

Characterizing At-Risk Students and Measuring Treatment Effects using Machine Learning Methods

Bevan Ian Smith

A thesis submitted to the Faculty of Engineering and the Built Environment, University of the Witwatersrand, Johannesburg, in fulfilment of the requirements for the degree of Doctor of Philosophy.

Johannesburg, June 2020

Declaration

I declare that this thesis is my own, unaided work, except where otherwise acknowledged. It is being submitted for the degree of Doctor of Philosophy in the University of the Witwatersrand, Johannesburg. It has not been submitted before for any degree or examination at any other university.

Signed this 3rd day of June 2020

A handwritten signature in black ink, appearing to read 'Bevan Ian Smith'. The signature is fluid and cursive, with the first name 'Bevan' being more legible than the last name 'Smith'.

Bevan Ian Smith

To Lanie.

“So then it is not of him who wills, nor of him who runs, but of God who shows mercy. ”

Romans 9:16

Acknowledgements

I would like to sincerely thank my supervisor Dr. Joke Bührmann for providing me with valuable advice and encouragement throughout this thesis. She was always generous with her time and challenged me to be more excellent in my work. Thank you very much Joke.

I am also extremely grateful to Dr. Charles Chimedza. I was always impressed with how he would be available for long discussions on my work and without fail, every time I saw him, I had a way to progress. Thank you very much Charles. This thesis was made possible by your generosity and knowledge.

Thank you to my wife Lanie, who has stood by me for many years as I fought my way through this PhD. I cannot thank her enough for her encouragement and most of all, her prayers, that the Lord would supply and strengthen me to keep going.

Special thanks is also due to Dr. Rodney Genga, who was the main factor for my receiving a sabbatical in order to accelerate this study. Without this sabbatical, the completion of this thesis would be severely delayed. Thank you Rodney.

Thank you to all my colleagues and friends who have participated in numerous discussions regarding my PhD and have taken interest in my journey.

Thank you to the Academic Information Systems Unit of the University of the Witwatersrand that graciously worked with me to extract all the data I needed for this study. They took time out of their busy schedules to accommodate me. Thank you very much.

Thank You to the Lord Jesus for supplying me richly during this study.

Abstract

Predicting and characterizing students that are at risk of failing is important because it would allow At-Risk students to be identified and recommended to various approaches that could improve their grades and help them pass. Not only is it important to predict At-Risk students, but also important to understand the features describing At-Risk students. By characterizing their features, the educator together with the student could identify which features could be changed in order to improve the probability of passing the course. In addition, At-Risk students could be recommended to various interventions such as supplementary tutorials, online videos or food programs and support groups. However, although these interventions are seen as beneficial, there is a great need to measure how effective these interventions are. Without measuring their effectiveness, i.e the treatment effects, there is no way to know if they are helping the student or not. Machine learning and statistical methods were used to address these objectives.

Preliminary work required a thorough preparation of the dataset to be used for training. A major component of the data preparation was the imputation of missing data using the random forest imputation method *missForest*. Further important preliminary work included detailed and thorough model selection procedures to identify the best performing models to be used for the main objectives. Lasso, random forest, adaboost, gradient boosting and neural networks were used in this study. Gradient boosting and neural network models performed the best in both classification and regression applications and were subsequently used in addressing the objectives. It is believed that these preliminary steps were vital to achieving better performance in addressing the objectives.

To better characterize At-Risk students, global and local model-agnostic machine learning methods were introduced. These carried out model explanations, or interpretations, describing which features are most important. Model-agnostic methods can be applied to any black-box model, for which model interpretation is difficult. Global (referring to the entire cohort of students) methods included permutation importances and partial dependence plots. In addition to these global agnostic methods, lasso β coefficients and tree-based feature importances were used for characterization. The results suggest that

using these global methods in combination provided confidence in describing overall features of At-Risk students as well as providing a way, via the partial dependence plots, to identify sub-regions in continuous features that could more accurately identify At-Risk students.

Although the global explanations provided insight into the overall characteristics, the results suggest that the local, or individual, methods may provide the most help to the student. Local methods used were the Local Interpretable Model-Agnostic Explanation (LIME) and Shapley Additive Explanations (SHAP) methods, developed by Ribeiro et al. (2016) and Lundberg et al. (2017), respectively. This provides explanations for individuals as to why they are At-Risk. The educator together with the student is then able to understand which features could be changed to change At-Risk status. To do this, counterfactual explanations were further generated. Counterfactual explanations, introduced for the first time to this field, perturbs a feature over a specific range and measures how the probability of being At-Risk changes as the feature is perturbed. This provided potential solutions to the student to see which feature can be changed to change the At-Risk status.

A machine learning method first introduced (as far as is known) by Beemer et al. (2017) and Beemer et al. (2018), was analyzed and extended. This method used machine learning to predict counterfactuals and treatment effects in observational studies. Traditional ways of measuring treatment effects in observational studies are to use propensity score matching (PSM) and this was the first known study validating this technique by comparing it with the PSM method. The results were found to be comparable. This method was further extended to be applied to not only compute global treatment effects but individual treatment effects. It was found that although global treatment effects were small, the individual treatment effects varied greatly, implying individual treatment effects provided more insight. A method was further developed to (1) predict an individual student's or sub-group of students' final grades and (2) predict their treatment effect. This would not only help predict the At-Risk status of the student, but also by how much an intervention would help them. Continuing with measuring treatment effects, the statistical method difference-in-differences (DID) was further used to measure treatment effects of supplementary online videos on performance. It was found that although global treatment effects were not-significant, there were significant local treatment effects.

Perhaps the main implication of this study was that using machine learning and statistical methods to analyze global predictions, characterizations and treatment effects are valuable, but focusing on smaller groups or individuals showed more potential for improving student success. Students are different and require different care. Machine learning is a powerful tool that can be used in conjunction with experienced caring educators to provide the best education to students.

Published Work

Aspects of this work have been submitted for publication as follows:

Bevan I. Smith, Charles Chimedza and Jacoba H. Bührmann. Random forest missing data imputation methods: Implications for predicting At-Risk students. *ISDA 2019: 19th International Conference on Intelligent Systems Design and Applications*, 2019. Accepted.

Bevan I. Smith, Charles Chimedza and Jacoba H. Bührmann. Individualized help for At-Risk students using model-agnostic explanations. *International Journal of Artificial Intelligence in Education*, 2019. Under review.

Bevan I. Smith, Charles Chimedza and Jacoba H. Bührmann. Global and individual treatment effects using machine learning methods. *International Journal of Artificial Intelligence in Education*, 2019. Revision submitted.

Bevan I. Smith, Charles Chimedza and Jacoba H. Bührmann. Treatment effects of online videos in a first year engineering mechanics course: Utilizing actual video watch time and difference-in-difference estimates. *African Journal of Research in Mathematics, Science and Technology Education*, 2019. Revision submitted.

Contents

Declaration	i
Acknowledgements	iii
Abstract	iv
Published Work	vi
Contents	vii
List of Figures	xviii
List of Tables	xxiii
List of Symbols	xxx
List of Acronyms	xxxii
1 Introduction	1
1.1 Predicting and Characterizing At-Risk Students	1
1.2 Providing Advice and Interventions	2
1.3 Measuring Treatment Effects of Interventions	2

1.4	Context	3
1.5	Contributions	3
1.6	Thesis Overview	4
2	Supervised Machine Learning	5
2.1	Introduction	5
2.2	Models Used	7
2.2.1	Lasso	7
2.2.1.1	Linear Regression	7
2.2.1.2	Logistic Regression	8
2.2.2	Random Forest	9
2.2.3	Boosting Methods	12
2.2.4	Deep Learning and Artificial Neural Networks	14
2.3	Model Evaluation Metrics	20
2.3.1	Regression	21
2.3.2	Classification	21
2.4	Model Assessment and Selection	27
2.5	Conclusion	28
3	Data Preparation	30
3.1	Introduction	30
3.2	Predictor and Target Features	30
3.2.1	Target Feature	31
3.2.2	Predictor Features	31

3.3	Normalizing and One-Hot Encoding	33
3.4	Literature on Missing Data	34
3.4.1	Types of Missingness	34
3.4.2	Determining Missingness Type	35
3.4.3	Methods for Addressing Missing Data	35
3.4.3.1	Listwise and Pairwise Deletion	35
3.4.3.2	Single Imputation - Mean, Mode, Median and Regression . .	36
3.4.3.3	Multiple Imputation	36
3.4.3.4	Random Forest Methods	38
3.4.3.5	Dealing with MNAR	38
3.4.4	Missing Values in Predicting Student Performance	39
3.5	Analysis of Missing Data	39
3.5.1	Original Data	39
3.5.2	MCAR Test	40
3.5.3	Determining MAR/MNAR	40
3.5.4	Missing Data Imputation	46
3.6	Correlated Features	48
3.7	Final Datasets	49
3.8	Conclusion	53
4	Characterizing At-Risk Students Using Model-Agnostic Methods - Model Selection	54
4.1	Introduction	54
4.2	Background	54

4.2.1	Predicting Student Performance	55
4.2.2	Imbalanced Data	58
4.3	Methodology	59
4.3.1	Model Hyperparameters	60
4.3.1.1	Lasso - Logistic regression with $L1$ regularization	60
4.3.1.2	Random Forest	61
4.3.1.3	Gradient Boosting	61
4.3.1.4	Adaboost	61
4.3.1.5	Artificial Neural Networks	62
4.3.2	Data Splitting	62
4.3.3	Addressing Imbalanced Data	63
4.3.4	cross-validation (CV) for Model Selection	64
4.3.5	Metrics Used	64
4.3.6	Model Performance	65
4.4	Best Models Overall	65
4.5	Best Model Per Dataset	67
4.6	Model Performance Over All Datasets	68
4.7	Overall Performance of Datasets	72
4.8	Conclusion	73
5	Characterizing At-Risk Students Using Model-Agnostic Methods - Model Explanations	75
5.1	Introduction	75
5.2	Background	76

5.2.1	Identifying Predictors of At-Risk Students	76
5.2.2	Interpretable machine learning (ML)	77
5.2.2.1	Interpretable Models	77
5.2.2.1.1	Regression Models	78
5.2.2.1.2	Feature Importances for Tree-Based Models	78
5.2.2.2	Global Model-Agnostic Techniques	78
5.2.2.2.1	Permutation Importances	79
5.2.2.2.2	Partial Dependence Plots	79
5.2.2.3	Local Model-Agnostic Techniques	80
5.2.2.3.1	LIME	80
5.2.2.3.2	SHAP	81
5.2.3	Counterfactual Explanations	82
5.3	Methodology	83
5.3.1	Global Techniques	84
5.3.2	Local Techniques	85
5.3.3	Counterfactual Models	85
5.4	Global Explanations	85
5.4.1	Feature Importances (FI)	85
5.4.2	Permutation Importances (PI)	88
5.4.3	Lasso β Coefficients	91
5.4.4	Partial Dependence Plots	93
5.5	Local Explanations	101

5.6	Counterfactual Explanations	103
5.7	Discussion	106
5.8	Conclusion and Future Work	108
6	Treatment Effects using Machine Learning Methods	110
6.1	Introduction	110
6.2	Theory of Treatment Effects	113
6.2.1	Randomized Control Studies	114
6.2.2	Observational Studies	114
6.2.3	Propensity Score Matching	115
6.3	Methodology	116
6.3.1	Overall Approach	117
6.3.2	Context	117
6.3.3	Control and Treatment Groups	117
6.3.4	Model Selection and Performance	118
6.3.5	Global Treatment Effects	118
6.3.6	Propensity Score Matching	118
6.3.7	Individual Treatment Effects	120
6.3.7.1	Subgroup Analysis	121
6.3.7.2	SHAP Values	121
6.4	Model Selection	122
6.5	Global Treatment Effects	122
6.5.1	Validation Using PSM	123

6.6	Subgroup Treatment Effects	124
6.6.1	Subgroup Treatment Effects on the Control (ITC)	124
6.6.1.1	Positive ITC Values	125
6.6.1.2	Negative ITC Values	126
6.6.2	Subgroup Treatment Effects on the Treated (ITT)	129
6.6.2.1	Positive ITT Values	129
6.6.2.2	Negative ITT Values	131
6.6.3	Individual Treatment Effects Using SHAP	132
6.7	Discussion	135
6.8	Conclusion	137
7	Treatment Effects of Online Videos Using Difference-in-Differences	138
7.1	Introduction	138
7.1.1	Main Contributions	139
7.2	Background	140
7.2.1	Online Videos	140
7.2.2	Measurement Techniques	141
7.2.2.1	Difference-in-Differences (DID)	142
7.2.2.2	Parallel Trends Assumption and Limitation	143
7.3	Methodology	144
7.3.1	Video Watch Time as a Predictor of Grades	144
7.3.1.1	Creating the Videos	145
7.3.1.2	Assessments	145

7.3.2	Difference-in-Differences (DID)	146
7.3.2.1	DID for Entire Cohort	146
7.3.2.2	DID Based on Initial Grades	146
7.3.2.3	Panel Data	146
7.3.3	Limitations and Assumptions	146
7.4	Data Analysis	147
7.4.1	Video Watch Time as a Predictor of Grades	147
7.4.2	Difference-in-Differences	151
7.4.2.1	DID Values for Entire Cohort	151
7.4.2.2	DID Values Based on Initial Grades	152
7.5	Discussion	154
7.6	Conclusion and Further Work	156
8	Conclusion	158
8.1	Introduction	158
8.1.1	Summary	158
8.1.2	Preliminary Work	159
8.2	Prediction, Characterization and Counterfactual Explanations	159
8.2.1	Limitations of ML for Prediction, Characterization and Counterfactual Explanations	160
8.3	Treatment Effects	161
8.3.1	MLTE Method	162
8.3.1.1	Limitations of the ML method on Treatment Effects	163
8.3.2	Difference-in-Differences	163

8.3.2.1	Limitations of the DID Method	164
8.4	Overall Implication of the Study	164
8.5	Generalizability and Future Work	165
8.6	Final Thoughts	166
Bibliography		168
A Model Selection and Performance		180
A.1	Imbalanced Data (IMB)	180
A.1.1	Logistic regression with $L1$ regularization - IMB	180
A.1.2	Random Forest - IMB	181
A.1.3	Gradient Boosting - IMB	182
A.1.4	Adaboost - IMB	184
A.1.5	Artificial Neural Networks - IMB	185
A.2	Random undersampling (RUS)	186
A.2.1	Logistic Regression with $L1$ regularization	186
A.2.2	Random Forest - RUS	188
A.2.3	Gradient Boosting - RUS	189
A.2.4	Adaboost - RUS	191
A.2.5	ANN - RUS	193
A.3	Resampled Data - Random oversampling (ROS)	195
A.3.1	Logistic Regression with $L1$ regularization - ROS	195
A.3.2	Random Forest - ROS	197
A.3.3	Gradient Boosting - ROS	199

A.3.4	Adaboost - ROS	201
A.3.5	ANN - ROS	203
A.4	Oversampling - SMOTE	205
A.4.1	Logistic regression with $L1$ regularization - SMOTE	205
A.4.2	Random Forest - SMOTE	207
A.4.3	Gradient Boosting - SMOTE	209
A.4.4	Adaboost - SMOTE	211
A.4.5	ANN - SMOTE	212
A.5	RUSBoost	214
B	Model Selection on Control and Treatment Groups	216
B.1	Lasso	216
B.1.1	Control Group	216
B.1.2	Treatment Group	217
B.2	Random Forest	217
B.2.1	Control Group	217
B.2.2	Treatment Group	218
B.3	Gradient Boosting	218
B.3.1	Control Group	218
B.3.2	Treatment Group	219
B.4	Adaboost	219
B.4.1	Control Group	219
B.4.2	Treatment Group	220

B.5	Artificial Neural Network	220
B.5.1	Control Group	220
B.5.2	Treatment Group	220

List of Figures

2.1	Single artificial neuron or perceptron.	15
2.2	Plots of activation functions.	17
2.3	Artificial neural network with a single hidden layer of multiple units and a single output unit [39].	17
2.4	Confusion matrix [53].	22
2.5	Receiver operating characteristic curve, ROC, according to James et al. (2013) [20]	24
2.6	Receiver operating characteristic curve, ROC, according to Hastie et al. (2009) [15]	24
3.1	Overall missing data analysis generated via <i>SPSS</i> (Version 25).	40
3.2	Features containing missing data generated via <i>SPSS</i> (Version 25).	41
3.3	Features containing missing data generated via <i>SPSS</i> (Version 25).	42
3.4	Summary of missing values after deleting total dropout observations. Generated by <i>SPSS</i> (Version 25).	43
3.5	MSE values for different imputed datasets.	47
3.6	Correlation heatmap.	48
3.7	Updated correlation heatmap with correlations between -0.7 and 0.7.	50
4.1	Accuracy boxplots of each model.	68

4.2	F1-score boxplots of each model.	69
4.3	AUC boxplots of each model.	70
4.4	κ score boxplots of each model.	71
4.5	Brier score boxplots of each model.	72
5.1	Mean feature importances for ROS-RF based on 100 fits to the test set. . . .	86
5.2	Mean feature importances for ROS-GB based on 100 fits to the test set. . . .	87
5.3	Mean permutation importances computed for ROS-RF based on 100 fits on the test set.	88
5.4	Mean permutation importances computed for ROS-GB based on 100 fits on the test set.	89
5.5	Mean permutation importances computed for IMB-ANN based on 100 fits on the test set.	90
5.6	β coefficients computed for IMB-Lasso.	92
5.7	PD-plots for March test PHYS1015 [Normalized] generated by ANN (top), GB (middle) and RF (bottom).	94
5.8	PD-plots for March.Test.MATH1014 [Normalized] generated by ANN (top), GB (middle) and RF (bottom).	95
5.9	PD-plots for March.Test.PHYS1014 generated by ANN (top), GB (middle) and RF (bottom).	97
5.10	PD-plots for Years.Since.Matric generated by ANN (top), GB (middle) and RF (bottom).	98
5.11	PD-plots for HS.Mathematics [Normalized] generated by ANN (top), GB (middle) and RF (bottom).	99
5.12	LIME results for first randomly selected At-Risk student.	101
5.13	SHAP results for first randomly selected At-Risk student.	102
5.14	LIME results for second randomly selected At-Risk student.	102

5.15	SHAP results for second randomly selected At-Risk student.	102
5.16	Counterfactual probabilities for March test PHYS1015	104
5.17	Counterfactual probabilities for March test MATH1014	104
5.18	Counterfactual probabilities for March test PHYS1014	105
5.19	Counterfactual probabilities for ADU.1st.Sem.No	105
5.20	Counterfactual probabilities for RESIDENCE.Yes	105
6.1	Individual treatment effects on the control (ITC) computed with the RF_{treat} model.	124
6.2	Top features for control group with positive ITC values.	126
6.3	Top features for control group with negative ITC values.	127
6.4	Individual treatment effects on the treated (ITT) computed with the $RF_{control}$ model.	129
6.5	Top features for treatment group with positive ITT values.	130
6.6	Top features for treatment group with negative ITT values.	131
6.7	SHAP results for control student with ITC value of 13.97%.	133
6.8	SHAP results for control student with ITC value of 11.0%.	133
6.9	SHAP results for control student with ITC value of -14.6%.	134
6.10	SHAP results for treated student with ITT value of 11.0%.	134
6.11	SHAP results for treated student with ITT value of 15.3%.	134
6.12	SHAP results for treated student with ITT value of -13.1%.	135
7.1	September Test 2017 grades of videos-group.	147
7.2	Histogram of September Test 2017. (a) Normal distribution of grades of videos group. (b) Right skewed distribution for the number of minutes of videos watched.	148

7.3	Grades of those watching videos for, top to bottom, October Test 2017, November Exam 2017 and Final-Year 2017.	149
7.4	Average grades for control and treatment groups for 2 nd semester 2017. . . .	151
A.1	Tuning of logistic regression parameter, C, based on AUC.	181
A.2	ROC- curve for lasso classifier.	181
A.3	ROC-curve for random forest classifier.	182
A.4	ROC-curve for gradient boosting classifier.	183
A.5	ROC-curve for adaboost classifier.	184
A.6	ROC-curve for AdaBoost classifier.	185
A.7	Tuning of logistic regression parameter, C, based on AUC.	186
A.8	ROC-curve for RUS Lasso regression classifier.	187
A.9	ROC-curve for RUS random forest classifier.	189
A.10	ROC-curve for RUS gradient boosting classifier.	190
A.11	ROC-curve for RUS adaboost classifier.	192
A.12	ROC-curve for RUS ANN classifier.	194
A.13	Tuning of logistic regression parameter, C, based on AUC.	195
A.14	ROC-curve for ROS Lasso regression classifier.	196
A.15	ROC-curve for ROS random forest classifier.	198
A.16	ROC-curve for ROS gradient boosting classifier.	200
A.17	ROC-curve for ROS adaboost classifier.	202
A.18	ROC-curve for ROS ANN classifier.	204
A.19	Tuning of logistic regression parameter, C, based on AUC.	205
A.20	ROC-curve for SMOTE lasso classifier.	206

A.21 ROC-curve for SMOTE random forest classifier.	208
A.22 ROC-curve for SMOTE gradient boosting classifier.	210
A.23 ROC-curve for SMOTE adaboost classifier.	212
A.24 ROC-curve for SMOTE ANN classifier.	213
A.25 ROC-curve for RUSBoost classifier.	214
A.26 Feature importances for RUSBoost classifier predicting At-Risk students. . .	215
B.1 Mean absolute error for lasso model tuned on the control dataset.	216
B.2 Mean absolute error for lasso model tuned on the treatment dataset.	217

List of Tables

2.1	Interpretation of Cohen's κ scores [57].	26
3.1	Features containing missing data generated via <i>SPSS</i> (Version 25).	40
3.2	Summary of features with missing values after deleting total dropout observations. Generated by <i>SPSS</i> (Version 25).	43
3.3	mean squared error (MSE) for Entire and Reduced dataset. Standard deviations are shown in brackets.	45
3.4	MSE for various imputed datasets with standard deviation in brackets, where applicable.	47
3.5	Correlated features removed from the dataset	51
3.6	Statistics of different datasets used in this study.	52
4.1	ML in student performance. The models with an asterisk indicate best performer in each study.	58
4.2	Random forest classifier hyperparameter ranges for model selection.	61
4.3	Gradient boosting parameter ranges for model selection.	61
4.4	AdaBoost parameter ranges for tuning.	62
4.5	ANN parameter ranges for model selection.	62
4.6	Class frequencies for training and testing datasets. Stratification preserves the class imbalance.	63

4.7	Class frequency.	63
4.8	Metrics for all the ML models trained and tested on all the datasets.	66
4.9	Summary of accuracy boxplots. Best mean values indicated by an asterisk.	68
4.10	Summary of F1-score boxplots.	69
4.11	Summary of AUC boxplots.	70
4.12	Summary of κ score boxplots.	71
4.13	Summary of Brier score boxplots.	72
4.14	Average scores for all datasets.	73
5.1	Top six features for each computation.	91
5.2	Impact of top features on being At-Risk using PD-plots.	100
6.1	Treatment effects estimated by Beemer et al. (2018) [9].	112
6.2	Statistics of different datasets used in this study.	119
6.3	Sample data of random student.	120
6.4	MAE for each model trained on the Control Group.	122
6.5	MAE for each model trained on the Treatment Group.	122
6.6	Treatment effects computed from the RF, GB and lasso models.	123
6.7	Comparison of ML technique with PSM for estimating treatment effects. p -value for PSM method in brackets.	124
6.8	Statistics for average treatment effects on control group, ATC, computed with the RF_{treat} model	125
6.9	Statistics of top features for control group with positive ITC values ($n = 175$).	126
6.10	Statistics of top features for control group with positive ITC values above 10% ($n = 54$).	127

6.11	Statistics for top features of control group with negative ITC values.	128
6.12	Statistics for top features of control group with negative ITC values (n = 71).	128
6.13	Statistics for average treatment effects on treated, ATT, computed with the RF _{control} model	129
6.14	Statistics for top features of treatment group with positive ITT values (n = 263).	130
6.15	Statistics for top features of treatment group with ITT values above 10% (n = 73).	131
6.16	Statistics for top features of treatment group with negative ITT values (n = 231).	132
6.17	Statistics for top features of treatment group with negative ITT values below -10% (n = 54).	132
7.1	Comparison of means between video and no-video groups for 2017. The T-test column shows the difference between the videos-group grades and the no-videos group.	150
7.2	Mid-Year 2017 and Final Year 2017 grades for videos and no-videos groups.	152
7.3	Changes and DID estimates based on failing or passing at Mid-Year for the time span from Mid-Year 2017 to Final Year 2017	153
7.4	Changes and DID estimates based on failing or passing at mid-year for the time span from Mid-Year 2017 to November Exam 2017.	153
7.5	DID estimates based on initial Mid-Year grades for the time span from Mid- Year 2017 to Final-Year 2017.	153
7.6	DID estimates based on initial Mid-Year grades for the time span from Mid- Year 2017 to November Exam 2017.	154
A.1	Confusion matrix for <i>lasso</i> classification.	180
A.2	Lasso classification metrics on test set.	181

A.3	Random forest classifier optimal hyperparameters.	182
A.4	Confusion matrix for random forest classification on imbalanced dataset. . . .	182
A.5	Random forest classification metrics on test set.	182
A.6	Gradient boosting optimal parameters.	183
A.7	Confusion matrix for gradient boosting classification on the imbalanced dataset.	183
A.8	Gradient boosting classification metrics on test set.	183
A.9	Optimal Adaboost parameters.	184
A.10	Confusion matrix for adaboost classification on imbalanced dataset.	184
A.11	Adaboost classification metrics on test set.	184
A.12	ANN parameter ranges for model selection on imbalanced data.	185
A.13	ANN classification metrics on test set.	185
A.14	Confusion matrix for ANN classification on imbalanced dataset.	185
A.15	Lasso classification metrics on RUS training set.	186
A.16	Confusion matrix for RUS lasso classification.	187
A.17	RUS lasso regression classification metrics on test set.	187
A.18	Random forest classifier optimal hyperparameters.	188
A.19	Random forest classification metrics on RUS training set.	188
A.20	Confusion matrix for RUS random forest classification.	188
A.21	RUS random forest classification metrics on test set.	188
A.22	Gradient boosting optimal parameters.	189
A.23	Gradient boosting classification metrics on RUS training set.	189
A.24	Confusion matrix for RUS gradient boosting classification.	189

A.25 RUS gradient boosting classification metrics on test set.	190
A.26 Gradient boosting optimal parameters.	191
A.27 AdaBoost classification metrics on RUS training set.	191
A.28 Confusion matrix for RUS AdaBoost classification.	191
A.29 RUS Adaboost classification metrics on test set.	191
A.30 Optimal ANN parameters for RUS dataset	193
A.31 ANN classification metrics on RUS training set.	193
A.32 Confusion matrix for RUS AdaBoost classification.	193
A.33 RUS ANN classification metrics on test set.	193
A.34 Lasso regression classification metrics on ROS training set.	195
A.35 Confusion matrix for ROS lasso classification.	196
A.36 ROS lasso classification metrics on test set.	196
A.37 Random forest classifier optimal hyperparameters. - ROS	197
A.38 Random forest classification metrics on ROS training set.	197
A.39 Confusion matrix for ROS lasso classification.	197
A.40 ROS random forest classification metrics on test set.	197
A.41 Gradient boosting ROS optimal parameters.	199
A.42 Gradient boosting classification metrics on ROS training set.	199
A.43 Confusion matrix for ROS gradient boosting classification.	199
A.44 ROS gradient boosting classification metrics on test set.	199
A.45 Gradient boosting ROS optimal parameters.	201
A.46 AdaBoost classification metrics on ROS training set.	201

A.47 Confusion matrix for ROS AdaBoost classification.	201
A.48 ROS Adaboost classification metrics on test set.	201
A.49 Optimal ANN parameters for ROS dataset	203
A.50 ANN classification metrics on ROS training set.	203
A.51 Confusion matrix for ROS ANN classification.	203
A.52 ROS ANN classification metrics on test set.	203
A.53 Lasso regression classification metrics on SMOTE training set.	205
A.54 Confusion matrix for SMOTE lasso regression classification.	205
A.55 SMOTE lasso classification metrics on test set.	206
A.56 Random forest classifier optimal hyperparameters for SMOTE dataset.	207
A.57 Random forest classification metrics on SMOTE training set.	207
A.58 Confusion matrix for SMOTE lasso regression classification.	207
A.59 SMOTE random forest classification metrics on test set.	207
A.60 Gradient boosting optimal parameters, SMOTE.	209
A.61 Gradient boosting classification metrics on SMOTE training set.	209
A.62 Confusion matrix for SMOTE gradient boosting classification.	209
A.63 SMOTE gradient boosting classification metrics on test set.	209
A.64 Adaboost optimal parameters, SMOTE.	211
A.65 AdaBoost classification metrics on SMOTE training set.	211
A.66 Confusion matrix for SMOTE AdaBoost classification.	211
A.67 SMOTE Adaboost classification metrics on test set.	211
A.68 Optimal ANN parameters for SMOTE dataset	212

A.69 ANN classification metrics on SMOTE training set.	212
A.70 SMOTE ANN classification metrics on test set.	212
A.71 Confusion matrix for SMOTE ANN classification.	213
A.72 RUSBoosting optimal parameters.	214
A.73 RUSBoost classification metrics on training set.	214
A.74 Confusion matrix for RUSBoost classification.	214
A.75 RUSboost classification metrics on test set.	215
B.1 Random forest regression hyperparameter ranges.	217
B.2 Random forest optimal hyperparameters for the Control Group.	218
B.3 Random forest optimal hyperparameters for the Treatment Group.	218
B.4 Gradient boosting hyperparameter ranges for tuning.	218
B.5 Optimal gradient boosting hyperparameters for Entire Group.	219
B.6 Optimal gradient boosting hyperparameters for Entire Group.	219
B.7 Adaboost hyperparameter ranges for tuning.	219
B.8 Optimal adaboost hyperparameters for the Control Group.	219
B.9 Optimal adaboost hyperparameters for Treatment Group.	220
B.10 ANN parameter ranges for model selection.	220
B.11 Optimal ANN parameters for Control Group	221
B.12 Optimal ANN parameters for Treatment Group	221

List of Symbols

The units of quantities defined by a symbol are indicated in square brackets following the description of the symbol. Quantities with no indicated units may be assumed to be dimensionless.

Ω Constraint on explanation model.

β Regression coefficient.

ϵ Learning rate.

κ Cohen's kappa.

$\mathcal{L}$ Cost function.

ν Learning rate.

ϕ Attribution effect.

π Proximity of synthetic instances.

ξ Explanation model.

λ Regularization parameter.

μ Mean value.

$\bar{y}$ Mean target.

σ Standard deviation.

$\mathbf{w}$ Weights in neural network.

$\mathcal{C}$ Cost function.

$\mathcal{L}$ Likelihood.

P Probability.

$\hat{y}$ Predicted target.

b Bias.

g Explanation model.

z Z-score.

c Number of classes.

E Entropy.

G Gini impurity.

I Impurity.

N Number of observations.

p Predictor.

p Probability.

r Residual.

S Subset of the data.

T Transpose.

X Input features.

x Input feature.

y Target feature.

List of Acronyms

adaboost adaptive boosting.

ADU Academic Development Unit.

ANN artificial neural network.

AUC area under the curve.

CV cross-validation.

DT decision tree.

FI feature importances.

FN false negatives.

FP false positives.

GB gradient boosting.

KNN k-nearest neighbor.

LIME Locally Interpretable Model-Agnostic Explanation.

MAR Missing at random.

MCAR Missing completely at random.

MI Multiple imputation.

MICE multivariate imputation by chained equations.

ML machine learning.

MLP multi-layer perceptron.

MLTE ML treatment effect.

MNAR Missing not at random.

MSE mean squared error.

PD-plot partial dependence plots.

PI permutation importances.

PSM propensity score matching.

RCT randomized controlled trial.

ReLU rectified linear unit.

RF Random forest.

ROC receiver operating characteristic curve.

ROS random oversampling.

RSS residual sum of squares.

RUS random undersampling.

SGD Stochastic gradient descent.

SHAP Shapley Additive Explanations.

SMOTE synthetic minority oversampling technique.

TN true negatives.

TNR true negative rate.

TP True positives.

TPR true positive rate.

Chapter 1 Introduction

A major challenge facing higher education is the failure and drop-out of students. Predicting student performance can potentially identify and characterize students at risk of failure, i.e. At-Risk students, who could then be recommended for various interventions aimed at improving pass rates. With regards to cost, in South Africa for example, one year at a university can cost up to R120,000 (approximately US\$8,200 in 2020) and improving graduation time by even a single semester could significantly reduce cost to the student or tax-payer. Further benefits include producing engineering graduates that can benefit both the economy and the graduate. Furthermore, it is crucial to measure how effective the interventions are in order to ascertain whether they are beneficial or not.

The objectives of this study are therefore to find better ways of:

1. Predicting and characterizing At-Risk students.
2. Providing advice or interventions for At-Risk students to help them switch to not being At-Risk.
3. Measuring the effectiveness of the interventions.

In this study machine learning (ML) methods primarily, as well as statistical methods, were proposed to address these important objectives.

1.1 Predicting and Characterizing At-Risk Students

ML is increasingly being used to train models that predict student performance [1, 2, 3, 4]. Although predictions are indeed valuable, characterizing, or describing, At-Risk students is also increasingly needed. There is a need to not only know who is At-Risk, but also to understand why they are At-Risk in order to gain insight into how they might be helped

[5]. To address this objective, global and local model-agnostic ML methods [6, 7, 8] were proposed. Global refers to the entire group of At-Risk students and local, to individual At-Risk students. The need to cater to individuals in various fields is growing [7, 8], and it is believed that there is the need to cater to individual students.

1.2 Providing Advice and Interventions

Characterizing of At-Risk students would provide the means for explaining both global and local reasons for failure that could be used by the students and educators to find potential ways to pass the course. These potential ways include changing certain features using counterfactual explanations and providing interventions. Counterfactual explanations [5] were further proposed that present potential ways individual students could change At-Risk status. These methods have not been used in this field thus far. However, care needs to be taken here to focus only on explanations, i.e. features, that can be changed and to not discriminate against any students based on any demographic characteristics. The use of counterfactual explanations is elaborated on in this study.

1.3 Measuring Treatment Effects of Interventions

Predicting and characterizing At-Risk students is the first step, followed by recommending various solutions and interventions that could help the students pass. However, simply providing an intervention does not mean that it is effective. There is the need to measure how effective these academic interventions are, i.e. measure treatment effects. Measuring treatment effects is challenging since the interventions in this study and similar studies in the literature [4, 9] are typically observational studies (not randomized control). That is, students that attend these interventions generally self-select to attend, making inferring causal (treatment) effects very difficult [10] due to potential selection bias. Measuring treatment effects in observational studies has been carried out in various fields by primarily using propensity score analysis [11, 12, 13]. Others [14] have studied causal effects using ML in fields such as economics, but no substantial work has used ML to study the effects of academic interventions in higher education. This study proposed the use of a ML method that estimates treatment effects, first presented by Beemer et al. (2017) [4] and Beemer et al. (2018) [9]. Their inspiring work has opened up a new way of measuring treatment effects by trainingⁱ ML models on both control and treatment (intervention) groups and using these models to estimate counterfactuals and treatment effects. However, there is the need to analyse the

ⁱTraining refers to the process whereby the algorithm learns the input-output relationship in the data.

effectiveness and validity of this ML method and further extend it, which this study sets out to do. This study refers to this method as the ML treatment effect (MLTE) method.

Interventions include various academic and non-academic activities such as face-to-face extra tutorials, online videos, food programmes and psychological assistance. In this study, academic interventions include both face-to-face and online video interventions. As indicated, the main focus is the use of ML to measure treatment effects. This study however also presents the use of a statistical technique called difference-in-differences to measure treatment effects in online videos.

1.4 Context

The context of this study was a first year engineering mechanics course, PHYS1015, for which extra face-to-face tutorials and online videos were provided as interventions for any students that wished to utilize these resources. Two cohorts of students were studied, one in the first semester of 2016, and the other in the second semester of 2017. In general, all students attended four hours of lectures per week and two hours of supplementary face-to-face tutorials were provided per week as the intervention. Furthermore, all students had access to supplementary online videos on the course content.

1.5 Contributions

This thesis makes the following contributions to the literature on predicting student performance:

1. Before any predictions or characterizations were performed, thorough missing data analysis and imputation was carried out in Chapter 3. This is the first study found in predicting student performance that performs such detailed analysis of missing data and implemented multivariate imputation by chained equations (MICE) and random forest ML methods to impute more accurate missing values. The results suggest that this imputation step provided better subsequent predictions and characterizations of At-Risk students.
2. In Chapter 5 global model-agnostic interpretable ML methods that identify or characterize At-Risk students were introduced. These methods identify predictors using any black-box ML model. These methods include permutation importances and partial-dependence plots.

3. In Chapter 5 local model-agnostic techniques, Locally Interpretable Model-Agnostic Explanation (LIME) developed by Ribeiro et al. (2016) [7] and Shapley Additive Explanations (SHAP), developed by Lundberg et al. (2017) [8], were introduced to identify characteristics of individual At-Risk students. These local models are able to provide individual explanations of why they are At-Risk.
4. Counterfactual explanations were further presented in Chapter 5 which provide potential ways that an individual student could switch from being At-Risk to not being At-Risk.
5. In Chapter 6 global and individual treatment effects were computed using the MLTE method. This is the first known study to validate this method with the traditional PSM technique.
6. Chapter 6 extends the MLTE method by computing subgroup and individual treatment effects.
7. Chapter 6 combines the MLTE method and local model-agnostic methods to develop a method for (1) predicting the final grade of students and (2) predicting individual treatment effects. This combining of methods allows for providing advice on whether individual students or subgroups would benefit or not from the interventions.
8. The statistical technique difference-in-differences (DID) was introduced in Chapter 7 for estimating the treatment effects of supplementary online videos on performance.

1.6 Thesis Overview

The subsequent sections of this thesis are summarized as follows. Chapter 2 presents a thorough overview of relevant ML concepts in order to lay a foundation for the remaining chapters. Chapter 3 presents the entire process of preparing the dataset for model training. In Chapters 4 and 5, At-Risk students were characterized using global and local ML methods. In Chapter 6 ML methods were used to measure treatment effects of interventions and Chapter 7 uses the statistical technique DID to measure treatment effects of online videos. Finally, Chapter 8 concludes the study and presents further work.

Chapter 2 Supervised Machine Learning

2.1 Introduction

ML forms the basis for this study. This chapter therefore presents a thorough overview of the ML models and concepts relevant to this research. Each subsequent chapter, Chapters 3 - 7, then presents reviews of the literature and/or theory related to that specific chapter. Furthermore, this thesis does not present a dedicated methodology chapter since each chapter presents its' own methodology.

ML is the ability of a computer program to learn from data and improve from experience [15]. In general, learning can be defined as the ability to remember, adapt and generalize [16]. For example, when humans learn, it means that they recognize a situation, remember what they did the last time they were in that situation and what the outcome was. If it was beneficial they might perform the same action or try something else if not beneficial. Generalizing means they use what was learned in certain situations to apply to new ones. Similarly, for machines to learn means the computer is able to adapt and generalize to new situations. Through experience, it is able to improve its performance, i.e. generate predictions or find trends [17]. Computers have generally been used to solve problems by being programmed step-by-step to perform tasks that are useful only under limited circumstances. However, in the field of, say, image recognition, it is impossible to program every single step for the computer to recognize every future image, i.e. to generalize. Therefore in ML, the machine learner, or model, is presented with data and it develops its own method of representing the knowledge [18].

ML comprises a few categories, namely supervised, unsupervised, semi-supervised and reinforcement learning [16, 18]. In supervised learning, the model is presented with data, called training data, that includes both input features and output labels/features, and the model learns, i.e. is trained, to associate the input with the output [19]. The presence of output features supervise (guide) the training process [15]. In unsupervised learning, no output labels are available, and the goal is to describe how the features are grouped or clustered together

[15]. Semi-supervised learning involves learning with examples that have both labeled and unlabeled data. Semi-supervised learning falls between supervised and unsupervised learning and often combines a small number of labeled data with a larger number of unlabeled. This is the case when there is a high cost to labelling data and therefore only a small number can be labeled. Reinforcement learning is concerned with how the model, or agent, interacts with the environment and receives a reward based on its' actions. The environment provides information on how good the agent's strategy is through a reward function [16]. *The research in this thesis makes use of supervised learning methods.*

Learning can be carried out on both regression (continuous) or classification problems. Regression is concerned with predicting a continuous output such as house price or grades of a student. Classification is concerned with predicting a class output, such as pass or fail, or whether a patient has cancer or not.

Two major uses of ML are to carry out prediction and inference (explanation) [20]. A model can be used to predict the outcome of a new instance. For example, a model can be trained on historical data of cancer patients, and then be used to predict a new patient's probability of cancer. Inference is not primarily concerned with prediction, but with understanding, or explaining, how features affect the output [20]. This is crucial to this study because it provides ways for characterizing At-Risk students. Model explainability is an aspect of ML that is becoming more important of late [6, 7, 8]. It can be performed based on traditional interpretable methods such as linear/logistic regression or tree-based models that can explain how the features affect the output. However, a new field of model-agnostic methods is being developed that is able to take any black-box ML model, such as neural networks, and infer, or explain, the impact of the features [6, 7, 8]. This new field allows this study to use multiple models and inference methods to both predict and characterize At-Risk students. Because explainability is a major component of this study, it is described in detail in Chapter 5 where it is primarily used.

This research utilizes supervised ML in the following ways:

- To impute (predict) missing data (Chapter 3).
- To characterize At-Risk students (Chapters 4 and 5), using prediction and inference.
- To measure treatment effects of interventions (Chapter 6), using prediction and inference.

The following section describes in detail the ML models used in this study which form the basis for *all* prediction and/or inference applications in subsequent chapters. Specific details of the ML applications are described in each chapter of this study.

2.2 Models Used

The literature on using ML to predict student performance, as well as other literature, shows that in general, three types of models, *random forest*, *boosting* and *neural networks* are the best performers overall [21, 22, 23]. These ML methods are therefore used in this study and their theory is described here in detail. In general, the models were first trained using model selection procedures (see Section 2.4) and *then* various inference methods (Chapter 5) were applied to the trained models. In addition to these methods, *lasso* was included since one of its' main uses is to carry out feature selection, i.e. find the features explaining the outcome, a major part of this study. The first method looked at next is lasso (Section 2.2.1), followed by the tree-based methods (Sections 2.2.2 and 2.2.3), and finally neural networks (Section 2.2.4).

2.2.1 Lasso

Lasso is a shrinkage method that fits a model with all available features, then constrains (regularizes) the feature coefficients (β 's) so that some coefficients are eliminated [15, 20]. This is very effective in (1) reducing model overfittingⁱ, (2) dealing with correlated features and (3) carrying out feature selection [20]. In the case of highly correlated features, features can behave erratically and the coefficients can become large. By regularizing the feature space, only one of the correlated features would remain and the feature coefficients would stabilize. Feature selection is the primary reason for selecting this method. In the related literature, lasso has been used to predict student performance and treatment effects [4, 9].

2.2.1.1 Linear Regression

Lasso can be applied to both regression and classification problems. In standard linear regression, the β coefficients are calculated by minimizing the residual sum of squares (RSS) shown in Equation 2.1. Shrinkage methods add a penalty term to penalize features with large coefficients. For lasso *regression*, the quantity in Equation 2.2 is minimized, where the penalty is $\lambda \sum_{j=1}^p |\beta_j|$ and p is the total number of predictors (features) [20]. The terms y_i and x_{ij} refer to outputs and inputs of individual observations. Lasso is also known as L_1 regularization since it penalizes the sum of the absolute value of the coefficients, i.e. the L_1 norm of coefficients. As λ is varied, the coefficients vary as well. If λ is zero then it is the equivalent of ordinary least squares and as λ gets larger, more coefficients reduce to zero.

ⁱOverfitting refers to the model fitting the errors or noise too closely [20]. The problem with overfitting is that the model does not generalize well to new observations.

Lasso yields sparse models, i.e. models that include only a subset of the original features [20]. This is a way of reducing over-fitting, eliminating unnecessary features in the model (such as correlated features), and thereby performing feature selection. Optimal λ values are obtained via CV discussed in Section 2.4.

$$RSS = \sum_{i=1}^p (y_i - (\beta_0 + \sum_{j=1}^p \beta_j x_{ij}))^2 \quad (2.1)$$

$$\sum_{i=1}^N (y_i - (\beta_0 + \sum_{j=1}^p \beta_j x_{ij}))^2 + \lambda \sum_{j=1}^p |\beta_j| \quad (2.2)$$

2.2.1.2 Logistic Regression

Lasso is also applied to classification problems by regularizing the logistic regression function. L_1 regularized logistic regression has been used as an inference tool, where the goal is to understand how the features explain the outcome [15]. The logistic function, shown in Equation 2.3, is a sigmoid function that takes in input values, X , and outputs a probability, P , between zero and one, that a given observation will be in a certain class. If the probability is above a certain threshold value, it is considered part of one class, otherwise it is considered part of the other class. The vector β describes the coefficients being estimated.

$$P(X) = \frac{e^{\beta^T X}}{1 + e^{\beta^T X}} \quad (2.3)$$

As with linear regression, the aim is to estimate the coefficients β in order to make predictions. To do this, the non-linear sigmoid function is transformed into a linear function by taking the logarithm of the odds, shown in Equation 2.4. Odds refers to the ratio of the probability of an event occurring to the probability of an event not occurring. It can be seen that the right-hand side of Equation 2.4 is in the familiar form of the regression equation. The probability of an event occurring has now been transformed into the $\log(\text{odds})$ of the event occurring.

$$\log(\text{odds}) = \log\left(\frac{P(X)}{1 - P(X)}\right) = \beta^T X \quad (2.4)$$

In linear regression models, the coefficients are estimated by minimizing the error (for example MSE). However, in logistic regression, coefficients are estimated by maximizing the likelihood, L , in Equation 2.5, where y_i refers to the output value of 0 or 1 (for example fail or pass) and s refers to the subset of observations.

$$L(\beta) = \prod_{s \text{ in } y_i=1} P(x_i) \cdot \prod_{s \text{ in } p_i=0} (1 - P(x_i)) \quad (2.5)$$

The first step is to assume values for the β coefficients and compute the log(odds), as per Equation 2.4. Thereafter the probability is computed as per Equation 2.3. Once the probability for each example in the training data has been computed, the likelihood (Equation 2.5) or the log-likelihood (Equation 2.6) is computed. The coefficients are repeatedly changed until a maximum likelihood is calculated. This computation is therefore an optimization problem.

$$\ell(\beta) = \sum_{i=1}^N y_i(\beta^T x_i) - \log(1 + e^{\beta^T x_i}) \quad (2.6)$$

β is then chosen that maximizes Equation 2.6, i.e.

$$\beta = \underset{\beta}{\operatorname{argmax}} \ell(\beta) \quad (2.7)$$

It is important to note that once the coefficients have been estimated, they are related not to the probability directly but to the log-odds of the probability. For example, in linear regression models, holding all other features constant, changing the first feature x_1 by one unit changes the output y_1 by the value of the coefficient β_1 . However, for logistic regression, a one unit change in x_1 changes the *log-odds* by β_1 (see Equation 2.4), or it changes the *odds* of the output by e^{β_1} . The coefficients in linear and logistic regression therefore have slightly different interpretations. Nevertheless, a positive coefficient means that increasing a feature x will increase the probability of the outcome and a negative coefficient will decrease the probability of the outcome [20].

Finally, lasso is applied to logistic regression by maximizing the regularized log-likelihood function, shown in Equation 2.8.

$$\ell(\beta) - \lambda \sum_{j=1}^p |\beta_j| \quad (2.8)$$

2.2.2 Random Forest

Random forest (RF) is an ensemble methodⁱⁱ that is a combination of multiple decision trees. A decision tree (DT) is used in both regression and classification problems by recursively

ⁱⁱA method that builds a model by combining the strengths of a number of simple base models [15].

splitting a data set into smaller subsets until it obtains the target outcome [24]. The aim is to split on a feature that will most reduce the uncertainty in prediction. Since there are multiple features in a dataset, a method is required for measuring how impure a subset is. Various methods exist for measuring the uncertainty or impurity of a prediction based on a feature. For *classification* problems these methods include entropy, misclassification error and Gini index [15, 20]. Impurity measured using entropy is given in Equation 2.9,

$$I_E(S) = - \sum_{i=1}^c P_i \log_2 P_i, \quad (2.9)$$

where S is a subset of the data, P_i is the probability (fraction) of a class i , and c is the number of classes in S . The higher the entropy (1 is a maximum, 0 is minimum) the higher the uncertainty (impurity). Misclassification error rate I_C , given by Equation 2.10, measures the fraction of observations in the subset that are not in the majority outcome [25].

$$I_C = 1 - \max(P_i) \quad (2.10)$$

The third method is the Gini index given by Equation 2.11. Similarly to entropy, higher values indicate higher impurity (more mixed outcome).

$$I_G(S) = - \sum_{i=1}^c P_i (1 - P_i), \quad (2.11)$$

Entropy and the Gini index are better suited for growing trees due to being differentiable and therefore more suited for optimization problems [15] and being more sensitive to probabilities in the nodes [20]. However, entropy is relatively slow to compute due to the $\log_2$ [25].

The above equations represent methods for measuring impurity at a node (subset). Deciding on which feature to split requires computing the amount of impurity reduction after splitting, known as information gain [26]. The greater the information gain, the greater the decrease in impurity. Features that result in the greatest information gain are chosen on which to split. The above therefore describes splitting for classification problems.

Regression problems however, deal with continuous features and therefore require a different method for measuring and reducing impurity. This method involves setting thresholds that create regions (subsets) in the data. The mean value of a region is assigned to all observations in that region. For example, the age of a student is a continuous feature and an age threshold could be set where above a specified age, students are assigned one mean value, and below

it, another mean value. For regression trees, the threshold value is chosen by minimizing the total MSE [25, 26]. If a continuous feature is therefore selected, a threshold value is chosen, regions are created and MSE values for *each region* are computed based on Equation 2.12, where for a given region S , N_S is the number of observations, y_i is the output for a single observation and $\bar{y}_S$ is the mean value.

$$\frac{1}{N_S} \sum_{i=1}^{N_S} (y_i - \bar{y}_S)^2, \quad (2.12)$$

For a specific threshold value, all the regions' MSE values are added up to obtain the total MSE. The threshold value is then repeatedly changed until a minimum total MSE is obtained.

The advantages of a decision tree is that:

- It is easy to understand and interpret [20].
- It can easily handle qualitative predictors without needing to create dummy variables [20].

The disadvantages [20] are that:

- The level of predictive accuracy is not as good as other techniques.
- It can only split data along axes-directions. For example, it cannot draw a diagonal classification line, only classifiers that are parallel to the axes.
- It is greedy and therefore does not select the globally optimal tree.
- Overfitting often occurs and therefore does not predict out-of-sample data well.
- Potential exists for high variance in output for small changes in input data.

RF however rarely overfits, reduces variance and can yield more accurate predictions than decision trees [27, 28]. Developed by Leo Breiman in 2001 [28], RF is an ensemble classifier that comprises multiple decision trees where each is created only from a subset of the original feature space. That is, each tree is generated from only a subset of the samples (observations) *and* features and therefore the ensemble of trees are independent of, or decorrelated [20] from, each other. Once the number of trees has been trained, predictions of new observations are made by voting for the most popular class in classification or averaging the outcome in regression. The steps for generating a RF [20] is:

1. Select bootstrap samplesⁱⁱⁱ from the training set.

ⁱⁱⁱRandom sampling with replacement.

2. Train a decision tree on each sample.
3. Randomly select only m features out of the total features p , for splitting.
4. Carry out predictions in each tree. For regression calculate the mean and for classification choose the outcome with the most votes.

Various hyperparameters affect the performance of the random forest model [29] and in general these hyperparameters need to be tuned to find the optimal values. The literature shows that the main parameters to consider are the *number of trees* in the forest and the *number of features* in each split [30]. In general, the higher the number of trees, the better the performance [29, 30]. However above a certain number, there is not necessarily significant improvement in performance. For example, some have shown that above 128 trees, there is no significant improvement in performance [31]. In terms of the number of features m for each split, in regression problems, some have suggested $m = p/3$ [28]; while others suggest $m = p$ performs best [32]. For classification problems, $m = \sqrt{p}$ has been shown to perform well [20, 30, 33] and it has been suggested to begin with $m = \sqrt{p}$ and then select values twice as large and half as large as $\sqrt{p}$ when selecting optimal hyperparameters [33].

One of the aims of random forest is to decorrelate the various trees used and therefore in terms of the *sampling size* parameter, smaller samples produces more diverse trees which results in lower correlation [29]. The standard choice is to sample the same number as the number of observations with replacement [29]. However, others found that sampling less than the total number of observations produced better results [34]. There are also options to sample with replacement (bootstrap) or without replacement. A parameter called *minimum samples for splitting*, which indicates the number of observations for splitting at a node, can also be adjusted. In the Python ML library *Scikit-learn*^{iv} the default is two. Another parameter is the minimum number of observations in a terminal node, or *minimum samples in a leaf*. The lower this value, the deeper each tree will be grown. Default values are generally around one. The best parameters are found by varying them until an optimal criteria is obtained, such as minimizing the MSE or maximizing the area under the curve (discussed in Section 2.3).

2.2.3 Boosting Methods

Whereas RF is trained on multiple trees simultaneously (each based on a bootstrap sample), boosting grows trees sequentially; each tree being grown from information obtained from the previous tree [20]. The idea is to iteratively improve a weak base-learner (model) such as a decision tree [35]. There are commonly two types of boosting methods, adaptive boosting

^{iv}A ML library developed for the Python programming language and the primary library used in this study.

(adaboost) [36] and gradient boosting (GB) [37, 38]. Adaboost grows an initial tree and carries out predictions. Then a process of re-weighting is carried out. Incorrectly predicted observations are weighted more highly than the correctly predicted observations, and a new tree is trained on the new re-weighted observations and added to the previous tree to update the model. The new tree therefore focuses more on the observations that were more difficult to initially predict [35]. This is an ensemble learning technique that combines a number of weak models into a superior models. In *Scikit-learn*, the tuning parameters for adaboost are the number of estimators (trees) and the learning rate.

Instead of training new trees on re-weighted observations, GB trains new trees on the *error residuals* from the previous tree's predictions. The procedure for GB is described as follows: Consider a training set comprising of input features and targets, (x_i, y_i) , for $i = 1, \dots, N$, where N is the number of observations in the training set. Further consider a differentiable cost function $C(y, f(x))$, for example the RSS computed between observed and predicted values, as shown in Equation 2.13. The predicted values are computed via a decision tree $f(x)$.

$$\sum_{i=1}^N \frac{1}{2} (y_i - f(x_i))^2 \quad (2.13)$$

The first step is to initialize a tree with a constant value, for example zero or the mean value (Equation 2.14). This initial tree therefore predicts a constant value for all input observations.

$$f(x) = 0 \quad (2.14)$$

Next, the residual r_i for each observation, that is, the negative gradient of the cost function, is computed, shown in Equation 2.15 [38].

$$\begin{aligned} r_i &= -\nabla C(y_i, f(x_i)) \\ &= -\left[\frac{\partial C(y_i, f(x_i))}{\partial f(x_i)}\right]_{f(x)=f_{t-1}(x)}, \quad \text{for } i = 1, \dots, N. \end{aligned} \quad (2.15)$$

Using Equation 2.13 as the cost function, the negative gradient is simply the ordinary residual $y_i - f_{t-1}$, which is the difference between the observation and previous prediction.

Next, a tree $f_t(x)$ is trained on the residuals calculated above, and a shrunk version of $f_t(x)$ is added to the original tree by multiplying by a learning rate ν , shown in Equation 2.16.

$$f(x) \leftarrow f(x) + \nu \cdot f_t(x) \quad (2.16)$$

After this, new residuals are calculated as per Equation 2.15, a new tree is trained on the residuals and the model is updated. The model $f(x)$ is updated until the total number of trees, T , has been reached. The final boosted model is given by Equation 2.17 [20]:

$$f(x) = \sum_{t=1}^T \nu \cdot f_t(x) \quad (2.17)$$

Various parameters affect the performance of the boosted trees and these parameters are selected via CV, discussed in Section 2.4. Some consider the number of trees T to be the main tuning parameter [35], since overfitting can occur if it is too large. Early-stopping is often required in order to reduce overfitting. The literature does not provide any guidance for T values and therefore needs to be obtained through optimization. The learning rate, ν , controls how quickly the boosted model learns. If it is too high then overfitting can occur. Boosting works best when the learning rate is low [20]. Typical values of the learning rate are between 0.001 and 0.01. However, smaller ν values lead to larger T values, which require more computation time [15]. The learning rate is also called shrinkage, since it shrinks the effect of the tree when updating the model. Tree size is also an important parameter with suggested values being between four and eight [15] with six reported as best [15, 38]. Finally, subsampling is where training takes place not on the whole sample but on a fraction (between 0 and 1) of the observations and grows the next tree using that subsample [15]. This is called stochastic gradient boosting [38]. Models with subsampling and shrinkage have been shown to perform better than models without [15]. The literature shows subsampling between 0.5 and 0.8 yields good results [38], with 0.5 being a typical value [15].

2.2.4 Deep Learning and Artificial Neural Networks

Deep learning is a part of the family of ML and is based on the artificial neural network (ANN) [19]. ANN originated in the 1940s, was popular again in the 1980s, but in the 1990s other ML methods such as boosting and support vector machines (SVMs) were favored [39]. Due to the increase in the availability of data [19, 40] and improvements in computer hardware and software [19, 39, 40], there was a resurgence in 2006, with the name deep learning [19]. Deep learning has shown to outperform other ML models in a host of domains [40].

ANNs are based on the neuron (Figure 2.1 [41]), also known as a perceptron [42], first modelled by McCulloch and Pitts [43]. The neuron was further developed by Rosenblatt in 1958 and could be trained [42], unlike that of McCulloch and Pitts [44].

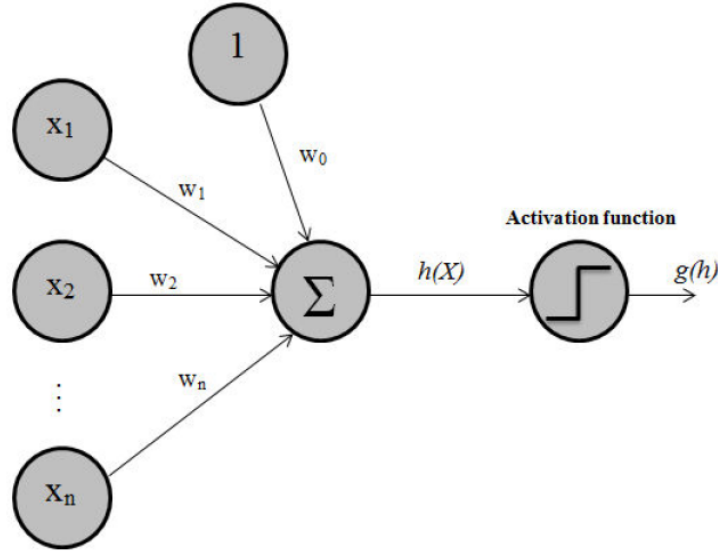


Figure 2.1: Single artificial neuron or perceptron.

The perceptron essentially performs two steps. The first step is to perform a linear summation shown in Equation 2.18 where X is the input features vector, $\mathbf{w}^T$ the transpose of the weights vector, and b the bias term (shown as w_0 in Figure 2.1).

$$h(X) = \mathbf{w}^T X + b \quad (2.18)$$

The second step is to pass the output $h(X)$ to an activation function $g(h)$ that transforms the output to a value between an upper and lower limit. If the value is above a certain threshold, then the neuron (or unit), activates, or fires. Activation functions introduce non-linearity into the model. Without the activation functions only linear problems could be addressed. There are various activation functions such as the sigmoid [15], tanh [44] and rectified linear unit (ReLU) [44] shown in Equations 2.19, 2.20 and 2.21 respectively. Plots of these functions are presented in Figure 2.2. The sigmoid function transforms $h(X)$ to zero if $h(X)$ is very negative, to one if very positive, and to some intermediary value if near zero. The function tanh operates similarly, but with lower and upper limits of -1 and +1.

$$g(h) = \sigma(h) = \frac{1}{(1 + e^{-h})} \quad (2.19)$$

$$g(h) = \tanh(h) = \frac{e^h - e^{-h}}{e^h + e^{-h}} \quad (2.20)$$

$$g(h) = \text{ReLU} = \begin{cases} h & \text{for } h > 0 \\ 0 & \text{otherwise} \end{cases} \quad (2.21)$$

Sigmoid and tanh activation functions were historically the most used but the ReLU function is increasingly becoming the default activation function and is considered most popular [19, 40]. ReLU typically learns faster than the other activation functions when implemented in a deep learning network due to saturation effects in the tanh and sigmoid functions [19]. During learning, the gradient of the cost function needs to be large and if functions saturate (become flat), the gradient becomes small and learning slows. ReLU however, does not saturate when g is above zero and allows for larger gradients and faster learning. The three activation functions mentioned here are the options available in the *Scikit-learn* library used in this study. An alternative to the ReLU function is the Leaky-ReLU [39] shown in Equation 2.22, where α is a negative slope parameter. When α is zero then it is equivalent to ReLU. If α is a small value like 0.01, then this allows for a small positive slope when the unit is not activated.

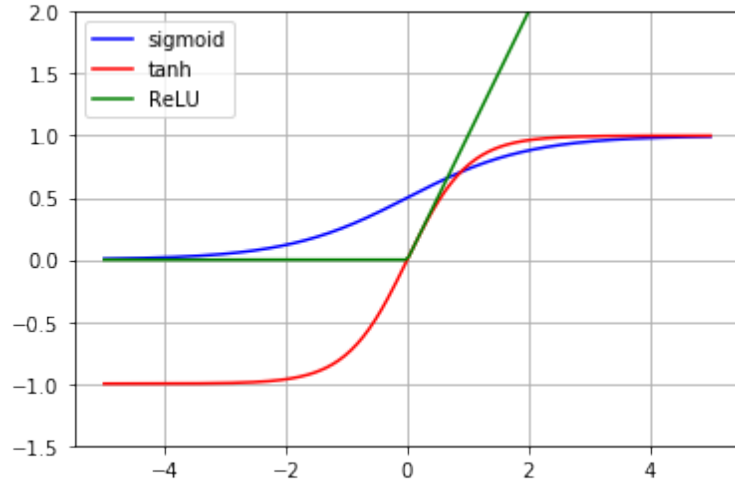


Figure 2.2: Plots of activation functions.

$$g(h) = \text{Leaky ReLU} = \begin{cases} h & \text{for } h > 0 \\ \alpha h & \text{otherwise} \end{cases} \quad (2.22)$$

Single perceptrons are limited in their capability [19] and therefore the feedforward ANN, also known as a multi-layer perceptron (MLP), was developed. A representation is shown in Figure 2.3 that contains a layer of input features, a hidden layer of neurons (or units), and an output layer. Networks with more than one hidden layer are known as deep neural networks, hence deep learning.

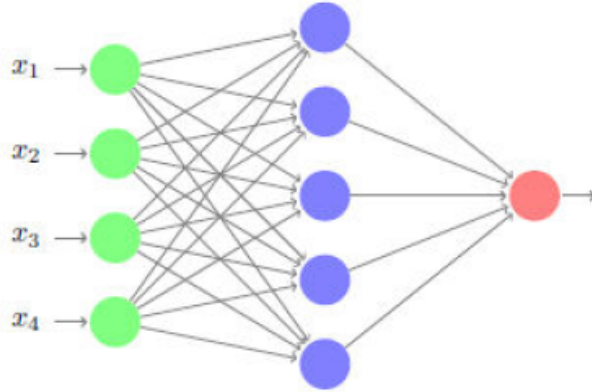


Figure 2.3: Artificial neural network with a single hidden layer of multiple units and a single output unit [39].

In a similar fashion to other ML models, neural networks learn by updating the weights of each input in each layer so that the predicted output matches as closely as possible an existing output. Updating the weights is therefore an optimization problem. In order to train the feedforward network and find optimal weights, two operations are carried out. The

first is *forward propagation* of the activations: the input features are fed into the first layer and information is propagated forward through each layer until the output is produced. The error is calculated between this output and the actual target output. For regression, the cost function C is given by the squared error, shown in Equation 2.23, where $\hat{y}_k$ and y_k are the predicted and actual target output respectively, and m is the number of units in the output layer.

$$C(\hat{y}_k, y_k) = \frac{1}{2} \sum_{k=1}^m (y_k - \hat{y}_k)^2, \quad (2.23)$$

For classification problems, most modern ANNs are trained using maximum-likelihood implying that the cost function is the negative log-likelihood, or cross-entropy, presented in Equation 2.24 [19]:

$$C(\hat{y}, y) = -y \cdot \ln(\hat{y}) - (1 - y) \ln(1 - \hat{y}), \quad (2.24)$$

In the second operation, the widely used algorithm *back-propagation* is used [45]. Via the chain rule of calculus, back-propagation computes the gradient of the cost function with respect to all the weights in the network. This allows information from the cost function to flow backward through the network [19], with the goal of updating the weights. Computing the gradient makes it possible to use gradient methods for minimizing the cost function and thereby training the weights. Other non-gradient methods for updating the weights also exist, for example evolutionary algorithms such as genetic algorithms [46].

Back-propagation is described next. Consider an ANN with three layers, input, hidden and output, with N_i inputs labelled x_i , N_a hidden units labelled a_j and N_o output units labelled o_k . The units use any activation function g discussed above. Further consider v_{ij} to be the weight connecting the i^{th} input and the j^{th} hidden unit and w_{jk} to be the weight connecting the j^{th} hidden unit to the k^{th} output unit.

In forward propagation, the output from an output unit is:

$$\begin{aligned} \hat{y}_k &= g(h_j) \\ &= g\left(\sum_j w_{jk} \cdot g_j\right) \end{aligned} \quad (2.25)$$

and g_j is the activation of a hidden unit given by:

$$\begin{aligned}
g_j &= g(h_j) \\
&= g\left(\sum_i v_{ij}x_i\right)
\end{aligned} \tag{2.26}$$

Therefore computing the partial derivative of the cost function with respect to w_{jk} , using the chain rule from calculus, gives the following:

$$\begin{aligned}
\frac{\partial C}{\partial w_{jk}} &= \frac{\partial C}{\partial \hat{y}_k} \cdot \frac{\partial \hat{y}_k}{\partial h_j} \cdot \frac{\partial h_j}{\partial w_{jk}} \\
&= -\sum_k (y_k - \hat{y}_k) \cdot g'(h_j) \cdot g_j
\end{aligned} \tag{2.27}$$

Further computing the partial derivative of the cost function with respect to v_{ij} gives the following:

$$\begin{aligned}
\frac{\partial C}{\partial v_{ij}} &= \frac{\partial C}{\partial \hat{y}_k} \cdot \frac{\partial \hat{y}_k}{\partial h_j} \cdot \frac{\partial h_j}{\partial g_j} \cdot \frac{\partial g_j}{\partial h_i} \cdot \frac{\partial h_i}{\partial v_{ij}} \\
&= -\sum_k (y_k - \hat{y}_k) \cdot g'(h_j) \cdot w_{jk} \cdot g'(h_i) \cdot x_i,
\end{aligned} \tag{2.28}$$

The gradients $\frac{\partial C}{\partial w_{jk}}$ and $\frac{\partial C}{\partial v_{ij}}$, also represented by $\nabla \mathbf{w}$ and $\nabla \mathbf{v}$, respectively, are subsequently used in the optimization process to update the weights shown in Equations 2.29 and 2.30. The weights are adjusted by taking small steps in the negative gradient direction, or descent direction, known as gradient descent. The parameter ϵ is the learning rate and is discussed later. During training with gradient descent, all the observations are passed into the network before making a gradient step [39]. Stochastic gradient descent (SGD) is a more efficient method than gradient descent and is currently the most common method used for training ANNs [19]. Instead of using all observations per gradient step, SGD passes in only minibatches of observations of size m at a time.

$$\mathbf{w} \leftarrow \mathbf{w} - \epsilon \cdot \frac{1}{m} \nabla_{\mathbf{w}} \sum_{i=1}^m C(\hat{y}_k, y_k), \tag{2.29}$$

$$\mathbf{v} \leftarrow \mathbf{v} - \epsilon \cdot \frac{1}{m} \nabla_{\mathbf{v}} \sum_{i=1}^m C(\hat{y}_k, y_k), \tag{2.30}$$

An improvement on SGD is to include a method called momentum to accelerate learning, by adding the term αv to the update equations above. Momentum accumulates an exponentially

moving average of the past gradients and forces the algorithm to move in that direction [19]. Momentum is given by the term v which plays the role of velocity: it gives the direction and speed at which the parameters move. Another term, the hyperparameter $\alpha \in [0, 1]$, affects how quickly the previous gradients decay and can be thought of as friction that slows down the momentum. Typical values of α are 0.5, 0.9 and 0.99 [19].

Similar to the other ML methods, the performance of the ANN depends not only on the weights, called parameters, but also on various other hyperparameters, which include the number of hidden layers, number of units in each layer, initial learning rate, learning rate schedule, activation function, optimization solver, momentum value, regularization value, number of epochs and batch size. The number of hyperparameters is indeed large and they are briefly discussed next.

For *hidden layers*, the literature has generally used between 1 and 5 [15, 39, 47, 48] while others simply say it is task specific [39]. In terms of *hidden units*, the literature has used between 2-24 [49], 39 [49], 3-19 [50], 2-15 [51], suggested between 5-100 [15] and suggested using 2/3 of input layer [49]. Training a neural network is really an art [15] and due to different datasets, it truly is an optimization process that varies per dataset.

The *learning rate* $\epsilon \in (0, 1]$ is said to be the most important hyperparameter [19, 52]. According to Bengio [52], if only one hyperparameter can be tuned, the learning rate is the best to tune. It is the step size taken when performing SGD. A too large learning rate can result in the error diverging. Typical values of between 10^{-6} and 10^{-1} are suggested [52]. A heuristic is to start with an initial learning rate and if the error diverges, to keep decreasing until no divergence is seen [52]. In this study, the *Scikit-Learn* modules *MLPClassifier* and *MLPRegressor* were used to construct the neural networks. These modules have two parameters relating to learning rate: *initial learning rate* (as the name says) and *learning rate schedule*, that indicates how the learning rate will be adapted at each time step. There are three types of schedules, *constant*, that keeps it at the initial value, *invscaling*, that gradually decreases it, and *adaptive*, that keeps it constant as long as the training error is decreasing.

2.3 Model Evaluation Metrics

Models are trained on training data and are evaluated by measuring their predictive performance on unseen data [15]. These evaluations are carried out by computing various error metrics. Measuring how well models perform on unseen data is crucial for deciding which models to choose, since this indicates how well a model will generalize to new data. Model

training and selection is presented in Section 2.4 and is based on this section that presents the evaluation metrics used for all the regression and classification models used in this study.

2.3.1 Regression

To evaluate regression models, mean squared error (MSE), mean absolute error (MAE) and R-squared (R^2) were used. MSE, already presented in Equation 2.12, calculates the average squared difference between the predicted and actual values (Equation 2.31).

$$MSE = \frac{1}{N} \sum_{i=1}^N (y_i - \hat{y}_i)^2 \quad (2.31)$$

MAE presented in Equation 2.32 calculates the average of the absolute errors between predicted and actual values. This was used when measuring the grades prediction in Chapter 6, since it provides a more intuitive indication of how far off the predictions are from the actual grade.

$$MAE = \frac{1}{N} \sum_{i=1}^N |y_i - \hat{y}_i| \quad (2.32)$$

Finally, Equation 2.33 presents the coefficient of determination, or R -squared, which is a measure of how well a line fits the data. In the right-most term in Equation 2.33, the numerator is the RSS that describes the variation of the model's predictions about the actual output and the denominator is the total sum of squares which is the variance of the actual output about the mean.

$$R^2 = 1 - \frac{\sum_{i=1}^N (y_i - \hat{y}_i)^2}{\sum_{i=1}^N (y_i - \bar{y})^2} \quad (2.33)$$

2.3.2 Classification

This study looks at binary classification (2-class) and for evaluating binary classification models, the metrics *accuracy*, *sensitivity*, *specificity*, *area under the curve (AUC)*, *precision* and *F1-score* were used. The basis of many of these classification metrics is the *confusion matrix* presented in Figure 2.4, which is a table summarizing the performance of a classifier. A classifier is trained on a training set and then tested on an out-of-sample test set (see Section 2.4). In supervised learning, the predicted classes on the test set are then compared with

the actual classes and the results are presented in a confusion matrix. The *actual* classes are presented in the rows with the positives and negatives on the top and bottom rows, respectively. The *predicted* classes are shown in the columns, with positives and negatives in the first and second column, respectively. True positives (TP) and true negatives (TN) occur when the predicted and actual classes match. These fall on the diagonal of the confusion matrix. When the predictions do not match the actual classes, the results are false positives (FP) and false negatives (FN), falling on the off-diagonal. Positive or negative classifications are defined beforehand. For example, in a study predicting students passing a course, a pass could be considered the positive classification. However, if a study is trying to predict failing students, then a fail can be considered the positive classification. Figure 2.4 is for a binary class, however confusion matrices can be extended to any number of classes.

		Predicted class	
		<i>P</i>	<i>N</i>
Actual Class	<i>P</i>	True Positives (TP)	False Negatives (FN)
	<i>N</i>	False Positives (FP)	True Negatives (TN)

Figure 2.4: Confusion matrix [53].

Accuracy, one of the most common classification metrics, is the percentage of true predictions out of all the predictions, shown in Equation 2.34. It serves as an initial indication of how good the classifier performs. However this metric is not ideal when dealing with imbalanced data (covered in Chapter 3).

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN}, \quad (2.34)$$

Sensitivity, also known as recall or true positive rate (TPR), defined in Equation 2.35, is the number of true positives out of the number of actual positives. The word sensitivity is helpful in that it indicates how sensitive the model is in picking up the positive cases. For example, if it was vitally important to pick up a patient with a disease (considered a positive), a highly sensitive model would not miss any positives. However, by increasing the sensitivity, it is likely to also include many false positives.

$$Sensitivity = \frac{TP}{TP + FN}, \quad (2.35)$$

Specificity, also called true negative rate (TNR), defined in Equation 2.36, is the percentage of true negatives out of all the actual negatives. This is similar to sensitivity but it measures how the model identifies *negative* cases. Considering again the example for identifying patients with or without a disease, specificity measures how many healthy people (considered negative cases) are correctly identified. Therefore high sensitivity means the diseased patients are not missed, and high specificity means a healthy patient is not incorrectly labelled as having the disease.

$$Specificity = \frac{TN}{FP + TN}, \quad (2.36)$$

Sensitivity and specificity can be combined into a graph called the receiver operating characteristic curve (ROC). The ROC curve is commonly used for assessing the tradeoff between sensitivity and specificity [15]. Classifiers classify based on a threshold. For example, a logistic regression model may classify an observation as positive if the probability is greater than a threshold of 0.5, and negative if it is below the threshold. The ROC curve is obtained by varying the threshold of a classifier and for each threshold, measuring and plotting the sensitivity and specificity. Although it measures the tradeoff between sensitivity and specificity, it is often plotted as sensitivity vs 1-specificity. The latter quantity is known as the false positive rate. An example of the ROC curve is shown in Figure [20]. Another way to construct this curve is according to Hastie et al. (2009) [15], who plot sensitivity vs specificity as shown in Figure 2.6. The overall performance of a classifier is then measured by calculating

the area under the ROC curve, called AUC. The better the model, the higher the AUC. A model that operates no better than random chance, will have an AUC value of 0.5, indicated by a diagonal curve. Classifiers are often compared via their AUC values.

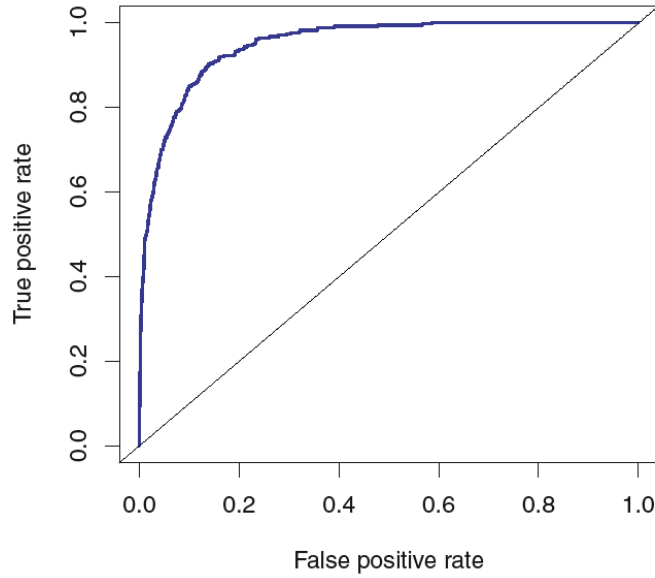


Figure 2.5: Receiver operating characteristic curve, ROC, according to James et al. (2013) [20]

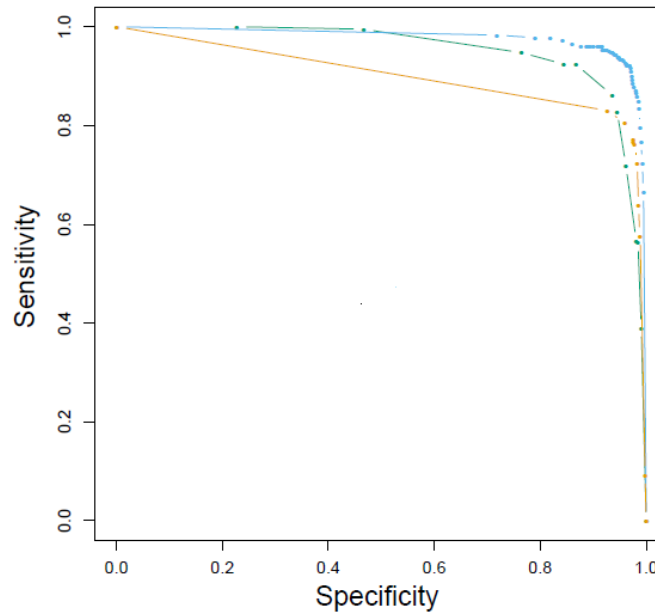


Figure 2.6: Receiver operating characteristic curve, ROC, according to Hastie et al. (2009) [15]

Precision is next presented in Equation 2.37. Whereas sensitivity is the fraction of true positives out of all the actual positives, precision is the fraction of true positives out of all

the *predicted* positives (TP + FP). It is also known as the *positive predictive power*. In other words, out of the observations that were predicted as positive, what fraction was relevant? Intuitively, this measures how many incorrect predictions were made.

$$Precision = \frac{TP}{FP + TP}, \quad (2.37)$$

Single metrics are never presented on their own. It is important to present them in pairs since a metric can be made to score highly simply by adjusting the threshold. A metric that combines sensitivity and precision into a single number is called the *F-measure* which is the weighted harmonic mean of sensitivity and precision, given in Equation 2.38 [54]. The β parameter gives different weights to precision and sensitivity. The higher the value, the greater the importance of precision. According to the *Scikit-learn* library [30], $\beta < 1$ lends more weight to precision whereas $\beta > 1$ favors sensitivity. A common value for β is 1, which is the harmonic mean of sensitivity and precision (Equation 2.39), and places equal importance on both.

$$F_\beta = \frac{(\beta^2 + 1) \cdot precision \cdot sensitivity}{\beta^2 \cdot precision + sensitivity}, \quad (2.38)$$

$$F_1 = \frac{2 \cdot precision \cdot sensitivity}{precision + sensitivity}, \quad (2.39)$$

Equation 2.39 is beneficial since if either precision or sensitivity is low, then the score will be low. For it to be high, both precision and sensitivity need to be high.

Another metric known as Cohen's Kappa, κ [55], is also used in this research. It is considered more robust than accuracy, since it also takes agreement by chance into account. The equation for κ is given below [56].

$$\kappa = \frac{accuracy - P(chance)}{1 - P(chance)}, \quad (2.40)$$

The term $P(chance)$ is the probability of chance agreement between the actual and predicted classes, described as follows:

$$P(chance) = P(positive) + P(negative), \quad (2.41)$$

where $P(positive)$ and $P(negative)$ are the probabilities that the actual and predicted positive and negative classes agree by chance. Consider again the confusion matrix (Figure 2.4) where the probability that both the prediction and actual classes are positive by chance is:

$$\begin{aligned}
P(positive) &= P(positive)_{actual} \cdot P(positive)_{predicted} \\
&= \frac{TP + FN}{TP + TN + FP + FN} \cdot \frac{TP + FP}{TP + TN + FP + FN}
\end{aligned} \tag{2.42}$$

The probability that both the prediction and actual classes are negative by chance is:

$$\begin{aligned}
P(negative) &= P(negative)_{actual} \cdot P(negative)_{predicted} \\
&= \frac{FP + TN}{TP + TN + FP + FN} \cdot \frac{FN + TN}{TP + TN + FP + FN}
\end{aligned} \tag{2.43}$$

Therefore the intuition behind Cohen's κ is that it removes the chance agreement between the predicted and the actual classes. The κ values are interpreted as per Table 2.1 [57] where values less than 0 are worse than random, values of 0 are completely random agreement and 1 is perfect agreement.

Table 2.1: Interpretation of Cohen's κ scores [57].

κ	Agreement
< 0.00	Worse than random
0.00 - 0.20	Slight
0.21 - 0.40	Fair
0.41 - 0.60	Moderate
0.61 - 0.80	Substantial
0.81 - 1.00	Near perfect

Finally, the *Brier score* was also used. It is a measure of the accuracy of a probabilistic prediction. According to the *Scikit-learn* documentation [30], across all items in a set of N predictions, the *Brier score* measures the MSE between the predicted probability, P_i of an item and its actual value O_i , shown in Equation 2.44. The *Brier score* value is always between 0 and 1, where the lower the value, the better the score, because lower values indicate smaller differences between predicted and actual probabilities.

$$\text{Brier score} = \frac{1}{N} \sum_{i=1}^N (P_i - O_i)^2 \quad (2.44)$$

2.4 Model Assessment and Selection

An important goal in ML is to compare models (using error metrics) in order to find the best performing ones. During model training two important tasks are accomplished. The first is to determine optimal model hyperparameters. Models typically have hyperparameters that affect model performance and it is necessary to tune these parameters to get optimal performance. This is known as *model selection* [15] and is performed on training data. The second task is to estimate predictive performance on out-of-sample, test data. This is known as *model assessment* [15].

Ideally, it would be preferable to randomly split the data into training, validation and testing sets [15]. *Model selection* can be carried out, i.e. the model hyperparameters could be tuned, by training the model on the training set and estimating prediction error on the validation set. Thereafter *model assessment* could be carried out by estimating predictive or generalization error on the test set [15]. However, if data is scarce, setting aside a validation set might reduce the ability to train effective models. This would then leave two sets, i.e. training and testing, where the model is trained and tested on the training and testing set, respectively. A problem with this method is that the test error is highly variable [20]. If training and testing is carried out multiple times on different 80%-20% splits of the data, the test error has been shown to vary greatly each time [20]. This high variance leads to overfitting the training data and obtaining poor prediction error. Some errors will be very low and others high.

Another method called k -fold CV aims to improve on the high variance problem inherent in training and testing sets. It randomly splits the dataset into typically $k = 5$ or 10 folds, trains on $k-1$ folds and tests on the k^{th} fold. This method improves on the previous train-test method by repeating this k -times, each time training and testing on a different $k-1$ fold and k^{th} observation. This results in not one test error but k errors and the variance is reduced by averaging these errors as per Equation 2.45. CV refers to CV error.

$$CV = \frac{1}{k} \sum_{i=1}^k MSE_i \quad (2.45)$$

Furthermore, the number of folds (k) can be varied and the bias and variance in the selected models are dependent on the number of folds used. The higher the number of folds, the

lower the bias and the higher the variance. The literature however, recommends $k = 5$ or 10 as a good compromise between bias and variance [15, 58]. Others have further suggested performing repeated CV to get more reliable results [58]. For example, ten-fold CV can be repeated ten times. The challenge with repeated CV is that it can become computationally very expensive. *In general, this study therefore uses ten-fold CV (without repeat) to estimate performance of all the models.*

Model selection and assessment were briefly introduced above and is described next. Models have parameters and hyperparameters that define them and model selection is to obtain the optimal hyperparameters by using k -fold CV: A model with a specific combination of hyperparameters is trained and tested using k -fold CV. The hyperparameters are then changed and CV is carried out again, obtaining another set of error metrics. This is repeated for a large range of hyperparameters until a minimum test error is found. The model is then selected based on the hyperparameters that resulted in this lowest test error. For example, when selecting a lasso model, the λ hyperparameter is varied between a lower and upper range, and for each value in this range, the test error is calculated via CV. The optimal λ value is then, in general, chosen at the minimum error.

For regression problems, CV is relatively simple to compute, due to the metrics being simple to calculate (Section 2.3). Classification problems on the other hand, pose more of a challenge due to having imbalanced datasets and needing to generate error metrics based on confusion matrices. Therefore for regression problems in this study, ten-fold CV was used on the entire dataset. However, for *classification* problems, the entire dataset was split first into 80% training and 20% testing, using stratified sampling. Model selection (i.e. hyperparameter tuning) was then performed on the 80% training set using ten-fold CV; a technique used by Hastie et al. (2009) [15]^v. Then in order to assess various metrics which are difficult to assess using k -fold CV, the 20% test set was used. Stratified sampling was used when performing the train-test split. This is important since randomly splitting imbalanced data will likely result in different class distributions between the training and testing sets. Stratification ensures that both sets have the same proportion of classes as the entire dataset. More on imbalanced data is presented in detail in Chapter 3.

2.5 Conclusion

The majority of this study is based on ML methods. This chapter therefore presented the relevant ML concepts, including the models used and model evaluation and selection methods. The applications of ML in this study includes both prediction and inference (explanation),

^vsee page 61, Section 3.3.4, of Elements of Statistical Learning.

which are both based on the models described in this chapter. This chapter therefore forms a solid basis for Chapters 3 through 6.

The models used comprise lasso, RF, adaboost, GB and ANN, since these were shown to perform the best in a number of empirical studies. They were described in detail in this chapter.

Furthermore, the importance and methods of model selection and assessment to obtain the best models, was described. This is to carry out hyperparameter tuning (model selection) using CV. For regression problems, based on the literature, model selection would be performed using ten-fold CV applied to the entire dataset. For classification problems, model selection would be performed using ten-fold CV on an 80% training set and tested on a 20% test set.

Each subsequent chapter presents a more thorough review of the literature related to that chapter. The following chapter presents the data preparation for the majority of this study and has a major component that uses ML to impute missing data.

Chapter 3 Data Preparation

3.1 Introduction

The context of this study is a first year engineering mechanics course, PHYS1015, presented at the University of the Witwatersrand in the first half of 2016. The objectives were to predict and characterize At-Risk studentsⁱ and to measure treatment effects of interventions carried out in this course. Both of these broad goals used ML and statistical methods. To carry this out, this chapter presents the entire process of preparing the dataset for model training in the later chapters.

This chapter:

1. Describes the dataset including the predictor and target features being used (Section 3.2).
2. Describes normalizing and encoding of features (Section 3.3).
3. Presents the literature on missing data imputation (Sections 3.4).
4. Analyses and performs missing data imputation (Section 3.5).
5. Addresses correlated features (Section 3.6).
6. Presents the final datasets to be used for subsequent analyses (Section 3.7).

3.2 Predictor and Target Features

The features used in training ML models are vital to the performance of the models. The predictor features include demographic, grades and extra tutorial attendance (intervention)

ⁱStudents are classified as At-Risk if they fail PHYS1015.

features provided by the University and recorded by the author.ⁱⁱ The aim was to predict the final PHYS1015 grade (obtained in June) as early as possible in the semester, which was taken to be the March series of tests. Therefore only features (demographic and grades) up until March were used in the models.

The original dataset had 901 observations (indicating 901 students) and 16 predictor features. The target and predictor features are described next.

3.2.1 Target Feature

- **PHYS1015 Final Grade:** This grade is the target feature calculated at the end of the semester in June 2016. It was either analyzed as pass/fail (classification) in Chapters 4 and 5, or as a continuous grade (regression) in Chapter 6. The code is given as PHYS1015.Mid-Year or Mid-Year.PHYS1015.

3.2.2 Predictor Features

The predictor features described next are attributes of students in the given dataset that have been identified as potential predictors of the target feature. The codes of the features are presented in most cases in brackets.

- **High school grades.** For this feature, three specific high school grades were chosen, namely Mathematics (HS.Mathematics), Physical Sciences (HS.Physical.Sciences) and English. These were chosen since the mathematics and physical sciences should have a direct influence on the performance in engineering mechanics. It was felt that language might also have a impact on performance and was therefore also included. The Mathematics and Physical Sciences were continuous variables. English is a categorical feature indicating whether the student took English as a (1) first language (HS.English.FIRST), (2) second language (HS.English.SECOND) or (3) as a foreign student (HS.English.FOREIGN).
- **Ethnicity.** Due to the background of South Africa, race is predicted to have a large effect on the educational backgrounds of the students which may influence performance. This feature has five categories, Black (Ethnicity.Black), White (Ethnicity.White), Coloured (Ethnicity.Coloured), Indian (Ethnicity.Indian) and Chinese (Ethnicity.Chinese).
- **Gender.** How does the student's gender predict performance? Male (Gender.Male) and female (Gender.Female) were encoded.

ⁱⁱEthical clearance was obtained from the University.

- **Bursary information.** It is predicted that the type of bursary might affect student performance. Four types of bursary categories were encoded: (1) no bursary (Bursary.None); (2) academic bursaries from the University (Bursary.UES); (3) bursaries from engineering companies (Bursary.OT); (4) bursaries from the National Student Financial Aid Scheme (Bursary.NSF).
- **Residence information.** Does the student live at home (RESIDENCE.No) or in the residence on campus (RESIDENCE.Yes)?
- **March 2016 Grades:** Three test results were chosen from the March tests: PHYS1015 (engineering mechanics, March.Test.PHYS1015), PHYS1014 (physics, March.Test.PHYS1014) and MATH1014 (mathematics, March.Test.MATH1014).
- **Type of student.** This feature describes if a student was a first year (Student.Type.First) or repeat student (Student.Type.Repeat) in PHYS1015.
- **Years since matriculation.** This feature refers to the number of years since the student graduated high school. It was proposed that this feature might affect performance. The code is Years.Since.Matric.
- **Citizenship.** Three categories of citizenship were chosen: (1) South African (Citizenship.RSA); (2) permanent resident (Citizenship.PR) or (3) resident of another country (Citizenship.ROC).
- **Age:** Does the age of the student affect student performance? The code is Age and is a continuous value.
- **Degree of study:** The PHYS1015 course is primarily taken by engineering students as well as other closely related degrees. Nine engineering degrees were considered which are described either in name form or by degree code shown in brackets. They are Mechanical (DegCd.EF005), Industrial (DegCd.EF007), Aeronautical (DegCd.EF006), Electrical (DegCd.EF003), Mining (DegCd.EF009), Civil (DegCd.EF001), Biomedical (EB000), Digital Arts (EB001) and Nuclear (SB000).
- **ADU.** This feature refers to Academic Development Unit (ADU), a name for supplementary tutorials on PHYS1015. It is considered an *intervention* and is provided for students that desire to make use of extra assistance and is completely voluntary. Students do not have to pay to attend and can join or leave these extra tutorials at any time during the semester. The student either attended ADU (ADU.1st.Sem.Yes) if they attended at least once, or not (ADU.1st.Sem.No). This feature plays a vital role, especially in Chapter 6 where treatment effects of ADU are analyzed.

The above-mentioned features were chosen primarily based on what was provided by the University as well as data collected by the author (ADU attendance). In future studies,

potential predictor features could include questionnaires on motivation or personality, more detail on parents education, details on province, town and high school, various information regarding whether the student receives a meal every day, whether the students have a support group of student peers, what kinds of accommodation they live in and what resources they have. There is considerable room for developing a follow-up study with a range of new features.

3.3 Normalizing and One-Hot Encoding

Before performing any analysis, it was important to first normalize the data [59, 60, 61]. In datasets, different features may have different ranges of data that might have a twofold effect on training ML models. One effect is on ML classifiers, such as nearest neighbor or support vector machine models, that calculate distances between points [62]. If one feature has a much larger range than others, distances will be dominated by that feature. The aim is to give each feature an equal weighting so that information is not lost. The other effect of non-normalized data is that when minimizing a cost function using gradient descent, if one feature has a much larger range, it takes longer to converge than if the ranges were similar.

Normalization is calculated as per Equation 3.1 by generating a z -score [63] for each value $\mathbf{x}$ in a feature. The z -score indicates the number of standard deviations above or below the mean value [63] and is computed by subtracting the mean μ of a numeric feature from each value in the feature column, and then dividing by the standard deviation σ . The result is a normalized vector of numeric features where the mean is zero and the standard deviation is one [15].

$$\mathbf{z} = (\mathbf{x} - \mu)/\sigma \quad (3.1)$$

In this study, the grades features all have ranges of approximately 100%; age has a range of approximately six years; and the categorical features, after encoding are either a one or zero. Normalization was applied to the seven continuous features in the dataset, i.e. the two high school grades (mathematics and physical sciences), the three March tests (mechanics, mathematics and physics), the age and years since matriculation.

Furthermore, it is important to encode the categorical features into a numeric form for model training. Two common methods are label encoding and one-hot encoding. Label encoding assigns numbers from 1 to $n-1$ for n -categories. Label encoding may be problematic since the ML model may attribute more importance to the category with the higher number, when in

fact there is no order of importance in the categories. One-hot encoding was therefore used in this study since it does not attribute an order of importance when encoding. It adds a binary presentation of the category: one if the category is present and a zero if absent. The categories in this dataset are all nominal, meaning there was no intrinsic ordering. One-hot encoding was carried out in *RStudio* [64] using the *dummyVars* function [65] and was performed on the nine categorical features in the dataset. *RStudio* is a programming language for statistical computing and graphics.

The important task of addressing missing data is presented next.

3.4 Literature on Missing Data

Missing data is common in all research [66, 67] and needs to be addressed since it can substantially affect the conclusions drawn [68, 67]. Although addressing missing data is an important part of data preparation, it has only been addressed superficially in the related literature [9, 69, 70, 71, 72, 73]. No thorough treatment of missing data has been presented. Could the results in the literature have been more trustworthy had the authors addressed missing values? It is believed that addressing missing data in a detailed way may provide more reliable results in later analyses, including prediction and characterizing of At-Risk students and predicting treatment effects. An overview of missing data and imputation techniques is presented next. Thereafter, the literature on missing data in predicting student performance is presented.

3.4.1 Types of Missingness

The first step in dealing with missing data is to determine the type of missingness; i.e., why the data are missing. This is important since it determines what type of methods can be used for addressing missing data. In general, there are three types of missingness [74]:

1. Missing completely at random (MCAR): This type of missingness results from random chance. For example, missing values may simply be due to arbitrary mistakes while capturing data. In this case, ignoring the missing data does not introduce any bias into the data. The probability of missingness is unrelated to any other variable in the dataset. If the data is MCAR, then methods such as listwise deletion can be applied, discussed in Section 3.4.3.1.
2. Missing at random (MAR): This type of missingness is conditional on observed data, not on missing data. Once a certain observed feature is controlled for, the missingness

becomes random [68]. For example, foreign students may have missing high school grades information, but within the foreign student group, the missing data is random. This type of missingness is known as ignorable. If missing data is MAR, then methods such as multiple imputation can be used, discussed in Section 3.4.3.3.

3. Missing not at random (MNAR): MNAR missingness depends on unobserved data, even on the missing feature itself. For example, in a survey that includes a question on income, a person with low income may choose to exclude their income. Therefore the missing data is dependent on itself. A further example of MNAR missingness is when there is dropout or attrition in a program. For example, a patient suffering with depression may drop out of a program if they begin to improve [74]. Whereas certain methods may not introduce bias into the data under the MCAR and MAR conditions, bias will be introduced under the MNAR condition [75]. MNAR is also known as non-ignorable.

3.4.2 Determining Missingness Type

Currently there is only a single statistical test, i.e. Little’s MCAR test, that can determine whether the missingness is MCAR or not [76]. Using a χ^2 test, it tests whether there are statistically significant differences between means of different missing-value patterns [77]. The null hypothesis states that there are not significant differences, i.e. that the missing data is MCAR. Therefore at a 95% confidence interval, p -values smaller than 0.05 computed for this test indicate *not* MCAR.

If the missingness is not MCAR, then the missingness is either MAR or MNAR. Schafer and Graham (2002) [74] say that there is no way to test for MAR, except by measuring follow-up data from the people that did not respond. In most cases, it is challenging to obtain follow-up data. There is the need therefore to make assumptions [78] and study the missingness case by case. It is suggested to examine patterns in the data to get some insight into the most likely mechanism for missingness [79]. The literature suggests that for many cases, assuming MAR is sufficient without introducing significant bias [74, 75].

3.4.3 Methods for Addressing Missing Data

3.4.3.1 Listwise and Pairwise Deletion

Various methods are used to deal with different types of missingness. If completely at random (MCAR) then *listwise deletion* and *pairwise deletion* are options [74]. Listwise deletion is

to delete an entire observation if one or more values are missing. If the missingness is *not* MCAR then deleting these observations might introduce bias into the data [74]. However, if missing data is less than about 5% or if the total number of observations deleted listwise is under 5%, then the bias in the data will likely be inconsequential [68]. Listwise deletion is the default approach to dealing with missing data in most statistical software packages [74, 80].

Pairwise deletion deletes cases only when a pair of features include a missing value. For example, if correlation is performed on two features, and one of the features has missing data, the correlation will exclude the pair of values where one value was missing [75]. Disadvantages of this method are that different subsets of observations are used for different analyses. This will tend to bias parameter estimates and likely result in a non-positive definite covariance matrix which cannot be used for most multivariate statistical analyses [68, 75].

3.4.3.2 Single Imputation - Mean, Mode, Median and Regression

Single imputation is a method that imputes single values into the missing data, for example the *mean* in continuous features [68, 75] and the *mode* in categorical features. Mean imputation, however, decreases the variance due to adding a number of mean values [75] and is not recommended by some [68]. Single imputation will introduce bias into the data for all types of missingness (MCAR, MAR and MNAR).

Regression is another single imputation method for estimating missing values. The variable that is missing is regressed on other measured variables. The problem with a single imputation procedure is that the filled in data does not have the same variation in the data, had it been complete [68, 75].

3.4.3.3 Multiple Imputation

Multiple imputation (MI) aims to restore the lost variance inherent in single imputation [68, 74] and can be applied in both MAR and MNAR cases. However under the MNAR cases, MI will introduce biased estimates into the data [81]. MI creates multiple different imputed data sets. Various analyses are performed on each imputed data set, then the results are averaged over the imputed datasets. For example, the MSE could be calculated for each imputed data set. Thereafter, the average MSE is calculated. The literature has suggested between three and fifty imputed datasets depending on the amount of missing data [68, 74, 82]. The higher number is said to reduce simulation error and improve power [81, 82]. Some suggest that ten

imputations are sufficient [74, 81]. Therefore to limit computation time, in this analysis, the number of imputed datasets was ten.

A current method for multiple imputation is known as MICE [81, 82]. In this method a series of regression models are fitted where each model is dependent on the other variables. This method can be applied in *RStudio* using the *MICE* package [82]. The way *MICE* carries out imputation is described below and is based on Azur et al. (2011) [81]:

1. Begin by imputing a simple value such as the mean for all missing values. This is viewed as a place- holder.
2. For a feature, F_m , remove the place-holder values and return values in that feature to missing.
3. The non-missing, observed values in F_m are then considered as dependent variables and are regressed against other features. Essentially this is to fit a regression model where the observed values in F_m are the predictions and the other features are the input variables.
4. The missing values in F_m are then imputed via prediction with the regression model from Step 3.
5. Then, consider another feature, remove that feature's place- holder values and impute using the same method. However, now include the imputed data in F_m in the previous regression model. This is then repeated for each feature until all the features have had their missing values imputed via these regression equations.
6. The above method is then repeated multiple times based on the idea that after multiple cycles the parameters in the regression model should have converged to become stable. Eventually a single imputed dataset is obtained.

There are however some disadvantages of MICE. The first is that this process is repeated multiple times to generate multiple imputation sets. Therefore for any further analysis, i.e. training and testing of ML models, the models need to be trained and tested on *all* the imputed datasets every time and the results averaged out. Secondly, the default for MICE is for the imputation regression models to not include interaction terms that could fit non-linear data [83]. Finally, MICE is a parametric model and therefore is sensitive to multicollinearity [83]. The next method aims to circumvent these limitations.

3.4.3.4 Random Forest Methods

The *missForest* method, developed by Stekhoven and Bühlmann [84], aims to circumvent the limitations of MICE discussed above. This method is based on RF methods [28] and has been shown to outperform MICE methods in a number of simulation studies [84, 85, 86]. RF was presented in detail in Section 2.2.2. The *MICE* function in *RStudio* also has the functionality of using random forests for multiple imputation. Thus far, no literature has been found that applies random forest methods, including *MICE-RF* and *missForest* to studies on predicting student performance. In the context of missing data, RF is able to deal with mixed-type data (continuous and categorical) and as a non-parametric method it allows for interactive and non-linear (regression) effects; it makes no structural assumptions regarding the data [84].

Consider a dataset with features that have missing data. For a given feature with missing data, there are two parts, the observed part and the missing part. *missForest* carries out the following two steps for each of these features in the dataset:

1. Fit a RF on the observed part,
2. Using the fitted RF model, predict the missing part.

The above two steps are repeated until some stopping criterion is met: for example as soon as the difference between the most recent imputed matrix and the previous one increases for both continuous and categorical features [84]. *missForest* was implemented in *RStudio* using the *missForest* package.

3.4.3.5 Dealing with MNAR

The above sections have described methods for dealing with MCAR and MAR types of missingness. If these two types cannot be assumed, then selection models and pattern-mixture models may be useful for dealing with MNAR missingness [87, 74]. This complicates the imputation method since additional modelling assumptions are required that influence the imputations [82], i.e. the missingness must be modelled [74, 78] and a distribution for the missingness must be specified [74].

To avoid these complications, the literature recommends assuming MAR [74] and further argues that models assumed in the MNAR may often be incorrect and perform worse than MAR methods [68]. Furthermore, Schafer et al. (2002) [74] state that where the cause of the missingness is not based on itself but on some other unmeasured feature that is moderately correlated (< 0.4) with the missing value, only minor bias might be introduced. Others agree

that the bias introduced when using MI may be less than other traditional approaches [75]. To make the MAR assumption reasonable, the literature recommends that as many predictors be included in the analysis as possible [78, 68].

3.4.4 Missing Values in Predicting Student Performance

The sections above presented the theory behind dealing with missing data. The relevant literature on predicting student performance has very little to say on analyzing missing data. In this field, listwise deletion methods have been used [9, 69, 73], but no detail was given indicating if the missing data was less than 5% or missing completely at random (MCAR) (a requirement for implementing listwise deletion). The most thorough description found was by Mduma et al. (2019) [88] who addresses missing values but simply imputes zeros and medians into missing values. Beaulac et al. (2019) [70] only states that missing values are a problem. Perhaps these studies did address the missing value problem but did not report on it. MICE was used in two case studies on predicting student performance [71, 72]. Nevertheless, no work has been found addressing missing data in a rigorous way. One aim of this study is to predict and characterize At-Risk students. It is proposed that the better the missing data is addressed, the better the predictions and characterizations will be.

3.5 Analysis of Missing Data

In Section 3.4 the type of missingness and various methods to address missing data were described. These methods were applied to the dataset in this study in order to obtain a final dataset. This section first analyzes the data to determine the type of missing data.

3.5.1 Original Data

After normalizing and encoding, the dataset had 901 observations and 38 features. Figure 3.1 shows the overall missing values in terms of variables (features), cases (i.e. observations) and values. It shows that 8.2% of cases (observations) contain missing values and overall there are only 1.28% missing values. If the missingness is MCAR, then it is possible to delete the observations listwise without introducing any bias.

Table 3.1 describes the features that have missing data. The maximum percentage was March test PHYS1014 with 4.1% and the minimum was years since matriculation (Years.Since.Matric) with 2.4%. The rest of the features in the dataset did not have any missing values.

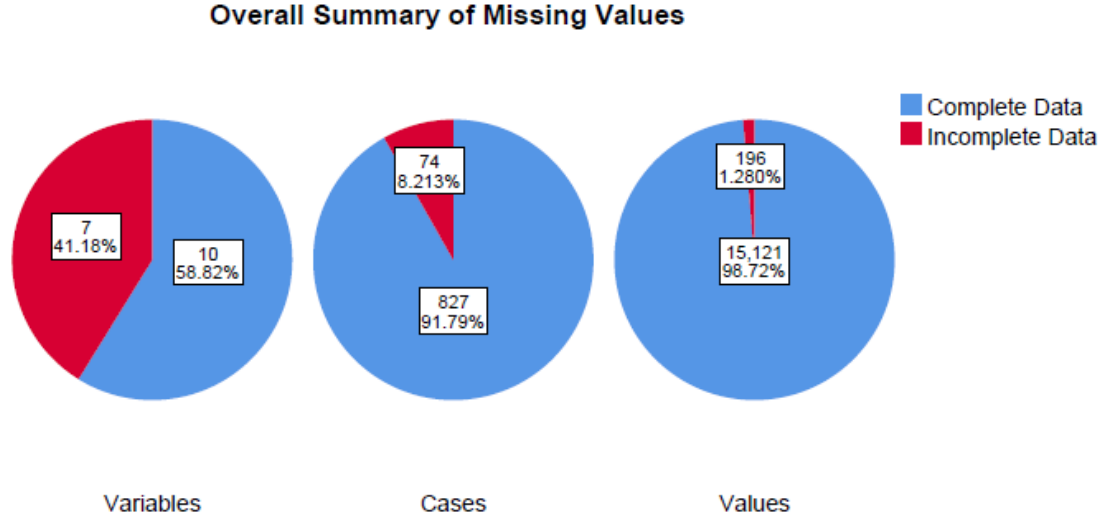


Figure 3.1: Overall missing data analysis generated via *SPSS* (Version 25).

Table 3.1: Features containing missing data generated via *SPSS* (Version 25).

Feature	Number	Percent [%]
March test PHYS1014	37	4.1
March test PHYS1015	35	3.9
March test MATH1014	29	3.2
High school physical sciences	25	2.8
High school mathematics	25	2.8
PHYS1015 Final Grade	23	2.6
Years since matriculation	22	2.4

3.5.2 MCAR Test

As indicated above, the percentage missing values was well below 5% (Figure 3.1) and therefore would allow for listwise deletion. However, this requires the missingness to be MCAR. Little's MCAR hypothesis test was carried out in the statistical package *SPSS* [89] and a p -value of 0.000 was calculated. This value is significant and suggests that the data was *not* MCAR but either MAR or MNAR. Therefore based on this, it is recommended to not carry out listwise deletion.

3.5.3 Determining MAR/MNAR

It is therefore important to determine if the missingness was MAR or MNAR. This is done by analyzing patterns in the missing data, presented in Figure 3.2 (using *SPSS*). This shows

different patterns where the same data is missing. For example, pattern 3 shows that there exists a number of cases (students) where years since matric, high school mathematics and high school physical sciences are missing simultaneously, indicated by the red blocks. That is, a number of students have exactly the same missing features. The aim here was to analyze these patterns to determine if MNAR or MAR missingness exists.

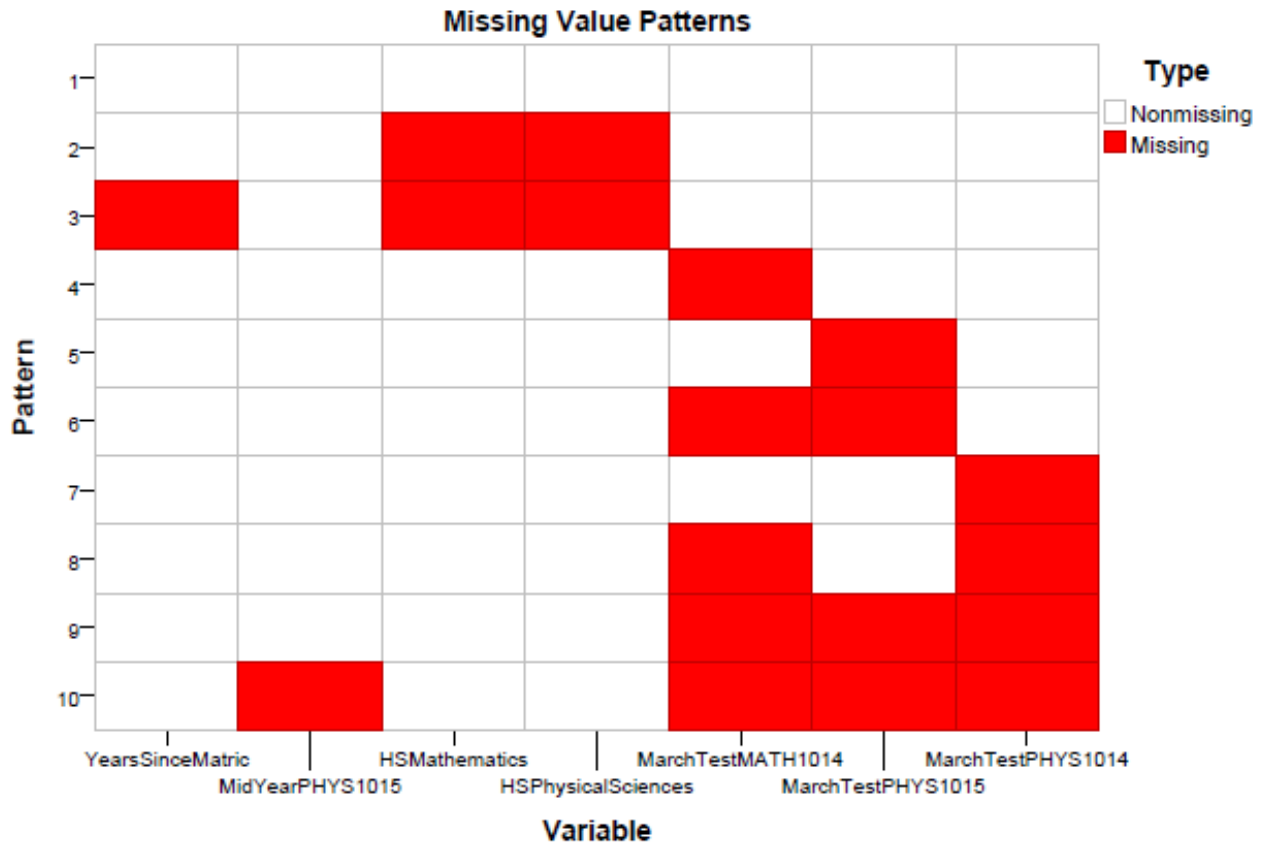


Figure 3.2: Features containing missing data generated via *SPSS* (Version 25).

Beginning with pattern 10, it is seen that March test PHYS1015, March test MATH1014, March test PHYS1014 and PHYS1015 Final Grade (Mid-Year.PHYS1015) are missing. This pattern includes two missing values related to the PHYS1015 course (March test and Mid-Year). This suggests that these students may have dropped out of the PHYS1015 course since there was no record for them in the beginning (March) and the end (Mid-Year). Therefore to more accurately determine what was happening here, more mechanics (PHYS1015) grades features were included in the dataset and the patterns of missingness were again analyzed. *Note that these extra features are not part of the desired dataset, but were only used to determine the missingness.* The reason they were not included in the desired dataset is because they were grades features captured after March, whereas the aim was to make predictions from March. However, these extra features were helpful solely in determining

the missingness type. The first was May test PHYS1015 (the second major test administered in May); the second was the overall tutorial grade for the 1st Semester PHYS1015 (the code is Tut.1st.PHYS1015); and the third was the PHYS1015 exam in June (the code is June.Exam.PHYS1015). The reason for adding these was to gain deeper insight into the missingness. However, they all occurred after March and were not included in the predictive features. Figure 3.3 shows the patterns when these extra PHYS1015 features are included.

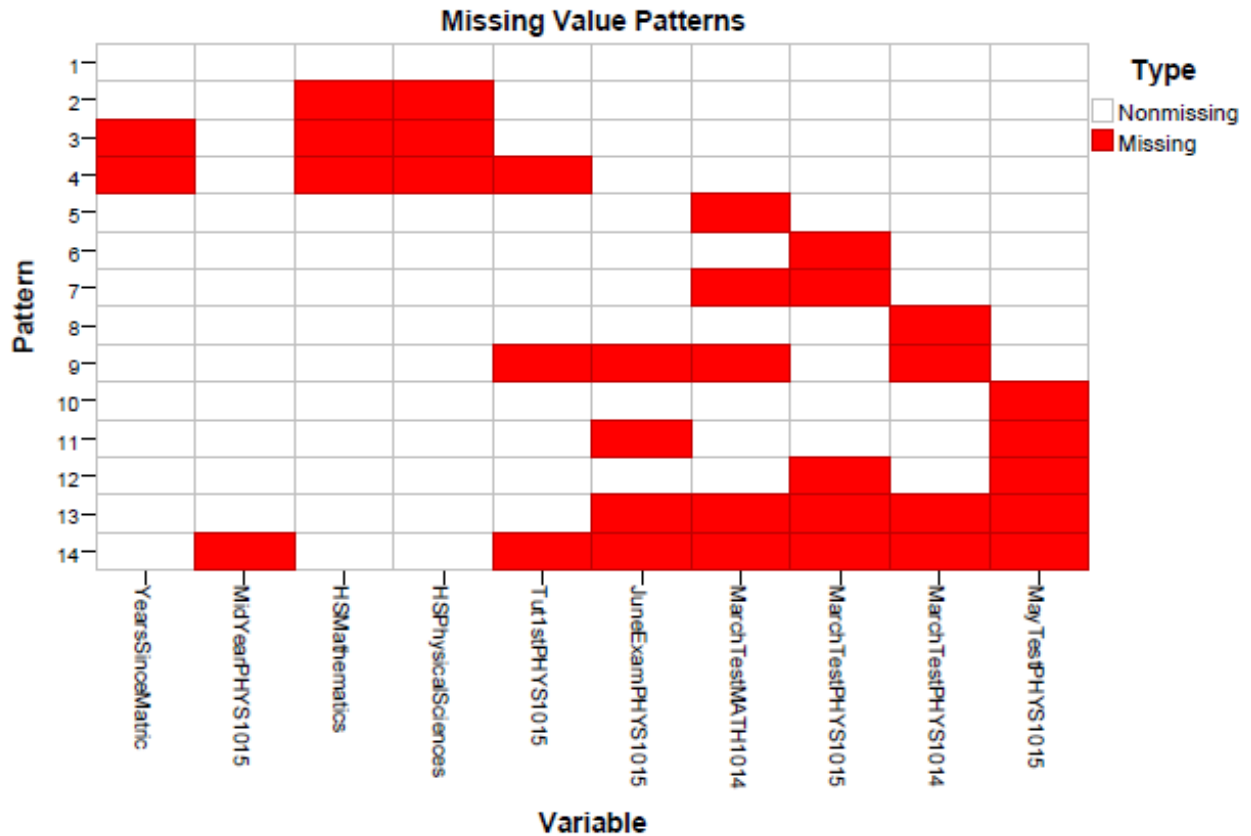


Figure 3.3: Features containing missing data generated via *SPSS* (Version 25).

Beginning with pattern 14, it can be seen that these students have no data for March test PHYS1015, May test PHYS1015, June exam PHYS1015 and the 1st semester tutorial grades. That is, there is no record at all for *all* their assessments. This is a clear indication that they dropped out before even commencing with the first assessment. *Therefore the observations from pattern 14 were completely deleted from the dataset.* This amounted to deleting 23 observations leaving 878 observations. An update of the missing values is presented in Figure 3.4 after deleting the total dropouts. The percentage missing values decreased from 1.28% to 0.66% and observations with missing data decreased from 8.2% to 5.8%. Figure 3.2 also shows how the maximum percent missing is now in the high school grades (2.8%).

Pattern 13 is almost identical to pattern 14, except that these students have a tutorial grade (Tut.1st.PHYS1015) which means that although they did not write any tests or the June

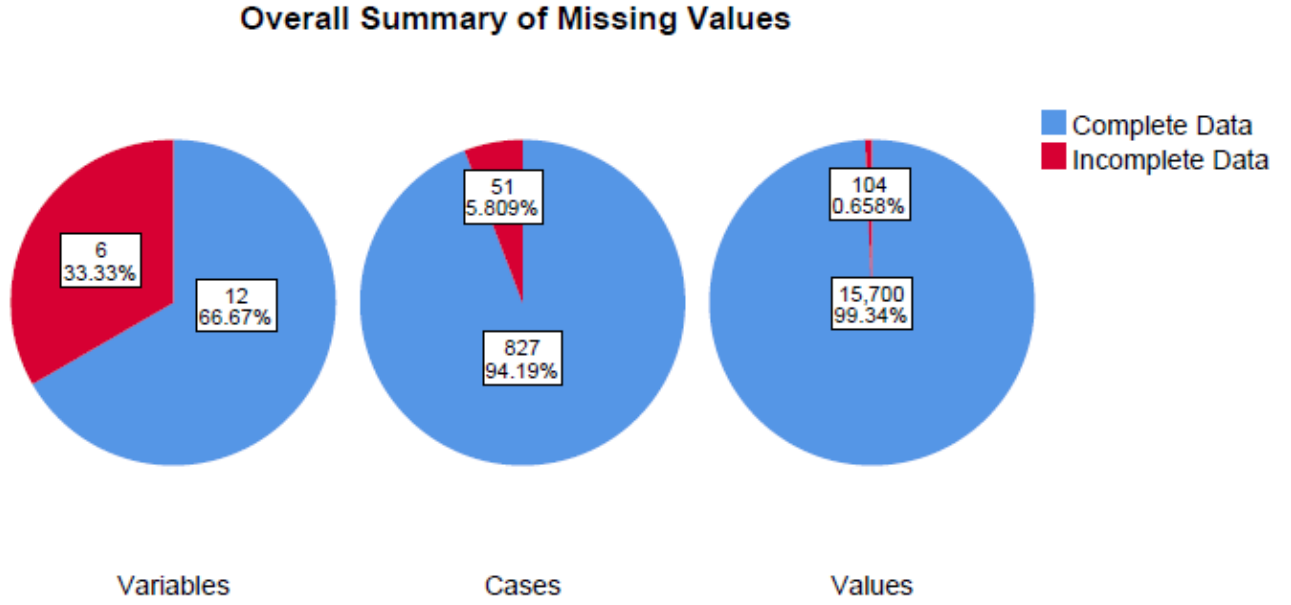


Figure 3.4: Summary of missing values after deleting total dropout observations. Generated by *SPSS* (Version 25).

Table 3.2: Summary of features with missing values after deleting total dropout observations. Generated by *SPSS* (Version 25).

Feature	Number	Percent [%]
March test PHYS1014	14	1.6
March test PHYS1015	12	1.4
March test MATH1014	6	0.7
High school physical sciences	25	2.8
High school mathematics	25	2.8
Years since graduation	22	2.4

exam, they did attend class to some extent. It is therefore assumed that these student dropped out early on. That is, they attended some tutorials and obtained a tutorial grade but never subsequently wrote any tests. When dropout occurs in a program, the subsequent missing values are known as non-ignorable or MNAR [74, 90, 91], because the missing value is dependent on itself or dependent on some unobservable factor. For example, the missing values for pattern 13 (i.e. March, May and June PHYS1015 tests) may be due to the students performing poorly in the tutorials. That is, the missing grades were *because* of poor grades. The other reason might be due to unobservable factors such as relocating to another city or switching to a another degree course. Nevertheless, both are considered MNAR. Further dropout is seen in pattern 11 where both March test PHYS1015 and 1st semester PHYS1015 tutorial grades are present but then May test PHYS1015 and June exam PHYS1015 are missing. Pattern 9 also likely shows dropout where March test PHYS1015 and May test

PHYS1015 are present but the tutorial and June exam PHYS1015 grades are missing. Therefore patterns 9, 11, and 13 in Figure 3.3 can be considered MNAR types of missingness. With regards to patterns 5, 6, 7, 8 and 10, it is not obvious whether the missing values are dependant on an observed feature. Due to the uncertainty, these missing types were also considered MNAR. Pattern 12 shows missing values for the March and May PHYS1015 tests, but have the tutorial grade and exam present. It is uncertain why these students did not write the two major tests but were still somehow allowed to write the exam. Due to uncertainty, this pattern was also considered MNAR. Patterns 5 through 14 have now been addressed.

The next patterns analyzed were 2, 3 and 4 in Figure 3.3. These patterns all include missing values for *high school information* (grades and date of matriculation). A further analysis of these students indicated that all the missing high school data is due to citizenship; all these students were residents of another country. The university therefore did not provide the high school grades for these foreign students in the same way as for the South African students. In total there were 33 foreign students, and 22 of these foreign students are missing high school mathematics and physical sciences grades. Due to this missingness being based on an observable feature, it was assumed to be MAR.

The pattern types have been analyzed and it was determined which patterns to completely delete, and which patterns are assumed to be MAR and MNAR. In general, using imputation techniques on MNAR will lead to biased results. However, in order to perform multiple imputation, the following presents the reasoning behind applying MAR methods to MNAR data in this dataset. The first reason is described by Schafer and Graham (2002) [74]. As discussed above, MNAR (patterns 9, 11, 12 and 13) may be due to either the outcome of the feature itself, or to an unobserved factor. If the true cause of the missing values is the value being measured (i.e. the missing grades are due to poor grades), then failure to account for MNAR might significantly bias the estimates. However, if the cause is not the value being measured, but unmeasured and unobservable factors that are not highly correlated with the outcome (less than 0.4), then only minor bias might be introduced. Schafer and Graham (2002) state that in many social science studies, the second reason is much more common. Therefore, in this study, it was assumed that dropout was due primarily to unobserved factors discussed above (i.e relocating, changing degrees, needing to work etc.). It is therefore assumed that analyzing the patterns (9, 11, 12 and 13) using MAR techniques will not introduce significant bias [74, 75]. These MNAR patterns make up six MNAR observations. There are however also MNAR missing types for patterns 5, 6, 7, 8 and 10 which amount to 32 observations. Therefore to test the assumption of whether MAR methods can be applied to the entire dataset, the following was carried out. Multiple imputation was applied to the entire dataset of 878 observations as well as to a reduced dataset that excludes the MNAR observations (total of 38). In this way comparisons can be made between the

datasets to see if any significant bias is introduced. If there is no significant difference in the outcome (in this case MSE), then there is confidence in this method.

The dataset that includes all observations is called Entire, with 878 observations. Thereafter the 38 MNAR observations were deleted and this dataset is called Reduced, with 840 observations. For each dataset, ten datasets were imputed using the MICE package in *RStudio*. The ten datasets were “measured” by performing linear regression using ten-fold CV and computing the MSE for each imputed dataset (ten in total). The mean MSE values and the standard deviations for the Entire and Reduced datasets are presented in Table 3.3. Furthermore a t-test was performed to compare mean values and a p -value of 0.97 was computed at $\alpha = 5\%$, showing that there is no statistically significant difference in mean values. This therefore strengthens the case for applying MAR methods to the Entire dataset without introducing any significant bias. Imputation using multiple imputation and random forest methods can therefore be performed, shown next.

Table 3.3: MSE for Entire and Reduced dataset. Standard deviations are shown in brackets.

Entire	Reduced	p -value
0.55 (0.019)	0.55 (0.002)	0.974

3.5.4 Missing Data Imputation

In order to compare imputed datasets, a baseline dataset was required. Although it was earlier said that listwise deletion is only possible if missing data is MCAR and less than 5%, the following assumption is made: a baseline dataset was generated consisting of the original dataset that was listwise deleted. The reasoning behind this is that the total missing values are only 0.68% (i.e. potentially insignificant) and statistical packages often automatically carry out listwise deletion when performing statistical analyses [74, 80]. When performing listwise deletion to obtain the baseline dataset, 64 observations had missing data and were completely deleted to leave 814 observations.

Four methods were used to impute data and were compared with the listwise-deleted baseline dataset.

The four imputation methods were:

1. Mean/mode
2. MICE using the default method
3. MICE using the random forest method with 10, 50 and 100 trees
4. missForest with 10, 50 and 100 trees

After imputation, linear regression models were trained on each dataset via ten-fold CV and mean MSE values were calculated for each dataset. The results of these imputed datasets are presented in Table 3.4 and Figure 3.5. In Table 3.4, MF indicates *missForest*. For the multiple imputations there were ten imputed datasets and therefore a range of MSE values is presented. The baseline dataset had an MSE value of 0.56. The mean/mode dataset performed the worst, with the highest MSE of 0.57. This corresponds to the literature that does not recommend using mean as an imputation method [68]. Next, what is interesting is that the baseline dataset performs similarly to all the MICE imputed datasets (both default and random forest). This suggests that the missing data was random, and if MICE was only being compared with the baseline, then the baseline would suffice. However, the *missForest* imputed datasets performed substantially better than the rest of the datasets. Whereas the MSE values for all the other datasets are 0.55 or 0.56, the *missForest* datasets all score 0.51 or 0.52. This result agrees with the literature where RF methods outperform MICE [84, 85, 86]. However, further study is needed to understand why *missForest* performed better than the *MICE-RF* method, since they were both RF methods. In addition to performing the best, a further benefit of the *missForest* method is that it only imputes a *single* dataset,

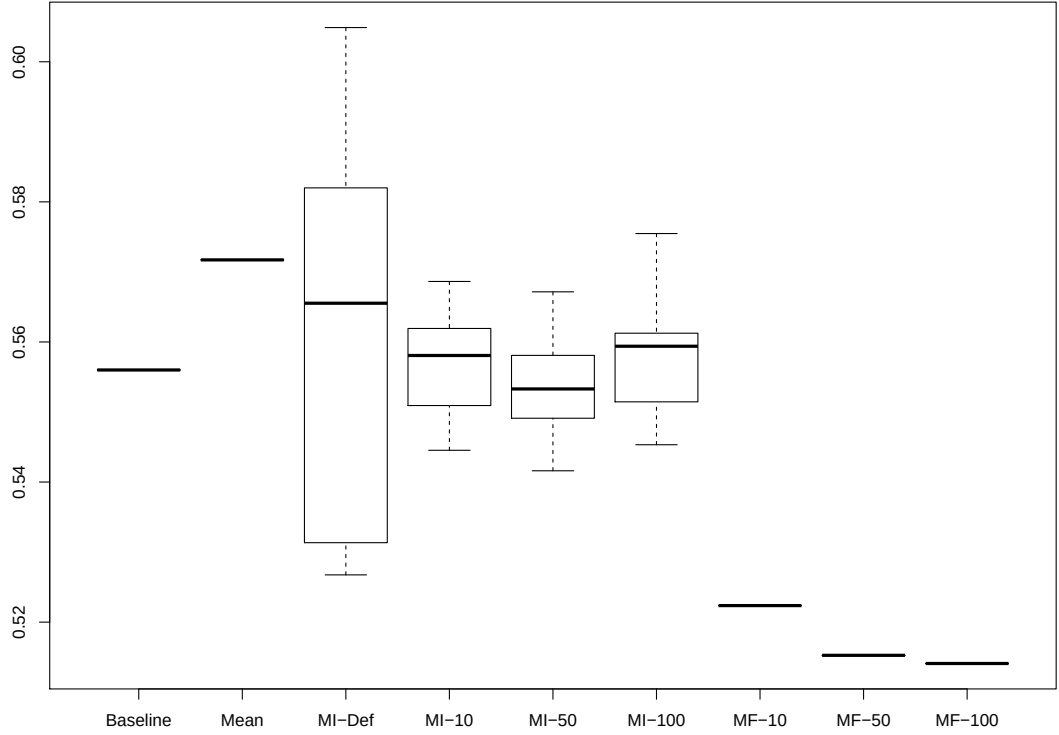


Figure 3.5: MSE values for different imputed datasets.

unlike MICE that imputed ten datasets. Therefore to conclude, *the dataset imputed using missForest with 100 trees was considered the final imputed dataset*. From hereon, all further analysis or computation of the data was based on this dataset.

Table 3.4: MSE for various imputed datasets with standard deviation in brackets, where applicable.

Dataset	MSE
Baseline	0.56
Mean/mode	0.57
MI (Default)	0.56 (0.020)
MI (RF, ntree = 10)	0.55 (0.008)
MI (RF, ntree = 50)	0.55 (0.007)
MI (RF, ntree = 100)	0.56 (0.008)
MF (ntree = 10)	0.52
MF (ntree = 50)	0.51
MF (ntree = 100)	0.51

3.6 Correlated Features

The final imputed dataset was obtained, called Entire dataset. However, feature correlation in datasets can lead to instability in modelling and there was the need to address these correlated features. Figure 3.6 presents a heatmap of the correlation matrix of the final imputed dataset.

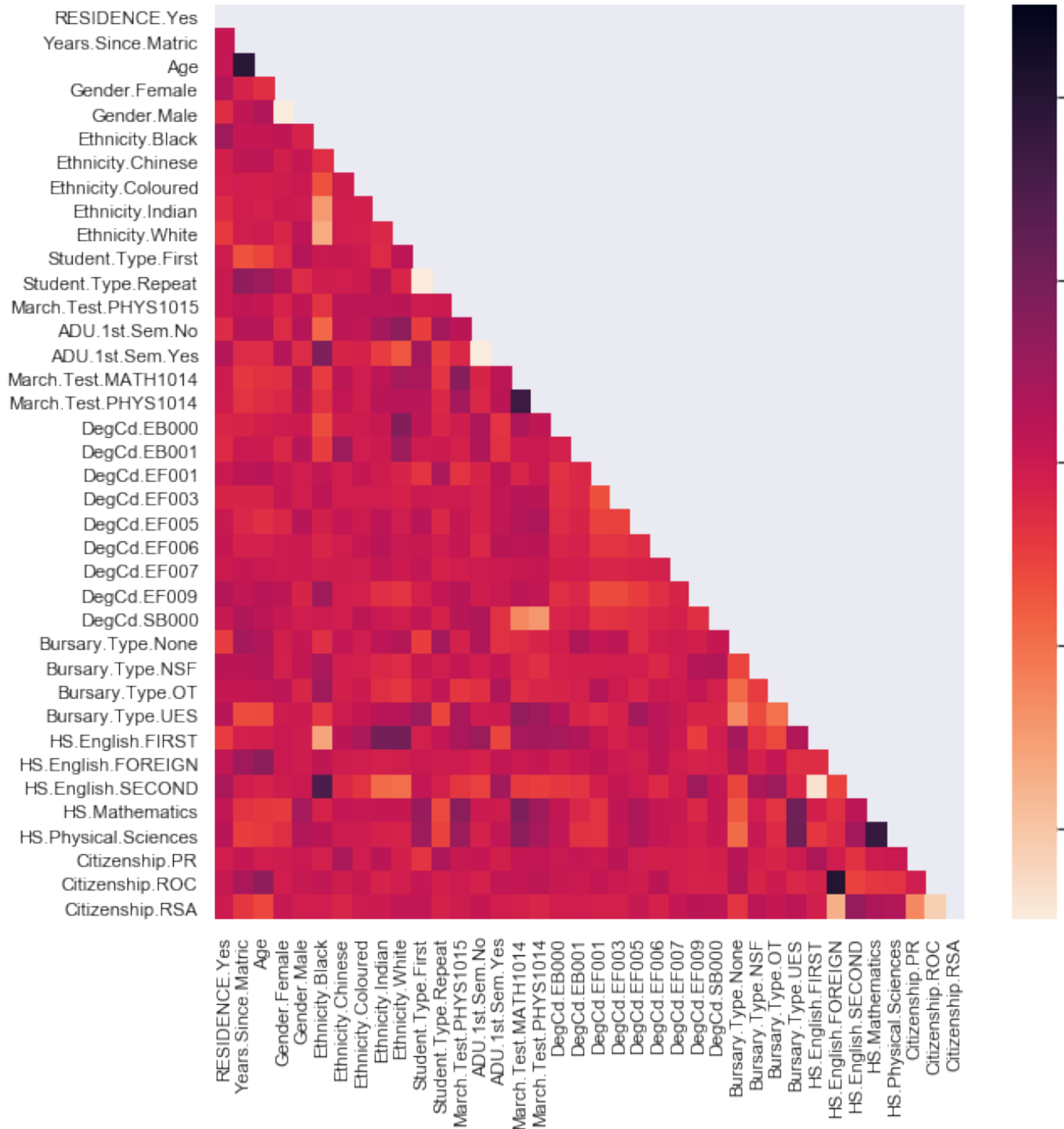


Figure 3.6: Correlation heatmap.

As can be seen in Figure 3.6, there are few highly positively and negatively correlated features. For example, the age feature is highly correlated (0.77) with years since matric. This makes sense since the older the student, the longer they have been out of high school before entering university. Another high correlation is between high school physical sciences and mathematics (0.71). The literature suggests that correlations of between 0.5 and 0.7 are moderate, whereas above 0.7 are high [92]. Therefore if a pair of features had a correlation above 0.7 or below -0.7, one feature was removed. The updated correlation heatmap with correlated features removed is shown in Figure 3.7.

Table 3.5 presents the features that were removed due to high correlation with another feature. Age was already discussed above as being correlated with years since matric. Male (Gender.Male), repeat student (Student.Type.Repeat) and attending the ADU tutorials (ADU.1st.Sem.Yes) were all one of a binary categorical feature and so were highly negatively correlated with the complementary feature. High school English as a second language (HS.English.SECOND) was highly correlated with being ethnically Black (Ethnicity.Black) since most of the South African Black students take English as a second language. High school physical sciences (HS.Physical.Sciences) and mathematics (HS.Mathematics) were also calculated to be highly correlated with each other. The final two removed features require a little bit of discussion. The feature describing foreign students (Citizenship.ROC) was highly correlated with high school English for foreign students (HS.English.Foreign) since ROC indicates resident of another country. However, the feature for South African citizenship (Citizenship.RSA) was highly *negatively* correlated with Citizenship.ROC (-0.85) which is understandable since this almost forms a binary category. That is, a student is either a citizen of RSA or resident of another country. However, the downside is that both these citizenship features needed to be removed and only the permanent resident feature (Citizenship.PR) remained. Although both Citizenship.RSA and Citizenship.ROC were removed, they were both linked to HS.English.Foreign, as shown in Table 3.5. Therefore, later when feature importances are computed, if HS.English.Foreign is selected as an important feature, there is a way of tracing this back to Citizenship.RSA. This will be similarly done for all removed features.

3.7 Final Datasets

In the previous section the final dataset Entire was established. The Entire dataset was subsequently used extensively in Chapters 4 and 5. In Chapter 6 control and treatment datasets were analyzed based on whether a student attended the ADU intervention (treatment) tutorials. Therefore out of the Entire dataset, those cases that didn't attend the ADU (control) and those that did attend (treatment) were extracted into further control and treatment

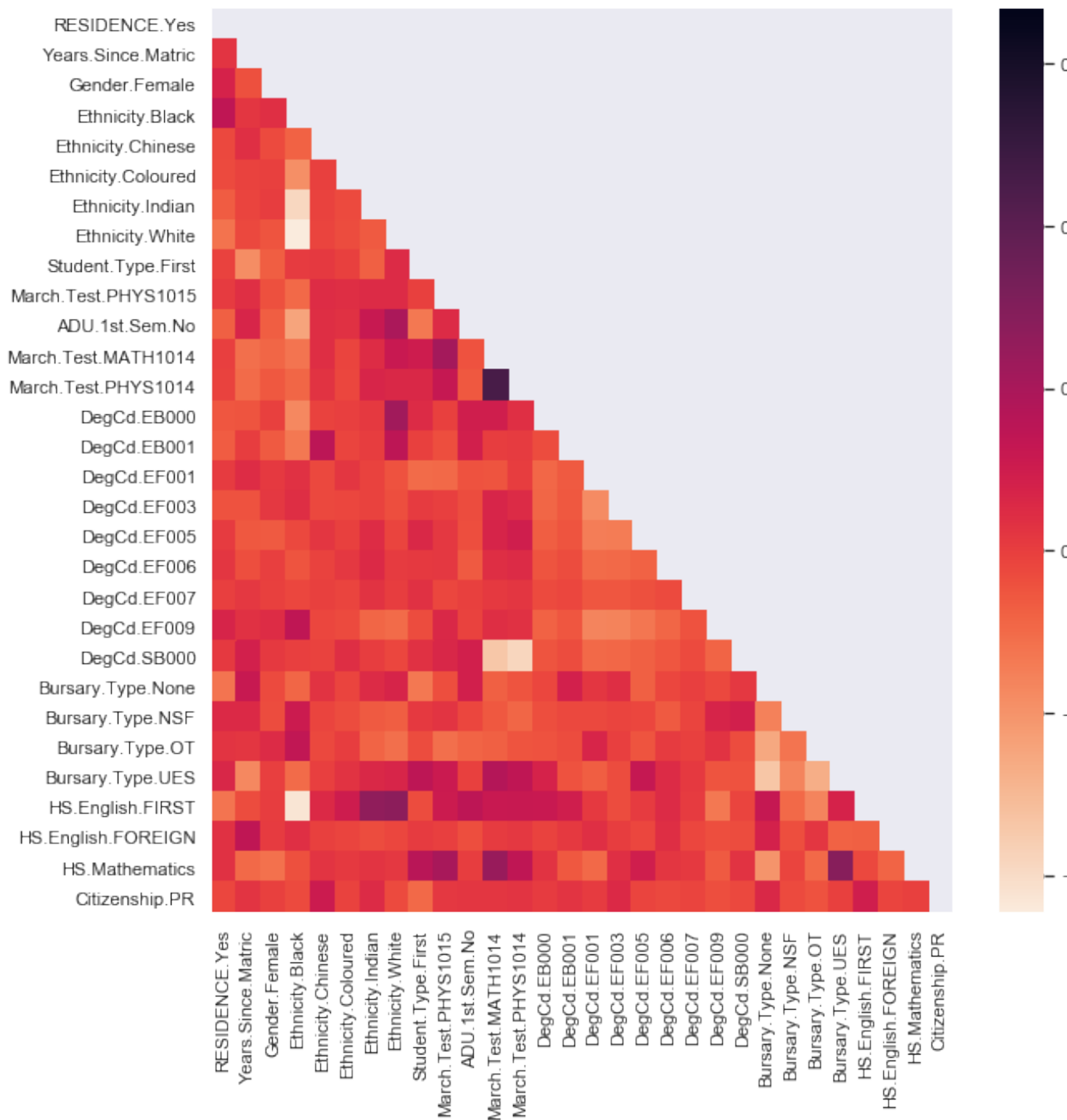


Figure 3.7: Updated correlation heatmap with correlations between -0.7 and 0.7.

subgroups, respectively. These datasets are presented in Table 3.6. Note that for categorical features, the total number in the category is shown, with the percentage in brackets. For a continuous feature, the average value is shown, with the standard deviation in brackets. Further note, although the correlated features have been removed in the dataset to be used for model training, for completion, the overall statistics for all features are still shown in Table 3.6. For example, the feature Male is removed from the dataset since it is correlated with Female, however it is shown in the table.

Table 3.5: Correlated features removed from the dataset

Removed Feature	Correlated Feature
Age	Years.Since.Matric
Gender.Male	Gender.Female
Student.Type.Repeat	Student.Type.First
ADU.1st.Sem.Yes	ADU.1st.Sem.No
HS.English.SECOND	Ethnicity.Black
HS.Physical.Sciences	HS.Mathematics
Citizenship.ROC	HS.English.Foreign
Citizenship.RSA	Citizenship.ROC

Table 3.6: Statistics of different datasets used in this study.

	Feature	Entire (878)	Control (384)	Treatment (494)
Gender	Male	631 (71.8)	296 (77.1)	335 (67.8)
	Female	247 (28.1)	88 (22.9)	159 (32.2)
Age		18.8 (1.1)	18.9 (1.2)	18.6 (1.1)
Ethnicity	African	710 (80.8)	248 (64.5)	462 (93.5)
	Caucasian	84 (9.5)	74 (19.3)	10 (2.0)
	Indian	65 (7.4)	49 (12.7)	16 (3.2)
	Coloured	16 (1.8)	10 (2.3)	6 (1.2)
	Chinese	3 (0.3)	3 (0.8)	0 (0)
Citizenship	RSA	833 (94.8)	367 (95.5)	466 (94.3)
	PR	33 (3.7)	10 (2.6)	23 (4.6)
	ROC	12 (1.3)	7 (1.8)	5 (1.0)
Student Type	First Year	781 (88.9)	315 (82.1)	466 (94.3)
	Repeat	97 (11.0)	69 (17.9)	28 (5.7)
Degree Code	Civil	189 (21.5)	71 (18.5)	118 (23.9)
	Electrical	183 (20.8)	72 (18.8)	111 (22.4)
	Mining	155 (17.6)	66 (17.1)	89 (18.0)
	Mechanical	126 (14.3)	47 (12.2)	79 (16.0)
	Aeronautical	65 (7.4)	18 (4.7)	47 (9.5)
	Industrial	19 (2.1)	8 (2.1)	11 (2.2)
	Biomedical	55 (6.2)	39 (10.1)	16 (3.2)
	Digital Arts	25 (2.8)	21 (5.4)	4 (0.8)
	Nuclear	61 (6.9)	42 (10.9)	19 (3.8)
Residence		205 (25.8)	79 (20.5)	149 (30.1)
Bursary Type	None	281 (32.0)	151 (39.3)	130 (26.3)
	UES	307 (34.9)	133 (34.6)	174 (35.2)
	OT	204 (23.2)	66 (17.2)	138 (27.9)
	NSF	86 (9.8)	34 (8.8)	52 (10.5)
Years Since Grad.		1.3 (0.77)	1.39 (0.85)	1.2 (0.68)
March Test	PHYS1015	72.2 (13.2)	73.4 (14.1)	71.2 (12.4)
	MATH1014	53.8 (20.9)	52.2 (24.4)	55.1 (17.7)
	PHYS1014	48.2 (71.2)	46.5 (20.3)	49.5 (14.3)
High School	Mathematics	80.9 (8.1)	81.0 (8.8)	80.9 (7.5)
	Physical Sciences	79.0 (8.0)	78.6 (9.1)	79.3 (7.1)
High School English	First	293 (33.3)	174 (45.3)	119 (24.0)
	Second	563 (64.1)	204 (53.1)	359 (72.6)
	Foreign	22 (2.50)	6 (1.5)	16 (3.2)
Target Feature	PHYS1015 Mid-Year	61.8 (13.0)	62.3 (14.9)	61.3 (11.4)

3.8 Conclusion

The aim of this chapter was to prepare the dataset for model training in later chapters. It began with describing the dataset, i.e. the target and predictor features. The target feature was the final engineering mechanics grade and the predictor features comprised seven continuous and nine categorical features. The features were simply ones provided by the University and recorded by the author. However suggestions were made in this chapter regarding a range of new features that could be generated in follow-up studies.

The first step was to normalize the continuous features, and one-hot encode the categorical features. These operations were necessary for improving the learning process and converting the categorical features into data that could be read by the model.

Thereafter missing data imputation was performed. This was the first known study on predicting student performance that carried out such a detailed analysis of missing data. It was proposed that carrying out detailed analysis and imputation of missing data would improve later model predictions. Addressing of missing data was performed using *SPSS* and *RStudio*. The majority of missing data patterns was found or assumed to be MNAR. However, it was concluded via a sensitivity computation that applying MAR methods such as *MICE* was acceptable, since it did not introduce significant bias into the dataset. Mean/mode, *MICE* and *missForest* were used as the imputation techniques and were all compared with a baseline listwise-deleted dataset. Mean/mode performed the worst and *MICE* did not perform any better than the baseline. However, the RF imputation method *missForest* performed substantially better than all the others. A *missForest* imputed dataset was subsequently chosen as the final dataset for all further analyses. The results suggest that performing the missing value analysis would result in better performance in ML predictions in the later chapters.

Finally, after final imputation, correlated features were removed to present the final dataset to be used in later chapters.

Chapter 4 Characterizing At-Risk Students Using Model-Agnostic Methods - Model Selection

4.1 Introduction

The previous chapter presented the entire process of preparing the dataset for model training and resulted in a final dataset to be used for model training and testing. This chapter presents the first of two chapters that address the first two objectives of this overall study:

- Predicting and characterizing first year engineering students that are At-Risk.
- Providing advice or interventions for At-Risk students to help them switch to not being At-Risk.

To address these objectives, recently developed model-agnostic interpretable ML techniques were utilized, which is fully presented in Chapter 5. Before introducing and applying these interpretable techniques, it was necessary to first identify the best performing ML models by carrying out detailed model selection. Once the best models were selected, they were then be used in Chapter 5.

4.2 Background

The background of this chapter presents literature related to:

- predicting student performance using ML, and
- addressing imbalance in data.

4.2.1 Predicting Student Performance

In this section literature is reviewed on how ML has been used to *predict* student performance (including At-Risk students). There have been a number of studies looking at what affects student success [93, 94], but the use of ML to predict student performance is relatively new. Kotsiantis et al. (2004) [95] were some of the first to carry this out. In predicting student performance using ML, the literature discusses:

1. what is being predicted,
2. what the input features in the models are, and
3. what types of ML models are being implemented.

Examples of predictions in the literature include overall grade point average (GPA) [73], final course grade [4, 96, 97], final exam grade [3, 4, 95], drop out rate [98], 3rd semester performance [99] and binary response, i.e. pass or fail [4]. Beemer et al. (2017) [4], predicted the effects of intervention programs.

Examples of input features are GPA [100], internal assessments (i.e. class attendance, quizzes, assignments [1, 98]), demographic information (such as gender [98, 4], family education, parental income [98, 101], marital status, children [95] and living arrangements [4]), online activity [97], psychometric behavior [102] and SAT scores [4]. Thus far, perhaps the strongest predictor of student performance is high school GPA [59, 103, 104]. However, Xu et al. (2017) [73] shows no correlation between final university GPA score and high school GPA. Other strong predictors are pre-university SAT scores [59], first year GPA [105, 106], previous semester performance [99], initial attendance [107], homework grades [107] and family income [101].

In terms of ML models, in the context of higher education, regression techniques have been used to predict continuous grades [4, 108]. However, the more common type of predictions have been to classify the output grades [2, 3, 4, 95, 108]. Classification algorithms used in the literature include logistic regression [97], neural networks [108], support vector machines [3, 101], decision trees [101], Bayes network [101], nearest neighbors [3] and ensemble learning algorithms [1, 4, 73]. The following will present the literature in more detail.

Kotsiantis et al. (2004) [95] made use of demographic and internal assessment features to train various models to predict students' passing or failing. Naive Bayes, C4.5 decision trees, neural networks, sequential minimal optimization (SMO), k-nearest neighbor (KNN) and logistic regression were the learners implemented. They conducted two experiments, one

with data from 354 students and the other from only 28 students. They conclude that *Naive Bayes* is the preferred algorithm since it also outperformed the other models with a 78% sensitivity for the first experiment. One way of improving on this study would be to work with larger datasets to see if the algorithms perform similarly on larger data sets.

Ibrahim et al. (2007) [109] aimed to predict final GPA grades using linear regression, neural networks and decision trees. Data from 206 students was obtained and computer knowledge, programming skills and family income were used as input features. *Neural networks* performed the best with decision trees and linear regression second and third, respectively.

Kotsiantis et al. (2010) [1] classified student performance in an online program by developing an *incremental* ensemble learner. In general, algorithms are trained on stationary data (i.e. data that doesn't change over time) but their aim was to train an algorithm that continuously receives new data, to update the model. Updating learning algorithms due to a changing environment is related to concept drift which is where the relationship between input data and target output changes over time [110]. Input features used in this study were written assignments at different points in the courses. Compared with the base learners, the *ensemble learner* had on average the highest accuracy with KNN performing second best.

Agrawal et al. (2015) [108] predicted future grades based on previous semester grades. Grades from the 3rd to 6th semester of an engineering course were collected and used to predict grades in 6 subjects. The population size was 80 and the model was trained using k-fold CV. They initially used multivariate linear regression to determine the relationship between the data to predict marks and obtained an accuracy rate of 70.48%. Thereafter they compared neural networks with Bayesian classifiers and found that *neural networks* outperform Bayesian networks, however they don't provide any further information. This research relies on small data sets, and to improve this research, larger data sets could be obtained.

Similar to Kotsiantis et al. (2010) [1], Xu et al. (2017) [73] trained an ensemble model on continuous data to predict final GPA. Their learning technique allowed for progressive prediction and continuous updating of student data. Their student sample size was 1169. They compared the ensemble learner with linear regression, logistic regression, random forest and k-nearest neighbors and found that as the year progressed and new data was added to the models, *random forest* performed the best of the base learners, and that the *ensemble learner* performed best overall. The worst performer generally, was KNN.

In their review paper on literature predicting student performance from 2002 to 2015, Shahiri et al. (2015) [96] ask two questions: (1) which features are important in predicting student performance and (2) which prediction methods are used? To answer question (1) they showed that cumulative GPA (cumulative final course results) and internal assessments (assignments,

class tests, attendance etc.) are often used predictors. Answering question (2) they showed that *neural networks* achieved the highest average predictive accuracy (98%) followed by decision trees (91%), support vector machines and KNN (both 83%) and then Naive Bayes (76%). Their answer to question (1) is slightly flawed however, in that they ask which features are important in predicting student performance, but instead only report the features that are the most commonly used in the literature. There is a difference between the most important features and the most commonly used features. This creates the need to carry out *feature selection* to determine the strongest predictors.

Shelton et al. (2017) [111] predicted At-Risk students in an online program using online activity and engagement as input features. However, instead of looking only at aggregated activity, for example total time online, they also studied the frequency of online interaction. Frequency was found to be a better indicator of student success than simply the total time of interaction. Predictions were categorical, where students fell into three groups, A, B (passing) or F (failing). They carried out two types of analyses using decision trees, boosting, logistic regression and rule induction. The first analysis consisted of the traditional aggregated data and the second involved time-series clustering (frequency). The time-series clustering approaches performed better at classifying at-risk students (i.e. failing) with the *decision tree* performing best. The number of students in the study was 509.

Anuradha et al. (2015) [2] predicted categorical student performance based on demographic information (i.e. gender, living location, family size, family income etc.), high school grades, and current university information (i.e. branch of study, previous semester grades, grades in the current course etc.). The grade categories were classified as distinction (<75%), first class (60% - 74%), second class (50% - 59%), third class(40% - 49%) and fail (<39%). Decision trees, Bayes network, Naive Bayes, k-nearest neighbors and rule learners were implemented. They found that *Bayes Network and Naive Bayes* performed the best out of all the algorithms, with KNN performing the worst overall.

A summary of the reviewed literature is presented in Table 4.1. In general, the literature shows that in studies on predicting student performance, ensemble methods, tree-based methods and neural networks perform the best overall. Literature unrelated to student performance has shown that random forest, boosting (adaboost, gradient boosting) and neural networks are currently performing the best in various empirical studies [21, 22, 23]. Except for the boosting models, this corresponds to the literature on predicting student performance. In Table 4.1, Acc. indicates accuracy; RMSE indicates root mean squared error; TP indicates true positives; Prec. indicates precision; AUC indicates area under the curve.

Table 4.1: ML in student performance. The models with an asterisk indicate best performer in each study.

Reference and Year	Sample size	Models	Classification/ Regression	Metric	Feature Selection
[95] 2004	354, 28	NB*, DT, NN, SMO, KNN, Log.R.	Class.	Acc.	No
[109] 2007	206	Lin.R., NN*, DT	Reg.	RMSE	No
[1] 2010	1347	Ensemble*, KNN	Class.	Acc.	No
[108] 2015	80	Lin.R., NN*, BN	Both	Acc.	Yes
[73] 2017	1169	Ensemble*, Lin.R., Log.R., RF*, KNN	Reg.	–	Yes
[111] 2017	509	DT*, Boost., Log.R., RI	Class.	Acc.	Yes
[2] 2015	–	DT, BN*, NB*, KNN	Class.	TP, Prec.	No
[96] 2015	–	NN*, DT, SVM, KNN, NB	Class.	Acc.	No
[4] 2017	1032	Ensemble*, Lasso	Both	RMSE, Acc. & AUC	No
[9] 2018	214	RF*, Lasso	Reg.	RMSE	Yes

4.2.2 Imbalanced Data

When datasets contain different classes of output, there is the need to address imbalanced data. Imbalanced data is when one class occurs more frequently than another. Addressing imbalance is crucial because imbalance may cause important cases to not be detected [112]. Applying metrics such as accuracy to highly imbalanced data might obtain very high accuracy, but not be very meaningful. Accuracy is sensitive to imbalanced data [54]. For example, if a disease occurs in 0.1% of the population, a model predicting that no-one has the disease (i.e. predicting the majority class) will have very high accuracy, but not necessarily detect the important cases. Instead of using accuracy, ROC-curve [112, 113, 114, 115, 116] and F-measure [112, 113, 114, 116] are perhaps the most recommended for imbalanced data, since they incorporate class distributions and are less likely to be affected by imbalanced distributions [113]. Cohen’s κ is another single metric that is good at handling imbalanced data problems.

In addition to using better metrics discussed above, other methods exist for addressing imbalance. The first is *resampling* of the data so that the classes are more balanced [116]. Resampling can either be carried out by random undersampling (RUS) or random oversampling

(ROS). ROS reduces the majority class by randomly sampling the majority class until a desired balance between the two classes is obtained. This ratio is commonly 50:50 but can be other ratios as well. This therefore reduces the original dataset S to S' where $|S'| < |S|$ and $S' \in S$ [117]. ROS generates more minority class samples until the desired ratio is obtained. ROS has the disadvantage of throwing away information but is able to train models in much less time due to the dataset being smaller. ROS has the benefit of not losing any information but requires longer training time. Another drawback of oversampling is overfitting since it reproduces the same observations in the dataset. In addition to these data resampling techniques, synthetic minority oversampling technique (SMOTE) is an oversampling technique that generates artificial minority class examples by extrapolating between existing minority examples [117]. This is different from ROS in that it generates new examples, rather than duplicating existing ones [116].

A second method for dealing with class imbalance is to use *ensemble* ML techniques, specifically boosting, as discussed earlier. Boosting is able to perform well on imbalanced data since it iteratively trains models on observations that were previously incorrectly predicted. The incorrectly predicted data tend to be the minority class, which is the class of interest.

Combinations of these two techniques have also been created, for example RUSBoost and SMOTEBoost combine the resampling techniques with adaboost [116]. When tested on 15 datasets, RUSBoost and SMOTEBoost have been shown to outperform adaboost, ROS and SMOTE techniques [116]. Although RUSBoost and SMOTEBoost performed equally well, it was concluded that RUSBoost is more beneficial due to reducing the size of the dataset and causing shorter training times. RUSBoost can be implemented in both *RStudio* and *Python* (using *Scikit-learn*).

4.3 Methodology

Neural networks, ensemble methods, tree-based methods (decision trees and random forest) and boosting were shown to perform the best, both in the literature on predicting student performance and other literature. These models were therefore used in this study.

However, there does not appear to be much rigor in the model selection process. That is, it appears that in the literature default hyperparameter values were used when comparing models and were often treated as black-box models. There is then the need for more detailed model selection via hyperparameter tuning to identify the best models. It is important to identify the best models using rigorous model selection techniques, and then use those models for evaluating feature importances. Furthermore, in the related work on predicting student

performance, there has been no known studies that address the issue of imbalance. In order for the subsequent chapters to be of value, it is crucial that a rigorous model selection process be carried out in addition to addressing the imbalance in the data.

The *overall approach* in this chapter was to select the best performing models by classifying At-Risk students. This is a classification problem: students are either At-Risk (failing) or not. To perform model selection the following was carried out:

- The model hyperparameter ranges for all the models were chosen based on Chapter 2. Lasso, random forest, gradient boosting, adaboost and artificial neural networks were used to classify At-Risk students.
- For all ML applications in this chapter, *Imbalanced-learn* and *Scikit-learn* libraries and modules were used in the Python programming language.
- The original dataset was split into an 80% training and 20% testing set, with stratification, to preserve the imbalance.
- Undersampling and oversampling techniques, as well as a hybrid technique called RUSBoost, were used when addressing imbalance.
- For each resampled dataset, model selection was carried out using ten-fold CV (Chapter 2.4).
- The best performing models were selected based on a range of criteria.

4.3.1 Model Hyperparameters

This section presents the models used in this study, together with the choice for their hyperparameters. The first model described is lasso.

4.3.1.1 Lasso - Logistic regression with $L1$ regularization

Since this is a classification problem, lasso was applied to logistic regression. The hyperparameter used to tune this model is a regularization parameter presented in Section 2.2.1 as λ . For lasso classification, the module *LogisticRegressionClassifier* was used. The regularization parameter, C , is the only tuning hyperparameter and is defined as the inverse of the λ value. The λ range was chosen from e^{-7} to e^1 , which translates to a C range from 0.367 to 1096.63.

4.3.1.2 Random Forest

The *Scikit-learn* module *RandomForestClassifier* was used for training random forest models [30]. Based on Section 2.2.2, for model selection, the ranges of values for each hyperparameter is shown in Table 4.2. In terms of number of trees, the literature showed that above 100 there is no significant improvement. Therefore a range from 10 to 100 trees was selected in increments of 10. However, much larger numbers of trees were also included here. The number of features for classification were as per the literature, i.e. $\sqrt{p}/2$, $\sqrt{p}$, $2\sqrt{p}$, but for completeness, p was also be included.

Table 4.2: Random forest classifier hyperparameter ranges for model selection.

Parameter	Range of values
Num. trees	10:100 in increments of 10; then 100, 500, 1000, 1500, 2000.
Num. features	$\sqrt{p}/2$, $\sqrt{p}$, $2\sqrt{p}$, p
Min. samples split	2, 4, 8
Min. samples leaf	1, 2, 4
Bootstrap	Yes, No

4.3.1.3 Gradient Boosting

Hyperparameter ranges for gradient boosting model selection are presented in Table 4.3. The *GradientBoostingClassifier* module from *Scikit-learn* was used.

Table 4.3: Gradient boosting parameter ranges for model selection.

Parameter	Range of values
Num. trees	100, 500, 1000, 1500, 2000, 3000
Learning rate	0.001:0.01, increments of 0.001
Subsample	0.1:0.9, increments of 0.1
Depth	4, 5, 6, 7, 8

4.3.1.4 Adaboost

The *Scikit-learn* module *AdaBoostClassifier* was used and only two tuning parameters are provided, number of trees and learning rate. The ranges of these parameters are presented in Table 4.4.

Table 4.4: AdaBoost parameter ranges for tuning.

Parameter	Range of values
Num. trees	100, 500, 1000, 1500, 2000, 3000
Learning rate	0.001:0.01, increments of 0.001

4.3.1.5 Artificial Neural Networks

Based on the literature, the following hyperparameter ranges were chosen for model selection of the ANN. The *MLPClassifier* module from *Scikit-learn* was used which refers to the multi-layer perceptron module.

Table 4.5: ANN parameter ranges for model selection.

Parameter	Range of values
Initial Learning rate	$10^{-6} : 10^{-1}$
Learning rate (schedule)	Constant, Adaptive, Invscaling
Activation	Identity, Logistic, Tanh, Relu
Hidden layers	1:3
Hidden layer sizes	10:50, 100
Momentum	0.5, 0.9, 0.99
Solver	Adam, SGD, LBFGS
Reg. param.	$10^{-6} : 10^{-1}$

4.3.2 Data Splitting

The entire dataset obtained from Chapter 3 was randomly split, with stratification, into 80% training and 20% hold-out (test) set. The split into 80% training (702 observations) and 20% testing(176 observations) was done using the *train_test_split* module in *Scikit-learn*, with stratification. That is, both the training and testing sets had the same distribution of positives (15.1%) and negatives (84.9%) as the entire set. The class frequencies are presented in Table 4.6. As shown here, failing of the course was considered to be a positive instance since that is what was being predicted

Table 4.6: Class frequencies for training and testing datasets. Stratification preserves the class imbalance.

Dataset	At-Risk (15.1%)	Not At-Risk (84.9%)	Total
Train (80%)	106	596	702
Test (20%)	27	149	176

4.3.3 Addressing Imbalanced Data

Before model selection was carried out, the data imbalance was addressed. As seen above, the At-Risk frequency is 15.1% which indicates a class imbalance. The total number of students for each class is shown in Table 4.7.

Table 4.7: Class frequency.

Class	Frequency [%]	Number
At-Risk	15.1	133
Not At-Risk	84.9	745

As discussed in Section 4.2.2, the use of accuracy is not the most appropriate metric for imbalanced datasets, since in this case, simply predicting everyone as not At-Risk will give an accuracy of 84.9%, which although looks impressive, might not help in identifying At-Risk students. Therefore the accuracy of 84.9% can serve as a baseline accuracy, which needs to be improved in the various models.

Also discussed in Section 4.2.2, resampling and ensemble methods can be used to deal with imbalanced datasets. *None of these resampling or hybrid techniques have been used in the literature predicting student performance.* This chapter addresses the data imbalance in five ways and all these methods used the *Imbalanced-learn* library [117] applied in Python.

1. Leave data as is, i.e. imbalanced. This dataset is called IMB.
2. Undersample using the RUS technique. For this, the module *imblearn.under_sampling.RandomUnderSampler* was used in Python.
3. Oversample using random oversampling, ROS. The module *imblearn.over_sampling.RandomOverSampler* was used.
4. Oversample using SMOTE. The module *imblearn.over_sampling.SMOTE* was used in Python.
5. Apply a hybrid technique using RUSBoost. Here the *imblearn.ensemble.RUSBoostClassifier* module was used in Python.

It is important to note that the resampling (undersampling, oversampling etc.) was performed only on the training set. This is because if resampling was performed on the entire dataset and *then* a train/test split was performed, the test datasets may include copies of data in the training set. This means the trained model would already have seen these observations in the test set which would lead to overfitting. According to Hastie et al. (2009) [15], feature selection is performed only on the training set after a train/test split. This means that the test set has not seen the manipulations of the training set. In the same way, manipulations such as resampling were carried out only on the training set and then all testing was carried out on the original same test set as per Table 4.6.

For each approach above, the models discussed in Section 4.3.1 were trained and tested.

4.3.4 CV for Model Selection

Ten-fold CV was carried out on the training set (original or resampled) in order to tune the various model hyperparameters for model selection. Once the hyperparameters were selected, the models were tested on the test set. CV is generally carried out on each hyperparameter combination. However this can lead to a very large number of model fits. For examples, for the random forest hyperparameters (Table 4.2) there are $14 \times 5 \times 3 \times 3 \times 2 = 1,260$ different combinations. And if for each combination, ten-fold CV is carried out, this would require *12,600* fits. The traditional way of obtaining optimal hyperparameters is to use a grid search method that searches through *every* possible combination. However, it has been shown that randomly selecting a subset of the combinations finds models as good or better than grid search, but in much less computational time [118, 52]. Therefore to reduce computation time, the *Scikit-learn* module *RandomizedSearchCV* was used and in general 100 random combinations of the hyperparameter space was selected. This reduced the number of fits to at most 1000.

4.3.5 Metrics Used

During model *selection* on the training set, AUC was used as the main metric. During model *testing* on the test set, accuracy, F1-score, AUC, Cohen’s Kappa and Brier’s score were used as measures. In addition to this, the confusion matrix was also calculated.

Model selection and testing model performance for each model was carried out on each dataset. For brevity, it is not included here but is presented in Appendix A. Only the final test performance is presented here and analyzed. The aim of this section was to compare the test performance of the various models on the various datasets.

4.3.6 Model Performance

Due to the need to tune hyperparameters on the 80% dataset (i.e. CV on the 80%), and test on 20%, it was not possible to perform training and testing multiple times in order to generate a range of metrics. The ideal would be to generate a range of metrics to compute mean and standard error values in order to obtain more confidence in the results. However, when the literature computes these mean and standard errors, it is based on already-selected hyperparameters. That is, they select hyperparameters for a model, and then run either ten-fold CV, or multiple training and testing, to generate the range of values. This study however needed to *first* carry out model selection using CV to obtain the hyperparameters. Then there was testing on the test set. If this study were to carry out multiple training and testing on different train/test sets each time, it would result a different set of hyperparameters each time, which would defeat the point of model selection on the 80% set. This means that it was not possible to generate a range of metric values and so provide means and standard errors to compute significance tests. This is a limitation in the study. And according to James et al. (2013) [20], different train/test splits produce high variance in the results, which suggests that the results obtained in testing might vary substantially for another test set. However, based on this limitation, it was assumed that for a specific dataset, e.g. Imbalanced Dataset (IMB), the models are tested on the same test set and therefore provide an *indication* of better performing models.

Due to there being different models as well as different datasets, model performance was analyzed first by model and then by dataset. The following was carried out:

1. first identify the best performing models overall (Section 4.4),
2. then compute the performance of each model, averaged over all the datasets (Section 4.5),
3. then compute the performance of each dataset, averaged over the five models 4.6.

4.4 Best Models Overall

The performance metrics for each model in each dataset is presented in Table 4.8. In the table the superscript *a* indicates the overall best performer; superscript *b* indicates the best performer within a dataset, and superscript *c* indicates the worst performer in a dataset. The models are labeled based on the dataset and model type. For example, IMB-RF is the random forest model trained on the original imbalanced dataset; RUS-ANN is the ANN model

trained on the RUS dataset. GB indicates gradient boosting, AdaB indicates adaboost. The IMB-ANN model was estimated to perform the best overall. This was determined by seeing that IMB-ANN performs best in accuracy, performs highly in AUC, performs second best in F1-score, highest in κ and second lowest in Brier score (lower is better for Brier score). ROS-RF and ROS-GB were estimated to perform slightly worse, but are also considered the overall better performers.

Table 4.8: Metrics for all the ML models trained and tested on all the datasets.

Model	Accuracy	AUC	F1-score	κ	Brier
IMB-Lasso	0.86	0.65	0.43	0.36	0.14
IMB-RF ^c	0.88	0.61	0.35	0.31	0.13
IMB-GB	0.88	0.65	0.45	0.39	0.13
IMB-AdaB	0.89	0.64	0.44	0.40	0.11
IMB-ANN ^{a,b}	0.89	0.69	0.51	0.49	0.12
RUS-Lasso	0.76	0.75	0.49	0.36	0.24
RUS-RF ^b	0.75	0.81	0.52	0.39	0.25
RUS-GB	0.77	0.77	0.51	0.39	0.23
RUS-AdaB ^c	0.71	0.71	0.43	0.27	0.29
RUS-ANN	0.79	0.75	0.49	0.37	0.22
ROS-Lasso	0.77	0.70	0.44	0.31	0.22
ROS-RF ^{a,b}	0.88	0.67	0.49	0.43	0.12
ROS-GB ^{a,b}	0.88	0.68	0.50	0.43	0.13
ROS-AdaB	0.76	0.70	0.44	0.30	0.24
ROS-ANN ^c	0.79	0.63	0.37	0.25	0.21
SMOTE-Lasso ^c	0.77	0.67	0.41	0.28	0.23
SMOTE-RF	0.85	0.59	0.32	0.25	0.15
SMOTE-GB	0.84	0.63	0.39	0.30	0.16
SMOTE-AdaB ^b	0.81	0.70	0.47	0.35	0.19
SMOTE-ANN	0.84	0.64	0.41	0.31	0.16
RUSBoost	0.77	0.70	0.44	0.30	0.23

4.5 Best Model Per Dataset

Looking at model performance within each dataset separately (Table 4.8), in the IMB dataset, the IMB-ANN performed the best and the IMB-RF performed the worst. This was determined by counting the number of times a model performs the best or worst. For the RUS dataset, RUS-ANN performed the best with RUS-AdaB the worst. For the ROS dataset, ROS-RF and ROS-GB perform the best with ROS-ANN the worst. For the SMOTE dataset, SMOTE-AdaB performed the best (although the accuracy was below baseline) with SMOTE-Lasso the worst. Finally, the hybrid model RUSBoost has comparable AUC with the top performers but much lower accuracy, F1-score, κ and higher Brier score. RUSBoost therefore does not perform as well as the top performing models.

4.6 Model Performance Over All Datasets

Here average model performance over all the datasets is looked at. The minimum, median and maximum values were also included. The *accuracy* of each model over all the datasets is looked at first in Table 4.9 and Figure 4.1. These results are also compared with the RUSBoost results shown previously in Table 4.8. First of all, the lasso (mean = 0.79) and AdaB (mean = 0.79) perform the worst overall. Furthermore, the AdaB range is the widest, suggesting it is sensitive to different datasets. RUSBoost similarly performed poorly with 0.77. The RF (mean = 0.84) and GB (mean = 0.84) models perform the best overall, with ANN having a slightly lower mean values (0.83). However, due to their low range, the ANN models perform the best overall. This suggests that in terms of accuracy, ANN is relatively insensitive to the different datasets.

Table 4.9: Summary of accuracy boxplots. Best mean values indicated by an asterisk.

Value	Lasso	RF	GB	AdaB	ANN
Min	0.76	0.75	0.77	0.71	0.79
Mean	0.79	0.84*	0.84*	0.79	0.83*
Median	0.77	0.87	0.86	0.79	0.82
Max	0.86	0.88	0.88	0.89	0.89

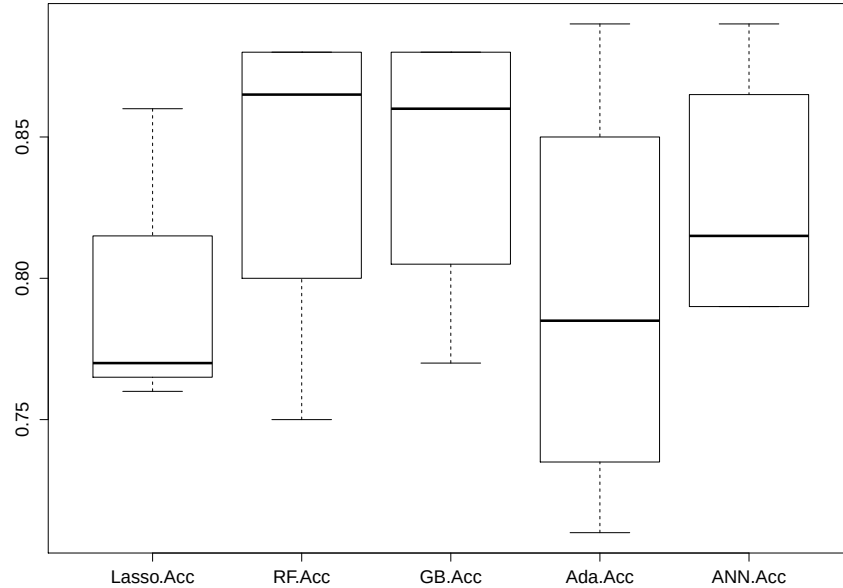


Figure 4.1: Accuracy boxplots of each model.

The *F1-score* was looked at next for each model, shown in Table 4.10 and Figure 4.2. Here GB (mean = 0.46) performs best over all datasets with ANN performing comparably (mean = 0.45). RF has the largest range, suggesting it is sensitive to the different datasets when measuring the F1-score. AdaB is shown to be least sensitive due to have the smallest range. This is different to what was found for accuracy, where RF was the most robust and AdaB most sensitive. The worst overall performer was lasso (mean = 0.44), although it too had a small range.

Table 4.10: Summary of F1-score boxplots.

Value	Lasso	RF	GB	AdaB	ANN
Min	0.41	0.32	0.39	0.43	0.37
Mean	0.44	0.42	0.46*	0.44	0.45*
Median	0.44	0.42	0.47	0.44	0.45
Max	0.49	0.52	0.51	0.47	0.51

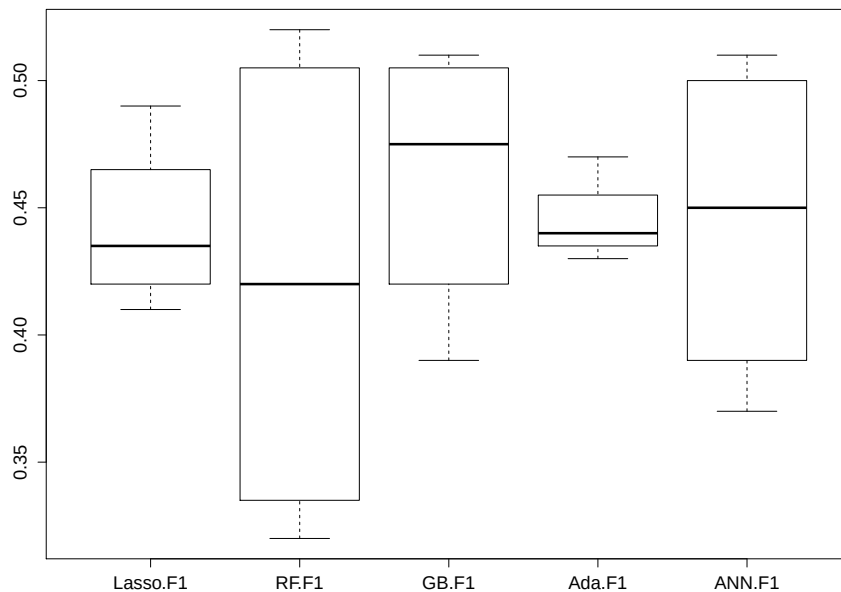


Figure 4.2: F1-score boxplots of each model.

The *AUC* values and boxplots are presented in Table 4.11 and Figure 4.3. As indicated before, it was not possible to perform multiple trains on the data in order to get a range of values. Therefore looking at the AUC results, it wasn't possible to see a clear indication of which model performed the best overall. For example, all the models have similar mean values. However, AdaB and lasso have the highest mean scores; and RF has the highest maximum score and largest range. RUSBoost scored an AUC of 0.70 which is higher than any mean value but not as high as the maximum values for RF and GB.

Table 4.11: Summary of AUC boxplots.

Value	Lasso	RF	GB	AdaB	ANN
Min	0.65	0.59	0.63	0.64	0.63
Mean	0.69*	0.67	0.68	0.69*	0.68
Median	0.69	0.64	0.67	0.70	0.67
Max	0.75	0.81	0.77	0.71	0.75

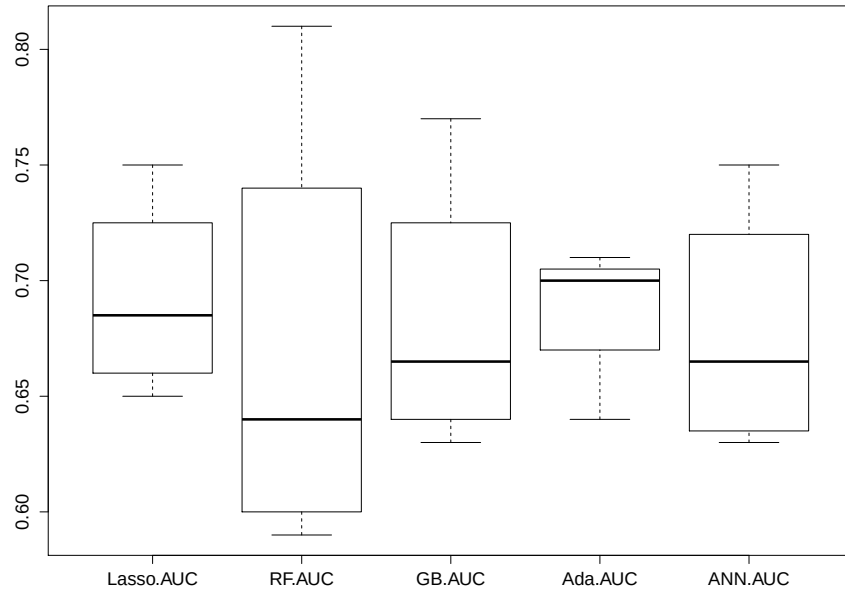


Figure 4.3: AUC boxplots of each model.

The κ performance was looked at next, for each model over all the datasets. GB (mean = 0.38) appears to perform the best since it has the highest mean and also the smallest range. Lasso performs the worst (mean = 0.33). ANN here has the largest range. RUSBoost scored 0.30 which is lower than any mean values.

Table 4.12: Summary of κ score boxplots.

Value	Lasso	RF	GB	AdaB	ANN
Min	0.28	0.25	0.30	0.27	0.25
Mean	0.33	0.35	0.38*	0.33	0.36
Median	0.33	0.35	0.39	0.33	0.34
Max	0.36	0.43	0.43	0.40	0.49

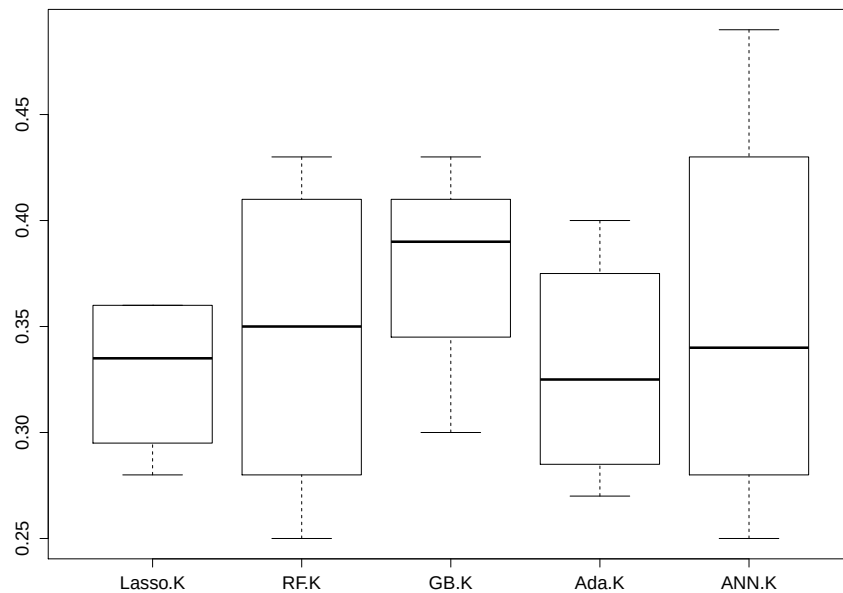


Figure 4.4: κ score boxplots of each model.

For *Brier score* lower values are better since the squared error between predicted probability and actual outcome is computed. RF (mean = 0.16) and GB (mean = 0.16) perform the best overall, with lasso (mean = 0.21) performing worst again. In this case RUSBoost (0.23) did not perform as well as the minimum mean value. The results show that AdaB has the largest range and best maximum value. ANN has the smallest range.

Table 4.13: Summary of Brier score boxplots.

Value	Lasso	RF	GB	AdaB	ANN
Min	0.14	0.12	0.13	0.11	0.12
Mean	0.21	0.16*	0.16*	0.21	0.18
Median	0.23	0.14	0.15	0.22	0.19
Max	0.24	0.25	0.23	0.29	0.22

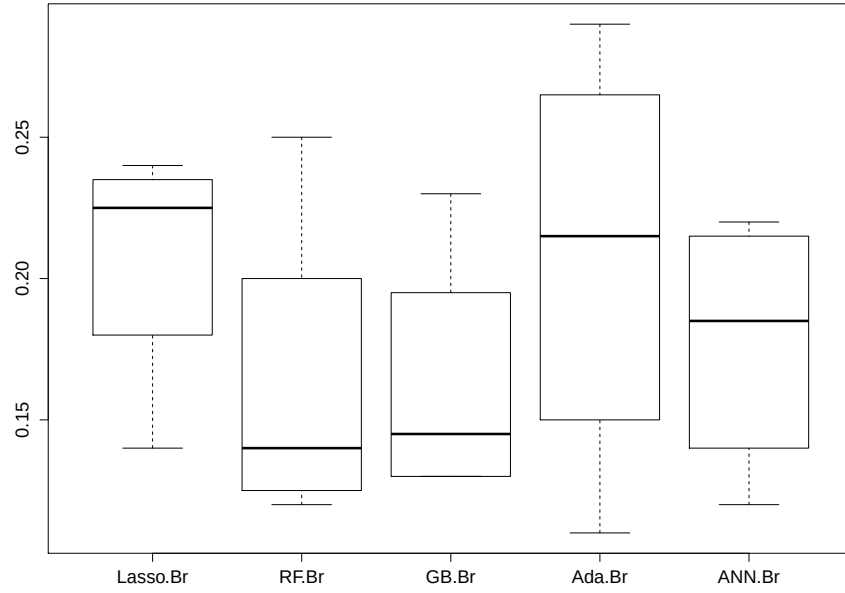


Figure 4.5: Brier score boxplots of each model.

The results suggest that averaged over all datasets, the best performers are *gradient boosting and neural networks*. These two models also were the best individual performers (IMB-ANN and ROS-GB). Lasso tends to perform the worst overall and this might be attributed to non-linearities in the data that a linear type model might not capture. For the majority of the cases, RUSBoost did not outperform the average model values.

4.7 Overall Performance of Datasets

Whereas the previous section analyzed the performance of each model over all the datasets, this section investigates the overall performance of each dataset by averaging each score, shown in Table 4.14.

Table 4.14: Average scores for all datasets.

Dataset	Accuracy	F1-score	AUC	κ score	Brier score
IMB	0.88	0.44	0.65	0.39	0.13
RUS	0.76	0.49	0.76	0.36	0.25
ROS	0.82	0.45	0.68	0.34	0.18
SMOTE	0.82	0.40	0.65	0.30	0.18

One of the aims was to see if resampling the data resulted in better overall model performance. Overall, the IMB performs the best with the RUS datasets performing second best, based on the number of times they score the highest in each category. However, only IMB has a mean accuracy above the baseline (or null accuracy) of 84.9%. Both undersampled datasets, RUS (0.76) and RUSBoost (0.82), had a reduction in accuracy. Random undersampling, although beneficial in terms of reducing computation time, can potentially throw away vital information of the majority class [116]. The results suggest that is what happened here and the accuracy has suffered as a result. However, the F1-score and AUC for RUS are all as good or better than the IMB dataset. ROS outperform IMB only in two metrics and SMOTE does worse than IMB in every metric. *Interestingly, the results suggest that the original dataset, IMB, is sufficient for training the models* and if resampling is to be done, then RUS is the next best choice (although the accuracy will suffer). That the IMB dataset performed well may be due the imbalance not being extreme (minority class = 15.1%) compared with cases such as fraud detection which is often below 1%. Furthermore, these results somewhat contradict those of Seiffert et al. (2010) who showed that SMOTE outperforms RUS most of the time [116]. In this case RUS tends to perform better than both oversampled sets three out of five times.

4.8 Conclusion

This chapter formed the first part of two chapters addressing the first two objectives of this thesis. This chapter obtained the best performing ML models while addressing imbalance in the data. The following chapter carries out the characterization of At-Risk students based on the models used in this study.

Due to imbalance in the data, lasso, RF, GB, adaboost and ANN were trained on various datasets comprising the original imbalanced set (IMB), a randomly undersampled set (RUS), a randomly oversampled set (RUS), a synthetically oversampled set (SMOTE) and a method using a hybrid technique (RUSBoost).

The following was found from this chapter:

1. IMB-ANN, ROS-GB and ROS-RF were the best individual performers and were subsequently used in Chapter 5.
2. Gradient boosting and neural networks performed the best over all datasets.
3. The original imbalanced dataset was sufficient for training and there is no need to resample. There being no need to resample may be attributed to the imbalance not being extreme.

The following chapter carries out detailed characterization of At-Risk students based on the best performing models obtained in this chapter.

Chapter 5 Characterizing At-Risk Students Using Model-Agnostic Methods - Model Explanations

5.1 Introduction

This chapter presents the second of two chapters addressing the first two objectives of this thesis. Chapter 4 obtained the best overall performing models that formed the basis for characterizing At-Risk students in this chapter.

ML is indeed valuable for predicting At-Risk students. However, interpreting or explaining these predictions helps to further understand why the predictions were made. These predictions are often made by complex models such as deep neural networks that are difficult to interpret. This chapter introduces model-agnostic methods used in other fields [6, 7, 8], that can explain any black-box model to gain insight into why students are At-Risk. These explanations can either be global (overall) or local (individual). Although *global* predictors (explanations) are helpful, it is becoming increasingly important to explain to individuals what features specifically resulted in receiving their outcome [5]. This can allow for tailor-made advice for the individual to understand what could be changed to receive a better outcome [5]. Model-agnostic methods are introduced to this field of predicting student performance and identifying At-Risk students.

The main contributions in this chapter are:

- Section 5.4 shows how global model-agnostic techniques such as permutation importances and partial-dependence plots were used to identify global features of At-Risk students.
- Section 5.5 presents how local model-agnostic techniques, LIME and SHAP, were used to identify features that affect individual students. These techniques are able to explain to individual students why they are At-Risk.

- Finally, in Section 5.6 counterfactual explanations for individual students were generated, showing how they could change from being At-Risk to being not At-Risk.

5.2 Background

In this background section, the related work on performing feature selection (model explanation) in the field of predicting student performance is first reviewed (Section 5.2.1). The theory on interpretable ML and model-agnostic methods is presented in Section 5.2.2. Thereafter counterfactual models and explanations (Section 5.2.3) are presented.

5.2.1 Identifying Predictors of At-Risk Students

Explaining ML predictions is becoming more important [7]. Although predictions are valuable, it is important to determine which features are impacting the predictions the most. With models such as regression and tree-based methods, this is carried out by estimating coefficients and feature importances, respectively. Therefore in terms of At-Risk students, these techniques can be used to identify features that characterize At-Risk students. Although much work has been done on *prediction* using regression and tree-based methods (see Table 4.1), there is little in terms of identifying predictors. Feature importances predicting At-Risk students have been implemented only in a couple of studies [69, 119].

More complex models such as neural networks pose challenges to interpreting feature importances. Therefore *model-agnostic* techniques (discussed in detail in Section 5.2.2) for identifying features provide an advantage since they can be applied to any black-box model. Model-agnostic techniques include global and local techniques. *Global* techniques such as partial dependence plots (PD-plot) and permutation importances (PI) [88] have been used sparingly in the literature. Mduma et al. (2019) [88] predicted dropout rates in schools and used PI to estimate top predictors. Others have used PI for feature selection to predict graduation rate and choice of major [70]. No literature is found that used PD-plots to analyze predictors of being At-Risk.

The use of *local* model-agnostic techniques, or local explanation models, for identifying features for individual students is in its' infancy. The major benefit of local methods is that they cater to individual students by identifying specific features that impact these individuals. Three main aims of explanation models have been proposed by Wachter et al. (2017) [5]. These are (1) to help the person understand why the model predicted their specific outcome,

(2) to provide grounds for the person to contest the decision and (3) to understand what could be changed in order to receive a different, more desirable, outcome.

This study introduces the LIME and SHAP local model-agnostic methods to the field of student performance. These methods have only been developed recently by Ribeiro et al. (2016) [7] and Lundberg and Lee (2017) [8] and open up an entire field for assessing individual students and catering to their needs. Thus far, LIME has been applied in a very brief way to predicting At-Risk students in an engineering program [120]. This literature suggests that further study is needed to see how LIME can be used for advising students. Others have used LIME for individual predictions to trace how different features impact a student in a massive open online course (MOOC) [121]. The literature is also scarce in terms of applying SHAP to predicting student performance [122]. The field therefore is wide open for further study and application in the field of predicting student performance.

Using these global and local techniques in combination will provide much insight into the features predicting At-Risk students both globally and individually. This area is therefore developed here.

5.2.2 Interpretable ML

Sections 5.2.1 presented the literature on explaining models using traditional and model-agnostic methods applied to the field of predicting student performance. This section presents the theory describing these methods.

Model interpretability is the degree to which a human can understand why a model made a certain prediction [6]. Two general techniques are presented that make up model interpretability, namely interpretable models and model-agnostic techniques. Furthermore, global and local feature importances are presented.

5.2.2.1 Interpretable Models

Interpretable models allow for relatively simple inference after model training, mainly due to being able to see the contribution of each feature to the output based on the feature weights.

5.2.2.1.1 Regression Models

Lasso, already discussed in Section 2.2.1, is a regularization technique applied to regression models such as linear regression (continuous) and logistic regression (classification). Regression models are *interpretable models* [6] since it is relatively easily to interpret via the β coefficients.

5.2.2.1.2 Feature Importances for Tree-Based Models

Interpretable models further include tree-based models (random forest, boosting etc.) discussed earlier. With tree-based methods, feature importances (FI) can be computed using the *mean decrease in impurity* [20, 26, 123]. The greater the decrease in impurity when splitting on a feature, the greater the importance of that feature. For a single tree, feature importance is decided by how much the impurity decreases after splitting. The larger the decrease, the higher the feature importance. For tree-based *ensembles* such as random forests and boosted trees, the decrease in impurity for each feature is added up for each tree and averaged over the number of trees. A disadvantage of using FI is that it is biased toward continuous features and those with a large number of categories [26, 124]. Another disadvantage of FI is that it only provides a positive value for the importance; it does not provide direction. That is, it is possible to tell the relative importance of a feature, but not if it has a positive or negative correlation with the outcome. A benefit of lasso β coefficients is that they provide direction, i.e. they indicate whether a feature is positively or negatively associated with the target. Therefore FI and lasso β coefficients can be used together, i.e. the feature importances can be calculated using the tree-based methods, and the lasso coefficients can indicate if those features are positively or negatively correlated with the target.

5.2.2.2 Global Model-Agnostic Techniques

The above two techniques are readily interpretable. There are however, other techniques called *model-agnostic methods* that allow for explainability of black-box models [6, 7]. For example, for an ANN trained with multiple layers, neurons and connections, it becomes very difficult to determine the importance of each feature in the network. Model-agnostic techniques can solve this problem.

The essential idea of model-agnostic methods is to train *any model* on training data, then *perturb* the inputs in some fashion, and to evaluate the outputs [7]. The following discusses these model-agnostic perturbation techniques that are used in this study. Sections 5.2.2.2.1 and 5.2.2.2.2 present *global* methods and Section 5.2.2.3 presents local agnostic methods.

5.2.2.2.1 Permutation Importances

PI [26], also known as *mean decrease accuracy* [123], is a technique where model performance, for example accuracy, is measured before and after randomly permutating the feature space. For example, a model is trained on training data and the accuracy is measured on out-of-sample data (the out-of-bag error for RF). A single feature column is then randomly shuffled, leaving all other feature columns and target column unchanged. The accuracy is measured again and compared with the original accuracy. How much the model performance deteriorated indicates how important the feature is. This is because the more important a feature is in predicting the outcome, the worse the model will perform after shuffling that feature [26]. This is repeated for all features and the change in accuracy is measured for all features. This technique is said to correct the bias introduced by feature importances [125], discussed in Section 5.2.2.1.2. Disadvantages of PI are that it overestimates the importance of correlated predictors [123, 126] and does not have directionality. A module called *Permutation Importance* by *ELI5*ⁱ [127], implemented in *Python*, is able to perform PI on any black-box estimator and therefore can be applied to any of the machine learners in this study.

5.2.2.2.2 Partial Dependence Plots

A PD-plot is a further global model-agnostic technique that computes the marginal effect of one feature (or more) on the target outcome [6]. The marginal effect refers to the effect a feature has on the outcome by changing that feature, while keeping other features fixed. The difference between PD-plots and FI is that partial dependence perturbs the feature over a range of values, from low to high, while FI shuffles the feature. PD-plot allow for seeing whether the feature has a linear, monotonic or non-linear relationship with the output. For linear regression, the partial dependence will be linear, but PD-plots allow for capturing non-linear effects as well. PD-plots are typically implemented after the most important features have been identified [15]. Consider features X_s for which PD-plots are to be computed and features $X_c = x_{1c}, x_{2c}, \dots, x_{Nc}$, which remain unchanged. A range is selected over which X_s is varied, and then for each value within this range the output, $\bar{f}_s(X_s)$ is evaluated, shown in Equation 5.1.

$$\bar{f}_s(X_s) = \frac{1}{N} \sum_{i=1}^N f(X_s, x_{ic}), \quad (5.1)$$

ⁱELI5 is a Python library which allows to visualize and debug various ML models using unified API. It has built-in support for several ML frameworks and provides a way to explain black-box models.

5.2.2.3 Local Model-Agnostic Techniques

Recently, local model-agnostic techniques have been developed [7, 8]. The techniques above all describe global techniques that are able to provide an overall importance of the features. There is however a need to focus also on individual observations, for example, needing to explain to a client exactly why *they* did not receive a loan. Or to explain to a student why he or she specifically is considered At-Risk. The global techniques are valuable for describing global, or overall behavior, but they are not able to describe why a specific observation received a certain prediction. The two methods, LIME and SHAP, were developed to focus on individual observations. If an At-Risk student is identified, these methods are then able to provide tailor-made advice for that student. The intuition behind these local model techniques is that they replace the complex model (i.e. ANN or RF) with a *local linear model* that is interpretable locally.

5.2.2.3.1 LIME

LIME stands for Local Interpretable Model-Agnostic Explanation developed by Ribeiro et al. (2016) [7]. This forms part of local surrogate models that are used to explain individual predictions [6]. As the name suggests, it fits local models, i.e. models that fit single or multiple similar observations. For example, a specific observation (student) that was predicted to be At-Risk might benefit from understanding which features are most contributing to their At-Risk status. This is different from identifying which global features affect At-Risk students.

For a selected observation, LIME generates synthetic instances in the vicinity of that observation and new predictions via the original model are generated for those synthetic points. Then on this new local dataset, LIME trains a local interpretable explanation model, such as lasso or a decision tree. This model is required to be locally but not globally faithful to the original black-box model.

The formal definition is shown in Equation 5.2. The local explanation model ξ is the model g , that for a given instance x , minimizes the cost function $\mathcal{L}$. This is shown in Equation 5.2. f is the original complex model (e.g. neural network); π indicates the proximity of the synthetic instances generated near the instance, x ; and Ω is a constraint on g . Therefore this method finds the explanation model that minimizes the difference between the predictions of g and f based on the synthetic instances. This is therefore an optimization problem and $\mathcal{L}$ is to be minimized based on varying g .

$$\xi(x) = \underset{g \in G}{\operatorname{argmin}} \mathcal{L}(f, g, \pi) + \Omega(g) \quad (5.2)$$

LIME practically works as follows:

1. Choose an observation (local instance), defined as x .
2. Perturb the data in a region (defined by π) around this instance, and then using the originally trained model f (e.g. neural network) make predictions $f(x)$.
3. Weight new samples by their proximity to the local instance.
4. Fit a new interpretable sparse linear explanation model g on the new data set, by minimizing $\mathcal{L}$.

Explaining a local prediction is then possible by interpreting the local explanation model [6].

5.2.2.3.2 SHAP

SHAP (SHapley Additive exPlanations), similar to LIME, computes the importance of each feature on a local level. It aims to explain a prediction of a single instance by computing the contribution of each feature to the prediction [6]. Lundberg et al. (2017) [8] define an additive feature attribution method (Equation 5.3) where the explanation model g is a linear function of binary variables.

$$g(z') = \phi_0 + \sum_{i=1}^P \phi_i z'_i \quad (5.3)$$

Here $z' \in (0,1)^P$ and P is the number of input features in the explanation model. Therefore this method attributes an effect ϕ_i to each feature and then sums up the effects of all feature attributions to approximate the output $f(x)$. The aim therefore with both LIME and SHAP, is to obtain the explanation function in order to estimate the effects, ϕ , which are the local feature importances. These effects are akin to regression coefficients.

However, the reason for developing the SHAP method is that it guarantees the properties *local accuracy*, *consistency* and *missingness* [6], derived from game theory. Local accuracy means that when approximating the original model f for a specific instance x , the explanation model must match the output of f for simplified inputs x' . These simplified inputs are just a simpler feature space used to train the explanation function. Consistency says that if due to a model changing, some simplified input's contribution increases or stays the same regardless of other inputs, the effect of that input should not decrease. The final property, missingness, says that if a feature is missing, then it's attribute is zero. Based on guaranteeing these

properties, it guarantees that the attributes, ϕ , are fairly distributed among the features, which LIME cannot guarantee [6].

Consider the idea of fair distribution. SHAP is based on Shapley values in game theory [128] and looks at how each player (feature) in a game (model) contributes to the outcome [8]. A Shapley value for a specific feature is basically the difference between a prediction with that feature and without that feature. However, this method computes this difference for all possible subsets, sums up these differences and calculates the mean. For example, if a dataset has three features, say F_1 , F_2 and F_3 , and the Shapley value for F_1 is to be computed, then the difference in predictions for all possible subset combinations is first computed. That is, predictions are computed for just F_2 , then F_1 is added to F_2 and the prediction and difference are computed. Then for F_2 and F_3 , the predictions are computed, F_1 is added, prediction and difference are computed again. Then finally, prediction for F_3 is computed, F_1 is added and prediction and difference are computed. The difference in predictions between F_1 and all possible subsets have now calculated. The average of these predictions is the Shapley value for F_1 . It presents fair distribution of attributes since it looks at how that feature impacts the prediction based on *all possible feature subsets*.

Formally, Shapley values are calculated as follows:

$$\phi_i = \sum_{S \subseteq F/i} \frac{|S|!(|F| - |S| - 1)!}{|F|!} [f_{S \cup i}(x_{S \cup i}) - f_S(x_S)] \quad (5.4)$$

where S is a subset of all features, F is the set of all features, $f_{S \cup i}$ is a model trained with the feature, $f_S(x_S)$ is a model trained without the feature.

5.2.3 Counterfactual Explanations

Once features for an individual student have been identified, i.e. the model has been explained, ideally the next step is to provide advise to an educator on how the student could change their At-Risk status. As discussed above, the aim of explanations is for the student to understand why they are At-Risk and what could be changed to alter the result [5]. A method of doing this is to generate counterfactual explanations that can be applied to individual observations [6]. A counterfactual explanation describes a “what if” scenario where an outcome could be predicted, had the input features been varied, or perturbed. For example, what would the value of a specific feature need to be in order to switch the student from At-Risk to *not* At-Risk? Or what would a client need to change in order to be granted a bank loan? Counterfactual models can be any black-box model [6] and according to Pearl and Mackenzie

(2018), a simple model such as Hooke’s Law can be considered a counterfactual model [129]. For example, if Hooke’s Law is defined as $F = kX$, where F is the spring force, k is the spring constant and X is the spring deformation, then the question can be asked: *what if* a force F is applied? What would the deformation X be? This establishes a “what if” scenario, a counterfactual, using a simple equation. Wachter et al. (2017) [5] argue that counterfactual explanations may be of more value than a local explanation model, since it helps the subject to not only know why they have their specific outcome, but to have potential ways to change their outcome.

There are however, some limitations to this method. For example, the features in the black-box model do not necessarily have a causal relationship with the outcome. It merely shows correlation. Therefore, although the input feature space can be perturbed and a different output can be obtained, further analysis is required to ascertain if that feature is directly or indirectly changing the output. Another limitation is the idea of constraints. In the Hooke’s Law example, if larger and larger forces are applied, eventually the spring will break or permanently deform. There is the need therefore of constraints in the counterfactual models. Counterfactuals need to be generated from a “possible world” [5]. These issues need to be addressed when applying this to the data.

Counterfactual explanations are valuable in estimating changes that a student can make to change their outcome. No literature has been found utilizing this technique in predicting student performance and it is considered a valuable method in assisting individual students.

5.3 Methodology

Chapter 4 selected the best performing models, IMB-ANN, ROS-RF and ROS-GB, which were used when performing feature selection in this chapter. A topic that is not covered much in the literature is whether to estimate feature importances on the training or testing data [6]. Hastie et al. (2009) [15] show that feature selection, i.e. identifying and then reducing features, should only be done on the training set and then *that* feature-selected model should be tested on the test set; i.e. the train/test split should be done before any manipulation. However, this study is not trying to modify the models through changing the features. It is only trying to identify the strongest predictors. Molnar (2019) [6] provides arguments for carrying out feature importances on both training and testing sets, and states that there is no conclusive literature arguing which is best. Machado et al. (2015) [130] investigated feature importances and PD-plots using only the training data and then reduced the model for testing on the test set. Due to the uncertainty in the literature, this study

investigated feature importances and permutation importances on the test set, since this closely matches the practice of trusting the test results more than the training results [6].

5.3.1 Global Techniques

With regard to the global methods, FI was only computed for the ROS-GB and ROS-RF models, since FI only works on tree-based models in *Scikit-learn*; but PI and PD-plots were applied to all three models.

Furthermore, each time a model was fit to the test data, there were differences due to the random nature of the models. For example, random forest chooses different random samples each time it fits to the data which results in slightly different feature importances each time. Similarly, PI generates random permutations of the features each time. Therefore due to this random nature, the following was carried out for FI and PI based on suggestions in the literature [6, 131]:

1. Fit model (IMB-ANN, ROS-RF, ROS-GB) on the test set.
2. Compute FI and PI and store the results for that round of fits.
3. Repeat 100 times. Molnar (2019) [6] simply suggests repeating the permutations and averaging the results in order to stabilize the measure but does not provide a specific method for determining the number of repetitions. In probability theory, the law of large numbers states the mean of the results obtained from a large number of experiments should approximate the expected value. Kuhn and Johnson (2013) suggest between 50 and 200 repetitions [131]. Due to computational resources, this number was chosen as 100.
4. Compute the mean value for each feature.

For all FI computations, the tree-based models were fit and then a *feature_importance_* function was applied to obtain the FI results.

For all PI computations, the package *eli5.sklearn.PermutationImportance* was used in Python. ELI5 is a Python package that helps to explain predictions of classifiers [132].

Although not part of the best performing models, the IMB-Lasso model was also utilized, simply to obtain directions for features via the β coefficients. The FI and PI results do not provide direction and therefore IMB-Lasso provided these.

Finally, PD-plots were computed primarily for continuous features, namely the PHYS1015, PHYS1014 and MATH1014 March tests, high school mathematics, and years since matriculation, to ascertain the marginal effects of these features with on the target outcome. The *Scikit-learn* module *sklearn.inspection.partial_dependence* was used for computing and plotting all PD-plots.

5.3.2 Local Techniques

In terms of the local techniques, LIME and SHAP, these were applied to all the models, since these are model-agnostic methods. The *LIME* package [133] was used to apply all LIME applications using the *lime.lime_tabular.LimeTabularExplainer* function. For all SHAP applications, the SHAP library was used [134].

5.3.3 Counterfactual Models

An important final step in this chapter was to provide the instructor with ways to advise the students on how to switch from being At-Risk to being not At-Risk. Counterfactual models were generated by perturbing the March test features for PHYS1015, PHYS1014 and MATH1014, as well as for ADU attendance and residence status. For the continuous features, perturbations were between -2 and 2 standard deviations, with the aim of ascertaining when the At-Risk probability dropped below 50%.

5.4 Global Explanations

5.4.1 Feature Importances (FI)

FI values were computed on the test set, based on the ROS-RF and ROS-GB models. The normalized feature importances were computed for 100 fits of the two models and then mean values were calculated. The results are plotted in Figures 5.1 and 5.2. First the two feature importance results based on the two models, are practically the same. They are in the same order and have similar mean values. The top predictor is March test PHYS1015 which is expected since the final PHYS1015 grade is being predicted. That is, it is expected that the March test earlier in the semester be a predictor of the final results. The next best predictors are the March tests for MATH1014 and PHYS1014. March test MATH1014 is shown to be a strong predictor (0.82) for PHYS1015. This shows that those who perform well in mathematics tend to perform well in the mechanics course. Furthermore, the top five

features are mainly grades features with one demographic feature, years since matriculation (Years.Since.Matric). This suggests that the age of the student is also important to predicting At-Risk students. After years since matriculation, there is a large decrease in importance (0.31 to 0.14) and then a gentle taper towards zero. This suggests that grades data are the top predictors and that the demographic features play a minor role. If the student is in residence, does not have a bursary, is a first year or repeat and gender are the next strongest predictors. These are all demographic features. However, these results are on a global scale and require more detailed view of these features.

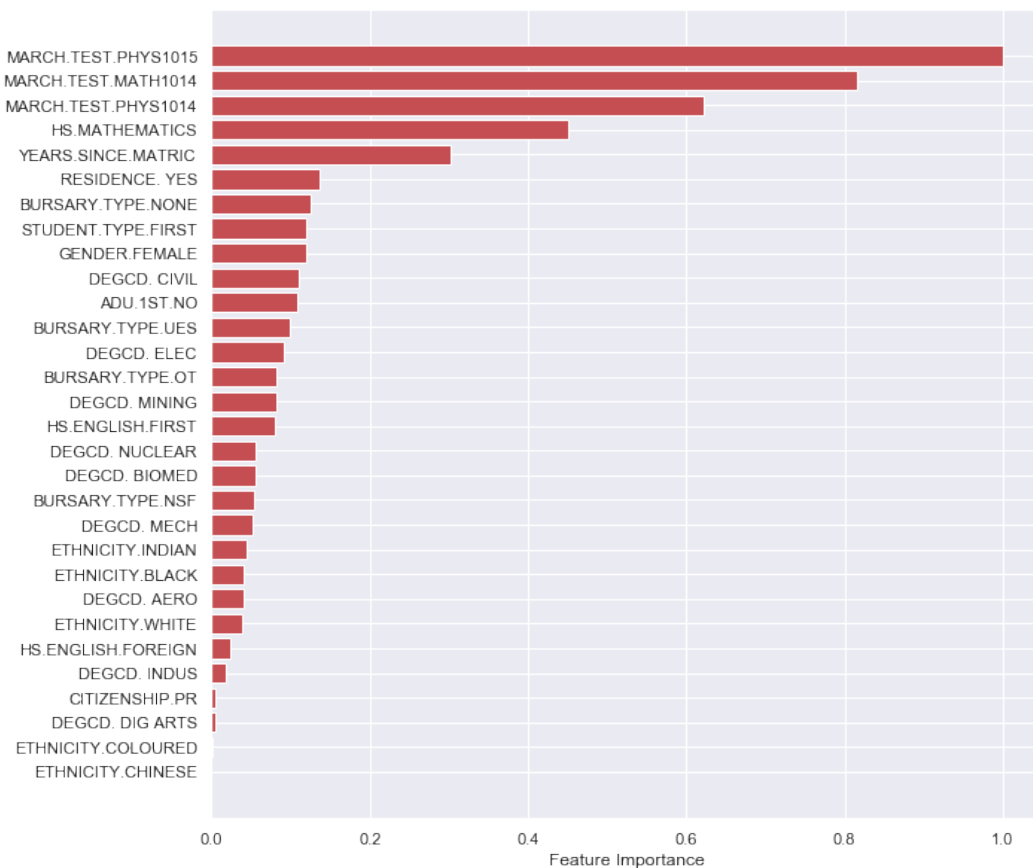


Figure 5.1: Mean feature importances for ROS-RF based on 100 fits to the test set.

As discussed in Section 5.2.2.1.2 there are a couple of shortcomings to FI. First is that the reduction in Gini impurity prefers continuous variables [70, 125]. Both models (ROS-RF and ROS-GB) generate continuous features as the top predictors. Are those the actual strongest predictors or a limitation in FI? To attempt to correct this potential bias, PI was implemented as discussed in Section 5.2.2.2.1 [125]. Second, FI does not indicate direction, i.e. whether the features are positive or negative predictors of being At-Risk. There was therefore the need to investigate the direction as well by computing lasso β coefficients and PD-Plots.

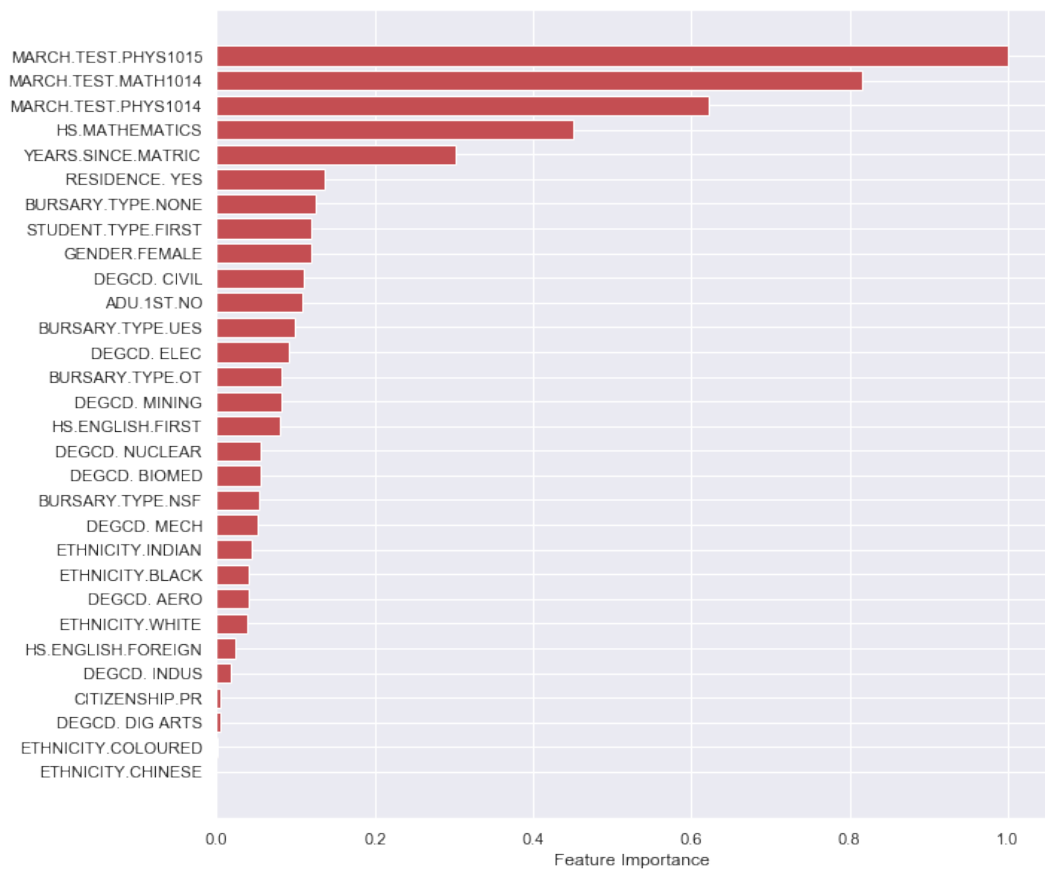


Figure 5.2: Mean feature importances for ROS-GB based on 100 fits to the test set.

5.4.2 Permutation Importances (PI)

Since PI is a model-agnostic technique, it was applied to all three models. PI for the ROS-RF is first presented in Figure 5.3. Note that there are negative PI values which mean that introducing random permutations of a feature actually result in better performance. This simply means that substituting the feature with noise is better than the original feature implying that these features are not important. Next, the top three features are the same as those for FI, but *here* March test PHYS1014 is swapped with March test MATH1014. In this case, attending the ADU, being a first year or repeat student and being a nuclear engineering student are featuring highly.

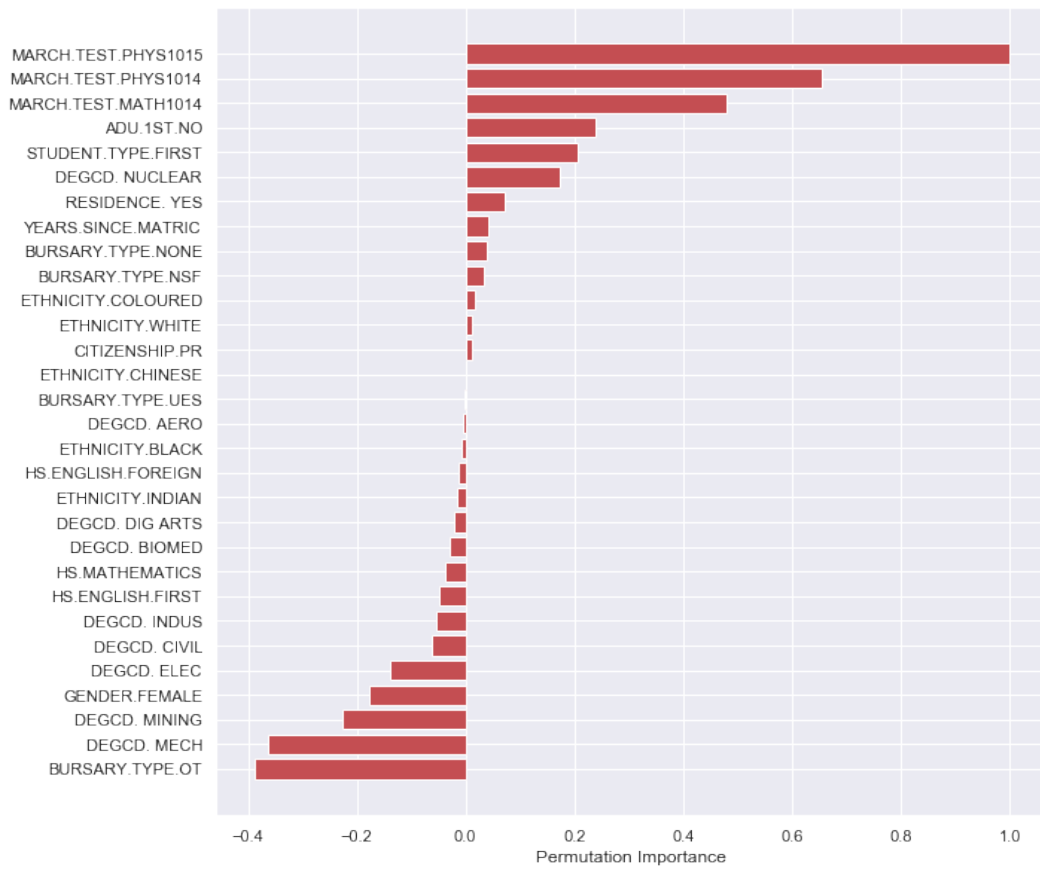


Figure 5.3: Mean permutation importances computed for ROS-RF based on 100 fits on the test set.

Figure 5.4 shows mean PI values for ROS-GB and generates similar results to that of ROS-RF. This is understandable since they are both tree-based models. However, type of student (Student.Type.First) and attending ADU (ADU.1st.No) are swapped around and here gender (Female) is considered important.

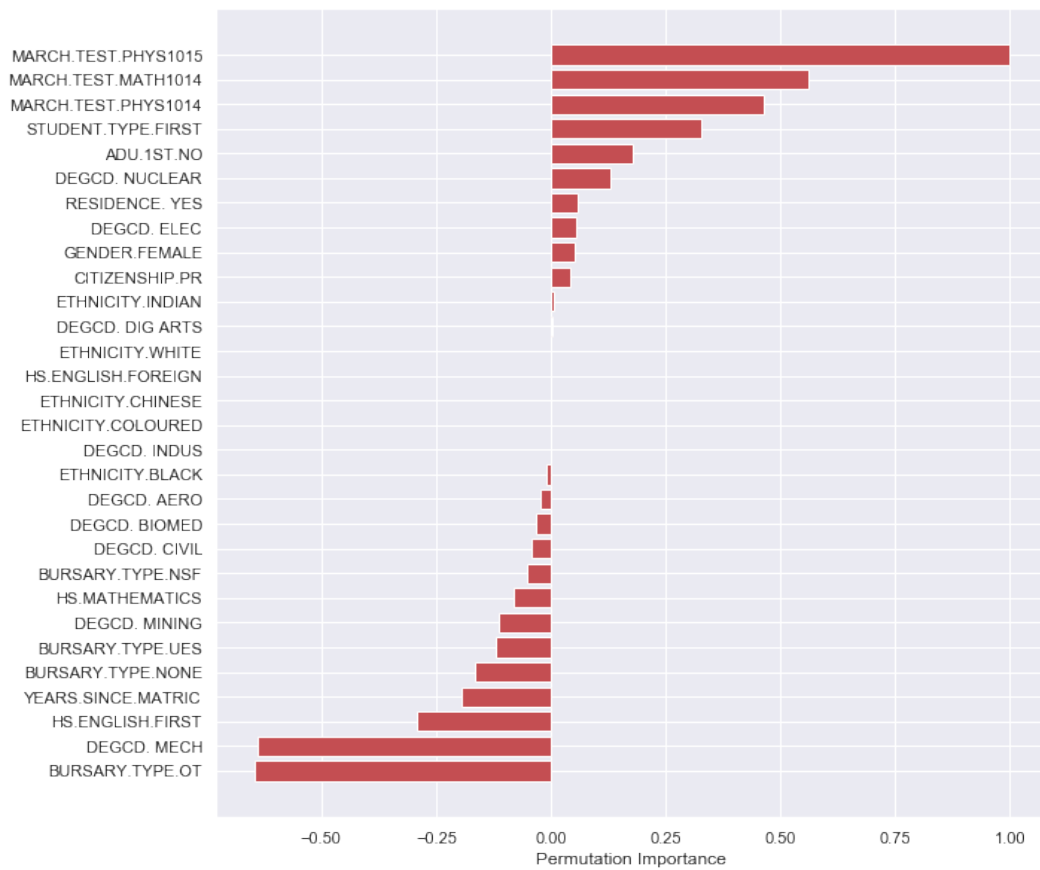


Figure 5.4: Mean permutation importances computed for ROS-GB based on 100 fits on the test set.

The final PI was computed for IMB-ANN fit to the test data, shown in Figure 5.5. Again March test PHYS1015 is first but here years since graduation (Years.Since.Matric) is second and March test MATH1014 is third. ADU attendance (ADU.1st.No) appears again and ethnicity (Indian) appears for the first time.

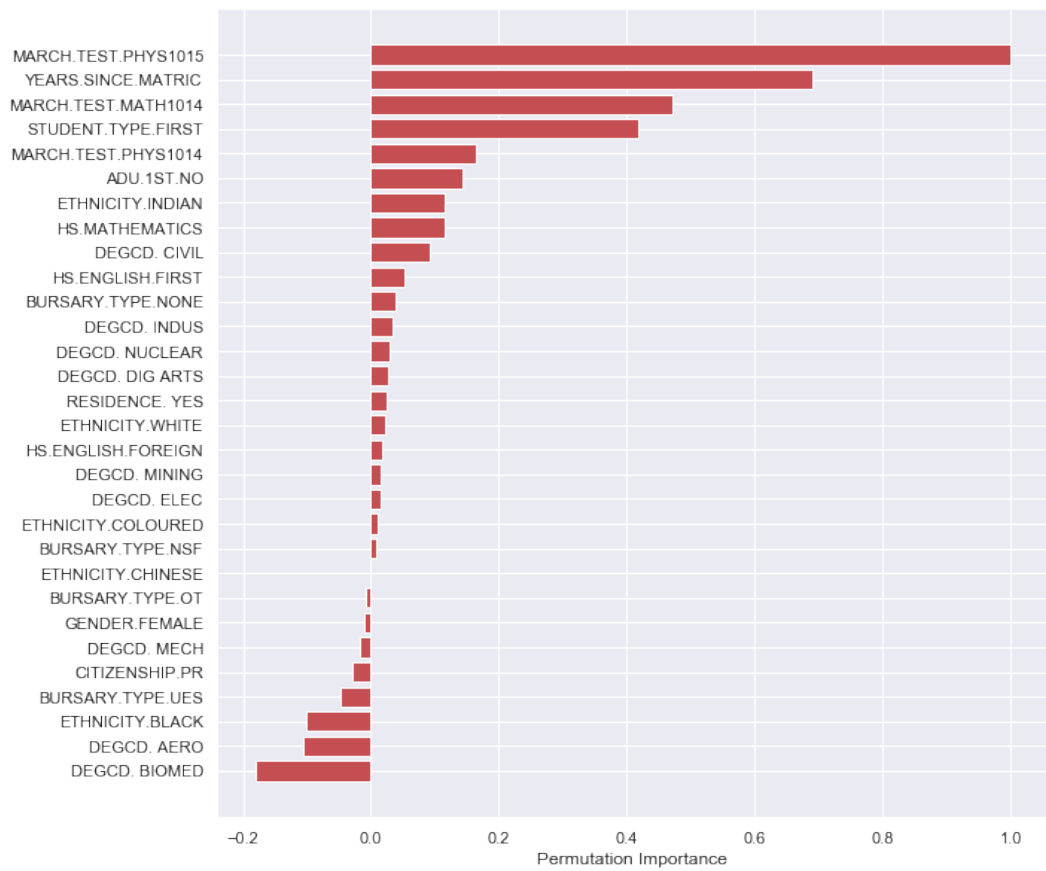


Figure 5.5: Mean permutation importances computed for IMB-ANN based on 100 fits on the test set.

Based on the above results, different techniques (FI and PI) and different models produce sometimes drastically different importance values. In Table 5.1 the top six features for each computation is presented. The three March tests appear in all, strongly suggesting that these indeed are the top predictors and global characteristics of At-Risk students. However, the remaining features appear in some and not in others.

Table 5.1: Top six features for each computation.

FI-RF	FI-GB	PI-RF	PI-GB	PI-ANN
March.Test.P1015	March.Test.P1015	March.Test.P1015	March.Test.P1015	March.Test.P1015
March.Test.M1014	March.Test.M1014	March.Test.P1014	March.Test.M1014	Years.Since.Matric
March.Test.P1014	March.Test.P1014	March.Test.M1014	March.Test.P1014	March.Test.M1014
HS.Mathematics	HS.Mathematics	ADU.1st.No	Student.Type.First	Student.Type.First
Years.Since.Matric	Years.Since.Matric	Student.Type.First	ADU.1st.No	March.Test.P1014
Residence.Yes	Residence.Yes	Degcd.Nuclear	Degcd.Nuclear	ADU.1st.No

5.4.3 Lasso β Coefficients

As discussed above, feature direction needs to be identified using β coefficients. The IMB-Lasso model was chosen to generate the lasso β coefficients, shown in Figure 5.6. The green bars indicate positive coefficients and the red bars, negative coefficients. It is important to note that the features in Table 5.1 and the β coefficients computed here indicate correlation with being At-Risk, and are not necessarily causal effects of being At-Risk. Based on the top features presented in Table 5.1, the three March tests all have negative coefficients. That is, these grades features are negatively correlated with being At-Risk. The negative coefficients indicate that performing better in the March tests are correlated with a lower chance of being At-Risk. This confirms our intuition. However, for high school mathematics (HS.Mathematics), there is a positive coefficient (although small relative to the other features) indicating a positive correlation with being At-Risk. This is a counterintuitive result since it would be expected that performing better in high school mathematics would be correlated with lower chance of being At-Risk. Years since matric (Years.Since.Matric) appears three times in Table 5.1 and here it has a negative coefficient. This suggests that the older the student is, the less chance of being At-Risk. Being in residence (Residence.Yes) has a positive effect on being At-Risk. Not attending ADU (ADU.1st.No) also has a positive effect on being At-Risk. This suggests that attending the ADU will decrease the students' chances of failing. Being a first year (Student.Type.First) is negatively correlated with being At-Risk suggesting that being a repeat student is a predictor of being At-Risk. Finally, being a Nuclear Engineering student appears to be strongly negatively correlated with being At-Risk. The university entrance requirements for nuclear engineering students is generally very high, suggesting that these are some of the higher performing students and so being At-Risk is low.

The above described how to combine FI, PI and β coefficients to characterize At-Risk students on a global scale.

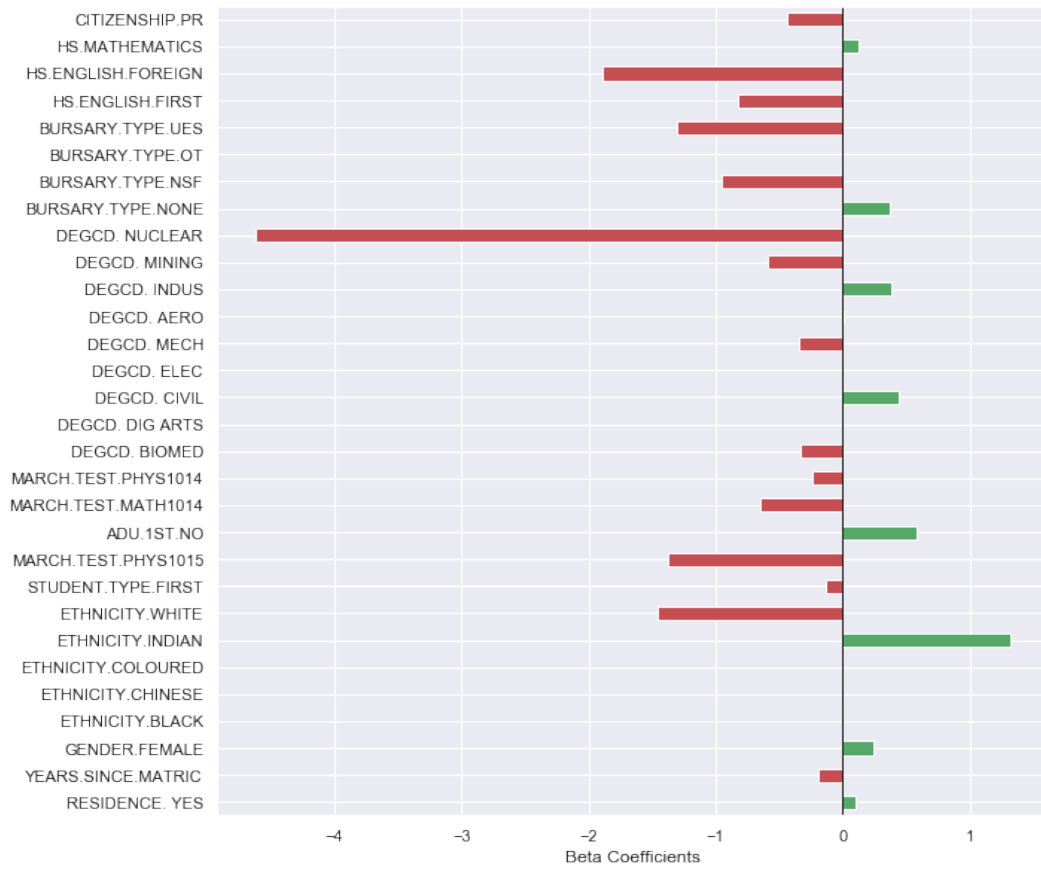


Figure 5.6: β coefficients computed for IMB-Lasso.

5.4.4 Partial Dependence Plots

Although the β coefficients provide an indication of the direction of the features, they only provide linear information. PD-plots, however, further allow for computing non-linear marginal effects. This provides an even deeper understanding of how each feature affects the output when changed.

PD-plots are interpreted here in a twofold way. The first is to describe the shape of the plot and the implications thereof. The second is to compare the plot with the global At-Risk probability of 15.1% (Table 4.6). Since global features are considered here, these are compared with the At-Risk probability for the entire group. Regions where the probability was higher than 15.1% were regions for identifying At-Risk students.

Figure 5.7 presents PD-plots for March test PHYS1015 (March.Test.PHYS1015) [Normalized] using ANN (top), GB (middle) and RF (bottom). The horizontal axis is the range of normalized grades and the vertical axis is the predicted probability of being At-Risk. According to Equation 3.1, the normalized grades indicate the number of standard deviations from the mean. In this analysis, the standard deviation of the grades was presented, together with the actual grade in parentheses. Beginning with the shape, all three models (Figure 5.7) show that as the grade increases for this course, there is a general decrease in At-Risk probability. This result simply confirms the β coefficients computed above. This suggests that the better the students perform in this test, the less chance there is of being At-Risk. Comparing with the global probability, for the ANN plot, any value less than -0.5 (65.6%) predicts At-Risk probabilities higher than 15.1%, i.e. higher than the expected probability of the entire cohort. For the GB and RF plots, grades less than approximately -1 (58%) result in higher At-Risk probabilities than the global expected value. Therefore for March test PHYS1015, students with grades of less than approximately 60% could be considered At-Risk of failing at the end of the semester.

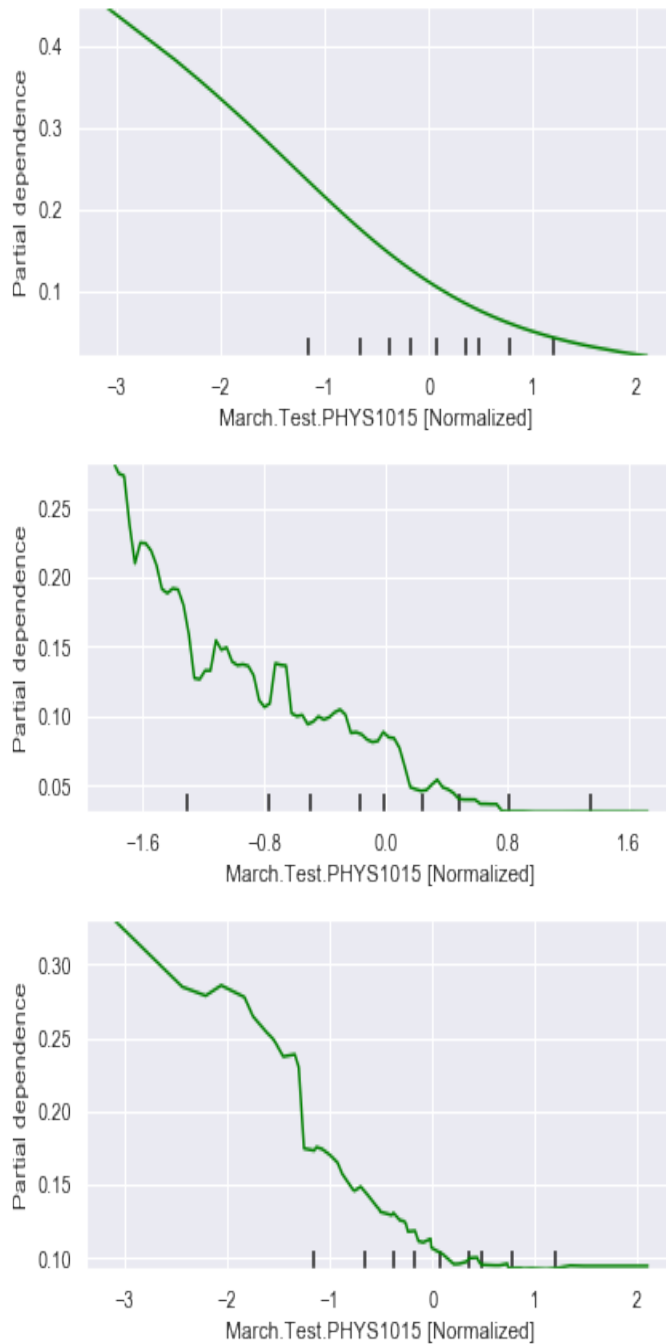


Figure 5.7: PD-plots for March test PHYS1015 [Normalized] generated by ANN (top), GB (middle) and RF (bottom).

PD-plots are shown for March test MATH1014 (March.Test.MATH1014) [Normalized] in Figure 5.8. In terms of the shape, there is a general decrease in being At-Risk as the grades improve. The ANN model is generally always smoother than the two tree-based models. For the GB and RF there is a relatively flat region from -2 to -1 (33%) which suggests any grades in this region do not affect the probability which remains in the region of 0.15 to 0.2. Then there is a negative slope between -1 (33%) until approximately 0.1 (58%) which suggests that

any improvement in this region are correlated with a decrease in At-Risk probability. For any grades above 0.1 we see the probability drop off completely and become zero, suggesting students above 0.1 have a zero probability of being At-Risk. Comparing with the global probability, grades below -0.4 (41.3%) on the ANN plot and approximately -1 on the GB and RF plots show At-Risk probabilities higher than the global value.

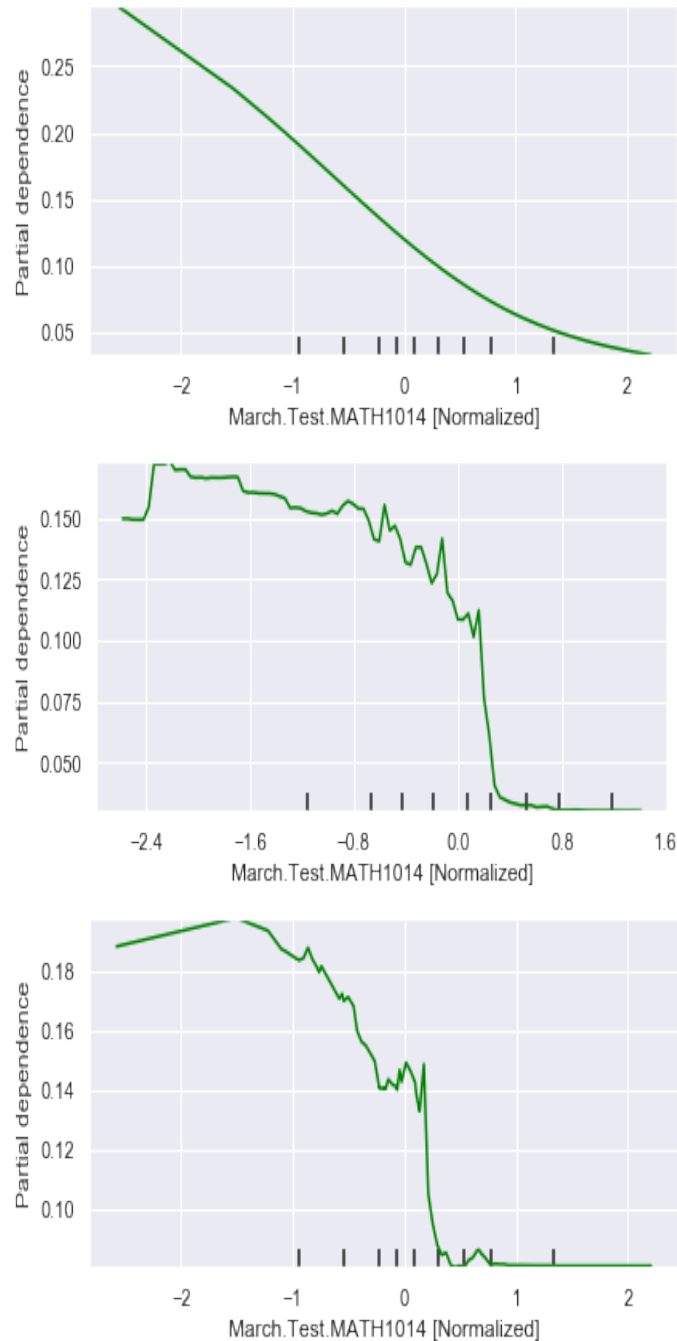


Figure 5.8: PD-plots for March.Test.MATH1014 [Normalized] generated by ANN (top), GB (middle) and RF (bottom).

The PD-plots for March test PHYS1014 (March.Test.PHYS1014) [Normalized] are shown in Figure 5.9. The plots are non-linear and have a turning point. From very low grades up until approximately -1 (31%) on the top plot (ANN), and approximately -0.5 (40%) for the GB and RF plots, there is a slightly positive slope. This suggests that the worse the students perform in this region, the lower their probability of being At-Risk, which is counter-intuitive. This can be explained in a couple of ways. The first is that the change in probability in this region is actually very small ($\tilde{0}.02$) and can be considered a flat region, meaning there is no change in probability. The second is that as students perform worse in this course (PHYS1014) they might accept that they will fail *this* course and subsequently place more emphasis on the PHYS1015 course, increasing their probability of passing. Above -0.5 there is a sharp decrease in probability of being At-Risk up until approximately 0.8 (63%), above which there is a constant probability of approximately 0.1. In terms of comparing with the global probability, the peak at the turning point climbs slightly above 15.1%. However, for this course, students below the peak (40%) are considered At-Risk.

Looking at the PD-plot for years since matriculation (Years.Since.Matric) [Normalized] feature (Figure 5.10), there is a notable turning point. The plots are concave and the turning point in the ANN plot is approximately 3 on the curve (3.6 years). For the GB and RF plots, there is no clearly defined turning point; the curve has a negative slope below 0 (1.3 years) and positive slope above 2 (2.8 years). This suggests that below 1.3 years, the less time out of high school increases the At-Risk probability and above 2.8 years, the longer time out of high school also increases the At-Risk probability. Furthermore, below about -1 (0.5 years) and above about 8 (7.5 years), At-Risk probability exceeds the global probability.

For the high school mathematics (HS.Mathematics) [Normalized] PD-plots (Figure 5.11) the ANN plot (top) is smooth and decreasing. However, the GB (middle) and RF (bottom) plots show more variability. For the GB and RF plots, below 0 (80%) the slope is negative, between -1.5 (68%) and -0.3 (78%) there is a flat region and for the RF plot, below -1.5 there is a negative slope until approximately -2.5 (60%). Furthermore, above 0 the probability increases again slightly, which suggests that those that score high grades in high school mathematics may have a slightly high risk of failing. This may be due to being overly confident and not working sufficiently hard. However, this increase is only approximately 0.02% and therefore could also be considered negligible. Finally, below approximately -1.8 (66%) the At-Risk probability is higher than the expected value.

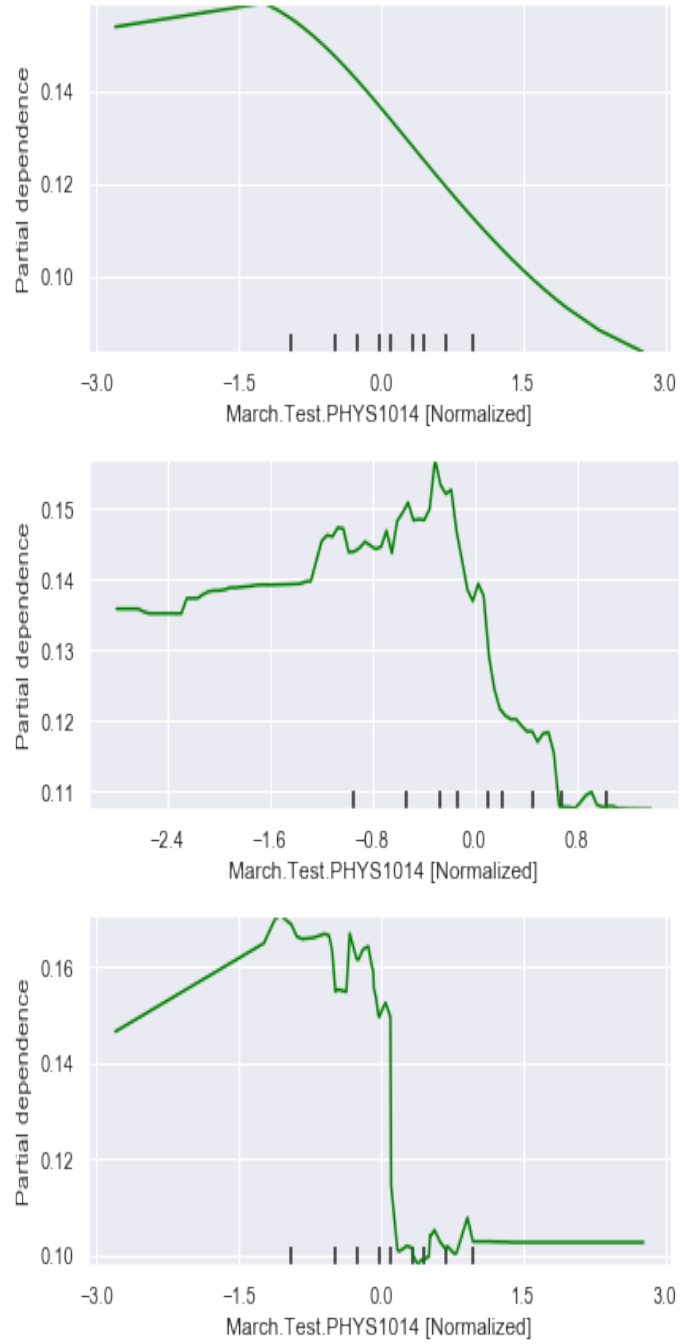


Figure 5.9: PD-plots for `March.Test.PHYS1014` generated by ANN (top), GB (middle) and RF (bottom).

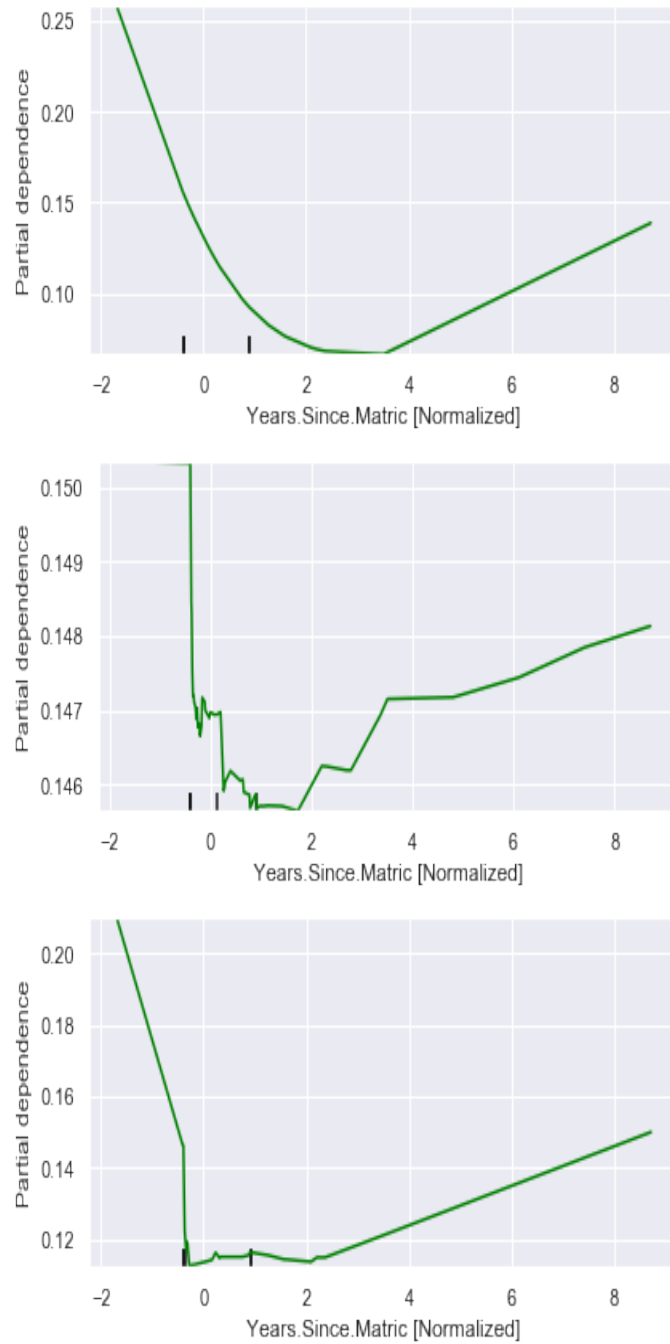


Figure 5.10: PD-plots for Years.Since.Matric generated by ANN (top), GB (middle) and RF (bottom).

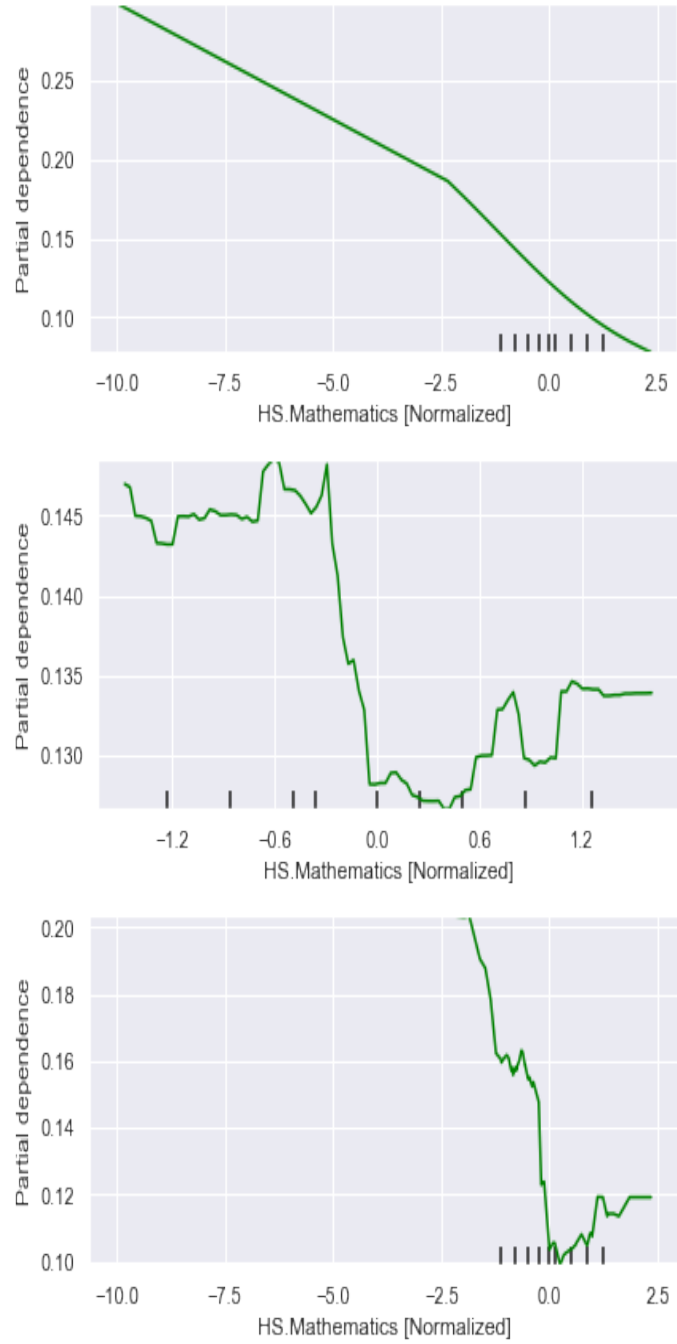


Figure 5.11: PD-plots for `HS.Mathematics [Normalized]` generated by ANN (top), GB (middle) and RF (bottom).

The remainder of the features were all categorical features and therefore would produce linear curves for the PD-plots, and therefore the plots were excluded. Table 5.2 describes the overall global features that identify and characterize At-Risk students. It summarizes how each feature is correlated with being At-Risk and the regions where At-Risk probability is higher than the global probability of 15.1%. The categorical features are described as follows. Higher means that that feature has a higher probability of being At-Risk than the complementary feature. For example, Residence.Yes being higher means being in residence has a higher At-Risk probability than not being in residence.

Table 5.2: Impact of top features on being At-Risk using PD-plots.

Feature	At-Risk impact	Regions higher than global probability
March.Test.P1015	Negative	< -0.5 (60%)
March.Test.M1014	Flat < -1 (33%) Negative between -1 (33%) & 0.1 (58%) zero > 0.1 (58%)	< -0.4 (41%)
March.Test.P1014	Positive up to 40% drop off & constant > 0.8 (63%)	< -0.5 (40%)
Years.Since.Matric	Negative < 0 (1.3 years) Positive > 2 (2.8 years)	< -1 (0.5 years) and > 8 (7.5 years)
HS.Mathematics	Negative < 0 (80%)	< -1.8 (66%)
ADU.1st.No	Higher	-
Residence.Yes	Higher	-
Student.Type.First	Lower	-
DegCd.Nuclear	Lower	-

5.5 Local Explanations

The previous sections focused on identifying the strongest predictors of being At-Risk *globally* and pointing in directions to identify, or characterize, At-Risk students. This section presents how to apply local model-agnostic, LIME and SHAP techniques, to cater to individual students and provide individual explanations.

To illustrate this, two students identified as At-Risk were randomly selected. Results for the first At-Risk student are presented in Figure 5.12 and 5.13 for LIME and SHAP respectively. Looking first at the LIME results in Figure 5.12, a 0.77 (77%) probability of being At-Risk (left bar figure) was predicted. Second, the middle bar-chart shows features pointing to the right (in orange) which contribute to the At-Risk prediction, and the feature pointing to the left (in blue) which contributes to being not At-Risk. The values of these features on the bar-chart indicate the values of the coefficients of the linear explanation model. On the right, the same features are shown but with their specific feature values for this student. For example, this specific student obtained -1.83 (48%) for March test PHYS1015. For this specific student, the features do correspond to the global features, showing that they performed poorly in the March tests, and they had a low high school mathematics grade. The only feature that contributed to being not At-Risk was their attending ADU (ADU.1st.Sem.No = zero), meaning they *did* attend the extra support tutorials. Also, the bursary type (Bursary.Type.UES = 0) appears to be irrelevant information.

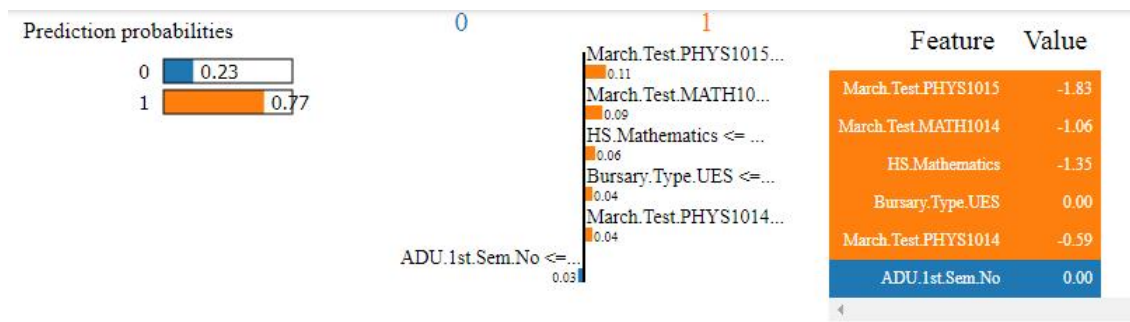


Figure 5.12: LIME results for first randomly selected At-Risk student.

Looking at the same student using SHAP results shown in Figure 5.13, the output value = 0.77, which corresponds to the 77% probability of the LIME prediction. This 0.77 output value is compared with the base value of 0.151 which corresponds to the 15.1% students that are At-Risk. Therefore this specific student had a 77% probability of being At-Risk, compared with the overall probability (15.1%) of the entire group. Next, the red bars show features that increase the chance of being At-Risk, whereas the blue bars decrease the chance. Also, each bar has a different length, indicating the size of the coefficient, similar to LIME. As discussed above, the LIME results identifies an irrelevant feature (Bursary.Type.UES = 0)

whereas the SHAP results show the relevant feature (Bursary.Type.None = 1). Therefore, this student could be told that they have a 77% probability of being At-Risk, that their low high school and March grades contribute to their being At-Risk, and that being in residence and attending ADU reduces their chance of being At-Risk.

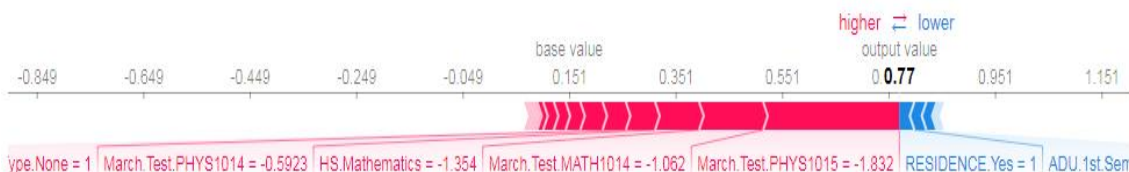


Figure 5.13: SHAP results for first randomly selected At-Risk student.

A second example is presented in Figures 5.14 and 5.15. Again, the results for LIME and SHAP are very similar, except that LIME again selects the complementary negative feature (DegCd.EB000 = 0) instead of the positive feature (DegCd.EF005=1). Also, SHAP selected the type of student feature (Student.Type.First = 0) as a positive contributor to being At-Risk which LIME did not select.

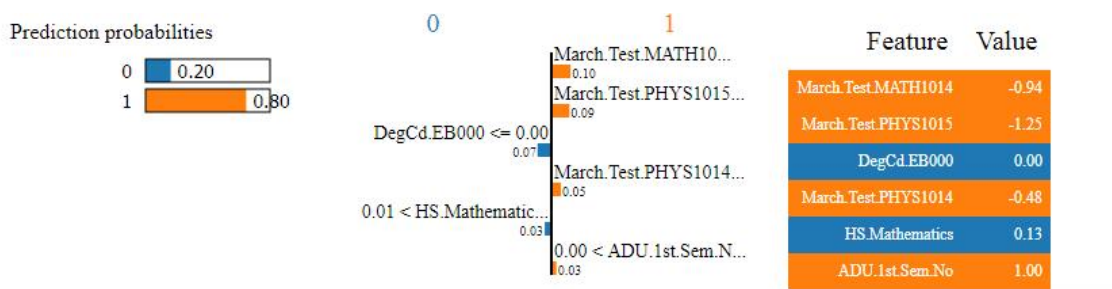


Figure 5.14: LIME results for second randomly selected At-Risk student.

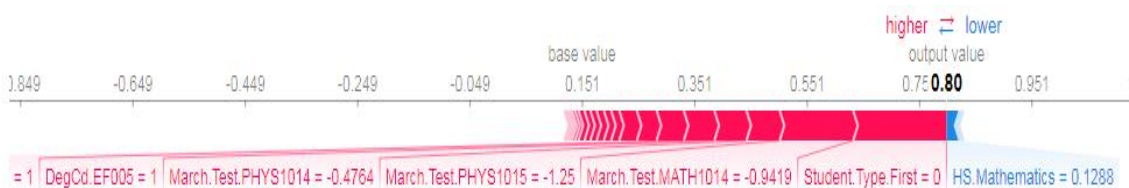


Figure 5.15: SHAP results for second randomly selected At-Risk student.

Although both techniques show similar results in these two cases, the literature suggests that LIME has its' drawbacks such as instability [6, 135] which means that in repeated sampling, explanations could be different. It was also shown that LIME and SHAP explanations can be inconsistent with each other when the underlying models are complex [135]. It is suggested that SHAP is a better technique since it is based on game theory and adheres to the three properties of local accuracy, missingness and consistency [6, 8]. This theory guarantees that

the output prediction is fairly distributed among the features, which LIME cannot guarantee [6].

The above section therefore presented the ways in which these local model-agnostic methods could be used to provide individual explanations to an At-Risk student.

5.6 Counterfactual Explanations

The next important benefit of local models is to use these identified features in counterfactuals models, i.e. potential "what if" scenarios that could help the students change their At-Risk status. The counterfactual models can be any model trained on the data [6]. In the case of this study, any of the original best performing models can be used (see Section 4.3.1) to estimate counterfactuals. The benefit of using the original models versus local models such as LIME and SHAP is that the values far away from the original value can be perturbed since the global models are valid throughout the feature space, unlike the local models.

Therefore to illustrate how to use counterfactual models, again consider the first At-Risk student discussed in Section 5.5, seen in Figures 5.12 and 5.13. Assuming that SHAP provides more reliable predictors for this student, the SHAP predictors can be perturbed to generate counterfactuals and estimate counterfactual probabilities. Counterfactuals for March test PHYS1015 were generated, shown in Figure 5.16, by perturbing this feature between -2 and 2 standard deviations. The horizontal red line is the 50% probability of being At-Risk and it can be noted when the probability crosses this line which indicates switching from either being At-Risk to being not At-Risk, or vice versa. Therefore, by perturbing this feature, the switching of the student's At-Risk status from positive to negative can be seen at approximately -1, which is 59%. Therefore, suggestions could be made to the person responsible for the student (instructor), that the student would need to score higher than approximately 60% in this test to switch to being not At-Risk. However, a limitation can be seen in applying this method to predicting At-Risk students: the March test PHYS1015 had already taken place and so the students cannot retake this March test to increase their score and decrease their At-Risk probability. That is, they cannot directly change *that* specific feature. In other applications such as qualifying for a bank loan, the client can focus on modifying the *same* factor to improve their status. For example, they know they need to earn a higher salary to improve chance of obtaining a loan. The idea here with predicting At-Risk students, is similar, but it is not possible to modify the same feature, such as March test PHYS1015. The students *can* however, modify a later PHYS1015 test grade to improve their chance of passing. This student's original score was -1.83 (48%) which is close to passing. A potential way of applying these results may be to inform the student that although they are

very close to passing, their grade is about 10% below what they should have, i.e. 60%, in order to eventually pass.

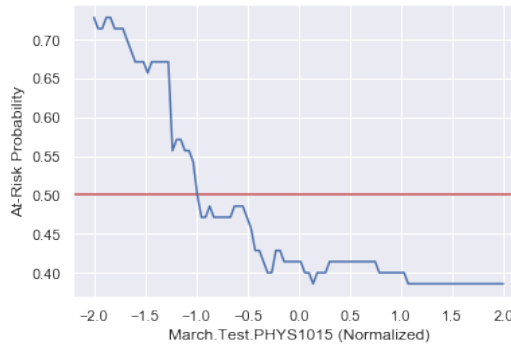


Figure 5.16: Counterfactual probabilities for March test PHYS1015

In Figure 5.17, March test MATH1014 is analyzed and at approximately 0 (54%) this student switches to being not At-Risk. This student had an actual score of -1.06 (31.7%). Note here that this feature is not directly, but indirectly linked to the target output, PHYS1015. Causation is not being implied here but it is simply shown that those students with these characteristics are At-Risk and might switch to being not At-Risk. However, due to PHYS1015 having a significant calculus component, MATH1014 might have a direct, though tenuous, causal relationship. This feature again suffers from the same limitation as discussed above for March test PHYS1015. That is, the student cannot modify this same feature but only a future test for this course. Nevertheless, this student is more than 20% below what they need to switch to being not At-Risk.

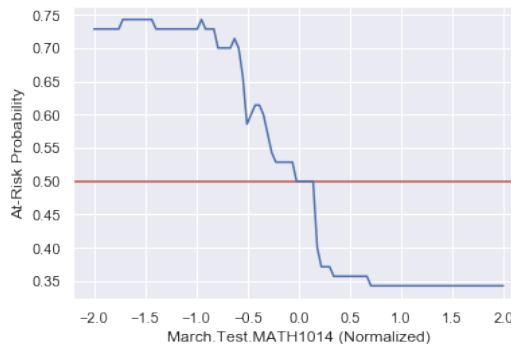


Figure 5.17: Counterfactual probabilities for March test MATH1014

For March test PHYS1014 (Figure 5.18), the student switches at approximately 0.1 (50%), whereas their actual score was 38.0%. Again, this course is not directly linked to PHYS1015, although there may be some overlap in the material.

Two categorical features are looked at next. Attending ADU (ADU.1st.Sem.No) is toggled between 0 (indicating *did attend ADU*) and 1 (*didn't attend ADU*). The counterfactual probabilities are shown in Figure 5.19. This student did attend the ADU and attending the

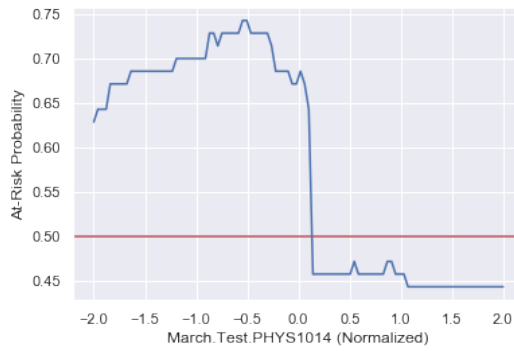


Figure 5.18: Counterfactual probabilities for March test PHYS1014

ADU did not result in being not At-Risk. However, if they had not attended (1), their At-Risk probability would be even higher than if they did (0). Similarly in Figure 5.20, for being in residence (RESIDENCE.Yes = 1) this student is At-Risk and by not being in residence (RESIDENCE.Yes = 0) the probability increases. Therefore it is better for this student to remain in residence, even though this specific feature does not result in being not At-Risk.

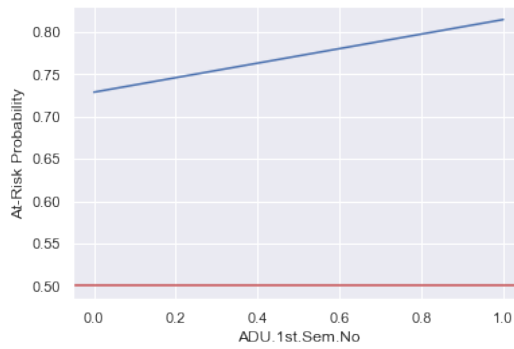


Figure 5.19: Counterfactual probabilities for ADU.1st.Sem.No

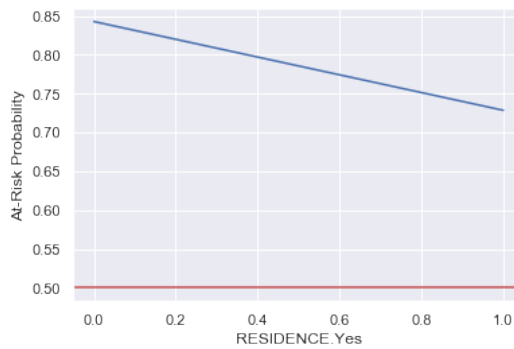


Figure 5.20: Counterfactual probabilities for RESIDENCE.Yes

5.7 Discussion

In terms of characterizing At-Risk students globally, a combination of global feature importance techniques were used. These techniques include FI and PI which were shown to agree on certain grades features but disagree on other demographic features. Nevertheless, combining these could provide a relatively confident indication of which features to consider when identifying At-Risk students. Perhaps the main limitation found with FI and PI was the lack of feature directionality which means that other methods were also required. This study made use of lasso coefficients to provide linear direction as well as PD-plots to provide non-linear information. The PD-plots presented regions for identifying At-Risk students, that are not captured by the other global methods. This study showed that the combination of methods used here are valuable because the FI and PI methods provide a “big-picture” approach of where to start looking for At-Risk students, and then the β coefficients and specifically the PD-plots provide a “zooming” method for further identifying At-Risk students globally.

It is vital in these types of applications to not mistake correlation with causation. Features varying with the target outcome does not imply that they are the cause of the outcome. For example, in the β coefficients computed by lasso, there is a gender feature, female, that is positively correlated with being At-Risk. Much care is required here since an erroneous concept could be assumed that being female leads to poorer grades and failing. In order to measure causality, there is a need for proper experiments such as a randomized controlled trial (RCT) or matching techniques such as PSM. Therefore these features are simply indicators or predictors of being At-Risk and should not be seen as causal. They nevertheless provide an area where the educator can begin to identify At-Risk students.

Following on from above, these results should be provided to an experienced educator that would use these results with the necessary caution so as to not be insensitive to any race or gender. The goal here is to assist the students to improve their pass rate, but there is the potential of using these results in an improper way and it is advised that educators in conjunction with psychologists use these results in the best way.

The global approach showed much value. The individual approach using the local model-agnostic methods can also add significantly more value to assisting individual At-Risk students. LIME and SHAP generally provided similar results and the literature suggested that SHAP was more reliable. The value in these methods is in providing explanations to an individual student as to why they were considered At-Risk and which areas they need to improve in. Again, care should be taken to provide these results to experienced educators who could in an appropriate way, present these results to a student. This method can be of value to the field of student performance since ML could be used to first predict which

students are At-Risk, then use these local techniques to find out why individual students are At-Risk. There is however, a twofold limitation of this approach. The first is that, as discussed above, these features are not necessarily causal features. This means that even if certain features are identified as predictors of being At-Risk, it doesn't mean that changing them will affect the outcome. There may well be confounding features that are the true cause. Second, even if certain features could be identified as problematic, they can't necessarily be changed. Certain features are simply static, for example race or age, and cannot be changed. Dynamic features such as grades or residence, could be changed, but again it is not known whether these changes will affect the outcome directly. For example, if SHAP predicted that an At-Risk student had positive residence SHAP values, then the student could move out of residence, since this is a dynamic feature. However, this is not necessarily a causal one.

Continuing with the thought on changing features, counterfactual explanations were generated. The use of counterfactual explanations is simple to understand in a bank loan scenario, for example. If the applicant was denied a loan, the local models could provide an indication of why they were denied and then counterfactual explanations would allow them to see what they need to change in order to qualify for a loan. It is slightly more complicated in the scenario for working with At-Risk students. It is possible to identify the students and then provide the instructor with each student's specific description (using LIME/SHAP). However, using counterfactuals to advise them on what to modify to switch their At-Risk probability requires further study. Since taking an academic course is time-based, where the student cannot repeat a test the same year, but only prepare for a next one, this method works best for advising general increases in grades to reduce being At-Risk. Furthermore, another limitation is that causal effects cannot be assumed (except perhaps for March test PHYS1015); only correlation. They however do point in directions where suggestions to the instructor could be made on how the students could improve. Therefore, saying that a student needs to improve in physics (PHYS1014) in order to pass mechanics (PHYS1015) is a difficult argument to make. There might be some overlap in the concepts between these two courses, so that doing well in the one might improve the other one. A similar argument can be made also for mathematics (MATH1014). Either there is some direct cause, or there could be a confounding feature that has not been measured. Another limitation in this study is that features in isolation were considered when generating counterfactuals. Future work would include performing optimization studies to obtain a range of counterfactual features that could switch At-Risk probability [6].

With regards to catering to individual students, it is suggested that the best way to address these different individual situations is to (1) identify At-Risk students using LIME/SHAP probability predictions, (2) generate individual LIME/SHAP values to describe each student,

(3) study these features and their potential effects with experts and, (4) suggest various solutions/interventions on a case by case basis. There are some features that could be suggested to improve. For example, the obvious one would be the PHYS1015 grade, if it was identified. Other suggestions could be attending a supplementary tutorial intervention, such as ADU. These methods may even point to hidden features that are affecting a number of measured features. This may be beneficial for uncovering a root cause of a student's poor performance and thereby allow for constructive advice. These methods therefore need to be applied on a case by case basis and require much sensitivity.

5.8 Conclusion and Future Work

This chapter formed the second part of two chapters that addressed the first two objectives of this study, which were to find better ways of predicting and characterizing At-Risk students and to provide advice or interventions to help them switch to not being At-Risk. The flow of this chapter was from the global to the local aspects of identifying characteristics of being At-Risk.

At-Risk students were first characterized using global model-agnostic methods. FI and PI techniques agreed on the grades features (March tests and high school grades) being important but they differed on demographic features. Using FI and PI, together with RF, GB and ANN models provided a multi-dimensional view into which global features were impacting At-Risk status. Although FI and PI provided the most important global predictors of being At-Risk, lasso β coefficients were needed, together with PD-plots, to provide direction to the features and indicate any non-linearities in the marginal effect of these top features. Much insight was provided by computing the partial dependence, since it showed regions where students had higher probabilities of being At-Risk than the global population.

Individualized care is becoming more important in various fields. Local explanations were generated using local model-agnostic techniques, LIME and SHAP, to show how individual students might be helped. SHAP was considered to be a more trustworthy method. These explanations can be presented to the instructor to highlight specific features that contribute toward a student being At-Risk, and the instructor can then map a way forward for improving the student's chance of passing.

Counterfactual explanations were introduced to this field and is a promising method to help direct the student on how to switch to not being At-Risk. This method showed how to perturb the features so that their marginal effect on the probability of passing is changed. It would allow for seeing how a student would need to modify certain features in order to change

At-Risk status. However, limitations were found in this method, in that certain features could not be modified.

The combination of global and local methods presented in this chapter promise much in terms of characterizing At-Risk students and aiming to improve the pass rate. However, this study shows that much care is needed to analyze each and every result with much care.

In terms of future work, there is the need, with the ethical permission of the University, to test these methods on future cohorts of students to see if similar results are found and to test the generalizability of this method.

Although the work on characterizing students requires follow up studies to test generalizability, perhaps the main area of future work is concerned with providing interventions using the local models and counterfactual explanations. This would require (1) identifying carefully which features could indeed be varied, i.e. dynamic features and (2) methods for measuring counterfactuals, i.e. after advising a student to modify a certain feature, there needs to be a way to measure if that modification indeed changed the At-Risk status using real data, and finally (3) combinations of counterfactuals need to be generated, not only individual ones as in this chapter; this would provide a range of ways for a student to be able to switch At-Risk status.

Chapter 6 Treatment Effects using Machine Learning Methods

6.1 Introduction

In the previous chapter methods of predicting and characterizing At-Risk student in a global and local sense, using ML, were presented. This chapter addresses the third objective of this study which is to find better ways of measuring the effectiveness of interventions, i.e. measure treatment, or causal, effects. A major discussion point from Chapter 5.7 was that the features did not necessarily imply causation, but correlation. This chapter however, aims to study causation in the intervention feature, ADU, discussed in Chapter 3.2. The ADU feature refers to extra face-to-face tutorials for PHYS1015. The aim was to use ML methods to measure how effective this intervention was on improving final grades.

Measuring the effectiveness of an intervention is highly important. Without measuring this effectiveness, it is not possible to know whether to continue with the intervention or modify it in any way. An intervention, or treatment, has an aim of bringing about an effect on a person or group receiving the treatment. The effectiveness is measured via a treatment effect, which is the difference between what a participant received, say the treatment (or control), and what they would have received, had they not received the treatment (or control). This is to compare potential outcomes [136]. Their outcome based on what they would have received in another scenario, is known as a counterfactual. However, a fundamental problem is that it is only possible for an individual to receive either the treatment or control at the same time [136]. The critical item therefore in measuring treatment effects is the *counterfactual*. Without the counterfactual, there is no way to measure a treatment effect.

ML has been used to estimate counterfactuals and thereby measure treatment effects in various fields [137, 138, 139]. Beemer et al. (2017) [4] and Beemer et al. (2018) [9] were some of the first to use ML to estimate counterfactuals in student performance, via prediction. Their work is inspiring. To measure the treatment effect of an intervention, they trained ensemble ML models on both the control and treatment groups and used the models to

predict the counterfactuals in the treatment effects. Consider Y_c to be the actual control group output. To generate the control group output counterfactuals, $\hat{Y}_c$, they fed the control group input features into a model trained on the treatment group. Similarly, consider Y_t as the actual treatment group outcome. The treatment group counterfactual, $\hat{Y}_t$ was predicted by feeding the treatment group input features into a model trained on the control group. The procedure is given below:

1. Train a model M_c on the control group.
2. Train a model M_t on the treatment group.
3. Predict the treatment group counterfactual $\hat{Y}_t$, by feeding in the treatment group features into M_c , the control group model. $\hat{Y}_t$ is therefore the predicted outcome of the treatment group, had they received the control.
4. Predict the control group counterfactual $\hat{Y}_c$, by feeding in the control group features into M_t , the treatment group model. $\hat{Y}_c$ is therefore the predicted outcome of the control group, had they received the treatment.
5. Estimate treatment effects $ATT = Y_t - \hat{Y}_t$ and $ATC = \hat{Y}_c - Y_c$.

The treatment effects ATT and ATC are the average treatment effect on the treated and average treatment effect on the control, respectively. In their paper, Beemer et al. (2017) did not use the terms ATT and ATC but ITE, individual treatment effect. This method is similar to that proposed by Athey and Imbens (2015) [14], by comparing predictions to actual outcomes. They have effectively utilized ML to predict what each student would score if they received both the control and the treatment, thereby addressing the counterfactual problem. Using this technique, they estimate improvements in the exam grades of approximately 9% and in the GPA of 0.45%. In predicting final exam score and course grade, as well as predicting if they will score above a C-grade, they show that the ensemble model performs better than all the base models. Their ensemble model consists of a number of base models: random forest, support vector machines, boosting, lasso, linear regression, k-nearest neighbor and Naive Bayes. Their population size was $n = 1032$ students and those enrolled in the intervention were $n = 169$. They suggested using the lasso model since it performed nearly as well as the ensemble but is computationally less expensive. They also suggest using a larger sample size to alleviate the correlation between the base learners. They however do not report on any details of how they developed the ML models.

In a similar study, Beemer et al. (2018) [9] estimate treatment effects on a course that had a traditional lecturing component comprising 157 students (considered the control group) as well as an online component of 57 students (considered the treatment group). Their aim was

to estimate the treatment effect of the online course on final grades. They applied a similar technique to the one discussed above for Beemer et al. (2017) [4] and chose two machine learners: random forest and lasso for their study. Table 6.1 shows that the random forest learner predicts an $ATT = 14\%$ suggesting the online group (treatment) would score 14% higher than if they received the control (i.e. only taken the traditional course). Also, ATC was calculated as 1% suggesting that the traditional group (control) would score 1% higher had they taken the online course (treatment). The lasso showed ATT and ATC scores of 7% and 2% respectively, although again, the paper only speaks of ITE.

Table 6.1: Treatment effects estimated by Beemer et al. (2018) [9].

Model	ATT [%]	ATC [%]
RF	14	1
Lasso	7	2

A few important points are made regarding the two inspiring studies carried out by Beemer et al. (2017) [4] and Beemer et al. (2018) [9].

- They do not take into account the difference in ATT and ATC . They compare ATT and ATC with each other when in fact they are treatment effects on different groups.
- There is no ground truth to compare their results against. How accurate are these counterfactual predictions? There is therefore the need to validate their method.
- This method potentially introduces bias into the counterfactual predictions due to the control and treatment groups potentially not having balanced features.

The above points therefore present areas to focus on in this study.

A further major thrust in the literature is that of individualized learning [4, 9, 140], as presented in Chapter 5. It follows therefore that not only global treatment effects need to be measured, such as ATT and ATC , but individual treatment effects, ITT and ITC . Although global treatment effects, i.e. average treatment effects on the entire group, might be helpful, understanding how different subgroups or individual students might be affected differently is perhaps considerably more valuable. It might be of great use to know how the benefits of an intervention vary with the features of the individuals [14, 140, 141]; i.e. who would benefit from this intervention? ML has been used in the literature to measure individual, or heterogeneous, treatment effects [4, 140], but is still in its' infancy in the field of student performance.

This chapter therefore proposes the following to contribute to the literature:

1. Section 6.5 presents how ML methods can be used to measure global treatment effects by predicting counterfactuals. This chapter builds on and extends the inspiring work shown in the literature [4, 9]. This method was validated by comparing it with the traditional PSM which is the first known case in the literature that compares these techniques.
2. In Section 6.6 it is shown how to compute subgroup and individual treatment effects and how they differ for different students. It is argued that it is more beneficial to compute treatment effects for subgroups and individual students than for the entire cohort. Characteristics of these subgroups are determined that allow for identifying who would and would not benefit from the intervention.
3. In Section 6.6.3 the focus shifts entirely to individual students. A method was presented for predicting the final course grade of individual students and predicting by how much the intervention will improve/worsen their final grade. Previous sections looked at groups, whereas this section caters to individual students to inform them specifically by how much the intervention would help them. In light of individualized learning, SHAP, presented in Chapter 5 was also used. The benefit of using this method is that it predicts the outcome of an individual student as well as explain the features that contribute towards that outcome.

The main objectives of this study have been presented above. The theory of treatment effects is presented in Section 6.2. Thereafter, the methodology governing this study is presented in Section 6.3. In Section 6.4 the performance of the ML models is shown. In terms of the results, Section 6.5 analyzes the treatment effects on the entire group, together with the validation of this technique. Section 6.6 then applies the prediction techniques to individual subgroups. Section 6.6.3 analyzes treatment effects on individual students and predicts by how much individual students will benefit from the intervention. Sections 6.7 and 6.8 present the discussion and conclusion, respectively.

6.2 Theory of Treatment Effects

As mentioned above, the critical component of measuring treatment effects is to measure the counterfactual. The following describe the various methods for measuring treatment effects.

6.2.1 Randomized Control Studies

A solution to the problem of the counterfactual is to create two randomized groups that have similar features statistically; i.e. balanced features. One group would then be assigned an intervention (treatment group) and the other would be assigned the control. This is known as a randomized controlled trial (RCT), the “gold standard” of experimental studies [10]. By having two groups that are the same statistically, with one receiving a treatment and the other not, a counterfactual is practically created. If a difference is measured between treatment and control groups, then due to the two groups having balanced features, causality can be attributed to the intervention. This difference, or average treatment effect (ATE), shown in Equation 6.1 is then simply the expected difference between the average outcome of the treatment group $E[Y_t]$ and the control group $E[Y_c]$ [142]. This will be valid for randomized studies.

$$\begin{aligned} ATE &= E[Y_t - Y_c] \\ &= E[Y_t] - E[Y_c] \end{aligned} \tag{6.1}$$

The challenge is that due to ethical reason, many academic interventions such as the one in this study, are not randomized. It might be considered unethical to randomly assign one group of students to receive an intervention and the other to not receive it. Many academic interventions are *observational studies*, discussed next.

6.2.2 Observational Studies

Observational studies are different from RCTs in that participants are not randomly assigned to the control and treatment groups; students self-select to attend the intervention, potentially resulting in selection bias [4, 143]. This results in the two groups not necessarily having balanced features. This practically means that *there is no counterfactual group*. If a difference is then measured between the groups, the difference, i.e. causal effects, cannot be attributed to the treatment but may be due to confounding features which impact both the outcome of the intervention and the choosing of the intervention. The academic intervention in this study is observational due to the students self-selecting to participate and not being randomly assigned. Therefore if the treatment group had overall better (or worse) results than the control, it cannot be concluded that the intervention caused the better (or worse) performance. It may be due to confounding features, such as the treatment group students being more diligent and harder working than the control. For this reason, the ATE cannot simply be used to measure the treatment effects, since there may be hidden biases. Morgan

and Winship (2007) [142] state that in observational studies the ATE essentially becomes a naive estimator ATE_{naive} that is rarely equal to the ATE under a randomized state. In order to estimate the “true” treatment effect in observational studies the ATE_{obs} is computed, shown in Equation 6.2 [142]. To obtain ATE_{obs} , counterfactual outcomes for the treatment and control group need to be estimated. That is, estimate what the treatment group would have received had it received the control, i.e. the treatment group counterfactual $\hat{Y}_t$, and what the control group would have received had it received the treatment, the control group counterfactual $\hat{Y}_c$. π is the percentage of observations receiving the treatment. Y_t is the actual average output of the treatment group and Y_c is the actual average output of the control group. The quantity $Y_t - \hat{Y}_t$ is known as the average treatment effect on the treated (ATT) and $\hat{Y}_c - Y_c$ is the average treatment effect on the control (ATC).

$$\begin{aligned} ATE_{obs} &= \pi(Y_t - \hat{Y}_t) + (1 - \pi)(\hat{Y}_c - Y_c) \\ &= \pi \cdot ATT + (1 - \pi)ATC \end{aligned} \tag{6.2}$$

As the name indicates, ATT estimates the difference between what the treatment group actually obtained and what they would have obtained (counterfactual) had they received the control. Similarly, ATC estimates the difference between what the control group would have obtained had they received the treatment and what they did actually obtain. As is evident, *estimating counterfactuals is the key to measuring true treatment effects* in observational studies.

6.2.3 Propensity Score Matching

Techniques such as matching and PSM are often used to generate counterfactuals and estimate treatment effects in observational studies [136, 144, 145]. Matching is a statistical technique that compares individuals in the observational control and treated groups that have matched (similar) features; i.e. it finds counterfactuals. The challenge with this is that it is often difficult to find close matches when the number of features become larger [136]. To circumvent this, PSM estimates a propensity e (see Equation 6.3) for an individual i , given a set of features, X , of receiving the treatment, $D = 1$. D is the variable indicating whether the individual is part of the treatment ($D = 1$) or control ($D = 0$) group. The propensity is the probability of receiving the treatment and is often computed using logistic regression.

$$e_i = P(D_i = 1|X_i) \tag{6.3}$$

The aim then is to match individuals based on this score which in effect is meant to balance the features in the two groups. For example two individuals that have similar propensity scores but are in different groups (treatment or control) can then be compared. This is essentially to generate counterfactuals. Examples of utilizing PSM are given in [11, 12, 13].

The procedure then is as follows:

1. Using logistic regression, compute the propensity scores (probabilities) of receiving the treatment, for all cases.
2. Split the entire group into treatment and control.
3. Match treatment and control cases that have similar propensity scores and measure differences.

Matching is performed in various ways, with nearest-neighbor [15] being a common method. Nearest-neighbor matching constructs a counterfactual for each treatment (or control) case using the control (or treatment) cases that are closest in propensity score. This method is known to use a caliper distance (measured in standard deviations) which is the maximum absolute difference allowed in the scores of matched subjects [145]. The literature has suggested values of 0.2 or 0.25 [145, 146, 147], but others have used values as small as 0.001 [142]. However, Lunt (2013) [147] states that the appropriate caliper depends on the dataset and should be tight enough to produce close match but at the same time produce a sufficient number of matches. That is, matching should not be so tight that the number of matched subjects is too small. Furthermore, optimal matching is said to occur when treated and control groups have balanced features [142]. It has been suggested to *proceed iteratively* until systematic differences between treated and control features have been eliminated [145].

The literature has suggested using nearest-neighbor caliper matching with replacement [142], where *with replacement* means that once a case has been used for matching, it is placed back in the pool to be used for matching on another case. The procedure for implementing PSM in this study is presented in Section 6.5.1.

6.3 Methodology

This section first presents the overall approach outlining the steps to address the objectives of this chapter. Thereafter it describes the context of the study, the datasets, the method behind selecting the best ML models, and the various global and individual treatment effects computations.

6.3.1 Overall Approach

1. Obtain the best performing ML models by training on control and treatment groups.
2. Measure treatment effects using the ML counterfactual prediction method.
3. Compare the ML treatment effect method with PSM.
4. Using the ML method, which is further referred to as the MLTE method, compute individual treatment effects and analyze ITT and ITC. Characterize various subgroups using the permutation importances feature selection method.
5. Use SHAP techniques to analyze individual students and predict how the intervention will affect individuals.

6.3.2 Context

The context of this study was a first year engineering mechanics course PHYS1015, where an intervention, a supplementary face-to-face tutorial called ADU (Academic Development Unit), was provided. The intervention was not compulsory, students self-selected to attend the tutorials. Two hours per week of extra tutorials were provided. These interventions involved teaching the content in a slower way, having more class discussions, doing more examples and problems and learning in groups.

6.3.3 Control and Treatment Groups

Over the course of the semester, student attendance for ADU class was recorded and those that attended ADU were considered the treatment group and those that didn't, the control group. It should be noted that even if a student attended a single ADU tutorial, they were considered part of the treatment group. Students were aware of class attendance. Chapter 3 concluded with the finalized Entire dataset together with the control and treatment datasets shown in Table 3.6. For convenience, this table is reproduced in Table 6.2. The total number of students was 878 with the control group comprising 384 (43.7%) students and the treatment group 494 (56.3%) students.

T-tests were performed comparing the features to test for feature balance. The table shows that a large number of features had significant differences (p -value < 0.05 at $\alpha = 5\%$), which suggests that the self-selection to attend the intervention resulted in unbalanced features in the control and treatment groups. Due to correlated features being removed, there were no

t-tests performed on those features. Furthermore, a sample dataset of a randomly chosen student is also presented in Table 6.3.

6.3.4 Model Selection and Performance

A vital part of this study was the use of ML models to train on the control and treatment datasets. The overall aim was to use ML models to predict counterfactuals and thereby estimate treatment effects. To do this, the best performing models were selected. In this study, lasso, random forest (RF), gradient boosting (GB), adaboost (AdaB) and artificial neural networks (ANN) were trained on both datasets with the aim of finding the best performing models. Ten-fold CV was performed to obtain the optimal hyperparameters for each model. The detailed procedure and results for model selection and performance on both control and treatment datasets are presented in Appendix B. The metric used was mean absolute error (MAE) since it was required to measure how far off the actual grade was when performing CV.

6.3.5 Global Treatment Effects

The ATT, ATC, ATE_{obs} quantities were computed based on the best performing models. In addition to the best performing models, the lasso model was also included to compare with the study of Beemer et al. (2018) [9] since they also compared lasso and RF models in their studies.

6.3.6 Propensity Score Matching

In the literature the ML technique implemented here has not been validated by standard methods [4, 9]. Furthermore, Athey and Imbens (2015) [14] state that in typical ML problems, there is a test set that is set aside for testing which is considered the “ground truth”; i.e. the actual target values. However, in this case, there is no “ground truth.” There is no test set to measure against to see how accurate the counterfactuals are, since the counterfactuals can never be measured. Therefore the traditional PSM method was used to generate counterfactuals and validate our technique. The *RStudio* module called *Match* was used to compute the ATE, ATT and ATC using PSM as follows. Nearest-neighbor matching with replacement was selected with a caliper distance of 0.25 as suggested in the literature. Then at this caliper distance, the degree of feature balance was measured using the *RStudio MatchBalance* module. *MatchBalance* compares each feature in the treated and control groups before and after matching, by computing various statistics and their p -values. If the p -values are less

Table 6.2: Statistics of different datasets used in this study.

	Feature	Entire (878)	Cont. (384)	Treat. (494)	p-value
Gender	Male	631 (71.8)	296 (77.1)	335 (67.8)	0.00
	Female	247 (28.1)	88 (22.9)	159 (32.2)	-
Age		18.8 (1.1)	18.9 (1.2)	18.6 (1.1)	0.00
Ethnicity	African	710 (80.8)	248 (64.5)	462 (93.5)	0.00
	Caucasian	84 (9.5)	74 (19.3)	10 (2.0)	0.00
	Indian	65 (7.4)	49 (12.7)	16 (3.2)	0.00
	Coloured	16 (1.8)	10 (2.3)	6 (1.2)	0.14
	Chinese	3 (0.3)	3 (0.8)	0 (0)	-
Citizenship	RSA	833 (94.8)	367 (95.5)	466 (94.3)	0.10
	PR	33 (3.7)	10 (2.6)	23 (4.6)	-
	ROC	12 (1.3)	7 (1.8)	5 (1.0)	-
Student Type	First Year	781 (88.9)	315 (82.1)	466 (94.3)	0.00
	Repeat	97 (11.0)	69 (17.9)	28 (5.7)	-
Degree Code	Civil	189 (21.5)	71 (18.5)	118 (23.9)	0.05
	Electrical	183 (20.8)	72 (18.8)	111 (22.4)	0.17
	Mining	155 (17.6)	66 (17.1)	89 (18.0)	0.74
	Mechanical	126 (14.3)	47 (12.2)	79 (16.0)	0.11
	Aeronautical	65 (7.4)	18 (4.7)	47 (9.5)	0.00
	Industrial	19 (2.1)	8 (2.1)	11 (2.2)	0.88
	Biomedical	55 (6.2)	39 (10.1)	16 (3.2)	0.00
	Digital Arts	25 (2.8)	21 (5.4)	4 (0.8)	0.00
	Nuclear	61 (6.9)	42 (10.9)	19 (3.8)	0.00
Residence		205 (25.8)	79 (20.5)	149 (30.1)	0.00
Bursary Type	None	281 (32.0)	151 (39.3)	130 (26.3)	0.00
	UES	307 (34.9)	133 (34.6)	174 (35.2)	0.85
	OT	204 (23.2)	66 (17.2)	138 (27.9)	0.00
	NSF	86 (9.8)	34 (8.8)	52 (10.5)	0.40
Years Since	Matric	1.3 (0.77)	1.39 (0.85)	1.2 (0.68)	-
March Test	PHYS1015	72.2 (13.2)	73.4 (14.1)	71.2 (12.4)	0.01
	MATH1014	53.8 (20.9)	52.2 (24.4)	55.1 (17.7)	0.05
	PHYS1014	48.2 (71.2)	46.5 (20.3)	49.5 (14.3)	0.01
High School	Mathematics	80.9 (8.1)	81.0 (8.8)	80.9 (7.5)	0.85
	Physical Sciences	79.0 (8.0)	78.6 (9.1)	79.3 (7.1)	-
High School English	First	293 (33.3)	174 (45.3)	119 (24.0)	0.00
	Second	563 (64.1)	204 (53.1)	359 (72.6)	-
	Foreign	22 (2.50)	6 (1.5)	16 (3.2)	0.10
Target Feature	PHYS1015 Mid-Year	61.8 (13.0)	62.3 (14.9)	61.3 (11.4)	0.27

Table 6.3: Sample data of random student.

Feature	Value
ADU	Yes
Residence	No
Age	-0.30
Gender	Male
Ethnicity	African
March test PHYS1015	-0.13
March test PHYS1014	0.36
March test MATH1014	0.69
Degree	Industrial
Bursary Type	None
High school English	First
High school Math.	0.25
Citizenship	RSA
Final grade PHYS1015	0.42 (67.3%)

than 0.05 then the feature in the two groups is statistically different. But if after matching, the p -value is larger than 0.05 it is assumed the feature in both groups is balanced. The caliper distance is gradually reduced until all the features have been balanced.

6.3.7 Individual Treatment Effects

Previously, the ATT and ATC were described that compute treatment effects on the entire group globally. Individual treatment effects, i.e. the individual treatment effect on the treated (ITT) and individual treatment effect on the control (ITC), were computed as per Equations 6.4 and 6.5, respectively.

$$ITT(i) = Y_t(i) - \hat{Y}_t(i) \quad (6.4)$$

$$ITC(i) = \hat{Y}_c(i) - Y_c(i) \quad (6.5)$$

The result of this computation is that each student would have their own individual treatment effect.

6.3.7.1 Subgroup Analysis

These individual results were analyzed first by looking at subgroups. The following subgroups were analyzed: (1) positive ITC/ITT values, (2) positive ITC/ITT values above 10%, (3) negative ITC/ITT values, (4) negative ITC/ITT values below 10%. The treatment effect of 10% was selected since the intervention may be of most help to those who are obtaining grades approximately 10% below passing or less.

The aim of this analysis was to find out what types of students would benefit (positive ITC/ITT) or would not benefit (negative ITC/ITT) from the treatment. Are there differences between the treatment and control students? Which students would benefit substantially (ITC/ITT >10%) or not benefit substantially (ITC/ITT <10%)?

The students were characterized by using the ML feature selection technique, permutation importances (PI), presented in Chapter 5. This technique allowed for identifying the main features for each subgroup that would allow for identifying the characteristics of the students that do and do not benefit.

6.3.7.2 SHAP Values

The previous section looked at subgroups. Within the individual treatment effects analysis, SHAP values were computed for randomly chosen control and treatment group individual students, to present how to (1) predict final grade and (2) predict the treatment effect. This was considered to be a highly valuable prediction since predictions could be made as to how individual students could be characterized.

6.4 Model Selection

Model performance on the control group is first summarized in Table 6.4 and the results from the complete model selection procedure is presented in Appendix B. For the models trained on the control group, three models performed the best, namely RF, GB and ANN. This is similar to the results obtained for the classification problem in Chapter 4.

Table 6.4: MAE for each model trained on the Control Group.

Model	MAE [%]
Lasso	7.57
RF	7.31
GB	7.33
AdaB	7.94
ANN	7.36

Model performance on the treatment group is shown in Table 6.5. Here the best performing models are RF and GB. Again, the RF and GB models are the better performers.

Table 6.5: MAE for each model trained on the Treatment Group.

Model	MAE [%]
Lasso	6.91
RF	6.62
GB	6.62
AdaB	6.86
ANN	6.90

6.5 Global Treatment Effects

The previous section identified the best performing models on the Control and Treatment Groups. The models $RF_{control}$ and $GB_{control}$ were subsequently used as the control group models and RF_{treat} and GB_{treat} were used as the treatment group models. $RF_{control}$ and $GB_{control}$ were used to compute ATT and RF_{treat} and GB_{treat} were used to compute ATC. The aim was then to feed the treatment group input features into the control models in order to obtain a counterfactual for the treatment group. Similarly, the control group input features were fed into the treatment models to generate control group counterfactuals. As discussed earlier, the lasso model was also included in this section to compare with the literature.

The ATE_{obs} , ATT and ATC were computed using the method indicated above. Results for these quantities are presented in Table 6.6. ATE_{obs} was computed as per Equation 6.2, with the fraction of treated student $\pi = 0.563$. For completion, the ATE_{naive} was calculated as per Equation 6.1, as -1.01%; i.e. the treatment group had an average grade higher than the control group. However, the difference was not significant ($p = 0.27$, $\alpha = 5\%$) and is biased.

Table 6.6: Treatment effects computed from the RF, GB and lasso models.

Model	ATT [%]	ATC [%]	ATE_{obs} [%]
RF	0.63	-0.58	0.10
GB	0.65	-0.63	0.09
Lasso	0.51	-0.61	0.02

The results firstly show that the models were consistent; i.e. they all estimated positive ATT values and negative ATC values. The results are suggesting that the treatment effects on the treated, ATT, had an average effect of improving their results by approximately 0.65% compared with had they not received the treatment. However for the control group, on average the intervention would not have helped them, indicated by the negative ATC values. The true ATE, i.e. ATE_{obs} , is a small positive value of approximately 0.1 %.

Therefore these results show that the ADU interventions, on average, do not appear to help much, since these values are small. In the literature by Beemer et al. (2018) [9] they recorded relatively large differences in treatment effects between the lasso and RF models (see Table 6.1). However, in this case, the lasso and the tree-based models have similar results. This may be due to performing a detailed model selection procedure to obtain optimal hyperparameters.

6.5.1 Validation Using PSM

The treatment effects were compared with PSM. In these computations, a caliper distance of 0.001 was found where the features were balanced. At this caliper distance, the ATT, ATC and ATE were computed as shown in Table 6.7 which was compared with the ML (ML) method. The PSM results first of all show that there were no significant treatment effects (p -values > 0.05), shown in brackets. Although these results are discouraging from a pedagogical point of view, the aim in this study was not to focus on the teaching aspect, but on first measuring how effective the intervention is. Once measurements can be taken, then the teaching can be modified. Next, the ML (ML) method computes ATT and ATE values that are in the ballpark of the PSM values. The ML technique computed a larger ATT value and a smaller ATC value than the PSM technique. Although difficult to determine, this could potentially be due to bias in the ML technique. Therefore although the PSM results show that there was no significant treatment effect, the ML results are relatively close to the

traditional method. Reasons for the difference in results can be attributed to the ML models having errors (MAE) of approximately 7% (see Tables 6.4 and 6.5) and not actually knowing the “true” propensity score. Follow up studies could include simulation studies where the bias in this method can more fully be ascertained.

Table 6.7: Comparison of ML technique with PSM for estimating treatment effects. p -value for PSM method in brackets.

Technique	ATT [%]	ATC [%]	ATE _{obs} [%]
ML	0.63	-0.58	0.10
PSM	0.44 (0.69)	0.02 (0.98)	0.28 (0.76)

6.6 Subgroup Treatment Effects

The previous section looked at the global treatment effects and non-significant treatment effects were computed. However, looking at how these interventions affect subgroups and individual students is perhaps more valuable.

6.6.1 Subgroup Treatment Effects on the Control (ITC)

This section looks at the treatment effects on the control group. Figure 6.1 shows the histogram of individual treatment effects on the control, ITC, computed with the RF_{treat} model. Statistics of this histogram are presented in Table 6.8.

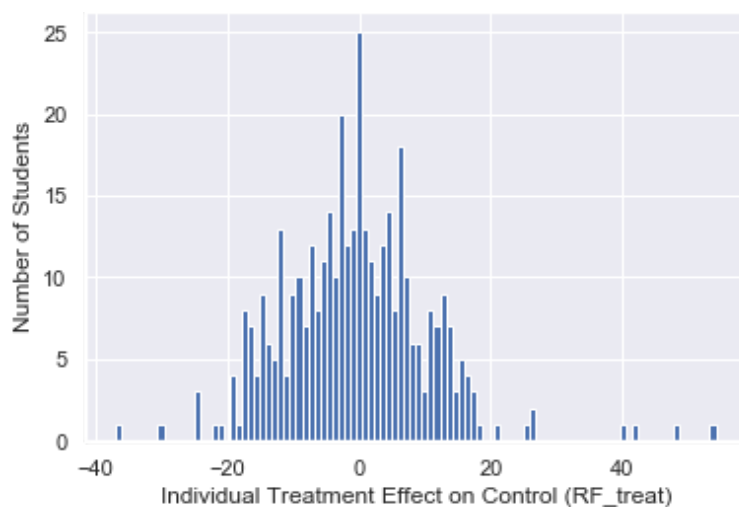


Figure 6.1: Individual treatment effects on the control (ITC) computed with the RF_{treat} model.

Table 6.8: Statistics for average treatment effects on control group, ATC, computed with the RF_{treat} model

Mean [%]	Std [%]	Min [%]	Max [%]
-0.58	10.9	-31.82	55.5

The majority of the control group students had ITC values within the -20% to 20% range although there were very large minimum and maximum values of -31.8% and 55.5%. Therefore although the ATC was -0.58% it can be seen that different control group students had different treatment effects. For some the treatment would be positive and for others it would be negative.

This now presents an important way of applying these results. First is the obvious way based on the question: *Would an individual student benefit from the intervention?* This method answers this question by simply computing the ITC for that individual control student. This method would simply identify a student as soon as possible in the semester and compute their ITC. The second way is related to the first but deals more with inference. The question is: *Who would benefit from the intervention?* That is, what are the characteristics of students who would benefit. This question is important since this would allow for targeting groups that might benefit more than other groups from the intervention.

6.6.1.1 Positive ITC Values

Here the most important features are looked at, shown in Figure 6.2 of the control students who received positive ITC values. These were computed using permutation importances (see Chapter 5.2.2.2.1).

The results show that the features March test PHYS1015, March test MATH1014, high school mathematics, March test PHYS1014 and age, are the top predictors in this group. The statistics of these features are shown in Table 6.9. The input features were all normalized. All the top features in the table are continuous and all mean values are below the control group mean, indicating that the control students for whom the ADU intervention is predicted to be effective, *performed worse than average* in the March test grades and are younger than the average age. This is suggesting that those control students who initially performed worse early in the year were predicted to receive the most benefit had they attended the interventions.

The students within the positive ITC subgroup that were predicted to benefit *substantially* from the intervention, i.e. >10%, were looked at next. What are the characteristics of those students who did not receive the treatment, yet will benefit greatly from the treatment? The

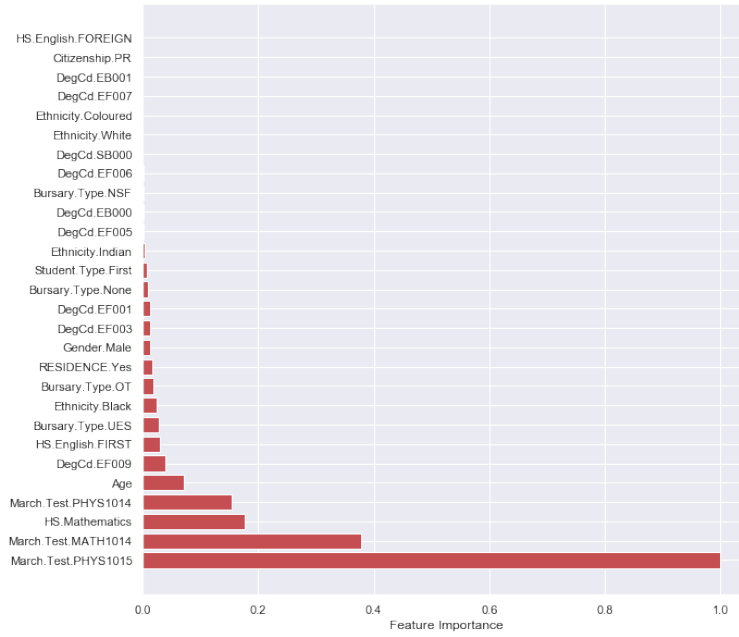


Figure 6.2: Top features for control group with positive ITC values.

Table 6.9: Statistics of top features for control group with positive ITC values ($n = 175$).

Feature	Mean [%]	Std [%]
March test P1015	-0.33	0.97
March test M1014	-0.08	0.80
March test P1014	-0.02	0.76
Age	-0.14	0.79
HS.Math.	-0.20	0.83

six features are presented in Table 6.10. The results show that March test PHYS1015 is 0.73 standard deviations below the mean grade. The actual non-normalized mean grade was 73.4% for the control group and this subgroup had a mean grade of 63.1%. Therefore these students could be advised that attending the ADU intervention may improve their final grade by at least 10%, since the ITC value is greater than 10%. This is under the assumption that the other grades are similar to what is found in Table 6.10. For this dataset, 54 control group students (6%) were predicted to benefit by more than 10%.

6.6.1.2 Negative ITC Values

The control students who were predicted to not benefit from the ADU intervention ($n = 209$), i.e. negative ITC values, were looked at next. The most important features are shown

Table 6.10: Statistics of top features for control group with positive ITC values above 10% (n = 54).

Feature	Mean [%]	Std [%]
March test P1015	-0.73	1.08
March test M1014	-0.12	0.72
March test P1014	0.01	0.64
Age	-0.05	0.99
HS.Math.	-0.33	0.88

in Figure 6.3 and are similar to those for the positive ITC values, but the values are quite different.

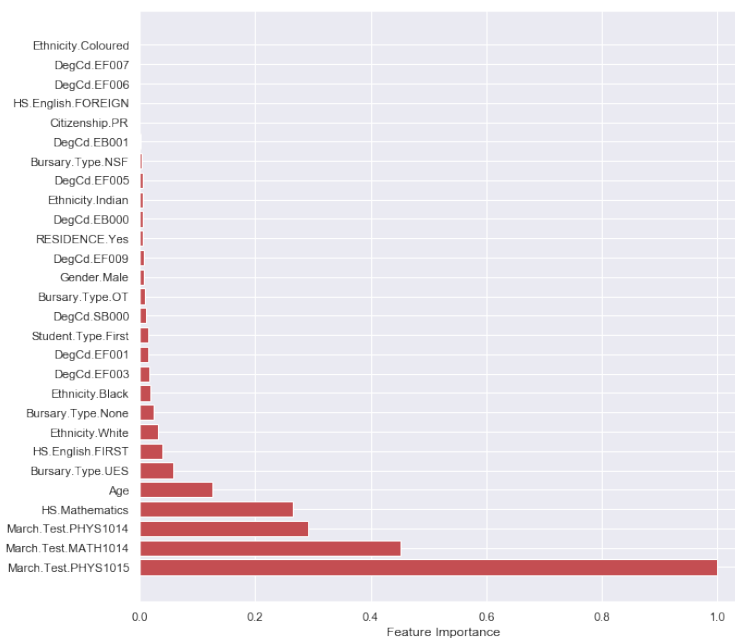


Figure 6.3: Top features for control group with negative ITC values.

The statistics of these features shown in Table for this group 6.11, show that the mean values are all positive suggesting that the control students who were predicted to not benefit from the intervention are those that had above average values for the March tests, high school mathematics and age.

These results suggest that for the control students, the initial grades are a determining factor of whether the treatment effects will be positive or negative. If the students had worse than average grades, and were younger on average, then the intervention would benefit them.

Table 6.11: Statistics for top features of control group with negative ITC values.

Feature	Mean [%]	Std [%]
March test P1015	0.26	0.93
March test M1014	0.06	1.12
March test P1014	0.01	1.15
Age	0.11	1.12
HS.Math.	0.16	1.09

Students who obtained ITC values less than -10% ($n = 71$) were also characterized. These are the ones who were predicted to be considerably worse off had they received the treatment. The statistics are shown in Table 6.12. What is interesting in the results are that the March test PHYS1015 and high school mathematics are double the values Table 6.11 but the other three have all tended close to the class average. Therefore those for whom the intervention is substantially counterproductive are those that scored 0.58% standard deviations above the mean for PHYS1015 and 0.45% standard deviations above the mean for high school mathematics. But these students were near to the average group values for the other features.

Table 6.12: Statistics for top features of control group with negative ITC values ($n = 71$).

Feature	Mean [%]	Std [%]
March test P1015	0.58	0.85
March test M1014	-0.03	1.33
March test P1014	-0.08	1.3
Age	-0.06	1.0
HS.Math.	0.45	0.85

6.6.2 Subgroup Treatment Effects on the Treated (ITT)

This section looks at subgroup treatment effects on the treated group. The histogram of ITT values is presented in Figure 6.4 and it shows that the majority is between -20% and +20% with some extreme values. Table 6.13 shows that this histogram has a positive mean value of 0.63%.

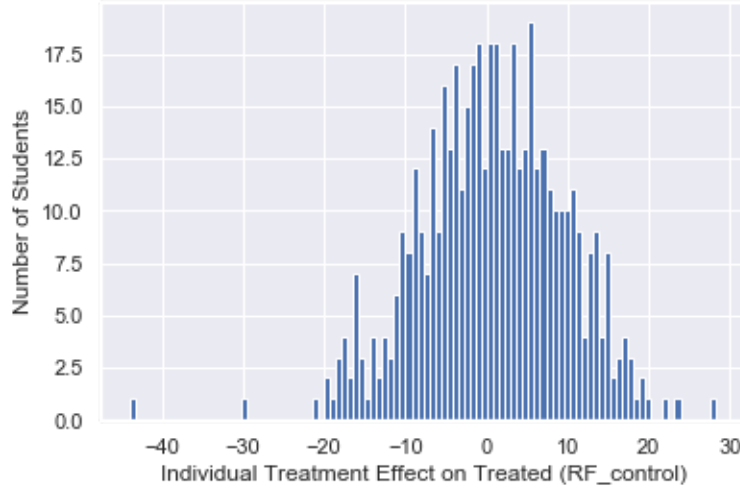


Figure 6.4: Individual treatment effects on the treated (ITT) computed with the $RF_{control}$ model.

Table 6.13: Statistics for average treatment effects on treated, ATT, computed with the $RF_{control}$ model

Mean [%]	Std [%]	Min [%]	Max [%]
0.63	8.9	-43.5	28.2

6.6.2.1 Positive ITT Values

In a similar way to the section on positive ITC values, this section looks at those treated students who had positive ITT values ($n = 263$). The top features are shown in Figure 6.5 and the statistics of these top features is shown in Table 6.14. Similarly to the ITC results, those treated students who are predicted to benefit from the intervention, all had negative mean grades which indicates they performed worse than average. However, the age feature here is positive as opposed to the negative age feature for the positive ITC values.

What are the characteristics of the treated students for whom the intervention was predicted to help *substantially*. The students with ITT values above 10% were selected and the statistics for the features are presented in Table 6.15. Compared with Table 6.13, the negative features

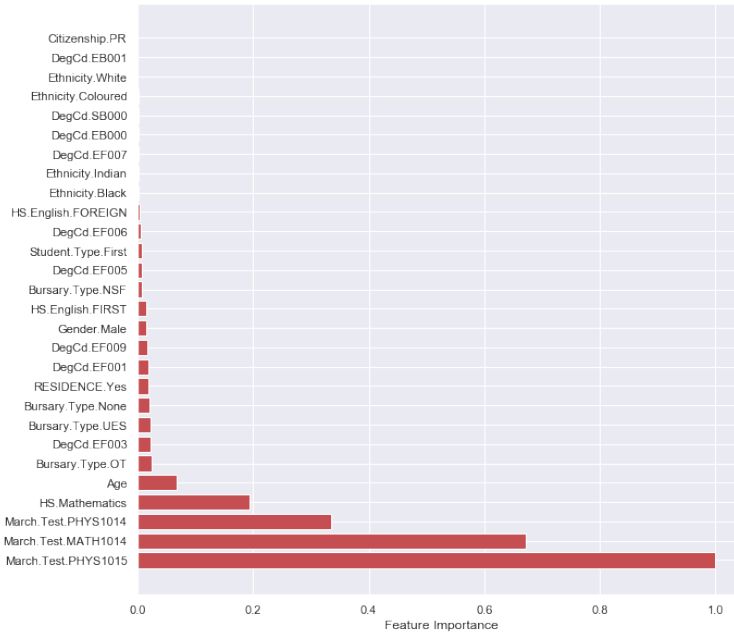


Figure 6.5: Top features for treatment group with positive ITT values.

Table 6.14: Statistics for top features of treatment group with positive ITT values (n = 263).

Feature	Mean [%]	Std [%]
March test P1015	-0.18	1.01
March test M1014	-0.09	0.99
March test P1014	-0.11	1.04
Age	0.02	1.02
HS.Math.	-0.15	0.93

are much more negative and the positive age feature has increased from 0.02% to 0.22%. The characteristics for the treated students who benefitted the most are similar to the control students who are predicted to benefit the most. They both have large negative values for March test PHYS1015, although the control group was more negative (-0.73 vs -0.47). High school mathematics for this group was very similar to the positive ITC group (0.33 vs 0.3). What is interesting is that the age in the treated group was a relatively large 0.22 vs -0.05 for the control group. This suggests that for those that receive the treatment, the older students will receive the most benefit. These students who obtained -0.47% in March test PHYS1015 obtained an actual mean value of 64.5%. Therefore for this group that obtained lower than this grade, combined with the values of the other features, the ADU intervention is predicted to improve their grade by 10% compared with what they would have had, had they not received the treatment.

Table 6.15: Statistics for top features of treatment group with ITT values above 10% (n = 73).

Feature	Mean [%]	Std [%]
March test P1015	-0.47	1.07
March test M1014	-0.11	0.84
March test P1014	-0.14	1.01
Age	0.22	1.28
HS.Math.	-0.30	0.88

6.6.2.2 Negative ITT Values

The treated students with negative ITT values (n = 231) were looked at next. These treated students were predicted to not have benefitted from the treatment. The top features and statistics for the treated group is presented in Figure 6.6 and Table 6.16. Positive mean grades values were found again, but negative age mean values.

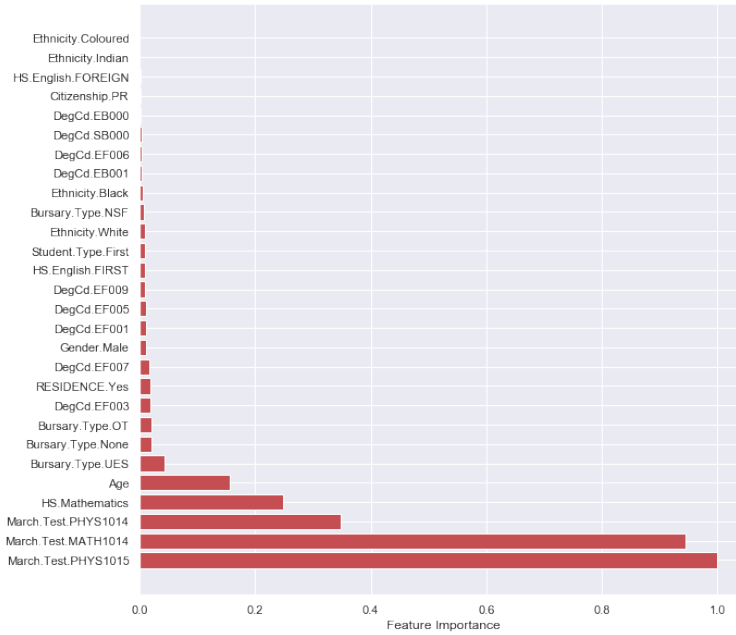


Figure 6.6: Top features for treatment group with negative ITT values.

The final subgroup was those treated students with ITT values smaller than -10%. These were students who received the intervention but were predicted to be worse off by more than 10%. The statistics for their top features are shown in Table 6.17 and, similar to the control group that had highly negative ITC values, this group had much higher average values for the March test PHYS1015 and high school mathematics.

Table 6.16: Statistics for top features of treatment group with negative ITT values ($n = 231$).

Feature	Mean [%]	Std [%]
March test P1015	0.22	0.93
March test M1014	0.11	1.00
March test P1014	0.13	0.93
Age	-0.02	0.97
HS.Math.	0.17	1.05

Table 6.17: Statistics for top features of treatment group with negative ITT values below -10% ($n = 54$).

Feature	Mean [%]	Std [%]
March test P1015	0.36	0.91
March test M1014	0.06	1.13
March test P1014	0.09	1.09
Age	0.38	1.48
HS.Math.	0.44	0.92

From the analysis of treatment effects for subgroups, it was found that for both control and treated groups, those students that had much lower March test grades for PHYS1015 and lower high school mathematics grades, were predicted to receive the most benefit from the intervention. And vice versa, those that performed better on average, were predicted to not receive the benefit. Therefore, although the interventions are open for any student to attend, the results suggest that focus could be placed on those subgroups where the students performed worse than average in those courses. The results show that these students receive the most benefit.

6.6.3 Individual Treatment Effects Using SHAP

In the previous section subgroups with different ITC and ITT values were analyzed to see what characterizes these groups. The subgroup analysis allows for marking out groups that could potentially benefit from the intervention. This section takes one step further to now focus on individual students by generating SHAP values as discussed in Chapters 5.2.2.3.2 and 5.5. Using these methods together with the methods shown above this method (1) predicts the final grade of any individual student, (2) predicts the treatment effect and (3) analyze the specific features that describe this student. This will allow for answering an important question: What will the student's final grade be and how much will the intervention help this specific student.

To illustrate this, a random control student was selected with an ITC value of 13.97%. For this student the SHAP values are generated in Figure 6.7 and the very low grades for March test PHYS1015, March test MATH1014 and high school mathematics describe (characterize) this student. Using the SHAP predictions he will obtain 35.44% for his final grade (much lower than the average base value of 62.34%) and the intervention will help him by 13.97%. Therefore adding 13.97% to 35.44% results in 49.4% which is close to passing. This illustrates how to use this method. Therefore this student can be recommended to receive the intervention.

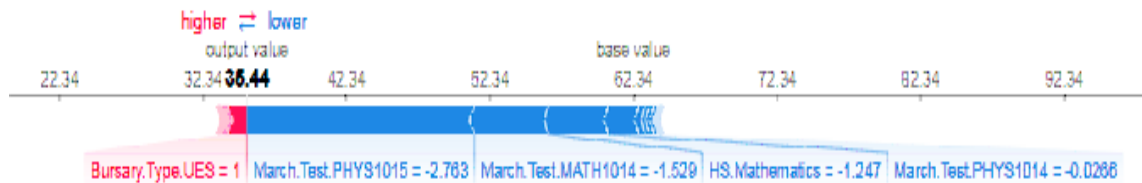


Figure 6.7: SHAP results for control student with ITC value of 13.97%.

Another example is presented where a control student obtained an ITC value of 11.0%. The SHAP values are presented in Figure 6.8 and the student is predicted to pass (52.81%), even without the intervention. But is predicted to have obtained 11.0% more had they received the intervention.

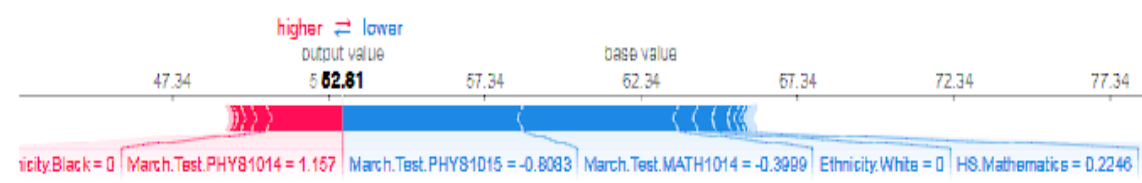


Figure 6.8: SHAP results for control student with ITC value of 11.0%.

Control individuals that had negative ITC values were looked at next. The first had an ITC value of -14.6% and their SHAP values are shown below in Figure 6.9. Their predicted final grade of 84.34% was much higher than the average and they also had grades above the mean for each course. Therefore using this technique, this student would not be recommended to receive the intervention since it would result in a reduction in predicted grade of 14.6%. This method again shows that those who perform better than average initially might not benefit from the intervention.

A treated student with an ITT value of 11.0% was chosen next, with their SHAP values shown in Figure 6.10. First of all, the SHAP values describe the student as performing well in March test PHYS1015, being slightly older than average, being in residence, but performing poorly

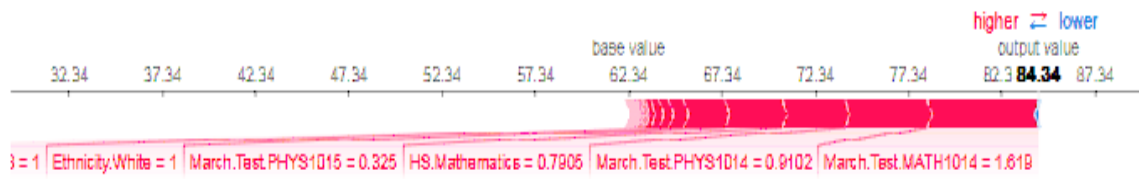


Figure 6.9: SHAP results for control student with ITC value of -14.6%.

in March test MATH1014 (-0.57%). For this student the ITT of 11.0% suggests that without the intervention, they would have obtained $66.62\% - 11.0\% = 55.62\%$ for their final grade.

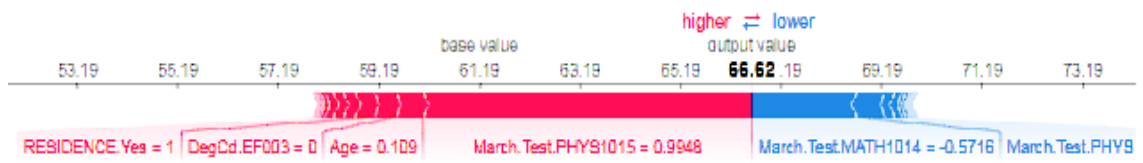


Figure 6.10: SHAP results for treated student with ITT value of 11.0%.

The next student had an ITT value of 15.3% and their SHAP values are shown in Figure 6.11. Although this student also had a relatively high positive ITT value compared with the previous student, their characteristics are not too similar. For this student, their March test PHYS1015 value was -0.19% and their final predicted grade was 79.2%. This student therefore substantially improved their grades. This method suggests that without the intervention, they would only have obtained $79.2\% - 15.3\% = 63.9\%$.

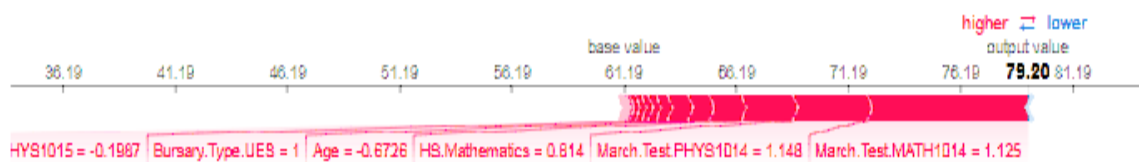


Figure 6.11: SHAP results for treated student with ITT value of 15.3%.

Finally a student with a relatively large negative ITT value of -13.1% was analyzed. Their SHAP values are shown in Figure 6.12. This treated student was predicted in March to obtain a final grade of 57.06% but the ITT results suggest that without the intervention, they would have obtained approximately 70% for the final grade ($57.06\% + 13.1\%$). For this student for whom the intervention is counterproductive, they performed above average in the March test PHYS1015 but below average in March test MATH1014 and were older on average than the rest of the treated students.

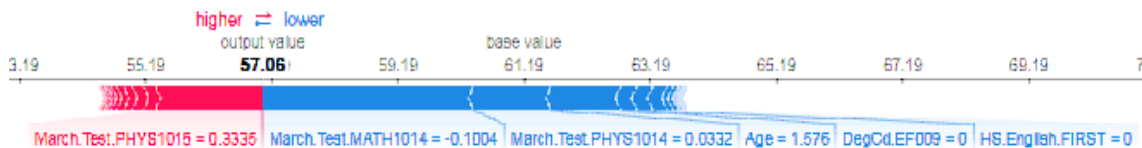


Figure 6.12: SHAP results for treated student with ITT value of -13.1%.

6.7 Discussion

Although using ML to predict student performance is common in the literature (see Table 4.1), using ML to predict treatment effects of interventions, i.e the MLTE method, is relatively new and promises much. The value in this method, developed from that proposed by Beemer et al. (2017) [4], is to predict not only average treatment effects but also individual treatment effects. It was shown that although average treatment effects on the overall cohort could be insignificant, individual treatment effects vary much for individual students. ML has been traditionally used to predict At-Risk students, which is certainly valuable. However, using ML to predict how much an intervention could help a student, or group of students, is also highly valuable. The author believes that perhaps the most important application of the MLTE method is to firstly predict the final grades of students and then predict how much the intervention would help, i.e the individual treatment effect. This would allow for targeting certain subgroups and encourage them to attend the intervention. Of course, no student should be disallowed from attending the interventions and all should be welcome. The subgroups and individuals were characterized using PI and SHAP, respectively, which is the first known case of carrying this out. This study was performed on historical data and future work will include making actual out-of-sample predictions on new students in order to test how accurate this method is.

This method further found that students that were predicted to have positive individual treatment effects, performed poorly in the March tests, i.e. had negative z-scores. And students that were predicted to have negative treatment effects, performed well (i.e. positive z-scores were found). This is an interesting finding and suggests that those who scored poorly in the initial tests might be more motivated to use the intervention more effectively. Also, these poorly performing students have more room for improvement than those that initially performed well. It would be valuable future study to study student motivation in interventions such as this one.

The MLTE method used in this study was validated using PSM and similar results were obtained. This is the first known study that has validated the MLTE method used by Beemer et al. (2017) [4]. This method is not limited to the field of predicting student

performance and any field where interventions are carried out, the varying treatment effects can be predicted. It does however have some limitations. The first is that it was not possible to know what the true treatment effects were, i.e. there was no “ground truth”. However, some confidence was found in the MLTE method since PSM is a traditional technique that has been used extensively in various fields [11, 12, 144, 145]. A second limitation is that the two groups, control and treatment, showed much imbalance in the features (see Table 6.2) and therefore using a model trained on one group to predict the counterfactual of the other group, could potentially introduce bias into the predictions [4]. However, again the PSM technique balanced the features in the two groups and computed the treatment effects which compared relatively well with the MLTE method. However, the MLTE method might compare well with PSM in this specific case, but it might be different in other situations. It is suggested for further work, to perform simulation studies where known bias could be introduced into the groups, and this method be further validated. Another limitation is that the counterfactual predictions are also dependent on how well the ML models carry out predictions. The best predictions in both the control and treatment models had average values of over 6% MAE. These prediction errors may also contribute to errors in the counterfactual predictions.

In terms of the models used, tree-based models outperformed the lasso model on both datasets. This is similar to Beemer et al. (2017) [4], where lasso, RF and boosting methods also perform better in one analysis, but then the results become mixed in the others. Further studies would include a larger range of ML models. Whereas Beemer et al. (2018) [9] used lasso and RF, this study used multiple models that were optimally found via hyperparameter optimization (see Chapter 2.4 and 6.4), and only then were the counterfactual predictions made using the best performing models. It is believed that without carrying out model selection via hyperparameter optimization, inferior results will be obtained.

Potential issues may arise from the features that are demographic, for example the ethnicity and gender of the students. This was previously discussed in Chapter 5.7. In this chapter, the main features were grades features and a single demographic feature, age. However, in other studies, ethnicity and gender might play a more significant role. It is important to show proper sensitivity when dealing with these types of predictors and is best to provide the results to experienced educators that have experience in handling this. It is important to note here that simply because features are identified as predictors of the outcome, it does not mean that they are causal. Further study would be required to identify any hidden confounding features. Nevertheless, the aim is to assist students to pass, and identifying which types of students would benefit is seen as highly valuable. These methods simply identify predictors of treatment effects, not the causes. The treatment that is being studied is the intervention itself.

Further on the topic of features, the better the features used in the training of the models, the potentially better the predictions will be. This study simply used the available features from the university as well as intervention attendance. It is therefore limited by the quality of the features used. More consideration is required in future studies to determine what new features might add substantially more value to the training of the models. It is proposed that better features would reduce the MAE, which could potentially result in more accurate counterfactual predictions.

6.8 Conclusion

This chapter extended the work in the literature [4, 9] that uses ML to measure treatment effects in observational studies, which this study calls MLTE. This study more thoroughly describes the different treatment effects and is the first, as far as is known, that validates the MLTE method against traditional PSM methods. The main outcome of using this technique was to essentially estimate counterfactual quantities that are used to compute the average treatment effect (ATE), the average treatment effect on the treated (ATT) and the average treatment effect on the control (ATC).

Furthermore, the importance of computing not only global treatment effects but subgroup and individual treatment effects was presented. The MLTE method was extended to be applied to individuals and sub-groups, by computing ITT and ITC, individual treatment effects on the treated and control, respectively. Specific characteristics of those students for whom the intervention would and would not be beneficial were found and it was shown that for both control and treatment groups, initial grades are a strong predictor of whether the intervention will benefit the students or not. This is important since subgroups can be identified that could be encouraged to attend the intervention.

Finally, using SHAP local explanations and predictions, the final grade of individual students could be predicted as and be used in conjunction with individual treatment effects to ascertain whether the intervention could help them pass.

Chapter 7 Treatment Effects of Online Videos Using Difference-in-Differences

7.1 Introduction

This chapter presents a study on treatment effects of supplementary online videos on student performance and falls under the third overall objective of this thesis: measuring treatment effects of interventions. Whereas the previous chapter looked at treatment effects of a face-to-face supplementary tutorial, this study focuses on supplementary videos as the intervention. This study was not performed on the same cohort of students studied in the earlier chapters but on a 2017 cohort of first year engineering students taking the PHYS1015 engineering mechanics course. Furthermore, to measure treatment effects, this study did not make use of ML, but of a statistical technique called difference-in-differences (DID).

Although the use of supplementary online videos to improve student learning promises much, it has produced mixed results. For example, the literature shows videos having a positive impact [148, 149, 150], videos having a negative impact [151, 152] and videos having no impact on performance [153, 154]. The mixed results have been attributed to different learning strategies [152], passive learning [154], attending less face-to-face lectures [154] and videos potentially not being effective across all fields of study [150].

Although these may be factors, this study proposes that a potential reason for the mixed results is due to the majority of the studies being observational, not randomized control [10], and therefore were not necessarily measured using the appropriate techniques. If interventions such as online videos are to be effective, it is crucial that appropriate measurement techniques be utilized. In observational studies, due to potential confounding variables, the grades of groups that utilize online videos cannot simply be compared with groups that don't. An example is Traphagan et al. (2010) [151] who, in their observational study, first found that those watching videos performed worse than the no-videos group. Thereafter they controlled for GPA and found that the two groups performed similarly. Furthermore, in observational studies, those that watch a large number of videos can't simply be compared with those that

watch a smaller number. Leadbeater et al. (2013) compared those watching more than five hours with those watching less than fifteen minutes and found no difference in performance [154]. The question being asked here is: could the mixed results also be due to the wrong measurement techniques employed to measure video impact in observational studies?

In order to address this problem, this study first captured detailed video watch time of a cohort of first year engineering mechanics students and compared the watch time with academic performance over multiple assessments. To the author's knowledge, this is the first known study that has captured actual watch time for each student. The aim was to determine if more video watching resulted in better performance. The students self-selected to watch the videos and therefore this was an observational study. What was found was that there was no association between the amount of video watching and academic performance.

Based on these findings, and due to only possessing grades information on the students, to measure the effects of the videos on performance, it was proposed to use the DID method that is able to capture treatment effects of an intervention in an observational study. The treatment effect in this study is defined as the average difference in grades between a treatment group and a control group. This method has thus far not been used in the literature on student success and is proposed since it can capture the treatment effects without needing a large number of features describing the students. It simply requires the grades of the students before and after the intervention.

7.1.1 Main Contributions

Based on the above, the main contributions of this observational study are as follows:

- This is the first known study that measured detailed video watch time for students and compares with performance in tests and exams. This is presented in Sections 7.3.1 and 7.4.1.
- Section 7.4.1, using data from multiple assessments of a 2017 cohort of 1st year engineering students, shows that there is *no association* between number of videos watched and academic performance. This study proposes that this is due to it being an observational study which requires a different technique for measuring the effect of videos.
- It is argued in Section 7.2.2.1 that it may be more insightful and accurate to measure video effectiveness using DID that looks at the *change* in grades, i.e. how the videos affect improvements in grades, not absolute grade. This is the first known study that applies this technique to measure treatment effects of online videos.

- Although no association was found using the techniques in Sections 7.3.1 and 7.4.1, significant treatment effects (DID values) were computed using DID as shown in Section 7.4.2.2.
- Non-significant treatment effects (DID values) were computed for the overall cohort of students (Section 7.4.2.1), but Section 7.4.2.2 shows that certain sub-groups are impacted by the videos and have significant DID values. There is a tendency for the initial grades at the beginning of the time span to be a predictor of video effectiveness. It was found that those initially failing tend to be impacted by the videos; specifically those that obtained 40%-50% initially. Initial grade combined with the videos has a significant impact on grades.

7.2 Background

This section presents related work on online videos, various measurement techniques used in the literature to measure impact of videos, and the theory of DID.

7.2.1 Online Videos

According to Muller (2005), the belief that multimedia will improve learning has been around for almost a hundred years [155]. For example, Thomas Edison believed that the motion picture would revolutionize learning by supplanting the textbook. Today mobile device use and internet access is prevalent and greatly affects the way students are accessing information. The area of multimedia that has seen the most growth recently is that of online education videos [156]. There are a range of methods used in online learning. For example, recording the lecture by the instructor and then making it available to the students online is a method employed by the Massachusetts Institute of Technology. YouTube also has a vast number of academic videos on various topics. The flipped classroom [157] is another application of online videos where the students watch video lectures and then attend class to discuss problems related to the video content. The idea there is to generate more active learning in the classroom. The Massive Open Online Course (MOOC) [158] has also seen increasing popularity over the last decade. The flipped classroom allows for guidance by the instructor but MOOCs are generally self-study and have relatively low completion rates.

The point here is that there are a variety of methods employing online learning through videos and although the use of videos promises much, the literature reports on varying impact on student learning. A major reason proposed in the literature as to why videos do not deliver what they promise, is due to superficial or passive learning [152, 154]. Students may feel

that simply by watching videos, they are learning. Hansch et al. (2015) [159] warns that it is problematic to equate engagement with learning. Guidance by instructors and the help of student peers is also seen as crucial to the learning process [160]. Although the students are doing the learning via videos, the instructor and/or student peers are still seen as crucial. Furthermore, attitude [161], motivation [162, 163] and attention [164] of the student have also been shown to be important in learning from multimedia. When investigating why videos are or are not effective, the above-mentioned reasons are certainly valid ones.

7.2.2 Measurement Techniques

Although much study has been performed on online learning, and various conclusions have been made regarding the varying effectiveness of the videos, the author believes that two important items are missing: (1) actual and detailed video watch time for each student, and (2) using the appropriate techniques for measuring video effectiveness in observational studies. According to Muller et al. (2005) [155], there needs to first be a reliable evaluation of methods before the teaching methods could be changed and re-tested. Before saying that one instructional technique is more effective than another, evaluating that technique is crucial. Therefore within the larger framework of learning using videos, this study is positioned within an important sub-section of measuring the effect of videos.

The measurement technique is based on the type of study being performed. As presented in detail in Chapter 6.2, in general there are two types of studies to measure intervention or treatment effects. Observational studies are studies where the participants (i.e. students) self-select to utilize the intervention (videos). This is in contrast to a RCT where participants are randomly assigned to either a control or treatment group. In the literature referenced earlier, the majority of the studies were observational [149, 150, 151, 152, 154]. Other literature made use of RCTs to measure treatment effects [148, 153, 165]. Although RCTs are seen as the “gold standard” for evaluating the efficacy of an intervention [10], the problem is that there may be ethical issues [10] related to randomly assigning students to watch or not watch videos. Furthermore, benefits of observational studies are that the participants can be viewed in their natural habitat; i.e. participants will act naturally when using the intervention and not be affected by being monitored in a study setting. Therefore simply allowing the students to self-select to watch the videos is seen as the appropriate way to provide this intervention. Although seen as beneficial, self-selection generally produces bias in the treatment effect due to imbalanced features.

Of those observational studies studying the effects of online videos on performance, Traphagan et al. (2010) [151] utilized one-way multivariate analysis of covariance (MANCOVA). Wieling et al. (2010) [149] and Williams et al. (2012) [150] performed multivariate regression to

measure the effect of online viewing on performance. Leadbeater et al. (2013) [154] used simple correlation to study how amount of video watching impacts performance. The problem with regression and correlation studies are that they essentially measure correlation and not causal (treatment) effects.

When measuring treatment effects in observational studies, the aim is to balance the features so that the treatment group and control group have balanced features. The aim is to create a counterfactual group to compare with the treatment group. A main way of carrying this out is to use PSM [145] which is beneficial if the study has a *range of features* for the participants and the effects are measured using cross-sectional data. In the absence of a range of descriptive features, the DID method can be used, which utilizes panel data; i.e. data captured over time. At the time of the study presented in this chapter, the author did not have access to a range of features (as in the previous chapters), but only panel data indicating the grades of students during the semester. Therefore DID was seen as the most appropriate method to measure treatment effects.

7.2.2.1 Difference-in-Differences (DID)

DID takes into account *changes* in outcome (i.e. grades) in order to measure effects. The intuition behind DID can be explained as follows. Consider measuring the change in grades (positive or negative) of the students that self-selected to watch videos (treatment group), after watching the videos. There could be the erroneous thought that the change in grades was due to the videos (the intervention). However, to properly measure the videos' effect, there needs to be a counterfactual group, i.e. a similar group that did not receive the intervention (control group). The counterfactual control group's average change in grades should be measured and compare with the treatment group change in grades. Whenever treatment effects are measured, there is the requirement to compare a treatment person/group with a counterfactual person/group. Thus far in the literature on measuring the effectiveness of videos on academic performance, this is missing in the observational studies. The control group is the counterfactual group that aims to explain what the treatment group would have obtained, had they not received the treatment. The DID method aims to remove or control for, the effects of selection bias. In the literature there are examples that use this technique to measure important concerns such as the effect that raising the minimum wage has on employment [166] and how disability benefits effect time out of work after injury [167]. However, this technique has not been used to measure treatment effects in online videos.

DID is calculated as per Equation 7.1. $\bar{y}_{tf}$ and $\bar{y}_{ti}$ refer to the average grade at the end and beginning of the time span, for the treatment group, respectively. Similarly, $\bar{y}_{cf}$ and $\bar{y}_{ci}$ refer to the average grade at the end and beginning of the time span, for the control

group, respectively. In this study, the beginning and end of the time span is taken to be grades obtained at the mid-year (before semester two begins), and final grades obtained at the end of the year. More details on this later. Equation 7.1 provides an intuition for the treatment effect, or DID. It shows that the change in the control group's average grade from the beginning to the end is subtracted from the change in the treated group's average grade over the same time.

$$DID = (\bar{y}_{tf} - \bar{y}_{ti}) - (\bar{y}_{cf} - \bar{y}_{ci}) \quad (7.1)$$

Although Equation 7.1 is a simple way to calculate the DID value, computing p -values and statistical significance is required and is based on Equation 7.2, where y refers to the grade (either Mid-Year or Final-Year); PrePost is a dummy variable indicating whether the grade is pre (Mid-Year) or post (Final-Year); Treat is a dummy variable indicating whether the student received treatment (videos) or not. Finally, PrePost*Treat is an interaction variable. In terms of the coefficients, β_0 refers to the control group's initial grade; β_1 is the difference between the treatment and control group's initial grades; β_2 indicates how much the control group changes grades over the time period of the intervention; and β_3 is the DID value indicating the treatment effect. The coefficients and p -values were calculated using Python and Excel by implementing Equation 7.2.

$$y = \beta_0 + \beta_1 PrePost + \beta_2 Treat + \beta_3 (PrePost * Treat) \quad (7.2)$$

7.2.2.2 Parallel Trends Assumption and Limitation

A major benefit of DID is that by comparing the change in grades of the treatment group with those of the control group, it takes into account the changes of all the features over time. That means that it is not necessary to measure multiple features for each group; the change in the outcome is sufficient. However, this benefit is based on the parallel trends assumption which states that the treatment (videos) group, in the absence of the treatment, would have the same change in grades as the control group (no-videos) over the time period considered. A method of proving the parallel trends assumption is to compare the control group's change in grades with the treatment group's change *before* the treatment group received the treatment (videos). Unfortunately for this study, no prior data was available showing who accessed the videos and who didn't, and therefore there was no possibility of measuring prior changes in grades for these groups. Therefore although this study is based on the parallel trends assumption, there was no way to prove this assumption.

7.3 Methodology

The main idea of this chapter was to use appropriate methods to measure treatment effects of online videos on academic performance using actual video watch time for each student. The methodology is presented here and describes two main components of this main idea. The first presents video watch time being a predictor of grades; and the second concerns measuring treatment effects using difference-in-differences.

7.3.1 Video Watch Time as a Predictor of Grades

This section studies the actual video watch time as a predictor of student performance. It is based on data from the 2017 engineering mechanics course, PHYS1015. Due to students self-selecting to watch the videos and thereby resulting in this study being observational [10], the aim was to see if the results would be similar to the literature, i.e. there is no relationship between videos and performance [151, 152, 153, 154]. This section focused on comparing the students' video watch time with the results of four assessments carried out in the second semester of 2017: two major tests (September Test and October Test), a final November Exam and the Final-Year grades for the course. The Final-Year grades are not a single assessment, per se, but a combination of all the major tests and tutorial tests during the semester.

To determine if there was any association of minutes watched with performance, the following analyses were carried out for each of the four assessments in 2017:

1. Plots were generated to show grades versus minutes watched for all the students, for each test or exam.
2. Correlation coefficients were calculated. Due to the skewness of the data, Spearman's correlation was used.
3. Linear regression was used to determine if the number of minutes of video watched was a significant predictor of student performance.
4. T-tests were performed to measure any significant difference between the means of the videos- and no-videos groups.

7.3.1.1 Creating the Videos

Supplementary videos were created for a first year engineering mechanics course. The videos were similar to Khan Academy style [168]. During the semester, 80 videos were recorded that covered chapters 12 - 15 of Engineering Mechanics: Dynamics [169]. These videos were uploaded onto Panopto™, a video hosting platform, and the students were able to access the videos at any time and place. The availability of the videos and the ability for students to utilize the videos at their own discretion resulted in a wide range of number of videos watched (i.e. number of views or number of minutes watched). Panopto™ was chosen as the platform since it provided very detailed viewing statistics for each student, that is, it was able to record the exact number of minutes watched per student for each video. At the beginning of the course, 1026 students were registered. However, varying numbers of students took the tests which are discussed later. The videos were typically between five to ten minutes long and covered both conceptual and practice content.

7.3.1.2 Assessments

Grades for the different assessments were compared with video watch time. The first assessment looked at was the September Test 2017 which covered material from Chapters 12 and 13 of Hibbeler et al. (2017) [169] and 44 videos were created covering these chapters. Therefore only the videos that were directly applicable to the content of the September Test 2017 were considered. A total of 953 students took the September Test 2017. Students that watched any number of videos (minutes >0) were included. Using this criteria, 652 students qualified as having watched the videos to varying degrees.

The next assessment was the October Test 2017, where 951 students took the test and based on the same criteria as above, 499 students made use of the videos to varying degrees. Here again, only the videos that were directly applicable to the content of October Test 2017 were considered. Therefore only the videos that were made after September Test 2017 up until October Test 2017 were analyzed, which focused only on Chapter 14 material. 19 videos were created for Chapter 14.

The November Exam comprised Chapters 12 to 15, 1022 students took the exam, and 609 students made use of the videos to varying degrees. Here student video data for the entire semester was looked at, since the November Exam covered all the work for the semester. An extra 17 videos were created to cover Chapter 15 which then gives a total of 80 videos created.

The overall final grades (Final-Year) for the course were also evaluated in light of the videos and 705 students made use of the videos to varying degrees and the video data for the entire semester was considered again.

7.3.2 Difference-in-Differences (DID)

DID estimates were computed for different cases, presented next.

7.3.2.1 DID for Entire Cohort

DID was first computed for the entire cohort of students. This was carried out for two time spans: Mid-Year to Final-Year and Mid-Year to November Exam. The aim here was to see if the videos had an effect on the entire group that watched them.

7.3.2.2 DID Based on Initial Grades

DID estimates were also computed based on whether students initially passed or failed at Mid-Year. The aim was to see if initial performance was an indicator of treatment effects. Thereafter, the initial Mid-Year grades were further broken down into ranges of grades from 30% to above 80% in ranges of 10%. The aim here was to see if different ranges had different treatment effects.

7.3.2.3 Panel Data

DID estimates are calculated from panel, or longitudinal data. In this study, DID was computed for two time spans, one from Mid-Year 2017 to November Exam 2017, and the other from Mid-Year 2017 to Final-Year 2017. This is because the November 2017 Exam was a stand alone assessment whereas the Final-Year 2017 grades were a combination of Mid-Year 2017 results as well as all the 2nd semester tests and tutorial tests. There might be different DID values for these two time spans and the aim was to investigate the difference.

7.3.3 Limitations and Assumptions

This study analyzes only the aggregated watch time (or number of views) and does not investigate aligned watching. For example, two students that both watched 100 minutes

of video may have watched different videos. A future study will be to investigate aligned watching. It is predicted that aligned watching might produce different results.

Further limitations of this study are that only a single year's cohort of students is analyzed which limits the ability to generalize.

This study is limited to panel data that only describes the initial and final grades of the students. The author did not have access to any demographic or other features describing the students, as in the previous chapters. The only information available to the author was the grades information, which makes using the DID method suitable.

7.4 Data Analysis

7.4.1 Video Watch Time as a Predictor of Grades

Figure 7.1 plots each student's grades for September Test 2017 against their accompanying minutes of videos watched. As stated earlier, there appears to be no correlation between the number of minutes watched and the grade for August Test 2017. There seems to be random scatter of the data points.

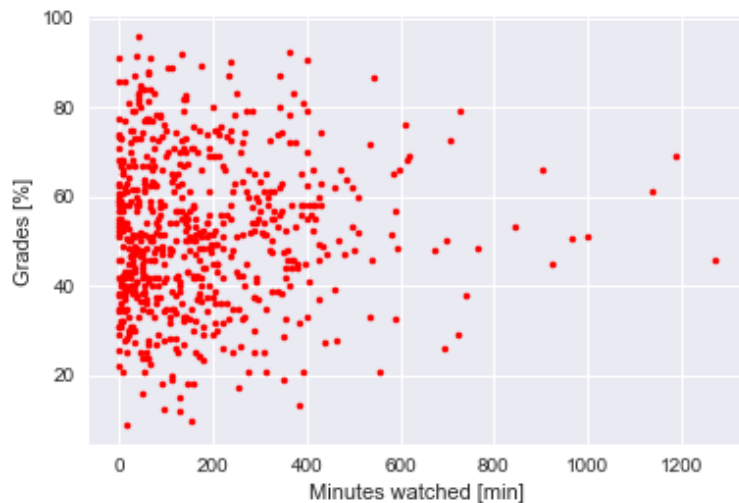


Figure 7.1: September Test 2017 grades of videos-group.

Although the distribution of grades is approximately normal (mean = 51.96%, std = 17.00%) for the videos group (see Figure 7.2a), the independent variable minutes watched, is skewed to the right (see Figure 7.2b). The right-skewing is similar to drop out rates seen in the literature where many students begin an online course but as time goes by the drop out rates increase [170]. Due to the skewness, Spearman's correlation was calculated as 0.051. The

results suggests that there is no correlation between the number of minutes watched and the September Test 2017 grades. Furthermore, carrying out linear regression on the data a β coefficient of 0.0035 was obtained and a p -value of 0.33 which indicates that as a whole, there is no relationship between the minutes watched and grades. To further study if there was any difference between the videos and no-videos groups, a t-test was conducted to measure differences between means. For September Test 2017, the mean for the non-video group (263 students) was 49.74%, and for the videos group (621 students), 50.99%. The videos group averaged 1.25% higher than the no-videos group, however the t-test resulted in a p -value of 0.35 indicating no significant difference between means at $\alpha = 5\%$.

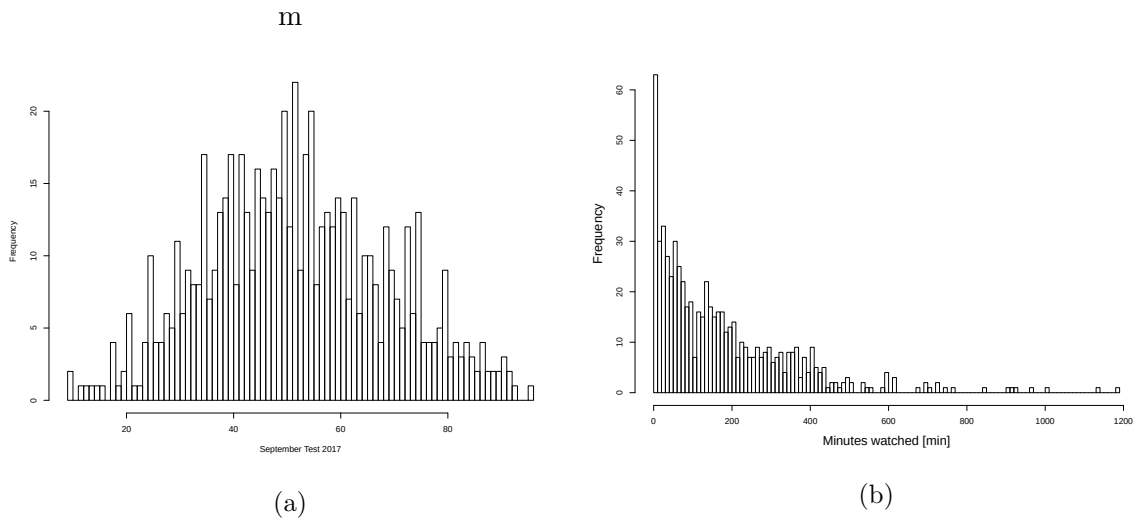


Figure 7.2: Histogram of September Test 2017. (a) Normal distribution of grades of videos group. (b) Right skewed distribution for the number of minutes of videos watched.

The results from this analysis suggest that the videos are not beneficial to the students in improving the academic performance. According to the literature, there may be a couple of reasons. First, the videos may encourage superficial learning [154]. That is, students may watch the videos passively and think that simply by watching, they are learning. Second, the students that watch a large number of videos may be weaker students to begin with and so even after using the videos, their grades might still not compare well with the stronger students. This second reason implies that the students having different starting points. This second reason is used to argue that the lack of association is due to this being an observational study that doesn't take into account these confounding variables, such as different starting points and motivation factors.

Figure 7.3 shows the grades and minutes video watched for, top to bottom, October Test 2017, November Exam 2017 and Final-Year 2017. Again, for October Test 2017, there appears to be no correlation between number of minutes watched and the student grades. Spearman's correlation was calculated as -0.011. Fitting a linear regression model, a beta coefficient of 0.000 (p -value = 0.99) was computed, which is not significant. This indicates there is no

relationship between minutes and grades. The average grades for the videos and no-videos groups were 54.2 % and 52.1 % respectively and t-test results actually show that the mean of the videos group is significantly higher ($p = 0.016$) than the no-videos group. The reason for this significance may be due to the time span between the September Test 2017 and October Test 2017 being short, only one month. This suggests that the Test content and the videos were very focused; they only covered one chapter. There was therefore a low variance in the range of material covered and perhaps this low variance made the videos effective.

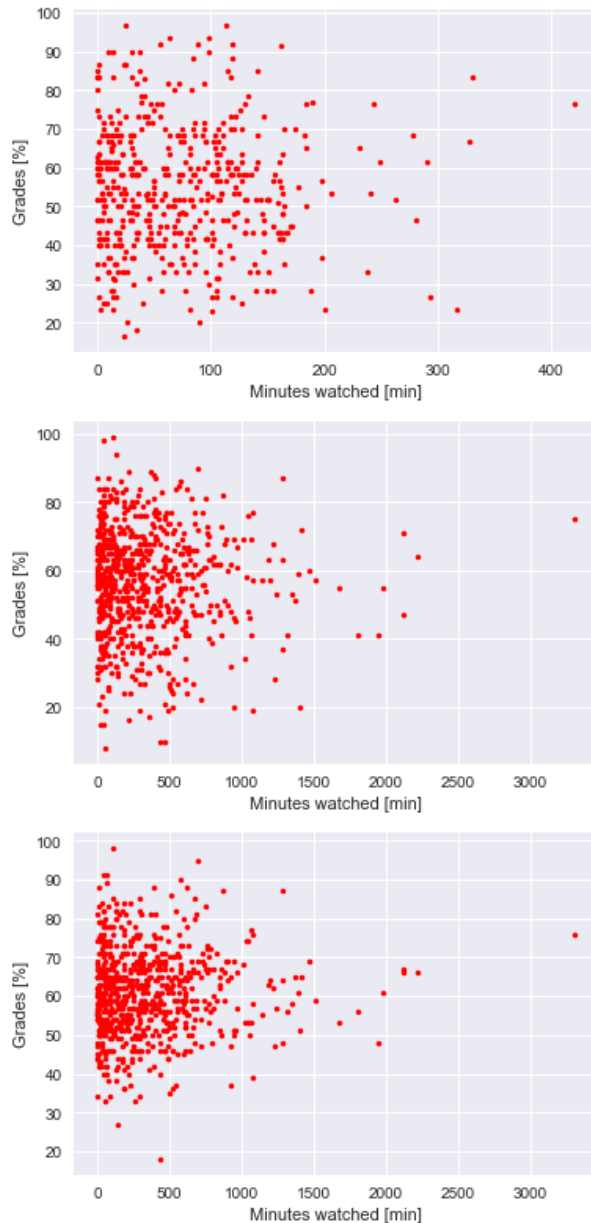


Figure 7.3: Grades of those watching videos for, top to bottom, October Test 2017, November Exam 2017 and Final-Year 2017.

November Exam 2017 again shows no clear trend in the data. Spearman's correlation coefficient was calculated as -0.039 showing no correlation between minutes watched and final

exam grades. A regression coefficient of -0.001 (p -value of 0.46) was calculated which is not significant. The means of the no-videos and videos groups were 56.5 % and 57.0 % respectively and the t-test calculated a p -value of 0.13 which suggests no significant difference in means.

Spearman's correlation for Final-Year grades was calculated as 0.019 which is again, not significant. The videos-group obtained a final year average of 61.4% and the no-videos group obtained 60.5%. There were 271 students that didn't use the videos and 594 that did. T-test results showed a p -value of 0.20 which again suggests that there is no significant effect of the videos on the final year marks. The regression coefficient was calculated as 0.001 ($p = 0.29$) which further suggests that the videos have no effect on the final year mark.

The following points are noted from the results of Section 7.4.1:

- No association between the amount of video watching and performance was found. Amount of video watching was not a predictor of performance as summarized in Table 7.1.
- When comparing videos and no-videos groups, the videos groups had higher averages than the no-video groups in all assessments (see Table 7.1 and Figure 7.4), indicated by the positive values in the *Difference* column which shows the videos-group grades minus the no-videos grades.
- However, as seen in the t-test column, no differences were significant except for October Test 2017.

Table 7.1: Comparison of means between video and no-video groups for 2017. The T-test column shows the difference between the videos-group grades and the no-videos group.

Assessment	Spearman's	β	Difference [%]	T-test
Sep. 2017	0.051	0.003 ($p=0.33$)	+ 1.2	($p = 0.35$)
Oct. 2017	-0.01	0.000 ($p=0.99$)	+ 2.1	($p = 0.016$)
Nov. 2017	-0.039	-0.001 ($p=0.46$)	+ 0.5	($p = 0.13$)
Final year	0.019	0.001 ($p=0.29$)	+ 0.9	($p = 0.20$)

This section showed how over multiple assessments, the number of videos watched showed no correlation with the grades. Did the videos really have no effect on the grades or are the results being looked at it in the wrong way? The author suggests that it is due to the different starting points that the students had before watching the videos. For example, consider a student who began with a high grade, say 80%, watched a very small number of videos and in the next assessment obtained a grade of 70%. That is, their grade decreased. Consider

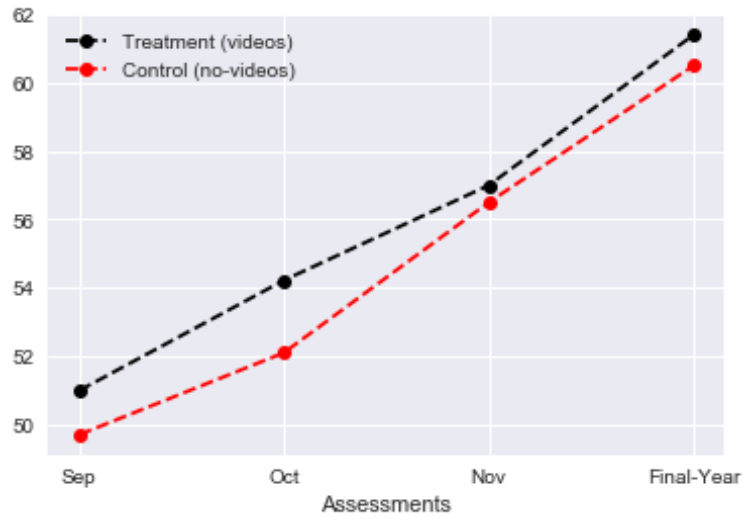


Figure 7.4: Average grades for control and treatment groups for 2nd semester 2017.

another student who began with a low grade, say 40%, watched a large number of videos, and then scored say 60% in the next assessment. If only the final grades are considered, it looks as if the second student that watched a large number of videos, was not benefitted at all. However, the first student who hardly watched any videos, performed better. Of course, there could be other factors in play as to why these students' grades changed. But a major flaw can be seen in simply comparing the number of videos watched with a single grade from an assessment. There is the need to look at the *change* in grades, not merely the final grades. The next section looks at difference-in-differences.

7.4.2 Difference-in-Differences

7.4.2.1 DID Values for Entire Cohort

This section therefore computes DID values to estimate treatment effects. First, the DID value for the entire cohort was computed. Table 7.2 presents the grades for the videos and no-videos groups at Mid-Year 2017 and Final-Year 2017. It can be seen that the videos group had an average change in grade of 3.25% and the no-videos group had an average change of 2.47%. Applying Equation 7.1, a DID value of $3.25 - (2.47) = 0.78\%$ was calculated, which suggests that the entire videos cohort obtained an average grade of 0.78% higher than what they would have obtained, had they not received the treatment. However, this was not significant $p = 0.48$ at $\alpha = 5\%$. It was concluded that there was no *overall* effect on academic performance.

Table 7.2: Mid-Year 2017 and Final Year 2017 grades for videos and no-videos groups.

	Videos group (n = 731 (76%))		No-videos group (n = 231 (24%))	
	Average [%]	Std. Dev. [%]	Average [%]	Std. Dev. [%]
Mid-Year	58.1	10.1	58.0	10.7
Final	61.4	10.7	60.5	11.0
Change in grades	3.25	-	2.47	-

Although not presented here, DID values were similarly computed for the time span Mid-Year 2017 to November Exam 2017. A DID value of 0.32% ($p=0.82$) was calculated which also is not-significant.

Therefore no global treatment effects were measured using DID;

7.4.2.2 DID Values Based on Initial Grades

The next aim was to compute DID values for various *sub-groups* based on their initial Mid-Year 2017 grades. It was proposed that the initial grades of the students at Mid-Year would affect (1) how they perform in the second-semester and (2) how they use the videos. Initially, those that failed ($<50\%$) and passed ($>50\%$) at Mid-Year were analyzed, independent of watching videos. These initial Mid-Year passing and failing groups were further broken up into treatment and control and DID values were computed. We therefore controlled for Mid-Year passing or failing grades and estimated the DID values. Table 7.3 presents the results for the Mid-Year to Final-Year time span and shows that those that failed at Mid-Year had a change of +4.9 % from Mid-Year to Final-Year, and those that passed at Mid-Year had a smaller change of +2.6 %. This suggests that, independent of videos, those that were initially failing had greater room for improvement and a larger incentive to improve their grades than those who were initially passing. Failing at Mid-Year looked to be a strong incentive for students to improve during the second-semester. Furthermore, the change for those failing at Mid-Year *and* watching videos was +5.7% and those failing and *not* watching videos, the change was +3.1%, giving a DID estimate of $5.7\% - 3.1\% = 2.6\%$ ($p = 0.05$). That is, for those failing at Mid-Year, the videos had an effect of improving grades by 2.6% from what they "would have obtained" without the videos. The DID estimate is significant at $\alpha = 5\%$. Again, looking at Table 7.3, the DID estimate for those passing at Mid-Year 2017 was 0.4 ($p = 0.72$) suggesting that for those passing at Mid-Year, there is no treatment effect.

Table 7.4 presents the results for the time span from Mid-Year 2017 to November Exam 2017. Although not significant at $\alpha = 5\%$, again the failing group shows a significant DID estimate of 4.3% at $\alpha = 10\%$, whereas the passing group has a non-significant DID estimate.

Table 7.3: Changes and DID estimates based on failing or passing at Mid-Year for the time span from Mid-Year 2017 to Final Year 2017

Mid-year [%]	Change [%]	Change (vid.) [%]	Change (no-vid.)	DID [%] (<i>p</i> -value)
Failing	4.9	5.7	3.1	2.6 (0.05)
Passing	2.6	2.7	2.3	0.4 (0.72)

Table 7.4: Changes and DID estimates based on failing or passing at mid-year for the time span from Mid-Year 2017 to November Exam 2017.

Mid-year [%]	Change [%]	Change (vid.) [%]	Change (no-vid.)	DID [%] (<i>p</i> -value)
Failing	3.1	4.3	-0.0	4.3 (0.07)
Passing	-2.3	-2.4	-1.9	-0.5 (0.7)

Therefore when controlling for passing or failing Mid-Year grades, those initially failing tend to receive significant treatment effects of the videos on their grades.

In addition to controlling for passing and failing Mid-Year grades, the students were further broken up into ranges of initial Mid-Year grades, and within each range, results were extracted for the treatment and control groups. The aim was to see how different initial ranges of Mid-Year grades affect DID values. Tables 7.5 and 7.6 show the DID values for these sub-groups for both times spans, respectively. For the group that obtained between 40% and 50% at Mid-Year, significant DID values were computed. The results suggest that those that are just under the passing grade are most impacted by the videos and that just missing the 50% mark may significantly motivate students to use the videos intervention. Further study needs to be carried out to determine why this specific subgroup is impacted whereas a close sub-group such as the 50%-60% group, is not. This will require interviewing students falling into these sub-groups to obtain further insight.

Table 7.5: DID estimates based on initial Mid-Year grades for the time span from Mid-Year 2017 to Final-Year 2017.

Mid-year grade [%]	Change [%](vid.)	Change [%] (no-videos)	DID [%] (<i>p</i> -value)
>80	3.64	-3.34	6.98 (0.18)
(70-80]	3.04	3.59	-0.55 (0.68)
(60-70]	2.27	1.34	0.93 (0.34)
(50-60]	2.84	2.77	0.07 (0.90)
(40-50]	5.49	2.43	3.10 (0.00)
(30-40]	4.90	7.30	-2.40 (0.41)

Table 7.6: DID estimates based on initial Mid-Year grades for the time span from Mid-Year 2017 to November Exam 2017.

Mid-year grade [%]	Change [%](videos)	Change [%] (no-videos)	DID [%] (<i>p</i> -value)
>80	-1.75	-15.85	14.1 (0.07)
(70-80]	-2.47	0.45	-2.92 (0.22)
(60-70]	-3.00	-4.56	1.56 (0.44)
(50-60]	-2.05	-0.20	-1.85 (0.27)
(40-50]	4.27	-1.40	5.67 (0.02)
(30-40]	0.74	8.73	-7.99 (0.13)

7.5 Discussion

This study has made two main contributions to the literature on online videos. The first is that this is the first known study to measure detailed video watch time of each student and compare it with test and exam results. The only literature found that has performed a similar study is Leadbeater et al. (2013) [154]. The literature has been extended in this study by measuring detailed watch time for each individual student and analyzing how each student performed in relation to the number of minutes of video watched, using regression and correlation. This was carried out for four assessment which allows for a degree of generalization. Ideally this would be carried out over multiple years to study better generalizability effects. What was found is that there was no association between grades obtained and minutes watched for each assessment. These results may seem discouraging but similar results are found by Leadbeater et al. (2013) [154]. Although no association was found, this does not necessarily mean that there is no treatment effect, since there could be bias in the results. Although there are many and varied reasons for the mixed results, this study proposed that a reason overlooked by the literature was the use of the wrong measurement techniques to measure treatment effects in observational studies. The author believes that the wrong question was being asked. When it comes to observational studies that have inherent bias, the question should not be: do the videos cause significantly better performance in the treatment group compared with the control? But, do the videos cause a significant *positive change (or improvement)* in grades for the treatment group compared with the control? In observational studies the aim should be to look at changes in grades, not the absolute grade.

Based on this argument, the second main contribution was to measure treatment effects of online videos using DID, that looks at changes in grades. This is the first known study to use this technique in the online video literature. Similar literature studying online learning has used randomized controlled studies to measure effects [153, 163, 165]. Figlio et al. (2013)

[165] stated in their study that due to selection problems it is not possible to simply look at differences between those watching videos and not and therefore conducted a randomized study. However, the author believes that randomizing the students into a control and treatment group is not the appropriate step to take since this coerces some students to not receive an intervention that might help them. It is seen as unethical. Therefore there is a need to rather investigate the effects of online videos but from the observational study perspective.

Although the previous method of measurements found no association, the DID values did find treatment effects. Although no significant global treatment effects were found for the overall cohort, significant effects were found for sub-groups. This is a similar finding to that of Chapter 6 where no treatment effects were measured for the overall group but large variations in individual treatment effects were found. A main finding was that the initial grade is a predictor of the treatment effect. Or conversely, the treatment effect is dependent on the initial grade and that for those initially failing, the intervention is found to be effective. The literature has linked initial factors to performance, but *no literature has been found that links initial factors to the treatment effect*. This finding suggests those that are failing (and specifically those in the 40%-50% range) may use the videos in the most productive way. A reason for the videos being effective for this group may lie more in the student than in the video itself. For example, these students may use the videos in a targeted or strategic way [154], meaning they use it to focus on topics they are struggling with. Further study is required to determine, through interviewing these students, whether this was the case. Another reason for video effectiveness in this group may be due to intrinsic motivation [155]. Those that perform poorly initially, may have a strong incentive to utilize the available resources as effectively as possible. Therefore although the intervention is seen as a valuable resource, the motivation of the student is seen to be a major factor in the videos' effectiveness. This study has perhaps found a proxy for motivation, i.e. the initial grade that a student obtains early in a semester. It suggests that for an intervention to be effective, it needs to be combined with other factors within the student. This was similarly found with Boaler et al. (2018) [163] who showed that mindset beliefs impact academic performance. Whereas Boaler et al. (2018) showed that the motivation impacts absolute performance, this study suggests that motivation impacts not the absolute grade, but the treatment effect.

Although future interventions will always be open to any students, this finding suggests that focus needs to be placed on the group of students that are just under the failing grade, to advise them to utilize the videos as much as possible. The hope is this practice will improve the pass rate in the course.

7.6 Conclusion and Further Work

This chapter presented work on addressing the third objective of this thesis which was to measure treatment effects in interventions for students. The intervention in this chapter was supplementary online videos that the students could access at any time. Students self-selected to watch the videos, making this an observational study.

Measuring video watch time and comparing test results with the watch time showed that there was no association of grades with watch time. This led the study to look at not absolute grades, but change in grades, using the DID statistical method. Although no significant global treatment effects were measured, significant sub-group effects were measured, showing that initial grades, specifically those that initially failed, were a predictor of treatment effects.

A major limitation of this study was that it was limited to a single year's cohort of students which restricts the generalizability of the results. Therefore the most important future work will be to generalize this study to more cohorts of students and in different courses, not only engineering mechanics.

Intrinsic factors within the student such as motivation and method of video usage are seen as major factors for how effective the videos intervention is. Therefore in follow-up studies, the students using the videos could be interviewed asking various questions pertaining to their motivational state and method of usage. This could be administered early and late in the semester. This opens up an important window into combining actual watch time, academic performance and motivational state of the students.

A further limitation here was that only the grades data was available for measuring treatment effects. Future studies would obtain a large range of features on the students and use this method in conjunction with the ML techniques presented in the previous chapters. It is also proposed as a future study, to compare how different the treatment effects are for face-to-face tutorials and videos.

Studying the quality of the videos also forms part of follow-up studies. This study focused simply on measuring how effective these specific videos are. A more thorough study of the videos themselves, in combination with students' views on the videos, could also open up a window into improving the performance. It is believed that the students themselves play an integral part in moving the conversation forward with regards to using videos as an online learning tool.

Finally, although global (group and subgroup) analyses have some value, the literature is increasingly showing the need to measure individual treatment effects [4, 9]. Therefore further research would focus on individual treatment effects that could assist the students in personalized learning.

Chapter 8 Conclusion

8.1 Introduction

Predicting and characterizing students at risk of failing is crucial because these At-Risk students could then be advised on various ways to switch from being At-Risk to not being At-Risk. The initial challenge therefore in this study was to find better ways of predicting and characterizing At-Risk students. Although predictions are indeed valuable, being able to also explain or interpret these predictions, i.e. characterize the students, is an important challenge to address. Knowing what kinds of students are At-Risk and why they are At-Risk will allow for identifying groups of students, or individual students, for which various advice or interventions could be provided. However, even though various interventions can be recommended, a further major challenge was to measure how effective these interventions were. It is not sufficient simply to recommend a student for an intervention without being able to measure whether the intervention actually helps the student or group of students. In summary, better ways were required of:

1. Predicting and characterizing At-Risk students.
2. Providing advice or interventions for At-Risk students to help them switch from being At-Risk to being not At-Risk.
3. Measuring the effectiveness of the interventions.

8.1.1 Summary

ML methods primarily, as well as statistical methods, were used to address these objectives. Based on thorough data preparation (Chapter 3) and model selection procedures (Chapter 4), this study introduced model-agnostic methods to the literature (Chapter 5) that were able to explain any ML model and thereby characterize groups or individual students. Furthermore,

counterfactual explanations were also introduced (Chapter 5) to provide potential ways that students could switch At-Risk status. Measuring treatment effects were addressed in Chapters 6 and 7.

8.1.2 Preliminary Work

Before any predictions, characterizations or treatment effects could be estimated, it was vital to carry out thorough data preparation. It is believed that without this first step, the predictions and characterizations in subsequent chapters would not be as reliable. In Chapter 3 various preparations were performed that included normalizing, encoding and dealing with correlated features. However, the most important part of this chapter was to perform a thorough analysis and imputation of missing data. It was found that the random forest method *missForest* performed the best when imputing missing data, outperforming *MICE* methods. Although it is difficult to measure based on this study, the results suggest that the dataset imputed by *missForest* resulted in subsequently better trained models that computed better characterizations of At-Risk students. This finding generates future work where the effects of missing data imputation on predictions and characterizations can be studied, potentially via simulation studies.

Thereafter, detailed and extensive ML model selection and testing was done to obtain the best performing models. This was carried out both in Chapters 4 and 6 where classification and regression models were trained. In general, the literature utilized models with default ML parameters. This study however, did not rely on default model parameters but performed an extensive model selection procedure by tuning the hyperparameters using CV. Five types of models were analyzed, namely lasso, random forest, gradient boosting, adaboost and neural networks. Random forest, gradient boosting and neural networks were the best performers. This compares well with the relevant literature. This preliminary step is considered highly important for addressing the objectives of this study. If optimal models are not first obtained, then the subsequent results from the models are not considered to be as trustworthy.

8.2 Prediction, Characterization and Counterfactual Explanations

This section discusses the implications of addressing the first two objectives. To characterize At-Risk students, this study introduced model-agnostic explanations in Chapter 5, which can be applied to any black-box model to extract the features that best predict the outcome. These explanations were either global, which refer to characterizing At-Risk students overall,

i.e. the entire group, or they were local, which refers to obtaining individualized or personalized explanations for individual students. For the global explanations, feature importances (FI), permutation importances (PI), lasso β coefficients and partial-dependence plots (PD-plots) were combined to identify the global features or characteristics. This was the first known study to combine these methods together and apply it to the field of student performance. The value in this method was to first obtain the top predictors using FI and PI, and then use β coefficients and PD-plots to provide direction, and compute non-linear marginal effects that identify smaller regions where students have higher At-Risk probabilities than the expected value. These methods promise much because it doesn't restrict the characterizing of At-Risk students to only using interpretable methods such as regression and tree-based methods. Complex high performing models can be trained and still have model-agnostic methods applied to them to identify which features most impact being At-Risk.

Although the global explanations were helpful, in that they can provide general areas in which to look to identify At-Risk students, the author believes that the real value is in combining the global with the local explanations, also presented in Chapter 5. Catering to individual students is crucial since students are different and have different needs. Locally Interpretable Model-Agnostic Explanations (LIME) and Shapley Additive Explanations (SHAP), developed by Ribeiro et al. (2016) [7] and Lundberg et al. (2017) [8], respectively, were introduced to the field of predicting student performance. It was shown how individual students that are predicted to be At-Risk can be presented with explanations on why they are At-Risk. The implication is that they could then identify areas that they could improve on so as to eventually pass. It is not only valuable to predict which students are At-Risk, but also to explain to these individual students potential ways to change to not being At-Risk. This led to the counterfactual explanations, also first introduced to this field in this study. Counterfactual explanations are "what if" scenarios generated by perturbing the features that were explained using the local models. This method promises much in terms of first, identifying At-Risk students, and then second, providing advice and solutions for these students to improve their grades and pass.

8.2.1 Limitations of ML for Prediction, Characterization and Counterfactual Explanations

Some important limitations were discovered in these applications. Although global and local features were identified that could allow for identifying At-Risk groups or individuals, much care is required in dealing with At-Risk students. Applying these results needs to be done using the help of experienced educators that would have the experience of dealing with students. Although these methods promise much, a couple of warnings are in order. The first

is that much care should be exercised to not confuse correlation with causation. For example, At-Risk features could include race or gender, and the analyst could erroneously believe that these features are causes for being At-Risk. However, they simply show correlation and more study would be required to identify the true causal factors for being At-Risk. The second warning, following on from the first one, is that these results could be used in an insensitive way by showing a student that they are going to fail. Again, experienced educators are needed to apply these results. The students' well being is of paramount importance and although these results promise much in terms of helping them, the way they are applied can help or be detrimental to the student. If they are told that they are going to fail, this might have one of two effects: it might motivate them or they might lose heart. Wisdom needs to be exercised in applying these results in the best way, caring for the welfare of the student.

Furthermore, although the aim is to identify features that could be modified, this is simply not possible for some features. Therefore counterfactual explanations are quite limited in their applications in terms of At-Risk students. A counterfactual explanation could suggest improving a test grade by a certain percentage in order to switch to not being At-Risk, but it is not possible in a time-based study to modify these types of features. The predictions and counterfactual explanations are based on certain test grades and it is not possible, in most cases, for the student to go back and re-write *that* test. Other features that cannot be changed include ethnicity, gender, or age, for example. Much study is still required with regards to the features that can be used when applying counterfactual explanations. There are certain static features that cannot be changed, as mentioned above, and dynamic features that could be changed. Dynamic features include attending interventions such extra tutorials, or food-programs and support groups. At the author's University, food-programs are in place for financially disadvantaged students to obtain one meal per day. Further value in these methods is that early on, At-Risk students could be identified and personal care could be administered to them. These ML methods might even allow for identifying personal non-academic issues in students' lives that could be addressed so that the student could be supported and eventually succeed.

8.3 Treatment Effects

The previous section discussed the implications of addressing the first two objectives. This section addresses the third objective.

8.3.1 MLTE Method

It is of great importance that interventions be measured in order to ascertain their effectiveness. This study used ML and statistical techniques to measure how effective single interventions were, namely face-to-face tutorials and online videos, presented in Chapters 6 and 7, respectively.

Chapter 6 extended the work by Beemer et al. (2017) [4] and Beemer et al. (2018) [9]. This work essentially used ML to predict counterfactuals, which are the basis for measuring treatment, or causal effects. This MLTE technique estimates counterfactuals by predicting what a treatment (or control group) would have had, had they received the control (or treatment). This study extended the literature by (1) more accurately describing the different treatment effects, (2) by validating this method against the traditional PSM method and (3) by studying global, subgroups and individual treatment effects. Again, this study showed that much value lies in combining both global and local effects. It is important to note that aim of this method is not to predict At-Risk students, but treatment effects.

Global treatment effects were computed and compared well with the traditional PSM methods. Validating the global values with PSM added some validity to this approach but it is felt that much more study is required to understand the scenarios where the MLTE method is and is not valid. Furthermore, global treatment effects were found to be small, but large variation of individual treatment effects was found. This suggests that there is more value in focusing on subgroups and individual students.

With regard to the subgroups, the MLTE method showed much potential in predicting *who would benefit from the intervention*. Using this method to predict counterfactuals and treatment effects allowed for finding subgroups that would benefit from the intervention. This was then used in combination with PI to explain or characterize these subgroups. For example, students were identified for whom the treatment effects were positive and using PI, the characteristics of these students were found. This would then allow for future subgroups to be identified that are predicted to benefit from the intervention. Also, this method predicts who would not benefit from the intervention. Although it is considered unethical to forbid any students from attending the intervention, the results suggest certain groups that could be focused on that would potentially receive the most benefit.

Zooming further into individual students, this study further developed a method to predict as early as possible (1) what the final grade of a student would be, (2) what the characteristics of the student are (using SHAP), and (3) what that specific student's treatment effect would be. The importance of this method is that an individual student could not only be predicted (identified) to be At-Risk, but they could be individually characterized, and most importantly,

predictions could be made for them to measure how much the intervention would help them. Although ML has been used much to predict student performance, this is one of the first studies to combine predicting performance with treatment effects. This would guide the educator as to how much the intervention would actually assist that student. Of course, all students should always be allowed to attend these interventions. But this method promises much in terms of focusing on students that have a much higher probability of receiving benefit from the intervention.

8.3.1.1 Limitations of the ML method on Treatment Effects

Although this method has shown promising applications, a few limitations exist. The first is that much work is still required in terms of validation. Although this study validated the technique using PSM, in reality there is no “ground truth” defining exactly what the real treatment effects (or counterfactuals) are. Second, the MLTE method had potential bias in the predictions due to unbalanced treatment and control groups. In RCTs or PSM methods, the aim is to balance the two groups so that a counterfactual is created. Thereafter differences between the two groups would then only be attributed to the treatment itself. However, in this case, due to unbalanced features, a model trained on a treatment group, would have the inherent characteristics of that group. Similarly, a model trained on the control would be characteristic of the control group. Therefore it is expected that applying control group features to the treatment model would most likely introduce a treatment effect that includes a bias. Bias in the treatment effect for the control group is similarly expected. This can actually be seen when comparing the global treatment effects, ATT and ATC, in that they were different. This study did not provide a qualitative or quantitative way of estimating the degree of bias. It is further expected that this bias might have a knock-on effect on the individual treatment effects (ITT, ITC). It is therefore a crucial follow-up study to focus on the bias generated in this kind of method. A third limitation was that this method was not tested on future data but on historic data. This allows much room for follow up studies, discussed below.

8.3.2 Difference-in-Differences

Chapter 7 presented the first known study to measure actual detailed video watch time and relate that to performance. It was shown over multiple assessments that there was no correlation between number of videos watched and performance. The DID technique was therefore used to measure treatment effects, since this was an observational study and only panel data was available.

It was again found, similar to Chapter 6, that global treatment effects were not significant, but that certain subgroups measured significant effects. An interesting finding from both these chapters on measuring treatment effects, was that initial grades were found to be a predictor of the treatment effect. That is, students with lower grades showed larger treatment effects. The results suggested that the group that had initial grades just lower than the passing grade would have significant treatment effects.

8.3.2.1 Limitations of the DID Method

The DID method is useful when there is a limitation in student information, and only grades data in panel form are available. DID is therefore dependent on the control group being a counterfactual to the treatment group, which is estimated via the parallel trends assumption.

Furthermore, this study was limited to a single year's cohort of students and would benefit much from carrying out multiple year's studies to obtain trends.

Finally, a limitation, similar to the previous studies, is that this method was not validated against any other techniques.

8.4 Overall Implication of the Study

This study focused on using ML and statistical methods to cater to students that are At-Risk of failing a university course. The value of this study for the field of At-Risk students is to not only be able to predict At-Risk students, but also to identify global and individual features of these students and predict whether interventions would be of use to At-Risk students. The implication is that students could be identified early on in a semester as being At-Risk of failing and then specific features could be identified as potential reasons for the At-Risk status. These specific features could then be addressed in cooperation with experienced educators and interventions could be provided. The methods in this study could then be applied to measuring if these interventions would actually provide a significant boost in final grades to help pass the course. The implication is that, although any student should be allowed to attend interventions, specific groups and individual students could be targeted and advised, using much sensitivity, to attend these interventions, because there is a high probability of it increasing their grades sufficiently to pass.

For there to be successful application of ML in the field of At-Risk students, adequate and insightful datasets are paramount. It is suggested that not only standard data be obtained such as seen in this study, but further psychological data such as student motivation also be

measured. Furthermore, it is advised to carry out detailed tuning of the various ML models to provide greater accuracy in the later stages. It is vital that these preliminary stages be carried out, otherwise there might not be sufficient confidence in the At-Risk predictions. If data is limited, the DID method is advised, although it would be of greater value to obtain rich datasets and apply ML methods.

When characterizing the students, combinations of methods are suggested in order to obtain better insight of the features that identify At-Risk students. These methods include the various ML models used together with the model-agnostic methods. When dealing with global characteristics, it is suggested to combine a method such as PI with a method that provides directionality, such as PD-plots. Furthermore, although global, sub-group and individual methods were presented, the results imply that more focus should be placed on the sub-group and individual applications. This is because in both Chapters 6 and 7, it was found that the sub-group and individual predictions and characterizations showed much more variation and significance. Although it might be more labor intensive, the future application of using model-agnostic methods is on catering to individual students. For individual characterizations, it is suggested to use the SHAP method since it is shown to be superior to LIME.

For both studies measuring treatment effects, initial status seemed to impact the treatment effect. This may imply that motivation combined with the treatment produce a positive result. It was suggested above that motivational information might be a valuable feature in future datasets.

In this study, the reader should be reminded that correlation should not be confused with causation and care should be exercised when using the results. The only time that causation is inferred is when measuring the treatment effects using the MLTE and DID methods.

8.5 Generalizability and Future Work

The methods in this study were applied to the field of student performance. However, there is much applicability both in this field, as well as any other fields where interventions are provided in an observational study.

A few key follow-up studies are identified and discussed next. The first is to test the generalizability of these methods on future student cohorts, proposed as follows: The models trained in this study could be used in a subsequent year at the first round of tests in March, and using the same input features, each student's final grade as well as their treatment effects could be predicted. The students could then be monitored throughout the semester and final

grades could be compared with predicted grades and final treatment effects compared with predicted treatment effects.

Second, whereas the first point discussed using the trained models on future data, new models could be trained on future data and generalizability could be measured in terms of whether similar features are found to characterize At-Risk students and if similar treatment effects are measured.

A third and obvious future study would be to perform a similar study using more features and a larger range of ML models. As discussed in Chapter 3 it may add significant value to include motivational features in the models, for example.

A fourth and vitally important future study would be to study the effects of the bias in the method for estimating treatment effects. This could be carried out using simulation studies where the “ground truth” treatment effect would be known and global and individual treatment effects could be computed and compared with the “ground truth”. There is considerable room here for future work.

Fifth, it is proposed to study how the ML models may change over time; a phenomenon known as concept drift [110]. This is important because using models trained in a single year’s cohort may not be valid in future years.

A final and exciting potential future study would be to compare treatment effects of videos and face-to-face interventions using these methods. How do they compare? Is one more effectual than the other? What about when they are used together?

8.6 Final Thoughts

The author believes that the use of ML in terms of helping students achieve success is the way forward. However, there is always the need, as seen in this study, for the human touch and wisdom to guide these ML application. The ML models are powerful for providing insight, but ultimately the student is in need of human care.

The author is reminded of a quote by Derek Muller [171], the popular YouTuber, who said: “As transformative as each new technology seems to be, ...what really matters is what happens inside the learner’s head. And making a learner think seems best achieved in a social environment with other learners and a caring teacher.” This quote is modified here to say:

As transformative as ML seems to be, what really matters is how the individual student is affected and to maximize the impact of ML on individual students requires a social environment with caring educators.

Bibliography

- [1] S. Kotsiantis, K. Patriarcheas and M. Xenos. ‘A combinational incremental ensemble of classifiers as a technique for predicting students’ performance in distance education’. In: *Knowledge-Based Systems* 23 (2010), pp. 529–535.
- [2] C. Anuradha and T. Velmurugan. ‘A comparative analysis on the evaluation of classification algorithms in the prediction of students performance’. In: *Indian Journal of Science and Technology* 8 (2015), pp. 1–12.
- [3] H. Al-Shehri et al. ‘Student performance prediction using support vector machine and k-nearest neighbor’. In: *2017 IEEE 30th Canadian Conference on Electrical and Computer Engineering (CCECE)* (2017), pp. 17–20.
- [4] J. Beemer et al. ‘Ensemble learning for estimating individualized treatment effects in student success studies’. In: *International Journal of Artificial Intelligence in Education* (2017).
- [5] S. Wachter, B. Mittelstadt and C. Russel. ‘Counterfactual explanations without opening the black box: Automated decisions and the GDPR’. In: *Harvard Journal of Law and Technology* 31.2 (2018).
- [6] C. Molnar. *Interpretable machine learning: A guide for making black box models explainable*. Leanpub, 2019.
- [7] M. T. Ribeiro, S. Singh and C. Guestrin. ‘Model-agnostic interpretability of machine learning’. In: *ArXiv preprint arXiv 1606.05386* (2016).
- [8] S. M. Lundberg and S.-I. Lee. ‘A unified approach to interpreting model predictions’. In: *31st Conference on Neural Information Processing Systems (NIPS 2017)*, (2017).
- [9] J. Beemer et al. ‘Assessing instructional modalities: Individualized treatment effects for personalized learning’. In: *Journal of Statistics Education* 26 (2018).
- [10] D. Faraoni and S. T. Schaefer. ‘Randomized controlled trials vs. observational studies: Why not just live together?’ In: *BMC Anesthesiology* 16.102 (2016).

- [11] Y. Ye and L. A. Kaskutas. ‘Using propensity scores to adjust for selection bias when assessing the effectiveness of Alcoholics Anonymous in observational studies’. In: *Drug and Alcohol Dependence* 104.1-2 (2009), pp. 56–64.
- [12] G. Hong and B. Yu. ‘Effects of kindergarten retention on children’s social-emotional development: An application of propensity score method to multivariate, multilevel data’. In: *Developmental Psychology* 44.2 (2008), pp. 407–421.
- [13] A. E. Wyse, V. Keesler and B. Schneider. ‘Assessing the effects of small school size on mathematics achievement: A propensity score-matching approach’. In: *Teachers College Record* 110 (2008).
- [14] S. Athey. ‘Machine learning and causal inference for policy evaluation’. In: *KDD ’15 Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (2015), pp. 5–6.
- [15] T. Hastie, R. Tibshirani and J. Friedman. *The elements of statistical learning. Data mining, inference, and prediction. Second Edition*. Springer, 2009.
- [16] S. Marsland. *Machine learning: an algorithmic perspective*. CRC Press, 2015.
- [17] T. M. Mitchell. *Machine learning*. McGraw-Hill Science/Engineering/Math, 1997.
- [18] R. Klein. ‘The Wits intelligent teaching system (WITS): A smart lecture theatre to assess audience engagement’. PhD thesis. University of the Witwatersrand, 2017.
- [19] I. Goodfellow, Y. Bengio and A. Courville. *Deep learning*. MIT Press, 2016.
- [20] G. James et al. *An introduction to statistical learning with Applications in R*. Vol. 7. Springer, 2013.
- [21] P. Baumann, D. Hochbaum and Y. Yang. ‘A comparative study of the leading machine learning techniques and two new optimization algorithms’. In: *European Journal of Operational Research* (2018).
- [22] R. Caruana and A. Niculescu-Mizil. ‘An empirical comparison of supervised learning algorithms using different performance metrics’. In: *In Proc. 23rd Intl. Conf. Machine learning (ICML’06)* (2006), pp. 161–168.
- [23] R. Caruana, N. Karampatziakis and A. Yessenalina. ‘An empirical evaluation of supervised learning in high dimensions’. In: *Proceedings of the 25th International Conference on Machine Learning (2008)* (2008), pp. 96–103.
- [24] M. Friedl and C. Brodley. ‘Decision tree classification of land cover from remotely sensed data’. In: *Remote Sensing of Environment* 61.3 (1997), pp. 399–409.
- [25] M. Krzywinski and N. Altman. ‘Classification and regression trees’. In: *Nat Methods* 14 (2017), pp. 757–758.
- [26] G. Louppe. ‘Understanding random forests: From theory to practice’. In: *PhD thesis, U. of Liege* (2014).

- [27] J. Ali et al. ‘Random forests and decision trees’. In: *IJCSI International Journal of Computer Science Issues* 9.5 (2012), pp. 272–278.
- [28] L. Breiman. ‘Random forests’. In: *Machine Learning* 45.1 (2001), pp. 5–32.
- [29] P. Probst, M. N. Wright and A.-L. Boulesteix. ‘Hyperparameters and tuning strategies for random forest’. In: *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery* 9.3 (2019), e1301.
- [30] F. Pedregosa et al. ‘Scikit-learn: Machine learning in Python’. In: *Journal of Machine Learning Research* 12 (2011), pp. 2825–2830.
- [31] T. M. Oshiro, P. S. Perez and J. A. Baranauskas. ‘How many trees in a random forest’. In: *MLDM’12 Proceedings of the 8th International Conference on Machine Learning and Data Mining in Pattern Recognition* 45 (2012).
- [32] P. Geurts, D. Ernst and L. Wehenkel. ‘Extremely randomized trees’. In: *Mach Learn* 63 (2006), pp. 3–42.
- [33] L. Breiman. ‘Manual on setting up, using and understanding random forests V3.1’. In: (2002).
- [34] G. Martínez-Muñoz and A. Suárez. ‘Out-of-bag estimation of the optimal sample size in bagging’. In: *Pattern Recognition* 43 (2010), pp. 143–152.
- [35] A. Mayr et al. ‘The evolution of boosting algorithms - From machine learning to statistical modelling’. In: *Methods Inf Med* 53.6 (2014).
- [36] Y. Freund and R. E. Schapire. ‘A decision-theoretic generalization of on-line learning and an application to boosting’. In: *Journal of Computer and System Sciences* 55.1 (1997).
- [37] J. H. Friedman. ‘Greedy function approximation: A gradient boosting machine’. In: *The Annals of Statistics* 29.5 (2001), pp. 1189–1232.
- [38] J. H. Friedman. ‘Stochastic gradient boosting’. In: *Computational Statistics and Data Analysis* 38 (2002).
- [39] B. Efron and T. Hastie. *Computer age statistical inference. Algorithms, Evidence, and Data Science*. Cambridge University Press, 2016.
- [40] Y. Lecun, Y. Bengio and G. Hinton. ‘Deep learning’. In: *Nature* 521 (2015), pp. 436–444.
- [41] Skymind. *A beginner’s guide to neural networks and deep learning*. URL: <https://skymind.ai/wiki/neural-network>.
- [42] F. Rosenblatt. ‘The perceptron: A probabilistic model for information storage and organization in the brain’. In: *Psychological Review* (1958), pp. 386–408.
- [43] W. S. McCulloch and W. H. Pitts. ‘A logical calculus of the ideas immanent in nervous activity’. In: *Bulletin of Mathematical Biophysics* 5 (1943), pp. 115–133.

- [44] J. Schmidhuber. ‘Deep learning in neural networks: An overview’. In: *Neural Networks* 61 (2015), pp. 85–117.
- [45] D. Rumelhart, G. Hinton and R. Williams. ‘Learning representations by back-propagating errors’. In: *Nature* 323 (1986), pp. 533–536.
- [46] M. Siddique and M. Tokhi. ‘Training neural networks: Backpropagation vs genetic algorithms’. In: *IJCNN’01. International Joint Conference on Neural Networks. Proceedings (Cat. No.01CH37222)* 4 (2001), pp. 2673–2678.
- [47] T. K. Gupta and K. Raza. ‘Optimizing deep neural network architecture: A tabu search based approach’. In: *Arxiv: 1808.05979* (2018).
- [48] T. Kavzoglu. ‘Determining optimum structure for artificial neural network’. In: *Proceedings of the 25th Annual Technical Conference and Exhibition of the Remote Sensing Society, Cardiff, UK* (1999), pp. 675–682.
- [49] K. G. Sheela and S. Deepa. ‘Review on methods to fix number of hidden neurons in neural networks’. In: *Mathematical Problems in Engineering* 2013 (2013), pp. 419–425.
- [50] S. Xu and L. Chen. ‘A novel approach for determining the optimal number of hidden layer neurons for FFN’s and its application in data mining’. In: *5th International Conference on Information Technology and Applications (ICITA 2008)* (2008).
- [51] L. Wang et al. ‘Optimal size of a feedforward neural network: How much does it matter?’ In: *Joint International Conference on Autonomic and Autonomous Systems and International Conference on Networking and Services* (2005).
- [52] Y. Bengio. ‘Practical recommendations for gradient-based training of deep architectures’. In: *arXiv: 1206.5533v2* (2012).
- [53] MLXTEND. *Confusion Matrix*. 2019. URL: http://rasbt.github.io/mlxtend/user_guide/evaluate/confusion_matrix/ (visited on 2019).
- [54] A. Tharwat. ‘Classification assessment methods’. In: *Applied Computing and Informatics* (2018).
- [55] J. Cohen. ‘A coefficient of agreement for nominal scales’. In: *Educational and Psychological Measurement* 20 (1960), pp. 37–46.
- [56] M. L. McHugh. ‘Interrater reliability: the kappa statistic’. In: *Biochem Med (Zagreb)* 22.3 (2012), pp. 276–282.
- [57] J. R. Landis and G. G. Koch. ‘The measurement of observer agreement for categorical data’. In: *Biometrics* 33 (1977), pp. 159–174.
- [58] I. H. Witten et al. *Data mining. Practical machine learning tools and techniques Fourth edition*. Morgan Kaufmann, 2016.
- [59] G. Zhang et al. ‘Student graduation: A longitudinal and cross-institutional study’. In: *Journal of Engineering Education* (2004), pp. 313–320.

- [60] Y. Qin et al. ‘Semi-parametric optimization for missing data imputation’. In: *Applied Intelligence* 27 (2007).
- [61] Y. V. Karpievitch, A. R. Dabney and R. D. Smith. ‘Normalization and missing value imputation for label-free LC-MS analysis’. In: *BMC Bioinformatics* 13.16 (2012).
- [62] S. Kotsiantis, D. Kanellopoulos and P. Pintelas. ‘Data preprocessing for supervised learning’. In: *International Journal of Computer Science* 1.1 (2006).
- [63] T. C. Urdan. *Statistics in plain English*. 3rd ed. Routledge, 2010.
- [64] RStudio Team. *RStudio: Integrated Development Environment for R*. RStudio, Inc. Boston, MA, 2016.
- [65] M. Kuhn et al. *caret: Classification and Regression Training*. R package version 6.0-82. 2019.
- [66] H. Kang. ‘The prevention and handling of the missing data’. In: *Korean Journal of Anesthesiology* 64.5 (2013), pp. 402–406.
- [67] T. Frisell. ‘SP0187 Why missing data is a problem, and what you shouldn’t do to solve it’. In: *Annals of the Rheumatic Diseases* 75 (2016).
- [68] J. W. Graham. ‘Missing data analysis: Making it work in the real world’. In: *Annual Review of Psychology* 60 (2009), pp. 549–576.
- [69] E. Aguiar et al. ‘Engagement vs performance: Using electronic portfolios to predict first semester engineering student retention’. In: *LAK 2014 : fourth International Conference on Learning Analytics and Knowledge: 24-28, March 2014, Indianapolis, IN, USA* (2014), pp. 103–112.
- [70] C. Beaulac and J. S. Rosenthal. ‘Predicting university students’ academic success and major using random forests’. In: *Research in Higher Education* (2019).
- [71] C. Crespo-Turrado et al. ‘Student performance prediction applying missing data imputation in electrical engineering studies degree’. In: *Hybrid Artificial Intelligence Systems. 11th International Conference, HAIS 2016* (2016), pp. 405–411.
- [72] E. Jove et al. ‘Attempts prediction by missing data imputation in engineering degree’. In: *SOCO-CISIS-ICEUTE*. 2017.
- [73] J. Xu, K. H. Moon and M. v. d. Schaar. ‘A machine learning approach for tracking and predicting student performance in degree programs’. In: *IEEE Journal of Selected Topics in Signal Processing* 11 (2017), pp. 742–753.
- [74] J. L. Schafer and J. W. Graham. ‘Missing data: Our view of the state of the art’. In: *Psychological Methods* 7.2 (2002), pp. 147–177.
- [75] J. L. Peugh and C. K. Enders. ‘Missing data in educational research: A review of reporting practices and suggestions for improvement’. In: *Review of Educational Research* 74.2 (2004), pp. 525–556.

- [76] R. J. A. Little. ‘A test of missing completely at random multivariate data with missing values’. In: *Journal of the American Statistical Association* 83.404 (1988).
- [77] C. Li. ‘Little’s test of missing completely at random’. In: *The Stata Journal* 13.4 (2013), pp. 798–809.
- [78] A. Gelman and J. Hill. *Data analysis using regression and multilevel/hierarchical models*. Cambridge University Press, 2007.
- [79] K. Grace-Martin. *How to diagnose the missing data mechanism*. URL: <https://www.theanalysisfactor.com/missing-data-mechanism/>.
- [80] J. Granberg-Rademacker. ‘A comparison of three approaches to handling incomplete state-level data’. In: *State Politics and Policy Quarterly* 7.3 (2007), pp. 325–338.
- [81] M. J. Azur et al. ‘Multiple imputation by chained equations: what is it and how does it work’. In: *International Journal of Methods in Psychiatric Research* 60.2 (2011), pp. 40–49.
- [82] S. van Buuren and K. Groothuis-Oudshoorn. ‘mice: Multivariate imputation by chained equations in R’. In: *Journal of Statistical Software* 45 (2011).
- [83] A. D. Shah et al. ‘Comparison of random forest and parametric imputation models for imputing missing data using MICE: a CALIBER study’. In: *American Journal of Epidemiology* 179.6 (2014).
- [84] D. J. Stekhoven and P. Bühlmann. ‘MissForest-non-parametric missing value imputation for mixed-type data’. In: *Bioinformatics* 28.1 (2012).
- [85] A. K. Waljee et al. ‘Comparison of imputation methods for missing laboratory data in medicine’. In: *British Medical Journal Open* 3 (2013).
- [86] B. Ramosaj and M. Pauly. ‘Who wins the Miss Contest for imputation methods? Our vote for Miss BooPF’. In: *arXiv preprint arXiv:1711.11394* 3 (2017).
- [87] J. W. Graham. *Missing data: Analysis and design*. New York: Springer, 2012.
- [88] N. Mduma, K. Kalegele and D. Machuve. ‘Machine learning approach for reducing students dropout rates’. In: *International Journal of Advanced Computer Research* 9 (2019), p. 42.
- [89] IBM Corp. ‘IBM Statistics for Windows, Version 25.0’. In: *Armonk, NY: IBM Corp* 83 (2017).
- [90] Y. Li, M. A. Palma and Z. P. Xu. ‘Impacts of playing after school on academic performance: a propensity score matching approach’. In: *Education Economics* 25.6 (2017).
- [91] M. Yang and S. E. Maxwell. ‘Treatment effects in randomized longitudinal trials with different types of nonignorable dropout’. In: *Psychological Methods* 19.2 (2014), pp. 188–210.

- [92] M. Mukaka. ‘A guide to appropriate use of Correlation coefficient in medical research’. In: *Malawi Medical Journal: The Journal of Medical Association of Malawi* 24.3 (2012).
- [93] M. N. van den Berg and W. Hofman. ‘Student success in university education: a multi-measurement study of the impact of student and faculty factors on study progress’. In: *Higher Education* 50 (2005), pp. 413–446.
- [94] B. F. French, J. C. Immekus and W. C. Oakes. ‘An examination of indicators of engineering students’ success and persistence’. In: *Journal of Engineering Education* 94 (2005), pp. 419–425.
- [95] S. Kotsiantis, C. Pierrakeas and P. Pintelas. ‘Predicting students’ performance in distance learning using machine learning techniques’. In: *Applied Artificial Intelligence* 18.5 (2004).
- [96] A. M. Shahiri, W. Husain and N. A. Rashid. ‘A review on predicting student’s performance using data mining techniques’. In: *Procedia Computer Science* 72 (2015), pp. 414–422.
- [97] M. Abdous, W. He and C.-J. Yen. ‘Using data mining for predicting relationships between online question theme and final grade’. In: *Educational Technology & Society* 15.3 (2012).
- [98] M. N. Quadri and N. V. Kalyankar. ‘Drop out feature of student data for academic performance using decision tree techniques’. In: 10.2 (2010), pp. 2–5.
- [99] T. Mishra, D. Kumar and S. Gupta. ‘Mining students’ data for prediction performance’. In: *Proceedings of the 2014 Fourth International Conference on Advanced Computing & Communication Technologies* (2014).
- [100] T. M. Christian and M. Ayub. ‘Exploration of classification using NBTree for predicting students’ performance’. In: *IEEE* (2014).
- [101] A. Daud et al. ‘Predicting student performance using advanced learning analytics’. In: *2017 International World Wide Web Conference Committee (IW3C2)* (2017), pp. 415–421.
- [102] S. Sembiring et al. ‘Prediction of student academic performance by an application of data mining techniques’. In: *2011 International Conference on Management and Artificial Intelligence (IPEDR)* 6 (2011), pp. 110–114.
- [103] A. Astin. *What matters in college?: Four critical years revisited*. Jossey-Bass Publishers, 1993.
- [104] J. Levin and J. Wyckoff. ‘Identification of student characteristics that predict persistence and success in an engineering college at the end of the sophomore year: Informing the practice of academic advising’. In: *Eric* (1990), pp. 1–42.

- [105] W. LeBold and S. Ward. ‘Engineering retention: National and institutional perspectives’. In: *Proceedings, 1988 American Society for Engineering Education Conference* (1988), pp. 843–851.
- [106] D. Budny, W. LeBold and G. Bjedov. ‘Assessment of the impact of freshman engineering courses’. In: *Journal of Engineering Education* 87 (1998), pp. 405–411.
- [107] E. Er. ‘Identifying at-risk students using machine learning techniques: A case study with IS 100’. In: *International Journal of Machine Learning and Computing* 2.4 (2012).
- [108] H. Agrawal and H. Mavani. ‘Student performance prediction using machine learning’. In: *International Journal of Engineering Research and Technology* 4.03 (2015), pp. 111–113.
- [109] Z. Ibrahim and D. Rusli. ‘Predicting students’ academic performance: Comparing artificial neural network, decision tree and linear regression’. In: *21st Annual SAS Malaysia Forum* (2007).
- [110] J. Gama et al. ‘A survey on concept drift adaptation’. In: *ACM Computing Surveys* 1.1 (2013).
- [111] B. E. Shelton, J.-L. Hung and P. R. Lowenthal. ‘Predicting student success by modeling student interaction in asynchronous online courses’. In: *Distance Education* 38 (2017), pp. 59–69.
- [112] S. Kotsiantis, D. Kanellopoulos and P. Pintelas. ‘Handling imbalanced datasets: A review’. In: *GESTS International Transactions on Computer Science and Engineering* 30.1 (2006).
- [113] G. Haixiang et al. ‘Learning from class-imbalanced data: Review of methods and applications’. In: *Expert Systems With Applications* 73 (1 2017).
- [114] Y. Qian et al. ‘A resampling ensemble algorithm for classification of imbalance problems’. In: *Neurocomputing* 143 (2014).
- [115] M. Galar et al. ‘A review on ensembles for the class imbalance problem: Bagging-, boosting-, and hybrid-based approaches’. In: *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)* 42.1 (2012).
- [116] C. Seiffert and T. M. Khoshgoftaar. ‘RUSBoost: A hybrid approach to alleviating class imbalance’. In: *IEEE Transactions on Systems, Man, and Cybernetics - Part A: Systems and Humans* 40 (1 2010).
- [117] G. Lemaître, F. Nogueira and C. K. Aridas. ‘Imbalanced-learn: A Python toolbox to tackle the curse of imbalanced datasets in machine learning’. In: *Journal of Machine Learning Research* 18.17 (2017), pp. 1–5.
- [118] J. Bergstra and Y. Bengio. ‘Random search for hyper-parameter optimization’. In: *Journal of Machine Learning Research* 13 (2012), pp. 281–305.

- [119] H. Lakkaraju et al. ‘A machine learning framework to identify students at risk of adverse academic outcomes’. In: *KDD’15* (2015).
- [120] R. K. Mothilal et al. ‘Predicting first-year engineering student success: From traditional statistics to machine learning’. In: *Proceedings of the 46th SEFI Annual Conference 2018; Vol. 46; pp. 322 - 329* (2018).
- [121] S. Nagrecha, J. Z. Dillon and N. V. Chawla. ‘MOOC dropout prediction: Lessons learned from making pipelines interpretable’. In: *Proceedings of the 26th International Conference on World Wide Web Companion* (2017).
- [122] W. Tian. ‘Predicting and interpreting students performance using supervised learning and Shapley Additive Explanations’. MA thesis. Arizona State University, 2019.
- [123] G. Louppe et al. ‘Understanding variable importances in forests of randomized trees’. In: *Advances in Neural Information Processing Systems 26* (2013).
- [124] C. Strobl et al. ‘Bias in random forest variable importance measures: Illustrations, sources and a solution’. In: *BMC Bioinformatics* 8 (25 2007).
- [125] A. Altmann et al. ‘Permutation importance: a corrected feature importance measure’. In: *Bioinformatics* (2010).
- [126] C. Strobl et al. ‘Conditional variable importance for random forests’. In: *BMC Bioinformatics* 9 (2008).
- [127] M. Korobov and K. Lopuhin. *ELI5 Documentation*. Release 0.9.0. 2019.
- [128] L. S. Shapley. ‘A value for n-person games’. In: *Contributions to the Theory of Games* (1953).
- [129] J. Pearl and D. Mackenzie. *The book of why*. Basic Books, 2018.
- [130] G. Machado, M. R. Mendoza and L. G. Corbellini. ‘What variables are important in predicting bovine viral diarrhea virus? A random forest approach’. In: *Veterinary Research* 46 (2015).
- [131] M. Kuhn and K. Johnson. *Applied Predictive Modelling*. Springer, 2013.
- [132] TeamHG-Memex. *ELI5*. 2017. URL: <https://github.com/TeamHG-Memex/eli5>.
- [133] T. L. Pedersen and M. Benesty. *lime: Local interpretable model-agnostic explanations*. 2019. URL: <https://CRAN.R-project.org/package=lime>.
- [134] S. Lundberg. *SHAP*. 2017. URL: <https://github.com/slundberg/shap>.
- [135] D. Alvarez-Melis and T. S. Jaakola. ‘On the robustness of interpretability methods’. In: *Proceedings of the 2018 ICML Workshop in Human Interpretability in Machine Learning* (2018).
- [136] E. A. Stuart. ‘Matching methods for causal inference: a review and a look forward’. In: *Stat. Sci.* 25.1 (2010), pp. 1–21.

- [137] J. Grimmer, S. Messing and S. J. Westwood. ‘Estimating heterogeneous treatment effects and the effects of heterogeneous treatments with ensemble methods’. In: *Political Analysis* 25.04 (2017), pp. 413–434.
- [138] S. Wager and S. Athey. ‘Estimation and inference of heterogeneous treatment effects using random forests’. In: *Journal of the American Statistical Association* 113 (2017).
- [139] A. Linden and P. R. Yarnold. ‘Using machine learning to assess covariate balance in matching studies’. In: *Journal of Evaluation in Clinical Practice* 22 (1988).
- [140] S. Zhao and N. Heffernan. ‘Estimating individual treatment effect from educational studies with residual counterfactual networks’. In: *Proceedings of the 10th International Conference on Educational Data Mining* (2017).
- [141] S. R. Künzel et al. ‘Metalearners for estimating heterogeneous treatment effects using machine learning’. In: *Proceedings of the National Academy of Sciences of the United States of America* 116.10 (2019).
- [142] S. L. Morgan and C. Winship. *Counterfactuals and causal inference. Methods and Principles for Social Research*. Cambridge University Press, 2007.
- [143] J. J. Heckman. ‘Selection bias and self-selection’. In: *Econometrics* (1990).
- [144] P. Rosenbaum and D. Rubin. ‘The central role of the propensity score in observational studies for causal effects’. In: *Biometrika* 70.1 (1983), pp. 41–55.
- [145] P. C. Austin. ‘An introduction to propensity score methods for reducing the effects of confounding in observational studies’. In: *Multivariate Behavioral Research* 46.3 (2011), pp. 399–424.
- [146] Y. Wang et al. ‘Optimal caliper width for propensity score matching of three treatment groups: A Monte Carlo study’. In: *PLOS ONE* 12.8 (2013), pp. 325–338.
- [147] M. Lunt. ‘Selecting an appropriate caliper can be essential for achieving good balance with propensity score matching’. In: *PLOS ONE* 179.2 (2013).
- [148] F. J. Boster et al. ‘Some effects of video streaming on educational achievement This project was supported by a contract from United Learning Corporation, now Discovery Education.’ In: *Communication Edition* 55:1 (2006), pp. 46–62.
- [149] M. Wieling and W. Hofman. ‘The impact of online video lecture recordings and automated feedback on student performance’. In: *Computers & Education* 54 (2010), pp. 992–998.
- [150] A. Williams, E. Birch and P. Hancock. ‘The impact of online lecture recording on student performance’. In: *Australasian Journal of Educational Technology* 28.2 (2012), pp. 1–9.

- [151] T. Traphagan, J. V. Kucsera and K. Kishi. ‘Impact of class lecture webcasting on attendance and learning’. In: *Educational Technology Research and Development* 58.1 (2010), pp. 19–37.
- [152] A. Le et al. ‘Online lecture accessibility and its influence on performance in skills-based courses’. In: *Computers & Education* 55.1 (2010), pp. 313–319.
- [153] J. A. Brotherton and G. D. Abowd. ‘Lessons learned from eClass’. In: *ACM Transactions on Computer-Human Interaction* 11.2 (2004), pp. 121–155.
- [154] W. Leadbeater et al. ‘Evaluating the use and impact of lecture recording in undergraduates: Evidence for distinct approaches by different groups of students’. In: *Computers & Education* 61.1 (2013), pp. 185–192.
- [155] D. Muller, J. Eklund and M. D. Sharma. ‘The future of multimedia learning: Essential issues for research’. In: *Paper presented at the Australian Association for Research in Education, Sydney, 2005* 18 (2005).
- [156] M. Ibrahim. ‘Implications of designing instructional video using cognitive theory of multimedia learning’. In: *Critical Questions in Education* 3.2 (2016).
- [157] M. Albert and B. J. Beatty. ‘Flipping the classroom applications to curriculum redesign for an introduction to management course: Impact on grades’. In: *Journal of Education for Business* (2014).
- [158] E. De Corte, L. Engwall and U. Teichler. ‘From books to MOOCs? Emerging models of learning and teaching in higher education’. In: *London: Portland Press Limited* (2016).
- [159] A. Hansch et al. ‘Video and online learning: Critical reflections and findings from the field’. In: *HIIG Discussion Paper Series No. 2015-02* (2015).
- [160] D. Muller. ‘Designing effective multimedia for physics education’. PhD thesis. University of Sydney, Australia, 2008.
- [161] V. Kettanurak, K. Ramamurthy and W. Haseman. ‘User attitude as a mediator of learning performance improvement in an interactive multimedia environment: an empirical investigation of the degree of interactivity and learning styles’. In: *International Journal of Human-Computer Studies* 54.4 (2001), pp. 541–583.
- [162] H. Astleitner and C. Wiesner. ‘An integrated model of multimedia learning and motivation’. In: *Journal of Educational Multimedia and Hypermedia* 13.1 (2004), pp. 3–21.
- [163] J. Boaler et al. ‘Changing student minds and achievement in mathematics: The impact of a free online student course’. In: *Frontiers in Education* 3.26 (2018).

- [164] C.-M. Chen and C.-H. Wu. ‘Effects of different video lecture types on sustained attention, emotion, cognitive load, and learning performance’. In: *Computers & Education* 80 (2015).
- [165] D. Figlio, M. Rush and L. Yin. ‘Is it live or is it internet? Experimental estimates of the effects of online instruction on student learning’. In: *Journal of Labor Economics* 31.4 (2013).
- [166] D. Card and A. B. Krueger. ‘Minimum wages and employment: A case study of the fast-food industry in New Jersey and Pennsylvania ’. In: *The American Economic Review* 84.4 (1994), pp. 772–793.
- [167] B. D. Meyer, W. K. Viscusi and D. L. Durbin. ‘Workers’ compensation and injury duration: Evidence from a natural experiment’. In: *The American Economic Review* 85.3 (1995), pp. 322–340.
- [168] K. Academy. *Khan Academy*. URL: <https://www.khanacademy.org/>.
- [169] R. C. Hibbeler. *Engineering Mechanics, Dynamics. Fourteenth Edition in SI Units*. 14th ed. Pearson, 2017.
- [170] C. Coffrin et al. ‘Visualizing patterns of student engagement and performance in MOOCs’. In: *LAK ’14 Proceedings of the Fourth International Conference on Learning Analytics and Knowledge* 1 (2014), pp. 83–92.
- [171] D. Muller. *This will revolutionize education*. URL: <https://www.youtube.com/watch?v=GEmuEWjHr5c>.

Appendix A Model Selection and Performance

This appendix presents the procedure for model selection for Chapter 4. It presents model selection for each model, i.e. lasso, random forest, gradient boosting, adaboost and neural network, on each dataset, namely imbalanced (IMB), random undersampled (RUS), random oversampled (ROS) and synthetic oversampled (SMOTE). It also presents results for the hybrid model RUSBoost.

Overall, for each dataset and model, the following presents optimal hyperparameters, scores on the training set, scores on the test set, confusion matrices and ROC-curves.

A.1 Imbalanced Data (IMB)

The original imbalanced dataset was first used to train the models.

A.1.1 Logistic regression with $L1$ regularization - IMB

For the lasso model, logistic regression with $L1$ regularization was used. Model selection via hyperparameter tuning was carried out on the training set (see Figure A.1) and an optimal AUC value was obtained at a C value of 2.60 ($\lambda = 0.038$).

Table A.1: Confusion matrix for *lasso* classification.

		Pred.		
		P	N	
Actual	P	143	6	149
	N	18	9	27
		161	15	

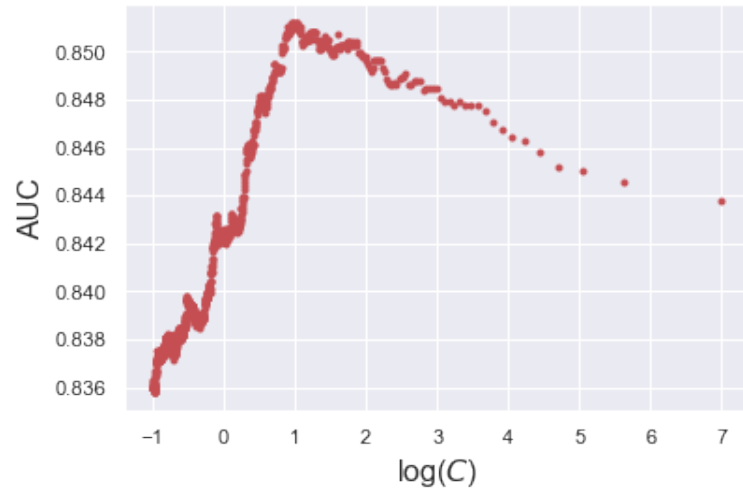


Figure A.1: Tuning of logistic regression parameter, C , based on AUC.

Table A.2: Lasso classification metrics on test set.

Accuracy	AUC	F1-score	κ	Brier score
0.86	0.65	0.43	0.36	0.14

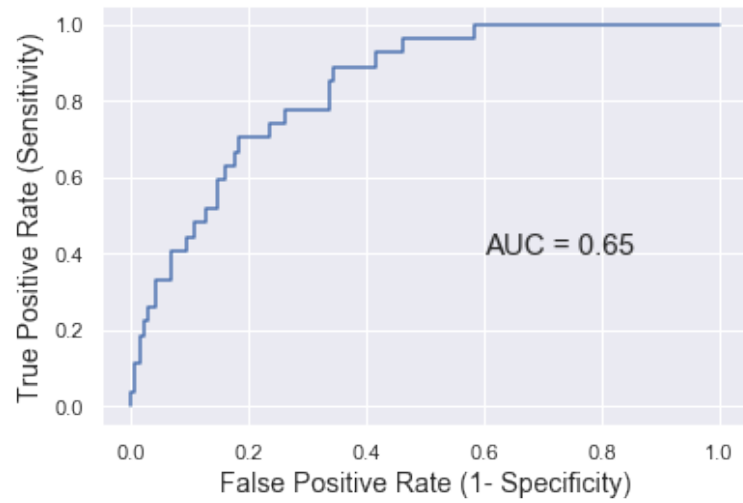


Figure A.2: ROC- curve for lasso classifier.

A.1.2 Random Forest - IMB

Table A.3: Random forest classifier optimal hyperparameters.

Parameter	Range of values
Num. trees	1500
Num. features	5
Min. samples split	4
Min. samples leaf	2
Bootstrap	Yes

Table A.4: Confusion matrix for random forest classification on imbalanced dataset.

		Pred.		
		P	N	
Actual	P	148	1	149
	N	21	6	27
		169	7	

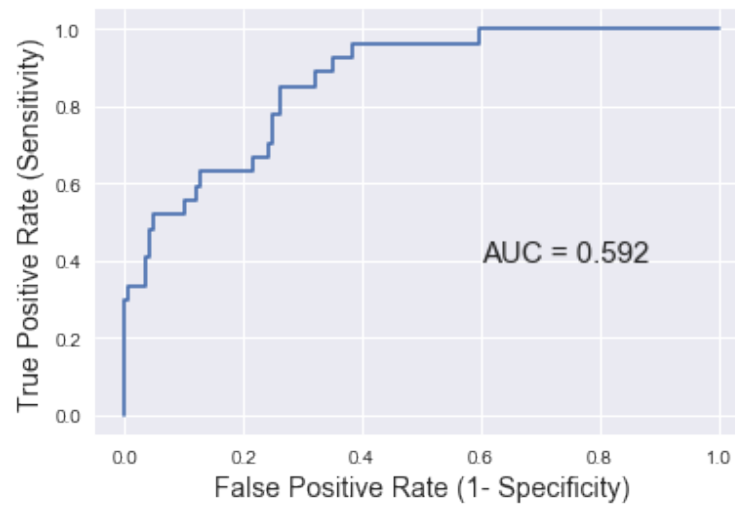


Figure A.3: ROC-curve for random forest classifier.

Table A.5: Random forest classification metrics on test set.

Accuracy	AUC	F1-score	κ	Brier score
0.88	0.61	0.35	0.31	0.13

A.1.3 Gradient Boosting - IMB

Table A.6: Gradient boosting optimal parameters.

Parameter	Range of values
Num. trees	1500
Learning rate	0.004
Subsample	0.5
Depth	4

Table A.7: Confusion matrix for gradient boosting classification on the imbalanced dataset.

		Pred.		
		P	N	
Actual	P	145	4	149
	N	18	9	27
		163	13	

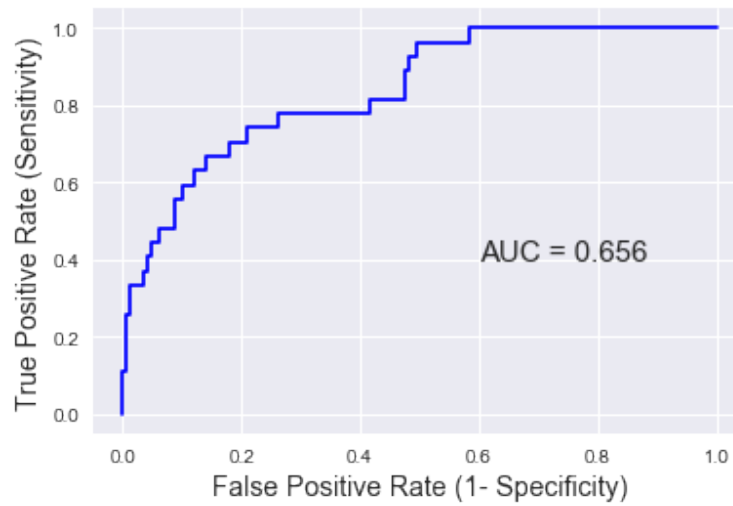


Figure A.4: ROC-curve for gradient boosting classifier.

Table A.8: Gradient boosting classification metrics on test set.

Accuracy	AUC	F1-score	κ	Brier score
0.88	0.65	0.45	0.39	0.13

A.1.4 Adaboost - IMB

Table A.9: Optimal Adaboost parameters.

Parameter	Range of values
Num. trees	1500
Learning rate	0.008

Table A.10: Confusion matrix for adaboost classification on imbalanced dataset.

		Pred.		
		P	N	
Actual	P	148	1	149
	N	19	8	27
		167	9	

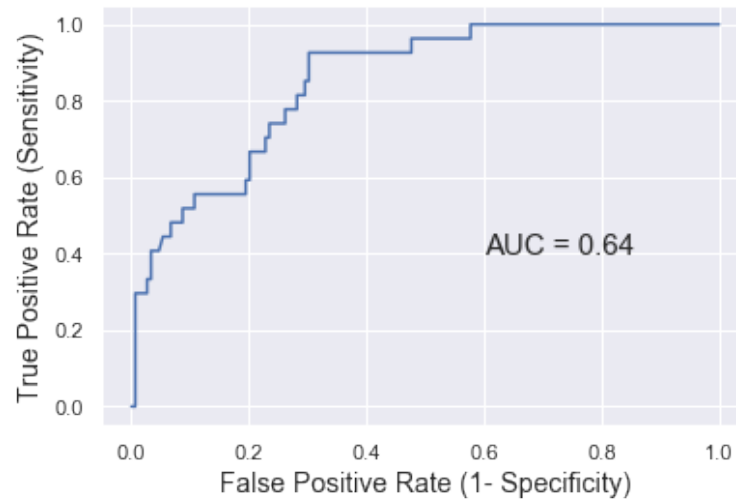


Figure A.5: ROC-curve for adaboost classifier.

Table A.11: Adaboost classification metrics on test set.

Accuracy	AUC	F1-score	κ	Brier score
0.89	0.64	0.44	0.40	0.11

A.1.5 Artificial Neural Networks - IMB

Table A.12: ANN parameter ranges for model selection on imbalanced data.

Parameter	Range of values
Initial Learning rate	0.009
Learning rate (schedule)	Adaptive
Activation	Tanh
Hidden layers	3
Hidden layer sizes	30,50,10
Momentum	0.9
Solver	SGD
Reg. param.	0.009

Table A.13: ANN classification metrics on test set.

Accuracy	AUC	F1-score	κ	Brier score
0.89	0.69	0.51	0.49	0.12

Table A.14: Confusion matrix for ANN classification on imbalanced dataset.

		Pred.		
		P	N	
Actual	P	144	5	149
	N	16	11	27
		170	16	

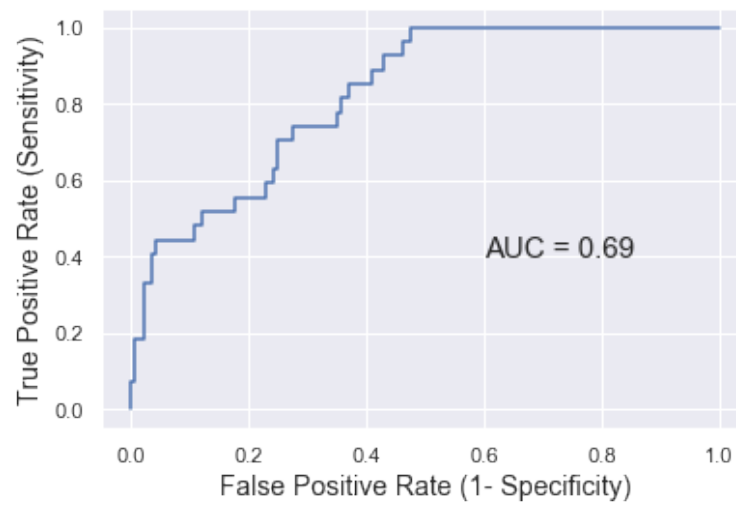


Figure A.6: ROC-curve for AdaBoost classifier.

A.2 Random undersampling (RUS)

Using the Imblearn under-sampling module called RandomUnderSampler, the Entire *training* set was reduced to a size of 266 rows and 30 columns. The 266 rows (observations) consist of an equal distribution of positive (at-risk) and negative classes, 133 each.

Training of the models was then carried out on the undersampled set and then tested on the original test set.

A.2.1 Logistic Regression with $L1$ regularization

The logistic regression model was tuned and a C parameter of 0.676 was obtained. Table A.15 and Figure A.7 show the training scores and the tuning of the parameter, respectively.

Table A.15: Lasso classification metrics on RUS training set.

Accuracy	AUC	F1-score
0.78	0.88	0.78

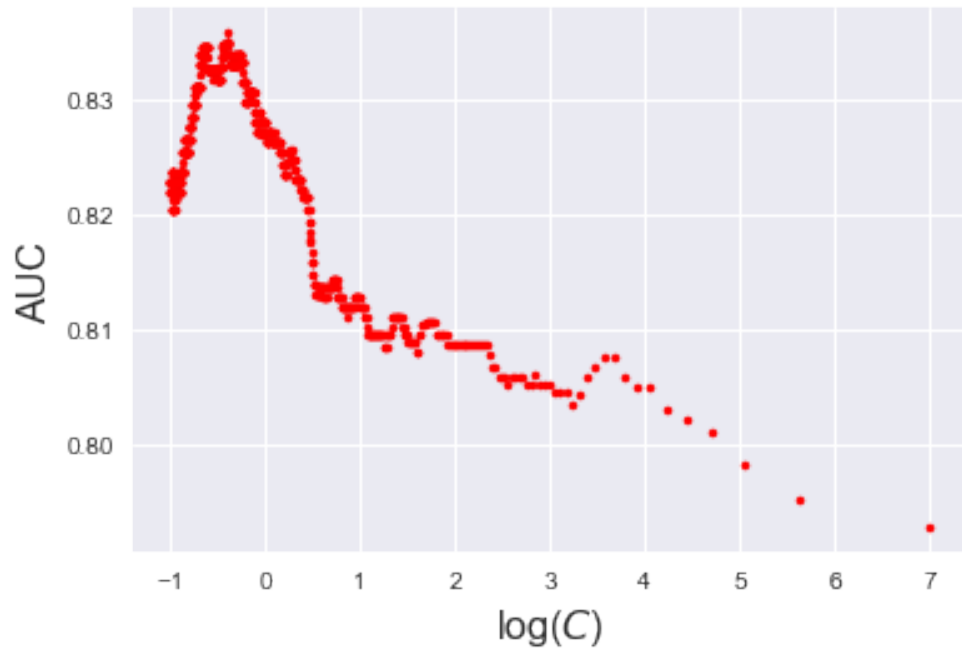


Figure A.7: Tuning of logistic regression parameter, C, based on AUC.

Table A.16: Confusion matrix for RUS lasso classification.

		Pred.		
		P	N	
Actual	P	114	35	149
	N	7	20	27
		121	55	

Table A.17: RUS lasso regression classification metrics on test set.

Accuracy	AUC	F1-score	Cohen's Kappa	Brier's Score
0.76	0.75	0.49	0.36	0.24

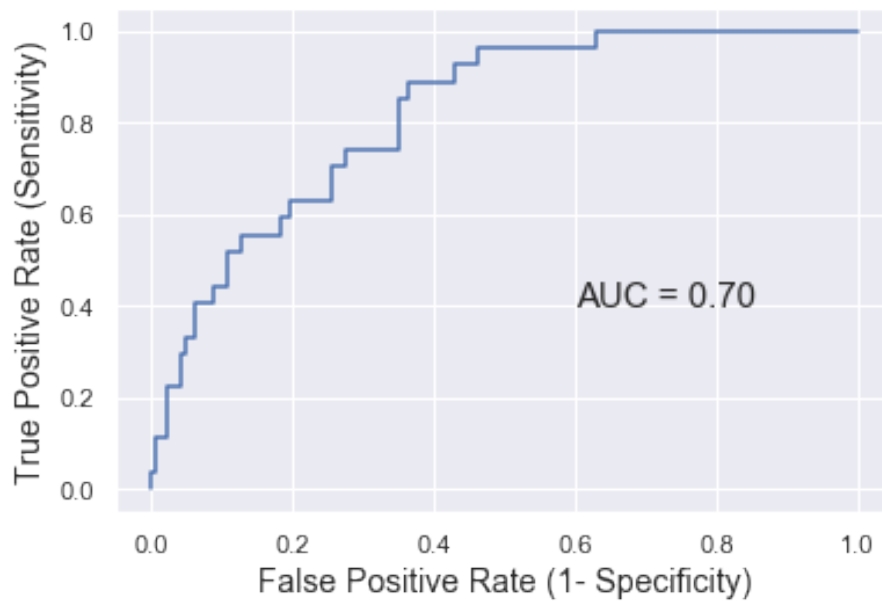


Figure A.8: ROC-curve for RUS Lasso regression classifier.

A.2.2 Random Forest - RUS

Table A.18: Random forest classifier optimal hyperparameters.

Parameter	Range of values
Num. trees	20
Num. features	5
Min. samples split	4
Min. samples leaf	4
Bootstrap	Yes

Table A.19: Random forest classification metrics on RUS training set.

Accuracy	AUC	F1-score
0.77	0.85	0.77

Table A.20: Confusion matrix for RUS random forest classification.

		Pred.		
		P	N	
Actual	P	108	41	149
	N	3	24	27
		111	65	

Table A.21: RUS random forest classification metrics on test set.

Accuracy	AUC	F1-score	Cohen's Kappa	Brier Score
0.75	0.81	0.52	0.39	0.25

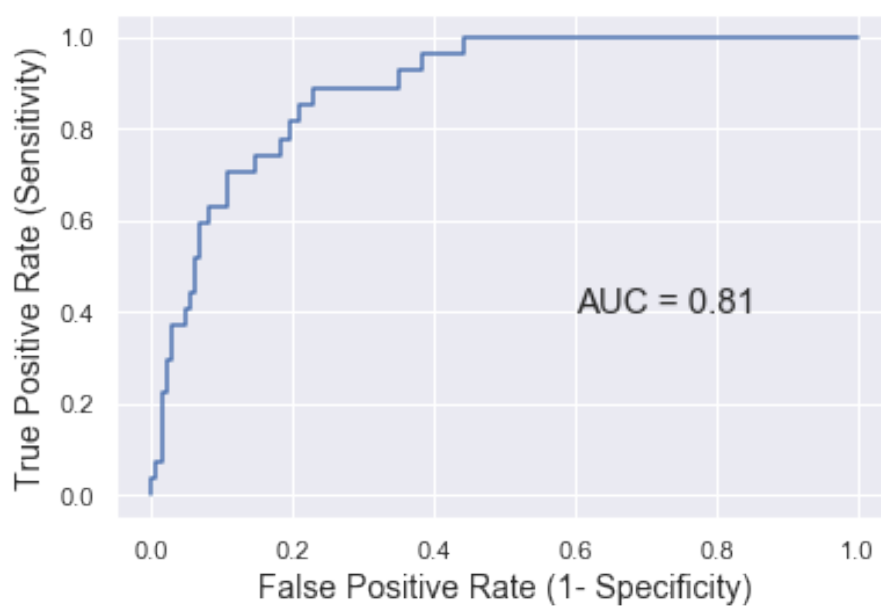


Figure A.9: ROC-curve for RUS random forest classifier.

A.2.3 Gradient Boosting - RUS

Table A.22: Gradient boosting optimal parameters.

Parameter	Range of values
Num. trees	500
Learning rate	0.007
Subsample	0.1
Depth	6

Table A.23: Gradient boosting classification metrics on RUS training set.

Accuracy	AUC	F1-score
0.78	0.88	0.79

Table A.24: Confusion matrix for RUS gradient boosting classification.

		Pred.		
		P	N	
Actual	P	115	34	149
	N	6	21	27
		121	55	

Table A.25: RUS gradient boosting classification metrics on test set.

Accuracy	AUC	F1-score	Cohen's Kappa	Brier Score
0.77	0.77	0.51	0.39	0.23

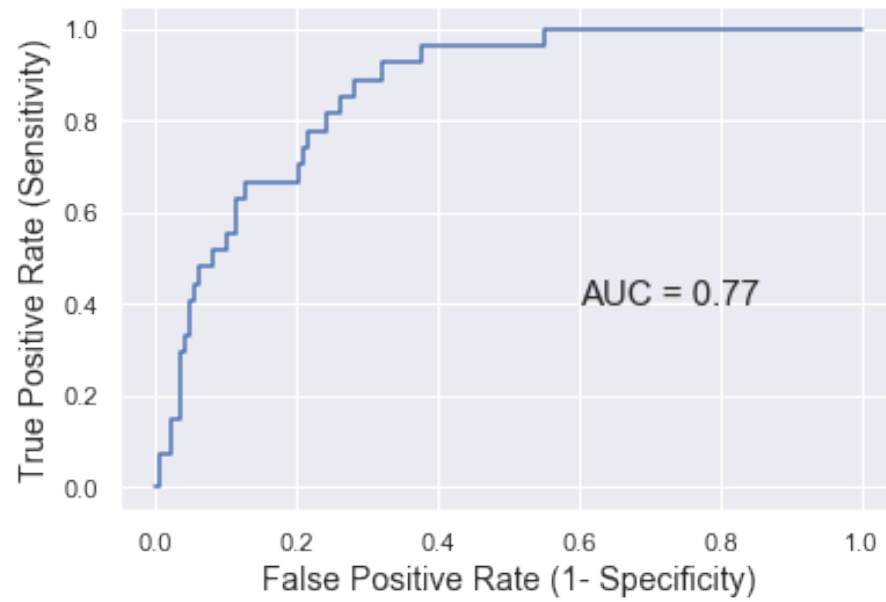


Figure A.10: ROC-curve for RUS gradient boosting classifier.

A.2.4 Adaboost - RUS

Table A.26: Gradient boosting optimal parameters.

Parameter	Range of values
Num. trees	3000
Learning rate	0.009

Table A.27: AdaBoost classification metrics on RUS training set.

Accuracy	AUC	F1-score
0.75	0.87	0.75

Table A.28: Confusion matrix for RUS AdaBoost classification.

		Pred.		
		P	N	
Actual	P	106	43	149
	N	8	19	27
		114	62	

Table A.29: RUS Adaboost classification metrics on test set.

Accuracy	AUC	F1-score	Cohen's Kappa	Brier Score
0.71	0.71	0.43	0.27	0.29

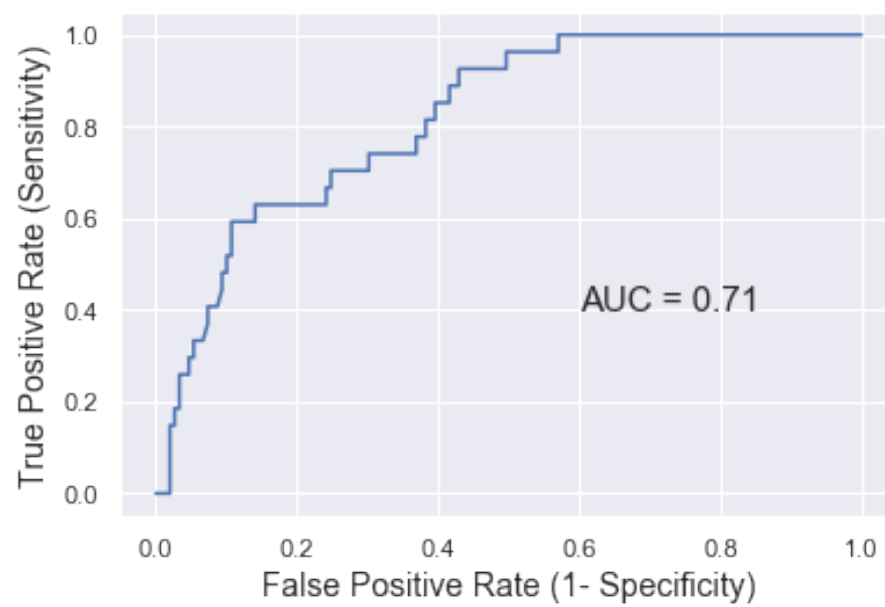


Figure A.11: ROC-curve for RUS adaboost classifier.

A.2.5 ANN - RUS

Table A.30: Optimal ANN parameters for RUS dataset

Parameter	Value
Solver	SGD
Momentum	0.5
Initial learning rate	0.015
Learning rate	Constant
Hidden layer size	(20,40,100)
Alpha	0.08
Activation	Tanh

Table A.31: ANN classification metrics on RUS training set.

Accuracy	AUC	F1-score
0.78	0.87	0.79

Table A.32: Confusion matrix for RUS AdaBoost classification.

		Pred.		
		P	N	
Actual	P	118	31	149
	N	8	19	27
		126	40	

Table A.33: RUS ANN classification metrics on test set.

Accuracy	AUC	F1-score	Cohen's Kappa	Brier Score
0.79	0.75	0.49	0.37	0.22

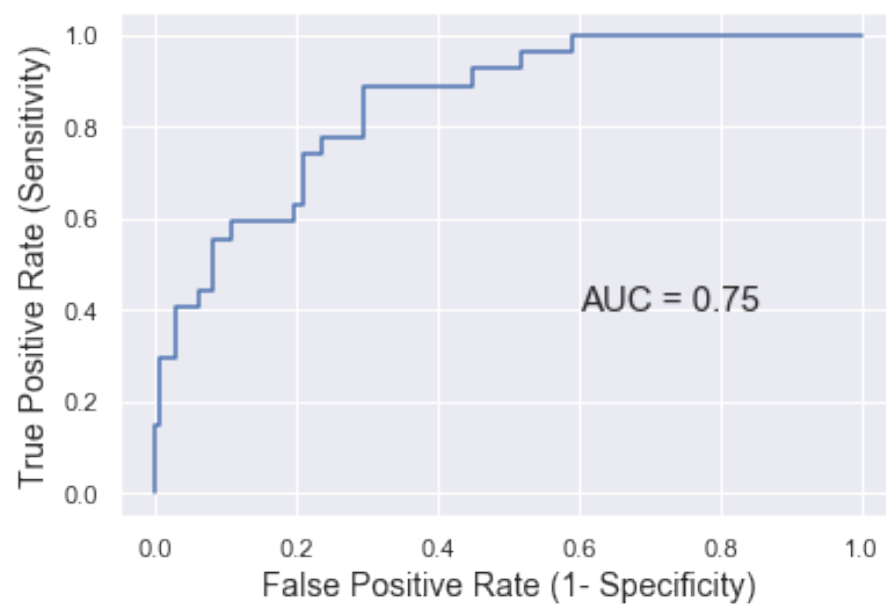


Figure A.12: ROC-curve for RUS ANN classifier.

A.3 Resampled Data - Random oversampling (ROS)

Oversampling was also applied to the Entire *training* set using the Imblearn oversampling module RandomOversampler, carried out in Python. The function oversamples the minority sample until it matches the majority. This resulted in a dataset of 1490 observations, evenly split between the positive and negative classes. Thereafter, an 80%/20% split was performed on the oversampled set, with stratification. All parameters for the various machine learning models was the same as the previous sections.

A.3.1 Logistic Regression with $L1$ regularization - ROS

Using the training data, the logistic regression model was tuned and a C parameter of 3.21 was obtained.

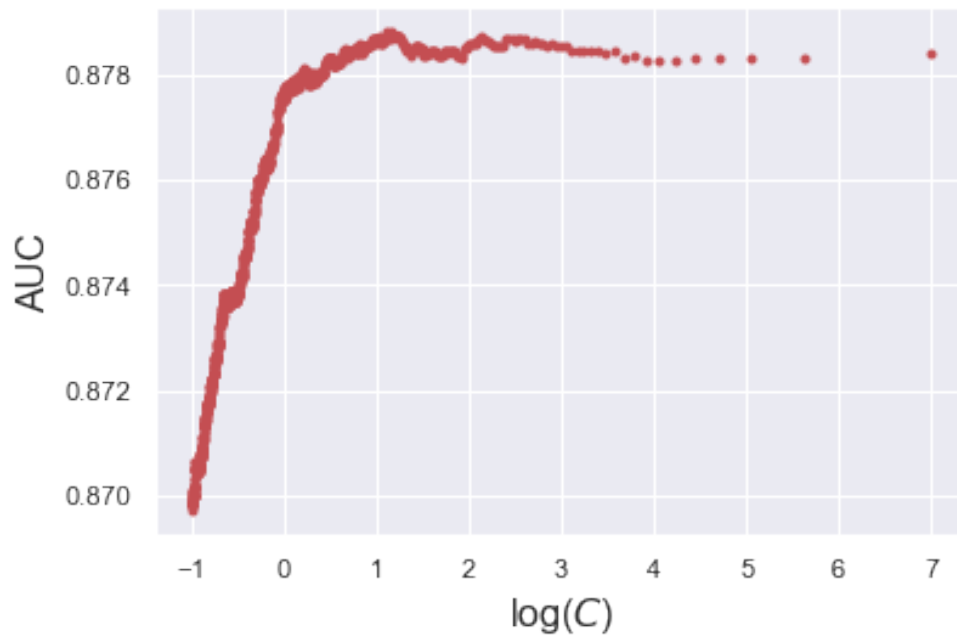


Figure A.13: Tuning of logistic regression parameter, C, based on AUC.

Table A.34: Lasso regression classification metrics on ROS training set.

Accuracy	AUC	F1-score
0.79	0.88	0.79

Table A.35: Confusion matrix for ROS lasso classification.

		Pred.		
		P	N	
Actual	P	120	29	149
	N	11	16	27
		131	45	

Table A.36: ROS lasso classification metrics on test set.

Accuracy	AUC	F1-score	Cohen's Kappa	Brier Score
0.77	0.70	0.44	0.31	0.22

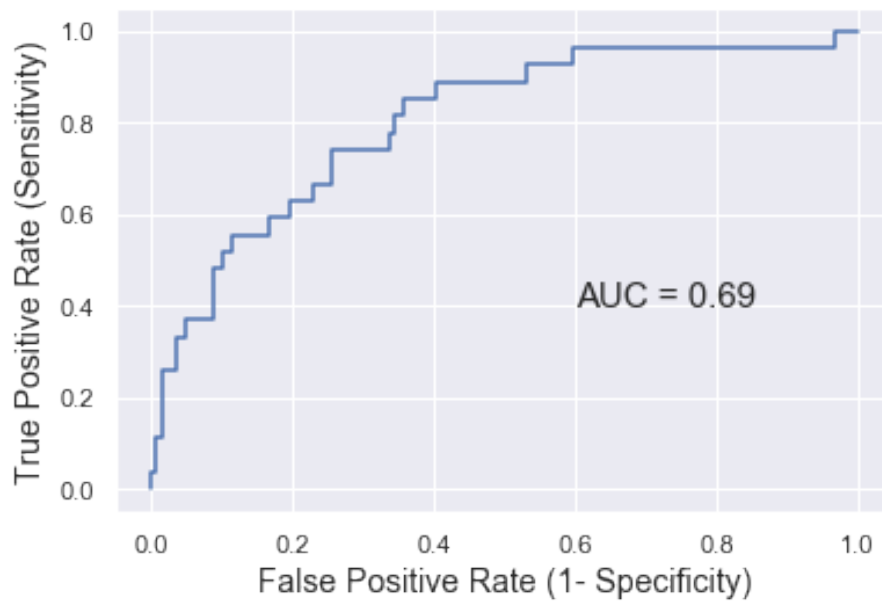


Figure A.14: ROC-curve for ROS Lasso regression classifier.

A.3.2 Random Forest - ROS

Table A.37: Random forest classifier optimal hyperparameters. - ROS

Parameter	Range of values
Num. trees	70
Num. features	11
Min. samples split	4
Min. samples leaf	1
Bootstrap	False

Table A.38: Random forest classification metrics on ROS training set.

Accuracy	AUC	F1-score
0.97	0.99	0.97

Table A.39: Confusion matrix for ROS lasso classification.

		Pred.		
		P	N	
Actual	P	145	4	149
	N	17	10	149
		162	14	

Table A.40: ROS random forest classification metrics on test set.

Accuracy	AUC	F1-score	Cohen's Kappa	Brier Score
0.88	0.67	0.49	0.43	0.12

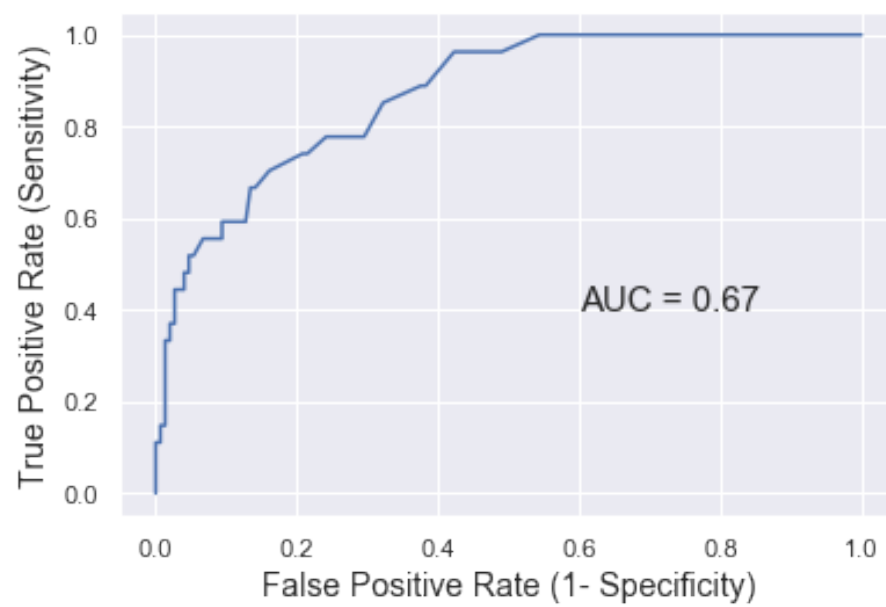


Figure A.15: ROC-curve for ROS random forest classifier.

A.3.3 Gradient Boosting - ROS

Table A.41: Gradient boosting ROS optimal parameters.

Parameter	Range of values
Num. trees	3000
Learning rate	0.007
Subsample	0.9
Depth	8

Table A.42: Gradient boosting classification metrics on ROS training set.

Accuracy	AUC	F1-score
0.97	0.99	0.87

Table A.43: Confusion matrix for ROS gradient boosting classification.

		Pred.			
		P	N		
Actual	P	143	6	149	
	N	16	11	27	
		159	17		

Table A.44: ROS gradient boosting classification metrics on test set.

Accuracy	AUC	F1-score	Cohen's Kappa	Brier
0.88	0.68	0.50	0.43	0.13

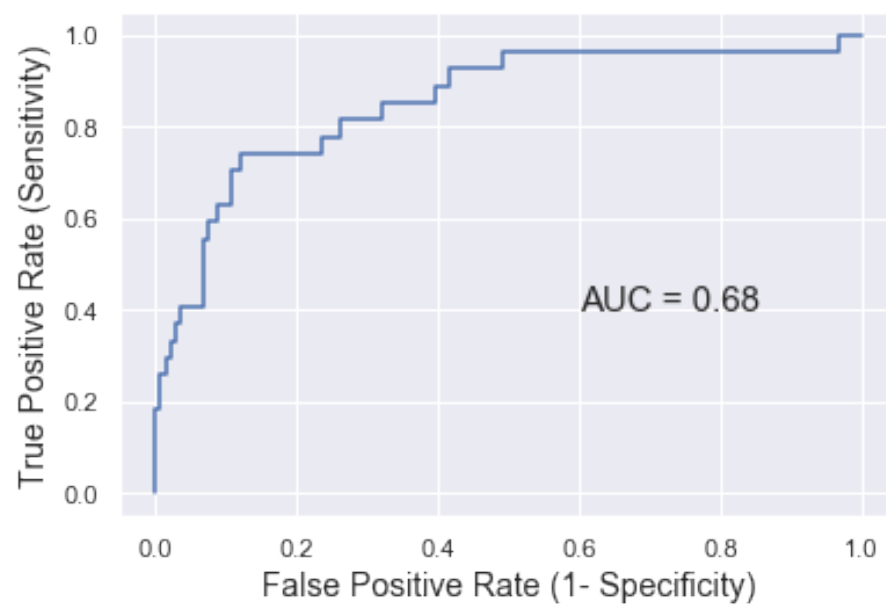


Figure A.16: ROC-curve for ROS gradient boosting classifier.

A.3.4 Adaboost - ROS

Table A.45: Gradient boosting ROS optimal parameters.

Parameter	Range of values
Num. trees	3000
Learning rate	0.01

Table A.46: AdaBoost classification metrics on ROS training set.

Accuracy	AUC	F1-score
0.81	0.91	0.81

Table A.47: Confusion matrix for ROS AdaBoost classification.

		Pred.		
		P	N	
Actual	P	116	33	149
	N	10	17	27
		126	50	

Table A.48: ROS Adaboost classification metrics on test set.

Accuracy	AUC	F1-score	Cohen's Kappa	Brier Score
0.76	0.70	0.44	0.30	0.24

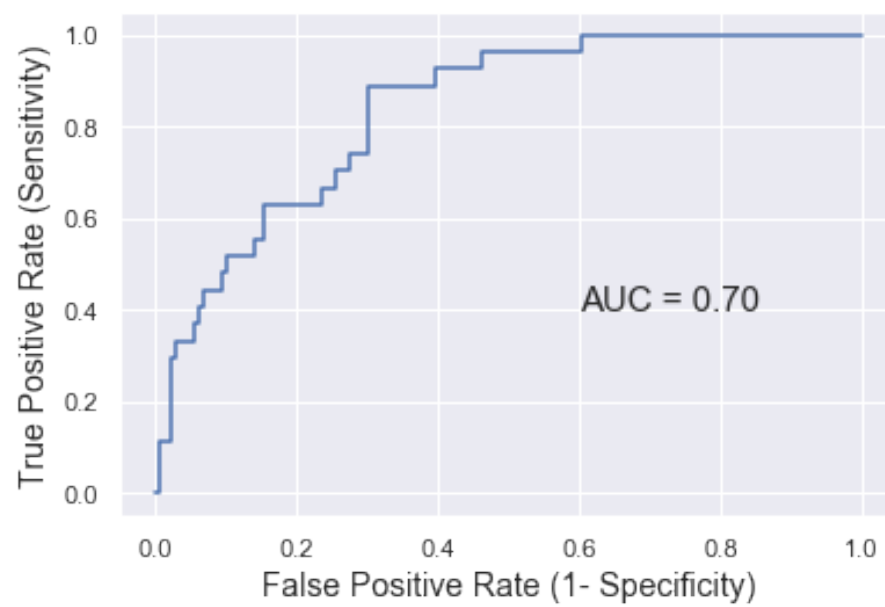


Figure A.17: ROC-curve for ROS adaboost classifier.

A.3.5 ANN - ROS

Table A.49: Optimal ANN parameters for ROS dataset

Parameter	Value
Solver	Adam
Momentum	0.9
Initial learning rate	0.031
Learning rate	Invscaling
Hidden layer size	(50,40,40)
Alpha	0.015
Activation	Tanh

Table A.50: ANN classification metrics on ROS training set.

Accuracy	AUC	F1-score
0.94	0.98	0.95

Table A.51: Confusion matrix for ROS ANN classification.

		Pred.		
		P	N	
Actual	P	128	21	149
	N	16	11	27
		144	32	

Table A.52: ROS ANN classification metrics on test set.

Accuracy	AUC	F1-score	Cohen's Kappa	Brier Score
0.79	0.63	0.37	0.25	0.21

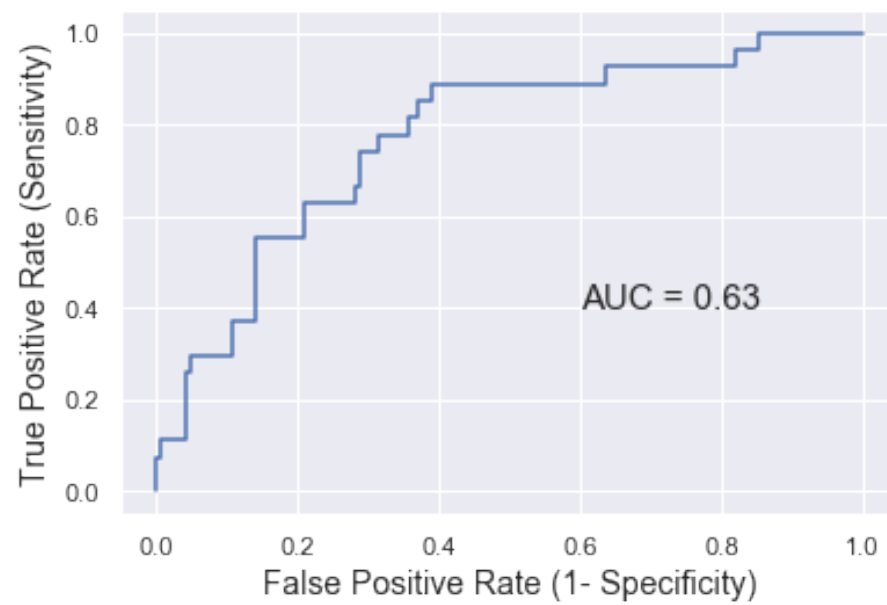


Figure A.18: ROC-curve for ROS ANN classifier.

A.4 Oversampling - SMOTE

Synthetic oversampling, SMOTE, was applied to the Entire *training* dataset. This resulted in a SMOTE dataset containing 1192 observations and 30 features.

A.4.1 Logistic regression with *L1* regularization - SMOTE

For the regularization parameter C , an optimal value of 22.50 was found.

Table A.53: Lasso regression classification metrics on SMOTE training set.

Accuracy	AUC	F1-score
0.82	0.90	0.82

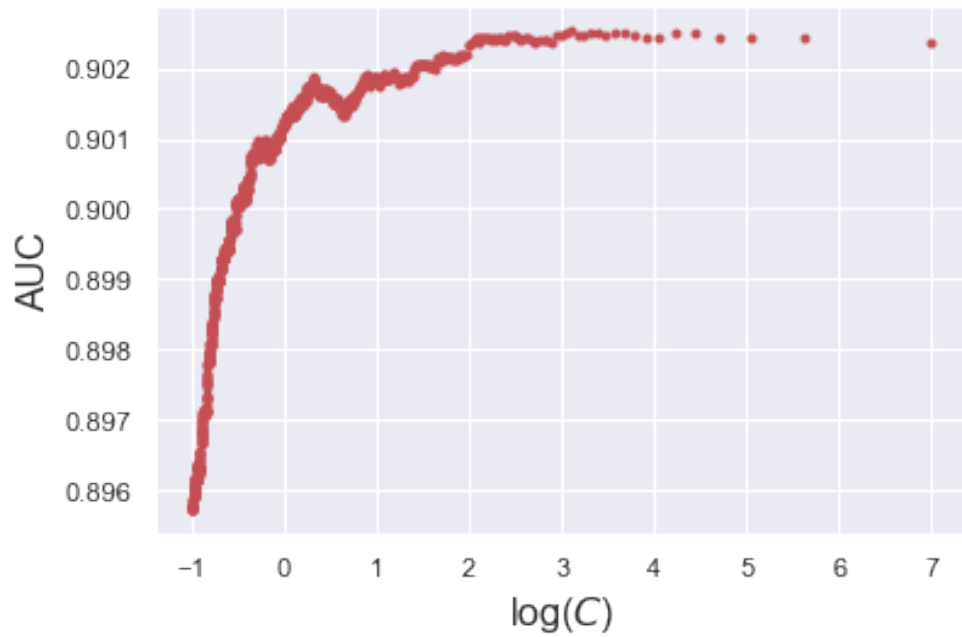


Figure A.19: Tuning of logistic regression parameter, C , based on AUC.

Table A.54: Confusion matrix for SMOTE lasso regression classification.

		Pred.		
		P	N	
Actual	P	122	27	149
	N	13	14	27
		135	41	

Table A.55: SMOTE lasso classification metrics on test set.

Accuracy	AUC	F1-score	Cohen's Kappa	Brier Score
0.77	0.67	0.41	0.28	0.23

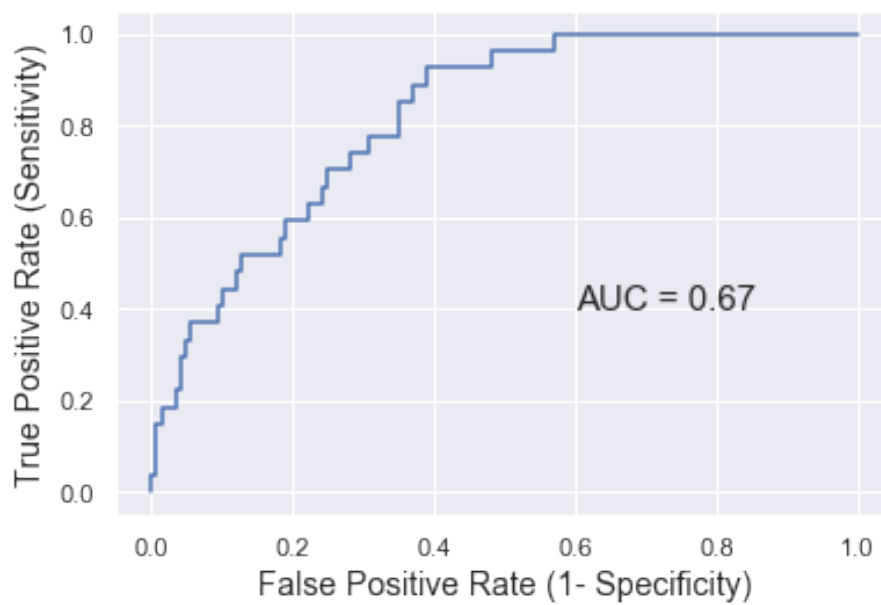


Figure A.20: ROC-curve for SMOTE lasso classifier.

A.4.2 Random Forest - SMOTE

Table A.56: Random forest classifier optimal hyperparameters for SMOTE dataset.

Parameter	Range of values
Num. trees	70
Num. features	3
Min. samples split	2
Min. samples leaf	1
Bootstrap	No

Table A.57: Random forest classification metrics on SMOTE training set.

Accuracy	AUC	F1-score
0.95	0.99	0.95

Table A.58: Confusion matrix for SMOTE lasso regression classification.

		Pred.		
		P	N	
Actual	P	144	5	149
	N	21	6	27
		162	14	

Table A.59: SMOTE random forest classification metrics on test set.

Accuracy	AUC	F1-score	Cohen's Kappa	Brier's
0.88	0.59	0.32	0.25	0.15

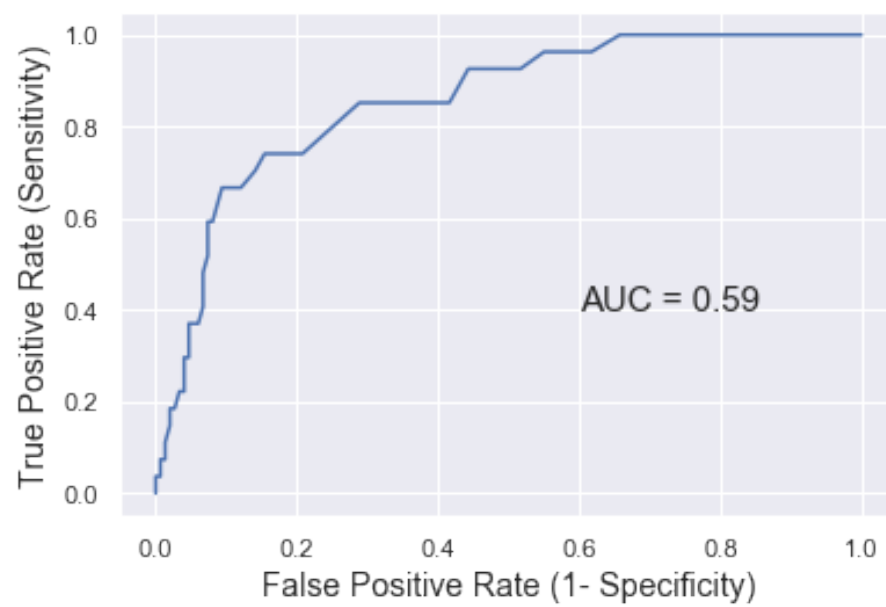


Figure A.21: ROC-curve for SMOTE random forest classifier.

A.4.3 Gradient Boosting - SMOTE

Table A.60: Gradient boosting optimal parameters, SMOTE.

Parameter	Range of values
Num. trees	3000
Learning rate	0.006
Subsample	0.7
Depth	7

Table A.61: Gradient boosting classification metrics on SMOTE training set.

Accuracy	AUC	F1-score
0.93	0.99	0.93

Table A.62: Confusion matrix for SMOTE gradient boosting classification.

		Pred.		
		P	N	
Actual	P	139	10	149
	N	18	9	27
		157	19	

Table A.63: SMOTE gradient boosting classification metrics on test set.

Accuracy	AUC	F1-score	Cohen's Kappa	Brier Score
0.84	0.63	0.39	0.30	0.16

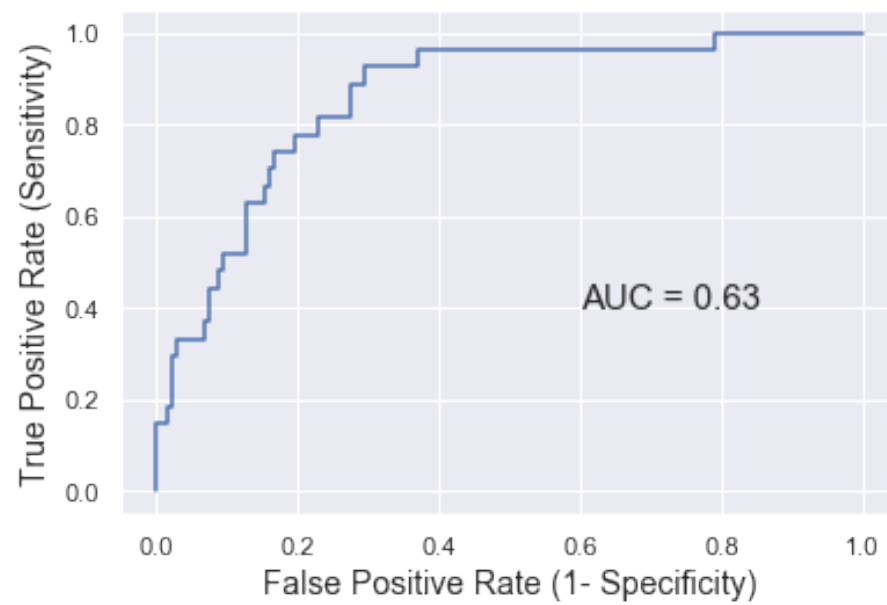


Figure A.22: ROC-curve for SMOTE gradient boosting classifier.

A.4.4 Adaboost - SMOTE

Table A.64: Adaboost optimal parameters, SMOTE.

Parameter	Range of values
Num. trees	3000
Learning rate	0.01

Table A.65: AdaBoost classification metrics on SMOTE training set.

Accuracy	AUC	F1-score
0.82	0.95	0.87

Table A.66: Confusion matrix for SMOTE AdaBoost classification.

		Pred.		
		P	N	
Actual	P	128	21	149
	N	12	15	27
		140	36	

Table A.67: SMOTE Adaboost classification metrics on test set.

Accuracy	AUC	F1-score	Cohen's Kappa	Brier
0.81	0.71	0.48	0.36	0.19

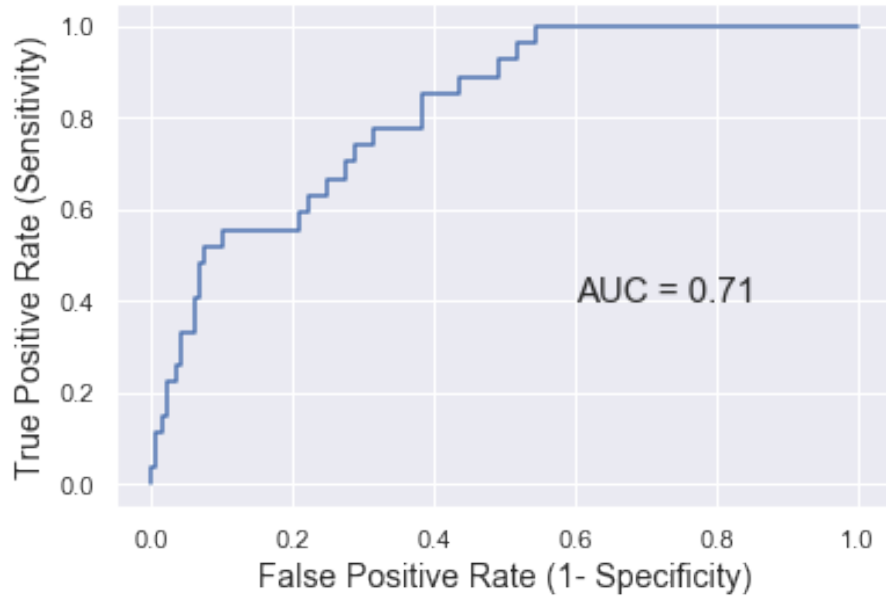


Figure A.23: ROC-curve for SMOTE adaboost classifier.

A.4.5 ANN - SMOTE

Table A.68: Optimal ANN parameters for SMOTE dataset

Parameter	Value
Solver	SGD
Momentum	0.9
Initial learning rate	0.035
Learning rate	Adaptive
Hidden layer size	(100,20,10)
Alpha	0.026
Activation	Relu

Table A.69: ANN classification metrics on SMOTE training set.

Accuracy	AUC	F1-score
0.94	0.98	0.95

Table A.70: SMOTE ANN classification metrics on test set.

Accuracy	AUC	F1-score	Cohen's Kappa	Brier Score
0.84	0.64	0.41	0.31	0.16

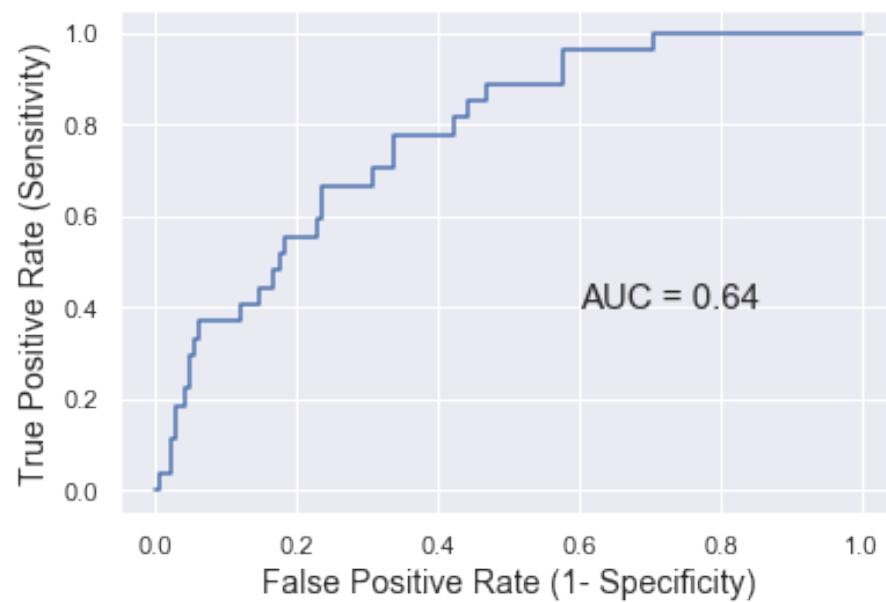


Figure A.24: ROC-curve for SMOTE ANN classifier.

Table A.71: Confusion matrix for SMOTE ANN classification.

		Pred.		
		P	N	
Actual	P	137	12	149
	N	17	10	27
		154	22	

A.5 RUSBoost

The model RUSBoost that combines random undersampling with boosting was carried out using the Scikit-learn function RUSBoostClassifier. The parameter ranges are the same as that for adaboost boost classifier, shown in Table 4.4. The optimal parameters are those in Table A.72 and the accuracy and AUC were calculated as 87.8% and 85.1 respectively. F1 score was 0.38

Table A.72: RUSBoosting optimal parameters.

Parameter	Range of values
Num. trees	100
Learning rate	0.001

Table A.73: RUSBoost classification metrics on training set.

Accuracy	AUC	F1-score
0.75	0.85	0.46

Table A.74: Confusion matrix for RUSBoost classification.

		Pred.		
		P	N	
Actual	P	115	34	149
	N	6	21	27
		121	55	

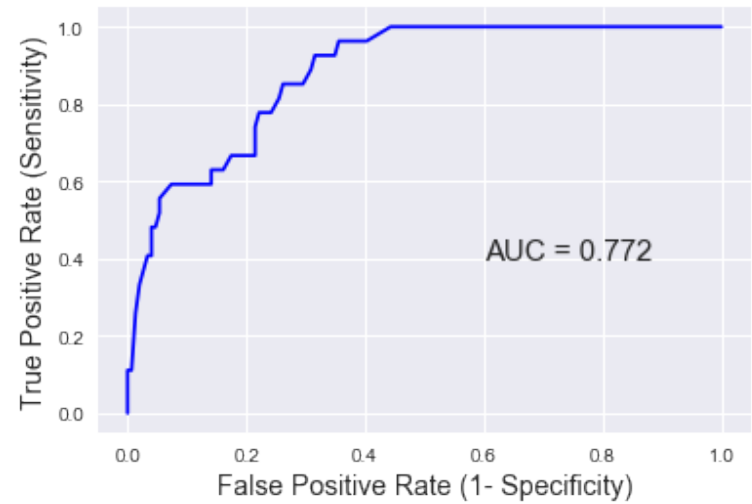


Figure A.25: ROC-curve for RUSBoost classifier.

Table A.75: RUSboost classification metrics on test set.

Accuracy	AUC	F1-score	Cohen's Kappa
0.79	0.77	0.53	0.41

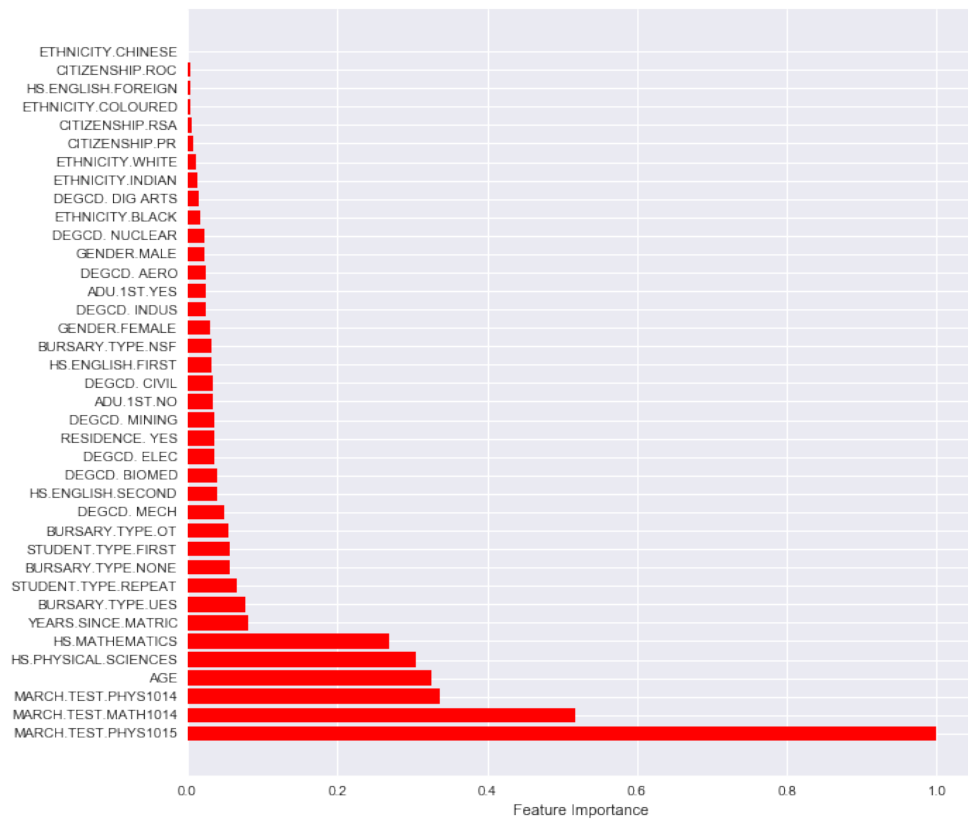


Figure A.26: Feature importances for RUSBoost classifier predicting At-Risk students.

Appendix B Model Selection on Control and Treatment Groups

For each model, ten-fold CV using RandomizedSearchCV from Scikit-learn was used for model selection.

B.1 Lasso

B.1.1 Control Group

The mean absolute error computed over the lasso hyperparameter range is presented in Figure B.1. An optimal lasso parameter of 0.0619 was obtained. Furthermore, a minimum MAE of 7.57% was calculated.

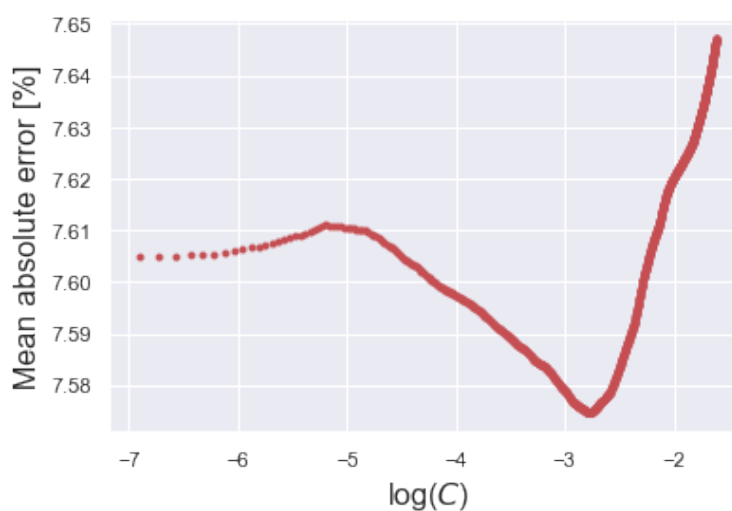


Figure B.1: Mean absolute error for lasso model tuned on the control dataset.

B.1.2 Treatment Group

A minimum MAE of 6.91% at a lambda parameter of 0.0938 was computed as shown in Figure B.2.

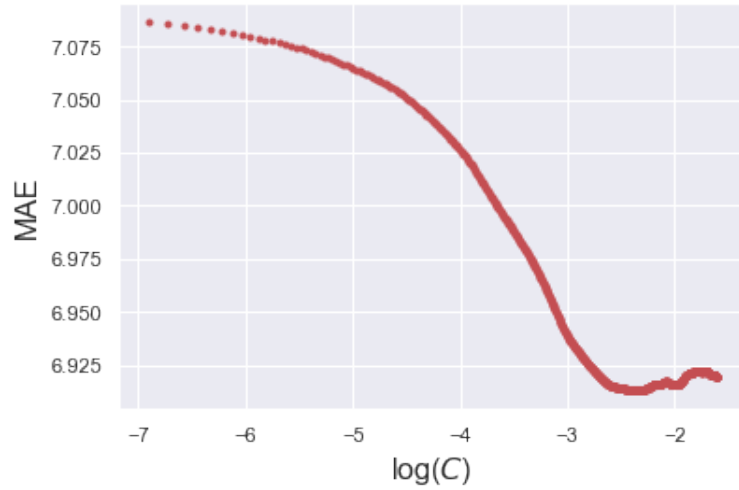


Figure B.2: Mean absolute error for lasso model tuned on the treatment dataset.

B.2 Random Forest

Table B.1: Random forest regression hyperparameter ranges.

Parameter	Range of values
Num. trees	10:100 in increments of 10; then 100, 500, 1000, 1500, 2000.
Num. features	$p/3$ (10) , $p/2$ (14), p (28)
Min. samples split	2, 4, 8
Min. samples leaf	1, 2 , 4
Bootstrap	Yes, No

B.2.1 Control Group

For the control group, a minimum MAE of 7.31% was calculated for random forest at the hyperparameters shown in Table B.2

Table B.2: Random forest optimal hyperparameters for the Control Group.

Parameter	Range of values
Num. trees	2000
Num. features	$p/3$ (9)
Min. samples split	2
Min. samples leaf	2
Bootstrap	Yes

B.2.2 Treatment Group

For the treatment group, a minimum MAE of 6.62% was calculated for random forest at the hyperparameters shown in Table B.3.

Table B.3: Random forest optimal hyperparameters for the Treatment Group.

Parameter	Range of values
Num. trees	80
Num. features	$p/3$ (9)
Min. samples split	2
Min. samples leaf	1
Bootstrap	Yes

B.3 Gradient Boosting

Table B.4: Gradient boosting hyperparameter ranges for tuning.

Parameter	Range of values
Num. trees	100, 500, 1000, 1500, 2000, 3000
Max depth	4, 5, 6, 7, 8
Learning rate	0.001: 0.01, increments of 0.001
Subsample	0.1: 0.9, increments of 0.1

B.3.1 Control Group

For the control group, a minimum MAE of 7.33% was calculated for gradient boosting at the hyperparameters shown in Table B.5.

Table B.5: Optimal gradient boosting hyperparameters for Entire Group.

Parameter	Value
Num. trees	3000
Max depth	7
Learning rate	0.001
Subsample	0.2

B.3.2 Treatment Group

For the treatment group, a minimum MAE of 6.62% was calculated for gradient boosting at the hyperparameters shown in Table B.6.

Table B.6: Optimal gradient boosting hyperparameters for Entire Group.

Parameter	Value
Num. trees	2000
Max depth	5
Learning rate	0.001
Subsample	0.5

B.4 Adaboost

Table B.7: Adaboost hyperparameter ranges for tuning.

Parameter	Range of values
Num. trees	100, 500, 1000, 1500, 2000, 3000
Learning rate	0.001: 0.01, increments of 0.001

B.4.1 Control Group

For the treatment group, a minimum MAE of 7.94% was calculated for gradient boosting at the hyperparameters shown in Table B.8.

Table B.8: Optimal adaboost hyperparameters for the Control Group.

Parameter	Value
Num. trees	100
Learning rate	0.008

B.4.2 Treatment Group

For the treatment group, a minimum MAE of 6.86% was calculated for gradient boosting at the hyperparameters shown in Table B.9.

Table B.9: Optimal adaboost hyperparameters for Treatment Group.

Parameter	Value
Num. trees	500
Learning rate	0.002

B.5 Artificial Neural Network

Table B.10: ANN parameter ranges for model selection.

Parameter	Range of values
Initial Learning rate	$10^{-6} : 10^{-1}$
Learning rate (schedule)	Constant, Adaptive, Invscaling
Activation	Identity, Logistic, Tanh, Relu
Hidden layers	1:3
Hidden layer sizes	10:50, 100
Momentum	0.5, 0.9, 0.99
Solver	Adam, SGD, LBGFS
Reg. param.	$10^{-6} : 10^{-1}$

B.5.1 Control Group

For the control group, a minimum MAE of 7.36% was calculated for ANN at the hyperparameters shown in Table B.9.

B.5.2 Treatment Group

For the treatment group, a minimum MAE of 6.90% was calculated for gradient boosting at the hyperparameters shown in Table B.9.

Table B.11: Optimal ANN parameters for Control Group

Parameter	Value
Solver	Adam
Momentum	0.9
Initial learning rate	1e-3
Learning rate	Adaptive
Hidden layer size	(50,100,10)
Alpha	1e-6
Activation	Relu

Table B.12: Optimal ANN parameters for Treatment Group

Parameter	Value
Solver	Adam
Momentum	0.99
Initial learning rate	1e-3
Learning rate	Adaptive
Hidden layer size	(50,100,10)
Alpha	1e-6
Activation	Relu