Applied Soft Computing Journal 174 (2025) 113035

Contents lists available at ScienceDirect

Applied Soft Computing

journal homepage: www.elsevier.com/locate/asoc

# A feature engineering based model architecture for modeling initial public offerings

Durga Vaidynathan [a], Parthajit Kayal [a], Moinak Maiti [b,*]

[a] Madras School of Economics, Behind Government Data Centre, Gandhi Mandapam Road, Kottur, Chennai 600025, India
[b] Department of Finance, School of Economics and Finance, University of the Witwatersrand, Johannesburg, South Africa

## HIGHLIGHTS

- A novel feature engineered based predictive model is developed to model Initial Public Offerings.
- SHAP values are used for model debugging.
- The study findings have high economic implications.

## ARTICLE INFO

*Keywords:*
IPO
Feature Engineering
SHAP
Game theory
Information asymmetry

## ABSTRACT

This study proposes a model architecture for modeling Initial Public Offerings (IPOs) by incorporating a diverse range of data sources, encompassing both textual and numerical inputs. Language models, machine learning models, and deep learning architectures are combined to make the final ensemble predictions. Several rich features are engineered and interpreted while providing scope for debugging using the game theory-based Shapley Additive exPlanations (SHAP) values. The study results indicate that the feature-engineering is highly eloquent in IPO performance modelling. The study findings have high economic implications range from detecting the market trends to overall market stability.

## 1. Introduction

Predictive modeling of Initial Public Offerings (IPO) is quite challenging given that the market needs to be better aware of the company except for the information in the media and the official prospectus and other such documents released by the company per the regulations [1]. Unlike well-established companies, investing in the stocks of emerging companies reflects systematic decision-making and a hunch or trust mechanism induced by the information available [2–4]. The earlier studies indicate that the OLS (ordinary least square) regression is still the preferred model for modelling the IPO. However, the linear regression models are inefficient in dealing with the tailed data that are common in IPO [5]. To deal with this issue related to the IPO modelling various studies have deployed a variety of artificial intelligence (AI) techniques and other advanced models [5–8].

To model the IPO performances earlier studies are heavily dependent on numerical data points like the cash flow ratio, return of assets, and macroeconomic variables [9–11]. Then several studies also highlighted that in addition to the various financial and economic indications, investors sentiment do affect the IPO performances significantly [5,12]. As a result, studies deploy various textual analysis tools to extract important information from the news articles and search trends to reflect on the decision-making behavior and model the same [13]. The vast amount of data on the internet and advancements in language models make it possible to abstract factual and up to date information for analysis [14,15].

Nevertheless, merely supplying available data to models' risks falling into a "garbage-in, garbage-out" scenario. Confidence in the machine learning (ML) models is crucial, and feature engineering is essential for maximizing the performance of it [16]. The distinctive contribution of the current study lies is that it attempts to propose a feature engineering approach to condense the kinds and amounts of online information available. It explores various ways to model a learning architecture that learns from rich data sources consisting of multiple sources of data. Utilizing the SHAP values the present study shows that the feature-engineering is highly eloquent in IPO performance modelling.

* Corresponding author.
*E-mail addresses:* fe22durga@mse.ac.in (D. Vaidynathan), parthajit@mse.ac.in (P. Kayal), maitisoft@gmail.com, moinak.maiti@wits.ac.za (M. Maiti).

https://doi.org/10.1016/j.asoc.2025.113035
Received 6 May 2024; Received in revised form 31 August 2024; Accepted 9 March 2025
Available online 20 March 2025

The remainder of the paper is organized as follows: Section 2 furnishes a concise review of the pertinent literature. Section 3 expounds on the data and methodology, encompassing the collection of data pertaining to distinctive features, the pre-feature engineering process undertaken to extract additional features prior to modeling, and the architecture of the model. Section 4 engages in a comprehensive discussion of the study's findings, addressing implications, limitations, and presenting avenues for future research. The paper concludes with the final Section 5.

## 2. Literature review

IPOs constitute a pivotal measure of new capital formation within an economy, functioning as a crucial indicator of long-term economic growth. Serving as catalysts for employment and capital infusion into production by directing savings, the returns from IPOs have garnered considerable interest in scholarly exploration. Traditionally, researchers have turned to regression techniques to elucidate factors contributing to IPO under-pricing and volatility. For instance, a recent study employed empirical models based on multiple linear regressions, determining that a firm's age, IPO years, ownership structure, issue size, and market capitalization collectively accounted for 44 % of the variation in issuer under-pricing [17]. Apart from under-pricing studies, if we turn our attention into information asymmetry theories, variables can be categorized into signalling variables, firm characteristics, and financial indicators [18]. The use of signalling variables helps quantify the signalling hypothesis, which advocates that certain characteristics in a company can send signals to potential investors on the credibility of the company. Empirical examinations by [19] affirmed the role of signalling in the valuation of IPOs of common stock. They argued that ownership retention sends a signal to investors on the value of the firm, thus reducing information asymmetry between issuer and investors. [18] extended this line of inquiry, utilizing auditors'/underwriter's reputation, ownership retention, and over-subscription rate as group of signalling variables and perform a hierarchical regression and step-wise regression to analyze the relationship and the associated statistical significance of the said relationships. The study further delved into the connections between signalling variables and financial variables for additional insights.

The study by [20] tackle the lower explanatory power of traditional econometric models in pricing IPOs by proposing an Artificial Neural Network (ANN) model that takes in 11 numerical input firm-specific variables and predicts the first-day close of the IPOs in the sample. The study findings show that neural networks consistently outperform the traditional methods. Also, it provides an approach to arrive at the feature importance which is the drawback when using a black box model like neural nets. The study computes the relative strength of each input unit on the output variable using a methodology proposed by [21]. It pioneered the first steps in using machine learning and deep learning models in pricing IPOs, which was later extended by various researchers [22–25]. In a more recent study by [5] use a data set of underpriced IPOs on Borsa Istanbul, and demonstrate the potential of random forest as a powerful alternative to linear regression for predicting initial returns. Their work reveals that random forest outperformed robust regression methods in terms of accuracy, with predictions closely matching actual returns and lower standard deviation. Notably, their analysis identified IPO proceeds and volume as the most influential factors, providing further insights into the role of ex-ante uncertainty in IPO performance. Taking a novel approach [26], propose a method for assessing IPO pricing efficiency through a Stochastic Frontier Analysis and Deep Neural Network. It demonstrated the superior performance of neutral networks compared to the traditional linear methods in estimating offer prices using US IPO data. It also suggests methods for analyzing their estimated models and draw economic interpretations for the impact of independent variables on dependent variables; while finding critical determinants of IPO efficiency like negative net income, EBITDA etc.

The Securities and Exchange Commission (SEC) mandates companies going public to file S-1 and form 424B, which is the prospectus of the company providing detailed information about the company, covering sections like the prospectus summary, risk factors, use of proceeds, management's discussions, and analysis. The study by [27] shed light on the role of ex ante uncertainty in IPO under-pricing. It argues that high frequencies of uncertain, weak modal, and negative words in the S-1 filing (a proxy for pre-IPO uncertainty) contribute to higher first-day returns and greater post-market volatility. These findings align with [28] theory of ex-ante uncertainty. In a more recent study [13], highlights the impact of forward-looking statements on valuation in the IPO prospectuses section. By constructing an FLS classifier built on deep learning techniques, it demonstrates the value of different features as predictors for IPO under-pricing (both pre- and post-IPO returns).

The issue of interpretability and the determination of feature importance is prevalent in most machine learning models. Further, the adoption of advanced machine learning models raises the level of difficulty in understanding the underlying intuitions behind their predictions [20]. Despite their superior performance, these models often function as "black box" systems and therefore, making it challenging for researchers and market practitioners to use the results for useful decision-making processes. In the present study, we try to enhance interpretability in machine learning models in the context of IPO pricing and thereby helping in decision-making process. There has been considerable amount of effort by the academic researchers to make machine learning models more interpretable. For example, feature importance analysis exhibits relative strength of each input variable on their impact on the output [20,29]. Additionally, models like decision trees, LIME (Local Interpretable Model-agnostic Explanations), and SHAP (Shapley Additive exPlanations) have helped to demystify the decision-making process of complex models [30].

Inclusion of textual data [13,27] enhance predicting IPO pricing. It allows us to investigate the language used in regulatory filings and thereby helps us gauging ex-ante uncertainty. It also helps in comprehending the impact of forward-looking statements on IPO valuation. For text analysis, integration of natural language processing techniques and sentiment analysis into the prediction model could provide valuable insights. In other words, it provides the linguistic cues that may influence market perceptions among the participants and thereby, contribute to dynamics of under-pricing. Despite the advancements in machine learning fields that allows textual data analysis, the basic challenge of balancing predictive power with reasonable level of model interpretability continues. Combination of textual data and numerical predictors poses more significant challenge as highlighted by [31]. The study simultaneously uses the textual features and financial variables inputs in machine learning models and achieve out-of-sample accuracy of 50 percent and more in most cases. However, the study perceives that combining textual and numerical data brings down models' predictive power. This is possibly due to the high dimensionality nature of the textual data. Nevertheless, their work provides useful methodological insights to handle plethora of textual features efficiently along with numerical data into the machine learning algorithms.

This study aims to combine the prospectus' textual data by quantifying the public interest in the company using Google Trends and Google search results analysis, along with traditional numerical predictors. It proposes feature engineering techniques to draw meaningful features from the textual sources. Exploring both concepts of textual regression [32–34] and classification techniques, it tries to derive ways to establish a model architecture to layer up the predictions for better interpretability. It also examines how one can better use the SHAP values for model debugging. Being a purely experimental study to devour the curiosity of how to make better use of the advanced language models and feature engineering methods to support the huge amount of data available online, it intends to provide a rich initial layer of prediction by providing scope for model-debugging, a less explored arena in studies using machine learning techniques for IPO prediction.

## 3. Data and methodology

### 3.1. Target variables

The National Association of Securities Dealers Automated Quotations, known as NASDAQ, is a popular repository of company-specific data. Their website[1] provides data related to IPO performances and all filings made by the respective companies. The IPO performances, spanning the first day of trading, thirty days of trading, and six months of trading, as provided on the website, serve as the targeted metrics for modeling the short-term, medium-term, and long-term performances of the IPOs, respectively. In NASDAQ's dataset, IPO performance is quantified as the percentage change in the stock price from the IPO price to the close of a specific period, be it the first trading day, 30 days, or six months since the IPO. To illustrate, if a company's IPO price was $10 per share and closed at $15 on its first trading day, the IPO performance would be calculated as 50 %. To examine the proposed architecture, this study utilizes these IPO performance metrics for the 15 stocks over the three specified time periods, offering a comprehensive perspective on the stocks' performances in the early stages and over an extended duration (see Fig. 1). While the histogram of IPO performances, unfortunately not depicted in this paper, reveals a balanced distribution for the first trading day, it exhibits a noticeable skew over more extended periods, indicating substantial increases or decreases in performance. Within individual data points, a diverse range is observed, with some companies significantly outperforming or underperforming their IPOs in the long run.

Notably, companies like "ACRV"[2] and "AESI"[3] demonstrate performances moving in opposite directions, highlighting the substantial variability in outcomes. Conversely, stocks such as "BLACU"[4] and "HSPOU"[5] exhibit minimal change in performance. It is important to note that while the primary objective of this study is not to achieve greater predictive accuracy but rather to create meaningful features for improved interpretability, data points were subjectively selected to ensure diversity and minimize bias in performance. Given the smaller dataset size, the target variable has been classified into five classes based on performance metrics (see Fig. 2): Strong Positive ($>$ 50 %), Positive (10–50 %), Neutral (-10–10 %), Negative (-50 % to $-$10 %), and Strong Negative ($<$ $-$50 %). This classification allows for a more nuanced analysis within the constraints of the available data, with a primary focus on text regression methodologies.

### 3.2. Predictors

As we delve into various data sources, this discussion begins with two fundamental types accessible to any IPO investor. The first source comprises the filings submitted by companies to the U.S. Securities and Exchange Commission (SEC). The SEC, as the repository for all filings made by public companies, provides comprehensive access to this information on its website.[6] Specifically, the SEC Form 424B4 contains the company's prospectus, offering investors essential information for decision-making. This includes details such as the IPO price, the quantity of shares offered, the concise financial history of the company, significant events, key stakeholders, associated risks, and other pertinent information. In our study, we extract and analyze the information outlined in Table 1 from this prospectus document.

Further, the second type of data crucial for any investor during the decision-making phase involves researching the company. A pivotal initial question in this phase is, "What does Google say about the company?" To address this, the study collects the top 10 search results obtained by googling the company for each designated time frame, with further details discussed in Section 3.3.1. Specifically, the study gathers the title of each search result and a brief snippet from the site displayed just below the title. These elements encapsulate the sentiment and may be considered the "first impression" about the company for investors during the given time frame. While more in-depth research is typically conducted into the industry and key individuals involved in subsequent phases, this study focuses on first-impression data, aiming to engineer features that yield powerful insights. The final layer of data is derived from an analyst's perspective, utilizing Google Trends data.[7] This dataset incorporates a parameter known as "interest over time," gauging the popularity of search queries in Google across various regions. Unlike the preceding predictors, which primarily stem from an individual investor's decision-making standpoint, the interest over time data offers a unique perspective by mapping the volume effect on the target variable. It serves as a link, transitioning from a singular decision-making node to a population's standpoint. In simpler terms, it can be thought of as the scalability variable for individual decision-making. It is essential to note that the interest over time data introduces a time series nature, while the query results pertain to a time frame (though not as time series data). This necessitates a definition of the first day of trading. Each IPO data point includes its prospectus filing date and the first day of trading. The source, NASDAQ, clarifies the exact date of the first trading day for each company, requiring its own definition. In this study, the first day of trading is defined as seven days from the prospectus filing date. Similarly, thirty days of trading span from the filing date, and the same principle applies for the six-month timeframe.

### 3.3. Data collection methodology

Given that one of the primary objectives of this study is to comprehend how to streamline the different information available online, it is necessary to delve into the automation of the data-collection process. It facilitates efficiency in data acquisition and sheds light on essential details about the central variables. In the subsequent sections, we provide details regarding the automated data-collection procedures and offer insights into the intricacies of handling the variables.

#### 3.3.1. Google search queries data

The Custom Search JSON API[8] option provided by Google is instrumental in filtering search results for each specified time frame. The selection of necessary search queries to gather initial impressions is an important aspect of the whole process. While investigating the IPO of a

[1] Website link: https://www.nasdaq.com
[2] ACRV: Acrivon Therapeutics Inc.
[3] AESI: Atlas Energy Solutions Inc.
[4] BLACU: Bellevue Life Sciences Acquisition Corp.
[5] HSPOU: Horizon Space Acquisition Corp.

[6] Website Link: https://www.sec.gov/edgar/searchedgar/companysearch
[7] Website Link: https://trends.google.com/trends/
[8] Website link https://developers.google.com/custom-search/v1/overview

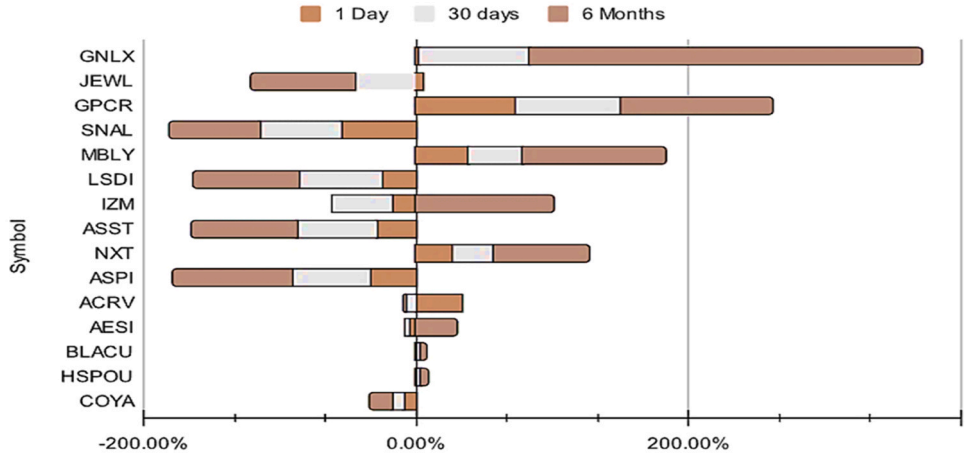


**Fig. 1.** IPO Performances over the three time periods.

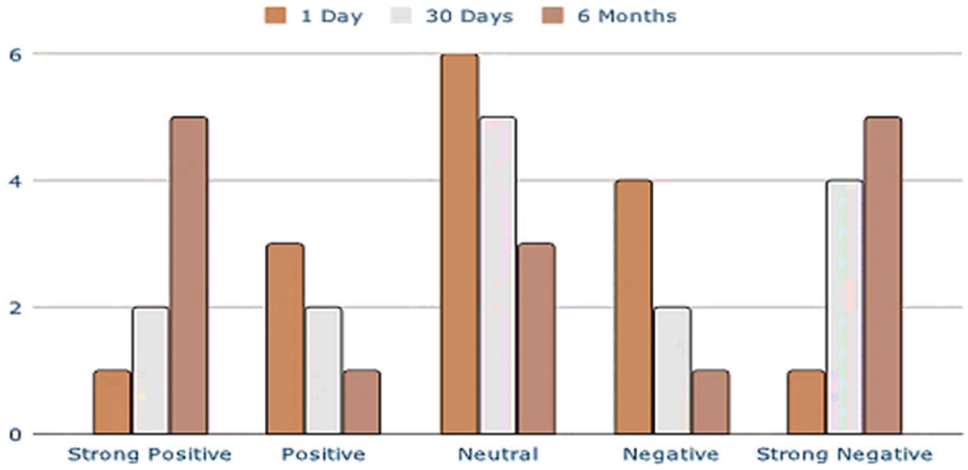


**Fig. 2.** Number of companies falling under each class as per their performance.

**Table 1**
List of predictors per data point obtained from each prospectus.

| Predictor | Description |
|---|---|
| Prospectus filed | Date of prospectus filing |
| Num. of shares | The number of shares being offered to the public |
| Offering price | The price of the IPO per share |
| NPL 2020 | The Net Profit/Loss for the financial year ending 2020 |
| NPL 2021 | The Net Profit/Loss for the financial year ending 2021 |
| NPL 2022 | The Net Profit/Loss for the financial year ending 2022 |
| Prospectus Summary | The prospectus summary section of the prospectus itself, which contains the overview of the company, summarizes the financial information, risk involved, significant events, key people, and any vital information. |

company like JEWL,[9] potential search queries could be: *"Adamas One Corp", "JEWL", IPO of adamas one corp," "Review of adamas one crop jewl ipo," "Ratings jwel ipo stock," "financial history adamas one corp," "What industry is adamas one corp"* etc. To navigate the numerous possibilities and streamline the search process, we use the operator feature in search queries. For instance, when exploring the IPO of Adamas One Corp, we could form a search query like "Adamas One Corp + JEWL". These search results should be filtered within the necessary time frame. While this approach may not be revolutionary, it effectively serves the intended purpose of extracting initial impressions from online searches. The iterative process involves applying this methodology to each company, enabling the retrieval of titles and corresponding snippets to encapsulate the first impressions obtained through Google searches.

### 3.3.2. Google Trends data

While it is possible to manually download trends data, the present study opted for a more efficient approach using the Python module "pytrends." [10] This decision stems from the need to collect approximately 45 sets of trends data (15 companies across three time periods). Employing an iterative process similar to the previous steps, the study iterates over each company and employs "pytrends" to fetch the interest over time data. This data assumes a time series structure, wherein the first/short period consists of seven steps (as defined in Section 3.2), the second/medium term comprises 30 steps, and the long-term period spanning 6 months incorporates around 183/184 steps. Each step carries a corresponding interest over time value, providing a comprehensive representation of the search query's popularity across different timeframes.

[9] Adamas One Corp.

[10] Authored by John Hogue and Burton DeWilde, pytrends is unofficial for Google trends available on open-source: https://pypi.org/project/pytrends/

**Table 2**
List of additional features obtained from the IOT data.

| Feature | Description | Interpretation |
|---|---|---|
| Event count | Represent the number of data points that are not zero that is, the number of significant spikes in the data (significant because, by nature, IOT is zero if the volume is not significant) | This offers insights into the intensity of public interest. A higher event count may explain a significant change in performance (whether positive or negative) |
| Second peak interest | The IOT value of the second most significant spike | By its nature, the maximum peak is 100; this differs from what we are interested in as it gives little meaning, except that it was popular at a point. The second peak of 60 may tell us that the interest has decreased by 40 units, but a value of 20 indicates a disinterested public. In a way, this helps us identify whether multiple spikes in search queries occur. |
| Time to the first peak | The duration from the beginning of the search period when the first peak (100) occurs | This tells the public's response time from the filing date; a stock might get popular after three months, maybe after some time. That might explain a sudden shift in the IPO performance. |
| Time to second peak | The duration between the first peak and the second peak occurs | This tells us whether there is a recurring interest in the IPO or if it is rare |
| Recent Interest | Reflects the popularity of the IPO term in the most recent data point | This tells the current level of public interest, capturing the latest sentiment |

### 3.4. Feature engineering

In this section, the study endeavours to engineer a multitude of features from the collected predictors. This pivotal step involves transforming raw data into meaningful predictors that contribute to the overall analysis. The study employs various techniques and experiences eureka moments throughout this process. It begins by delving into the utilization of the interest over time data, as outlined below:

#### 3.4.1. Interest Over Time (IOT) data from Google Trends

Normalization techniques are not required for this dataset, as Google Trends inherently normalizes search data. Google Trends focuses on relative popularity rather than absolute volume, simplifying the analysis by highlighting the "popularity" of an IPO that influences stock prices. The study is particularly interested in discerning significant shifts in search trends rather than the precise number of individuals searching for a specific stock. To achieve this, each data point is computed by dividing the number of searches for the term by the total number of searches on Google at that specific time and location, with a global perspective in our case.[11] Subsequently, these values are scaled on a range from 0 to 100, reflecting a topic's proportion to all searches on all topics. A score of 100 signifies high popularity, while 0 indicates very low volume or, at times, none. In essence, these data points signify statistical significance. The study proceeds to define additional features derived from this time series data, representing just a subset of the features obtained from the IOT data (Table 2).

#### 3.4.2. Google search results

The textual data is organized as a list of objects per data point, with each object encompassing the title and snippet texts of the first ten query results displayed by Google during the specified time frame. One classic feature engineering technique applicable to text data involves obtaining the sentiment score for each result (Table 3).

**Table 3**
List of additional features drawn from the search results data.

| Feature | Description | Interpretation |
|---|---|---|
| Number of positive search results | Count of positive sentiment scores among the sentiment scores from search queries | A higher count of positive sentiment scores might indicate a more positive sentiment toward the company. |
| Number of negative search results | Count of negative sentiment scores among the sentiment scores from search queries | A higher count of negative sentiment scores indicates a more negative sentiment toward the company. |
| Mean positive scores | The average of positive score values | Higher positive scores, on average, indicate a favourable atmosphere for the company. |
| Mean negative scores | The average of positive score values | Higher negative scores, on average, indicate an unfavourable atmosphere for the company. |

The study explores features drawn from the sentiment scores data of the search results, offering a robust set of features to work with. Moving forward, the study delves into a more comprehensive understanding of each query result, extending beyond sentiment analysis. Leveraging the Bidirectional Encoder Representations from Transformers (BERT), a pre-trained model, the study generates contextualized embeddings from the textual data found in both the titles and snippets. These BERT embeddings provide a nuanced comprehension of the textual content and are subsequently integrated with the non-textual features. This integration enables the creation of a holistic representation of IPO-related data. The BERT representations, coupled with sentiment scores, act as a bridge between raw text and the predictive model, elevating the quality of features for subsequent layers in the model architecture. One notable advantage of incorporating BERT embeddings lies in their interpretability. The study gains transparency into the influence of specific textual elements on predictions by tracing them back to the BERT embeddings. This interpretability element enhances the ability to draw meaningful inferences regarding the impact of textual content on IPO outcomes.

#### 3.4.3. The text data from the "Prospectus Summary" section in the prospectus

Similarly, the textual data from the IOT data is processed using the BERT model, following the approach detailed in Section 4.

#### 3.4.4. Numerical predictors (from Table 1)

The study incorporates the number of shares, IPO price, and net profit/loss for the years 2020, 2021, and 2022 as numerical predictors (Table 4). However, it is worth noting that some companies may lack profit/loss information for specific years, with these values designated as "none." Given the significance of profit/loss information in influencing investor decision-making, the study extracts three features from this scenario. It defines three Boolean variables indicating whether profit/loss data is available for the years 2020, 2021, and 2022. To address missing values, the study fills them with zeros, ensuring a comprehensive representation of this financial information.

The inclusion of binary features enables the model to consider the impact of financial indicators when the corresponding data is available,

**Table 4**
List of binary features obtained from profit/loss availability.

| Feature | Description |
|---|---|
| Profit/Loss Availability 2020 | 1 if available, 0 if otherwise |
| Profit/Loss Availability 2021 | 1 if available, 0 if otherwise |
| Profit/Loss Availability 2022 | 1 if available, 0 if otherwise |

[11] A brief on the normalization process is explained here: https://support.google.com/trends/answer/4365533?hl=en#:~:text=Google%20Trends%20normalizes%20se

**Table 5**
List of final variables and their type.

| Variable type | Variables |
|---|---|
| Text | Prospectus summary, Search results from Google |
| Time series | Interest over time data from Google trends |
| Numerical (including Boolean) | Number of shares, Offering price, NPL 2020, NPL 2021, NPL 2022, one_day_scores_count_positiv, one_day_scores_count_negative, one_day_scores_mean_negative, one_day_scores_mean_positive[a], event_count, second_peak_interest, time_to_first_peak, time_to_second_peak, recent_interest |

[a] Thirty- and Six-month scores are computed for each model accordingly

adding valuable insights to the feature set for enhanced predictive accuracy. Subsequently, the study processes other numerical data using Min-Max scaling. This transformation standardizes the values to a range between 0 and 1, preventing any single feature from exerting disproportionate influence on the predictive model. The application of Min-Max scaling ensures a balanced contribution from all numerical predictors, contributing to a more equitable dataset despite the inherent constraints of a small sample size.

### 3.5. The model architecture

An ensemble learning algorithm proves to be a valuable tool, particularly when confronted with multiple data types and the challenge of interpreting features. As of now, the present study encompasses three distinct data types (Table 5):

Considering the predictors, the study formulates four model architectures contingent on the type of target variable. As outlined in Section 3.1, the target variable is treated both as a continuous target and a classified target. The study proceeds to explore both the regression problem and the classification problem based on the nature of the target variable.

The data pipeline developed for this study is designed to efficiently handle and integrate diverse data types, including textual, numerical, and time series data, to predict IPO performance (See Fig. 3). The pipeline begins with the **Data Collection** phase, where data is sourced from IPO prospectuses, Google search snippets, and Google Trends (volume of interest). Following data collection, **Temporal Segmentation** is applied, particularly to the Google Trends and search snippet data, which is divided into three distinct time frames: 1 Day, 30 Days, and 6 Months post-IPO. This segmentation allows the model to capture temporal dynamics in public interest and sentiment.

In the **Feature Engineering** stage, relevant features are extracted and processed from the collected data. Textual data is embedded using BERT to capture contextual information, while numerical and categorical data undergo standard preprocessing techniques such as normalization. **Data Merging** then combines these engineered features into a single dataset for each time frame, using a unique identifier (e.g., the IPO symbol) to ensure accurate alignment of features across different data sources.

The pipeline then branches into two distinct modeling paths. The first path leverages the **Multimodal Toolkit** and employs a **Gating Mechanism** that dynamically integrates numerical, categorical, and textual features. This combined representation is fed into a fully connected neural network in the **Final Prediction Layer** to generate predictions. The second path applies an **Ensemble Modeling** approach, where individual models—BERT for textual data, LSTM for time-series data, and XGBoost for numerical data—make independent predictions. These predictions are then combined using an ensemble method, such as voting or averaging, to form the final prediction. Both branches of the pipeline undergo rigorous interpretability analysis using SHAP values, which provide insights into the contributions of various features and allow for reverse-engineering of predictions to better understand the model's decision-making process. This comprehensive and flexible pipeline structure enables the integration of multiple data types and modeling approaches, ensuring robust and interpretable predictions of IPO performance.

#### 3.5.1. Transformer architecture for text data

Addressing a critical challenge in our curated dataset involves the amalgamation of unstructured text and structured tabular data. Despite the impressive capabilities of transformers and BERT in natural language processing, they exhibit limitations in handling multimodal data, thereby complicating the model architecture. These models are predominantly text-centric and lack inherent mechanisms for seamlessly incorporating preprocessing, feature engineering, and model integration in a typical pipeline. To overcome this challenge [35], proposed an elegant solution with their open-source Multimodal toolkit package. This toolkit integrates multimodal data on top of text data for classification and regression problems. Leveraging hugging face transformers as the base text model, they introduce a combining module that incorporates the outputs of the transformer along with categorical and numerical features, generating rich multimodal features for downstream tasks such as regression or classification. The package introduces various methods for effectively combining heterogeneous data points, referred to as "Combine Feature Methods." [12]

Building on this innovative approach, the study explores two model architectures—one with only textual data and another with both textual and numerical data. The "text_only" architecture utilizes solely the text columns before the final classifier layer. On the other hand, the "gating_on_cat_and_num_feats_then_sum" architecture performs a gated summation of outputs from transformers, numerical, and categorical features before reaching the final output layer, as proposed by [36]. Fig. 4 illustrates a simplified version of the model architecture developed using the multi-modal kit.

Based on this architecture, four models are derived depending on the input provided: Classifier with all features (text, numerical, categorical)

a. Classifier with just text-based features (omitting the gating mechanism)
b. Regression model with all features
c. Regression model with just text-based features

For regression tasks, the Mean Squared Error (MSE) is employed to compute the loss function. This involves comparing the logits with the true values. Meanwhile, classification tasks utilize a cross-entropy loss function.

#### 3.5.2. BERT model architecture for textual data

In this study, we employed the Bidirectional Encoder Representations from Transformers (BERT) model, specifically the bert-base-uncased variant, to process and analyze textual data extracted from IPO prospectuses and Google search results. BERT was selected for its superior capability to capture the contextual meaning of words within text, making it highly suitable for our application.

*Text Tokenization and Embedding:*

- Tokenizer Setup: We utilized Hugging Face's BertTokenizer, designed for the bert-base-uncased model. The tokenizer converts text into a sequence of tokens that BERT can process.
- Prospectus Summary: The "Prospectus Summary" section of each document was tokenized, with text sequences truncated to 512 tokens for computational efficiency, and shorter sequences padded. The tokenized text was then converted into PyTorch tensors for input into the model.

[12] See "Included Methods" section at -https://github.com/georgian-io/Multimodal-Toolkit#included-methods

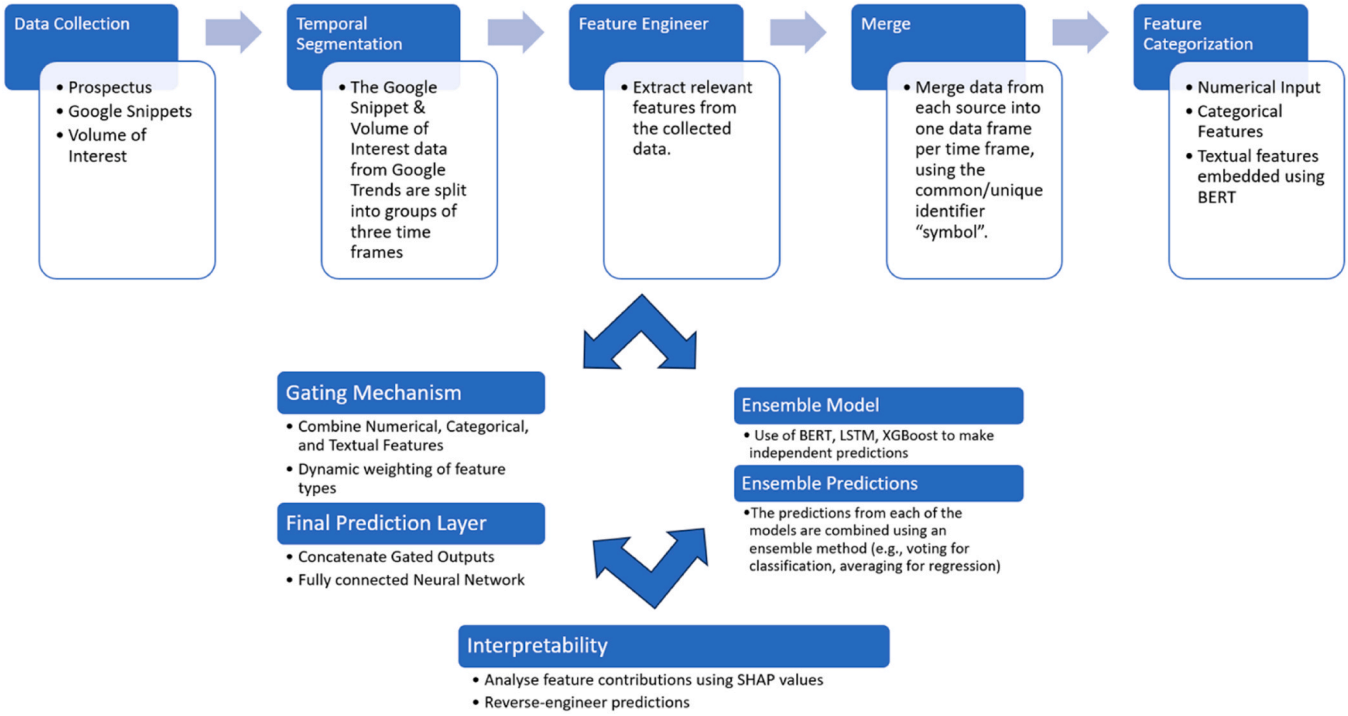


**Fig. 3.** Data pipeline overview.

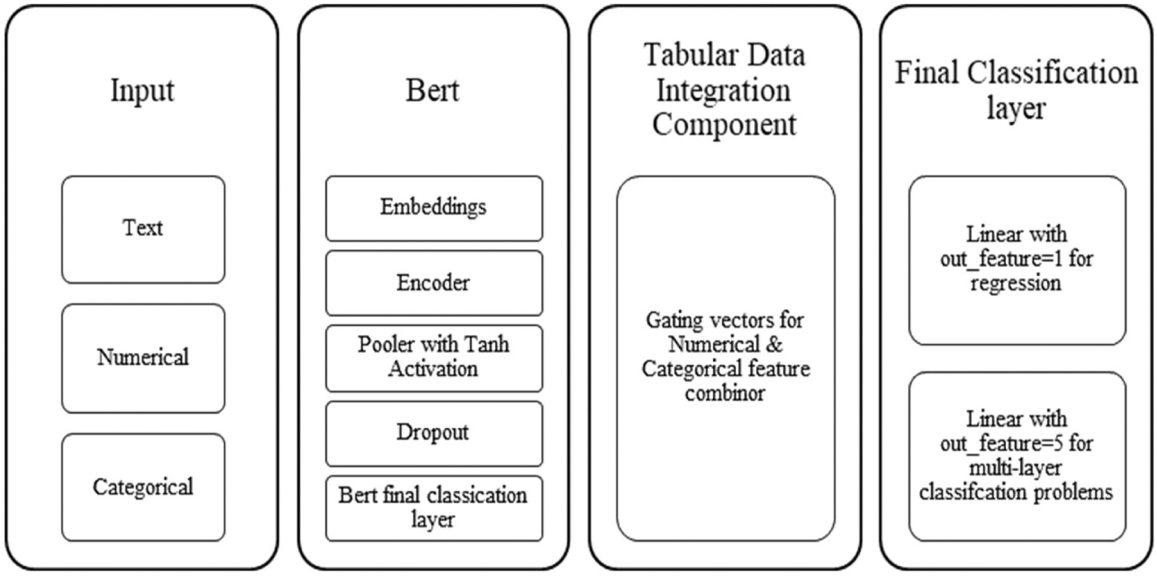


**Fig. 4.** Transformer architecture.

- Search Query Data: Titles and snippets from the one-day search data were concatenated into single strings before tokenization. This allowed us to capture the initial impressions conveyed by search results. Dynamic padding was applied to optimize processing during batch training.

*Model Configuration and Hyperparameters:*

- Model: The bert-base-uncased variant was employed for text embedding.
- Max Sequence Length: A maximum sequence length of 512 tokens was set, balancing the need to capture full context with computational efficiency.
- Batch Size: Depending on memory constraints, batch sizes were set to either 16 or 32.
- Learning Rate: A learning rate of 2e-5 was used for fine-tuning BERT.
- Training Epochs: Training was conducted over three epochs, with early stopping based on validation loss to prevent overfitting.
- Optimizer: The AdamW optimizer, with weight decay, was utilized to enhance model performance.
- Dropout Rate: A dropout rate of 0.1 was applied during fine-tuning to mitigate overfitting.

BERT's integration into our multimodal model allowed us to combine the textual embeddings with numerical features, facilitating a richer representation of the IPO-related data. This integration was

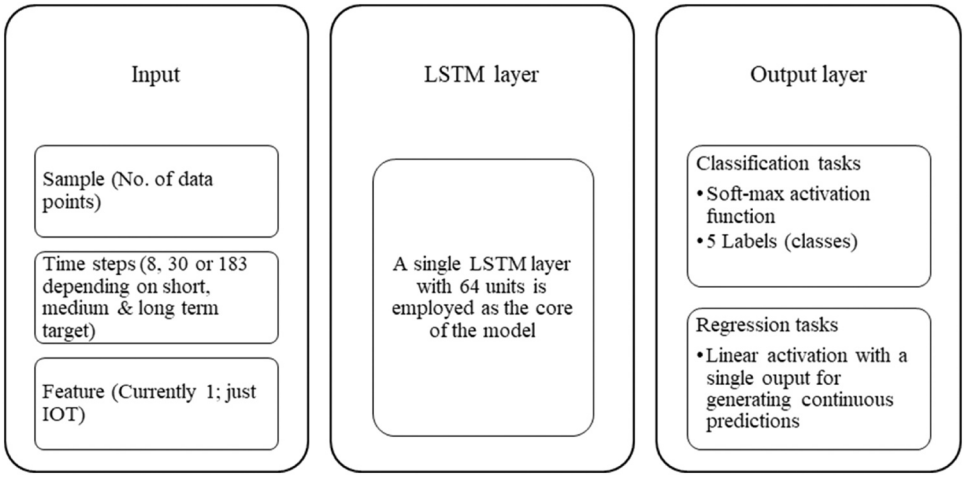


**Fig. 5.** LSTM architecture for fitting time series features.

**Table 6**
Performance metrics from the classifier models.

| | Classifier with all features (short term) | Classifier with all features (medium term) | Classifier with all features (long term) | Classifier with text only (Short term) | Classifier with text only (medium term) | Classifier with text only (Long term) |
|---|---|---|---|---|---|---|
| **Accuracy** | 0.40 | 0.47 | 0.20 | 0.13 | 0.40 | 0.67 |
| **Precision** | 0.16 | 0.27 | 0.04 | 0.14 | 0.24 | 0.45 |
| **Recall** | 0.40 | 0.47 | 0.20 | 0.13 | 0.40 | 0.67 |
| **F1 Score** | 0.23 | 0.34 | 0.07 | 0.10 | 0.30 | 0.53 |
| **Confusion Matrix** | [[0, 0, 1, 0, 0], [0, 0, 4, 0, 0], [0, 0, 6, 0, 0], [0, 0, 3, 0, 0], [0, 0, 1, 0, 0]] | [[2, 0, 2, 0, 0], [1, 0, 1, 0, 0], [0, 0, 5, 0, 0], [2, 0, 0, 0, 0], [0, 0, 2, 0, 0]] | [[0, 0, 5, 0, 0], [0, 0, 1, 0, 0], [0, 0, 3, 0, 0], [0, 0, 1, 0, 0], [0, 0, 5, 0, 0]] | [[1, 0, 0, 0, 0], [4, 0, 0, 0, 0], [5, 0, 1, 0, 0], [2, 0, 1, 0, 0], [0, 0, 1, 0, 0]] | [[3, 0, 1, 0, 0], [2, 0, 0, 0, 0], [2, 0, 3, 0, 0], [1, 0, 1, 0, 0], [0, 0, 2, 0, 0]] | [[5, 0, 0, 0, 0], [1, 0, 0, 0, 0], [1, 0, 0, 0, 2], [1, 0, 0, 0, 0], [0, 0, 0, 0, 5]] |
| **Count of False predictions** | 9 | 8 | 11 | 13 | 9 | 5 |
| **General interpretation** | This model performs moderately well in terms of accuracy and recall but has poor precision. The F1 score is also low, indicating suboptimal performance in both precision and recall. The confusion matrix reveals a significant number of false positives and false negatives, especially in class 2. | This model shows improvement compared to the previous one, with higher accuracy, precision, recall, and F1 score. However, there are still misclassifications, particularly in classes 0, 2, and 4, as evident from the confusion matrix. | This model performs poorly with very low precision, recall, and F1 score. It fails to effectively classify any class, as indicated by the confusion matrix, with numerous misclassifications in all classes. | This model also performs poorly with very low precision, recall, and F1 score. It struggles to correctly classify any class, as shown by the confusion matrix, with numerous misclassifications in all classes. | This model performs moderately well, with accuracy, precision, recall, and F1 score around 40 %. However, it still has room for improvement, as shown by the confusion matrix, which indicates misclassifications across different classes. | This model performs relatively well, with accuracy, precision, recall, and F1 score higher than other models. It shows good capability in predicting classes, especially for long-term predictions. The confusion matrix indicates fewer misclassifications compared to other models, with most predictions concentrated on the diagonal, indicating correct classifications. |

critical for improving the overall prediction accuracy and interpretability of our model.

### *3.5.3. LSTM architecture for handling time series data*

Long Short-Term Memory (LSTM) is a remarkable architecture for Recurrent Neural Networks capable of handling sequential data by allowing information to persist over time [37]. They can capture long-term patterns and dependencies, making them ideal for predicting time series data. The choice of LSTM is owed to the non-linear fashion in the interest over time data from Google trends. It can be easily configured for classification and regression tasks, thus perfectly fitting out analysis (See Fig. 5) [38]. demonstrate the effectiveness of LSTM over the traditional Auto Regressive Moving Averages model in forecasting the number of cases using Google trends [39]. shows outperformance of the LSTM model over traditional approaches in machine learning in forecasting international migration using Google Trends data. Moreover, this model can be concatenated with other sets of time series features if we wish to integrate the stock price data in measuring the medium and long-term performances[13] [40].

Model Configuration

[13] "Thirty day" IPO Performance is referred as Medium term and "Six month" IPO Performance is referred as Long term (see Section 2.1)

- Input Shape: The LSTM was set up to handle sequences of numerical data, with each sequence reflecting a specific period's data (e.g., daily trends).
- LSTM Layer: We used a single LSTM layer with 64 units. This layer was chosen for its ability to capture patterns without making the model overly complex.
- Output Layer:

a. For classification tasks, we used a Dense layer with softmax activation, outputting probabilities for each class. The number of output units matched the number of classes (e.g., 5 for IPO performance classification).
b. For regression tasks, a single neuron with a linear activation function was used to predict continuous values.

Hyperparameter Tuning

- Loss Function:

a. We used categorical_crossentropy for classification.
b. For regression, mean squared error (MSE) was the choice.

- Optimizer: We selected the Adam optimizer for its efficiency in adapting the learning rate during training.
- Batch Size and Epochs:

a. Batch size was set to 32 to manage memory and generalize better.
b. We started with 1 epoch to prevent overfitting, with the possibility to adjust based on validation performance.

- Regularization: Although dropout wasn't explicitly added, the model's simplicity helped prevent overfitting.

The LSTM was part of an ensemble with XGBoost and BERT-based models. This setup allowed us to combine the LSTM's temporal insights with the strengths of the other models, improving overall accuracy.

#### 3.5.4. Extreme gradient boosting for numerical features

In this study, we utilized the Extreme Gradient Boosting (XGBoost) algorithm to model the numerical features associated with IPO performance. XGBoost is a powerful machine learning algorithm known for its efficiency and effectiveness, particularly in handling small to medium-sized datasets, making it an ideal choice given the constraints of our sample size.

Model Configuration:

- Numerical Features: The numerical features considered included the number of shares, offering price, net profit/loss for the years 2020, 2021, and 2022, as well as sentiment scores derived from Google search results. Additionally, time series features such as event count, second peak interest, time to first peak, time to second peak, and recent interest were included.
- Handling Small Datasets: XGBoost was selected for its ability to deliver robust performance even with limited data points, which was a critical consideration given our dataset size of only 15 IPOs.
- Feature Importance: One of the key advantages of using XGBoost is its ability to provide clear insights into feature importance, which is particularly valuable for interpreting the model's predictions.

Hyperparameter Tuning:

- Learning Rate: A lower learning rate was chosen to allow the model to learn gradually and avoid overfitting, especially important with a small dataset.
- Maximum Depth: The maximum depth of the trees was carefully adjusted to prevent the model from becoming too complex and overfitting the data.
- Subsample: A subsample ratio of the training data was used for each iteration to introduce variability and enhance generalization.
- Regularization: L2 regularization (Ridge regression) was applied to reduce the risk of overfitting by penalizing large coefficients.

One may argue that a major limitation in our experimental setup is the restricted sample size (15 data points). This study can possibly engineer various features for each data point. However, this is mainly constrained by the limited sample size due to computational and time constraints associated with integrating data from diverse sources. To address this constraint, we opt to leverage the Extreme Gradient Boosting (XGBoost) algorithm for fitting numerical features. This choice is primarily motivated by XGBoost's efficacy in handling small datasets [41]. Another advantage of adopting a conventional machine learning approach, in contrast to the deep learning/transformer architectures used in previous sub-sections, is the ease of tracking feature interpretability and importance. XGBoost is applied to fit both classification and regression models, aligning with the methodology employed in earlier cases.

#### 3.5.5. Ensemble of models

In the context of ensemble predictions, while various ensemble methods could be considered, the study acknowledges the constraint posed by the limited sample size. Therefore, a straightforward averaging approach has been employed to derive the final prediction, incorporating individual predictions from all the models. For classification models, a simple voting mechanism is deemed sufficient for the ensemble process.

## 4. Results and discussions

Upon scrutinizing the data histogram over a condensed timeframe, specifically one day, a discernible pattern of balanced distribution manifests. However, with the extension of the timeframe, the distribution skews, signalling the susceptibility of the stock to pronounced volatilities over prolonged periods. This observation underscores the intricate challenge inherent in modelling extreme fluctuations in stock prices, thereby raising doubts regarding the viability of tracking a singular IPO over time and constructing a resilient model, particularly given the discrete nature of the inputs employed. Additionally, when examining the same IPO across various timeframes, the distribution consistently exhibits skewness, indicative of the anticipation of heightened premiums, denoting extreme returns, aligned with the associated risks.

### 4.1. Initiating model debugging

Our study encompasses both classification and regression models; however, our primary focus for debugging purposes centres on the examination of classification models. This decision stems from the greater familiarity with classification methodologies compared to the relatively novel concept of text regression, which was only explored to satisfy intellectual curiosity. Our initial approach involves analyzing the probability distribution of class predictions generated by the models when applied to the training dataset. While predicting on the training data itself may seem unconventional, it serves a crucial purpose in the debugging process. The first step in model debugging entails assessing whether the features encode predictive signals and establishing a baseline or heuristic for predicting the labels. Backtracking model predictions across various data points can be cumbersome. Thus, adopting a strategy of starting small, by hand-selecting data points of varying intensities, allows for a supervised debugging approach to evaluate the predictive capabilities and weights of each feature on the final

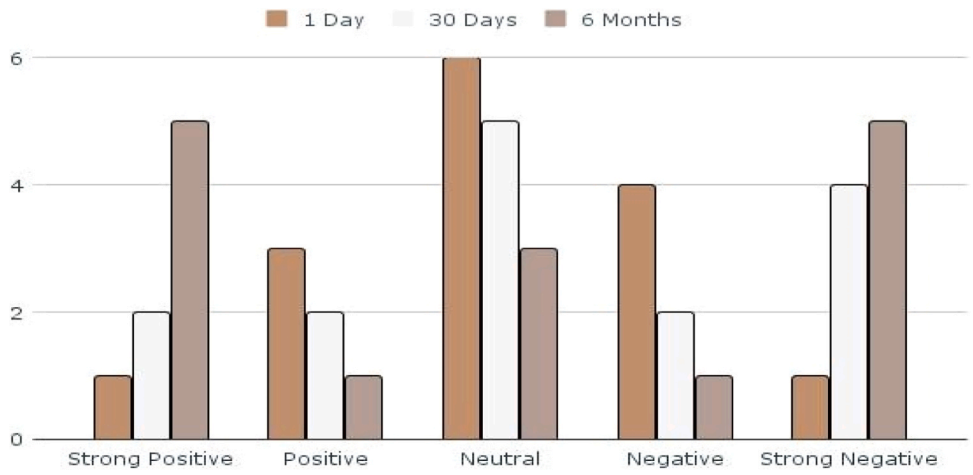


**Fig. 6.** Distribution of classes within our sample.

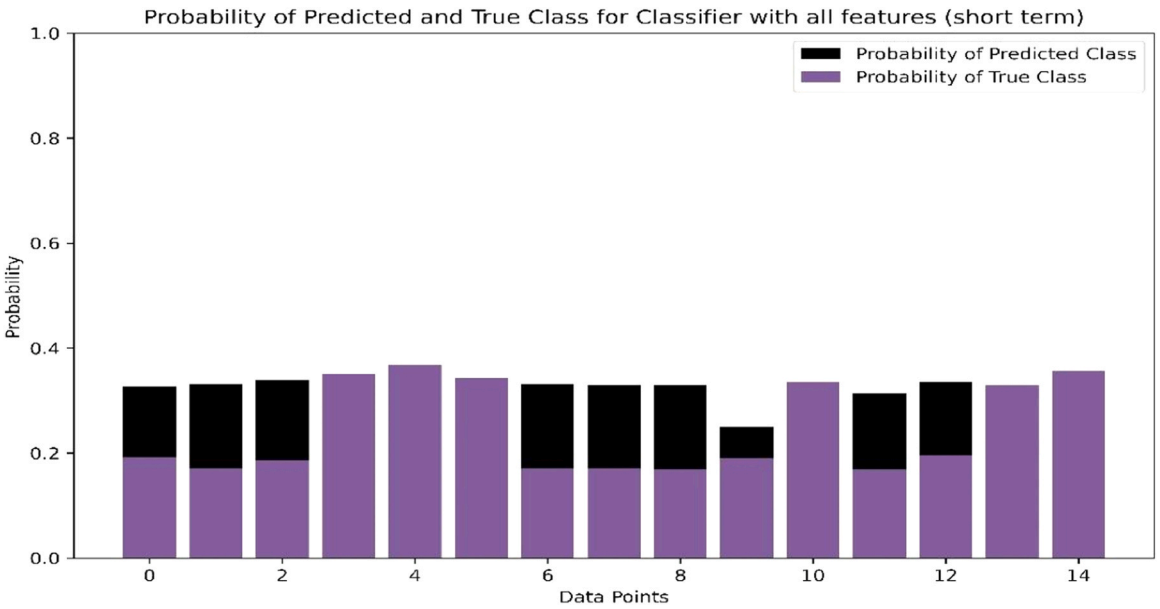


**Fig. 7.** Probability of predicted and true class for classifier with all features (Short term).

prediction. Moreover, testing a poorly performing model on new data may yield predictable outcomes, rendering such endeavours futile.

We can conceptualize deep learning models as compositions of successive (non-linear) functions, wherein each layer computes increasingly complex features. Consequently, we possess the capability to monitor these emergent features. This analogy is particularly relevant for deep convolutional neural network models, wherein the features computed by each layer can be visualized. Like traditional software debugging, where breakpoints enable the monitoring of intermediate variable values, we can employ a comparable approach to monitor the intermediate outputs of our deep learning model. Therefore, our exploration will entail scrutinizing the models on their training data itself to glean insights that can inform subsequent analysis and refinement. By examining the distribution of predictions and the behaviour of features at different stages of the model, we aim to identify and rectify any anomalies or weaknesses in the classification process. This meticulous debugging process is crucial for ensuring the robustness and reliability of our classification models, ultimately enhancing the accuracy and effectiveness of our analyses in comprehensively understanding IPO performance. Table 6 detail the performance metrics of the classifier models used by the present study.

### 4.2. *Introduction to heuristics*

In the domain of supervised algorithms, particularly during the phase of model debugging, the incorporation of human judgment stands as a fundamental heuristic. This method aids in discerning whether the model effectively encapsulates predictive information. If humans encounter difficulty in predicting the data, whether images or text, it suggests that the machine learning model will likely encounter similar challenges. For intricate machine learning architectures, experts advocate for the utilization of high-quality yet concise datasets, favouring them over expansive training data sets. While some may perceive this strategy as counterproductive, a deeper examination of our dataset is warranted. Fig. 6 illustrates the distribution of classes within our sample.

Upon initial scrutiny of Fig. 6, the distribution within the one-day timeframe appears relatively balanced, with a prevalence of stocks categorized as neutral. However, as the timeframe extends and additional information inundates the dataset from diverse online sources, a noticeable skew in distribution becomes evident. This presents a significant hurdle in modelling IPO performance across the designated timeframes, as the distribution deviates from equilibrium.

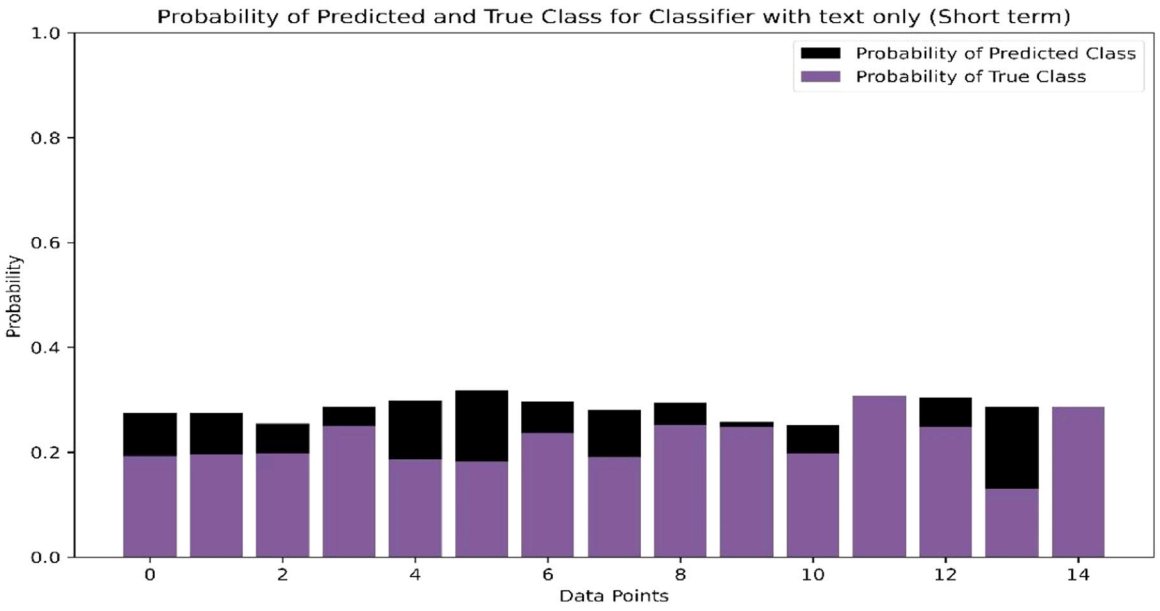


**Fig. 8.** Probability of predicted and true class for classifier with text only (Short term).

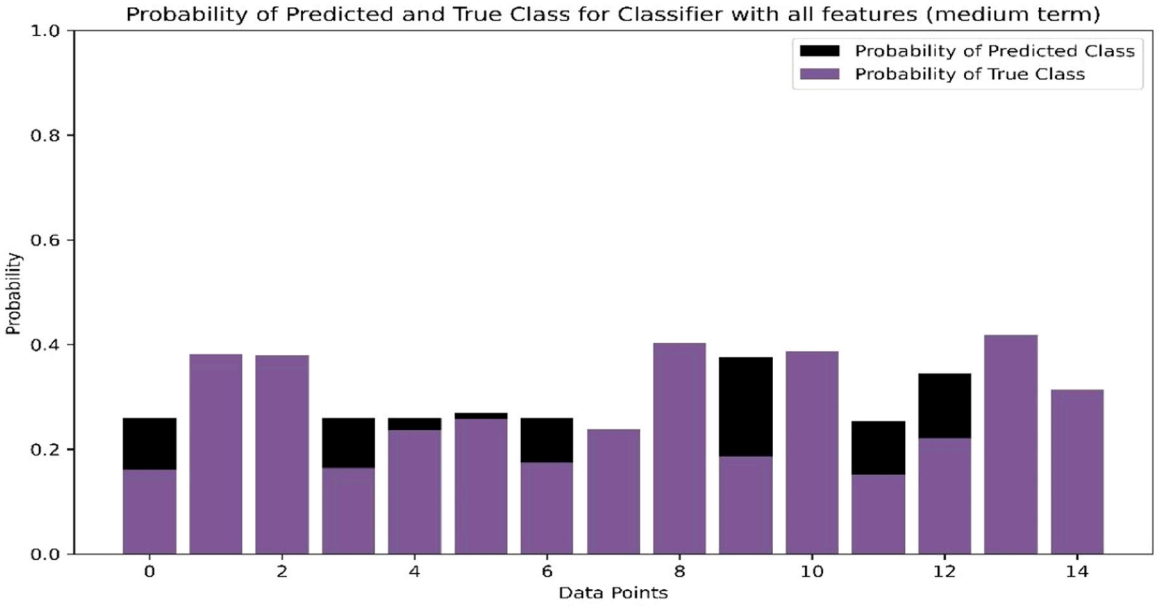


**Fig. 9.** Probability of predicted and true class for classifier with all features (medium term).

Consequently, when constructing such models, the selection of data points reflecting normal or realistic scenarios within the given timeframe becomes imperative. It is imperative to acknowledge that stocks are subject to various risk factors, and the market inherently exhibits volatility for new companies, often taking years for this volatility to subside within a negligible range. Furthermore, our study's maximum timeframe is constrained to six months, during which quarterly earnings reports play a pivotal role. Positive outcomes are likely to result in a surge in IPO value, commonly referred to as a "POP," while adverse results may precipitate a decline. Analyzing the actual distribution of IPO POPs on the first day, as reported by NASDAQ, reveals that approximately 31 % of IPOs experience a surge in value on their debut, with an average return of 18.4 %.

Moving to the confusion matrices of our models, with reference to Table 6, the first model, "Classifier with all features (short term)," demonstrates a tendency to classify all data points into the neutral class, adopting a risk-averse approach akin to human decision-making. Conversely, the contrast model, "Classifier with text only (short term)," exhibits a distribution of predictions across the strong negative and neutral classes. Notably, despite the inclusion of text-only information leading to discernible differences in predictions, the accuracy of this model is notably inferior. It is crucial to recognize the inherent limitations of relying solely on metrics such as accuracy, recall, F1 score, and precision, particularly within the context of a small dataset comprising only 15 observations. To gain a deeper understanding of our features and their predictive signals, we analyzed the probability assigned to predicted and true classes and compared the disparities (see Figs. 7–8).

In our quest for a deeper comprehension of our features and their predictive efficacy, we conducted an analysis of the probabilities assigned to both predicted and true classes, unveiling several notable observations regarding the alignment between the model's predictions

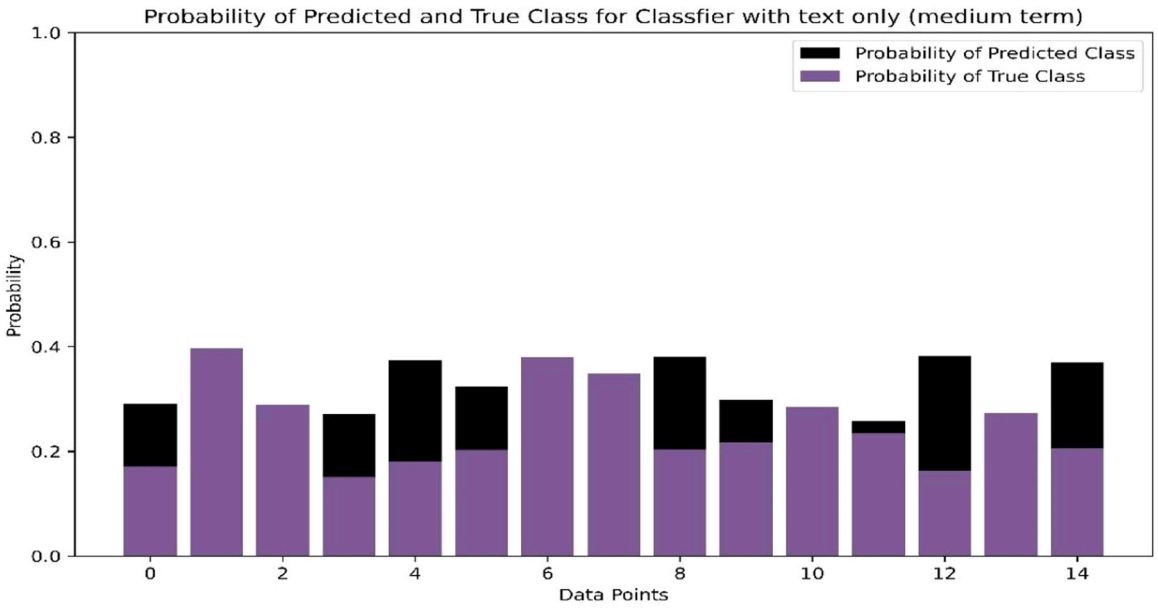


**Fig. 10.** Probability of predicted and true class for classifier with text only (medium term).

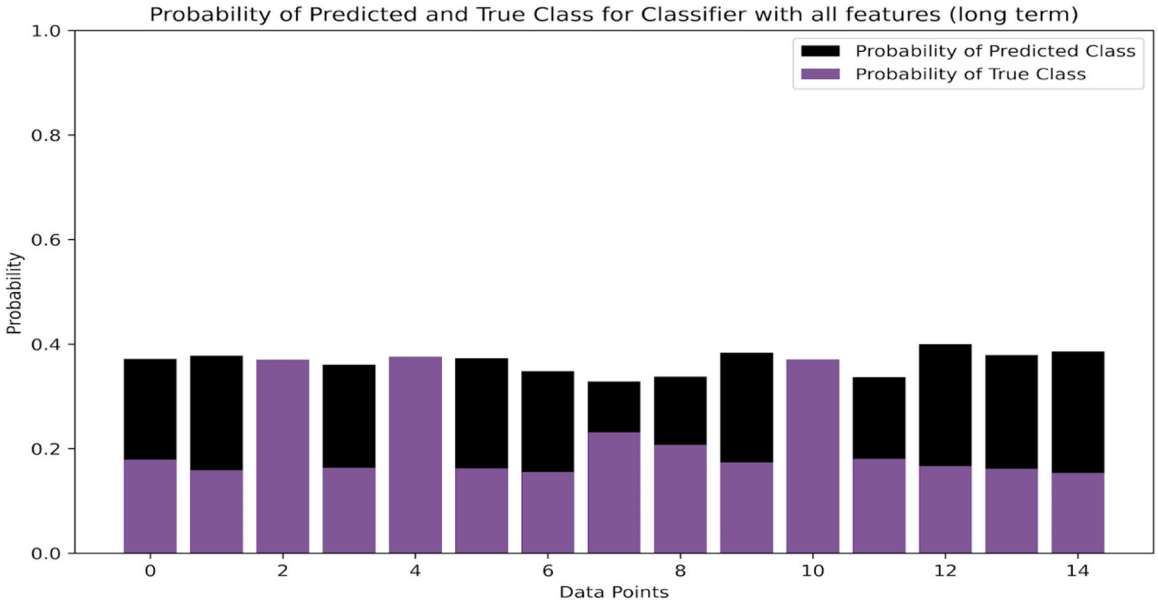


**Fig. 11.** Probability of predicted and true class for classifier with all features (Long term).

and the true class labels. Firstly, the model demonstrates a discernible lack of confidence in its predictions, as evidenced by probabilities consistently falling below the 50 % threshold for both classes. Interestingly, the class with the highest prediction probability is invariably assigned as the predicted class. Moreover, the disparities between the probabilities of the true class and the predicted class are relatively minor, typically ranging between 5 % and 19 %. This consistent pattern implies a degree of concordance between the model's predictions and the true class labels. However, it would be premature to assert that the second model, featuring text-only features, outperforms the first model in the short term based solely on this analysis. Achieving enhanced accuracy by minimizing these prediction disparities remains an overarching objective. While the modest sample size may contribute to these discrepancies, it prompts an exploration into potential inherent characteristics within our predictors that could be leveraged to augment predictive performance.

While our discussion primarily delves into the short-term timeframe, there exists a natural curiosity regarding the outcomes of medium (see Figs. 9–10) and longer-term (see Figs. 11–12) periods. Therefore, we offer distribution graphs for these timeframes. Notably, the classifier demonstrates proficiency in the long run, exhibiting confidence levels surpassing 65 % and accurately predicting a greater number of classes. However, the abrupt surge in confidence raises concerns regarding potential overfitting or memorization of the training data.

Upon scrutinizing the distribution of classes within the long-term sample, it becomes evident that the model tends to favour predicting extreme classes over intermediate ones. For instance, despite the true class being negative, the model assigns a marginally higher probability to the strong negative class, suggestive of a strategic utilization of dataset patterns. Remarkably, the classifier demonstrates proficiency over longer timeframes, showcasing confidence levels exceeding 65 % and achieving a greater accuracy in predicting a wider array of classes.

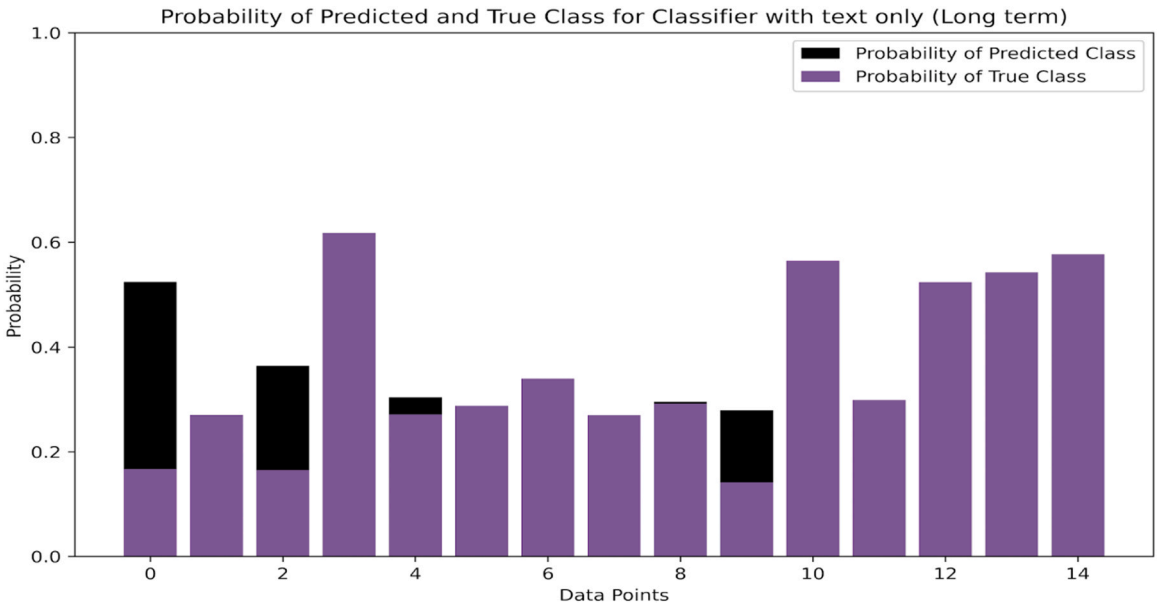


**Fig. 12.** Probability of predicted and true class for classifier with text only (Long term).

**Table 7**
Class distribution within the long-term sample.

| Data point | Predicted Class | True Class | Probability of Predicted Class | Probability of True Class |
|---|---|---|---|---|
| 0 | Strong Negative | Neutral | 0.524528 | 0.166942 |
| 1 | Strong Positive | Strong Positive | 0.270443 | 0.270443 |
| 2 | Strong Negative | Positive | 0.364266 | 0.164968 |
| 3 | Strong Negative | Strong Negative | 0.617861 | 0.617861 |
| 4 | Strong Positive | Neutral | 0.304418 | 0.270982 |
| 5 | Strong Positive | Strong Positive | 0.287952 | 0.287952 |
| 6 | Strong Positive | Strong Positive | 0.339882 | 0.339882 |
| 7 | Strong Positive | Strong Positive | 0.270012 | 0.270012 |
| 8 | Strong Negative | Negative | 0.295697 | 0.291633 |
| 9 | Strong Positive | Neutral | 0.279625 | 0.141558 |
| 10 | Strong Negative | Strong Negative | 0.564715 | 0.564715 |
| 11 | Strong Positive | Strong Positive | 0.299087 | 0.299087 |
| 12 | Strong Negative | Strong Negative | 0.524123 | 0.524123 |
| 13 | Strong Negative | Strong Negative | 0.542752 | 0.542752 |
| 14 | Strong Negative | Strong Negative | 0.577107 | 0.577107 |

However, the abrupt escalation in confidence levels raises legitimate concerns regarding potential overfitting or the model's tendency to memorize the intricacies of the training data.

Upon meticulous examination of the class distribution within the long-term sample, as illustrated in the Table 7, it becomes evident that the model exhibits a predilection for predicting extreme classes over intermediate ones.

Notably, despite the true class indicating negativity, the model tends to assign a marginally higher probability to the strong negative class. This phenomenon suggests a strategic manoeuvre leveraging dataset patterns, potentially aimed at optimizing predictive performance. This selective behaviour prompts curiosity regarding why this specific model exhibits such tendencies while others do not. A plausible explanation can be attributed to the incorporation of transfer learning within the model architecture, particularly leveraging a well-trained language model such as BERT, renowned for its proficiency in contextual comprehension. It is hypothesized that the inclusion of text-only data, combined with a diverse array of information amassed over the six-month timeframe, enhances the model's accuracy and dependability. Armed with these insights, our subsequent undertaking entails a

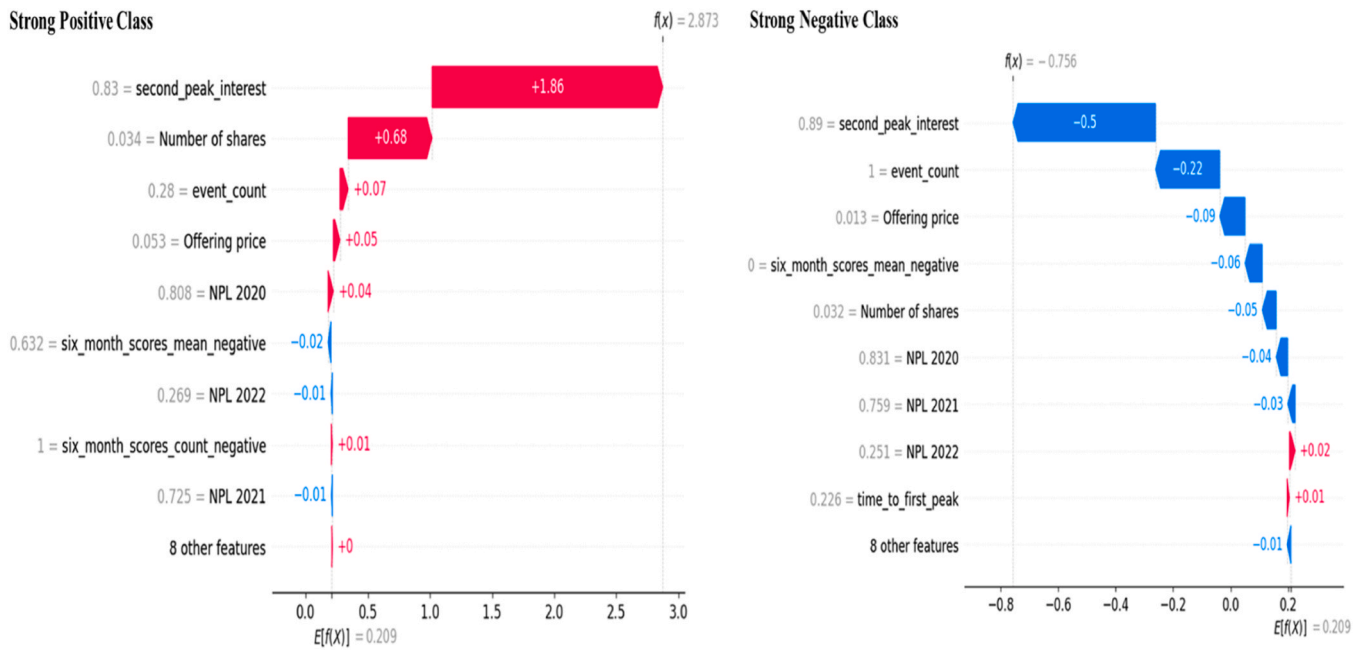


**Fig. 13.** Water Fall plot of SHAP values for data point with strong positive and negative performances over six months.

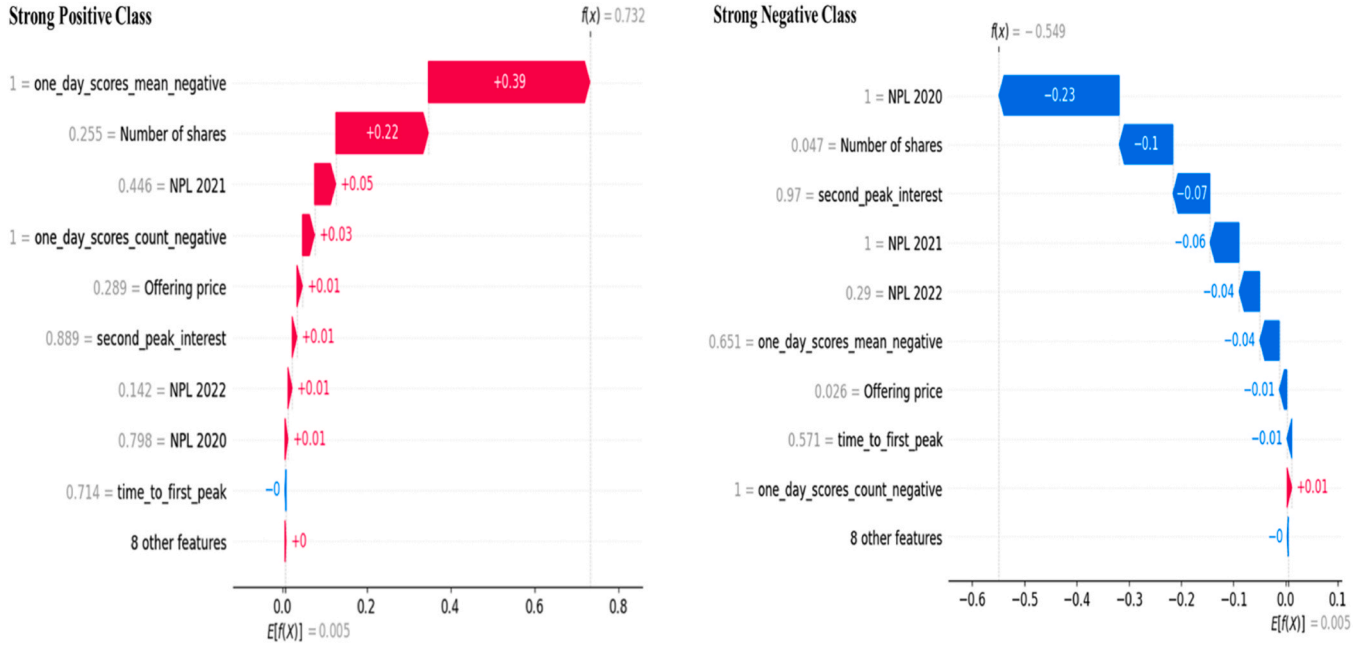


**Fig. 14.** Water Fall plot of SHAP values for data point with strong positive and negative performances over 1 Day.

detailed examination of each feature's contribution to prediction, facilitating a deeper understanding of the rationale guiding the model's decisions.

### 4.3. SHAP values: A tool for model interpretation

A pivotal aspect of model debugging entails the identification of features exerting significant influence on predictions, with SHAP (SHapley Additive exPlanations) values serving as a precise tool for this purpose, facilitating the identification of key variables shaping a model's output. Proposed by [42] and further elaborated upon in subsequent research [43], SHAP values offer a game theoretic approach to explain the output of any machine learning model, allowing computation of each feature's contribution to a specific prediction. As elucidated by Molnar (2020), SHAP values can be conceptualized as a game where feature values enter a room randomly, with each value contributing to the prediction, and the SHAP value of a feature represents the average change in prediction when that feature joins the existing set of features. Leveraging SHAP values enables assessment of the relative contribution of each feature, thereby providing insights into the key factors driving the model's predictions. Analysis of SHAP values across multiple instances facilitates evaluation of model consistency and identification of features with excessive impact, which may introduce bias or compromise prediction reliability. Consequently, SHAP values emerge as a potent instrument for pinpointing influential features within a model's prediction landscape, aiding in model refinement and

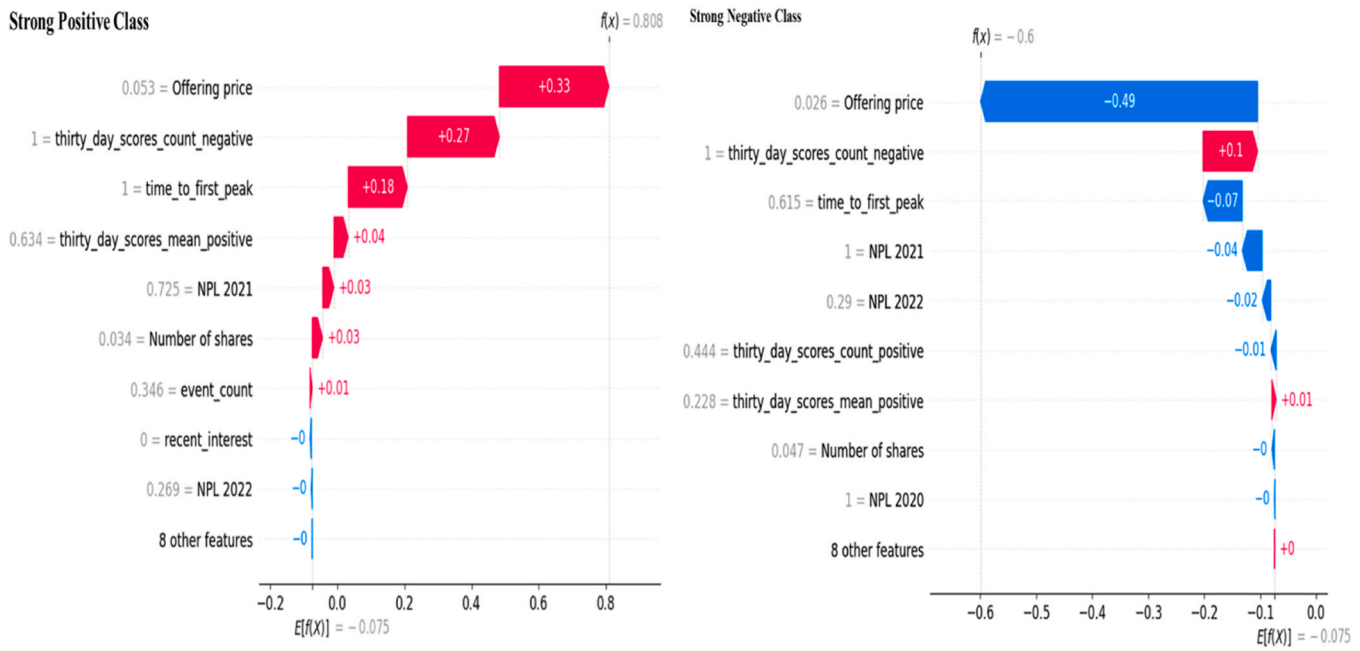


**Fig. 15.** Water Fall plot of SHAP values for data point with strong positive and negative performances over thirty days (medium run).

**Table 8**
Sample interpretations of Figs. 13, 14 and 15.

| | Strong Positive | Strong Negative |
|---|---|---|
| Short term (1 Day) | All features demonstrate a positive contribution from the base value towards driving up the IPO price, suggesting potential underpricing initially. Nevertheless, assessing the corresponding neural class performance is imperative to draw any conclusive inference. | The variable ranking has shifted, with all feature values now moving in the opposite direction compared to the strong positive class's feature values. Lower values are now influencing a decline in performance. |
| Medium term (30 Day) | The variables derived from sentiment scores of query results have risen in the ranks, indicating a heightened amount of available information in the medium term. The increase in the number of negative scores and mean positives is driving up the base value. | The time to first peak is notably shorter compared to the strong positive class, suggesting a delayed popularity of the IPO, which could potentially explain the performance decline. Monitoring this data point across all time periods would provide valuable insights into information dissemination and response dynamics. |
| Long term (6 Month) | Interestingly, the second peak interest is ranked first, suggesting sustained interest in IPOs over a longer timeframe. This is corroborated by the prominence of the variable 'event_count,' indicating continued attention even after the initial peak of interest. | The rankings exhibit notable similarity, with differences primarily attributed to the variables driving down the target. A critical observation is the recent emergence of 'time_to_first_peak,' unlike the strong positive class. Furthermore, in the long run, NPL variables assume significance, indicating a trend of investors seeking anticipatory price surges. Speculative investors seem to be diminishing, with a rise in actual investors displaying genuine interest in the company. |

debugging, while summary and dependence plots serve as effective visualization aids for understanding feature importance. Models may occasionally produce perplexing outputs or unexpected behaviours, underscoring the importance of comprehending their inner workings. With SHAP values, one can delve into each prediction, unravelling the rationale behind the model's decisions and paving the way for model improvement.

In this section, we demonstrate how to explore our model architecture to support our feature engineering objectives, focusing on regression models unlike the previous section, and examining specific instances with extreme categories such as "Strong positive" and "Strong negative." This approach aligns with the objective of demonstrating methods to debug and enhance our model, with the waterfall plot utilized for visualization, offering a clear representation of the impact of evidence provided by each feature on the model's prediction. Figs. 13, 14, and 15 illustrate the SHAP values for two data points characterized by "strong positive" and "strong negative" performances across the three time periods, respectively. These values are arranged by their contribution and presented in a Waterfall graph. In interpreting the Waterfall graph, one starts from the base value located at the bottom of the x-axis, denoted as $E[f(X)]$, or the "base value." This base value signifies the expected prediction when no specific features are considered, obtained by setting all features to default values, such as the mean. Conceptually, it can be envisioned as the value resulting from a linear regression where the predictors are set to their mean values for simplicity. Subsequently, each value corresponding to each feature in the chart indicates how the contribution of that feature influences the movement of the value from the base value. The cumulative sum of these contributions to the base value leads to the final prediction, denoted as $f(x)$, shown at the top of the chart. A detailed explanation is provided in Table 8.

In the case of the time series architecture detailed in Section 3.5.2, a straightforward time series plot of the interest over time data can offer substantial insights. However, for those interested in examining the LSTM's learning process [44], introduce force plots. In this study, the "Kernel Explainer" method is employed to elucidate the LSTM's behaviour using a force plot. Interpreting the plot can be more intricate, where the numbers on the y-axis represent the discrete denotation of each sample in the dataset (15 in total). Each sample is associated with approximately 8 stacks, each corresponding to a time-step feature.

It is not unusual to find conflicting interpretations in Table 7 or Figs. 12–14, where various features may influence predictions in contradictory ways. These discrepancies are central to the core focus of the study, allowing us to analyze and refine our model using heuristic

feedback derived from SHAP values. For instance, certain features are expected to guide the direction of classification, such as 'interest over time' for gauging a specific ticker's popularity, while 'sentiment' and 'offering price' (potentially combined with current trading price) contribute to the magnitude of deviation from expected trends. Although there could be further discussions on this topic, the essence of this research lies in the methodology to extract and utilize this feedback for model debugging. The approaches to debugging such as adjusting feature weighting, removing specific features, or translating data differently during training are subjective and ultimately depend on the model creator's intentions and heuristics for achieving the desired outcome.

### 4.4. Economic implications

The present study findings have several important economic implications as follows. First, precise predictions of IPO performance can potentially increase the chances of detecting the market trends, making more informed decisions, and decrease the associated investment losses. Beyond its immediate financial implications for the investors, IPO performance indicates macroeconomic growth of an economy. The success of IPOs could indicate a robust foundation for ongoing innovation and industrial growth within an economy. These can also be simultaneously reflected in other economic indicators such as output growth. Second, the companies that are planning for successful IPOs can be benefitted from the feature-engineered models to optimize their IPO timing, pricing, and marketing strategies. Lastly, the regulators and policy makers can monitor the IPOs performances and take policy measures to maintain the overall market stability. Our feature-engineered based model could suggest potential fraud in the financial markets. Maintaining transparency, stricter regulation, accountability, and integrity in financial markets could reduce such incidents. Thus, understanding the dynamics of IPO performance provides valuable insights into the broader economic landscape and helps policymakers make informed decisions to sustain economic growth and stability.

## 5. Conclusion

In this study, various approaches to modeling a learning architecture that learns from diverse data sources have been explored. The emphasis has been on understanding how pre-modeling feature engineering contributes meaningfully to the model's predictions, as evidenced by the SHAP values. The study highlights the potential economic implications of such architectures, providing empirical evidence related to efficient market and information asymmetry theories. The analysis of long-term IPO performance is recognized as crucial for assessing economic growth, reflecting the health of newly public companies essential for innovation and industrial development. While the study serves as a significant contribution to the field of IPO performance modeling, it acknowledges the potential for further extension and exploration. The focus on feature interpretability, interaction, ranking, and text-based SHAP values offers avenues for future research. The study suggests that more in-depth debugging of the model, particularly with a larger sample size, and exploring error correction methods to determine the causation and contribution of features would be valuable avenues for further research. The exploration of text-based SHAP values can be extended further by delving into topic modeling, summarization, and correlation analyses of text-based features derived from various textual sources, such as the prospectus and Google search results. The process of modeling is ongoing, and debugging based on insights gained from SHAP values is a crucial aspect.

This study primarily aimed to establish an experimental setup for predictive modeling of IPO performance, rather than to fully explore or validate the predictive power of the models developed. However, few limitations as well as potential areas for future work are important to acknowledge. The possible limitation of this study could be the small sample, and potential biases associated with feature engineering. Further studies can extend its primary findings by enriching it with more diversified datasets. Then this study has explored the elementary feature interpretability, interaction, and ranking. The use of SHAP values in this study primarily serves to enhance the interpretability of the model by identifying the contributions of various features to the predictions. So, future studies may consider to include various statistical analysis for further robustness checks, and model's scalability.

## CRediT authorship contribution statement

**Vaidynathan Durga:** Writing – original draft, Visualization, Validation, Software, Methodology, Investigation, Formal analysis, Data curation, Conceptualization. **Kayal Parthajit:** Writing – review & editing, Writing – original draft, Validation, Supervision, Project administration, Formal analysis, Conceptualization. **Maiti Moinak:** Writing – review & editing, Writing – original draft, Supervision, Resources, Funding acquisition, Formal analysis, Conceptualization.

## Declaration of Competing Interest

There is no conflict of interest.

## Data Availability

Data will be made available on request.

## References

[1] M. Lowry, M.S. Officer, G.W. Schwert, The variability of IPO initial returns, J. Financ. 65 (2) (2010) 425–465.
[2] F. Allen, G.R. Faulhaber, Signaling by underpricing in the IPO market, J. Financ. Econ. 23 (2) (1989) 303e323.
[3] J.R. Ritter, I. Welch, A review of IPO activity, pricing, and allocations, J. Financ. 57 (4) (2002) 1795e1828.
[4] K. Rock, Why new issues are underpriced, J. Financ. Econ. 15 (2) (1986) 187e212.
[5] B. Baba, G. Sevil, Predicting IPO initial returns using random forest, Borsa Istanb. Rev. 20 (1) (2020) 13–23.
[6] D. Quintana, C. Luque, J.M. Valls, P. Isasi, Evolution strategies for IPO underpricing prediction, Financ. Decis. Mak. Using Comput. Intell. (2012) 189–208.
[7] B. Reber, B. Berry, S. Toms, Predicting mispricing of initial public offerings, Intell. Syst. Account., Financ. Manag. 13 (1) (2005) 41e59.
[8] D. Wang, X. Qian, C. Quek, A.H. Tan, C. Miao, X. Zhang, Y. Zhou, An interpretable neural fuzzy inference system for predictions of underpricing in initial public offerings, Neurocomputing 319 (2018) 102–117.
[9] T. Jewartowski, J. Lizinska, Short- and long-term performance of polish IPOs, Emerg. Mark. Financ. Trade 48 (2) (2012) 59e75.
[10] L. Tian, Regulatory underpricing: determinants of Chinese extreme IPO returns, J. Empir. Financ. 18 (1) (2011) 78e90.
[11] B. Wadhwa, Insights into the IPO underpricing for listing on National stock exchange, J. Bus. Thought 5 (2014) 38e58.
[12] E. Fedorova, S. Druchok, P. Drogovoz, Impact of news sentiment and topics on IPO underpricing: US evidence, Int. J. Account. Inf. Manag. 30 (1) (2022) 73–94.
[13] J. Tao, A.V. Deokar, A. Deshmukh, Analysing forward-looking statements in initial public offering prospectuses: a text analytics approach, J. Bus. Anal. 1 (1) (2018) 54–70.
[14] A. Lazaridou, E. Gribovskaya, W. Stokowiec, N. Grigorev. Internet-augmented language models through few-shot prompting for open-domain question answering. arXiv preprint arXiv:2203.05115, 2022.
[15] H. Zhang, H. Song, S. Li, M. Zhou, D. Song, A survey of controllable text generation using transformer-based pre-trained language models, ACM Comput. Surv. (2022).
[16] A. Gosiewska, A. Kozak, P. Biecek, Simpler is better: lifting interpretability-performance trade-off via automated feature engineering, Decis. Support Syst. 150 (2021) 113556.
[17] R. Bansal, A. Khanna, Determinants of IPO's initial return: extreme analysis of Indian market, J. Financ. Risk Manag. 1 (04) (2012) 68.
[18] M.S. Wei Leong, S. Sundarasen, IPO initial returns and volatility: a study in an emerging market, Int. J. Bus. Financ. Res. 9 (3) (2015) 71–82.
[19] D.H. Downes, R. Heinkel, Signaling and the valuation of unseasoned new issues, J. Financ. 37 (1) (1982) 1–10.
[20] B.A. Jain, B.N. Nag, Artificial neural network models for pricing initial public offerings, Decis. Sci. 26 (3) (1995) 283–302.
[21] Y. Yoon, G. Swales Jr, T.M. Margavio, A comparison of discriminant analysis versus artificial neural networks, J. Oper. Res. Soc. 44 (1) (1993) 51–60.
[22] S. Coy, R. Balasubramanian, B.L. Golden, O. Kwon, H. Beirjandi. Using neural networks to predict the degree of underpricing of an initial public offering. In

Proc., 3rd International Conference on AI Applications on Wall Street, 1995, pp. 6-9.
[23] D. Quintana, Y. Saez, P. Isasi, Random forest prediction of IPO underpricing, Appl. Sci. 6 (7) (2017).
[24] S.J. Robertson, B.L. Golden, G.C. Runger, E.A. Wasil, Neural network models for initial public offerings, Neurocomputing 18 (1-3) (1998) 165–182.
[25] B.K. Wong, Y. Selvi, Neural network applications in finance: a review and analysis of literature (1990–1996), Inf. Manag. 34 (3) (1998) 129–139.
[26] D.P. Neghab, R. Bradrania, R. Elliott, Deliberate premarket underpricing: new evidence on IPO pricing using machine learning, Int. Rev. Econ. Financ. 88 (2023) 902–927.
[27] T. Loughran, B. McDonald, IPO first-day returns, offer price revisions, volatility, and form S-1 language, J. Financ. Econ. 109 (2) (2013) 307–326.
[28] R.P. Beatty, J.R. Ritter, Investment banking, reputation, and the underpricing of initial public offerings, J. Financ. Econ. 15 (1-2) (1986) 213–232.
[29] D. Vaidynathan, P. Kayal, M. Maiti, Effects of economic factors on median list and selling prices in the US housing market, Data Sci. Manag. 6 (4) (2023) 199–207.
[30] Y. Li, J. Chan, G. Peko, D. Sundaram, An explanation framework and method for AI-based text emotion analysis and visualisation, Decis. Support Syst. (2023) 114121.
[31] A.G. Katsafados, I. Androutsopoulos, I. Chalkidis, M. Fergadiotis, G.N. Leledakis, E. G. Pyrgiotakis, Textual Inf. IPO Under.: A Mach. Learn. Approach (2020).
[32] N. Dereli, M. Saraclar. Convolutional neural networks for financial text regression. In Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics: Student Research Workshop, 2019, pp. 331-337.
[33] M. Joshi, D. Das, K. Gimpel, N.A. Smith. Movie reviews and revenues: An experiment in text regression. In Human language technologies: The 2010 annual conference of the North American chapter of the Association for Computational Linguistics, 2010, pp. 293-296.
[34] R. Liang, W. Zhang, H. Ye, Interpretable deep learning based text regression for financial prediction, Expert Syst. 40 (9) (2023) e13368.
[35] K. Gu, A. Budhkar. A package for learning on tabular and text data with transformers. In Proceedings of the Third Workshop on Multimodal Artificial Intelligence, 2021, pp. 69-73.
[36] W. Rahman, M.K. Hasan, S. Lee, A. Zadeh, C. Mao, L.P. Morency, E. Hoque. Integrating multimodal information in large pretrained transformers. In Proceedings of the conference. Association for Computational Linguistics. Meeting, NIH Public Access, 2020, Vol. 2020, p. 2359.
[37] M.K. Ho, H. Darman, S. Musa, Stock Price Prediction using ARIMA, Neural Network and LSTM Models, in: Journal of Physics: Conference Series, 1988, 2021 012041 (IOP Publishing).
[38] S. Prasanth, U. Singh, A. Kumar, V.A. Tikkiwal, P.H. Chong, Forecasting spread of COVID-19 using google trends: a hybrid GWO-deep learning approach, Chaos Solitons Fractals 142 (2021) 110336.
[39] N. Golenvaux, P.G. Alvarez, H.S. Kiossou, P. Schaus. An lstm approach to forecast migration using google trends. arXiv preprint arXiv:2005.09902, 2020.
[40] S. Hochreiter, J. Schmidhuber, Long short-term memory, Neural Comput. 9 (8) (1997) 1735–1780.
[41] T. Chen, C. Guestrin. XGBoost: A scalable tree boosting system. In Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining, 2016, pp. 785-794.
[42] S.M. Lundberg, S.I. Lee, A unified approach to interpreting model predictions, Adv. Neural Inf. Process. Syst. 30 (2017).
[43] S.M. Lundberg, G. Erion, H. Chen, A. DeGrave, J.M. Prutkin, B. Nair, S.I. Lee, From local explanations to global understanding with explainable AI for trees, Nat. Mach. Intell. 2 (1) (2020) 56–67.
[44] S.M. Lundberg, B. Nair, M.S. Vavilala, M. Horibe, M.J. Eisses, T. Adams, S.I. Lee, Explainable machine-learning predictions for the prevention of hypoxaemia during surgery, Nat. Biomed. Eng. 2 (10) (2018) 749–760.