Data Science in Finance and Economics

https://www.aimspress.com/journal/dsfe

DSFE, 4(4): 570–600.
DOI:10.3934/DSFE.2024024
Received: 21 June 2024
Revised: 10 September 2024
Accepted: 14 November 2024
Published: 09 December 2024

Research article

Performance evaluation metric for statistical learning trading strategies

Jiawei He[1], **Roman N. Makarov**[2,*], **Jake Tuero**[3] **and Zilin Wang**[2]

[1] Department of Mathematics & Statistics, 50 Stone Road East, University of Guelph, Guelph, Canada

[2] Department of Mathematics, Wilfird Laurier University, 75 University Avenue West, Waterloo, Canada

[3] Department of Computer Science, University of Alberta, 116 Street and 85 Avenue, Edmonton, Canada

* **Correspondence:** Email: rmakarov@wlu.ca; Tel: +1-548-889-3870.

Abstract: We analyze how the sentiment of financial news can be used to predict stock returns and build profitable trading strategies. Combining the textual analysis of financial news headlines and statistical methods, we build multi-class classification models to predict the stock return. The main contribution of this paper is twofold. Firstly, we develop a performance evaluation metric to compare multi-class classification methods, taking into account the precision and accuracy of the models and methods. By maximizing the metric, we find optimal combinations of models and methods and select the best approach for prediction and decision-making. Secondly, this metric enables us to construct profitable option trading strategies, which can also be used as an assessment tool to analyze models' prediction power. We apply our methodology to historical data from Apple stock and financial news headlines from Reuters from January 1, 2012 to May 31, 2019. During validation (May 31, 2018, to May 31, 2019), our models consistently outperformed the market, with two-class one-stage models yielding returns between 30% and 45%, compared to the S&P500 index's 1.73% return over the same period.

Keywords: Statistical learning; logistic regression; sentiment analysis of financial news; multi-class classification metrics; option trading strategies

JEL Codes: C53, C38, C52, G17

1. Introduction

Financial news texts and headlines have proven to be a rich data source for stock market prediction and decision-making. Financial news headlines often capture key market data and can gauge the sentiment of the market, potentially influencing traders' decisions and, in turn, impacting the stock market. Textual sentiment reflects the sentiments or emotions conveyed in text, which may be positive, neutral, or negative.

Various studies have proved the effectiveness and usefulness of the information on sentiment analysis when forecasting stock market prices. Researchers have found that sentiment derived from news headlines can significantly impact stock prices, as it captures investor mood and market sentiment (Heston and Sinha, 2017; Li et al., 2014; Mohan et al., 2019; Shah et al., 2018).

The systematic literature review in Ashtiani and Raahmei (2023) focuses on studies that use machine learning and text mining to predict the stock market based on news. The authors found that news headlines were widely used to predict the stock market. When analyzing the models' and methods' performance, it was seen that including additional information from the bodies of news articles did not improve prediction accuracy compared to using just the headlines. The most frequently used machine learning models include neural networks, support vector machines, regression, and random forest.

In another literature review by Nazareth and Reddy (2023), the authors examined recent advances in machine learning and deep learning applied to quantitative finance. They demonstrated that the extensive use of machine learning in stock markets has been highly successful, with its strong predictive performance challenging the efficient market hypothesis (EMH). The study highlights the application of this technology in predicting stock prices, stock direction, stock volatility, market indices, and return forecasting. Additional literature reviews devoted to news-based prediction of financial markets using text mining and machine learning are available in Nti et al. (2020).

Those reviews confirm our approach to model construction in this paper. We use financial news headlines to build stock return predictive models based on text-processing techniques as well as statistical and machine learning methods, such as logistic regression and the support vector machines (SVM). Stock prices are affected by numerous unpredictable factors, making precise forecasting difficult. Due to the intricacies and uncertainties inherent in natural language, accurately predicting stock market prices based on financial news headlines is challenging. Directional prediction models, which simplify the outcome to an increase or decrease, are better suited to handle market uncertainty, providing a more practical approach to decision-making in financial markets. Therefore, we focus on predicting whether the stock return exceeds a certain upper threshold or falls below a lower limit. As in Nevasalmi (2020); Barucci et al. (2021) and other studies, we introduce a multinomial response variable with two, three, four, or five classes derived from the stock log returns, where the classes are determined by return thresholds. For example, a three-class model predicts whether financial news headlines carry sentiments that can have either a positive, neutral, or negative effect on a particular stock.

Our first contribution is to construct multistage classification models using stock returns and the results of sentiment analysis of news headlines. First, we expanded a standard bag-of-words (BOW) model by including bigrams, words with positions (within headlines), and sentiment scores as features. Second, we apply principal component analysis (PCA) to reduce the dimension of covariates and mitigate possible multicollinearity problem. Third, we build multistage classification models with more than two classes by stacking two or three binary classifiers.

Sentiment data extracted from unconventional sources, such as online news and social media posts, provide insights into how the market perceives a company or financial instrument and its popularity among investors. In the next step, the prediction of the model, derived from the sentiments of the financial news, can be used to make investment decisions. Yang et al. (2017) developed a trading strategy based on Twitter sentiment. Duz Tan and Tas (2021) further confirmed that firm-specific Twitter sentiment can predict stock returns, independent of news sentiment, suggesting the potential for trading strategies that leverage social media sentiment. Frattini et al. (2022) developed a stock picking and trading algorithm based on the results of the classification procedure for stock returns.

In this paper, we propose option-based trading strategies relying on classification models for stock returns. Particularly, the stock return predictive models are used to construct trading strategies composed of standard European call and put options. If the model signals that the stock return will be positive, we buy a call option. Conversely, we buy a put option, knowing that the return is expected to be negative. Such strategies represent an efficient application of classification models and an additional assessment tool for comparison methods since the profit generated by a trading strategy can be used to examine the predictive power of the models.

The comparison of multi-class classification models is not straightforward when the number of classes exceeds two and most units are assigned to a single class, causing imbalanced datasets. The most commonly used evaluation metrics for multi-class classification models are the macro average accuracy, precision, recall, and macro F1-score (Grandini et al., 2020). However, these performance evaluation metrics may not take into account the intended application of a multi-class classification model.

The other contribution of this paper is a new performance metric for classification tasks. It is linked to the trading application of the stock-return classification models considered in this paper. Like many other traditional performance measures studied in the literature (Sokolova and Lapalme, 2009; Ballabio et al., 2018), it is based on the confusion matrix. The proposed performance metric offers the advantage of penalizing incorrect predictions. It is based on a user-defined weight matrix that can be adjusted to suit a specific classification task.

To summarize, we use sentiment analysis and natural language processing (NLP) techniques to identify the connection between financial news headlines and the stock market. Our approach consists of three steps. First, we conduct sentiment analysis of the keywords extracted from the financial news headlines using the Valence Aware Dictionary and sEntiment Reasoner (VADER). Second, we classify the impact of the headlines posted on a given trading date using pre-determined thresholds based on daily log returns of the stock of interest. Finally, we analyze the relationship between the impact of the headline and the results of the sentiment analysis using statistical methods such as logistic regression, least absolute shrinkage and selection operator (LASSO), and support vector machines (SVM) techniques combined with principal component analysis (PCA). We then develop a new metric that helps us compare and choose optimal multi-class classification models. It takes into account the intended use of the models by appropriately penalizing wrong predictions and rewarding correct predictions. To compare models, we define a performance metric as a dot (element-wise) product of weight and precision matrices. To test our models, we construct options trading strategies and simulate running them using the last year's dataset not used for training the models.

The paper is organized as follows. Section 2 describes the data, derived variables, and classes for the analysis. Section 3 describes the methodology. In Section 4, we apply our models to historical Apple stock and financial headline news from Reuters collected from January 1, 2012, to May 31, 2019. Section 5 constructs and compares options trading strategies. Section 6 provides conclusions and suggests some further avenues of study.

2. Derived variables and classes

The news headlines need to be pre-processed and cleaned following the procedures described in Plisson et al. (2004) and Abdul-Rauf et al. (2019). After that, each headline is represented as a binary vector, where each entry corresponds to a specific feature. Index s equals 1 if the sth feature appears in the headline and 0 otherwise. In addition to individual words, we also consider bigrams, which are pairs of words appearing together, and include information about whether a particular word appears at the beginning or end of a headline. People may give more attention to specific words that appear near the front of the headline; conversely, we may pay less attention to the same word appearing near the end. Considering every possible word, bigram, and whether each feature is at the start or end of the headline would make the feature space too complex. Therefore, we discard features with a low frequency. The resulting "bag of features" is then used to derive variables for the analysis.

Sentiment analysis is an NLP technique used to extract opinions or emotions from text. After pre-processing the headlines, we use VADER to calculate sentiment scores. VADER relies on a lexical sentiment dictionary that maps linguistic features to sentiment intensities. The scores generated by

VADER include positive, negative, neutral, and compound scores. The positive, negative, and neutral scores represent the proportion of their total sentiment scores, so their sum is always one for each headline (Hutto and Gilbert, 2014). The normalized compound score ranges from -1 to 1. The final set of feature-derived variables includes sentiment scores and binary variables representing features such as single words, bigrams, and word positions that appear at least five times in the total collection of headlines.

The stock log returns (`logreturn`) are calculated for each business day t using the adjusted closing prices (`close`) as follows:

$$\texttt{logreturn}[t] = \ln\left(\frac{\texttt{close}[t]}{\texttt{close}[t-1]}\right).$$

We include only the days when at least one piece of financial news has been posted. The log return, $\texttt{logreturn}[t]$, is used to classify the impact of a headline posted on day t.

The sentiment polarity of headlines toward the target can be determined by setting a threshold for stock returns (Tang et al., 2015). Using the empirical quantile function for log returns, we select the lower and upper thresholds based on the appropriate percentiles. For example, let the sentiment of a headline be classified as negative, neutral, or positive. If the log return on a given date is below the lower threshold, then the headlines posted on that date are considered to have a negative sentiment. If the log return is above the upper threshold, the corresponding headlines are classified as having a positive sentiment. All other headlines are regarded as neutral, indicating they do not significantly impact stock returns.

We use the quantile function to select the upper and lower thresholds, choosing the 90% or 95% percentile of log returns as the upper threshold and the 5% or 10% percentile as the lower threshold. To evaluate the four threshold combinations—5% and 95%, 5% and 90%, 10% and 90%, and 10% and 95%—we train regression models and identify the best set of impact thresholds. For binary classification models, we use one threshold to classify data into impactful (positive or negative) and non-impactful categories. Models with three or four categories utilize two or three thresholds, respectively. For a model with five categories, we select four thresholds to classify headlines as having strongly negative, negative, neutral, positive, or strongly positive impacts on log returns.

3. The methodology

3.1. Logistic regression

A logistic regression technique is commonly used to model the relationship between the probabilities of classifications with two outcomes and some independent variables. Suppose that the two-class response variable is the impact of the headline news and the probability $f(\mathbf{X})$ that the headline is impactful can be modeled with a logistic regression as

$$\ln\left(\frac{f(\mathbf{X})}{1-f(\mathbf{X})}\right) = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \cdots + \beta_k X_k, \tag{1}$$

where $\mathbf{X} = (X_1, X_2, \ldots, X_k)$ is the vector of independent variables from the "bag of features" derived from the sentiment analysis and natural language processing of the headlines. Estimating the logistic regression parameters, β_0,β_1,...,β_k, not only identifies the relationship between the classification probability and the independent variables but also provides an estimate of the classification probability (Stoltzfus, 2011).

Once the estimated probability is calculated using the model, it is essential to decide on a probability threshold that indicates the headline's impact. If the estimated probability exceeds this threshold, the prediction is classified as a success, meaning the headline is impactful; otherwise, the prediction is classified as a failure. When determining the probability threshold, we consider both the accuracy and precision of the prediction models. Generally, accuracy and precision are evaluated using sensitivity and specificity, which are defined as follows:

$$\textbf{Sensitivity} = \frac{\text{True positive}}{\text{True positive + False negative}},$$
$$\textbf{Specificity} = \frac{\text{True negative}}{\text{True negative + False positive}}.$$

We choose the appropriate probability threshold by maximizing the sum of sensitivity and specificity. The receiver operating characteristic (ROC) curve is known to find the best probability threshold (Hoo et al., 2017). The ROC curve is created by plotting sensitivity (true positive rate) on the y-axis against the success-specificity (false positive rate) on the x-axis to evaluate the performance of the binary classifier. The points at the top left of the ROC curve are typically selected as the threshold.

3.2. Multi-stage classification models

To construct a multi-class classification model, we combine several logistic regression models (Swiderski et al., 2012). Each logistic regression model provides its own probability threshold that maximizes the respective sum of sensitivity and specificity. Depending on the number of classes, we carry out the analysis in multiple stages.

For the three-class scenario with positive, negative, and neutral headlines, we design a two-stage classification model. In the first stage, a logistic model distinguishes between positive and non-positive impactful headlines. In the second stage, non-positive headlines are further classified as either neutral or negative. Alternatively, the model can first classify headlines as negative or non-negative in the initial stage and then separate the non-negative headlines into positive and neutral categories. Figure 1 illustrates how the positive two-stage logistic regression model works.

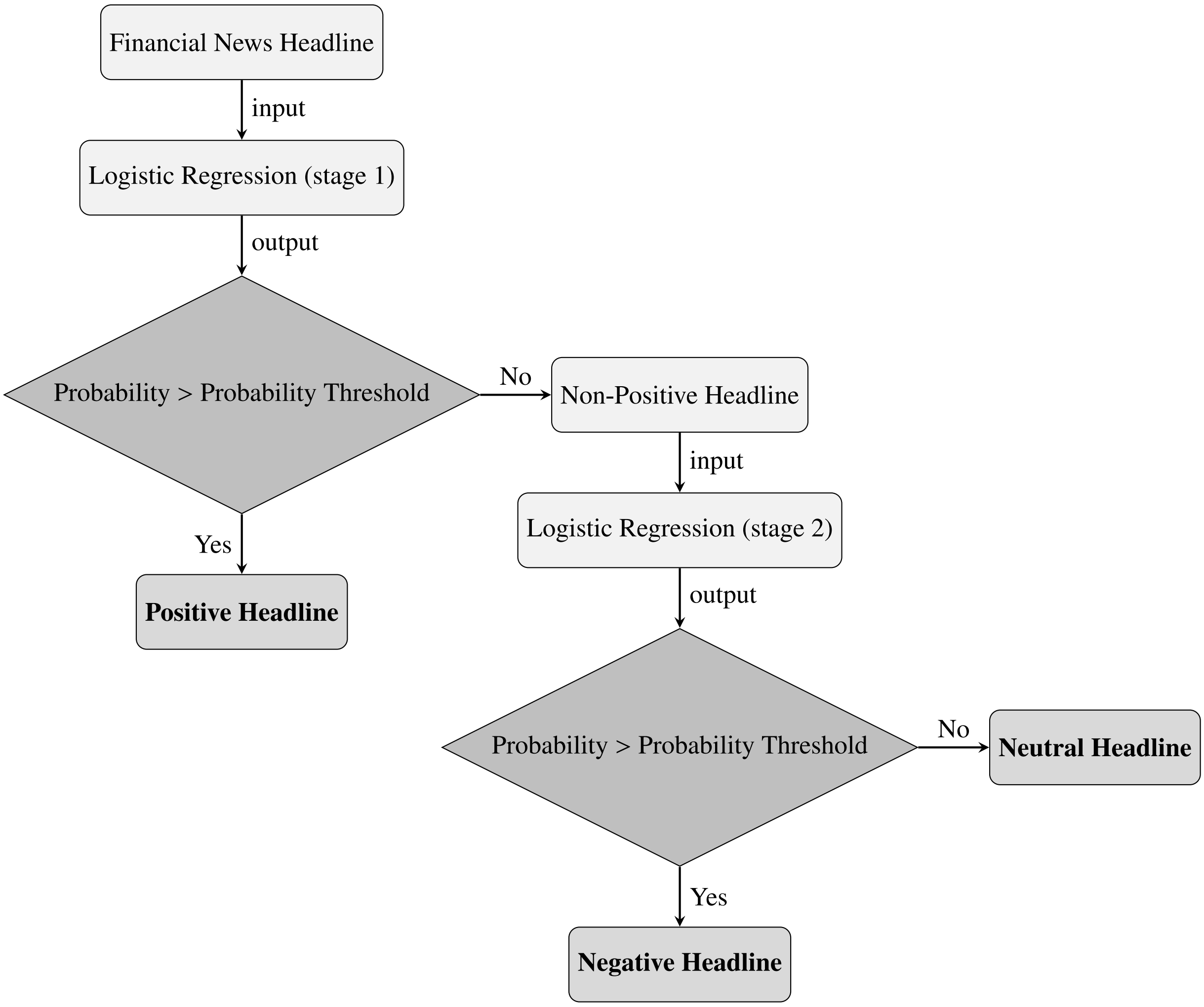
Financial News Headline
input
Logistic Regression (stage 1)
output
Probability > Probability Threshold
No
Non-Positive Headline
input
Logistic Regression (stage 2)
output
Yes
Positive Headline
Probability > Probability Threshold
No
Neutral Headline
Yes
Negative Headline

Figure 1. A positive two-stage model with three classes.

Additionally, we can develop a neutral-based two-stage model. In the first stage, this model classifies headlines as neutral (non-impactful) or non-neutral (positively or negatively impactful). In the second stage, non-neutral headlines are further classified as either positive or negative.

The two-stage logistic regression model provides a more detailed classification than the one-stage model and offers richer sentiment information. With two-stage logistic regression models, we can develop advanced trading strategies with a wider range of options. For example, by using positive two-stage models, we can create trading strategies that incorporate both call and put options, as discussed in Section 5.

Additionally, we can implement a more aggressive trading strategy by adding a larger number of options to the portfolio when headlines indicate strongly positive or strongly negative sentiments. At the second stage of the logistic regression model, we can further classify positive and negative headlines into strongly positive or non-strongly positive, and strongly negative or non-strongly negative, respectively. This approach categorizes sentiments into four states, allowing us to extract more comprehensive information from the headlines and develop two-stage models with four classes.

Similarly, a three-stage model can be constructed to account for strongly positive and strongly negative sentiments, enabling the creation of even more aggressive options trading strategies.

3.3. Performance evaluation metrics

To examine the performance of a multi-category classification model, we propose a new evaluation metric. Note that this new metric includes some traditional metrics as special cases. First, we calculate the confusion matrix **C** as defined in Tables 1 and 2.

Table 1. Two-class confusion matrix $\mathbf{C}_2$ (positive/non-positive or negative/non-negative). TP and FP stand for "true predicted" and "false predicted," respectively.

		PREDICTION	
		Pos (1)	Non-Pos (2)
ACTUAL	Pos (1)	TP_{11}	FP_{12}
	Non-Pos (2)	FP_{21}	TP_{22}

		PREDICTION	
		Neg(1)	Non-Neg (2)
ACTUAL	Neg (1)	TP_{11}	FP_{12}
	Non-Neg (2)	FP_{21}	TP_{22}

Given the confusion matrix $\mathbf{C}_2$ for the case with two sentiment categories as in Table 1, the precision matrix $\mathbf{P}_2$ is calculated as follows:

$$\mathbf{P}_2 = \begin{bmatrix} \dfrac{TP_{11}}{TP_{11} + FP_{21}} & \dfrac{FP_{12}}{FP_{12} + TP_{22}} \\ \dfrac{FP_{21}}{TP_{11} + FP_{21}} & \dfrac{TP_{22}}{FP_{12} + TP_{22}} \end{bmatrix}. \tag{2}$$

Table 2. Three-class confusion matrix $\mathbf{C}_3$. TP and FP stand for "true predicted" and "false predicted," respectively.

		PREDICTION		
		Negative (1)	Neutral (2)	Positive (3)
ACTUAL	Negative (1)	TP_{11}	FP_{12}	FP_{13}
	Neutral (2)	FP_{21}	TP_{22}	FP_{23}
	Positive (3)	FP_{31}	FP_{32}	TP_{33}

For the case with three sentiment categories as in Table 2, the precision matrix $\mathbf{P}_3$ is calculated from entries of $\mathbf{C}_3$ as follows:

$$\mathbf{P}_3 = \begin{bmatrix} \dfrac{TP_{11}}{TP_{11} + FP_{21} + FP_{31}} & \dfrac{FP_{12}}{FP_{12} + TP_{22} + FP_{32}} & \dfrac{FP_{13}}{FP_{13} + FP_{23} + TP_{33}} \\ \dfrac{FP_{21}}{TP_{11} + FP_{21} + FP_{31}} & \dfrac{TP_{22}}{FP_{12} + TP_{22} + FP_{32}} & \dfrac{FP_{23}}{FP_{13} + FP_{23} + TP_{33}} \\ \dfrac{FP_{13}}{TP_{11} + FP_{21} + FP_{31}} & \dfrac{FP_{23}}{FP_{12} + TP_{22} + FP_{32}} & \dfrac{TP_{33}}{FP_{13} + FP_{23} + TP_{33}} \end{bmatrix}. \tag{3}$$

Similarly, we construct the matrix $\mathbf{P}_n$ for models with n categories (where $n = 4, 5$).

The multi-class classification model's performance score is calculated as the dot product of a weight matrix $\mathbf{W} = (w_{ij})$ with entries satisfying $-1 \le w_{ij} \le 1$ and the precision matrix $\mathbf{P} = (p_{ij})$. For an n-class model with n-by-n matrices $\mathbf{W}_n$ and $\mathbf{P}_n$, the performance score is calculated as follows:

$$\mathbf{Score} = \mathbf{W}_n \cdot \mathbf{P}_n = \sum_{i=1}^{n} \sum_{j=1}^{n} w_{ij} p_{ij}. \tag{4}$$

Note that one can also use the sensitivity matrix or an average of the precision and sensitivity matrices in (4). However, we focus here on the case with the precision matrix.

The design of the weight matrix is driven by the application of constructed models to build options trading strategies. First, let us consider a two-category model that classifies a headline as having either a positive or non-positive impact on the stock return. Suppose we purchase a call option when a positive sentiment prediction is given, as we expect the stock price to rise, and do not make the purchase when a non-positive sentiment prediction is given. In this case, we are only concerned with whether the true positive is predicted as positive. Thus, we can set the weight matrix as $\mathbf{W}_2 = \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix}$. Notice that the **Score** calculated using (4) with the matrix $\mathbf{W}_2$ is the precision calculated for the first class.

Similarly, suppose the model classifies a headline as having a negative or non-negative impact on the stock return. In that case, we purchase a put option for a negative sentiment prediction since we expect the stock to fall, and we do not buy the option in the other case. Again, we are only concerned with the precision defined by $\dfrac{TP_{11}}{TP_{11} + FP_{21}}$ because it is the only factor that makes our trading strategy profitable.

Alternatively, one can set $\mathbf{W}_2 = \begin{bmatrix} 1 & -1/2 \\ -1/2 & 0 \end{bmatrix}$ where the weights of $-1/2$ penalize wrong predictions that lead to a potential loss or unrealized gain.

For two-stage regression models, constructing a trading strategy is a bit more complicated since we can use two types of options. We can also build a more aggressive strategy by utilizing strongly positive or strongly negative predictions.

Let us consider the case with three categories: Negative (1), Neutral (2), and Positive (3). If a headline is classified as having a positive impact (or a negative impact) on the stock return, we purchase a call option (or a put option). Otherwise, if the prediction is neutral, no option is purchased. In this case, we generate a profit when the prediction is correct. The respective entries of the weight matrix, w_{11} and w_{33}, are set to 1. We also set $w_{22} = 1$ to "reward" the model for making a correct prediction in the Neutral category, even though no option is purchased in this case.[*] Alternatively, one can set $w_{22} = 0$ since no action is taken when a headline is predicted to have no impact.

We incur a loss when a positive headline (or a negative headline) is predicted to have a negative effect (or a positive effect) on the stock return since an incorrect type of option is purchased. For example, a positive sentiment prediction leads us to buy a call option when we should buy a put option. Thus, the respective entries of the weight matrix are set to -1. The resulting matrix $\mathbf{W}_3$ is as follows:

$$\mathbf{W}_3 = \begin{bmatrix} 1 & 0 & -1 \\ 0 & 1 & 0 \\ -1 & 0 & 1 \end{bmatrix}.$$

The two-stage regression models with four categories of sentiments are treated in the same way. First, let us consider the strongly positive two-stage models with four classes: Negative (1), Neutral (2), Positive (3), and Strongly Positive (4). The weight matrix $\mathbf{W}_4^+$ in (5) can be used to calculate the performance score for this model.

$$\mathbf{W}_4^+ = \begin{bmatrix} 1 & 0 & -1 & -1 \\ 0 & 1 & 0 & 0 \\ -1 & 0 & 1 & 1/2 \\ -1 & 0 & 1/2 & 1 \end{bmatrix} \text{ and } \mathbf{W}_4^- = \begin{bmatrix} 1 & 1/2 & 0 & -1 \\ 1/2 & 1 & 0 & -1 \\ 0 & 0 & 1 & 0 \\ -1 & -1 & 0 & 1 \end{bmatrix}. \tag{5}$$

The most critical elements are the diagonal values in the output precision matrix. They represent accurate predictions. Thus, we put ones on the main diagonal of the weight matrix. The entries equal to $1/2$ represent tolerable mispredictions, because we can still make profits even if the model classifies positive headlines as strongly positive or vice versa. However, some predictions are intolerable because they eliminate our chances of making profits and cause losses within our trading strategies. Such instances include forecasts where positive or strongly positive headlines are erroneously classified as negative, thereby misleading our trading decisions and causing losses. Similarly, when negative headlines are classified as positive or strongly positive, such incorrect predictions may cause losses due to the design of aggressive trading strategies. Consequently, a value of -1 is appropriately assigned to positions within the matrix that have the potential to inflict losses on our trading strategies.

[*] If a headline is classified as having no impact, one can also purchase a butterfly spread that combines four standard options.

Similarly, for a strongly negative two-stage model with four classes, namely, Strongly Negative (1), Negative (2), Neutral (3), and Positive (4), we construct the weight matrix $\mathbf{W}_4^-$ as given in (5).

Finally, let us consider the three-stage regression model that classifies the headlines into five states of sentiment: Strongly Positive (1), Positive (2), Neutral (3), Negative (4), and Strongly Negative (5). The corresponding weight matrix $\mathbf{W}_5$ is defined as follows:

$$\mathbf{W}_5 = \begin{bmatrix} 1 & 3/4 & -1/4 & -3/4 & -1 \\ 3/4 & 1 & -1/4 & -1/2 & -3/4 \\ -1/4 & -1/4 & 1 & -1/4 & -1/4 \\ -3/4 & -1/2 & -1/4 & 1 & 3/4 \\ -1 & -3/4 & -1/4 & 3/4 & 1 \end{bmatrix}.$$

Within this matrix, certain elements carry particular significance. The diagonal values remain the most pivotal, directly influencing the profitability of our trading strategies. In contrast, the top right and bottom left corners continue to represent the most detrimental factors, leading to additional losses.

The entries of $\mathbf{W}_5$, corresponding to the cases when strongly positive or strongly negative headlines are incorrectly classified as positive or negative, respectively, have been assigned a value of $3/4$. This choice is based on the potential for still generating profits from these inaccurate predictions. Similarly, if positive or negative headlines are misclassified as strongly positive or strongly negative, we also attribute a value of $3/4$ to those instances.

Conversely, when headlines with positive or negative sentiment are misclassified as strongly negative or strongly positive, respectively, we believe that a value of $-3/4$ sufficiently accounts for the impact of this errors. Additionally, for headlines of positive or negative sentiment inaccurately predicted as negative or positive, we have set the value at $-1/2$. This reflects the losses incurred by these errors.

Furthermore, the cases when neutral headlines are incorrectly predicted as strongly positive, positive, negative, or strongly negative are identified with a value of $-1/4$ in the matrix. This accounts for the potential misdirection these predictions could introduce, leading to unprofitable trading decisions. Similarly, strongly positive, strongly negative, positive, and negative headlines that have been misclassified as neutral can affect returns and profits by providing misleading information, although they do not incur direct losses. Therefore, we have assigned a value of $-1/4$ to these positions within the matrix.

Note that the performance metric defined in (4) includes some other multi-class classification metrics as special cases. For example, if $\mathbf{W}_n$ is a diagonal matrix with $1/n$'s on its main diagonal, the dot product $\mathbf{W}_n \cdot \mathbf{P}_n$ gives us the macro average precision metric introduced in Grandini et al. (2020).

4. Analysis of Apple stock and Reuters headline news

To examine the practical feasibility and performance of the proposed methodology, we perform an extensive analysis using Apple stock data and Apple related headlines collected from the Reuters website from January 1, 2012, to May 31, 2019.

4.1. Data

For the sentiment analysis, we pre-process the headlines by removing Reuters-specific artifacts, converting all words to lowercase, removing noise words such as articles, and eliminating the delimiting characters and punctuation marks while capturing similar phrases. We then use a word stemmer to remove prefixes and suffixes and apply lemmatization to find the normalized forms of all words (Plisson et al., 2004). A word list is generated through the tokenization process, as described in Abdul-Rauf et al. (2019). Using the pre-processed headline data, we generate sentiment scores and identify regular words, bigrams, and words with positions that appear at least five times in the total dataset of the headlines. The total number of features generated from the sentiment analysis is 1178.

Thus, the dimensions of the dataset for the analysis are 1502×1178, where the number of columns (1178) represents the features derived from the sentiment analysis, and the number of rows (1502) corresponds to the trading days with posted headlines from January 1, 2012, to May 31, 2019. For evaluating the methods and applying performance metrics, we use the first seven years of trading days as the training data, with dimensions 1302×1178, and the remaining data as the test set, with dimensions 200×1178.

Note that the number of variables (features) obtained from the sentiment analysis is very large, and some of them are correlated. Principal component analysis (PCA) is an unsupervised learning method used to examine a dataset containing several correlated dependent variables. These principal components are determined through the computation of eigenvectors from the covariance matrix of the original data (Abdi and Williams , 2010).

Table 3. Selection of principal components (PCs) by cumulative variance ranging from 80% to 95%.

PCA	Selection 1	Selection 2	Selection 3	Selection 4
Number of PCs	183	232	302	418
Proportion of Variance	0.119%	0.087%	0.058%	0.031%
Cumulative Proportion	80.074%	85.061%	90.056%	95.018%

Here, we choose the principal components that can retain 80% to 95% of the total variance. As summarized in Table 3, we select the first 183, 232, 302, or 418 principal components, which retain 80%, 85%, 90%, or 95% of the total variance of the dataset, respectively. At the tuning stage, we select the number of principal components that yields the largest value of the performance score.

4.2. Model building

We begin with a logistic regression model and use a penalized version (LASSO) to prevent overfitting. Additionally, we employ a non-statistical classification approach from machine learning, namely, the support vector machines (SVM) method.

The performance metric is crucial for selecting optimal parameters of the proposed classification models. First, we aim to determine the optimal number of principal components, considering four scenarios. Second, for models with two and three classes, we seek to identify the best combination of log-return thresholds (from four possible combinations). Third, we evaluate various designs for multi-stage models: negative, neutral, and positive.

Consequently, we build and train 24 two-class models (one-stage), 48 three-class models (two-stage), 8 four-class models (strong two-stage), and 4 five-class models (three-stage) for each method. Within each group, we select the model with the highest performance score. These scores are calculated as aggregated values using 10-fold cross-validation, as explained in Subsection 4.2.4.

4.2.1. Logistic regression model

The GLM (generalized linear model) function in R (Team, 2013) is used to implement the logistic regression model in (1). The one-stage models classify data into two states: positive versus non-positive or negative versus non-negative. For these models, we also select the impact thresholds for log returns and determine the optimal number of principal components.

Among all the one-stage logistic models, the best-performing model is the negative model, using the first 183 principal components and 10% and 90% as the upper and lower quantile impact thresholds, respectively. This model achieves the highest performance score of 20.23.

There are three variations of the two-stage, three-class logistic regression model: positive, negative, and neutral. Again, we tune the impact thresholds and the number of principal components for each model.

Figure 2 illustrates the performance of the logistic models in the two-stage logistic regression. The best-performing two-stage logistic regression model is the negative model, which uses impact thresholds of 10% and 95%, along with the first 302 principal components.

In addition, the two-stage logistic models can classify log returns into four states, including a strongly negative or strongly positive class. The two-stage model with the strongly negative state outperforms the other models. Notably, the performance score decreases as the number of principal components increases. For example, the strongly positive two-stage logistic model with the largest number of principal components has the lowest performance score. The best model among these is the strongly negative two-stage model that uses the first 232 principal components.

The three-stage logistic models classify data into five categories: strongly negative, negative, neutral,

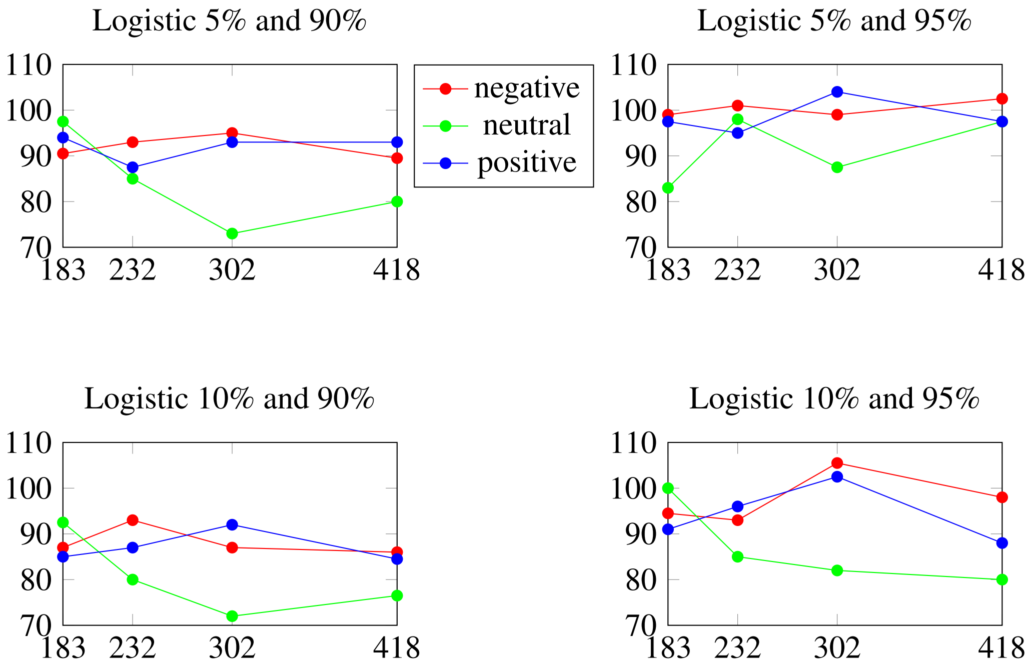

Figure 2. Two-stage logistic regression models with different impact thresholds and numbers of principal components. The x-axis represents the number of principal components that correspond to 80%, 85%, 90%, and 95% of the cumulative variance. The y-axis indicated the performance score for the models. *Positive*, *negative*, and *neutral* represent positive, negative, and neutral two-stage logistic models, respectively.

positive, and strongly positive, with fixed impact thresholds. The best three-stage logistic model uses the first 302 principal components. A summary of all the best logistic models is provided in Table 4.

Table 4. Performance of the best logistic models. PCs is the number of principal components.

Best Logistic Models	Sentiment	Thresholds	PCs	Performance Score
One-stage	Negative	10%, 90%	183	20.23
Two-stage	Negative	10%, 95%	302	105.72
Strong two-stage	Negative	Fixed	232	53.67
Three-stage	Fixed	Fixed	302	27.02

4.2.2. LASSO method

LASSO (least absolute shrinkage and selection operator) is a penalized regression method that applies regularization to prevent overfitting by shrinking the coefficients of less important predictors (Friedman et al., 2010). The optimal model parameters are obtained by solving the following minimization problem:

$$\underset{\boldsymbol{\beta}}{\text{minimize}}\left(-\left[\frac{1}{n}\sum_{i=1}^{n} y_i \cdot (\beta_0 + \mathbf{X}_i^{\top}\boldsymbol{\beta}_1) - \ln\left(1 + e^{\beta_0 + \mathbf{X}_i^{\top}\boldsymbol{\beta}_1}\right)\right] + \lambda\|\boldsymbol{\beta}\|_1\right),$$

where y_i is the response variable indicating whether the headline represented by the predictor vector $\mathbf{X}_i$ is impactful or not. The vector $\boldsymbol{\beta} = (\beta_0, \boldsymbol{\beta}_1)$, with $\boldsymbol{\beta}_1 = (\beta_1, \beta_2, \ldots, \beta_k)$, contains all the model's coefficients. Here, n is the number of observations, $\|\boldsymbol{\beta}\|_1$ denotes the L_1 norm $\boldsymbol{\beta}$ (it's a sum of the absolute values of β_i), and λ is the tuning parameter that controls the penalty's strength. LASSO effectively reduces multicollinearity and selects a subset of features for the model.

The glmnet package (Tibshirani, 1996) in R is used to build LASSO regression models and tune the penalized coefficients λ. The optimal λ is chosen to minimize the mean-squared error calculated using the 10-fold cross-validation method.

Table 5 summarizes the best LASSO models.

Table 5. Performance of the best LASSO models. PCs is the number of principal components.

Best LASSO Models	Sentiment	Thresholds	PCs	Performance Score
One-stage	Negative	10%, 90%	302	22.22
Two-stage	Neutral	5%, 95%	183	106.40
Strong two-stage	Positive	Fixed	418	50.90
Three-stage	Fixed	Fixed	232	38.97

4.2.3. Support vector machines

The support vector machines (SVM) method classifies data by finding a hyperplane that best separates different classes (Ukil, 2007). The optimal hyperplane is chosen to maximize the margin, or the distance between points from different classes and the hyperplane.

In R Studio, the `e1071` package provides kernel options to map data from one dimension to another, often from a lower to a higher dimension. Since our focus is on logistic regression, we use the linear kernel to map the data. The cost parameter in SVM models is tuned during the learning process. This parameter, known as the soft margin, controls the tolerance for misclassification errors; it determines how much misclassification the model can accept while learning. The default cost value in SVM is 1, but we tune it using the following list of values: Cost $= [0.001, 0.01, 0.1, 1, 10, 100]$. Generally, Cost = 0.01 yields the best performance score. Table 6 lists the best SVM model in each category.

Table 6. Performance of the best SVM models. PCs is the number of principal components.

Best SVM Models	Sentiment	Thresholds	PCs	Performance Score
One-stage	Negative	10%, 90%	183	22.74
Two-stage	Neutral	5%, 95%	183	111.63
Strong two-stage	Negative	Fixed	183	52.76
Three-stage	Fixed	Fixed	232	38.75

4.2.4. Cross-validation

Cross-validation involves splitting the dataset into training data, used to train regression models, and testing data, used to validate the model's performance and tune its parameters (Refaeilzadeh et al., 2009). In k-fold cross-validation, the data is divided into k non-overlapping folds. One fold is selected as the testing dataset, while the remaining $k - 1$ folds serve as the training dataset. This process is repeated k times, with each fold used exactly once as the testing set.

Nested cross-validation is used when constructing LASSO and SVM models. This approach helps in selecting the optimal hyperparameter(s) through inner cross-validation, while the outer cross-validation provides an unbiased estimate of the model's performance (Wainer and Cawley, 2021).

An unbalanced division of training and testing datasets can lead to increased misclassification. To address this, when building regression models with three categories, the dataset is divided into three collections, each containing only positive, neutral, or negative headlines. The data is then split into k folds, ensuring that each fold contains samples selected proportionally from these three collections.

We specifically use 10-fold cross-validation, where 10% of the total data is reserved for testing and 90% is used for training. Data from January 1, 2012, to May 31, 2018, is utilized for training and validating the regression models. Once we have identified the best model using the 10-fold cross-validation, we train it using all the training data and then apply the model to the evaluation data from June 1, 2019, to May 31, 2019.

The 10-fold cross-validation process generates ten confusion matrices during training. We sum the results from each category in these confusion matrices to create an aggregate confusion matrix for further analysis.

4.2.5. Comparison of the models

First, we observe that the models that classify neutral versus non-neutral classes in the first stage generally outperform the other two-stage regression models. Second, one-stage models that predict negative versus non-negative headlines demonstrate higher accuracy, suggesting that these models identify more effectively how negatively impactful headlines influence stock returns.

Table 7. Performance of the best one-stage LASSO, logistic, and SVM models, where *negative* represents the classification of negative versus non-negative headline sentiments. PCs is the number of principal components. 183 PCs capture 80% of the cumulative variance, and 302 PCs correspond to 90% of the cumulative variance.

One-Stage Models	Sentiment	Thresholds	PCs	Performance Score
LASSO	Negative	10%, 90%	302	22.22
Logistic	Negative	10%, 90%	183	20.23
SVM	Negative	10%, 90%	183	22.74

Table 7 lists the best one-stage models. From this table, we can conclude that all top models classify negative versus non-negative classes. Among these, the SVM model achieves a higher performance score than both the logistic and LASSO models.

We introduced two types of two-stage regression models: one with three headline sentiments (positive, neutral, and negative) and another with four headline sentiments (positive, neutral, negative, and strongly positive or strongly negative). Table 8 presents the best models selected from the two-stage models in the first case. We can observe that the neutral two-stage regression models outperform the others. Moreover, these neutral two-stage models perform better when the 5% and 95% quantiles are used as thresholds. Additionally, the two-stage SVM model outperforms the two-stage LASSO and logistic models, achieving the highest performance score.

Table 8. Performance of the best two-stage LASSO, logistic, and SVM models, where *neutral* represents classifying neutral versus non-neutral headline sentiments at the first stage and positive and negative at the second stage. PCs is the number of principal components. 183 PCs capture 80% of the cumulative variance, and 302 PCs correspond to 90% of the cumulative variance.

Two-Stage Models	Sentiment	Thresholds	PCs	Performance Score
LASSO	Neutral	5%, 95%	183	106.40
Logistic	Neutral	10%, 95%	302	105.72
SVM	Neutral	5%, 95%	183	111.63

Now, let us consider the four-class two-stage models. Table 9 shows that strongly negative models generally outperform strongly positive models. The strongly negative logistic model, in particular, has the highest performance score among all such models.

Recall that for the three-stage models, the impact thresholds are set at 5%, 30%, 70%, and 95% for strongly negative, negative, positive, and strongly positive outcomes, respectively. Table 10 demonstrates that both the LASSO and SVM models outperform the logistic models. The best three-stage LASSO and SVM models utilize the first 232 principal components and exhibit similar performance scores.

Table 9. Performance of the best strongly positive and strongly negative two-stage LASSO, logistic, and SVM models. The threshold combinations are fixed for these two-stage models. PCs is the number of principal components. 183 PCs correspond to 80% of the cumulative variance, 232 PCs correspond to 85% of the cumulative variance, and 418 PCs correspond to 95% of the cumulative variance.

Strong Two-Stage Models	Sentiment	PCs	Performance Score
LASSO	Positive	418	50.90
Logistic	Positive	302	47.84
SVM	Positive	418	51.28
LASSO	Negative	183	50.13
Logistic	Negative	232	53.67
SVM	Negative	183	52.76

Table 10. Performance of the best three-stage LASSO, logistic, and SVM models. PCs is the number of principal components. 232 PCs correspond to 85% of the cumulative variance, and 302 PCs correspond to 90% of the cumulative variance.

Best Three-Stage Models	PCs	Performance Score
LASSO	232	38.97
Logistic	302	27.02
SVM	232	38.75

5. Model evaluation with option strategies

Investors can benefit from trading strategies based on well-collected and aggregated information (Zhang and Skiena, 2010). To evaluate the predictive power of our classification models, we construct several trading strategies involving standard European call and put options. For simplicity, we focus on at-the-money options, where the strike price equals the spot price. Additionally, we assume that all options have one day to expiry date (also referred to as maturity or expiration date). To determine the fair price of an option, we use the Black–Scholes pricing formulas (refer to Campolieti and Makarov (2021) for a derivation of the formulas).

After the headlines are posted, we predict the stock return relative to the closing price on the previous trading day. Specifically, we use the headlines posted on day t to forecast the stock return relative to the closing price on day $t-1$. To examine the model's effectiveness and the market's efficiency, we develop two types of strategies, which we refer to as *unrealistic* and *plausible*.

First, we assume that, on day $t-1$, we already have knowledge of the headlines that will be posted on day t. This allows us to predict the stock return for day t and purchase options that mature in one day. While this scenario is unrealistic, it serves as a valuable tool for analyzing the performance of our models.

We also build more plausible trading strategies based on the idea that positive and negative headlines can influence the stock market for more than a single day. We assume that the headlines posted on day

t will similarly impact the closing price on day $t + 1$. The primary difference between plausible and unrealistic trading strategies is the timing of the option purchase. This section focuses on unrealistic strategies, which allows us to test the predictive capabilities of our models.

We construct seven trading strategies that use sentiment predictions, as well as three additional strategies without predictions (referred to as *casual strategies*). Each strategy consists of standard European call and put options. The ten trading strategies described below are implemented for each type of classification model. Strategies 1 through 7 can be designed as either plausible or unrealistic.

Strategy 1 consists exclusively of call options. It is based on a positive one-stage model that provides positive sentiment predictions. Strategy 1 involves purchasing one call option on each trading day when a headline is predicted to have a positive impact on the log return.

Strategy 2 consists exclusively of put options. It is based on a negative one-stage model that provides negative sentiment predictions. Strategy 2 involves purchasing one put option on each trading day when a headline is predicted to have a negative impact on the log return.

Strategy 3 consists of both call and put options. It uses predictions from the positive and negative one-stage models. Strategy 3 involves purchasing one call option when a positive sentiment is predicted and one put option when a negative sentiment is predicted. If there is an overlap in predictions, both kinds of options are purchased on the same day.

Strategy 4 consists of both call and put options. It uses predictions from two-stage models that classify three sentiments: positive, negative, and neutral. Since the two-stage model predicts these sentiments without overlap, Strategy 4 involves purchasing one call option when a positive sentiment is predicted and one put option when a negative sentiment is predicted.

Strategy 5 consists of both call and put options. It is based on a strongly positive two-stage model that predicts strongly positive, positive, neutral, and negative sentiments. Strategy 5 involves purchasing ten call options for each predicted strongly positive sentiment and one call or one put when a positive or negative sentiment is predicted, respectively.

Strategy 6 consists of both call and put options. It is based on a strongly negative two-stage model that predicts strongly negative, negative, neutral, and positive sentiments. Strategy 6 involves purchasing ten put options for each predicted strongly negative sentiment and one call or one put when a positive or negative sentiment is predicted, respectively.

Strategy 7 consists of both call and put options. It is based on three-stage models that predict strongly negative, negative, neutral, positive, and strongly positive sentiments. Strategy 7 involves purchasing ten put options for each predicted strongly negative sentiment or ten call options for each predicted strongly positive sentiment. Additionally, one call or one put is purchased when a positive or negative sentiment is predicted, respectively.

Strategy 1c is a trading strategy that disregards predictions and involves purchasing call options daily throughout the evaluation period. Given that there are 200 trading days with relevant headlines from June 1, 2018, to May 31, 2019, Strategy 1c involves a total of 200 calls.

Strategy 2c is a trading strategy that disregards predictions and involves purchasing put options daily throughout the evaluation period. Strategy 2c involves a total of 200 puts.

Strategy 3c is a trading strategy that disregards predictions and involves purchasing both call and put options daily throughout the evaluation period. Strategy 3c involves a total of 200 calls and 200 puts.

5.1. Standard European options and Black–Scholes formulae

European options are contracts that give their holders the right to buy or sell an underlying asset at a predetermined price on a specific future date. These options can either be calls or puts. A call option grants the holder the right to purchase the underlying asset at a strike price of K on the expiration date, while a put option grants the holder the right to sell the underlying asset at the strike price of K on the expiration date. The potential profit from exercising European call and put options are determined by the following payoffs:

$$\text{Payoff}_{\text{Call}} = \max(S-K,0) \text{ and } \text{Payoff}_{\text{Put}} = \max(K-S,0), \tag{6}$$

where S represents the underlying asset's price on the expiration date, and K is the fixed strike price. In constructing our trading strategies, we utilize at-the-money options with one day to expiry, meaning the strike price is set equal to the current stock price. If an option is purchased on day t, its strike price is $K = S_t$. The option then expires on day $t+1$, at which point the holder may exercise it. Thus, if a call option is bought on day t for C_t dollars, the profit realized is $\max(S_{t+1} - S_t, 0) - C_t$. Conversely, if a put option is bought for P_t dollars, the profit is $\max(S_t - S_{t+1}, 0) - P_t$.

We use the Black–Scholes formula to calculate option prices for several reasons. Our dataset contains only daily stock prices; without access to historical option values, we are unable to use market prices for options to compute strategy returns or calibrate a more sophisticated asset price model under the risk-neutral measure.

The Black–Scholes model is a complete market model, allowing us to estimate volatility using historical stock returns under the real-world measure and applying the same value as a risk-neutral parameter. While we could use nonlinear local-volatility models, such as the constant elasticity of variance (CEV) model—also a complete market model—we did not observe a significant difference in the one-day options prices generated by the CEV and Black–Scholes models.

Using the Black–Scholes model, we calculate each option's no-arbitrage price and the next day's profit. We then sum all the costs and profits to determine the total return at the end of the evaluation period. The Black–Scholes model provides us with the following pricing formulas for call and put options:

$$C_t = \mathcal{N}(d_1)S_t - \mathcal{N}(d_2)Ke^{-rh} \text{ and } P_t = \mathcal{N}(-d_2)Ke^{-rh} - \mathcal{N}(-d_1)S_t \tag{7}$$

where

$$d_1 = \frac{\ln\frac{S_t}{K} + (r + \frac{\sigma_t^2}{2})h}{\sigma_t\sqrt{h}} \text{ and } d_2 = d_1 - \sigma_t\sqrt{h}. \tag{8}$$

Here, h is equal to $\frac{1}{252}$, reflecting the fact that there are 252 trading days in a year. The variable t represents the day on which the options are written, and $\mathcal{N}$ denotes the standard normal cumulative distribution function. In equation (7), C_t and P_t represent the no-arbitrage prices of the call and put options at time t, respectively. S_t is the closing stock price on the day the options are purchased. Since we are dealing with at-the-money options, the strike price K is set equal to S_t.

The parameter r is the risk-free interest rate, for which we use the average annual yield of a 10-year U.S. Treasury bond, set at 2.91%. The parameter σ_t represents the annual standard deviation of the log returns over the year preceding day t. It is calculated as follows:

$$\sigma_t^2 = \frac{1}{nh}\sum_{i=t-n}^{t-1}(x_i - \bar{x}_t)^2,$$

where $n = 252$ represents the total number of trading days in the year prior to the options purchase, x_i denotes the log return on day i, and $\bar{x}_t = \frac{1}{n}\sum_{i=t-n}^{t-1} x_i$ is the average log return over the same period.

5.2. One-stage trading strategies

This subsection compares plausible and unrealistic trading strategies based on one-stage classification models. The positive one-stage model predicts whether a headline will have a positive or non-positive impact on the stock price. The best positive one-stage model is the SVM model with an upper threshold of 90% and the first 183 principal components as features. We apply the SVM model to the evaluation data from June 1, 2018, to May 31, 2019, and purchase call options on trading days with predicted positive headlines. A summary of our predictions on the test data is shown in Table 11.

The evaluation dataset contains 200 trading days, so the total number of predicted positive and non-positive days is 200. First, we create an unrealistic strategy; the results are presented in Table 13. Since

Table 11. Prediction table for the positive one-stage SVM model.

	Predict Positive	Predict Non-Positive
Actual Positive	9	15
Actual Non-Positive	29	147

38 days are predicted to have positive headlines, 38 call options are purchased for the trading strategy.

In Table 13, the following terms are defined:

Purchased: The total number of options bought in this trading strategy.

Exercised: The total number of options exercised when profitable.

Price: The total cost of all purchased options.

Payoff: The cumulative payoff of the exercised options.

Profit: The difference between the payoff and the cost.

Return: The profit divided by the cost.

Based on Table 13, we conclude that the trading strategy performs well, yielding a much higher return than Strategy 1c, which involves purchasing options every day when a headline is posted. Although the total profit from this trading strategy is small, it could be amplified by increasing the number of options purchased.

Table 14 presents the results from the plausible trading strategy based on the positive one-stage regression model. The plausible strategy also buys 38 call options, just like the unrealistic one, because both use the same predictions. However, in the plausible strategy, options are purchased on the same day a positively impactful headline is posted. The plausible strategy exercises 20 options, confirming that the positive impact of a headline can last for more than one trading day. Although this strategy exercises 20 options, it yields a lower return than the unrealistic strategy.

Table 12. Prediction table for the negative one-stage SVM model.

	Predict Negative	Predict Non-Negative
Actual Negative	11	14
Actual Non-Negative	31	144

The best negative one-stage model is the SVM model with an impact threshold of 10% and the first 183 principal components as predictors. Table 12 displays the predictions from the model applied to the evaluation dataset. Based on Table 12, the unrealistic trading strategy is expected to buy 42 put options, but only 11 of these predictions are accurate. The trading strategy based on these predictions is more profitable than Strategy 2c. Note that Strategy 2c has a higher annual return than Strategy 1c, indicating that the market tends to be more volatile when stock prices decline.

Table 14 provides the results from a plausible strategy using the negative one-stage model. This strategy also buys 42 put options but only exercises 22 of them, resulting in a smaller annual return than the unrealistic strategy.

While one-stage models can predict positive or negative headline sentiments, their predictions can be combined to form a trading strategy based on both models. For instance, based on predictions from the positive one-stage SVM model, we buy a call option. Conversely, based on predictions from the negative one-stage SVM model, we buy a put option. Since predictions come from two different models, some overlap may occur, leading to the purchase of both call and put options. This combination of call and put options is known as a straddle. Table 13 presents the results from the combined unrealistic strategy. It yields a lower return than the negative one-stage strategy.

Table 13. Summarized results for one-stage unrealistic trading strategies.

Unrealistic Strategy	Purchased	Exercised	Price	Payoff	Profit	Return
Strategy 1	38	22	48.045	78.029	29.984	62.41%
Strategy 2	42	27	52.270	124.402	72.131	138.00%
Strategy 3	80	49	100.316	202.431	102.115	101.79%
Strategy 1c	200	106	252.740	285.435	32.696	12.94%
Strategy 2c	200	94	242.611	287.201	44.591	18.38%
Strategy 3c	400	200	501.018	566.112	78.328	15.67%

We also develop a combined plausible strategy using the two one-stage models. The annual return of the combined plausible strategy is 35.11%, which is higher than that of Strategy 3c. We find that the annual return of the combined unrealistic strategy is higher than that of the plausible strategy, which supports the efficient market theory.

Table 14. Summarized results for one-stage plausible trading strategies.

Plausible Strategy	Purchased	Exercised	Price	Payoff	Profit	Return
Strategy 1	38	20	48.320	67.786	19.465	40.28%
Strategy 2	42	22	52.143	67.950	15.807	30.32%
Strategy 3	80	42	100.463	135.736	35.273	35.12%

5.3. Two-stage trading strategies

The two-stage logistic regression model is designed to classify headlines into three sentiments, enabling the construction of trading strategies that involve both call and put options. In the following, we focus on analyzing unrealistic strategies.

Positive two-stage models first classify headlines as either positive or non-positive. In the second stage, the non-positive headlines are further categorized as either neutral or negative. The best-performing positive two-stage model is the logistic regression model, which uses impact thresholds of 5% and 95% and the first 302 principal components as features.

Table 15. Prediction table for the positive two-stage logistic model.

	Predict Positive	Predict Neutral	Predict Negative
Actual Positive	4	7	2
Actual Neutral	16	142	11
Actual Negative	1	16	1

Table 16. Results of an unrealistic trading strategy for the positive two-stage logistic model.

Unrealistic strategy	Purchased	Exercised	Price	Payoff	Profit	Return
Strategy 4	35	22	43.786	86.321	42.535	97.14%
Strategy 3c	400	200	495.248	572.830	77.582	15.67%

Table 15 presents the predictions from the best positive two-stage model. It shows that 21 headlines are predicted to be positive, while 14 headlines are predicted to be negative. Consequently, the trading strategy based on these predictions includes 21 calls and 14 puts. From Table 16, we can conclude that the positive two-stage model yields a higher return compared to the positive one-stage model and Strategy 3c.

The neutral two-stage model first classifies headlines as either neutral or non-neutral. In the second stage, the non-neutral headlines are further classified as positive or negative. The best neutral two-stage model is the SVM model, using impact thresholds of 5% and 95%, along with the first 183 principal components as features.

Table 17. Prediction table for the neutral two-stage SVM model.

	Predict Positive	Predict Neutral	Predict Negative
Actual Positive	3	6	4
Actual Neutral	19	136	14
Actual Negative	4	10	4

Table 18. Results of unrealistic trading strategies for the neutral two-stage SVM model.

Unrealistic strategy	Purchased	Exercised	Price	Payoff	Profit	Return
Strategy 4	48	28	60.517	116.323	55.806	92.22%
Strategy 3c	400	200	495.248	572.830	77.582	15.67%

In Table 17, we see that 26 headlines are predicted to be positive and 22 headlines are predicted to be negative. Consequently, our trading strategy involves 26 calls and 22 puts. From Table 18, it is evident that our strategy only exercises 28 out of 48 options, indicating that the predictions are inaccurate approximately 40% of the time. The neutral trading strategy performs slightly better than the strategy based on the positive two-stage model.

The negative two-stage model first classifies headlines as either negative or non-negative. In the second stage, the non-negative headlines are further classified as either positive or neutral. The best negative two-stage model selected is the logistic model, which uses 10% and 95% as the impact thresholds and the first 302 principal components as features.

Table 19. Prediction table for the negative two-stage logistic model.

	Predict Positive	Predict Neutral	Predict Negative
Actual Positive	0	13	0
Actual Neutral	9	136	17
Actual Negative	0	17	8

According to Table 19, there are 9 headlines predicted to be positive and 25 headlines predicted to be negative. Consequently, the trading strategy involves 9 calls and 25 puts. Notably, none of the predicted positive headlines was accurate.

Table 20. Results of unrealistic trading strategies for the negative two-stage logistic model.

Unrealistic Strategy	Purchased	Exercised	Price	Payoff	Profit	Return
Strategy 4	34	25	41.376	111.411	70.035	169.27%
Strategy 3c	400	200	495.248	572.830	77.581	15.67%

The trading strategy based on the negative two-stage model proves to be the most effective one we have constructed so far. It is evident that most options in this strategy are exercised, resulting in a return of 169.265%, which significantly surpasses the return of Strategy 3c. The negative two-stage model's emphasis on classifying negative headlines contributes to its higher accuracy in predictions. Given that the market tends to be more sensitive to negative news, negative two-stage models generate more profitable trading strategies.

5.4. Strongly positive and strongly negative two-stage models

Recall that we also incorporate strong sentiments into the two-stage logistic regressions, classifying headlines into four categories: negative, neutral, positive, and strongly positive or strongly negative. Due to the limitations of two-stage models, we can only incorporate one strong sentiment at a time.

First, we generate a trading strategy using the strongly positive two-stage models. The best model in this category is the SVM model with the first 418 principal components. Table 21 summarizes the predictions made by the strongly positive two-stage SVM model.

Table 21. Prediction table for the strongly positive two-stage SVM model.

Actual\Predict	Strongly Positive	Positive	Neutral	Negative
Strongly Positive	5	1	3	4
Positive	8	12	12	18
Neutral	13	14	23	19
Negative	7	21	15	25

Based on these predictions, the trading strategy involves 378 call options and 66 put options. As shown in Table 22, while the annual return of this aggressive strategy is higher than that of a casual strategy, it does not outperform trading strategies based on two-stage models that exclude strong sentiments.

Table 22. Results of unrealistic trading strategies for the strongly positive two-stage SVM model.

Unrealistic strategy	Purchased	Exercised	Price	Payoff	Profit	Return
Strategy 5	444	285	556.190	881.004	324.814	58.40%
Strategy 3c	400	200	495.248	572.830	77.582	15.67%

We also develop a trading strategy based on strongly negative two-stage models. In these models, the first stage classifies headlines into negative and non-negative categories. Subsequently, negative headlines are further classified as strongly negative or negative, while non-negative headlines are classified as positive or neutral. The best strongly negative two-stage model is the logistic model using the first 232 principal components.

Table 23. Prediction table for the strongly negative two-stage logistic model.

Actual\Predict	Positive	Neutral	Negative	Strongly Negative
Positive	21	20	15	7
Neutral	21	27	16	5
Negative	9	19	17	5
Strongly Negative	1	7	6	4

Table 24. Results of unrealistic trading strategies for the strongly negative two-stage logistic model.

Unrealistic strategy	Purchased	Exercised	Price	Payoff	Profit	Return
Strategy 6	326	175	396.636	708.378	311.742	78.60%
Strategy 3c	400	200	495.248	572.830	77.582	15.67%

According to Table 23, the trading strategy involves 52 call options and 264 put options. Table 24 shows that this strategy performs well, generating a higher profit compared to the strategy based on the strongly positive two-stage model. This result reinforces the observation that the market is more sensitive to negative news.

5.5. Three-stage trading strategies

The three-stage model classifies headlines into five sentiment categories. This allows us to develop even a more aggressive trading strategy. The impact thresholds for this model are fixed at 5%, 30%, 70%, and 95% to distinguish between strongly negative, negative, positive, and strongly positive sentiments. The best-performing three-stage model is the LASSO model, using the first 232 principal components.

Table 25. Prediction table for the three-stage LASSO model.

Actual\Prediction	Strongly Positive	Positive	Neutral	Negative	Strongly Negative
Strongly Positive	8	0	1	2	2
Positive	15	11	12	6	6
Neutral	7	17	28	13	4
Negative	8	13	13	9	7
Strongly Negative	1	1	5	2	9

In this strategy, we buy 10 calls or 10 puts when a headline is predicted to have strongly positive or strongly negative sentiment. For positive or negative predictions, we purchase one option each. According to Table 25, the trading strategy comprises 432 calls and 312 puts. The results of this strategy are summarized in Table 26.

Table 26. Results of the unrealistic trading strategy 7 based on the three-stage LASSO model.

Unrealistic strategy	Purchased	Exercised	Price	Payoff	Profit	Return
Strategy 7	744	478	927.753	2080.406	1152.653	124.24%
Strategy 3c	400	200	495.248	572.830	77.582	15.67%

As shown in Table 26, the aggressive trading strategy based on the three-stage logistic regression outperforms aggressive strategies derived from two-stage models. Its return is significantly higher than that of Strategy 3c. Overall, the trading strategy based on the three-stage model is profitable, although aggressive strategies do not yield returns as high as regular ones.

5.6. Summary of unrealistic trading strategies

Table 27. Summary of the best unrealistic trading strategies.

Unrealistic strategy	Model	Return
Strategy 2	Negative one-stage SVM	137.996%
Strategy 4	Negative two-stage logistic	169.265%
Strategy 6	Strongly negative two-stage logistic	78.597%
Strategy 7	Three-stage LASSO	124.241%

In this section, we constructed and analyzed trading strategies for various regression models, including one-stage, two-stage, and three-stage models. Table 27 summarizes the performance of the best unrealistic strategies.

The table reveals that the negative one-stage SVM, negative two-stage logistic, and strongly negative two-stage logistic models are the most effective among all the one-stage and two-stage models. This suggests that the market is more sensitive to negative news, resulting in higher returns for strategies based on negative models compared to others.

6. Conclusions

In this paper, we built and trained several classification models to forecast log return directions using financial news headlines. We employed stacked binary classification methods, such as logistic regression and support vector machines, to build models with three or more classes. This approach provides greater flexibility compared to multinomial classification models, allowing for various configurations. For example, we developed three versions of the two-stage three-class model (negative, neutral, and positive).

To optimize model parameters and select the best approach, we compared all variants using a proposed performance metric. Additionally, we demonstrated the construction of the weight matrix used to compute performance scores in the context of option-based trading strategies. Although the weight matrix is user-defined, the performance metric remains practical and adaptable. Practitioners can tailor the weight matrix to suit their own strategies, depending on the financial products used.

Using these models, we proposed option-based trading strategies and evaluated their performance. These strategies, whether simple (using only call or put options) or more sophisticated and aggressive, are a natural application of classification models predicting short-term log return trends. Our results showed that prediction-based trading strategies significantly outperformed those without predictions, emphasizing the value of sentiment analysis in financial markets. However, we found that aggressive trading strategies based on four- and five-class models did not perform as well as those based on two- and three-class models. Additionally, the study revealed that the market was more sensitive to negative news. Despite the promising results, the strategies were based on unrealistic assumptions. Future work should focus on testing these strategies in more realistic settings, using options with several days to expiry and varying moneyness.

Disclaimer

The information provided in this article does not constitute financial, investment, or other professional advice. The author and publisher are not responsible for any losses or damages related to your reliance on this information.

Acknowledgments

The authors thank the Editor and three anonymous referees for their helpful and insightful comments. This work was supported by the research grants RGPIN-2015-311762 and RGPIN-2020-04782 from the Natural Sciences and Engineering Research Council of Canada (NSERC).

Use of AI tools declaration

The authors declare they have not used Artificial Intelligence (AI) tools in the creation of this article.

Conflict of interest

All authors declare no conflicts of interest in this paper.

References

Abdi H, Williams LJ (2010) Principal component analysis. *Wires Comput Stat* 2: 433–459. https://doi.org/10.1002/wics.101

Abdul-Rauf S, Kiani K, Zafar A, et al. (2019) Exploring transfer learning and domain data selection for the biomedical translation. In *Proceedings of the Fourth Conference on Machine Translation*, 3: 156–163. https://doi.org/10.18653/v1/W19-5419

Ashtiani MN, Raahemi B (2023) News-based intelligent prediction of financial markets using text mining and machine learning: A systematic literature review. *Expert Syst Appl* 217: 119509. https://doi.org/10.1016/j.eswa.2023.119509

Ballabio D, Grisoni F, Todeschini R (2018) Multivariate comparison of classification performance measures. *Chemometr Intell Lab* 174: 33–44. https://doi.org/10.1016/j.chemolab.2017.12.004

Barucci E, Bonollo M, Poli F, et al. (2021) A machine learning algorithm for stock picking built on information based outliers. *Expert Syst Appl* 184: 115497. https://doi.org/10.1016/j.eswa.2021.115497

Campolieti G, Makarov RN (2021) *Financial Mathematics: A Comprehensive Treatment in Discrete Time*. CRC Press. https://doi.org/10.1201/9781315373768

Duz Tan S, Tas O (2021) Social media sentiment in international stock returns and trading activity. *J Behav Financ* 22: 221–234. https://doi.org/10.1080/15427560.2020.1772261

Frattini A, Bianchini I, Garzonio A, et al. (2022) Financial technical indicator and algorithmic trading strategy based on machine learning and alternative data. *Risks* 10: 225. https://doi.org/10.3390/risks10120225

Friedman J, Hastie T, Tibshirani R (2010) Regularization paths for generalized linear models via coordinate descent. *J Stat Softw* 33: 1–22.

Grandini M, Bagli E, Visani G (2020) Metrics for multi-class classification: an overview. *arXiv Preprint*. https://doi.org/10.48550/arXiv.2008.05756

Heston SL, Sinha NR (2017) News vs. sentiment: Predicting stock returns from news stories. *Financ Anal J* 73:67–83. https://doi.org/10.2469/faj.v73.n3.3

Hoo ZH, Candlish J, Teare D (2017) What is an roc curve? *Emerg Med J* 34: 357–359. https://doi.org/10.1136/emermed-2017-206735

Hutto C, Gilbert E (2014) VADER: A parsimonious rule-based model for sentiment analysis of social media text. In *Proceedings of the international AAAI conference on web and social media*, 8: 216–225. https://doi.org/10.1609/icwsm.v8i1.14550

Li X, Xie H, Chen L, et al. (2014) News impact on stock price return via sentiment analysis. *Knowl-Based Syst* 69: 14–23. https://doi.org/10.1016/j.knosys.2014.04.022

Mohan S, Mullapudi S, Sammeta S, et al. (2019) Stock price prediction using news sentiment analysis. In *2019 IEEE fifth international conference on big data computing service and applications (BigDataService)*, 205–208. IEEE. https://doi.org/10.1109/BigDataService.2019.00035

Nazareth N, Reddy YVR (2023) Financial applications of machine learning: A literature review. *Expert Syst Appl* 219: 119640. https://doi.org/10.1016/j.eswa.2023.119640

Nevasalmi L (2020) Forecasting multinomial stock returns using machine learning methods. *J Financ Data Sci* 6: 86–106. https://doi.org/10.1016/j.jfds.2020.09.001

Nti IK, Adekoya AF, Weyori BA (2020) A systematic review of fundamental and technical analysis of stock market predictions. *Artif Intell Rev* 53: 3007–3057. https://doi.org/10.1007/s10462-019-09754-z

Plisson J, Lavrac N, Mladenic D (2004) A rule based approach to word lemmatization. In *Proceedings of IS*, 3: 83–86.

Refaeilzadeh P, Tang L, Liu H (2009) Cross-validation. *Encyclopedia database syst* 5: 532–538.

Shah D, Isah H, Zulkernine F (2018) Predicting the effects of news sentiments on the stock market. In *2018 IEEE International Conference on Big Data (Big Data)*, 4705–4708, IEEE. https://doi.org/10.1109/BigData.2018.8621884

Sokolova M, Lapalme G (2009) A systematic analysis of performance measures for classification tasks. *Inform Process Manag* 45: 427–437. https://doi.org/10.1016/j.ipm.2009.03.002

Stoltzfus JC (2011) Logistic regression: a brief primer. *Acad Emerg Med* 18: 1099–1104. https://doi.org/10.1111/j.1553-2712.2011.01185.x

Swiderski B, Kurek J, Osowski S (2012) Multistage classification by using logistic regression and neural networks for assessment of financial condition of company. *Decis Support Syst* 52: 539–547. https://doi.org/10.1016/j.dss.2011.10.018

Tang D, Qin B, Feng X, et al. (2015) Effective LSTMs for target-dependent sentiment classification. *arXiv Preprint*. https://doi.org/10.48550/arXiv.1512.01100

Team RC (2013) R: A language and environment for statistical computing. R foundation for statistical computing, vienna, austria. Available from: http://www. R-project. org/.

Tibshirani R (1996) Regression shrinkage and selection via the lasso. *J R Stat Soc B* 58: 267–288. https://doi.org/10.1111/j.2517-6161.1996.tb02080.x

Ukil A (2007) Support vector machine. In *Intelligent Systems and Signal Processing in Power Engineering*, 161–226. Springer.

Wainer J, Cawley G (2021) Nested cross-validation when selecting classifiers is overzealous for most practical applications. *Expert Syst Appl* 182: 115222. https://doi.org/10.1016/j.eswa.2021.115222

Yang SY, Mo SYK, Liu A, et al. (2017) Genetic programming optimization for a sentiment feedback strength based trading strategy. *Neurocomputing* 264: 29–41. https://doi.org/10.1016/j.neucom.2016.10.103

Zhang W, Skiena S (2010) Trading strategies to exploit blog and news sentiment. In *Fourth international aAAI conference on weblogs and social media*, 4: 375–378.