mathematics

MDPI

Article

Using Financial News Sentiment for Stock Price Direction Prediction

Bledar Fazlija * and Pedro Harder *

School of Management and Law, ZHAW Zurich University of Applied Sciences, 8400 Winterthur, Switzerland
* Correspondence: fazl@zhaw.ch (B.F.); hardeped@students.zhaw.ch (P.H.)

Abstract: Using sentiment information in the analysis of financial markets has attracted much attention. Natural language processing methods can be used to extract market sentiment information from texts such as news articles. The objective of this paper is to extract financial market sentiment information from news articles and use the estimated sentiment scores to predict the price direction of the stock market index Standard & Poor's 500. To achieve the best possible performance in sentiment classification, state-of-the-art bidirectional encoder representations from transformers (BERT) models are used. The pretrained transformer networks are fine-tuned on a labeled financial text dataset and applied to news articles from known providers of financial news content to predict their sentiment scores. The generated sentiment scores for the titles of the given news articles, for the (text) content of said news articles, and for the combined title-content consideration are posited against past time series information of the stock market index. To forecast the price direction of the stock market index, the predicted sentiment scores are used in a simple strategy and as features for a random forest classifier. The results show that sentiment scores based on news content are particularly useful for stock price direction prediction.

Keywords: sentiment analysis; natural language processing; machine learning; stock prize prediction

MSC: 91-10

check for updates

Citation: Fazlija, B.; Harder, P. Using Financial News Sentiment for Stock Price Direction Prediction. *Mathematics* **2022**, *10*, 2156. https://doi.org/10.3390/math10132156

Academic Editor: Anatoliy Swishchuk

Received: 20 April 2022
Accepted: 14 June 2022
Published: 21 June 2022

1. Introduction

The assumption that financial markets are random and cannot be predicted has been investigated in numerous studies, whereby varying results were obtained [1] (pp. 7653–7670). According to Fama [2] (p. 414), the efficiency of a financial market can be divided into three categories: weak, semi-strong, and strong. In the weak form of efficiency, asset prices reflect past market prices. In the semi-strong form of efficiency, asset prices reflect all publicly available information; and in the strong form of efficiency, asset prices reflect all information, including non-public information. Depending on the actual market efficiency, certain information about asset prices should be reflected in the daily news about the financial markets. The Internet has significantly increased the amount of available data in recent years, which has also affected the financial sector. With the advancement of research in machine learning, new opportunities have opened up in generating business-driving information from the available data. We will make use of natural language processing and other machine learning methods to distill relevant information from large amounts of news articles.

More precisely, we conduct an empirical investigation using financial text data to predict the stock market index price direction of the Standard & Poor's 500 index (S&P 500).

This is done in three stages: (A) First, a fine-tuned version of a pre-trained language model is used to label a dataset of news articles from Bloomberg and Reuters; (B) The predicted sentiment scores are used as features (alone and in combination with time series data) to predict the price direction of S&P 500; (C) the trained model's output is used to choose a strategy where either a short or long position is taken for the following trading day.

In the natural language processing part, state-of-the-art transformer models [3] are used. After fine-tuning them on financial textual data, pre-trained transformer models are able to predict the sentiment scores of financial texts with high accuracy. The sentiment scores of the titles, the content, and their sentiment scores combined with past time series information are used to predict the price direction of the S&P 500 stock market index.

We contribute to the existing literature in the following ways:

(a) We analyze the effectiveness of title- and content sentiment for stock price direction prediction.

(b) We propose several models for predicting stock price directions using sentiment scores, which give promising results.

Structure of This Paper

In Section 2, we give an overview of the relevant related literature. The datasets and proposed models are presented and described in Section 3. The fourth chapter discusses the results. This chapter also presents the limitations of this work, formulates recommendations for further research, and gives an outlook on possible developments.

2. Related Works

Sentiment analysis has been extensively studied in recent years, using a variety of machine learning techniques—including natural language processing (NLP). Traditional NLP methods have the disadvantage that they do not work well with long sentences and with specialized text. However, by incorporating recurrent deep learning models and fine-tuning large language models, these problems can be countered. To improve the quality of models used in a wide variety of domains, a dictionary can also be used. In previous research [4], natural language processing models using a dictionary have been shown to outperform classical bag-of-words models [5]. One of the advantages of using dictionaries is that they can be adapted for a variety of industries. An even more advanced form of representing words is provided by word embeddings, which take into account aspects such as linguistic similarities–and thus also provide better classification results, see [6].

State-of-the-art models have proven efficient in capturing sentiment in texts from various thematic fields. Part of the research on sentiment analysis has also dealt with applications in finance–thus dealing with the challenge of predicting the sentiment of financial text data sufficiently well.

The research around sentiment analysis in finance focuses on the following core problems: The mere sentiment classification of financial text data, and the analysis and use of text sentiment for predicting, for instance, stock prices or price directions–or vice-versa, predicting the sentiment of future news texts using, among others, market time series data.

Some authors have used classical machine learning models like support vector machines or tree-based models [7,8], to approach questions around predicting stock prices based on the sentiment of financial news or social media content.

Recently, a higher focus is put on neural networks-based models [4,9–13]—this is mainly due to the enormous progress in the field of deep neural networks. For instance, [12] uses feed-forward neural networks and recurrent neural networks. As the latter can capture relations across time, they are often used to model time series data and are shown in [12] to perform better for stock price prediction.

Furthermore, ref. [13] uses financial market performance data to predict the sentiment of news items. For this, they used models based on long short-term memory (LSTM) networks.

The work in [10] shows that the inclusion of past time series information such as price movement over the last 4, 6, 8 as well as 10 trading days in combination with a natural language processing model can increase the performance of sentiment analysis substantially. Claiming that a single classifier is not sufficient for satisfactory price direction prediction, ref. [10] constructs a multichannel collaborative network using candlestick-chart and social-media data for stock trend predictions.

Results of other previous studies [4,11] indicate that the best performance is achieved by using the transformer models, which nowadays represent the golden standard in the natural language processing field.

Ref. [14] focuses on sentiment classification for financial text data and shows that this task can be solved reasonably well with a wide range of techniques and models. Among them are transformers networks, which are shown to provide superior results as compared to other more classical natural language processing models, such as ones based on bag-of-words, ones augmented with financial lexica but also recurrent neural networks or convolutional neural networks.

The transformer is a new state-of-the-art network architecture [3] (pp. 6000–6010). They are not based on a recurrent network architecture and rely on an attention mechanism, achieving a better performance in several NLP fields than models based on recurrent or convolutional architectures. An open-source software was made available in 2019 [14] (pp. 4171–4186). Since then, transformers have been applied to many problems in NLP and gave rise to many other models, some of which are relevant for current research in finance.

Bidirectional encoder representations from transformers (BERT), were first introduced in [14]. The pre-trained BERT models can be loaded from the Python package "transformers" and can thus be easily used for natural language tasks. BERT achieves state-of-the-art results in eleven NLP tasks and can be fine-tuned by adding an additional layer for specific NLP tasks, which achieves excellent results compared to traditional methods, as shown in [4].

FinBERT results from BERT by pre-training it on financial text data and was proposed in [15], with the aim of predicting sentiment based on textual financial data.

Refs. [9,10,16] use sentiment analysis for stock prize (direction) prediction. Ref. [10] shows that the stock trend (i.e., the stock price increasing or decreasing in the near future) can be predicted with an accuracy of up to around 0.75 for selected stocks.

In [11], a transformer-based model, named FinALBERT, is proposed. To label a given text, the price predictions are used—as opposed to using the sentiment to predict the stock price direction prediction.

3. Materials and Methods

In this chapter, we discuss the relevant datasets and the proposed models. Figure 1 below provides an overview of this.

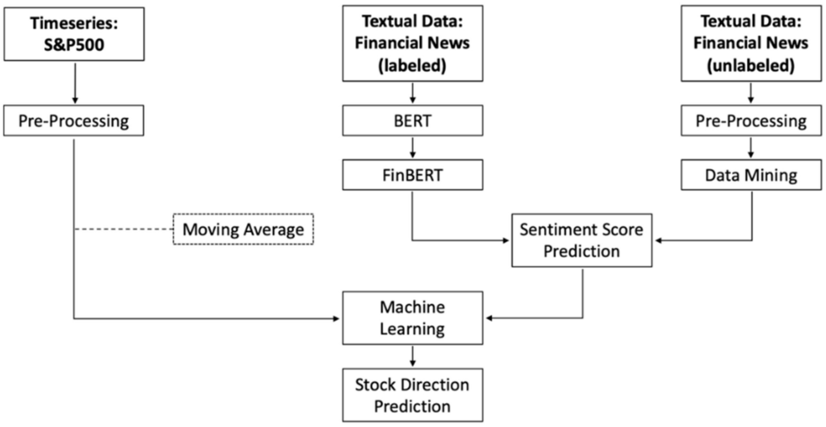
Timeseries: S&P500
Textual Data: Financial News (labeled)
Textual Data: Financial News (unlabeled)
Pre-Processing
BERT
Pre-Processing
FinBERT
Data Mining
Moving Average
Sentiment Score Prediction
Machine Learning
Stock Direction Prediction

Figure 1. Overview of the empirical research.

The used datasets and the applied methods will be discussed in the next sections. The time series data of the S&P 500 stock index is used as a target variable. For sentiment classification, transformer models are applied, using BERT models implemented in Python. To extract the sentiment score from news items, BERT is first fine-tuned on a pre-labeled dataset. In a second step, the fine-tuned BERT model is applied to unlabeled financial news data to compute a sentiment score. This sentiment score is then used as an explanatory variable for the prediction of the stock direction of the S&P 500 stock index and is fed as input into a machine learning model.

3.1. Datasets

This section introduces and discusses the different datasets used in this paper. The preprocessing of the data, whenever necessary, is then described, which is required for it to be used as input to a learning algorithm.

3.1.1. Financial Phrase Bank

The Financial Phrase Bank dataset [17] contains 5000 phrases in the field of finance and economics and was first used and published in 2013. It is intended for establishing new standards for modeling techniques in a financial context. It is labeled to allow sentiment classification, i.e., to classify each sentence into a positive, negative, or neutral category by considering only the information explicitly available in that sentence. Each sentence was annotated by 16 annotators with sufficient background expertise in financial markets. Three of the annotators were researchers, and the remaining 13 were master's students at the Aalto University School of Business with majors in finance, accounting, and economics. The annotators were asked to consider the sentences from only an investor's point of view, that is, whether the news could have a positive, negative, or neutral impact on the stock price. Sentences that did not appear to be relevant were considered neutral [17].

Once the 5000 sentences had been classified by the annotators, it was necessary to aggregate these classifications. They provided four datasets that differ in the agreement rate of the classifications. The agreement rates of the classifications in the datasets were 100%, more than 75%, more than 66%, and more than 50%. In their published paper, it turned out that the performance of a classification model did not vary significantly, depending on which of these datasets it was trained on. Accordingly, we decided to use the dataset with more than 50% agreement. The advantage is that this dataset contains 4846 observations, respectively classified sets, whereas the number of observations decreases for the more precisely classified datasets. In Table 1 a positive, a neutral, and a negative sentence are shown.

Table 1. Sentences of the Financial Phrase Bank [17].

Positive Sentiment: Sales have risen in other export markets.
Neutral Sentiment: However, the brokers' ratings on the stock differ.
Negative Sentiment: The company slipped to an operating loss of EUR 2.6 million from a profit of EUR 1.3 million.

The dataset contains 2879 neutral, 604 negative, and 1363 positive observations. It is used further in the next chapter of the empirical research to fine-tune a sentiment classification model.

3.1.2. Financial News Dataset from Bloomberg and Reuters

The financial news dataset [18] collected in 2014 was used to predict the sentiment and, in the next step, to predict stock price movements. This dataset is one of the largest publicly available datasets which can be downloaded from Remy & Ding [19]. The Bloomberg dataset contains 447,279 observations, which include the publication date, the title, and the content of the news. Figure 2 provides an overview of the number of news articles by publication date between 3 January 2007 and 26 November 2013.

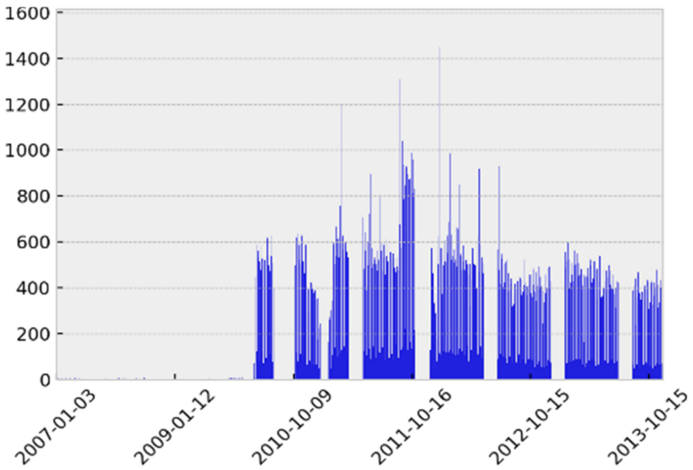

Figure 2. Amount of Bloomberg news by publication date.

It can also be seen that the number of messages per day varies significantly, and that at the beginning, there are only a few news items available for each day. Furthermore, the dataset shows some timestamps which do not contain any news. To further analyze the dataset, the number of messages per weekday was plotted in a bar chart.

Figure 3 illustrates that most of the news items were published on the midweek days. The volume of news published on Saturday or Sunday is around a quarter compared to the other days of the week. After the Bloomberg dataset has been considered, the same procedure is performed for the Reuters news dataset. Unlike the Bloomberg dataset, the Reuters news dataset contains 8,556,310 news items published between 1 January 2007 and 16 August 2016. The dataset includes the publication date, title, and URL link, but not the content, for each observation.

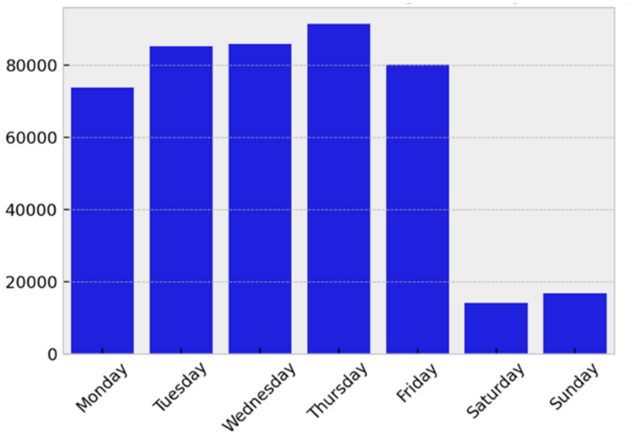

Figure 3. Amount of Bloomberg news by weekday.

Figure 4 illustrates that the number of news items per publication date does not fluctuate as much as in the Bloomberg dataset. Furthermore, the dataset does not show any time periods when no news was published.

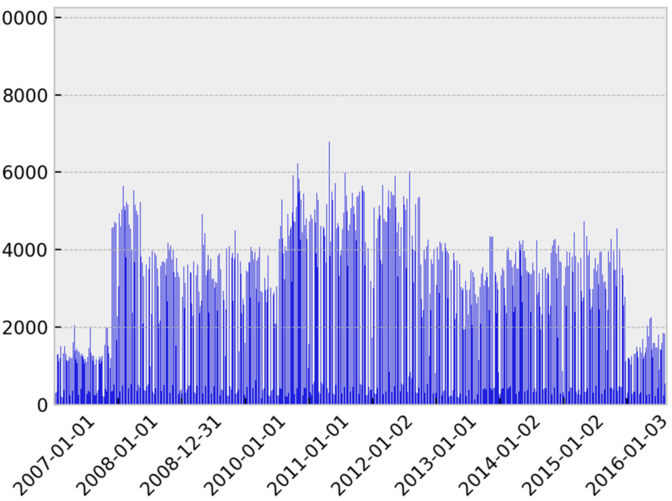

Figure 4. Amount of Reuters news by publication date.

In Figure 5, the number of news titles published per weekday runs similarly to the Bloomberg dataset.

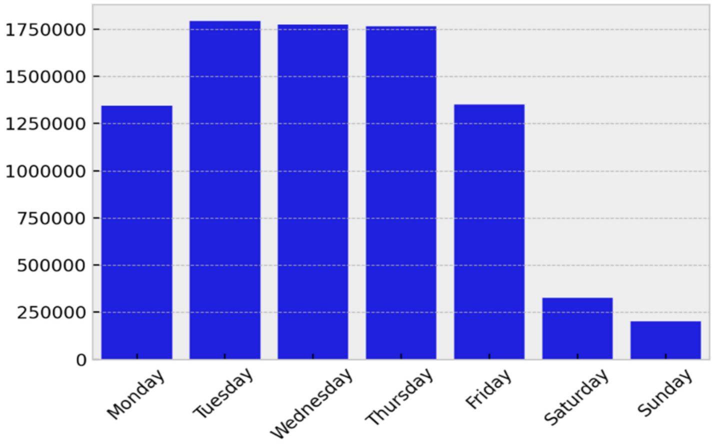

Figure 5. Amount of Reuters news by weekday.

3.1.3. Data Mining and Preprocessing of the Financial News Datasets

The first step is deleting all observations with empty content in both datasets. Since the Bloomberg dataset has both the title and the associated content, it is used as the reference dataset. This means that for days with too few or no observations in the Bloomberg dataset, additional news items were downloaded. As mentioned previously, the Reuters dataset shows, per observation, the publication date, the title, and the URL, which is used for data mining. However, to determine which news items are additionally downloaded, the number of published news items in the Bloomberg dataset was analyzed. Thereby, the date when fewer than 50 news items are available is stored in a date vector. This vector is then used to filter out the URLs from the Reuters dataset.

For a better overview of the process, we highlight the conducted steps:

1. Filter out the dates when the Bloomberg dataset contains 50 or fewer observations and generate a date vector for these cases.
2. Extract the observations of the Reuters dataset according to the date vector of step (1).
3. Use the URL of the filtered observations from step (2) to download additional articles. If there are more than 200 articles, randomly select 200 among them.
4. Since the Reuters dataset has a longer date vector than the Bloomberg dataset, additional 200 articles per day are downloaded from 26 November 2013 (date of the last

observation of the Bloomberg dataset) to 16 August 2016 (date of the last observation of the Reuters dataset).

In steps (3) and (4), some URLs which did not work were adjusted by a trial-and-error method to make the relevant content accessible. The problem with these URLs was that the year in which the news was published was also part of the URL and had to be removed. To check if these really were the correct news items, we considered a sufficiently large random sample and checked the contents and publication dates. During data mining, the content, as well as the title, were downloaded. In a second control, the downloaded title was compared to the titles from the Reuters dataset. Once the 345,776 additional articles were downloaded, this data was merged with the Bloomberg dataset. However, to use the dataset as input for our random forest classifiers, a constant number of observations per day is required. Consequently, the dataset is preprocessed to contain 58 observations per day, as this is the smallest number of available observations per day. The 58 observations were selected randomly using a random number generator.

3.1.4. Timeseries Data

The Standard & Poor's 500 Index (S&P 500®) was used as the time series data. The S&P 500 is considered the best single indicator for US large cap equities. The index comprises 500 leading companies from the United States and covers approximately 80% of the available market capitalization. It is rebalanced once a year, whereby new companies can be included in the index or existing companies can be excluded. Rebalancing takes place quarterly, whereby the weights of the individual companies in the index are adjusted [20].

Figure 6 shows the closing prices of the S&P 500 total return index. Total return means that, in the performance measurement, not only the price development but also the reinvested cash distributions, such as reinvested dividends, are considered. The reason for using a total return index and not a price index is that, among others, dividend distributions influence the price of a stock, and news reports write about these distributions. Since we discuss price direction prediction, an attempt is made to predict the direction of the price for the next day. Accordingly, the time series must still be preprocessed such that a binary classification can be performed. First, the daily returns are calculated using the formula below.

$$r_t = \frac{P_t - P_{t-1}}{P_{t-1}} \tag{1}$$

where r_t is the relative return at time t, P_t is the price at time t, P_{t-1} is the price at time $t-1$.

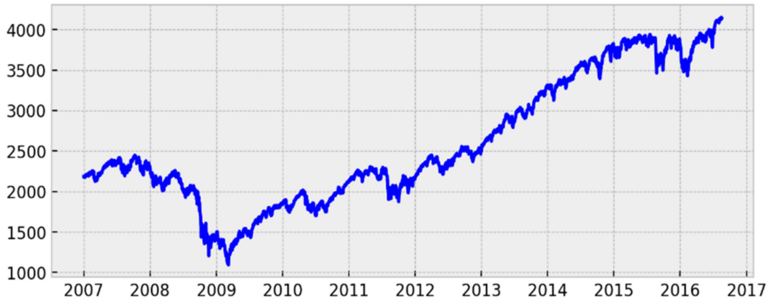

Figure 6. Price chart of the S&P 500 total return index.

Once the daily relative returns have been calculated, they are further preprocessed using the signum function.

$$d_t = sign(r_t) = \begin{cases} 1\,if\ r_t > 0 \\ 0\,if\ r_t = 0 \\ -1\,if\ r_t < 0 \,| \end{cases} \tag{2}$$

where d_t is the price direction at time t and r_t is the relative return at time t.

According to the above formula, the price direction has a value of zero if the return also has a value of zero. In this case, the problem to be solved would be a multi-classification problem. Since it is more of interest to predict a rising or falling return, the price directions which have a value of zero were over-sampled with the previous day's value. In this case, the problem to be solved turned into a binary classification problem.

The annualized returns are calculated using the geometric average, and the volatility is calculated using the standard deviation of the returns. To provide a better overview of the calculation, the formulas are listed below:

$$returnp.a. = \left(\prod_{t=1}^{T}(1 + return_t)\right)^{\frac{252}{T}} - 1 \tag{3}$$

$$volatility = \left(\frac{1}{T-1}\sum_{t=1}^{T}\left(return_t - \overline{return}\right)\right)^{\frac{1}{2}} \cdot (252)^{\frac{1}{2}} \tag{4}$$

$$return/risk = \frac{return\ p.a.}{volatility} \tag{5}$$

$$max.drawdown = \frac{Peak - Lowest\ value}{Peak} \tag{6}$$

where *peak* refers to the peak value before the largest drop, and the lowest value refers to the lowest value before a new peak is established, considering the whole period.

3.2. Proposed Models

The purpose of the empirical research presented here is to predict the S&P 500 stock market index price direction using the data outlined in the previous subsection. To this end, a pre-trained transformer network was fine-tuned on a labeled dataset, the Financial Phrase Bank dataset, presented in Section 3.1.1. After the model is fine-tuned, the preprocessed textual data from the Bloomberg and Reuters dataset, presented in Section 3.1.2, is fed as input to the model, and sentiment scores are calculated for the 58 daily news items. The predicted sentiment scores are then used as labels.

Based on the computed sentiment scores, two strategies were implemented, a simple one that does not make use of machine learning and one that does, based on a random forest classifier, to predict the price direction. These models are then further compared to the so-called random walk model—a simple and well-known baseline model.

As pointed out in previous research [21] where textual data is used to predict stock prices, there are disagreements on whether to use the title or the whole content of the news items for sentiment classification. For this reason, we developed different models that predict stock price direction using the title, the content, the title, and content combined, and the title combined with the content and some one-day lagged moving averages of the S&P 500 stock market index.

Note that transaction costs are neglected across all the models presented in this work.

The implementation of the pre-trained transformer networks is explained in Section 3.2.1. In Section 3.2.2, first, a simple stock price direction prediction model is presented, which is based on the sentiment scores but makes no use of machine learning. The model is then extended with the use of random forest models. A comparison of the results of the applied models is provided in Section 4.

3.2.1. Sentiment Scores

As mentioned above, the model architecture, as well as the concept of BERT, is based on the transformer networks. During the pre-training, the BERT model was trained by the publicists on unlabeled data over different tasks. Therefore, by adding a single output layer and subsequent fine-tuning, the model performs well in countless NLP tasks such as question-answering, language inference, etc.

While performing the pre-training, the WordPiece embeddings were used [14]. In contrast to the transformer architecture published by Vaswani [3], the architecture of BERT does not include any transformer decoder layer but has transformer encoder layers. The encoder layers serve to decompose and understand the inputs, whereas the additional output layer is task-specific and may be regarded as a decoder from the point of view of its functionality. Another difference is that, in the self-attention mechanism, not only the words on the left side of the sentence but also the words on the right side of the sentence are considered. This is also the reason why there are no decoders in the BERT architecture. The authors of the Python package published two BERT architectures, which differ based on their size. $\text{BERT}_{\text{BASE}}$ contains 12 encoder layers, and $\text{BERT}_{\text{LARGE}}$ contains 24 encoder layers.

The BERT model we used here is FinBERT [15], which uses the $\text{BERT}_{\text{BASE uncased}}$ model as a pre-trained model. Uncased means that the text is lowercased before training is started. In addition to the $\text{BERT}_{\text{BASE uncased}}$ model, a classification layer was added as the output layer, which predicts the labels as positive, neutral, or negative. The model parameters are listed in the Table 2 below.

Table 2. Tested parameters.

Parameter	Value
train_batch_size	32
eval_batch_size	32
max_seq_length	48
learning_rate	2×10^{-5}
epochs	4
lower_case	True
encoder_no	12

A batch size of 32 is used in the training as well as in the evaluation phase. The parameter "max_seq_length" stands for the maximum length of the sentences that are used. This means that for a sentence with a length of 70 words, only the first 48 words are used. The learning rate and the epochs were initiated with values of 2×10^{-5} and 4, respectively. The model parameter "lower_case" means that the texts are written in lower case. Since the BERT base model is the $\text{BERT}_{\text{BASE uncased}}$ model, it makes no difference if this parameter is set to True since this is already done in the base model, anyway. The parameter "encoder_no" stands for the number of pre-trained encoders that are fine-tuned during the downstream task. In this case, all 12 encoders are adjusted during fine-tuning.

Before proceeding with a discussion of the models used to predict price directions, some technical preliminary remarks are due.

3.2.2. Performance Measures

To assess the accuracy of machine learning models, appropriate performance measures must be used. For classification tasks, the confusion matrix (see Table 3) contains a lot of information. This is a 2 × 2 matrix for binary classification and represents the correctly predicted instances (true positives and true negatives) and the two incorrectly predicted instances (false positives and false negatives). A good model results in a confusion matrix with a high number of true positives and true negatives and a low number of entries off the diagonal.

Table 3. Confusion matrix for binary classification.

		Predicted Class	
		Positive	**Negative**
True Class	**Positive**	True positives (TP)	False negatives (FN)
	Negative	False positives (FP)	True negatives (TN)

The confusion matrix can easily be computed for k classes in a multi-class classification and is then represented by a $k \times k$-matrix.

Below we will consider the confusion matrix for the classification of sentiment into the three classes—positive, neutral, and negative.

Several performance measures can be calculated from the confusion matrix. The usual performance measures (accuracy, precision, recall, and F1 score) have the disadvantage that they give biased results when the data sets are unbalanced (i.e., unequal number of occurrences of the classes we want to predict). A simple way around this problem is to calculate these values per class and then calculate the weighted value (weighted precision, weighted recall, or weighted F1 score). The advantage is that this score is weighted by the number of observations in the classes and thus does not give a biased calculation [22].

$$weightedscore = \frac{\sum_{m=1}^{M} n_m \cdot score_m}{N} \tag{7}$$

Here M is the number of classes, n_m is the number of datapoints belonging to class m, $score_m$ is the used performance measure (so, either precision, recall, or F1-score) on class m, and N is the number of datapoints in the whole dataset.

In addition to the above measures, there are others that are considered useful in evaluating binary classifiers. We will use the so-called (binary) Brier score and the Matthews correlation coefficient (MCC) to evaluate the price direction classifier (see [23,24]).

3.2.3. Using k-Fold Cross-Validation for Machine Learning Models on Time Series Data

When validating machine learning models, especially the ones with high variance, k-fold cross-validation is often used to provide a more stable measure of model accuracy. In addition, k-fold cross-validation is used for hyperparameter tuning. In practice, these two methods are often combined simultaneously. With respect to time series data, the applicability of k-fold cross-validation has been critically discussed because it does not take into account the temporal sequence and, therefore, one may suspect that the obtained model performance estimates are overly optimistic. We will nevertheless use k-fold cross-validation to perform hyperparameter tuning and obtain stable accuracy estimates for our models. There are two main reasons why k-fold cross-validation is useful here.

- There are theoretical arguments from the literature [25] that show that k-fold cross-validation can be used for common machine learning models applied to time series data.
- We perform hyperparameter tuning using k-fold cross-validation, but ultimately test the model out-of-sample on a dataset not used in the validation step.

3.3. Random Walk Model for Price Direction Prediction

To assess the effectiveness of the proposed models, their performances in predicting the price direction of the S&P 500 index are compared with that of the well-known random walk model [26]. Formally, the stock price is modeled by the following equation:

$$S_{t+1} = S_t + \mu S_t \Delta t + S_t \sigma \epsilon \sqrt{\Delta t} \tag{8}$$

Here, S_{t+1} is the current stock price, Δt is the considered time interval, μ and σ are the mean and the standard deviation, respectively, of the returns computed from historical data, and ϵ is a stochastic error term which is standard normal distributed. As explained in Equation (2), the price direction is then computed by applying the signum function on the returns.

The first two terms on the right-hand side of equation 8 can be interpreted as follows: The current stock price is changed within the time interval Δt by the μ%. Whereas the third term includes randomness through ϵ and is modulated by the value of σ.

Simulating the above model on the same historical time series data of the S&P 500 index will result in different predictions each time due to the stochastic term involving ϵ. This in turn will lead to different results in terms of the accuracy of the prediction of the price direction. To get a good idea of the behavior of the model in describing the price direction, we run the above model 1000 times on the same data, calculate the mean over the predicted index values, and finally apply the signum function to obtain a binary result.

3.4. Simple Strategy for Price Direction Prediction

Next, we present a simple stock price direction prediction based on the sentiment scores computed from the contents and titles of news items. In a first step, the mean of the 58 sentiment scores is calculated per day. In a second step, the signum function is applied to this mean. If the mean is greater than 0 and thus represents a rather positive sentiment, it is rounded up to 1. Accordingly, a negative average value is rounded down to -1. Subsequently, this vector with the trading signals is shifted by one day and multiplied by the returns of the next trading day.

As an example, one can use the news that was published on Monday. First, the sentiment scores are calculated as described in the previous chapter, and then the mean value is calculated thereof, which then is rounded to 1 or -1. A positive value is rounded up to 1, which stands for a buy signal, and is multiplied by the return on Tuesday. A negative value is rounded to -1 and multiplied by the return on Tuesday, which represents a 100% short position. For this purpose, transaction costs are neglected; however, it is possible to replicate this trading strategy cheaply using S&P 500 futures.

3.5. Price Direction Prediction with Random Forest

Random forest (RF) can be applied to solve classification and regression problems and falls within the area of machine learning called ensemble learning. Different machine learning models show different strengths and weaknesses, depending on the nature of the problem at hand. The main idea of ensemble learning is to build a strong predictive model based on a collection of several, possibly weak, models. Ensemble learning often helps to reduce the widespread machine learning problem of poor out-of-sample prediction accuracy due to high variance [27] (pp. 605–507).

The random forest has several hyperparameters which can be tuned to find an optimal model, among which are n_estimators (number of trees in the forest); max_depth (maximum depth of each tree); bootstrap (whether bootstrap samples are used when building trees), and criterion (function that is used to measure the quality of a split, whereby the Gini impurity and entropy are the supported methods).

We will use random forest classifiers to predict the price direction based on (a) sentiment scores and (b) sentiment scores in combination with time series data (moving avarages). The input variables are always the predicted sentiment scores (as given by the fine-tuned language model), and the target variable is the price direction of the next trading day. For the experiments, all settings are kept constant except the training and test data.

To get a comprehensive estimation of the out-of-sample performance, we will train and test several models, according to the sets given in Table 4. The implemented random forest models are first fitted in the first four years and tested in the following year. In a second run, the models are fitted in the first five years and tested in the sixth year, and so on.

Table 4. In-sample and out-of-sample periods.

In-Sample Period	Out-of-Sample Period
01/01/2007 to 12/31/2010	01/01/2011 to 12/31/2011
01/01/2007 to 12/31/2011	01/01/2012 to 12/31/2012
01/01/2007 to 12/31/2012	01/01/2013 to 12/31/2013
01/01/2007 to 12/31/2013	01/01/2014 to 12/31/2014
01/01/2007 to 12/31/2014	01/01/2015 to 12/31/2015
01/01/2007 to 12/31/2015	01/01/2016 to 08/16/2016

During the fitting of the in-sample time period, hyperparameter tuning is conducted—with the aim of finding the best-performing model. For this, the Scikit-learn grid search method "GridSearchCV" is used, with $k = 8$ folds.

So, in the learning process, the entire training dataset is split into numerous training and validation samples. The model is then fitted on the training datasets and tested on the validation datasets, with the resulting cross-validated score being the average over all validation runs. In the case of the k-fold cross-validation, the original training dataset is split into k subsets with the same size for all subsets without replacement. The model is trained on $k - 1$ training subsets and evaluated on the remaining kth subset, i.e., the validation subset. This process is repeated until each of the k subsets is used once as the validation set. The resulting cross-validated performance is the average of the k performance measurements on the k validation subsets [28].

Cross-validation (as opposed to a single train-test-split) allows for a more stable estimation of the out-of-sample performance of the model (as obtained on the test data) and is particularly useful for models that exhibit high variance, such as decision trees.

The left column of the table above lists the parameters which are used for hyperparameter tuning. The values tested are shown in the right column. GridSearchCV tests all possible parameter combinations and cross-validates them simultaneously.

This means that the model tests different random forests with the specified parameters from Table 5, cross-validates them, and returns the random forest model with the highest cross-validated score as output. The model with the highest cross-validated score is then fitted to the whole in-sample dataset and applied during the out-of-sample period.

Table 5. Tested hyperparameters.

Parameter	Value
n_estimators	250, 500, 750, 1000
max_depth	1, 3, 5, 10, 20, 30
max_features	2, 5, 10, 15, 30, 40, 58
criterion	gini, entropy
random_state	333
bootstrap	True

The weighted F1-score and the weighted precision score are used as performance scores. It is important to note that this procedure is applied to all simulations. The results of the simulations where the weighted precision score is used as a performance score are shown in the next subsections. Since the procedure remains the same for different performance metrics, the results where the weighted F1-score is used as a performance metric are only shown in Section 3 and compared with the other results.

After the FinBERT model is fine-tuned on the Financial Phrase Bank dataset, the sentiment score prediction of the aggregated Reuters and Bloomberg dataset is started.

For this purpose, the fine-tuned FinBERT model is used, and the sentiment scores of the titles as well as the contents are predicted. In BERT models, individual sentences are given as input instead of the whole text as observations. The added output layer in FinBERT provides as output the probabilities for the financial text being positive, neutral, or negative. The classification is done according to the highest of the three probabilities. To calculate the sentiment score, it is defined that a score of 1 is highly positive and a score of -1 is highly negative. A sentiment score of 0 can be considered neutral.

Let us consider a sample title: "*TREASURIES–Bonds creep higher in thin trade*". This sentence is a title of an observation of the Reuters dataset and was published on 2 January 2007. The FinBERT model provides the following probabilities for the sentiment classification:

$$\begin{gathered} \textit{positive} = 0.965 \\ \textit{negative} = 0.0175 \\ \textit{neutral} = 0.0175 \end{gathered}$$

The calculation of the sentiment score is simple and can be calculated from the difference between the probability that a text is *positive* and the probability that a text is *negative*.

$$Sentiment\ score = 0.965 - 0.0175 = 0.9475$$

The *sentiment score* in this case is high and corresponds to a text being classified as positive. Texts that are classified as neutral usually have low probabilities of being positive or negative. Consequently, their sentiment scores are close to 0. Since, in BERT, sentences and not whole texts are considered as observations and a news story often consists of several sentences, the average value of all sentiment scores from the sentences is used as sentiment score for the entire news content.

Once the sentiment scores of the 58 news stories per day are predicted, two new DataFrames are created, which can be used as input for a machine learning model. The DataFrames have the date vector as index and columns with the 58 sentiment scores, whereby the datasets differ in that one contains the sentiment scores of the titles and the other the sentiment scores of the content.

In the random forest models, the sentiment scores are used as features, and the price direction of the S&P 500 stock market index is used as the target variable. Since the stock index is only traded on weekdays, and thus contains a different date vector than the DataFrames with the sentiment scores, the DataFrames of the sentiment scores are reindexed to the date vector of the S&P 500 stock index. This means that published messages on weekends are deleted from the DataFrames and that if there are no sentiment scores on a trading day, the values are forward-filled with the previous sentiment scores. This also prevents news, respectively sentiment scores, from being used for the prediction of the return direction of the same day. To continue using the scores from the weekends, the sentiment scores from Friday, Saturday, and Sunday are averaged so that they can be used to predict the price direction for Monday.

4. Results

4.1. Sentiment Classification

First, let us consider the performance of the fine-tuned model when it comes to predicting the sentiment of given news items. For fine-tuning, the subset with more than 50% agreement among the annotators of the Financial Phrase Bank dataset is used. This is split into a train, validation, and test datasets with the proportions of 72-8-20. As pointed out, the target variable is the sentiment of the news, with the three labels being positive (label 0), negative (label 1), and neutral (label 2). Accordingly, the model is trained, validated, and then tested on the test dataset. The resulting confusion matrix is shown in Figure 7.

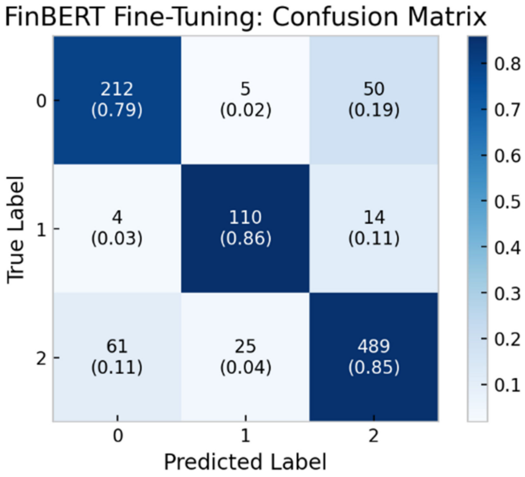

Figure 7. Confusion matrix for FinBERT sentiment classification.

This yields the following results, given in Table 6.

Table 6. Classification accuracy using FinBERT.

Accuracy	Weighted Precision	Weighted Recall	Weighted $F_1 - Score$
0.836	0.839	0.836	0.837

4.2. Simple Strategy Based on Sentiment Scores

To compare the simple strategy (without a random forest classifier) with the strategies explained later in this chapter, the returns are calculated and indexed from 1 January 2011 to 16 August 2016, where the starting value is 100. This can be considered as investing 100 USD at the beginning.

In Figure 8 above, the simple strategy based on the content sentiment scores outperforms the S&P 500 stock market index. A closer look shows that an outperformance is generated at the end of 2011, which explains the outperformance of this strategy since, in the following years, all price downturns are taken along and are not correctly reflected by the content sentiment scores, as it is the case for example in the price decline in 2015. The strategy based on the title sentiment scores is generally poor and even shows a negative performance in this period. This also resonates with findings from previous research [21] that the content of the news should also be considered rather than just the title for sentiment scoring.

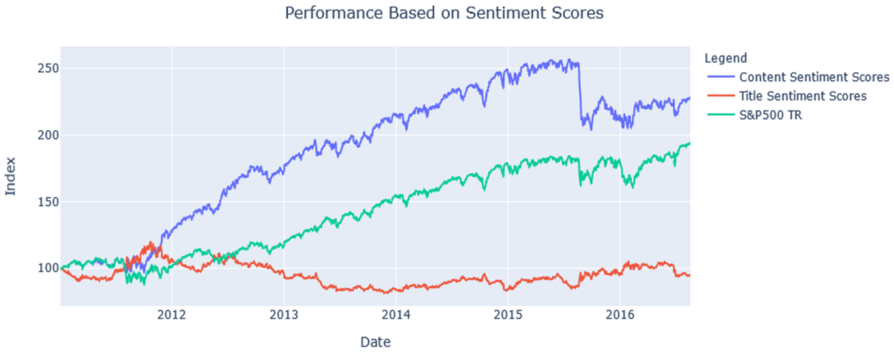

Figure 8. Performance based on sentiment scores.

Table 7 gives an overview of the relevant numbers. The simple strategy based on the sentiment of the text content leads to a higher return and is also preferable in terms of the risk-return ratio.

Table 7. Performance measures of strategies based on sentiment scores from 1 January 2011 to 16 August 2016.

	S&P 500	Content SC [1]	Title SC [1]
Return p.a.	12.045%	15.208%	−0.934%
Volatility	15.108%	15.097%	15.128%
Return/Risk	0.797	1.008	−0.062
Max. Drawdown	18.641%	20.598%	32.089%

[1] The abbreviation SC stands for sentiment scores.

4.3. Random Forest Classifier Based on Sentiment Scores

Figure 9 shows that no strategy can outperform the S&P 500 stock market index, when the sentiment of the title is solely considered. In this strategy, such random forest classifiers are trained which are based on different sets of features, in particular, ones related to sentiment, are trained. It is worth mentioning that the strategy “RFC Title SC” stands for random forest classification based on title sentiment scores and takes a short position when a negative return is predicted. In the second strategy, "RFC Title SC2", if a negative return is predicted, only divestment is done, which means that the simulated return on such days is zero. Between the two strategies, strategy 2 performs better because in the case of a false negative prediction, only divestment is done, and no short position is taken.

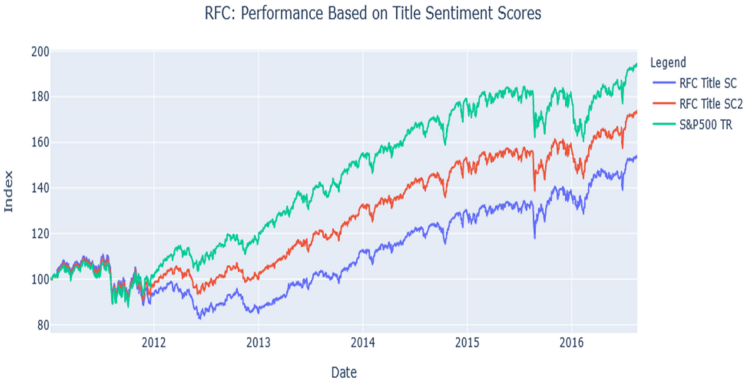

Figure 9. Out-of-sample performance of random forest classification based on title sentiment scores.

Table 8 shows that despite using a random forest classifier, title sentiment leads to worse results than the previous strategy, which is based solely on text content sentiment and does not use a classification model at all. However, when it comes to using the title sentiment, the advantage of the random forest classifier is clearly visible. For instance, compare the returns in Table 8 with those in the last column of Table 7.

Table 8. Performance measures of strategies based on RF on title sentiment scores from 1 January 2011 to 16 August 2016.

	S&P 500	RFC Title SC	RFC Title SC2
Return p.a.	12.045%	7.673%	9.920%
Volatility	15.108%	15.119%	14.606%
Return/Risk	0.797	0.508	0.679
Max. Drawdown	18.641%	25.518%	18.669%

Next, we consider the results for the sentiment predicted solely on the content of the news items, while using a random forest classifier in Figure 10.

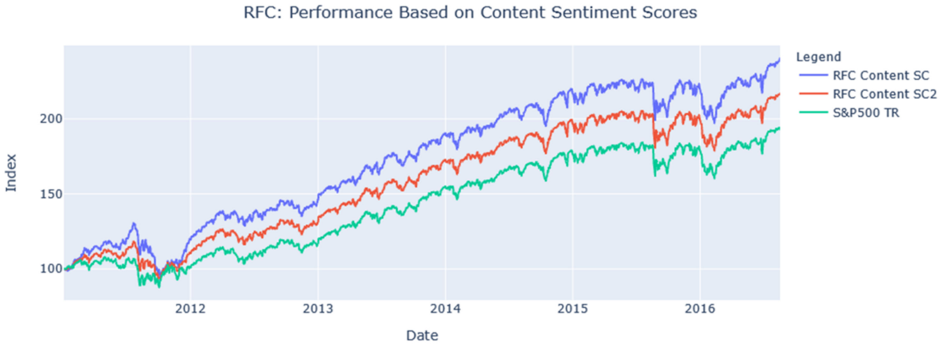

Figure 10. Out-of-sample performance random forest classification based on content sentiment scores.

Table 7 shows that the use of text content sentiment in a simple strategy leads to promising results even without using a machine learning classifier. The comparison in terms of title sentiment also showed a significantly better result using a random forest classifier over using the simple strategy solely based on the sentiment. Table 9 shows that the classifier also improves performance for text content, although the difference is not as large as in the cases mentioned previously.

Table 9. Performance measures of strategies based on RF classification on content sentiment scores from 1 January 2011 to 16 August 2016.

	S&P 500	RFC Content SC	RFC Content SC2
Return p.a.	12.045%	16.261%	14.226%
Volatility	15.108%	15.089%	14.442%
Return/Risk	0.797	1.078	0.985
Max. Drawdown	18.641%	28.187%	20.755%

Now, we consider the usage of the sentiment based on the title and the content of news items in combination.

Due to the relatively weak performance of all the models based on the title sentiment, one might suspect that this feature will have a negative impact on models also when used in combination with the text content sentiment. Figure 11 shows that the results of the two strategies do not differ much from the stock market index. Nevertheless, both strategies show a slight outperformance. Therefore, it can be concluded that both strategies have a more attractive return characteristic, as shown in Table 10.

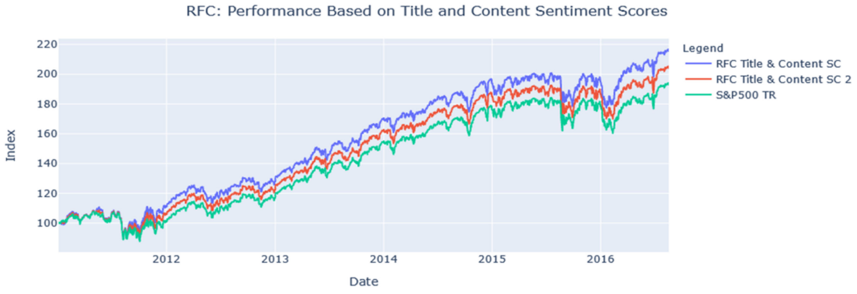

Figure 11. Out-of-sample performance random forest classification based on title and content sentiment scores.

Table 10. Performance measures of strategies based on RF classification on title and content sentiment scores from 1 January 2011 to 16 August 2016.

	S&P 500	RFC Title & Content SC	RFC Title & Content SC2
Return p.a.	12.045%	14.143%	13.093%
Volatility	15.108%	15.096%	14.975%
Return/Risk	0.797	0.937	0.874
Max. Drawdown	18.641%	18.810%	17.587%

4.4. Comparison to Random Walk Model

Above, we discussed and compared the different strategies and forms of sentiment scores used. For a more nuanced comparison, we next compare the price direction prediction of the sentiment-based random forest classifier with the benchmark random walk model. Figure 12 clearly shows that the prior outperforms the random walk model.

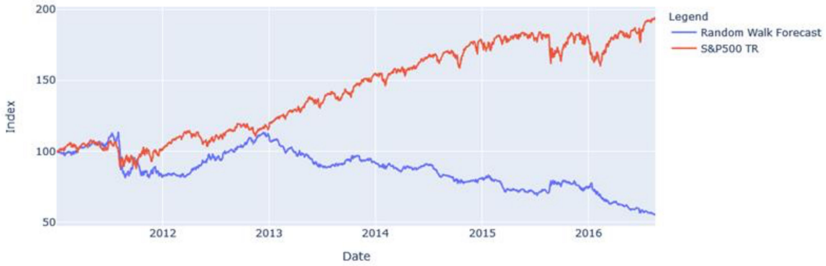

Figure 12. Comparison of the random forest classifier with the random walk model.

To get sufficient proof of the superiority of our models, several scores are computed and compared. Besides accuracy, binary classification models are judged through scores such as Brier scores, and Matthews Correlation Coefficient (MCC), among others (see [23,24,29]).

It was highlighted in [24] that MCC is more informative than Brier scores and that, especially for Brier scores of around 0.25, the results cannot be interpreted unambiguously (they can be either very good or very bad). Given the nature of the prediction task, one would not expect overly high accuracy or MCC, yet the results in Table 11 show significant differences between the models. The model that uses the random forest classifier based on content sentiment produces the best results. This is evident not only in accuracy, but also in MCC, showing a strong enough difference between using a random forest classifier for title sentiment and for content sentiment.

Table 11. Summary of performance measures.

	Accuracy	Binary Brier Score	Matthews Correlation Coefficient
Content Sentiment	0.518	0.482	0.026
Title Sentiment	0.206	0.794	0.007
RFC Title	0.535	0.465	−0.023
RFC Content	0.561	0.439	0.069
RFC Title & Content	0.554	0.446	0.039
RFC Title, Content & MA	0.552	0.448	0.045
Random Walk Model	0.499	0.501	−0.011

Additional to the above results, Tables A1 and A2 in Appendix A summarize the results for all models and strategies as optimized by the weighted precision score and the weighted F1-score, respectively.

5. Discussion

As already mentioned, the random forest classification models were optimized with respect to the performance measures weighted precision and weighted F1-score. The results are listed in two tables in Appendix A. Another useful optimization would be to optimize the random forest models according to the performance measure recall or weighted recall, because this performance measure minimizes the number of false negative predictions. Since the random forest models minimized the number of false negative predictions by classifying all or almost all predictions into the label positive, their results are not listed in this chapter.

Based on the findings of this paper, it can be said that the sentiment scores calculated by state-of-the-art NLP methods can aid stock price forecasting through the lens of sentiment. In practice, there are different scenarios where the models shown here can be used. Since the sentiment scores should also reflect the market sentiment, the scores can be used for a risk-based approach, whereby a combination with other proven indicators is recommended. Another possibility arises from the use of the NLP methods shown in relation to sustainable investing. For example, reports can be used to find out whether the company takes ethical or ecological aspects into account, and, accordingly, by automating the reading of the reports, a pre-selection of sustainable and less sustainable companies can be made.

Recommendations for further research can be made on various levels. First and foremost, we did not take transaction costs into consideration. This is a significant limitation and should be addressed in future studies. From a data perspective, the author proposes to use news on all companies included in the stock market index. In this way, the company-specific sentiment scores can be considered features. From the methods standpoint, further machine learning models can be used for the classification task. Additionally, other past time series information such as a moving value at risk or a moving expected shortfall of the previous days can be added as input for improving the quality of the predictions.

Author Contributions: Conceptualization, B.F. and P.H.; methodology, B.F. and P.H.; software, P.H.; validation, P.H.; formal analysis, P.H.; investigation, P.H.; data curation, P.H.; writing—original draft preparation, P.H.; writing—review and editing, B.F. and P.H.; visualization, P.H.; supervision, B.F. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

Acknowledgments: Open access funding provided by ZHAW Zurich University of Applied Sciences. We thank the two anonymous reviewers for their valuable comments that enhanced the readability and content of this paper significantly. We thank Enis Abdyli for proofreading the article.

Conflicts of Interest: The authors declare no conflict of interest.

Appendix A

Table A1. Summary of empirical results, random forest classification optimized by the weighted precision score.

	Accuracy	Weighted Precision	Weighted Recall	Weighted F1-Score	Binary Brier Score	Matthews Correlation Coefficient	Return p.a. *	Volatility *	Return / Risk	Max. Drawdown *
S&P 500							12.05	15.11	0.80	18.64
Content Sentiment	0.518	0.307	0.518	0.386	0.482	0.026	15.21	15.10	1.01	20.60
Title Sentiment	0.206	0.308	0.206	0.247	0.794	0.007	−0.93	15.13	−0.06	32.0
RFC Title	0.535	0.487	0.535	0.437	0.465	0.023	7.67	15.12	0.51	25.52
RFC Content	0.561	0.574	0.561	0.447	0.439	0.069	16.26	15.09	1.08	28.19
RFC Title & Content	0.554	0.568	0.554	0.412	0.446	0.039	14.14	15.10	0.94	18.81
RFC Title, Content & MA	0.552	0.535	0.552	0.483	0.448	0.045	11.66	15.10	0.77	18.41
Random Walk Model	0.499	0.500	0.499	0.499	0.501	0.011	−9.75	15.1	−0.65	−51.44

* Measurements are in %.

Table A2. Summary of empirical results, random forest classification optimized by the weighted F1-score.

	Accuracy	Weighted Precision	Weighted Recall	Weighted F1-Score	Binary Brier Score	Matthews Correlation Coefficient	Return p.a. *	Volatility *	Return / Risk	Max. Drawdown *
S&P 500							12.05	15.11	0.80	18.64
Content Sentiment	0.518	0.307	0.518	0.386	0.482	0.026	15.21	15.10	1.01	20.60
Title Sentiment	0.206	0.308	0.206	0.247	0.794	0.007	−0.93	15.13	−0.06	32.0
RFC Title	0.534	0.509	0.534	0.482	0.466	0.006	4.20	15.13	0.28	−37.92
RFC Content	0.531	0.506	0.531	0.486	0.469	0.002	3.71	15.12	0.25	23.06
RFC Title & Content	0.540	0.517	0.540	0.486	0.460	0.020	7.38	15.11	0.49	24.79
RFC Title, Content & MA	0.524	0.513	0.524	0.512	0.476	0.016	5.45	15.12	0.36	21.35
Random Walk Model	0.499	0.500	0.499	0.499	0.501	0.011	−9.75	15.1	−0.65	−51.44

* Measurements are in %.

References

1. Nassirtoussi, A.K.; Aghabozorgi, S.; Wah, T.Y.; Ngo, D.C.L. Text mining for market prediction: A systematic review. *Expert Syst. Appl.* **2014**, *41*, 7653–7670. [CrossRef]
2. Fama, E. Efficient capital markets: A review of theory and empirical work. *J. Financ.* **1970**, *25*, 383–417. [CrossRef]
3. Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A.; Kaiser, L.; Polosukhin, I. Attention Is All You Need. In Proceedings of the 31st International Conference on Neural Information Processing Systems (NIPS'17), Long Beach, CA, USA, 4–9 December 2017; Curran Associates Inc.: Red Hook, NY, USA, 2017; pp. 6000–6010.
4. Mishev, K.; Gjorgjevikj, A.; Vodenska, I.; Chitkushev, L.; Trajanov, D. Evaluation of Sentiment Analysis in Finance: From Lexikons to Transformers. *IEEE Access* **2020**, *8*, 131662–131682. [CrossRef]
5. Consoli, S.; Barbaglia, L.; Manzan, S. Fine-grained, aspect-based sentiment analysis on economic and financial lexicon. *Knowl.-Based Syst.* **2022**, *247*, 108781. [CrossRef]
6. Barbaglia, L.; Consoli, S.; Wang, S. Financial Forecasting with Word Embeddings Extracted from News: A Preliminary Analysis. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*; Springer: Cham, Switzerland, 2021; Volume 1525, pp. 179–188.
7. Chen, L.; Qiao, Z.; Wang, M.; Wang, C.; Du, R.; Stanley, H.E. Prediction of stock market index movement by ten data miningtechniques? *Mod. Appl. Sci.* **2009**, *3*, 28–42.
8. Chakraborty, P.; Pria, U.S.; Rony, M.R.A.H.; Majumdar, M.A. Predicting stock movement using sentiment analysis of Twitter feed. In Proceedings of the 2017 6th International Conference on Informatics, Electronics and Vision & 2017 7th International Symposium in Computational Medical and Health Technology (ICIEV-ISCMHT), Himeji, Japan, 1–3 September 2017; pp. 1–6.
9. Li, X.; Xie, H.; Chen, L.; Wang, J.; Deng, X. News impact on stock price return via sentiment analysis. *Knowl.-Based Syst.* **2014**, *69*, 14–23. [CrossRef]
10. Ho, T.-T.; Huang, Y. Stock Price Movement Prediction Using Sentiment Analysis and CandleStick Chart Representation. *Sensors* **2021**, *21*, 7957. [CrossRef] [PubMed]
11. Jaggi, M.; Mandal, P.; Narang, S.; Naseem, U.; Khushi, M. Text Mining of Stocktwits Data for Predicting Stock Prices. *Appl. Syst. Innov.* **2021**, *4*, 13. [CrossRef]
12. Khoa, N.L.D.; Sakakibara, K.; Nishikawa, I. Stock price forecasting using back propagation neural networks with time and profit based adjusted weight factors. In Proceedings of the 2006 SICE-ICASE International Joint Conference, Busan, Korea, 18–21 October 2006; pp. 5484–5488.
13. Souma, W.; Vodenska, I.; Aoyama, H. Enhanced news sentiment analysis using deep learning methods. *J. Comput. Soc. Sci.* **2019**, *2*, 33–46. [CrossRef]
14. Devin, J.; Chang, M.; Lee, L.; Toutanova, K. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Minneapolis, MN, USA, 2–7 June 2019; pp. 4171–4186.

15. Araci, D. FinBERT: Financial Sentiment Analysis with Pre-Trained Language Models. University of Amsterdam. Available online: https://arxiv.org/pdf/1908.10063.pdf (accessed on 20 April 2022).
16. Mohan, S.; Mullapudi, S.; Sammeta, S.; Vijayvergia, P.; Anastasiu, C. Stock Price Prediction Using News Sentiment Analysis. In Proceedings of the 2019 IEEE Fifth International Conference on Big Data Computing Service and Applications (BigDataService), Newark, CA, USA, 4–9 April 2019; pp. 205–208.
17. Malo, P.; Sinha, A.; Korhonen, P.; Wallenius, J.; Takala, P. Good debt or bad debt: Detecting semantic orientations in economic texts. *J. Assoc. Inf. Sci. Technol.* **2013**, *65*, 782–796. [CrossRef]
18. Ding, X.; Zhang, Y.; Liu, T.; Duan, J. Using Structured Events to Predict Stock Price Movement: An Empirical Investigation. In Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), Doha, Qatar, 25–29 October 2014; pp. 1415–1425. Available online: http://emnlp2014.org/papers/pdf/EMNLP2014148.pdf (accessed on 20 April 2022).
19. Remy, P.; Ding, X. Financial News Dataset from Bloomberg and Reuters. Available online: https://github.com/philipperemy/financial-news-dataset (accessed on 20 April 2022).
20. S&P Dow Jones Indices. S&P U.S. Indices Methodology. Available online: https://www.spglobal.com/spdji/en/indices/equity/sp-500/# (accessed on 20 April 2022).
21. Liu, Q.; Cheng, X.; Su, S.; Zhu, S. Hierarchical Complementary Attention network for Predicting Stock Price Movements with News. In Proceedings of the 27th ACM International Conference on Information and Knowledge Management (CIKM '18). Association for Computing Machinery, New York, NY, USA, 22–26 October 2018; pp. 1603–1606.
22. Grandini, M.; Bagli, E.; Visani, G. Metrics for Multi-Class Classification: An Overview. *arXiv* **2008**, arXiv:2008.05756.
23. Rufibach, K. Use of Brier score to assess binary predictions. *J. Clin. Epidemiol.* **2010**, *63*, 938–939. [CrossRef] [PubMed]
24. Chicco, D.; Warrens, M.J.; Jurman, G. The Matthews Correlation Coefficient (MCC) is More Informative Than Cohen's Kappa and Brier Score in Binary Classification Assessment. *IEEE Access* **2021**, *9*, 78368–78381. [CrossRef]
25. Bergmeir, C.; Hyndman, R.; Koo, B. A note on the validity of cross-validation for evaluating autoregressive time series prediction. *Comput. Stat. Data Anal.* **2018**, *120*, 70–83. [CrossRef]
26. Hull, J.C. *Options Futures and Other Derivatives*; Pearson Education India: Chennai, India, 2003.
27. Hastie, T.; Tibshirani, R.; Friedman, J. *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*, 2nd ed.; Springer: New York, NY, USA, 2009.
28. Berrar, D. Cross-Validation. Available online: https://www.researchgate.net/publication/324701535_Cross-Validation (accessed on 20 April 2022).
29. Lahiri, K.; Yang, L. Forecasting binary outcomes. In *Handbook of Economic Forecasting*; Elsevier: Amsterdam, The Netherlands; Volume 2, pp. 1025–1106.