Assessing the Correlation between News Sentiment and Stock Price Fluctuations

Duygu Bayram*
d.bayram.1@student.rug.nl

Konstantin Chernyshev*
k.chernyshev@student.rug.nl

Sara Nabhani*
s.nabhani@student.rug.nl

Hulma Naseer*
h.naseer.1@student.rug.nl

Abstract

Stock market prediction is an active research area. It can help investors evaluate the expected return on their investment and there is a great benefit in automating this process. One of the sources of information that can be used for prediction is financial news which covers various aspects of the market and economic situation. The sentiment about the company in financial news can be used as a predictor of its stock price increase or decrease. In this paper, we build a sentiment analysis model and investigate whether there is a correlation between historical news sentiment classified by our model and real stock price changes. A significant correlation suggests using financial news sentiment as a factor in determining trends for the stock market.

1 Introduction

The earliest function of the stock market dates back to the Dutch East India Company in the 1600s (Petram and others, 2011). With the country heavily relying on ship trade (EH, 2023), merchants turned to citizens to acquire more funding for their journeys in return for sharing a portion of the profits. In the modern world, investors buy shares, or stocks, from companies, essentially funding them in exchange for a portion of their profits (Sincere and others, 2014).

However, this process requires investors to predict how successful a certain company will be, and how much return they will be able to get for their investments in the form of profits (Sincere and others, 2014). Because the success of a company often relies on the popularity of their products, public opinion plays an important role in this prediction (Mishev et al., 2020). The basic premise is that as popularity grows, so does the stock value. Similarly, if a company starts losing public interest, the value of the stocks could start to decrease, prompting the investors to sell their stocks to profit from their initial buying price. Thus, the trend in values is an additional factor for investment decisions.

Such prediction can be performed intuitively, by following the news, discussions, and trends in the current market prices. However, one can also turn to statistics to follow the trends in stock values. While there are rudimentary ways of approaching this issue, such as feature engineering (Pang et al., 2002; **?**; **?**), or lexicon-based approaches (Stone et al., 1962; **?**); it is also possible to do it with machine learning.

The practice of machine learning has spread across our lives over the past decade. It has found its use in genome studies (Yip et al., 2013), in physical modeling (Willard et al., 2020), in natural language processing (Olsson, 2009), in image processing (Sungheetha and Sharma, 2021), and many other fields. With its strong foundation in statistics and data handling, it is no surprise that it has also found its place in the analysis of economic practices (Mishev et al., 2020).

(Mishev et al., 2020) have reviewed the use of machine learning applications to analyze news sentiments for future stock price prediction. (Wang et al., 2015) employ Naive Bayes, Decision Trees, and SVM models to perform sentiment analysis on stock tweets.

[0] Authors alphabetically sorted

Another study makes use of regression models and various feature extraction methods on news (Atzeni et al., 2017).

As such, a use of machine learning relevant to our project is its application in natural language processing. While language is difficult to define without reduction, one can think of it as a way for humans to store, exchange, and transfer information as it is a crucial factor in these functions (Buckley et al., 2005). Today, with the increasing popularity of the internet (Hoffman et al., 2004), there is a vast amount of textual data of people engaging in opinion sharing. Considering this, automated methods of collecting and analyzing this data could be helpful for gaining insight into public opinion and human-related events.

One way of assessing public opinion on a topic is the use of sentiment analysis. Sentiment analysis, also sometimes called opinion mining (Nanli et al., 2012), can be treated as a supervised classification task. In the most basic sense, the model is first trained with data that is annotated with positive, negative, or neutral labels, then, this model can be used to label unseen data with the same sentiments. (Chakraborty et al., 2022) have employed classical machine learning algorithms such as Naive Bayes (NB) (Hand and Yu, 2001), Support Vector Machine (SVM) (Cortes and Vapnik, 1995), Adaboost (Freund et al., 1999), Decision Trees (DT) (Quinlan, 1986), and Random Forest (RF) (Breiman, 2001) for sentiment analysis in their work. (Ain et al., 2017) mention the use of K-Nearest Neighbours (KNN) (Peterson, 2009) in their models review. In more recent years, transformers such as BERT (Devlin et al., 2018) have been adopted for the task (Habimana et al., 2020), showing promising results.

This technology could be beneficial in the context of trade. As discussed, stock prices are influenced by public opinion (Mishev et al., 2020); however, without this technology, investors have to follow trends based on their intuition and conscious follow-through of news. This takes considerable time and effort, furthermore, intuition is prone to error and bias (Kahneman and Tversky, 1977). Using sentiment analysis to gauge public opinion to predict stock market behavior could be a helpful development to automate this process.

However, this idea would need to be tested. There are many factors that can influence stock market prices, such as order flow (Gerig, 2008), corporate policy-making, regulations, and macroeconomic factors (Cutler et al., 1988); public opinion and interest is only one of them. In fact, (Cutler et al., 1988) argue that news are not sufficient to explain share price fluctuations. We aim to investigate the extent to which public opinion correlates with the changes in stock values by conducting sentiment analysis on news data and performing correlation analysis between these expressed sentiments and historical market prices.

2 Method

In this project, we train classical machine learning models (as detailed in Section 2.3.1) and transformers (Section 2.3.2) with a labeled dataset from the Huggingface Hub to classify the sentiment of stock market news between the years of 2008-2020. Next, to investigate whether our proposal is a viable approach, we look at the correlation between the detected sentiments of the news articles and the real stock price percentage increase or decrease on the corresponding dates. That is, if the news sentiment and the stock market behavior are correlated, models which predict future stock prices based on similar news data can be built. We approach the problem in three steps:

1. We train classical machine learning models as a baseline on an available dataset to classify our news data.

2. We fine-tune pre-trained language models on the same available dataset to classify our news data.

3. We test if there is a significant correlation between news sentiments and stock price changes.

2.1 Data

Our task requires three different sets of data. To represent the public sentiment of the market, we use the US Equities News Dataset[1] (USEN) which serves as an archive

[1] https://www.kaggle.com/datasets/gennadiyr/us-equities-news-data?resource=download

of NYSE/NASDAQ stock exchange news between the years of 2008 and 2020. To find correlations between the classified sentiment of a given time duration and the actual stock prices, we use the Yahoo! Finance API[2] (yfinance) to pull market price data for the same years. We also use a separate labeled dataset, titled Sentiment Analysis Data[3], for training and analyzing the performance of the sentiment analysis models.

2.1.1 US Equities News

The USEN dataset is collected from investing.com, a financial news website. The dataset includes a ticker for each article to identify particular stocks, the news title, the news body, the date of the article, the attribution to the author, a link to the article, and the article ID as collected from the website. It is important to note that we do not have *time of the day* information about publishing.

We filtered the given dataset to select the top 10 companies (10 unique tickers) by news number. The dataset used consists of a total of 35,467 headlines between the dates of 2008.12.19 and 2020.01.23 (2856 trading days). For our purposes, we only use the ticker, headline, news body, and the date information. An overview of the corresponding number of news and number of days the company is mentioned in news can be viewed in Table 1.

2.1.2 Yahoo Finance

To prepare the Yahoo Finance (YF) dataset, we pull the relevant data from Yahoo Finance using the open-source yfinance library. The final dataset consists of 28,124 entries of opening and closing prices, as well as the highest and lowest values, for the dates and the tickers that are available to us in the USEN dataset. Table 1 shows the total number of days where pricing information is available for each ticker.

	#News	**#Prices**	**#Days**
AAPL	6626	2856	1353
BAC	4621	2856	1762
BA	4028	2856	1235
GOOGL	3567	2856	1176
MSFT	3361	2856	1185
GS	3158	2856	1132
AMZN	2972	2856	984
TSLA	2553	2420	915
TGT	2401	2856	661
INTC	2180	2856	948

Table 1: USEN-YF dataset statistics. Total number of News articles (#News), Number of Days with stock price data available (#Prices) and Number of Days with News articles available (#Days) for 10 Tickers.

2.1.3 Sentiment Analysis Data

Since the USEN dataset does not contain sentiment labels, we need a separate dataset for training our sentiment analysis models. For this purpose, we chose an available dataset as close to the domain as possible. The Sentiment Analysis Data[4] combines the Financial Phrasebank[5] dataset from the Huggingface Hub and a Financial Text[6] dataset from Kaggle. The data is already split into train and test sets of 4551 and 506 samples respectively. Each news item contains a positive (2), neutral (1), or negative (0) label.

2.2 Evaluation

Given the news sentiments for a certain company calculated using our sentiment analysis model on a certain day and the corresponding percentage increase in stock price, we aim to find the correlation between the two over the course of almost 12 years of data.

Correlation between two variables is a statistical measure that indicates the extent to which two or more variables fluctuate in relation to each other (Wigmore, 2020). A positive correlation shows that when one variable increases, the other also increases, and vice versa. A correlation is negative if the increase

[2] https://pypi.org/project/yfinance/

[3] https://huggingface.co/datasets/nickmuchi/financial-classification

[4] https://huggingface.co/datasets/nickmuchi/financial-classification

[5] https://huggingface.co/datasets/financial_phrasebank

[6] https://www.kaggle.com/datasets/percyzheng/sentiment-classification-selflabel-dataset

in one variable is accompanied by a decrease in the other.

Here, we will find the strength of the correlation between our two variables i.e. news sentiment and stock price change. A positive relationship is desired in order for being able to predict the stock market changes using the news sentiment. Treating our sentiment data as categorical and continuous, we find two different correlation values which are further discussed in experiments.

2.3 Sentiment Analysis Model

Since USEN dataset has no sentiment labels, we need to train a separate model for labeling. For this purpose, we used the Financial Classification dataset mentioned before. We considered two approaches: 'Classical' Machine Learning models as a baseline to be compared to the performance of the advanced Transformers Neural Networks, described below. LinearSVM showed the best performance as a baseline model, although the experimented transformer models performed better.

2.3.1 Classical Models

To select a baseline model to experiment with, we first performed a rough model search, training some widely-used models against some popular vectorization methods. All models were trained with 4-fold cross-validation and f1-macro out-of-fold evaluation.

The experiments were carried out using the scikit-learn (Pedregosa et al., 2011) library. Initial searches were conducted using default parameters. The following classifiers were employed: Logistic Regression, SVM, Random Forest, Gradient Boosting, Naive Bayes, and K-Nearest Neighbors, in conjunction with Count Vectorizer, Tf-Idf Vectorizer, and Hashing Vectorizer.

As indicated in Table 2, the best model and vectorizer combination was LinearSVM with Tf-Idf vectorizer, which was also one of the fastest models. As a result, this combination was chosen as the baseline model for further hyperparameter tuning in subsequent experiments.

Linear Support Vector Machines (Linear SVM) (Cortes and Vapnik, 1995) and Term Frequency-Inverse Document Frequency (Tf-Idf) vectorization (Jones, 1972) are two popular algorithms used in natural language processing for the classification tasks. Linear SVM operates by finding a hyperplane that maximizes the distance between samples of two classes. Mathematically, given a feature space of text representation, SVM finds the weight vector, w, and bias term, b, that satisfies the constraint $w^T x_i + b \geq 1$ for the first class samples and $w^T x_i + b \leq -1$ for the second. For multi-class setup we used One-vs-Rest approach, where, a separate SVM model is trained for each class against all other classes. During prediction, each separate model is used, and the class with the highest predicted probability is selected as the final prediction. Tf-Idf vectorization represents each sample as a vector of term frequencies, weighted by the inverse of the term's document frequency, so that terms that appear frequently in many documents are given a lower weight than terms that appear infrequently. Mathematically, Tf-Idf for a term t in document d is given by $Tf(t,d) \times \log(\frac{N}{df(t)})$, where $Tf(t,d)$ is the term frequency of t in d, N is the number of documents, and $df(t)$ is the number of documents that contain the term t.

Next, we optimized the hyperparameters of the selected model. This was achieved through the use of two libraries, BAYESIAN-OPTIMIZATION (Fernando Nogueira, 2017) and HYPEROPT (Bergstra et al., 2013), which employ two common approaches to hyperparameter optimization. The first library implements classical Bayesian Global Optimization with Gaussian Processes (BGO-GP), as outlined in (Schweidtmann et al., 2020). It combines the strengths of Bayesian optimization and Gaussian process regression. The algorithm uses a Gaussian process model to approximate the unknown function, where the prior distribution is updated with each new sample point, resulting in a posterior distribution that provides a probabilistic estimate of the function. The algorithm then uses an acquisition function to choose the next sample maximizing the probability of finding the global optimum. The second one uses Tree of Parzen Estimators (TPE) algorithm (Bergstra et al., 2011) - a Bayesian Optimization algorithm utilizing decision trees. In the first step, it estimates the underlying probability density function (pdf)

	Count	Tf-Idf	Hash	FastText
LogReg	0.688	0.638	0.622	0.444
SVM	0.590	0.603	0.598	0.620
LinearSVM	0.676	**0.691**	0.688	0.572
RandomForest	0.605	0.598	0.529	0.477
GradientBoosting	0.662	0.646	0.656	0.587
NaiveBayes	0.627	0.415	*(not applicable)*	*(not applicable)*
K-Neighbors	0.453	0.583	0.539	0.558

Table 2: 7 classical ML models against 4 popular vectorizers trained using default parameters and basic preprocessing. Out-of-Fold f1-macro scores for 4-folds Cross Validation are reported. Bold font indicates the best result.

Model	**Trials**	**f1-macro**
Bayesian	128	**0.737**
Hyperopt	64	0.729

Table 3: Hyperparameters Optimization of the Tf-Idf vectorizer with Linear SVM models using two different searching methods: Global Bayesian Optimization and Tree Parzen Estimator. The f1-macro scores on the test set are reported. Bold font indicates the best result.

using Parzen Estimators method. Then, a decision tree is built to split the search space into regions with different levels of expected improvement. At each iteration, the algorithm samples a new point from the highest-improvement region to update the model.

The models were trained and tuned with 4-folds cross-validation on the train data. The 2 corresponding optimization methods result are available in Table 3. The best model was found using the Bayesian optimization library, achieving an f1-macro score of 0.737 on the test set.

2.3.2 Transformers

We used language models that are trained based on the transformer architecture (Vaswani et al., 2017). The Transformer model is a State of the Art deep learning architecture for natural language processing (suitable for a wide range of tasks).

Transformers has an Encoder-Decoder architecture, the encoder consists of multiple encoding layers that are fed the input to process it consecutively to extract contextual information and then outputs the extracted features encodings. This output is then fed into the decoder that consists of multiple decoder layers and gets processed using the contextual information to generate output sequence. This architecture employs a self-attention mechanism in each encoder and decoder layer. The self-attention mechanism calculates a weighted sum of all layer inputs generating each input representation. Then these representations are passed through a feedforward neural network. The model is trained using maximum likelihood estimation with a cross-entropy loss function. This architecture allows the model to capture long-range dependencies and context-dependent representations of the input tokens. Also, the Transformer models are quite efficient, due to efficient parallel computation. All this made it possible to create large Language Models having a good 'memory', which are trained on large corpora of texts, and later require only small finetuning for a particular task.

We conducted experiments on some of the state of the art language models:

- BERT (Devlin et al., 2018) - one of the first Language Models based on a bidirectional Transformer training. The model was trained on Next Sentence Prediction (NSP) and Masked Language Modeling (MLM) tasks using 16 GB of text data. Using Masked Language Modeling (MLM), 15% of each words sequence fed into the model is replaced with the [MASK] token, then based on the context represented by the other non-masked words the model attempts to predict the masked words. In Next Sentence Pre-

Base Model	Model on HF	Params	f1-macro
bert-base-uncased	ML-ns-bert-base-uncased	109M	0.852
roberta-base	ML-ns-roberta-base	124M	0.875
finbert	ML-ns-finbert	124M	0.880
twitter-roberta-base-sentiment	ML-ns-twitter-roberta-base-sentiment	124M	**0.885**

Table 4: Fine-tuning Language Models. The first column shows the model being fine-tuned.The second column shows the name of fine-tuned model uploaded to HuggingFace under saranabhani/{Model on HF}. The third column shows the number of model parameters. Models trained for 4 epochs with a learning rate of 5e-6 and weight decay of 0.01 and a batch size of 32. The f1-macro scores on the test set are reported. Bold font indicates the best result.

diction (NSP) training, the model is fed pairs of sentences and attempts to predict whether the second sentence in each pair is a subsequent to the first sentence. The model training aims at minimizing the combined loss of Next Sentence Prediction (NSP) and Masked Language Modeling (MLM).

- RoBERTa (Liu et al., 2019) - a robustly optimized version of BERT. Trained for a longer time on larger batches and learning rate on the Masked Language Modeling, however, with the objective of Next Sequence Prediction removed. This has resulted in a better performance on the Masked Language Modeling task as compared to that of BERT. In addition, it achieved high improvement on downstream tasks performance.

- FinBert (Liu et al., 2021) - a BERT base model that was trained on a large financial corpus and fine-tuned for financial sentiment classification.

- Twitter-roBERTa-base for Sentiment Analysis (Barbieri et al., 2020) - a RoBERTa base model that was trained on 58M tweets and fine-tuned for sentiment analysis task as part of creating a unified evaluation framework TWEETEVAL.

We further trained the above-listed models on our Sentiment Analysis data and fine-tuned them to do the news sentiment classification task.

All of the experimented models performed well on the test data, with f1-macro score greater than 0.85. The f1-macro scores per model are available in Table 4. FinBert was trained on a financial corpus and Twitter-roBERTa-base models was trained and fine-tuned to do sentiment analysis task, as a result they performed better than BERT and RoBERTa. However, the highest score was achieved by the Twitter-roBERTa-base model with 0.885 f1-macro score. Thus, it is the one we used to predict the sentiment of the US Equities News data.

2.4 Experiments

Correlation tells us about the association between the two variables. The two variables here are the sentiment of news articles and the percentage change in the stock price on the day. For each ticker, we only consider days where we have both news data and stock price data available and merge the two.

After we have trained a suitable model for sentiment analysis, we use this model to annotate our US Equities news data with sentiments for 10 tickers. We use both the headline of the news and the content to predict a single sentiment value for the news i.e. either 'positive', 'negative', or 'neutral'. It is important to note that on a single day, a single company can have multiple news articles hence multiple sentiment values. To find a single sentiment value for a certain day, we encode our sentiments as 0 for 'neutral', 1 for 'positive', and -1 for 'negative'. Then, we take the mean across all news for a single day and a single company. For example, if a company has 4 news articles in a single day with sentiment scores encoded as -1, 1, 0, and 1 the mean will be 0.25 which is slightly positive as we have two positive news articles. So the mean for each day will be a value between -1 and 1 always with all positive news articles being 1 and all negative news articles being -1. We can treat this as a continuous variable representing the sentiment. We also created a categorical variable using this mean score where anything above 0 gets a positive class, 0 is neutral, and below zero is negative.

For the stock price changes, we use two approaches to calculate the percentage change (increase or decrease). For a single day, firstly we calculate the percentage change of the closing price of the day from the opening price of the same day. Secondly, we calculate the percentage change of the high price from the high price of the same company from the previous day. The closing price of one day is not the same as the starting price of the next day and both approaches make sense. This is a continuous variable.

2.4.1 Spearman's Rho Correlation

In order to find the correlation between said variables, we start by considering the sentiment mean as a continuous variable. The method we will use here is Spearman's Rho correlation (Bhat, 2023). The difference between Pearson's R and Spearman's Rho is that Pearson's R looks for a linear relationship between the two variables with a specific slope (rate of increase). In turn, Spearman's Rho looks for any monotonic relation. Since we do not expect a constant rate of increase, we go with Spearman's correlation latter. The formula is the following:

$$\rho = 1 - \frac{6 \sum d_i^2}{n(n^2-1)}$$

where d is the pairwise distances of the ranks of the variables and n is the number of samples. For each of the 10 tickers, we find Spearman's Rho coefficient for the entire data.

A positive or negative (non-zero) Spearman's Correlation would suggest that there is a correlation between sentiment score and daily stock price increase.

2.4.2 Kruskal-Wallis Test

The other method we used for checking correlation is Kruskal-Wallis Test (or one-way ANOVA on ranks or non-parametric ANOVA) (Kruskal and Wallis, 1952). This method is used to test if there is a significant difference between the stock price changes among positive, negative, and neutral sentiments. The sentiment here is the categorical variable we created. This test is a non-parametric alternative to the one-way ANOVA test and is suitable when the data does not follow a normal distribution for each category which after inspection turned out to be the case with our data. The Kruskal-Wallis test statistic is calculated using the following formula:

$$H = \frac{12}{n(n+1)} \sum_{i=1}^{k} n_i (R_i - \frac{n+1}{2})^2$$

where k is the number of groups, n is the number of samples, n_i is the number of samples in group i, and R_i = the sum of the ranks for group i.

A significant result from the Kruskal-Wallis test would suggest that there are differences in the mean values of the daily stock price percentage change among negative, neutral, and positive groups.

The results from Spearman's Rho correlation and the Kruskal-Wallis test will be discussed and interpreted in the next section to determine the correlation between the sentiment scores and stock price changes.

3 Results

We conducted experiments using two approaches of stock price change calculation: the changes between the Opening and the Closing price (In-Day) and between the High price of the current day and the previous day (WithPrev). Each of these is calculated as a percentage. These values are compared to 1) the mean sentiment of each day (if there are several news, their sentiment labels are averaged). The calculated final sentiments will take values $[-1; +1]$, with -1 as purely negative, 0 as neutral, and +1 purely positive 2) the categorical variable created from mean sentiment with anything above 0 being positive, 0 being neutral and anything below zero being negative. Each of the Tickers was considered independently.

The results of the Spearman's Rho correlation and the Kruskal-Wallis test are shown in Table 5 and Table 6 respectively. The

	In-Day	**With-Prev**
AAPL	0.134	0.218
AMZN	0.102	0.161
BA	0.070	0.124
BAC	0.125	0.283
GOOGL	0.064	0.095
GS	0.022	0.151
INTC	0.145	0.257
MSFT	0.156	0.196
TGT	0.012	0.108
TSLA	0.074	0.183

Table 5: Spearman's correlation between mean daily sentiment label (-1 negative, 0 neutral, +1 positive) and stock price changes in percentage. The increase is measured between the open and close price of the same day (In-Day) or the current and previous day's highest price (With-Prev).

	In-Day	**With-Prev**
AAPL	*<0.001*	*<0.001*
AMZN	0.001	*<0.001*
BA	0.017	*<0.001*
BAC	*<0.001*	*<0.001*
GOOGL	0.180	0.005
GS	0.505	*<0.001*
INTC	*<0.001*	*<0.001*
MSFT	*<0.001*	*<0.001*
TGT	0.964	0.013
TSLA	0.019	*<0.001*

Table 6: p-values of the Kruskal–Wallis Test on mean stock value increase in percentage between 3 groups: negative, neutral, positive sentiment calculated from the mean sentiment of the day. The increase is measured between the open and close price (In-Day) or using the current and previous day's high price (With-Prev). Italic font indicates a significant difference.

Spearman's Rho coefficient ranges from 0.012 to 0.283, indicating a weak, but stably positive correlation between the sentiment means and stock price change. An interesting thing to note is that the correlation is consistently higher when we consider With-Prev stock price changes as compared to In-Day price changes.

Similarly, the Kruskal-Wallis test shows a significant difference in stock price changes across the sentiment groups (p<0.001) for the majority of the tickers, particularly for the With-Prev stock price change. This suggests that there is a significant association between news sentiment on a day and the stock price change on the same day.

At the same time, we can see that 2 tickers GOOGL and TGT are seemingly less correlated with the news sentiment when considering Spearman's Rho. Similarly, for these stocks, the hypothesis of the dependence of the mean stock price change on the news sentiment is not rejected, suggesting that there is no significant difference among sentiment groups.

4 Conclusion

The stock market is subject to fluctuations due to diverce factors such as company growth, current events, social and political situations. One aspect of these factors is the daily news sentiment regarding a particular company. We trained a transformer-based NLP model for classifying the news about company into one of three categories: positive, neutral, or negative. Using the average daily sentiment about a company, we conducted a correlation analysis between the sentiment and the stock price changes. We found a significant correlation for the majority of the tickers into consideration. This implies that the tone of company news, particularly its positive or negative connotation, can serve as an additional predictor in the stock price changes forecasting.

References

Qurat Tul Ain, Mubashir Ali, Amna Riaz, Amna Noureen, Muhammad Kamran, Babar Hayat, and A Rehman. 2017. Sentiment analysis using deep learning techniques: a review. *International Journal of Advanced Computer Science and Applications*, 8(6).

Mattia Atzeni, Amna Dridi, and Diego Reforgiato Recupero. 2017. Fine-grained sentiment analysis on financial microblogs and news headlines. In *Semantic Web Challenges: 4th SemWebEval Challenge at ESWC 2017, Portoroz, Slovenia, May 28-June 1, 2017, Revised Selected Papers*, pages 124–128. Springer.

Francesco Barbieri, Jose Camacho-Collados, Luis Espinosa Anke, and Leonardo Neves. 2020. TweetEval: Unified benchmark and comparative evaluation for tweet classification. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 1644–1650, Online, November. Association for Computational Linguistics.

James Bergstra, Rémi Bardenet, Yoshua Bengio, and Balázs Kégl. 2011. Algorithms for hyper-parameter optimization. In J. Shawe-Taylor, R. Zemel, P. Bartlett, F. Pereira, and K.Q. Weinberger, editors, *Advances in Neural Information Processing Systems*, volume 24. Curran Associates, Inc.

James Bergstra, Daniel Yamins, and David Cox. 2013. Making a science of model search: Hyperparameter optimization in hundreds of dimensions for vision architectures. In Sanjoy Dasgupta and David McAllester, editors, *Proceedings of the 30th International Conference on Machine Learning*, volume 28 of *Proceedings of Machine Learning Research*, pages 115–123, Atlanta, Georgia, USA, 17–19 Jun. PMLR.

Adi Bhat. 2023. Spearman correlation coefficient: Formula + calculation, Jan.

Leo Breiman. 2001. Random forests. *Machine learning*, 45:5–32.

Peter J Buckley, Martin J Carter, Jeremy Clegg, and Hui Tan. 2005. Language and social knowledge in foreign-knowledge transfer to china. *International Studies of Management & Organization*, 35(1):47–65.

Partha Chakraborty, Farah Nawar, and Humayra Afrin Chowdhury. 2022. Sentiment analysis of bengali facebook data using classical and deep learning approaches. In *Innovation in Electrical Power Engineering, Communication, and Computing Technology: Proceedings of Second IEPCCT 2021*, pages 209–218. Springer.

Corinna Cortes and Vladimir Vapnik. 1995. Support-vector networks. *Machine learning*, 20(3):273–297.

David M Cutler, James M Poterba, and Lawrence H Summers. 1988. What moves stock prices?

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. BERT: pre-training of deep bidirectional transformers for language understanding. *CoRR*, abs/1810.04805.

EH. 2023. The dutch economy in the golden age (16th – 17th centuries). *Economic History Encyclopedia*.

PhD Fernando Nogueira. 2017. Bayesianoptimization. `https://github.com/fmfn/BayesianOptimization`.

Yoav Freund, Robert Schapire, and Naoki Abe. 1999. A short introduction to boosting. *Journal-Japanese Society For Artificial Intelligence*, 14(771-780):1612.

Austin Gerig. 2008. A theory for market impact: How order flow affects stock price. *arXiv preprint arXiv:0804.3818*.

Olivier Habimana, Yuhua Li, Ruixuan Li, Xiwu Gu, and Ge Yu. 2020. Sentiment analysis using deep learning approaches: an overview. *Science China Information Sciences*, 63:1–36.

David J Hand and Keming Yu. 2001. Idiot's bayes—not so stupid after all? *International statistical review*, 69(3):385–398.

Donna L Hoffman, Thomas P Novak, and Alladi Venkatesh. 2004. Has the internet become indispensable? *Communications of the ACM*, 47(7):37–42.

Karen Spärck Jones. 1972. A statistical interpretation of term specificity and its application in retrieval. *Journal of Documentation*, 28:11–21.

Daniel Kahneman and Amos Tversky. 1977. Intuitive prediction: Biases and corrective procedures. Technical report, Decisions and Designs Inc Mclean Va.

William H. Kruskal and W. Allen Wallis. 1952. Use of ranks in one-criterion variance analysis. *Journal of the American Statistical Association*, 47(260):583–621.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. cite arxiv:1907.11692.

Zhuang Liu, Degen Huang, Kaiyu Huang, Zhuang Li, and Jun Zhao. 2021. Finbert: A pretrained financial language representation model for financial text mining. In *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence*, IJCAI'20.

Kostadin Mishev, Ana Gjorgjevikj, Irena Vodenska, Lubomir T Chitkushev, and Dimitar Trajanov. 2020. Evaluation of sentiment analysis in finance: from lexicons to transformers. *IEEE access*, 8:131662–131682.

Zhu Nanli, Zou Ping, LI Weiguo, and Cheng Meng. 2012. Sentiment analysis: A literature review. In *2012 International Symposium on Management of Technology (ISMOT)*, pages 572–576. IEEE.

Fredrik Olsson. 2009. A literature survey of active machine learning in the context of natural language processing.

Bo Pang, Lillian Lee, and Shivakumar Vaithyanathan. 2002. Thumbs up? sentiment classification using machine learning techniques. *arXiv preprint cs/0205070*.

F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. 2011. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830.

Leif E Peterson. 2009. K-nearest neighbor. *Scholarpedia*, 4(2):1883.

Lodewijk Otto Petram et al. 2011. *The world's first stock exchange: how the Amsterdam market for Dutch East India Company shares became a modern securities market, 1602-1700*. Ph.D. thesis, Universiteit van Amsterdam [Host].

J. Ross Quinlan. 1986. Induction of decision trees. *Machine learning*, 1:81–106.

Artur M. Schweidtmann, Dominik Bongartz, Daniel Grothe, Tim Kerkenhoff, Xiaopeng Lin, Jaromil Najman, and Alexander Mitsos. 2020. Global optimization of gaussian processes.

Michael Sincere et al. 2014. *Understanding stocks*. McGraw-Hill Education.

Philip J Stone, Robert F Bales, J Zvi Namenwirth, and Daniel M Ogilvie. 1962. The general inquirer: A computer system for content analysis and retrieval based on the sentence as a unit of information. *Behavioral Science*, 7(4):484.

Akey Sungheetha and Rajesh Sharma. 2021. 3d image processing using machine learning based input processing for man-machine interaction. *Journal of Innovative Image Processing (JIIP)*, 3(01):1–6.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *CoRR*, abs/1706.03762.

Gang Wang, Tianyi Wang, Bolun Wang, Divya Sambasivan, Zengbin Zhang, Haitao Zheng, and Ben Y Zhao. 2015. Crowds on wall street: Extracting value from collaborative investing platforms. In *Proceedings of the 18th ACM Conference on Computer Supported Cooperative Work & Social Computing*, pages 17–30.

Ivy Wigmore. 2020. What is correlation?: Definition from techtarget, Aug.

Jared Willard, Xiaowei Jia, Shaoming Xu, Michael Steinbach, and Vipin Kumar. 2020. Integrating physics-based modeling with machine learning: A survey. *arXiv preprint arXiv:2003.04919*, 1(1):1–34.

Kevin Y Yip, Chao Cheng, and Mark Gerstein. 2013. Machine learning and genome annotation: a match meant to be? *Genome biology*, 14(5):1–10.