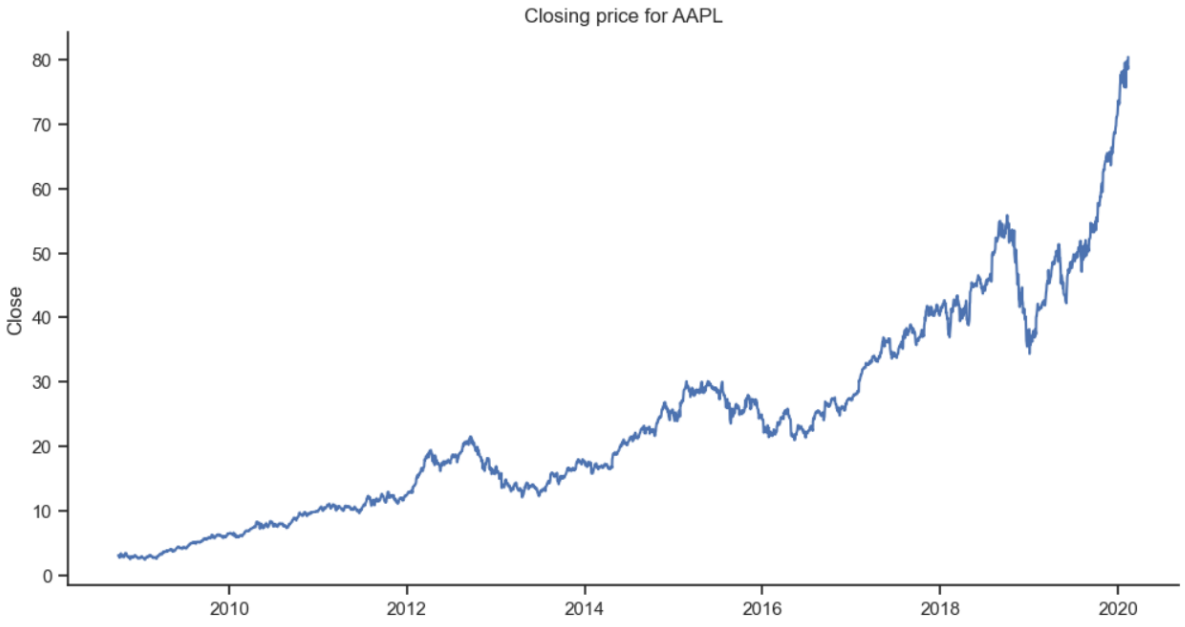
Natural Language Processing

Report FinCon4U

Predicting stock price movements based on news



Team 07: Camila Matoba, Marta Nowakowicz & Nadine van Raaij

16-10-2022

Table of Contents

Executive summary

The purpose of this project was to predict stock price movements of Apple Inc. based on news articles. We used the BoW approach and sentiment analysis of titles of news articles related to Apple. The best result was achieved with GaussianNB Classifier with bigrams and sentiment analysis, with weighted precision of 0.641 as the main performance metric. In particular, section four contains the main words that influence these results. The last section refers to limitations and possible future considerations in case more in-depth analysis is of interest.

Pre-processing

1. Document Filtering: Relevant corpus for the experiment

The data used during the research is a news archive representing years 2008-2020 of the “US equities publicly traded on NYSE/NASDAQ which still has a price higher than 10$ per share”. The data is open-sourced and available on Kaggle.

As the goal of the project was to predict the closing price of Apple shares, we decided that the relevant news is only the one related to Apple. Therefore to create the corpus we filtered news with the ticker “AAPL”. This subset represents around 9.3% of the total news dataset, which sums up to 19985 news articles. An assumption that is made here is that the news not related to Apple do not affect the stock price of AAPL or do not affect the stock price the same as the news articles related to Apple do.

Furthermore, the dates of the AAPL stock price and the news articles were set in parallel. In other words, we assume that news only affects the short-term fluctuations of the stock price. This implies that the news items affect the next business day. In case of days which happen after not-business days (e.g Mondays or holidays relevant to the Stock Market) we take into account all news published since the last business day.

A majority of research in stock predictions based on news articles use only the title of these news articles and Nemes and Kiss (2021) proved that the sentiment of news article titles only have significant correlation with the stock market value changes. However, Liu et al. (2018) found in their experiment that including the content of news articles significantly improves the accuracy of stock predictions. Furthermore, they successfully captured the complementary information in the news titles and content by designing a new measurement, namely score-inverse similarity (SIS), for calculating attention weights. Therefore we decided to try this as well. Unfortunately we did not succeed in this because of lack of computational power. In the end, we decided to run on the titles of the news instead of the full content.

2. Target variable labels

The target variable was based on a condition - whether the closing Stock price of AAPL is higher or lower than the opening price. We assign label “1” if the closing price is higher than the opening price and link them to news published in the previous business day. In all other scenario’s the label “0” is the given. As we have two labels, the problem becomes a binary classification task.

3. Kaggle corpus and Pre-processing

The corpus from Kaggle has different approaches in pre-processing this data. The one we are using for this project has notedly removed punctuation. Furthermore, no html tags were seen.

Tetlock (2007) proposes a simple method on tokenizing the newspaper articles into numbers. He does so by counting the number of words into 77 categories according to Harvard IV-4 dictionary for each available day. In an example case on Kaggle, they implement a similar method with the use of two available dictionaries of Harvard IV-4 or Loughran and McDonald Financial Sentiment, the latter being oriented for financial news. He also created dummy variables for the holidays and weekend days in the U.S. to control for other potential return anomalies, which we did for the same reason as well.

Others from the kaggle competition limit themselves more to general pre-processing steps orientated at textual data like equalising upper and lowercase letters.

Because it would have been possible that certain authors only write articles with an extremely high negative or positive sentiment which would influence the model unjustified, we have investigated the sentiment distribution among authors of their articles. It was found that for the top 20 authors (representing more than 95% of the total articles written) a normal distribution could be drawn for the sentiment given to their articles. Only an insignificant part of the data got an extremely positive sentiment whilst for negative sentiment this is not found in the data. We performed these analyses on both the subset of articles about the AAPL ticker and the total dataset. These results are interesting since it might suggest that there is some “objective” truth. In a sense of “if there is an event, authors will likely judge it with the same sentiment” or it can be that there is a mainstream interpretation which covers the small authors writing only a few articles with an extreme sentiment score.
Based on these results it was chosen not to give weights to certain authors who write articles with extreme sentiment scores since these are not representative in the data.

To check if any other dataset balancing technique would need to be applied, we checked that there are 901 days in the dataset labelled as 1, that is the closing price of the day is higher than the opening price, and 829 as the opposite. With his small percentage difference of 8.32% between classes, we consider it relatively balanced.

To sum up, we have chosen to implement the following pre-processing steps:

- Lowercase all words
- Stopword removal (for english)
- Remove Unicode characters
- Removing numbers, since we are focused on the effect that the words in the news have on the Stock market
- Removal of too long words over 25 characters (since these might be a typo or outlier) or single letters
- Removal of repeated spaces
- Contradiction equalisation
- Lemmatization
- Sentiment classification of positive and negative (assumed that positive sentiment positively influences the stock price, neutral does not affect the stock price and negative sentiment negatively influences the stock price)

Modelling

4. Modelling Considerations

We tested two approaches simultaneously. Firstly, BoW representation of text. Secondly, sentiment analysis of financial news. We decided to try sentiment analysis as there are a number of studies which confirm that there is a strong correlation between stock prices and publication of news articles (e.g Mohan et. al, 2017).

When it comes to BoW, after applying TfidfVectorizer the number of features was 85031, which we considered too many for a dataset with news regarding only 1730 days. Furthermore, that amount of features is too large to allow feasible testing and might include unnecessary or meaningless words. For that reason, we performed a dimensionality reduction with RandomForestClassifier with a threshold equal to the mean. There are many ways to reduce dimensionality of the data. We decided to use Random Forest for feature selection, as it allows us to keep the most important features and keep information about them (unlike PCA).

This yielded us 406 features. In this set, the 15 most important terms were: 'qcom', 'nxp', 'outlook', 'st', 'positioned slump', 'file new', store', 'break', 'bullish', 'google', 'pressure', 'apple supplier', 'future point', 'proxy', 'payment'.

NB: These are the most relevant for classification, meaning they could bring both higher or lower stock values.

When it comes to sentiment analysis, there are many available tools for that task. Nemes and Kiss (2021) made a comparison of sentiment analysis tools (TextBlob, NLTK-VADER lexicon, Recurrent neural network (RNN) and Bidirectional encoder representations from transformers (BERT) used to emotionally analyse and classify different economic news headlines and examine their impact on different stock market value changes. Emotions were classified into the usual positive, negative and neutral categories. Neutral categories appeared for TextBlob and NLTK, but not for RNN. The result they obtained was that RNN outperformed Textblob and NLTK, emphasising that there was no neutral emotional value in this case either. Since BERT performed the best, this sentiment tool was chosen for this project. While the traditional NLP models follow a unidirectional approach, that is, reading the text either from left to right or right to left, BERT reads the entire sequence of words at once. To do so BERT uses a Transformer which is essentially a mechanism to build relationships between the words in the dataset. In its simplest form, a BERT consists of two processing models namely an encoder and a decoder. The function of the encoder is reading the input text. The function of the decoder is to produce the predictions. However, since BERTs' main goal is to create a pre-trained model, the encoder takes priority over the decoder (Nemes and Kiss 2021).

Furthermore, Liu S. (2020) showed that adding bigrams have a positive effect on models performance which is why we implemented this as well. The model performance improved in our case as well for all algorithms except for the SVC (results are shown in Appendix I).

5. Model Selection and Evaluation

We tested a number of classifiers (see Appendix I), including ensemble classifiers and neural networks. At first glance, the best model that seemed to have the best performance was a neural network that achieved precision of 0.597. However after further investigating, the model seems to be underfitting (Appendix II), which is understandable given the high complexity of text analysis. Thus the chosen classifier was **GaussianNB**, as it was the best classifier between the ones we tested, with a precision of 0.641. Other models tested have their results in Appendix I.

6. Optimal parameters

For training and evaluating our models we created a function where all the results are saved and a summary of the performance of the model with the best parameters are shown (Appendix III). Multiple parameters were chosen to be tested based on what was proven in literature (Ching Chen et al. 2020) to workout well with the classification task at hand. With a test size of 20%, we used RandomizedSearchCV to find what the optimal parameters are. The validation split for the Neural network model was 20% of the test size.

Evaluating

7. Training and Evaluation Experimental Design

The experimental design that was used for this project is like a cyclic process of pre-processing, modelling, evaluating, repeating this till all options we read about in literature were attempted and we found a final model with the best of all three phases. For evaluating the model we have looked at the scores (error) over the development of our model (epochs) and tried different layers (including dropout) and different activations for the neural network. Besides, we created confusion matrices to check the amount of true and false positives and negatives since this influences the precision directly.

8. Model Performance

Given that the objective is predicting when the closing price of the day is higher than the opening price, this is classified in our model as 1. Additionally, (Willman P. et. al. 2002) states that financial managers focus on avoiding losses rather than making gains, as humans in general are loss averse. Therefore, we want to avoid False Positives, or type 1 errors, and thus the performance metric chosen was precision. This metric is defined on sklearn that it is "*the ratio tp / (tp + fp) where tp is the number of true positives and fp the number of false positives. The precision is intuitively the ability of the classifier not to label as positive a sample that is negative*". As shown in table 1, the best chosen model has the best score for predicting 1 correctly (predicting if the closing price is higher than the opening price) with a precision of 0.863. Predicting the price at closing being equal or lower than the opening price (predicting 0 correctly) also has a moderate recall of 0.691 which comes together to a weighted precision of 0.641.[1]

[1] See sklearn.metrics.precision_recall_fscore for detailed description of this calculation.

Therefore, using this model could nudge financial managers to be less loss averse. Since the model shows accurately when the stock price will rise, this stimulates them to buy stocks earlier before the prices arise and sell them later when the prices arise, which is the most commonly used and proven to be profitable trading strategy (Willman P. et al. 2002).

	0	1	recall	precision	f1	kappa
0	56	107	0.691358	0.343558	0.459016	0.212796
1	25	158	0.596226	0.863388	0.705357	0.212796

Table 1: Confusion matrix for the best model, GaussianNB

9. Limitations and Future Work

As Liu et al. (2018) found in their experiments, other methods than BoW obtain significantly better results. They show that the Hierarchical Complementary Attention Network (HCAN) they created achieves the best performance, while the performance of BoW is the worst. They argue that this is because "*BoW uses only the simple statistics of words in the news and cannot capture the semantic information of the news. In addition, it encounters the curse of dimensionality problems when the vocabulary size is too large*" (another reason why we haven chosen to only use the title of news articles). Other methods like FastText outperform BoW as well, since their ability of representing textual information is more powerful than BoW. An example of a caveat of BoW is that the word order in documents does not matter for the model, while there is also ambiguity, when a term can belong to more than one topic.

Another limitation of our research one could argue is that we are only using the titles of the news articles. However, for this more in depth research needs to be performed on the content of these articles and the sentiment spread over an article. This is because paragraphs can have different sentiments since most articles discuss a certain topic from multiple sides (Fazlija B. and Harder P. 2022) which could result in various sentiments which could also normalise the sentiment. We also made a number of assumptions, regarding impact of the news on stock prices - i.e we used only Apple related articles and searched for short term impact. It can be argued that the global situation on the stock market impacts the prices of Apple shares as well. In particular, the word that influenced the prediction the most was "qcom" and the second one was NXP. As Qualcomm [2] is an American multinational corporation that creates semiconductors, software, and services related to wireless technology and mobile communications standards, and NXP is a Semiconductor manufacturing company, it does make sense that they have common interests. In more in depth studies it might be interesting to include that ticker as well to see its direct influence.

One limitation on the sentiment analysis with FinBert is that 73% of the misclassifications of FinBERT are between positive and neutral labels, while the same number is 5% for negative and positive, according to the author (Araci D. 2019).

A last thing that might improve the performance of the model in the future is implementing trigrams. The reason for that is that changing from unigrams (single words) to bigrams (up to two word combinations) had one of the biggest improvements in the model. Sentiment analysis only had a slight improvement on the overall performance and it takes quite some time to train. Thus, if a very high precision is not needed and speed is preferred, this final step can be skipped altogether.

[2] https://en.wikipedia.org/wiki/Qualcomm

Appendix

I. Precision results of top 10 the models evaluated

Model	Input	precision_macro	precision_micro	precision_weighted
GaussianNB	Bigram with sent	**0.64379222**	**0.61849711**	**0.641042751**
GaussianNB	Bigram	0.639556277	0.604046243	0.636237582
MLPClassifier	Bigram	0.602545441	0.601156069	0.602297339
MultinomialNB	Bigram with sent	0.600904977	0.601156069	0.600957288
MultinomialNB	Bigram	0.583492063	0.583815029	0.585106891
GaussianNB	Unigrams	0.578388108	0.575144509	0.57803334
RandomForestClassifier	Bigram with sent	0.572598162	0.572254335	0.574317296
LogisticRegression	Bigram	0.569886364	0.569364162	0.571627036
LogisticRegression	Bigram with sent	0.569394407	0.572254335	0.57041581
MultinomialNB	Unigrams	0.570506305	0.569364162	0.57036846

Table 2: Precision results of the top 10 models tested, highlighted for the best value for each metric

II. Neural Network Performance

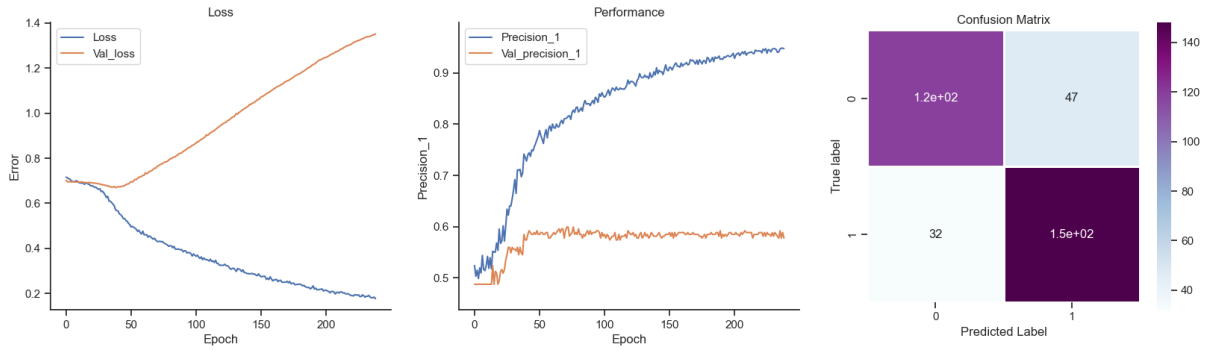


Figure 2: Underfitting Neural Network Performance

III. Full code

In case there is more interest to get more in depth insights about our analysis, the full code is available on GitHub, at https://github.com/octokami/news_stock_market containing the following elements:

A. The main notebook, where each step is thoroughly explained

There are 2 main folders: model_output containing the results without sentiment analysis and model_output_sem with them.

B. For each classification model we present:
    1. The confusion matrix or error matrix, that is, the sum for each class on what was predicted against how they were classified
    2. Plot of the words that most influenced precision
    3. The best tuned model
C. For the Neural network model:
    1. The hypertuned model
    2. The figure evaluating the training performance
    3. Kt_trials folder containing information about each hypertuning trial
D. results.csv is the aggregation of results from all models, including the ones that did not perform as well as the top 10 presented in this report.
E. utils contains the .yml file to rebuild the environment.

References

Ching Chen et al. (2020) Selecting critical features for data classification based on machine learning methods. Journal of Big Data volume 7, Article number: 52

Fazlija B. and Harder P. (2022) Using Financial News Sentiment for Stock Price Direction Prediction via https://www.mdpi.com/journal/mathematics

Liu et al. (2018) Hierarchical Complementary Attention Network for Predicting Stock Price Movements with News. CIKM'18, Torino, Italy pages 1606-1606. DOI 10.1145/3269206.3269286

Liu S. (2020) Sentiment Analysis of Yelp Reviews: A Comparison of Techniques and Models. University of Waterloo

Mohan, S., Mullapudi, S., Sammeta, S., Vijayvergia, P., & Anastasiu, D. C. (2019, April). Stock price prediction using news sentiment analysis. In 2019 IEEE Fifth International Conference on Big Data Computing Service and Applications (BigDataService) (pp. 205-208). IEEE.

Nemes L. and Kiss A. (2021) Prediction of stock values changes using sentiment analysis of stock news headlines. Pages 375-394 | Received 22 Nov 2020, Accepted 07 Jan 2021, Published online: 01 Feb 2021

Pedregosa, F., Varoquaux, Ga"el, Gramfort, A., Michel, V., Thirion, B., Grisel, O., … others. (2011). Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, https://scikit-learn.org/stable/modules/generated/sklearn.metrics.precision_score.html

Shafi M. (2014) Determinants influencing individual investor behaviour in stock market: a cross country research survey. Arabian Journal of Business and Management Review (Nigerian Chapter) Vol. 2, No. 1, 2014

Tetlock P. C. (2007) Giving Content to Investor Sentiment: The Role of Media in the Stock Market. Journal of Finance, Forthcoming, Pages 51.

Willman P., Fenton-O'Creevy M., Nicholson N., Soane E., (2002). Traders, managers and loss aversion in investment banking: a field study. Accounting, Organisations and Society, Volume 27, Issues 1–2, Pages 85-98, ISSN 0361-3682. https://doi.org/10.1016/S0361-3682(01)00029-0.