NLP applied to the US stock market
Machine Learning for Natural Language Processing 2022

CALYAKA Emin
DSSA
emin-ali.calyaka@ensae.fr

Second AuthorKHROYAN David
DSSA
david.khroyan@ensae.fr

Abstract

1 Problem Framing

1.1 Introduction

The US Stock market is one of the most globalized and accessible marketplaces in the world thanks to the democratization of brokerage accounts, computers and information. A full ecosystem with different players emerged in order to make this marketplace a viable place. Every trader has got his own strategies or believes that lead to buy or sell stock. Some people are specialised in high frequency trading and provide liquidity, others analyze quarterly reports in order to cherry-pick a portfolio. And despite all this diversity, the news impact on stock market became so heavy, that the majority of investors have to deal more and more with news analysis. Even traditional money management firms, who were historically based on fundamental analysis performance, now use quantitative methods to capture news titles, Google trends and some other pieces of information, in order to re-adjust their Portfolio weights. Let alone the quantitative firms who constantly analyse the news topics and try to make clever decisions in order to anticipate panic, bubbles and, more generally, trends.

1.2 Data

A lot of different Data providers can give access to data. The first arising thought here should be Bloomberg. Thanks to the fact that one of us is an ESSEC student, we had access to Bloomberg computers freely, but the data there can't be directly extracted into a Jupyter Notebook. That is why we preferred to work with Thompson Eikon, also known as Reuters, with an integrated python-based notebook interface. It offers almost the same data, but directly in a dataframe format from pandas.

The US stock market is composed of approximately 10 thousand companies, and it is obvious that we couldn't work with all of them. We decided to chose companies from the same index, because indexes are usually composed of companies that present some similarities, thus allowing investors to make bets on or against all of the companies with this common pattern. Thus led us to restrict our choice of companies to the SP500, the Standard and Poor's index of the 500 biggest companies in the US stock market. In reality, the number 500 may vary every quarter, and we had 503 companies in our data set. The choice of SP500 wasn't made only because of the popularity of this index. In reality, after doing some random tests and research we found out that the best articles topics quality was observed for these companies. The index includes some famous companies like Apple, Tesla, McDonald's and articles' titles are more interesting and present more opinion-based words than in the case of smaller companies. We did a comparison with the SP400, that are the 400 companies after the 500 biggest ones, and concluded that SP500 is an index that is usable enough for our purpose.

Nevertheless, one can't limit the needed data only on News articles' titles. We also needed some price changes, in order to be able to see the impact of a news on the stock price. In order to do that we downloaded prices and made aggregates of 30 minutes. The 30-minutes time interval is an arbitrage, because it has to be short enough to isolate the news impact, and long enough so that investors had time to get familiar with the news and to react.

The training data set was sent to you. previously and it is taken from Kaggle. The dataset is composed of Positive/Neutral/Other classification for 4837 news articles used in finance industry. It was first presented by Malo, P., Sinha, A., Takala, P., Korhonen, P. and Wallenius, J. (2014) in their paper: "Good debt or bad debt: Detecting semantic orientations in economic texts." Journal of the American Society for Information Science and Technology.

2 Experiments Protocol

We first trained a model on the pre-labeled dataset, using supervised learning models.

First, we cleaned the textual data by deleting numbers, punctuation, symbols, etc... Then we lemmatized by letting us choose between several classic lemmatizers/stemmers: WordnetLemmatizer, PorterStemmer and Spacy.

Naive Bayes and SVM are often used as a baseline. We chose the naive Bayesian classification model because it is quite simple, and it generally performs better than SVM on shorter texts. The naive Bayesian classification method is a supervised learning algorithm that classifies a set of observations according to probability rules derived from Bayes' law. The naive Bayesian classifier implies that the classes of the training dataset are known and provided, hence the supervised character of the tool.

For the naive bayes, we have mainly looked at the influence of several factors. First, the word embedding with two methods which are the frequency counting and the Tf-idf method. Then, we looked at the influence of n-grams, in particular up to the trigram.
In order to stay close to models that are more easily understood, we decided to focus on a model that is accessible. This model is a variant of Naive Bayes and SVM; it was presented by Sida Wang and Christopher D. Manning in the article "Baselines and Bigrams: Simple, Good Sentiment and Topic Classification" (link).
Indeed, we use the log-ratio counts of Naive Bayes as input to an SVM. We have tried to code this model simply from the classifiers already implemented in sklearn of the Naive Bayes and the SVM.

Finally, in order to process directly on the news base that we have collected via the financial API, we have used a pre-trained model on financial vocabulary: finBERT (link).

The link of our Colab is available here :
https://colab.research.google.com/drive/1D5pkZbfDxHZ4VDYrqvYMW7f7X7wz6z6Y?usp=sharing

3 Results

In terms of results, the different metrics on the basis of pre-labeled news is available in figure

The base on which we applied the models gave very little oriented news headlines, and therefore was not exploited. We had less than 5% of oriented stocks which gives us too little information about the stocks on which the news was focused. finBERT has even classified our entire stock as "neutral".

4 Limits

There are clearly apparent data limits into our study. The fact that it is hard to get real oriented news surged during the study and it was hard to overcome. One other limit are the stock returns that are made for 30-minute intervals. In a bigger study different types of intervals might be tried, even different window sizes depending on the importance of the news may be measured by regression or other techniques. The last limit is about the training dataset that isn't exactly on the US stock market of 2022, it makes it harder to do predictions.

5 Discussion/Conclusion

There are clearly apparent data limits into our study. The fact that it is hard to get real oriented news surged during the study and it was hard to overcome. One other limit are the stock returns that are made for 30-minute intervals. In a bigger study different types of intervals might be tried, even different window sizes depending on the importance of the news may be measured by regression or other techniques. The last limit is about the training dataset that isn't exactly on the US stock market of 2022, it makes it harder to do predictions.

6 Appendix

	precision	recall	f1-score	support
positive	0.68	0.46		
neutral	0.78	0.84		
negative	0.60	0.58		
accuracy				
macro avg	0.68	0.63		
weighted avg	0.71	0.72		

[]

	precision	recall	f1-score	support
positive	0.83	0.25	0.38	181
neutral	0.72	0.91	0.80	864
negative	0.59	0.45	0.52	409
accuracy			0.70	1454
macro avg	0.72	0.54	0.57	1454
weighted avg	0.70	0.70	0.67	1454

[]

	precision	recall	f1-score	support
positive	0.65	0.51	0.57	181
neutral	0.77	0.85	0.81	864
negative	0.66	0.58	0.62	409
accuracy			0.73	1454
macro avg	0.70	0.65	0.67	1454
weighted avg	0.73	0.73	0.73	1454

[]

Figure 1: Results :
a = Baseline NB (Count Embedding, No real influence of Ngram
b = NB (Tf-Idf Embedding, No real influence of Ngram)
c = NB into SVM (Count Embedding + 1,2-grams)