Quantitative and Qualitative Analysis of Stocks

Om Pandey
B. Tech, Computer Science and engineering
Indian Institute of Technology
Mandi, HP
b18182@students.iitmandi.ac.in

Sanskar Gupta
B. Tech, Electrical engineering
Indian Institute of Technology
Mandi, HP
b18140@students.iitmandi.ac.in

Vishal Singh
B. Tech, Computer Science and engineering
Indian Institute of Technology
Mandi, HP
b18150@students.iitmandi.ac.in

Rahul Kumar
B. Tech, Computer Science and engineering
Indian Institute of Technology
Mandi, HP
b18079@students.iitmandi.ac.in

Shashank
B. Tech, Electrical engineering
Indian Institute of Technology
Mandi, HP
b18192@students.iitmandi.ac.in

***Abstract*—An attempt to measure effect of qualitative factors like tweets andnews on stock market and subsequent prediction of the sentiment of the stock market using a big data approach**

***Keywords*—Spark, Stock Market, Random Forest, Python, Sentiment Analysis**

I. Introduction

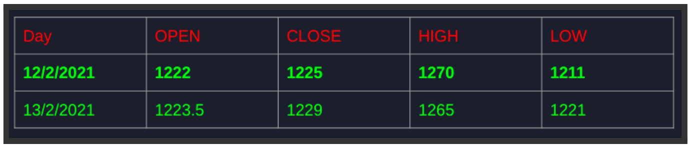
The stock market and its tendencies are particularly turbulent in the financial world. Researchers are drawn to it in order to capture the volatility and forecast its next actions. Investors and market analysts research market behaviour and devise buy and sell strategies based on their findings. Because the stock market generates a significant quantity of data every day, it is extremely difficult for an individual to incorporate all current and historical data while projecting a stock's future direction. There are primarily two approaches for predicting market trends, technical and fundamental.

Technical analysis looks at previous price and volume to forecast future trends, whereas fundamental analysis looks at financial data to get insight. The efficient-market theory, which claims that stock market prices are inherently unpredictable, casts doubt on the usefulness of both technical and fundamental analysis.

This study uses a combination of technical and fundamental analysis techniques to forecast the future trend of a stock by using tweets about a firm as primary data and attempting to categorise them as good (positive) or bad (negative). If the emotion is good, the stock price is more likely to rise, but if the sentiment is negative, the stock price is more likely to fall.

II. Quantitative analysis (Microsoft Ticker)

1. The yFINANCE Python module was used to create the database, which allows users to access historical ticker data.
2. As a vector, we used [High, Low, Open Close] of Microsoft Stocks over 1000 days.



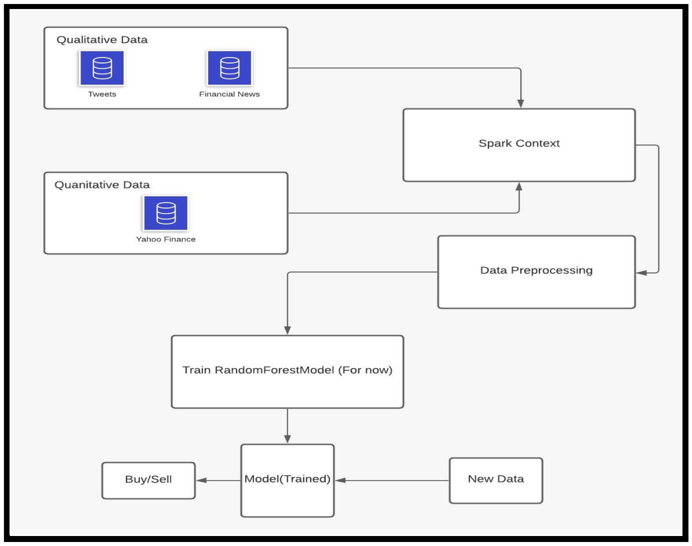
Day	OPEN	CLOSE	HIGH	LOW
12/2/2021	**1222**	**1225**	**1270**	**1211**
13/2/2021	1223.5	1229	1265	1221

3. We analysed the closing prices of consecutive days and divided the data into two groups:
   a. The following day's closing price is higher than today's closing price.
   b. The following day's closing price is lower than today's closing price.
4. We fed [Open, High, Low, Close] into the random forest model to train it.
5. The outcome (for Microsoft) is 63 percent, which is insufficient, thus we advise a change.

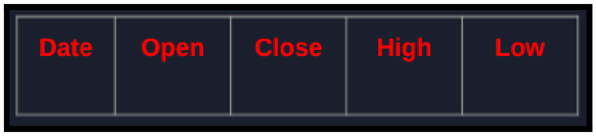
III. A qualitative cum quantitative approach

We merged the quantitative data from the yFINANCE Python module with the qualitative data from the tweets in this method. Then we preprocessed the final dataset by passing it via the spark context. The random forest model was then trained to predict whether tomorrow's closing price will be higher or lower than today's closing price.

A. Flow Chart

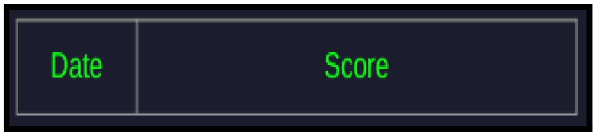


B. Quantitative Data Format



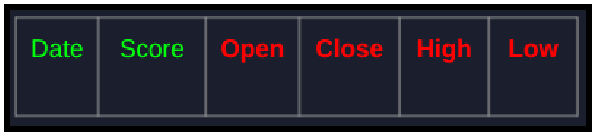
Date	Open	Close	High	Low

C. Qualitative Data Format



Date	Score

D. Final Data Format



Date	Score	Open	Close	High	Low

IV. Qualitative Analysis of Tweets

Initially, we were planning to use both news headlines and tweets as qualitative data, but after discovering that most of the news headlines were either neutral or a question, we decided to go with the tweets only, which are usually a lot more polar than their news headlines.

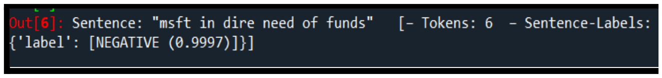
A. Collecting Data

After trying to build our own web crawler and being disappointed with the results, we decided to go with the tweets. The next big disappointment was the limitation of the tweepy library which only allowed users to fetch tweets from the last seven days only. We decided to use the twint library, which provided access to old tweets as well. We collected tweets by their “cashtags' ', which were attached to tweets just like hashtags, but started with a $ symbol and was followed by ticker of the relevant stock, e.g: $msft. We collected 100 tweets per day for the last 1000 days to build up our dataset.

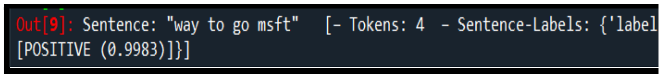
B. *Data Preprocessing and Random FOrest Classifier*

The flair library, which comes pre-loaded with a wonderful sentiment analysis model, was used to analyse the sentiment of each piece of data.

The following example demonstrates the use of a sentiment analyzer:



```
Out[6]: Sentence: "msft in dire need of funds"   [- Tokens: 6  - Sentence-Labels:
{'label': [NEGATIVE (0.9997)]}]
```



```
Out[9]: Sentence: "way to go msft"   [- Tokens: 4  - Sentence-Labels: {'label
[POSITIVE (0.9983)]}]
```

For each day, we calculated the sentiments of the tweets with the cashtag of the stocks we are working on multiplied by -1 if sentiment was negative. Then we took the average of the sentiments and stored it in a vector named as score. Then we linked the score vector to the previously used data for qualitative analysis.

We used Random Forest Classifier to model the data using the final data format as input which will predict whether tomorrow's closing price will be higher or lower than today's closing price.

C. *Results*

The results were, if we try to put it lightly, as expected. The accuracy of the model was increased by a whopping 11% to reach around 75% of accuracy. Now although this does not seems to be much, it is still a lot considering the highly volatile nature of stock market

V. Is this Big Data?

Currently, due to hardware and bandwidth constraints, we are fetching 100 tweets per day for the last 1000 days to build our dataset. This number surely does not come under the regime of big data, but given access to the right hardware this number can be extended to data of the last 10-15 years and to tens of thousands of tweets per day. If this is done, then this data will surely be “big” enough. And the good thing is that our model based on spark and python would still work as our code is scalable and was written with big data in mind.

VI. Results across various companies

The following table presents the result of our model on a few companies (three to be exact):

Company Name	Accuracy (Without tweets)	Accuracy (With tweets)
Microsoft	64	75
Reliance Industries	71.8	76
Tesla	65.7	74.5

.

As the results show, the accuracy shifted from 64-66 percent to around 75-76 percent for stocks listed in NASDAQ, whereas there wasn’t a major change in accuracy for reliance industries. This is because the cashtag of reliance industries is not used much on twitter by indian users, so because of inadequate data, the results did not change much

VII. Challenges and Limitations

The biggest hurdle in this project was without doubt the collection of data. Our initial plan was to scrape financial news from the web using a crawler, but the news collected seemed to be of no value as most of the news headlines were either questions or neutral. Then we tried to collect tweets pertaining to the stock ticker using tweepy library (official twitter library), but it only allowed one to fetch tweets from the last seven days. Finally we stumbled upon a library called twint, but the data extraction is still taking around 3.5 hours even if we are fetching only 100 tweets per day. Moreover, sometimes this library is unable to retrieve old tweets or throws 403 errors.

Apart from this, another major problem was that we were searching tweets based on cashtags, but

Indian users on twitter do not use cash tags that often in contrast to the NASDAQ and other stock exchanges. Because of this, we were not able to extract many tweets for stocks listed in indian stock exchanges.

VIII. Future Improvements

Our original plan was to incorporate traditional indicators like moving averages and bollinger bands into our model, but because of time constraints, we were not able to do so. We can still incorporate these indicators in our model. Also because of the aforementioned time constraints, we were not able to test any models other than the random forest classifier, so there is room for improvement there also.

Apart from this, we created the labels with short term trading in mind, so we can customize this and take this as an input from the user too. Finally, we haven't prepared a user friendly gui yet, which is also a part of future improvements.

IX. Conclusion

We have seen through this project that addition of qualitative data, i.e. sentiment analysis of tweets related to a particular stock into the traditional quantitative data can improve the performance of a model by close to 11-12%. This improvement too, however, will make the model reach an accuracy of around 75%, which is still not much, but good considering the volatile nature of the stock market. Although there is still a lot of scope of improvement, this project works as a proof of concept for the effect of sentiment of tweets on the stock market.

Acknowledgment

We would like to acknowledge Dr. Arti Kashyap, who provided us with an opportunity to choose and work on a project that we find interesting as part of the CS561 course (Big Data and map reduce).

References

1. https://github.com/twintproject/twint
2. https://pypi.org/project/yfinance/
3. https://github.com/flairNLP/flair
4. https://seekingalpha.com/