Trumpefe

Fabio Taddei Dalla Torre
Matr. 214924

Alessandro Emanuele Piotti
Matr. 215191

Introduction

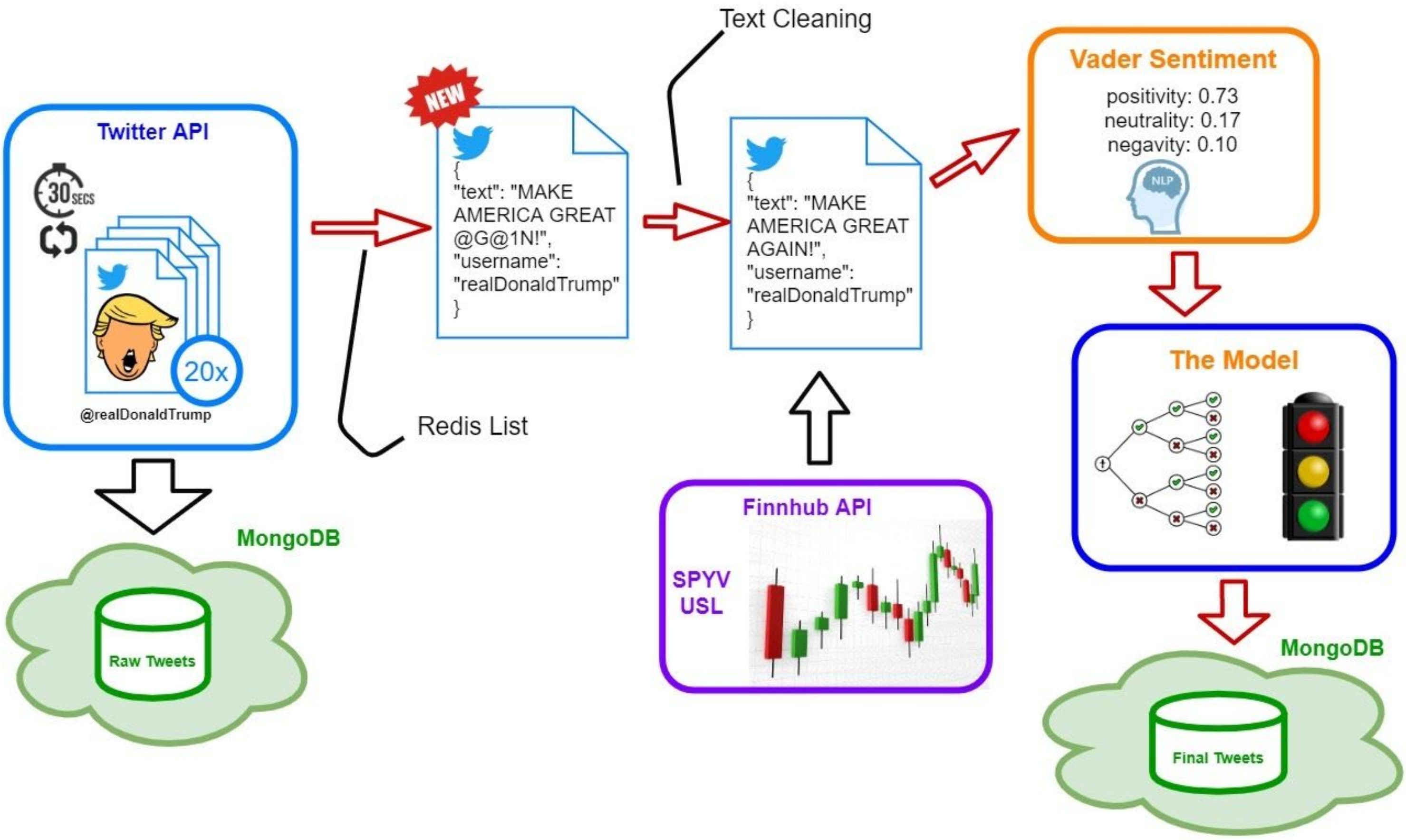
Because of the variety of variables involved in the stock's price making process, it is hard to predict how investors will behave. Nevertheless, JPMorgan, one of the biggest financial services companies, has developed the Volfefe index. This index measures the impact of Trump tweets on the volatility of certain stocks.
Our task required us to develop a system capable of computing a model that allows us to predict the influence of President Trump tweets on the stock market.
Long term stocks are in general less volatile and, after evaluating a wider range of options, we decided to focus our attention on two American stocks: USL that is the 12-month USA oil fund and SPYV that is linked to S&P500 Value Index.

First Ideas

The main idea of the system is to provide a useful alert system in case of the current President of The United States, Mr. Donald Trump, publishes a Tweet that could affect significantly the volatility of some given stocks. It could be a worthwhile solution for the ones that are in the stock market and they want to always know the latest news. In order to have always the most recent tweets available, it is important to stay connected to the Twitter feed and thanks to the Twitter API, it is possible to get the last 20 tweets published by his account @realDonaldTrump.

Design

It was decided to implement a Big Data system to manage the data ingestion and the subsequent stages of data preparation (cleaning) and data transformation. The main sources of information are the tweets published by President Trump and the stocks used in the model. Together with the System, it was projected and implemented a web application that makes it possible to access the *Trumpefe* solution in a natural and intuitive way. The focus of the application is the access of the main information processed by the system.



System

The entire pipeline is composed of different independent stages. The most important ones are the stock extraction and the linguistic analysis. Since it is straightforward and fairly linear, there is only one initial *checkpoint* in which the new raw tweets are immediately stored in the database, and the other one is at the end of the processing. One relevant bottleneck is represented by the stocks API, since they are provided with a limited usage.

Application

It consists of a simple webapp implemented with python programming language and based on the Flask microframework, the DB is the same of the system accessed in a read-only fashion.

Data Model

Having a system that revolves around tweet objects leads to the natural choice of preserving the semi-structured profile of json files and this flexibility allows to edit the existing fields and insert new ones into the original object.
From a higher level we can exploit these characteristics to keep all the atomic information inside a single object extracting the interesting aspects when needed. The structure is completed by the NoSQL database (document-based) scalable by design thanks to the AWS hosting.

Big Data Choices

Redis: for its simplicity and reliability for atomic operation. Creating a layer between Twitter API and to give to the system the time necessary to transform each tweet from the processing list.
MongoDB Atlas: a fully managed replicated MongoDB Database hosted on AWS Frankfurt with a set of operations tailored to work with these kinds of data.
Python: besides its simplicity, it was complete and robust enough to build an entire solution (system + application) with just one programming language. The best choice with tons of useful packages to manage the environment, the API calls and the database.
Github: version control and hosting.
API: Twitter API (tweets), Twitter oEmbed API (tweets display), Finnhub API (stocks), MongoDB API (database), Dandelion API (sentiment, besides its *turbulence*).

Functioning

System

The system requests the last 20 tweets of @realDonaldTrump every 30 seconds, the new tweets that are not already present in the system are therefore pushed in a Redis List waiting to be processed. In the file `redis_sentiment.py` is where the main processing happens. The tweets are cleaned and is extracted some secondary information like number of hashtags, mentions, etc. After the insertion of the volatility, The clean text is processed with the vader sentiment package in order to obtain the sentiment analysis (compound, positivity, neutrality and negativity) and with all those elements it is possible to make a prediction with the model (decision tree) already created and store the final result back in the cloud.

Efficiency

Model Choice

All the manipulation on the data was done in order to provide a training and a test set for two classification trees used in order to predict whether the tweet has an extreme bad or extreme good impact on the stock market.

Finnhub API

All the manipulation on the data has been done in order to provide a training and a test set for two classification trees used in order to predict whether the tweet has an extreme bad or extreme good impact on the stock market.

The starting point of the model is a database created through the API service provided by the website Finnhub.io. For each observation (tweet) we considered stock data, in particular: opening, closure, high and low, from an hour before to half an hour after the tweet, with a frequency of almost one minute. We only considered tweets that were published on the opening hour of the stock market with an hour margin with respect to the opening and half hour to the closure. Those decisions were dictated by the fact that President Trump publishes an average of 20 tweets per day, hence considering a wider range of observations for each tweet could create too much overlapping between each tweet. In order to estimate the volatility before and after the tweet we used the Parkinson formula for intraday volatility measure

$$\sigma = \sqrt{\frac{1}{4Nln2}\sum_{i=1}^{N}(ln\frac{h_i}{l_i})^2}$$

where N is the number of observations, h are high and l are low prices, then we labelled the observation as 0 or 1 according to the variation of volatility. Moreover, we created another two columns that we labelled according to the variation in the volatility and the variation in the stock prices. This choice was taken because the market is going particularly bad if the prices and the volatility are decreasing (that means the market is oriented towards a negative trend), on the other hand the market is going particularly well if the volatility is decreasing and the prices are increasing (for the same but opposite reason).

Model Evaluation

We measured the performance of the models through a test error rate, in particular we assigned the 80% of the observation for training and the remaining 20% for testing. In general, the model performs quite well showing a test error that is in general lower than the 28%, only the model used for the case in which USL is going particularly bad that shows a test error rate of 39%.

Moreover, the training procedures of the trees are very fast, a factor that makes it possible to easily train the trees on bigger sets.

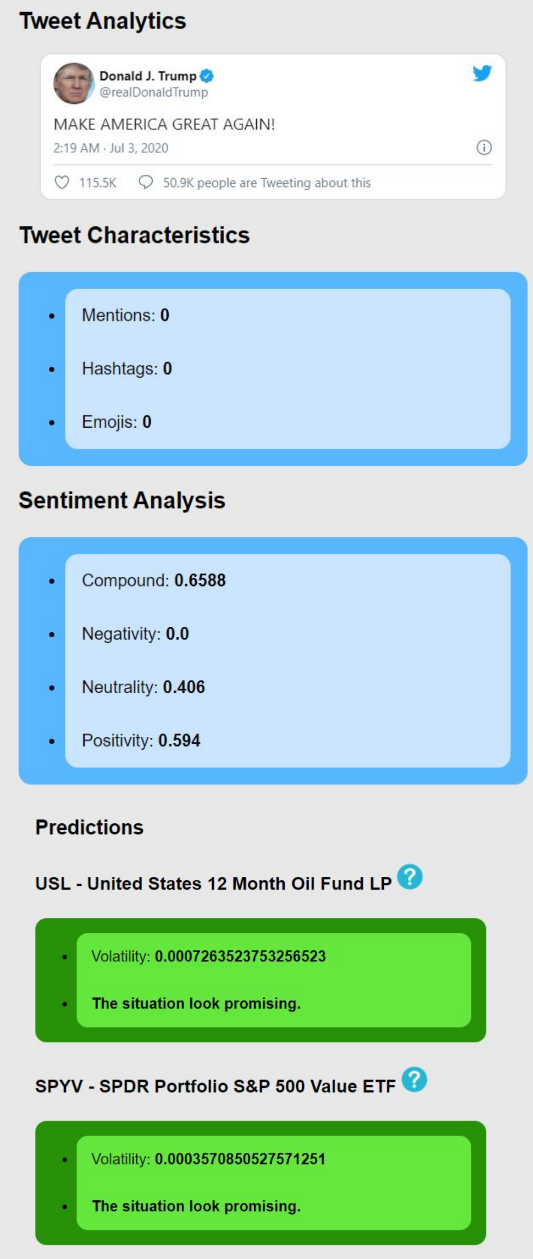
Conclusion

Trumpefe can be seen as the first experiment of stock prediction with a marginal help given by the stocks indicators (high, low, volatility). The main part of the project was focused on finding the approach to process these information in an innovative way, to extract meaningful information and to improve the prediction. Clearly a bunch of words and hashtags cannot arrogantly interpret the complex reality of our days. On the other hand, with the increase of information, testing new stocks and new temporal patterns could lead to interesting discoveries. Some of them could happen thanks to the possibility to recombine this system with new tweet sources (senators, politicians), new stocks ticker and to scale it with more information. At the moment we are satisfied with the application showing us some basic investing insights.

> In order to run the project, please follow the instructions in the `README.md` file.

Appendix

Views taken from the Application:



Tweet Analytics

Donald J. Trump
@realDonaldTrump

MAKE AMERICA GREAT AGAIN!

2:19 AM · Jul 3, 2020

♡ 115.5K 50.9K people are Tweeting about this

Tweet Characteristics

- Mentions: **0**
- Hashtags: **0**
- Emojis: **0**

Sentiment Analysis

- Compound: **0.6588**
- Negativity: **0.0**
- Neutrality: **0.406**
- Positivity: **0.594**

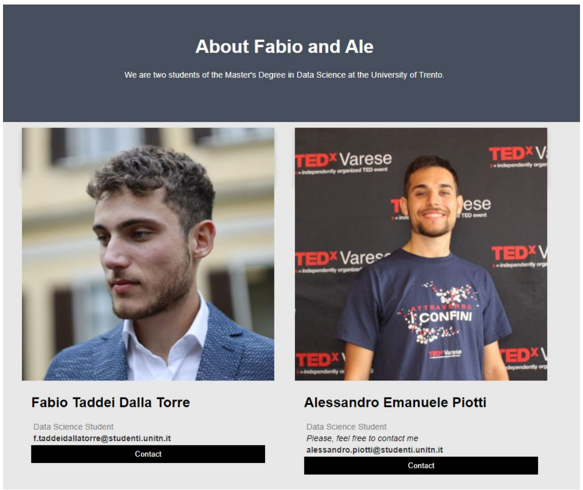
Predictions

USL - United States 12 Month Oil Fund LP ?

- Volatility: **0.00072635237532566523**
- **The situation look promising.**

SPYV - SPDR Portfolio S&P 500 Value ETF ?

- Volatility: **0.0003570850527571251**
- **The situation look promising.**



About Fabio and Ale

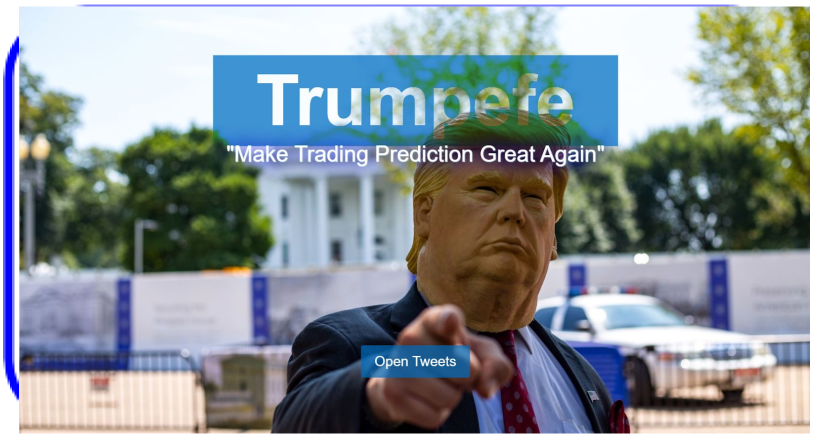
We are two students of the Master's Degree in Data Science at the University of Trento.

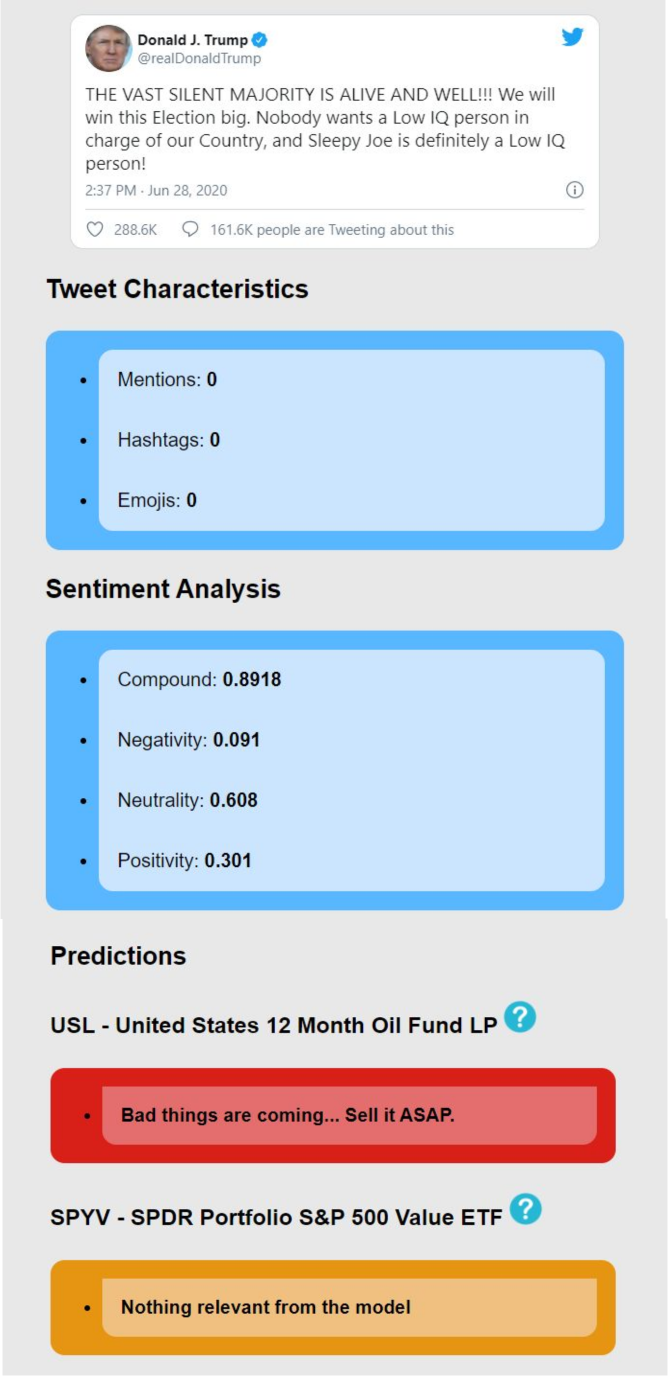
Fabio Taddei Dalla Torre

Data Science Student
f.taddeidallatorre@studenti.unitn.it

Alessandro Emanuele Piotti

Data Science Student
Please, feel free to contact me
alessandro.piotti@studenti.unitn.it





Donald J. Trump
@realDonaldTrump

THE VAST SILENT MAJORITY IS ALIVE AND WELL!!! We will win this Election big. Nobody wants a Low IQ person in charge of our Country, and Sleepy Joe is definitely a Low IQ person!

2:37 PM · Jun 28, 2020

♡ 288.6K 161.6K people are Tweeting about this

Tweet Characteristics

- Mentions: **0**
- Hashtags: **0**
- Emojis: **0**

Sentiment Analysis

- Compound: **0.8918**
- Negativity: **0.091**
- Neutrality: **0.608**
- Positivity: **0.301**

Predictions

USL - United States 12 Month Oil Fund LP ?

- **Bad things are coming... Sell it ASAP.**

SPYV - SPDR Portfolio S&P 500 Value ETF ?

- **Nothing relevant from the model**