Donald Trump’s Tweets

& Stock Trends

Predictive Analytics Model

Group Members: Maria Yolovska | SNR: 2062642

Lidia Nikolova | SNR: 2057873

Ivaylo Papazov | SNR: 2051866

Team Number: 1

1. Model's Objective

In today's modern world of communication, approximately 45% of the population relies on social media and shares opinions, preferences and trends with the other users in various social media platforms. A prominent example for such an environment is Twitter where 206 million active daily users post about their interests and activities.

This background information inspired our team's predictive analytics model which provides insights into the tweets of the former president of the United States, Donald Trump. We were specifically interested what effect his tweets can have on the trends in the NASDAQ Stock Exchange.

Having this in mind, we created a model based on a set of data that can predict the behavior of stocks in the NASDAQ Stock Exchange according to the opinions that Donald Trump leaves in the form of personally posted tweets. As an influential public figure, the opinion of the former president has an impact at the organization level of NASDAQ's stock trends which then results in a varying direction of the stocks' price.

The respective price fluctuations can be advantageous for some investors, but also detrimental to others. That is why our team wanted to gain insight into how the stock price variations can be predicted, so that investors can have a general idea of their investments' current valuation. A further interesting point while working on the predictive model is how the influence of Donald Trump's tweets affected the direction of the NASDAQ stocks. Taken altogether, these aspects will be explained in more detail in the next sections of this paper.

2. Data Preparation

The data used in the model comes from two separate files, namely Donald Trump Tweets 2019 and NASDAQ stock prices. The two files were used following the teacher's recommendation after facing troubles with the data for our formerly chosen company, United Airlines. They can be found on Canvas, under the Module "Predictive Analytics - Assignment".

Throughout the data preparation process we faced several challenges. First, we found the data we needed for United Airlines. However, while trying to merge both files, PythoCurrentn gave an error we could not find a solution to. Consequently, we used the Donald Trump data, available on Canvas. After using Donald Trump and NASDAQ stock data we did not face any challenges.

The transformation process of the data consists of several steps. First we removed some of the unnecessary variables and all observations that were not from Donald Trump. Next, we filtered the file with stocks in order to have the necessary data. Furthermore, we divided the time column into two columns - Day-Month-Year and Seconds-Minutes-Hours of the tweets and stocks. Finally, we filtered the columns in descending order - the date and time are filtered from oldest to newest, along with the Tweets and the rest of the information in both files.

3. **Modeling and Analytics**

To establish the accuracy of the model, we used a Naive Bayes sentiment analysis. Therefore, the analytical model and techniques we have applied are connected to machine learning models.

More specifically, the type of machine learning model that is described in this report is an unsupervised learning model, since the data we have available does not have any labels provided.

Furthermore, we also tried applying a Decision Tree technique in order to see if the accuracy would be higher, but Python gave an error we could not tackle appropriately.

To begin with the techniques applied towards sentiment analysis, the first thing we had to do is, given the data we had already merged, to tokenize it. Tokenizing is a technique you use to dismantle the given tweet into a set of strings so that Python can understand it. We defined it by using the function *def*.

Afterwards, we created another function, which removed the stopwords from all the tweets. These words are unnecessary words such as "a", "an", etc. which do not provide any important information for our analysis. We created this function by downloading the stopwords packing from the NLTK import.

The next step was to find the 7000 most used words that Trump used in his tweets throughout 2019. We did that by yet another function, and the obtained data, combined with the filtering from stopwords, is what we used to continue the analysis.

Then, we created a featureset using the previously mentioned top 7000 most used words. By doing this featureset, we provided the future data that we will use for the training and testing set in the Naive Bayes sentiment analysis.

Finally, using the NLTK Python package, we put in the code for the Naive Bayes model. The first step was to establish a training set and a testing set for the model to get data from. We did that by dividing the data in parts of 75/25, where 75 percent of the filtered information was used to train the model, and the rest 25 percent to test it.

Finally, we added one last line of code to view the accuracy of the model we created, along with the 10 most informative features to get further insight from our model.

4. Validation

The current accuracy of the model is 0.6140. The initial accuracy was 0.49, so we had to do a few changes to our model to increase it to the current level. Those changes will be described below.

The validation method used for our model is train/split test. The sets, created from our filtered data, had originally a 75:25 ratio split, as mentioned in the Modeling and Analytics section, but after running the model, we found out that changing the ratio to 80:20 instead, gave a slight boost to our accuracy. Therefore, we decided to keep the new and improved ratio.

Furthermore, we saw that reducing the number of features - namely the most used words that Donald Trump used in his tweets, also contributed to increasing the model accuracy. Consequently, we reduced the number of features from 7000 to 4000.

Finally, we added another line of code regarding the order of the words in the filtered data. Using the random package provided by Python, we shuffled the featureset, overcoming the issue of the alphabetic ordering of the data. This resulted in increasing the accuracy to our current final number of 0.6140.

5. Evaluation Form:

Maria Yolovska - Increased the accuracy of the model; Data Preparation part of the paper; Made the Presentation

Lidia Nikolova - Introduction of the paper; Final check of all parts

Ivaylo Papazov - Create the Python Model; Modeling and Analytics & Validation parts of the paper