DSS

Assignment 3

Group 10

Stock price prediction using sentiment analysis

|  |  |
| --- | --- |
| Group 10: |  |
| Names: | SNR: |
| Velin Todorov | 2019916 |
| Ivo Vloet | 2027533 |
| Tim van Asten | 2041767 |
| Luuk Verbiesen | 2042510 |

# Introduction

According to Mehta, Pandya and Kotecha (2021), social media sites like Facebook and Twitter allow users to share their opinion. These opinions can reflect the public’s view of current events. In the context of the stock market, opinions expressed on the previously mentioned social media websites can have an impact on the stock price. Therefore, this paper will describe an approach to predicting stock price using sentiment analysis. More specifically, this paper will describe how we utilized twitter and stock price data about Apple Inc (APPL) to create a model. The twitter data is obtained from Kaggle, while the stock price data is obtained from Yahoo Finance. The twitter data contains already calculated sentiment polarity per volume of tweets for every single day. The twitter data is measured on a daily basis, where the first observation begins from 01/01/2016 and the last observation is on 02/09/2019. To match the time frame of the twitter data, the stock price data will be obtained from the same period (01/01/2016 - 02/09/2019). For these reasons, the model that our group proposes can be used at individual level. Since individuals can use the proposed model to first identify what the overall mood of people is regarding a company by applying sentiment analysis and then use the results from the sentiment analysis in combination with stock price data to come up with a final prediction whether the stock will go up or down.

# Data Preparation

For this assignment, there will be two datasets - one which contains already calculated sentiment polarities and tweets volume, and another containing stock price data. The twitter dataset is obtained from Kaggle, while the stock price data is obtained from Yahoo Finance. Both datasets are related to Apple Inc. (APPL). The stock price data starts from 04/01/2016, while the twitter data starts from 01/01/2016. Therefore, the first data preparation technique is to match the time frame on the two datasets - days which are holidays, and weekends need to be deleted from the twitter data. In order to check if the data preparation was successful, the dates in the twitter dataset were first transformed into a text like ‘Mon’, ‘Tue’ etc. This was done to obtain all the days which are not part of the weekend and delete weekend observations - Sat and Sun. Then, the date from the stock price data was imported into the twitter dataset. Another variable called ‘Match’ was created, which consisted of boolean values - True or False. In every cell of the match column, the following formula was used - ‘=Cell1=Cell2’, in order to check if the dates matched. The formula returns ‘True’ if there is a match, and ‘False’ if there is no match. Once ‘True’ was obtained everywhere, and the ‘False’ values were deleted, the match variable was removed. The columns ‘ts\_polarity’ and ‘twitter\_volume’ from the twitter dataset were moved into the stock price dataset. After that the columns ‘Open’, ‘High’, ‘Low’, ‘Close’, ‘Adjusted Close’ were formatted to two numbers after the decimal point in order to achieve consistency. This process created the final dataset that is used for the analytics part. The most challenging part of this process was matching the data time frame between the two datasets, because both files are big and this task was done manually. After this, the data was also processed again in Python, where the dataset was transformed into a dataframe using the library pandas.

# Modelling and analytics

The modelling in this project was done using Random Forest Classification (RFC). According to Donges (2021), random forest is a supervised machine learning approach that uses multiple decision trees and combines those trees to increase the accuracy of the results. Our model uses 20 trees. Random forest can be used for either classification or regression models (Donges, 2021).

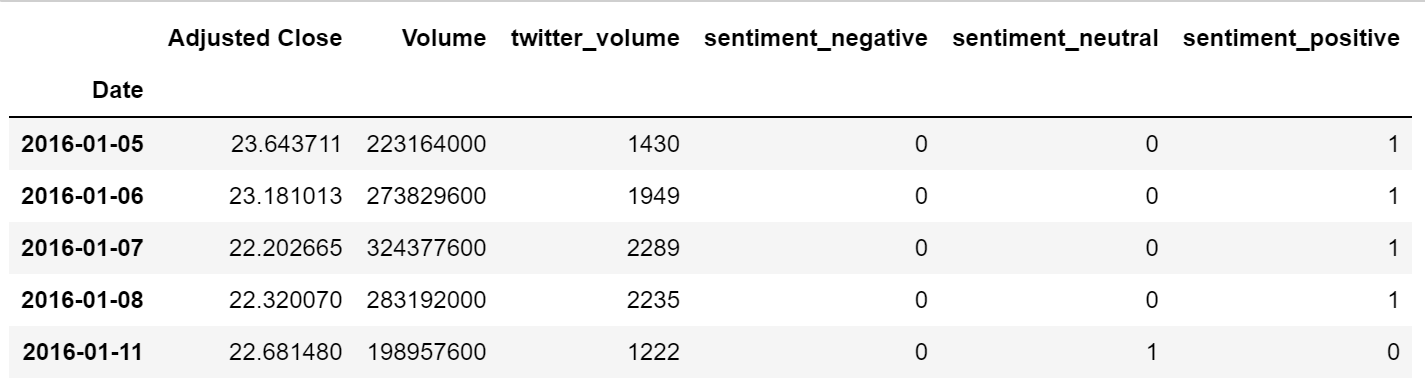
Donges (2021) states that random forest can be used in stock trading, to determine a stock’s future behaviour. Because of this, our team decided to use this approach. The first thing we did was to classify the sentiments from the Apple twitter sentiment data into positive, negative and neutral.

To do this, the following algorithm was implemented:

## Figure 1. Algorithm to determine the sentiment polarity

After determining the sentiment polarity, we created a variable called ‘price diff’, which contained positive or negative numbers. If there is a positive number, this indicates an increase in the price at time t compared to the price at t-1, and if there is a negative number, this indicates a negative trend. The formula to determine this is: (price at time t - price at time t-1) / price at time t-1. From this variable we created a dummy variable, called ‘trend’, which contained 1 for positive increase and 0 for everything else. The column ‘trend’ was dropped from the feature set. Then, we created 3 separate dummy columns for the sentiments, called ‘sentiment\_negative’, ‘sentiment\_neutral’, ‘sentiment \_ positive’. In the end, our feature set consisted of the following columns:

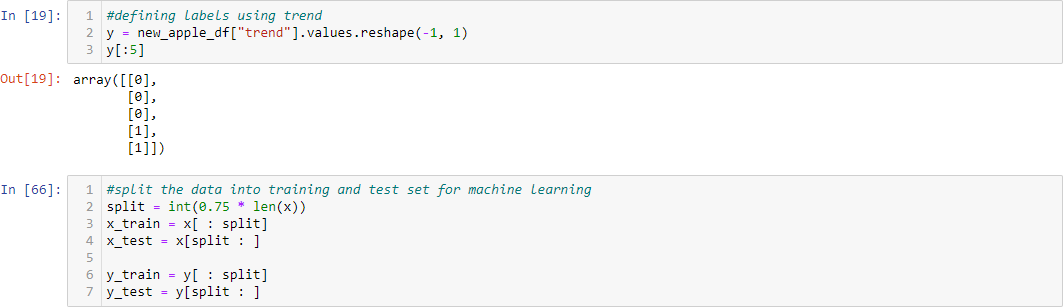
## Figure 2. Feature set



After obtaining the feature set, we obtain a one-dimensional array from the ‘trend’ column. We do this because this array is going to represent the dependent variable that we are trying to predict.

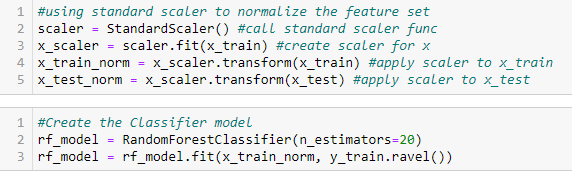
After obtaining the feature set(IV) and the array(DV), we split the data in a ratio 75% / 25%, as this ratio produced the highest accuracy score.

## Figure 3. Splitting the data and creation the of DV



Once done with creating the training and testing set, we standardized the train and test data sets. This is an attempt to increase the accuracy of the model. After that, we created the random forest classification model.

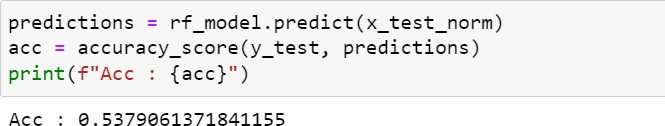
## Figure 4. Creation of the Classifier model



# Validation

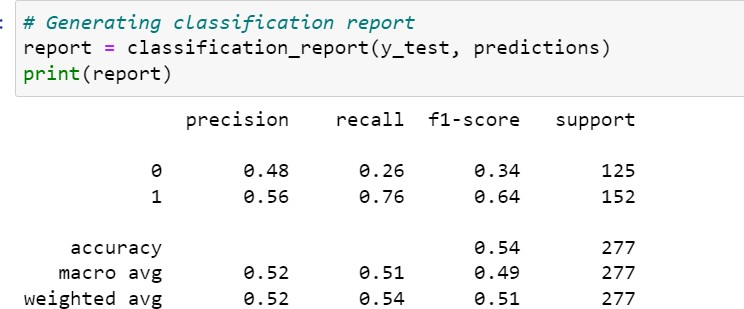
After the data was ready and split into train and test sets and the model instance was created, data predictions were made and saved. Afterwards, Sklearn metrics were used to calculate the accuracy of our model. As seen in figure 5, the accuracy of the random forest classification model was 53,8%.

## Figure 5. Calculation of predictions and accuracy



After calculating the accuracy, Sklearn metrics was used to generate a classification report. This classification report is shown in figure 6. In the table it can be seen that the precision for negative sentiment tweets is less than the precision of positive sentiment tweets. This difference is most likely due to the fact that there are more observations with a positive sentiment rather than a negative sentiment. In the classification report it can be seen that the precision for negative sentiment tweets is about 48% and the precision for positive sentiment tweets is about 56%.

## Figure 6. Creation of the classification report

In accordance with our findings in the classification report, more observations result in a

more precise prediction. This could be used to improve the accuracy of our model. If there is more data available, the algorithm has more data to train with, therefore making the predictions more accurate. One thing that could also be done is using a different model. In this project we used a classification algorithm and this resulted in a 53% accuracy. Using more powerful statistical methods, like regression, could improve the precision of the predictions significantly.

References:

1. Malik, U. (2018, June 13). Random Forest Algorithm with Python and Scikit-Learn. Stack Abuse. <https://stackabuse.com/random-forest-algorithm-with-python-and-scikit-learn/>
2. Teo, B. G. (2021, May 15). How to Predict Stock Prices Change with Random Forest in Python. Medium.

[https://python.plainenglish.io/how-to-predict-stock-prices-change-with-random-forest-in-python-f](https://python.plainenglish.io/how-to-predict-stock-prices-change-with-random-forest-in-python-f707e101d5c4) [707e101d5c4](https://python.plainenglish.io/how-to-predict-stock-prices-change-with-random-forest-in-python-f707e101d5c4)

1. Markham, K. (2020, February 3). Simple guide to confusion matrix terminology. Data School. <https://www.dataschool.io/simple-guide-to-confusion-matrix-terminology/>
2. Mehta, P., Pandya, S., & Kotecha, K. (2021). Harvesting social media sentiment analysis to enhance stock market prediction using deep learning. PeerJ Computer Science, 7, e476. <https://doi.org/10.7717/peerj-cs.476>
3. Dubey, A. (2018, December 15). Feature Selection Using Random forest - Towards Data Science. Medium. <https://towardsdatascience.com/feature-selection-using-random-forest-26d7b747597f>
4. Real Python. (2021, August 20). Split Your Dataset With scikit-learn’s train\_test\_split(). <https://realpython.com/train-test-split-python-data/>
5. Sklearn Random Forest Classifiers in Python. (2018, May 16). DataCamp Community. <https://www.datacamp.com/community/tutorials/random-forests-classifier-python>
6. Twitter Sentiments dataset: <https://www.kaggle.com/nadun94/twitter-sentiments-aapl-stock>