ISM 6136 Data Mining Final Project

Shani Nuyts, Motahareh Pourbehzadi, Pankaj Tiwari, and Sarmad Kiani

Introduction

We selected Fintech for our project. With this project, we were interested in analyzing the relationship between news sentiments and adjusted stock prices. We wanted to see how our model can bring in valuation for the companies. We wanted to see how news for a certain company can impact the valuation of that company in the stock market (sentiment analysis). The companies we chose were Facebook, Pfizer, Dogecoin. We wanted to see how particular news about these companies impact their outlook in stock valuation. We collected news data from different sources and then ran multiple ML models including decision tree, Support Vector Machines model. These models are useful for data extraction and pattern prediction. This model uses some particular words from the news like "conspiracy" and then sees its outcome on the valuation of stock. Recent news of data breach of facebook , conspiracies related to Pfizer vaccines and Elon musk tweeting for dogecoin are some of the many news that companies use for analysis. Sentiment analysis is becoming more and more popular and we wanted to work on sentiment analysis for our final project.

Data Characteristics

Since sentiment analysis is a part of natural language processing which tries to identify the opinion within a given text across news, blogs, reviews, social media, etc, we usually have the raw input data as text phrases. For this project, we have raw data as news headlines specifically shortlisted based on 3 keywords: Facebook, Pfizer and Dogecoin. This data has been taken from two different data sources: Kaggle and newsapi.org. Data from Kaggle is a collection of 5000 news headlines whereas the similar dataset from newsapi.org is used for testing the model. We have ensured that there is no discrepancy between these two datasets. As our model is using binary classification, so we have categorized our data either as positive or negative and have avoided the 3rd category of 'neutral.'

For this model, "News Headlines" is the independent variable whereas "Sentiment label" column which has either number 1 (means positive sentiment) or -1 (means negative sentiment) is our dependent variable. We use 80% of the data to train our model and 20% of the data will be used to test the model. So our data is split in an 80:20 ratio for training and testing. We have tried to solve business problem of identifying news impact on market sentiments and thereby on company's share price by adding below aspects of sentiment analysis in the dataset:

Domain specific tasks: As sentiment Analysis is heavily dependent on domain expertise, addition of domain specific concepts influences the overall semantic orientation of sentences.
Verbs and expressions to detect direction of events: Verbs and expressions help to detect direction of events. For eg. verbs like increasing profit or decreasing profit, etc.
Polarity of different concepts with their relation to the expected direction of events: Addition of information on how the polarity of different concepts depends on the expected direction of events. For eg. sentiments are positive if results are expected to increase and vice versa.

DM Model Construction

The data preprocessing for this model is done in three steps: normalization, feature hashing, and filter-based feature selection.

Normalization: In Sentiment analysis, as a 1st step of normalization, we have removed all words which do not contain sentiments. We used R code to remove punctuation, digits, special characters, stop words and then all the phrases are converted into lower case. This normalized data is then made as non-categorical data and it is given input as to Feature Hashing.

Feature Hashing: This step takes the input as a variable length text input and gives output as a fixed length numeric vector in the form of unigrams and bigrams in text.

Filter-Based Feature Selection: The output from feature hashing is huge and we select only the most relevant features. For this model, we use the top 1500 most relevant features.

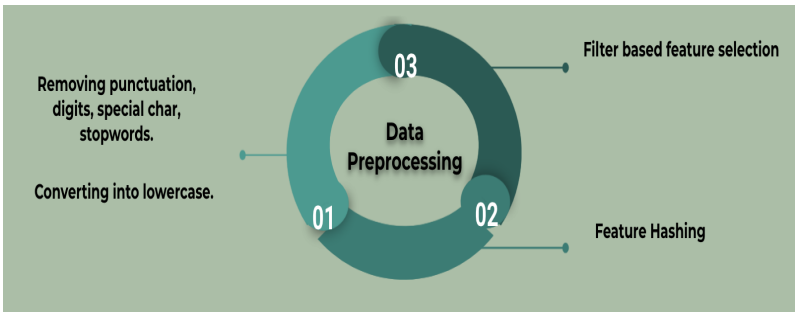

Fig.1. Data Preprocessing Steps

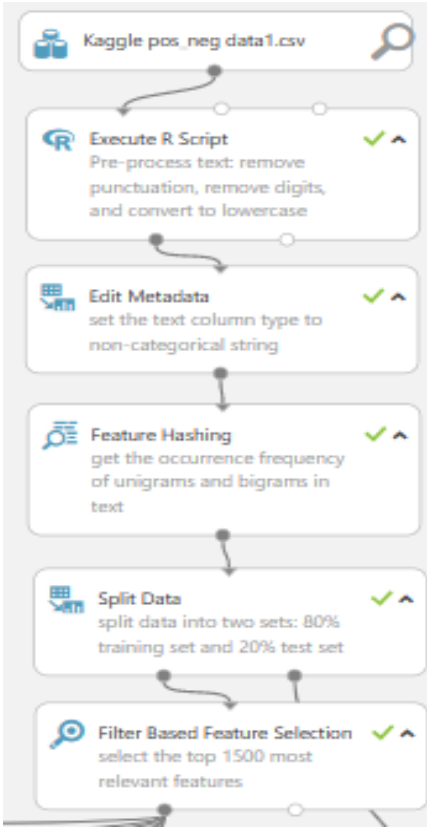

Fig. 2. Data Preprocessing Model

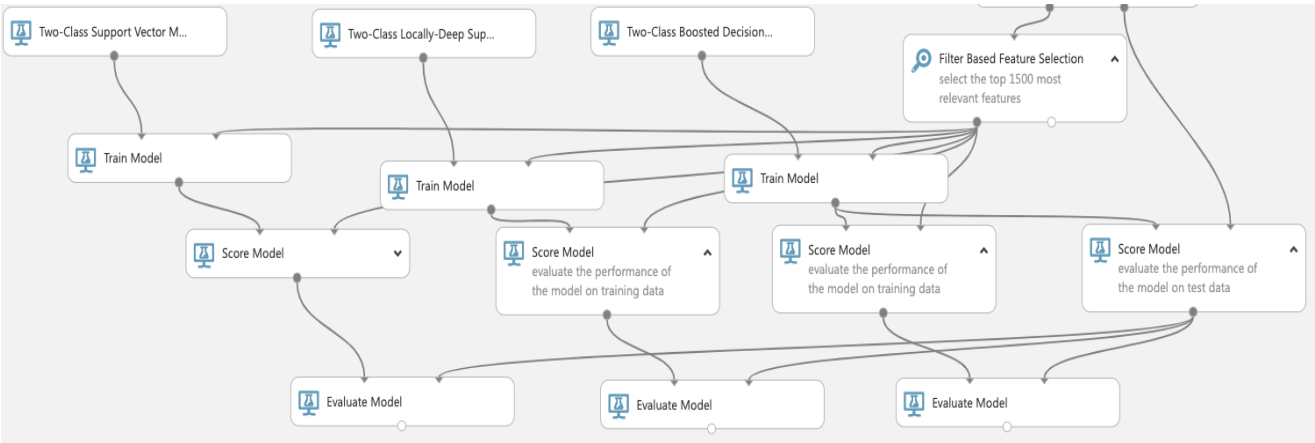

Fig. 3. Data Mining Model

Based on the data characteristics and the problem statement, we must determine how to construct the data mining model. To do so, we first look at the necessary requirements of our model. The models used must be supervised models since we are using our independent variable, the text, to determine a dependent variable, the sentiment label. It is clear that the models must be classification models since the goal of the data mining model is to classify the text as having either a positive or a negative sentiment. Therefore, the models we use must be two-class classification models. Interpretability is not very important for our model, since we simply want an accurate understanding of whether a subject's current media coverage is positive or negative. However, if we changed the objective of our model so that we would like to know why a certain text has a positive or negative sentiment, a different analysis would need to be performed.

The first of our models is a two-class boosted decision tree. Similar to a random forest model, the boosted decision tree consists of multiple decision trees where each tree is dependent on prior trees, as shown in figure 4 (Yildrim). This leads to a greater accuracy than that of the regular decision tree model. The boosted decision tree model is highly efficient when it comes to both classification and regression models and has a higher accuracy than a random forest model (Zhang et al.,a). However, the boosted decision tree is prone to overfitting and is sensitive to outliers, which must be accounted for during implementation (Yildrim).

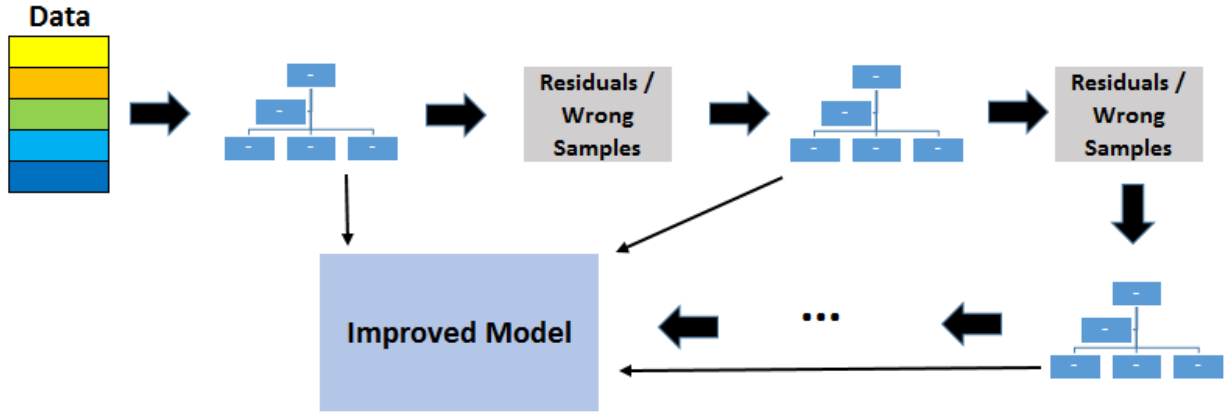

Fig. 4. Boosted Decision Tree Model process

The second of our models is the two-class support vector machine. The two-class support vector machine (SVM) model is based on the support vector machine algorithm. In this particular case, the SVM model is a classification model, but SVM can also be used as a regression model.This model has a variety of uses, particularly in text and image classification. The SVM model takes the training data and plots it on an n-dimensional space, where n is the number of features of the data. The algorithm then plots an n-dimensional hyperplane to distinguish between the two classes. The hyperplane is created in a way to maximize the margin between the data points and the hyperplane, as seen in figure 5 (Ghandi).

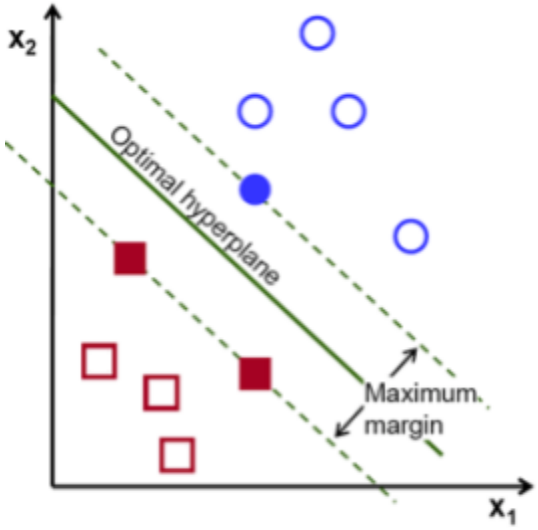

Fig. 5. Support Vector Machine data and hyperplane mapping

We also use the two-class locally deep support vector machine. This algorithm is the result of research which tried to take the widely popular support vector machine model and tried to increase its training speed. The kernel function, used to map the data points onto the n-dimensional space, is slightly altered in this model to reduce the training speed without losing accuracy. This is useful when working with very large training datasets. By using both the two-class support vector machine model and the locally deep support vector machine model, we could run both and compare whether or not the locally deep SVM did indeed reduce the training speed of the model for the dataset we were working with.

Results

Now that we have discussed the data characteristics and the models that we have selected based on the specifications of the problem that we are trying to tackle, it is time to go through the results as shown in Figures 6-8. Each figure demonstrates the performance of the model on the training data, the performance of the model on the testing data and their ROC curves. The first figure illustrates the performance of the 2-class boosted decision tree. In this model the total area under curve (AUC) for the model on the training set is 0.992 and the AUC for the testing data is 0.903. Overall for the total number of 430 data points in the training dataset, 355 data points are correctly classified. Figure 7, represents the results of the 2-class locally deep support vector machine (SVM). The performance of the model on the training data is much better than the previous model with the accuracy of 0.99 and the AUC equal to 1. Even though the performance of the model on the training data is better than the previous model, the 2-class deep SVM model has exactly the same AUC as the 2-class decision tree. Finally, it comes to the third model which is the 2-class SVM model. The performance of this model is almost exactly the same as the first model on both the training and the testing data. There is only a slight difference in the model's performance using the training data, where the AUC is equal to 0.987. After elaborating on the results of all three models, it can be seen that the accuracy and the AUC of all three models are almost the same for the testing sets. Therefore, we used the total execution time of the models. The execution times of the models 1 to 3 are 120, 122 and 118 seconds respectively. Therefore, as per our analysis, for this specific testing set, the best model is the 2-class SVM model. We also realized that there is no significant improvement when using the deep SVM model compared to the regular 2-class SVM model.

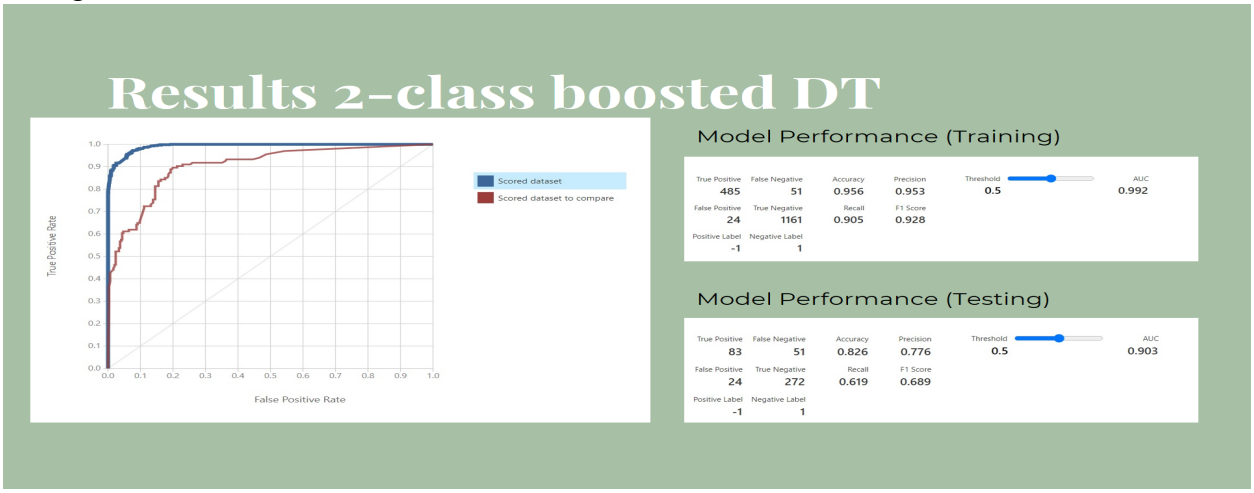

Fig. 6. Performance of the 2-class boosted decision tree model

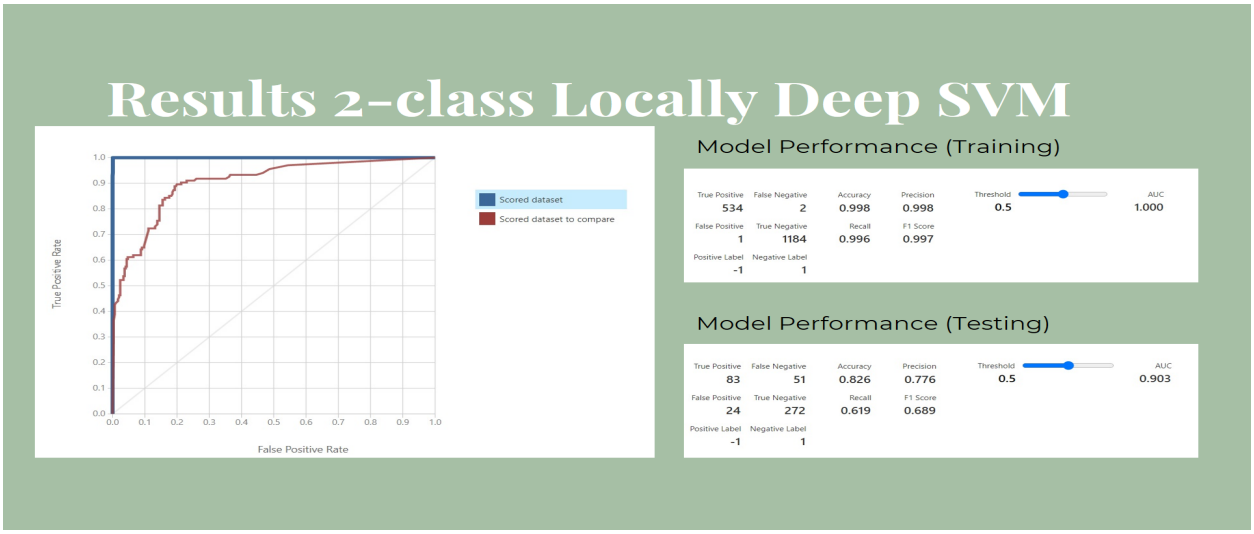

Fig. 7. Performance of the 2-class locally deep SVM model

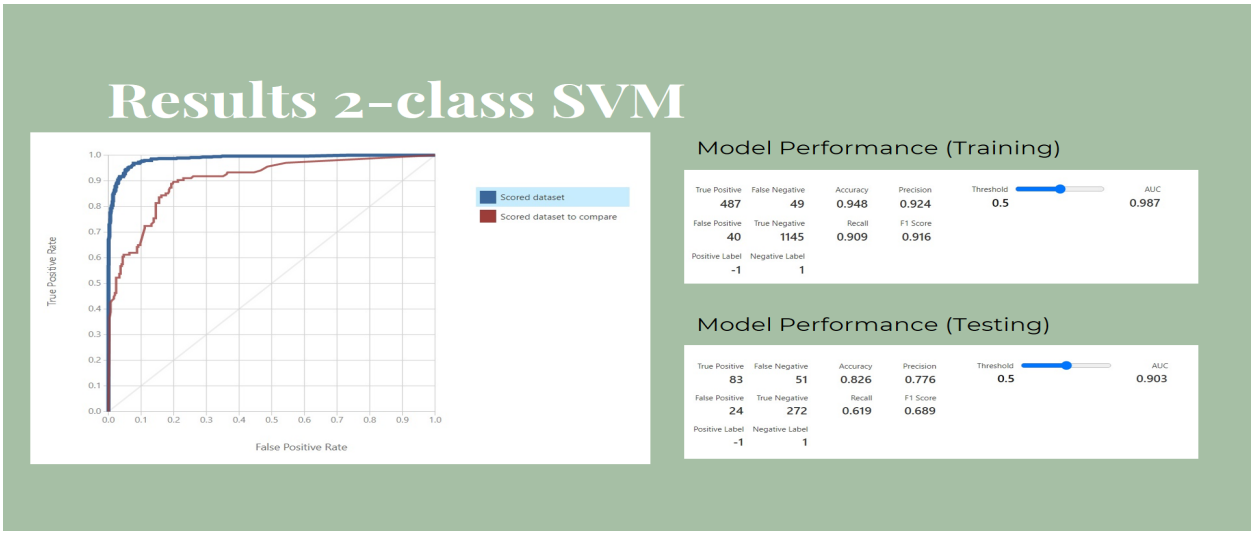

Fig. 8. Performance of the 2-class SVM model

After the implementation of the aforementioned models, we have to see if there are any relations between the classified positive and negative sentiments of the news headlines and the stock values of Facebook, DogeCoin, and Pfizer. Also, note that there are days when the stock market prices are not announced for Pfizer and Facebook during the weekends as opposed to DogeCoin, which is a cryptocurrency and can be traded any day of the week. Another point that must be taken into consideration is the fact that sometimes there are positive and negative sentiments in the news everyday and there are days when no sentiments or no stock prices are announced. In order to demonstrate this, we should take the classification results from Azure ML Studio, sort them by date and categorize them as positive and negative sentiments. The problem here is the output of the Azure ML Studio doesn't show the classification results for every input. However, by estimate, the results of the classification vs. the actual stock market price is illustrated in Figures 9-11. The estimation is based on the output of the evaluation module, where the total numbers of true positives and true negatives are shown. By looking at the results, we can see that

the positive sentiments have a positive correlation with the increase in adjusted stock price values of DogeCoin. When there is a sequence of positive sentiments in five executive days, we can see a considerable increase in the adjusted stock price. Almost the same can be seen in Pfizer stock prices, where after four consecutives days with positive news in the media, the adjusted stock price increases rapidly. On the other hand, the behaviour of the Facebook adjusted stock price is very different from the other two. Subsequent positive sentiments of the news had a positive impact on the adjusted stock price, but the fluctuation is higher compared to the other two companies. This shows that there is no meaningful relationship between the sentiments and facebook stock prices. We suspect that this is happening due to the previous occasions, when facebook, instagram and WhatsApp were down for almost a day on Oct 4th, 2021.
Overall, the results show that often consecutive positive sentiments in the media is correlated with an increase in stock price. In contrast, we have not found a correlation between negative sentiments and a decrease in the stock price.

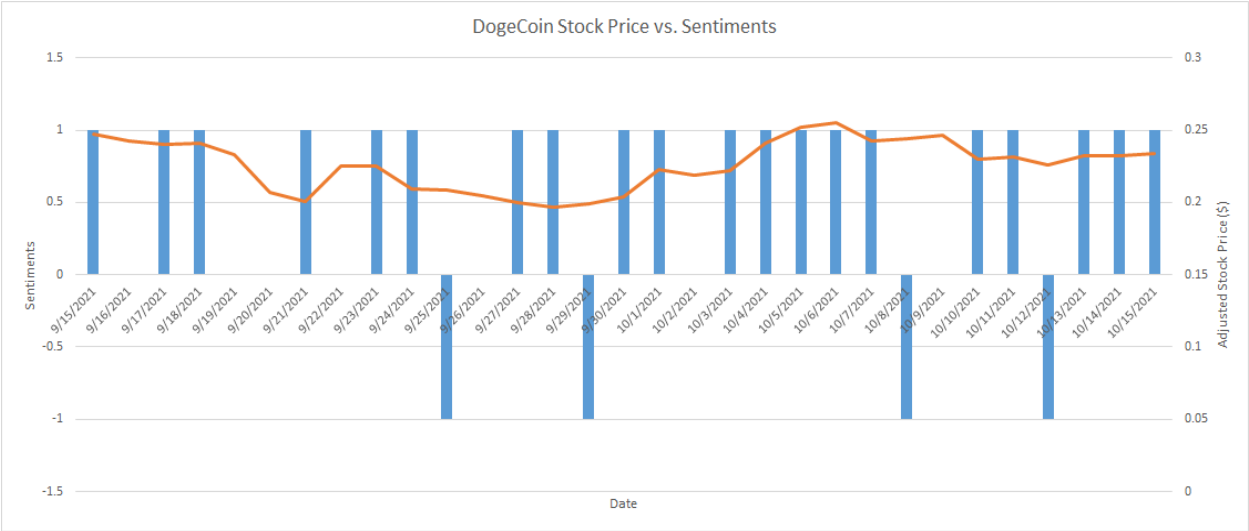

Fig.9. DogeCoin Stock Price vs. Sentiments

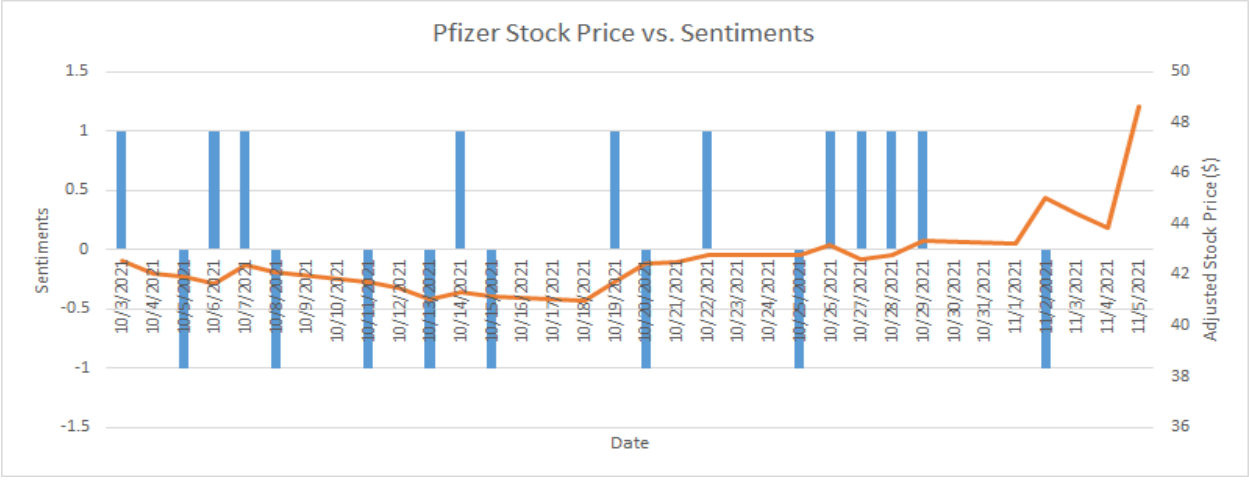

Fig.10. Pfizer Stock Price vs. Sentiments

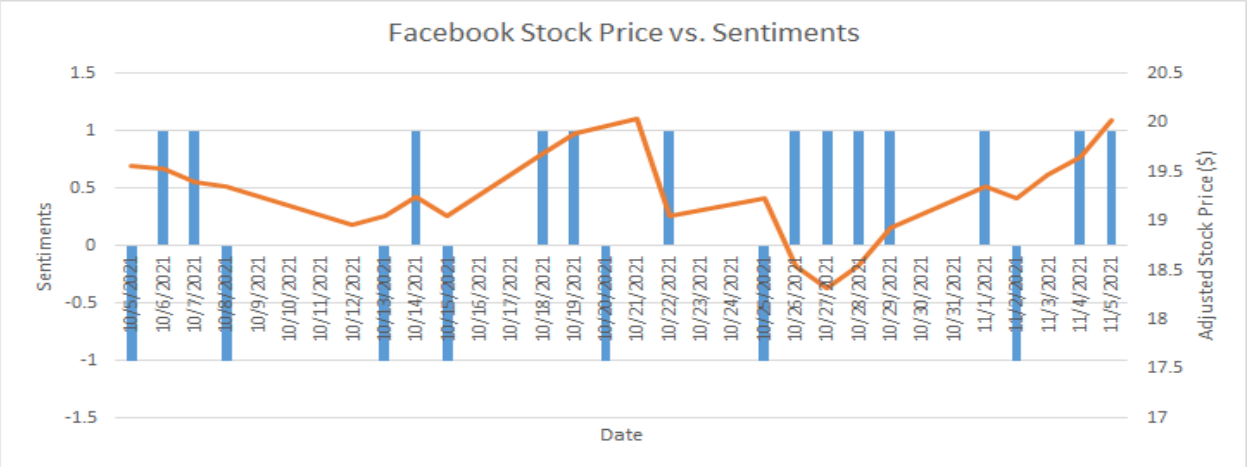

Fig.11. Facebook Stock Price vs. Sentiments

Appendix

************R code to normalize data************

```
# Map 1-based optional input ports to variables
dataset <- maml.mapInputPort(1) # class: data.frame

 # Separate the label and tweet text
sentiment_label <- dataset[[1]]
tweet_text      <- dataset[[2]]

 # Replace punctuation, special characters and digits with space
tweet_text <- gsub("[^a-z]", " ", tweet_text, ignore.case = TRUE)

 # Convert to lowercase
tweet_text <- sapply(tweet_text, tolower)
 data.set <- as.data.frame(cbind(sentiment_label,tweet_text), stringsAsFactors=FALSE)

# Select data.frame to be sent to the output Dataset port
maml.mapOutputPort("data.set")
```

References

Gandhi, R. (2018). Support Vector Machine - Introduction to Machine Learning Algorithms. Retrieved from https://towardsdatascience.com/support-vector-machine-introduction-to-machine-learning-algorithms-934a444fca47

Joy, C. A. (2020). Two Class Support Vector Machine. Retrieved from https://www.c-sharpcorner.com/article/two-class-support-vector-machine/

Yildrim, S. (2020). Gradient Boosted Decision Trees-Explained. Retrieved from https://towardsdatascience.com/gradient-boosted-decision-trees-explained-9259bd8205af

Zhang, X., Gronlund, C. J., Howell, J., Li, B., Lu, P., Gilley, S., … Takaki, J. (2021). Two-Class Boosted Decision Tree. Retrieved from https://docs.microsoft.com/en-us/azure/machine-learning/studio-module-reference/two-class-boosted-decision-tree

Zhang, X., Gronlund, C. J., Lu, P., Martens, J., Gilley, S., Parente, J., … Takaki, J. (2021). Two-Class Locally Deep Support Vector Machine. Retrieved from https://docs.microsoft.com/en-us/azure/machine-learning/studio-module-reference/two-class-locally-deep-support-vector-machine