CONSTRUCTOR UNIVERSITY

Text Analysis and Natural Language Processing

Data Science for Society and Business

Author: Akshatha.

Prof: Seeback, AndreasWilhelm, AdalbertF.X.

Date: May 17 2022

Text analysis and Natural Language Processing

Topic: How does the sentiment of social media posts affect the stock market?

Abstract:

The sentiment of social media posts has become an increasingly important factor in understanding and predicting the stock market trends. This document aims to explore the relationship between social media sentiment and stock market by analyzing the chatgpt generated posts. Though this study may not fully uncover the potential correlations between sentiment trends and stock market movements, as I had made collected limited rows of data. With this research aiming to provide valuable information for investors, traders, and policymakers interested in understanding the dynamics between social media sentiment and the stock market.

Table of Contents

Introduction:

The stock market is a complex system that is influenced by a variety of factors such as sentiment of social media posts from Twitter, Stocktwits, Telegram etc. Making the accurate predictions on stock price is challenging as the best time to invest or hold depends on the factors like interest rates, inflation, quarterly financial reports, change in supply and demand. Though many factors help in determining the stock price in the market, psychological factors such as users' sentiment regarding policy changes, new investments or natural disasters also greatly influence how the stock price change. For example, after Elon musk announced that he bought twitter, his new investment led to a sharp decline in stock price for Tesla in the second half of 2022. With lots of news, or online platform available it becomes challenging for investors to keep track of all the information that can impact their investments.

Methodology:

Using Text mining to analyze the sentiment of social media posts and their impact on social market.

Data Collection:

Data is collected from the Chatgpt for the analysis.

Data Set:

Data set is collected from chatgpt and it contains three columns

Field Name	Description	DType
Year	Year when the posts is posted it has data from 2015 to 2023.	int64
Company	Name of the company, about which post s posted	object
Corpus	Posts	object

Data Preprocessing and Cleaning:

- As per the below sns heatmap visualization Fig.1 there are no duplicates value found from the dataset.

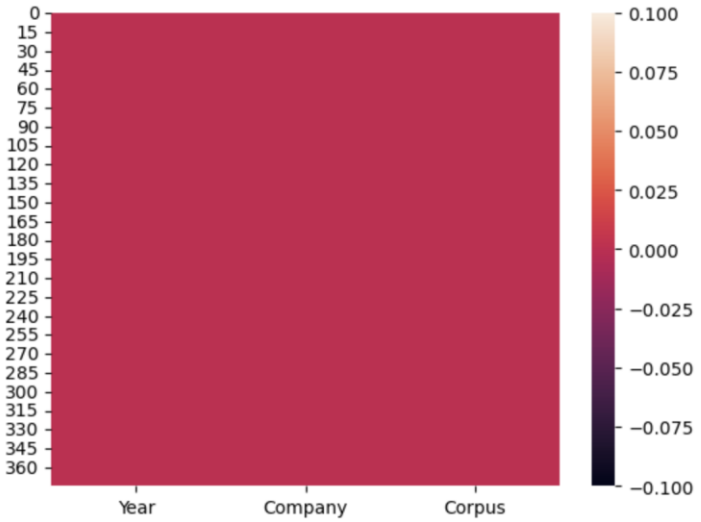


Fig.1 sns heatmap showing non null values for all the columns

- In the Fig.2 showing count of text lengths, there are 80 counts are available for the text length of 140. And the text length in the range from 80-130 counts are above 50. We do not have too short or too long texts in our datasets. And in the Fig3 shows the distribution of word counts.

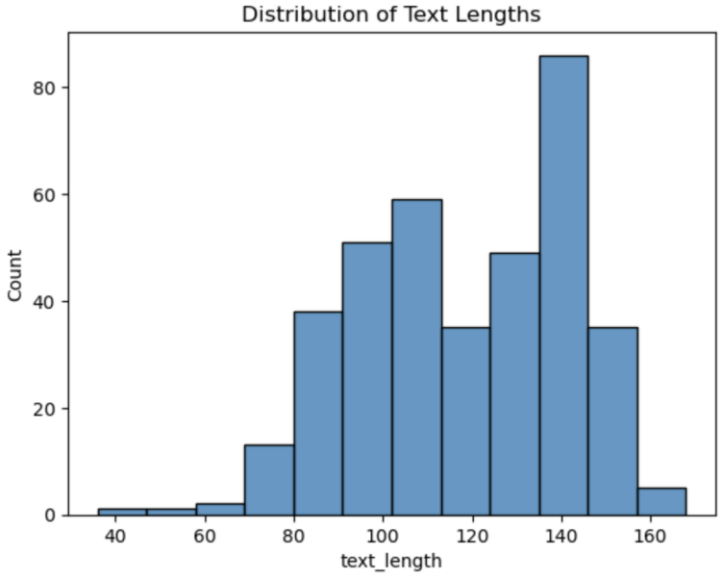


Fig.2 Histogram plot showing distribution of text lengths

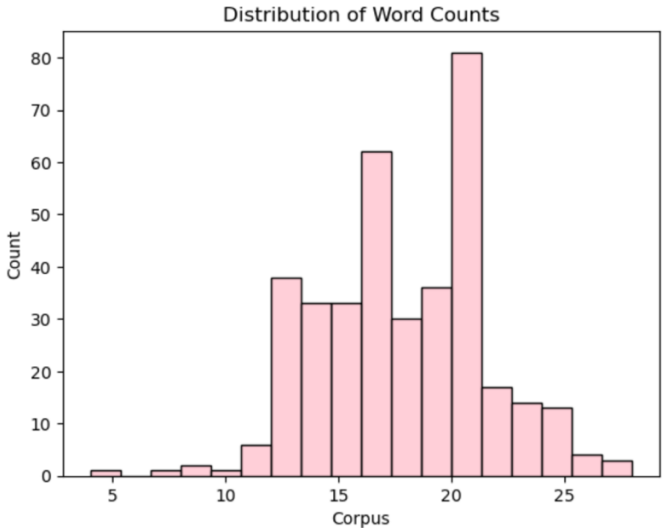


Fig.3 Histogram plot showing distribution of word count

- **RegexpTokenizer** class from the 'nltk.tokenize' module is used to split a text into smaller units in Corpus column.
- **Stop words** like I, me, he, what, into, between, then, once, not etc. are removed also removing infrequent words from tokenized text allowing to focus more on informative and meaningful words.
- Creating a frequency distribution which records the number of times each word has occurred.
- **Lemmatization** is performed to reduce the dimensionality of the data and improve the accuracy of text analysis by treating different inflected forms of word as a single entity.
- Below Fig.4 Showing the word cloud visualization, showing the most frequently occurring words in the dataset.

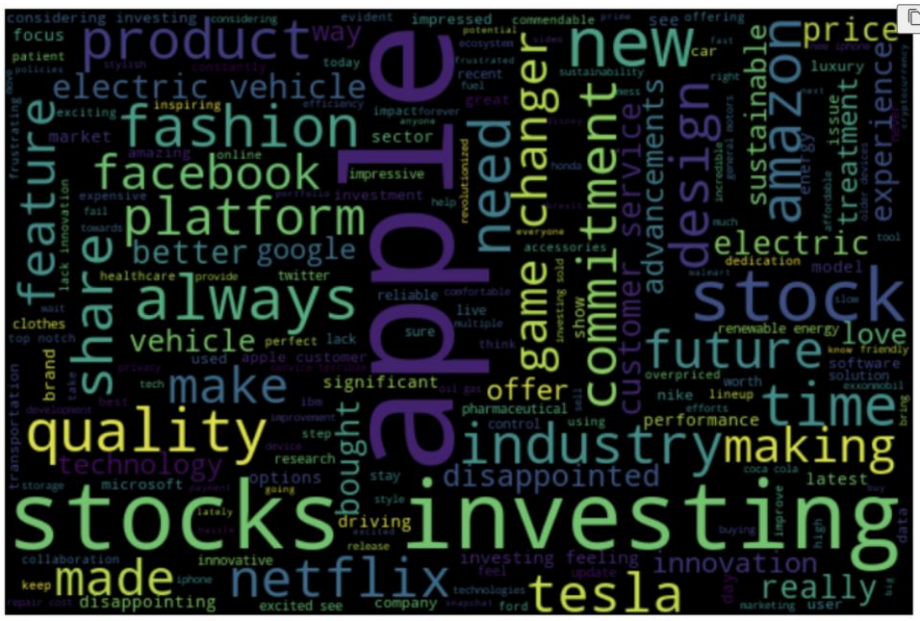


Fig 4 Word cloud in the dataset

- From the below Fig.5 we can see that investing, apple and stocks are the words are most frequently found in the dataset.

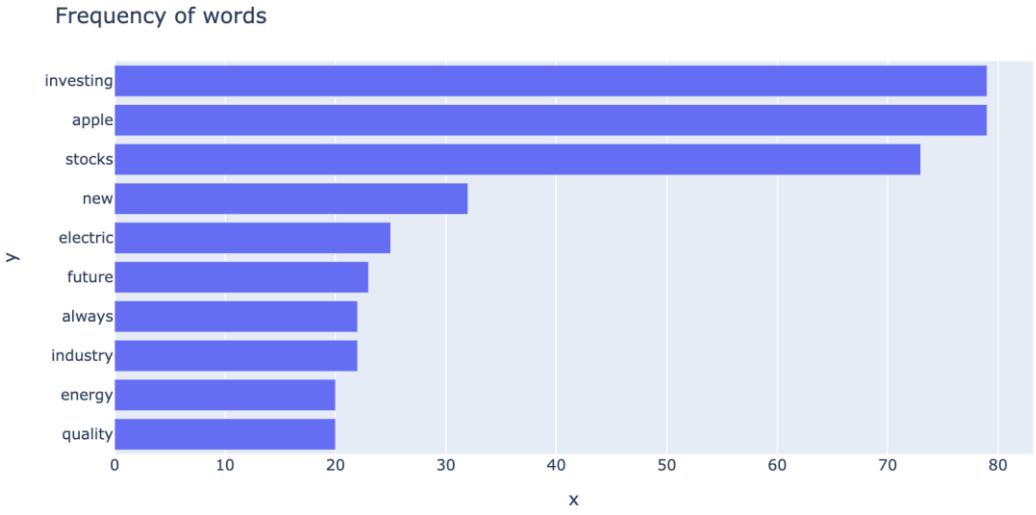


Fig 5 showing frequency of words

- Using the package SentimentIntensityAnalyzer(), polarity of the preprocessed text is found, which is giving Positive, Negative and Neutral and Compound scores and then

defining variable '**Sentiment'** if compound score >0 sentiment is considered as 'Positive', compound score <0 sentiment is considered as 'Negative' otherwise compound score =0 sentiment is 'Neutral'.

- Below Fig.6 showing the distribution of text by sentiment and we see that count is more for positive from my dataset.

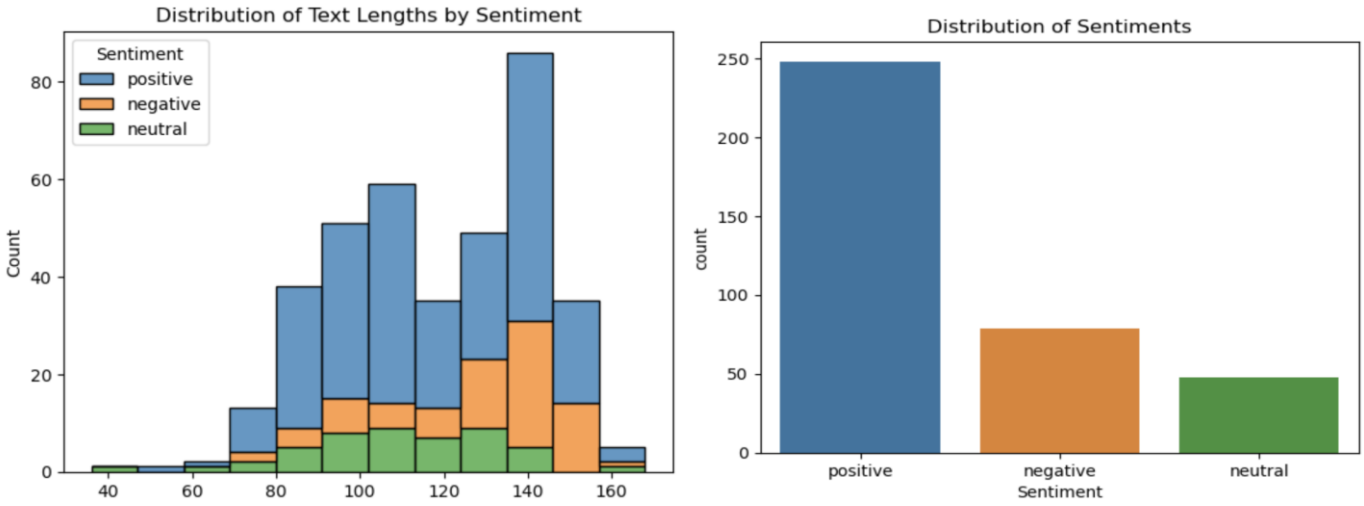


Fig 6. Distribution of texts sentiment

Model Building:

Word2Vec Model Training:

Word2Vec is a popular technique which used in learning word embeddings, which are dense vector representations of words in a continuous vector space. These word embeddings capture semantic and syntactic relationships between words, allowing the model to capture similarities and contextual information. Vectors found from this model, which captures the learned representation of words in the sentences, where each dimension represents a specific feature of the word. The value in the vectors shown in below Output.1 indicate the strength or presence of those features for each word. I have used Skip-Gram architecture with window size of 3, which determines the number of surrounding words used to predict the target words.

	0	1	2	3	4	5	6	7	8	9	...	20	
0	-0.029545	0.027451	0.452395	0.008651	-0.076219	0.019064	0.178948	0.170935	-0.391405	-0.181178	...	0.171906	-0.1241
1	-0.033010	0.029090	0.451499	0.005864	-0.076037	0.017898	0.179433	0.170322	-0.389684	-0.182456	...	0.174845	-0.1222
2	-0.031958	0.028306	0.449768	0.004852	-0.073904	0.022067	0.177395	0.169224	-0.389957	-0.177281	...	0.176189	-0.1233
3	-0.033342	0.031360	0.462797	0.002995	-0.084522	0.012296	0.170470	0.155902	-0.392495	-0.185635	...	0.173440	-0.1155
4	-0.036518	0.032676	0.447767	0.006038	-0.073016	0.016926	0.172778	0.165627	-0.386299	-0.181188	...	0.179276	-0.1206

Output.1 word2vec_df

Model Training and Evaluation:

Training the classification models like Random Forest Classifier, Logistic Regression, Decision Tree Classifier, and K-Nearest Neighbor Classifier using Word2Vec as input features and the sentiment labels from the training data.

```
Time taken to fit the Random Forest Classifier model with word2vec vectors: 0.042701005935668945
Time taken to fit the Logistic Regression model with word2vec vectors: 0.013882875442504883
Time taken to fit the Decision Tree Classifier model with word2vec vectors: 0.004781007766723633
Time taken to fit the KNeighbors Classifier model with word2vec vectors: 0.0006809234619140625
Accuracy for RFC Model: 0.7
Accuracy for Logistic Regression: 0.6733333333333332
Accuracy for Decision Tree Classifier: 0.5566666666666666
Accuracy for KNeighbors Classifier : 0.5666666666666667
```

Output 2. Model Output

From the above Output 2. We can see that time taken by the KNeighbor Classifier is small compared to all other model which is 0.00068 seconds. Accuracy for the Random Forest Classifier is 0.7 which means that model achieved 70% accuracy on average across the cross-validation folds. The higher the accuracy, the better the model performs in classifying the sentiment of the data.

Below Output 3 is the Classification Report for each of the trained model using trained data. Here I'm trying to find various metrices like Precision, Recall, F1-Score and Support to evaluate the performance of the models.

```
Classification report for the Random Forest Classifier Model :
              precision    recall  f1-score   support

    negative       1.00      0.98      0.99        61
     neutral       1.00      1.00      1.00        37
    positive       1.00      1.00      1.00       202

    accuracy                           1.00       300
   macro avg       1.00      0.99      1.00       300
weighted avg       1.00      1.00      1.00       300

Classification report for the Logistic Regression Model :
              precision    recall  f1-score   support

    negative       0.00      0.00      0.00        61
     neutral       0.00      0.00      0.00        37
    positive       0.67      1.00      0.80       202

    accuracy                           0.67       300
   macro avg       0.22      0.33      0.27       300
weighted avg       0.45      0.67      0.54       300
```

```
Classification report for the KNeighbors Classifier Model :
              precision    recall  f1-score   support

    negative       0.52      0.54      0.53        61
     neutral       0.45      0.14      0.21        37
    positive       0.77      0.86      0.81       202

    accuracy                           0.71       300
   macro avg       0.58      0.51      0.52       300
weighted avg       0.68      0.71      0.68       300

Classification report for the Decision Tree Classifier Model :
              precision    recall  f1-score   support

    negative       1.00      1.00      1.00        61
     neutral       1.00      1.00      1.00        37
    positive       1.00      1.00      1.00       202

    accuracy                           1.00       300
   macro avg       1.00      1.00      1.00       300
weighted avg       1.00      1.00      1.00       300
```

Output 3. Classification Report

Looking at the specific classification reports:

- **Random Forest Classifier (RFC) Model**:The RFC model shows high precision, recall, and F1-score for all classes, indicating excellent performance. It can predict the sentiment classes accurately.
- **Logistic Regression (LR) Model**:The LR model shows poor performance for the "negative" and "neutral" classes, with precision, recall, and F1-score of 0. This indicates that the LR model struggles to correctly predict these classes. However, it performs relatively well for the "positive" class.
- **KNeighbors Classifier (KNC) Model**:The KNC model shows moderate performance across all classes. Precision, recall, and F1-score vary for each class, with the highest scores observed for the "positive" class.
- **Decision Tree Classifier (DTC) Model**:The DTC model shows excellent performance with high precision, recall, and F1-score for all classes, indicating accurate predictions. It achieves 100% accuracy on the training data.

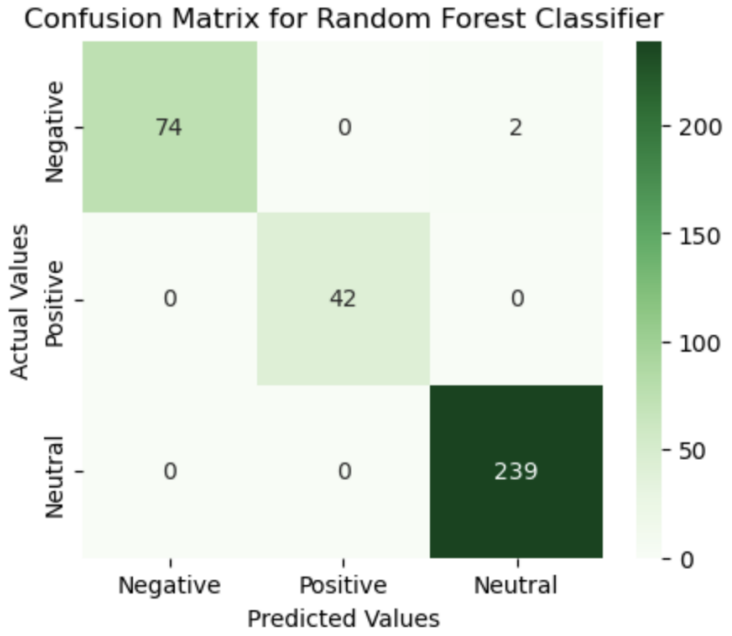
Confusion Matrix

Below Fig 7 shows the confusion matrices for each of the classifiers. The confusion matrix visualizes the performance of a classification model by showing the counts of true positive, true negative, false positive, and false negative predictions.

Random Forest Classifier:

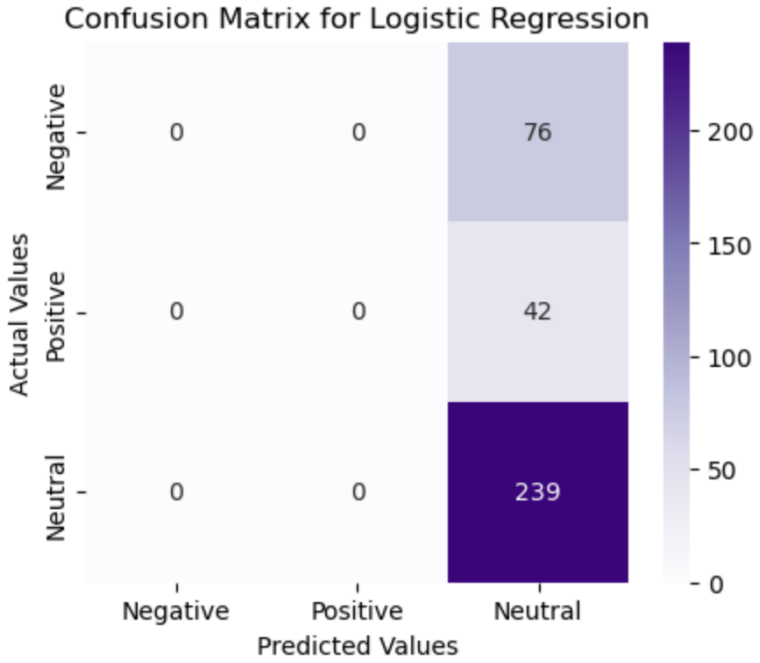
- Accuracy: High (almost perfect) as it correctly predicts the majority of samples.

- Precision: High for all classes, indicating that the model performs well in predicting positive, negative, and neutral classes.
- Recall: High for all classes, indicating that the model can effectively identify positive, negative, and neutral samples.
- Overall, the Random Forest Classifier seems to perform well based on the provided confusion matrix.



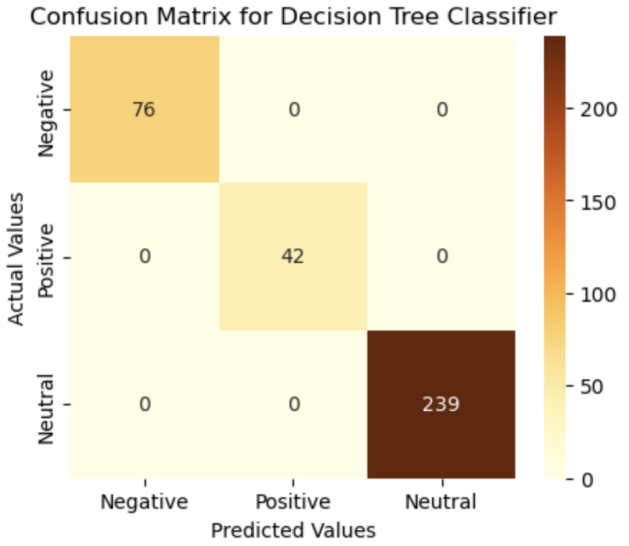
Logistic Regression:

- Accuracy: Low (only predicts neutral class correctly) as it fails to predict the positive and negative classes.
- Precision: Not applicable since no positive or negative samples were predicted.
- Recall: Not applicable since no positive or negative samples were predicted.
- The Logistic Regression model performs poorly based on the provided confusion matrix.



Decision Tree Classifier:

- Accuracy: High (almost perfect) as it correctly predicts the majority of samples.
- Precision: High for all classes, indicating that the model performs well in predicting positive, negative, and neutral classes.
- Recall: High for all classes, indicating that the model can effectively identify positive, negative, and neutral samples.
- Similar to the Random Forest Classifier, the Decision Tree Classifier performs well based on the provided confusion matrix.



K Nearest Neighbors (KNN) Classifier:

- Accuracy: Moderate, as it predicts a relatively high number of samples correctly but also misclassifies a substantial number of samples.
- Precision: Varies for different classes, with higher precision for the negative class compared to the positive and neutral classes.
- Recall: Varies for different classes, with higher recall for the positive class compared to the negative and neutral classes.
- The KNN Classifier shows moderate performance based on the provided confusion matrix.

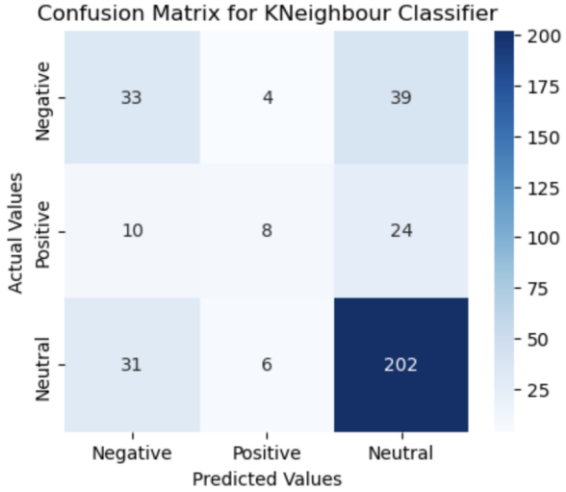


Fig 7. Plot showing confusion matrix for different models

Based on these observations, the Random Forest Classifier and the Decision Tree Classifier seem to perform well overall, with high accuracy, precision, and recall values for all classes.

Business Use Cases:

If any Investors or traders wants to invest in any of the stocks and they want to understand what is the sentiment or how this stock has market growth, this model can be used to predict the sentiment about the social posts about the stocks and can take decision if to worth investing or not.

As example, shown in the below Fig 8 for company ‘Apple’ making use of RFC model. We can see that most of the reviews are positive overall from 2015 till 2023.

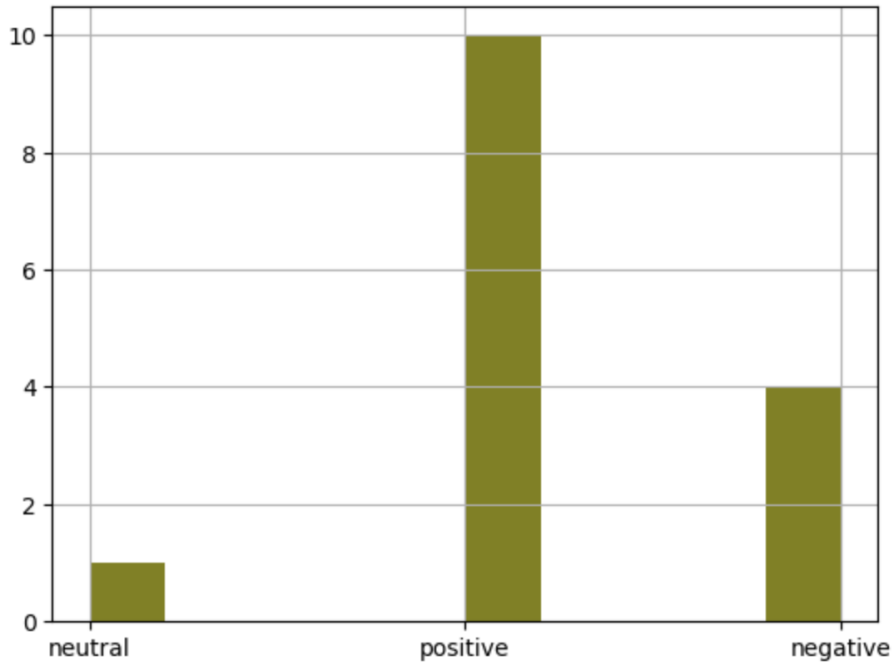


Fig 8 Sentiment for Apple stocks

In the below Fig 9 plot which shows for particular year, for 2019 Apple stocks has both positive and negative sentiment however more having positive posts, whereas in 2022 sentiment found to be only positive.

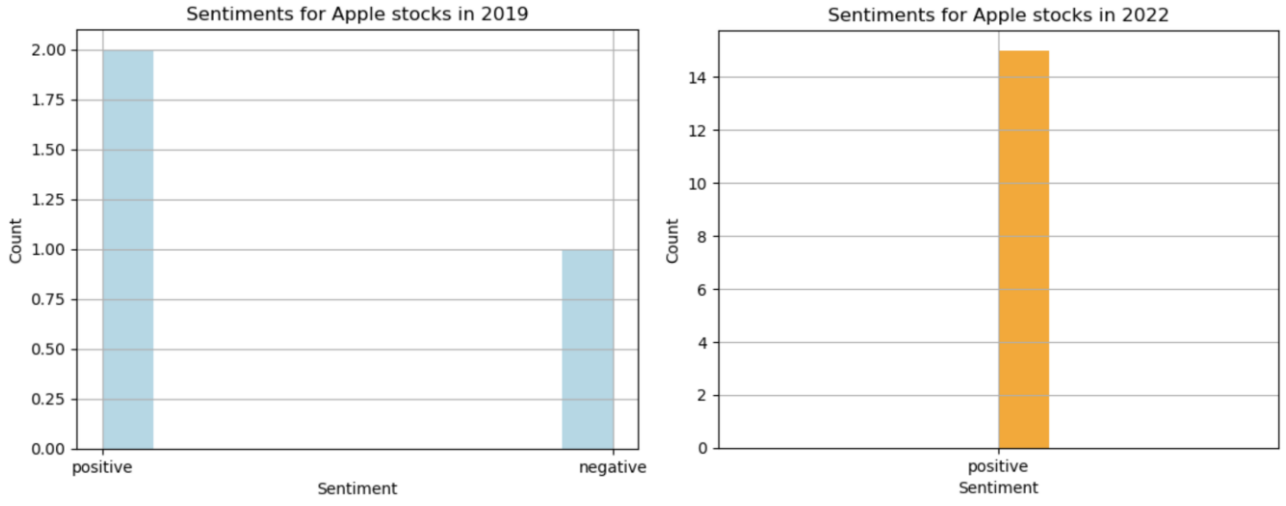


Fig 9 Sentiment for Apple stocks for the year 2016 and 2018

With relation to my plot in the Fig.9 and in the below figure from the website (https://www.counterpointresearch.com/apple-iphone-market-share-quarter/, https://www.asktraders.com/learn-to-trade/stock-trading/apple-vs-samsung/) shows apple has high increase in stock for the year 2022 compared to 2019. We believe that people now take into account social media posts when making investing decisions, although these don't really affect the stock market's rise or fall all that much on their own. However, considering people's propensity for convenience and comfort, relying on social media posts may someday play a large role in determining investing decisions.

Fig 10. Online images

Limitation:

The result may not be 100% accurate with the model I have implemented, as I have used only the few rows of data which is may result in overfitting of the model. This can be overcome by collecting more data.

Conclusion:

I have made use of classification models like Random Forest Classifier, Logistic Regression, Decision Tree Classifier, and K-Nearest Neighbor Classifier using Word2Vec. Among which Random Forest Classifier provided the good result and accuracy of the model was 70% followed by the Decision Tree Classifier. Whereas Logistic regression model showed poor performance, where it struggled to predict the classes.

The research seeks to answer the question of how positive or negative sentiment in social media posts can affect the stock market.

It is expected that social media posts have significant effect on the stock market. Positive posts are likely to result in an increase in stock price, while negative posts are likely to result in a decrease in stock prices. Neutral sentiment may have a negligible effect on stock market performance. Aiming to provide valuable insights into the relationship between social media sentiment and stock market performance. The result of this study can be used by investors and traders to make informed decisions regarding the purchase or sale of stocks. Emotional analytics work has the potential to improve pre-word processing through deep neural networks and can extend this research to neural convolution networks.

References:

https://towardsdatascience.com/nlp-in-the-stock-market-8760d062eb92

https://research.aimultiple.com/sentiment-analysis-stock-market/

https://www.researchgate.net/publication/360726066_NLP_in_Stock_Market_Prediction_A_Review

https://medium.com/nerd-for-tech/wallstreetbets-sentiment-analysis-on-stock-prices-using-natural-language-processing-ed1e9e109a37

https://chat.openai.com/

Code can be found in my GitHub Page
https://github.com/aakshathak/Text_analysis_and_NLP/blob/main/text_analysis.ipynb