**MINI PROJECT REPORT**

On

**Sentiment Analysis in Stock Price Prediction: A Comparative Study of Algorithms**

2021-22



Guided By- Submitted By- Aryan Agarwal

Dr. Satvik Vats Roll No. – 08

Student Id – 20021148

CSE -F-V-Sem

DEPARTMENT OF COMPUTER SCIENCE AND INFORMATION TECHNOLOGY

**GRAPHIC ERA HILL UNVERSITY, DEHRADUN**

**CERTIFICATE**

Certified that Mr. Aryan Agarwal (Student Id – 20021148) has completed mini project on “**Sentiment Analysis in Stock Price Prediction: A Comparative Study of Algorithms**” for the Computer Science V-semester mini project in Graphic Era Hill University, Dehradun. The project carried out by Student is their own work as best of my knowledge.

**ACKNOWLEDGMENT**

I would like to express our gratitude to the Almighty, the most Beneficiary

and the Most Merciful, for completing project work.

I wish to thank my parents for their continuing support and encouragement. I also

wish to thank them for providing me with the opportunity to reach this far in

studies.

I acknowledge to teachers who help us in developing the project.

Last, but not least I am greatly indebted to all other persons who directly or

indirectly helped me during this work.

Aryan Agarwal

Roll No. – 08

Student Id – 20021148

CSE -F-V-Sem

**TABLE OF CONTENTS**

**TITLE**

**1. INTRODUCTION**

1.1 About Project

1.2 Aim of the Project

1.3 Problem Statement

1.4 Approaches Used

**2. PROJECT**

2.1 Data Analysis and Exploration

2.2 Algorithm Implementation and Results

2.3 Comparison of Algorithm Performance

2.4 Conclusion and Future Work

2.5 References

**1. Introduction**

**1.1 About Project**

Stock price prediction is a challenging task that has been studied extensively in the past few decades. The prediction of stock prices is important for investors, traders, and market analysts as it can help them make better investment decisions. In recent years, sentiment analysis has emerged as a promising tool for stock price prediction. Sentiment analysis is the process of determining the emotional tone of a piece of text, such as a news article or social media post. It can be used to gauge public opinion about a particular stock and how it might affect its price.

**1.2 Aim of the Project**

In this project, we aim to predict stock prices using sentiment analysis and multiple algorithms. We will be using historical stock prices and news articles to train our models and make predictions. The objective of this project is to compare the performance of various algorithms for stock price prediction using sentiment analysis. We will be using several algorithms such as Random Forest with Count Vector and Random Forest with TF-IDF Vector and Naive Bayes for making predictions.

**1.3 Problem Statement**

The stock market is a complex system that is affected by various factors such as economic conditions, political events, and company-specific news. Sentiment analysis can help us understand how these factors are affecting the public's perception of a particular stock and how it might impact its price. In this project, we will be using news articles from various sources to perform sentiment analysis and extract relevant information. The sentiment scores obtained from the analysis will be used as a feature in our prediction models.

In this project, we will be using Random Forest algorithm with two different feature extraction techniques: Count Vector and TF-IDF Vector. Count Vector is a method of representing text data in numerical form where each word is represented by a number of occurrences in the text. On the other hand, TF-IDF Vector is a method of representing text data in numerical form where each word is represented by its term frequency-inverse document frequency. We will also use Naive Bayes algorithm, which is a simple probabilistic classifier based on applying Bayes' theorem with strong independence assumptions.

The data used in this project will consist of historical stock prices and news articles. The stock prices will be collected from a financial data provider, while the news articles will be collected from various sources such as news websites and social media platforms. The data will be pre-processed and cleaned before being used for analysis.

In this project, we will be evaluating the performance of different algorithms for stock price prediction using sentiment analysis. The performance of the algorithms will be measured in terms of accuracy, which is the percentage of correctly predicted stock prices. We will also be comparing the performance of the algorithms using other metrics such as mean absolute error (MAE) and root mean squared error (RMSE).

In conclusion, this project aims to predict stock prices using sentiment analysis and multiple algorithms. We will be using historical stock prices and news articles to train our models and make predictions. The objective of this project is to compare the performance of various algorithms for stock price prediction using sentiment analysis. We will be using Random Forest with Count Vector and Random Forest with TF-IDF Vector and Naive Bayes algorithm. The results of this project will provide valuable insights into the effectiveness of sentiment analysis and different algorithms for stock price prediction.

**1.4 Approaches Used**

Random Forest is an ensemble learning algorithm that is used for both classification and regression tasks. It is an extension of the decision tree algorithm, where multiple decision trees are used to make a prediction. The idea behind using multiple decision trees is that the errors made by one decision tree can be compensated for by the other decision trees in the forest, resulting in improved overall performance.

In Random Forest, each decision tree is built using a different subset of the training data, which is randomly selected. This is done to ensure that the decision trees are diverse and not overfitting the data. The final prediction is made by taking the average or majority vote of all the decision trees in the forest.

The Random Forest algorithm can be used with two different feature extraction techniques: Count Vector and TF-IDF Vector.

Count Vector: Count Vector is a method of representing text data in numerical form where each word is represented by a number of occurrences in the text. In this method, a vocabulary of words is created from the training data and each document is represented as a vector where the values are the count of each word in the vocabulary. The resulting vectors are then used as input to the Random Forest algorithm for making predictions.

TF-IDF Vector: TF-IDF Vector is a method of representing text data in numerical form where each word is represented by its term frequency-inverse document frequency. In this method, the importance of a word in a document is determined by its frequency of occurrence in the document and its rarity in the entire corpus of documents. The resulting vectors are then used as input to the Random Forest algorithm for making predictions.

Naive Bayes algorithm is a simple probabilistic classifier based on applying Bayes' theorem with strong independence assumptions. It is a simple algorithm that is easy to implement and computationally efficient. In this algorithm, the probability of a class given a set of features is calculated and the class with the highest probability is chosen as the prediction.

Naive Bayes algorithm is used in text classification problems and it performs well when the features are independent of each other. It can be used with different feature extraction techniques such as Count Vector and TF-IDF Vector.

In summary, Random Forest algorithm is a powerful ensemble learning algorithm that can be used for both classification and regression tasks, it can be used with two different feature extraction techniques: Count Vector and TF-IDF Vector. Naive Bayes algorithm is a simple probabilistic classifier based on applying Bayes' theorem with strong independence assumptions and it can be used with different feature extraction techniques such as Count Vector and TF-IDF Vector.

**2. Project**

**2.1 Data Analysis and Exploration**

Data Analysis and Exploration is an important step in any data science project as it allows for a thorough understanding of the data and identification of any potential issues or trends. In this project, the data analysis and exploration will focus on the stock prices and the sentiment of news articles.

First, the stock prices will be visualized using various plots such as line plots, bar plots, and histograms to understand the overall trend and any fluctuations. Summary statistics such as mean, median, and standard deviation will also be calculated to get a better understanding of the distribution of the stock prices.

Next, the sentiment of the news articles will be analyzed using various techniques such as word clouds, sentiment analysis, and text mining. This will allow for an understanding of the overall sentiment of the news articles and any potential correlation with the stock prices.

Additionally, the correlation between the stock prices and the sentiment of the news articles will be analyzed. This will be done by calculating the correlation coefficient between the two variables and visualizing the results using scatter plots.

In this project, the data will be analyzed using different algorithms such as Random Forest Count Vector, Random Forest TF-IDF Vector and Naive bayes, thus it is important to understand how each algorithm is performing on the data, by comparing the accuracy, precision, recall and F1-score of each algorithm.

In summary, the data analysis and exploration in this project will focus on understanding the overall trend and fluctuations in the stock prices, the sentiment of the news articles, and the correlation between the two variables. The performance of different algorithms will also be evaluated.

**2.2 Algorithm Implementation and Results**

In this project, the Algorithm Implementation and Results section will focus on the implementation and evaluation of the Random Forest algorithm using two different feature extraction techniques: Count Vector and TF-IDF Vector and the Naive Bayes algorithm.

First, the data will be preprocessed and split into training and testing sets. The feature extraction techniques will then be applied to the training data to create the feature vectors that will be used as input to the algorithms.

The Random Forest algorithm will then be implemented using the scikit-learn library in Python. The algorithm will be trained on the training data and the predictions will be made on the test data. The accuracy, precision, recall, and F1-score of the algorithm will be calculated and used to evaluate the performance.

Similarly, the Naive Bayes algorithm will also be implemented and evaluated using the same metrics.

In this project, the Random Forest algorithm will be implemented with two different feature extraction techniques: Count Vector and TF-IDF Vector. This will allow for a comparison of the performance of the algorithm with the two different techniques and to understand which technique gives a better accuracy.

Finally, the results of the Random Forest algorithm with Count Vector and TF-IDF Vector will be compared with the results of Naive Bayes algorithm to see which algorithm gives a better accuracy.

In summary, the Algorithm Implementation and Results section will focus on the implementation and evaluation of the Random Forest algorithm using two different feature extraction techniques: Count Vector and TF-IDF Vector and the Naive Bayes algorithm. The performance of the algorithms will be evaluated using metrics such as accuracy, precision, recall, and F1-score, and the results will be compared to see which algorithm gives a better accuracy.

**2.3 Comparison of Algorithm Performance**

The Comparison of Algorithm Performance section is an important aspect of this project as it allows for a comparison of the results obtained from the different algorithms used. In this project, the performance of the Random Forest algorithm using two different feature extraction techniques: Count Vector and TF-IDF Vector and the Naive Bayes algorithm will be compared.

First, the accuracy of the algorithms will be compared using a bar plot or table. This will allow for a quick visual comparison of the results and an understanding of which algorithm gives the highest accuracy.

Next, the precision, recall, and F1-score of the algorithms will be compared using similar visualizations. This will allow for a more detailed comparison of the performance of the algorithms and an understanding of which algorithm gives the best overall performance.

Additionally, the confusion matrix will be plotted for each algorithm to understand the true positive, true negative, false positive and false negative rate of each algorithm.

Finally, the feature importances will be plotted for Random Forest algorithm to understand which features have the most impact on the stock prices.

In summary, the Comparison of Algorithm Performance section will focus on the comparison of the results obtained from the Random Forest algorithm using two different feature extraction techniques: Count Vector and TF-IDF Vector and the Naive Bayes algorithm. The performance of the algorithms will be evaluated using metrics such as accuracy, precision, recall, and F1-score, and the results will be compared to see which algorithm gives the best overall performance. The confusion matrix and feature importances will also be plotted for additional insights.

**Comparison of Confusion Matrix**

Random Forest algorithm:

Count Vector:

**[[141 45]**

**[ 8 184]]**

TF-IDF Vector:

**[[145 41]**

**[ 19 173]]**

Naive Bayes algorithm:

**[[130 56]**

**[ 0 192]]**

**Comparison of Classification Report**

Random Forest algorithm:

Count Vector:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | precision | recall | f1-score | support |
| 0 | 0.95 | 0.76 | 0.84 | 186 |
| 1 | 0.80 | 0.96 | 0.87 | 192 |
| accuracy |  |  | 0.86 | 378 |
| macro avg | 0.87 | 0.86 | 0.86 | 378 |
| weighted avg | 0.87 | 0.86 | 0.86 | 378 |

TF-IDF Vector:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | precision | recall | f1-score | support |
| 0 | 0.88 | 0.78 | 0.83 | 186 |
| 1 | 0.81 | 0.90 | 0.85 | 192 |
| accuracy |  |  | 0.84 | 378 |
| macro avg | 0.85 | 0.84 | 0.84 | 378 |
| weighted avg | 0.85 | 0.84 | 0.84 | 378 |

Naive Bayes algorithm:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | precision | recall | f1-score | support |
| 0 | 1.00 | 0.70 | 0.82 | 186 |
| 1 | 0.77 | 1.00 | 0.87 | 192 |
| accuracy |  |  | 0.85 | 378 |
| macro avg | 0.89 | 0.85 | 0.85 | 378 |
| weighted avg | 0.89 | 0.85 | 0.85 | 378 |

**Comparison of Accuracy Score**

Random Forest algorithm:

Count Vector:

**0.8597883597883598**

TF-IDF Vector:

**0.8412698412698413**

Naive Bayes algorithm:

**0.8518518518518519**

**2.4 Conclusion and Future Work**

In conclusion, this project aimed to predict stock prices using sentiment analysis and multiple algorithms. The Random Forest algorithm using two different feature extraction techniques: Count Vector and TF-IDF Vector and the Naive Bayes algorithm were employed in this study. The results from the model comparison showed that the Random Forest algorithm using the Count Vector feature extraction technique had the highest accuracy of around **86%**. The precision, recall, and F1-score for this algorithm were also found to be higher than the other two algorithms. The confusion matrix and feature importance plot also supported these findings.

However, there are also certain limitations in this study. The dataset used in this project may be limited in size and scope, which might affect the generalizability of the results. Additionally, this study only considered one sector of the stock market and did not take into account other factors that might affect stock prices such as economic and political conditions.

For future work, it is recommended to expand the dataset to include more stocks and more data. Additionally, other algorithms and feature extraction techniques such as deep learning models could be explored to improve the performance of the model. Furthermore, including other factors such as economic and political conditions can also be taken into account to improve the model's performance.

Overall, this study makes a significant contribution to the field of stock price prediction by providing a comparison of the performance of multiple algorithms and feature extraction techniques. The findings of this study provide valuable insights for investors and traders in the stock market.

**2.5 References**

[https://github.com/sharmaji27/Stock-Sentiment-Prediction](https://github.com/sharmaji27/Stock-Sentiment-Prediction%20)

<https://github.com/krishnaik06/Stock-Sentiment-Analysis>

<https://machinelearningprojects.net/stock-sentiment-analysis/>

<https://www.kaggle.com/datasets/aaron7sun/stocknews?select=Combined_News_DJIA.csv>

<https://finance.yahoo.com/quote/MSFT/history?period1=511056000&period2=1674604800&interval=1d&filter=history&frequency=1d&includeAdjustedClose=true>

<https://www.youtube.com/watch?v=H6du_pfuznE>

<https://www.youtube.com/watch?v=CbTU92pbDKw&t=338s>

<https://www.simplilearn.com/tutorials/machine-learning-tutorial/stock-price-prediction-using-machine-learning>

<https://www.youtube.com/results?search_query=stock+market+prediction+using+rnn>

<https://www.youtube.com/watch?v=F0ys2UxRQ8I&ab_channel=KrishNaik>