Mini Project Presentation
on

"Stock Price Prediction Using Sentiment Analysis of News Headlines"

Presented by

(Subham Kumar - A082, Jeet Guhathakurta - A056, Aashvi Trivedi - A089)

Under the Guidance of

(Prof Sandip Bankar)

Project Title

Stock Price Prediction Using Sentiment Analysis of News Headlines

Introduction

- Stock market data analysis needs the help of artificial intelligence and data mining techniques.
- This study basically shows the effect of emotion classification of financial news to the prediction of stock market prices.
- In order to find correlation between sentiment predicted from news and original stock price and to test efficient market hypothesis.

Index

Literature

Author Name	Title	Methods Used	Result Analysis	Advantages	Disadvantages
Oshi Gupta	Sentiment Analysis of News Headlines For Stock Trend Prediction	Bag of Words, Dictionary approach, Random Forest, Naive Bayes Classifier, Sentiment Analysis, NLP	Random forest and Naive Bayes was used wherein both gave same accuracy.The latter gave 100% precision	Has relevant methods used to obtain basic understanding of stock prediction	Doesn't include BERT to set benchmark and compare other methods
László Nemes & Attila Kiss	Prediction of stock values changes using sentiment analysis of stock news headlines	TextBlob, NLTK-Vader Lexicon Tools, BERT & RNN Model, Sentiment Analysis, NLP	Out of all methods used, BERT was the best and was set as a benchmark.	Comparison of multiple methods to obtain best possible outcome	
Anurag Nagar & Michael Hahsler	Using Text and Data Mining Techniques to extract Stock Market Sentiment from Live News Streams	NLP, Sentiment Analysis, Data Mining using R Programming Language	Estb. the correlation of their method with actual stock price movement. Collected top 20 most relevant headlines before creating text corpus		Does not implement multiple methods to compare results obtained.

Problem Statement

To derive relevant information for investors from a large chunk of Top news headlines using Natural Processing Language to predict the movement of stock market.

Objective

The project gathers non-quantifiable information about a company from content such as financial news articles and other sources, and then predicts its stock trend based on these headlines using machine learning techniques such as:

K-means clustering

Decision Trees

Random Forests

Naive Bayes

Support Vector Machines.

The main objectives:-

1. Sentiment Analysis of Stocks - It assists investors and businesses in understanding the social sentiment of their portfolios and creates a data dictionary of news headlines and label the data having positive, negative or neutral sentiments.
2. Comparing Random Forests and Naïve Bayes - classify the text and evaluate each algorithm's precision
3. Stock Values - Stock value is impacted by market capitalization size. A stock's worth is influenced by its market acceptance or perceived value.
4. Predict future stock movement for different stocks - Once the data dictionary is loaded on the Sentiment Analysis program, it will be capable of producing forecasts of the stock market based on previously labelled data using supervised machine learning.

Possible Algorithms

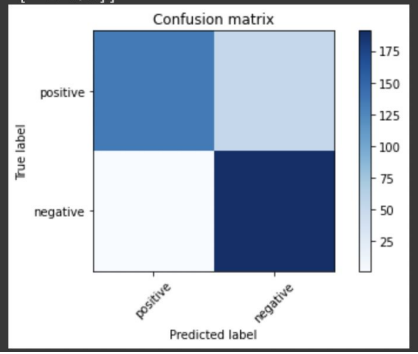
- Naive Bayes
- Random Forest with CountVectorizer
- Random Forest with TF-IDF
- TextBlob
- NLTK - VADER Lexicon
- Recurrent Neural Network (RNN)
- Bidirectional Representations from Transformers (BERT)

Result

- Random Forest with CountVectorizer

```
[34] ## Import library to check accuracy
     from sklearn.metrics import classification_report,confusion_matrix,accuracy_score

     score=accuracy_score(test['Label'],predictions)
     print(score)

     0.8597883597883598

[36] report=classification_report(test['Label'],predictions)
     print(report)

                   precision    recall  f1-score   support

                0       0.99      0.72      0.83       186
                1       0.79      0.99      0.88       192

         accuracy                           0.86       378
        macro avg       0.89      0.86      0.86       378
     weighted avg       0.89      0.86      0.86       378
```

```
from sklearn.metrics import classification_report,confusion_matrix,accuracy_score
plt.figure()
cm=confusion_matrix(test['Label'],predictions)
print(cm)
plot_confusion_matrix(cm)
plt.show()

[[134  52]
 [  1 191]]
```



Result

- Random Forest with Tf-idf

```
score=accuracy_score(test['Label'],predictions)
print(score)
```

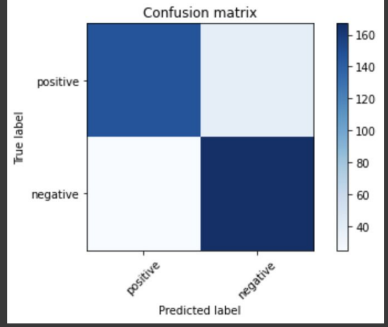
```
0.8306878306878307
```

```
[55] report=classification_report(test['Label'],predictions)
     print(report)
```

	precision	recall	f1-score	support
0	0.85	0.79	0.82	186
1	0.81	0.87	0.84	192
accuracy			0.83	378
macro avg	0.83	0.83	0.83	378
weighted avg	0.83	0.83	0.83	378

```
plt.figure()
cm=confusion_matrix(test['Label'],predictions)
print(cm)
plot_confusion_matrix(cm)
plt.show()
```

```
[[147  39]
 [ 25 167]]
```



Result

- Using Multinomial Naïve Bayes

```
score=accuracy_score(test['Label'],predictions)
print(score)
```

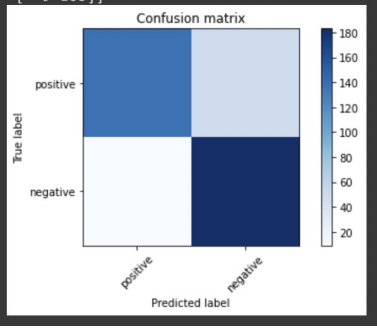
```
0.8465608465608465
```

```
report=classification_report(test['Label'],predictions)
print(report)
```

```
              precision    recall  f1-score   support

           0       0.94      0.74      0.83       186
           1       0.79      0.95      0.86       192

    accuracy                           0.85       378
   macro avg       0.86      0.84      0.84       378
weighted avg       0.86      0.85      0.84       378
```

```
from sklearn.naive_bayes import MultinomialNB
nb=MultinomialNB()
nb.fit(traindata_x,train['Label'])

predictions = nb.predict(test_data)
matrix=confusion_matrix(test['Label'],predictions)
print(matrix)

plot_confusion_matrix(matrix)
```

```
[[137  49]
 [  9 183]]
```



Summary

Algorithms		Precision	Recall	f1-score	Accuracy Score
Random Forest with Countvectorizer	0	0.99	0.72	0.83	0.86
	1	0.79	0.99	0.86	
Random Forest with TF-IDF	0	0.85	0.79	0.82	0.83
	1	0.81	0.87	0.84	
Naive Bayes	0	0.94	0.74	0.83	0.85
	1	0.79	0.95	0.86	

Future Scope

- Our future work could further include deep learning techniques like Recurrent Neural Network (RNN) and Bidirectional Encoder Representations from transformers (BERT) .
- We have considered an existing dataset for now but in future we can extract financial news in real time from platforms like **Bloomberg** and perform sentiment analysis on them.
- We can also use **Twitter developer API** to extract investors' tweet in real time to determine their sentiment to predict the movement of stock.

References

Journal Paper

[1] R. Goonatilake and S. Herath, The volatility of the stock market and news, International Research Journal of Finance and Economics, 2007, 11: 53-65.

[2] László Nemes & Attila Kiss (2021) Prediction of stock values changes using sentiment analysis of stock news headlines, Journal of Information and Telecommunication, 5:3, 375-394, DOI: 10.1080/24751839.2021.1874252

[3] Oshi Gupta, Sentiment Analysis of News Headlines for Stock Trend Prediction, International Journal for Research Trends and Innovation, 2020, Vol. 5, Issue 12, ISSN: 2456-3315

[4] W.B. Yu, B.R. Lea, and B. Guruswamy, A Theoretic Framework Integrating Text Mining and Energy Demand Forecasting, International Journal of Electronic Business Management. 2011, 5(3): 211-224

[5] Yauheniya Shynkevich, T.M. McGinnity, Sonya Coleman, Ammar Belatreche, Predicting Stock Price Movements Based on Different Categories of News Articles, 2015 IEEE Symposium Series on Computational Intelligence

References

Conference Paper

[1] Anurag Nagar, Michael Hahsler, Using Text and Data Mining Techniques to extract Stock Market Sentiment from Live News Streams, IPCSIT vol. XX (2012) IACSIT Press, Singapore

[2] J. Bean, R by example: Mining Twitter for consumer attitudes towards airlines, In Boston Predictive Analytics Meetup Presentation, 2011

Website

[3] http://www.iraj.in/journal/journal_file/journal_pdf/14-481-153485282984-86.pdf

[4] https://arxiv.org/pdf/1607.01958.pdf

Thank you!