Stock Sentiment Analysis

JUNE 20, 2025

Enrollment No. 23112090
Authored by: Aridaman Singh
Chemical Engineering

REPORT

STOCK SENTIMENT ANALYSIS USING MACHINE LEARNING MODEL

Project Overview

Stock Sentiment Analysis uses machine learning model to analyze the news headlines present on internet and generate features like subjectivity, polarity, negative, neutral, FinBERT feature, GloVe feature using advanced natural language processing (NLP) techniques which is used to train the ML model and provide relationship between news sentiment and movement in stock prices.
Using the predicted label from ML model we get a trading strategy for the stocks.
I have done news sentiment analysis for 3 companies 'TESLA', 'AMAZON' and 'META'.

Methodology

1. Data Collection:

First collect the data from reliable sources for sentiment analysis from different websites like market insider, finviz etc. with the help of web scraping. Then merge the data collected from different websites into one dataframe.
Using yahoo finance library present in python collect the historical stock data.

```
for page in range(1,330): # Loop for a range of pages (1 to 330)

  market_insider_url= url2 + ticker + '-stock?p=' + str(page)  # here we construct the URL for e
  time.sleep(2)
  Response2= requests.get(market_insider_url)
  Response2.raise_for_status()
  html= Response2.text

  soup = BeautifulSoup(html, 'lxml')  # Parse the HTML content using BeautifulSoup.

  articles = soup.find_all('div', class_="latest-news__story") # Finding all news articles on th


  for article in articles:
      Datetime = article.find('time', class_="latest-news__date").get('datetime')
      news1= article.find('a', class_= "news-link").text
      L.append([ticker,Datetime,news1])
      df1= pd.DataFrame(L, columns=df1.columns) # storing data of the list into dataframe df1.
```

This is the code for web scraping 300 pages of website market insider.

2. Data Preprocessing:

Text news collected is first cleaned and then preprocessed by removing stopwords, punctuation, lemmitization and performing tokenization using spacy.'en_core_web_sm' is a package in spacy for English language used for performing all this cleaning process.

```
nlp = spacy.load('en_core_web_sm')

def clean_text(text):
    doc = nlp(text)
    tokens = [token.lemma_ for token in doc if not token.is_stop and not token.is_punct]
    return ' '.join(tokens)


sorted_df['Cleaned_News'] = sorted_df['News'].apply(clean_text)
print(sorted_df)
```

This code is used for Data Preprocessing.

3. Feature Extraction:

Sentiment score were extracted using the news column in my dataframe with help of vader and NLP models like FinBERT and Glove, textual data was converted into numerical features using such NLP models.

FinBERT – This is a basic BERT model fine-tuned for financial sentiment analysis.

Word2Vec – This model provides vector using sementic relationships between words.

GloVe – This model provides word embeddings based on word co-occurrence statistics.

Vader – This model provides subjectivity, polarity, positive, negative, neutral and compound by processing cleaned news.

```
finbert = BertModel.from_pretrained('yiyanghkust/finbert-tone')
tokenizer = BertTokenizer.from_pretrained('yiyanghkust/finbert-tone')
def FinBERT_Features(text):
    Inputs = tokenizer(text, return_tensors='pt', max_length=512, truncation=True, padding='max_length')
    Outputs = finbert(**Inputs)
    last_hidden_state = Outputs.last_hidden_state

    feature_vector = last_hidden_state.mean(dim=1).detach().numpy().flatten()
    return feature_vector

merged_data['FinBERT_Features'] = merged_data['Cleaned_News'].apply(FinBERT_Features)
```

This code snippet is an example of feature extraction using FinBERT model which gives a vector using textual data.

4. Model Training:

With the help of final data frame the diff models were trained and at last ensemble was applied to get best model.

```
ensemble_model = VotingClassifier(estimators=[('lda', best_lda),('rf', best_rf),('xgb', best_xgb)], voting='soft')

# Train ensemble model.
ensemble_model.fit(x_train, y_train)

# To evaluate ensemble model.
y_pred = ensemble_model.predict(x_test)
y_prob = ensemble_model.predict_proba(x_test)


accuracy = accuracy_score(y_test, y_pred)
print("Accuracy Score:", accuracy)
print("Classification Report:")
print(classification_report(y_test, y_pred))
```

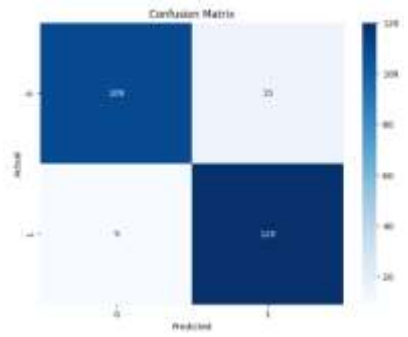
Code shows the ensemble method using voting classifier for the fine - tuned ML models.

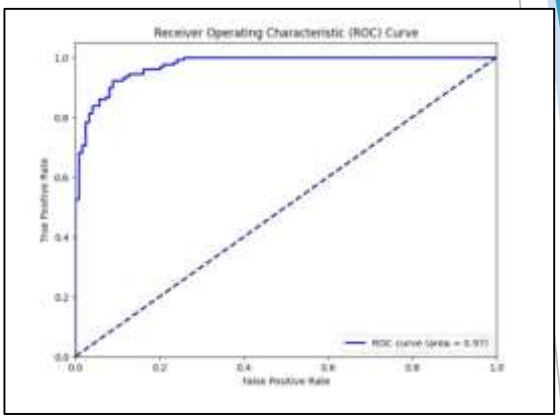
5. Predictions:

Few models were trained like, Randomforest Classifier, Logistic Regression, XGBoost, Linear Discriminant Analysis, SVM, MLP Classifier and after fine tuning them using GridSearchCV OR RandomizedSearchCV and analysing the scores we selected few models (XGBoost, Logistic Regression, Random Forest Classifier) and ensembled them using voting classifier using Hard voting and Soft Voting and best ensemble model was selected to get predicted Label.

- **Scores, Confusion Matrix & ROC Curve:**

Tesla



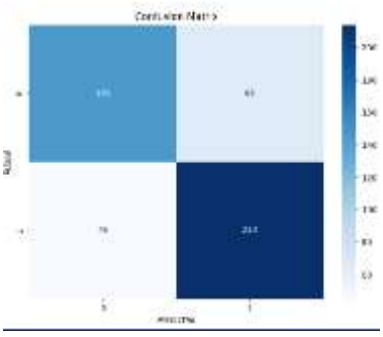


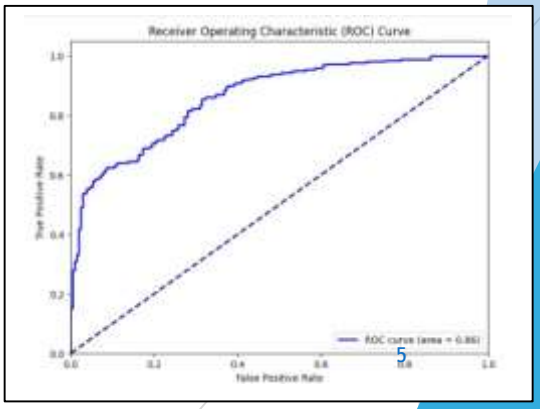

```
Accuracy Score: 0.9051383399209486
Classification Report:
              precision    recall  f1-score   support

           0       0.92      0.88      0.90       124
           1       0.89      0.93      0.91       129

    accuracy                           0.91       253
   macro avg       0.91      0.90      0.90       253
weighted avg       0.91      0.91      0.91       253
```

Apple

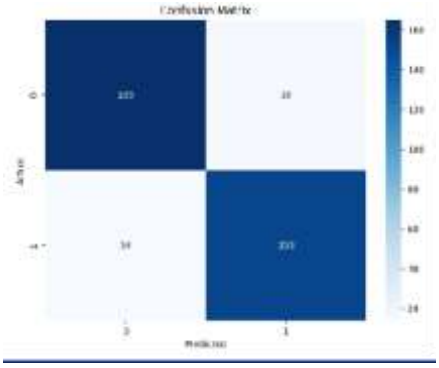


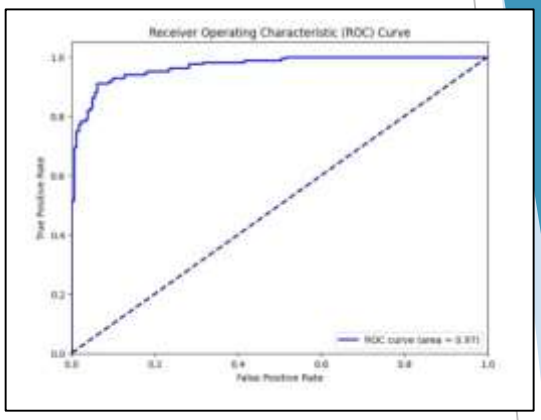


Accuracy Score: 0.7654584221748401
Classification Report:

	precision	recall	f1-score	support
0	0.76	0.69	0.72	210
1	0.77	0.83	0.80	259
accuracy			0.77	469
macro avg	0.77	0.76	0.76	469
weighted avg	0.77	0.77	0.76	469

Amazon





Accuracy Score: 0.9137931034482759
Classification Report:

	precision	recall	f1-score	support
0	0.92	0.91	0.92	181
1	0.91	0.92	0.91	167
accuracy			0.91	348
macro avg	0.91	0.91	0.91	348
weighted avg	0.91	0.91	0.91	348

6. Trading Strategy:

With the help of predicted label and historical stock data a trading strategy has been employed.

- **Predicted Label has been used as a signal.**
- **If the signal is 1 and capital we have is greater than 0 then it buys the maximum stock we can buy at opening price.**
- **If the signal is 0 and stock owned is greater than then 0 it sells all the stocks we have at the opening price.**

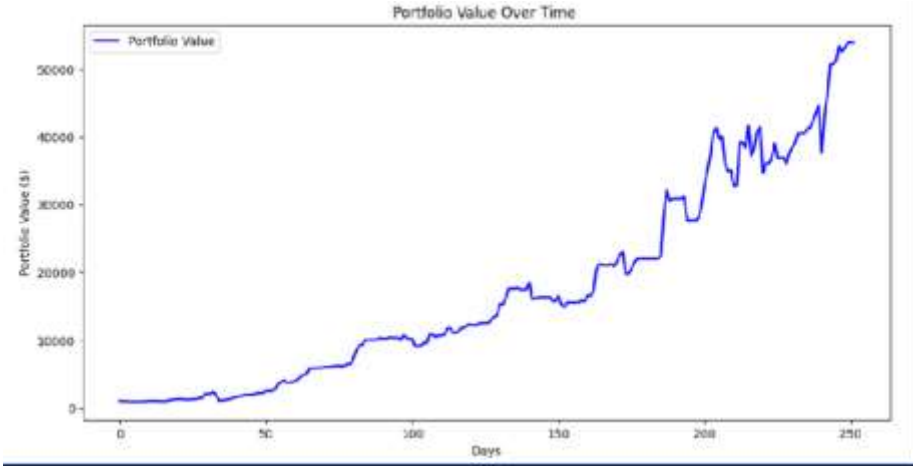
- **In any other case we hold the stocks and cash available.**

Sharpe Ratio: The Sharpe Ratio is a measure used to evaluate the performance of an investment by adjusting for its risk.

Maximum Drawdown: Maximum Drawdown is a measure used to assess the risk of a portfolio or investment strategy by examining the largest drop from peak to trough in the value of a portfolio during a specific period of time.

Win Ratio: The Win Ratio, in the context of trading or investment strategies, refers to the proportion of profitable trades relative to the total number of trades executed.

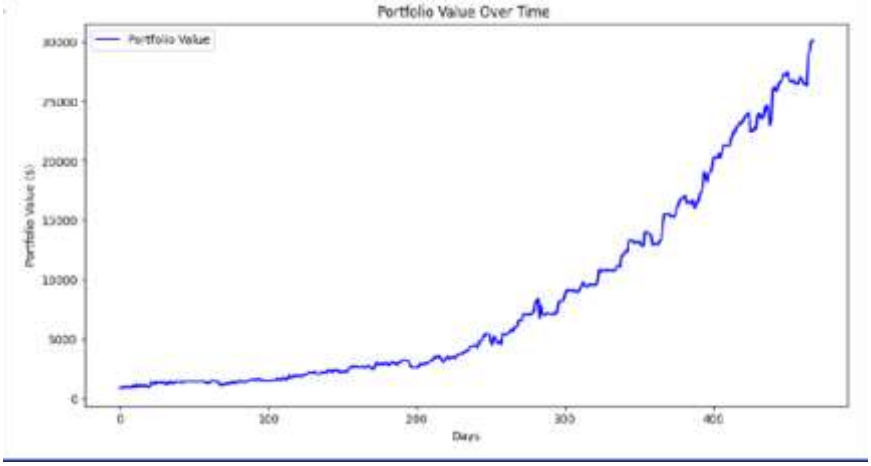
TESLA



	Portfolio_value	Stocks_held	Action
0	996.868061	58.0	Buy
1	914.276018	0.0	Sell
2	948.996077	60.0	Buy
3	884.396009	60.0	Buy
4	868.476076	60.0	Buy

Sharpe Ratio: 4.18
Maximum Drawdown: -55.92%
Number of Trades Executed: 198
Win Ratio: 51.56%
Final Portfolio Value: 53945.07

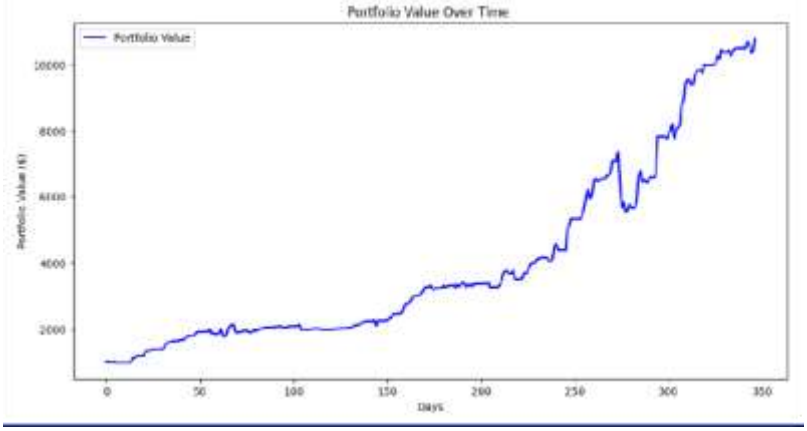
APPLE



	Portfolio_value	Stocks_held	Action
0	875.765515	65.0	Buy
1	1012.712461	0.0	Sell
2	879.854824	65.0	Buy
3	856.329394	65.0	Buy
4	1038.480238	0.0	Sell

Sharpe Ratio: 2.71
Maximum Drawdown: -23.35%
Number of Trades Executed: 372
Win Ratio: 48.94%
Final Portfolio Value: 10869.38

AMAZON



	Portfolio_value	Stocks_held	Action
0	1004.889984	20.0	Buy
1	988.489990	20.0	Buy
2	980.970001	0.0	Sell
3	980.970001	0.0	Hold
4	995.818467	19.0	Buy

Sharpe Ratio: 3.85
Maximum Drawdown: -24.57%
Number of Trades Executed: 255
Win Ratio: 53.49%
Final Portfolio Value: 10780.61