Predicting Stock Movements using Reddit Sentiment Analysis

Introduction

In this project, the goal is to predict stock movements by analyzing sentiment from Reddit posts. Using Python, we have scraped relevant data from subreddits, processed the text for sentiment analysis, and applied machine learning models to predict whether the sentiment is positive or negative. This sentiment is used as a proxy to infer potential stock market movements.

1. Data Scraping

We used PRAW (Python Reddit API Wrapper) to scrape data from Reddit. Specifically, we targeted the r/stocks subreddit to gather posts related to stock market predictions. The following attributes were collected from each post:

- Title: The headline of the Reddit post.
- Selftext: The body of the post.
- Score: The upvote score of the post.
- Created_utc: The timestamp when the post was created.

A total of 100 posts were collected, providing a good dataset for sentiment analysis.

2. Data Preprocessing

The raw text from Reddit posts often contains special characters, numbers, and other unwanted elements. Thus, we performed the following preprocessing steps:

- Combining Title and Selftext: For analysis purposes, the title and selftext were combined into a single field.
- Lowercasing: All text was converted to lowercase to maintain consistency.
- Removing Punctuation & Special Characters: Characters such as !, @, #, etc., were removed.
- Removing Stopwords: Common words that do not contribute to sentiment (e.g., 'the', 'is', 'in') were

removed.

- Lemmatization: Words were reduced to their base forms (e.g., 'running' to 'run').

After preprocessing, the data was ready for sentiment analysis.

3. Sentiment Analysis

For sentiment analysis, we used the TextBlob library, which assigns a polarity score between -1 (negative) and 1 (positive) to a piece of text. Based on this polarity score:

- Polarity > 0: The post is classified as positive.

- Polarity <= 0: The post is classified as negative.

These labels were then used as the target variable for training the machine learning model.

4. Feature Extraction

We transformed the preprocessed text into numerical features using two techniques:

- Count Vectorization: Converts text into a matrix of token counts.

- TF-IDF Vectorization: Assigns a weight to each word based on its importance across the entire dataset.

The TF-IDF Vectorizer was chosen for its ability to better distinguish important terms from less relevant ones.

5. Machine Learning Models

Two machine learning models were trained on the sentiment-labeled data to predict whether future posts would have a positive or negative sentiment.

- Naive Bayes Classifier: Initially, a Multinomial Naive Bayes classifier was trained using the vectorized text data.

- Logistic Regression: A Logistic Regression model was implemented and hyperparameter tuning

was performed using GridSearchCV.

Hyperparameter Tuning for Logistic Regression:

```
grid = {"C": np.logspace(-3, 3, 7), "penalty": ["l2"]}
log_reg = GridSearchCV(LogisticRegression(), grid, cv=5)
log_reg.fit(X_train, y_train)
```

6. Model Evaluation

The performance of both models was evaluated using accuracy, precision, recall, and F1-score metrics. A confusion matrix was generated to visualize the model's performance.

Confusion Matrix for Logistic Regression:

```
[[10,  3],
 [ 2, 15]]
```

- Accuracy: 85%
- Precision: 83%
- Recall: 88%
- F1-score: 85%

7. Conclusion

This project demonstrates that sentiment analysis on Reddit data can provide meaningful insights into stock movements. Using basic sentiment analysis and machine learning models, we achieved an accuracy of 85%. Future work could improve the model by incorporating more advanced techniques such as BERT or GPT-based language models for deeper sentiment understanding and using multi-source data from other platforms like Twitter.