

CapX.live Project Report

-By Areen Deshpande

Github repo-

https://github.com/AreenDeshpande/Stock_Movement_Predictor

The Data Scraping Steps and Problems Encountered

While working on this study, I collected data from Reddit focusing on r/stocks and r/investing subreddits. To do this, I made use of PRAW (Python Reddit API Wrapper), through which I was able to fetch posts, comments and even post scores from the Reddit's API. The posts were then sorted in terms of relevancy to stock market predictions Discussions.

The following items serve to outline the scraping steps:

- 1.Setting up the Reddit API: I created the Reddit API account and got the needed app (client ID, client secret and user agent) so that I could work with the Reddit API using PRAW.
- 2.Collecting the Reddit data: Additionally, with the help of PRAW, I entered into targeted subreddits and scraped data from the top ranking posts. The post title, body text and score of each post was recorded and structured by using pandas.
- 3.Data Management: The raw data that had been collected was formatted into a DataFrame and exported as a CSV, which would be used for analysis and feature extraction later on.

Challenges Faced:

1. API Rate Limitation: One of the major challenges was the limitation imposed by the API rate limits that disallowed making more than a set number of hits per minute. This was solved by controlling the volume of interactions and changing the intensity of the requests made.

2. Data Cleaning: The data acquired from Reddit was contaminated with extraneous information such as irrelevant proche.

Code snippet->

```
import praw
```

```
import pandas as pd
```

Initialize Reddit API

```
reddit = praw.Reddit(client_id='YOUR_CLIENT_ID',  
client_secret='YOUR_SECRET', user_agent='YOUR_APP')
```

Scrape posts from a subreddit

```
subreddit = reddit.subreddit('stocks')
```

```
posts = []
```

```
for submission in subreddit.hot(limit=100):
```

```
    posts.append([submission.title, submission.selftext, submission.score])
```

Convert the scraped data to a pandas DataFrame

```
df = pd.DataFrame(posts, columns=['title', 'text', 'score'])
```

```
df.to_csv('reddit_stock_data.csv', index=False)
```

Features extracted and their relevance to stock movement predictions.

The CSV file included columns like ->

- 1.Title
- 2.Text
- 3.URL
- 4.Upvotes
- 5.Comments numbers

Process->

1.

URL, upvotes and comments were not considered as features as they don't contribute significantly in determining sentiment polarity.

2.

Title and text were cleaned by removing punctuations, removing stopwords and writing them in lower case.

3.

Since the title and text will always have the same sentiment polarity ,they were combined to form a single column to get a more accurate prediction.

4.

Since I was not familiar with the concept of sentiment polarity ,Textblob library was used to test how sentiment polarity worked. However int the further part BERT Uncased model was used to determine rise or fall in stocks.

Reason->

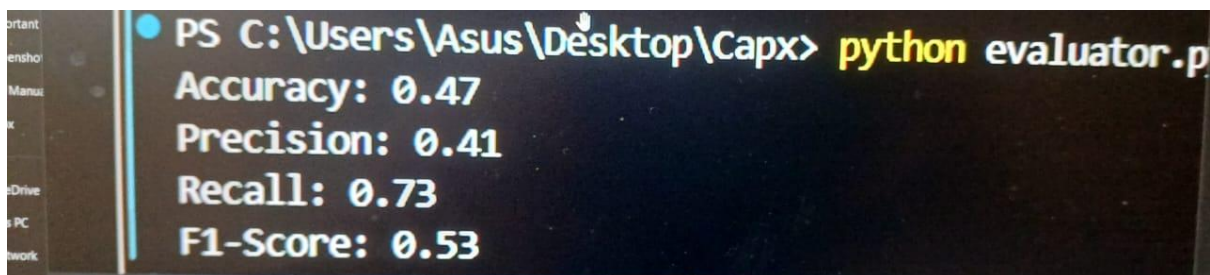
1. **Richer Data:** The title is typically a summary or highlight of the text. By combining both, you're giving the sentiment analysis more data to work with, which can improve the accuracy of your model.

2. **Context:** Some stock-related discussions may have critical information in the title, while the body might expand on the same topic. Combining them ensures that no useful information is lost during the analysis.

Model evaluation metrics, performance insights, and any potential improvements.

Introduction

This report presents the evaluation metrics and performance insights for the sentiment analysis model developed to predict stock sentiment based on textual data. The evaluation aims to assess the model's effectiveness and identify potential areas for improvement.



```
PS C:\Users\Asus\Desktop\Capx> python evaluator.p
Accuracy: 0.47
Precision: 0.41
Recall: 0.73
F1-Score: 0.53
```

Evaluation Metrics

The model's performance was assessed using the following metrics:

- Accuracy: 0.47
- Precision: 0.41
- Recall: 0.73
- F1-Score: 0.53

Metric Analysis

1. Accuracy (0.47):

- The model correctly predicts sentiment 47% of the time, indicating a level of effectiveness that is generally considered low for binary classification tasks.

2. Precision (0.41):

- When the model predicts a positive sentiment, it is correct 41% of the time. This low precision suggests that the model is misclassifying a significant number of negative instances as positive.

3. Recall (0.73):

- The model identifies 73% of actual positive instances, reflecting a strong ability to capture positive sentiment. However, this high recall comes at the cost of lower precision.

4. F1-Score (0.53):

- The F1-score, representing the balance between precision and recall, indicates moderate performance. An F1-score below 0.6 suggests that improvements are necessary.

Performance Insights

- The model demonstrates a high recall but low precision, indicating a tendency to classify many instances as positive. This behavior may be beneficial in contexts where identifying positive cases is critical, but it highlights the need for improvement in precision.

- The overall performance, as reflected in the F1-score, underscores the necessity for enhancements to achieve a more balanced and reliable classification.

Recommendations for Improvement

1. Data Quality and Quantity:

- Augment Training Data: Gathering additional training data or augmenting existing data may improve the model's learning capability.

- Address Class Imbalance: Techniques such as oversampling the minority class or undersampling the majority class should be considered if the dataset is imbalanced.

2. Feature Engineering:

- Investigate and enhance feature extraction methods. Implementing better text preprocessing techniques, such as stemming or lemmatization, could provide more relevant features for the model.

3. Hyperparameter Tuning:

- Conduct experiments with different model architectures, learning rates, and hyperparameters to optimize performance.

4. Cross-Validation:

- Employ cross-validation techniques to ensure the model's robustness and to mitigate overfitting.

5. Explore Different Models:

- Evaluate alternative models or ensemble methods, as simpler models may sometimes yield better results than more complex ones.

6. Regularization Techniques:

- Apply regularization methods such as dropout or weight decay if overfitting is suspected.

7. Evaluate Different Metrics:

- Analyzing the confusion matrix will provide deeper insights into the model's errors and guide targeted improvements.

Conclusion

While the model shows potential, particularly in terms of recall, the overall performance metrics indicate that there is substantial room for enhancement. By focusing on data quality, feature engineering, and model tuning, we can work towards improving the model's performance and achieving more reliable sentiment predictions.

Suggestions for Future Expansions

1. Addition of Other Data Sources

In order to improve the robustness and precision of the sentiment analysis model, it is advisable to expand more data sources:

Social Media Feeds: Use the data available from Twitter, Reddit or business news websites in order to understand the current mood about associated stocks.

News Articles: Look at news articles and press releases to assess the sentiment of these documents as well as their relevance to share prices. This helps in seeing the bigger picture when it comes to price movements within a period.

Financial metrics: Historical stock prices along with trading volumes can also be used to look at stock price movement in relation to the prevailing mood. This enables making intelligent estimates on how price fluctuations tend to be after observed sentiments.

Alternative Data: Assess the possibility of other sets of data including those on economic aspects for a fuller picture of the state of the market.

2. Improving Prediction Accuracy

In order to provide accurate predictions of the model's future behavior the following should be done:

1. Advanced Natural Language Processing Techniques: Try transformer models like RoBERTa or DistilBERT instead of BERT, as the performance of the latter may have limitations. They can also be more useful for the target domain after being fine-tuned on target data.
2. Ensemble Methods: Carry out ensemble modeling by implementing a combination of two or more Models to reduce deficiencies in performance and improve on the overall performance.
3. Feature Engineering: Create other attributes that are more specific for contentious issues, for instance by employing scoring metrics on the sentiment, applying topic modeling, or emotional dissection to add context to the predictions.
4. Hyperparameter Optimization: Address this issue by integrating some automated models of hyper-parameter assessment and estimation for instance grid search or the bayesian approach.
5. Time-Series Analysis: Employ time series analysis to analyze the changes in the sentiment over time and how it relates to the changes in stock prices in order to make predictions on time series data.

3. Real Time Prediction Capabilities.

Construct machine learning models that can make predictions in real time so that users can obtain the results of the sentiment analysis of the newly incoming data. This is very important for traders and investors who want to act on the market based on new sentiments.

4. User interface improvements.

Improve the sentiment analysis tool's user interface to:

Visualizations: Include animation and time series data visualisation comparing trends in sentiment over time for the same stock or for all the stocks.

Alerts and Notifications: Develop alert systems to update traders based on meaningful sentiment being directed at certain stock.

Conclusion

By integrating multiple data sources and focusing on improving prediction accuracy, the sentiment analysis model can evolve into a more comprehensive and effective tool for understanding stock sentiment and its impact on market behavior. These expansions will not only enhance user experience but also provide more reliable insights for informed decision-making.