Stock Tweets Sentiment Analysis

**Overview**

Nowadays, numerous people started to talk about stocks and investments openly on social media. Last year we’ve seen Gamestop, AMC and Tesla stocks skyrocketing, and it is trending on twitter for several weeks. A lot of people are affected by what social media react to these stocks, which makes understanding social media content a very intuitive method in learning the stock market. For companies and industries, understanding social media will help them to predict the actions of future investors, and even assist in building models that will forecast future stock price. NLP will help with identify the relationship between text data from social media and the sentiment behind it. By building a model that can detect social media reaction towards stock market, we will be able to understand how social media affects stock market and investors in the future.

**Data Exploration**

The dataset contains two column, one text column along with a sentiment label which is available at: <https://www.kaggle.com/code/sandeepkumarkushwaha/sentimentanalysis-of-stock-market-spp500>

here are some graphs from exploratory analysis:

Chart, bar chart

Description automatically generated1.Frequency of sentiment (1 is positive/-1 is negative)

2.Word Cloud graphs

Text

Description automatically generated

**NLP Method**

I used bag of word approach for text processing. Here are the packages and methods implemented:

NLTK - The Natural Language Toolkit offers text processing libraries for tokenization, parsing, classification, stemming, tagging and semantic reasoning.

re.sub('[^a-zA-Z]', " ", x[i]) - This code is implemented to filter out only English letters, replace other characters with space.

df.lower() / df.split() - This code is used to split the sentence into word array after lowering all letters.

stopwords.words('english') - I used this package to remove stop words in every sentence.

WordNetLemmatizer() - Instead of stemming, lemmatization is more reasonable when the content is important for sentiment analysis, I used this code to lemmatize each word after removing stop words.

CountVectorizer() - Text data are transformed into vectors to fit in predictive models later

Comparison Table between raw and processed data:

|  |  |
| --- | --- |
| Raw Data | Processed Data |
| Kickers on my watchlist XIDE TIT SOQ PNK CPW BPZ AJ trade method 1 or method 2, see prev posts | kicker watchlist xide tit soq pnk cpw bpz aj trade method method see prev post |
| user: AAP MOVIE. 55% return for the FEA/GEED indicator just 15 trades for the year. AWESOME. | user aap movie return fea geed indicator trade year awesome |

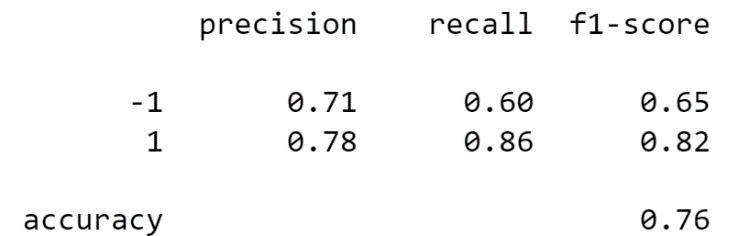
**Model Evaluation**

*I. Decision Tree*

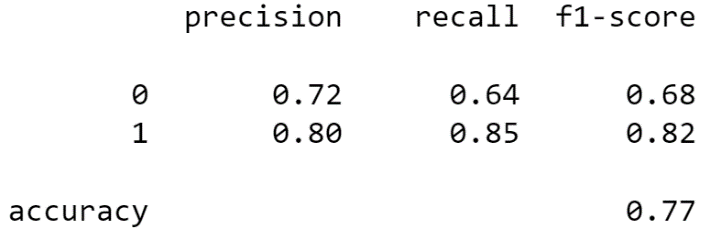
Table

Description automatically generated

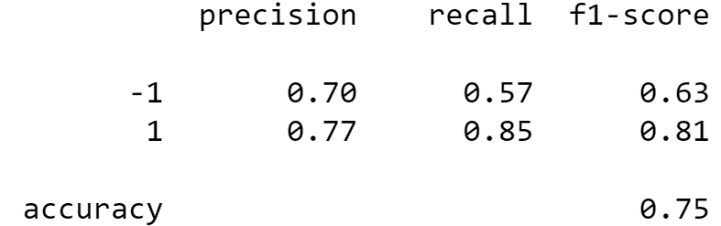
*II. Random Forest*



*III. XGBoost*



*IV. LGBM*

****

**Results and Findings**

Based on previous model performance part, XGBoost classifier gives the best performance. The dataset is clean but small, text processing part serves as the most important step in making this project.

**Conclusion and Future Steps**

The raw data is very hard to interpret since 80% of tweets about stocks are numbers and company stock symbols, however, after text processing a lot of tree-ensembled models can give an accuracy of around 80% which is surprising to me. In the future studies, I would like to implement some deep learning methods such RNN model on it. With more sufficient stock tweets data, this study can be further improved, and I can get a better understanding on how powerful NLP is in discovering the values behind texts from our daily social media platforms.

**References**

<https://www.analyticsvidhya.com/blog/2015/06/tuning-random-forest-model/>

<https://www.kaggle.com/code/diveki/classification-with-nlp-xgboost-and-pipelines/notebook>

<https://towardsdatascience.com/xgboost-fine-tune-and-optimize-your-model-23d996fab663>