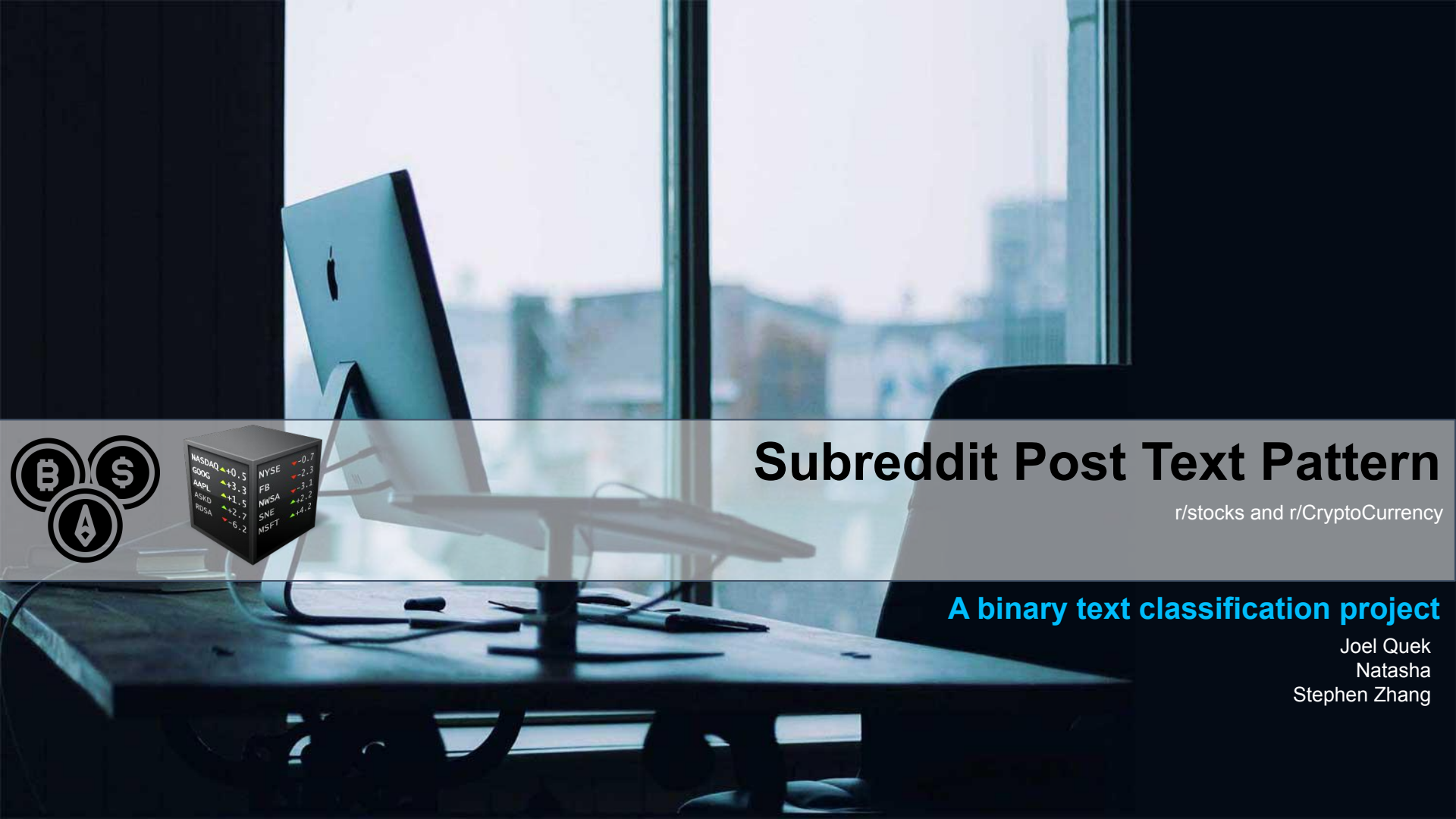


Subreddit Post Text Pattern

r/stocks and r/CryptoCurrency

A binary text classification project

Joel Quek
Natasha
Stephen Zhang

1 Background

As a part of marketing division of an investment club, we would like to create a list of suggested post/news based on comments left by user in the discussion thread. This enables us to create a curated content that suits our member's interest

Data of Interest: People tend to discuss and share their analysis before making any trade.
Reddit is a platform to have an open discussion on many topics

- A broad topic can be categorized to a subreddit where people can share their thoughts, ask questions or opinions from others and user anonymity is guaranteed in this platform

2 Subreddits

From the beginning of COVID, people shows significant interest on how to grow their money. There are many investment instruments to do so such as stock and crypto currency.

There are many differences among them, but while discussing these topics, the words used is quite similar.

3 Problem Statement

Creating a model to do binary text classification (stocks/ CryptoCurrency) based on the post where the model then can be implemented to classify member's discussion thread.

The **goal of our model** is to get a good degree of separation between the two classes which is represented by Receiver Operating Characteristics Area Under the Curve **(ROC-AUC)**.

- The higher it is, the better the model at predicting the binary class.

Introduction
Analysis on text pattern on r/stocks and r/CryptoCurrency

Subreddit Post First Outlook
What similarities or differences we can observe from EDA?

Pre-processing and Exploration
Vectorizer analysis and diving on the modeling

Summary
What is wrong and how can it go wrong?

Introduction

r/stocks and r/CryptoCurrency

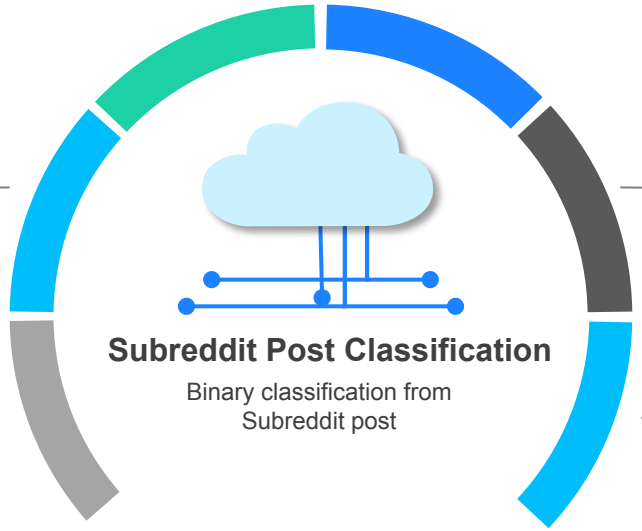
Methodology

Exploratory Data Analysis
What pattern that we can observe?

Natural Language Processing Application
Hyperparameter tuning for potential model

Data Pre-Processing
How do we create effective features?

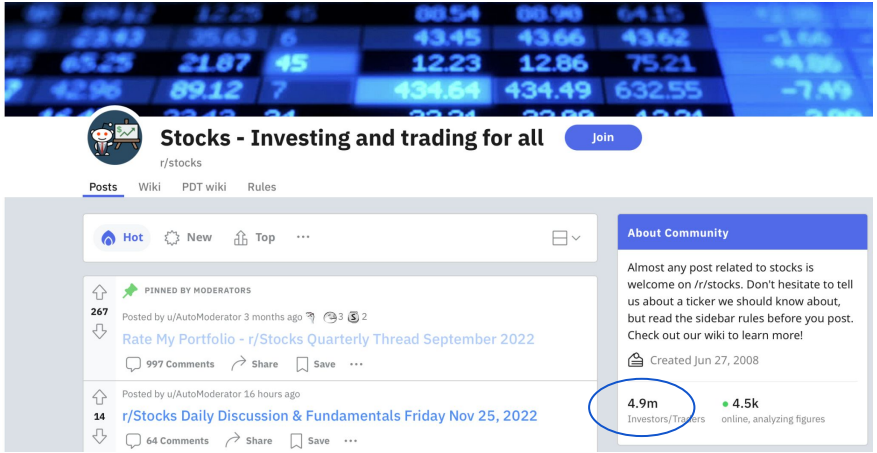
Final Model Adjustment
Optimal threshold selection for final model

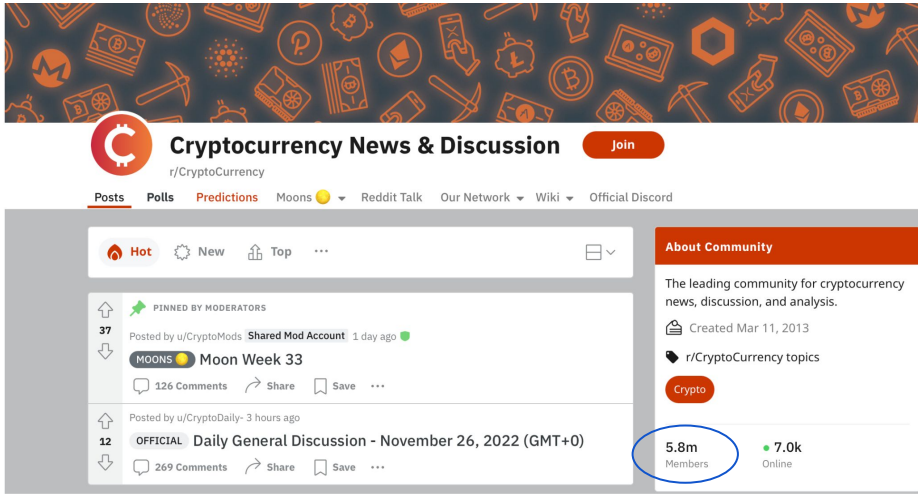


Scraping
Using API push shift to pull title and self text from Reddit post

Error Analysis
Why did our model predict the wrong class? Is there any limitations or potential improvement for final model?

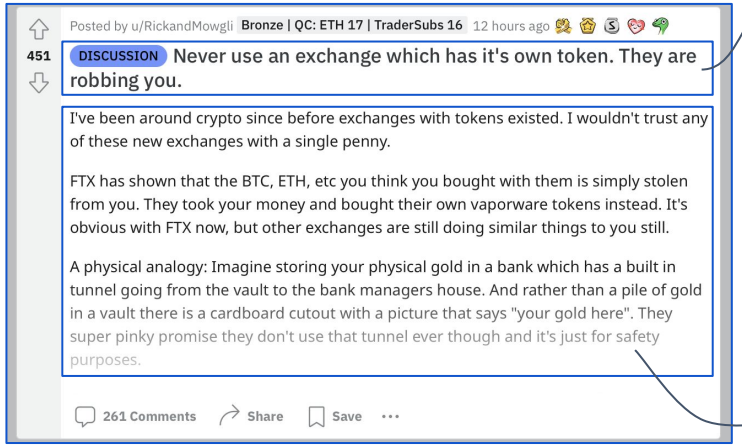
Webscraping





Huge users in both subreddits

Web-scraping

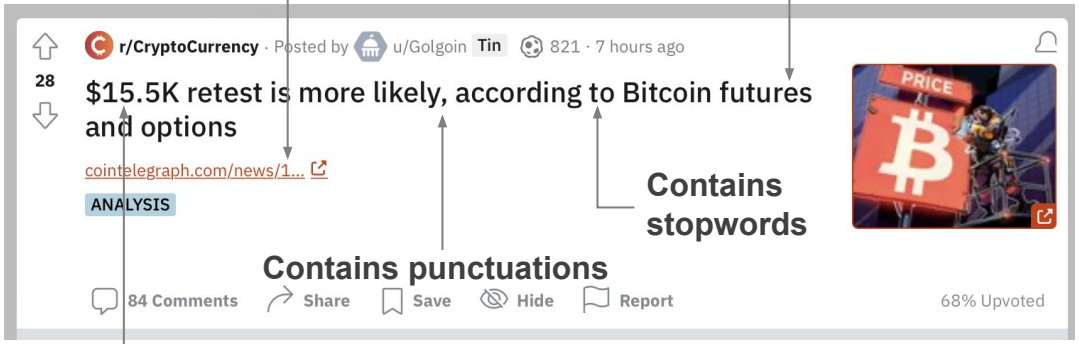


> Scraping through title and self text

Pre-processing

Contains link

Lemmatize: “futures” becomes “future”



Contains stopwords

Contains punctuations

Contains number

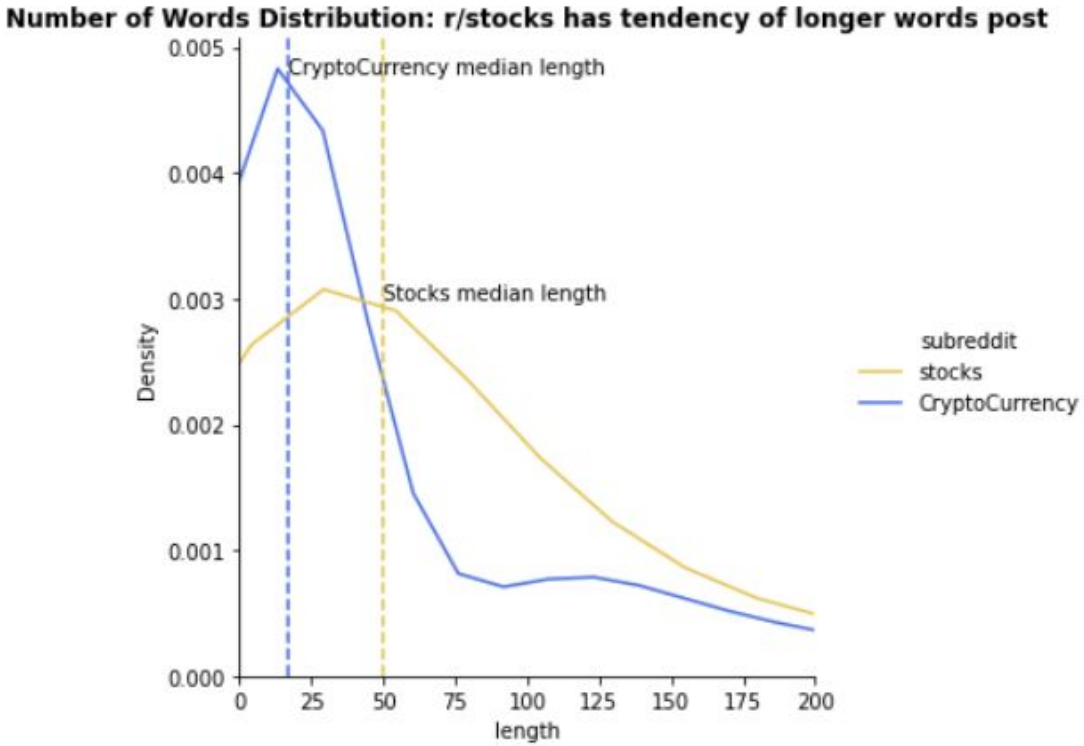
Post that contains empty selftext post scraping: removed!

```
0                                                    NaN
1    [Link to the full article (4 min read)](https:...
2                                                    NaN
3                                                    NaN
4                                                    NaN
Name: selftext, dtype: object
```

Subreddit First Outlook (EDA)

r/stocks and r/CryptoCurrency

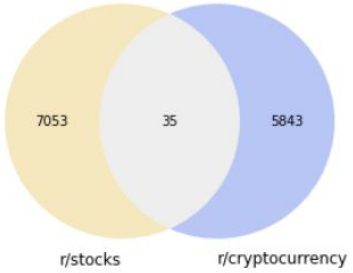
EDA: Post Length and Overlapping User



Balanced Data

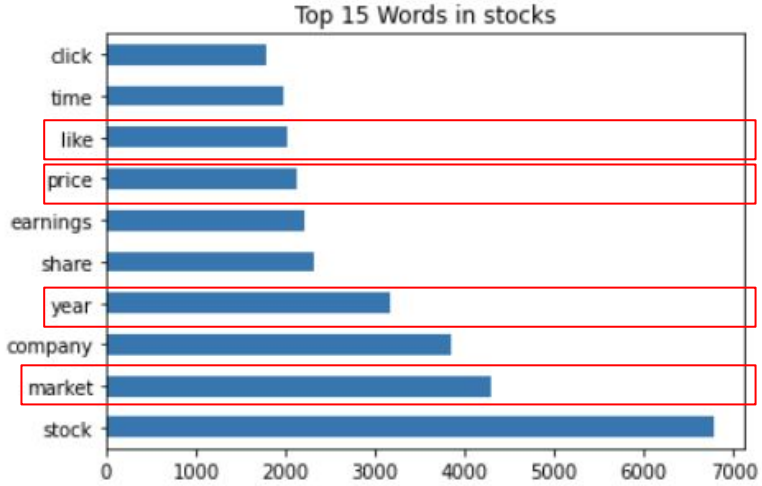
stocks	0.54666
CryptoCurrency	0.45334

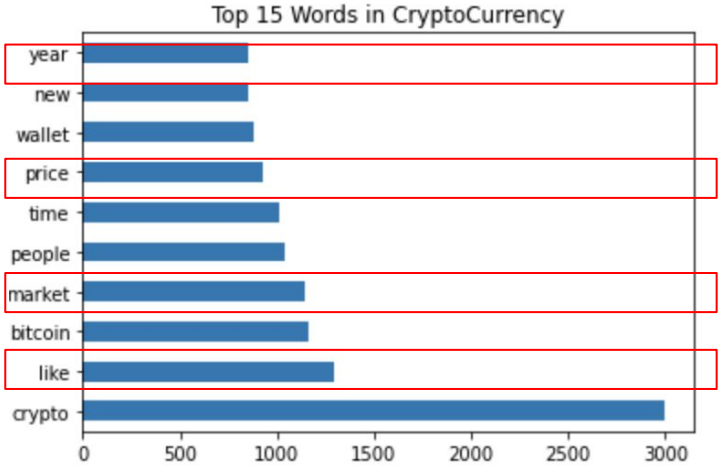
There are 35 common authors between stocks and CryptoCurrency subreddits.



EDA: Overlapping Token

“stock” and “crypto” are the most distinct feature, we will remove them for our model training





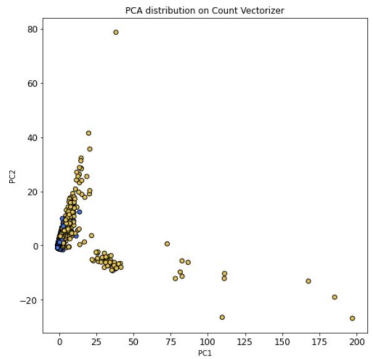
“Most of the people and analyst saying ***market*** is gonna crash next year, SP500 below 3400,huge recession incoming, terrible earnings etc.
Yes we may get a bit higher unemploymet(3.8-4.2%), ***earnings*** will maybe have a miss but nothing too devastating, economy still strong throughout 2023.
How things are going, we might achieve soft landing.
Your opinion?”

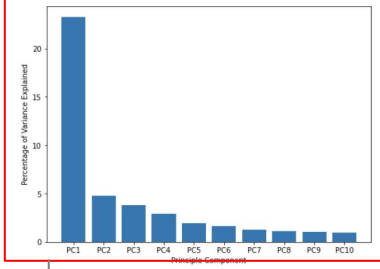
“For those unfamiliar, CZ (will admit like 90% of you I can't say his real name right) is the leader of the largest ***crypto*** exchange and pretty much an indirect reason why the ***market*** has tumbled further. He sped up the process of outing Sam and helped collapse FTX which we all know the rest…
Turns out it was BS. Coinbase has pretty much revealed they may as well rename themselves BTC incorporated as they absolutely dwarf Binance in Btc reserves.
Moral of the story, there are lots of people trying to create FUD and looking to further shake out retailers. CZ is now a living meme”

Principal Component Analysis

CountVectorizer

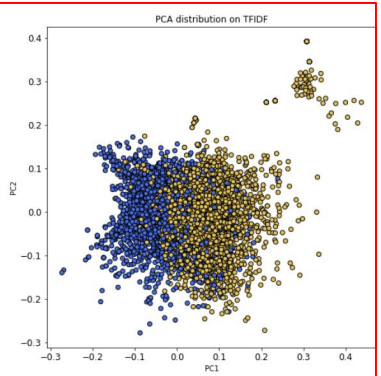
Convert post to bag of words based on token frequency in the document

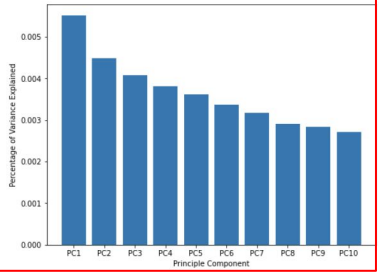




Term Frequency-Inverse Document Frequency

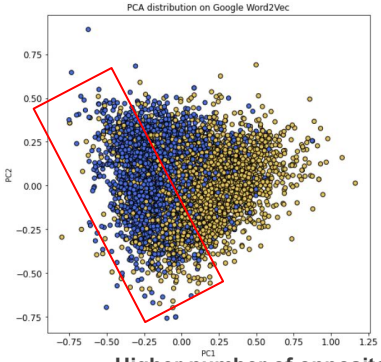
Provides the importance of the words in the corpus

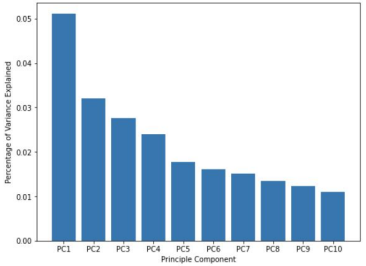




2013 Google Word2Vec

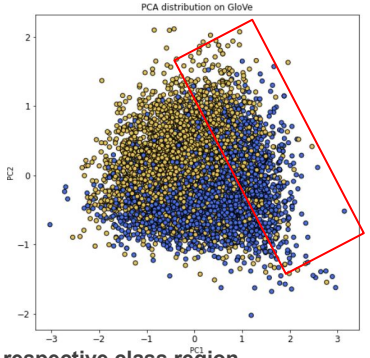
Pre-trained embedded vectors on 100 billion words from Google news dataset.

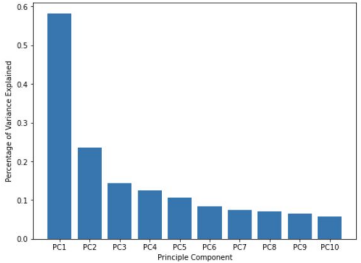




2014 Twitter Global Vectorizer (GloVe)

Pre-trained embedded vectors on 2 billion twitter tweets by Stanford University Computer Science Department





Higher number of opposite class in respective class region

Post related to stocks and crypto are very dynamic, the terminology might not have been captured in both Google Word2Vec and Global Vectors.

Expecting better degree of separation compare to word embedding vectors

NLP Modeling

r/stocks and r/CryptoCurrency

Model Selection

0: r/stocks
1: r/CryptoCurrency

Pre-processed Post

Vectorizer

CountVectorizer **TFIDF** **Google Word2Vec** **GloVe**

Supervised Machine Learning Model (Classifier) based on ROC

ROC AUC Score	CountVectorizer	TFIDF	Word2Vec	GloVe
Logistic Regression	91.24	90.78	86.59	86.29
KNN	89.19	89.18	82.63	84.17
Naive Bayes	**91.47**	90.76	74.04	65.35
Random Forest	88.96	87.64	85.64	85.83

- Processing time: Naive Bayes is significantly faster than random forest
- Naïve Bayes can outperform other algorithms if the feature variables are independent.
- Both model are not interpretable, but we can try using **Lime Interpreter!**

Final Model

0: r/stocks
1: r/CryptoCurrency

Multinomial Naive Bayes with Count Vectorizer

Classifier threshold at 0.5: which means <0.5 belongs to stock and above is crypto

Training Accuracy Score

99.77%

Testing Accuracy Score

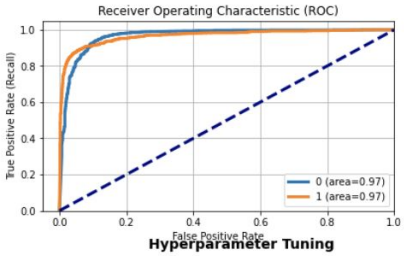
96.65%

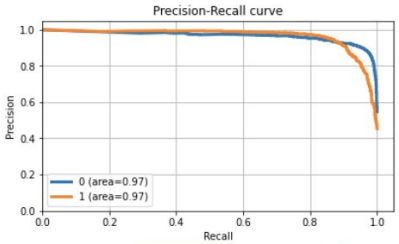
ROC Score: 91.47%

Recall Score: 88.1%

Precision Score: 93.42%

F1 Score: 90.68%



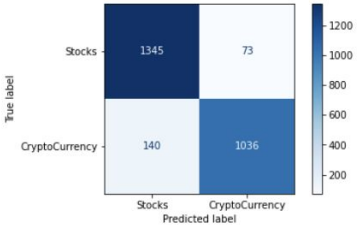


Hyperparameter Tuning

ngram_range metrics for cvec : (1, 2)

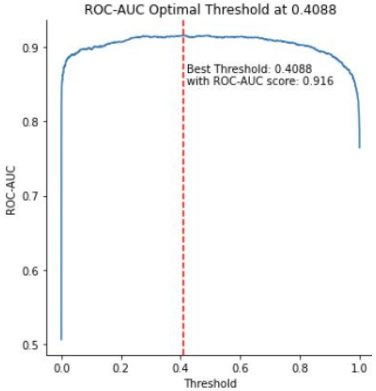
max_features metrics for cvec : None

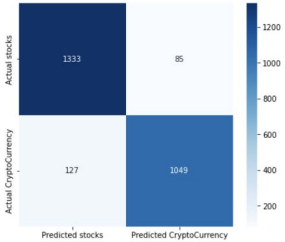
max_df metrics for cvec : 0.85



Optimal Threshold Selection = 0.4088

There is **precision-recall tradeoff** observed as the False Negative decreases while the number of False Positive increases.
Comparing with the before and after threshold adjustment, **precision 93.42% to 92.5% and recall from 88.1% to 89.2%.**





Recall Score: 89.2%

Precision Score: 92.5%

F1 Score: 90.82%

Error Analysis

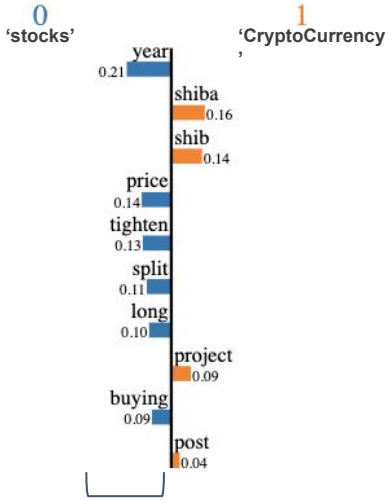
r/stocks and r/CryptoCurrency

Misclassification: False Negative

0: r/stocks
1: r/CryptoCurrency

Predicting ‘stocks’ subreddit while the true class is ‘CryptoCurrency’

```
True: 1 --> Pred: 0 | Prob: 0.0002
```



Text with highlighted words

market tighten day scared come self split situation shib started buying past year buying way managed average nowi bought month price good know belive project personally long admit im starting lose mentallyeveryone month wont regret year recently saw post year shiba mindset keeping long portfoliowhat shoesbest regard silly investor currently

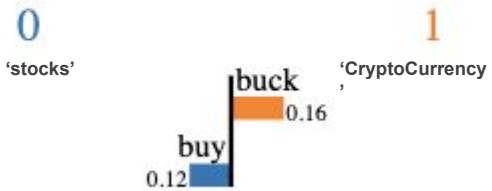
shiba is actually a good indicator for CryptoCurrency as it is one of the well-known crypto but as we can see t**here are more features/token corresponding to stocks subreddit instead of cryptocurrency**, therefore, model wrongly classified it as stocks instead of CryptoCurrency

Misclassification: False Positive

0: r/stocks
1: r/CryptoCurrency

Predicting ‘CryptoCurrency’ subreddit while the true class is ‘stocks’

```
True: 0 --> Pred: 1 | Prob: 0.5233
```



Text with highlighted words

buck buy buy

It is hard to even differentiate for us to classify this post from this post!

Conclusion

1. **Pre-trained embedded vectors perform worst across all models**. This is expected from the first principal component analysis
2. Across all models, **ngram_range = (1,2) is found to be beneficial** to increase our accuracy score
3. There is **precision-recall tradeoff** observed when we tune our model
4. **If a post contains a lot of vocabulary from the opposite class, our model is unable to predict correctly** (False Negative example).
5. Last but not least, looking at the learning curve plot, we observe our Cross-Validation score has not reached plateau yet, which means we can try to **pull more training examples to improve our model score**.