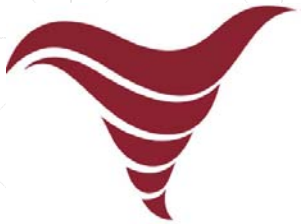


Web APIs and NLP Classification

Rufus Ayeni

Problem Statement

Use the Pushshift API to collect text data (posts) from two subreddits that focus on financial investing and trading. After collecting the data, do the following:

- Use NLP techniques to train various models to analyze a subset of the data.
- Test how well the various models classify posts.

The following classes correspond to the two subreddits from which the data was collected:

- Class 0: Subreddit r/pennystocks
- Class 1: Subreddit r/CryptoCurrency

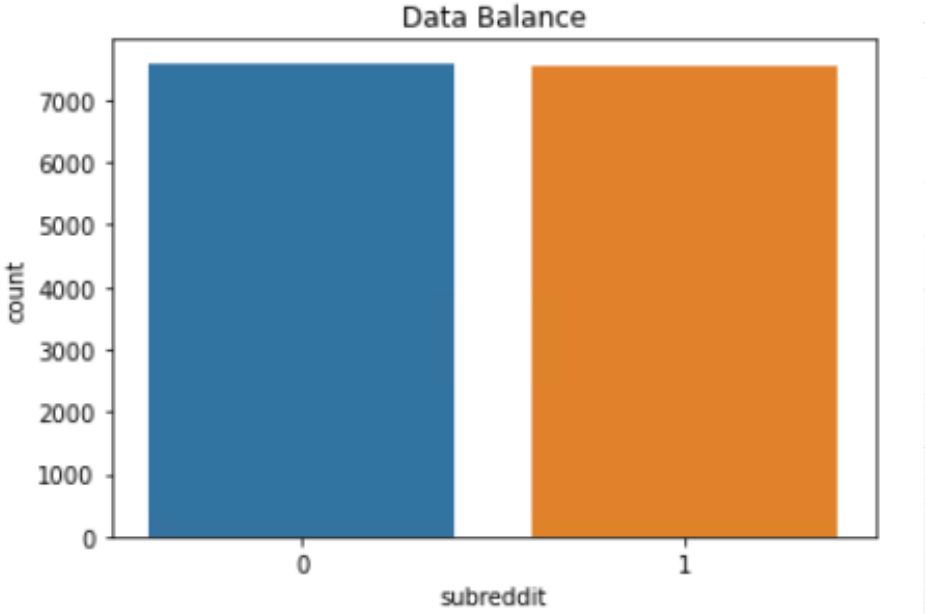
Data Cleaning

Cleaning the data wasn't as challenging as previous cleanings.

- **Time:** 75% of our overall work was dedicated to cleaning data.
- **[removed], [deleted], special chars, links:** Most of the cleaning was dedicated to removing the [removed] and [deleted] tags.
- **title -> selftext :** Instead of dropping rows that had the [removed] and [deleted] tags, we decided to replace the tags with values from the `title` column.

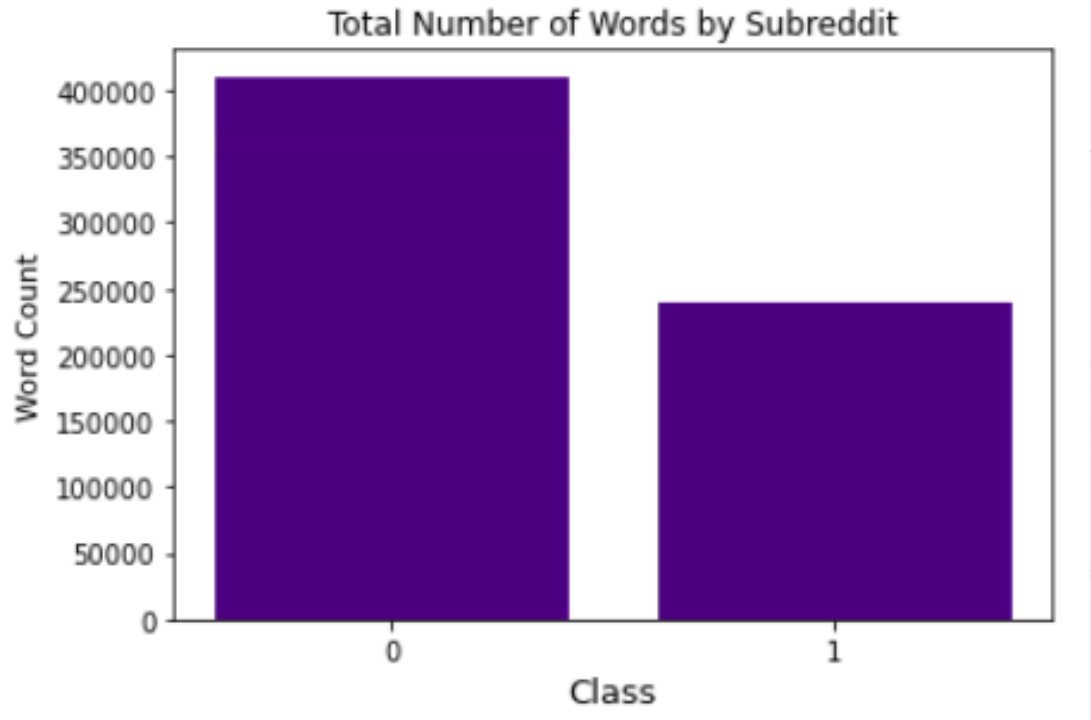
EDA

- **Balance**: Our dataset is fairly balanced with **50%** of observations are of class 0 and **49%** are of class 1. The number of observations is 15,199.



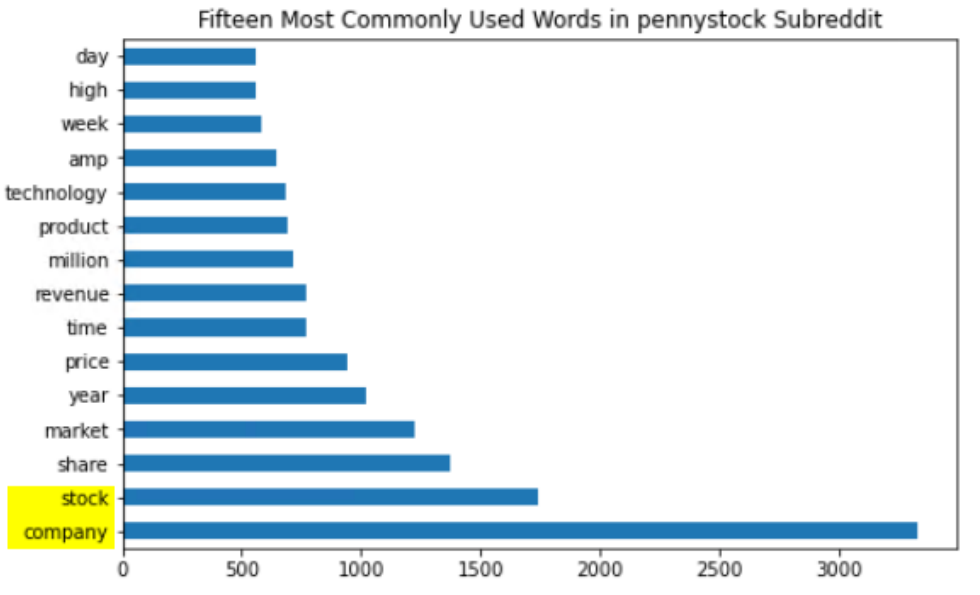
EDA

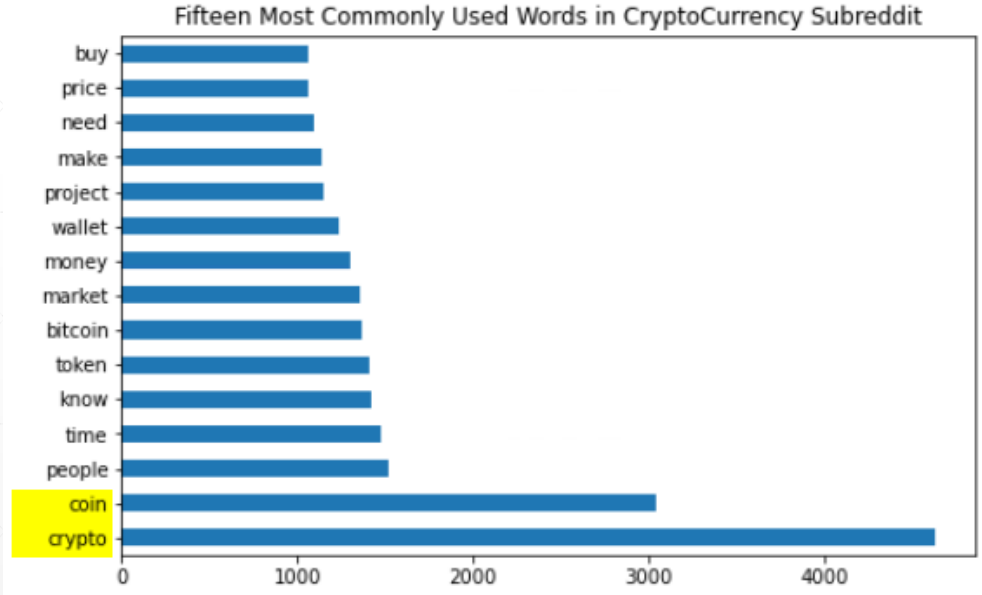
- **Total Words**: Class 0: 410,588, Class 1: 240,309



EDA (Commonly Used Words)

- Class 0: `company` & `stock`
- Class 1: `coin` & `crypto`
- Note: `'ha' and 'wa': second pass needed to remove stop-like words`





Model Evaluation (Overfitting)

Model	Trng Score	Test Score	Variance
Random Forest	0.8523	0.8457	0.0066
Naïve Bays	0.8752	0.8637	0.0115
AdaBoost	0.8789	0.8633	0.0156
Simple Vector Classifier	0.9809	0.8965	0.0844

Model Evaluation (Specificity & Sensitivity)

Model	Specificity	Sensitivity
Random Forest	0.7433	0.9488
Naïve Bays	0.8035	0.9243
AdaBoost	0.9548	0.7496
Simple Vector Classifier	0.9085	0.8845

Closing Comments.

- All the models some overfitting. In terms of fitting, the **Random Forest (RF)** model performed the best, because it had the least amount of overfitting, and the highest amount of bias. In addition to best generalizing new data, the RF model had the highest sensitivity score (true positive rate). However, it also had the lower specificity score (true negative rate).
- The models could have performed better if more attention were given to eliminating stop-like words (e.g., 'ha', and 'wa'). In the initial observation, the aforementioned stop-like words ranked in the 15 most commonly used words. Those two stop-like words were removed. However, I am sure others remained.
- An increase in the number of observations improved the performance of all models. Initially, we collected 4,000 observations. Then, we increased the number to 15,000.