Project 3: Using APIs and NLP for Prediction

subreddits: r/CryptoCurrency and r/StockMarket

By:
Sileshi Hirpa

Dec. 17/21

content:

1. Background
2. Problem Statement
3. Methodology
4. Conclusions and Recommendation

Project Goal:

Classification of comments from the two subreddits

Background:

Photo : Google

According to its website,

- **Reddit** is a network of communities (with 430 million+ monthly active users) where people can dive into their interests, hobbies and passions.
- **subreddits** are subsidiary threads or categories within the **Reddit** website(source).

- My two subreddits (with 6M+ members) are r/CryptoCurrency/ and /StockMarket/.
- CryptoCurrency will be assumed as my positive target and stock will be negative
- The optimization parameter for my model is going to be accuracy.

Problem statement:

With stock and crypto investors in mind, I am using Reddit's API for webscraping posts from two subreddits, r/CryptoCurrency and r/StockMarket, and use NLP to train a classifier on which subreddit a given post came from. The model will predict to which subreddits class a text belongs to.

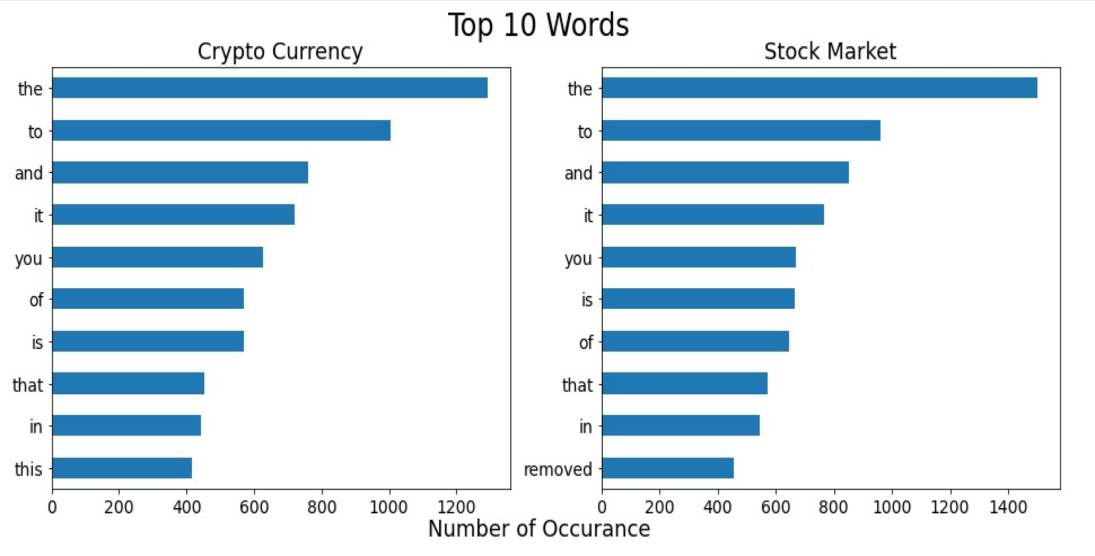
Methodology:

- **Data collection**: Data Scraping using Reddit API through **pushshift.io** and collected more than 4000 posts (2000 Crypto, 2063 Stock posts)
    - Takes long time downloading,
- **Data Cleaning and EDA**:
    - cleaning html tags, emojis, etc. needs more time
    - dropping duplicates
- **Preprocessing and Modeling**:
    - lemmatizing and customizing stop words by adding “lol”, “wa”, “ha”, “don”, etc. to stopwords
    - EDA for most common (top 10) words from both subreddits
    - train/test split (default size, stratify)
    - **Models used/tested**: Random Forest, Logistic Regression, Support Vector Machine, and Multinomial Naive Bayes
- **Modeling tools used**: Pipelines, and GridSearch
- **Evaluation methods**: accuracy score, precision from classification report, confusion matrix to see False Positives and False Negative, ROC curve to visualize model performance.

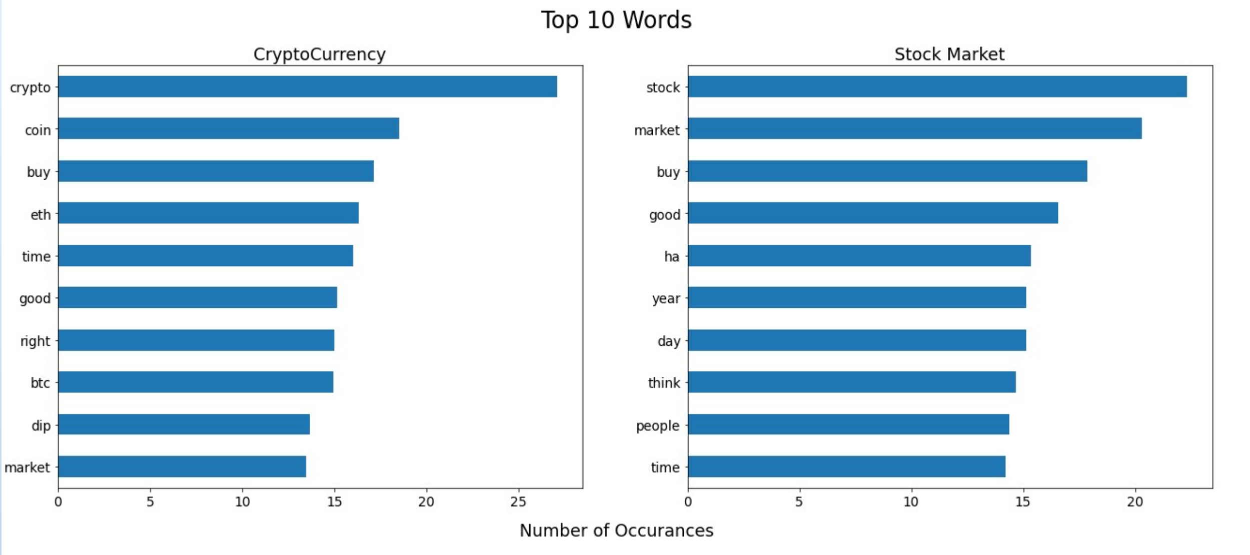
- the two subreddits converted to DataFrames and then merged
- the merged DataFrame has ‘body’ and ‘target’ columns; where,
    - ▸ the ‘body’ column is the text message for each post,
    - ▸ the ‘target’ column categorizes each text to its subreddit.
- binarized my ‘target’ column as the CryptoCurrency is my positive target for the classification.
- checked for my baseline accuracy before any data cleaning:

Target	Baseline accuracy	Interpretation
1	0.500125	Approximately by 50% subreddit posts are crypto
0	0.499875	

- Visualized top 10 most occurred words:



All stop words



After customizing my stopwords:

Model Selection

I double checked my baseline accuracy before model deployment:

Target	Baseline accuracy	Interpretation
1	0.541151	➢ Changed by 4% (from 50 to 54%) subreddit posts are crypto
0	0.458849	

- Tested **Logistic Regression**, **Random Forest Classifier, and Support Vector Machine Classifier.** The best among those is **TF-IDF** and **Logistic Regression:**

Train score is 0.9089

Test score is 0.7438

(Model Selection … cont'd)

With TF-IDF and Logistic Regression with train score = 0.9089 & test score = 0.7438, the corresponding confusion matrix table is:

	Predicted Negative	Predicted Positive
Actual Negative	TN	FP
Actual Positive	FN	TP

	Predicted Stock	Predicted Crypto
Stock	238	133
Crypto	74	363

Specificity: spec = tn / (tn + fp) = 0.6415

⇢ the model predicted 64.15% of the posts belong to the stock market subreddit

⇢ Type I Error (or FP) = 1- spec = 0.3585

- ➢ the model incorrectly predicted 35.85% of the post as cryptocurrency subreddit

Sensitivity: sens = tp/(tp+fn) = 0.8307

⇢ the model correctly predicted 83.07% of the posts belong to the cryptocurrency subreddit

⇢ Type II Error (or FN) = 1- sens = 0.1693

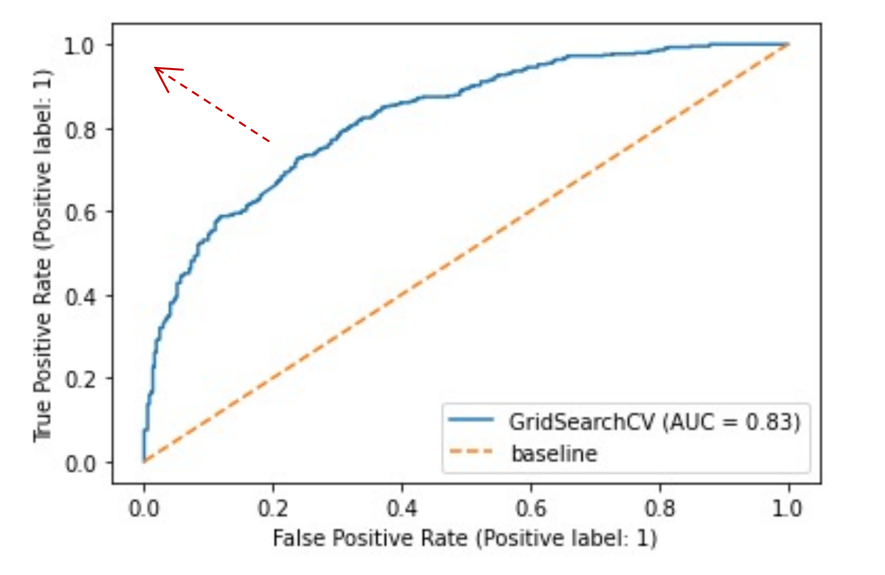
- ➢ the model incorrectly predicted 16.93% of the post as StockMarket subreddit

Accuracy: acc = (tp + tn)/(tp + tn + fp + fn) = 0.7438

⇢ the model predicted 74.38% of the posts correct

GENERAL ASSEMBLY

Receiver Operating Characteristic (ROC) Curve



- Area under the ROC curve = 0.83

I want see the blue curve to be as close as possible to a square corner, thus making the area under the curve as close to 1 as possible, but it’s far but not bad.

False positive rate = type I error
=1 − specificity = FP / (FP + TN)
False negative rate = type II error
= 1 − sensitivity = FN / (TP + FN)

The ROC curve is a plot of the True Positive Rate (sensitivity) vs. the False Positive Rate (1 - specificity) for all possible decision thresholds.

Conclusions and recommendations

My Best scoring model: Logistic regression, Train / test score: 0.9089/0.7438

Potential improvements: collect more training data, do more data cleaning and preprocessing (remove more stop words i.e., numbers, stem/lemmatize i.e. -ing verbs), more intensive gridsearching to optimize models, try more models (boosting, SVM)

>>> Steps Forward:

Getting real-time data using webscraping of the subreddits, make fresh predictions and make Sentiment Analysis.

Thank You

Time for suggestions, Comments and questions