Classifying posts from r/Stocks & r/CryptoCurrency using NLP, ML

By Nicholas Nguyen

Table Of Contents

Problem

The goal of this presentation is to:

- Show the audience which classifier model performs best at predicting which subreddit the posts came from based on given subreddit texts

1. Gathering Data

Subreddits:

- r/CryptoCurrency
- r/Stocks

API:

- Pushshift.io
- Grabbed only submissions, no comments
- Pulled 5k submissions from each subreddit

2. Data Preparations

Cleaning:

- Dropped “u/Dont-know-stocks”
  - This would show up frequently in my pulls
- Dropped duplicates & null values
- Dummifying the subreddit values

Results:

- 9300 posts left.

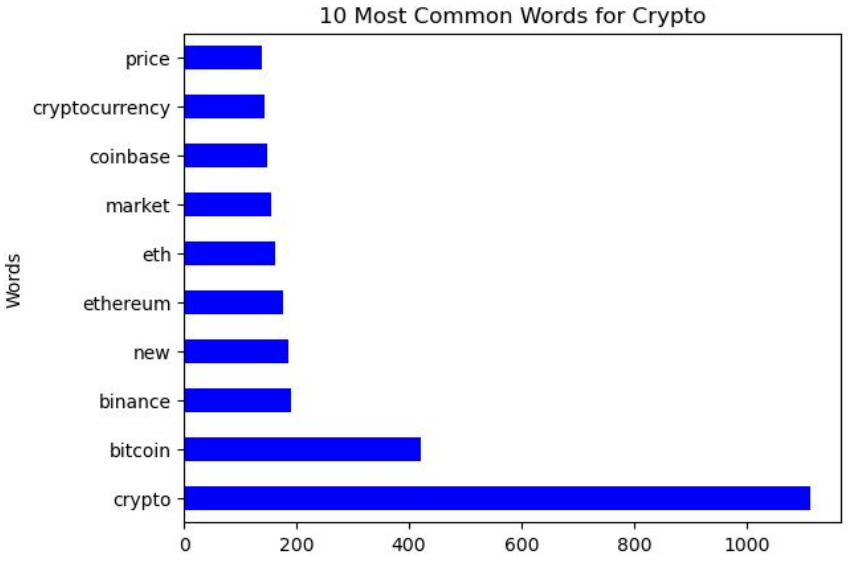
3. Preprocessing

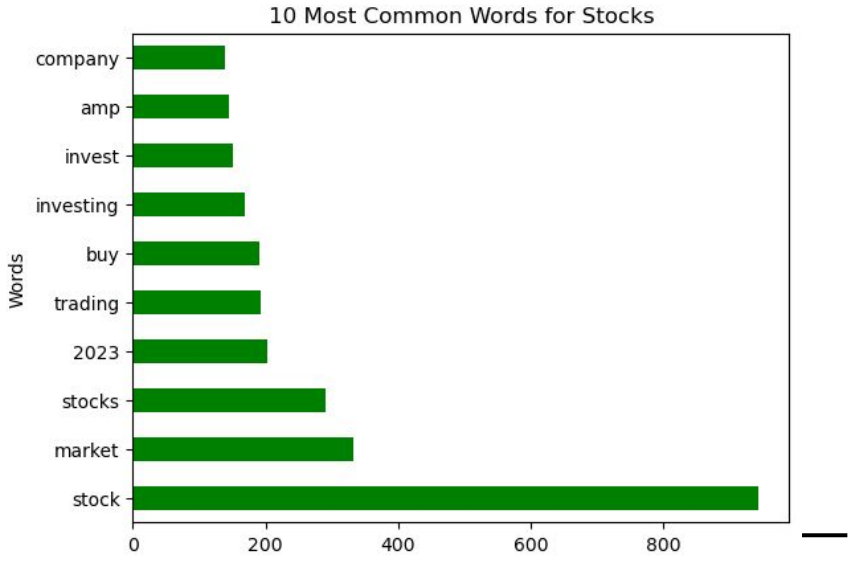
1. Lemmatize
2. Train/Test split (0.25 test size)
3. CountVectorizer: “Grabbed most frequently used words”
    a. Charted out results for both subreddits
4. TfidfVectorizer: “Identified the most important words”
    a. Charted out results for both subreddits

EDA: Most Common Words

Results from CountVectorizer:

- Stop words: ‘english’

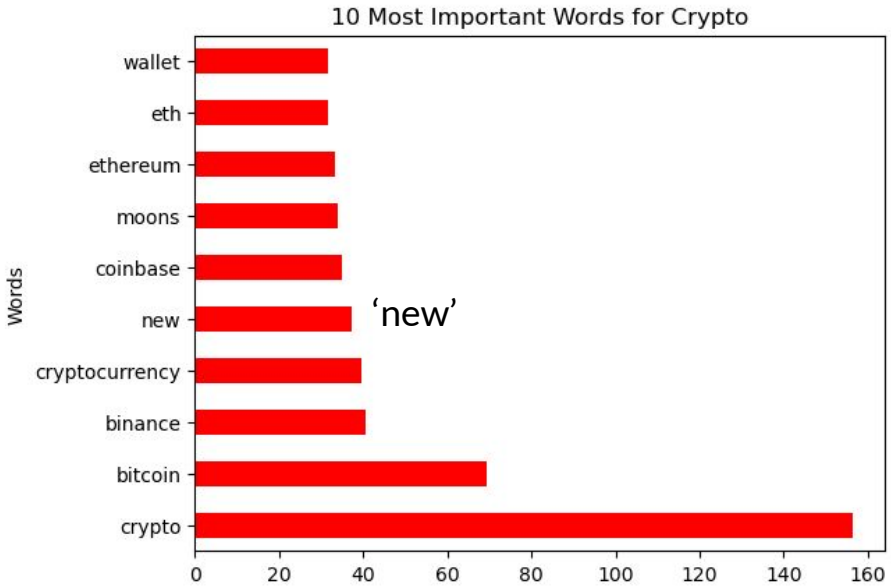


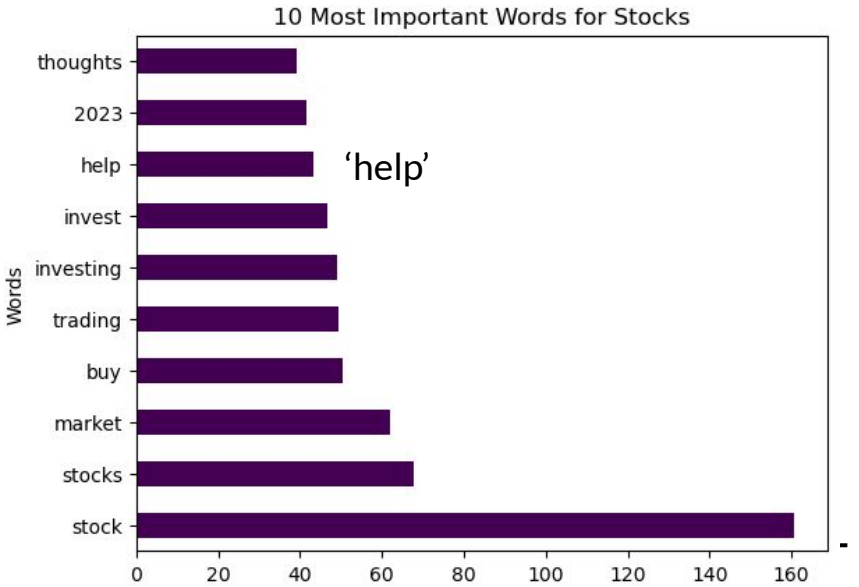


EDA: Most Important Words

Results from TfidfVectorizer:

- Stop words: ‘english’





Final Step. Modeling:
Logistic Regression
K-Nearest Neighbors Class.
Support Vector Classifier

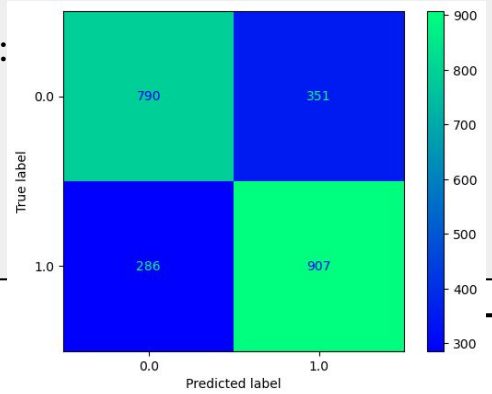
Modeling (K-Nearest Neighbors Classifier)

Two Separate Pipelines, Confusion Matrix to represent result!

Count Vectorizer:

- Training score: 0.8278
- Test score: 0.7270
- **Specificity: 0.6923**
- **Accuracy: 0.7270**

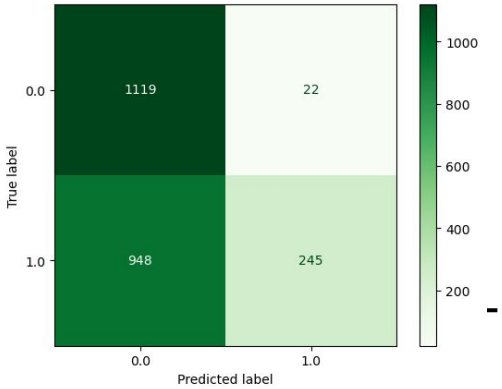
Confusion Matrix:



Tfidf Vectorizer (highly overfit):

- Training score: 0.9530
- Test score: 0.5844
- **Specificity: 0.9807**
- **Accuracy: 0.5844**

Confusion Matrix:

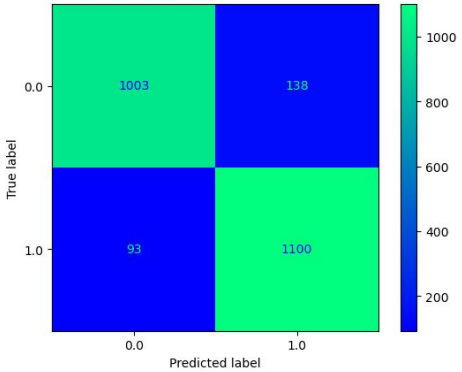


Modeling (Logistic Regression)

Two Separate Pipelines, Confusion Matrix to represent result!

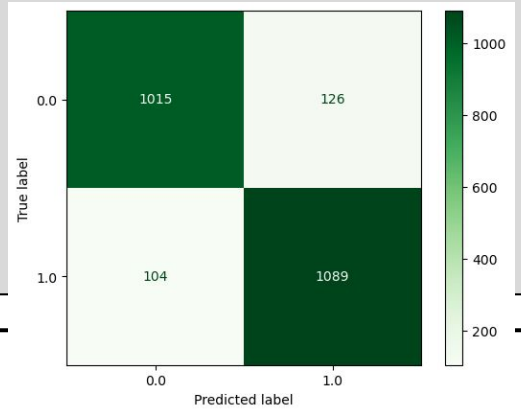
Count Vectorizer:

- Training score: 0.9588
- Test score: 0.9010
- **Specificity: 0.8790**
- **Accuracy: 0.9010**



Tfidf Vectorizer:

- Training score: 0.9530
- Test score: 0.9014
- **Specificity: 0.8895**
- **Accuracy: 0.9014**

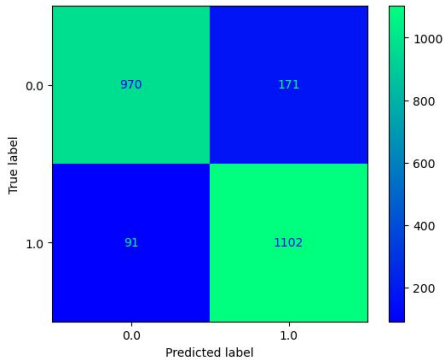


Modeling (Support Vector Classifier)

Two Separate Pipelines, Confusion Matrix to represent result!

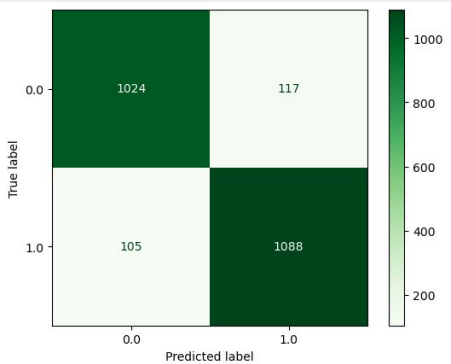
Count Vectorizer:

- Training score: 0.9522
- Test score: 0.8877
- **Specificity: 0.8501**
- **Accuracy: 0.8877**



Tfidf Vectorizer:

- Training score: 0.9910
- Test score: 0.9048
- **Specificity: 0.8974**
- **Accuracy: 0.9048**



Tuning for best vect. Per model

KNN (CVEC):

- Max df. 0.9
- Max Feat. 8000

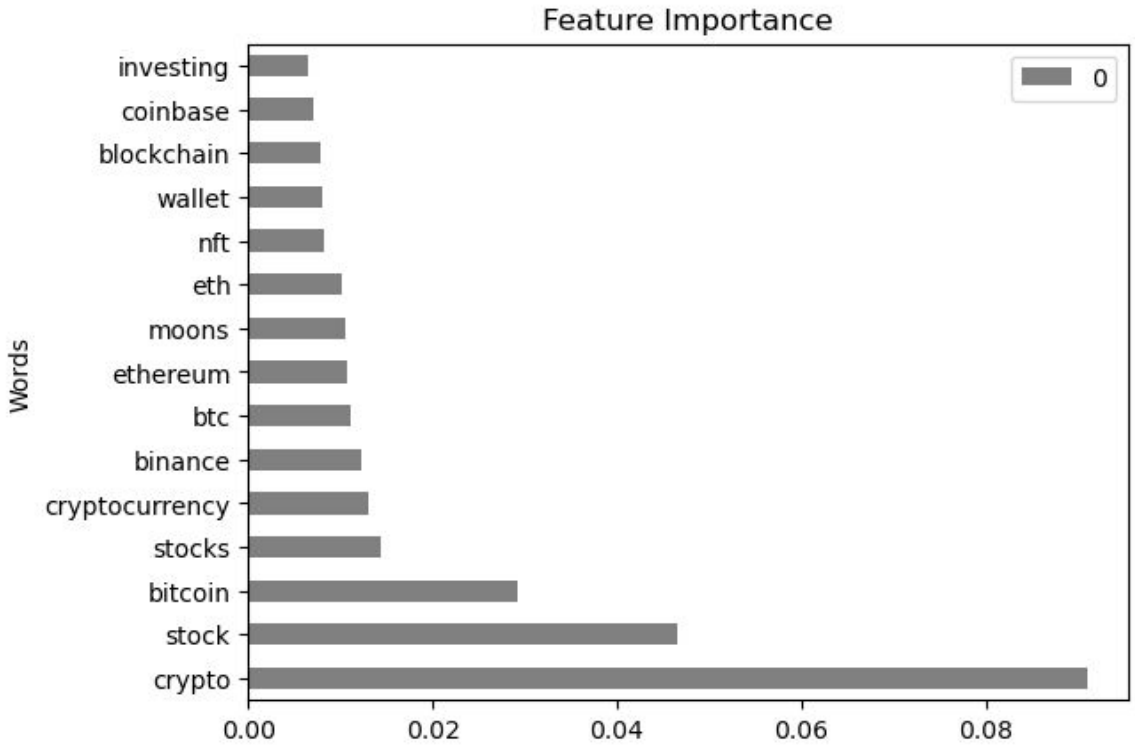
SVC(TFIDF):

- Ngram (1,1)
- Max Feat. 7000

LogisticRegression(TFIDF):

- C: 1.0
- Penalty: L2

Extra: Feature Importance using RF



Conclusion:

The best scoring model for this problem is: **Support Vector Classifier**

Training score: 0.9910

Test score: 0.9048

Questions?