WEB API &
Natural Language Processing

Prepared by Clarence Alvarez

Project Goal

Construct classification models and choose one that predicts which of two subreddits a random post belongs to.

Models used:

- K-Nearest Neighbors
- Random Forest Classifier
- Multinomial Naive Bayes

Metric of Success: Accuracy of models against Null model

Data Collection

Intent:

Automate collection with a while-loop, while also dropping posts that:

- Were removed/deleted
- Contain media files (video/photos)
- Duplicates (weekly discussion posts)
- Empty ‘selftext’

Anticipated resulting data to be unique posts that require little cleaning

Collected a total of 20,053 observations

Note: Wait time = 5 seconds between requests

Data Cleaning and EDA

Word Count positively skewed

Minimum: 1

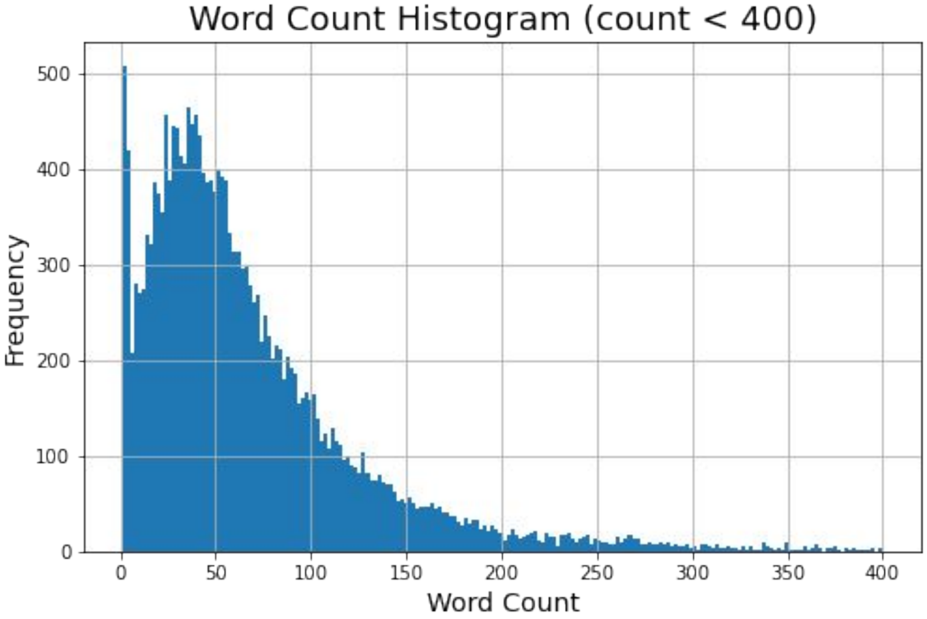
Maximum: 4048

Mean: 71.49

Median: 52

Mode: 40

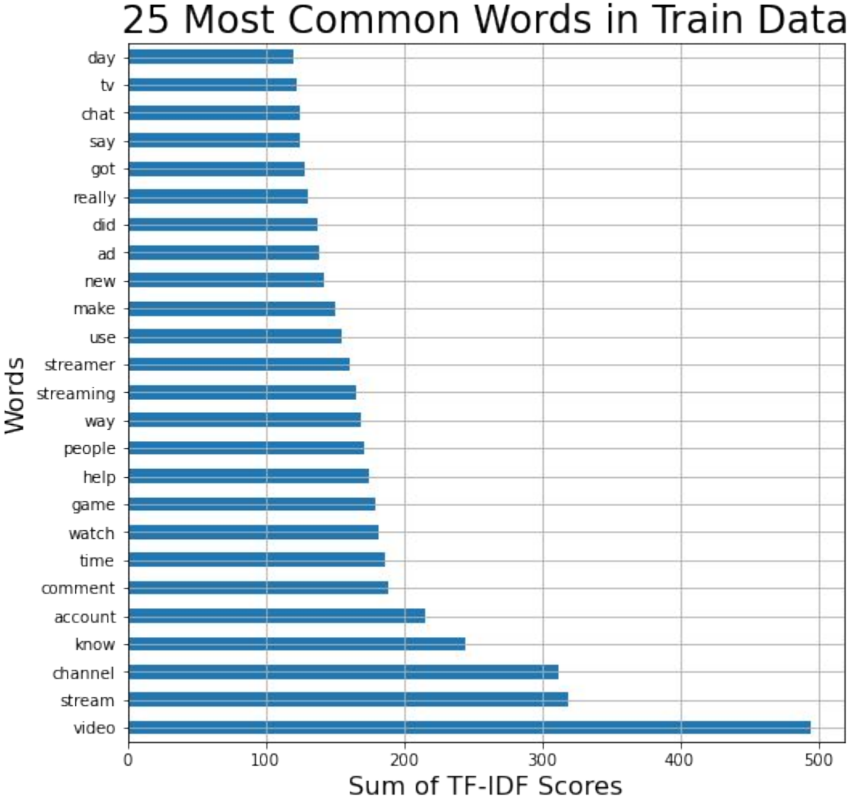
Standard Deviation: 79.16



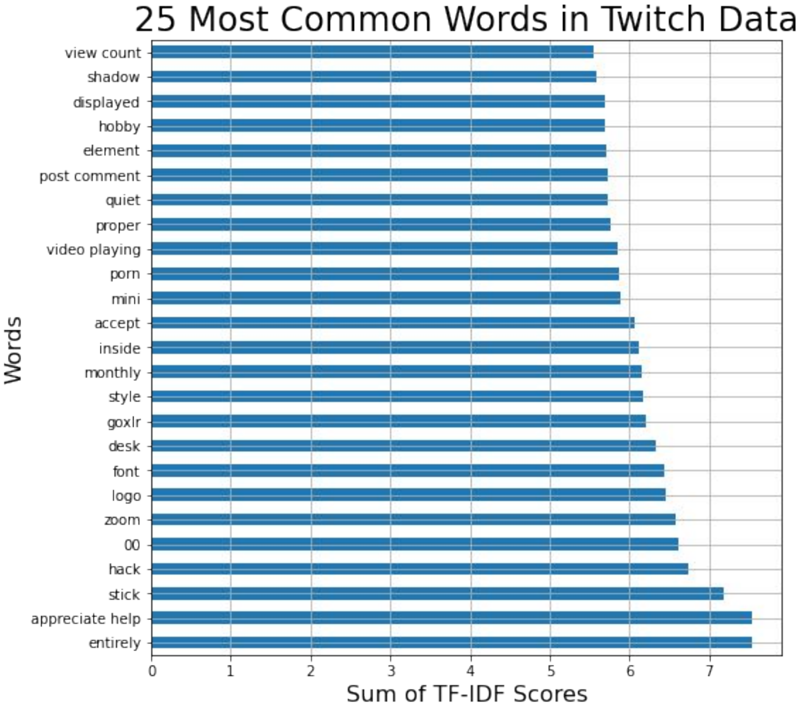
Data Pre-processing

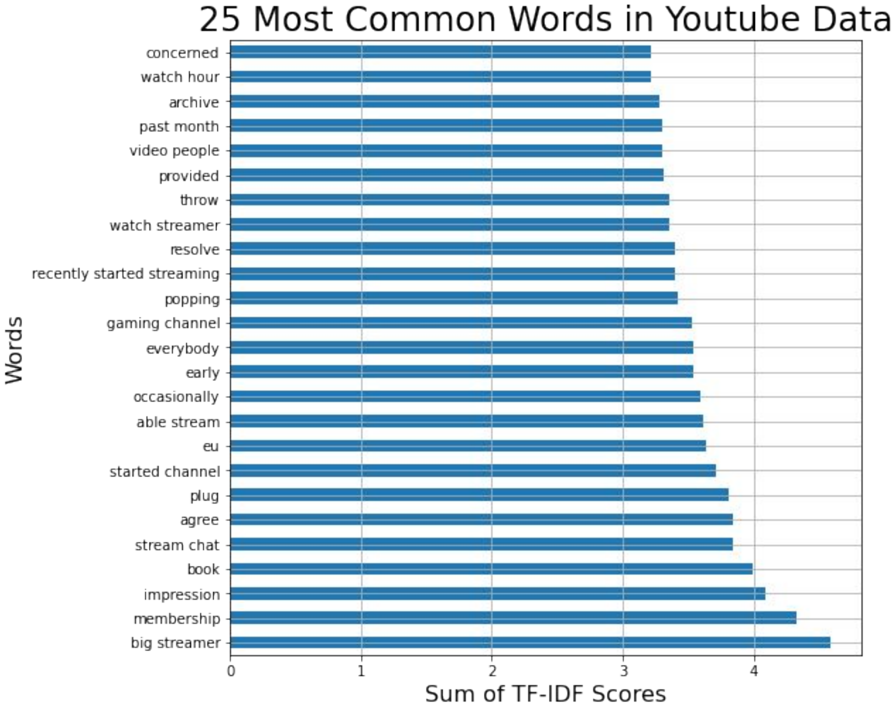
Steps taken on predictors (‘selftext’):

- Expanded contractions
- Made all text lower case
- Lemmatized
- Removed punctuations
- Created a custom list of stopwords and combined with those found in ‘text’ module and TF-IDF Vectorizer
- Train-Test-Split (test size = 0.20)
- Used TF-IDF Vectorizer
  - max features = 50,000
  - ngram_range = (1, 3)



Most Common Words Unique to Each Subreddit





Please note: differences in scales against the overall top words. Suggests similar verbage used in both subreddits.

Model Hyperparameters

Grid-Search used to tune the following hyperparameters for the given models

K-Nearest Neighbors	Random Forest Classifier	Multinomial Naive Bayes **tuned cvec hyperparameters*
n_neighbors = [51, 1,501] varying steps to decrease fitting time	max_depth = [66, default] default: nodes expanded until all leaves pure or less than min_samples_split	max_features = [8,000, 10,000] default consider only top max_features ordered by term frequency across corpus

weights = [uniform, distance]	min_samples_split = [5, 20]
	default = 2
		max_df = [0.8, 0.9]
		default = 1.0

	min_samples_leaf = [3, 10]
	default = 1
		min_df = [2]
		default = 1

K-Nearest Neighbors

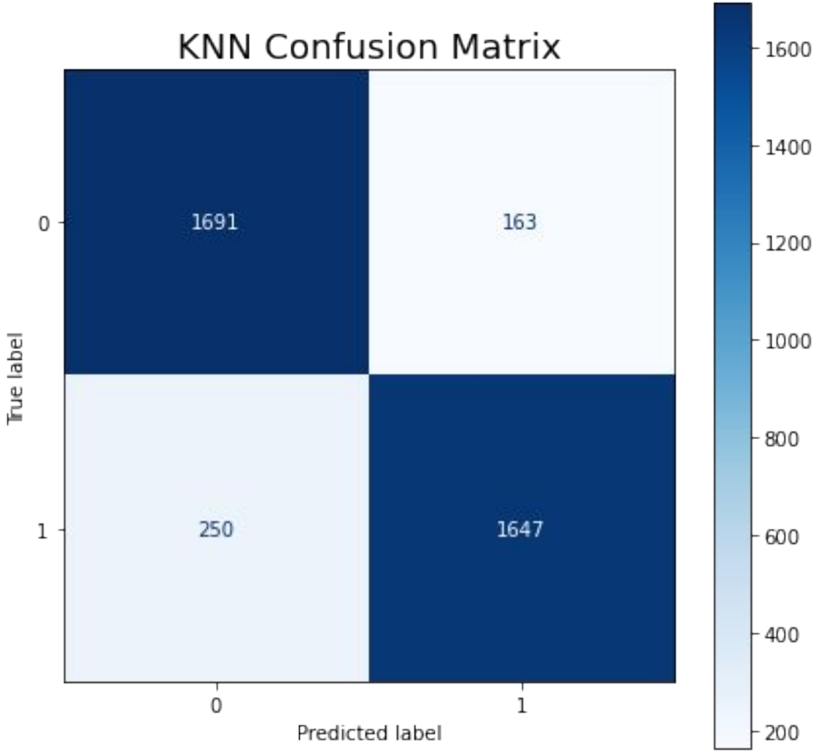
Best Parameters:

n_neighbors =	701
weights =	distance

Scores:

Train:	0.9996
Test:	0.8898

Model is overfit, but better than baseline



Baseline: 0.5057

Train Size = 15,002, Test Size = 3,751

1 = twitch, 0 = youtube

Random Forest Classifier

Best Parameters:

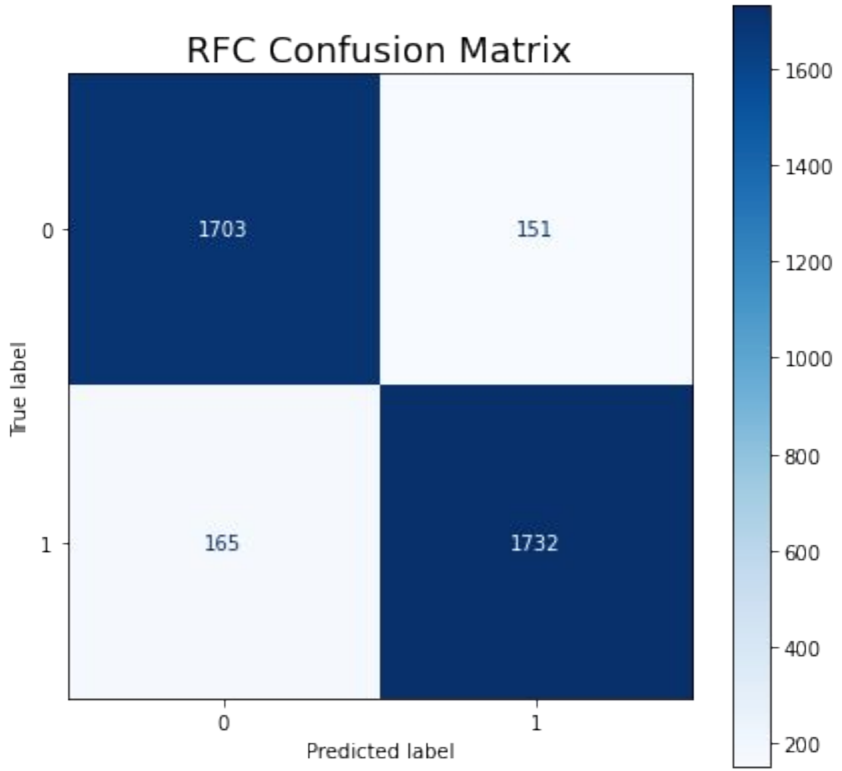
max_depth = 100

min_samples_split = 5

min_samples_leaf = 3

Scores:

Train:	0.9407
Test:	0.9157



Model is less overfit than KNN, beats baseline

Baseline: 0.5057

Train Size = 15,002, Test Size = 3,751

1 = twitch, 0 = youtube

Multinomial Naive Bayes

Best (cvec) Parameters:

min_df = 2

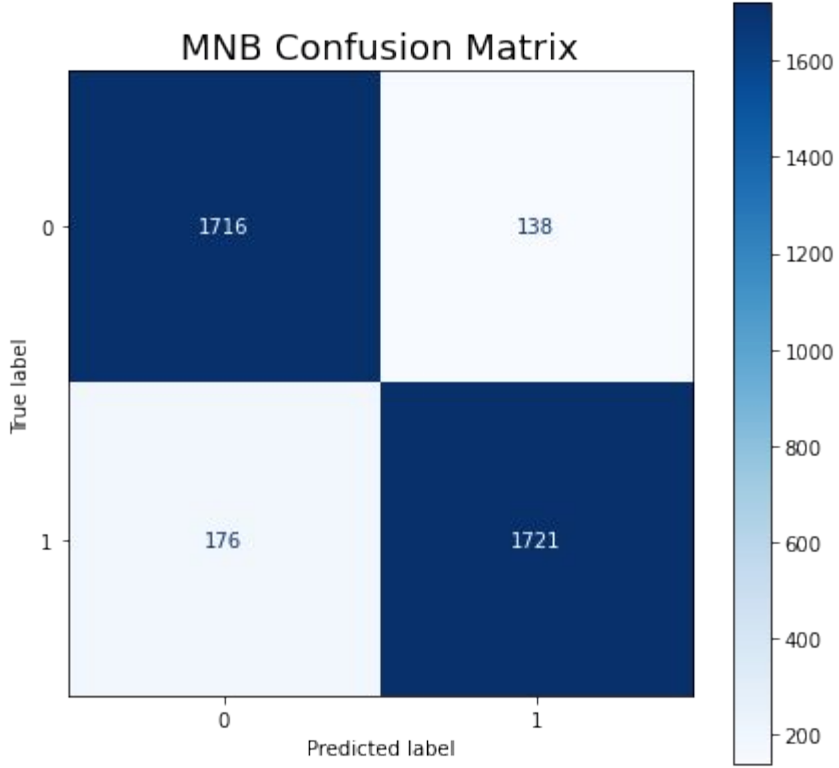
max_df = 0.8

max_features = 10,000

Scores:

Train:	0.9216
Test:	0.9162

Model is fairly balanced and beats baseline.



Baseline: 0.5057

Train Size = 15,002, Test Size = 3,751

1 = twitch, 0 = youtube

Conclusion/Recommendation

In choosing amongst these selected models to determine which of two subreddits a random post belongs to, I recommend using Random Forest Classifier.

- Performed better than KNNClassifier
- Similar in performance to MNB, but also offers insight into which features are important in reducing Gini impurity
    - Top five features:
        - ‘video’
        - ‘stream’, ‘streamer’, ‘streaming’
        - ‘game’
        - ‘comment’
        - ‘chat’