Natural Language Processing (NLP)

Patriots Posts
vs
NFL Posts

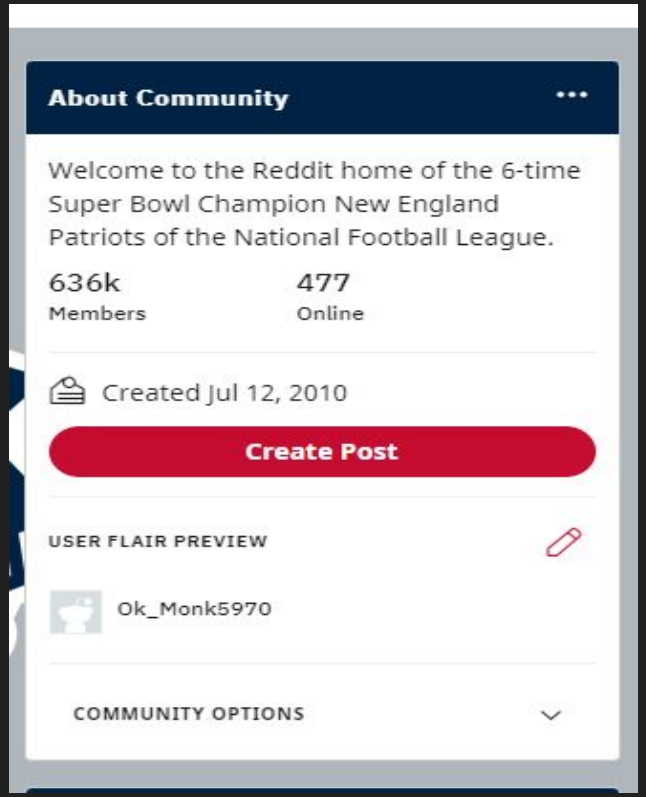
Martin Berberi GA DSIR-222

How can Natural Language Processing better distinguish posts from different sources?

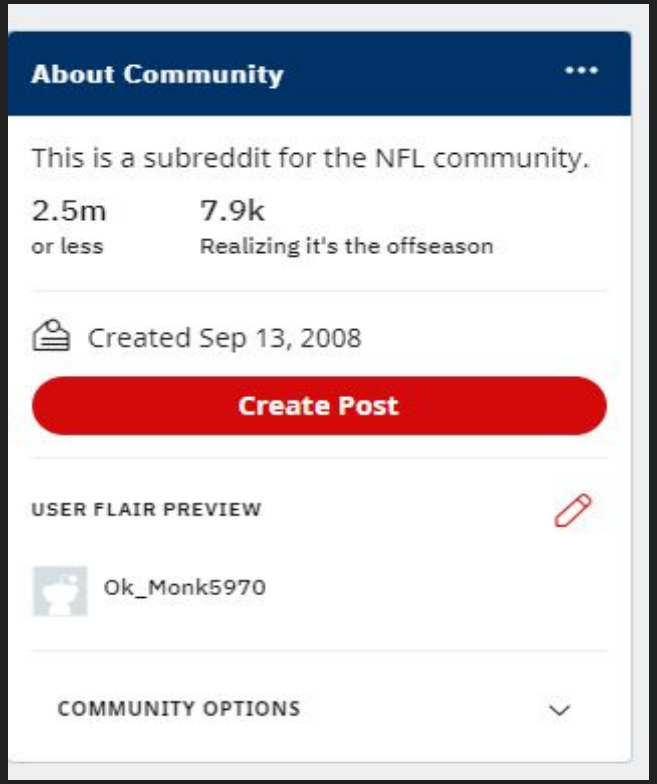
CASE STUDY:

Detection between Patriots titles and NFL titles by building a classification model trained on each Subreddit titles.

r/Patriots



r/NFL



Methodology

1

Data Gathering

Scrapping ~40,000 posts from Patriots & NFL with the help of Pushshift API

2

Exploratory Data Analysis

Investigate data and create data visualization to observe patterns and distinguish each category characteristics

3

Natural Language Processing

Prepare the data for modeling. After cleaned by duplicates, punctuation Vectorize the data

4

Modeling

Find the combination of model and vectorizer for the best accuracy score.

Data Gathering

- 39,989 posts: 19,993 posts from r/Patriots and 19,996 posts from r/NFL
  - 38,141 posts after dropping duplicates
  - Features: Subreddit (target), title (predictor), author, domain, created_utc

- Data cleaning:
  - Drop Duplicates
  - Check for null values
  - Cleaning the title from punctuation, numbers, extra spaces

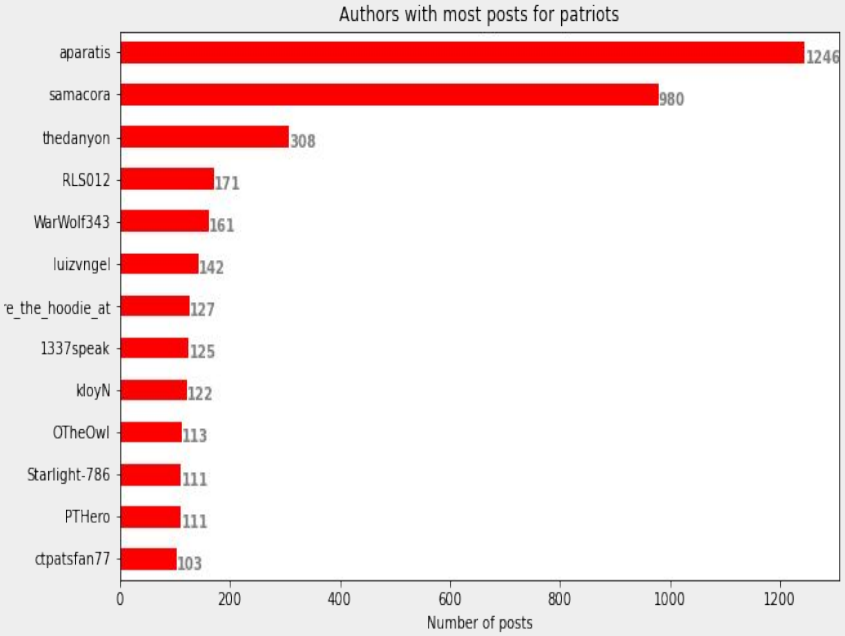
Exploratory Data Analysis

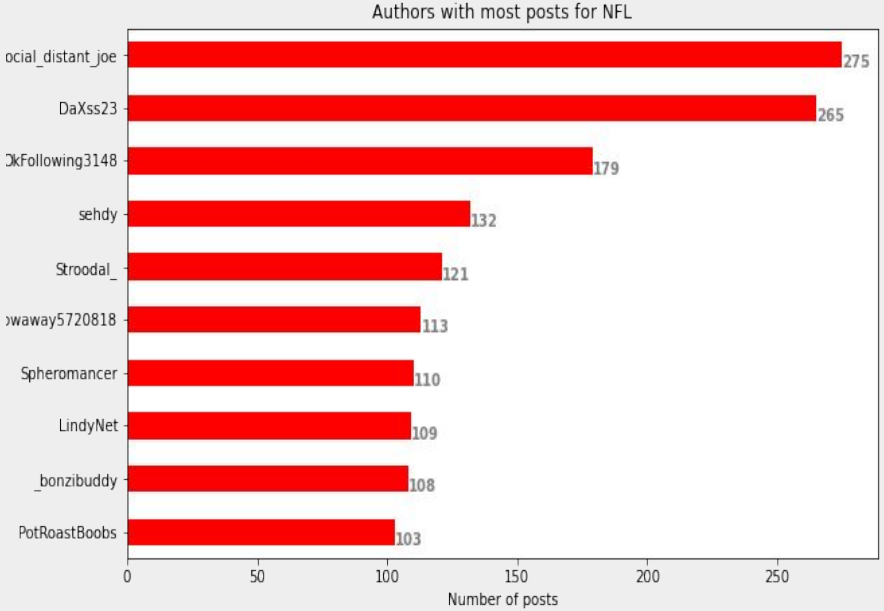
EDA

Top Authors r/Patriots & r/NFL

- Aparatis
  - samacora
    - thedanyon

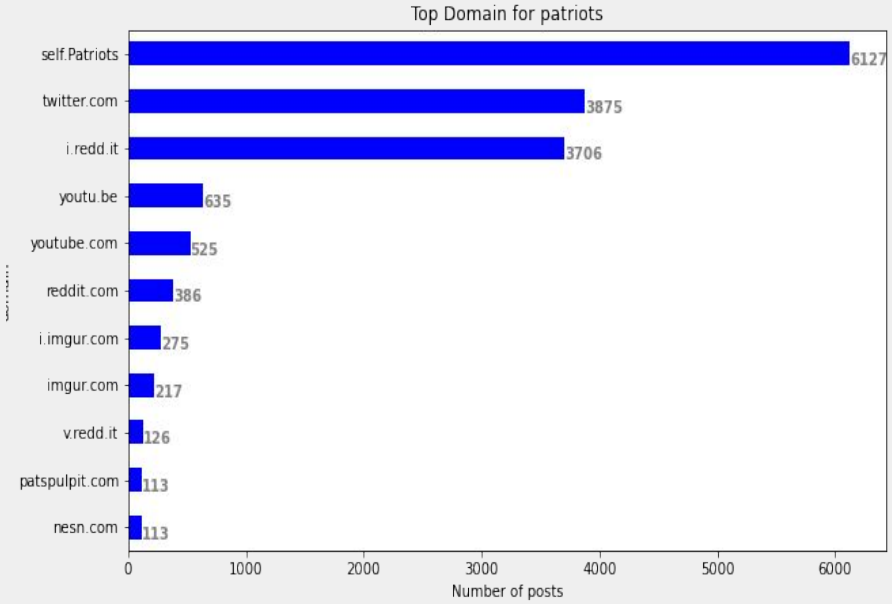
- Social_distant_joe
  - DaXss23
    - OkFollowing3148



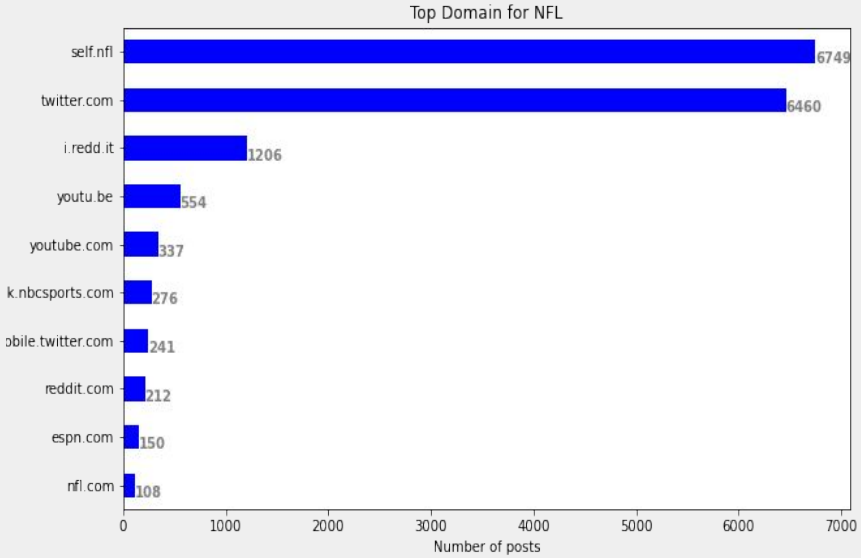


Top Domains r/Patriots & r/NFL

- self.Patroits
  - Twitter
    - i.redd.it



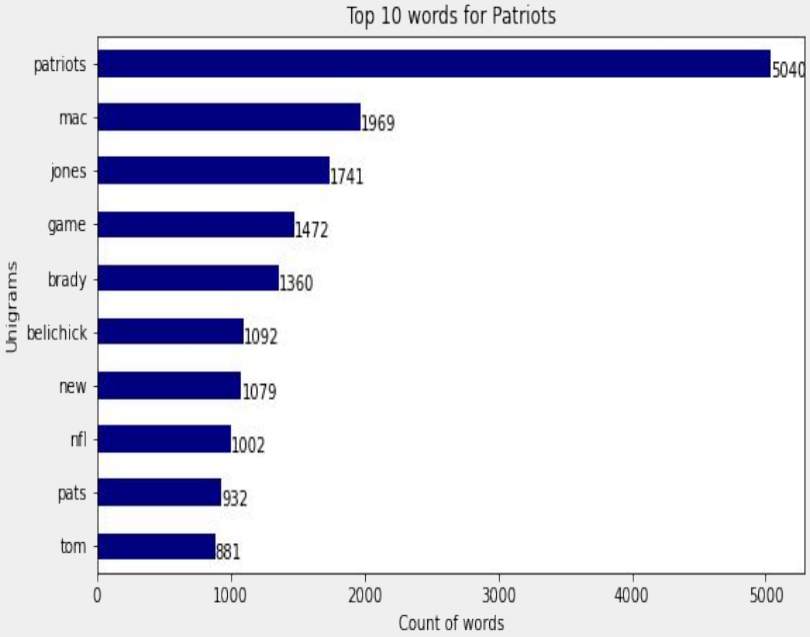
- Self.nfl
  - twitter
    - i.redd.it



Top 10 Words r/Patriots & r/NFL

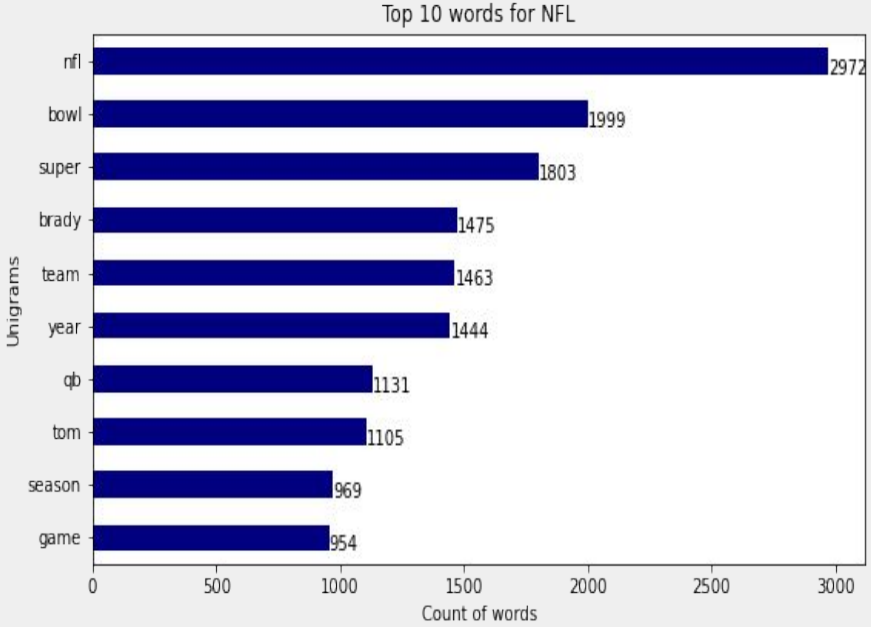
Game

Brady



Tom

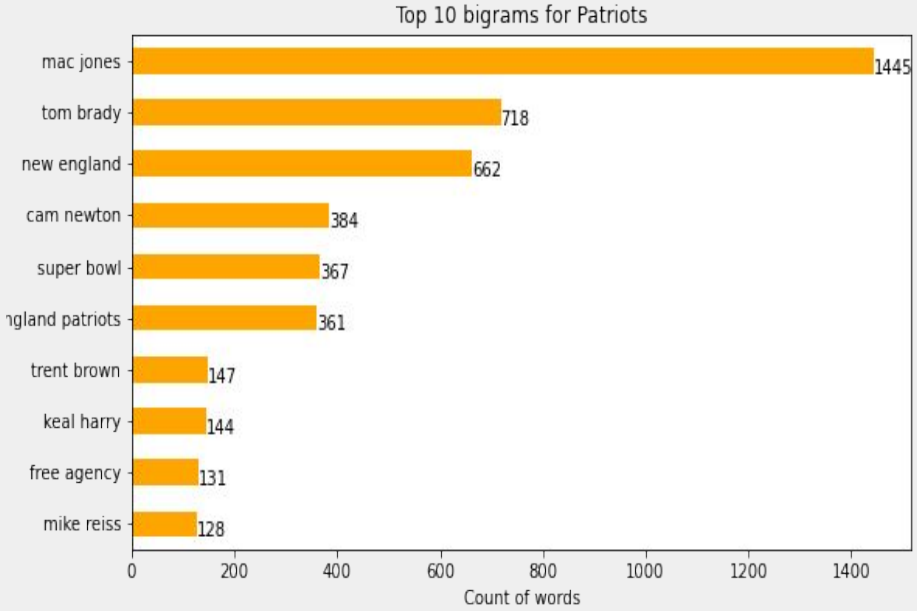
NFL

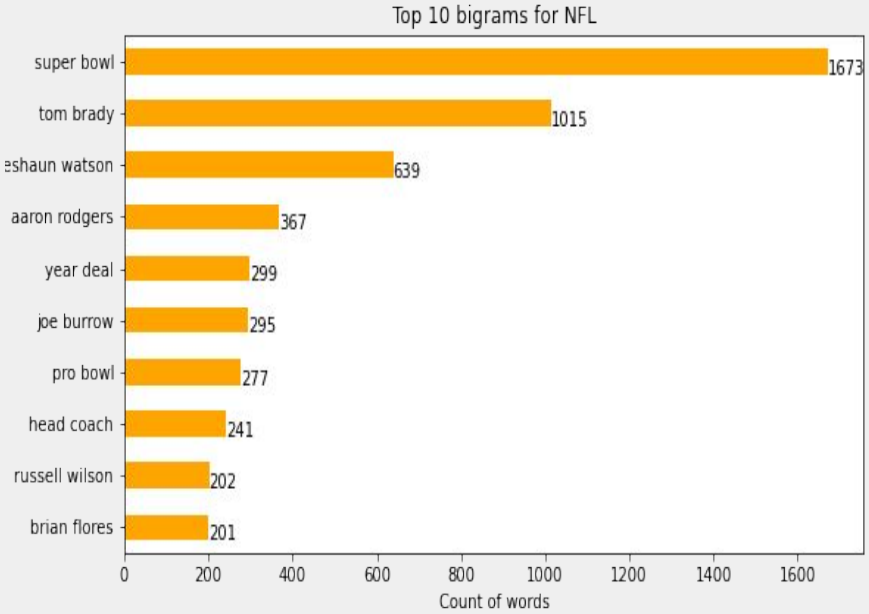


Top 10 Bi-Grams r/Patriots & r/NFL

- **Tom**
  - **Brady**

- **Super**
  - **Bowl**





MODELING

Pipeline & GriedSearchCV

Model 1

CountVectorizer()
stop_words: [None, ‘english’, my_stop_words]
ngam_range: [(1, 2), (1, 3)]
max_df: [0.75, 0.85]
min_df: [2, 3]
Logistic Regression()
C: [0.35, 0.75, 1.0]

Model 2

TfidVectorizer()
stop_words: [None, ‘english’, my_stop_words]
ngam_range: [(1, 2), (1, 3)]
max_df: [0.75, 0.85]
min_df: [2, 3]
Logistic Regression()
C: [0.5, 1.0]

Model 3

CountVectorizer()
stop_words: [None, ‘english’]
ngam_range: [(1, 2), (1, 3)]
max_df: 0.75
min_df: 2
MultinomialNB()
alpha: 1.0

Model 4

TfidVectorizer()
stop_words: None
ngam_range: [(1, 2), (1, 3)]
max_df: 0.75
min_df: 2
MultinomialNB()
alpha: 1.0

Model 1. Best Accuracy Score: 86%

Model 1

CountVectorizer()
stop_words: None
ngam_range: [(1, 3)]
max_df: 0.75
min_df: 2
Logistic Regression()
C: 0.35
Accuracy : 86.16%

Model 2

TfidVectorizer()
stop_words: None
ngam_range: [(1, 2)]
max_df: 0.75
min_df: 2
Logistic Regression()
C: 1.0

Accuracy: 86.10%

Model 3

CountVectorizer()
Stop_words: None
ngam_range: [(1, 2)]
max_df: 0.75
min_df: 2
MultinomialNB()
alpha: 1.0

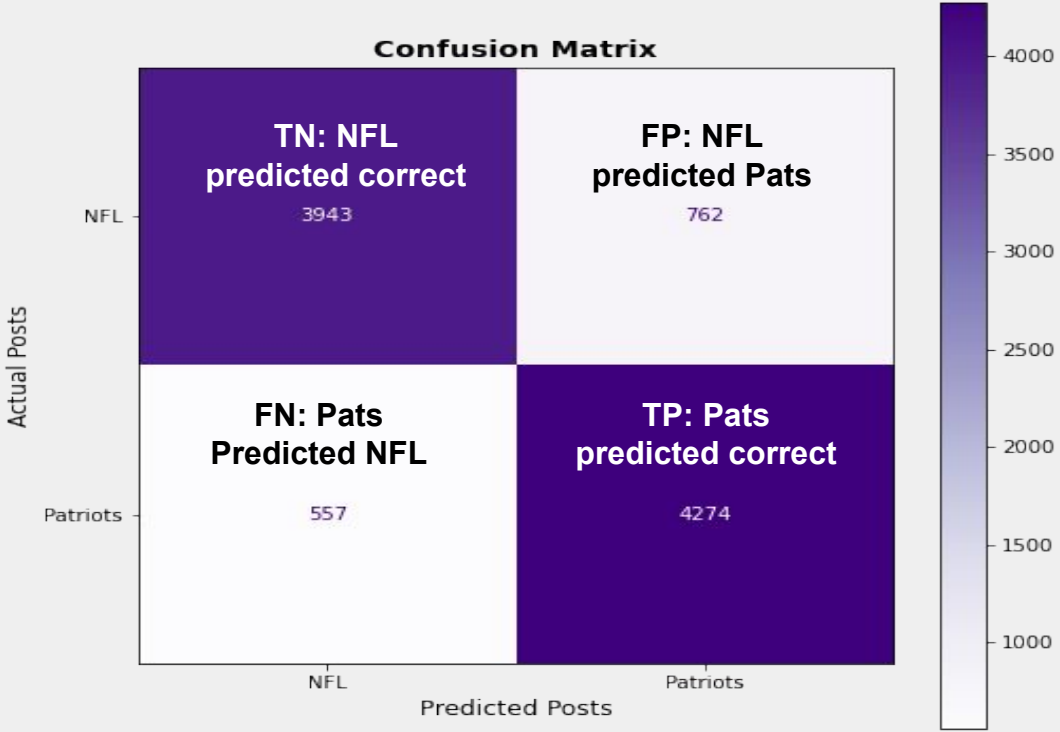
Accuracy: 85.39%

Model 4

TfidVectorizer()
stop_words: None
ngam_range: [(1, 2), (1, 3)]
max_df: 0.75
min_df: 2
Logistic Regression()
alpha: 1.0

Accuarcy: 85.42%

Model 1. Confusion Matrix



Accuracy
The model predict correct ~86% of the posts

Precision
For all pats prediction, ~85% are predicted correctly.

True Positive Rate(Sensitivity)
For all pats posts, ~88% are predicted correctly

True Negative Rate(Specificity)
For all NFL posts, ~84% are predicted correctly

Misclassification Rate
For all predictions ~14% predicted incorrectly

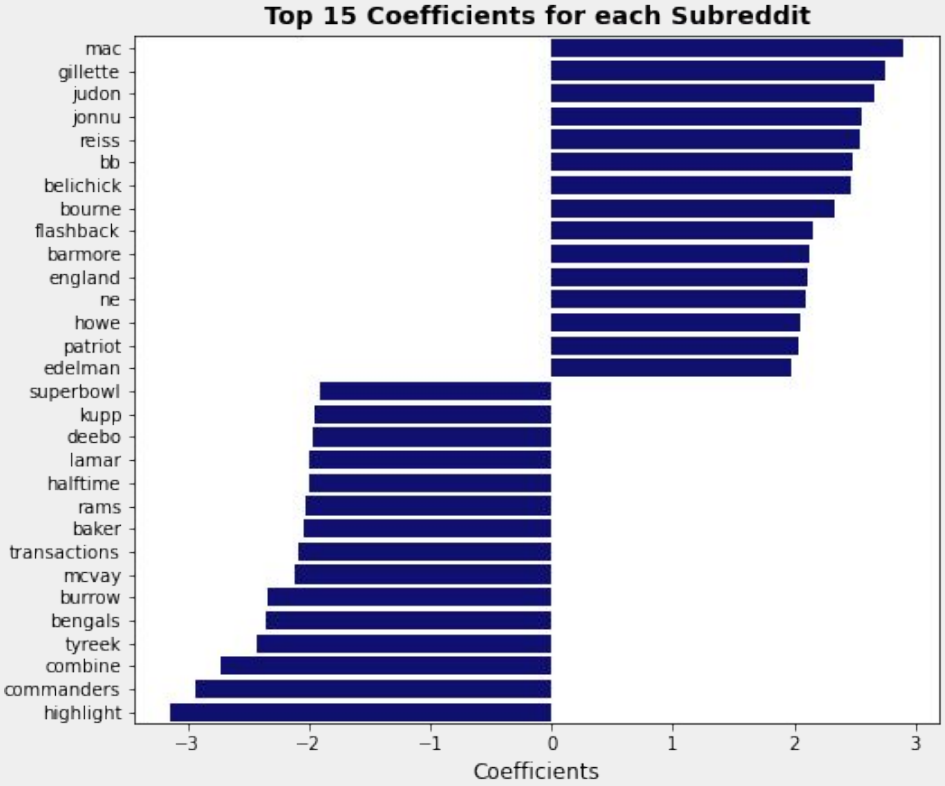
Logistic Regression Coefficients

r/Patriots:

The words that shows the most positively coefficients rate are ‘mac’ followed by ‘gillette’ and ‘judon’. *Increasing the presence of word ‘’mac” by 1 in title, that title is 18.16 times as likely to be classified as Patriots subreddit.*

r/NFL

The words that shows the most positively coefficients rate are highlight’ followed by ‘commanders’ and ‘combine’.
Increasing the presence of word 'highlight' by 1 in title, that title is 23 times as likely to be classified as NFL subreddits



Conclusions

The best model to distinguish the patriots post from nfl post is Logistic Regression with CounterVectorizer with accuracy score of 86%.(86% of the posts are predicted correctly).

Disadvantages

Natural Langage Processing (NLP) with images????