Project 3

Reddit API Classification & NLP

Chan Song Yuan
General Assembly DSI14
May 18, 2020

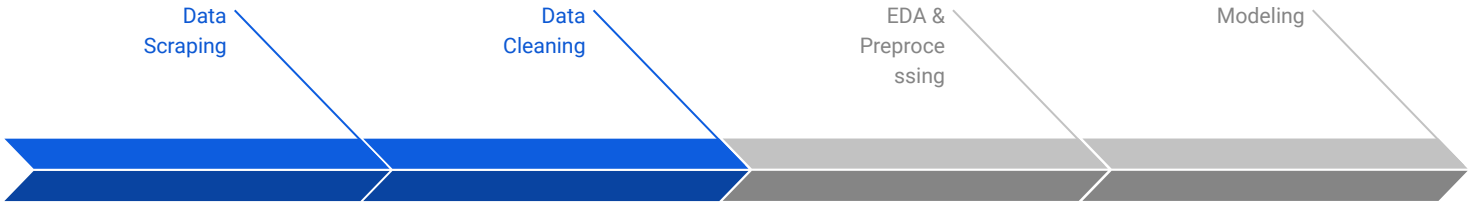
Outline

- **Problem Statement**
- **Data Information**
- **Data Cleaning & EDA**
- **Modeling**
- **Evaluation**
- **Conclusion & Recommendation**

Problem Statement

In this project, we will explore how well Natural Language Processing Model differentiate post content from two similar subreddits which is /r/Python & /r/bigdata. Which combinations of model and classifier works best? What is the accuracy and how much of the miss-classification will occur between two different subreddit posts?

My Collected Data



Data Collection

Scrape Data from Reddit API

Data Cleaning

Data cleaning and removal of special characters.

EDA and Preprocessing

To visualize the data to understand the data before proceed to pre-process the data collected

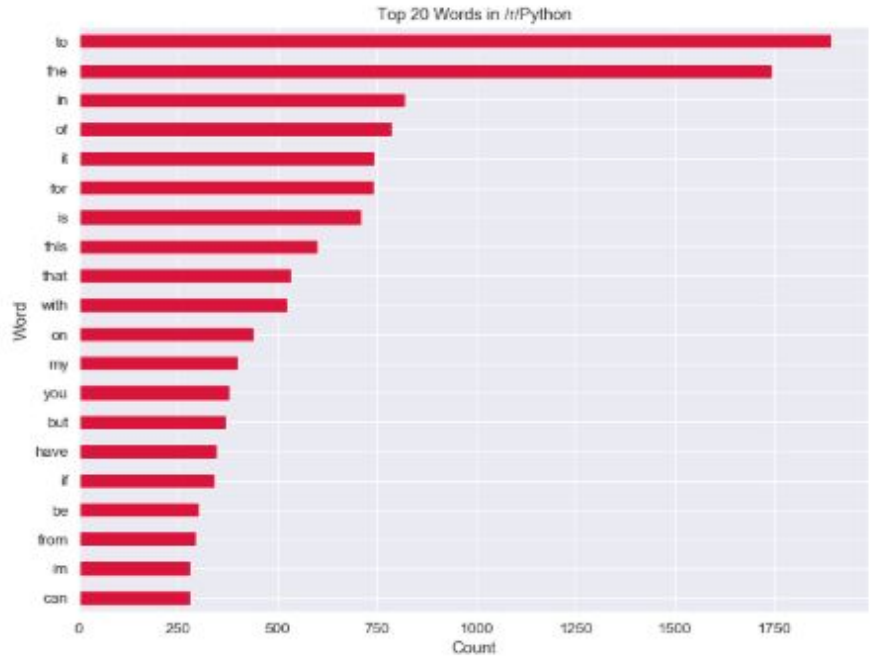
Modeling

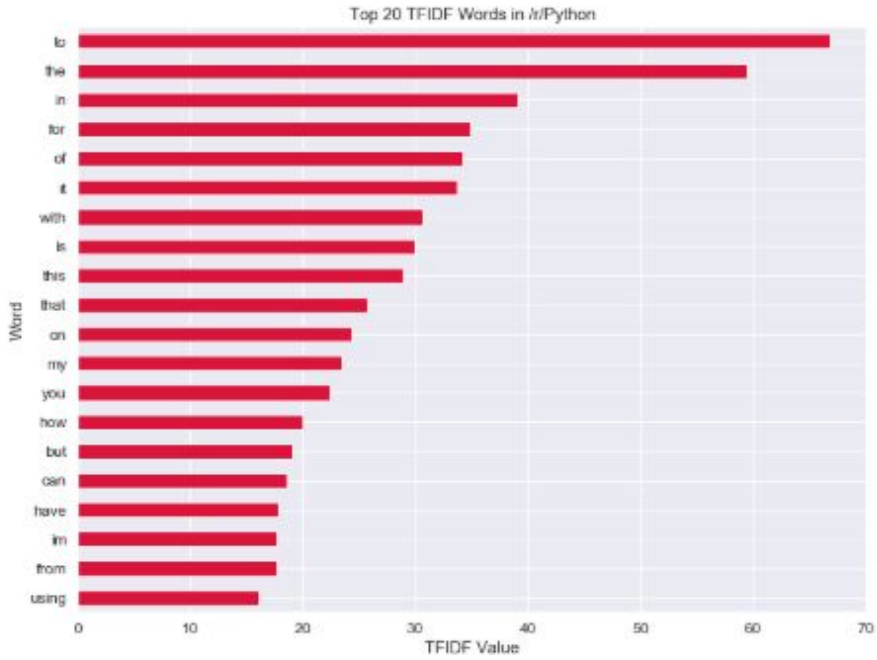
Different combinations of modeling to get best accuracy

- Total of 1837 rows of data collected after drop duplicates
  - 866 from r/Python
  - 971 from r/bigdata

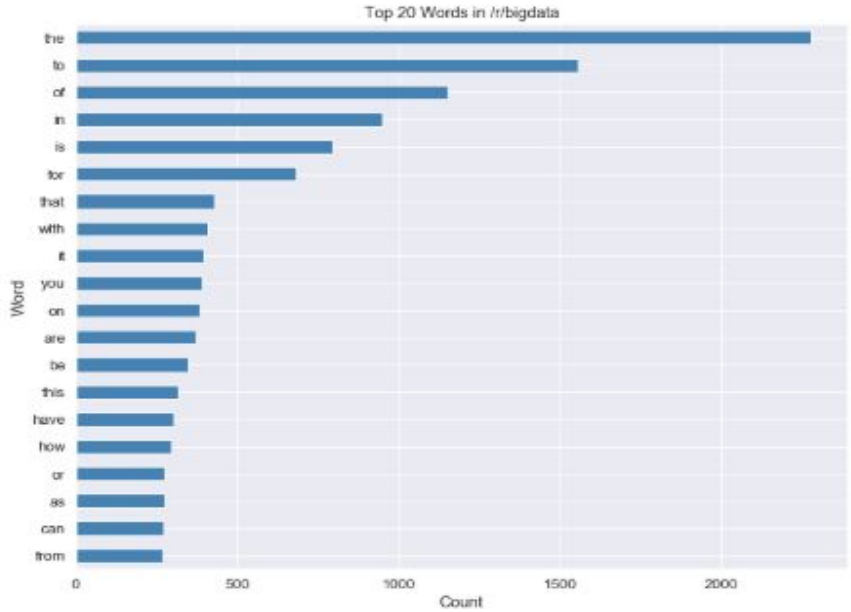
Frequent Words in r/Python and r/bigdata

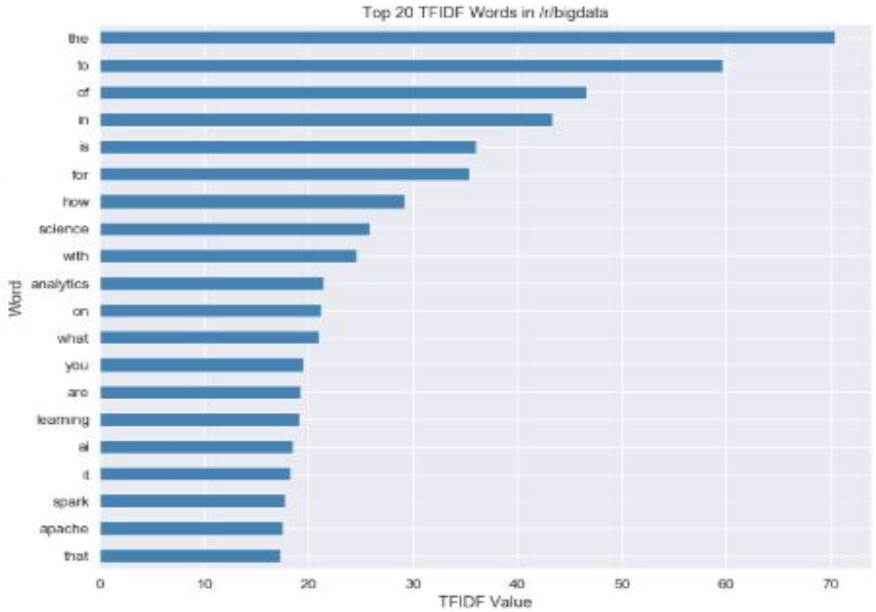
Frequent Words in r/Python





Frequent Words in r/bigdata





Modeling

Modeling with Best Hyperparameter Set with GridSearch CV

Model	Train Score (CountVectorizer)	Test Score (CountVectorizer)	Train Score (TfidfVectorizer)	Test Score (TfidfVectorizer)
Logistic Regression	0.9912	0.8397	0.9959	0.8397
Naive Bayes (Multinomial)	0.9176	0.8614	0.9367	0.8668
Random Forest	0.9966	0.8207	0.9939	0.8016
Decision Tree	0.9959	0.769	0.9966	0.7418

Comparing Classification Report

CountVectorizer with Naive Bayes (Multinomial)

	precision	recall	f1-score	support
r/Python	0.84	0.87	0.85	173
r/bigdata	0.88	0.86	0.87	195
accuracy			0.86	368
macro avg	0.86	0.86	0.86	368
weighted avg	0.86	0.86	0.86	368

TF-IDF Vectorizer with Naive Bayes (Multinomial)

	precision	recall	f1-score	support
r/Python	0.84	0.88	0.86	173
r/bigdata	0.89	0.85	0.87	195
accuracy			0.87	368
macro avg	0.87	0.87	0.87	368
weighted avg	0.87	0.87	0.87	368

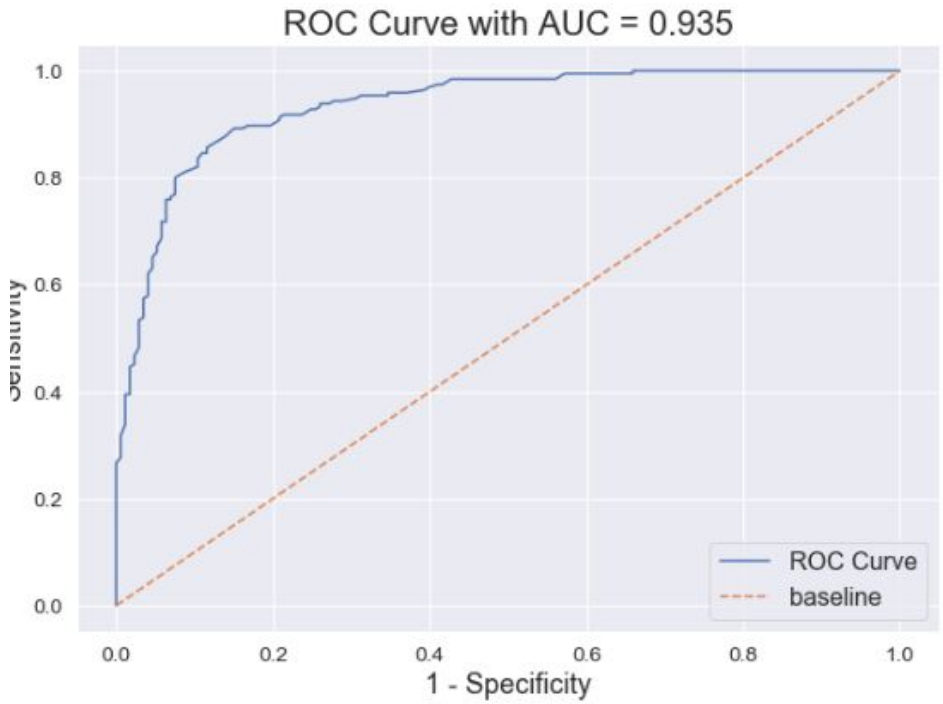
Confusion Matrix

Confusion Matrix based on TF-IDF Vectorizer with Naive Bayes Multinomial

	Predict r/Python	Predict r/bigdata
Actual r/Python	153 / 368	20 / 368
Actual r/bigdata	29 / 368	166 /368

ROC AUC

ROC AUC SCORE : 0.935



Conclusion

- **Naive Bayes with TF-IDF Vectorizer worked fairly well with accuracy of 87%, with both subreddits were related.**
- **Naive Bayes with Countvectorizer works well too with accuracy of 86%.**
- **Scope can be expanded to improve modes further:**
  - **Collect more subreddit posts**
  - **Tuning parameters for each model to achieve better scores, will take longer time to tune and compile best parameters**
  - **Consider other classifiers, AdaBoost, Gradient Boost, etc.**

THE END