Project 3: Web APIs & NLP

Benjamin Toh (DSIF-2)

Problem Statement: To identify the right subreddit given a specific post

In this project, we will be look at ‘NBA’ and ‘PremierLeague’ subreddits, with the intent to web scrape 5000 posts per subreddit

Timeline

Web Scraping

❑ Create a function to web scrape

Pre Processing

❑ Tokenize
❑ Lemmatize
❑ CountVectorizer

Modeling

❑ Naive Bayes
❑ Random Forest Classifier

Evaluation

❑ Confusion Matrix
❑ ROC - AUC

Conclusion

❑ Findings
❑ Future work

Web Scraping

- ❑ A function was created for web scraping where 3 inputs are required (subreddit, no. of post to scrape and no. of times to scrape) and remove duplicates

Note: Limited to max 100 posts per scrape

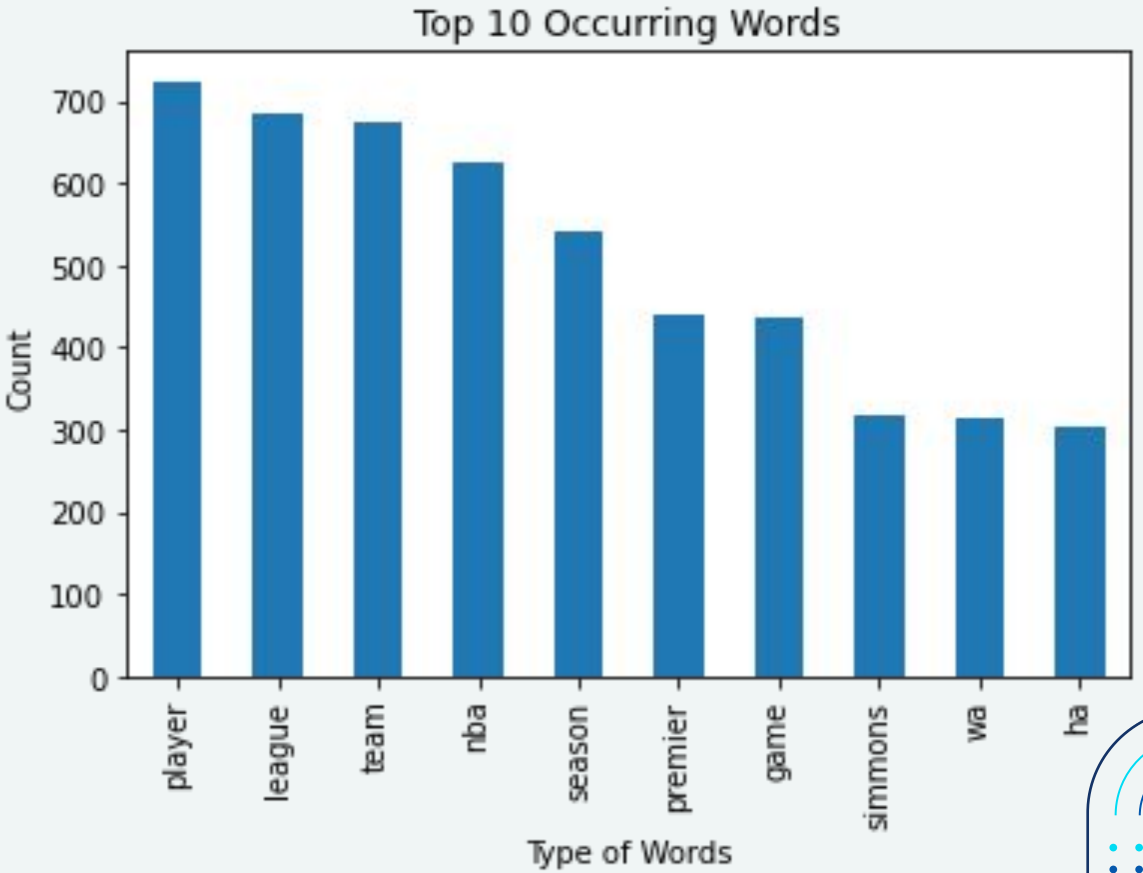
- ❑ 4737 posts from ‘NBA’ and 4761 posts from ‘PremierLeague’

Pre Processing

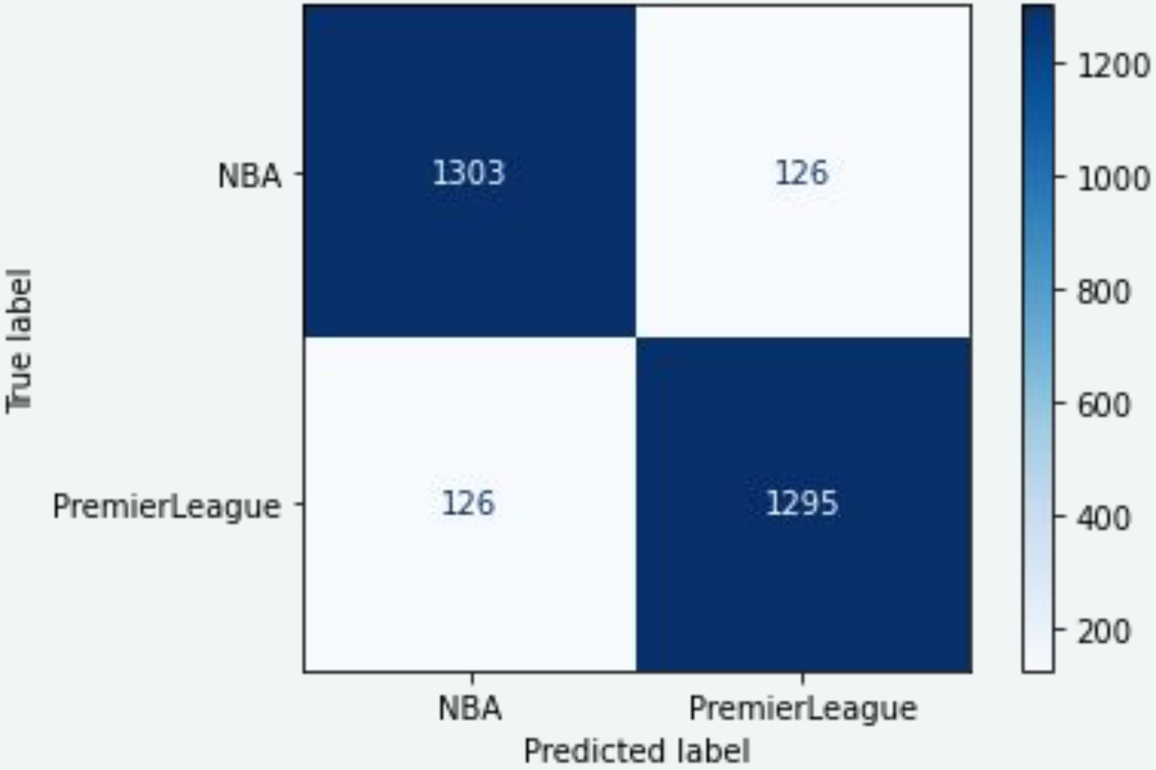
- ❑ Tokenize: Split a string into substrings and remove special characters
- ❑ Lemmatize: Shortening words to combine similar forms of the same word
- ❑ CountVectorizer: Transform a given text into a vector based on the frequency it appears in the whole text and remove stop words.

Modeling

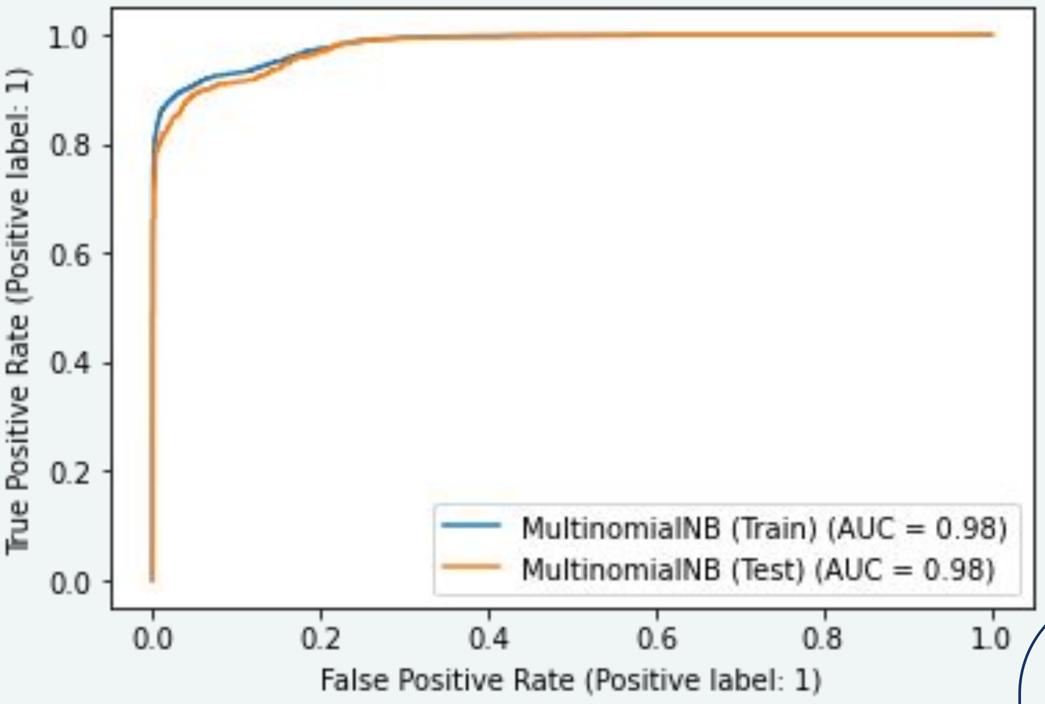
- ❏ Naive Bayes: Train test split (70%/30%), fit and predict
- ❏ Random Forest Classifier: Train test split (70%/30%), Hyperparameter tuning (GridSearchCV), fit and predict



Evaluation (Naive Bayes)



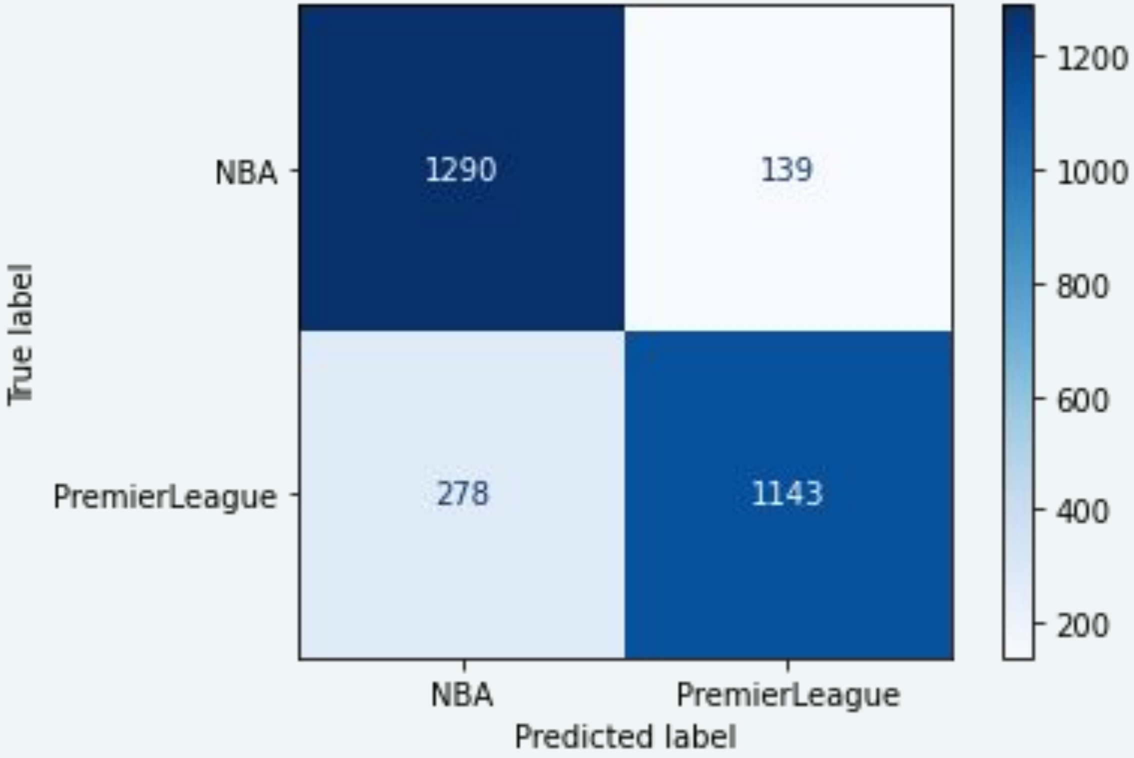
ROC Curves Comparison for Train and Test Data Set (Naive Bayes)

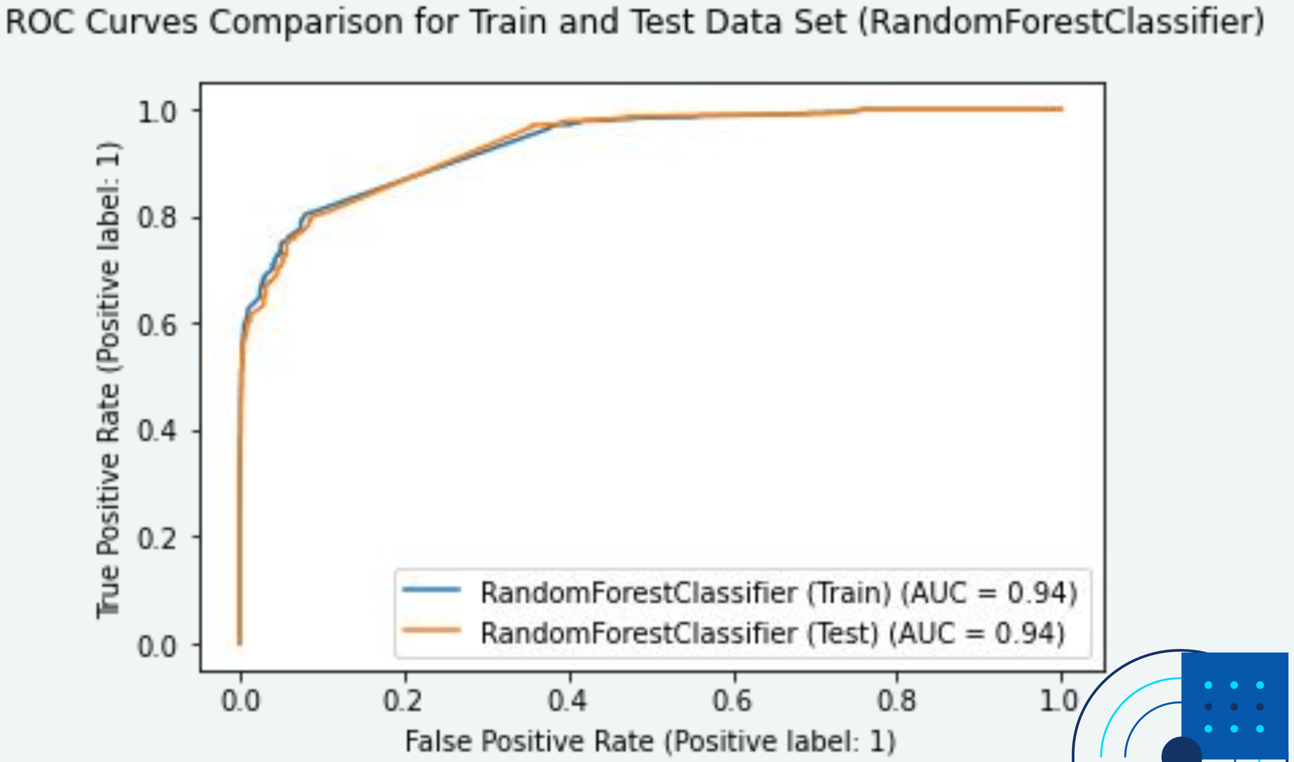


F1 Score: 0.911

Accuracy : 0.912

Evaluation (Random Forest Classifier)





F1 Score: 0.846

Accuracy : 0.854

Based on ROC curves, it's a better fitted model as compared to Naive Bayes

Conclusion

Findings

❑ Naive Bayes has better scores

❑ Random Forest Classifier has a better fit and is a more consistent model

❑ Common words within the top 10 occurring words

Future Work

❑ Try out TF-IDF Vectorizer

❑ Remove the common words which have high frequency in both subreddits

❑ Using other models (eg. Logistics Regression, SVM, etc)

Q&A

Thank You