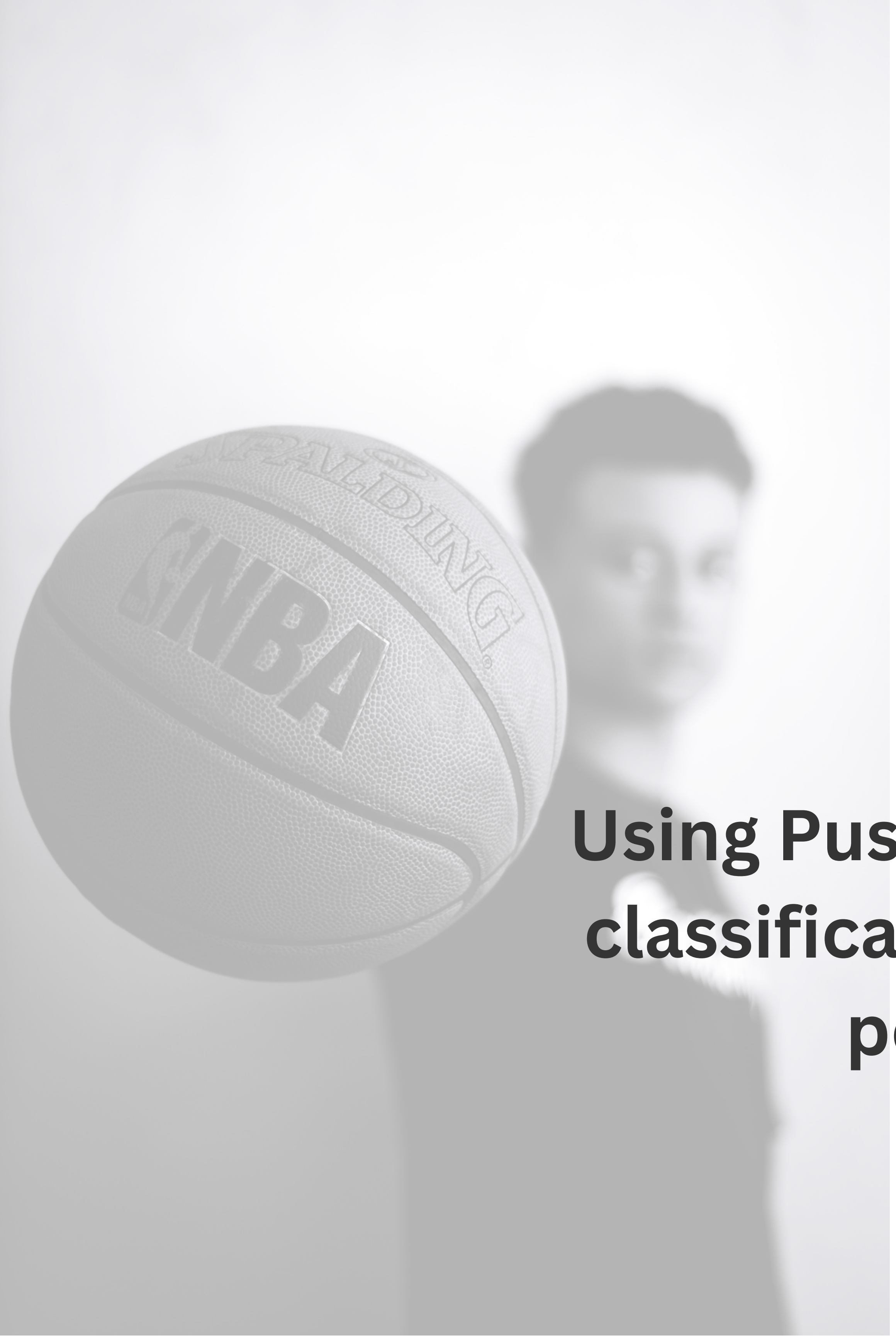
Web APIs & NLP

Prediction on Subreddits

by Ian Stack

Problem Statement



Using Push shift's API, I will gather data and build classification models to predict which subreddit posts came from r/nba and r/nhl

•••

Background

- Professional sports has become one of the biggest entertainment industries
- More platforms are being used to share information across the internet

Subreddit:

r/nba

- The NBA is a professional basketball league played in the USA
- r/nba: A subreddit dedicated to NBA news and discussion.

•••

Subreddit:

r/nhl

- The NHL is a professional hockey league played in the USA

- r/nhl: A subreddit dedicated to NHL news and discussion.

Plan of Action

1. Data Collection

Using Pushshift's API, collect 3,000 posts from r/nba and r/nhl

2. EDA

analyze the data using visual techniques

3. Modeling

Use models: Multinomial Bayes, K-Nearest Neighbor, & Logistic Regression

4. Evaluation

Interpret findings and finalize conclusion

...

Data Collection

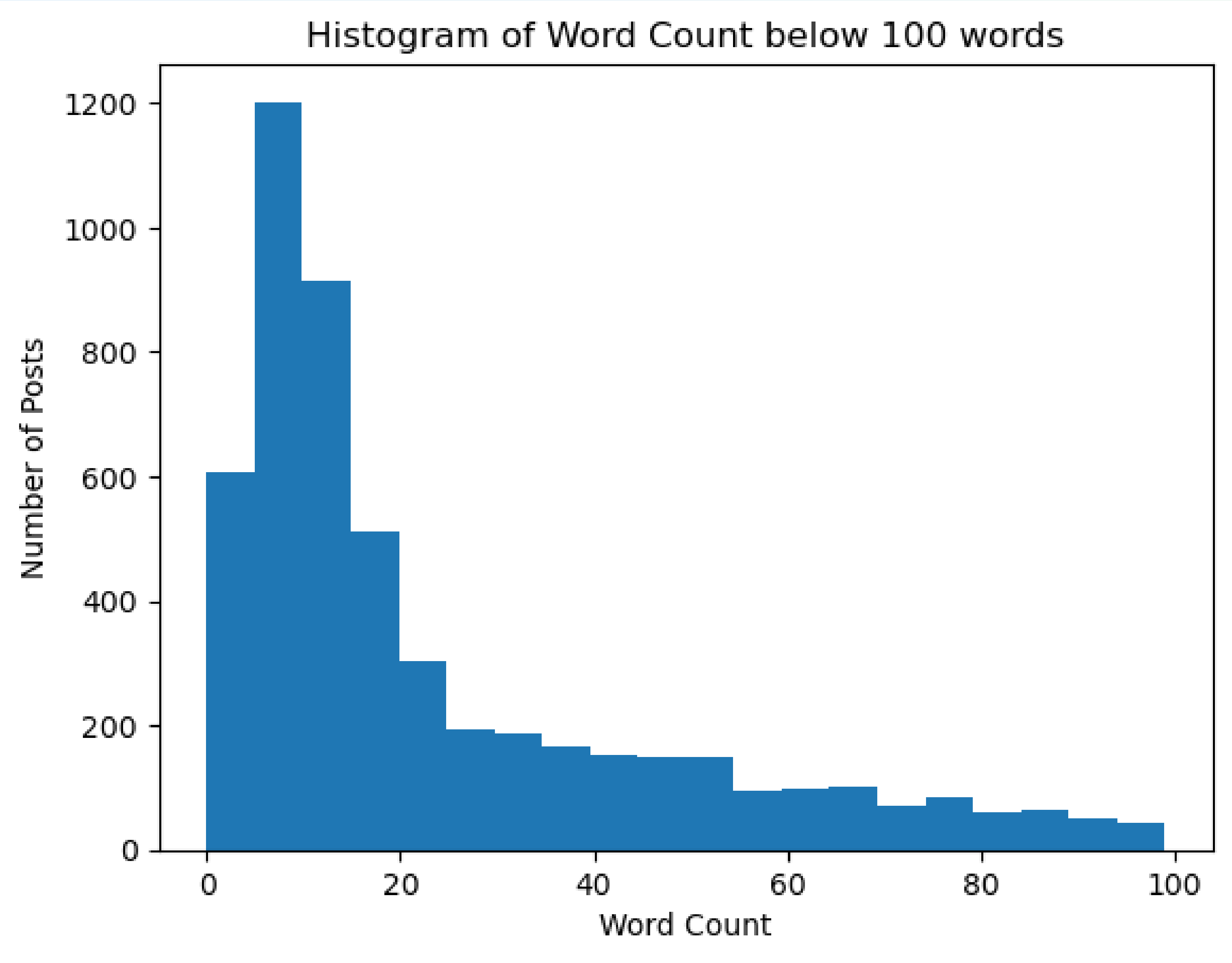
- Data in from of subreddit posts

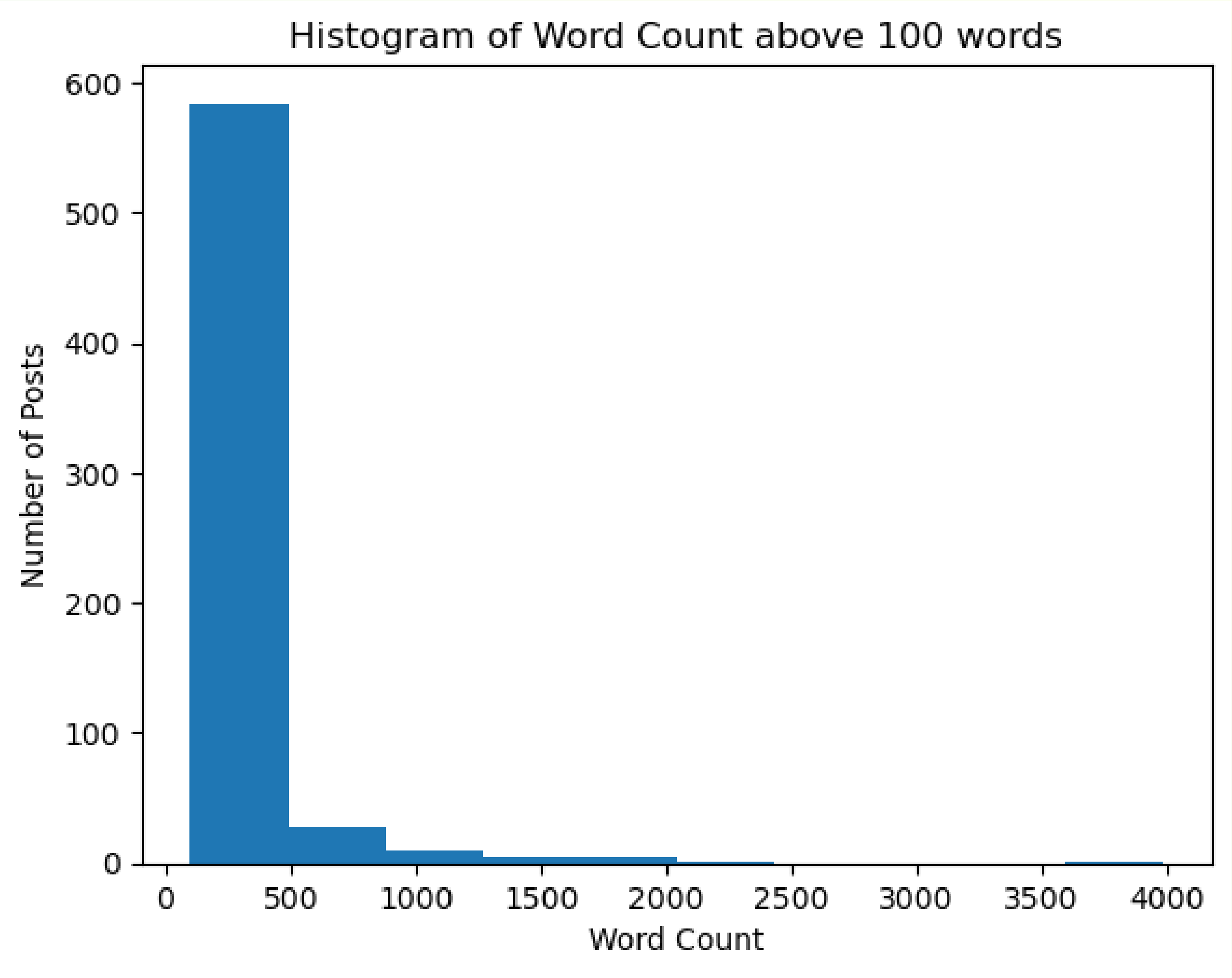
- Cleaned Data

- Only Columns used:
    - 'Selftext'
    - 'Title'
    - 'Subreddit'
        - r/nba: 1
        - r/nhl: 0

- Created New Columns:
    - word count
    - word length
    - tokenized

Histogram Word Count





Word Count per Subreddit

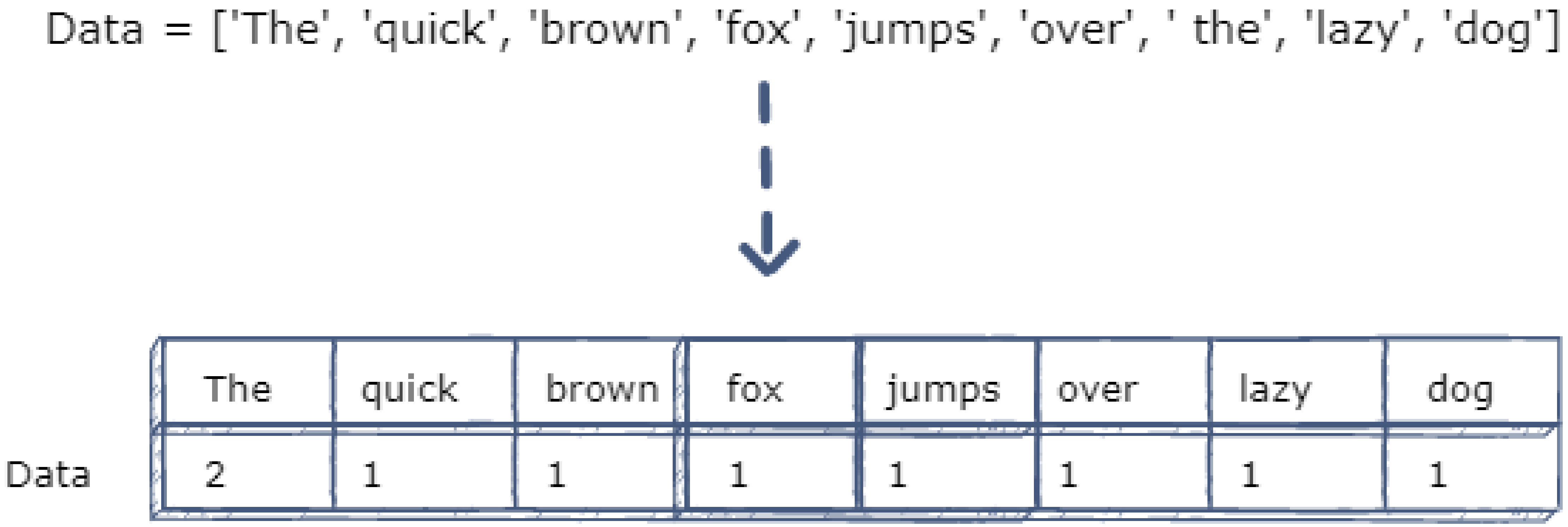
r/nba vs. r/nhl

- r/nba had higher avg.
  - word count
  - character length

Subreddit	Avg. word count	Avg. character count
r/nba	66.08	388.62
r/nhl	28.99	160.65

Count Vectorization: Converts each document into a matrix of words and their counts.

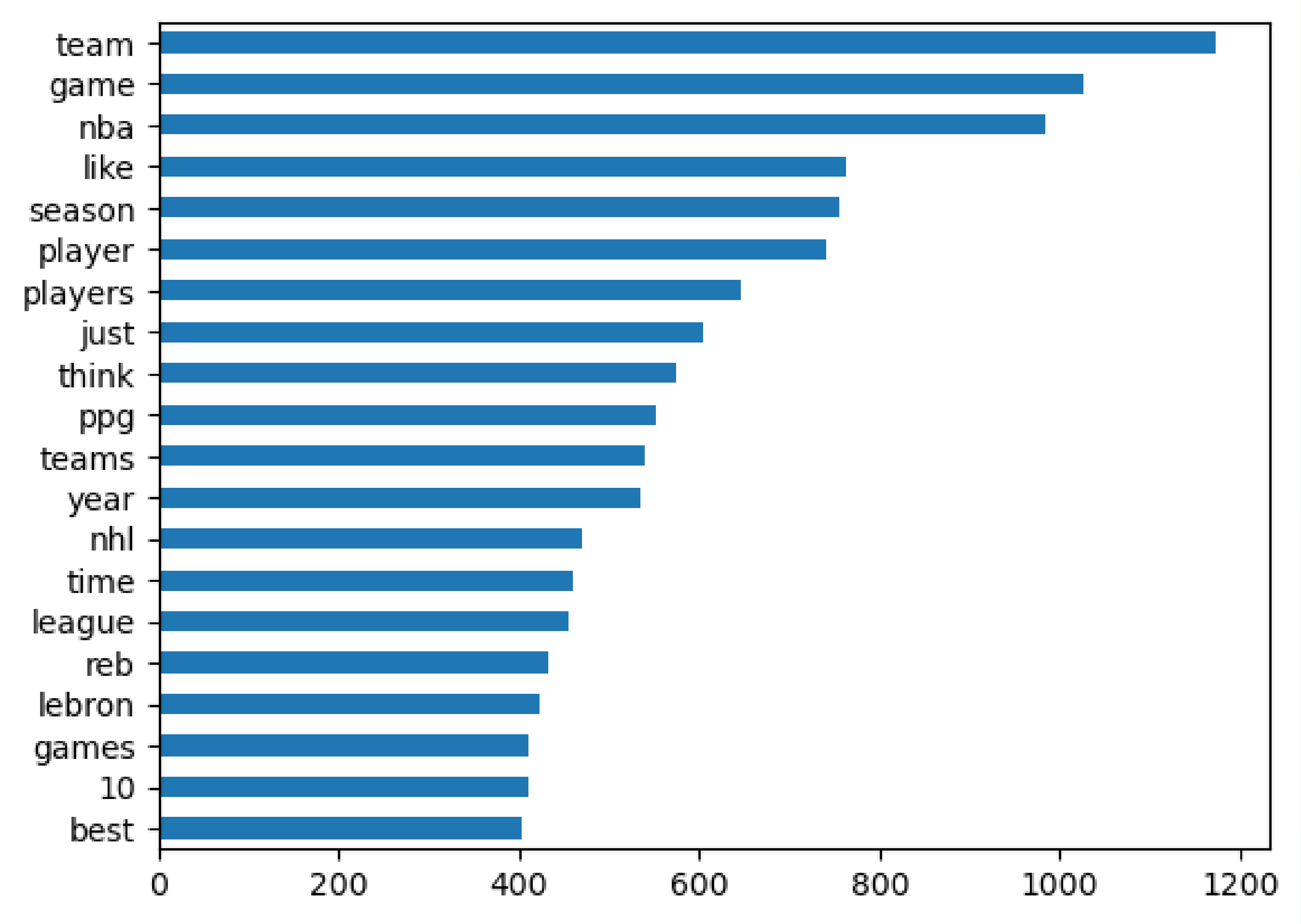
TF-IDF Vectorization: transform the text into a meaningful representation of integers or numbers which is used to fit machine learning algorithm for predictions



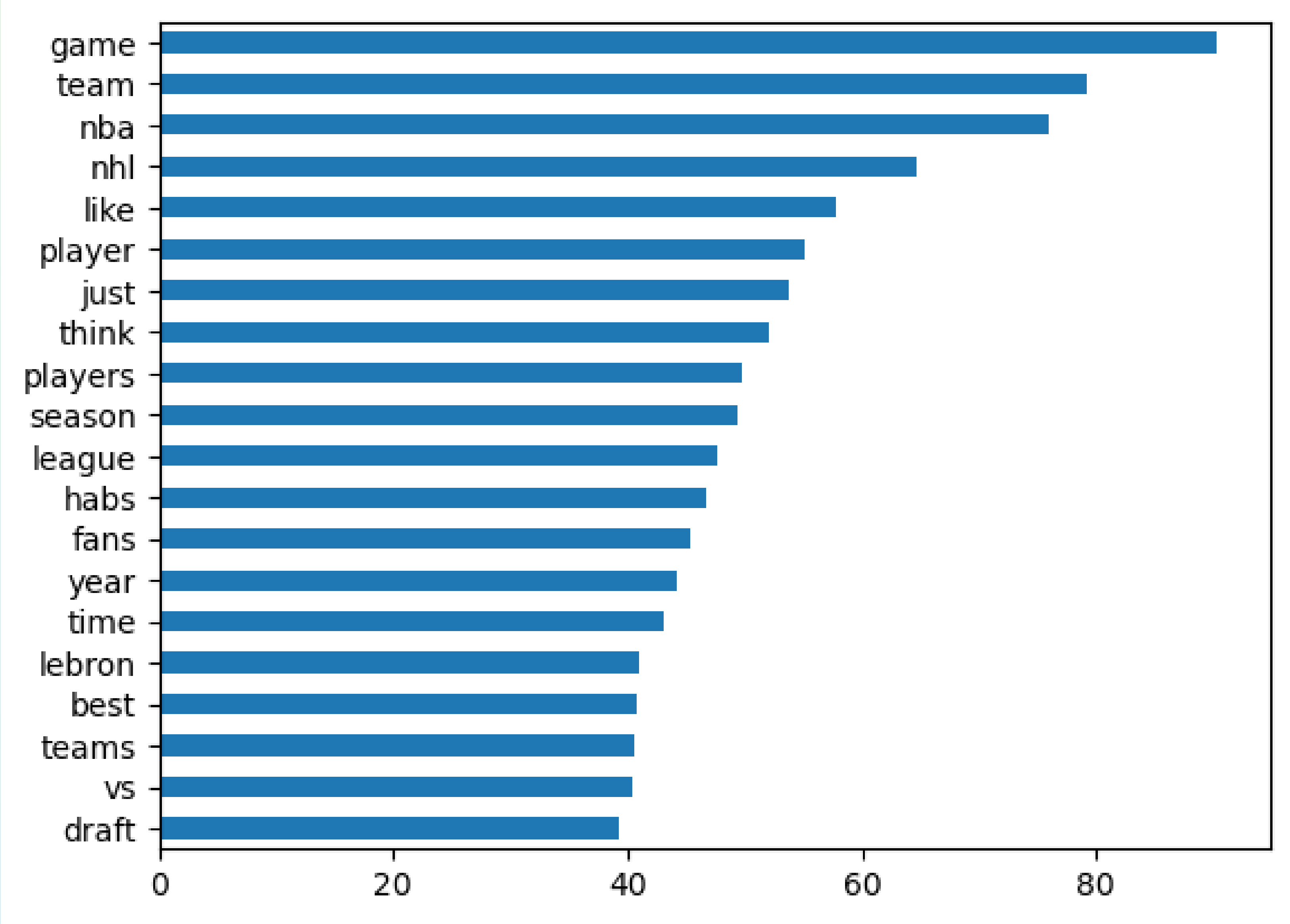
Data = ['The', 'quick', 'brown', 'fox', 'jumps', 'over', ' the', 'lazy', 'dog']

	The	quick	brown	fox	jumps	over	lazy	dog
Data	2	1	1	1	1	1	1	1

Top 20 Word Count

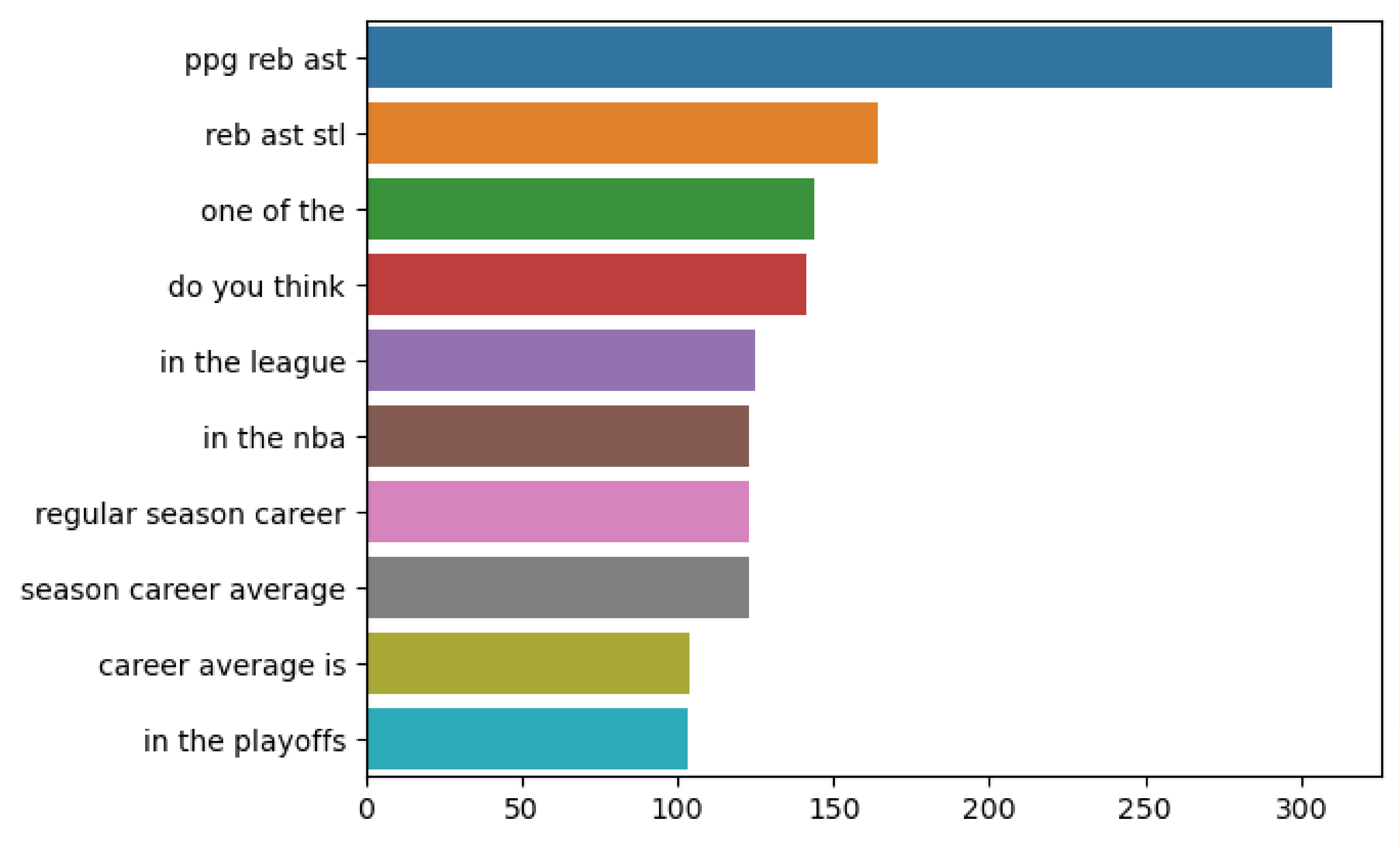


CountVectorizer



TF-IDF Vectorizer

N-Gram: 3 Words



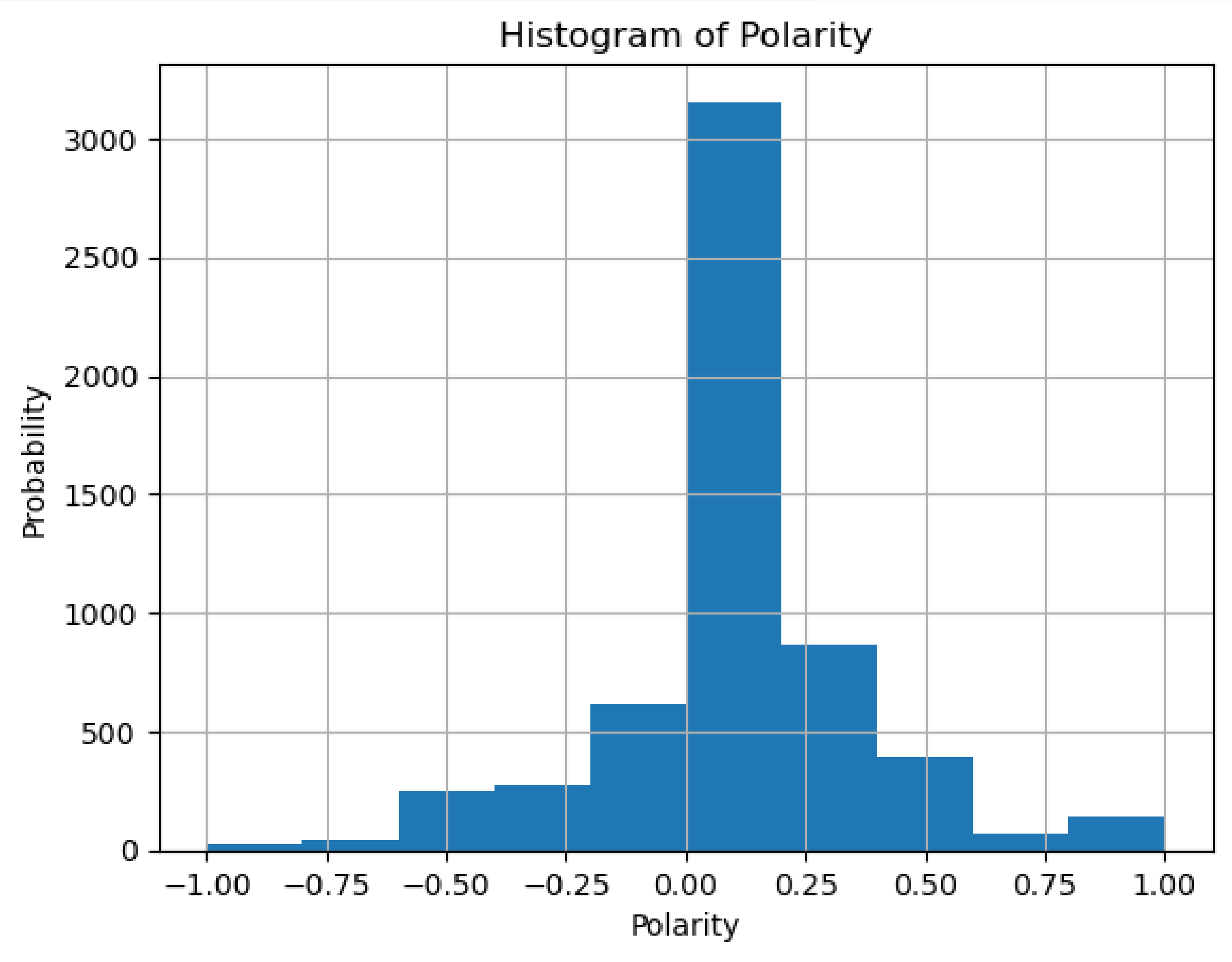
N-Gram: order of N-words

N = Number of Words

Sentiment Analyzer

Lexicon-based sentiment analyzer

analyzing data and classifying it based on if it is positive or negative.



- Sensitivity: For those who posted on r/nba, how many did I get correct

- Specificity: For those who did NOT post on r/nba(posted on r/nhl) how many did I get correct

3. Modeling

Goal: Using Train-Test-Split, find best performing model

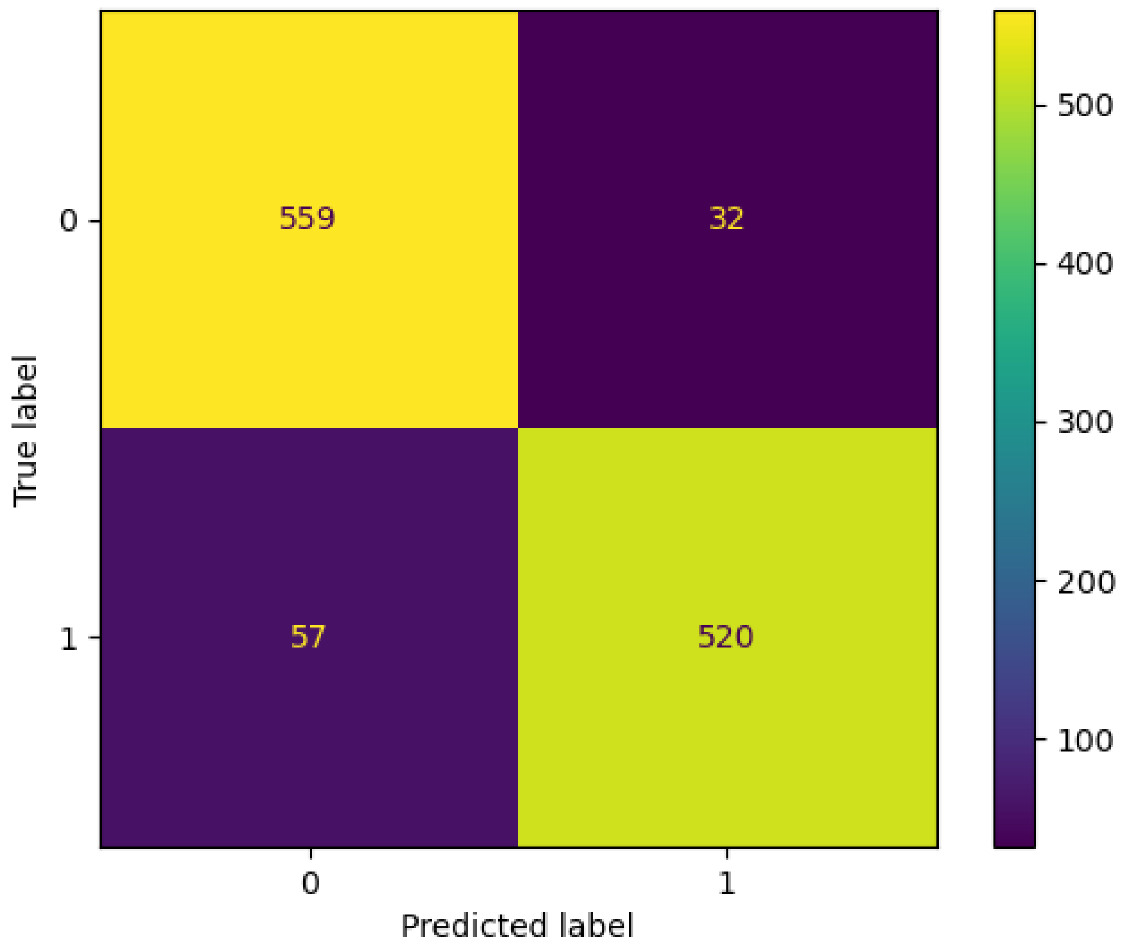
Multinomial Naive Bayes

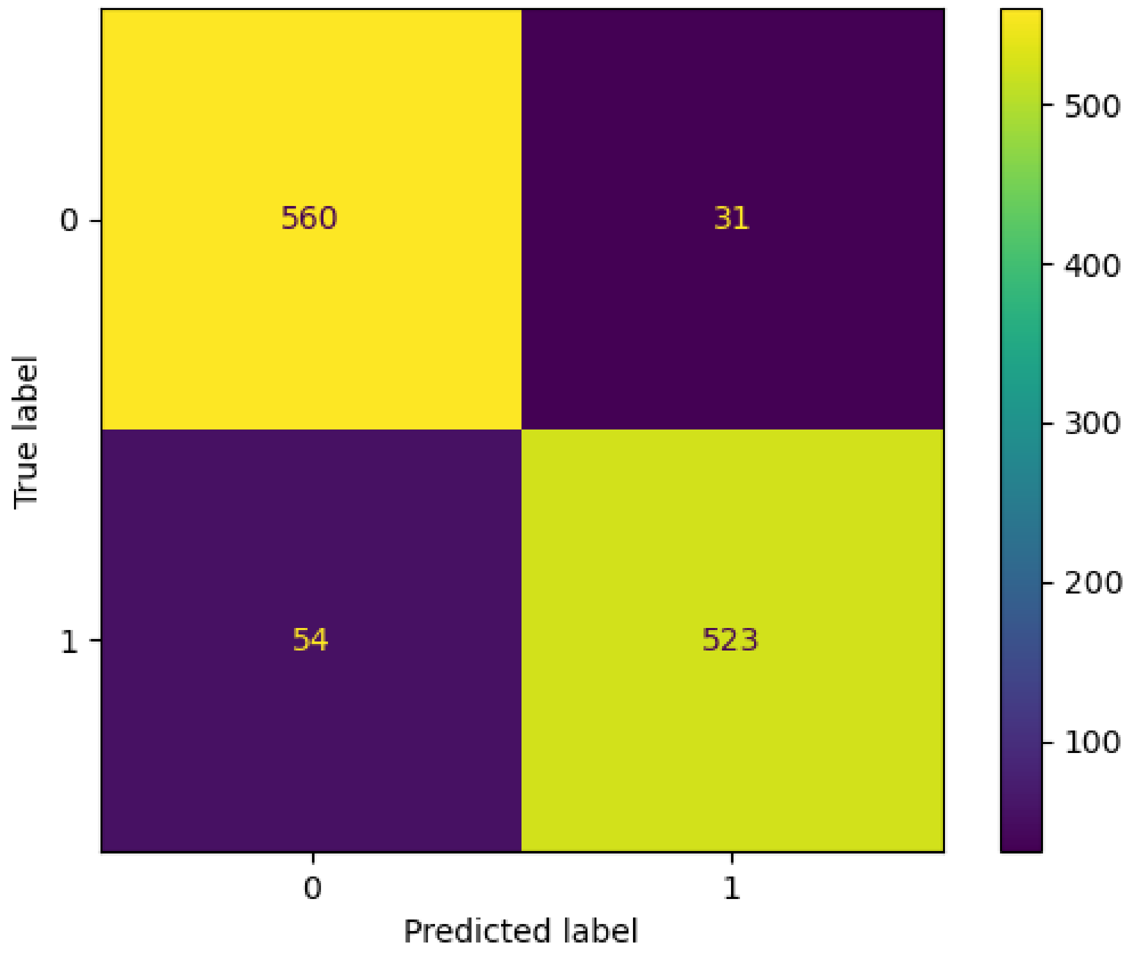
- Count Vectorizer:
  - Train RMSE: 0.946
  - Test RMSE: 0.923

  - Sensitivity: 0.901
  - Specificity: 0.945

- IF-IDF Vectorizer:
  - Train RMSE:0.956
  - Test RMSE: 0.923

  - Sensitivity: 0.906
  - Specificity:0.947





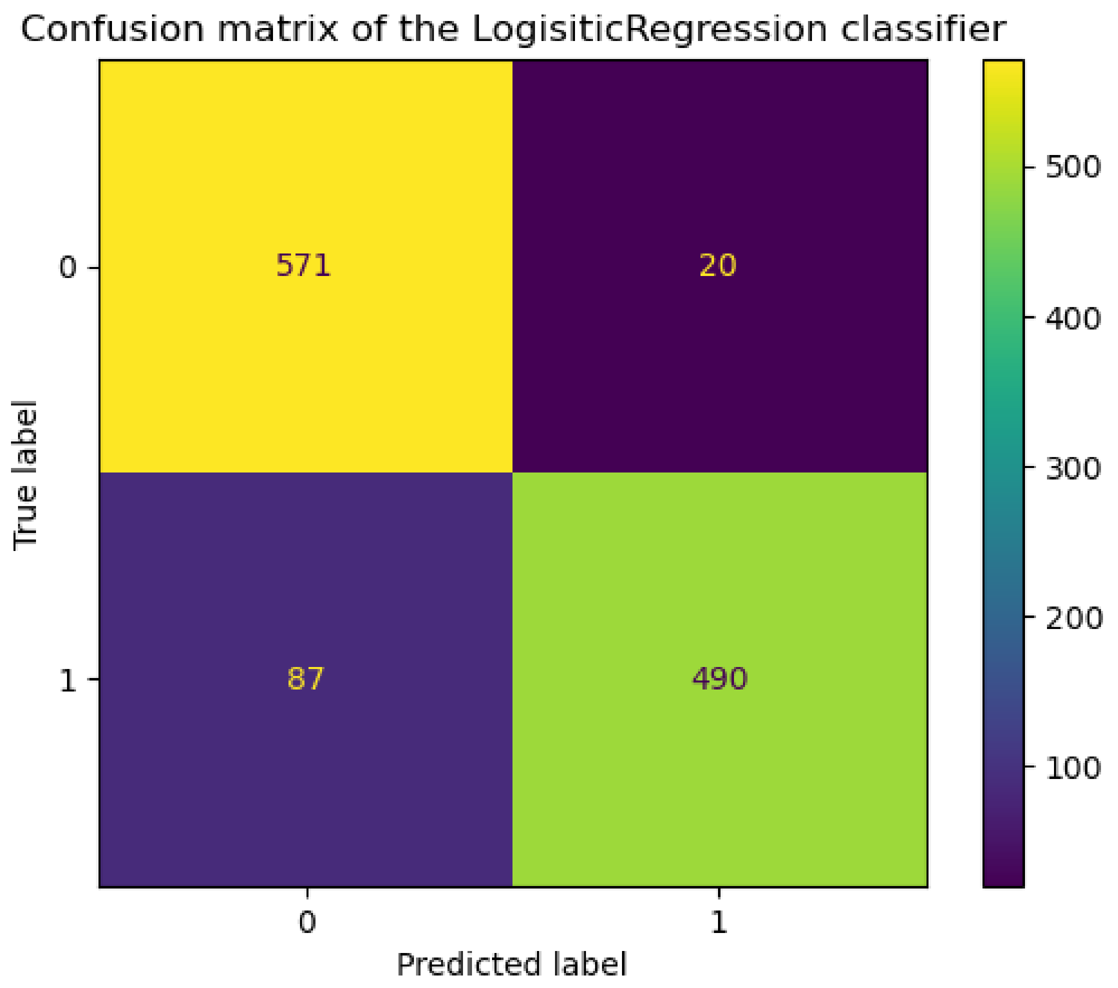
Logistic Regression

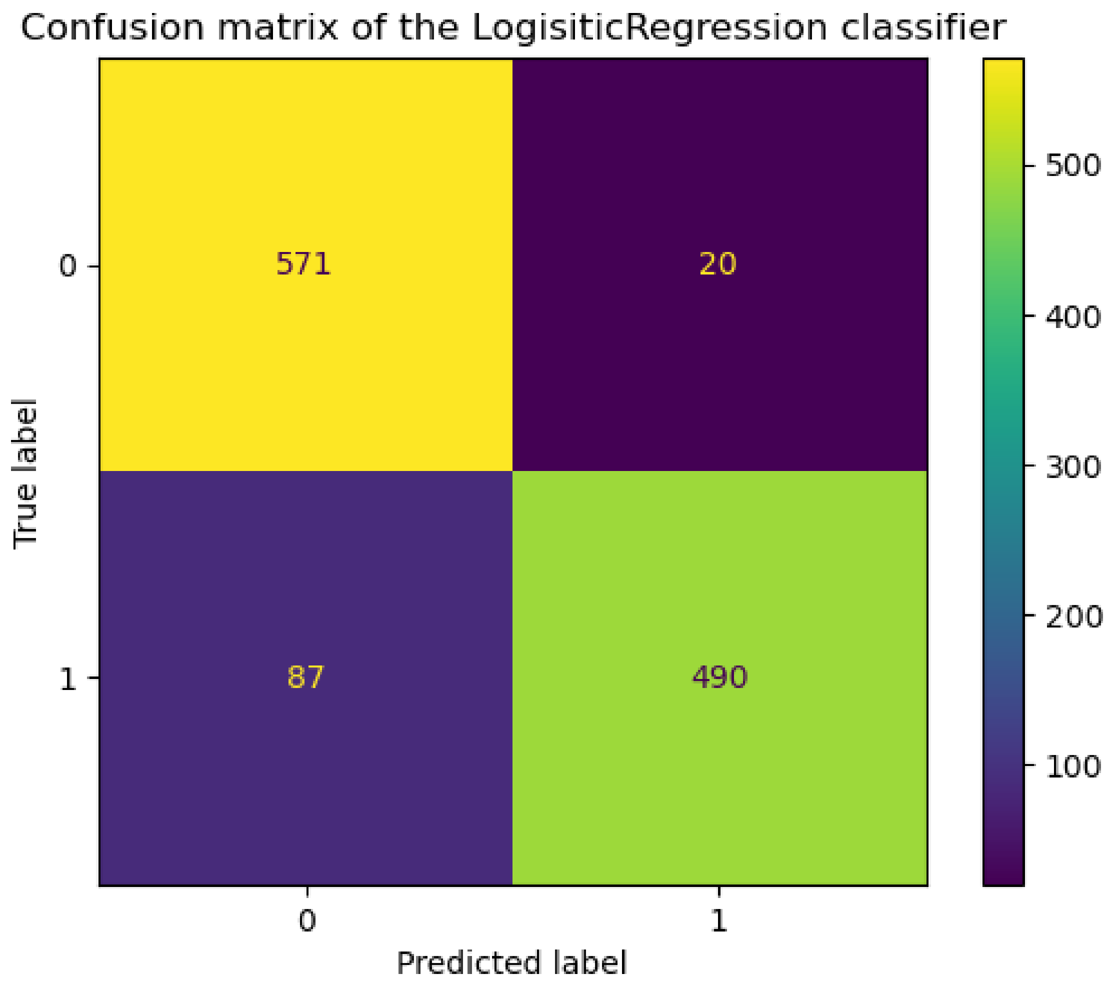
- Count Vectorizer:
  - Train RMSE: 0.929
  - Test RMSE: 0.908

  - Sensitivity: 0.849
  - Specificity: 0.966

- IF-IDF Vectorizer:
  - Train RMSE:0.929
  - Test RMSE: 0.908

  - Sensitivity: 0.849
  - Specificity:0.966





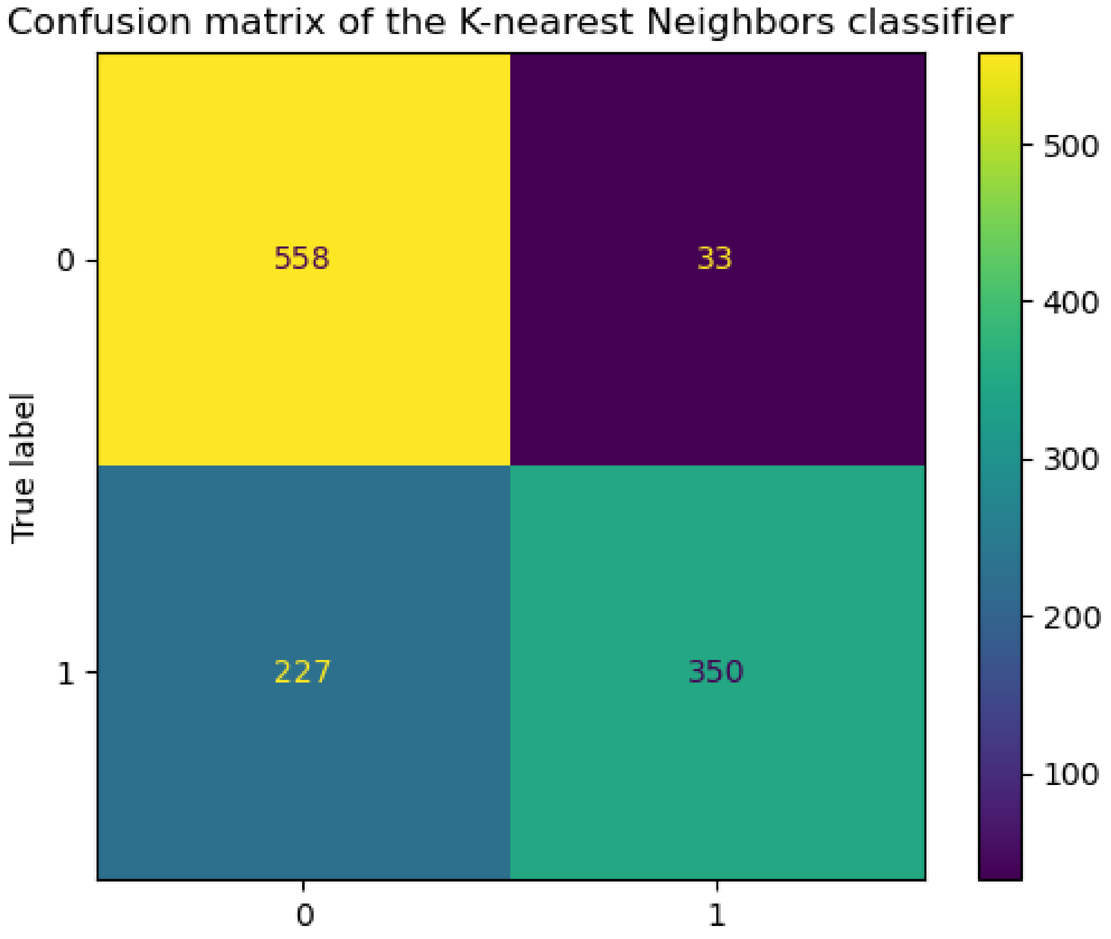
K-Nearest Neighbor

- Count Vectorizer:
  - Train RMSE: 0.844
  - Test RMSE: 0.777

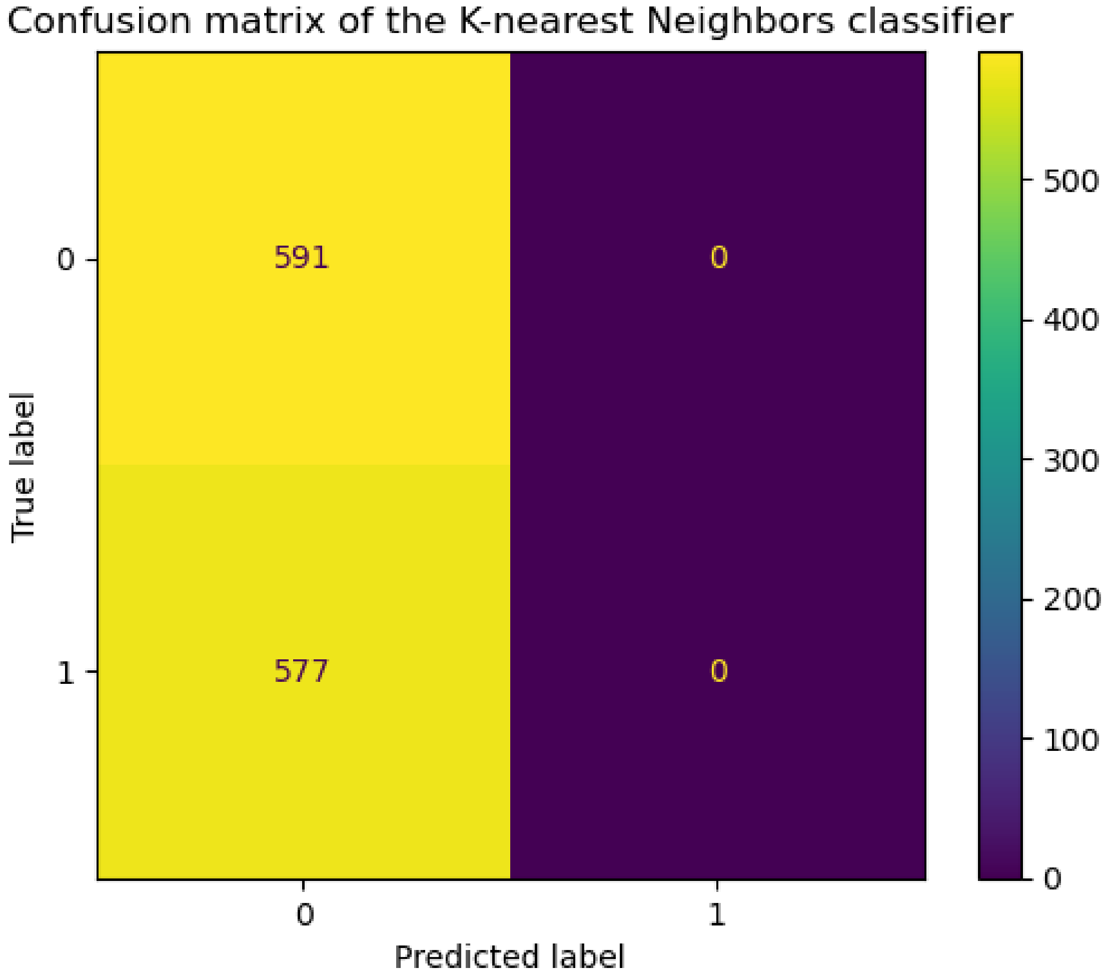
  - Sensitivity: 0.606
  - Specificity: 0.944

- IF-IDF Vectorizer:
  - Train RMSE:0.505
  - Test RMSE: 0.505

  - Sensitivity: 1.0
  - Specificity:0.0



Confusion matrix of the K-nearest Neighbors classifier



Confusion matrix of the K-nearest Neighbors classifier

Predictions vs. Actual

(MNB model)

[18]: '[Overtime/Twitter] Kevon Looney getting cooked at his own camp'

-> Thought it was r/nhl but it was not

[4010]: '[Sean Shapiro on Twitter] Since everyone asks, from what I've heard ESPN didn't seriously consider bringing back Gary Thorne.

-> thought it was r/nba but it was not

Conclusion & Recommendations

Overall:

User's posts are being posted on correct subreddit.

Models Overfit

Would adjust parameters & stop words(names, terminology)

Multinomial Naive Bayes #1

Best Model with best accuracy and specificity

Introduce more columns

Analyze more from subreddit posts: comments

Thank you!

Let us know if you have questions or clarifications.

Sources

https://www.projectpro.io/recipes/use-tf-df-vectorizer#:~:text=TF%2DIDF%20will%20transform%20the,documents%20the%20word%20appears%20in.

https://www.google.com/imgres?imgurl=https%3A%2F%2Fwww.educative.io%2Fapi%2Fedpresso%2Fshot%2F5197621598617600%2Fimage%2F6596233398321152&imgrefurl=https%3A%2F%2Fwww.educative.io%2Fanswers%2Fcountvectorizer-in-python&tbnid=AYGlEZbt6k_ZMM&vet=12ahUKEwi4tv6OvIn8AhXhB0QIHc0iAowQMygCegQIARBp..i&docid=JwXI4_tIS6teBM&w=565&h=205&q=graphic%20explaining%20count%20vectorizer&ved=2ahUKEwi4tv6OvIn8AhXhB0QIHc0iAowQMygCegQIARBp