NLP & Classification Modeling

Christiaan Dageforde

Problem Statement

- Subreddit fields have been replaced with null values
- Develop models to distinguish between subreddits

Data & Method

- Using Reddit's API, pulled posts from two different Subreddits
  - r/Batman
  - r/Joker
- Data
  - 4,230 Posts
    - 2,150 from r/Joker
    - 2,080 from r/Batman
  - Comments/"selftext"
    - Too many non-text or null posts to use in modeling
  - Titles
    - All posts had titles, and so were used for my text data

Data & Method, Cont.

- Three classification models considered
    - Multinomial Naive Bayes
    - Random Forest Classifier
    - Logistic Regression
  - Performed Gridsearch operations to find the optimal parameters to tune each model
  - Selected the two most accurate models for comparison
- Natural Language Processing Methods
  - Count Vectorizer
  - TF-IDF

Natural Language Processing

NLP

- Methods to contextualize human language in a way that computers can understand
- A variety of feature extraction techniques were used in the process of training our models

NLP Methods

- TF-IDF (Term Frequency - Inverse Document Frequency)
    - Transforms unstructured text into a matrix of term frequencies by document frequency
  - TF
    - How many times does a given term appear in a document?
  - IDF
    - How many documents in our corpus include this term?
- Tells us which words are most unique to each document
  - Valuable in distinguishing between documents

NLP Methods

- CountVectorizer
  - Transforms unstructured text into a matrix of term frequencies
- Stop Words
  - Commonly-used words that contain little information that would help to identify a given document
    - “Noise”; can cause difficulty during the modeling process
- N-Grams
  - Breaks phrases into smaller segments in order to better contextualize words

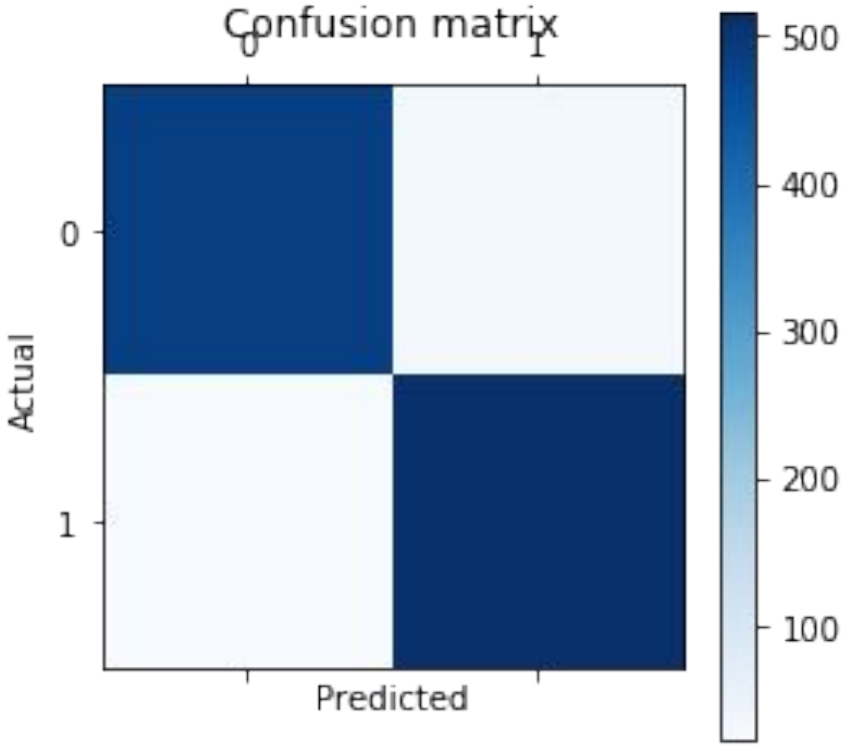
Models

Model ‘A’

- Random Forest Classifier
  - Few parameters to tweak
  - Commonly high-performing
  - Number of estimators: 15
    - Number of trees in forest
  - CountVectorizer
    - Stop Words: None
    - Max. Features: 4,000
    - N-Gram Range: 1,1

Scoring

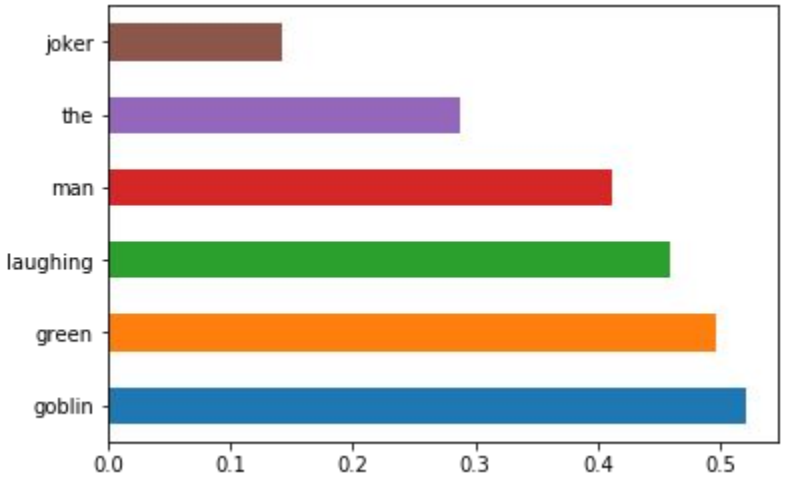
- 1,058 Predictions
- Correct Predictions
  - True Positive: 516
  - True Negative: 487
- Incorrect Predictions
  - False Positive: 22
  - False Negative: 33
- Scoring Metrics
  - Accuracy: 94.8%
  - Sensitivity: 95.7%
  - Specificity: 94%
  - Misclassification: 5.2%



Model ‘B’

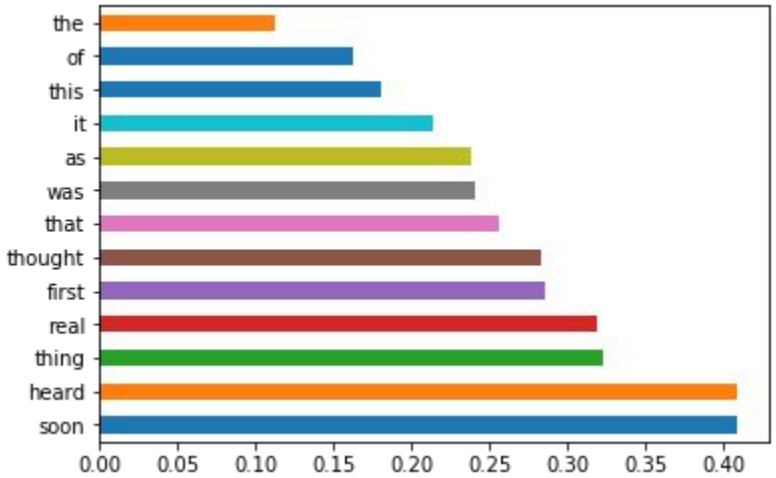
- Logistic Regression
  - Continuous predictions to classify between ‘1’ and ‘0’
    - C: 10
  - TF-IDF
    - Analyzer: Word
    - N-Gram Range: 1,2
    - Stop Words: None

Term Frequency



Positive Class

- ‘1’
- r/batman

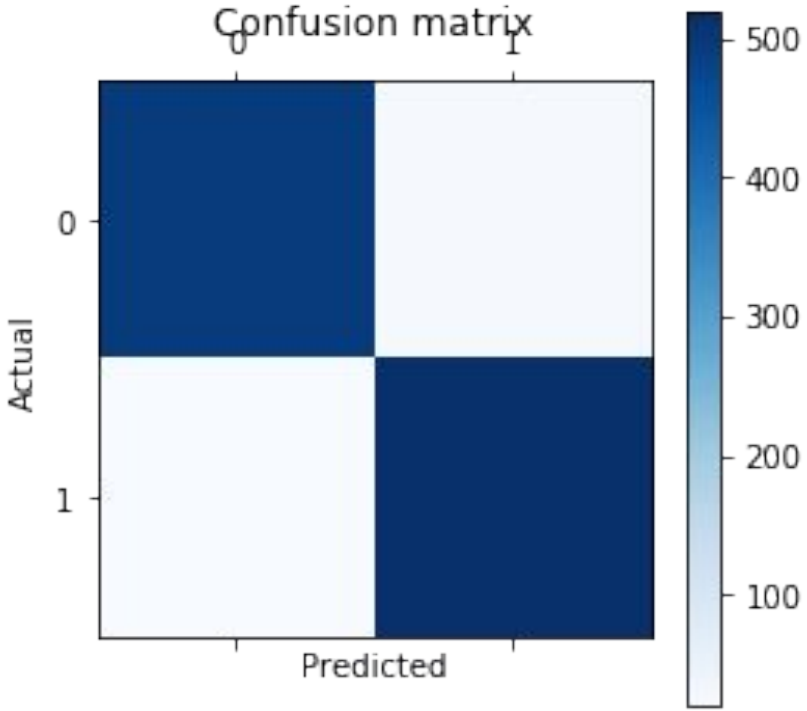


Negative Class

- ‘0’
- r/joker

Scoring

- 1,058 Predictions
- Correct Predictions
  - True Positive: 519
  - True Negative: 496
- Incorrect Predictions
  - False Positive: 19
  - False Negative: 24
- Scoring Metrics
  - Accuracy: 95.9%
  - Sensitivity: 96.3%
  - Specificity: 95.6%
  - Misclassification: 4.1%



Conclusions

...and the winner is...

- Model ‘B’
  - Highly accurate
  - Low misclassification rate