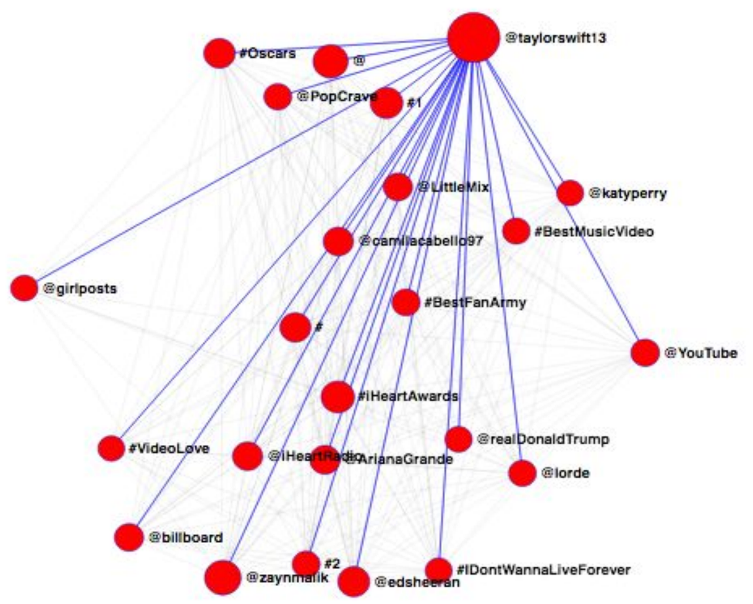
Data Mining Twitter for Sentiment Analysis and Hate Speech Detection

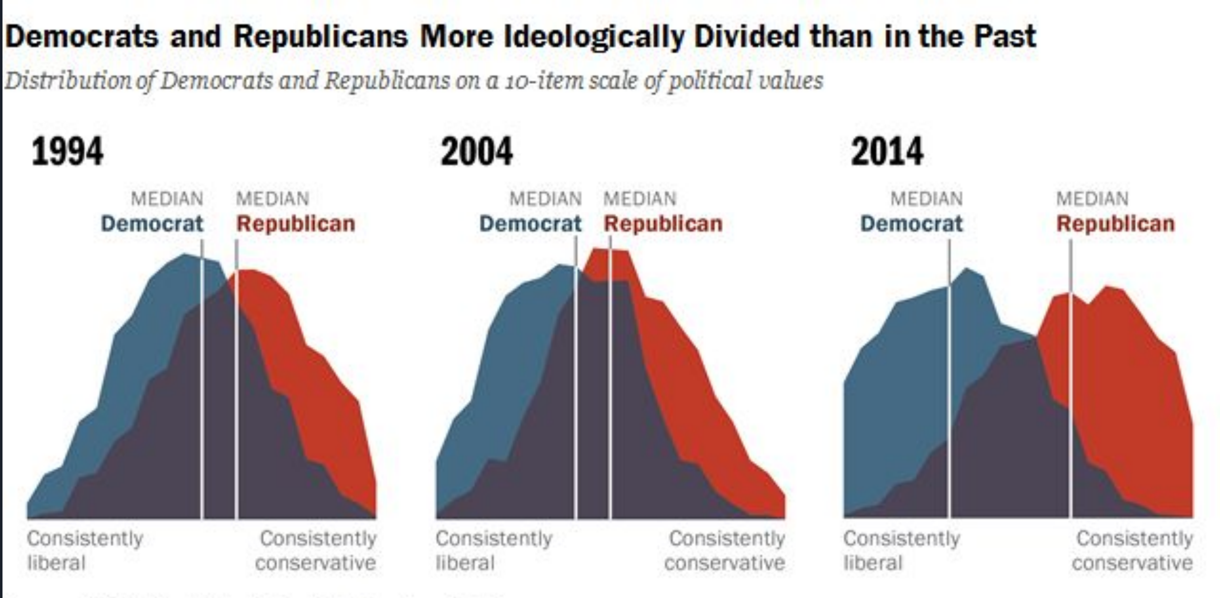
Arshjit Pelia 0374870
Nikholas Pcenicni 0389088



TWITTER BACKGROUND

- Micro-blogging Interface
- The majority of interactions occur amongst unfamiliar parties
- Significant factor in political polarization
- Establishes user networks across similar interests

TWITTER NETWORK



Network	Clust.	Left	Right	Undec.	Nodes
Retweet	A	1.19%	93.4%	5.36%	7,115
	B	80.1%	8.71%	11.1%	11,355
Mention	A	39.5%	52.2%	8.18%	7,021
	B	9.52%	85.7%	4.76%	154

- Political Discourse takes place across 2 network topologies:
  - Mentions
  - Retweets
- Retweet networks consist of more cross-ideological engagement
  - Also cultivate more polarization
- Polarized groups also use positive or negative hashtags rather than neutral

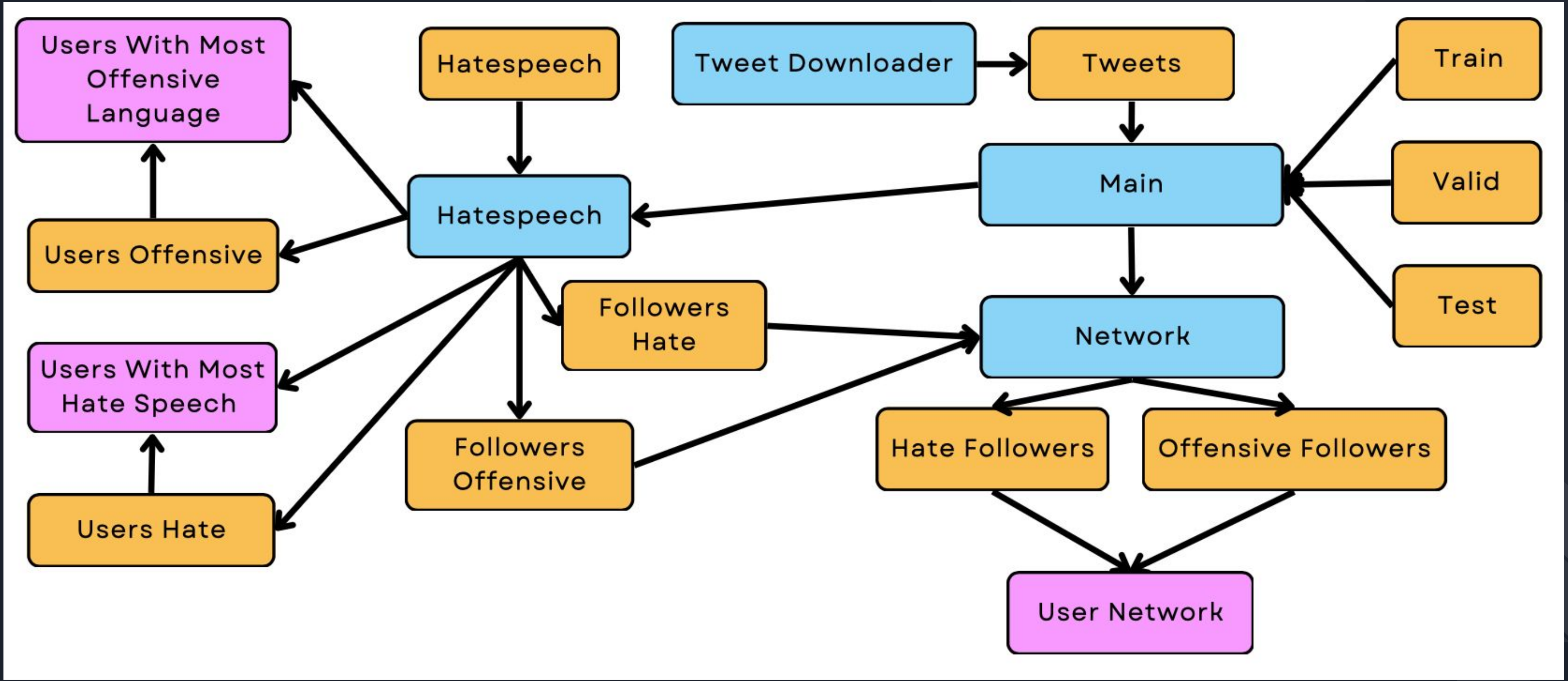
Objectives

- Perform sentiment analysis on tweets related to the 2022 World Cup
- Train a classification model to automate labelling
- Implement hate speech Detection on negative tweets
- Construct user network frequently associated with hateful or offensive subjects

Contributions

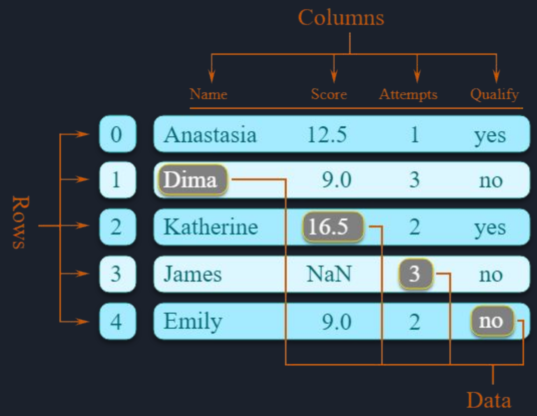
- Proposal of an efficient classification model to automate annotation of hateful or offensive speech sentiment
- Efficacy comparison across Naive-Bayes and Support Vector Machine
- Extension of distinguished research findings related to opinion-mining tweets.
- Introduce a method to compile dangerous user networks

Methodology Overview



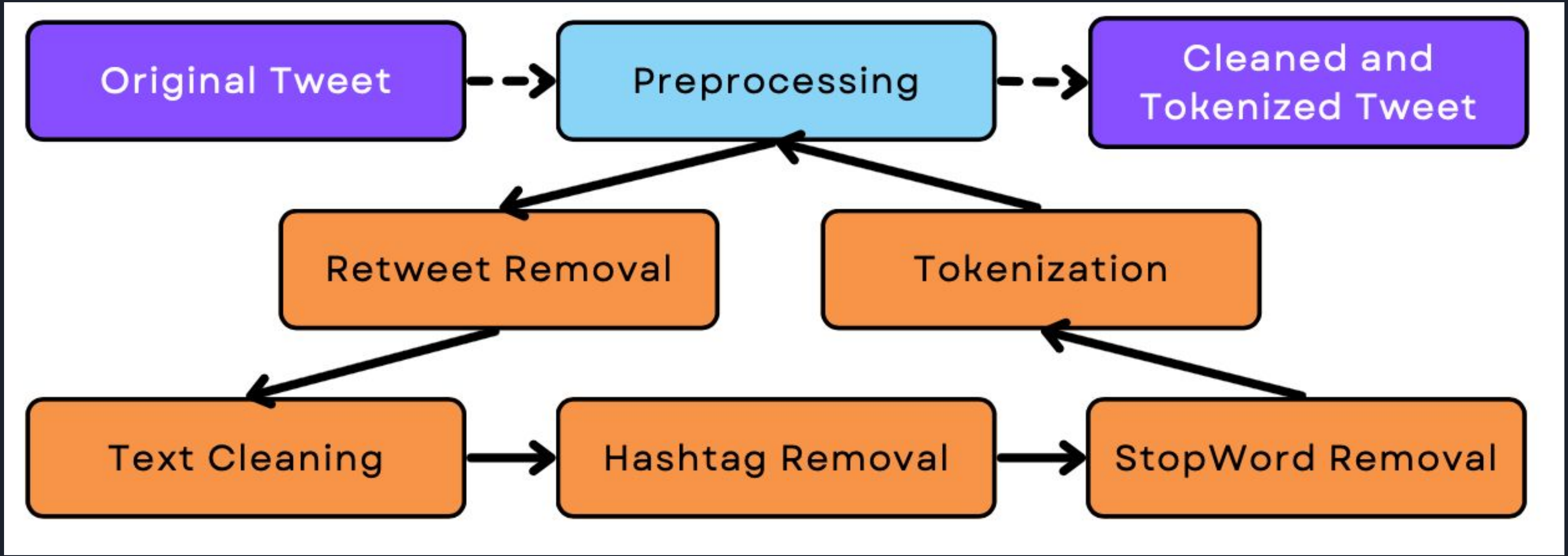
TWEET CORPUS

- Employed Tweepy module and an authenticated developer account to connect and explore metadata within the Twitter API
- Search Parameter for scraping: keywords and hashtags matching “World Cup”
- Collected Over 60,000 tweets to be used for training and analysis
- Datasets were stored in a csv file and operated on using Dataframes from panda Library



Sample of Scraped Dataset

timestamp	tweet_OG	username	all_hashtags	location	followers_count	retweet_count	favorite_cou
2022-12-03 01:07:18+00:00	@sawed_sajid Here, I will support Argentina to win the World Cup.	Farah99165380	[]		0	0	
2022-12-03 01:07:18+00:00	RT @brfootball: The World Cup bracket is set ⍰üèÜ https://t.co/8GRBmCffrl	KSmorls	[]		100	5392	
2022-12-03 01:07:18+00:00	RT @NAACP: Tonight, FoxCorp, the owner of FOX News, has been trying to pad its deep pockets by bullying @DirecTV into p	dbactnow	[]	Sarasota springs	1803	7	
2022-12-03 01:07:18+00:00	RT @johncrossmirror: Match report: Cameroon 1 Brazil 0. Cameroon captain Vincent Aboubakar scored the winner, got sent	KafelGurja	[]		229	16	
2022-12-03 01:07:18+00:00	Bro my barber is going to the World Cup semis and finals ⍰üò≠⍰üò≠	Ronn_Knee	[]	Texas, USA	399	0	
2022-12-03 01:07:18+00:00	RT @FlyerTalkerinA2: NEW TBB Blog Post - #StockMarket Bubbles, #FTX Collapse, Clapping for #SBF, #WorldCup Total #Foc	feeonlyplanner	[{'text': 'StockMark	Ann Arbor, Michigan	10560	1	
2022-12-03 01:07:17+00:00	RT @StokeyyG2: Qatar really cooked up one of the best World Cup‚Äôs ever ⍰üíÄ⍰üíÄ	Grimissad	[]	Nigeria	41	11119	
2022-12-03 01:07:17+00:00	RT @iam_wilsons: This World Cup has been nothing short of exciting. Congratulations to all the Manchester United players	kvnq_sammie	[]	On the streets	2730	17	
2022-12-03 01:07:17+00:00	RT @jeonjkloops: me hearing dreamers (a world cup song) being played at world cup https://t.co/r2v7ty6obz	deadness_7	[]	⍰üå¨⍰üå∏she/her	47	2426	
2022-12-03 01:07:17+00:00	RT @jeonjkloops: me hearing dreamers (a world cup song) being played at world cup https://t.co/r2v7ty6obz	Jhxxpjk	[]	Princes Of Global Pop	624	2426	



PRE-PROCESSING SEQUENCE

Preprocessing Tweet Text

1. Begin by removing redundant information from each tweet
    a. URLs, Emojis, punctuation, stopwords are all inconsequential for NLP processing
2. Embed Text using BertTokenizerFast
    a. Assign numerical value to each word for indexed representation
3. After removing empty tweets, perform Lemmatization and Stemming
    a. Lemmatize: Evaluate the morphological context of words
    b. Stem: Account for prefixes and suffixes in an inflected word
4. Shuffle the dataset and reset the index
5. Proceed to Sentiment Analysis

Sentiment Analysis with VADER

```
def get_sentiment(tweet):
    #get sentiment using vaderSentiment and classify tweets as extre
    sid = SentimentIntensityAnalyzer()
    ss = sid.polarity_scores(tweet)
    if ss['compound'] >= 0.05:
        return 'Extremely Positive'
    elif ss['compound'] > -0.05 and ss['compound'] < 0.05:
        return 'Neutral'
    else:
        return 'Extremely Negative'
```

- Evaluate each tweet according to a negative or positive Polarity
- VADER: Valence Aware Dictionary for Sentiment Reasoning
    - Trained model stemming from the NLTK library
- Map lexical features to emotional intensities based on library
- Distributes polarity score between 0 and 1.0
    - Identifies how negative or positive a text is by summing up the intensity of each word

Data Preparation

1. Balance the corpus on the target variable (Sentiment) with oversampling
    a. Randomly sample the training dataset to remove class bias (majority vs minority)
    b. Initialize independent (text) and dependant (sentiment) training variables
2. Split the balanced corpus into training and testing sets
    a. 90% of the tweets used to train Model, 10% reserved for testing model
3. Perform One Hot Encoding
    a. Assign Integer to each unique nominal value
    b. Transform categorical entities to a numerical distribution

```
red,     green,   blue
1, 0, 0
0, 1, 0
0, 0, 1
```

4. Store data frame objects in CSV files for model implementation

MODEL EVALUATION CRITERIA

Accuracy: fraction of correctly classified documents in relation to the total number of documents is called accuracy

- Primary performance measure

Precision: Indicates the distribution of correct predictions

Recall: number of positive class predictions made out of all positive examples in the dataset.

F-Measure: Harmonic mean between Precision and Recall as a single value

$$Accuracy = \frac{TP + TN}{TP + FP + TN + FN}$$

$$Precision = \frac{TP}{TP + FP}$$

$$Recall = \frac{TP}{TP + FN}$$

$$F - Measure = \frac{2 \times Precision Recall}{Precision + Recal}$$

TP	TRUE POSITIVE
TN	TRUE NEGATIVE
FP	FALSE POSITIVE
FN	FALSE NEGATIVE

Naive Bayes Classifier: Implementation

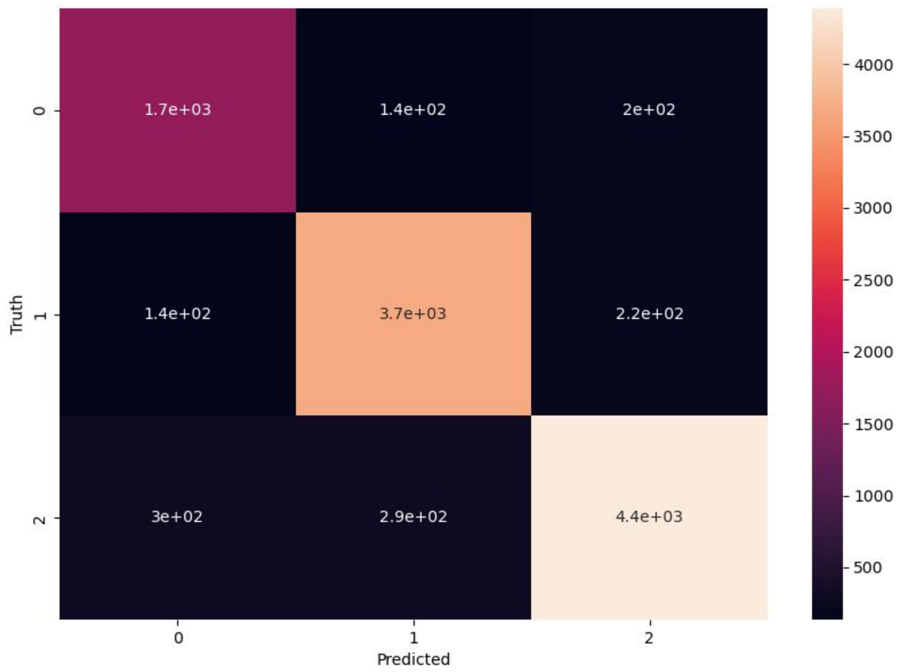
- Train Naive Bayes classifier using pre classified data set
- Analyze sentiment of test data from dataset
- Determine performance and accuracy of classification
- Plot relations
- Graph word bubbles of words with positive and negative sentiment in tweets

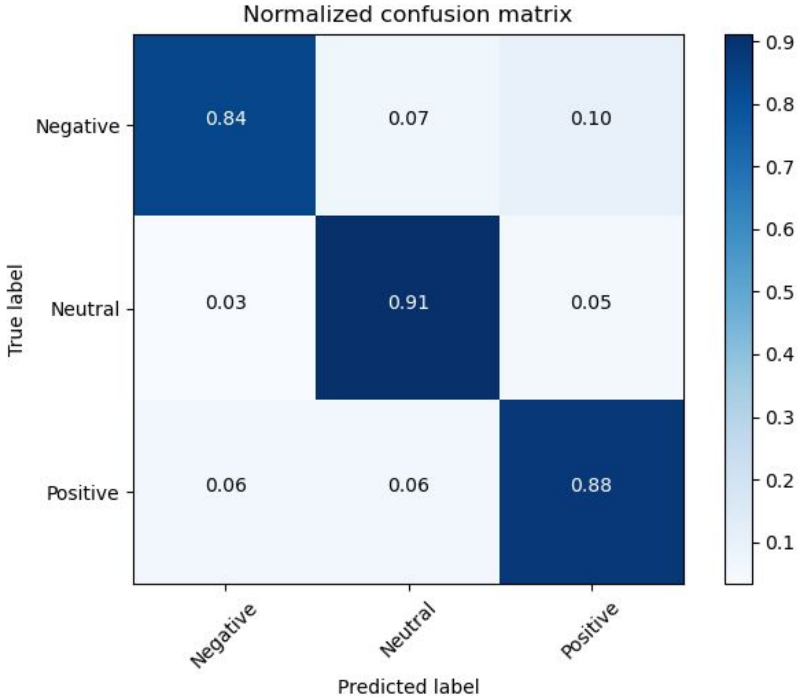
Naive Bayes Classifier: Results

Accuracy: 0.8840775823184484

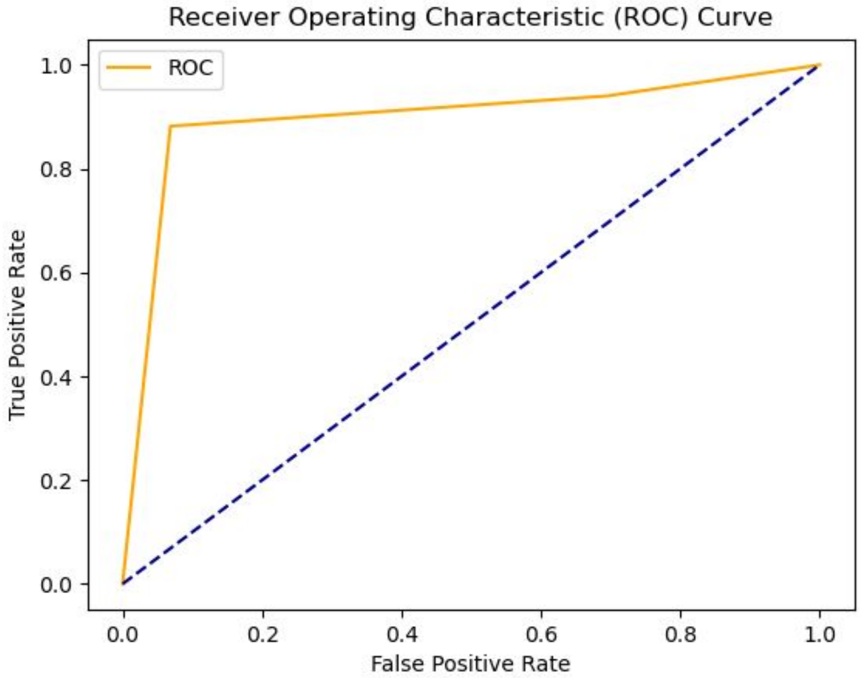
	Precision	Recall	F1-Score	Support
Negative	0.80	0.84	0.82	2068
Neutral	0.90	0.91	0.90	4041
Positive	0.91	0.88	0.90	4976
Accuracy			0.88	11085
Macro Avg	0.87	0.88	0.87	11085
Weighted Avg	0.89	0.88	0.88	11085

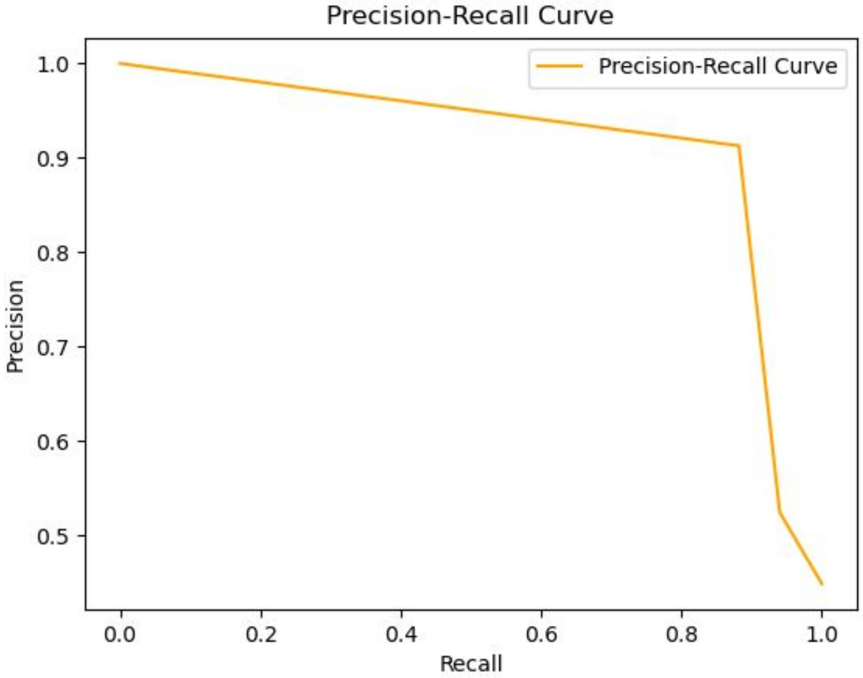
Naive Bayes Classifier: Results



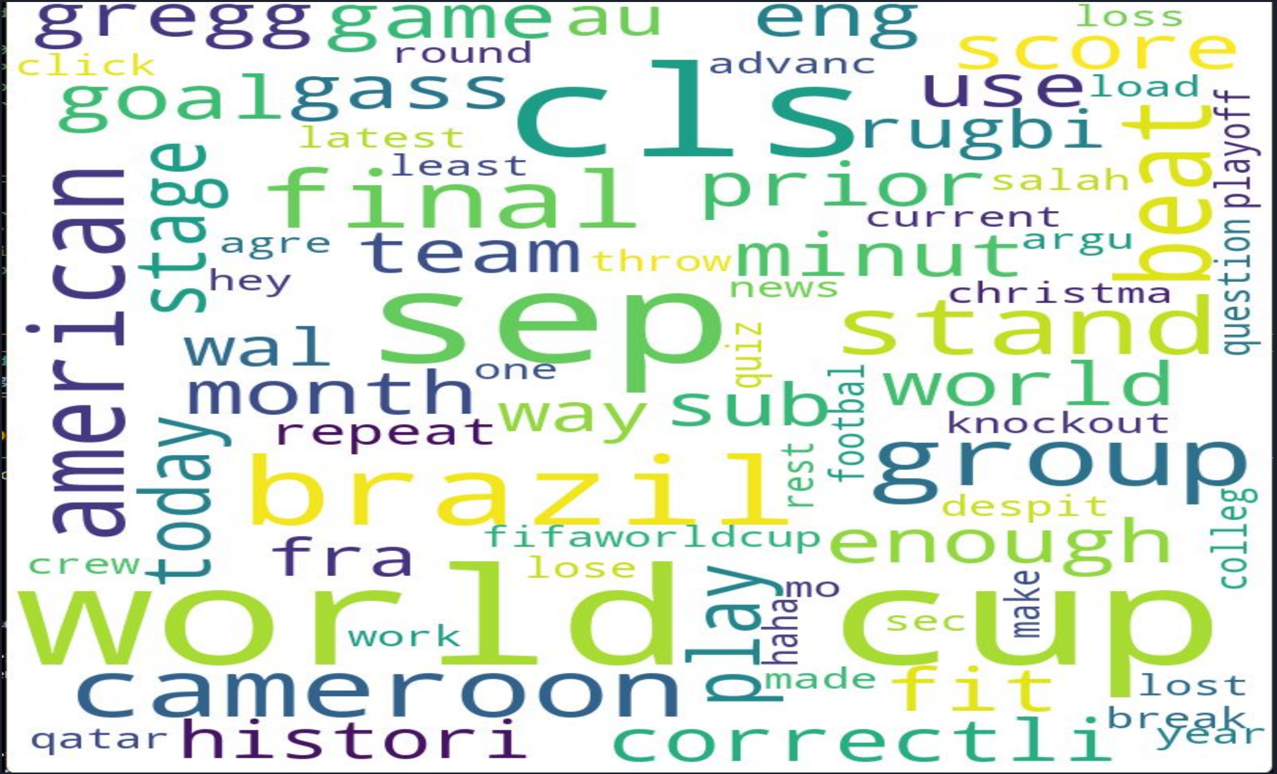


Naive Bayes Classifier: Results





Naive Bayes Classifier: Wordcloud



SVM Classification Model: Implementation

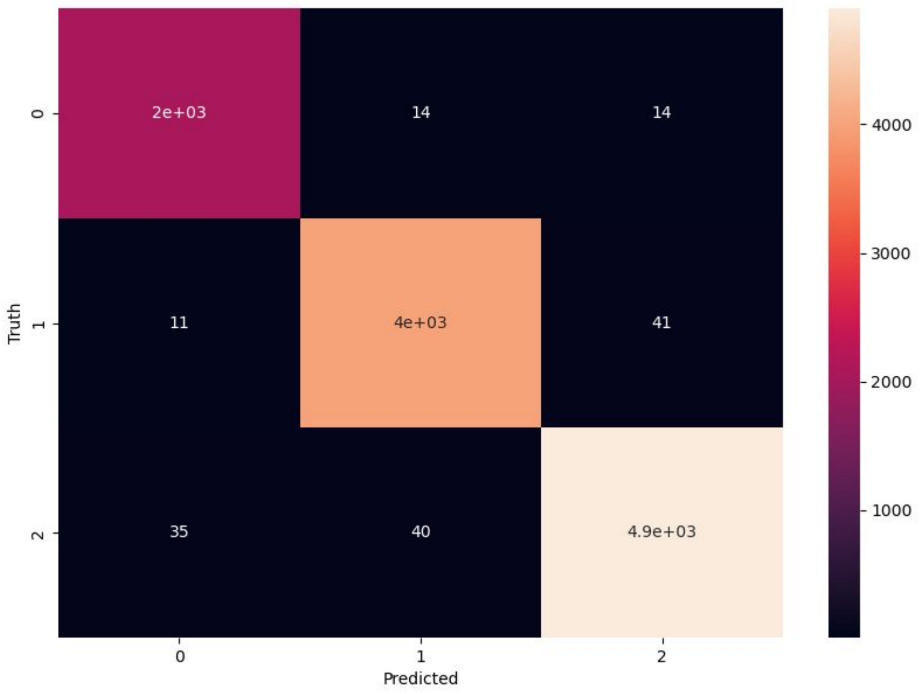
- Train Support Vector Machine classifier using pre classified data set
- TFID transformations to remove recurring text from the tweet
- Analyze sentiment of test data from dataset
- Determine performance and accuracy of classification
- Plot relations
- Graph word bubbles of words with positive and negative sentiment in tweets

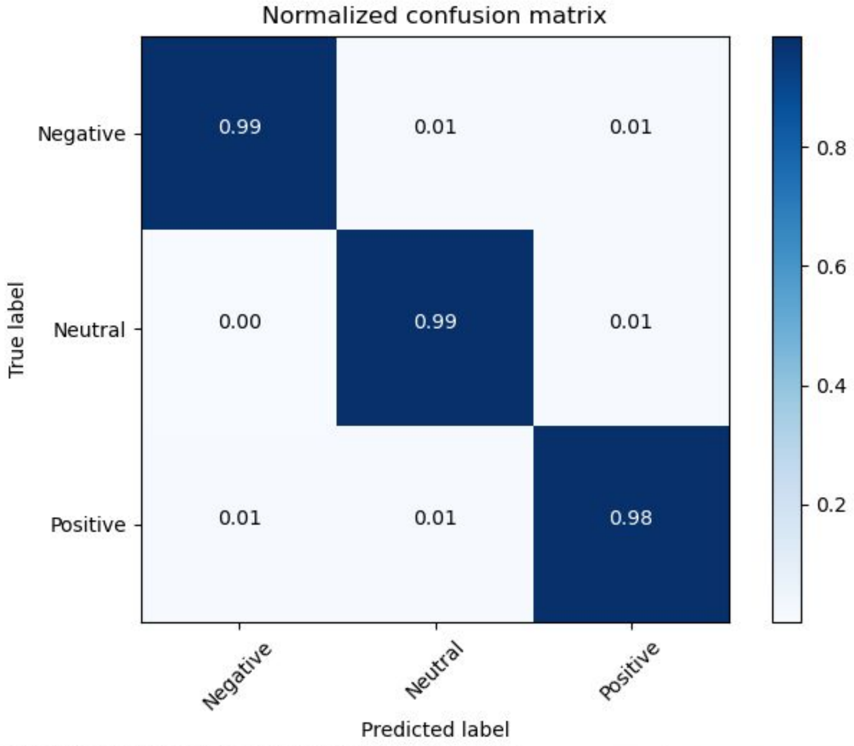
SVM Classification Model: Results

Accuracy: 0.9860171402796571

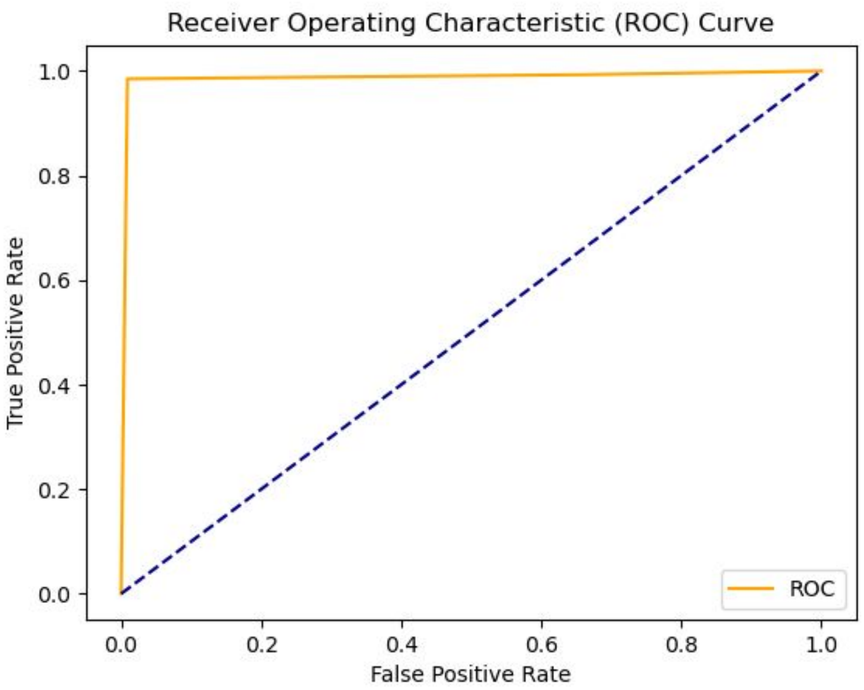
	Precision	Recall	F1-Score	Support
0	0.98	0.99	0.98	2068
1	0.99	0.99	0.99	4041
2	0.99	0.98	0.99	4976
Accuracy			0.99	11085
Macro Avg	0.98	0.99	0.99	11085
Weighted Avg	0.99	0.99	0.99	11085

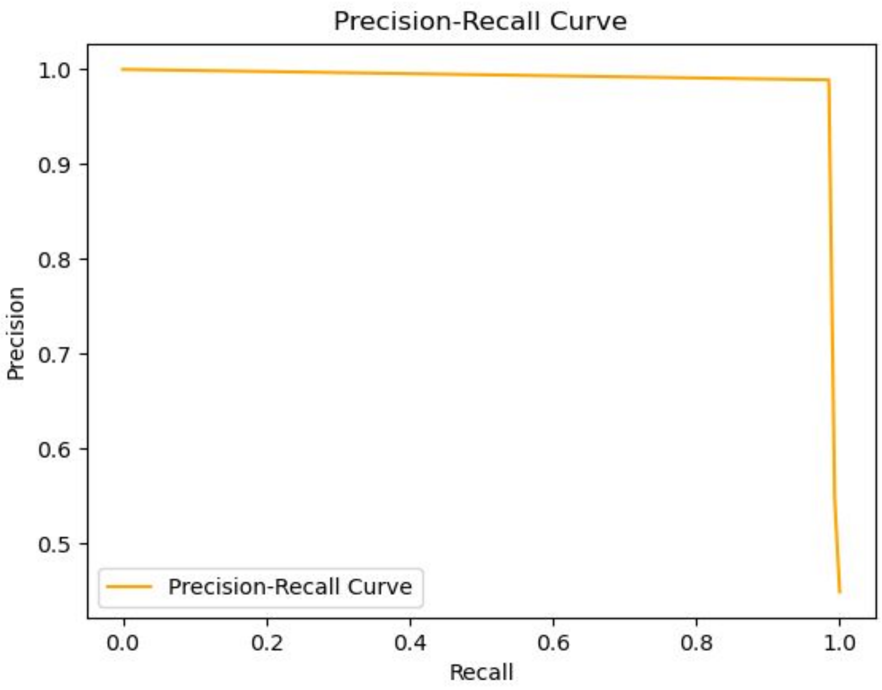
SVM Classification Model: Results



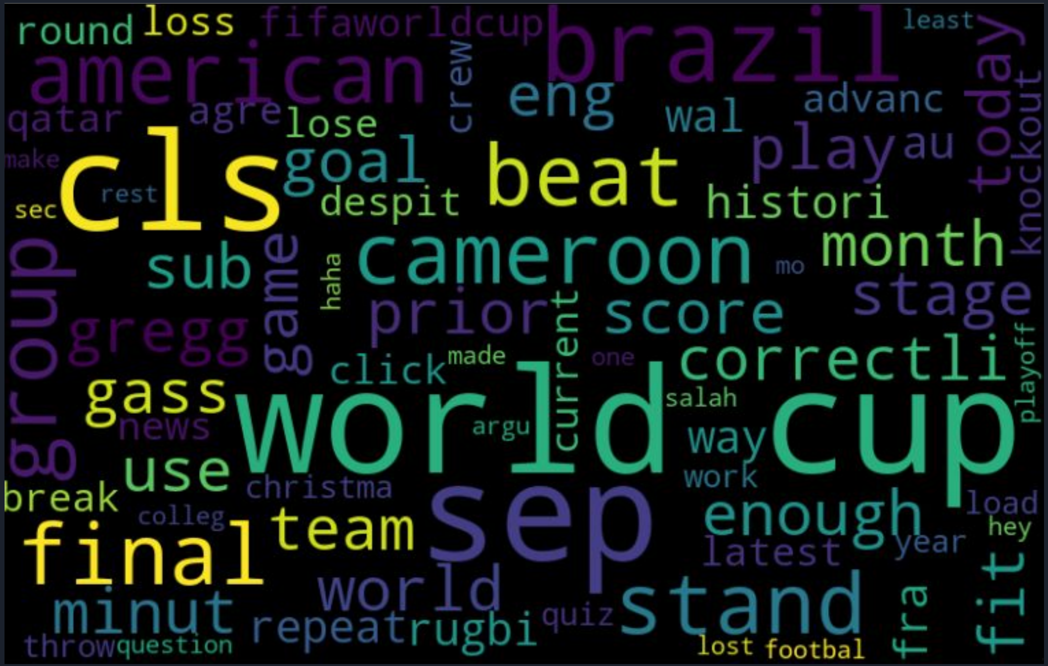


SVM Classification Model: Results





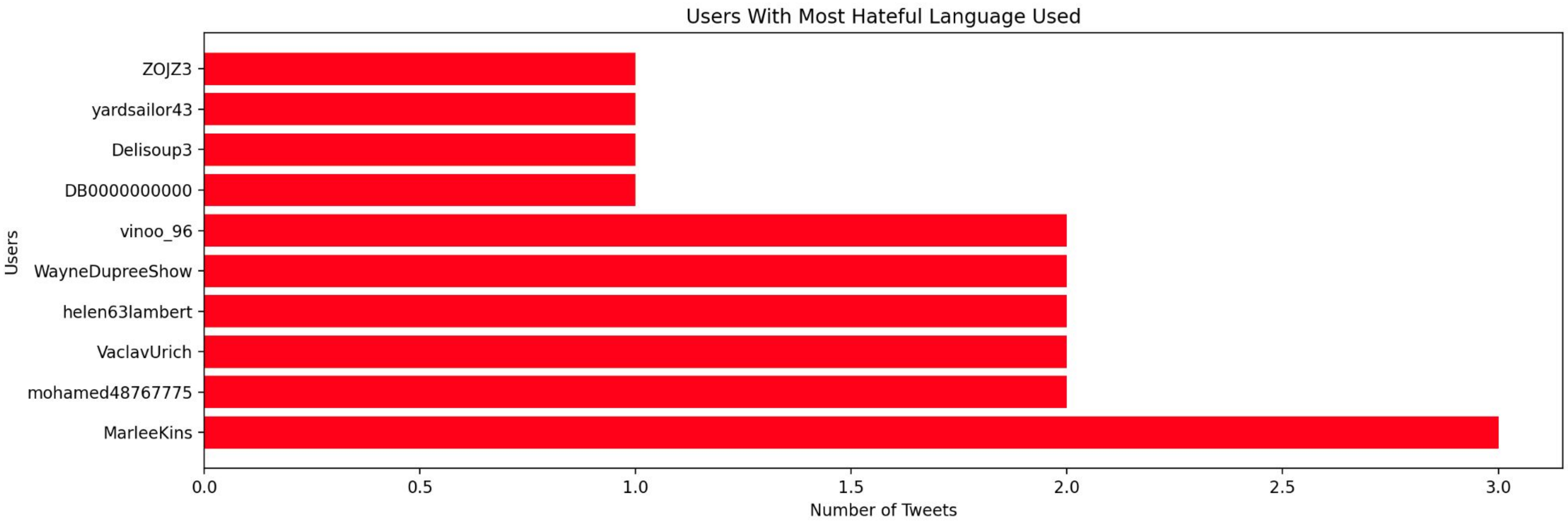
SVM Classification Model: Results



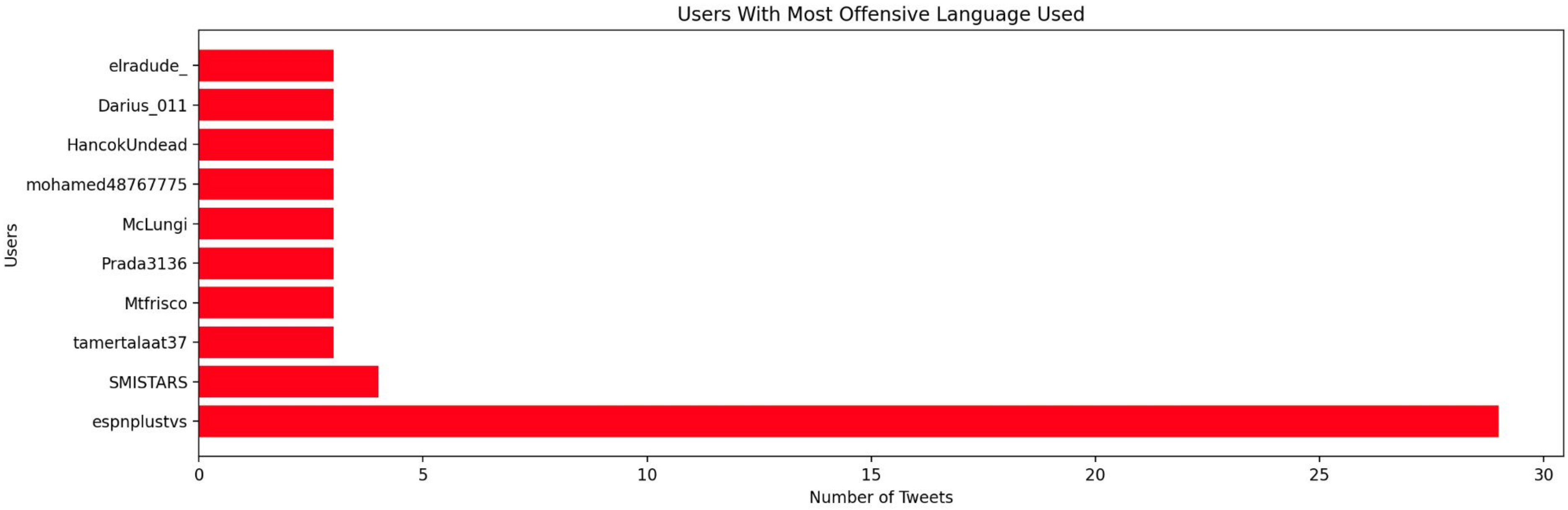
Hate Speech Detection: Implementation

- Analyze tweets with negative sentiment to determine if they contain hate speech
- Build Dataframes of users with with offensive and hateful tweets, and the respective tweets
- Graph users with the most offensive and hateful tweets

Hate Speech Detection: Results



Hate Speech Detection: Results



User Network of Hate Speech Users: Implementation

- Gather followers from users with the most hate speech and offensive language used
- Gather followers followers to determine links between users
- Model Data to show relationships between users and users followers

User Network of Hate Speech Users: Results