Twitter Sentiment Analysis

By Ashour Dankha & Viren Mody

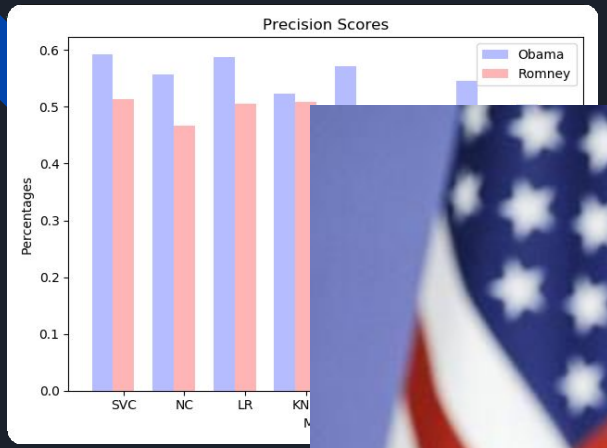
Preprocessing: 3 Stages

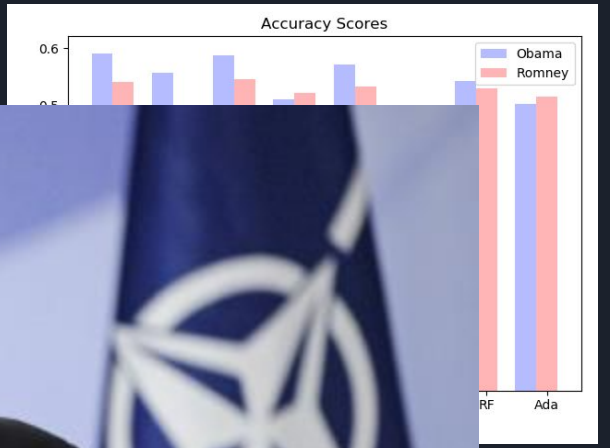
- ★ Regex Stage:
  - Links
  - Accents/Non UTF-8 letters
  - Digits
  - Hashtags
  - Punctuations
- ★ Stop Words Stage:
  - Removed All Stop Words (Using a NLTK Library)
- ★ Stemmer Stage:
  - Used a library (Porter Stemmer + One other) to get root of words. (i.e. Having -> hav)

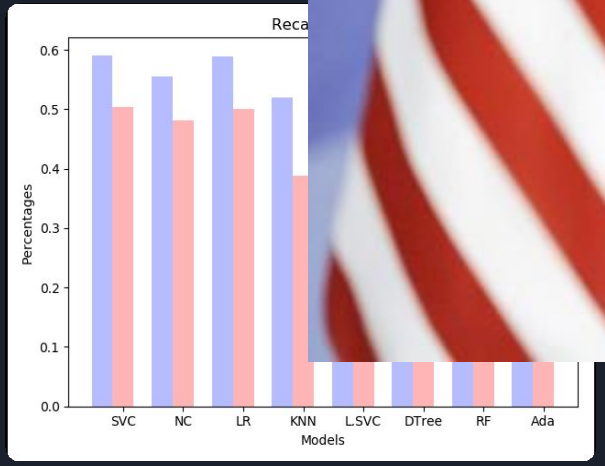
```
curr_tweet = regex_tweet(regex, curr_tweet)
curr_tweet = remove_stop_words(curr_tweet)
curr_tweet = porter_stemmer(curr_tweet)
return curr_tweet
```

Classification Stage

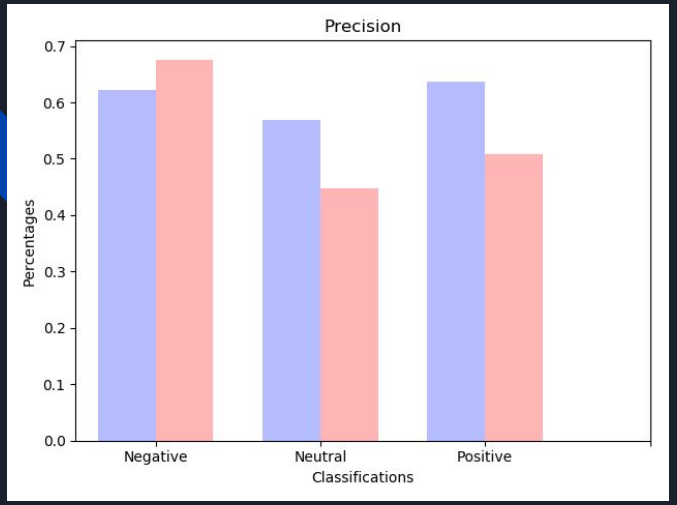
- ★ Evaluated multiple models (10+) and found/used the one with highest F-Score(s)
- ★ Tuned parameters to achieve higher scores for each individual model.
- ★ Cross Validation was 10, as per request.

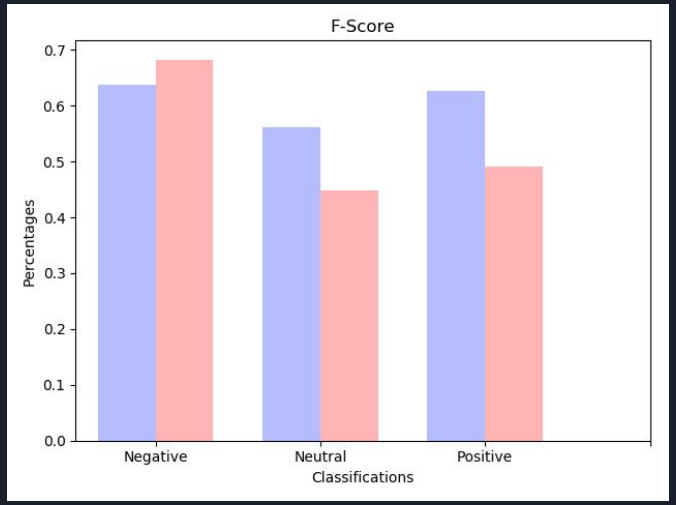

Precision Scores
Obama
Romney
Percentages
0.0
0.1
0.2
0.3
0.4
0.5
0.6
SVC
NC
LR
KN

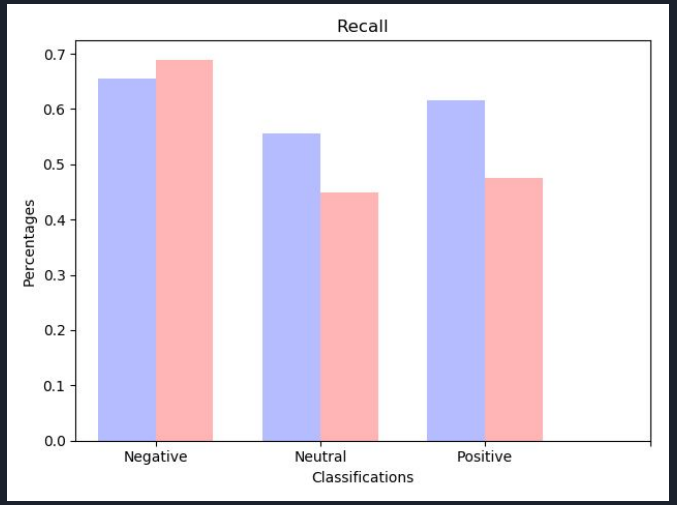

Accuracy Scores
Obama
Romney
0.6
RF
Ada

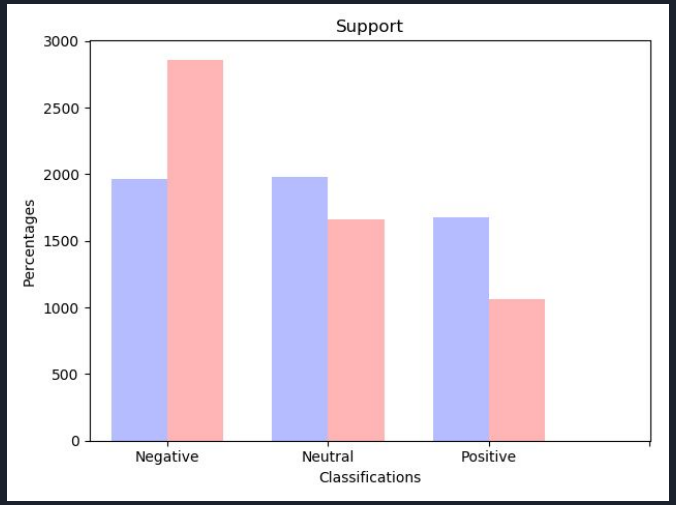

Obama
Romney
Percentages
0.0
0.1
0.2
0.3
0.4
0.5
0.6
SVC
NC
LR
KNN
LSVC
DTree
RF
Ada
Models

Obama
Romney
0.0
SVC
NC
LR
KNN
LSVC
DTree
RF
Ada
Models


Precision
Percentages
0.0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
Negative
Neutral
Positive
Classifications


F-Score
Percentages
0.0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
Negative
Neutral
Positive
Classifications


Recall
Percentages
0.0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
Negative
Neutral
Positive
Classifications


Support
Percentages
0
500
1000
1500
2000
2500
3000
Negative
Neutral
Positive
Classifications

Current/Future Strategies and Goals

- ★ Figure out better parameters for each model given our dataset.
- ★ Use other Vectorization Algorithms/Strategies to create features.
- ★ Find other stemmer algorithms that work well with Tweets.
- ★ Try to optimize run-time as things are a bit slow with a cv=10.
- ★ Achieve 75% F-Score averages without Deep-Learning.
- ★ Try out some Deep Learning Strategies (if time permits)