Machine Learning Application to create classification of Tweets, based on twitter data

Neel Khalade PA-67 IMLA- 36

Abstract-

Nowadays, machine learning is used in almost all the types of applications where large amount of dataset is generated. I have used ML (Machine Learning) on the Dataset of Republican vs Democrat Tweets of US 2016 elections.

Based on the humungous data on the twitter we can predict a lot of things, in my project I have tried to predict just a simple thing based on textual analysis

I have used various methods & algorithms to generate insights & predict whether a particular tweet is from a Republican or a Democrat.

Motivation-

Twitter generates millions of tweets per hour, so I was motivated to use twitter data.

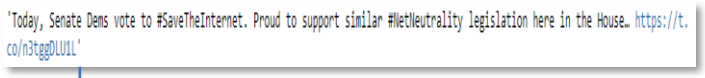
Political decisions are made by politicians but there is no actual trace of what people think of a particular decision or of a political event.

I was really motivated that 2016 election was heavily influenced by using Machine Learning & even UK Brexit elections was heavily influenced by using ML, so understanding the power of this, I have tried to understand how this works.

Preprocessing-

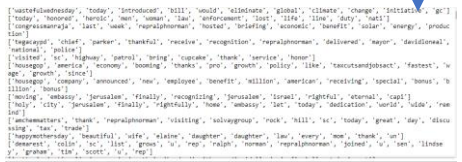
The dataset which I am using has tweets from the 200 political leaders in USA, so these tweets were having a lot of retweets, external links, hashtags, tagging of users etc. As the tweets had natural language of these people.

So, I had to remove all the stopwords, the words which are unnecessary, like is, an, the, who which occur a lot of times, also I had to remove the tags, retweets & hashtags, brackets etc.



Took nltk Stopwords

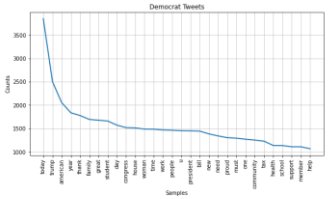
Tokenized the words

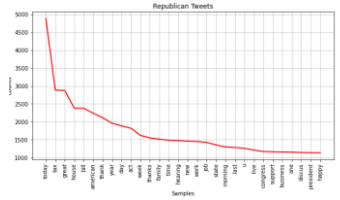


Converted words in proper lemma

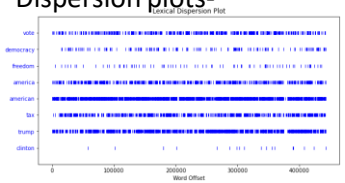
EDA & Sentiment analysis-

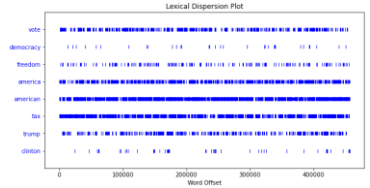
Did basic frequency distribution of words to find the most used words-





Dispersion plots-





Used textblob for measuring sentiments-

```
democratblob.sentiment
```

```
Sentiment(polarity=0.16536069955855845, subjectivity=0.4648999095113135)
```

```
republicanblob.sentiment
```

```
Sentiment(polarity=0.19836825482009132, subjectivity=0.4590716830328091)
```

This code was just to determine the political party, so the classification was only between to 2 outputs, we can use a similar strategy for 100ds of such results from the data available to us on the social media.

I imported all these types of classifiers & trained the model by tokenizing the training data using tfIdf & then looping through the data & the models, I calculated the score of each model

Result:

Multi-NB - 0.8144390966308775

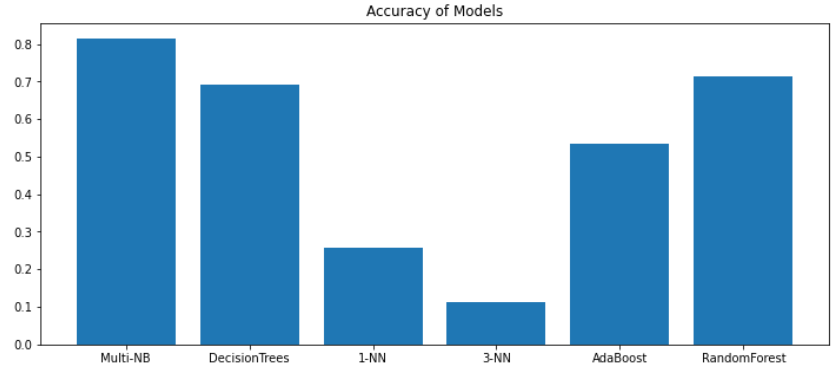
DecisionTrees - 0.6927787319841684

1-NN - 0.2583576760791951

3-NN - 0.11117413499716393

AdaBoost - 0.5350530674408643

RandomForest - 0.7135092556124458



Conclusion-

The multi-nb had the best accuracy of determining the political party, whereas AdsBoost & Randomforest being very good at classification couldn't compete with the score of multi-naiive bayes which was surprising. KNN failed because I had textual data which can be total random that's why I did frequency & dispersion plots to see how some meaningful words are affecting the model on both sides