Political Prediction Using Twitter Tweets

PA29 Shrutika
PA62 Shriyash Shingare
PA67 Neel Khalade

- Introduction
- Data set
- Data cleaning
- Data visualization techniques

Political data analysis

- 2016 US election data
- Extracted tweets (kaggle data)
- Data visualization techniques
  - Matplot
  - Wordcloud
  - D3

Data set

```
In [3]: data.head(10)
```

Out[3]:

	Party	Handle	Tweet
0	Democrat	RepDarrenSoto	Today, Senate Dems vote to #SaveTheInternet. P...
1	Democrat	RepDarrenSoto	RT @WinterHavenSun: Winter Haven resident / Al...
2	Democrat	RepDarrenSoto	RT @NBCLatino: .@RepDarrenSoto noted that Hurr...
3	Democrat	RepDarrenSoto	RT @NALCABPolicy: Meeting with @RepDarrenSoto ...
4	Democrat	RepDarrenSoto	RT @Vegalteno: Hurricane season starts on June...
5	Democrat	RepDarrenSoto	RT @EmgageActionFL: Thank you to all who came ...
6	Democrat	RepDarrenSoto	Hurricane Maria left approx $90 billion in dam...
7	Democrat	RepDarrenSoto	RT @Tharryry: I am delighted that @RepDarrenSo...
8	Democrat	RepDarrenSoto	RT @HispanicCaucus: Trump's anti-immigrant pol...
9	Democrat	RepDarrenSoto	RT @RepStephMurphy: Great joining @WeAreUnidos...

Extracted Tweets from kaggle

Data cleaning

1- DATA CLEANING

Before starting the analysis, we should remove like @, rt, html, # etc. objects.

In [5]:
```
import re
import nltk
from nltk.corpus import stopwords
import nltk as nlp
stopwords = stopwords.words('english')
stopwords.append('rt');
#(such as “the”, “a”, “an”, “in”)
```

In [6]:
```
#we created 2 different class as democrat and republican
democrat=data[data.Party=="Democrat"]
republican=data[data.Party=="Republican"]
```

In [7]:
```
#Cleaning democrat party tweets
democrat_list=[]
for d in democrat.Tweet:
    d=re.sub(r'http\S+', '', d) #remove links
    d=re.sub("[^a-zA-Z]", " ", d) #remove all characters except letters
    d=d.lower() #convert all words to lowercase
    d=nltk.word_tokenize(d) #split sentences into word
    d=[word for word in d if not word in set(stopwords)] #add to stopwords list if unnecessary words.
    lemma=nlp.WordNetLemmatizer()
    d=[lemma.lemmatize(word) for word in d] #identify the correct form of the word in the dictionary #eg . Voting to
    d=" ".join(d)
    democrat_list.append(d) #append words to list
```

In [9]:
```
#first 5 tweets in the list
democrat_list[0:5]
```

Out[9]:
```
['today senate dems vote savetheinternet proud support similar netneutrality legislation house',
 'winterhavensun winter resident alta vista teacher one several recognized repdarrensoto national teacher apprecia'
,
 'nbclatino repdarrensoto noted hurricane maria left approximately billion damage congress allocated',
 'nalcabpolicy meeting repdarrensoto thanks taking time meet latinoleader ed marucci guzman nalcabpolicy',
 'vegalteno hurricane season start june st puerto rico readiness well pwr puertorico repdarrensoto espaillatny']
```

In [10]:
```
#first 5 tweets in the list
republican_list[0:5]
```

Out[10]:
```
['wastefulwednesday today introduced bill would eliminate global climate change initiative gc',
 'today honored heroic men woman law enforcement lost life line duty nati',
 'congressmanraja last week repralphnorman hosted briefing economic benefit solar energy production',
 'tegacaypd chief parker thankful receive recognition repralphnorman delivered mayor davidloneal national police',
 'visited sc highway patrol bring cupcake thank service honor']
```

split sentences into words

In [11]:

```
democrat_tweets=str(democrat_list).split()
republican_tweets=str(republican_list).split()
democrat_tweets=[word.replace("'","") for word in democrat_tweets ]
democrat_tweets=[word.replace("[", "") for word in democrat_tweets ]
democrat_tweets=[word.replace("]","") for word in democrat_tweets ]
democrat_tweets=[word.replace(",", "") for word in democrat_tweets ]

republican_tweets=[word.replace("'","") for word in republican_tweets ]
republican_tweets=[word.replace("[", "") for word in republican_tweets ]
republican_tweets=[word.replace("]","") for word in republican_tweets ]
republican_tweets=[word.replace(",", "") for word in republican_tweets ]
```

Now lets check length of two list.

In [12]:

```
print("Democrat tweets word length:",len(democrat_tweets))
print("Republican tweets word length:",len(republican_tweets))
```

```
Democrat tweets word length: 443138
Republican tweets word length: 457293
```

Frequency of Usage of Words by Parties

In [19]:
```
democratclass=[]
for each in new.FrequencyDemocrat:
    if each<50:
        democratclass.append("Very Low")
    elif 49<each<150:
        democratclass.append("Low")
    elif 149<each<500:
        democratclass.append("Medium")
    elif 499<each<1500:
        democratclass.append("High")
    else:
        democratclass.append("Very High")

new["democratclass"]=democratclass
```

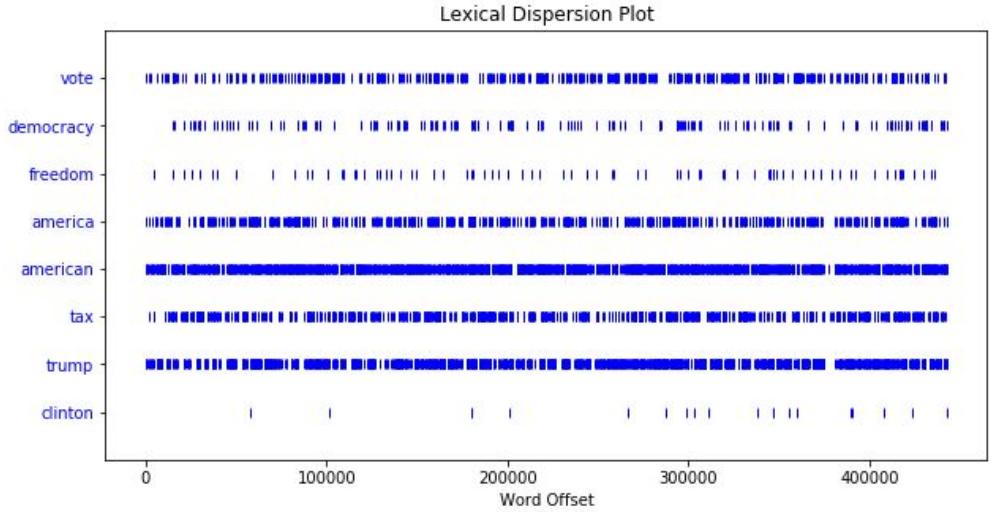
In [20]:
```
republicanclass=[]
for each in new.FrequencyRepublican:
    if each<50:
        republicanclass.append("Very Low")
    elif 49<each<150:
        republicanclass.append("Low")
    elif 149<each<500:
        republicanclass.append("Medium")
    elif 499<each<1500:
        republicanclass.append("High")
    else:
        republicanclass.append("Very High")

new["republicanclass"]=republicanclass
```

Dispersion Plots

Democrat Tweets Plot

```
In [26]: plt.subplots(figsize=(10,5))
         democrat_tweet.dispersion_plot(["vote","democracy","freedom","america","american","tax","trump","clinton"])
```



Wordcloud

In [5]:

```
# Wordcloud top 100 words and color by party
highest <- words[1:100,]
wordcloud(words = highest$words, freq = highest$count,
          ordered.colors = T,
          scale=c(4,.5),
          colors = c('#0015BC', '#FF0000')[factor(highest$party)])
```

Analysing different classifier score

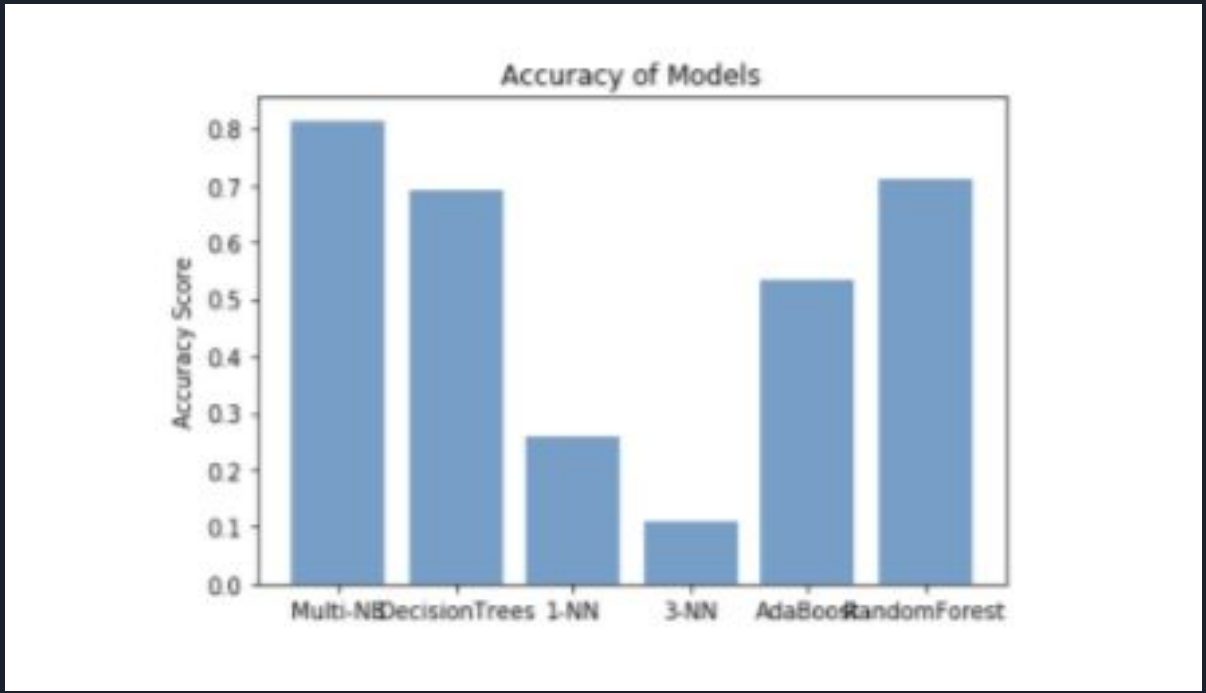
In [10]:

```
predictScores = []

for i in range(len(clfs)):
    classifierFunction(clfs[i], X_train, Y_train)
    Y_pred = predictFunction(clfs[i], X_test)
    score = f1_score(Y_test, Y_pred)
    predictScores.append(score)
    print(score)
```

```
0.8142936493241993
0.691502184722013
0.25874716001298276
0.11116625310173696
0.533038589072036
0.7094591905766245
```

D3


Accuracy of Models
Accuracy Score
0.0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
Multi-NB
DecisionTrees
1-NN
3-NN
AdaBoost
RandomForest