**Twitter Sentiment Analysis on Election 2018 Tweets**

Prepared by:

Bodduna Satwick - [11810077]

V Venkata Sai KushwanthReddy - [11810002]

#Loading requried Libries

import numpy as np

import pandas as pd

import matplotlib.pyplot as plt

from textblob import TextBlob

from nltk.corpus import stopwords

import re

%matplotlib inline

**Loading Data**

Election\_data = pd.read\_csv('TelanganaTweets.csv'); #Loading the Scraped Data from twitter using pandas

print("Total Tweets avaliable : %d "% (len(Election\_data))) #Total Number of Tweets Available

Election\_data.info() #Checkcing info of the Scrapped data

pattern= r"(b')|\\x([0-9a-fA-F]+)|\\n" #assinging customized regex pattern which for processing the tweets

**Processing Data**

processed\_data =[];

for test in range(len(Election\_data.iloc[:,0])): #looping the first column of the df which is Tweet

processed\_data.append(re.sub(pattern,"",Election\_data.iloc[test,0])); #Using the re.sub we are matching the pattern and replacing it with blank

preprocess\_data = processed\_data #Saving the data into a temp

preprocess\_data[1:5] #Checking the output

for tweets in range(len(Election\_data.iloc[:,0])): #Cleaning the tweets further removing all the symbols except numbers,aplhabates and #,@ we will be using #,@ for few analysis

processed\_data[tweets] = re.sub('[^ a-zA-Z|#|@|0-9]','',processed\_data[tweets])

processed\_data[tweets] = processed\_data[tweets].lower() #making all the tweets lowercase

preprocess\_data[1:5] #Checking the output

Election\_data['processed\_text']=processed\_data #Adding the temp data which processed and cleaned to the main dataframe

Election\_data.head() #Checking the data

**Analysis on the tweets**

#Assinging the tweets for a party by using the hashtags

Election\_data['Split\_Data'] = '';

for counting in range(len(preprocess\_data)): #Using the Preprocess data for spliting

preprocess\_data[counting] = str.split(preprocess\_data[counting]);

#Spliting the tweets into words for matching the dictonary items and for assiging party

Election\_data['Split\_Data']= preprocess\_data; #adding this words of the tweet to the dataframe

Election\_data['Party']=''; #Creating a new column to store the party id

for tweet in range(len(Election\_data['Split\_Data'])): #using the tweet words which we have processed earlier

TRS\_Dict = ['@trspartyonline','@ktrtrs','#trs','#ktr','rao','#trs','#kcr','@asadowaisi','#telanganawithkcr','#aimim','#harishrao','@telanganacmo','#voteforcar','#phirekbaarkcr','@raokavitha','#kcrfailedtelangana','#votefortrs','#trsparty','#askktr','b"@ktrtrs','#trs,','#trs105#phirekbaarkcr','#prajaashirvadasabha','#phirekbaarkcr','#voteforpadmarao'];

bjp\_Dict = ['@bjp4india','@bjp4telangana','modi','@bjp4india','#govotebjp','@narendramodi']

Congress\_Dict = ['tdp','congress','#mahakutami','#congress','@incindia','@jaitdp','#prajakutami','#mahakutami','#chandrababunaidu','@rahulgandhi','#revanthreddy','@uttamtpcc','#byebyetrs','alliance','#congressiswinning','#cpi','#cbnints','#cbnintelangana','#rahulgandhi','kutami','#prajakutamiwinning','#peoplesfront','#tdpcongress']

#creating a list of hashtags or words which belongs to a each particular party which we are going to check in each tweet and assign to a party

TRS\_count = 0;

bjp\_count =0;

Congress\_count = 0;

other\_count =0;

#intializing the variable with 0

for words in Election\_data.iloc[tweet,10]: #Creating a for loop getting the count of each words and party weightage

if words in TRS\_Dict: #If a tweet find the word in its party dict it will assign 1 and if it find again it will increment the value

TRS\_count += 1;

elif words in bjp\_Dict:

bjp\_count +=1;

elif words in Congress\_Dict:

Congress\_count +=1;

else:

other\_count =1; #If it dont find any of the words belongs to the any party then it will assign the value 1 to others

if TRS\_count > bjp\_count and TRS\_count > Congress\_count: #here we will check the weightage of each party , this basically assign the party tag to the tweet which has the high weight

Election\_data.iloc[tweet,11] = 'TRS';

elif bjp\_count >TRS\_count and bjp\_count > Congress\_count:

Election\_data.iloc[tweet,11] = 'BJP'

elif Congress\_count >TRS\_count and Congress\_count>bjp\_count:

Election\_data.iloc[tweet,11] = 'Congress'

else:

Election\_data.iloc[tweet,11] = 'Neutral' #if all the party weightage is zero then it will tag netural

Election\_data.head() #Checking the data after tagging the party to every tweet

**Sentimental Analysis on tweets**

#Creating the function for checking the sentiment of the tweet

def sentimentscore(tweet):

analysis = TextBlob(tweet) #passing the tweet into the textblob for analysing the polarity

if analysis.sentiment.polarity > 0: #Here we are assing one if the polarity is greater than 0 as positive tweet

return 1

elif analysis.sentiment.polarity == 0: #Here we are assing zero if the polarity is is equals to0 as netural tweet

return 0

else:

return -1 #Here we are assing -1 if the polarity is lessthan than 0 as negative tweet

Election\_data['sentiment'] = Election\_data['processed\_text'].apply(sentimentscore)

#Applying the above defined function to the Processed text

#Checking the number of positve,Negative and neutral tweets

print("Total Positve Tweets : %d "% (len(Election\_data[Election\_data['sentiment']>0])))

print("Total Neutral Tweets : %d "% (len(Election\_data[Election\_data['sentiment']==0])))

print("Total Negative Tweets : %d "% (len(Election\_data[Election\_data['sentiment']<0])))

#Giving tag for the dataframe for further analysis on tableau

Election\_data['State']='Telangana'

Election\_data.head() #Checking the data

Election\_data[Election\_data['Party']!='Neutral'].Party.value\_counts().plot(kind='bar') #Plotting the basic graph for review

**Analysis on Hashtag's**

#word frequency of hashtags

stop = stopwords.words('english') #Loading the stopwords from library

Election\_data["Clean\_tweet\_stopwords"] = Election\_data["processed\_text"].apply(lambda x: ' '.join([word for word in x.split() if word not in (stop)]))

#by using the stopwords library we are searching in the tweets and replacing it with blank

FrequencyTags=pd.Series(np.concatenate([x.split() for x in Election\_data.Clean\_tweet\_stopwords])).value\_counts()

#Calculating the frequency of each tag

HashFreq = pd.DataFrame(FrequencyTags) #converting the pandas series to dataframe

HashFreq=HashFreq.reset\_index() #resting the index

HashFreq.columns = ['HashWord', 'Frequency'] #adding the colum names to the new df

HashFreq['State']='Telangana' #adding Tag for further analysis in Tableau

HashFreq.head() #Checking the data

**Storing the processed data into csv**

Election\_data.to\_csv('TelanganaSentimentProcessed.csv') #storing the processdata into the csv

HashFreq.to\_csv('TelanganaWords.csv') #Sotring the word frequency into csv