import tweepy

from textblob import TextBlob

from wordcloud import WordCloud

import pandas as pd

import numpy as np

import re

import matplotlib.pyplot as plt

plt.style.use('fivethirtyeight')

# Load the data

from google.colab import files

uploaded = files.upload()

# Get the data

log = pd.read\_csv("Login.csv")

# Twitter Api Credentials

consumerKey = log["key"][0]

consumerSecret = log["key"][1]

accessToken = log["key"][2]

accessTokenSecret = log["key"][3]

# Create the authentication object

authenticate = tweepy.OAuthHandler(consumerKey, consumerSecret)

# Set the access token and access token secret

authenticate.set\_access\_token(accessToken, accessTokenSecret)

# Creating the API object while passing in auth information

api = tweepy.API(authenticate, wait\_on\_rate\_limit = True)

# Extract 100 tweets from the twitter user

posts = api.user\_timeline(screen\_name="BillGates", count = 100, lang ="en", tweet\_mode="extended")

# Print the last 5 tweets

print("Show the 5 recent tweets:\n")

i=1

for tweet in posts[:5]:

print(str(i) +') '+ tweet.full\_text + '\n')

i= i+1

# Create a dataframe with a column called Tweets

df = pd.DataFrame([tweet.full\_text for tweet in posts], columns=['Tweets'])

# Show the first 5 rows of data

df.head()

# Create a function to clean the tweets

def cleanTxt(text):

text = re.sub('@[A-Za-z0–9]+', '', text) #Removing @mentions

text = re.sub('#', '', text) # Removing '#' hash tag

text = re.sub('RT[\s]+', '', text) # Removing RT

text = re.sub('https?:\/\/\S+', '', text) # Removing hyperlink

return text

# Clean the tweets

df['Tweets'] = df['Tweets'].apply(cleanTxt)

# Show the cleaned tweets

df

# Create a function to get the subjectivity

def getSubjectivity(text):

return TextBlob(text).sentiment.subjectivity

# Create a function to get the polarity

def getPolarity(text):

return TextBlob(text).sentiment.polarity

# Create two new columns 'Subjectivity' & 'Polarity'

df['Subjectivity'] = df['Tweets'].apply(getSubjectivity)

df['Polarity'] = df['Tweets'].apply(getPolarity)

# Show the new dataframe with columns 'Subjectivity' & 'Polarity'

df

allWords = ' '.join([twts for twts in df['Tweets']])

wordCloud = WordCloud(width=500, height=300, random\_state=21, max\_font\_size=110).generate(allWords)

plt.imshow(wordCloud, interpolation="bilinear")

plt.axis('off')

plt.show()

# Create a function to compute negative (-1), neutral (0) and positive (+1) analysis

def getAnalysis(score):

if score < 0:

return 'Negative'

elif score == 0:

return 'Neutral'

else:

return 'Positive'

df['Analysis'] = df['Polarity'].apply(getAnalysis)

# Show the dataframe

df

# Printing positive tweets

print('Printing positive tweets:\n')

j=1

sortedDF = df.sort\_values(by=['Polarity']) #Sort the tweets

for i in range(0, sortedDF.shape[0] ):

if( sortedDF['Analysis'][i] == 'Positive'):

print(str(j) + ') '+ sortedDF['Tweets'][i])

print()

j= j+1

# Printing negative tweets

print('Printing negative tweets:\n')

j=1

sortedDF = df.sort\_values(by=['Polarity'],ascending=False) #Sort the tweets

for i in range(0, sortedDF.shape[0] ):

if( sortedDF['Analysis'][i] == 'Negative'):

print(str(j) + ') '+sortedDF['Tweets'][i])

print()

j=j+1

# Plotting

plt.figure(figsize=(8,6))

for i in range(0, df.shape[0]):

plt.scatter(df["Polarity"][i], df["Subjectivity"][i], color='Blue') # plt.scatter(x,y,color)

plt.title('Sentiment Analysis')

plt.xlabel('Polarity')

plt.ylabel('Subjectivity')

plt.show()

# Print the percentage of positive tweets

ptweets = df[df.Analysis == 'Positive']

ptweets = ptweets['Tweets']

ptweets

round( (ptweets.shape[0] / df.shape[0]) \* 100 , 1)

# Print the percentage of negative tweets

ntweets = df[df.Analysis == 'Negative']

ntweets = ntweets['Tweets']

ntweets

round( (ntweets.shape[0] / df.shape[0]) \* 100, 1)

# Show the value counts

df['Analysis'].value\_counts()

# Plotting and visualizing the counts

plt.title('Sentiment Analysis')

plt.xlabel('Sentiment')

plt.ylabel('Counts')

df['Analysis'].value\_counts().plot(kind = 'bar')

plt.show()