PROJECT

PROBLEM STATEMENT :

The aim of the project is to apply the principles of machine learning on twitter data to analysis sentiment on '*Demonitisation*'. This project is aimed to determine the amount of positive, negative and neutral tweets with help of a pie-chart.

1 EXTRACTION OF TWITTER DATA :

1.1 IMPORTING LIBRARIES:

import tweepy

import pandas as pd

import numpy as np

from IPython.display import display

import matplotlib.pyplot as plt

1.2 CREATING TWITTER APP :

In order to extract tweets for a posterior analysis, we need to access to our Twitter account and create an app. This can be done from https://apps.twitter.com

We saved the requirements in credentials.py:

- Consumer Key (API key)
- Consumer Secret (API secret)
- Access Token
- Access Token Secret

eg: the script goes as follows:

Twitter App access key for the user:

CONSUMER_KEY = ' 5IKNCorAoNQZsaxIRSrbhMpZ4'
CONSUMER_SECRET = 'GjjvFO6UbdU8q1bZ8ct3p5VHjixYM3t4BgSEylkB7a6nkgl3SM'

ACCESS_TOKEN=' 952466985148796928-mVcheCwoQxjkgDplStBmTAeASsM3eKE'
ACCESS_SECRET= '7B8bWQALMH81KFaWsUeqFTTUAxVzexojO3R1zIvCm1cKd'

```
# we import our access keys :

 from credentials.py import *

# API's setup:


def twitter_setup() :

    """Utility function to setup the Twitter's API
    with our access keys provided. """

    # Authentication and access using keys:

    auth = tweepy.OAuthHandler(CONSUMER_KEY, CONSUMER_SECRET)
    auth.set_access_token(ACCESS_TOKEN, ACCESS_SECRET)

    # Return API with authentication:

    api = tweepy.API(auth)
    return api
```

1.3. Tweets extraction

```
#As mentioned I have used 'demonitisation' as keyword to search the twitter app database
and extract all tweets on this topic and store it in variable named 'tweets'.

extractor = twitter_setup()

tweets=extractor.search('demonitisation')

1.4
 #We printed latest 5 tweets on 'demonitisation'

print(tweets[:5])

data = pd.DataFrame(data=[tweet.text for tweet in tweets], columns=['Tweets'])

1.5

def clean_tweet(tweets):

'''Utility function to clean the text in a tweet by removing
    links and special characters using regex.'''


    return ' '.join(re.sub("(@[A-Za-z0-9]+)|([^0-9A-Za-z \t])|(\w+:\/\/\S+)", " ", tweets).split())

1.6 Analysis of nature of satement :
```

```
    analysis = TextBlob(clean_tweet(tweets))
    if analysis.sentiment.polarity > 0:
        return 1
    elif analysis.sentiment.polarity == 0:
        return 0
    else:
        return -1



data['SA'] = np.array([ analize_sentiment(tweet) for tweet in data['Tweets'] ])

pos_tweets = [ tweet for index, tweet in enumerate(data['Tweets']) if data['SA'][index] > 0]
#print(pos_tweets[:2])

neg_tweets = [ tweet for index, tweet in enumerate(data['Tweets']) if data['SA'][index] < 0]

nutral_tweets = [ tweet for index, tweet in enumerate(data['Tweets']) if data['SA'][index] ==
0]
```

2. Visualisation of the result on tweets:

```
# picking positive tweets from tweets
#ptweets = [tweet for tweet in tweets if tweet['sentiment'] == 'positive']
# percentage of positive tweets

print("Positive tweets percentage: {} %".format(100*len(pos_tweets)/len(tweets)))
print("Negative tweets percentage: {} %".format(100*len(neg_tweets)/len(tweets)))
print("Nutral tweets percentage: {} %".format(100*len(nutral_tweets)/len(tweets)))

lables=['Positive','Negative','Neutral']

valu=[(100*len(pos_tweets)/len(tweets)),(100*len(neg_tweets)/len(tweets)),(100*len(nutral_t
weets)/len(tweets))]

fig1,ax1=plt.subplots()

ax1.pie(valu,labels=lables, autopct='%1.1f%%',
        shadow=True, startangle=180)
ax1.axis('equal')

plt.show()
```

RESULT COMES AS: