In Class Assignment 2,3

My Experience:

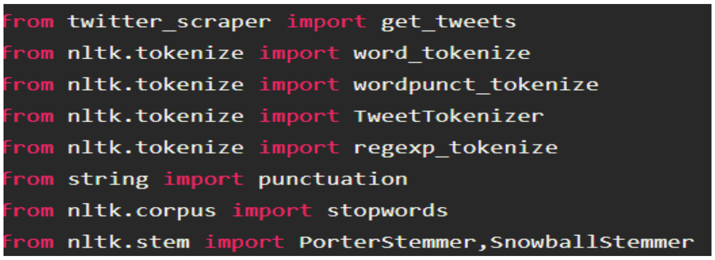
a) In the First in Class Assignment I gained hands on experience about extracting tweets from my twitter developer account.

b) In this Assignment, I gained hands on experience of Converting the tweets to vectors, calculating the similarity among the texts, reduction of the dimensionality of the text using the PCA Algorithm, Using the K-Means clustering algorithm and plotting the graphs.

c) Please note I have converted the Jupyter Notebook into the “dark mode” using `jupyter-themes`

Step 1: Importing Libraries

One of the most important part of performing all the above-mentioned operations is the usage of the Libraries. I have used the following Libraries. I did try using additional, different libraries. However, the below gave me the expected output.



```
from twitter_scraper import get_tweets
from nltk.tokenize import word_tokenize
from nltk.tokenize import wordpunct_tokenize
from nltk.tokenize import TweetTokenizer
from nltk.tokenize import regexp_tokenize
from string import punctuation
from nltk.corpus import stopwords
from nltk.stem import PorterStemmer,SnowballStemmer
```

Screenshot 1: Importing Libraries

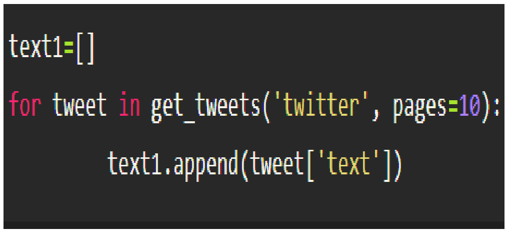
Step 2: Extracting Tweets:

I have Implemented this using two approaches to demonstrate Extracting tweets.

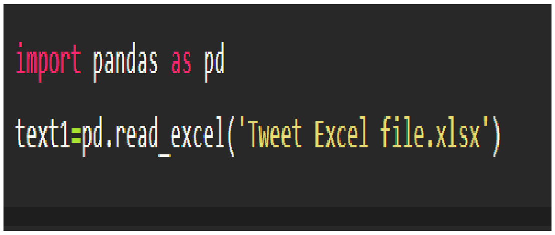
In **Scenario 1**: I am extracting tweets using the *twitter_scraper* library using the #as twitter followed by page number as 10.

In **Scenario 2**: I have extracted the tweets (10 tweets) from an Excel File.

My Intention here was to demonstrate flexibility in the extraction of tweets onto the Jupyter Notebook console.



```
text1=[]
for tweet in get_tweets('twitter', pages=10):
        text1.append(tweet['text'])
```



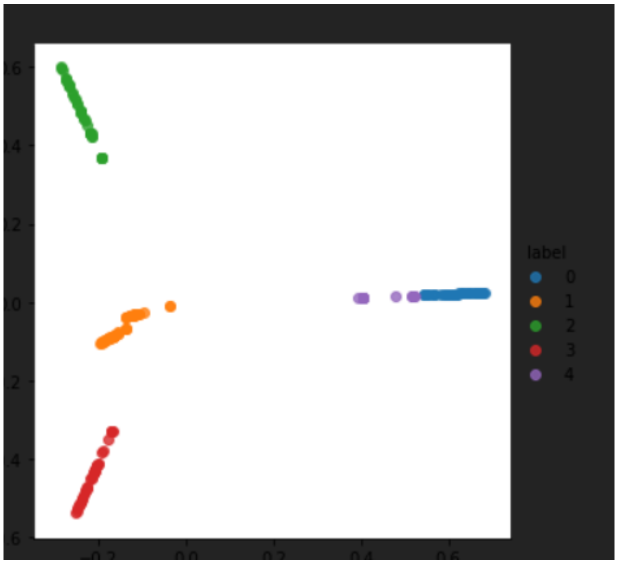
```
import pandas as pd
text1=pd.read_excel('Tweet Excel file.xlsx')
```

Step 3: In this step I had to decide the usage of Algorithms to get relevant output. Before application of algorithms it is very Important to ensure the Data is Cleaned.

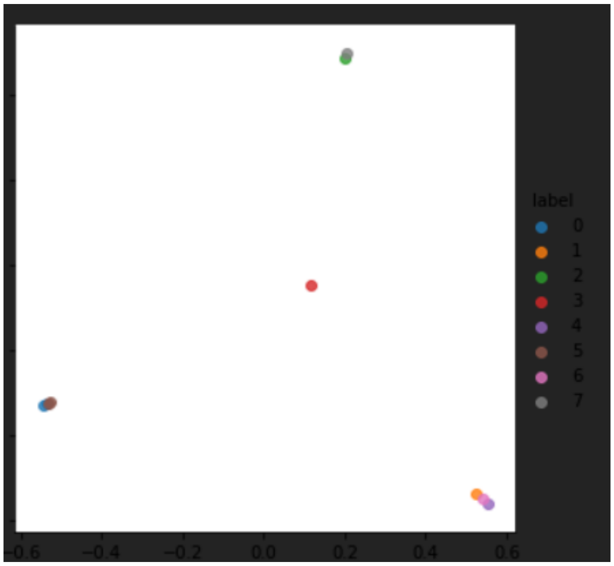
Latent Dirichlet Allocation: I am now using the LDA Algorithm for topic modelling that is used to generate topics based on word frequency from a set of documents. It is useful in finding similarity features from the set of text data.

The Principal Component Analysis: I used this Algorithm for the dimensionality reduction feature**.**

Finally, I used the **K-Means Clustering Algorithm** to display the Desired output.



Output 1



Output 2

'Does the cluster indicate relevant tweets?'

Answer: I have considered both the possible scenarios to answer this question in detail. Below is my answer indicating both the possible output.

Output 1: As seen above the density of output 1 is better and spread over plethora of tweets. This is because I have used 10 pages of tweets rather than only 10 tweets for output 1. The Labels [0-4] indicate the tweets plotted using the K-Means Cluster. As Per the spread the cluster clearly has grouped all relevant tweets together with each label indicating different shade of colour per tweet. Therefore, I conclude the cluster Indicates relevant tweets.

Output 2: As seen in the above screenshot, the density of the tweets is lesser as I am displaying the output for only 10 tweets in output 2. The labels [1-7] Indicate different colors per similar tweets. However, as seen the tweets are not forming a clear/Exact cluster (as seen the colors are mixed among each other indicating tweets are not fully relevant compared to Output 1). Therefore, I conclude the cluster Indicates partially relevant tweets only.

Please Find Attached: Python Jupyter notebook as well as .html file for your reference.