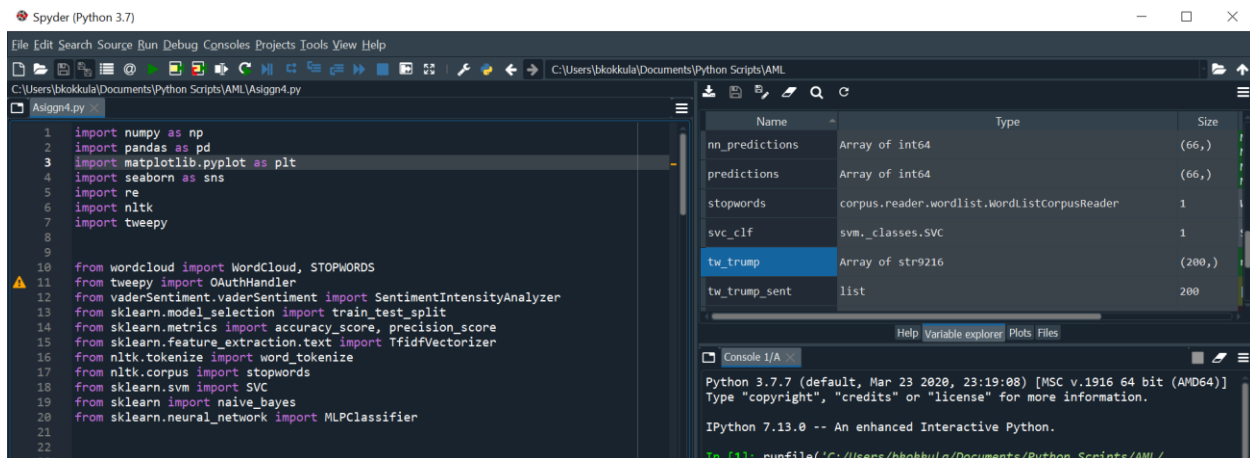


AML Major Assignment1: Twitter Data

Scraping tweets from twitter Sentiment: Analysis is a special case of text classification where users opinions or sentiments regarding a product are classified into predefined categories such as positive, negative, neutral etc. Public sentiments can then be used for corporate decision making regarding a product which is being liked or disliked by the public.

Importing Libraries: Using Python for developing a sentiment analysis model, you need to import the required libraries. The following script does that:



```
1 import numpy as np
2 import pandas as pd
3 import matplotlib.pyplot as plt
4 import seaborn as sns
5 import re
6 import nltk
7 import tweepy
8
9
10 from wordcloud import WordCloud, STOPWORDS
11 from tweepy import OAuthHandler
12 from vaderSentiment.vaderSentiment import SentimentIntensityAnalyzer
13 from sklearn.model_selection import train_test_split
14 from sklearn.metrics import accuracy_score, precision_score
15 from sklearn.feature_extraction.text import TfidfVectorizer
16 from nltk.tokenize import word_tokenize
17 from nltk.corpus import stopwords
18 from sklearn.svm import SVC
19 from sklearn import naive_bayes
20 from sklearn.neural_network import MLPClassifier
```

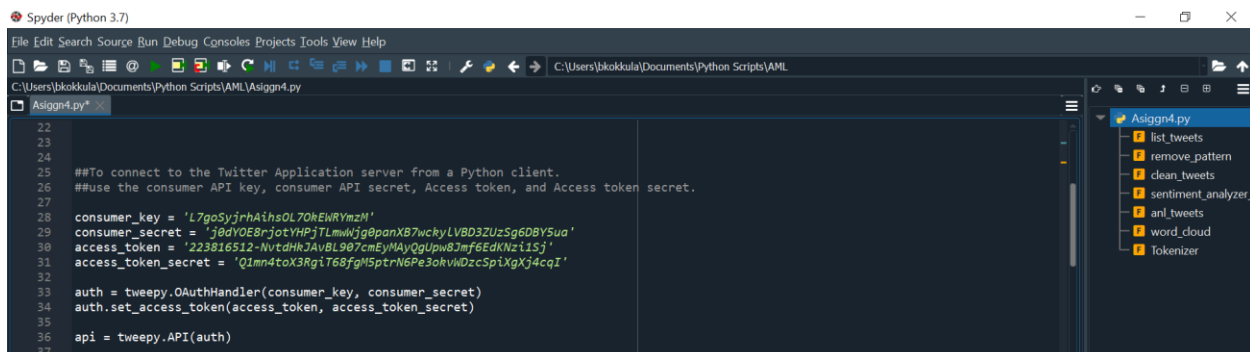
The screenshot shows the Spyder Python IDE with a script named 'Assign4.py'. The script imports various libraries for sentiment analysis, including numpy, pandas, matplotlib, seaborn, re, nltk, tweepy, wordcloud, vaderSentiment, sklearn, and nltk. The variable explorer on the right shows the following variables:

Name	Type	Size
nn_predictions	Array of int64	(66,)
predictions	Array of int64	(66,)
stopwords	corpus.reader.wordlist.WordListCorpusReader	1
svc_clf	svm.classes.SVC	1
tw_trump	Array of str9216	(280,)
tw_trump_sent	list	280

The console output shows the Python version (3.7.7) and the IPython version (7.13.0).

In the script above, we import “Numpy”, “Pandas”, “Matplotlib” “seaborn” “tweepy” “NLTK” and “re” libraries.

Connecting Python Client Application to Twitter Server: To connect to the Twitter Application server from a Python client, use the consumer API key, consumer API secret, Access token, and Access token secret. Execute the following script:



```
22
23
24
25 ##To connect to the Twitter Application server from a Python client.
26 ##Use the consumer API key, consumer API secret, Access token, and Access token secret.
27
28 consumer_key = 'L7goSjyrhAih5OL70kEWRYmZM'
29 consumer_secret = 'j0dYOE8rjotYHPjTLmWkjg8panXB7uchyLV8D3ZUzSg6D8YSua'
30 access_token = '223816512-NvtDhKJAvBL907cmEyMAyQqUpw8Jmf6EdKNzi15j'
31 access_token_secret = 'Q1mn4toX3RgiT68fgMSptrN6Pe3okvWdzCSp1XgXj4cqI'
32
33 auth = tweepy.OAuthHandler(consumer_key, consumer_secret)
34 auth.set_access_token(access_token, access_token_secret)
35
36 api = tweepy.API(auth)
37
```

The screenshot shows the Spyder Python IDE with a script named 'Assign4.py'. The script defines the consumer API key, consumer API secret, Access token, and Access token secret. It then creates an OAuthHandler object and sets the access token and access token secret. Finally, it creates a Tweepy API object using the OAuthHandler object.

The variable explorer on the right shows the following variables:

- list_tweets
- remove_pattern
- clean_tweets
- sentiment_analyzer
- anl_tweets
- word_cloud
- Tokenizer

Scraping Tweets: We have successfully connected to the Twitter API. The next step is to fetch tweets. Next, create an empty list alltweets which will contain the scraped tweets. In the search query specify the string “realDonaldTrump” which means that you want to search the tweets that contain the word “realDonaldTrump”.

```
37 # Creating Twitter List
38
39
40 def list_tweets(user_id, count, prt=False):
41     tweets = api.user_timeline(
42         user_id, count=count, tweet_mode='extended')
43     tw = []
44     for t in tweets:
45         tw.append(t.full_text)
46         if prt:
47             print(t.full_text)
48             print()
49     return tw
50
51 # trump user name in twitter.
52 user_id = 'realDonaldTrump'
53 count=200
54
55
56
```

Once you execute the script above, you will see 200 most recent tweets containing the string “realDonaldTrump” will be stored in the all tweets list and with that, we end the first part of the article.

Performing Vader Sentimental Analysis:

We have scraped live tweets from twitter. To create a vader sentimental analysis model using existing dataset and to use that model to predict sentiments for the 200 tweets that you scraped. Follow these steps to perform sentiment analysis on scraped tweets:

```
77
78 # Performing vader sentiment analysis
79
80
81 analyser = SentimentIntensityAnalyzer()
82 analyser.polarity_scores("The movie is good")
83
84
85 def sentiment_analyzer_scores(text):
86     score = analyser.polarity_scores(text)
87     lb = score['compound']
88     if lb >= 0.05:
89         return 1
90     elif (lb > -0.05) and (lb < 0.05):
91         return 0
92     else:
93
```

Data Preprocessing: we will divide the data into the label and feature set and then will remove special characters and empty spaces from the tweets. Execute the following script to do so:

```
55
56 #Cleaning Twitter Dataset
57
58
59 def remove_pattern(input_txt, pattern):
60     r = re.findall(pattern, input_txt)
61     for i in r:
62         input_txt = re.sub(i, '', input_txt)
63     return input_txt
64
65
66 def clean_tweets(lst):
67     # remove twitter Return handles (RT @xxx:)
68     lst = np.vectorize(remove_pattern)(lst, "RT @[\\w]*:")
69     # remove twitter handles (@xxx)
70     lst = np.vectorize(remove_pattern)(lst, "@[\\w]*")
71     # remove URL links (httpxxx)
72     lst = np.vectorize(remove_pattern)(lst, "https?://[A-Za-z0-9./]*")
73     # remove special characters, numbers, punctuations (except for #)
74     lst = np.core.defchararray.replace(lst, "[^a-zA-Z#]", " ")
75     return lst
76
77
78
```

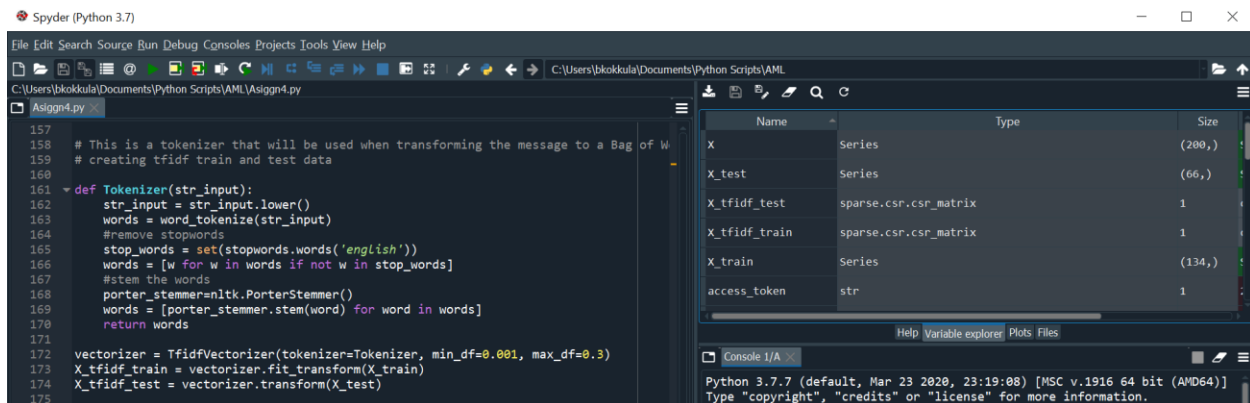
Name	Type	Size
nn_predictions	Array of int64	(66,)
predictions	Array of int64	(66,)
stopwords	corpus.reader.wordlist.WordListCorpusReader	1
svc_clf	svm.classes.SVC	1
tw_trump	Array of str9216	(200,)
tw_trump_sent	list	200

```
Python 3.7.7 (default, Mar 23 2020, 23:19:08) [MSC v.1916 64 bit (AMD64)]
Type "copyright", "credits" or "license()" for more information.

IPython 7.13.0 -- An enhanced Interactive Python.

In [1]: runfile('C:/Users/bhokkula/Documents/Python Scripts/AML/
Assign4.py', wdir='C:/Users/bhokkula/Documents/Python Scripts/AML')
```

TF-IDF for Text to Numeric Conversion and creating transform train and test data sets: You can use the TFIDF scheme to convert text to numbers. The following script does that:

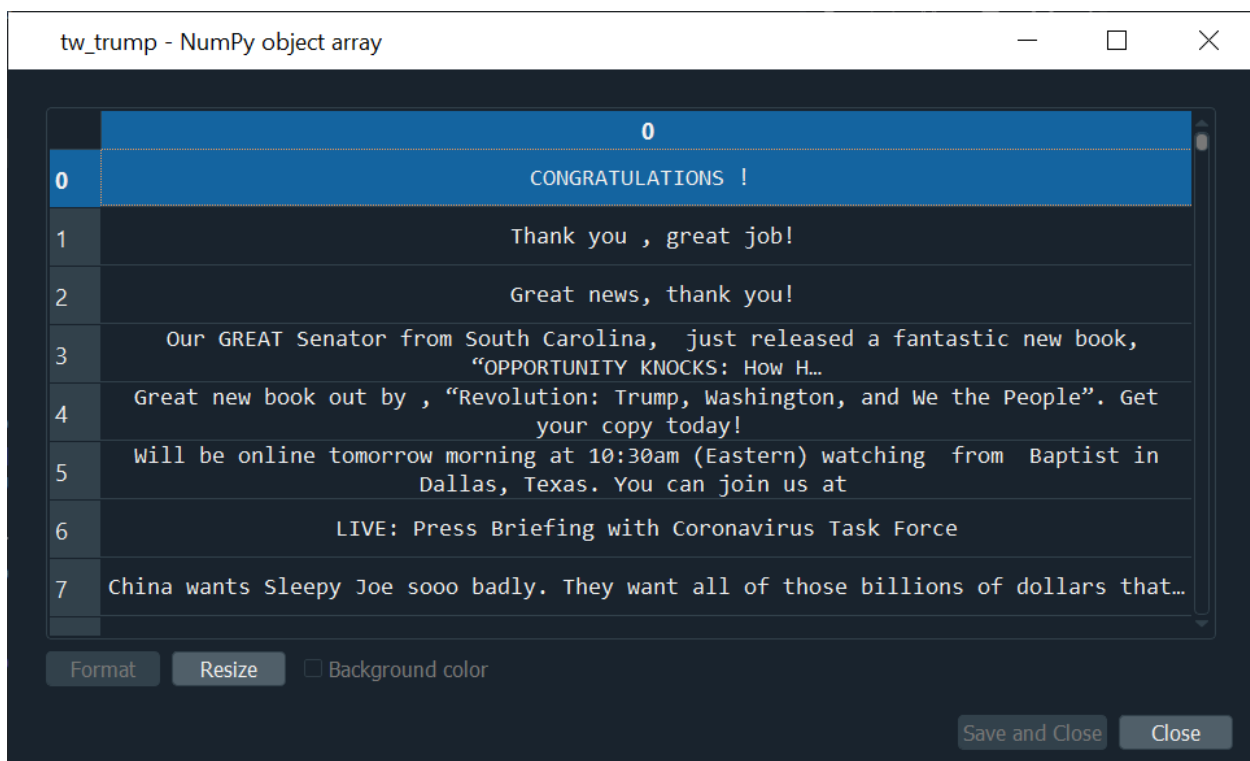


```
157
158 # This is a tokenizer that will be used when transforming the message to a Bag of W
159 # creating tfidf train and test data
160
161 def Tokenizer(str_input):
162     str_input = str_input.lower()
163     words = word_tokenize(str_input)
164     #remove stopwords
165     stop_words = set(stopwords.words('english'))
166     words = [w for w in words if not w in stop_words]
167     #stem the words
168     porter_stemmer=nltk.PorterStemmer()
169     words = [porter_stemmer.stem(word) for word in words]
170     return words
171
172 vectorizer = TfidfVectorizer(tokenizer=Tokenizer, min_df=0.001, max_df=0.3)
173 X_tfidf_train = vectorizer.fit_transform(X_train)
174 X_tfidf_test = vectorizer.transform(X_test)
175
```

Name	Type	Size
X	Series	(200,)
X_test	Series	(66,)
X_tfidf_test	sparse.csr.csr_matrix	1
X_tfidf_train	sparse.csr.csr_matrix	1
X_train	Series	(134,)
access_token	str	1

Console 1/A
Python 3.7.7 (default, Mar 23 2020, 23:19:08) [MSC v.1916 64 bit (AMD64)]
Type "copyright", "credits" or "license" for more information.

In the output, you will see each of the 200 scraped tweets along with its sentiment. A screenshot of the output from the Spyder console is shown below:

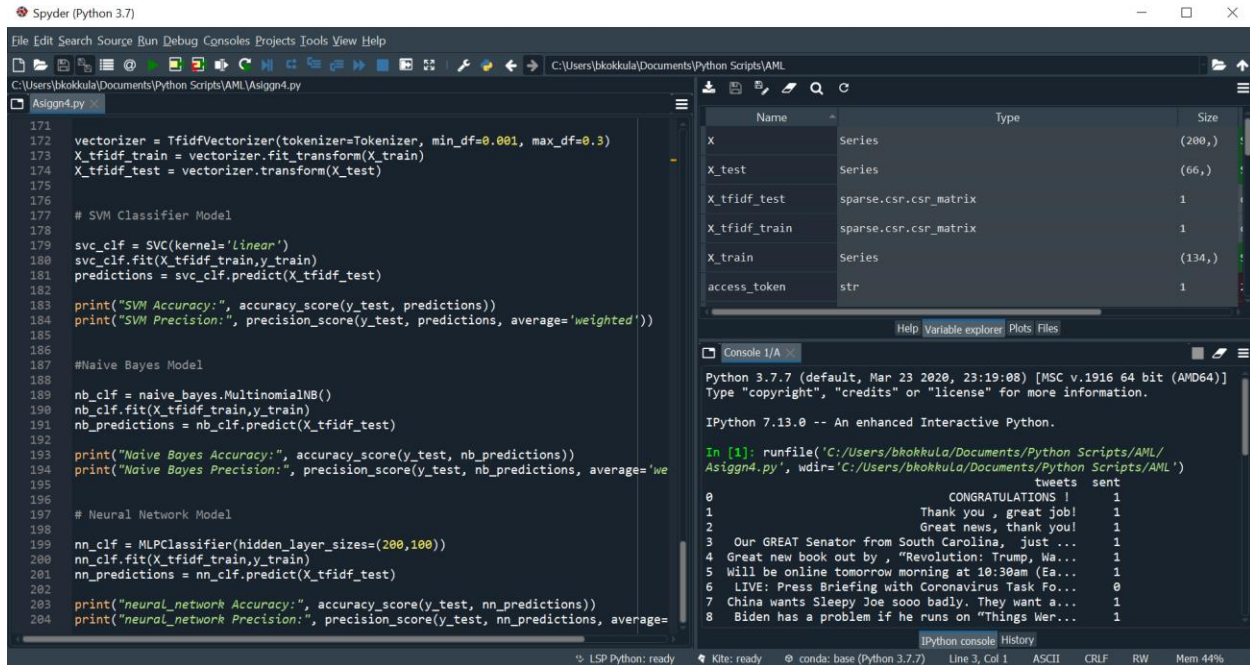


tw_trump - NumPy object array

	0
0	CONGRATULATIONS !
1	Thank you , great job!
2	Great news, thank you!
3	Our GREAT Senator from South Carolina, just released a fantastic new book, "OPPORTUNITY KNOCKS: How H...
4	Great new book out by , "Revolution: Trump, Washington, and We the People". Get your copy today!
5	Will be online tomorrow morning at 10:30am (Eastern) watching from Baptist in Dallas, Texas. You can join us at
6	LIVE: Press Briefing with Coronavirus Task Force
7	China wants Sleepy Joe sooo badly. They want all of those billions of dollars that...

Format Resize Background color Save and Close Close

The performance of algorithms on the scraped tweets and see output from below:



```
171
172 vectorizer = TfidfVectorizer(tokenizer=Tokenizer, min_df=0.001, max_df=0.3)
173 X_tfidf_train = vectorizer.fit_transform(X_train)
174 X_tfidf_test = vectorizer.transform(X_test)
175
176
177 # SVM Classifier Model
178
179 svc_clf = SVC(kernel='linear')
180 svc_clf.fit(X_tfidf_train, y_train)
181 predictions = svc_clf.predict(X_tfidf_test)
182
183 print("SVM Accuracy:", accuracy_score(y_test, predictions))
184 print("SVM Precision:", precision_score(y_test, predictions, average='weighted'))
185
186
187 #Naive Bayes Model
188
189 nb_clf = naive_bayes.MultinomialNB()
190 nb_clf.fit(X_tfidf_train, y_train)
191 nb_predictions = nb_clf.predict(X_tfidf_test)
192
193 print("Naive Bayes Accuracy:", accuracy_score(y_test, nb_predictions))
194 print("Naive Bayes Precision:", precision_score(y_test, nb_predictions, average='we
195
196
197 # Neural Network Model
198
199 nn_clf = MLPClassifier(hidden_layer_sizes=(200,100))
200 nn_clf.fit(X_tfidf_train, y_train)
201 nn_predictions = nn_clf.predict(X_tfidf_test)
202
203 print("neural_network Accuracy:", accuracy_score(y_test, nn_predictions))
204 print("neural_network Precision:", precision_score(y_test, nn_predictions, average=
```

Name	Type	Size
X	Series	(200,)
X_test	Series	(66,)
X_tfidf_test	sparse.csr.csr_matrix	1
X_tfidf_train	sparse.csr.csr_matrix	1
X_train	Series	(134,)
access_token	str	1

```
Python 3.7.7 (default, Mar 23 2020, 23:19:08) [MSC v.1916 64 bit (AMD64)]
Type "copyright", "credits" or "license" for more information.

IPython 7.13.0 -- An enhanced Interactive Python.

In [1]: runfile('C:/Users/bkakkula/Documents/Python Scripts/AML/
Assign4.py', wdir='C:/Users/bkakkula/Documents/Python Scripts/AML')

tweets sent
0          CONGRATULATIONS !          1
1          Thank you , great job!      1
2          Great news, thank you!      1
3          Our GREAT Senator from South Carolina, just ... 1
4          Great new book out by , "Revolution: Trump, Wa... 1
5          Will be online tomorrow morning at 10:30am (Ea... 1
6          LIVE: Press Briefing with Coronavirus Task Fo... 0
7          China wants Sleepy Joe sooo badly. They want a... 1
8          Biden has a problem if he runs on "Things Wer... 1
9          With all the grounded flights and ghosted air... 1

SVM Accuracy: 0.6363636363636364
SVM Precision: 0.6207285622179239
Naive Bayes Accuracy: 0.6212121212121212
Naive Bayes Precision: 0.7037337662337662
neural_network Accuracy: 0.696969696969697
neural_network Precision: 0.7344389844389845
```

Output Results: SVM Accuracy: 0.6363636363636364;

SVM Precision: 0.6207285622179239

Naive Bayes Accuracy: 0.6212121212121212

Naive Bayes Precision: 0.7037337662337662

neural_network Accuracy: 0.696969696969697

neural_network Precision: 0.7344389844389845

Conclusion: The sentimental analysis is one of the most important tasks in corporate decision making. Being aware of the public sentiment about a product can play a crucial role in the success or failure of the product. By see above accuracy of each model the neural network model performed best and followed by SVM model and NB model.