Group C 1 (Mini Project)

Aim:

Use the following dataset and classify tweets into positive and negative tweets. https://www.kaggle.com/ruchi798/data-science-tweet

----

Requirement:

- Anaconda Installer
- Windows 10 OS
- Jupyter Notebook

Theory:

What is Sentiment Analysis?

Sentiment Analysis is the process of 'computationally' determining whether a piece of writing is positive, negative or neutral. It's also known as **opinion mining**, deriving the opinion or attitude of a speaker.

Why Sentiment Analysis?

- Business: In marketing field companies use it to develop their strategies, to understand customers' feelings towards products or brand, how people respond to their campaigns or product launches and why consumers don't buy some products.
- Politics: In political field, it is used to keep track of political view, to detect consistency and inconsistency between statements and actions at the government level. It can be used to predict election results as well!
- Public Actions: Sentiment analysis also is used to monitor and analyse social phenomena, for the spotting of potentially dangerous situations and determining the general mood of the blogosphere.

Libraries Used:

1. Pandas: Pandas is an open source Python package that is most widely used for data science/data analysis and machine learning tasks. It is built on top of another package named Numpy, which provides support for multi-dimensional arrays.

2. String: The string module contains a number of functions to process standard Python strings.

3. Sklearn: It provides a selection of efficient tools for machine learning and statistical modeling including classification, regression, clustering and dimensionality reduction via a consistence interface in Python.

Conslusion:

Hence, we successfully implemented sentiment analysis using python.

In [1]:
```
import pandas as pd
```

In [4]:
```
df=pd.read_csv('tweets.csv')
```

In [5]:
```
df.shape
```

Out[5]:
```
(31962, 3)
```

In [6]:
```
df
```

Out[6]:

	id	label	tweet
0	1	0	@user when a father is dysfunctional and is s...
1	2	0	@user @user thanks for #lyft credit i can't us...
2	3	0	bihday your majesty
3	4	0	#model i love u take with u all the time in ...
4	5	0	factsguide: society now #motivation
...	...	...	...
31957	31958	0	ate @user isz that youuu?ð ð ð ð ð ð...
31958	31959	0	to see nina turner on the airwaves trying to...
31959	31960	0	listening to sad songs on a monday morning otw...
31960	31961	1	@user #sikh #temple vandalised in in #calgary,...
31961	31962	0	thank you @user for you follow

31962 rows × 3 columns

In [7]:
```
df=pd.read_csv('tweets.csv',nrows=10000)
```

In [8]:
```
df
```

Out[8]:

	id	label	tweet
0	1	0	@user when a father is dysfunctional and is s...
1	2	0	@user @user thanks for #lyft credit i can't us...
2	3	0	bihday your majesty
3	4	0	#model i love u take with u all the time in ...
4	5	0	factsguide: society now #motivation
...	...	...	...
9995	9996	0	@user my routine is out of whack! evening wal...
9996	9997	0	i'm dead but still happy #poledance #madrid ##...
9997	9998	0	â #united kingdom claimant count rate up to...

	id	label	tweet
9998	9999	0	rip my friend ð ¢ð ¢ #shocked #dismay #hea...
9999	10000	0	how to open... your , loving hea #thursdayth...

10000 rows × 3 columns

In [9]:
```
df.shape
```

Out[9]:
```
(10000, 3)
```

In [10]:
```
df['tweets_len']=df['tweet'].apply(lambda x : len(x))
```

In [11]:
```
df
```

Out[11]:

	id	label	tweet	tweets_len
0	1	0	@user when a father is dysfunctional and is s...	102
1	2	0	@user @user thanks for #lyft credit i can't us...	122
2	3	0	bihday your majesty	21
3	4	0	#model i love u take with u all the time in ...	86
4	5	0	factsguide: society now #motivation	39
...	...	...	...	...
9995	9996	0	@user my routine is out of whack! evening wal...	120
9996	9997	0	i'm dead but still happy #poledance #madrid ##...	90
9997	9998	0	â #united kingdom claimant count rate up to...	106
9998	9999	0	rip my friend ð ¢ð ¢ #shocked #dismay #hea...	102
9999	10000	0	how to open... your , loving hea #thursdayth...	78

10000 rows × 4 columns

In [65]:
```
sent='Hii , Where are you ?'
```

In [66]:
```
import string
```

In [67]:
```
string.punctuation
```

Out[67]:
```
'!"#$%&\'()*+,-./:;<=>?@[\\]^_`{|}~'
```

In [68]:
```
count=sum([1 for x in sent if x in string.punctuation])
```

In [69]:
```
per=count/(len(sent)-sent.count(' '))
```

```
In [70]: per

Out[70]: 0.125
```

```
In [71]: import string
```

```
In [72]: string.punctuation

Out[72]: '!"#$%&\'()*+,-./:;<=>?@[\\]^_`{|}~'
```

```
In [73]: def count_punct(sent):
             count =sum([1 for x in sent if x in string.punctuation])
             p=round(count/(len(sent)-sent.count(' '))*100,2)
             return p
```

```
In [74]: count_punct(sent)

Out[74]: 12.5
```

```
In [75]: df['punct%']=df['tweet'].apply(lambda x:count_punct(x))
```

```
In [84]: from nltk.corpus import stopwords
         from nltk.stem import PorterStemmer
         ps=PorterStemmer()
```

```
In [85]: s_words=stopwords.words('english')
```

```
In [86]: #analyser funtion
         def clean_text(text):
             data=[x for x in text if x not in string.punctuation]
             data=''.join(data)
             data=[ps.stem(x) for x in data.split() if x not in s_words]
             return data
```

```
In [87]: clean_text(sent)

Out[87]: ['hii', 'where']
```

```
In [88]: # inputdata
         X=df.drop(['label','id'],axis=1)
         # output data
         y=df['label']
```

```
In [89]: X

Out[89]:
                                          tweet   tweets_len   punct%
   0   @user when a father is dysfunctional and is s...          102     3.66
```

	tweet	tweets_len	punct%
1	@user @user thanks for #lyft credit i can't us...	122	7.92
2	bihday your majesty	21	0.00
3	#model i love u take with u all the time in ...	86	5.71
4	factsguide: society now #motivation	39	6.25
...	...	...	...
9995	@user my routine is out of whack! evening wal...	120	11.22
9996	i'm dead but still happy #poledance #madrid ##...	90	11.84
9997	â #united kingdom claimant count rate up to...	106	10.47
9998	rip my friend ð ¢ð ¢ #shocked #dismay #hea...	102	7.95
9999	how to open... your , loving hea #thursdayth...	78	10.61

10000 rows × 3 columns

In [90]:

```
y
```

Out[90]:

```
0       0
1       0
2       0
3       0
4       0
       ..
9995    0
9996    0
9997    0
9998    0
9999    0
Name: label, Length: 10000, dtype: int64
```

In [92]:

```
from sklearn.feature_extraction.text import TfidfVectorizer
tfidf=TfidfVectorizer(analyzer=clean_text)
X_trans=tfidf.fit_transform(X['tweet'])
```

In [93]:

```
X_trans.shape
```

Out[93]:

```
(10000, 18712)
```

In [97]:

```
X_vect=pd.concat([X[['tweets_len','punct%']].reset_index(drop=True),pd.DataFrame(X_t
```

In [98]:

```
from sklearn.model_selection import train_test_split
X_train,X_test,y_train,y_test=train_test_split(X_vect,y,stratify=y,random_state=0)
```

In [99]:

```
from sklearn.linear_model import LogisticRegression
clf=LogisticRegression()
clf.fit(X_train,y_train)
```

```
C:\Users\ganes\anaconda3\lib\site-packages\sklearn\linear_model\_logistic.py:763: Co
nvergenceWarning: lbfgs failed to converge (status=1):
STOP: TOTAL NO. of ITERATIONS REACHED LIMIT.
```

```
Increase the number of iterations (max_iter) or scale the data as shown in:
    https://scikit-learn.org/stable/modules/preprocessing.html
Please also refer to the documentation for alternative solver options:
    https://scikit-learn.org/stable/modules/linear_model.html#logistic-regression
  n_iter_i = _check_optimize_result(
```

Out[99]:

```
LogisticRegression()
```

In [101…

```
y_pred=clf.predict(X_test)
```

In [102…

```
from sklearn.metrics import accuracy_score
```

In [103…

```
accuracy_score(y_test,y_pred)
```

Out[103…

```
0.9336
```

In [104…

```
accuracy_score(y_test,y_pred)*100
```

Out[104…

```
93.36
```

In [ ]:

```
```