```
! pip install kaggle
```

```
Requirement already satisfied: kaggle in /usr/local/lib/python3.10/dist-packages (1.6.14)
Requirement already satisfied: six>=1.10 in /usr/local/lib/python3.10/dist-packages (from kaggle) (1.16.0)
Requirement already satisfied: certifi>=2023.7.22 in /usr/local/lib/python3.10/dist-packages (from kaggle) (2024.2.2)
Requirement already satisfied: python-dateutil in /usr/local/lib/python3.10/dist-packages (from kaggle) (2.8.2)
Requirement already satisfied: requests in /usr/local/lib/python3.10/dist-packages (from kaggle) (2.31.0)
Requirement already satisfied: tqdm in /usr/local/lib/python3.10/dist-packages (from kaggle) (4.66.4)
Requirement already satisfied: python-slugify in /usr/local/lib/python3.10/dist-packages (from kaggle) (8.0.4)
Requirement already satisfied: urllib3 in /usr/local/lib/python3.10/dist-packages (from kaggle) (2.0.7)
Requirement already satisfied: bleach in /usr/local/lib/python3.10/dist-packages (from kaggle) (6.1.0)
Requirement already satisfied: webencodings in /usr/local/lib/python3.10/dist-packages (from bleach->kaggle) (0.5.1)
Requirement already satisfied: text-unidecode>=1.3 in /usr/local/lib/python3.10/dist-packages (from python-slugify->kaggle) (1.3)
Requirement already satisfied: charset-normalizer<4,>=2 in /usr/local/lib/python3.10/dist-packages (from requests->kaggle) (3.3.2)
Requirement already satisfied: idna<4,>=2.5 in /usr/local/lib/python3.10/dist-packages (from requests->kaggle) (3.7)
```

```
!mkdir -p ~/.kaggle
!cp kaggle.json ~/.kaggle/
!chmod 600 ~/.kaggle/kaggle.json
```

```
cp: cannot stat 'kaggle.json': No such file or directory
chmod: cannot access '/root/.kaggle/kaggle.json': No such file or directory
```

```
!kaggle datasets download -d kazanova/sentiment140
```

```
Dataset URL: https://www.kaggle.com/datasets/kazanova/sentiment140
License(s): other
Downloading sentiment140.zip to /content
 70% 57.0M/80.9M [00:00<00:00, 174MB/s]
100% 80.9M/80.9M [00:00<00:00, 182MB/s]
```

```
from zipfile import ZipFile
dataset ='/content/sentiment140.zip'

with ZipFile(dataset,'r')as zip:
  zip.extractall()
  print('the dataset is extracted')
```

```
the dataset is extracted
```

```
import numpy as np
import pandas as pd
import re
from nltk.corpus import stopwords
from nltk.stem.porter import PorterStemmer
from sklearn.feature_extraction.text import TfidfVectorizer
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import accuracy_score
```

```
import nltk
nltk.download('stopwords')
```

```
[nltk_data] Downloading package stopwords to /root/nltk_data...
[nltk_data]   Unzipping corpora/stopwords.zip.
True
```

```
print(stopwords.words('english'))
```

```
['i', 'me', 'my', 'myself', 'we', 'our', 'ours', 'ourselves', 'you', "you're", "you've", "you'll", "you'd", 'your', 'yours', 'yourse
```

```
twitter_data=pd.read_csv('/content/training.1600000.processed.noemoticon.csv',encoding = 'ISO-8859-1')
```

```
twitter_data.shape
```

```
(1599999, 6)
```

```
twitter_data.head()
```

	0	1467810369	Mon Apr 06 22:19:45 PDT 2009	NO_QUERY	_TheSpecialOne_	@switchfoot http://twitpic.com/2y1zl - Awww, that's a bummer. You shoulda got David Carr of Third Day to do it. ;D
0	0	1467810672	Mon Apr 06 22:19:49 PDT 2009	NO_QUERY	scotthamilton	is upset that he can't update his Facebook by ...
1	0	1467810917	Mon Apr 06 22:19:53	NO_QUERY	mattycus	@Kenichan I dived many times for the ball. Man

```
column_names =['target','id','data','flag','user','text']
twitter_data=pd.read_csv('/content/training.1600000.processed.noemoticon.csv',names=column_names,encoding = 'ISO-8859-1')
```

```
twitter_data.shape
```

```
(1600000, 6)
```

```
twitter_data.head()
```

	target	id	data	flag	user	text
0	0	1467810369	Mon Apr 06 22:19:45 PDT 2009	NO_QUERY	_TheSpecialOne_	@switchfoot http://twitpic.com/2y1zl - Awww, t...
			Mon Apr 06			is upset that he can't

```
twitter_data.isnull().sum()
```

```
target    0
id        0
data      0
flag      0
user      0
text      0
dtype: int64
```

```
twitter_data['target'].value_counts()
```

```
target
0    800000
4    800000
Name: count, dtype: int64
```

```
twitter_data.replace({'target':{4:1}}, inplace=True)
```

```
twitter_data['target'].value_counts()
```

```
target
0    800000
1    800000
Name: count, dtype: int64
```

0 ---> negative tweet 1---> positive tweet

stemming: stemming is the process of reducing a word to its root word

```
port_stem = PorterStemmer()
```

```
def stemming(content):

  stemmed_content=re.sub('[^a-zA-Z]','',content)
  stemmed_content=stemmed_content.lower()
  stemmed_content=stemmed_content.split()
  stemmed_content= [port_stem.stem(word) for word in stemmed_content if not word in stopwords.words('english')]
  stemmed_content= ''.join(stemmed_content)

  return stemmed_content
```

```
twitter_data['stemmed_content'] =twitter_data['text'].apply(stemming)
```

```
twitter_data.head()
```

	target	id	data	flag	user	text	
0	0	1467810369	Mon Apr 06 22:19:45 PDT 2009	NO_QUERY	_TheSpecialOne_	@switchfoot http://twitpic.com/2y1zl - Awww, t...	switc
1	0	1467810672	Mon Apr 06 22:19:49 PDT 2009	NO_QUERY	scotthamilton	is upset that he can't update his Facebook by ...	is
			Mon Apr				

```
print(twitter_data['stemmed_content'])
```

```
0          switchfoothttptwitpiccomyzlawwwthatsabummeryou...
1          isupsetthathecantupdatehisfacebookbytextingita...
2          kenichanidivedmanytimesfortheballmanagedtosave...
3                       mywholebodyfeelsitchyandlikeitsonfir
4          nationwideclassnoitsnotbehavingatallimmadwhyam...
                                 ...
1599995            justwokeuphavingnoschoolisthebestfeelingev
1599996    thewdbcomverycooltohearoldwaltinterviewshttpbl...
1599997         areyoureadyforyourmojomakeoveraskmefordetail
1599998     happythbirthdaytomybooofallltimetupacamarushakur
1599999    happycharitytuesdaythenspccsparkscharityspeaki...
Name: stemmed_content, Length: 1600000, dtype: object
```

```
print(twitter_data['target'])
```

```
0          0
1          0
2          0
3          0
4          0
          ..
1599995    1
1599996    1
1599997    1
1599998    1
1599999    1
Name: target, Length: 1600000, dtype: int64
```

```
# separating the data and label
X=twitter_data['stemmed_content'].values
Y=twitter_data['target'].values
```

```
print(X)
```

```
['switchfoothttptwitpiccomyzlawwwthatsabummeryoushouldagotdavidcarrofthirddaytodoitd'
 'isupsetthathecantupdatehisfacebookbytextingitandmightcryasaresultschooltodayalsoblah'
 'kenichanidivedmanytimesfortheballmanagedtosavetherestgooutofbound' ...
 'areyoureadyforyourmojomakeoveraskmefordetail'
 'happythbirthdaytomybooofallltimetupacamarushakur'
 'happycharitytuesdaythenspccsparkscharityspeakinguphh']
```

```
print(Y)
```

```
[0 0 0 ... 1 1 1]
```

SPITTING the data to training data and test data

```
X_train,X_test,Y_train,Y_test = train_test_split(X,Y, test_size=0.2,stratify=Y,random_state=2)
```

```
print(X.shape,X_train.shape,X_test.shape)
```

```
(1600000,) (1280000,) (320000,)
```

```
print(X_train)
```

```
['abouttowatchsawivanddrinkalilwin' 'hatermagazineimin'
 'eventhoughitsmyfavouritedrinkithinkitsthevodkaandcokethatwipesmymindallthetimethinkimgonnahavetofindanewdrink'
 ... 'iseagerformondayafternoon'
 'hopeeveryoneandtheirmotherhadagreatdaycantwaittohearwhattheguyshaveinstoretomorrow'
 'ilovewakinguptofolgerstoobadmyvoicewasdeeperthanhi']
```

```
print(X_test)
```

```
['mmangenmdoingfineihaventhadmuchtimetochatontwitterhubbyisbackforthesummeramptendstodominatemyfreetim'
 'atahsmayshowwruthkimampgeoffreysanhueza'
 'ishataramaybeitwasonlyabayareathangdammit' ...
 'destinineverthelesshoorayformembersandhaveawonderfulandsafetrip'
 'notfeelingtoowel' 'supersandrothanky']
```

```
#converting the textual data to numerical data
vectorizer = TfidfVectorizer()
X_train = vectorizer.fit_transform (X_train)
X_test = vectorizer.fit_transform(X_test)
```

```
print(X_train)
```

```
  (0, 4269)     1.0
  (1, 370478)   1.0
  (2, 259878)   1.0
  (3, 544662)   1.0
  (4, 1125692)  1.0
  (5, 183084)   1.0
  (6, 655943)   1.0
  (7, 427969)   1.0
  (8, 372293)   1.0
  (9, 414371)   1.0
  (10, 453206)  1.0
  (11, 946311)  1.0
  (12, 1165280) 1.0
  (13, 334893)  1.0
  (14, 1029544) 1.0
  (15, 1251389) 1.0
  (16, 26694)   1.0
  (17, 620642)  1.0
  (18, 371055)  1.0
  (19, 166384)  1.0
  (20, 939661)  1.0
  (21, 392755)  1.0
  (22, 288391)  1.0
  (23, 202689)  1.0
  (24, 1253472) 1.0
  :     :
  (1279975, 97244)      1.0
  (1279976, 1110776)    1.0
  (1279977, 938969)     1.0
  (1279978, 783480)     1.0
  (1279979, 800934)     1.0
  (1279980, 865418)     1.0
  (1279981, 587370)     1.0
  (1279982, 952975)     1.0
  (1279983, 412833)     1.0
  (1279984, 81684)      1.0
  (1279985, 59066)      1.0
  (1279986, 1015971)    1.0
  (1279987, 777243)     1.0
  (1279988, 1218709)    1.0
  (1279989, 450972)     1.0
  (1279990, 465151)     1.0
  (1279991, 249222)     1.0
  (1279992, 1169847)    1.0
  (1279993, 1170137)    1.0
  (1279994, 75555)      1.0
  (1279995, 1106989)    1.0
  (1279996, 561111)     1.0
  (1279997, 516692)     1.0
  (1279998, 400780)     1.0
```

```
(1279999, 469313)     1.0
```

```
print(X_test)
```

```
(0, 193539)   1.0
(1, 15208)    1.0
(2, 131537)   1.0
(3, 75524)    1.0
(4, 106316)   1.0
(5, 108758)   1.0
(6, 127957)   1.0
(7, 301475)   1.0
(8, 199386)   1.0
(9, 112690)   1.0
(10, 32110)   1.0
(11, 74325)   1.0
(12, 33926)   1.0
(13, 215744)  1.0
(14, 304704)  1.0
(15, 146801)  1.0
(16, 251714)  1.0
(17, 35333)   1.0
(18, 304368)  1.0
(19, 54669)   1.0
(20, 171971)  1.0
(21, 282298)  1.0
(22, 34327)   1.0
(23, 313179)  1.0
(24, 280164)  1.0
:     :
(319975, 173561)      1.0
(319976, 151782)      1.0
(319977, 27990)       1.0
(319978, 248013)      1.0
(319979, 98636)       1.0
(319980, 315946)      1.0
(319981, 137258)      1.0
(319982, 243642)      1.0
(319983, 116305)      1.0
(319984, 152248)      1.0
(319985, 14252)       1.0
(319986, 50822)       1.0
(319987, 306608)      1.0
(319988, 315360)      1.0
(319989, 2060)        1.0
(319990, 126089)      1.0
(319991, 303395)      1.0
(319992, 143608)      1.0
(319993, 156126)      1.0
(319994, 98973)       1.0
(319995, 278420)      1.0
(319996, 269151)      1.0
(319997, 52661)       1.0
(319998, 210896)      1.0
(319999, 265750)      1.0
```

TRAINING THE MACHINE LEARNING MODEL

- LOGISTIC REGRESSION

```
model = LogisticRegression(max_iter=1000)
```

```
model.fit(X_train, Y_train)
```

```
▾ LogisticRegression
LogisticRegression(max_iter=1000)
```

model evalution

accuracy score

```
X_train_prediction =model.predict(X_train)
train_data_accuracy =accuracy_score(Y_train, X_train_prediction)
```

```
print('accuracy score on the training data:', train_data_accuracy)
```

```
accuracy score on the training data: 0.99812265625
```

```
import pickle
```

```
filename = 'trained_model.sav'
pickle.dump(model,open(filename,'wb'))
```

```
loaded_model = pickle.load(open('/content/trained_model.sav', 'rb'))
```

```
X_new =X_train[200]
print(Y_train[200])

prediction = model.predict(X_new)
print(prediction)

if (prediction[0]==0):
  print('negative tweet')

else:
  print('positive tweet')
```

```
1
[1]
positive tweet
```

```
X_new =X_train[45]
print(Y_train[45])

prediction = loaded_model.predict(X_new)
print(prediction)

if (prediction[0]==0):
  print('negative tweet')

else:
  print('positive tweet')
```

```
0
[0]
negative tweet
```