```
! 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) (2024.6.2)
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)
```

upload your kaggle json file

```
#configure the path of kaggle.json file
!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
```

importing twitter sentiment dataset

```
#Api to fetch the dataset from kaggle
!kaggle datasets download -d kazanova/sentiment140
```

```
Dataset URL: https://www.kaggle.com/datasets/kazanova/sentiment140
License(s): other
sentiment140.zip: Skipping, found more recently modified local copy (use --force to force download)
```

```
#unzip the dataset
from zipfile import ZipFile
data_set = '/content/sentiment140.zip'
with ZipFile(data_set, 'r') as zip:
  zip.extractall()
  print("Done")
```

```
Done
```

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
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]   Package stopwords is already up-to-date!
True
```

```
#printing the stopword in English
print(stopwords.words('english'))
```

```
['i', 'me', 'my', 'myself', 'we', 'our', 'ours', 'ourselves', 'you', "you're", "you've", "you'll", "you'd", 'your', 'yours', 'yourself', 'yourselves', 'he', 'him', 'his', 'him
```

data processing

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

```
#Checking the dataset
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 PDT 2009	NO_QUERY	mattycus	@Kenichan I dived many times for the ball. Man...
2	0	1467811184	Mon Apr 06 22:19:57 PDT 2009	NO_QUERY	ElleCTF	my whole body feels itchy and like its on fire
			Mon Apr 06 22:19:57 PDT			

```
# naming the columns and readin the data_set
column_names = ['target', 'id', 'date','flag', 'user', 'text']
```

```
twitter_data.columns = column_names
```

```
twitter_data.head()
```

	target	id	date	flag	user	text
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 PDT 2009	NO_QUERY	mattycus	@Kenichan I dived many times for the ball. Man...
2	0	1467811184	Mon Apr 06 22:19:57 PDT 2009	NO_QUERY	ElleCTF	my whole body feels itchy and like its on fire
3	0	1467811193	Mon Apr 06 22:19:57 PDT 2009	NO_QUERY	Karoli	@nationwideclass no, it's not behaving at all....
4	0	1467811372	Mon Apr 06 22:20:00 PDT 2009	NO_QUERY	joy_wolf	@Kwesidei not the whole crew

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

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

```
#checking the distribution of the data_set
twitter_data['target'].value_counts()
```

```
target
4    800000
0    799999
Name: count, dtype: int64
```

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

```
#checking the distribution of the data_set
twitter_data['target'].value_counts()
```

```
target
1    800000
0    799999
Name: count, dtype: int64
```

0 -----> Negative Tweets 1 -----> Positive tweets

stemming

```
port_stem = PorterStemmer()
```

```
def stemming(content):

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

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

```
twitter_data.head()
```

```
# seprating the data and labels
X = twitter_data['stremmed_content'].values
Y = twitter_data['target'].values
```

```
print(X)
```

splitting the data into train_test split

```
X_train, X_test, Y_train, Y_test = train_test_split(X, Y, test_size=0.2, random_state=3)
```

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

```
print(Y.shape, Y_train.shape, Y_test.shape)
```

Convert the textual data to numerical data

```
vectorizer = TfidfVectorizer()
X_train = vectorizer.fit_transform(X_train)
X_test = vectorizer.transform(X_test)
```

traning the machine learning model

```
model = LogisticRegression(max_iter=1000)
model.fit(X_train, Y_train)
```

Accuraccy Score

```
x_train_prediction = model.predict(X_train)
training_data_accuracy = accuracy_score(x_train_prediction, Y_train)
```

```
print('Accuracy on training data : ', training_data_accuracy)
```

Accuracy score is 0.81

```
X_test_prediction = model.predict(X_test)
test_data_accuracy = accuracy_score(X_test_prediction, Y_test)
```

```
print(test_data_accuracy)
```

model accuracy is 77.8%

```
import pickle
```

```
file_name = 'sentiment_analysis_model.sav'
pickle.dump(model, open(file_name, 'wb'))
```

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

```
X_new = X_test[200]
print(Y_test[200])
prediction = loaded_model.predict(X_new)
print(prediction)
if (prediction[0]==0):
  print('Negative Tweet')
else:
  print('Positive Tweet')
```