```
#import the library
import numpy as np
import matplotlib.pyplot as plt
import pandas as pd

#read the data set
data=pd.read_csv(r"C:\Users\sunil\Desktop\DK\vs code\sentiment prediction by text\Restaurant_Review

import re
import nltk
from nltk.corpus import stopwords # for stopwords
from nltk.stem.porter import PorterStemmer # for stem the words

# blank cor[]
corpus=[]

# take to proper format
for i in range(0,1000):
    review = re.sub('[^a-zA-Z]', ' ', data["Review"][i])
    review = review.lower()
    review = review.split()
    ps=PorterStemmer()
    #review = [ps.stem(word) for word in review if not word in set(stopwords.words('english'))]
    review = ' '.join(review)
    corpus.append(review)


# Creating the TF-IDF model
from sklearn.feature_extraction.text import TfidfVectorizer
cv = TfidfVectorizer()
X = cv.fit_transform(corpus).toarray()
y=data.iloc[:,1].values

'''
# Creating the Bag of Words model
from sklearn.feature_extraction.text import CountVectorizer
cv = CountVectorizer()
X = cv.fit_transform(corpus).toarray()
y=data.iloc[:,1].values
'''

from sklearn.model_selection import train_test_split
x_train,x_test,y_train,y_test=train_test_split(X,y,test_size=0.15,random_state=0)
'''
#1 decission tree
from sklearn.tree import DecisionTreeClassifier
classifier=DecisionTreeClassifier()
classifier.fit(x_train,y_train)

#2 svc classifier
from sklearn.svm import SVC
classifier=SVC()
classifier.fit(x_train,y_train)

#3 knn classifier
from sklearn.neighbors import KNeighborsClassifier
classifier=KNeighborsClassifier()
```

```
classifier.fit(x_train,y_train)
'''
#4 logostic regression
from sklearn.linear_model import LogisticRegression
classifier=LogisticRegression()
classifier.fit(x_train,y_train)
'''
# 5 Bernouli naive base
from sklearn.naive_bayes import BernoulliNB
classifier=BernoulliNB()
classifier.fit(x_train,y_train)

#6 gausian naive based
from sklearn.naive_bayes import GaussianNB
classifier=GaussianNB()
classifier.fit(x_train,y_train)

#7 random forest
from sklearn.ensemble import RandomForestClassifier
classifier=RandomForestClassifier()
classifier.fit(x_train,y_train)

#8 xgbost
from xgboost import XGBClassifier
classifier=XGBClassifier()
classifier.fit(x_train,y_train)

#9 lightGBM
from lightgbm import LGBMClassifier
classifier=LGBMClassifier()
classifier.fit(x_train,y_train)
'''
# predict
y_pred=classifier.predict(x_test)

#accuracy score
from sklearn.metrics import accuracy_score
ac=accuracy_score(y_test,y_pred)
print(ac)
'''
# confusion matrix
from sklearn.metrics import confusion_matrix
cm=classifier.confusion_matrix(y_test,y_pred)
print(cm)
'''
# bais variance
bais=classifier.score(x_train,y_train)
var=classifier.score(x_test,y_test)
print("bais:-", bais)
print("variance:-",var)

import pickle
filename="model.pkl"
with open (filename,"wb") as file:
    pickle.dump(classifier,file)

#TfidfVectorizer file
```

```
tfidf_filename = 'tfidf.pkl'
with open(tfidf_filename, 'wb') as scaler_file:
    pickle.dump(cv, scaler_file)

import os
os.getcwd()
```