import pandas as pd

import re

from nltk.stem.porter import PorterStemmer

from nltk.corpus import stopwords

from sklearn.feature\_extraction.text import CountVectorizer

from sklearn.preprocessing import LabelEncoder

from sklearn.model\_selection import train\_test\_split

from sklearn.naive\_bayes import GaussianNB

from sklearn.ensemble import RandomForestClassifier

from sklearn.metrics import accuracy\_score, precision\_score, recall\_score, f1\_score

dataSet = pd.read\_csv('spam.csv', encoding='cp437')

y = dataSet.iloc[:, 0].values

stemmedReviews = []

for i in range(y.size):

review = re.sub('[^a-zA-Z]', ' ', dataSet['v2'][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)

stemmedReviews.append(review)

cv = CountVectorizer(max\_features=3000)

x = cv.fit\_transform(stemmedReviews).toarray()

le = LabelEncoder()

y = le.fit\_transform(y)

X\_train, X\_test, y\_train, y\_test = train\_test\_split(x, y, test\_size=0.20)

classifier = GaussianNB()

classifier.fit(X\_train, y\_train)

pred = classifier.predict(X\_test)

print("\nNAIVE BAYES : \n")

print('Accuracy score: {}'.format(accuracy\_score(y\_test, pred)\*100))

print('Precision score: {}'.format(precision\_score(y\_test, pred)\*100))

print('Recall score: {}'.format(recall\_score(y\_test, pred)\*100))

print('F1 score: {}'.format(f1\_score(y\_test, pred)\*100))

classifier1 = RandomForestClassifier(n\_estimators=15, criterion='entropy')

classifier1.fit(X\_train, y\_train)

predRF = classifier1.predict(X\_test)

print("\nRANDOM FOREST CLASSIFIER : \n")

print('Accuracy score: {}'.format(accuracy\_score(y\_test, predRF)\*100))

print('Precision score: {}'.format(precision\_score(y\_test, predRF)\*100))

print('Recall score: {}'.format(recall\_score(y\_test, predRF)\*100))

print('F1 score: {}'.format(f1\_score(y\_test, predRF)\*100))

stemmedReviews = []

dataset = pd.read\_csv('test1.csv', encoding='cp437')

for i in range(0, 1):

review = re.sub('[^a-zA-Z]', ' ', dataset['v1'][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)

stemmedReviews.append(review)

temp = cv.transform(stemmedReviews)

predNaiveBayes = classifier.predict(temp.toarray())

predRandomForest = classifier1.predict(temp.toarray())

if predNaiveBayes == 1:

OutputNB = "Spam"

else:

print("Not spam")

OutputNB = "Not spam"

if predRandomForest == 1:

print("Spam")

OutputRF = "Spam"

else:

print("Not spam")

OutputRF = "Not spam"

print("\nOutput for Gaussian NB = {0} {1}".format(classifier.predict(temp.toarray()), OutputNB))

print("\nOutput for Random Forest Classifier = {0} {1}".format(classifier1.predict(temp.toarray()), OutputRF))