# -\*- coding: utf-8 -\*-

"""EMAIL\_SPAM\_DETECTION.ipynb

Automatically generated by Colaboratory.

Original file is located at

https://colab.research.google.com/drive/1BPPF1OcNjhM4NuFtMbWrqkW-x5OcUW8w

# \*\*EMAIL SPAM DETECTION WITH MACHINE LEARNING\*\*



#Download \*\*Dataset\*\* here [EMAIL SPAM DETECTION](https://drive.google.com/file/d/19Yz5vBQ3T174xeSOOqVzw3rWWm39stsf/view?usp=share\_link)

### Knowing about the Dataset

\*\*Importing the Required Libraries\*\*

"""

# Numpy Library for Numerical Calculations

import numpy as np

# Pandas Library for Dataframe

import pandas as pd

# Matplotlib and for Plottings

import matplotlib.pyplot as plt

# Pickle Library for Saving the Model

import pickle

# RE Library for Regular Expression

import re

# NLTK Library for Natural Language Processing

import nltk

nltk.download('stopwords') # Downloading the Stopwords

# Stopwords for removing stopwords in the Text

from nltk.corpus import stopwords

# PorterStemmer for Stemming the Words

from nltk.stem.porter import PorterStemmer

# CountVectorizer for Bagging of Words and Vectorizing it

from sklearn.feature\_extraction.text import CountVectorizer

# Train\_Test\_Split for splitting the Dataset

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

# Decision Tree Classifier, Random Forest Classifier and Multinomial Naïve Bayes are Models

from sklearn.tree import DecisionTreeClassifier

from sklearn.ensemble import RandomForestClassifier

from sklearn.naive\_bayes import MultinomialNB

# Accuracy Score and Confusion Matrix is for Analysis of Models

from sklearn.metrics import confusion\_matrix

from sklearn.metrics import accuracy\_score

"""\*\*Reading informations in the Dataset\*\*"""

from google.colab import drive

drive.mount('/content/drive')

spam = pd.read\_csv("/content/drive/My Drive/Oasis Infobyte/Data Science - Internship/Email-Spam-Detection/spam.csv", encoding='ISO-8859-1')

"""\*\*Checking for null values in Data\*\*"""

spam.isnull().sum()

"""\*\*Checking the First Five Values in the Data\*\*"""

spam.head()

"""\*\*Checking the Last Five Values in the Data\*\*"""

spam.tail()

"""\*\*Taking the required Columns in the Dataset\*\*"""

spam = spam[['v1', 'v2']]

spam.columns = ['label', 'message']

spam.head()

"""\*\*Dimensions of the Dataset\*\*"""

spam.shape

"""\*\*Checking for the classes in the Data\*\*"""

spam.groupby('label').size()

"""### Plotting

\*\*Plotting the Label in the Dataset\*\*

"""

spam['label'].value\_counts().plot(kind='bar')

"""### NLP

\*\*Preprocessing the Text in the Dataset\*\*

"""

ps = PorterStemmer()

corpus = []

for i in range(0, len(spam)):

review = re.sub('[^a-zA-Z]', ' ', spam['message'][i])

review = review.lower()

review = review.split()

review = [ps.stem(word) for word in review if not word in stopwords.words('english')]

review = ' '.join(review)

corpus.append(review)

# Printing the first 5 values in the corpus list

corpus[1:6]

"""\*\*Creating Bag of Words Model\*\*"""

cv = CountVectorizer(max\_features = 4000)

X = cv.fit\_transform(corpus).toarray()

Y = pd.get\_dummies(spam['label'])

Y = Y.iloc[:, 1].values

"""### Data Modeling

\*\*Splitting the Dataset into Training and Testing Set\*\*

"""

X\_train, X\_test, Y\_train, Y\_test = train\_test\_split(X, Y, test\_size = 0.20, random\_state=42)

"""### Model Building

\*\*Creating the Models\*\*

"""

# Model 1 - Random Forest Classifier

model1 = RandomForestClassifier()

model1.fit(X\_train, Y\_train)

# Model 2 - Decision Tree Classifier

model2 = DecisionTreeClassifier()

model2.fit(X\_train, Y\_train)

# Model 3 - Multinomial Naïve Bayes

model3 = MultinomialNB()

model3.fit(X\_train, Y\_train)

"""\*\*Prediction\*\*"""

pred1 = model1.predict(X\_test)

pred2 = model2.predict(X\_test)

pred3 = model3.predict(X\_test)

"""### Model Testing

\*\*Testing the Model\*\*

"""

# Model 1 - Random Forest Classifier

print("Random Forest Classifier")

print("Confusion Matrix: ")

print(confusion\_matrix(Y\_test, pred1))

print("Accuracy: ", accuracy\_score(Y\_test, pred1))

print("--------------------------------")

# Model 2 - Decision Tree Classifier

print("Decision Tree Classifier")

print("Confusion Matrix: ")

print(confusion\_matrix(Y\_test, pred2))

print("Accuracy: ", accuracy\_score(Y\_test, pred2))

print("--------------------------------")

# Model 3 - Multinomial Naïve Bayes

print("Multinomial Naïve Bayes")

print("Confusion Matrix: ")

print(confusion\_matrix(Y\_test, pred3))

print("Accuracy: ", accuracy\_score(Y\_test, pred3))

from sklearn.metrics import confusion\_matrix

cm = confusion\_matrix(Y\_test, pred3)

import seaborn as sns

sns.heatmap(cm, annot=True)

report1 = classification\_report(Y\_test, pred1)

print("Classification Report for RFC \n", report1)

report2 = classification\_report(Y\_test, pred2)

print("Classification Report for DTC \n", report2)

report3 = classification\_report(Y\_test, pred3)

print("Classification Report for MNB \n", report3)

"""\*\*Best Model is Multinomial Naïve Bayes\*\*

### Saving Models

Saving all the Models

"""

filename = "RFC.pkl"

pickle.dump(model1, open(filename, 'wb'))

filename = "DTC.pkl"

pickle.dump(model2, open(filename, 'wb'))

filename = "MNB.pkl"

pickle.dump(model3, open(filename, 'wb'))

print("Saved all Models")