AD3391 - AI MINI PROJECT

DETECTION OF SPAM MAIL USING ML ALGORITHM

# DATASET DESCRIPTION:

* The dataset consists of 3 columns i.e. ID, Label and Message. This dataset is my own mail which I converted into csv using setup-mbox-migrator tool. This contains both spam and ham mails.

# **PREPROCESSING:**

import numpy as np

import pandas as pd

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

from sklearn.feature\_extraction.text import TfidfVectorizer

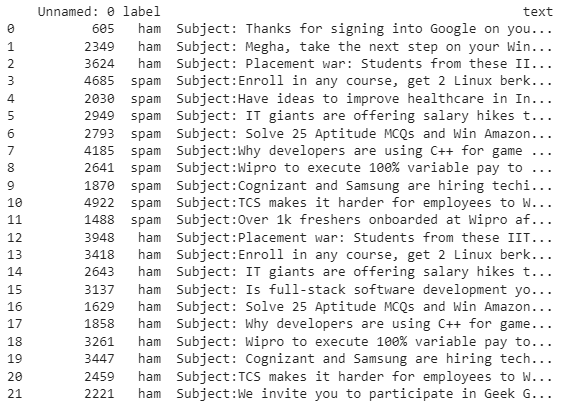
from sklearn.linear\_model import LogisticRegression

from sklearn.metrics import accuracy\_score

# IMPORTING DATASETS:

raw\_mail\_data = pd.read\_csv('/content/spam mail dataset.csv')

print(raw\_mail\_data)

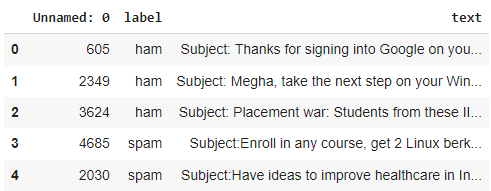


# replace the null values with a null string

mail\_data = raw\_mail\_data.where((pd.notnull(raw\_mail\_data)),'')

# printing the first 5 rows of the dataframe

mail\_data.head()



# checking the number of rows and columns in the dataframe

mail\_data.shape



# label spam mail as 0;  ham mail as 1;

mail\_data.loc[mail\_data['text'] == 'spam mail', 'text',] = 0

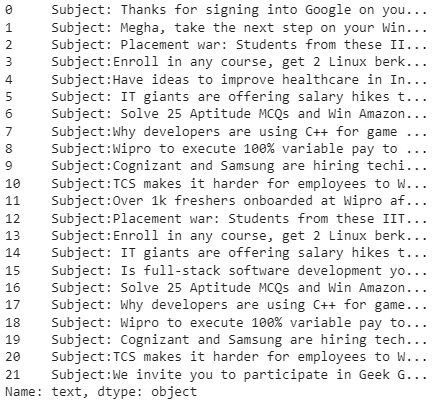
mail\_data.loc[mail\_data['text'] == 'ham mail', 'text',] = 1

# separating the data as texts and label

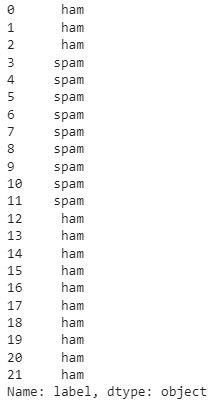
X = mail\_data['text']

Y = mail\_data['label']

print(X)



print(Y)



X\_train, X\_test, Y\_train, Y\_test = train\_test\_split(X, Y, test\_size=0.2, random\_state=3)

print(X.shape)

print(X\_train.shape)

print(X\_test.shape)



# Feature Extraction

# transform the text data to feature vectors that can be used as input to the Logistic regression

feature\_extraction = TfidfVectorizer(min\_df = 1, stop\_words='english',

                                     lowercase='True')

X\_train\_features = feature\_extraction.fit\_transform(X\_train)

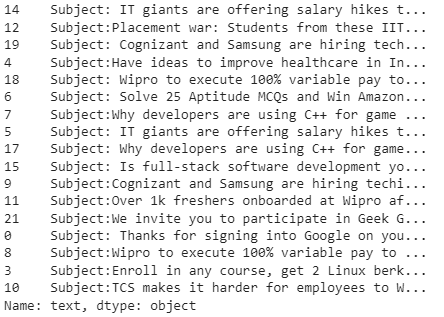
X\_test\_features = feature\_extraction.transform(X\_test)

# convert Y\_train and Y\_test values as integers

Y\_train = Y\_train.astype(str);

Y\_test = Y\_test.astype(str);

print(X\_train)



print(X\_train\_features)

# 

# Training the Model using Logistic Regression

model = LogisticRegression()

# training the Logistic Regression model with the training data

model.fit(X\_train\_features, Y\_train)

# prediction on test data

prediction\_on\_test\_data = model.predict(X\_test\_features)

accuracy\_on\_test\_data = accuracy\_score(Y\_test, prediction\_on\_test\_data)

# **Identification of ham mail**

input\_mail = ["Subject: Megha, take the next step on your Windows device by confirming your Google Account settings"]

input\_data\_features = feature\_extraction.transform(input\_mail)

# making prediction

prediction = model.predict(input\_data\_features)

print(prediction)

## OUTPUT



# **Identification of spam mail**

input\_mail = ["Subject:Enroll in any course, get 2 Linux berks as a bonus."]

input\_data\_features = feature\_extraction.transform(input\_mail)

# making prediction

prediction = model.predict(input\_data\_features)

print(prediction)

## OUTPUT



# **PERFORMACE METRICS:**

prediction\_on\_training\_data = model.predict(X\_train\_features)

accuracy\_on\_training\_data = accuracy\_score(Y\_train, prediction\_on\_training\_data)

print('Accuracy on training data : ', accuracy\_on\_training\_data)

****