**Building a Smarter AI-Powered Spam Classifier – Phase 2**



**Name:** Pavan Kumar V

**Reg No:** 513521104034

**Dept:** CSE

**Year:** III

**NM id:** au513521104034

**E-Mail:**

[pavankumar.v1301@gmail.com](mailto:pavankumar.v1301@gmail.com)

**Introduction:**

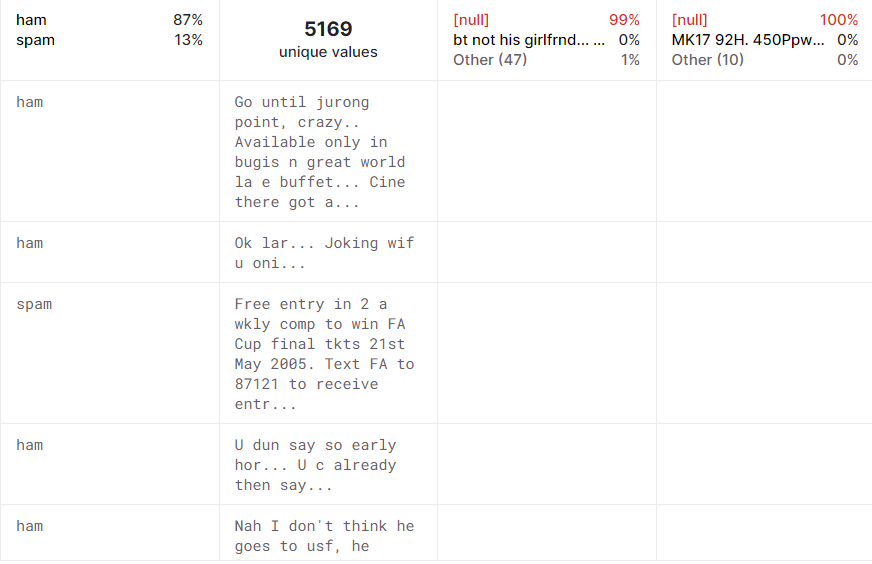
* In today's digital landscape, where spam messages inundate communication channels, this project embarks on the mission to develop a robust spam classifier. Accurate spam classification is indispensable for users seeking to separate legitimate messages from unwanted content, thereby preserving the reliability of their digital communications.
* Recognizing the critical importance of this task, it's essential to acknowledge the formidable challenges it presents. Spam messages continually evolve, employing a diverse array of tactics to evade detection, often leading to suboptimal performance when conventional classification methods are employed.
* The primary objective of this project is to construct a spam classifier that excels in precision and adaptability. By doing so, we aim to empower users to regain control of their digital inboxes, providing them with a more efficient and pleasant communication experience. Through a blend of advanced techniques and innovative approaches, we aspire to overcome the challenges posed by ever-evolving spam and offer users a robust solution to their spam-related concerns.

**Content for Project Phase 2:**

* Consider exploring advanced regression techniques like SVM, Naïve Bayes, Random Forests for improved Prediction accuracy.
* In the event of classifying a dataset which has been defined as a spam base, logistic regression is the most adaptable decision-based technique for detecting spam emails from this type of dataset.

**Data Source:**

* We will need a dataset containing labeled example of spam and non-spam message. We can use Kaggle dataset for this purpose.
* **Dataset Link:**[**https://www.kaggle.com/datasets/uciml/sms-spam-collection-dataset**](https://www.kaggle.com/datasets/uciml/sms-spam-collection-dataset)
* A snapshot of the data is presented in fig.
* The dataset consists of both spam and ham.



**Data Collection and Preprocessing:**

* Importing the dataset: Obtain a comprehensive dataset containing relevant features such as which message should be classified as spam and which should be classified as ham
* Data preprocessing: Clean the data by handling missing values, outliers, and categorical variables. Standardize or normalize numerical features.

**Exploratory Data Analysis (EDA):**

* Visualize and analyze the dataset to gain insights into the relationships between

variables.

* Identify correlations and patterns that can inform feature selection and engineering.
* Present various data visualizations to gain insights into the dataset.
* Explore correlations between features and the target variable.
* Discuss any significant findings from the EDA phase that inform feature selection.

**Pre-Trained Language Model and Transfer Learning:**

* A pre-trained language model is a transformer model, which is trained on large amount of language data for specific tasks.
* The idea behind using pre-trained model is that, model has really good understand of language which we can borrow for our (NLP) task as it is and just focus on training unique part of task in our model. This is called as transfer learning.

**Deployment:**

* Deploy the chosen model for spam classification.

**Program**:

Spam classifier

import numpy as np

import pandas as pd

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

from sklearn.naive\_bayes import MultinomialNB

from sklearn.svm import SVC

from sklearn.ensemble import RandomForestClassifier

from sklearn.metrics import accuracy\_score

import os

for dirname, \_, filenames **in** os.walk('/kaggle/input'):

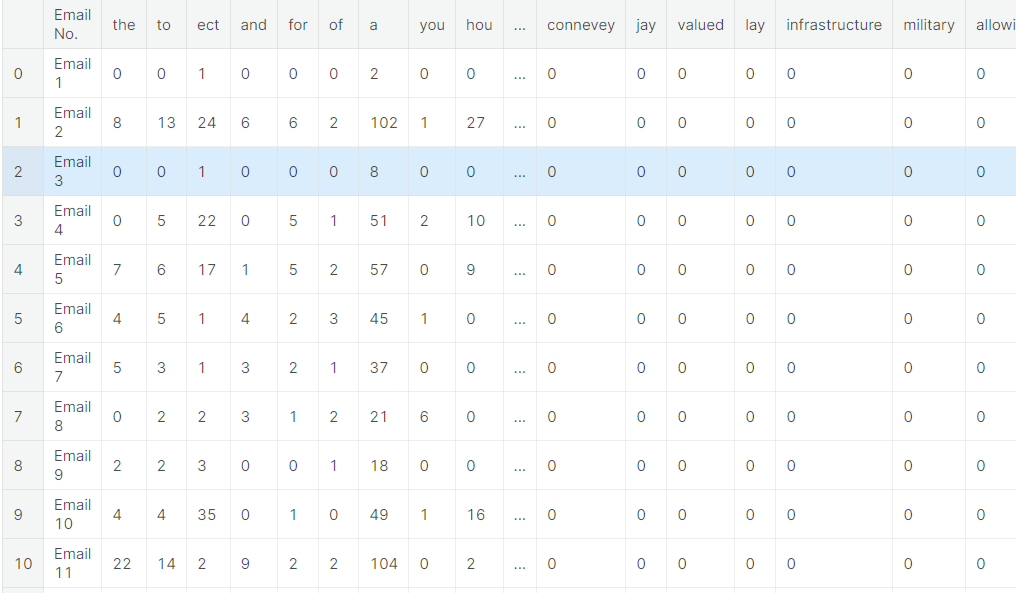
for filename **in** filenames:

print(os.path.join(dirname, filename))

df=pd.read\_csv("../input/email-spam-classification-dataset-csv/emails.csv")

df.head(20)

Output:

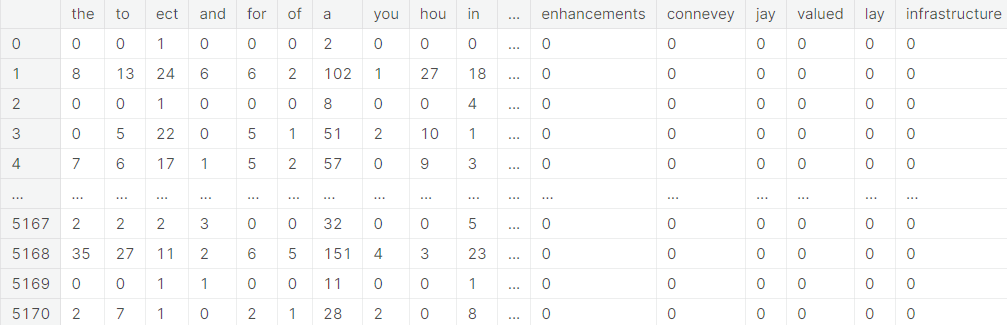


**Creating the NB Model:**

* In this project we are classifying mails typed in by the user as either 'Spam' or 'Not Spam'. Our original dataset was a folder of 5172 text files containing the emails.
* Now let us understand why we have separated the words from the mails. This is because, this is a text-classification problem. When a spam classifier looks at a mail, it searches for potential words that it has seen in the previous spam emails. If it finds a majority of those words, then it labels it as 'Spam'. **Why did I say majority? -->**
* CASE 1: suppose let's take a word 'Greetings'. Say, it is present in both 'Spam' and 'Not Spam' mails.
* CASE 2: Let's consider a word 'lottery'. Say, it is present in only 'Spam' mails.
* CASE 3: Let's consider a word 'cheap'. Say, it is present only in spam.

X = df.iloc[:,1:3001]

X



# Support Vector Machines

Support Vector Machine is the most sought after algorithm for classic classification problems. SVMs work on the algorithm of Maximal Margin, ie, to find the maximum margin or threshold between the support vectors of the two classes (in binary classification). The most effective Support vector machines are the soft maximal margin classifier, that allows one misclassification, ie, the model starts with low bias (slightly poor performance) to ensure low variance later.

svc = SVC(C=1.0,kernel='rbf',gamma='auto')

svc.fit(train\_x,train\_y)

y\_pred2 = svc.predict(test\_x)

print("Accuracy Score for SVC : ", accuracy\_score(y\_pred2,test\_y))

Accuracy Score for SVC: 0.9010054137664346

# Random Forests (Bagging)

* Ensemble methods turn any feeble model into a highly powerful one. Let us see if ensemble model can perform better than Naive Bayes

rfc = RandomForestClassifier(n\_estimators=100,criterion='gini')

rfc.fit(train\_x,train\_y)

y\_pred3 = rfc.predict(test\_x)

print("Accuracy Score of Random Forest Classifier : ", accuracy\_score(y\_pred3,test\_y))

Accuracy Score of Random Forest Classifier: 0.9760247486465584

**CONCLUSION AND FUTURE WORK (PHASE 2)**

**Project Conclusion**: In the Phase 2 conclusion, we will summarize the key findings and insights from the

advanced regression techniques. We will reiterate the impact of these techniques on improving the accuracy and robustness of spam classifier.

**Future Work:** We will discuss potential avenues for future work, such as incorporating

additional data sources (e.g., real-time economic indicators), exploring deep learning models

for prediction, or expanding the project into a web application with more features and

interactivity.