**AIES Mini Project Report**

1. **Project Title**:

**- Movie Recommender System: SMS spam classifier System**

2. **Team Members**:

- Team Member 1: Dittee Salian, roll no.:43

- Team Member 2: Tejas Redkar, roll no.: 44

- Team Member 3: Gayatrini Neogi, roll no.: 49

- Team Member 4: Ishani Patil, roll no.:58

3. **Overview of Various Technologies Used and Datasets**:

* **User Interface (UI)**: Design a user-friendly interface for users to input text (email or SMS).
* **Frontend:** User inputs are sent to the backend for processing.
* **Backend:**

Preprocessing: Tokenization, cleaning, and feature extraction.

Machine Learning Model: Train or load a pre-trained model for spam classification.

Post-processing: Interpret model results and classify emails or SMS.

* **Database**: Store user data for future improvements or analysis.
* **Deployment**: Deploy the application using Docker for scalability.
* **Future Enhancement**: Regular model retraining using new data to improve accuracy. Integration with real-time data sources for continuous learning.

4. **Workflow/Architecture Diagram with Explanation:**

**All the Machine Learning Algorithms works on two stages:-**

1. Training Stage.
2. Testing Stage.

So In the Training Stage Naive Bayes create a Lookup table in which they store all the possibility of probability which we are going to use in the Algorithm for predicting the result.

And In the testing phase let Suppose you have given a test point to the algorithm to predict the result , they fetch the values from the lookup table in which they store all the possibility of probability and use that value to predict the result .

Our first target is to make a list of all the word which are used in that 5172 Email. For this we have some step:

1. Load the “e-mail” folder in Jupiter Notebook With the help of OS in which each file is one Email.
2. Open each file with the help of **f=open(e-mail)  
   In this f=open (e-mail) if you have give one file in f=open() it open that file to read.**
3. Read the File.  
   ***f.read() it read all the content of that email file and store in string format.***
4. Split the file with the spaces (“ “)and append in the list.

In this time we have a list of Words in which we have all the words stored which are used in 5172 Email. But we don’t know which word occur how much time , for finding this we are going to *import counter from collection* ,this counter will give you the result that which word occur how much time  
**from collections import Counter**And pass the word list in counter it form a dictionary which show  
which word occur how much time  
**word\_dict=Counter(words)**

Now we have a Word\_dict in which we have store which word occur how much time but we don’t use all these word because it may reduce the accuracy of our Algorithm So we use the most Common 3000 word , you may take any number like 2500 word or any thing but Here I will take top 3000 word.We have a method to find the most common word from a dictionary  
**word\_dict=word\_dict.most\_common(3000)**

word\_dict is look Something Like this:-  
**Here Key is the word and value is the how much time it occur.**

Now this one is the very important part of this Email-Spam-Classifier:-

As we all know that the for training the data we have to make data in row and column Style , So we are going to make a tabular data in which there are some rows and columns.where each row is one of the email and each column is one the word which are present in that word\_dict and the value is a integer which shows that the number of time that particular word from the word\_dict occur in that particular email. So it form shape like (5172 x 3000) in which 5172 email and the all the 3000 most common word which are stored in word\_dict.

Here you see test\_size =0.2 it means 80% of data we give to the Algorithm to learn the or for training the model and rest 20% we we using for testing.

**This accuracy score help us to predict the accuracy of your Algorithm.**

Here we take a mail in a variable new\_email. After that we split that whole email. We are then going to count that most common word in this input Email which are stored in word\_dict. Next, we convert that list into a numpy array and reshape into (1,3000). Finally, we predict the result with our model which is present in clf object because all the logic of spam or Ham is present in clf object of our model. Here 1 represent Spam and 0 represent Ham.

5. **Future Scope and Conclusion:**

* Future Scope:
* Integration with more communication channels (e.g., social media, chat applications).
* Enhancing model accuracy with more advanced machine learning techniques.
* User feedback system for continuous improvement.

**Conclusion :** Email Spam Classifier is one of the Best Project in the Machine Learning Field. Filtering the Spam e-mail really helps a lot because it become a big trouble over the internet.Machine Learning is the best way to filter the spam out.In this Filtering process we can use multiple of Algorithm but We are Working on Naive Bayes because the performance or Accuracy of this Algorithm is Better than others.