**PHASE 4**

**BUILDING A SMARTER AI-POWERED SPAM CLASSIFIER**

**(development part 2)**

**TEAM MEMBERS:**

1. NALINKUMAR S - (2021504304)
2. DINESHKUMAR S - (2021504710)
3. PRAVEEN G R - (2021504307)
4. VENGADESHWARAN S - (2021504708)

**Abstract:**

Email is a very important way for businesses to talk to each other. Even though there are other ways to communicate, more and more people are using email. But, there's a problem - lots of the emails we get are actually spam, which is like unwanted junk mail. More than half of all emails are spam! This means spammers are wasting our time and resources with messages that don't really matter. These spammers are smart and use tricky methods to send out these annoying emails. So, we need to figure out how to tell the good emails from the bad ones. This paper focuses on using smart computer programs (called machine learning algorithms) to do just that. We look at lots of different ways these programs can be used to figure out if an email is spam or not. We also talk about what future research could be done in this area and what challenges might come up. This information can help other researchers in the future.

**Objective:**

The goal of this research is to use machine learning algorithms, which are like smart programs that learn from data, to sort out spam emails from regular ones. This means training the program to recognize patterns in the words used in emails and decide if they're more like spam or safe emails. It's like teaching the program to be really good at telling the difference between annoying junk mail and important messages.

**Introduction:**

Email is the main way many of us talk officially on the internet. But lately, there's been a big increase in annoying emails called spam. It's like getting a bunch of unwanted messages. The good emails are called 'ham' emails. On average, a regular email user gets about 40-50 emails every day. Surprisingly, spammers make a lot of money, about 3.5 million dollars a year, from sending spam. This causes problems for regular people and businesses.

It also means we spend a lot of time dealing with these annoying emails. Shockingly, spam makes up more than half of all the emails we get, clogging up our inboxes. This not only wastes our time but also makes us less productive. Even worse, spammers use spam for bad things like stealing personal information and causing financial problems. It's a big problem we need to solve!

**Dataset Overview:**

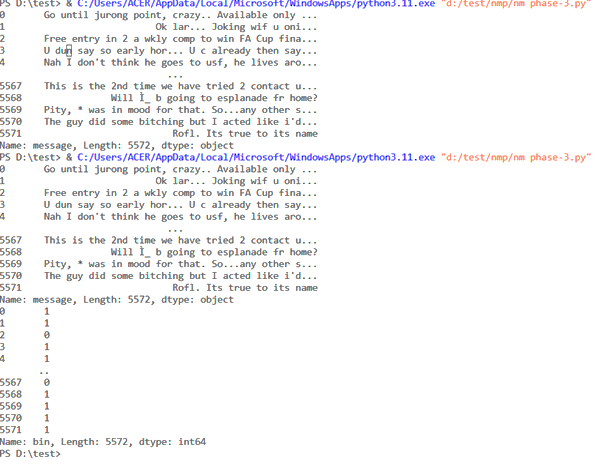
The dataset we used is called the "SMS Spam Collection." It's a bunch of text messages, like the ones you send on your phone. There are 5,574 messages in total, and they're all in English. Some of these messages are normal, like messages from friends or family (we call them "ham"). Others are annoying spam messages. We've carefully labeled each message to tell which is which. This dataset helps us teach computers to tell the difference between regular messages and annoying spam ones.

PROGRAM

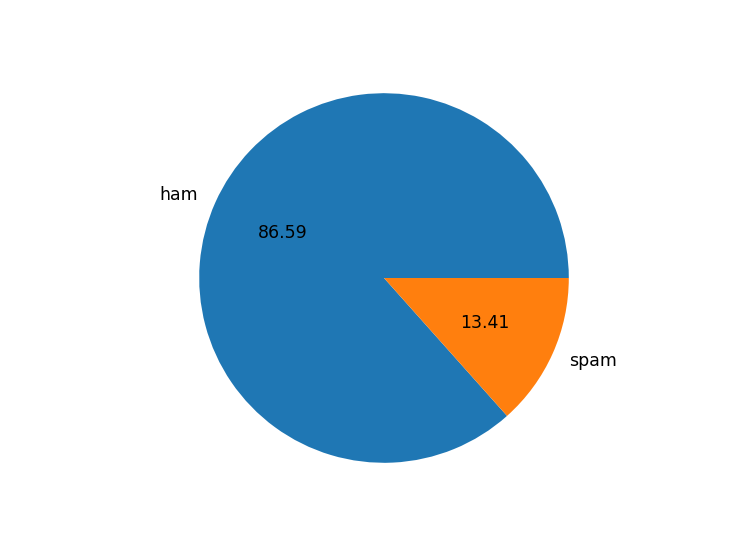
Import necessary libraries



Output after imported spam sample file

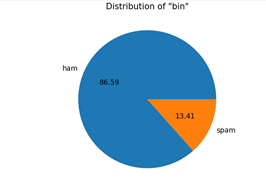


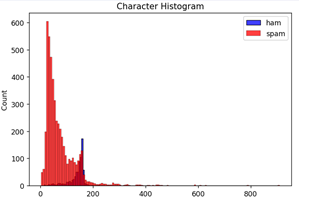
RATIO OF SPAM IN THE SAMPLE





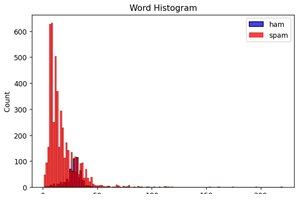
**Spam Representation**

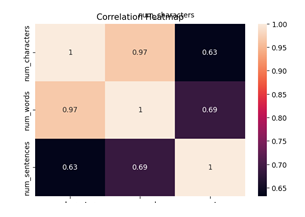
distribution of bin



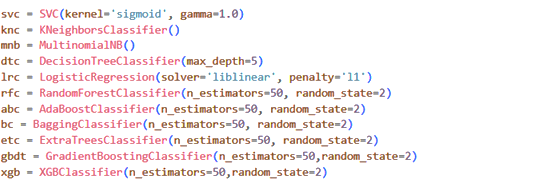
Character Histogram

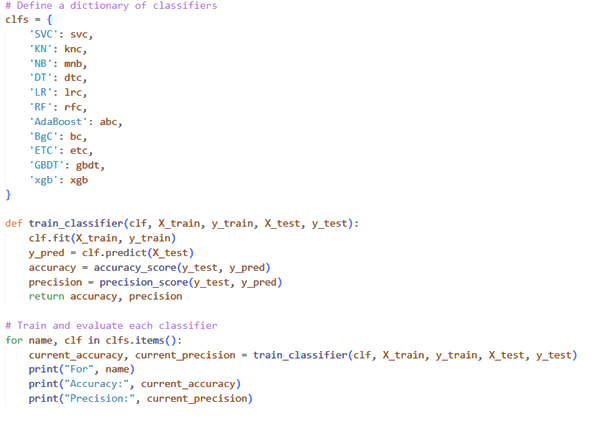
word histogram



Correction heatmap









**Accuracy report:**

An accuracy report is a document or summary that tells us how well a model, system, or process is doing its job correctly. It measures how often it gets things right compared to how often it makes mistakes. In simpler terms, it helps us understand how accurate and reliable the model or system is in performing its tasks.

**Conclusion:**

The creation of an SMS Spam Classifier, achieved through steps like modifying features, training the model, and assessing its performance, plays a vital role in combating the SMS spam problem. Through our evaluation of different classification methods, we gathered the following key insights:

- Support Vector Classifier (SVC) and Random Forest (RF) stood out with the highest accuracy, both at around 97.58%.

- Naive Bayes (NB) achieved a perfect precision score, meaning it had zero false positives.

- Other models like Gradient Boosting, Adaboost, Logistic Regression, and Bagging Classifier performed well, with accuracy scores ranging from 94.68% to 96.03%.

When choosing the best model, it's important to consider factors beyond just accuracy, like how efficiently it runs and the specific needs of the application. It's recommended to fine-tune and validate the model further before making a final decision.