Comment Spam Classifier for Reducing Chat Moderators

Sunrit Pal

IT Department

IEM Kolkata

COMMENT SPAM CLASSIFIER FOR REDUCING CHAT MODERATORS

A REPORT SUBMITTED IN PARTIAL FULFILLMENT FOR

THE DEGREE OF

BACHELOR OF TECHNOLOGY

IN

INFORMATION TECHNOLOGY

FROM

INSTITUTE OF ENGINEERING AND MANAGEMENT KOLKATA

BY

SUNRIT PAL

ENROLLMENT NUMBER: 12019002004041

UNIVERSITY REGISTRATION NUMBER: 027887 of 2019-20

Under the guidance of

PROF. PARTHA SARATHI PAUL

DEPARTMENT OF INFORMATION TECHNOLOGY

INSTITUTE OF ENGINEERING AND MANAGEMENT, KOLKATA

Affiliated to

MAULANA ABUL KALAM AZAD UNIVERSITY OF TECHNOLOGY, KOLKATA, WEST BENGAL

INSTITUTE
OF ENGINEERING & MANAGEMENT
Salt Lake Electronics Complex, Kolkata-700 091, WB, INDIA

Phone : (033) 2357-2969/2059/2995
(033) 2357-8189/8908/5389
Fax : 91-33-2357-8302
E-mail : director@iem.edu.in
Website : www.iem.edu.in

CERTIFICATE

I hereby recommend that the thesis entitled **"COMMENT SPAM CLASSIFIER FOR REDUCING CHAT MODERATORS"** prepared by **Sunrit Pal** (Enrollment No: 12019002004041, University registration No: 027887 of 2019-20) under my supervision be accepted in partial fulfilment of the requirements for the degree of "Bachelor of Technology" in **"Information Technology"** of **Institute of Engineering and Management, Kolkata**.

(Prof. Partha Sarathi Paul)

Project Guide

DEPARTMENT OF INFORMATION TECHNOLOGY,

INSTITUTE OF ENGINEERING AND MANAGEMENT KOLKATA

COUNTERSIGNED

(Dr. Arun Kumar Bar)

Principal

IEM Kolkata

Prof. Dr. Arun Kumar Bar
Principal
Institute of Engineering & Management
Sector-V, Salt Lake Electronics Complex
Kolkata-700091

(Prof. (Dr.) Moutushi Singh)

Head of the Department

Dept. of IT, IEM Kolkata

Declaration

I declare that this project report titled **"COMMENT SPAM CLASSIFIER FOR REDUCING CHAT MODERATORS"** submitted in partial fulfillment of the degree of B. Tech in Information Technology is a record of original work carried out by me under the supervision of Prof. Partha Sarathi Paul, and has not formed the basis for the awardof any other degree or diploma, in this or any other Institution or University. In keeping with the ethical practice in reporting scientific information, due acknowledgements have been made wherever the findings of others have been cited.

Sunrit Pal

Sunrit Pal

Enr. No.- 12019002004041

IEM Kolkata

Place: Kolkata

Date: 10/5/23

Acknowledgement

I am very much indebted to my guide Prof. Partha Sarathi Paul for giving me the opportunity to pursue my B. Tech. thesis work under her guidance. I would like to express my sincere gratitude for her valuable guidance and insightful suggestions throughout my thesis work. It is her exceptional encouragement and precious knowledge in the field of machine learning that has given this project a meaningful shape. She has always encouraged me to get my work published in various journals of repute.

It is beyond my literal and material means to express my heartfelt thanks and deep gratitude to my mentor Prof. Partha Sarathi Paul for her constant encouragement and advice throughout this project.

I was extremely fortune to have Prof. (Dr.) Moutushi Singh as the Head of the department. I thank her for all the support she has provided during the entire duration of the project.

It is my pleasure to acknowledge all my teachers for their extraordinary support and encouragement.

Finally, I thank my parents who have always supported me with deep love and sacrifice throughout my life. I pray for their happiness and good health and I dedicate my work to them in the sincerest way I can think of.

Sunrit Pal

Sunrit Pal

Place: Kolkata

Date: 10/05/23

Abstract

With modern day technology and more and more people looking forward to express themselves on live stream in social media sites like YouTube, Twitch etc., spamming unnecessary and inappropriate sentences and/or links over and over again in the comment section is a major issue. To reduce and/or prevent this from happening, people tend to hire chat moderators so that others don't see unnecessary messages and/or links. Spammed comments are basically unwanted junk that are spammed in bulk in the comment section of indiscriminate streamers' comment section. These comments are basically irrelevant to the stream so it is important to separate these from the real comments that the streamer may read and give adequate attention. Itis for this reason that we are building a comment spam classifier. The aim for this project is the classify comments into spam and ham (not spam) based on huge train data set already fed to the system and various decision algorithms so that real comments are passed through and spams are rejected unless specifically manually allowed by the streamer and/or moderator. So basically, with this project we are achieving the task of binary classification of the comments into two groups. As already mentioned, the reason for doing this is simple: by detecting unsolicited and unwanted comments, we can prevent spam messages from creeping into the comment section, thereby reducing hired chat moderators. We are going to use a spam.csv file, which contains a dataset of both spam and genuine comments, to train our model. We have obtained this dataset from online survey. Our classifier model has been deployed on google colab platform.

Keywords: social media, streamers, binary classification, spammed comments, ham.

Table of Contents

Description	Page Number
Title	i
Certificate	ii
Declaration	iii
Acknowledgement	iv
Abstract	v
List of figures	viii
List of Abbreviations	ix
Chapter 1: Introduction	**10**
1.1 Motivation of the work	10
1.2 The need of Comment Spam Detection	10
1.3 System Design and Work	11
Chapter 2: Literature Review	**12**
2.1 Literature surveys	12
Chapter 3: Spam Messages	**17**
3.1 Standard Filtering	17
3.2 Use of Machine Learning for Spam Detection	17
3.3 Machine Learning Models	18
Chapter 4: Dataset Preparation	**21**
4.1 Implementation of the Problem	21
4.2 Source of the Dataset	21
4.3 How the Dataset is Used	21
Chapter 5: Building the Project	**23**
5.1 Brief Overview	23
5.1.1 Loading the Data	23

5.1.2 Data Cleaning 24
5.1.3 EDA 26
5.1.4 Data Preprocessing 30
5.1.5 Model Building 33
Chapter 6: Results 36
6.1 Results 36
Chapter 7: Limitations 37
7.1 Limitations 37
Chapter 8: Conclusion 38
8.1 Conclusion 38
Chapter 9: Future Prospects 39
9.1 Increasing the Accuracy 39
9.2 Instant Detection 39
Chapter 10: References 40

List of Figures

1. Figure 1: Machine Learning for Spam Detection
2. Figure 2: Decision Tree
3. Figure 3: Naïve Bayes
4. Figure 4: Random Forest
5. Figure 5: Support Vector Machine
6. Figure 6: Basic Goal of our Project
7. Figure 7: Ham VS Spam Pie Chart in our Dataset
8. Figure 8: Histogram of TF-IDF
9. Figure 9: Histogram and Random Forest of Train Data Set
10. Figure 10: Heatmap of IDF
11. Figure 11: WordPress Spam Messages from Train Data Set
12. Figure 12: WordPress Ham Messages from Train Data Set
13. Figure 13: Bar-plot of Spam Tokens based on above WordPress Figures
14. Figure 14: Bar-plot of Ham Tokens based on above WordPress Figures
15. Figure 15: Accuracy VS Precision Relational Database of various ML Algorithms used
16. Figure 16: Accuracy VS Precision based of above table
17. Figure 16: Accuracy and Precision of the Comment Spam Classifier developed

List of Abbreviations

AI	Artificial Intelligence
ML	Machine learning
TF	Term Frequency
IDF	Inverse Document Frequency
Spam	Sales Promotional Advertising Mail
Colab	Colaboratory
SVM	Support Vector Mail
csv	comma-separated values
KNN	K-th Nearest Neighbour
Test data	Data used for testing
Training Data	Data used for training
NN	Neural Network
DT	Decision Tree
EDA	Exploratory Data Analysis
py	Python
Validation data	Data Used for validating algorithm

Chapter 1

Introduction

1.1 Motivation of the work

Comment spam, also referred to as junk comments, spammed comments, or simply spam, is unsolicited messages and/or links sent in bulk by users or machines(spamming). These useless messages are basically occupying unnecessary attention in the streamers comment section and also making the user's experience very poor as streamers' attention is spent on unnecessary spams and unable to sort out the real comments. Not only are spam comments wasting the space of comment section and streamers' attention, there are other dangerous problems posed by these messages. Spam comments can be really dangerous. It can include malicious links that can infect your computer with malware or inappropriate content. This is an even worse problem than space wastage because malware can permanently damage our computer systems and data. Hence, identification of spam comments is a very important need for the users.

1.2 The need of Comment Spam Detection

A comment spam filter is a necessity for every large streamer and organization having a huge fan base as large number of comments are posted at the same time. A fairly popular streamer roughly receives 100-1000 comments a second, out of which an average of 40% of comments are spam and/or inappropriate for the stream. At its very root, keeping your communications flow smooth requires a reliable comment spam filter.

Digging deeper into this topic, let's discuss the reasons that make comment spam filters an absolute-necessity.

Security

Out of all the comments received by an individual throughout the day, the possibility of a phishing attack or cyber threat is never zero. With the benefits of comment spam filters, the security risk can be reduced since the streamers get in hand the comments that have gone through various spam checks. Moreover, these comment spam filters throw out malware, malicious, and virus-infested comment along with links and enhance user experience.

Time-Saving

Let us go back to the comment stats we discussed at the beginning of this section. Having to manually filter out the 60% important messages out of the average 40% clutter by chat moderators (sometimes paid) does seem time-consuming. This can be of greater concern if these stats are put into a fairly large streamers' chat. By filtering out the important chat and blocking the spam messages, a spam filter saves time for the user and keeps the comment section free from malicious links and/or spammed comments.

Increased-Productivity

Along the lines of the time-saving benefit of comment spam filters, these tools facilitate increased productivity of the user by keeping away unwanted comments. As mentioned in the types of comment spam filters, in certain cases the users can set up standards for spam division. By keeping away the unnecessary comments that might distract or waste the time of the user, these spam filters can keep the comment section clean and reduce cost of paid moderators. This in turn increases productivity.

1.3 System Design and Work

Our project will help solve this particular problem faced by users. We are going to train a spam classifier that will do binary classification of the comments into either “spam” or “ham” (ham=not spam) thereby making it easier for the user to identify these junk comments. We will use a dataset obtained from online survey to train our spam classifier. We will use Google Colab for our project as we know that it supports the Python programming language. The model will block spammed comment and/or links and pass other real messages. More details on how we will code and train our spam classifier is explained better in later sections of this report.

Chapter 2

Literature Review

The Literature Survey is used to provide a brief overview and explanation of the reference papers. The literature survey conveys the technical details related to the project in a proper and detailed manner.

2.1 Literature survey

[S.K Tuteja] (2016) [1]

The author has worked with different machine learning algorithms for email classification such as Neural Network (NN), Support Vector Machine (SVM), J48 Decision Tree based classifier, Naïve Bayes. The dataset used by the author was Spam Base dataset. In this paper work, the author didn't mention advantages and disadvantages of any algorithm.

[G. Mujtaba] [L. Shuib] [R. G. Raj] [N. Majeed] [M. A. Al- Garadi] (2017) [2]

Proposed the basic three steps which are common in every classification process. The first step is pre-processing in which the given text is converted into tokens and this step is also used for removal of stop words. The second step is learning process and, in this feature, set is built which is very much necessary for the classification of emails. The last step is classification of email as ham or spam by using efficient algorithm. Algorithms like support vector machine, logistic regression, regression trees and random forest are considered for classification. They used the Phishing Corpus dataset and with the help of Bag of words as feature extraction approach classified the email as ham or spam. In his study, they did not mention the different tools for reduction methods for email classification.

[S. Ajaz] [M. T. Nafis] [V.Sharma] (2017) [3]

They collected email dataset from the online available websites and used Naïve Bayes for filtering of emails. He proposed a hybrid approach using secure hash method and Naive Bayes to filter email data but could not provide information regarding the misuse of storage resources and network bandwidth. By using Secure Hash Algorithm, the email is considered as a message M due to a generated function. The message M is further classified into S and L where L stands for ham email or genuine email and on the other hand S stands for spam email.

[Abdulhamid Muhammad Shafi] [M.S. Osho] [Ismaila] [J.K. Alhassan] (2018) [4]

They did performance analysis for different machine learning classification

techniques such as Radial Basic Function (RBF) Network, Lazy Bayesian Rule, Random Tree, Bayesian Logistic Regression and J48. They did a comparison between all these given algorithms based on Precision, Recall, Root Mean Squared Error, F- Measure and Accuracy. They used the dataset from UCI Machine Learning Repository. For finding the precision and recall value, they applied the F-measure method. The highest F- measure was obtained by using Rotation Forest algorithm and lowest for Naive Bayes algorithm. They used the Kappa Statistics for the statistical result and the best result was obtained for Rotation Forest Algorithm with 87.9. The best accuracy was obtained by using Rotation Forest algorithm with 94% accuracy and lowest accuracy was obtained by REP Tree algorithm with 89%. Other algorithms such as Naive Bayes gave the accuracy of 88.5% and J48 gave 92.3% accuracy.

[N.F. Rusland] [N.Wahid] [S.Kasim] [H.Hafit] (2017) [5]

Performed analysis on email classification on two different datasets by using Naïve Bayes algorithm based on the Accuracy, Precision, F-Measure and Recall. The process was divided into 3 steps. First step is data pre- processing in which all articles, conjunctions and undesired words is removed from the text. Next is the feature extraction followed by training of the Naive Bayes model. Based on the training of the model, it predicts whether the given text is ham or spam. By using Spam data Dataset, the author achieved an accuracy of 91.13% and for the other Spam Base dataset, accuracy achieved was 88%. By his analysis, the author concluded that the performance of Naïve Bayes algorithm is better on Spam data dataset compared to Spam Base.

[A.S Yuksel] [S.F. Cankaya] [I.S. Uncu] (2017) [6]

They compared Support Vector Machine (SVM) and Decision Tree for email filtering. The given dataset was divided into training set and testing set. Each of the model gets trained separately and based on its training, its accuracy is measured. The author made use of supervised learning for both the algorithms and obtained an accuracy of 92% on SVM and an accuracy of 82% on Decision Tree method. Based on his work, the author concluded that SVM performed better than Decision Tree.

[T. Verma] (2017) [7]

Proposed a method using SVM algorithm and feature extraction for filtering emails. This method consists of several steps such as Email Collection in which data is obtained from the dataset. After that it sent for pre- processing where unnecessary contents are removed and only desired content is sent for further process. Then the process of feature extraction followed by training of SVM model. The author used the dataset from Apache Public corpus. In the proposed solution, special symbols, HTML tags, URL and unnecessary alphabets were removed. The author used the Vocab file to map all the words from the dictionary. By using the SVM algorithm on pre- processed dataset, an accuracy of 98% was obtained.

[V.K Singh] [S. Bhardwaj] (2018) [8]

They worked on the solution for combining classification technique to get better result for spam filtering. The author took help of data mining and by using that collected all the information regarding success, current problems and previous failures of spam filtering. The method was based on binary classification where 1 was used for Spam email and 0 was used for Ham emails. They combined the 2 method that is Machine Learning and Knowledge Engineering for the filtering of emails. The performance of the proposed method was very poor on the combined KNN and SVM algorithm.

[Priti Sharma] [Uma Bhardwaj] (2017) [9]

Performed comparison between the Naive Bayes and J48 decision tree algorithm for classification of emails. They used the dataset of size 1000.They performed three experiments and based on the results, the algorithms are compared by evaluating the different performance parameters like accuracy, recall, precision, true negative rate, F-measure. First experiment performed by using Naive Bayes classifier and accuracy achieved was 83.5% along with precision value of 85.26% and recall value of 85.26%. Second experiment performed by using J48 decision tree classifier and accuracy achieved was 91.5% with precision value of 93.68% and recall value of 89%. The third experiment was done by using hybrid bagged approach. In last experiment accuracy achieved was 87.5% with precision value of 89.47% and recall value of 85%. For future enhancement, the concept of boosting approach can be used as it might replace the weak classifier's learning features with the strong classifier's features.

[Manmohan Singh] [Rajendra Pamula] [Shudhanshu Kumar Shekhar] (2018) [10]

The author has compared the Gaussian Kernel and the Linear Kernel by using the Support Vector Machine algorithm for classification of emails. Linear separable problem is the one in which Linear Decision Boundary can be used to separate a class. For a particular problem, there can be several decision boundaries but a good and an efficient decision boundary is the one that perfectly fits the data given and also able to classify any new data. Gaussian boundary can be used where linear decision boundary is not effectively fitting to the given dataset. They used the spamTrain.mat dataset of size 4k for training purpose which contain both ham and spam emails. SpamTest.mat file containing 1k entry was used for testing purpose. Both the training and testing files are a subset of Spam Assassin Public Corpus. Along with the testing accuracy, they also focused on the training time and testing time in both approaches. Accuracy obtained by using Linear kernel was 98.5% with training time of 134 (in sec) and by using Gaussian Kernel, the accuracy was 97.1% with training time of 190(in sec).So based on the result, the author concluded that training time for linear kernel is far less than Gaussian kernel and also Linear Kernel has higher accuracy than that of Gaussian Kernel. Though the Gaussian kernel is more advanced and better fitting kernel than linear kernel but the dataset used has large number of features and thus a dataset having large number of features fit more better using linear kernel than Gaussian Kernel.

[Linda Huang] [Julia Jia] [Emma Ingram] [Wuxu Peng] (2018) [11]

Proposed a solution to increase the accuracy of Naive Bayes and reduce the false positive rate. Naive Bayes is a supervised machine learning algorithm which is based on Bayes theorem and can be used as a probabilistic model for classification of emails. Although Naive Bayes classifier provides higher accuracy but still spammers are able to bypass the filter by using leetspeak and diacritics. Leetspeak is a coded spelling system and language used in very informal communication on the internet, featuring letters combined with numbers or special characters in place of letters that they may resemble, and including inventive misspellings, jargon, and slang. Diacritic is a sign, such as an accent or cedilla, which when written above or below a letter indicates a difference in pronunciation from the same letter when unmarked or differently marked. They have done some modification in Naive Bayes to convert the symbols present in the text into possible letters and used a spell check to make sure that the corrected symbol is a word and then it is passed through the algorithm for classification. By doing this, they improved the accuracy from 23.9% to 62%.

[Prachi Gupta] [Ratnesh Kumar Dubey] [Dr. Sadhna Mishra] (2019) [12]

In this, they have compared the performance of Naive Bayes and Support Vector Machine algorithm for classification of emails. The dataset they have used consists of 5574 rows and 2 columns. One column is used for storing emails and other is used as label (Ham or Spam). Totally, they used 4 steps as Data Collection, Data Preprocessing, Data Transformation and Classification System for classification of emails. Data Pre-processing was used to clean the data and make it free from any kind of ambiguities, errors, redundancy. In Data Transformation, pre-processed data is converted into lowercase and converted into format as desired by the algorithm for classification. And at last desired attributes are identified and by using feature extraction, algorithm classifies the content into Ham or Spam. Accuracy obtained by using Naive Bayes was 99.49% and it was 86.35% by using Support Vector Machine. So, the author concluded that Naive Bayes algorithm performed exceptionally well as compared to SVM for classification of emails.

[U.K Sah] [N.Parmar] (2017) [13]

Proposed a method for classifying an email as Ham or Spam by using feature selection and also worked to improve the training time as well as the accuracy of the spam filtering model. They also performed a comparison between the Naive Bayes algorithm and Support Vector Machine. Based on the accuracy as well as the computation time of the algorithm for the given dataset. The whole process was divided into four steps. First was preparation of data in which the given dataset was divided into training set consisting of 702 mails and testing set with 260 mails. The second step was creation of word dictionary followed by the third step which was feature selection process by generating the feature vector matrix. The last step was to train the model and based on its training the model predicts the email as ham or spam. Based on the results obtained, the author concluded that Naive Bayes

gives better accuracy in comparison with Support Vector Machine.

Reference Number	Evaluation Metrics	Dataset	Future Work
[1]	Neural Network, Support Vector Machine, J48 Decision Tree	Spam Base Phishing Corpus	Algorithm can be used with the dataset having larger size
[2]	Support Vector Machine, Logistic Regression, Regression Tree, Random Forest	UCI Machine Leaning Repository	
[4]	Radial Basic function, Lazy Bayesian Rule, Random Tree, J48		Efficient Algorithm is required to achieve more accuracy
[7]	Support Vector Machine	Apache Public Corpus	
[8]	k-nearest neighbors, Support Vector Machine Naive Bayes, J48	Online Available Websites	Efficient method to achieve high accuracy, Acceptable Recall and Precision Value
[9]		Ling Spam Dataset	Concept of boosting approach can be used as it might replace the weak classifiers leaving features
[10]	Gaussian Kernel, Linear Kernel using Support Vector Machine	Spam Assassin Public Corpus	
[11]	Naive Bayes	Ling Spam Corpus	To Increase Speed and Efficiency and Also Detect other Forms of Email Messages Larger Size Dataset is Required

Chapter 3

Spam Messages

3.1 Standard Filtering

The comment spam definition is ambiguous since it is streamers' platform dependent. At present, comment spam is getting the attention of everyone. Comment spam ordinarily includes particular spontaneous messages sent in mass by individuals and/or machines. In the era of technology, the dodger/spammer shows a story where the unfortunate casualty needs forthright financial help so that the fraudster can gain a lot bigger total of cash, which they would then share.

The fraudster will either earn a profit or avoid communication when the unfortunate victim completes the instalments. Currently, various companies develop different techniques and algorithms for efficient spam detection and filtering. But none have paid much attention to the comment section with the rise of streamers in recent days. We address some filtering strategies in this section to understand the filtering process. Standard spam filtering is a filtering system that implements a set of rules and works with that set of protocols as a classifier. In the first step, content filters are implemented and using machine learning techniques to figure out the spam. Backlist filters are applied to the comment to clinch the data coming from the backlist file to avoid spam messages. After this stage, rule-based filters are implemented, recognizing the sender using the subject line and user-defined parameters. Eventually, allowance and task filters are used by implementing a method that allows the account holder to send the message.

3.2 Use of Machine Learning for Spam Detection

Machine learning is one of the most important and valuable applications of artificial intelligence (AI), which gives computer systems the ability of automatically learning and enhancing their functionality without explicit programming. The primary purpose of machine learning algorithms is to build automated tools to access and use the data for training. Machine learning consists of three major kinds, used for numerous tasks. For the last decade, researchers have been trying to make comment section communication better than today. Spam filtering of comments is one of the most critical ways of protecting message section networks. Many research articles have been published using various machine learning approaches to identify and process spam emails, but there are still some research gaps in its application on comment sections. Junk comment is one of the central, attractive research fields for filling the gaps. That is why, this paper is presented to make a summarized version of different existing machine learning models and approaches that are being used for comment spam detection and removal. This paper also evaluates the most common machine learning approaches like KNN, SVM, random forest, and Naive Bayes.

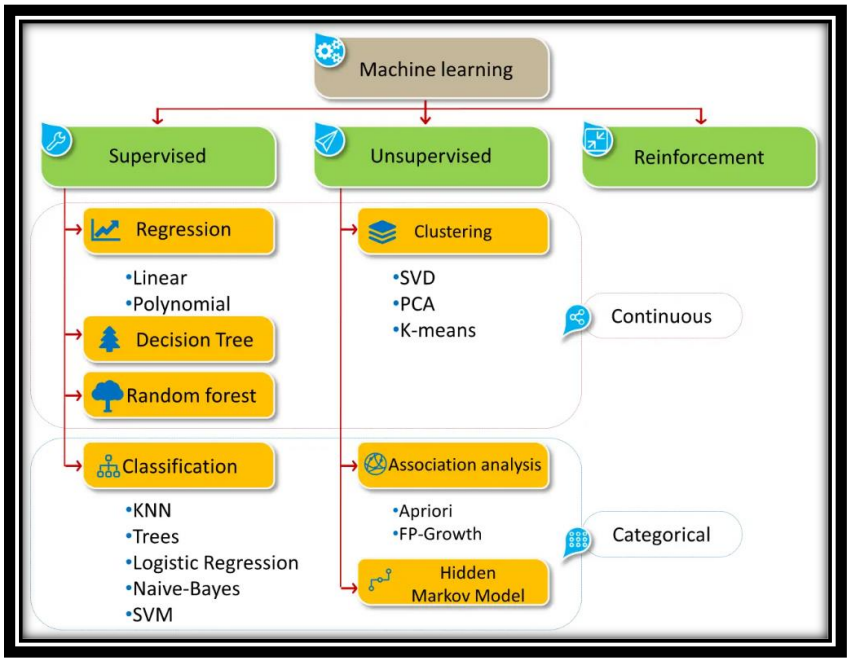

Fig 1: Machine Learning for Spam Detection

3.3 Machine Learning Models

Decision Tree:
Decision Tree is a Supervised learning technique that can be used for both classification and Regression problems but mostly it is preferred for solving Classification problems. It is a true-structured classifier, where internal nodes represent the features of the dataset, branches represent the decision rules and each leaf node represents the outcome.

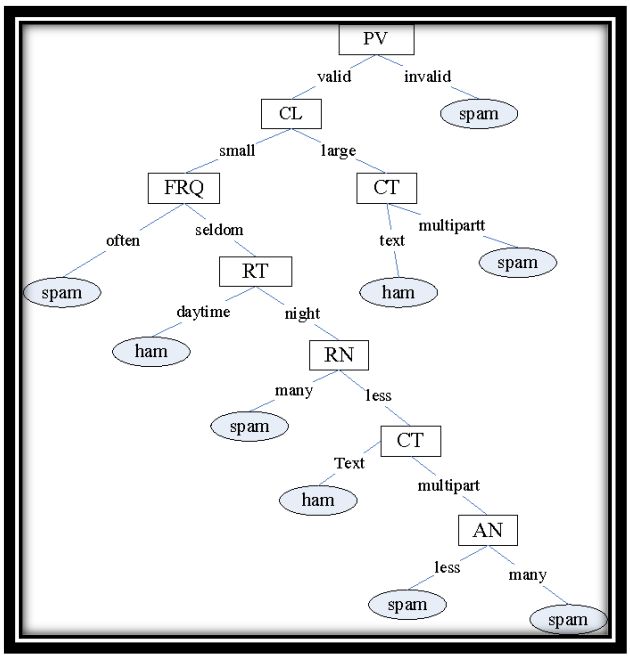

Fig 2: Decision Tree

Naive Bayes:

Naive Bayes Classifier is one of the simple and most effective Classification algorithms which helps in building the fast machine learning models that can make quick predictions. It is a probabilistic classifier, which means it predicts on the basis of the probability of an object.

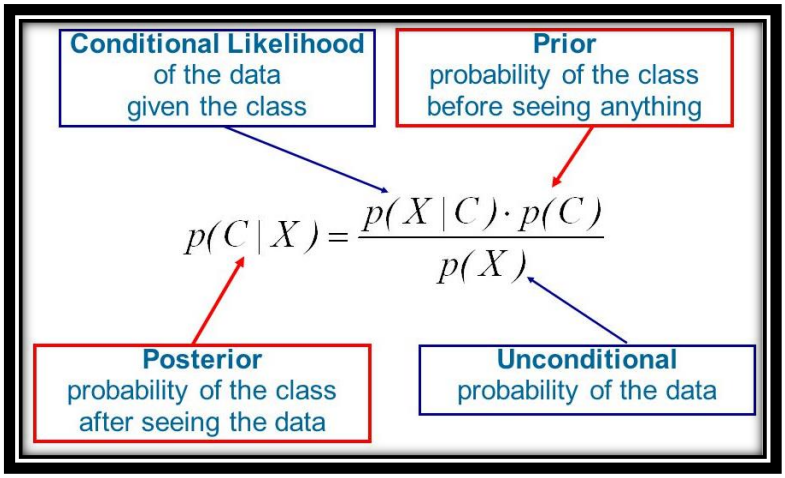

Fig 3: Naïve Bayes

Random Forest:

As the name suggests, "Random Forest is a classifier that contains a number of decision trees on various subsets of the given dataset and takes the average to improve the predictive accuracy of that dataset." Instead of relying on one decision tree, the random forest takes the prediction from each tree and based on the majority votes of predictions, and it predicts the final output.

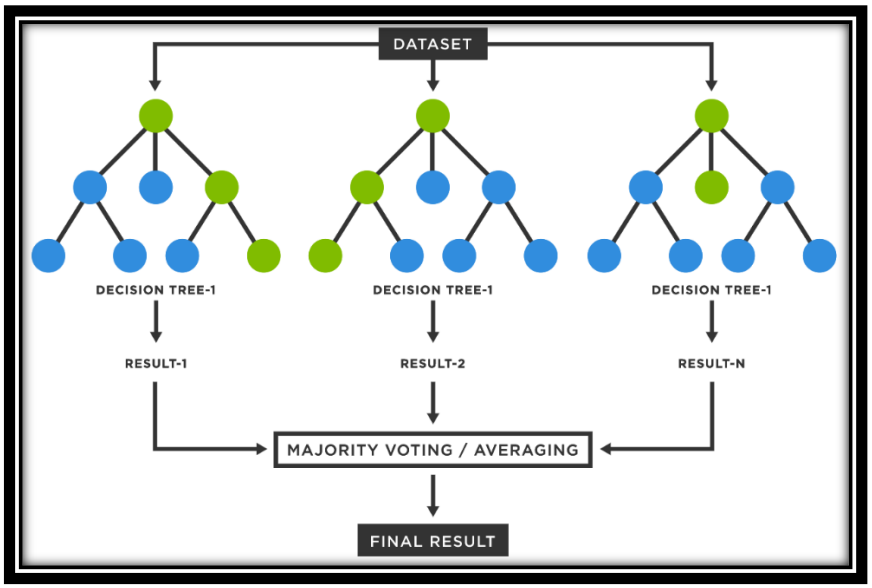

Fig 4: Random Forest

Support Vector Machine (SVM):

The goal of the SVM algorithm is to create the best line or decision boundary that can segregate n-dimensional space into classes so that we can easily put the new data point in the correct category in the future. This best decision boundary is called a hyperplane.

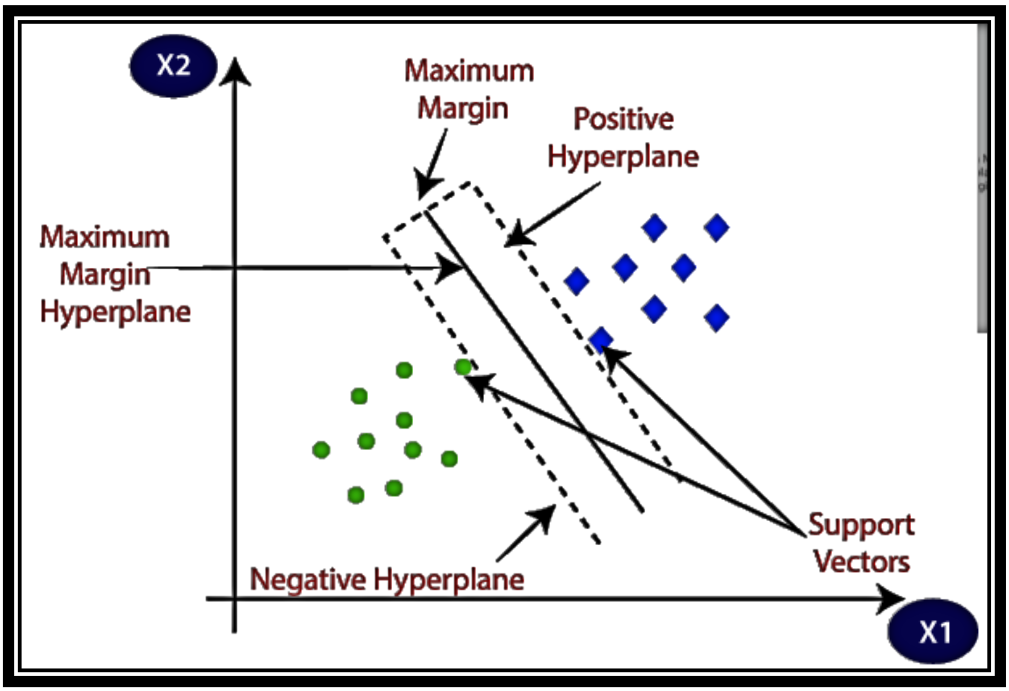

Fig 5: Support Vector Machine

Chapter 4

Dataset Preparation

4.1 Implementation of the problem

Identification of spam messages is a very important need for the users. Our project will help solve this particular problem faced by users by reducing the demand of paid chat moderators. We are going to train a spam classifier that will do binary classification of the comments into either “spam” or “ham”(ham=not spam) which in turn will block the spam comments and/or links will allowing the hams to pass through. We will use a dataset obtained from online survey to train our spam classifier.

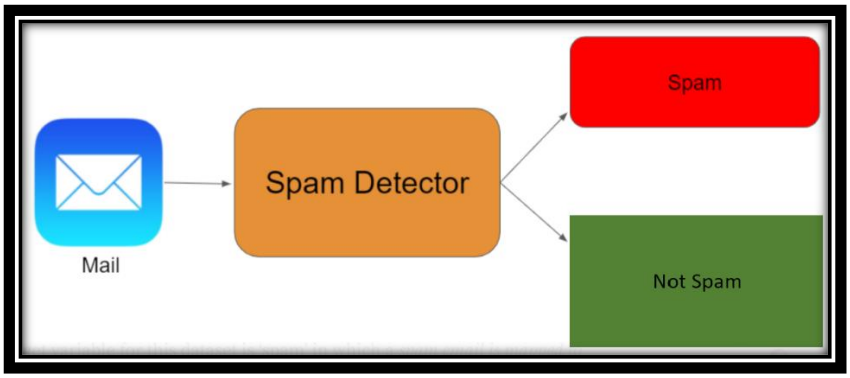

Fig 6: Basic Goal of our Project

4.2 Source of the Dataset

The Data Set that we have used to train our spam classifier has been obtained from online survey. It contains one set of comment messages in English of 5,574 messages, tagged according being ham (legitimate) or spam.

4.3 How the Dataset is used

The files contain one message per line. Each line is composed by two columns: v1 contains the label (ham or spam) and v2 contains the raw text.

This corpus has been collected from online survey:

- A collection of 425 spam messages was manually extracted from the survey. The identification of the text of spam messages in the claims is a very hard and time-consuming task, and it involved carefully scanning hundreds of web pages.

- A subset of 3,375 randomly chosen ham messages of the Corpus, which is a dataset of about 10,000 legitimate messages collected for research at the online survey.

- A list of 450 Email ham messages collected from Caroline Tag's PhD Thesis.

- Finally, we have incorporated the Spam Corpus v.0.1 Big. It has 1,002 ham messages and 322 spam messages.

Please note that the following is an incomplete list of the data set that we are using just to give a rough idea of the nature of the data that we are using. This data has been represented in a tabular format with two columns. The first column v1 tells whether the email is spam or ham. The column v2 displays the actual content of the email.

V1	V2
ham	I've been searching for the right words to thank you for this breather. I promise i wont take your h...
spam	URGENT! You have won a 1 week FREE membership in our $100,000 Prize Jackpot! Txt the word: CLAIM to ...
spam	WINNER!! As a valued network customer you have been selected to receivea $900 prize reward! To claim...
ham	I'm gonna be home soon and i don't want to talk about this stuff anymore tonight, k? I've cried enou...

Chapter 5

Building the Project

5.1 Brief Overview

- First of all, we will download the data set from online survey and convert them into .csv format which we will use to train our Spam Classifier. More information related to this data set including how large this data set is, the various sources from where these data have been collected etc. will be explained in detail in a subsequent section of the project report.
- Open Google Colab and upload the spam.csv file onto it. So now our dataset has been uploaded.
- Create a Python Notebook named "Comment Spam Classifier" and read the entire dataset onto the python notebook.
- Now we will do Data Cleaning on this dataset which is basically identifying incomplete, incorrect, inaccurate, or irrelevant parts of the data and then replacing, modifying, or deleting the dirty or coarse data. The code required to do this data cleaning is provided in detail in later sections.
- Next is the process of EDA (Exploratory Data Analysis) which is an approach to analyze the data using visual techniques. It is used to discover trends, patterns, or to check assumptions with the help of statistical summary and graphical representations. The detailed coding approach is given in subsequent sections.
- Next is Text Preprocessing where we will transform the text into a clean and consistent format that can then be fed into a model for further analysis and learning. This includes vectorization of the data, stemming and removal of stop-words. Detailed code of this is given later.
- Next is Model Building.
- Next is Evaluation of this model that we build.
- Next is Improvement, which basically means that based on the evaluation of our model if any improvements must be made then we will make those.

5.1.1 Loading the data

Firstly, we are going to load the data of spam.csv into our Python Notebook. Here is the following code for that:

```python
import numpy as np
import pandas as pd
```

```python
df = pd.read_csv('spam.csv')
```

5.1.2 Data Cleaning

Data cleaning means fixing bad data in your data set. Bad data could be: Empty cells. Data in wrong format. Wrong data. Here is the following code for data cleaning with output:

```python
df.info()
```

Output:

```
RangeIndex: 5572 entries, 0 to 5571
Data columns (total 5 columns):
 #   Column      Non-Null Count  Dtype
---  ------      --------------  -----
 0   v1          5572 non-null   object
 1   v2          5572 non-null   object
 2   Unnamed: 2  50 non-null     object
 3   Unnamed: 3  12 non-null     object
 4   Unnamed: 4  6 non-null      object
dtypes: object(5)
memory usage: 217.8+ KB
```

```python
# drop last 3 cols
df.drop(columns=['Unnamed: 2','Unnamed: 3','Unnamed: 4'],inplace=True)
```

```python
df.sample(5)
```

Output:

	v1	v2
1947	ham	The battery is for mr adewale my uncle. Aka Egbon
2712	ham	Hey you still want to go for yogasana? Coz if ...
4428	ham	Hey they r not watching movie tonight so i'll ...
3944	ham	I will be gentle princess! We will make sweet ...
49	ham	U don't know how stubborn I am. I didn't even ...

```python
# renaming the cols
df.rename(columns={'v1':'target','v2':'text'},inplace=True)
df.sample(5)
```

Output:

	target	text
1418	ham	Lmao. Take a pic and send it to me.
2338	ham	Alright, see you in a bit
88	ham	I'm really not up to it still tonight babe
3735	ham	Hows the street where the end of library walk is?
3859	ham	Yep. I do like the pink furniture tho.

```
from sklearn.preprocessing import LabelEncoder
encoder = LabelEncoder()
```

```
df['target'] = encoder.fit_transform(df['target'])
```

```
df.head()
```

Output:

	target	text
0	0	Go until jurong point, crazy.. Available only ...
1	0	Ok lar... Joking wif u oni...
2	1	Free entry in 2 a wkly comp to win FA Cup fina...
3	0	U dun say so early hor... U c already then say...
4	0	Nah I don't think he goes to usf, he lives aro...

```
Out[14]: target    0
         text      0
         dtype: int64
```

```
In [15]: # check for duplicate values
         df.duplicated().sum()
```

```
Out[15]: 403
```

```
In [17]: # remove duplicates
         df = df.drop_duplicates(keep='first')
```

```
In [18]: df.duplicated().sum()
```

```
Out[18]: 0
```

```
In [19]: df.shape
```

```
Out[19]: (5169, 2)
```

5.1.3 EDA (Exploratory Data Analysis):

EDA is applied to investigate the data and summarize the key insights. It will give you the basic understanding of your data, it's distribution, null values and much more. You can either explore data using graphs or through some python functions. Below is some of the EDA code with given output:

In [29]:
```
df.head()
```

Out[29]:

	target	text
0	0	Go until jurong point, crazy.. Available only ...
1	0	Ok lar... Joking wif u oni...
2	1	Free entry in 2 a wkly comp to win FA Cup fina...
3	0	U dun say so early hor... U c already then say...
4	0	Nah I don't think he goes to usf, he lives aro...

```
df['target'].value_counts()
```

```
0    4516
1     653
Name: target, dtype: int64
```

```
import matplotlib.pyplot as plt
plt.pie(df['target'].value_counts(), labels=['ham','spam'],autopct="%0.2f")
plt.show()
```

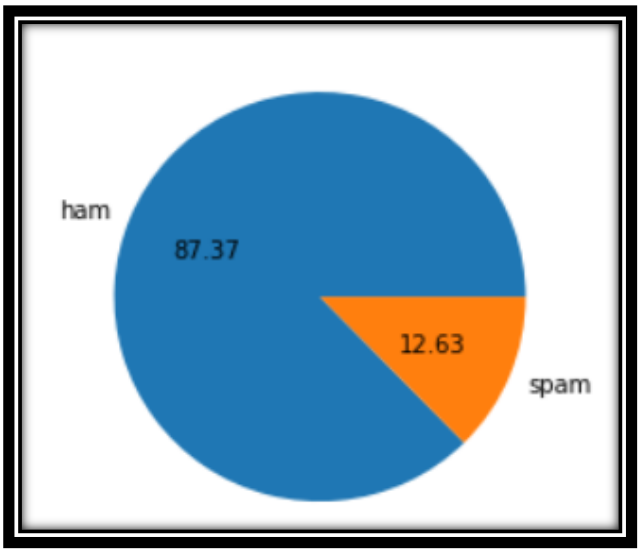

Fig 7: Ham VS Spam Pie Chart in our Dataset

In [53]:
```
df['num_sentences'] = df['text'].apply(lambda x:len(nltk.sent_tokenize(x)))
```

In [54]:
```
df.head()
```

Out[54]:

	target	text	num_characters	num_words	num_sentences
0	0	Go until jurong point, crazy.. Available only ...	111	24	2
1	0	Ok lar... Joking wif u oni...	29	8	2
2	1	Free entry in 2 a wkly comp to win FA Cup fina...	155	37	2
3	0	U dun say so early hor... U c already then say...	49	13	1
4	0	Nah I don't think he goes to usf, he lives aro...	61	15	1

In [55]:
```
df[['num_characters','num_words','num_sentences']].describe()
```

Out[55]:

	num_characters	num_words	num_sentences
count	5169.000000	5169.000000	5169.000000
mean	78.923776	18.456375	1.962275
std	58.174846	13.323322	1.433892
min	2.000000	1.000000	1.000000
25%	36.000000	9.000000	1.000000
50%	60.000000	15.000000	1.000000
75%	117.000000	26.000000	2.000000
max	910.000000	220.000000	38.000000

In [58]:
```
# ham
df[df['target'] == 0][['num_characters','num_words','num_sentences']].describe()
```

Out[58]:

	num_characters	num_words	num_sentences
count	4516.000000	4516.000000	4516.000000
mean	70.456820	17.123339	1.815545
std	56.356802	13.491315	1.364098
min	2.000000	1.000000	1.000000
25%	34.000000	8.000000	1.000000
50%	52.000000	13.000000	1.000000
75%	90.000000	22.000000	2.000000
max	910.000000	220.000000	38.000000

In [59]:

```
#spam
df[df['target'] == 1][['num_characters','num_words','num_sentences']].describe()
```

Out[59]:

	num_characters	num_words	num_sentences
count	653.000000	653.000000	653.000000
mean	137.479326	27.675345	2.977029
std	30.014336	7.011513	1.493676
min	13.000000	2.000000	1.000000
25%	131.000000	25.000000	2.000000
50%	148.000000	29.000000	3.000000
75%	157.000000	32.000000	4.000000
max	223.000000	46.000000	9.000000

In [78]:

```
import seaborn as sns
```

In [84]:

```
plt.figure(figsize=(12,6))
sns.histplot(df[df['target'] == 0]['num_characters'])
sns.histplot(df[df['target'] == 1]['num_characters'],color='red')
```

Out[84]:

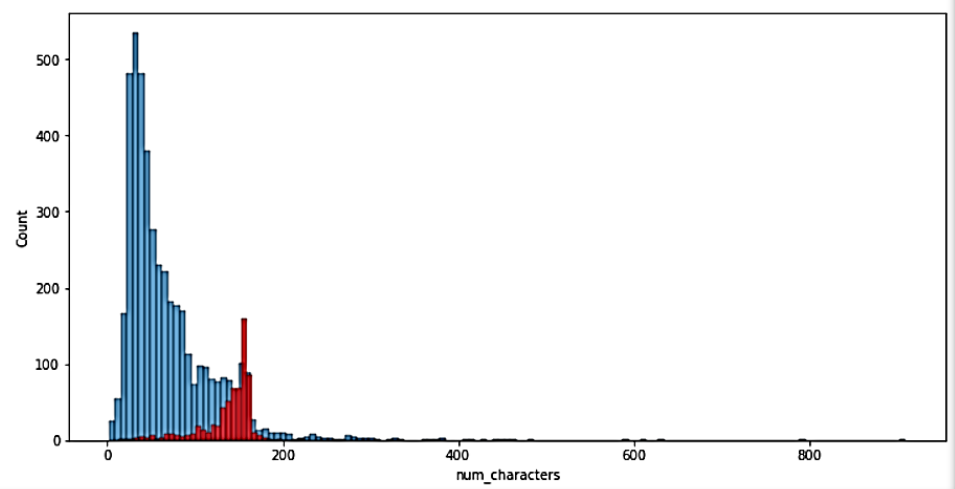

Fig 8: Histogram of TF-IDF

In [86]:

```
sns.pairplot(df,hue='target')
```

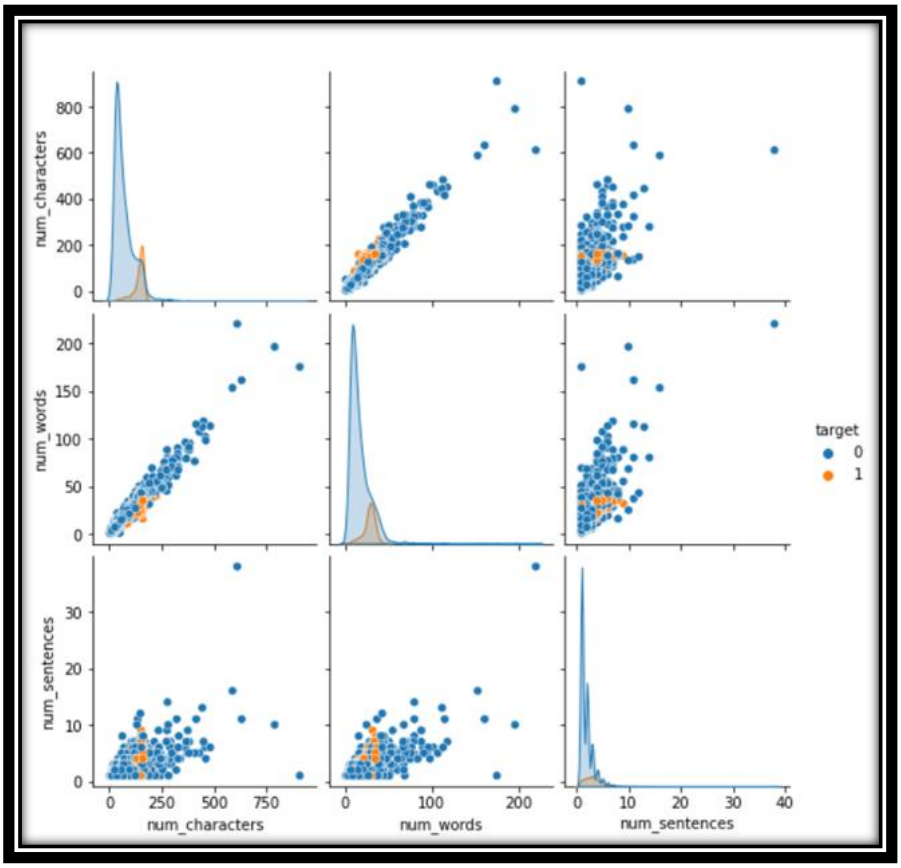

Fig 9: Histogram and Random Forest of Train Data Set

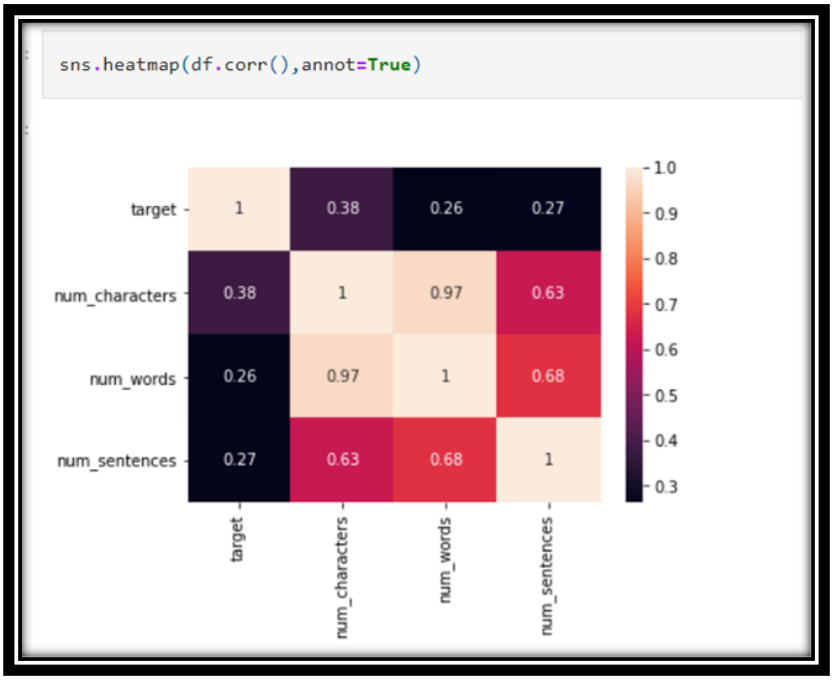

Fig 10: Heatmap of IDF

5.1.4 Data Preprocessing:

For machine learning algorithms to work, it's necessary to convert raw data into a clean data set, which means we must convert the data set to numeric data. We do this by encoding all the categorical labels to column vectors with binary values.

In data preprocessing, we do the following-

5.1.4.1 Lower case:

Converting a word to lower case (NLP->nlp). Words like *Book* and *book* mean the same but when not converted to the lower case those two are represented as two different words in the vector space model (resulting in more dimensions).

5.1.4.2 Tokenization:

Splitting the sentence into words and classifying them into keywords, identifiers, string constant/literals, operator, separator, special characters etc.

5.1.4.3 Stop Words Removal:

Stop words are very commonly used words (a, an, the, etc.) in the documents. These words do not really signify any importance as they do not help in distinguishing two documents.

5.1.4.4 Stemming:

It is a process of transforming a word to its root form. Eg, preprocessing to process.

5.1.4.5 Lemmatization:

Lemmatization reduces the words to a word existing in the language. Libraries such as **nltk**, and **spaCy** have stemmers and lemmatizers implemented. These are built based on a rule-based approach.

Example of messages before and after transformation are in the below chart:

Out[267…

	target	text	num_characters	num_words	num_sentences	transformed_text
0	0	Go until jurong point, crazy.. Available only ...	111	24	2	go jurong point crazi avail bugi n great world...
1	0	Ok lar... Joking wif u oni...	29	8	2	ok lar joke wif u oni
2	1	Free entry in 2 a wkly comp to win FA Cup fina...	155	37	2	free entri 2 wkli comp win fa cup final tkt 21...
3	0	U dun say so early hor... U c already then say...	49	13	1	u dun say earli hor u c alreadi say
4	0	Nah I don't think he goes to usf, he lives aro...	61	15	1	nah think goe usf live around though

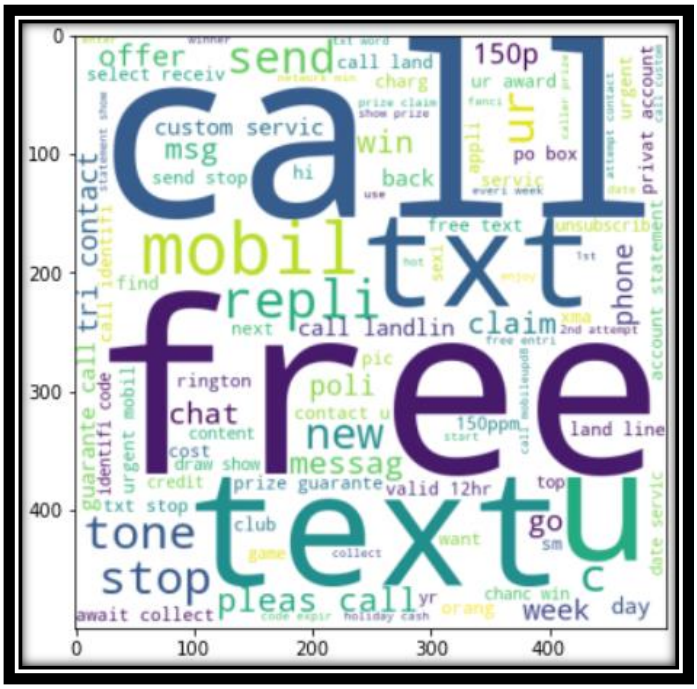

Fig 11: WordPress Spam Messages from Train Data Set

Fig 12: WordPress Ham Messages from Train Data Set

Note: The above figure of Ham and Spam messages are based on the DF technique. The higher the occurrence of a word, the larger the size of the word in the wordpress figures.

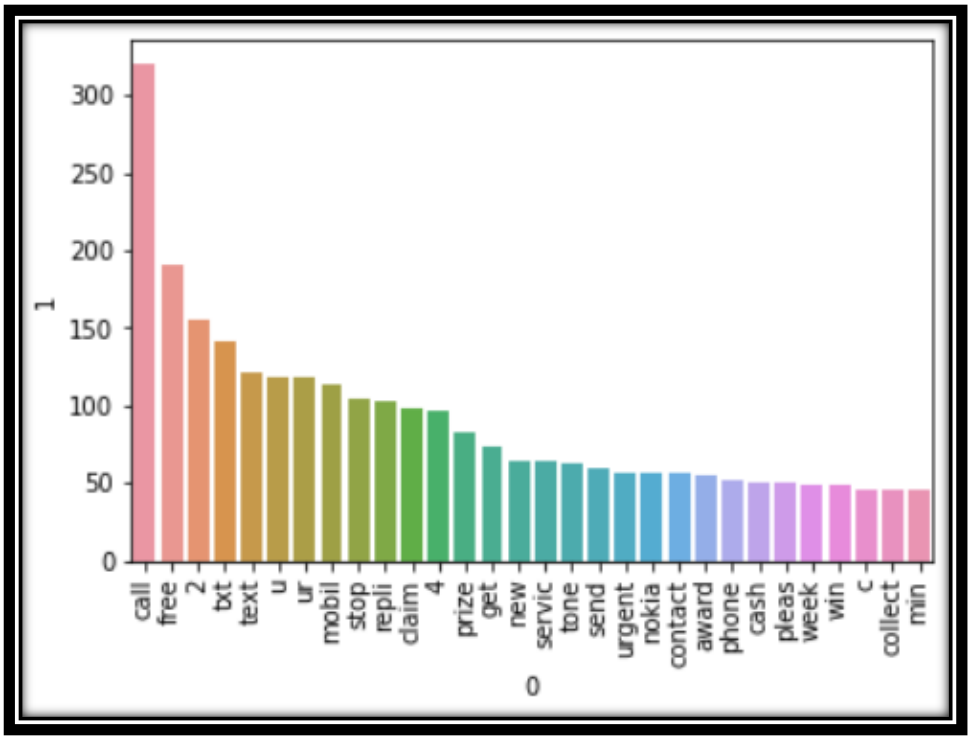

Fig 13: Bar-plot of Spam Tokens based on above WordPress Figures

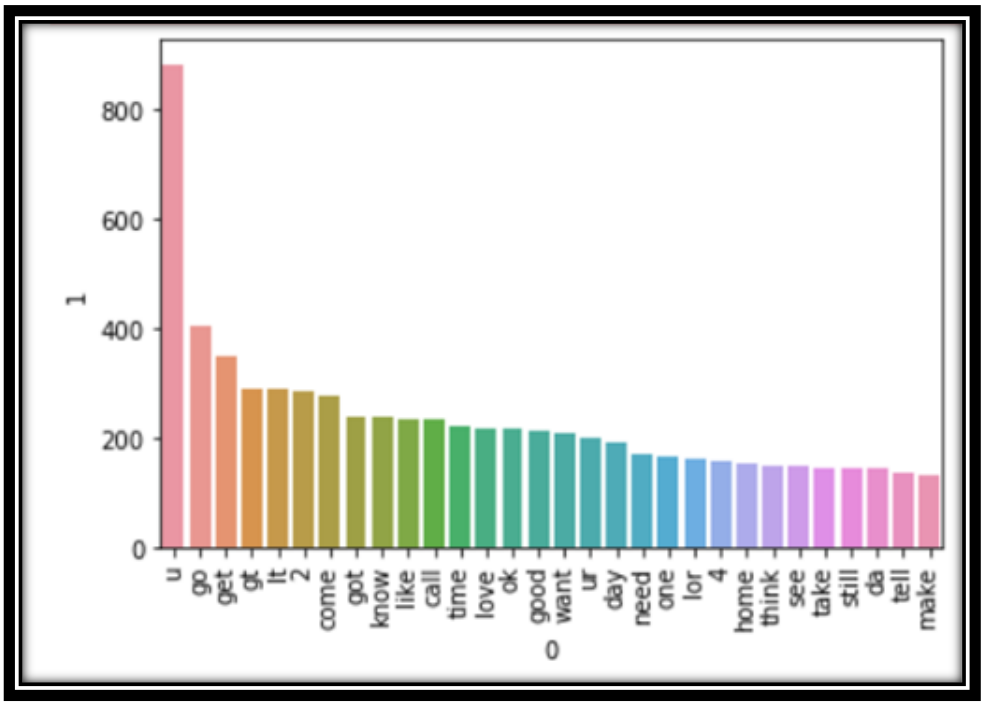

Fig 14: Bar-plot of Spam Tokens based on above WordPress Figures

5.1.5 Model Building:

Building a model in machine learning is creating a mathematical representation by generalizing and learning from training data. Then, the built machine learning model is applied to new data to make predictions and obtain results.

In [522…
```
from sklearn.feature_extraction.text import CountVectorizer,TfidfVectorizer
cv = CountVectorizer()
tfidf = TfidfVectorizer(max_features=3000)
```

In [523…
```
X = tfidf.fit_transform(df['transformed_text']).toarray()
```

In [470…
```
#from sklearn.preprocessing import MinMaxScaler
#scaler = MinMaxScaler()
#X = scaler.fit_transform(X)
```

In [483…
```
# appending the num_character col to X
#X = np.hstack((X,df['num_characters'].values.reshape(-1,1)))
```

In [524…
```
X.shape
```

Out[524…
```
(5169, 3000)
```

In [525…
```
y = df['target'].values
```

In [494…
```
from sklearn.linear_model import LogisticRegression
from sklearn.svm import SVC
from sklearn.naive_bayes import MultinomialNB
from sklearn.tree import DecisionTreeClassifier
from sklearn.neighbors import KNeighborsClassifier
from sklearn.ensemble import RandomForestClassifier
from sklearn.ensemble import AdaBoostClassifier
from sklearn.ensemble import BaggingClassifier
from sklearn.ensemble import ExtraTreesClassifier
from sklearn.ensemble import GradientBoostingClassifier
from xgboost import XGBClassifier
```

```
svc = SVC(kernel='sigmoid', gamma=1.0)
knc = KNeighborsClassifier()
mnb = MultinomialNB()
dtc = DecisionTreeClassifier(max_depth=5)
lrc = LogisticRegression(solver='liblinear', penalty='l1')
rfc = RandomForestClassifier(n_estimators=50, random_state=2)
abc = AdaBoostClassifier(n_estimators=50, random_state=2)
bc = BaggingClassifier(n_estimators=50, random_state=2)
etc = ExtraTreesClassifier(n_estimators=50, random_state=2)
gbdt = GradientBoostingClassifier(n_estimators=50,random_state=2)
xgb = XGBClassifier(n_estimators=50,random_state=2)
```

	Algorithm	Accuracy	Precision	Accuracy_max_ft_3000	Precision_max_ft_3000	Accuracy_scaling	Precision_scaling	Accuracy_num_chars	Precision_num_chars
0	KN	0.900387	1.000000	0.905222	1.000000	0.905222	0.976190	0.928433	0.771186
1	NB	0.959381	1.000000	0.971954	1.000000	0.978723	0.946154	0.940039	1.000000
2	ETC	0.977756	0.991453	0.979691	0.975610	0.979691	0.975610	0.976789	0.975000
3	RF	0.970019	0.990826	0.975822	0.982906	0.975822	0.982906	0.974855	0.982759
4	SVC	0.972921	0.974138	0.974855	0.974576	0.971954	0.943089	0.866538	0.000000
5	AdaBoost	0.962282	0.954128	0.961315	0.945455	0.961315	0.945455	0.971954	0.950413
6	xgb	0.971954	0.950413	0.968085	0.933884	0.968085	0.933884	0.970019	0.942149
7	LR	0.951644	0.940000	0.956480	0.969697	0.967118	0.964286	0.961315	0.971154
8	GBDT	0.951644	0.931373	0.946809	0.927835	0.946809	0.927835	0.948743	0.929293
9	BgC	0.957447	0.861538	0.959381	0.869231	0.959381	0.869231	0.968085	0.913386
10	DT	0.935203	0.838095	0.931335	0.831683	0.932302	0.840000	0.943907	0.877358

Fig 15: Accuracy VS Precision Relational Database of various ML Algorithms used

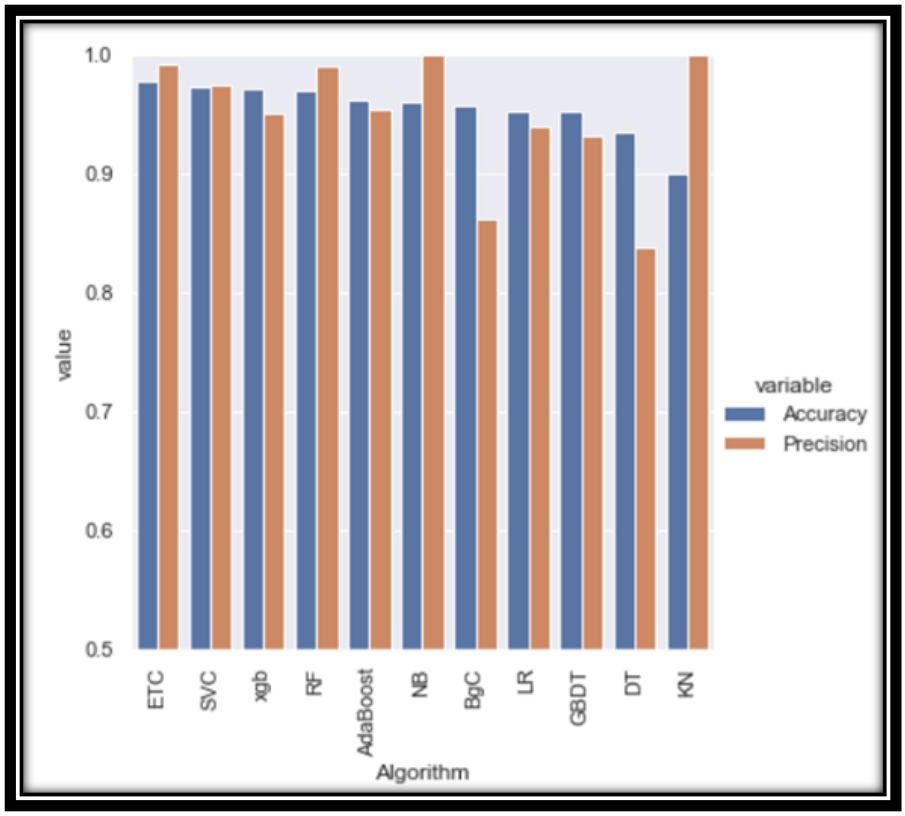

Fig 16: Accuracy VS Precision based of above table

In [521…

```
clf.fit(X_train,y_train)
y_pred = clf.predict(X_test)
print("Accuracy",accuracy_score(y_test,y_pred))
print("Precision",precision_score(y_test,y_pred))
```

```
Accuracy 0.9787234042553191
Precision 0.9328358208955224
```

Fig 17: Accuracy and Precision of the Comment Spam Classifier developed

Chapter 6

Results

After training our spam comment detection model with the spam.csv dataset, we have obtained a spam detector which can classify the comments into the two categories of "genuine comments" and "spam comments" with a high level of accuracy. The following images describe the working of our spam comment detector:

Input The Comment into the User box:

```
message= input()
You win Lottery 88$
```

The Result Displaying Whether the Comment is Genuine or Spam:

```
MultinomialNB() This is a Spam email

SVC(C=1000, gamma=0.001) This is a Spam email

KNeighborsClassifier(n_neighbors=3) This is a Real email

DecisionTreeClassifier() This is a Spam email

RandomForestClassifier() This is a Spam email


#Let's make a predict with Naive Bayes
spam_dect(clf_NaiveBaised, message)

MultinomialNB() This is a Spam email
```

Chapter 7

Limitations

Some critical challenges faced by spam filters are discussed as follows:

I. The growing amount of data on the Internet with various new features is a big challenge for spam detection systems.

II. Features' evaluation from several dimensions such as temporal, writing styles, semantic, and statistical ones is also challenging for spam filters.

III. Most of the models are trained on balanced datasets, while self-learning models are not possible.

IV. Many spam detection models face adversarial machine learning attacks that will decrease their effectiveness. Adversaries can throw a variety of attacks during the training and testing of ML models. Adversaries can harm training data to cause a classifier to classify the data incorrectly (poisoning attack), create unfavorable samples during testing to evade detection (evasion attack), and obtain sensitive training data via a learning model (privacy attack).

Chapter 8

Conclusion

In the last two decades, spam detection and filtration gained the attention of a sizeable research community. The reason for a lot of research in this area is its costly and massive effect in many situations like consumer behavior and fake reviews. The survey covers various machine learning techniques and models that the various researchers have proposed to detect and filter spam in emails and IoT platforms. The study categorized them as supervised, unsupervised, reinforcement learning, etc. The study compares these approaches and provides a summary of learned lessons from each category. This study concludes that most of the proposed email and IoT spam detection methods are based on supervised machine learning techniques. A labelled dataset for the supervised model training is a crucial and time-consuming task. Supervised learning algorithms SVM and Naive Bayes outperform other models in spam detection. The study provides comprehensive insights of these algorithms and some future research directions for email spam detection and filtering.

Chapter 9

Future Prospects

9.1 Increasing the Accuracy

Although our project is fairly accurate in detecting spam or not spam emails, it is still not perfect and there is scope for improvement. One big drawback for us is that the data set we are using is relatively small, however if we had larger data nearly of the order of one hundred thousand then our spam classifier model would be much more well trained and our accuracy would be much more than it is currently. There is another problem with the data set we are using and that is the fact that there is an imbalance in the number of spam and ham messages that we have in the data.

9.2 Instant Detection

Another problem is that since we have deployed our project on google collab, the users have to manually go over there and type their message to see if it is spam or not. This is not very user friendly. So, for potential future changes, one thing that we definitely need to do is to integrate our project model with the user's already existing emailing system so that when the users receive the emails, our model automatically tags it with "spam" or "not spam" so that the users can immediately identify it. This way the users will not waste extra time again retyping/copy-pasting this message onto our website and checking. So, integrating our spam detector model with the current comment system of the user is the future of our project. It is only with this step that our project can actually be put to good use for the user's benefit immediately.

Chapter 10

References

[1] https://en.wikipedia.org/wiki/Email_spam

[2] https://www.cisco.com/c/en_in/products/security/email-security/what-is-spam.html#~types-of-spam

[3] https://www.kaggle.com/datasets/uciml/sms-spam-collection-dataset/code

[4] https://www.researchgate.net/publication/319370844_Classification_Techniques_in_Machine_Learning_Applications_and_Issues

[5] Introduction to Machine Learning with Python: A Guide for Data Scientists Book by Andreas C. Müller and Sarah Guido

[6] https://www.sciencedirect.com/science/article/pii/S2405844018353404

[7] https://www.ijirt.org/master/publishedpaper/IJIRT156181_PAPER.pdf

[8] Yüksel, A. S., Cankaya, S. F., & Üncü, İ. S. "Design of a Machine Learning Based Predictive Analytics System for Spam Problem", 2017