# SMS Spam Classification

*Project report submitted to the Amrita Vishwa Vidyapeetham in partial fulfilment of the requirement for the Degree of*

**BACHELOR of TECHNOLOGY**

**in**

**COMPUTER SCIENCE AND ENGINEERING**

*Submitted by*

**Karpurapu Rahul AM.EN.U4CSE18505**
**Pasupuleti Bhanu Prakash AM.EN.U4CSE18142**
**Pothula Rohith Kumar Reddy AM.EN.U4CSE18141**
**Bodapati Abhi Teja AM.EN.U4CSE18112**


**AMRITA SCHOOL OF ENGINEERING**

**AMRITA VISHWA VIDYAPEETHAM**
**(Estd. U/S 3 of the UGC Act 1956)**
**AMRITAPURI CAMPUS**

**KOLLAM -690525**

**MAY 2021**

**DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING**

**AMRITA VISHWA VIDYAPEETHAM**
**(Estd. U/S 3 of the UGC Act 1956)**

**Amritapuri Campus**

**Kollam -690525**

# BONAFIDE CERTIFICATE

This is to certify that the project report entitled **"SMS Spam Classification"** submitted by **Karpurapu-Rahul(AM.EN.U4CSE18505),PasupuletiBhanuPrakash(AM.EN.U4CSE18142), PothulaRohithKumarReddy(AM.EN.U4CSE18141) and BodapatiAbhiTeja(AM.EN.U4CSE18112))**, in partial fulfillment of the requirements for the award of Degree of Bachelor of Technology in Computer Science and Engineering from Amrita Vishwa Vidyapeetham, is a bonafide record of the work carried out by them under my guidance and supervision at Amrita School of Engineering, Amritapuri during Semester 8 of the academic year 2020-2021.

Your Guides Name
Project Guide

Coordinator name
Project Coordinator

Dr. Siji Rani
Chairperson
Dept. of Computer Science & Engineering

Reviewer

Place : Amritapuri
Date : 17 May 2021

**DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING**

**AMRITA VISHWA VIDYAPEETHAM**
**(Estd. U/S 3 of the UGC Act 1956)**

**Amritapuri Campus**

**Kollam -690525**


# DECLARATION

We, **Karpurapu Rahul(AM.EN.U4CSE18505),Pasupuleti Bhanu Prakash (AM.EN.U4CSE18142),Pothula Rohith Kumar Reddy(AM.EN.U4CSE18141),Bodapati Abhi Teja(AM.EN.U4CSE18112)** hereby declare that this project entitled **"Name of your project"** is a record of the original work done by us under the guidance of **Your Guides name**, Dept. of Computer Science and Engineering, Amrita Vishwa Vidyapeetham, that this work has not formed the basis for any degree/diploma/associationship/fellowship or similar awards to any candidate in any university to the best of our knowledge.

Place : Amritapuri

Date : 17 May 2021

Signature of the student

Signature of the Project Guide

**Abstract**

The fast growth of mobile networks has greatly enriched our life disseminating information and providing communications at any time and anywhere. However, at the same time, when people gather and exchange useful information, they also receive unwanted data and content, such as spam and unsolicited commercial advertisements. SMS (Short Message Service) spam is one typical example of unwanted content, which has caused a serious problem to mobile users by intruding on their devices, occupying device memories and irritating the users.

The easy accessibility and simplicity of Short Message Services (SMS) have made it attractive to malicious users thereby incurring unnecessary costing on the mobile users and the Network providers' resources.

SMS has reached more than 6 billion users globally with approximately 9.5 trillion SMS sent globally in 2009. This tremendous growth in mobile devices has made SMS a very attractive area to malicious organizations for carrying out illegal activities and influencing security risks such as SMS spam, Phishing, License to Kill Spyware, Malware, and privacy issues to mobile data.

Based on the statistics from the Singapore Police Force, from January till June 2020, the amount cheated through scams have increased by more than 8 million dollars

# Contents

# List of Figures

# List of Tables

# Chapter 1

# Problem Definition

Short Message Service (SMS) is one of the well-known communication services in which a message sends electronically. Spam messages include advertisements, free services, promotions, awards, etc. People are using the ubiquity of mobile phone devices is expanding day by day as they give a vast variety of services by reducing the cost of services. Short Message Service (SMS) is one of the broadly utilized communication service. This has prompted an expansion in mobile phones attacks like SMS Spam. In this problem, preliminary results are mentioned or explained herein based on real time dataset(by extracting messages from phones and making dataset).

# Chapter 2

# Related Work

## A weighted feature enhanced HMM for spam SMS filtering

## Pre-process

In order to remove redundant information for better processing, every SMS message is first pre-handled toward the start. The pre-process incorporates tokenization and stops word expulsion.

### Tokenization

Tokenization is to extract the words from an SMS message body. For some oriental languages, such as Chinese, Korean and Japanese, words are firmly associated without spaces in a sentence. In this manner, the SMS messages in these languages must be first partitioned into words and punctuations sequence through a division interaction.

### SMS word weighting

SMS words might contain diverse semantic data and the SMS word weighting calculation can allocate a legitimate weight worth to each word. A word weight value demonstrates the shot at recognizing a spam or ham SMS message with the word's event. A word that conceivably happens in the spam set has a more modest negative weight and a word which in all likelihood shows up in the ham set has a greater positive weight. It indicates the HMM to think about additional significant words and be prepared unequivocally ideal.

## SMS property prediction

While foreseeing the SMS property, the SMS messages in the testing set are first pre-handled as same as the preparing set. After tokenization and stop-word evacuation, the testing SMS messages become word arrangements with the first request. Then, at that point, these words acquire the word loads in the preparing set. They are subbed by the heaviness of a similar word in preparing a set. For the words that don't exist in the preparation set, known as obscure words, weight 0 is applied. So the word arrangements become weight successions which are the perception groupings. Individually, each weight grouping of the testing SMS messages takes care of the prepared HMM which deciphers spam or ham name arrangement for each testing SMS message thus.

# Soft techniques for SMS spam classification: Methods, approaches and applications

## Importance of spam detection

The continuous escalation of mobile devices over the years has given users an unbeatable communication experience which has increased users' performance efficiently. The most popular and widely used service of the Global System for Mobile communication (GSM) is the Short Message Services known as SMS. This fast and ever-growing service has reached more than 6 billion users globally with approximately 9.5 trillion SMS sent globally in 2009. This tremendous growth in mobile devices has made SMS a very attractive area to malicious organizations for carrying out illegal activities and influencing security risks such as SMS spam, Phishing, License to kill Spyware, Malware, and privacy issues to mobile data. So the SMS span detection is very important area.

# Aim

The aim of this paper is to identify and review existing state of the art methodology for SMS spam based on some certain metrics: AI methods and techniques, approaches and deployed environment and the overall acceptability of existing SMS applications.

# Content Based Approach

It involves the uses of words or character frequency mostly called bag of words for document representation. Here, the frequencies of words are used

as features within a classification method for detecting spamicity. Spamicity is the factor of the frequency of occurrence of the same words in the token database with assigned values to each word between the range 0.0 to 1.0. In Bag of words-based approaches, the sequence of words and their semantic relations are not considered.

## Non Content Based Approach

On the contrary, non-content based approaches utilize certain message characteristics or signature patterns as features for detecting anomalies within a network. These features could be based on a static measure (total number of messages sent per time); the size of the message and time stamp.

## Hybrid Method

Hybrid approaches combine features from content and non-content based approaches for classification purposes.

## Architecture

Based on the 83 selected studies, existing SMS spam filter architecture is based on three main layers which are: Client-based (solution resides on the mobile device), Server-based (solutions reside at the network provider's side, SMSC) and collaborative-based (solutions reside on both the client and the server layer). Quite a number of the selected studies solutions were based at the client side.

## Dataset

The issue of ensuring user's privacy is a major factor affecting SMS data collection globally. Most of the existing English SMS corpus is based on reusability and compilation of smaller SMS databases to build more robust databases, examples of some databases formed from smaller ones include UCI corpus (comprises of NUS corpus, Caroline Tag and Grumble text), British English Corpus (comprises of Caroline Tag and Grumble text corpus).

## Discussion, limitations and taxonomy

This survey emphasizes the overall research contributions on SMS spam detection and classification while also presenting a summary of existing studies based on their state-of-the-art-methodologies, approaches, architectures, status, SMS databases and existing anti-spam solutions. Classification of short messages is vital in ensuring that the users' security is preserved since mobile devices are important tools for daily activities. The significance of SMS spam classification is numerous. Based on the selected publications reviewed in this study, a lot of work has been done to solve the SMS spam problem using machine learning. However, some methods are yet to be fully explored such as Deep learning. This method is being maximized in areas like text mining, image processing, pattern recognition, etc. Based on the selected publication, this survey presents an overall summary of the pros and cons of the different areas involved in SMS spam classification as shown in Table above.

# Conclusion and future direction

The analysis result obtained for each search strategies are as follows: Existing methods on SMS spam shows that Machine learning (ML), Statistical analysis and evolutionary methods are 49

# Chapter 3

# Requirements

The design of this project contains both hardware and software. The specifications are listed below.

## Hardware

Leveraging the power of cloud computing, we have used Python 3 Google Compute Engine for the processing power. The provisioned server has 12GB Ram and 108GB of storage. the storage was used to upload the dataset and work on it.

## Software

The algorithm was implemented in Google Collab in python using multiple libraries. The libraries include

1. NUMPY
2. PANDAS

3. NLTK
4. SKLEARN
5. MATPLOTLIB
6. SEABORN

# Chapter 4

# Proposed System

This section will discuss the overall proposed methodology for SMS classification. Here, we constructed two models, model A, B. Model A will classify the messages as ham or spam. However, model B classifies ham messages into four more classes, i.e., OTP, transaction, delivery, and personal. Figure 4.1 and Figure 4.2 show the abstract view of the proposed models .As shown in Figure 4.1, there are six steps namely 1) Data Collection, 2) Data Preprocessing, 3) Feature Engineering, 4) Data Splitting, 5) Predictive Model Construction, and 6) Model Evaluation. However, Figure 4.2 shows the steps followed for model B same as model A except the data collection and data preprocessing. Furthermore, section 4.1.1 shows the exploratory data analysis on the given dataset. The details are explained in subsequent sections.

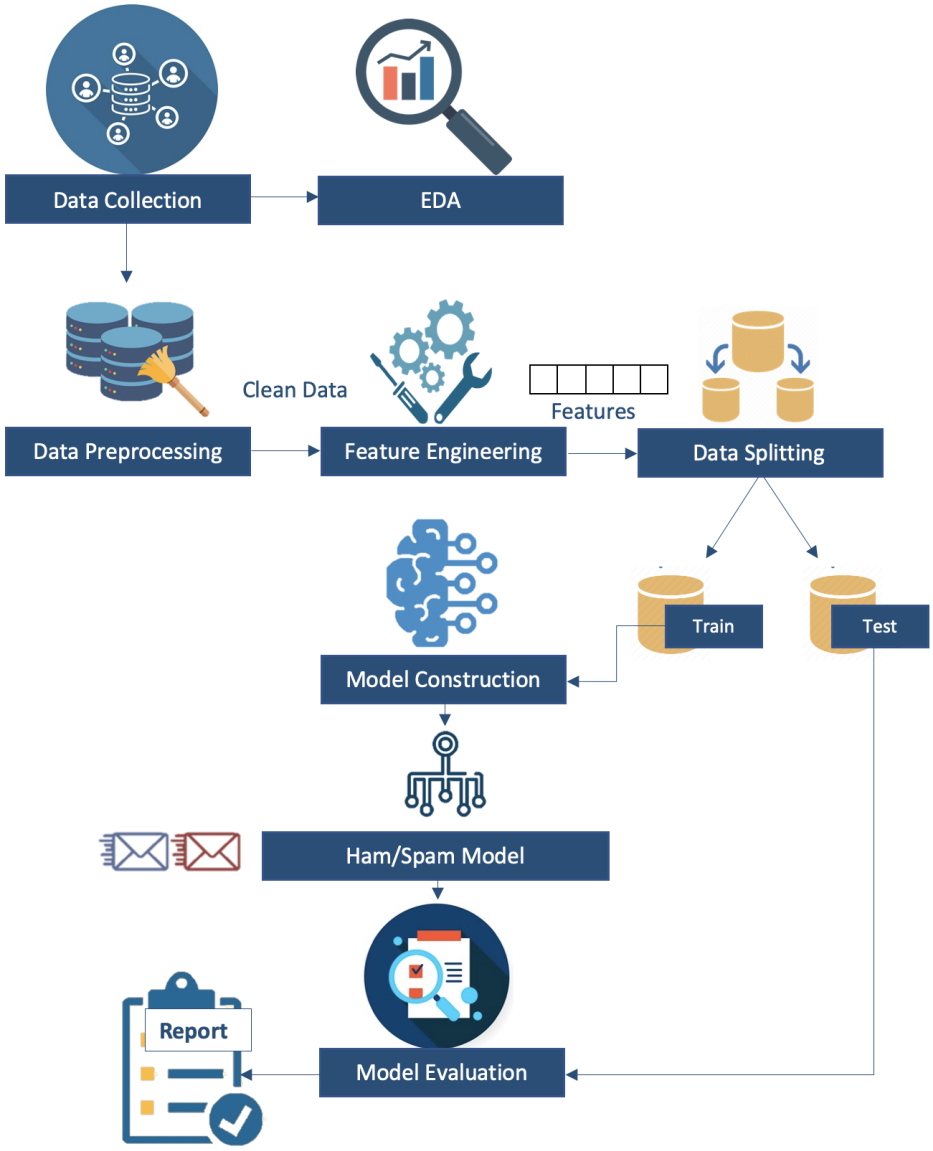


Figure 4.1: Proposed Methodology for Ham and Spam Classification

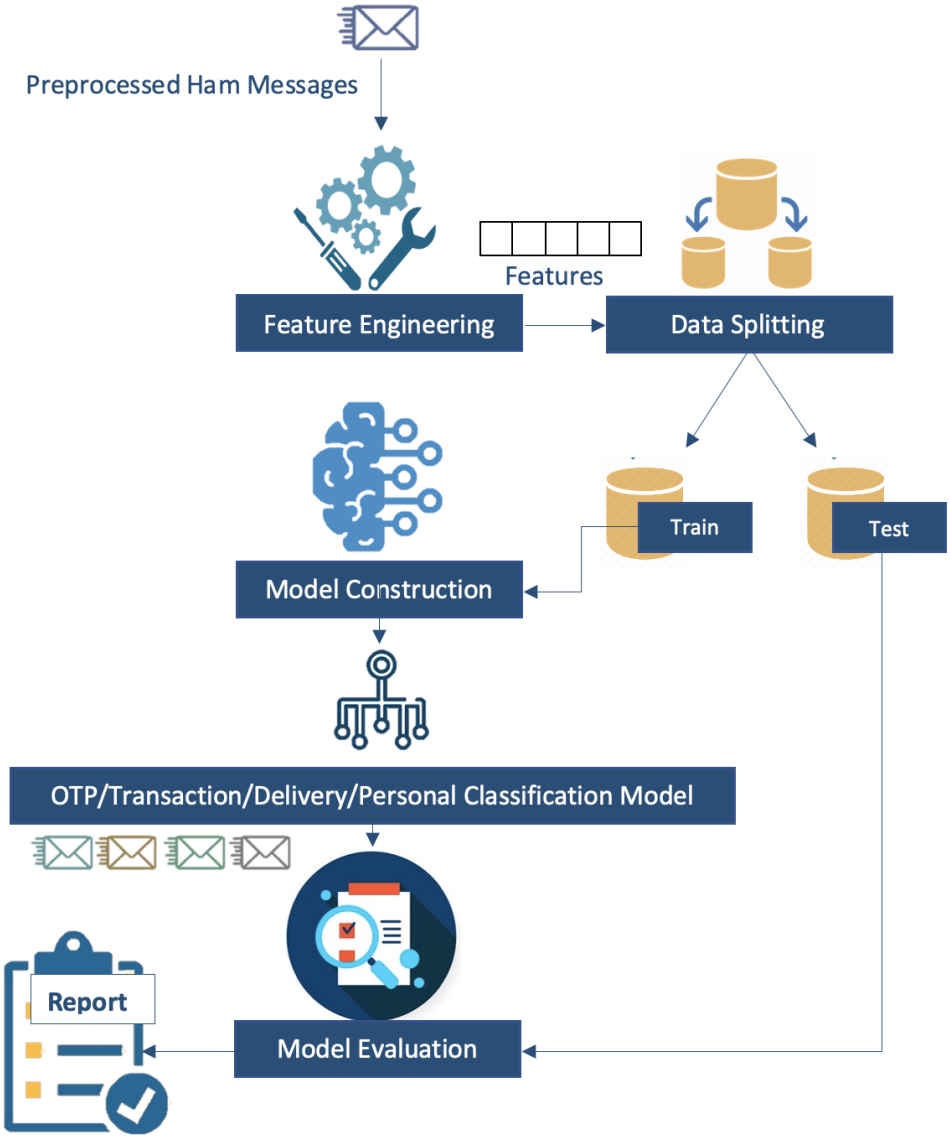


Figure 4.2: Proposed Methodology for OTP/transaction/delivery/personal

## 4.1 Data Collection

In India, there are many ways to spam people through SMS, getting random messages like you are pre-approved for a loan, you have won 10,000/- and to claim it, you need to click the given links. Therefore, there is a need to identify spam messages. For ham/spam classification models, many datasets

are available throughout the internet, but most conversations are from the US and UK. To the best of our knowledge, there is no dataset having Indian conversation messages. Therefore, to keep this scenario, the data is created from scratch. The curated data has conversations specifically from India. The dataset consists of 3893 data points. First, we labeled the collected data as ham or spam using the manual labeling approach. Second, we performed the below steps to label the ham SMS as OTP, transaction, delivery, or personal.

1. Because, the dataset is in three languages (i.e. English, Hindi, and Tamil). First, we translated the Hindi and Tamil data into English using the google translate API.

2. Next, the data points having the words "otp, verification code, one time password, use code, login code, code , the code, whatsapp code" are labeled as the "OTP" category, the data points having the words "debit, credit, txn, bank, debited, credited, a/c, withdrawn, upi, atm, transferred, withdrawal, cash withdrawal, transaction" are labelled as "Transaction" category and the data points having the words "shipped, delivered, track id, shipment, shipment id, ready to ship, tracking id, order, picked, out for delivery, packed, placed, ready to ship, out for pickup, confirmed, arriving, dispatched" are labelled as "Delivery" category. Finally, the remaining data points are labeled as the "Personal" category.

Figure 4.3 shows the final shape of dataset in terms of class labels. Furthermore, the details of the exploratory data analysis is given in the below section.

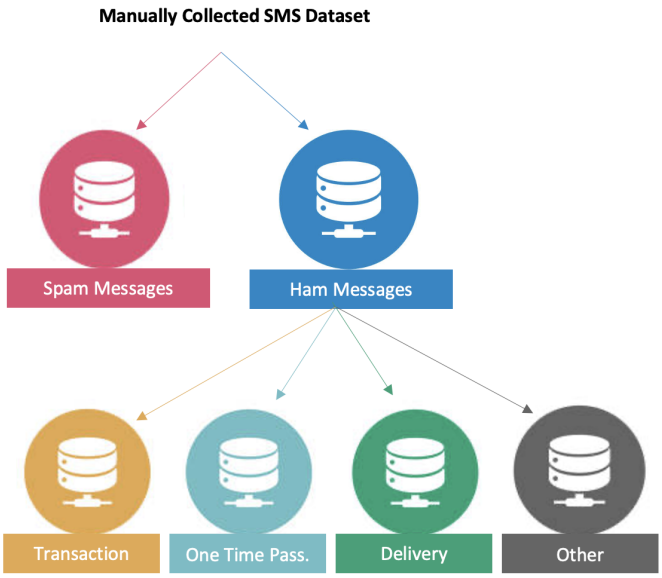


Figure 4.3: Dataset in Terms of Labels

### 4.1.1 Exploratory Data Analysis

Before constructing the models for SMS classification, it is necessary to become familiar with the dataset. The initial investigation of data is known as exploratory data analysis, or EDA [1]. The primary focus of EDA is to inspect the data using visuals. Data visualization helps to tell stories by curating data into an easy form to understand, highlighting trends and outliers. A good visualization tells a story, removing the noise from data and highlighting more information. Therefore, in this study, we performed exploratory data analysis. The details have given below.

1. The first step is to check the distribution of the target variables, i.e., the number of spam/ham and OTP/Transaction/Delivery/Personal messages. It also helps to understand how messages are distributed evenly and hint at the performance of the dataset. Figure 4.4 shows that the dataset has 72% of SMS belonging to ham and the remaining 28% SMS

belonging to the spam category. However, Figure 4.5 shows the distribution of ham messages using OTP, transaction, personal, and delivery classes. As shown here, 55%, 32%, 8%, and 5% of SMS belong to the personal, transaction, OTP, and delivery classes,respectively. However, Table 4.1 and Table 4.2 show the data points distribution of ham/spam and OTP/transaction/personal/delivery categories, respectively.

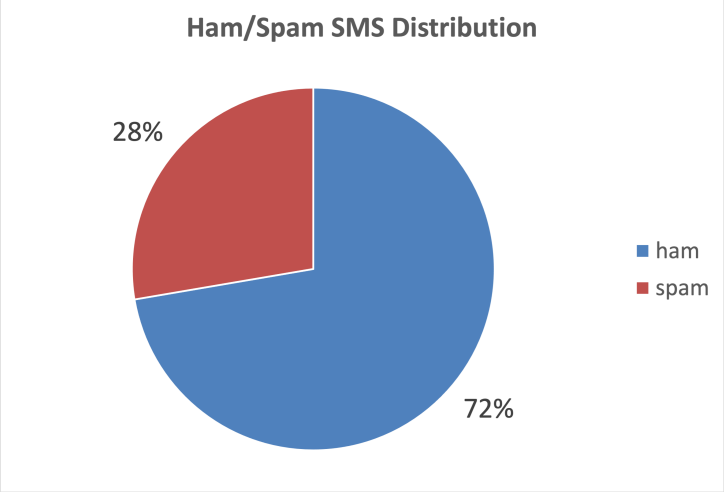


Figure 4.4: Dataset Distribution on Ham/Spam

| | **Total Percentage** | **Datapoints** |
|---|---|---|
| **Ham** | 72% | 2815 |
| **Spam** | 28% | 1078 |
| | **Total** | 3893 |

Table 4.1: Ham/Spam Data points in Dataset

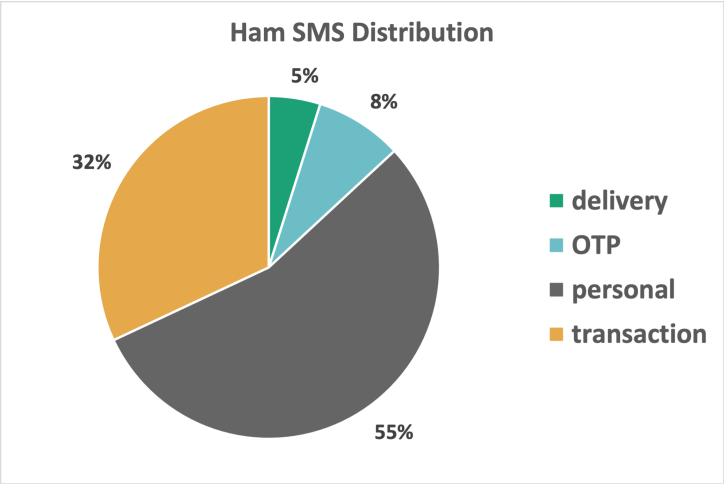


Figure 4.5: Ham SMS Distribution

| | **Total Percentage** | **Data Points** |
|---|---|---|
| **Personal** | 55% | 1546 |
| **Transaction** | 32% | 900 |
| **OTP** | 8% | 232 |
| **Delivery** | 5% | 137 |
| | **Total** | 2815 |

Table 4.2: Personal/Transaction/OTP/Delivery Data Points

2. The length of a message may also help to predict whether a message is a ham or spam. For example, an SMS too short containing a link is often spam, or an SMS too long persuading you into a product may also be spam. Graphs in Figure 4.6 show the length of ham and spam messages. As shown here, the ham messages are shorter than spam messages.

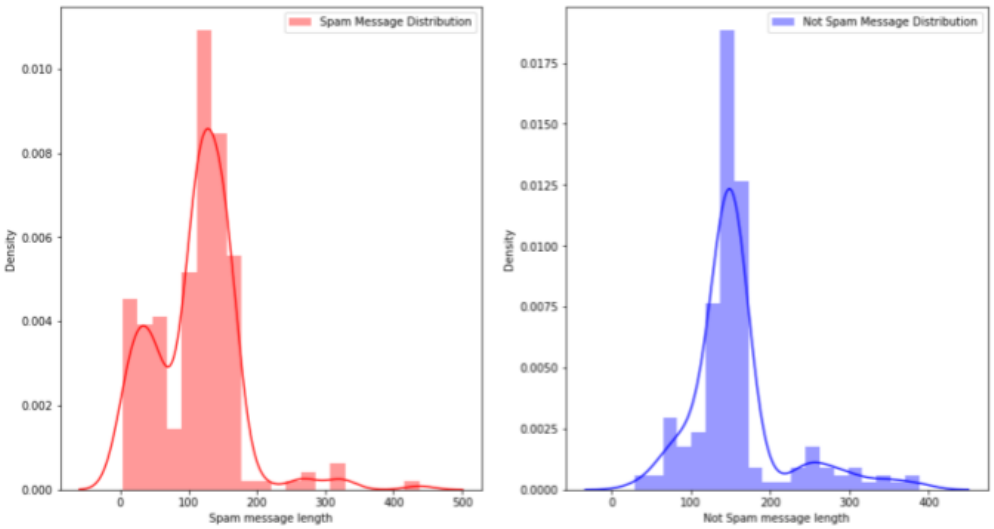


Figure 4.6: Length of Ham and Spam Messages

3. In the exploratory data analysis part, we used the word cloud[2] to understand more about data. Word cloud gives a rough estimate of the sentiment in the messages. Word cloud shows the words that occur the most and the least. Additionally, word clouds help to understand the content of the dataset. Figure 4.7 and Figure 4.8 show the word clouds of ham and spam messages, respectively. As shown here, spam messages mostly contain words like click, Rs, free, HTTPS, visit, offer, etc. Furthermore, the word clouds of personal, transaction, OTP, and delivery messages are shown in Figure 4.9, Figure 4.10, Figure 4.11, and Figure 4.12, respectively. Figure 4.10 shows that in transaction messages, the frequent words are account, money, balance, debit card, etc. However, the word cloud for OTP shows that words like OTP, code, use code, one time, share, pay, time password words are more frequent than others. Furthermore, for class delivery, the most frequent words are order, amzn, delivered, track, shipped, etc.

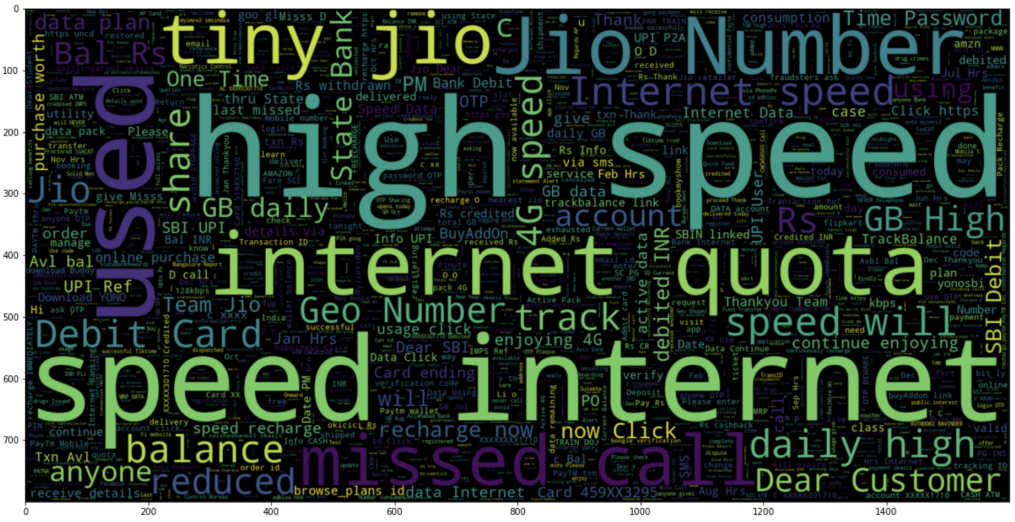


Figure 4.7: Ham Word Cloud

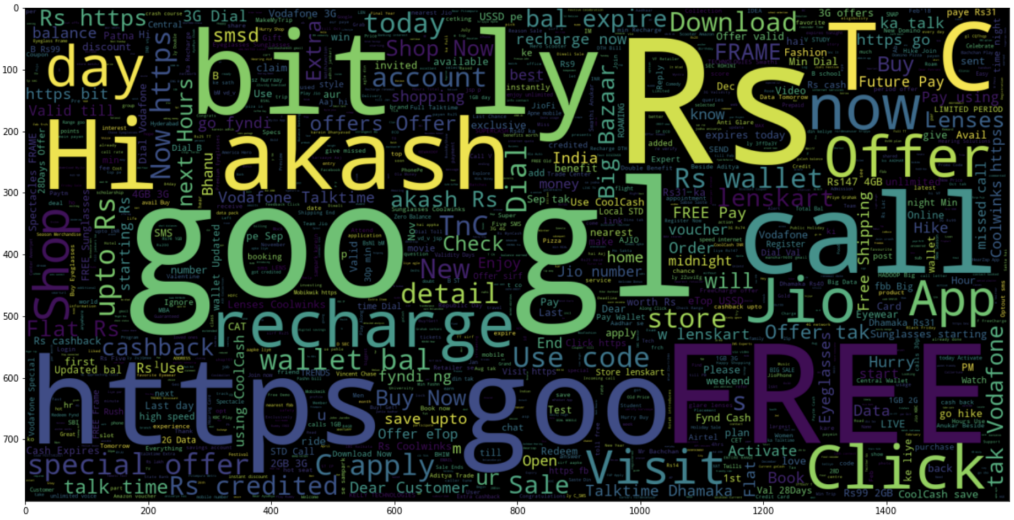


Figure 4.8: Spam Word Cloud

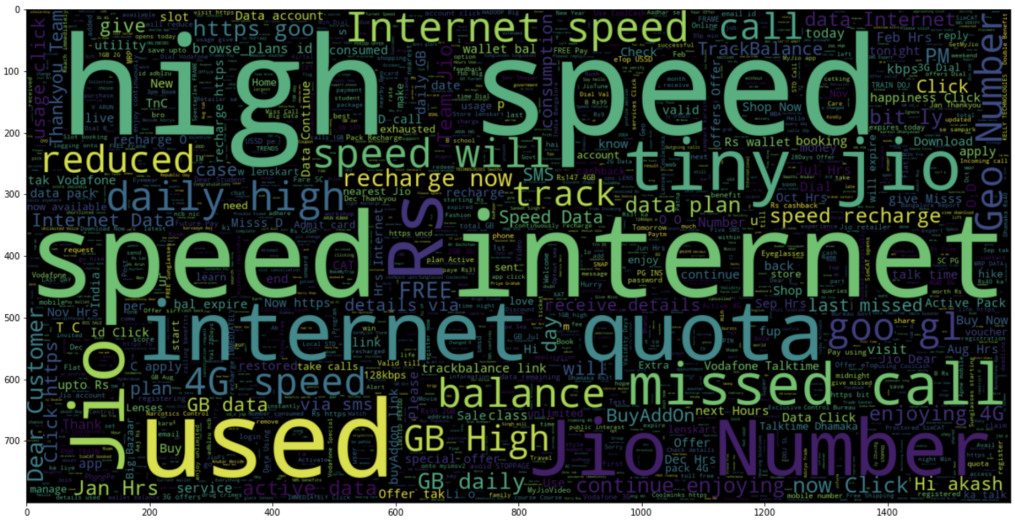


Figure 4.9: Personal Messages Word Cloud

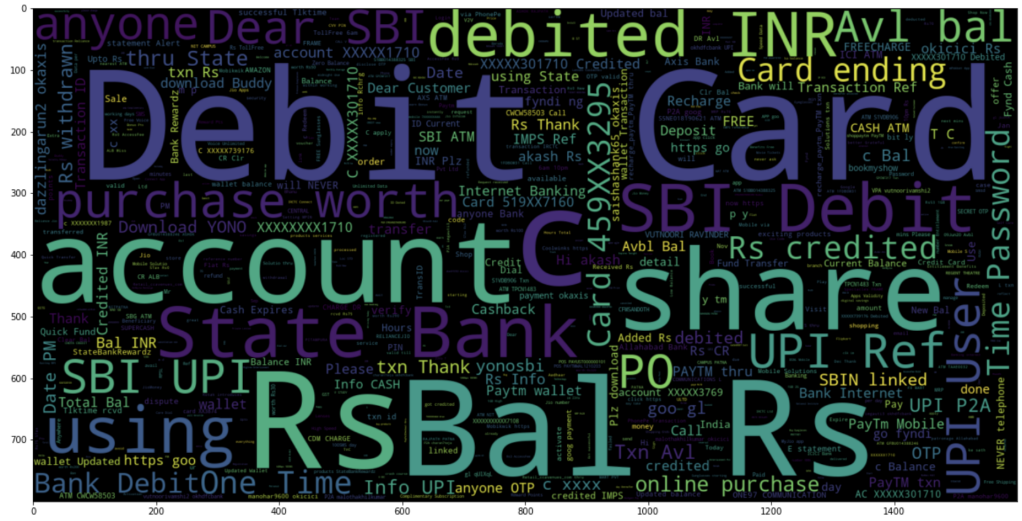


Figure 4.10: Transaction Messaged Word Cloud

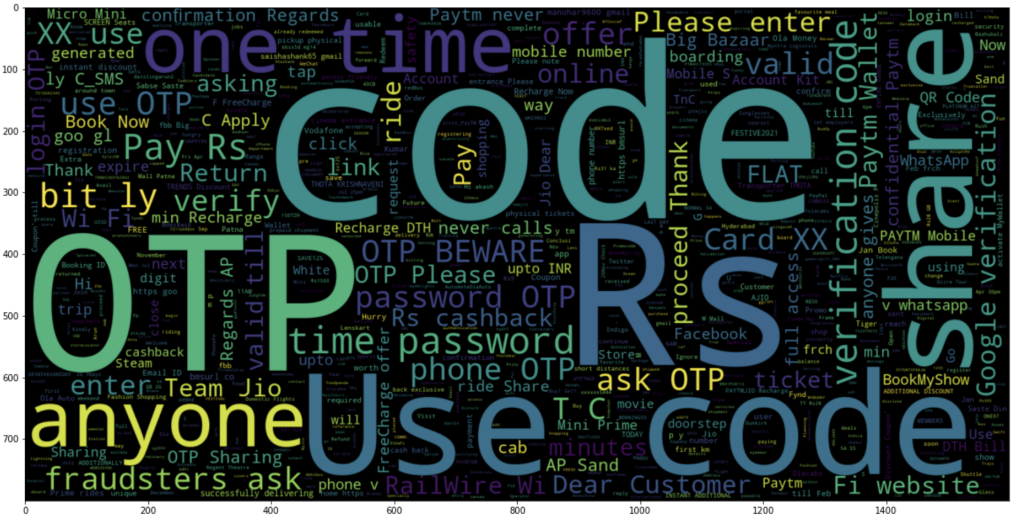


Figure 4.11: OTP Messages Word Cloud

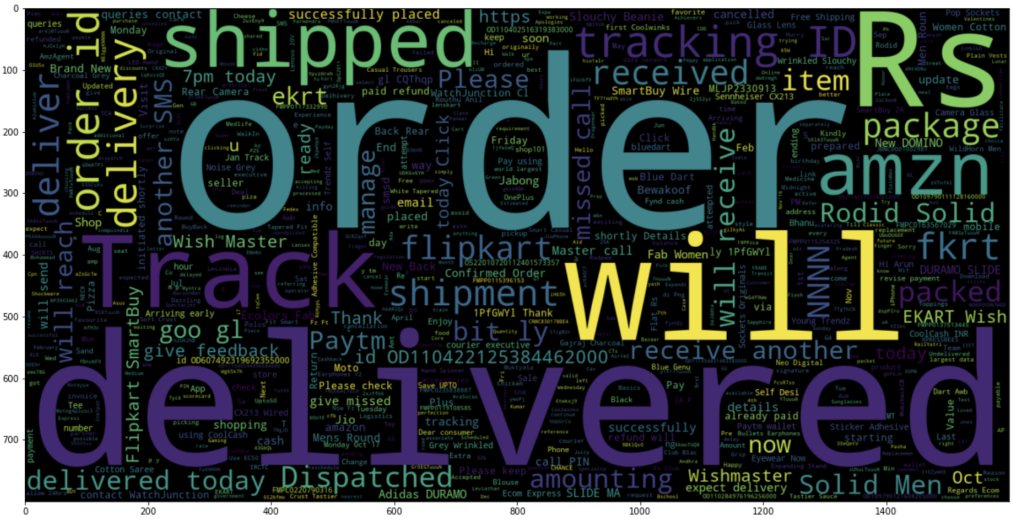


Figure 4.12: Delivery Messages Word Cloud

## 4.2 Preprocessing

Preprocessing is a data mining technique that transforms raw data into an understandable format. There are the following possible reasons to preprocess the data. First, past studies have shown that predictive models with preprocessed data give better classification results than raw or uncleaned

data [3]. Secondly, there are pieces of information that are not important for predictive models, e.g., upper cases, stop words, etc. Lastly, as real-world data is incomplete, it is necessary to preprocess the data before going through a model. In this study, many tasks are performed for cleaning and preprocessing the collected data. As proposed models based on text classification, there is a need to remove unnecessary or non-informative features like upper cases, URLs, usernames, or hashtags from an SMS text. We used the pattern matching technique to remove the noise. We also used a few natural language preprocessing techniques, i.e., removing punctuation and stopwords. The punctuation (i.e., full stops, commas, etc.) and stopwords (i.e., words like a the, is, are) act as noise to the model. Therefore, it is essential to remove them from the text. This project used the predefined set of stop words given by the Natural Language Tool Kit to remove stop words to avoid noise for the model. Figure 4.13 explains the preprocessing steps with an example.


Figure 4.13: Data Preprocessing

## 4.3 Feature Engineering

In machine learning, the predictive models do not understand the predictive rules from the raw data. These models work on numeric vector to understand the predictive rules. In machine learning, feature engineering is an essential step to transfomed the raw text into vectors. This step extracts the features from raw data and represent these features are a numeric vector. In this study, we have used feature engineering technique named count vectorizationand TF-IDF [4]. The details are given in subsequent sections.

### 4.3.1 Count Vectorization

Count Vectorization is a great tool used to transform the given text into a vector-based on the frequency count of each word that occurs in the entire text.

**Key Observations**

1. There are 1913 unique words in the document, represented as columns of the table.

2. There are 412 text samples in the document, each represented as rows of the table.

3. Every cell contains a number, that represents the count of the word in that particular text.

4. All words have been converted to lowercase.The words in columns have been arranged alphabetically

### 4.3.2 TF-IDF

TF-IDF is a method which gives us a numerical weightage of words which reflects how important the particular word is to a document in a corpus. Tf is Term frequency, and IDF is Inverse document frequency. Tf(Term Frequency): Term frequency can be thought of as how often does a word 'w' occur in a document 'd'. IDF(inverse document frequency): Sometimes, words like 'the' occur a lot and do not give us vital information regarding the document. To minimize the weight of terms occurring very frequently by incorporating the weight of words rarely occurring in the document. Combining these two we come up with the TF-IDF score. In this study, tri-gram

with TFIDF was used to extract the numeric features that would be used for model construction.

## 4.4 Data Splitting

In machine-learning, splitting data into train and test data is commonly used approach before model construction. Train data would be used by model to learn the classification rules. However, test data used for model evaluation. In this study, we used the Pareto Principle to split the given data i.e., "80% of effects come from 20% of causes" [5]. This principle is also known as 80:20 ratio. Using this principle, 80% and 20% of data will be used for training and testing, respectively. Table 4.3 and Table 4.4 show the splitting of data on basis of ham/spam and otp/delivery/personal/transaction, respectively.

| | **Class** | **Total Instances** | **Train Set** | **Test Set** |
|---|---|---|---|---|
| **0** | **Ham** | 2815 | 2262 | 553 |
| **1** | **Spam** | 1078 | 852 | 226 |
| | **Total** | **3893** | **3114** | **779** |

Table 4.3: Ham/Spam: Details of Dataset after Data Splitting

| | **Class** | **Total Instances** | **Train Set** | **Test Set** |
|---|---|---|---|---|
| **0** | **OTP** | 232 | 191 | 41 |
| **1** | **Delivery** | 137 | 101 | 36 |
| **2** | **Personal** | 1546 | 1242 | 304 |
| **3** | **Transaction** | 900 | 718 | 182 |
| | **Total** | **2815** | **2252** | **563** |

Table 4.4: OTP/delivery/personal/transaction: Details of Dataset after Data Splitting

## 4.5 Model Construction

This section explains the model construction. As discussed earlier, two models are constructed for SMS classifications. The details of the models are

given in subsequent section.

### 4.5.1 Model A: Ham/Spam SMS Classification Model

In classification, past literature showed that no machine-learning algorithm performs best on all kinds of data [6]. Hence, researchers deploy different machine learning algorithms on the same data to discover the best-performing model. Therefore, we also trained five different machine-learning models namely LR[7], NB[8], SVM[9], RF[10], and ensemble classifier[11] using the given vector of train data. The training models will predict whether SMS is spam or ham using test data.

### 4.5.2 Model B: OTP/Transaction/Delivery/Personal SMS Classification Model

Furthermore, the ham SMS could also classify into subcategories. Therefore, we also trained a model that classifies whether a ham SMS belongs to OTP, transaction, delivery, or personal class. Same like, ham/spam classification, we also trained five different machine-learning models using train data for second-level classification i.e., LR[7], NB[8], SVM[9], RF[10], and ensemble classifier[11].

## 4.6 Model Evaluation

In classification, precision, recall, accuracy, and F-measure score are commonly used metrics to evaluate the performance of trained models. As Figure 4.4 and Figure 4.5 show that the instances of given dataset are not equally distributed. Therefore, we used precision, recall, F-measure, and ac-

curacy to evaluate the performance of constructed models of ham/spam and OTP/transaction/delivery/personal. Seliya et al.[12] have discussed these performance metrics and their relationship in detail.

# Chapter 5

# Experimental Results and Analysis

This section discusses the different experimental results observed during the deployment and evaluation of predictive models (i.e., Model A for ham or spam classification, Model B for OTP, transaction, delivery, and personal SMS classification). Additionally, this section also compares the performance of deployed machine-learning models. The subsequent sections discuss the details of all obtained results.

## 5.1 Ham/Spam SMS Classification Results

As mentioned in section 4.5, we have deployed different machine-learning algorithms to predict whether an SMS is a ham or spam. Table 5.1 shows the results obtained by different models using test data, i.e., 20% of data. As shown here, SVM achieved the highest accuracy, i.e., 93%, followed by the RFC and ensemble classifiers with an accuracy of 92%.

| | | Accuracy | Precision | Recall | F-Measure |
|---|---|---|---|---|---|
| **3-gram** | **LR** | 90 % | 0.90 | 0.90 | 0.89 |
| | **MNB** | 88% | 0.90 | 0.88 | 0.87 |
| | **RFC** | 92% | 0.93 | 0.92 | 0.92 |
| | **SVM** | 93% | 0.93 | 0.93 | 0.92 |
| | **Ensemble** | 92% | 0.92 | 0.92 | 0.92 |

Table 5.1: Ham/Spam Classification Results

### 5.1.1 Ham/Spam Classification: Models Comparison

Table 5.1 shows that SVM model outperformed with accuracy of 93%. As shown here, the RFC and ensemble models almost performed same on given data set with accuracy of 92%. But precision of RFC model is higher than ensemble model. Therefore, to discover the best-performing model, we have compared the precision of these models in terms of class labels, i.e., ham and spam. Figure 5.1 shows the comparison of deployed models using precision, recall, and F1 score. As shown here, the precision rate of the SVM model is 0.92 and 0.94 for ham and spam, respectively. Additionally, the RFC model achieved same precision rate as SVM model for ham class. For class spam, RFC and ensemble models outperformed than SVM model with precision rate of 0.95. Although, the overall performance of LR and MNB is lower than other models. But for class spam, these models have higher precision rate than others i.e., MNB has 0.98 and LR has 0.96 precison rate. Furthermore, MNB has lowest precision rate for class ham.

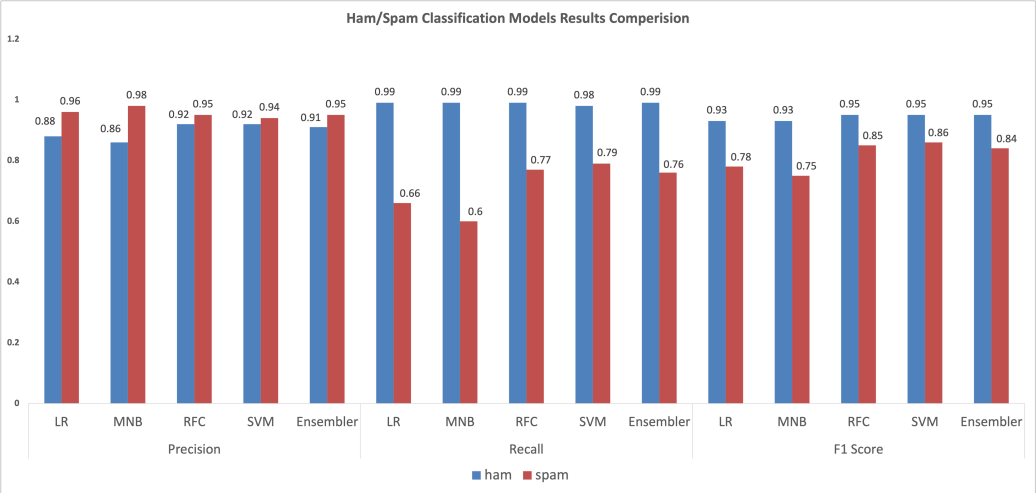


Figure 5.1: Ham/spam Classification Models Comparison

### 5.1.2 Model A: Class wise Performance

As Table 5.1 shows that the precision rate of LR and MNB for spam class is higher than best performing models, i.e., SVM. Therefore, we compared the class-wise performance of deployed models using confusion metrics. Figure 5.2 shows the confusion metric of best-performing model. Out of 779 test instances, the SVM model correctly classified 722 instances into 548 and 174 as ham and spam, respectively. The remaining 57 instances were wrongly classified (i.e., 12 as spam and 45 as ham). However, Figure 5.3 shows the confusion metrics of RFC and ensemble models. As shown here, RFC correctly classified 552 as ham and 168 as spam, i.e., 720 of 799 test data. However, the remaining 8 and 51 instances were incorrectly classified as spam and ham, respectively. And the ensemble model correctly classified 552 of 560 as ham and 166 of 219 as spam instances. As mentioned before, the SVM correctly classified 548 of 560 instances as ham. However, the RFC and ensemble models outperformed SVM by correctly classifying 552 of 560 as ham. Moreover, confusion metrics in Figure 5.4 depict the class-wise performance of LR and MNB classifiers. As shown here, the performance of

these models is lower than other models for class spam. But LR and MNB outperformed for class ham as compared to others. For class ham, the MNB and LR model correctly classified 557 and 554 out of 560 instances, respectively. Only 3 and 6 instances were incorrectly classified as spam by MNB and LR models, respectively.

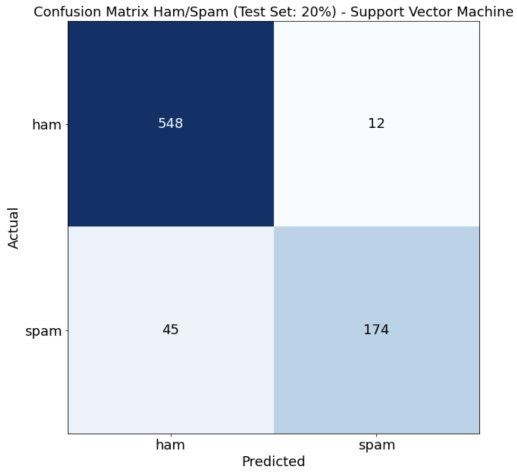


Figure 5.2: Confusion Matrix: SVM Classifier

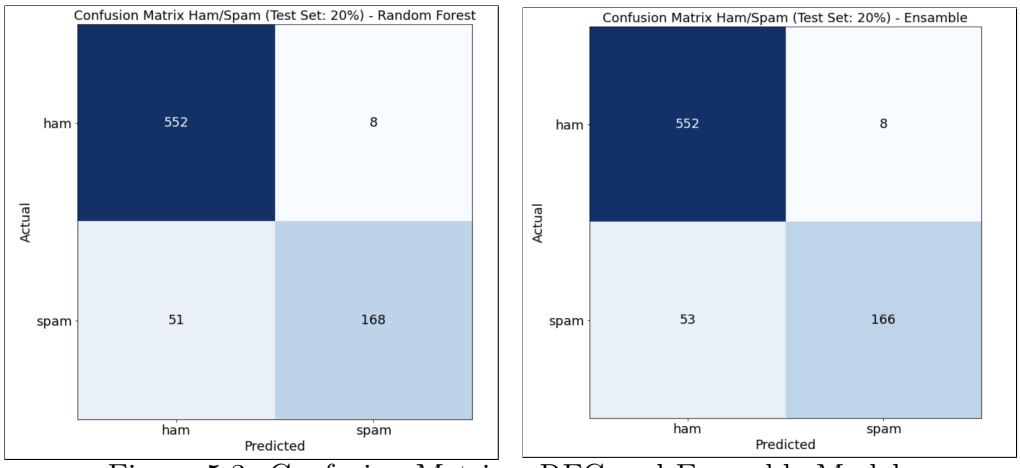


Figure 5.3: Confusion Metrics: RFC and Ensemble Models

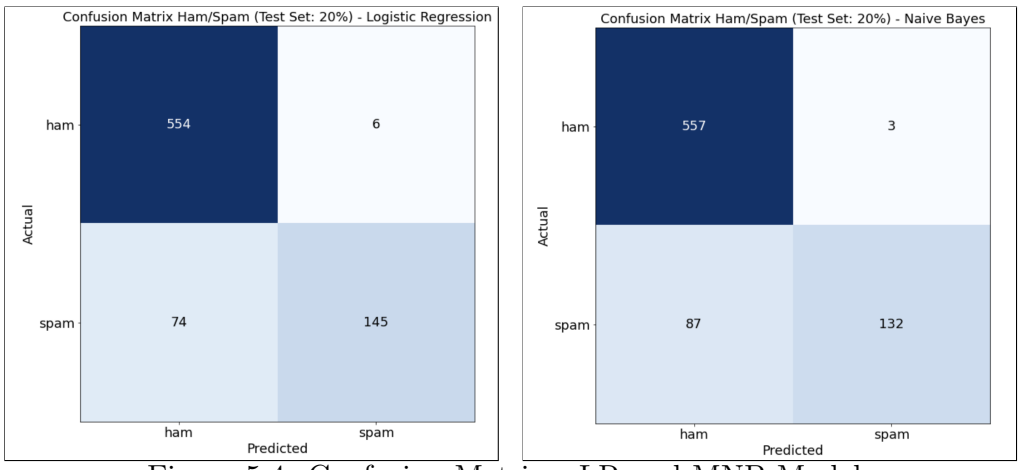


Figure 5.4: Confusion Metrics: LR and MNB Models

## 5.2 OTP/Transaction/Delivery/Personal SMS Classification Models Results

Table 5.2 shows the results obtained by different machine-learning models. As shown here, the RFC model obtained the highest accuracy, i.e., 98% followed by the SVM and ensemble classifier with an accuracy of 97% and 96%, respectively. The MNB performed lowest with an accuracy of 88%.

| | | **Accuracy** | **Precision** | **Recall** | **F-Measure** |
|---|---|---|---|---|---|
| **3-gram** | **LR** | 94 % | 0.95 | 0.94 | 0.94 |
| | **MNB** | 88% | 0.89 | 0.88 | 0.86 |
| | **RFC** | 98% | 0.98 | 0.98 | 0.98 |
| | **SVM** | 97% | 0.97 | 0.97 | 0.97 |
| | **Ensemble** | 96% | 0.97 | 0.96 | 0.96 |

Table 5.2: OTP/Transaction/Delivery/Personal SMS Classification Results

### 5.2.1 OTP/Transaction/Delivery/Personal SMS Classification: Models Comparison

This section compares the performance of different machine-learning models for each class label. Here, we used precision rate to discover the best-performing model. Precision, recall, and F1 score of deployed models for each classes are shown in Figure 5.5. As shown here, all models achieved the same precision for the class OTP i.e., 1. For class delivery, the LR and MNB have precision of 1 followed RFC. However, SVM and ensemble achieved same precision rate in delivery class. Furthermore, for personal SMS class, RFC outperformed with precision of 0.97 followed by SVM and ensemble model with 0.95 and 0.94 of precision rate, respectively. In personal class, the MNB achieved lowest precision rate i.e., 0.85. For class transaction, ensemble model has highest precision rate i.e., 1. The lowest precision rate was achieved by MNB model i.e., 0.91. However, other three models have same precision rate for transaction class, i.e., 0.99.

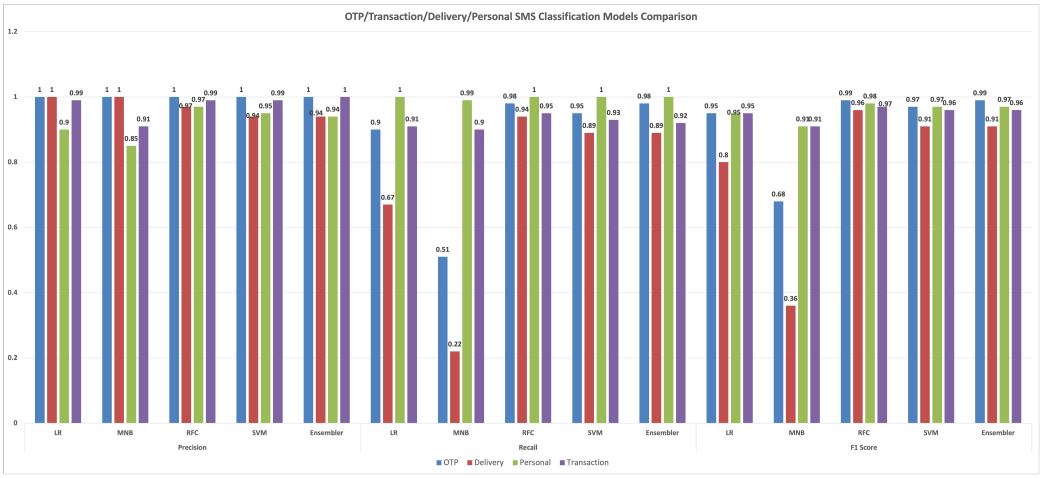


Figure 5.5: OTP/Transaction/Delivery/Personal Classification Models Comparison

### 5.2.2 Model B: Class wise Performance

This section discusses the class-wise performance of model B using confusion metrics. Figure 5.6 shows the confusion metrics of RF model. Out of 563 test instances, the RF model correctly classified 40, 34, 304, and 173 instances as OTP, delivery, personal, and transaction, respectively. RF model wrongly classified 12 instances (i.e., 1 of 41 OTP as transaction, 2 of 36 delivery as personal, 1 and 8 of 182 transactions as delivery and personal). Figure 5.7 shows the confusion metrics of SVM and Ensemble models. As shown here, SVM and Ensemble correctly classified 554 and 543 of 563 instances, respectively. Confusion metrics of LR and MNB shown in Figure 5.8. As shown here, For class delivery, the MNB model gave the least performance of all deployed models, only 8 out of 36 instances correctly classified as a delivery class. The remaining 32 instances wrongly classified as personal. The confusion metrics of models show that these models correctly classified all 304 delivery instances except the MNB model. The MNB model correctly classified 300 of 304 delivery messages, but the remaining 4 instances incorrectly classified as transaction messages.

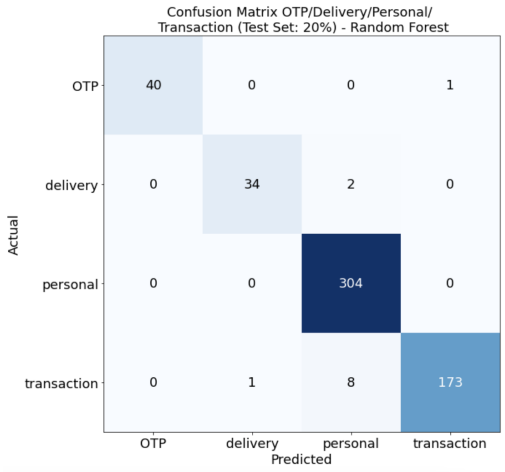


Figure 5.6: Confusion Matrix: RF Classifier

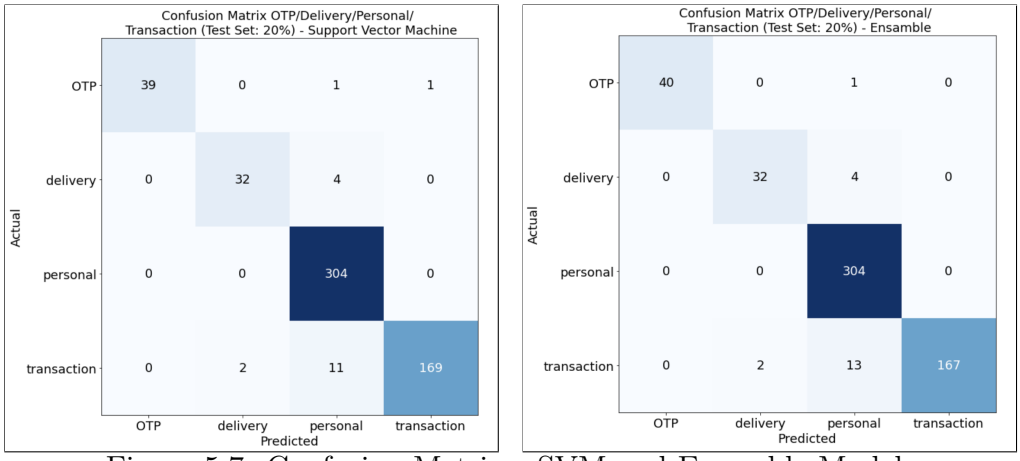


Figure 5.7: Confusion Metrics: SVM and Ensemble Models

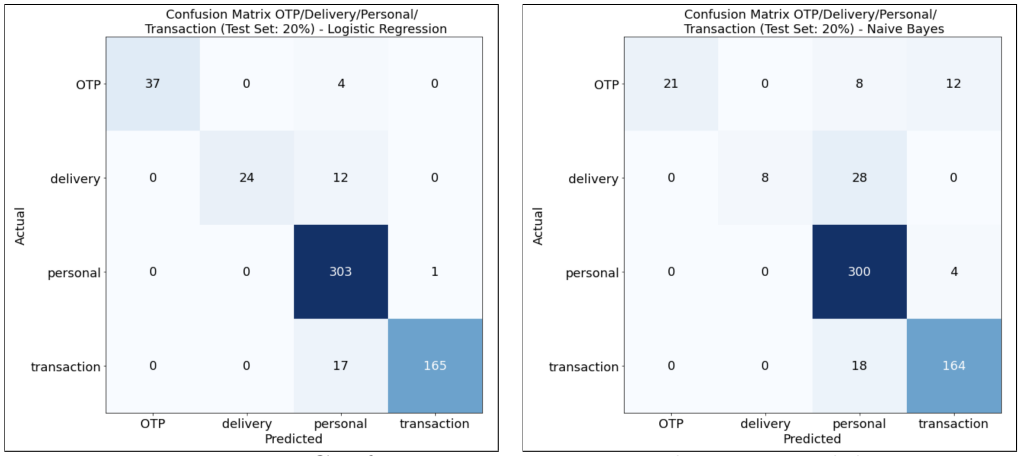


Figure 5.8: Confusion Metrics: LR and MNB Models

# Chapter 6

# Conclusion

This study used automated text classification techniques to detect ham and spam messages. Moreover, this study also compared five ML algorithms to classify messages as spam or ham. Additionally, the model B was also constructed to classify the ham messages into further four classes. The experimental results showed that RFC, SVM, and ensemble algorithms achieved better classification results than LR and NB classifiers. The SVM model outperformed with the accuracy of 93% for ham or spam classification. However for model B, RFC obtained the highest accuracy of 98%. For future enhancement of work following guidelines are suggested and will be the direction of focus:

- Adding more data specific to Indian messages to improve the performance of the models
- Proposed methodology will be extended to incorporate deep neural networks.

# Bibliography


[1] V. Cox, "Exploratory data analysis," in *Translating Statistics to Make Decisions*. Springer, 2017, pp. 47–74.

[2] F. Heimerl, S. Lohmann, S. Lange, and T. Ertl, "Word cloud explorer: Text analytics based on word clouds," in *2014 47th Hawaii international conference on system sciences*. IEEE, 2014, pp. 1833–1842.

[3] S. Shaikh and S. M. Doudpotta, "Aspects based opinion mining for teacher and course evaluation," *Sukkur IBA Journal of Computing and Mathematical Sciences*, vol. 3, no. 1, pp. 34–43, 2019.

[4] J. Ramos *et al.*, "Using tf-idf to determine word relevance in document queries," in *Proceedings of the first instructional conference on machine learning*, vol. 242, no. 1. Citeseer, 2003, pp. 29–48.

[5] R. Dunford, Q. Su, and E. Tamang, "The pareto principle," 2014.

[6] Y.-C. Ho and D. L. Pepyne, "Simple explanation of the no-free-lunch theorem and its implications," *Journal of optimization theory and applications*, vol. 115, no. 3, pp. 549–570, 2002.

[7] L. M. Gladence, M. Karthi, and V. M. Anu, "A statistical comparison of logistic regression and different bayes classification methods for machine

learning," *ARPN Journal of Engineering and Applied Sciences*, vol. 10, no. 14, pp. 5947–5953, 2015.

[8] D. D. Lewis, "Naive (bayes) at forty: The independence assumption in information retrieval," in *European conference on machine learning*. Springer, 1998, pp. 4–15.

[9] T. Joachims, "Text categorization with support vector machines: Learning with many relevant features," in *European conference on machine learning*. Springer, 1998, pp. 137–142.

[10] B. Xu, X. Guo, Y. Ye, and J. Cheng, "An improved random forest classifier for text categorization." *J. Comput.*, vol. 7, no. 12, pp. 2913–2920, 2012.

[11] O. Sagi and L. Rokach, "Ensemble learning: A survey," *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, vol. 8, no. 4, p. e1249, 2018.

[12] N. Seliya, T. M. Khoshgoftaar, and J. Van Hulse, "A study on the relationships of classifier performance metrics," in *2009 21st IEEE international conference on tools with artificial intelligence*. IEEE, 2009, pp. 59–66.

# Appendix A

# Source code

Collab Link