Session 2019

PROJECT REPORT
on

AN APP FOR ORGANIZING TEXT MESSAGES AND NOTIFICATIONS

Submitted By:

Name	**Roll No**	**Branch**	**Year**	**Semester**
Ashi Agarwal	R110216043	B-TECH CSE CCVT	3rd	V1
Ashish Saini	R110216045	B-TECH CSE CCVT	3rd	V1
Avik Jain	R110216048	B-TECH CSE CCVT	3rd	V1
Ayushman Bharadwaj	R110216051	B-TECH CSE CCVT	3rd	V1

Under the guidance of
Mr. Sandeep Pratap Singh
Assistant Professor (Virtualization Department)
UPES, Dehradun

School of Computer Science

UNIVERSITY OF PETROLEUM AND ENERGY STUDIES
Virtualization Department
Dehradun-248007

DECLARATION

I/We hereby certify that the project work is **AN APP FOR ORGANIZING TEXT MESSAGES AND NOTIFICATION** in partial fulfilment of the requirements for the award of the Degree of Bachelor of Technology in Computer Science And Engineering with Specialization in Cloud Computing and Virtualization Technology and submitted to the Department of Virtualization at School of Computer Science, University of Petroleum And Energy Studies, Dehradun, is an authentic record of my/our work carried out during a period from **January, 2019** to **May, 2019** under the supervision of **Mr. Sandeep Pratap Singh, Assistant Professor, Department of Virtualization**

The matter presented in this project has not been submitted by me/ us for the award of any other degree of this or any other University.

Ashi Agarwal

Roll No R110216043

Ashish Kumar Saini

Roll No R110216045

Avik Jain

Roll No R110216048

Ayushman Bhardwaj

Roll No R110216051

This is to certify that the above statement made by the candidate is correct to the best of my/our knowledge.

Dr. Deepshikha Bhargava
Head Department of Virtualization
School of Computer Science.
University of Petroleum And Energy
Studies Dehradun - 248001
(Uttarakhand)

Mr. Sandeep Pratap Singh
Project Guide
Asst. Professor(CIT)
School of Computer Science,
UPES, Dehradun

ACKNOWLEDGEMENT

We wish to express our deep gratitude to our guide **Mr. Sandeep Pratap Singh**, for all advice, encouragement and constant support he has given us throughout our project work. This work would not have been possible without his support and valuable suggestions.

We sincerely thank to our Head of the Department, **Dr. Deepshikha Bhargava**, for her great support in doing our project **An App for Organizing Text Messages and Notifications** at SoCS.

We would like to thank all our **friends** for their help and constructive criticism during our project work. Finally, we have no words to express our sincere gratitude to our **parents** who have shown us this world and for every support they have given us.

Name	**Ashi Agarwal**	**Ashish Kumar Saini**	**Avik Jain**	**Ayushman Bharadwaj**
Roll No.	**R110216043**	**R110216045**	**R110216048**	**R110216051**

CERTIFICATE OF ORIGINALITY

This is to certify, that the research project submitted by me is an outcome of my independent and original work. We have duly acknowledged all the sources from which the ideas and extracts have been taken. The project is free from any plagiarism and has not been submitted elsewhere for publication.

Name of Author (s): Ashish Kumar Saini, Ayushman Bharadwaj, Avik Jain, Ashi Agarwal

Designation: B.Tech in Computer Science Engineering with specialization in Cloud Computing and Virtualization Technology.

Affiliated Institutions: UPES, DEHRADUN

Title of Project: An App for Organizing Text Messages and Notifications

Email: 500052000@stu.upes.ac.in

500052388@stu.upes.ac.in

500053031@stu.upes.ac.in

500052703@stu.upes.ac.in

Contact No: +91 8957212220

ABSTRACT

The quantity of short message service (SMS) and spam is growing rapidly. Solutions which have been proposed, that for the most part, use classical text classification techniques. But that require another computer system to create a classifier using a large quantity of SMS and notifications information in advance to set up the classifier into the cellular phones for filtering out the incoming text messages. These type of solution decreases the independence of a mobile user because the user has to store the incoming message or notification into a computer system to train the classifier. Firstly, storing messages in a computer system either locally or remote messages which also contain private data, leads to the many privacy problem. And secondly, it furthermore increases hardware cost and communication cost among mobile phones and a computer. Therefore, we propose a completely standalone filtering system that does not require a computer system to function. The training process is done on a computer machine and the model is loaded in the mobile device. We are analyzing machine learning algorithms such as Naive Bayes, Support Vector Machines and Neural Networks etc. to find out the best model to perform spam filtering.

Keywords: **Naive Bayes**, **Support Vector Machine**, **Message, Spam**.

TABLE OF CONTENTS

List of figures

1.Introduction

Also due to the increase in the number of smartphones, the number of short communication channel is also increasing. This increase in the number of messages being exchanged on day to day basis result in malicious attacks, which include spam attacks, that can be a direct or indirect threat to the privacy and security of the user. Unlike, email spam filtering, SMS spam filtering is a relatively new task but it inherits all the traditional problem of the email filtering systems, including some additional problems that are text message specific.

New security problems arise every day, maybe due to a faulty update from the manufacturer or an untrusted application downloaded by a user, but one issue remain consistent throughout, which spam messages. At the least, spam can cause interrupts in our busy days, forcing us to waste time opening and deleting messages. In a more serious and very common case, spam could release a virus into our systems which can prove to be fatal for both software as well as hardware of the system.

According to a recent survey, out of all the messages we receive in our day to day life, more than 70 percent messages are either spam or not relevant to the particular user. Preventing Spammers from sending spam messages may not be possible just yet, for the time being the best we can do is have anti-spam applications in your systems that may work as a filter, filtering every message before it bothers us. Anti-spam applications traditionally use more than one filtering methods to identify spam and prevent it from reaching user's inbox.

Different spam-filtering systems uses different techniques:

- Some run checks on every message to determine the likelihood of it being a spam.
- Other just simply block every message from any known spammers.
- Other only allow messages from a certain sender.
- Some spam filtering methodologies works on the user response, i.e. they identify the spammer or spam messages based on users response to a particular message or a particular sender [1]

Detail description of all the different filters are given below:

List-Based Filters

List-Based Filters works by organizing the list of senders in different types of list and then allow or disallow messages from those senders.

Different types of list used in List- based Filters are as follows:

- **Blacklist**

A very popular filtering technique that attempts to stop messages by blocking messages according to the list of senders in the blacklist. Upon the arrival of any message, the spam filter checks if its senders name or number in blacklist. To determine whether the message should be blocked or not.

- **Real-Time Blackhole List**

This method is same as the blacklist method but it requires much less maintenance. This is because in Real-Time Blackhole list the list of senders that needs to be blacklisted are made by a 3rd party application which provides that list at a minimal cost and the user system application can refer to that list in deciding whether the message from a particular sender should be blocked or not.

- **Whitelist**

A whitelist technique works almost opposite to Blacklist. In whitelist, a list of trusted senders in prepared and messages from only those senders will be accepted. This method is very useful in system requiring high security.

Content-Based Filters

Content-based filters calculates words and phrases found in each message to judge whether a particular message should be allowed or not.

- **Word-Based Filters**

A word-based spam filter is the simplest type of content-based filter. In word-based Filters a bag of spam words is prepared and every message is searched for these words and if any of those words are fount the message, the message is blocked.

- **Heuristic Filters**

Heuristic filters take things a step forward then the simple word-based filter. Heuristic filters take multiple terms found in a message into consideration while deciding whether to block a message or not.

- **Bayesian Filters**

Bayesian filters are considered to the most advanced form, which apply the laws of probability to determine which messages will be allowed and which will not. In order for a Bayesian filter to effectively block spam, the end user must initially "train" train the system by flagging spam messages and the system learns as more and more messages are flagged as spam or not. [1]

Classification Methods

Some of the popular classification techniques used in spam filtering algorithm.

- **Naive Bayes Algorithm (NB)**

The Naive Bayes technique is used for creating probabilistic model for classification of messages. Naive Bayes algorithm makes an assumption that all the features in a modal are statistically independent of one another.

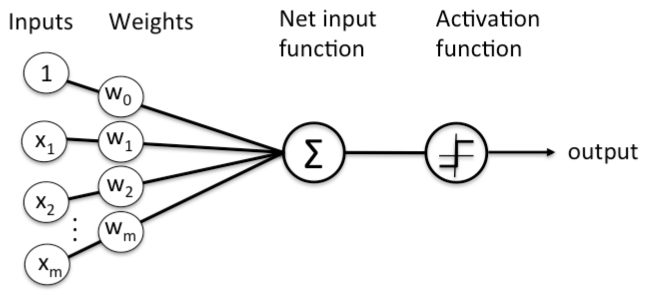
Pros of Naive Bayes - Only a small quantity of training data is used to determine or estimate all the parameters necessary to perform the intended classification.[2].

- **Support Vector Machine (SVM)**

Supervised learning models, which study data and understand patterns, Classification is the major use of SVM. From training examples, with every example marked as belonging to its individual category, an SVM training algorithm builds a model that will assign new examples into its suitable category, it is also known as non-probabilistic binary linear classifier. An SVM model is a representation of the examples as points in space, mapped so that the examples of the separate categories are divided by a clear gap between them. New examples are then mapped into that same space as the training data and are predicted based on which side of the gap they fall on and depending on that they are assigned to a category.[3].

- **Neural Networks**

Neural Networks are a class of general function approximations, and be applied to every ML problem which is based on studying a complex mapping from the input or initial space to the outer space. Every neural network is composed of many layers. Each layer is made of nodes. A node is responsible for all the calculation. Summation of input data is taken with the weights associated with each input which determines whether that particular input should be dampened or amplified. All the input*weight are summed which is then passed through an activation function, which tells us that whether a particular neuron should be active or not.



[4]

Fig-1 Neural Network

2.Literature Review

- Neelam Choudhary and Ankit Kumar Jain et al. [5] explained a machine learning approach to detect and classify spam messages by using an algorithm known as Random Forest Algorithm. They proposed a general architecture a system and what step will it involve. The steps mainly included Dataset collection, Pre-processing phase, Training classifier, Testing Classifier, classification result and performance evaluation. A detailed description of what a spam message contains and what are the most common terms in a spam message.

- Tiago A. Almeida et al. [6] provided the shortcomings of a content-based filter for mobile phone messages, the reason being that the mobile messages are very small. A large collection of non-encoded spam messages. A conclusion was presented that SVM performs better than any other classifier, hence, it could be used for further comparison as a baseline. They advocated in favor of SVM over any other model of message classification.

- M. Nivaashini et al. [7] deals with the problem of high false positive rate associated with the current models and present a solution by deep learning approach. They also proposed framework to classify the SMS spam by using of Deep Neural Network (DNN) classifier using SMS Spams database from UCI Machine Learning repository. The results of the comparison between the proposed deep learning techniques with the existing machine learning algorithms in terms of measures like Accuracy, True Positives, False Positives was presented.

- Thusitha Dayaratne et al. [8] proposed a hybrid Spam detection solution, which can be used by both a smartphone as well as a feature phone. In which a feature phone user can use a general solution for filtering out his SMS and Smartphones can configure and filter SMSs based on their own preferences rather than sticking in to a general filter. They also evaluated the accuracy of neural network using spam huge dataset along with some randomly used personal SMSs.

3.PROBLEM STATEMENT

In today’s era where any an average individual who uses a mobile phone, receives hundreds of mobile SMS every day. Some of them are useful to him/her, most of them are not. Some messages are spam containing a possible virus and could prove to be fatal for any device. Every message we receive on our phone gives us a notification or an alert and this could prove to very irritating for the user if the notification was about something not important or relevant to him. So there needs to a system that does all the classification and only displays the messages that the relevant to the user based on his/her feedback and also does spam filtering for him/her.

Most of the systems which exist requires additional computing power to determine whether a message is spam or not, so there is a need for a system which does this classification in real time after getting trained on a dataset. The System should be able to perform real-time classification and filtering of text messages.

4.OBJECTIVES

Here we are providing the user an android application that will classify the Text message, notification and even the messages from other messaging platforms into various categories like Primary, updates, promotional, OTP, banking transactions, etc. Our system will filter spam notifications from apps on the device also.

Sub-Objective:

- To filter out spam Text Messages and Notifications.
- We use various Machine Learning models and will evaluate which is best for filtering spams.

  Categorizing Notifications and Messages on the basis of the content.

5.Methodology

To Build a spam filter we will be needing lots of training data of SMS messages, due to unavailability of a good and large data sets we will be constructing our own data set.

5.1Building a Data Set

1. Collection of messages in form of a csv file.
2. Cleaning the data collected.

5.2Training the Machine Learning Model

1. Dataset is split into Test and Training dataset with Cross-Validation.
2. All the different selected models are trained using the dataset
3. Models are tested using the Test sets and Performance is evaluated for each algorithm.
4. On the basis of various evaluation factors an algorithm is decided that will be used in the Android application.

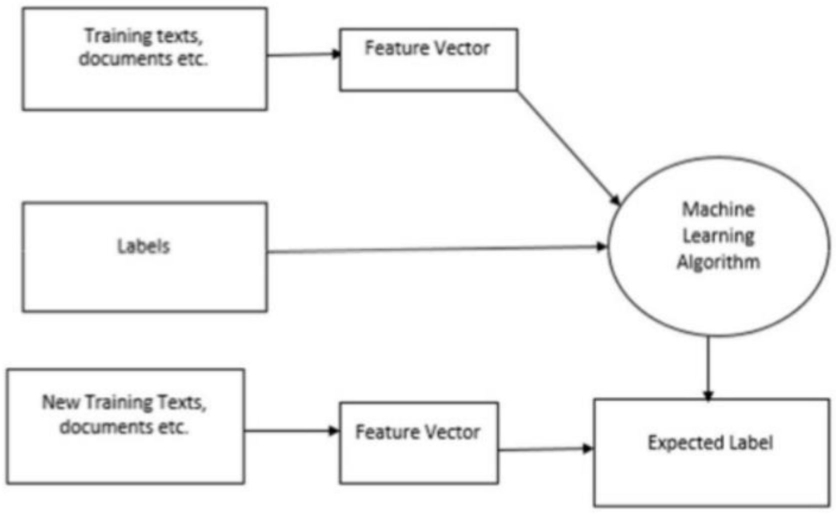
5.3Building the Android Application

1. UI/UX will be Build.
2. API of the Model will be integrated in the application
3. Testing will be performed.

6.DESIGN

6.1Data Flow Diagrams

Level 0


Training texts, documents etc.
Feature Vector
Labels
Machine Learning Algorithm
New Training Texts, documents etc.
Feature Vector
Expected Label

Level 1

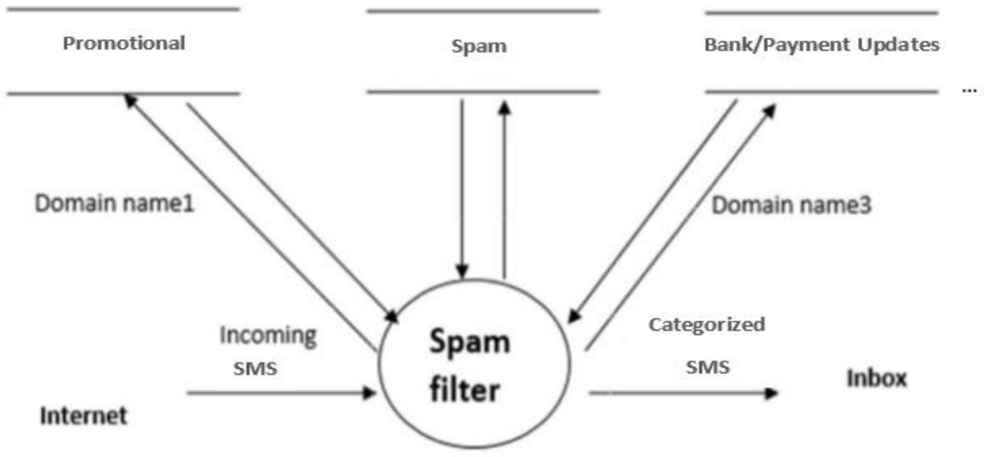


Fig-2 Data Flow Diagram

6.2) Use Case Diagram

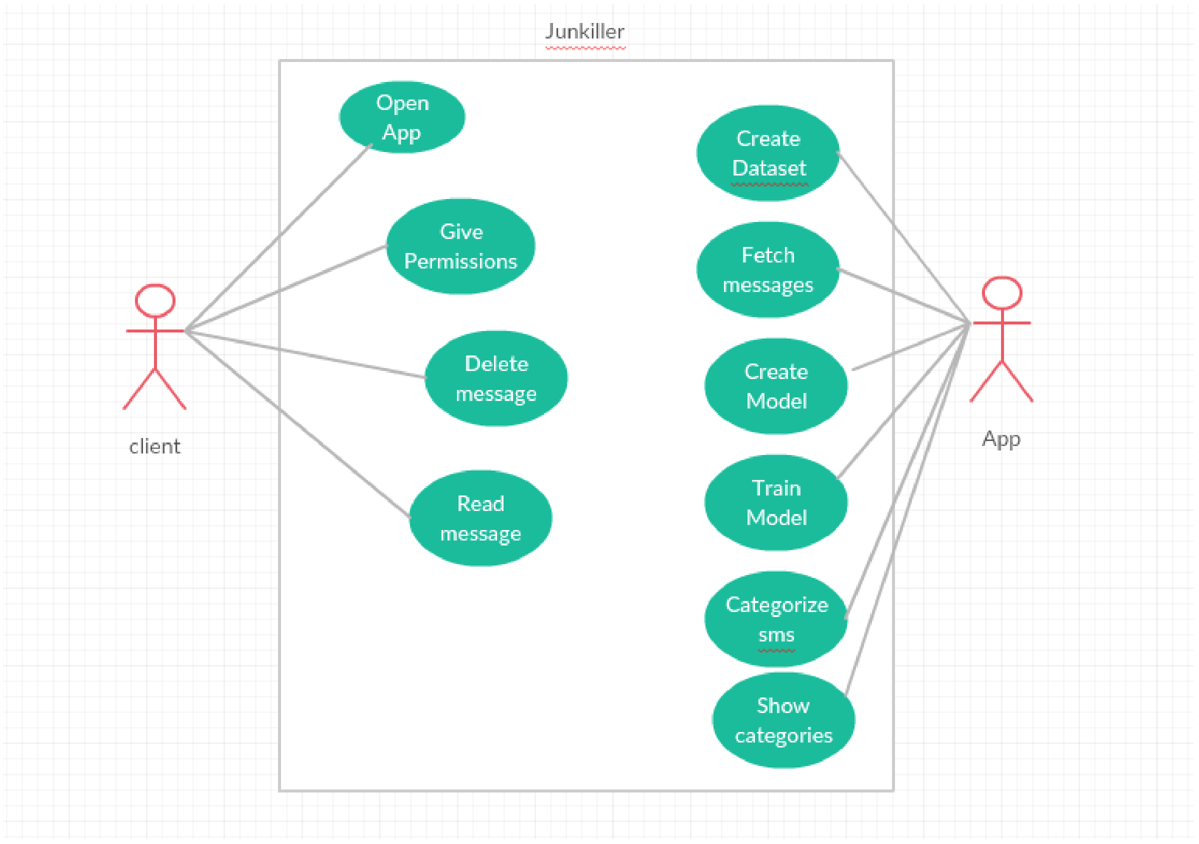


Fig-3 Use case diagram

6.3) Activity Diagram

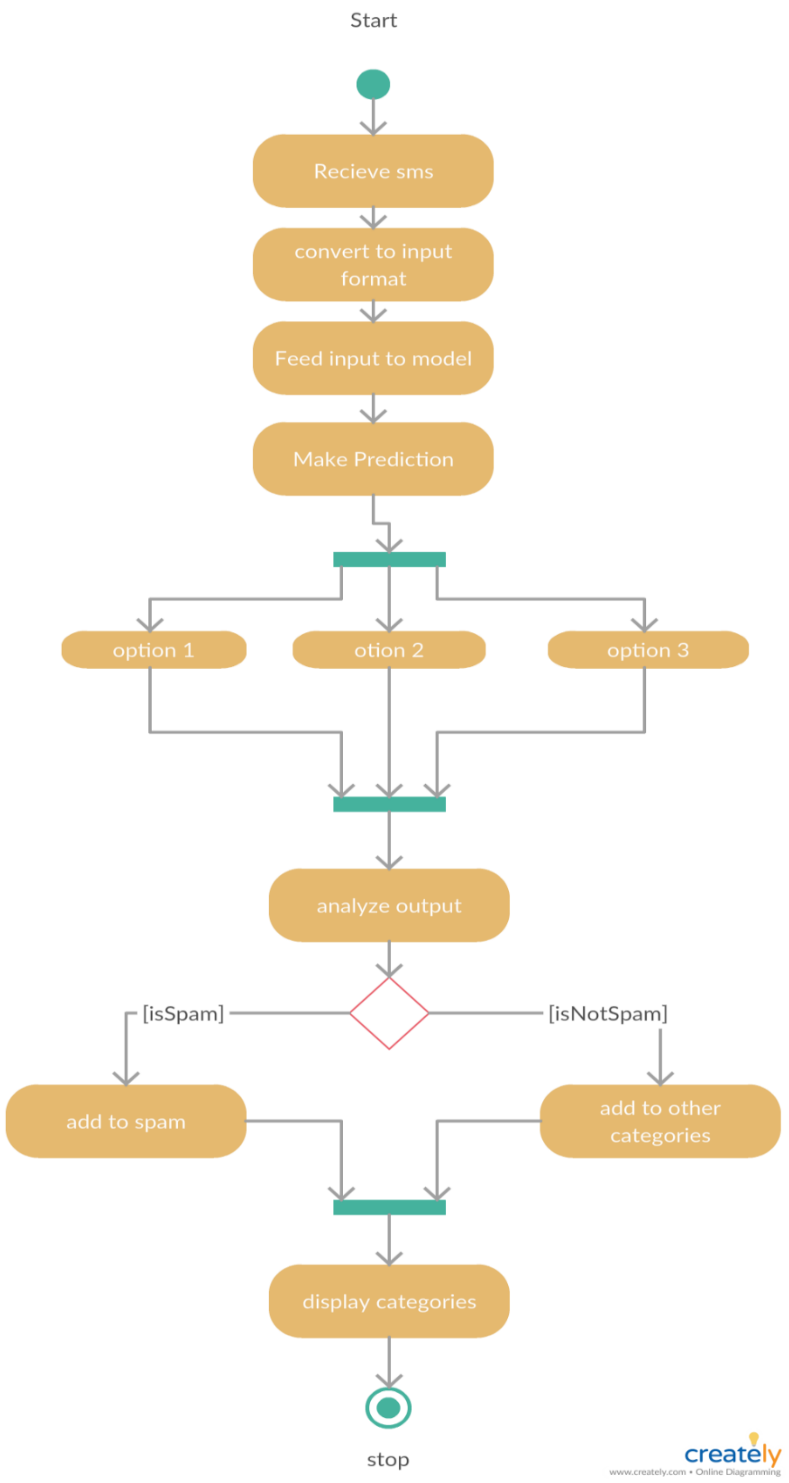
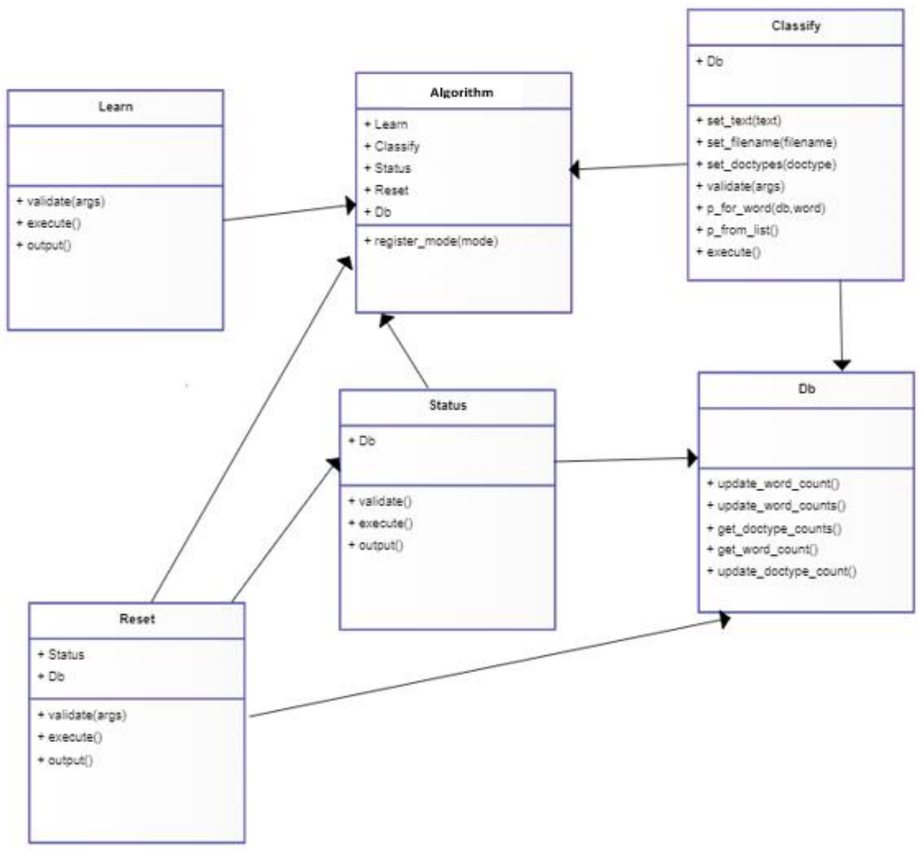


Fig-4) Activity Diagram

6.4) Class Diagram



- **Class Diagram (Android Application)**

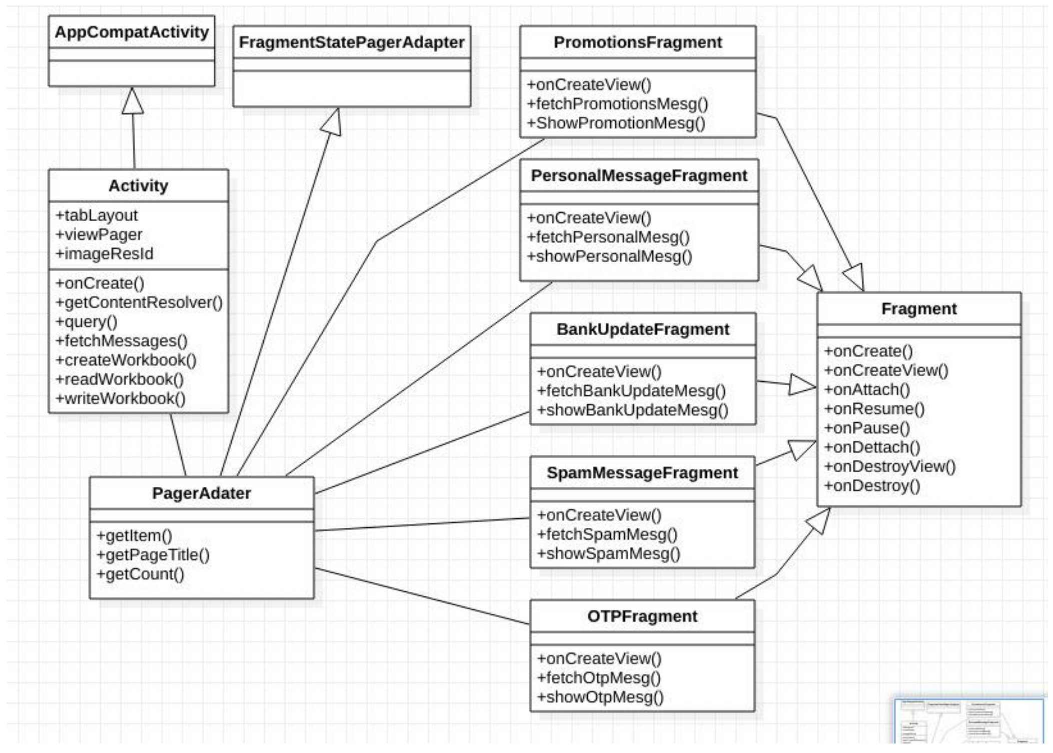
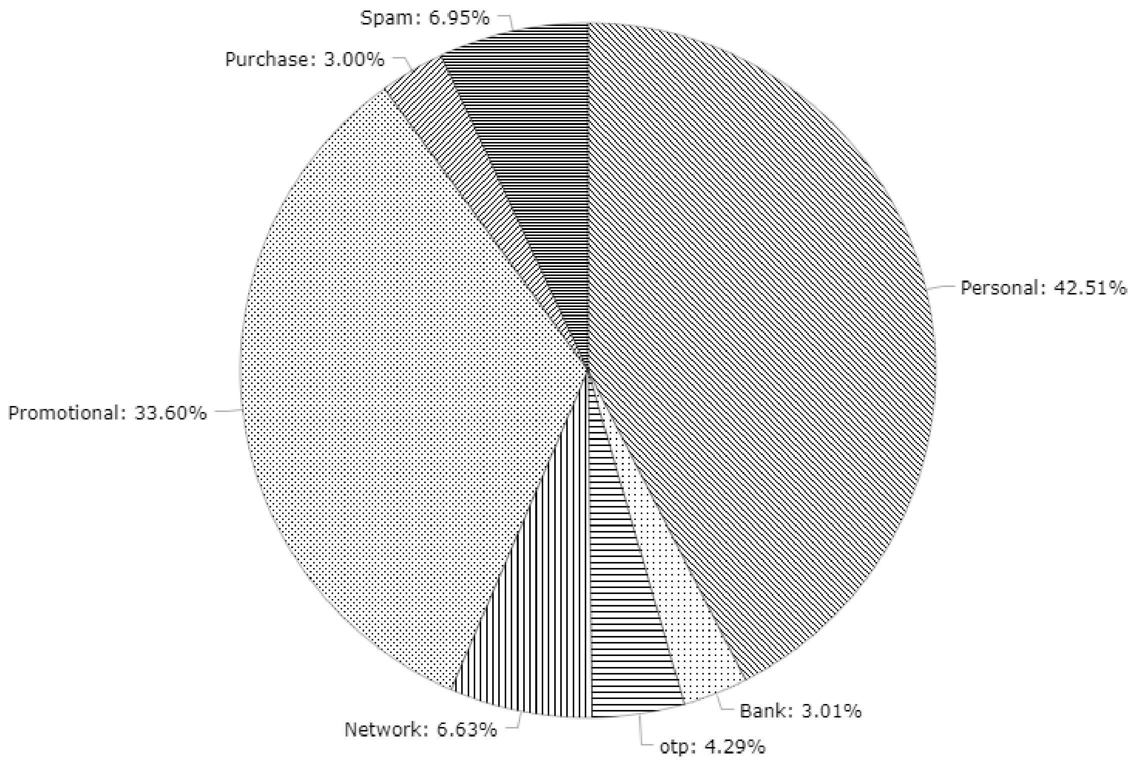


Fig-5 Class Diagram

6.5) Dataset Visualization



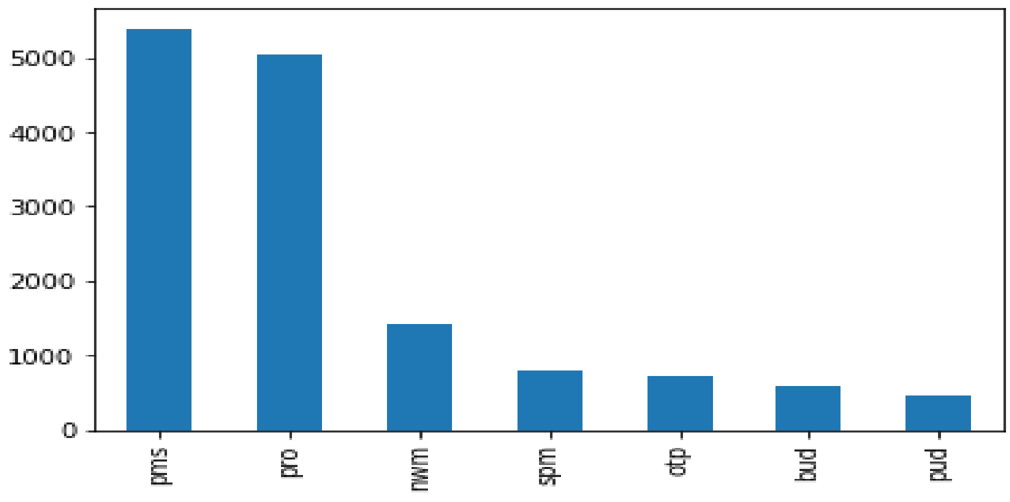
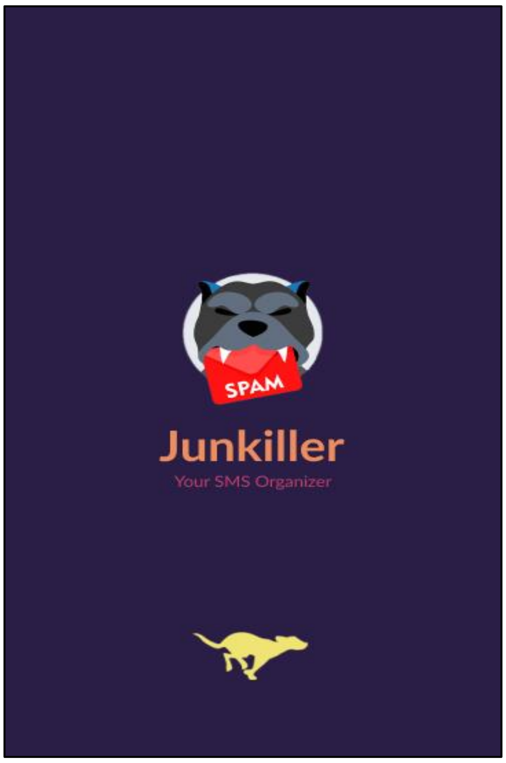
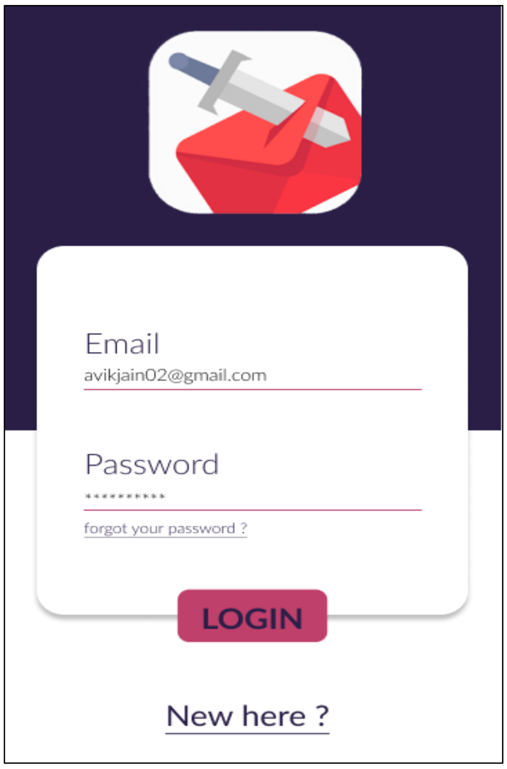
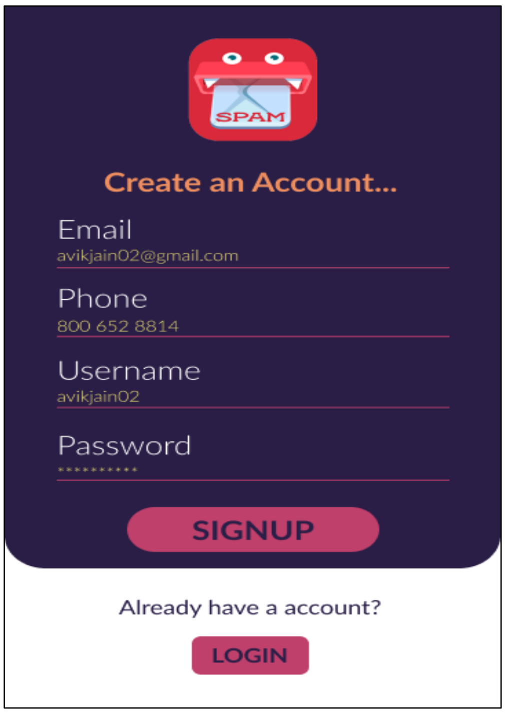


Fig-6 Data Visualization

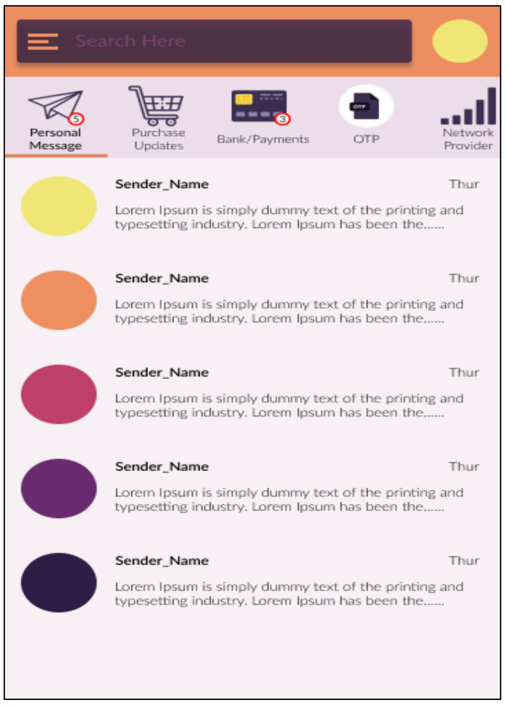
6.6) Android application UI Design




Email
avikjain02@gmail.com
Password
****



Email
avikjain02@gmail.com
Phone
800 652 8814
Username
avikjain02
Password
****



Sender_Name Thur
Lorem Ipsum is simply dummy text of the printing and typesetting industry. Lorem Ipsum has been the......

Sender_Name Thur
Lorem Ipsum is simply dummy text of the printing and typesetting industry. Lorem Ipsum has been the......

Sender_Name Thur
Lorem Ipsum is simply dummy text of the printing and typesetting industry. Lorem Ipsum has been the......

Sender_Name Thur
Lorem Ipsum is simply dummy text of the printing and typesetting industry. Lorem Ipsum has been the......

Sender_Name Thur
Lorem Ipsum is simply dummy text of the printing and typesetting industry. Lorem Ipsum has been the......

Fig-7 Android UI

7.Design Methodology

7.1) Naïve Bayes Algorithm:

Naïve Bayes is based on Bayes theorem of mathematical probability.

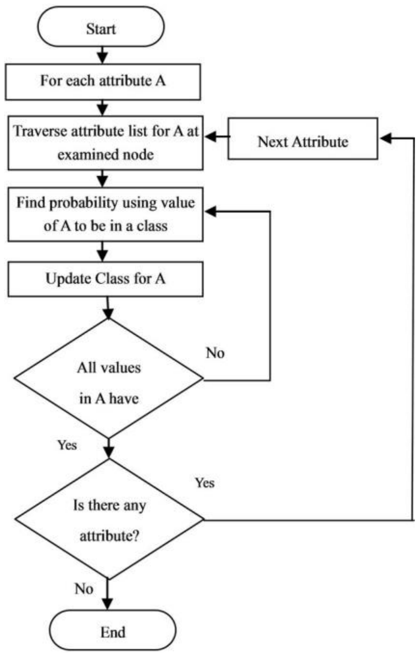
For classifying a dataset using Naïve Bayes the following steps Are performed:

- The dataset is divided into two parts matrix i.e a feature matrix and response matrix.
  feature matrix: set of dependent variables, variables used for predicting the outcome.
  Response matrix: Outcome.
- Bayes' Theorem finds the probability of an event occurring given the probability of another event that has already occurred. Mathematical equation:

$$P(A|B) = \frac{P(B|A)P(A)}{P(B)}$$

- Naïve Assumption:

  A and B are independent events:



$P(A,B) = P(A)P(B)$

Fig-8(Naïve Bayes Flowchart)

7.2) Support vector Machine

Support Vector Machines is a machine learning algorithm. Its main goal is finding a hyperplane of an N-dimensional space (N is the no. of features) and it classifies the points distinctly. Many possible hyperplanes can be chosen to separate the classes that data points have (may be two). Our goal is finding maximum margin plane that is largest distance of data points of the classes (may be two) and by doing this we can add some reinforcement as in future data points could be classified with more accuracy. Hyperplanes are the decision boundaries which can help in classifying the data points.

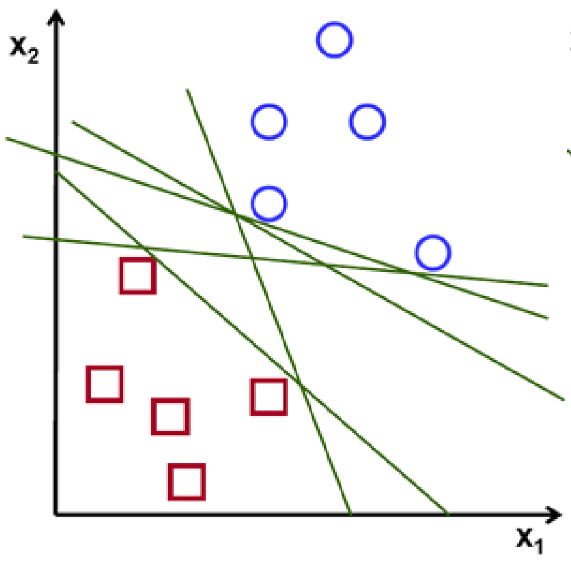


Fig 9- Different hyperplanes.

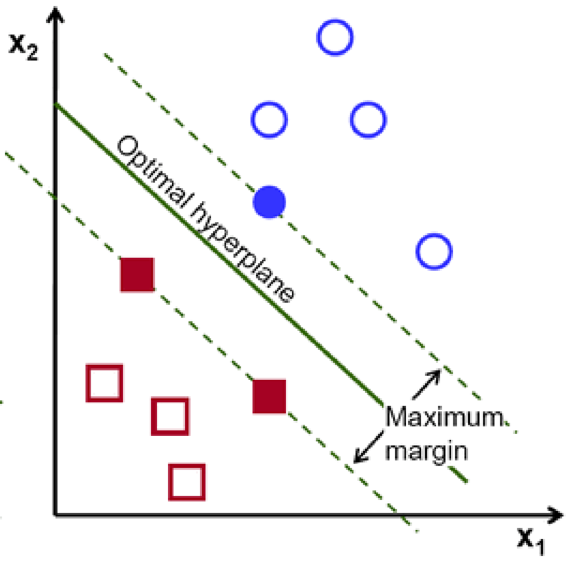


Fig 10- Maximized margin.

The data points that are closer to the hyperplane are called support vector machines and these can influence the position and orientation of the hyperplane. By using these, we can max the margin of the classifier.

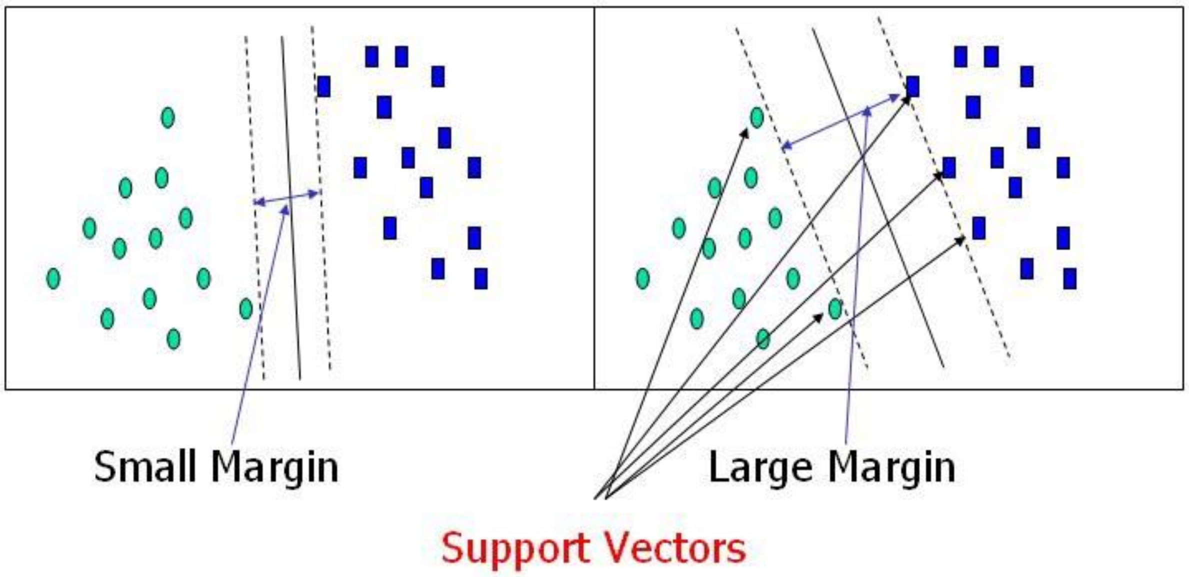


Fig 11- Support vectors category.

7.3. Neural Networks:

Neural Networks are some group of algorithms loosely modeled to copy human brain which can be used for pattern recognition. They label or cluster sensory data. Recognized patterns can be numerical, contained in vectors, into which every real-world data (audio, image, text) must be converted.

NN helps in clustering and classifying. They help to bunch unlabeled information as per likeliness among the model sources of info, and they classify information when they have a marked dataset to train on.

NN are generally made up of several layers. Layers are nothing but set of nodes grouped together. A node does some computation and gives result just like human neuron. A node work is to increase or decrease the input by assigning weights to them. Then the sum of weight-input products is passed to a layer's activation function, to decide if and to what degree that signal should advance further through the network to influence a definitive result, say, a demonstration of classification. On the off chance that the signals goes through, the neuron has been "activated."

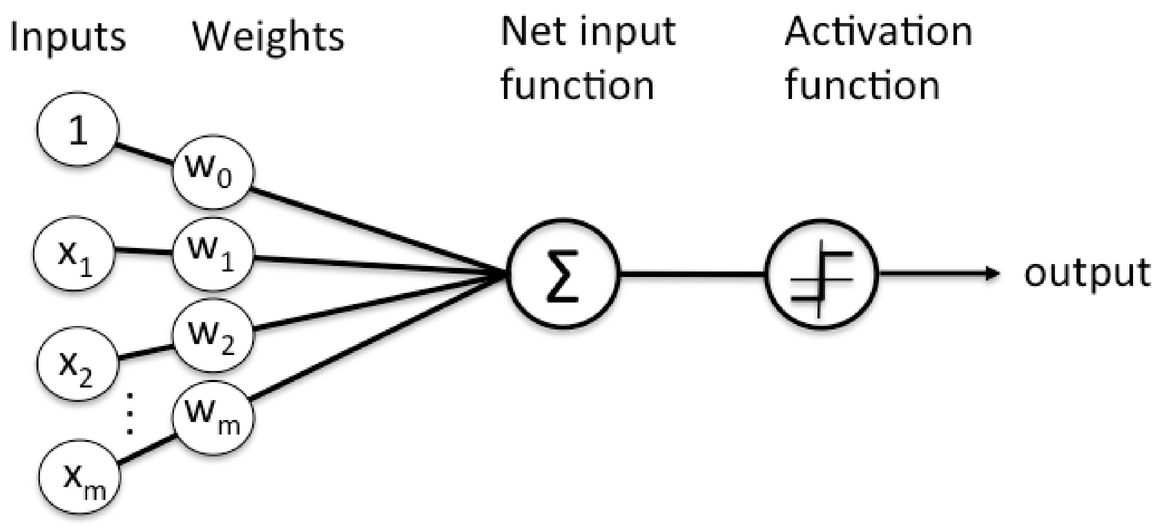


Fig 13- Perceptron general diagram.

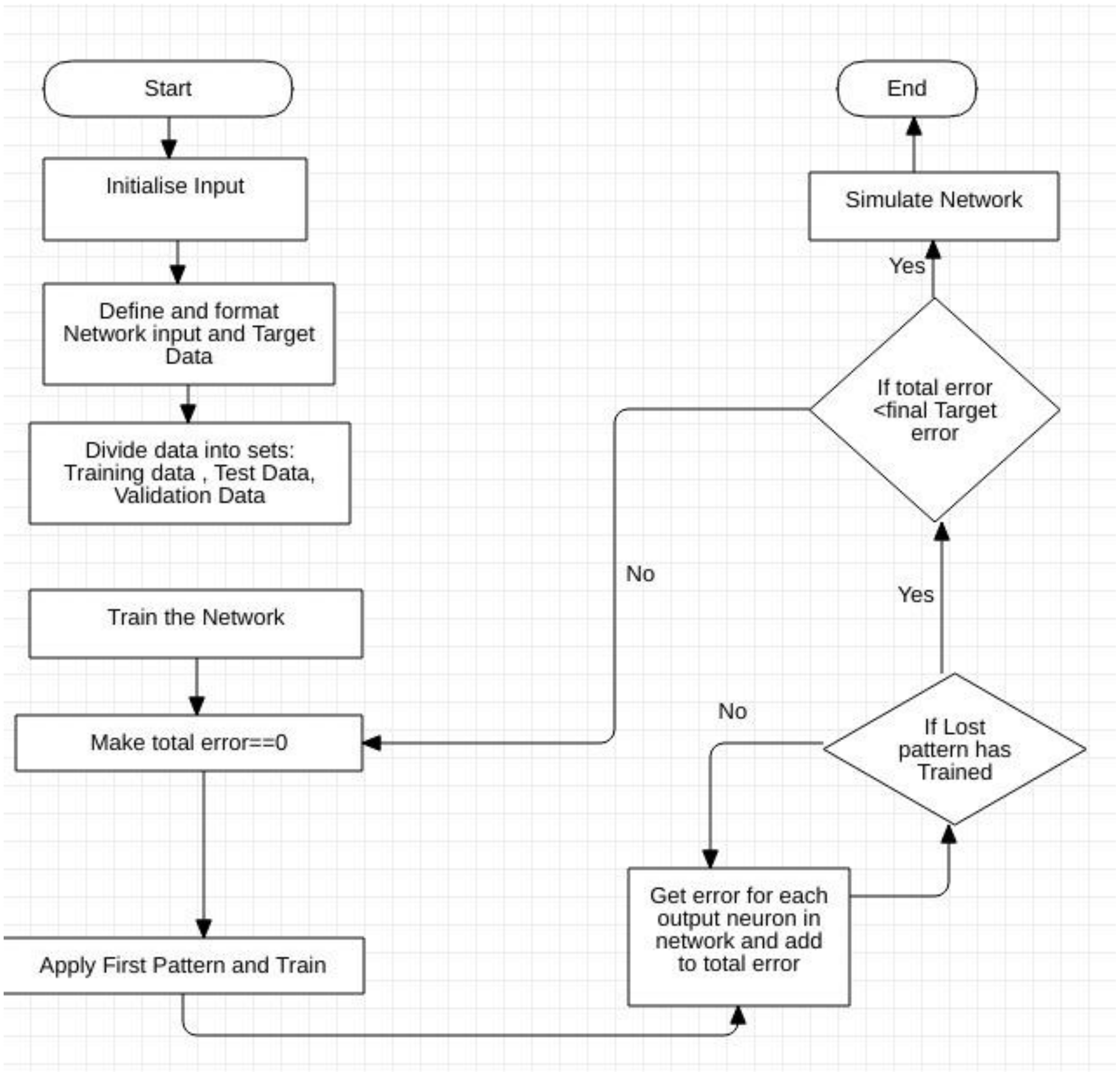

Start
Initialise Input
Define and format Network input and Target Data
Divide data into sets: Training data , Test Data, Validation Data
Train the Network
Make total error==0
Apply First Pattern and Train
Get error for each output neuron in network and add to total error
If Lost pattern has Trained
No
Yes
If total error <final Target error
No
Yes
Simulate Network
End

Fig-14(Neural Network Flowchart)

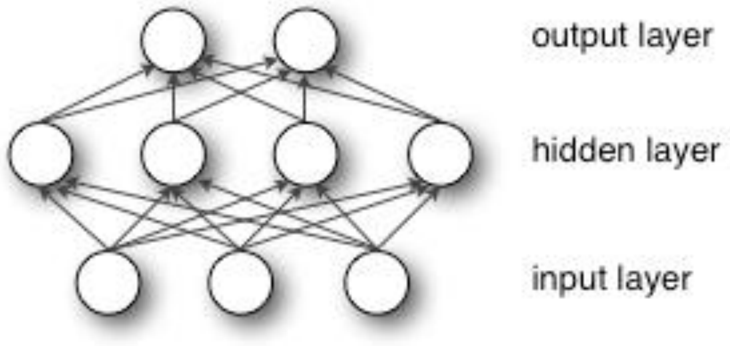


Fig 15- Each Layer's output is used as an input for upcoming layer.

7.4.k-nearest neighbors algorithm (**k-NN**) is a supervised machine learning algorithm. It is used for pattern recognition for classification and regression. K closest examples of training in input in the feature space.

- k-NN classification consists of class membership output.
- k-NN regression consists of property value of object output.

The training period of the calculation comprises just of putting away the feature vectors and class labels of the training samples. In the classification stage, k is a client characterized steady, and an unlabeled vector (a question or test point) is classified by allocating the mark which is most continuous among the k training samples nearest to that inquiry point. A normally utilized separation metric for constant factors is Euclidean separation. A downside of the essential "majority voting" classification happens when the class circulation is skewed. That is, instances of a progressively visit class will in general rule the forecast of the new precedent, since they will in general be regular among the k nearest neighbors because of their huge number. One approach to beat this issue is to weight the classification, considering the separation from the test point to every one of its k nearest neighbors.

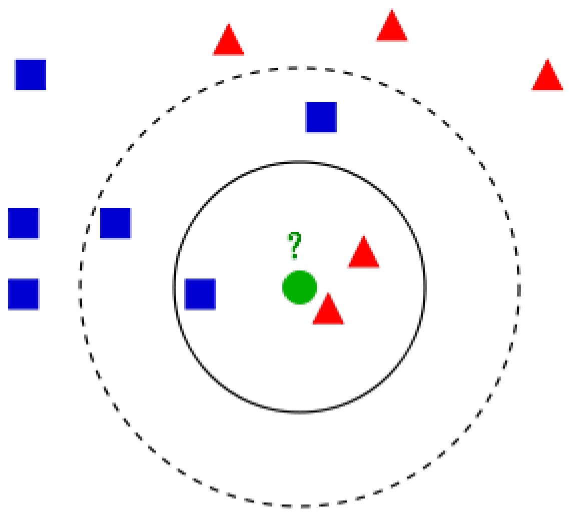


Fig- 15 Example of k-NN classification. The test sample (green dot) should be classified either to the first class of blue squares or to the second class of red triangles.

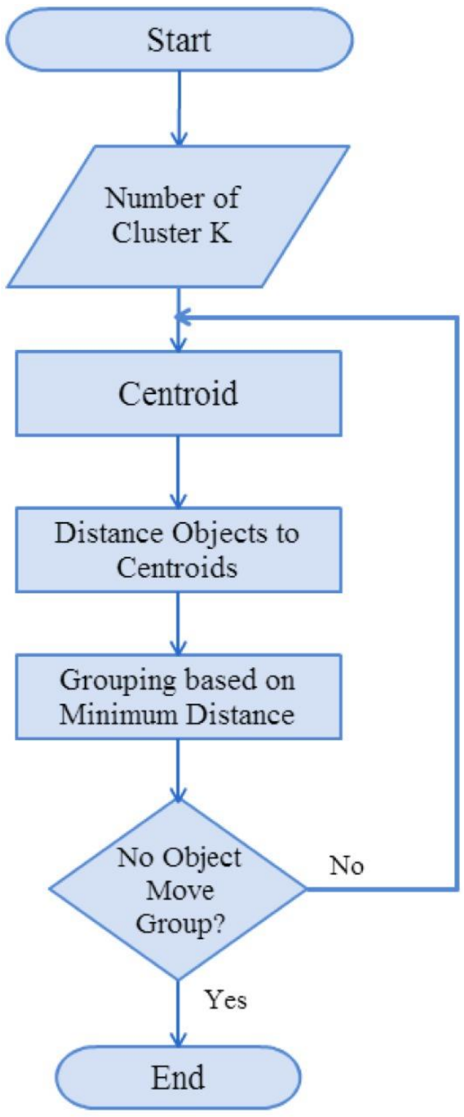

Start
Number of Cluster K
Centroid
Distance Objects to Centroids
Grouping based on Minimum Distance
No Object Move Group?
No
Yes
End

Fig-17 KNN flowchart

8.System Requirements (Software/Hardware)

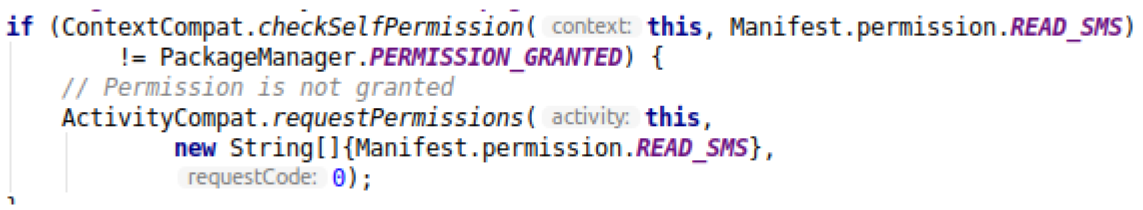
- Hardware Interface:
  - 64 bits processor architecture supported by windows.
  - Minimum RAM requirement for proper functioning is 8 GB.
  - Required input as well as output devices.
  - Required sufficient Graphic card for model training.
- Software Interface:
  - This system is developed in Java and Python programming language.
  - Anaconda Distribution
  - Adobe XD
  - Zeplin
  - Android Studio

9.Implementation

In Andriod:

For fetching SMS from a phone:

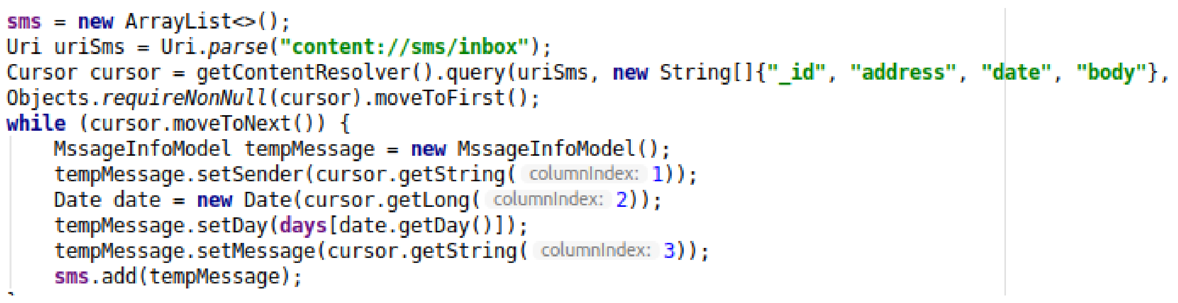
- We need to ask user his/her permission for accessing(reading) messages, this is done by using method, requestPermission(READ_SMS) .



```java
if (ContextCompat.checkSelfPermission( context: this, Manifest.permission.READ_SMS)
        != PackageManager.PERMISSION_GRANTED) {
    // Permission is not granted
    ActivityCompat.requestPermissions( activity: this,
            new String[]{Manifest.permission.READ_SMS},
            requestCode: 0);
```

- If permission is granted in the above step, ContentProvider class is used for fetching SMS by using “content://sms/inbox”

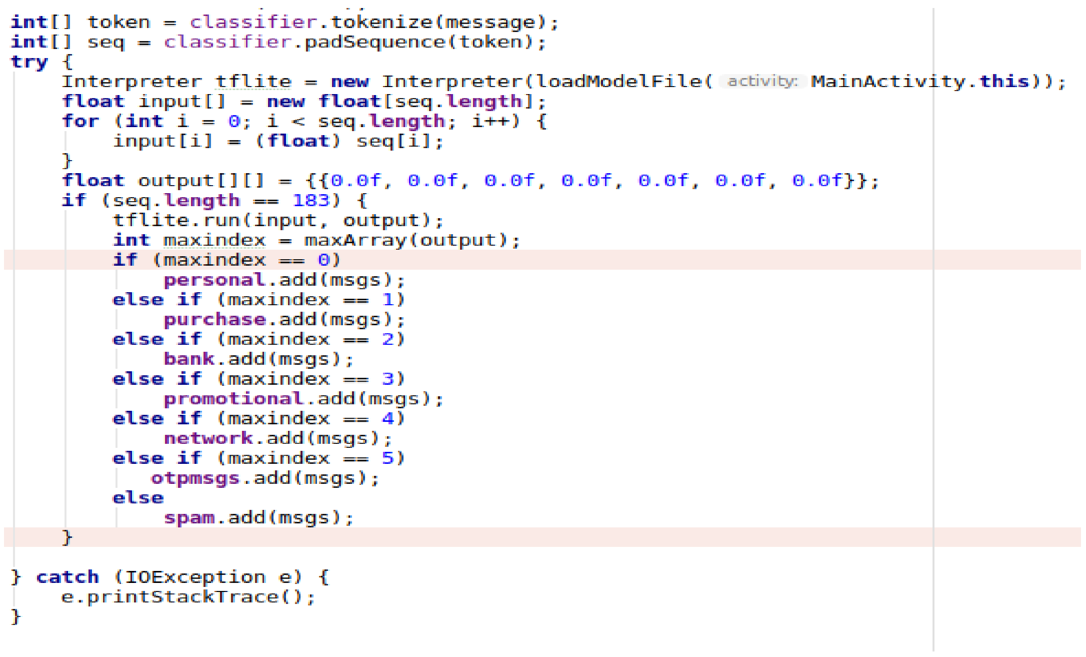
- After database is quarried, using ContentResolver we get the reference of cursor to SMS database.
- We create an SMS model, which contain 3 attributes, Sender, date & message.



```java
sms = new ArrayList<>();
Uri uriSms = Uri.parse("content://sms/inbox");
Cursor cursor = getContentResolver().query(uriSms, new String[]{"_id", "address", "date", "body"},
Objects.requireNonNull(cursor).moveToFirst();
while (cursor.moveToNext()) {
    MssageInfoModel tempMessage = new MssageInfoModel();
    tempMessage.setSender(cursor.getString( columnIndex: 1));
    Date date = new Date(cursor.getLong( columnIndex: 2));
    tempMessage.setDay(days[date.getDay()]);
    tempMessage.setMessage(cursor.getString( columnIndex: 3));
    sms.add(tempMessage);
```

Classification of SMS:

- Initialise and load dictionary to tokenize our messages in the form of an array using the word_dict.json file.
- Set maximum length of messages according to the dataset.
- Tokenize message and loop over each sms from phone
- Load Tflite model using Interpreter with our model.tflite file.
- Run tflite model on each SMS and predict probability of each category
- Find the category which has maximum probability and add to the list of that category



```java
int[] token = classifier.tokenize(message);
int[] seq = classifier.padSequence(token);
try {
    Interpreter tflite = new Interpreter(loadModelFile( activity: MainActivity.this));
    float input[] = new float[seq.length];
    for (int i = 0; i < seq.length; i++) {
        input[i] = (float) seq[i];
    }
    float output[][] = {{0.0f, 0.0f, 0.0f, 0.0f, 0.0f, 0.0f, 0.0f}};
    if (seq.length == 183) {
        tflite.run(input, output);
        int maxindex = maxArray(output);
        if (maxindex == 0)
            personal.add(msgs);
        else if (maxindex == 1)
            purchase.add(msgs);
        else if (maxindex == 2)
            bank.add(msgs);
        else if (maxindex == 3)
            promotional.add(msgs);
        else if (maxindex == 4)
            network.add(msgs);
        else if (maxindex == 5)
          otpmsgs.add(msgs);
        else
            spam.add(msgs);
    }

} catch (IOException e) {
    e.printStackTrace();
}
```

Data Pre-Processing

- This Module basic task is to prepare the data so that our classification algorithms can be implemented on it. The Algorithms do not directly take the text as an input as all of them work on Numbers.

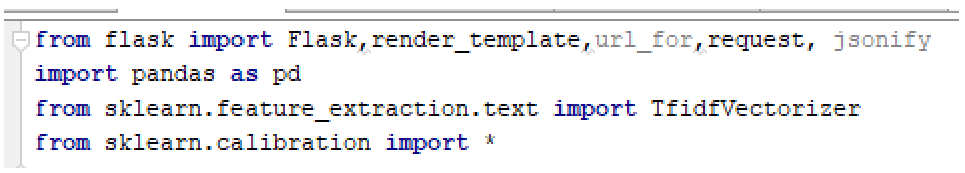
```
dataframe  =  pd.read_csv('dataset_final.csv'  ,  encoding='iso8859'  ,  usecols=[
'label' , 'text' ] )
dataframe = dataframe[dataframe['text'].notnull()]
```

- We first load our dataset into numpy arrays and then change the 7 classes labels into numbers.
- Then we tokenize the text the sms part of the data using keras class tokenizer.
- We also pad messages to make all of them of same length.

```
tokenizer = tf.keras.preprocessing.text.Tokenizer()
tokenizer.fit_on_texts( texts )
tokenized_messages = tokenizer.texts_to_sequences( texts )
padded_messages          =         tf.keras.preprocessing.sequence.pad_sequences(
tokenized_messages , maxlen )
```

In Web App:

- **Importing required modules: -**
    1. **Flask:** Library to implement flask microframework.
    2. **Jinja2:** A templating engine.
    3. **Pandas:** Python library for data manipulation and analysis.
    4. **Sklearn:** Python library for various classification, regression and clustering algorithms including support vector machines.

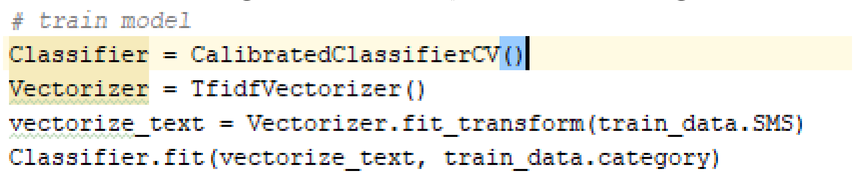


```python
from flask import Flask,render_template,url_for,request, jsonify
import pandas as pd
from sklearn.feature_extraction.text import TfidfVectorizer
from sklearn.calibration import *
```

- **Read CSV file using read_csv and prepare train_data and test_data**

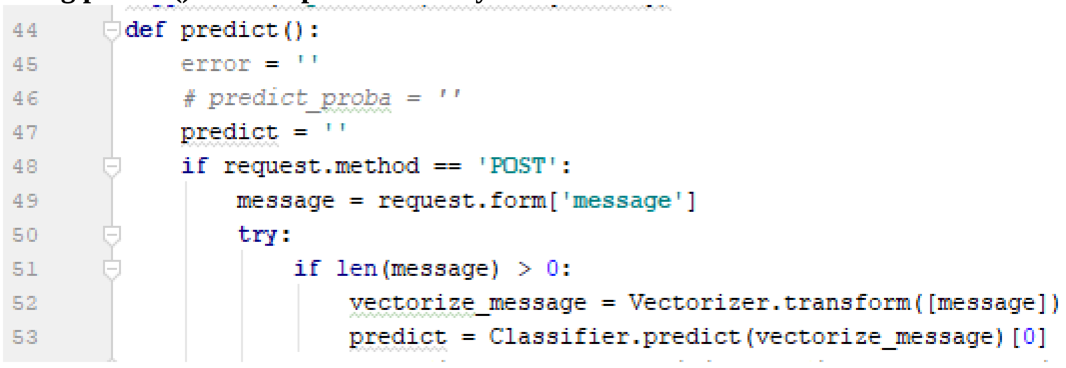
```
df = pd.read_csv('junkiller.csv', encoding='latin-1')
df.drop(['Unnamed: 2', 'Unnamed: 3', 'Unnamed: 4'], axis=1, inplace=True)
df['SMS'] = df['text']
df['category'] = df['label'].map({'bud': 0, 'pms': 1, 'pro': 2, 'nwm': 3, 'otp': 4, 'pud': 5, 'spm': 6})
df.drop(['label', 'text'], axis=1, inplace=True)
train_data = df[:10114]
test_data = df[10114:]
```

- **Tokenize data using TfidTokenizer() and fit data using fit_transform()**



```
# train model
Classifier = CalibratedClassifierCV()
Vectorizer = TfidfVectorizer()
vectorize_text = Vectorizer.fit_transform(train_data.SMS)
Classifier.fit(vectorize_text, train_data.category)
```

- **Using predict() method predict/classify**



```
def predict():
    error = ''
    # predict_proba = ''
    predict = ''
    if request.method == 'POST':
        message = request.form['message']
        try:
            if len(message) > 0:
                vectorize_message = Vectorizer.transform([message])
                predict = Classifier.predict(vectorize_message)[0]
```

10.Result

After comparison among Machine Learning algorithms Multinomial Naïve Bayes, Support Vector Classifier and k nearest neighbors. Below were the accuracies achieve on the data collected by us.

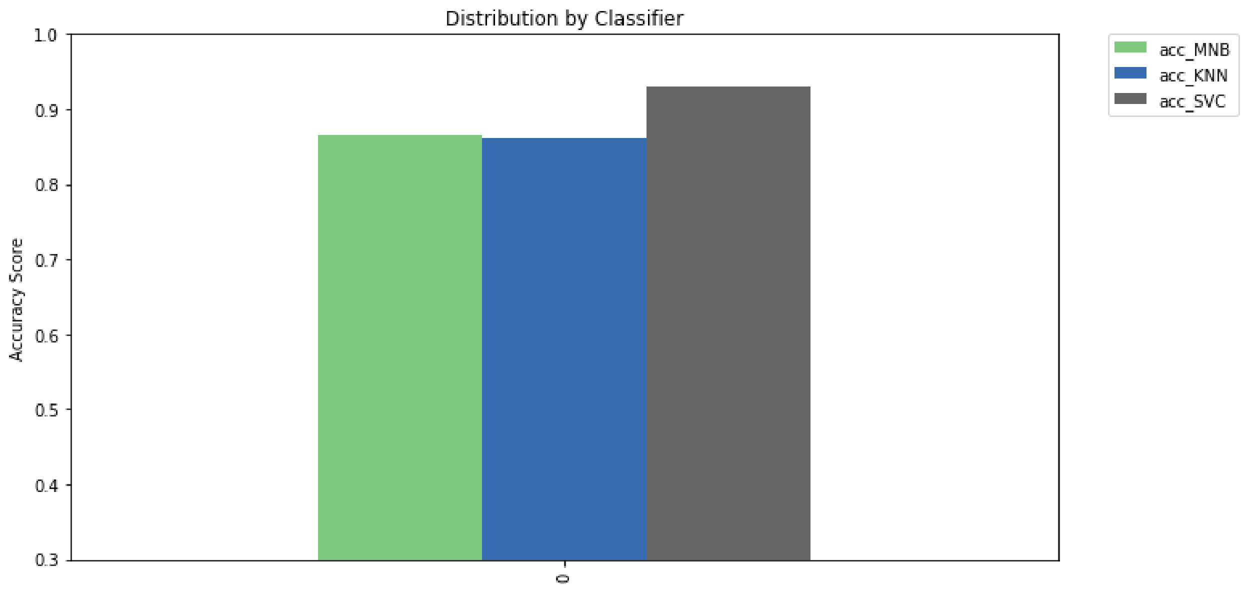


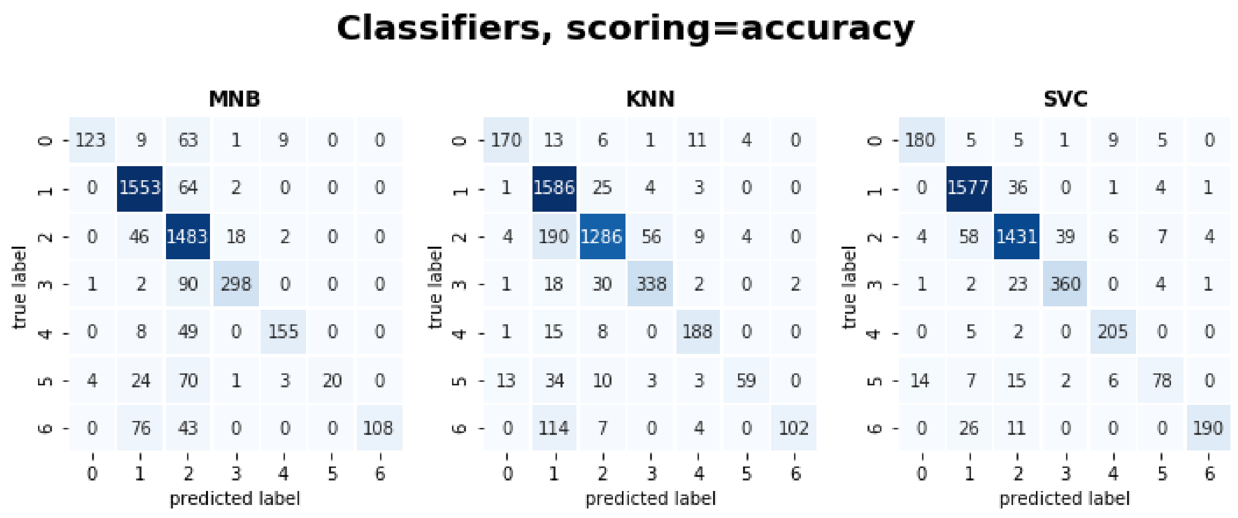
Fig-18(Accuracy)

Following are the accuracy achieved on test dataset.

Algorithm	Accuracy
MNB	0.8647398843930636
KNN	0.8621965317919075
SVC	0.9297109826589596

The confusion matrix will give an overview of classification results: The elements on diagonal are the number of points for which the predicted label is same to the true label, while the off-diagonal elements are those the classifier mislabeled. It is better if the diagonal values of confusion matrix are higher, this is indicating that many correct predictions are made by the algorithm. In the confusion matrix the rows correspond to the actual classes and columns denote the predicted classes.

For a Multi class classification task where number of classes in our case is 7, there are 49 possible results. Below is the Confusion matrix for each of the algorithm.



MNB

true label \ predicted label	0	1	2	3	4	5	6
0	123	9	63	1	9	0	0
1	0	1553	64	2	0	0	0
2	0	46	1483	18	2	0	0
3	1	2	90	298	0	0	0
4	0	8	49	0	155	0	0
5	4	24	70	1	3	20	0
6	0	76	43	0	0	0	108

KNN

true label \ predicted label	0	1	2	3	4	5	6
0	170	13	6	1	11	4	0
1	1	1586	25	4	3	0	0
2	4	190	1286	56	9	4	0
3	1	18	30	338	2	0	2
4	1	15	8	0	188	0	0
5	13	34	10	3	3	59	0
6	0	114	7	0	4	0	102

SVC

true label \ predicted label	0	1	2	3	4	5	6
0	180	5	5	1	9	5	0
1	0	1577	36	0	1	4	1
2	4	58	1431	39	6	7	4
3	1	2	23	360	0	4	1
4	0	5	2	0	205	0	0
5	14	7	15	2	6	78	0
6	0	26	11	0	0	0	190

Along with this we made a Neural Network on the same dataset to integrate into our android app.

We Trained on 11532 samples, validate on 2883 samples. And 2883 samples we kept for testing.

Final Hyper Parameters that we chose after multiple iterations are

Batch Size – 50

Epoch- 200

Learning Rate – 0.0001

Value of alpha for Leaky reLU – 0.2

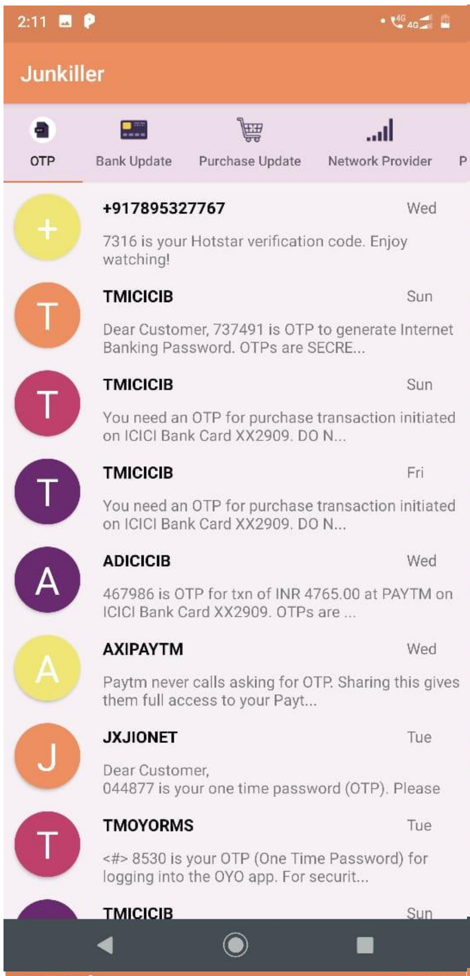
While training the last sample - loss: 0.9335 - acc: 0.6899 - val_loss: 0.9661 - val_acc: 0.6878

While testing sample - loss: 0.9661 - acc: 0.6878

Loss of 0.9660796909500979 Accuracy of 68.78252029418945 %

Accuracy of nearly 69% was achieved and the android app was built using the tflite model of this Keras model.

Screenshots of the result are attached on the next page.



+917895327767 Wed
7316 is your Hotstar verification code. Enjoy watching!

TMICICIB Sun
Dear Customer, 737491 is OTP to generate Internet Banking Password. OTPs are SECRE...

TMICICIB Sun
You need an OTP for purchase transaction initiated on ICICI Bank Card XX2909. DO N...

TMICICIB Fri
You need an OTP for purchase transaction initiated on ICICI Bank Card XX2909. DO N...

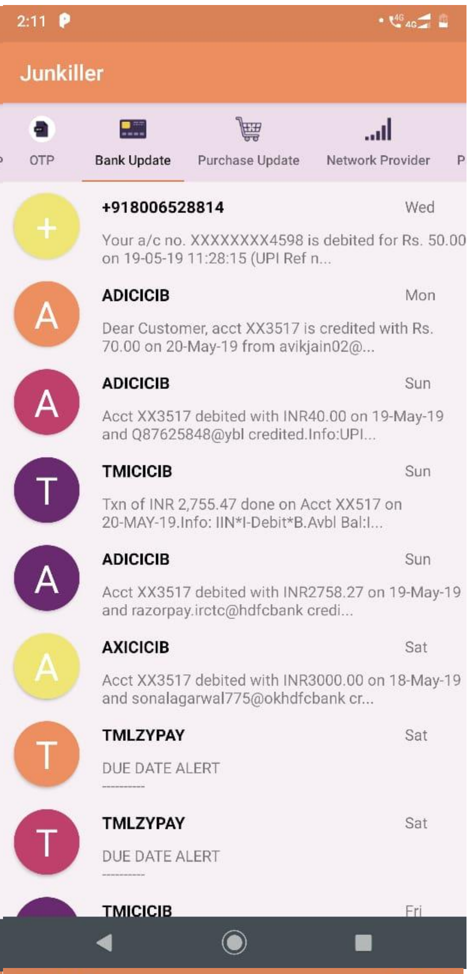
ADICICIB Wed
467986 is OTP for txn of INR 4765.00 at PAYTM on ICICI Bank Card XX2909. OTPs are ...

AXIPAYTM Wed
Paytm never calls asking for OTP. Sharing this gives them full access to your Payt...

JXJIONET Tue
Dear Customer,
044877 is your one time password (OTP). Please

TMOYORMS Tue
<#> 8530 is your OTP (One Time Password) for logging into the OYO app. For securit...

TMICICIB Sun



+918006528814 Wed
Your a/c no. XXXXXXXX4598 is debited for Rs. 50.00 on 19-05-19 11:28:15 (UPI Ref n...

ADICICIB Mon
Dear Customer, acct XX3517 is credited with Rs. 70.00 on 20-May-19 from avikjain02@...

ADICICIB Sun
Acct XX3517 debited with INR40.00 on 19-May-19 and Q87625848@ybl credited.Info:UPI...

TMICICIB Sun
Txn of INR 2,755.47 done on Acct XX517 on 20-MAY-19.Info: IIN*I-Debit*B.Avbl Bal:I...

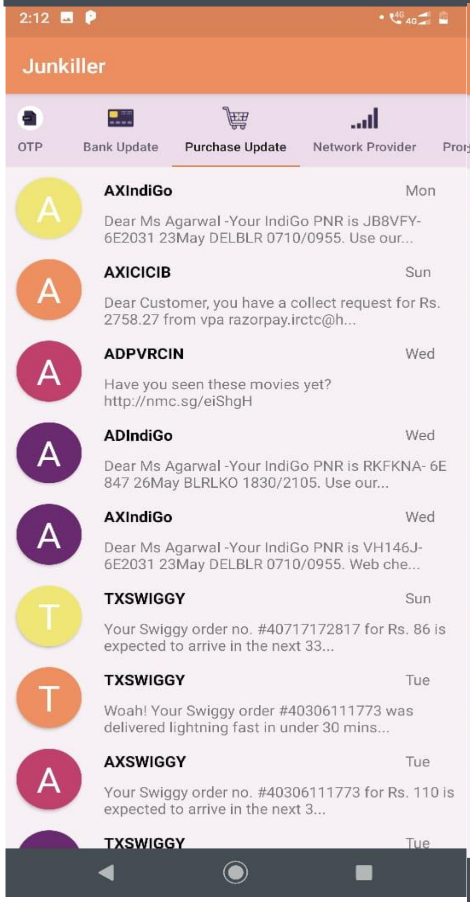
ADICICIB Sun
Acct XX3517 debited with INR2758.27 on 19-May-19 and razorpay.irctc@hdfcbank credi...

AXICICIB Sat
Acct XX3517 debited with INR3000.00 on 18-May-19 and sonalagarwal775@okhdfcbank cr...

TMLZYPAY Sat
DUE DATE ALERT

TMLZYPAY Sat
DUE DATE ALERT

TMICICIB Fri



AXIndiGo Mon
Dear Ms Agarwal -Your IndiGo PNR is JB8VFY-6E2031 23May DELBLR 0710/0955. Use our...

AXICICIB Sun
Dear Customer, you have a collect request for Rs. 2758.27 from vpa razorpay.irctc@h...

ADPVRCIN Wed
Have you seen these movies yet? http://nmc.sg/eiShgH

ADIndiGo Wed
Dear Ms Agarwal -Your IndiGo PNR is RKFKNA- 6E 847 26May BLRLKO 1830/2105. Use our...

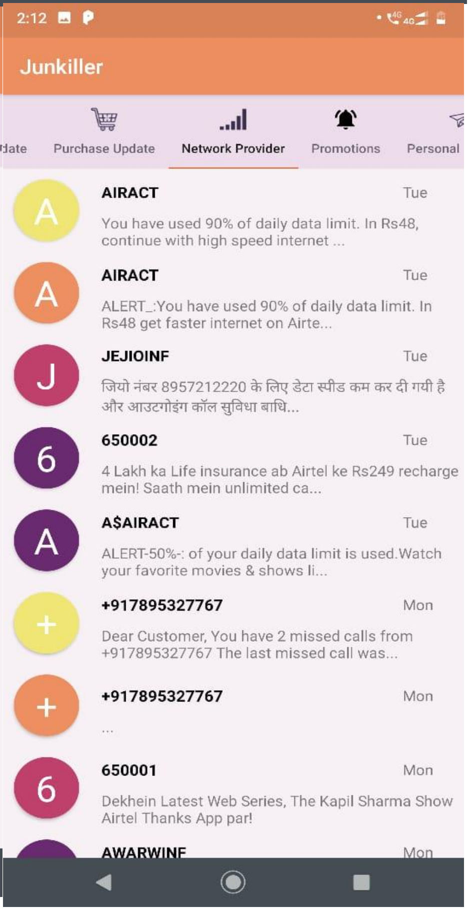
AXIndiGo Wed
Dear Ms Agarwal -Your IndiGo PNR is VH146J-6E2031 23May DELBLR 0710/0955. Web che...

TXSWIGGY Sun
Your Swiggy order no. #40717172817 for Rs. 86 is expected to arrive in the next 33...

TXSWIGGY Tue
Woah! Your Swiggy order #40306111773 was delivered lightning fast in under 30 mins...

AXSWIGGY Tue
Your Swiggy order no. #40306111773 for Rs. 110 is expected to arrive in the next 3...

TXSWIGGY Tue



AIRACT Tue
You have used 90% of daily data limit. In Rs48, continue with high speed internet ...

AIRACT Tue
ALERT_:You have used 90% of daily data limit. In Rs48 get faster internet on Airte...

JEJIOINF Tue
जियो नंबर 8957212220 के लिए डेटा स्पीड कम कर दी गयी है और आउटगोइंग कॉल सुविधा बाधि...

650002 Tue
4 Lakh ka Life insurance ab Airtel ke Rs249 recharge mein! Saath mein unlimited ca...

A$AIRACT Tue
ALERT-50%-: of your daily data limit is used.Watch your favorite movies & shows li...

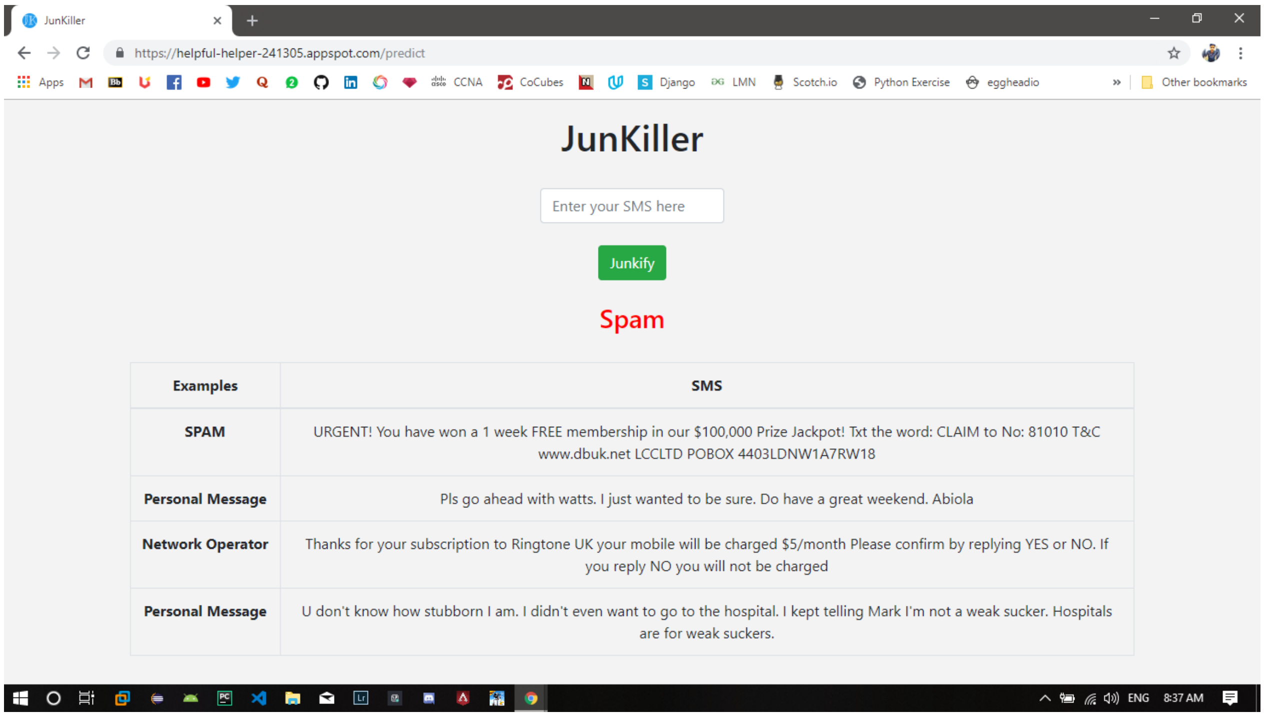
+917895327767 Mon
Dear Customer, You have 2 missed calls from +917895327767 The last missed call was...

+917895327767 Mon
...

650001 Mon
Dekhein Latest Web Series, The Kapil Sharma Show Airtel Thanks App par!

AWARWINF Mon

Fig-19(App Screenshots)



JunKiller

Spam

Examples	SMS
SPAM	URGENT! You have won a 1 week FREE membership in our $100,000 Prize Jackpot! Txt the word: CLAIM to No: 81010 T&C www.dbuk.net LCCLTD POBOX 4403LDNW1A7RW18
Personal Message	Pls go ahead with watts. I just wanted to be sure. Do have a great weekend. Abiola
Network Operator	Thanks for your subscription to Ringtone UK your mobile will be charged $5/month Please confirm by replying YES or NO. If you reply NO you will not be charged
Personal Message	U don't know how stubborn I am. I didn't even want to go to the hospital. I kept telling Mark I'm not a weak sucker. Hospitals are for weak suckers.

Fig-20- Web app Interface.

11. Conclusion and future scope

Although Neural Networks are considered the best for this classification when data is huge but in the case like our when the data is imbalanced, they tend to perform rather poorly as compared to traditional machine learning approaches. SVC out performed every other technique that we used.

Future Scope –

- Improvement in Neural Network accuracy.
- Increasing the size of the dataset so that our model can generalize better.
- Functionality to Classify messages from other apps
- Functionality to Classify messages also on the basis of sender.

.

References

[1] https://www.techsoupcanada.ca/en/learning_center/10_sfm_explained.

[2] Rish I., “An empirical study of the Naive Bayes classifier”, In Proceeding of IJCA Workshop on Empirical Methods in AI,2001.

[3] Richard D., Peter E. and David S., “Pattern Classification”,

[4] https://skymind.ai/wiki/neural-network

[5] W Stephen M., “Machine learning An Algorithm Perspective”, Chapman & Hall/CRC Publisher, 2009.

[6] “Towards Filtering of SMS Spam Messages Using Machine Learning Based Technique” Neelam Choudhary and Ankit Kumar Jain.

[7] “Contributions to the Study of SMS Spam Filtering: New Collection and Results” Tiago A. Almeida

[8] “SMS Spam Detection using Deep Neural Network” M. Nivaashini, R.S.Soundariya, A.Kodieswari, P. Thangaraj.

[9] “Content based Hybrid SMS Spam Filtering System” Thusitha Dayaratne