Dharmsinh Desai University, Nadiad

Faculty of Technology, Department of Computer Engineering

B.Tech. CE Semester – VI

Subject: (CE-621) System Design Practice

Project Title:

Spam Text Filter

Developed By:

Hetul Bhatt (Roll No.: CE015, ID: 17CEUOS123)

Jainik Gadara (Roll No.: CE047, ID: 17CEUON116)

Guided By:

Prof. Hariom A. Pandya

Dharmsinh Desai University, Nadiad

Faculty of Technology, Department of Computer Engineering

CERTIFICATE

This is to certify that System Design Practice project entitled “Spam text filter” is the bona fide report of work carried out by

1) **Hetul Bhatt (17CEUOS123)**
2) **Jainik Gadara (17CEUON116)**

of Department of Computer Engineering, Semester VI, academic year 2019-20, under our supervision and guidance.

Guide	HOD
Prof. Hariom A. Pandya Assistant Professor of Department of Computer Engineering, Dharmsinh Desai University, Nadiad.	**Dr. C. K. Bhensdadia** Head of the Department of Department of Computer Engineering, Dharmsinh Desai University, Nadiad.

Table of Contents

1. Abstract

Currently, SMS is one of the most important methods of communication. However, the increasing of spam SMS causes traffic congestion, decreasing productivity, phishing, which has become a serious problem for our society. And the number of spam SMS is increasing every year. Therefore, spam SMS filtering is an important, meaningful and challenging topic. The aim of this research is to find an effective solution to filter possible spam SMS. We have developed this software in accordance with these points so that seamless visual SMS segregation takes place.

2. Introduction

We all might have received many SMS from an unknown, congratulating your victory in contest or lottery and requesting you to share your bank account details to avail benefit. Here Spam text filter comes into picture, which helps us by labeling such messages as spam so that we are not mistaken.

Spam text filter is a mobile application for detecting spam messages from your message inbox.

Application reads all your inbox messages and checks whether message is spam or not. Application send message body to API and API in turn returns response whether message is spam or ham.

Application also preserves privacy before reading messages from inbox. User must permit application to read message inbox after which application can do its work.

We have used Bayes Theorem for calculating probability. And a Naive Bayes Classifier to develop our predicting model.

2.2 Tools/Technologies Used

Technologies:

- Flutter SDK
- Python libraries
    - NumPy
    - Pandas
    - NLTK
    - Scikit-learn
- Flask

Platforms:

- Android
- iOS

Tools:

- Google Colab
- Visual Studio Code
- Android Studio

3. Software Requirements Specification

Functional Requirements

R1 Display Spam messages
Description: User can see list of all spam messages.
Input: User selection
Output: List of Spam messages

R1.1 Delete message(s)
Description: After identifying message as spam, user has option to delete spam messages.

R1.1.1 Delete single message
Description: User delete a Spam message.
Input: User selection.
Output: Updated list of Spam messages.

R1.1.2 Delete multiple messages
Description: User can delete multiple spam messages by selecting them.
Input: User selected messages.
Output: Updated list of spam messages.

R1.2 Block sender
Description: User can block a sender who has send spam message.
Input: User selects the sender.
Output: Success or failure message.

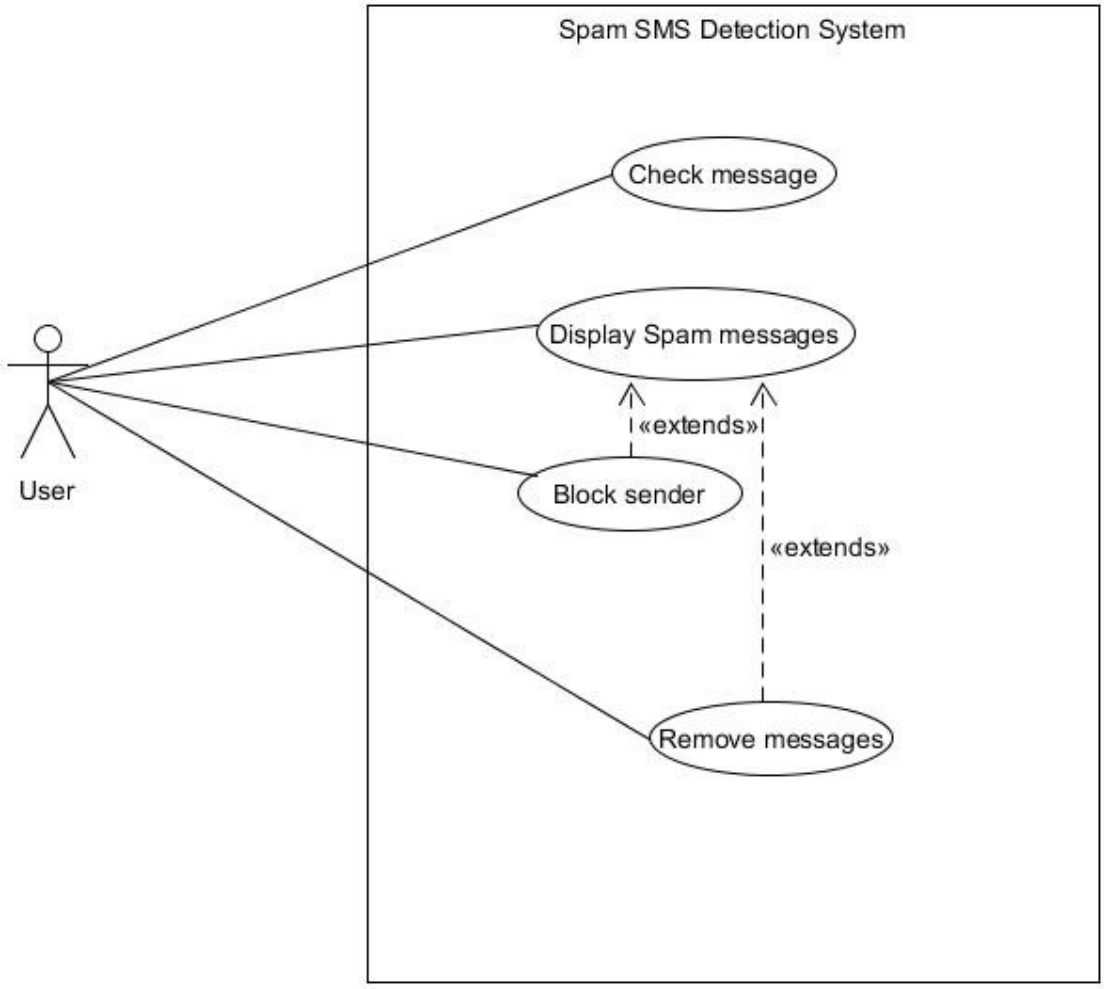
R2 Check message if Spam or nor
Description: When user receives a message, system automatically checks whether the received message is spam or not.
Input: Received message
Output: Informing whether message is spam or not.

Non-Functional Requirements

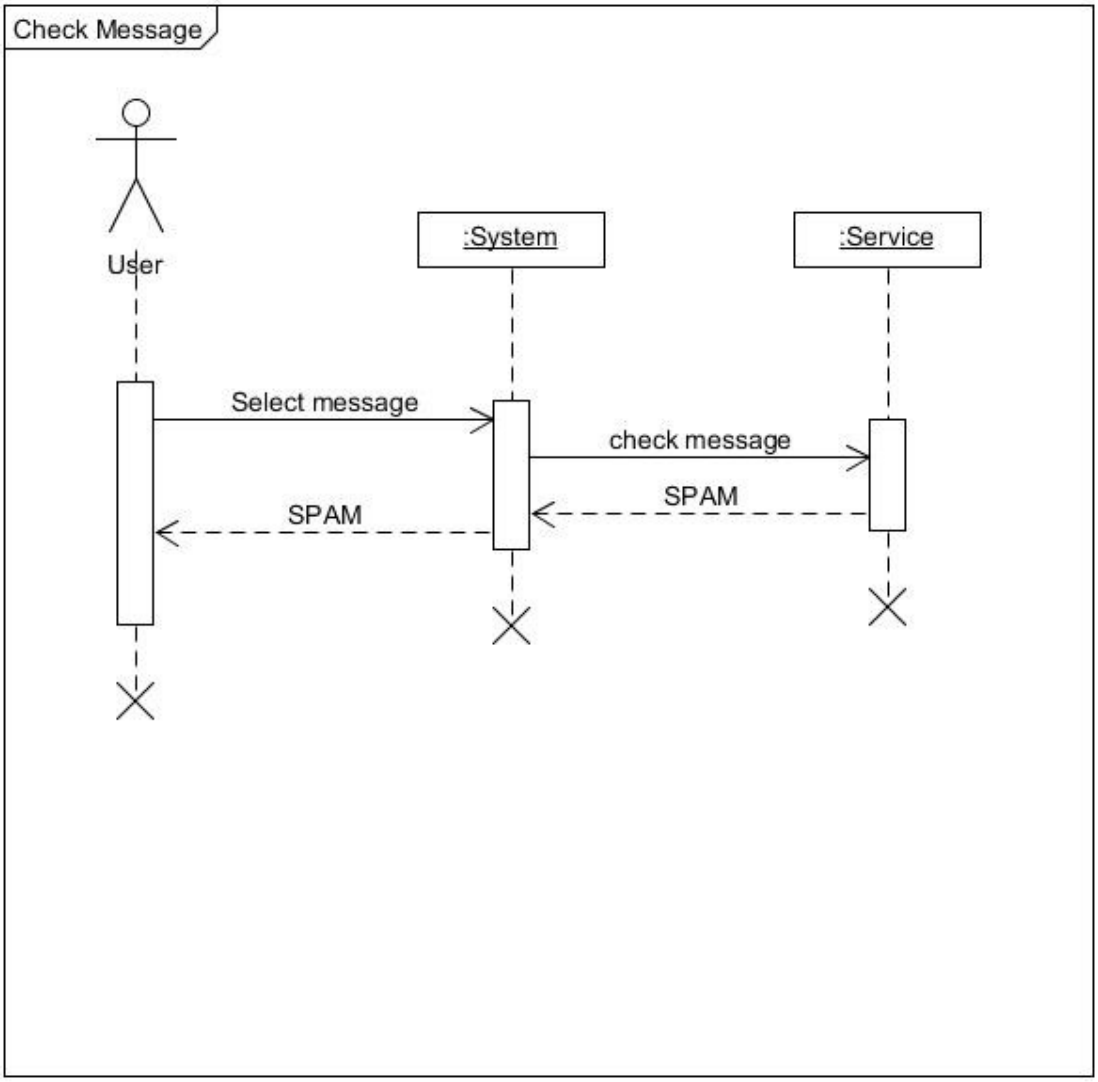
- System should always be available.
- System should be architected in such a way that future expansions is easy.
- System should be easy to access.
- System should be highly secured so that no can intercept messages in between.

4. Designs

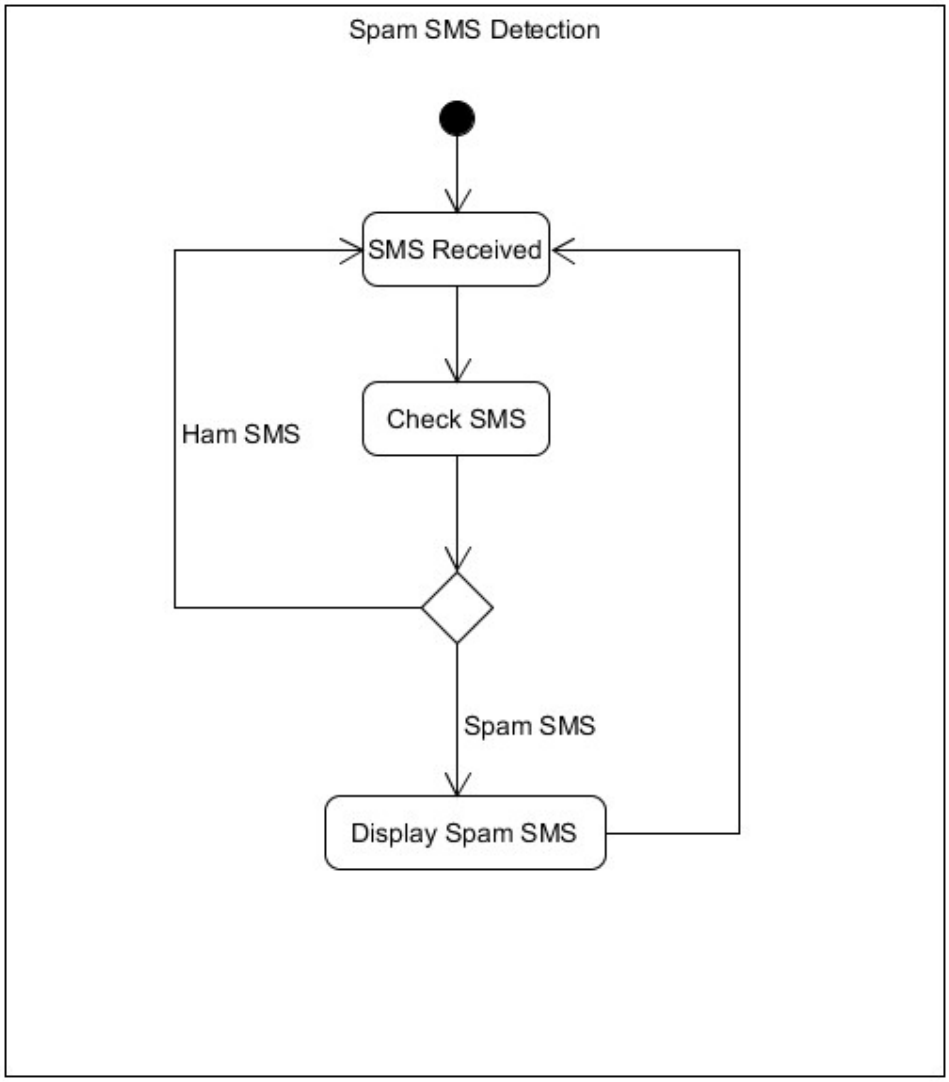
4.1 Use Case Diagram



4.2 Sequence Diagram



4.3 Activity Diagram



5. Implementation Details

Prediction Module

This module is the key feature of the system. Prediction module predicts whether a message is spam or ham. Module takes SMS body as an input and output string “Spam” or “Ham”.

System is a machine learning model where we have used concept of Bayes probability theorem to predict output. Bayes Theorem is also widely used in the field of machine learning. Bayes theorem is used in developing models for classification predictive modelling problems such as the Bayes Optimal Classifier and Naive Bayes.

For developing model for our system, we have used Naïve Bayes Modelling approach. We have divided dataset in ratio of 80:20, where 80% of data is for training and remaining 20% is for testing.

First, model removes all the stops words and punctuations from the message. After that messages in tokenizes into a matrix and padded with zeros if matrix size if not enough. Finally, the array after padding with zeros is converted into a csr matrix and is input to Bayes classifier’s predict method which outputs whether message is ham or spam.

Display Module

We have created our frontend as a mobile application where user can check spam messages. User needs to provide permission for reading all inbox messages. After that app makes API calls to predict whether the message is spam or not.

We have used flutter framework for developing android app. We have used SMS package to query messages from inbox. All messages are displayed in Listview for better readability along with its output as ‘ham’ or ’spam’.

Calculated example showcasing underlying logic of Naïve Bayes' Classifier

According to the theory of conditional probability,

$$P(A|B) = \frac{P(A \cap B)}{P(B)} \Rightarrow P(A \cap B) = P(B) \cdot P(A|B) \quad -(1)$$

$$P(B|A) = \frac{P(B \cap A)}{P(A)} \Rightarrow P(B \cap A) = P(A) \cdot P(B|A) \quad -(2)$$

considering eqn 1 and 2;

$$P(B) \cdot P(A|B) = P(A) \cdot P(B|A)$$

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

The formula written above is the mathematical representation of the Bayes Theorem

Now consider, out of 100 SMSs → 75 Ham / 25 Spam

Word 'Buy' occures in 20/25 (Spam SMS)

5/75 (Ham SMS)

Ly; 'Cheap' → 15/25 (S)

10/75 (H)

$$P(S|\text{'Buy'}) = \frac{P(S) \cdot P(\text{'Buy'}|S)}{P(\text{'Buy'})}$$

$$= \frac{0.25 \times 0.8}{0.25} = 80\%.$$

Ly; $P(S|\text{'cheap'}) = \frac{P(S) \cdot P(\text{'cheap'}|S)}{P(\text{'cheap'})}$

$$= \frac{0.25 \times 0.6}{0.25} = 60\%.$$

What we just saw is good enough for similar words but in SMSs multiple words are contained. This methodology turns out to be extremely costly in such scenarios. Thus, for multiple words, we make the following naive assumption.

$P(B \cap C)$ => probability of 'Buy' and 'cheap' containing phrase being a spam.

$P(B \cap C) = P(B) \cdot P(C)$ (Scalar multiplication of probabilities of both the words being/making Spam independently).

thus; $P(S \mid B \cap C) = \dfrac{P(B|S)\, P(C|S)\, P(S)}{P(B|S) \cdot P(C|S) \cdot P(S) + P(B|H) \cdot P(C|H) \cdot P(H)}$ (by Bayes th^m)

($\because P(B \cap C \mid S) = P(B|S) \cdot P(C|S)$ i.e. naive assumption)

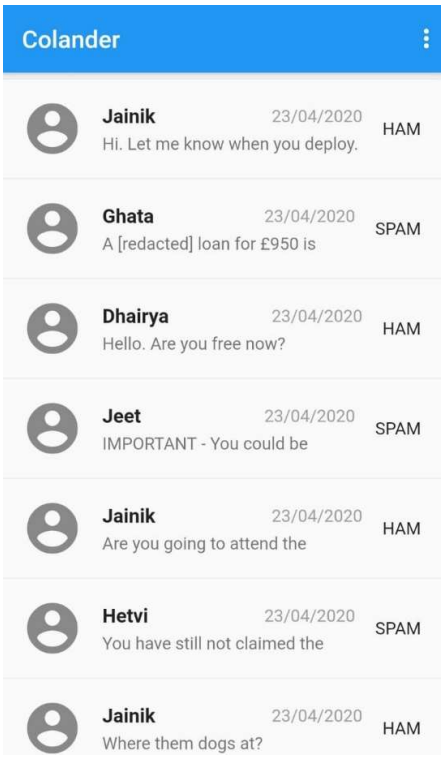
$$= \frac{0.8 \times 0.6 \times 0.25}{(0.8 \times 0.6 \times 0.25) + (5/75 \cdot 10/75 \cdot 75/100)}$$

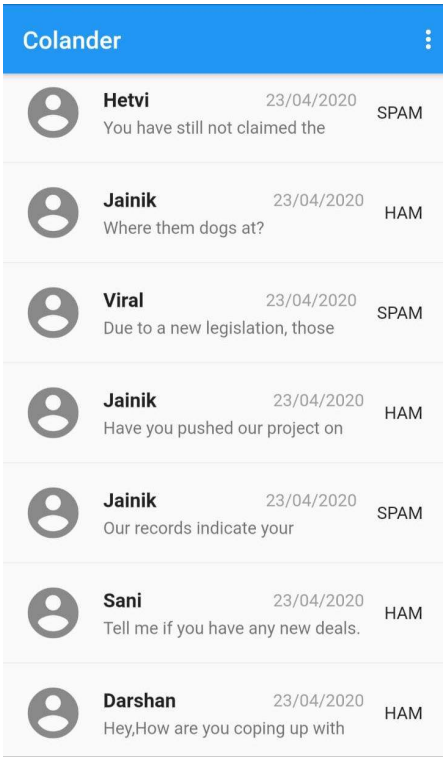
$$= \frac{0.12}{0.12 + 0.008} = 0.9473$$

Thus, we have calculated the probability by applying a naive assumption over the Bayes theorem. Thus, we call this classifier, 'Naive Bayes Classifier'.

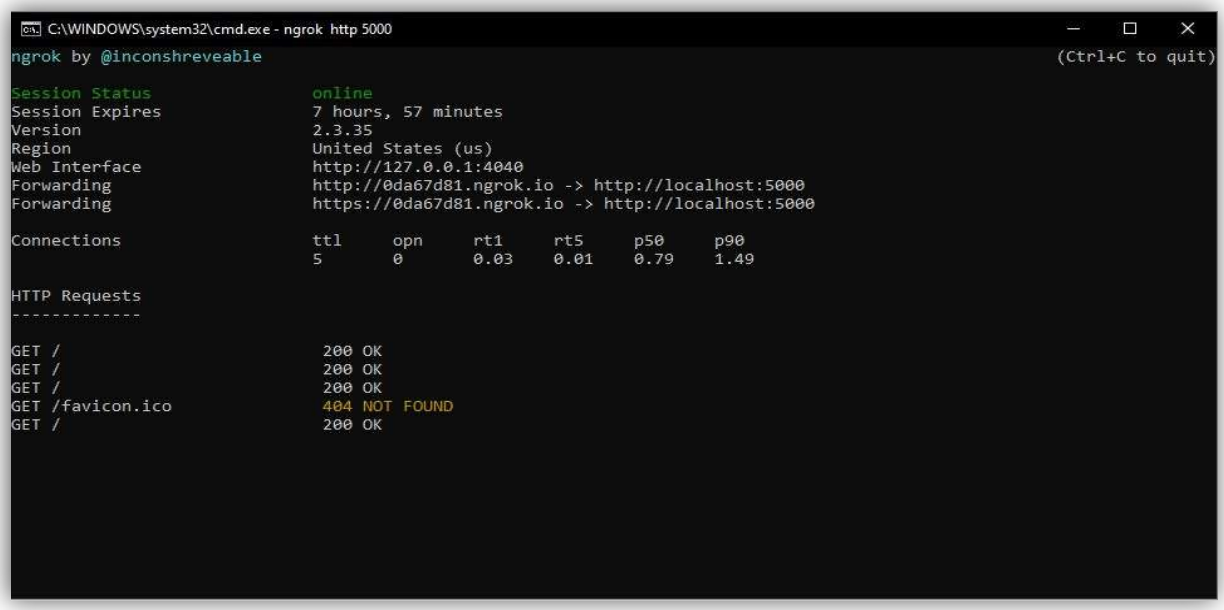
6. Screenshots

6.1 Reading messages from inbox and predicting as spam or ham.

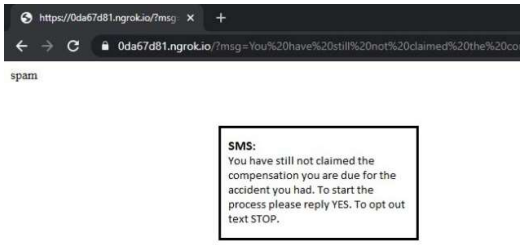


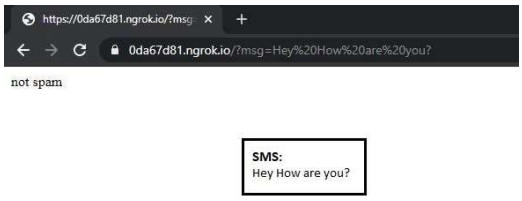


6.2 Deploying the API and opening the port using tunnel to localhost



6.3 API call from browser





7. Conclusions

We have successfully developed a machine learning model using various python libraries viz., scikit-learn, NumPy, pandas, NLTK, and flask to host API.

Clients can check whether the messages are spam or not. Our UI is more robust, attractive, and easy to use. API predicts the output at an accuracy of 95%.

8. Limitations

The software is almost complete. However, we can upload the service on the cloud for no downtime. This service can be used from computers for email as well. We are trying to figure out a way to run python scrips natively from android/iOS so that API calls need not be made. We have tried to find some alternatives for TFlite on flutter but haven't succeeded as of now.

We are also planning to have a couple of more user functionalities like to block a spam contact and delete spam messages.

9. Bibliography

Following links and websites were referred during the development of this project.

https://scikit-learn.org/stable/user_guide.html

https://numpy.org/doc/

https://pandas.pydata.org/docs/

https://www.nltk.org/py-modindex.html

https://stackoverflow.com/