A MINI PROJECT REPORT

ON

# "SPAM AND HAM DETECTION"


Sinhgad College of Engineering

Department of Computer Engineering

LABORATORY PRACTICE- II

(B.E Computer Engineering)

BY

| | |
|---|---|
| Tushar Agarwal | 405004 |
| Shloka Bhalgat | 405016 |
| Chirag Khandhar | 405073 |
| Bhumika Khetan | 405074 |

Year 2018-19

# CERTIFICATE


"**SPAM AND HAM DETECTION**"

Sinhgad College of Engineering

Department of Computer Engineering


| | |
|---|---|
| Tushar Agarwal | 405004 |
| Shloka Bhalgat | 405016 |
| Chirag Khandhar | 405073 |
| Bhumika Khetan | 405074 |


Prof. S. S. Pandhare
Internal Guide
Department of Computer Engineering

Prof. M. P. Wankhede
Head of Dept.
Department of Computer Engineering

Dr. S.D. Lokhande
Principal
SCOE, Pune

# CONTENTS

# Abstract

Numerous times we have been warned about Spam e-mails and text messages sent over the mobile phone. Spam text messages often times cause a nuisance to the user and may also pose as a threat to the device as well. Text messages sent on the mobile phone being shorter in size, makes it difficult for a normal spam detector to detect it as an unauthorized message. Advertisers rely on SMS spam because of the frequency at which they can resend a single message to multiple users without being identified as spam. Moreover, they tend to overpower automated filters which are pre-installed on the device.

This project involves a web application made using R where we simulate a system which can detect a text message as spam or ham depending on the words used in that *short message*. We have used the Naïve Bayes Text Classification algorithm to classify the messages depending on their class. The Bayesian formula along with the Naïve Classification algorithm helps to train the algorithm such that it avoids biasing and incorrect results. Using image media the results have been generated in the form of text and graphs. We have implemented word cloud representation for the dataset that is in action.

# List of Figures

# Acknowledgements

We would like to take this opportunity to thank all the people who were part of this project in numerous ways, people who gave unending support right from the initial stage. In particular, we wish to thank,

Our internal project guide **Prof. S. R. Hiray** and **Prof. S. S. Pandhare** who have given their co-ordination timely and precious guidance without which this project would not have been a success. We thank them for reviewing the entire project with painstaking efforts and more of them uncanny ability to spot the mistakes.

We would like to thank our HOD **Prof. M. P. Wankhede** for his continuous encouragement, support and guidance at each and every stage of development of this project.

And last but not the least we would like to thank all my friends who were associated with me and helped me in preparing my project. The project named **'Spam and Ham Detector'** would not been possible without the extensive support of people who were directly or indirectly in its successful evolution.

# Chapter 1

# Introduction

## 1.1 Background and Basics

As the worldwide use of mobile phones has grown, a new avenue for electronic junk mail has opened for disreputable marketers. These advertisers utilize Short Message Service (SMS) text messages to target potential consumers with unwanted advertising known as SMS spam. This type of spam is particularly troublesome because, unlike e-mail spam, many cellular phone users pay a fee per SMS received. Developing a classification algorithm that could filter SMS spam would provide a useful tool for cellular phone providers.

Since Naïve Bayes has been used successfully for e-mail spam filtering, it seems likely that it could also be applied to SMS spam. However, relative to e-mail spam, SMS spam poses additional challenges for automated filters. SMS messages are often limited to 160 characters, reducing the amount of text that can be used to identify whether a message is junk. The limit, combined with small mobile phone keyboards, has led many to adopt a form of SMS shorthand lingo, which further blurs the line between legitimate messages and spam. Let's see how a simple Naïve Bayes classifier handles these challenges.

## 1.2 Problem Statement and Scope

To design a text classification web application used determine whether the entered text is Spam or Ham using the Naïve Bayes classification algorithm.

# Chapter 2

# Project Planning and Methodology

## 2.1 System Requirements

The software's used for this project on Developers side are:

- **OS:** Windows 7 or above
- **Technology:** R-3.5.1
- **Framework:** RStudio
- **Server:** Shiny's built in server
- **Database**: none

The various packages used in the backend coding are:

- quanteda
- dplyr
- RColorBrewer
- ggplot2
- pROC
- shiny
- shinydashboard

The software's required for End-User are:

- **Operating System**: Windows/ MacOS/ Unix/ Android

- **Processor**: Core II or above (min 1.1 GHz)
- **Browser:** Google Chrome/ Mozilla Firefox
- **Network Requirements:** Internet Connectivity

## 2.2 Project Methodology

### 2.2.1 Prediction Using Naïve Bayes Classifier

Naive Bayes classifiers are a class of simple linear classifiers which are *conditional probability* models based on **Bayes** Theoram i.e

$P(Y\ 2\ Kj\ |Xi) = P(X1|Y\ ).P\ (X2|Y\ ).....P\ (Xi|Y\ ).\ P(Y\ 2\ Kj)$

*where Xi* are the number of inputs and Y is discrete response variable and *Kj* are the number of class labels.

The special thing about Naive Bayes classifiers are that they follow *Conditional Independence Theorem* i.e. the features *Xi* are uncorrelated and independent of each other which is often always crude. Secondly, they assume that the data samples are drawn from an identical and independent distribution- **IID** is the term which is famous in Statistics.

```
#separating Train and test data
spam.train<-spam[1:4458,]
spam.test<-spam[4458:nrow(spam),]

msg.dfm <- dfm(msg.corpus, tolower = TRUE)  #generating document freq matrix
msg.dfm <- dfm_trim(msg.dfm, min_count = 5, min_docfreq = 3)
msg.dfm <- dfm_weight(msg.dfm, type = "tfidf")

#trining and testing data of dfm
msg.dfm.train<-msg.dfm[1:4458,]

msg.dfm.test<-msg.dfm[4458:nrow(spam),]
```

**Fig. 2.2.1** Project Methodology-I

### 2.2.2 Training the Naïve Bayes classifier

```
nb.classifier<-textmodel_NB(msg.dfm.train,spam.train[,1])
nb.classifier
```

**Fig. 2.2.2** Project Methodology-II

```
## Fitted Naive Bayes model:
## Call:
##  textmodel_NB.dfm(x = msg.dfm.train, y = spam.train[, 1])
##
##
## Training classes and priors:
## spam  ham
##  0.5  0.5
##
##        Likelihoods:      Class Posteriors:
## 30 x 4 Matrix of class "dgeMatrix"
##                 spam          ham       spam       ham
## you     5.001507e-03 0.0096798156 0.34067144 0.6593286
## have    4.322289e-03 0.0042303673 0.50537386 0.4946261
## 1       2.695748e-03 0.0009529526 0.73882413 0.2611759
## new     3.492485e-03 0.0010753934 0.76457487 0.2354251
## .       6.965338e-03 0.0168302131 0.29271598 0.7072840
## please  2.339097e-03 0.0011593603 0.66860811 0.3313919
## call    1.058603e-02 0.0021859571 0.82884759 0.1711524
## i       8.439760e-04 0.0112106647 0.07001254 0.9299875
## wait    1.860817e-04 0.0011538316 0.13887596 0.8611240
## for     5.699340e-03 0.0045025239 0.55865674 0.4413433
## hope    2.334040e-04 0.0017258550 0.11912872 0.8808713
## tonight 1.137075e-04 0.0011106417 0.09287182 0.9071282
```

```
## too     3.802754e-05 0.0017024748 0.02184860 0.9781514
## bad     1.232420e-04 0.0006045270 0.16934219 0.8306578
## as      1.339518e-03 0.0020699791 0.39287852 0.6071215
## well    2.938089e-04 0.0017334850 0.14492664 0.8550734
## but     2.528948e-04 0.0043933716 0.05442968 0.9455703
## rock    3.802754e-05 0.0002684845 0.12406542 0.8759346
## night   3.003905e-04 0.0017976398 0.14317739 0.8568226
## anyway  3.802754e-05 0.0005405216 0.06572915 0.9342709
## going   1.538819e-04 0.0023951976 0.06036762 0.9396324
## a       7.856726e-03 0.0064918622 0.54756091 0.4524391
## now     6.254232e-03 0.0028758075 0.68501697 0.3149830
## good    6.723203e-04 0.0030342352 0.18138681 0.8186132
## speak   8.003838e-04 0.0004728416 0.62862694 0.3713731
## to      1.113210e-02 0.0075761991 0.59503541 0.4049646
## soon    2.642059e-04 0.0010608467 0.19939274 0.8006073
## today   1.120666e-03 0.0019041688 0.37048833 0.6295117
## is      4.451802e-03 0.0058153446 0.43359683 0.5664032
## accept  3.802754e-05 0.0003188419 0.10655871 0.8934413
```

### 2.2.3 Testing to model

```
pred<-predict(nb.classifier,msg.dfm.test)

#generating a confusion matrix

# use pred$nb.predicted to extract the class labels
table(predicted=pred$nb.predicted,actual=spam.test[,1])
```

```
##          actual
## predicted ham spam
##      ham  943    4
##      spam  19  149
```

```
#16 wrongly classified for ham and 7 examples wrongly classified for spam

#acccuracy of the classifier on Test data
mean(pred$nb.predicted==spam.test[,1])*100
```

```
## [1] 97.93722
```

**Fig. 2.2.3 (A)** Testing to model

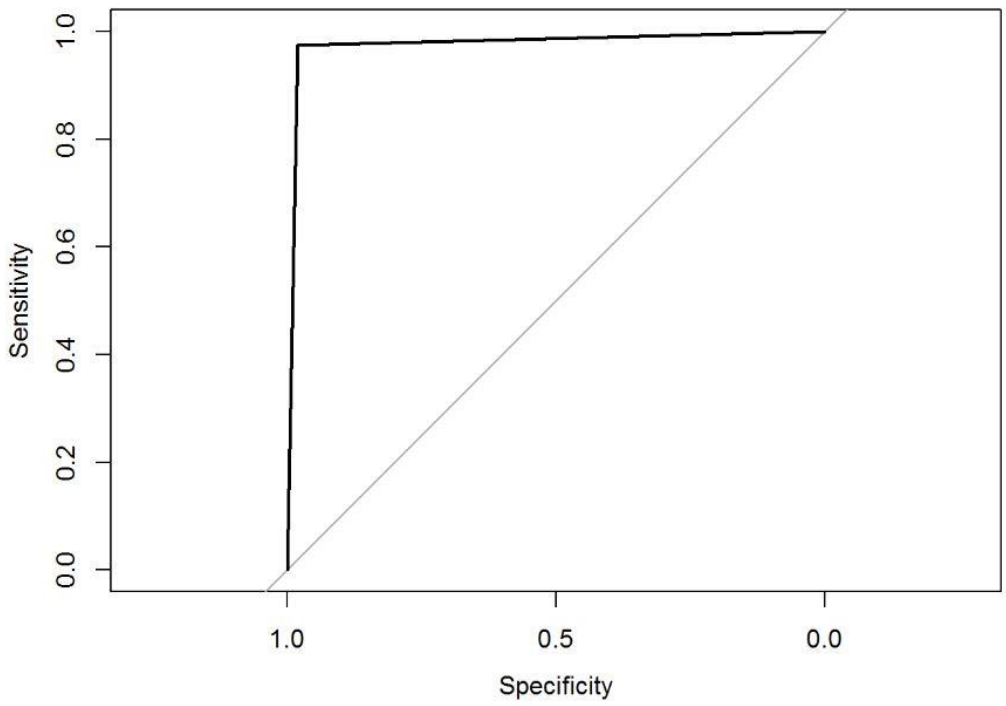



**Fig. 2.2.3 (B)** ROC Curve

## Area under the curve: 0.9771

# Area under the curve: 0.9679

In the ROC curve the area under the curve is **0.9679** which is a very nice score and implies that the model can easily recognize text messages as either spam or ham. ROC curve is plotted between **Sensitivity**-i.e. true positive rate (positive classes being classified correctly) vs. the **Specificity**-i.e. true negative rate (negative classes being classified correctly)

In the Confusion matrix, the **diagonals** are the correctly classified examples while the **off-diagonals** the incorrectly classifies examples.

# Chapter 3

# Implementation and Coding

## 3.1 Selenium

Selenium is an open-source tool that is used for test automation. It is licensed under Apache License 2.0. Selenium is a suite of tools that helps in automating only web applications.

The Selenium-IDE (Integrated Development Environment) is an easy-to-use Firefox plug-in to develop Selenium test cases. It provides a Graphical User Interface for recording user actions using Firefox which is used to learn and use Selenium, but it can only be used with Firefox browser as other browsers are not supported.

However, the recorded scripts can be converted into various programming languages supported by Selenium and the scripts can be executed on other browsers as well.

In our project we will be testing our “Spam or Ham Detector” web application which is built using Shiny in Rstudio.

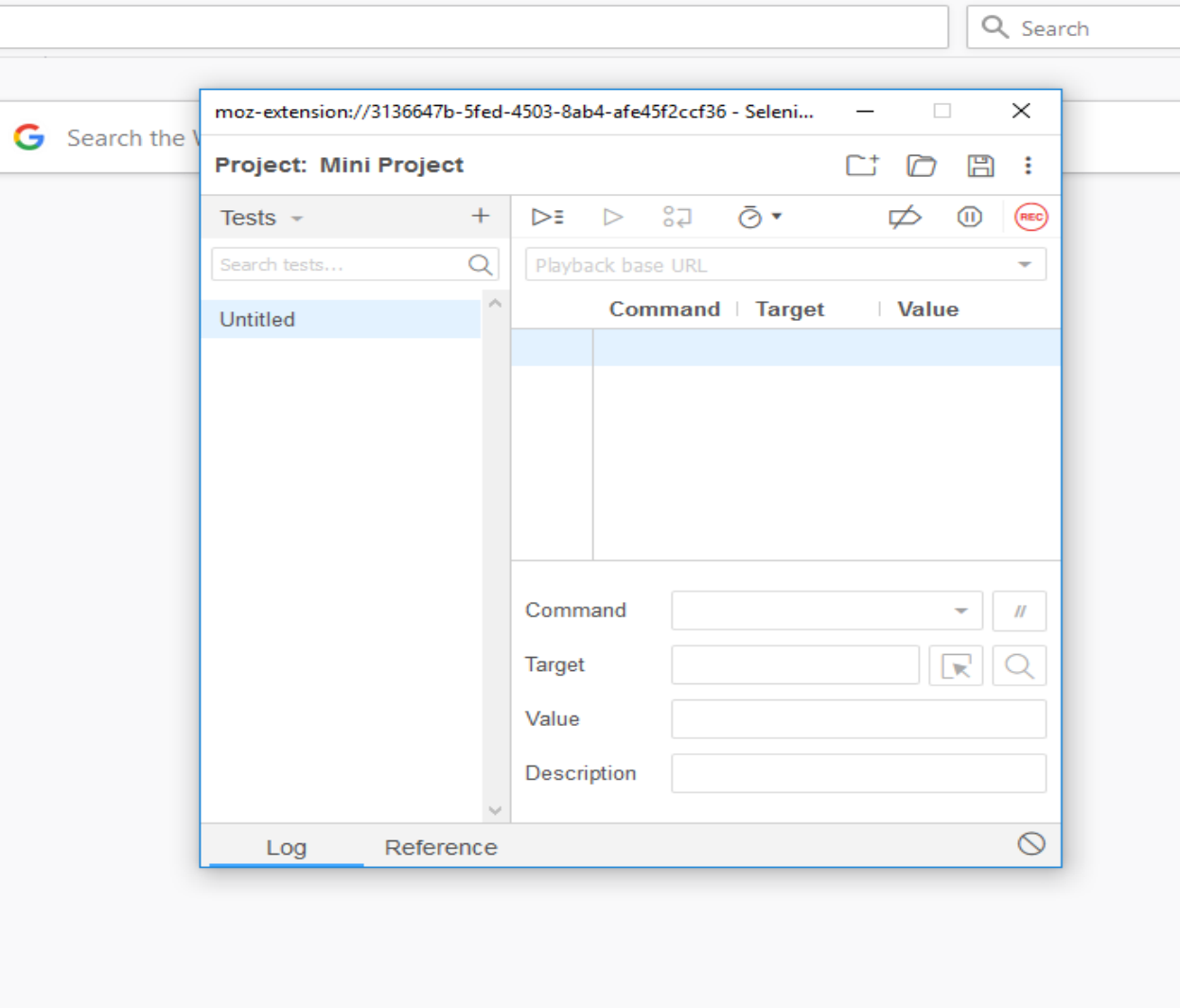



**Fig. 4.1.1 Selenium IDE**

## 3.2 GUI Design

R has a variety of GUI frameworks like RGtk2, gWidgets, and likewise. We have opted to use RShiny which is an R package that makes the developers work of making interactive web apps easier. It can be built straight from R. You can host standalone apps on a webpage or embed them in R Markdown documents or build dashboards. We can extend the Shiny apps with CSS themes, htmlwidgets, and JavaScript actions.

## 3.2.1 GUI Design/Screenshots

**A.** The home page or welcome window contains two buttons i.e. Dashboard and Details. Dashboard allows the user to give input to check whether a specified text is spam or ham. On the other hand the details section give the output for bar charts and word cloud

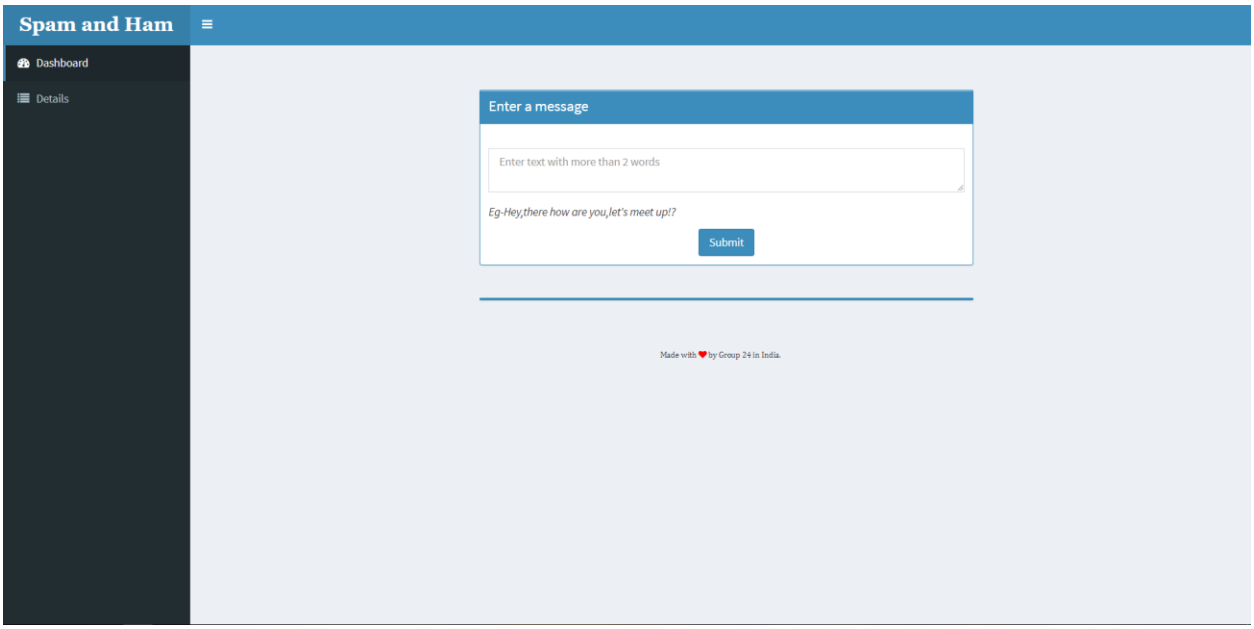



**Fig. 3.2.1.A1** Dashboard

**Fig. 3.2.1.A2** Details- Data Distribution


**Fig. 3.2.1.A3** Details- Word Clouds

**B.** After entering the test input "*Hi Emily, it was nice meeting you, let's catch up on Saturday at 9.30 p.m. at Ruby's Café*", the system detects the message as HAM.

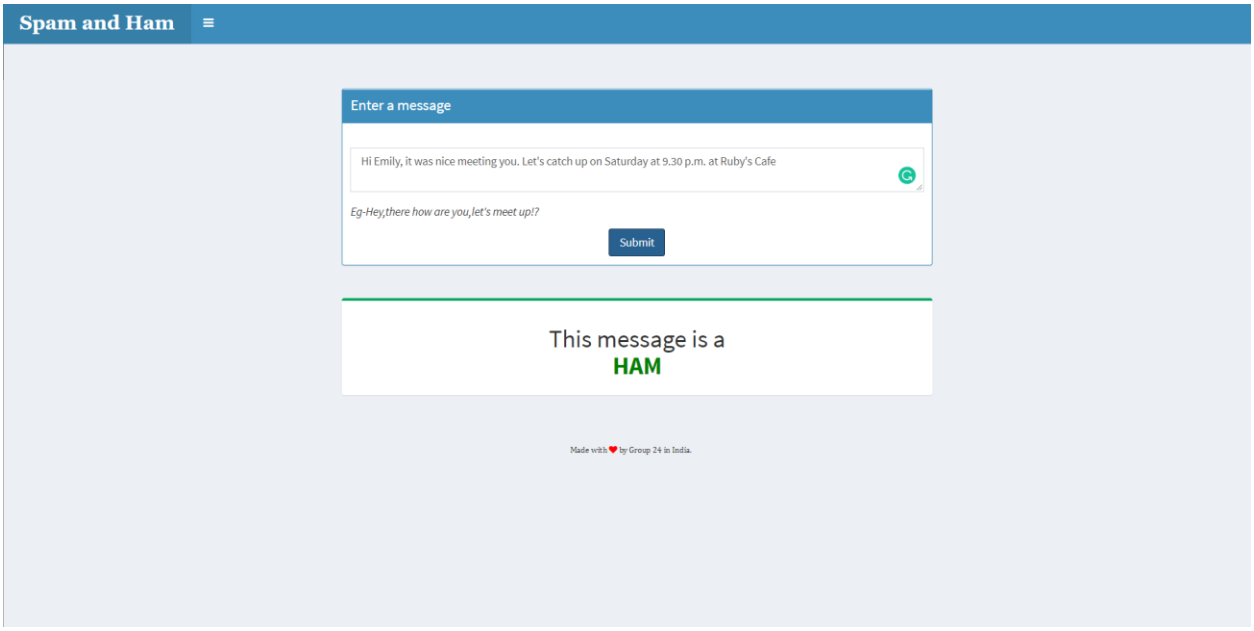


**Fig. 3.2.1(B)** Dashboard after Output- HAM

**C.** The details page gives final output in the form of statistical information with histograms and bar chart. The word cloud for spam and ham words is also displayed.

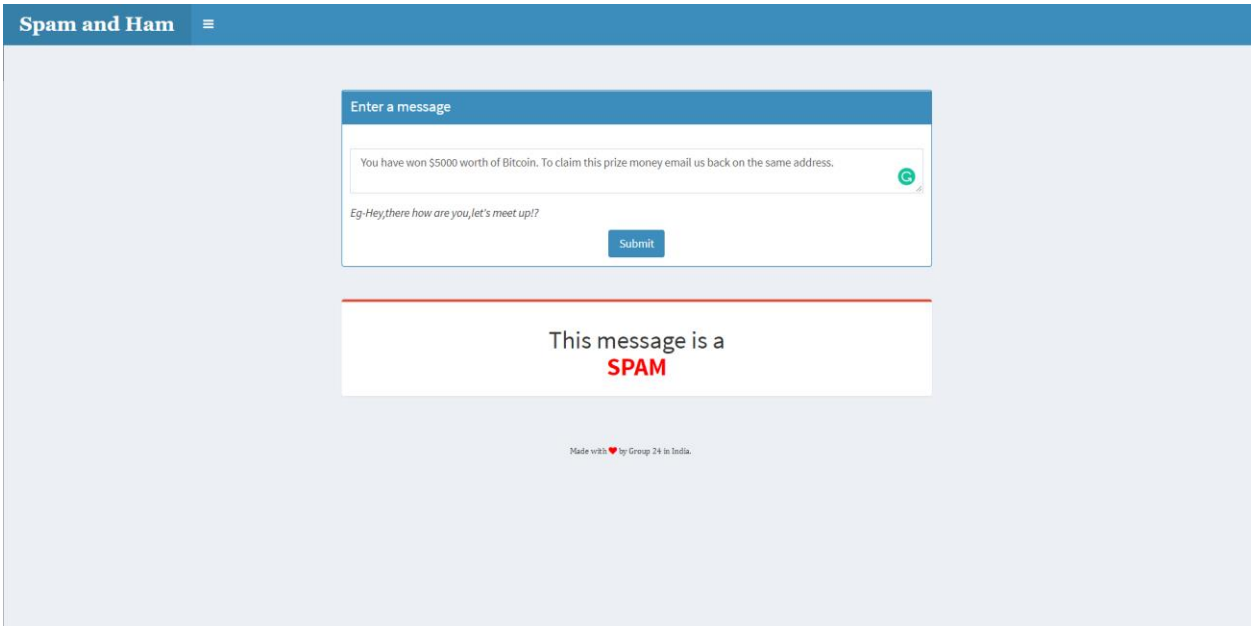



**Fig. 3.2.1(C)** Dashboard after Output- SPAM

# 3.3 Algorithms

## 3.3.1. Introduction:

Conceptually, this is a way to go from P (Evidence| Known Outcome) to P (Outcome Known Evidence). Often, we know how frequently some particular evidence is observed, *given a known outcome*. We have to use this known fact to compute the reverse, to compute the chance of that *outcome happening*, given the evidence.

P (Outcome given that we know some Evidence) = P (Evidence given that we know the Outcome) times P (Outcome), scaled by the P (Evidence)

## 2. Naïve Bayesian Classification:

It is based on the Bayesian theorem, it is particularly suited when the dimensionality of the inputs is high. Parameter estimation for naive Bayes models uses the method of maximum likelihood. In spite over-simplified assumptions, it often performs better in many complex real world situations.

**Advantage**: Requires a small amount of training data to estimate the parameters.

**Example:**

- **Training set**
    - Round-red
    - Round-orange
    - Oblong-yellow
    - Round-Red
- **Dataset**
    - round-red
    - round-orange
    - round-red
    - round-orange
    - oblong-yellow
    - round-red
    - round-orange
    - oblong-yellow
    - oblong-yellow
    - round-red

Naive Bayes classifiers are a class of simple linear classifiers which are conditional probability models based on **Bayes** Theorem i.e.

**P(Y 2 Kj |Xi) = P (X1|Y).P (X2|Y).....P (Xi|Y). P(Y 2 Kj)**

Where- Xi are the number of inputs and Y is discrete response variable and Kj are the number of class labels.

The special thing about Naive Bayes classifiers are that they follow Conditional Independence Theorem i.e. the features Xi are uncorrelated and independent of each other which is often always crude. Secondly, they assume that the data samples are drawn from an identical and independent distribution- **IID** is the term which is famous in Statistics.

## 3.4 Important libraries

### 3.4.1 Natural language processing package

**quanteda** makes it easy to manage texts in the form of a **corpus**, defined as a collection of texts that includes document-level variables specific to each text, as well as meta-data for documents and for the collection as a whole. quanteda includes tools to make it easy and fast to manipulate the texts in a corpus, by performing the most common natural language processing tasks simply and quickly, such as tokenizing, stemming, or forming ngrams. quanteda's functions for tokenizing texts and forming multiple tokenized documents into a document-feature matrix are both extremely fast and extremely simple to use. quanteda can segment texts easily by words, paragraphs, sentences, or even user-supplied delimiters and tags.

### 3.4.2 Building Spam Ham Word clouds

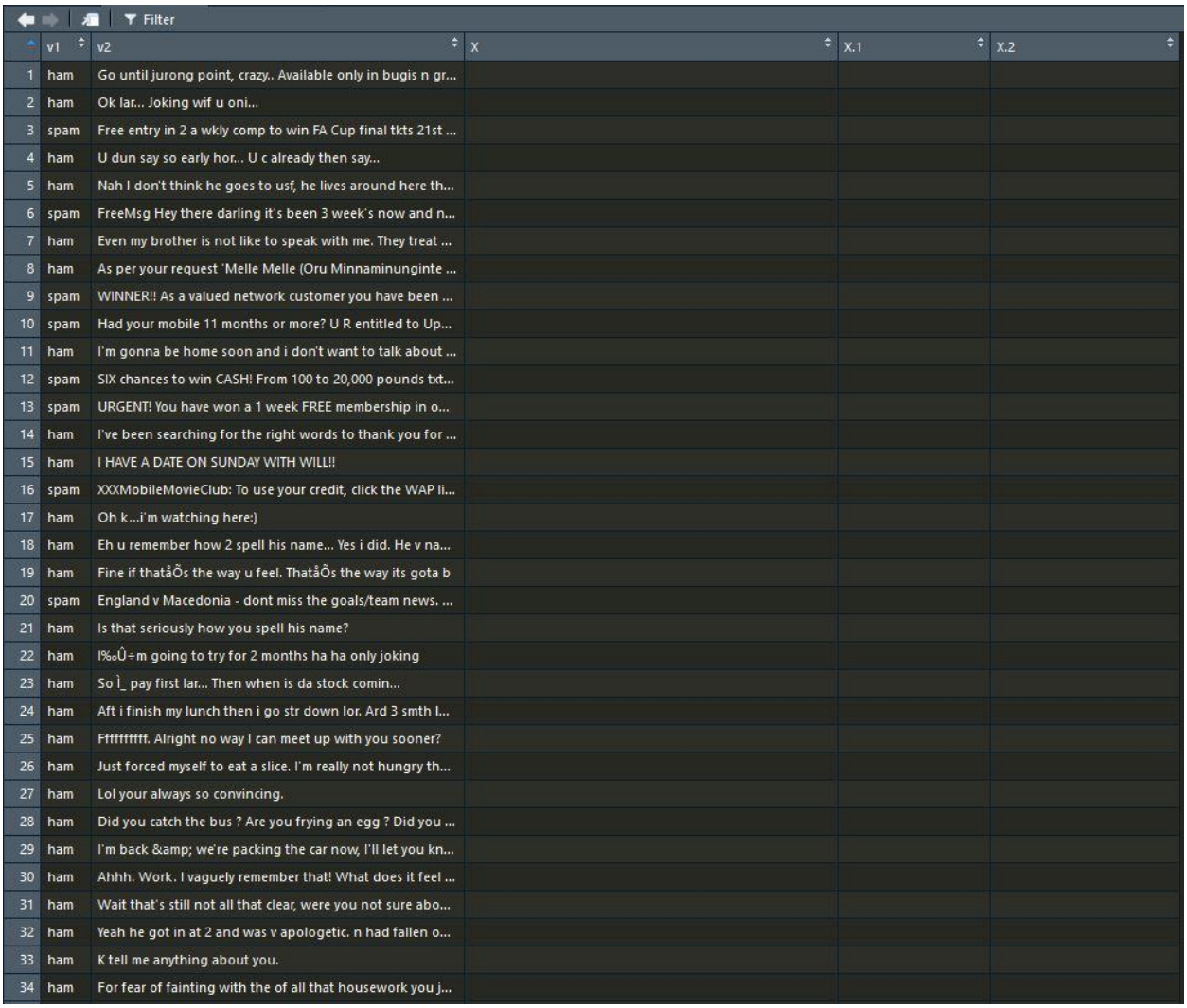
We'll use quanteda's corpus() command to construct a corpus from the Text field of our raw data. A corpus can be thought of as a master copy of our dataset from which we can pull subsets or observations as needed. After this I will attach the Label field as a document variable to the corpus using the docvars() command. We attach Label as a variable directly to our corpus so that we can associate SMS messages with their respective ham/spam label later in the analysis.

# Chapter 4

# Results and Discussion

## 4.1. Data Analysis

The R code has been written in such a way that whenever we enter a phrase, the data is searched for matching class.



Filter

| | v1 | v2 | X | X.1 | X.2 |
|---|---|---|---|---|---|
| 1 | ham | Go until jurong point, crazy.. Available only in bugis n gr... | | | |
| 2 | ham | Ok lar... Joking wif u oni... | | | |
| 3 | spam | Free entry in 2 a wkly comp to win FA Cup final tkts 21st ... | | | |
| 4 | ham | U dun say so early hor... U c already then say... | | | |
| 5 | ham | Nah I don't think he goes to usf, he lives around here th... | | | |
| 6 | spam | FreeMsg Hey there darling it's been 3 week's now and n... | | | |
| 7 | ham | Even my brother is not like to speak with me. They treat ... | | | |
| 8 | ham | As per your request 'Melle Melle (Oru Minnaminunginte ... | | | |
| 9 | spam | WINNER!! As a valued network customer you have been ... | | | |
| 10 | spam | Had your mobile 11 months or more? U R entitled to Up... | | | |
| 11 | ham | I'm gonna be home soon and i don't want to talk about ... | | | |
| 12 | spam | SIX chances to win CASH! From 100 to 20,000 pounds txt... | | | |
| 13 | spam | URGENT! You have won a 1 week FREE membership in o... | | | |
| 14 | ham | I've been searching for the right words to thank you for ... | | | |
| 15 | ham | I HAVE A DATE ON SUNDAY WITH WILL!! | | | |
| 16 | spam | XXXMobileMovieClub: To use your credit, click the WAP li... | | | |
| 17 | ham | Oh k...i'm watching here:) | | | |
| 18 | ham | Eh u remember how 2 spell his name... Yes i did. He v na... | | | |
| 19 | ham | Fine if thatåÕs the way u feel. ThatåÕs the way its gota b | | | |
| 20 | spam | England v Macedonia - dont miss the goals/team news. ... | | | |
| 21 | ham | Is that seriously how you spell his name? | | | |
| 22 | ham | I‰ÛÏm going to try for 2 months ha ha only joking | | | |
| 23 | ham | So Ì_ pay first lar... Then when is da stock comin... | | | |
| 24 | ham | Aft i finish my lunch then i go str down lor. Ard 3 smth l... | | | |
| 25 | ham | Fffffffff. Alright no way I can meet up with you sooner? | | | |
| 26 | ham | Just forced myself to eat a slice. I'm really not hungry th... | | | |
| 27 | ham | Lol your always so convincing. | | | |
| 28 | ham | Did you catch the bus ? Are you frying an egg ? Did you ... | | | |
| 29 | ham | I'm back & we're packing the car now, I'll let you kn... | | | |
| 30 | ham | Ahhh. Work. I vaguely remember that! What does it feel ... | | | |
| 31 | ham | Wait that's still not all that clear, were you not sure abo... | | | |
| 32 | ham | Yeah he got in at 2 and was v apologetic. n had fallen o... | | | |
| 33 | ham | K tell me anything about you. | | | |
| 34 | ham | For fear of fainting with the of all that housework you j... | | | |

**Fig. 4.1.1** Data stored in Comma Separated Values File

## 4.2. Graphs and Charts


**Fig. 4.2.1** Ham Word cloud


**Fig. 4.2.2** Spam Word cloud

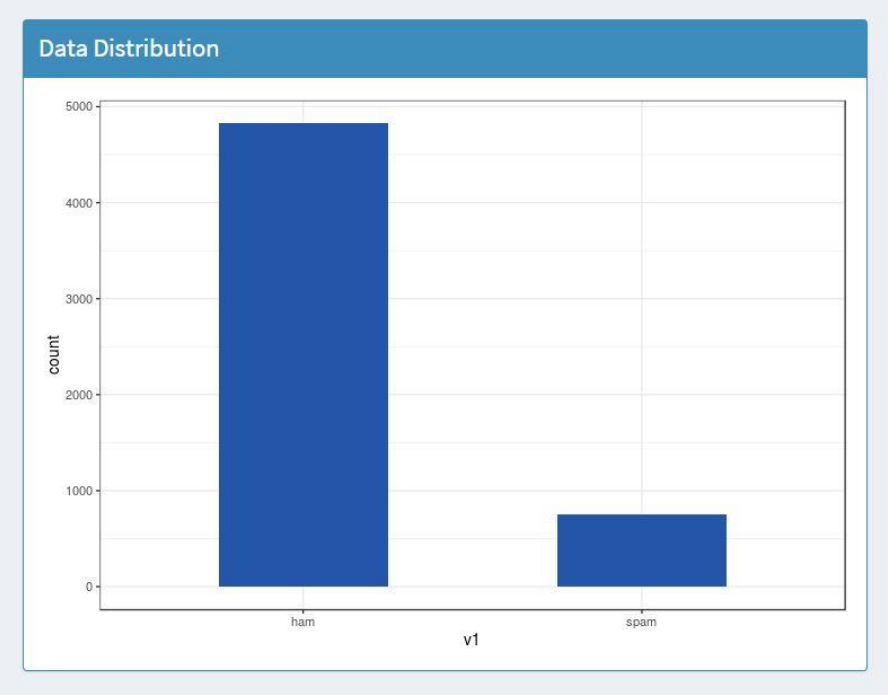



**Fig. 4.2.3** Bar Chart for Data Distribution

# Conclusion

Thus we learned about classification using Naïve Bayes. This algorithm constructs tables of probabilities that are used to estimate the likelihood that new examples belong to various classes. The probabilities are calculated using a formula known as Bayes' theorem, which specifies how dependent events are related. Although Bayes' theorem can be computationally expensive, a simplified version that makes so-called "naive" assumptions about the independence of features is capable of handling extremely large datasets.

The Naïve Bayes classifier is often used for text classification. To illustrate its effectiveness, we employed Naïve Bayes on a classification task involving spam SMS messages. Preparing the text data for analysis required the use of specialized R packages for text processing and visualization. Ultimately, the model was able to classify over 98 percent of all the SMS messages correctly as spam or ham.

# References

**[1]** Dea Delvia Arifin, Shaufia, Moch. Arif Bijaksana "**Enhancing spam detection on mobile phone Short Messaging Service (SMS) performance using FP-growth and Naïve Bayes Classifier** ", Institute of Electrical and Electronics Engineer , 10.1109/APWiMob.2016.7811442, 01-2017.

# Software Testing Report

## Introduction

***Document overview***

This document is the software test report of the testing phase of the Spam and Ham software development project. It contains the results of tests, which were executed during the testing phase. Detailed analysis of various test cases, report of those test cases and results of those test cases are documented in this report. Overall the techniques used for testing our software and how it is used to validate the application has also been documented. Overall this report is used to improve the overall data distribution of the application.

## Project References

***References***

| # | Document Identifier | Document Title |
|---|---|---|
| [D1] | 0-7695-2562-8 | Automating functional tests using Selenium, IEEE, A. Holmes, M. Kellog |
| [D2] | 2277 128X | A Study on Functioning of Selenium Automation Testing Structure, IJARCSSE, Jyoti Devi, Kirti Bhatia, Rohini Sharma. |

***Standard and Regulatory References***

| # | Document Identifier | Document Title |
|---|---|---|
| [D1] | 978-1-5090-2767-5 | Dea Delvia Arifin, Shaufia, Moch. Arif Bijaksana "Enhancing spam detection on mobile phone Short Messaging Service (SMS) performance using FP-growth and Naïve Bayes Classifier ", Institute of Electrical and Electronics Engineer , 10.1109/APWiMob.2016.7811442, 01-2017. |

# Overview of Tests Results

***Tests log***

The Spam and Ham application (version 1.0) was tested on the Selenium testing platform . The tests of the test phase (ref. software test plan) where executed. The testing log included tests for verification of different tests and then determining whether the execution of these tests was successful or not.

Further evaluation of this test cases were made to help improve the overall functionality of the application.

***Rationale for decision***

After executing a test, the decision is defined according to the following rules:

- **Passed:** The test sheet is set to "Passed" state when all steps are in "Passed" state. The real result is compliant to the expected result.
- **Failed:** The test sheet is set to " Failed " state when all steps of the test are set to "Failed" state or when the result of a step differs from the expected result.
- **Partially passed:** The test sheet is set to "Partially passed" state when at least one step of the test is set to " Partially passed " state or when the result of a step is partially compliant to the expected result. → Keep it or remove. Source of inconsistencies: criteria to set if result is Partially passed may be qualitative
- **NOT RUN:** Default state of a test sheet not yet executed.
- **NOT COMPLETED:** The test sheet is set to "Not Completed" state when at least one step of the test is set "Not Run" state.

***Overall assessment of tests***

- All tests with interfaces passed, graphical user interface is optimized for screens of the test platform
- All tests passed and is favorable for user acceptance.
- All test cases were executed successfully on the system.
- All test cases were executed automatically thereby conforming to automated testing.

***Impact of test environment***

The testing environment included Selenium IDE as the main resource. Selenium IDE helped in generating automated test cases for effective execution and automated testing.

**Selenium IDE** is an integrated development environment for Selenium scripts. It is implemented as a Chrome and Firefox extension, and allows you to record, edit, and debug tests.

Selenium IDE is not only a recording tool: **it is a complete IDE**. You can choose to use its recording capability, or you may edit your scripts by hand. With autocomplete support and the ability to move commands around quickly, Selenium IDE is the ideal environment for creating Selenium tests no matter what style of tests you prefer.

Features of Selenium IDE:

- Easy record and playback
- Intelligent field selection will use IDs, names, or XPath as needed
- Autocomplete for all common Selenium commands
- Walk through tests
- Debug and set breakpoints
- All in one project file, containing all test cases and suites

## Detailed Tests Results

For each executed test, this document contains:

- Test identification;
- Test title;
- Test decision;
- A comment containing additional information or problems encountered during execution and differences with the test procedure.

For the problems leading to a bug, the bug ID is reported in the result of the step where problem was encountered.

***Detailed test results***

**1. Test ID- SOHT01**

| **Test ID** | **Same ID as in test desc.** | **Comment** | **Decision** |
|---|---|---|---|
| Test description | SOHT01 | Verifying whether entered text is Ham | Passed |
| Verified Requirement | SRS-REQ-1 | Selenium IDE, working web application link | Passed |
| Initial conditions | 1. Working Internet Connection<br>2. Compatible Browser<br>3.Successful loading of website | Same as in test desc. | Passed |
| Tests inputs | A line of text | "how are you?" | Passed |
| Data collection actions | Automated | N/A | Passed |
| Tests outputs | This message is HAM | The message detected is a Ham | Passed |
| Assumptions and constraints | N/A | N/A | N/A |
| Expected results and criteria | The test passes all constraints | N/A | Passed |
| **Step number** | **Operator actions** | **Expected result and evaluation criteria** | **Result** |

| | | | |
|---|---|---|---|
| 1 | Start Selenium IDE | Selenium IDE is started | Passed |
| **2** | Establish connection to web app | Web application is opened | Passed |
| **3** | Run the .side file and evaluate the test case | Test case is being evaluated | Passed |

**2. Test ID- SOHT02**

| **Test ID** | **Same ID as in test desc.** | **Comment** | **Decision** |
|---|---|---|---|
| Test description | SOHT02 | Verifying whether entered text is Ham | Passed |
| Verified Requirement | SRS-REQ-2 | Same as in test desc. | Passed |
| Initial conditions | 1. Working Internet Connection<br>2. Compatible Browser<br>3.Successful loading of website | Same as in test desc. | Passed |
| Tests inputs | A line of text | Call me at 1234 to claim | Passed |
| Data collection actions | Automated | N/A | Passed |
| Tests outputs | This message is SPAM | The message detected is a SPAM | Passed |

| | | | |
|---|---|---|---|
| Assumptions and constraints | N/A | N/A | N/A |
| Expected results and criteria | The IDE and the web application are connected and the test is being evaluated | The test passes all constraints | Passed |
| **Test procedure** | | | |
| **Step number** | **Operator actions** | **Expected result and evaluation criteria** | **Result** |
| 1 | Start Selenium IDE | Selenium IDE is started | Passed |
| 2 | Establish connection to web app | Web application is opened | Passed |
| 3 | Run the .side file and evaluate the test case | Test case is being evaluated | Passed |

**3. Test ID- SOHT03**

| **Test ID** | **Same ID as in test desc.** | | **Decision** |
|---|---|---|---|
| Test description | SOHT03 | Verifying the title of the web application | Passed |
| Verified Requirement | SRS-REQ-001 | Same as in test desc. | Passed |
| Initial conditions | 3. Working Internet Connection | Same as in test desc. | Passed |

| | | | |
|---|---|---|---|
| | 4. Compatible Browser<br>3.Successful loading of website | | |
| Tests inputs | N/A | The title of the application | Passed |
| Data collection actions | Automated | N/A | Passed |
| Tests outputs | The title is "Spam And Ham" | The title is verified successfully | Passed |
| Assumptions and constraints | N/A | N/A | N/A |
| Expected results and criteria | Title should be verified as "Spam and Ham" | Title is verified as "Spam and Ham" | Passed |
| **Test procedure** | | | |
| **Step number** | **Operator actions** | **Expected result and evaluation criteria** | **Result** |
| 1 | Start Selenium IDE | Selenium IDE is started | Passed |
| 2 | Establish connection to web app | Web application is opened | Passed |
| 3 | Run the .side file and evaluate the test case | Test case is being evaluated | Passed |

**4. Test ID- SOHT04**

| **Test ID** | **Same ID as in test desc.** | **Comment** | **Decision** |
|---|---|---|---|
| Test description | SAHT04 | Connection loss after 1 min is acceptable for this test phase. | Passed |
| Verified Requirement | SRS-REQ-001 | Same as in test desc. | Passed |
| Initial conditions | Application should start successfully. | Application should start successfully | Passed |
| Tests inputs | N/A | N/A | N/A |
| Data collection actions | N/A | N/A | N/A |
| Tests outputs | Connection failed | The connection to the application is faiiled | Failed |
| Assumptions and constraints | N/A | N/A | N/A |
| Expected results and criteria | CONNECTION IS ESTABLISHED | N/A | Failed |
| **Test procedure** | | | |
| **Step number** | **Operator actions** | **Expected result and evaluation criteria** | **Result** |
| 1 | Start Selenium IDE | Selenium IDE is started | Failed |
| 2 | Establish connection to web app | Web application is opened | Failed |

| | | | |
|---|---|---|---|
| 3 | Run the .side file and evaluate the test case | Test case is being evaluated | Failed |

## Execution of the test cases

The following section contains the analysis of the test cases documented above in the table along with screenshots of actual execution.

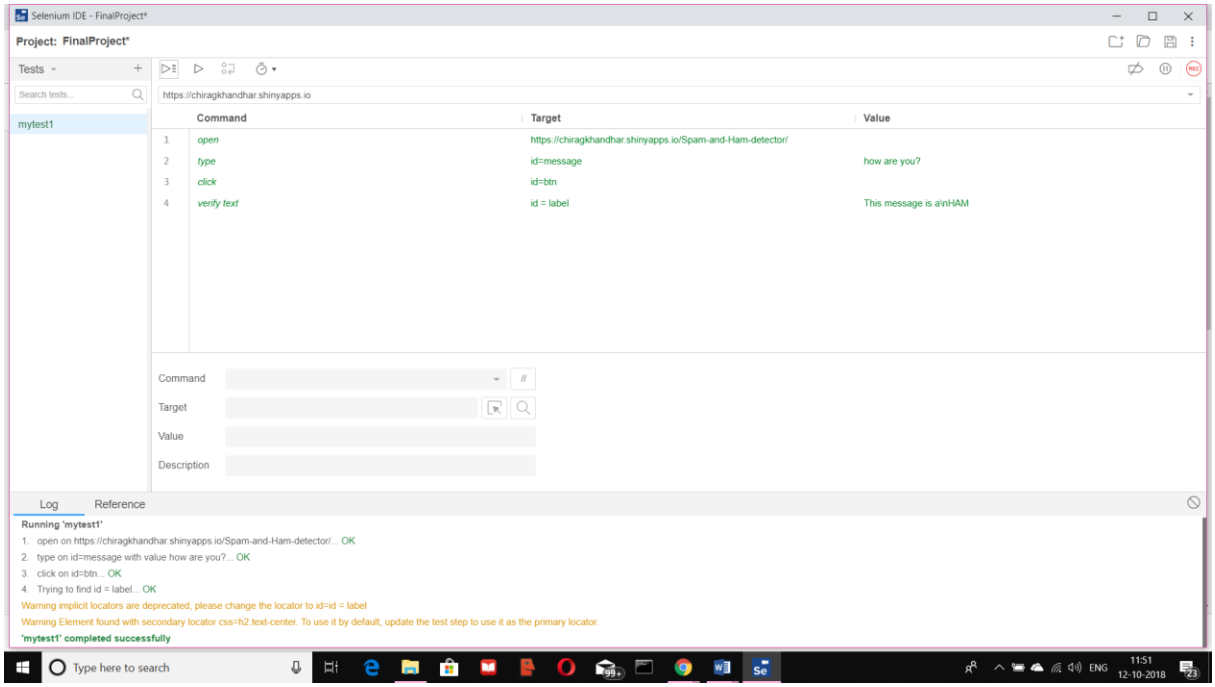



Fig 1: Test case ID(SOHT01)

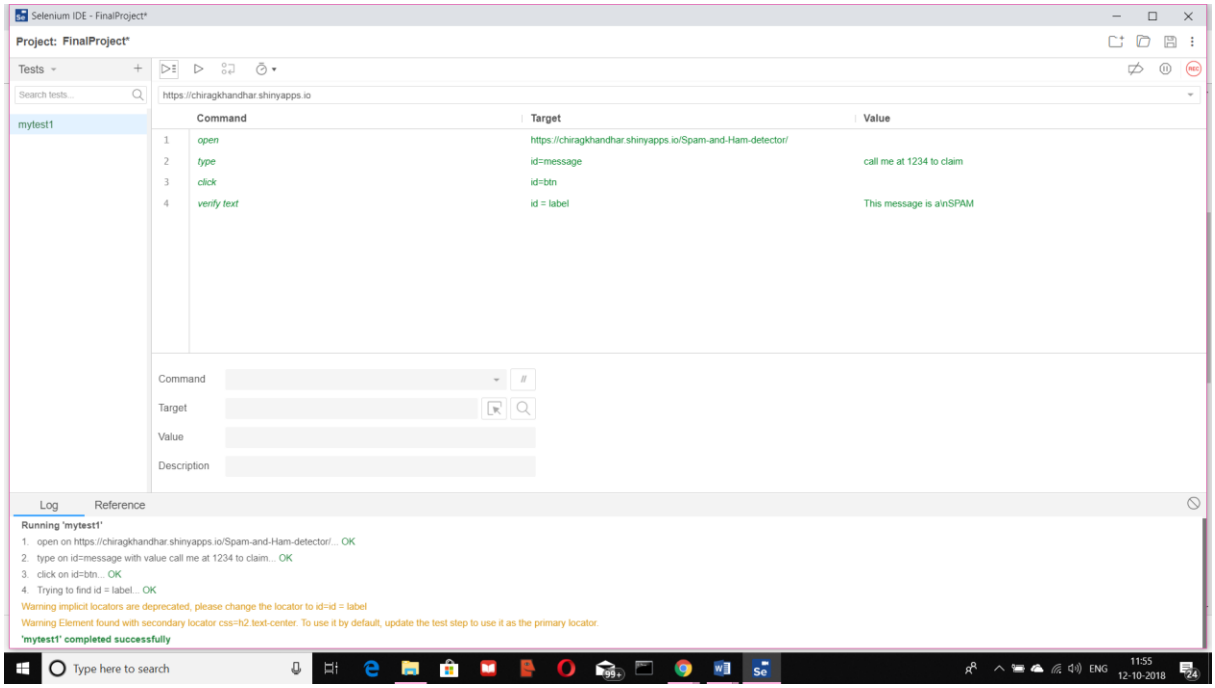



Fig 2: Test case ID(SOHT02)

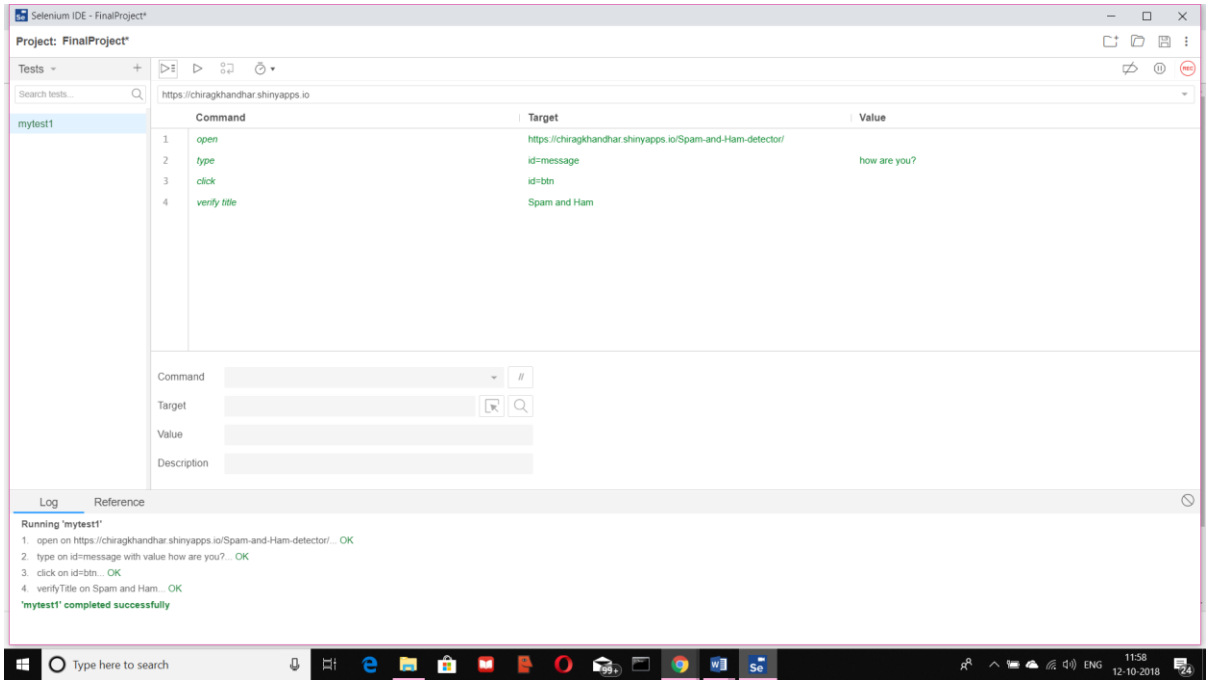



Fig 3: Test case ID(SOHT03)