BATCH NO:18A10

SPAM REVIEW DETECTION USING LINGUISTIC AND BEHAVIORAL METHODS

Project report submitted in partial fulfilment of the academic requirement for the award of the degree of B. Tech in Information Technology.

By

G.Chandana - 18251A1210

J.Sriharshini - 18251A1242

M.Deekshitha - 18251A1252

G. Narayanamma Institute of Technology and Science
(for Women)
(AUTONOMOUS)
Approved by AICTE, New Delhi & Affiliated to JNTUH, Hyderabad
Accredited by NBA & NAAC Shaikpet, Hyderabad – 500104, TS.
Sept 2021 - June 2022

G. NARAYANAMMA INSTITUTE OF TECHNOLOGY & SCIENCE

(AUTONOMOUS)

(For Women)

(Sponsored by G. Pulla Reddy Charities Trust, Hyderabad)
Accredited by NBA & NAAC, Approved by AICTE & Affiliated to JNTUH
An ISO 9001 : 2015 Certified Institution

CERTIFICATE

This is to certify that the project report entitled **Spam Review Detection Using Linguistic and Behavioral Methods** is bonafide work done by

G.Chandana - 18251A1210

J.Sriharshini - 18251A1242

M.Deekshitha - 18251A1252

Under the guidance of **Mrs. Ch. Sravanthi-Asst.prof**, during **Sept 2021 to June 2022**, in partial fulfillment for the award of degree in B.Tech in Information Technology, from G. Narayanamma Institute of Technology and Science.

Mrs. Ch. Sravanthi-Asst. Prof	Dr. I. Ravi Prakash Reddy
Internal Guide	**Head of the Department**

ACKNOWLEDGEMENT

We would like to express our sincere thanks to **Dr K. Ramesh Reddy, Principal**, GNITS, for providing the good working environment and facilities in the college. A sincere gratitude to **Dr I. Ravi Prakash Reddy**, **Professor and HOD**, Dept. of Information Technology, GNITS for encouraging us to take up the project and all the timely support during the time of mini-project. We respectfully acknowledge
Ms. Ch. Sravanthi-Asst. Prof, Department of Information Technology for her immense support and guidance during the whole time of the project. Finally, we would also like to thank all the faculty and staff of IT Department who helped me directly or indirectly, parents and friends for their cooperation in completing the project work.

ABSTRACT

Online reviews regarding different products or services have become the main source to determine public opinions. Consequently, manufacturers and sellers are extremely concerned with customer reviews as these have a direct impact on their businesses. Unfortunately, to gain profits or fame, spam reviews are written to promote or demote targeted products or services. This practice is known as review spamming. In recent years, the spam review detection problem has gained much attention from communities and researchers, but still there is a need to perform experiments on real-world large-scale review datasets. This can help to analyze the impact of widespread opinion spam in online reviews. In this work, two different spam review detection methods have been proposed: (1) Spam Review Detection using Behavioral Method (SRD-BM) utilizes thirteen different spammer's behavioral features to calculate the review spam score which is then used to identify spammers and spam reviews, and (2) Spam Review Detection using Linguistic Method (SRD-LM) works on the content of the reviews and utilizes transformation, feature selection and classification to identify the spam reviews. To the best of our knowledge, this is the first study of its kind which uses a large-scale review dataset to analyze different spammers' behavioral features and linguistic method utilizing different available classifiers.

LIST OF FIGURES

LIST OF TABLES

TABLE OF CONTENTS

1.INTRODUCTION

1.1. General/Domain Description

Review spam is usually related to email and web spam. The web spam is used to attract people by manipulating the content of the page so that the web page will be ranked highly by the search engines. Email spam is mainly used for advertising purposes. However, spam reviews are different in a sense as these give the wrong opinion about a product/ service and it is very difficult to detect spam reviews manually. Therefore, existing web spam or email spam detection techniques are not suitable for spam review detection. Spam review detection is a challenging task as no one can detect a review as spam by simply reading its text. Review websites are usually open to public reviews. Therefore, any user can act as spammer to write spam reviews about any product and/or service. Spam reviews appear as legitimate until different spammer behavioral features and/or the review text is analysed to identify the spam reviews. Based on these perspectives, existing approaches of Spam Review Detection (SRD) utilizes spammer behavioral features or linguistic features for the detection of spammers and spam reviews respectively. The linguistic feature considers review text to identify the reviews as spam or not spam; whereas behavioral features reflect the behavior of reviewer in terms of time stamp of review, review rating, user profile, etc.

The aim of this work is to develop an SRD model adapting a vast set of behavioral and linguistic features on large-scale real-world dataset.The User Login to the website and uploads the dataset to be trained. Any dataset can be uploaded for training containing the same schema. Later, the input is given as the review that he/she want to check whether spam or not. The result is displayed along with the spam score.

1.2. Objective and Scope of the Project

In this work, two different spam review detection methods have been proposed: (1) Spam Review Detection using Behavioural Method (SRD-BM) utilizes thirteen different spammer's behavioural features to calculate the review spam score which is then used to identify spammers and spam reviews, and (2) Spam Review Detection using Linguistic Method (SRD-LM) works on the content of the reviews and utilizes transformation, feature selection and classification to identify the spam reviews.

The aim of this work is to develop an SRD model adapting a vast set of behavioural and linguistic features on large-scale real-world dataset.

To develop a game that recognizes the static hand gestures which are recorded in stable lighting and simple background conditions.

1.3. Project Definition

Consequently, manufacturers and sellers are extremely concerned with customer reviews as these have a direct impact on their businesses. Unfortunately, to gain profits or fame, spam reviews are written to promote or demote targeted products or services. This practice is known as review spamming. In recent years, the spam review detection problem has gained much attention from communities and researchers, but still there is a need to perform experiments on real-world large-scale review datasets. This can help to analyze the impact of widespread opinion spam in online reviews. In this work, two different spam review detection methods have been proposed: (1) Spam Review Detection using Behavioral Method (SRD-BM) utilizes thirteen different spammer's behavioral features to calculate the review spam score which is then used to identify spammers and spam reviews, and (2) Spam Review Detection using Linguistic Method (SRD-LM) works on the content of the reviews and utilizes transformation, feature selection and classification to identify the spam reviews.

2.LITERATURE SURVEY

2.1. Existing System

Spam review detection using the spammer behavioural method finds the unusual patterns and relationships between different spammers. Only a few studies have explored spam review detection using the spammer behavioural method to date. They are:

[1]Mukherjee developed a spam review detection method using a clustering technique by modelling the spam city of the reviewer to identify spammer and non-spammer clusters.[1]**Heydari** have proposed a model incorporating only the time series feature of the reviewer on an amazon real dataset.[1]**Kumar** has proposed a hierarchical supervised learning method. This method analysed reviewers behavioural features and their interaction using multivariate distribution.[1]**Zang** recommended a supervised model based on reviewer features to identify spam reviews.[1]**Li** used the feature based sparse additive generative model and the svm classifier to discover the general rule for spam review detection.

[5] have proposed a hierarchical supervised-learning method. This method analysed reviewer’s behavioural features and their interactions using multivariate distribution. Zhang et al.

[6] **Ahmed and Danti** used various rule-based machine learning algorithms. Moreover, the authors compared the effectiveness of the proposed method through a ten-fold cross-validation training model for sentiment classification.

2.2. Issues in Existing System

Based on the review of spammer behavioural models ,it has been observed that most of the existing studies have only utilized time series based spammer behavioural feature. It is analysed that utilizing rich set of behavioural features can help in improving the accuracy of spammer identification. Therefore, the proposed behavioural framework utilizes thirteen different spammer behavioural features to calculate spam score in spam review identification.

Based on the literature review it has been observed that most of the existing studies did not incorporate a number of important linguistic features while designing linguistic feature-based SRD models and utilized only one classifier to train their proposed models.

2.3. Proposed System

Proposed methods utilized real world large scale amazon review dataset. Proposed SRD-BM which incorporated thirteen different behavioural features to identify spammers and spam reviews. Proposed SRD-LM which utilized linguistic features and classifiers to identify spam reviews.Compared and analysed the accuracy of proposed SRD-BM and SRD-LM.

2.4 Advantages of proposed system

The linguistic feature considers review text to identify the review as spam or not spam :where as behavioural features reflect the behaviour of reviewer in terms of time stamp of the review rating user profile etc. Most of the existing systems have only utilized the uni-gram linguistic approach to classify reviews. Usually the uni-gram approach produces good results but fails in some cases.

Based on the literature review, it has been observed that most of the existing studies did not incorporate a number of important linguistic features while designing the linguistic feature based SRD models and utilized only one classifier to train their proposed models. The current study therefore extends the SRD domain to design a linguistic model utilizing several features, including stemming and N-gram techniques. These features have significantly improved the accuracy of the proposed model in spam review identification. Moreover the proposed model utilizes and compares the accuracy of four different classifiers, including Naive Bayes(NB),Logistic Regression(LR),Support Vector Machine(SVM),and Random Forest Classifier(RF) to further improve the accurate prediction of spam review.

3.REQUIREMENT SPECIFICATION

3.1. Introduction

The requirements are grouped by their stakeholders, and functional and non-functional requirements are separated.

3.2. System environment

Spam Review Detection Model environment consists of 1 active Actor and Co-operating system. The user accesses the platform using the internet. Any active user can login through the system using his saved credentials otherwise by registering himself to the system.

3.3. Functional requirements specification

3.3.1 Use case descriptions

This section outlines the brief description of the use cases for the user.

USE CASE: LOGIN/REGISTER

Use Case Name	Login/Register
Trigger	The User enters the credentials to login/register
Precondition	The User should initially have access to the website
Basic Path	User login through the website by providing his/her credentials.
Alternative Paths	The user should create an account if doesn't exists
Postcondition	The User is directed to the input grid window
Exception Paths	The User may abandon the operation at any time.
Other	None

Table 3.3.1.1 – Use case Login/Register

Diagram:

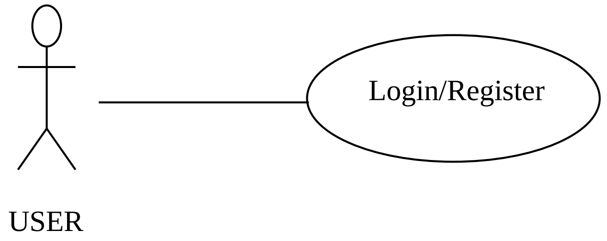


Fig. 3.3.1.1 – Login /Register

Brief Description:

The user should provide the accurate credentials to login. If the account does not exist the user should register.

Initial Step-By-Step Description:

Before, this use case is initiated, the user has already access to internet and opened the website

1. The user clicks on Login button.
2. The user enters the username in the space provided.
3. The user enters the respective password.
4. If the user forgets his/her password, they click on forgot password option.
5. The user clicks on submit button to login successfully.

USE CASE: INPUT REVIEW

Use Case Name	Input Review
Trigger	The user should provide the review and click submit button
Precondition	The user should login with correct credentials and have access to the online website
Basic Path	User enter the correct credentials and write the input review in the grid
Alternative Paths	User may logout if not
Postcondition	The system displays the review is spam or not and user can also get the spam score by clicking the spam score button.
Exception Paths	The User may abandon the operation at any time.
Other	None

Table 3.3.1.2 – Use case Input Review

Diagram:

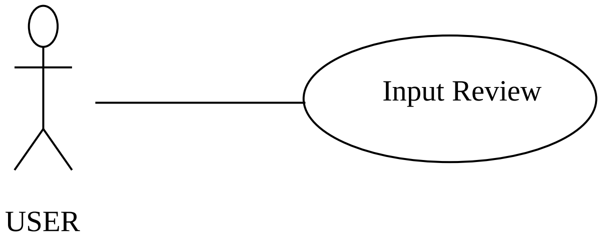


Fig. 3.3.1.2 – Use case Input Review

Brief Description:

The user has access to the website and can provide the input review to check the review is spam or not.

Initial Step-By-Step Description:

Before this use case is initiated the user is already login to the website.

1. The user sees the space provided to write the review.
2. The user enters his/her review in the input grid.
3. The user clicks on spam result button.

USE CASE: VIEW SPAM RESULT

Use Case Name	View Spam Result
Trigger	Submitting the input review by the user in the provided input grid
Precondition	The user initially should the review in the input grid
Basic Path	After submitting the system displays the spam result
Alternative Paths	The user can also get the spam score of the provided input review
Postcondition	The user can be redirected to the input grid window or can view the spam score.
Exception Paths	The attempt may be abandoned at any time.
Other	None

Table 3.3.1.3 – View Spam Result

Diagram:

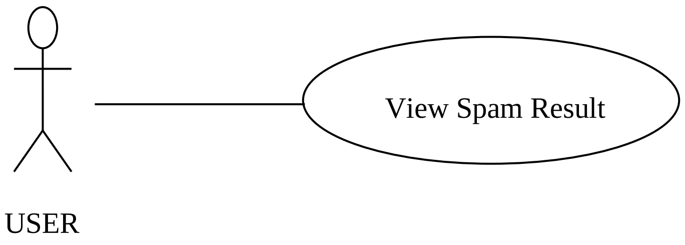


Fig. 3.3.1.3 – Use case View Spam Result

Brief Description:

The spam results can be checked by user after entering the correct input and clicking the spam result button.

Initial Step-By-Step Description:

Before this use case in initiated the user is already provided the review in the input grid.

1. The user can view the result of the provided review.
2. The user understands if the provided review is spam or non-spam review.

USE CASE: VIEW SPAM SCORE

Use Case Name	View Spam Score
Trigger	The user should click the spam score button
Precondition	The user should provide the input review
Basic Path	The user gets to view the pie-chart after submitting
Alternative Paths	The activity is completed ,user can go to the home page
Postcondition	User can get the spam score later redirected to the input grid page or user may logout.
Exception Paths	The attempt may be abandoned at any time.
Other	None

Table 3.3.1.4 – Use case View Spam Score

Diagram:

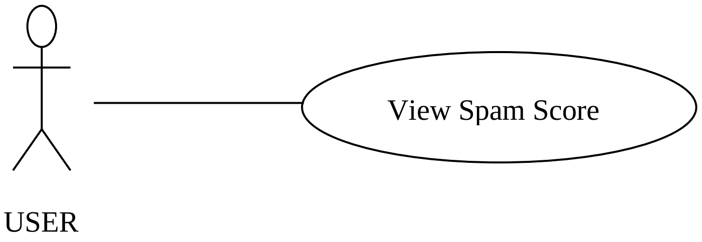


Fig. 3.3.1.4 – Use case View Spam Score

Brief Description:

The user gets the spam score in the form of pie-cart by clicking the spam score button.

Initial Step-By-Step Description:

Before this use case in initiated the user is already provided the review in the input grid.

1. The user clicks on view spam score button.
2. Pie-chart is displayed, which indicates the percentage of spam the review is.
3. User can logout of the website.

USE CASE: PROVIDE SPAM RESULT

Use Case Name	Provide Spam Result
Trigger	The admin receives the data from the user & provide the spam result
Precondition	The user initially should provide the review in the input grid
Basic Path	The admin sends the input review to the algorithm
Alternative Paths	The admin can also provide the spam score
Postcondition	The admin waits for another new input review to provide the spam result.
Exception Paths	The attempt may be abandoned at any time.
Other	None

Table 3.3.1.5 – Provide Spam Result

Diagram:

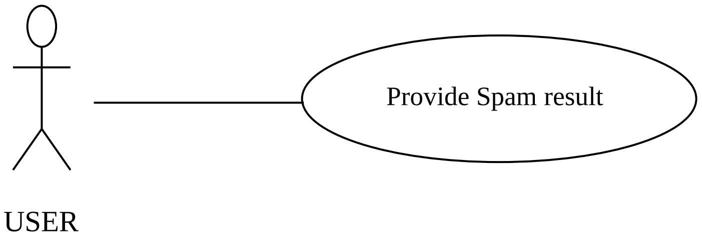


Fig. 3.3.1.5 – Use case Provide Spam Result

Brief Description:

After providing the input review by user the system provides the spam result.

Initial Step-By-Step Description:

Before this use case in initiated the admin receives the review from the input grid provided by the user

1. The admin provides the result of the provided review.

USE CASE: PROVIDE SPAM SCORE

Use Case Name	Provide Spam Score
Trigger	The admin receives the data from the user & provide the spam score
Precondition	The user initially should provide the review in the input grid
Basic Path	The admin sends the input review to the algorithm
Alternative Paths	The process can be terminated if the input is incorrect
Postcondition	The admin waits for another new input review to provide the spam score
Exception Paths	The attempt may be abandoned at any time.
Other	None

Table 3.3.1.6 – Use case Provide Spam Score

Diagram:

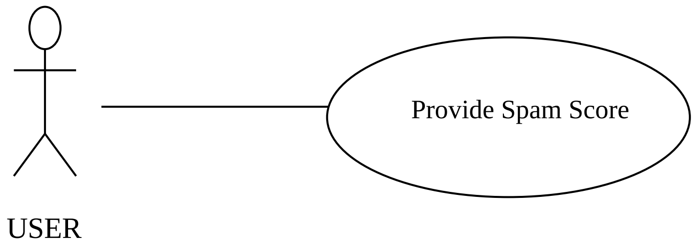


Fig. 3.3.1.6 – Use case Provide Spam Score

Brief Description:

The system provides the spam score after the user clicks the spam score button.

Initial Step-By-Step Description:

Before this use case in initiated the admin is already provided with the review from the input grid.

1. Pie-chart is displayed, which indicates the percentage of spam the review is.
2. Admin can terminate the process if the input is not provided by the user.

3.3.2 List of Functional Requirements

STAKEHOLDERS

Table:3.3.2.1

U	User
A	Admin

Functional Requirements of User

FR1	The system must enable user to register themselves by providing their personal details.
FR2	The system must allow users to login through their username and password after registering.
FR3	The system must allow users to reset their profile.
FR4	The system must provide the spam score to the user
FR5	The system must show whether the review is spam or not

Functional Requirements of Admin

FR6	The system must allow Admin to login with valid credentials.
FR7	The System must allow Admin to access the database of the application.
FR8	The system must allow Admin to add/remove/block users profile.
FR9	The system must provide admin all control over credentials details of users.

3.4 Non-Functional Requirements for User and Admin

Table:3.4

NF1	**Usability** The system is designed with completely automated process hence there is no or less user intervention.

NF2	**Reliability** The system is more reliable because of the qualities that are inherited from the chosen platform Anaconda. The code built by using python is more reliable.
NF3	**Performance** This system is developing in the high level languages and using the advanced front-end and back-end technologies it will give response to the end user on client system with in very less time.
NF4	**Supportability** The system is designed to be the cross platform supportable.
NF5	**Security** The System shall provide security to user's data.

3.5. H/W & S/W Requirements

Software Requirements

Operating System	: Windows 11
Web Framework	: Flask
Frontend	: Java Servlet Pages(JSP),HTML,CSS, Java Script
Database	: My SQL

Hardware Requirements

Processor	: Intel Core 5 (I5)
RAM	: 8 GB
Monitor	: 14*11 inches
Memory	: 1 TB
Drive Type	: Solid State Drive(SSD)

4. DESIGN SPECIFICATION

4.1 Overall Use Case Diagram

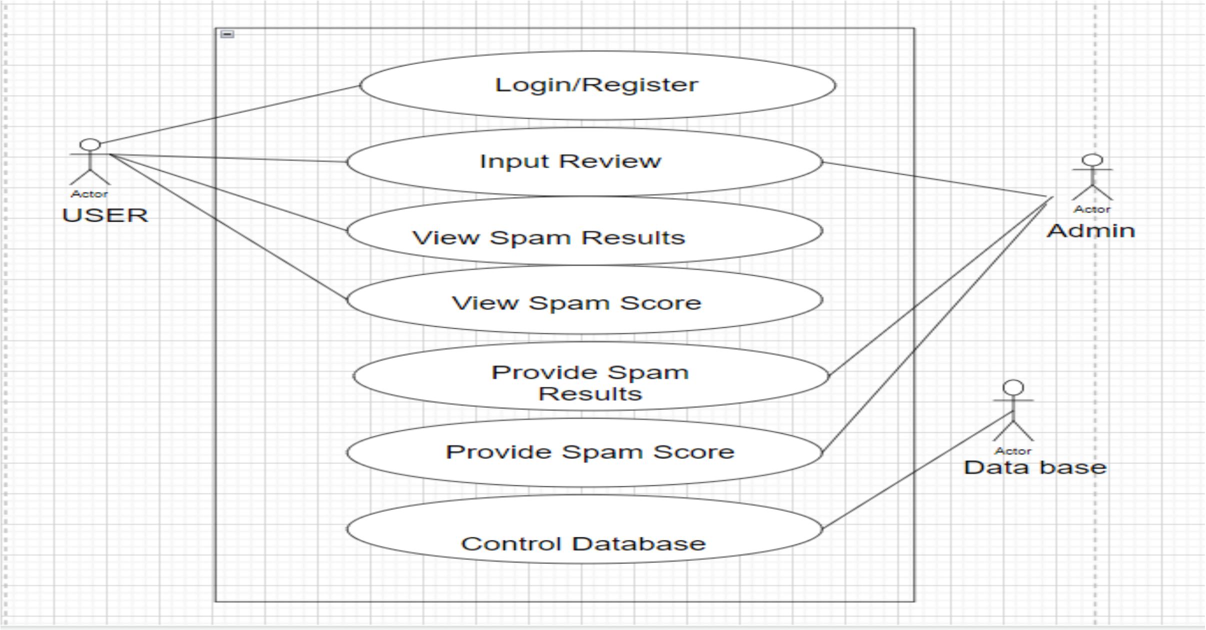


fig 4.1 – Use case Diagram

4.2 Database Description

Our dataset is of size 40,433. Data set consists of 4 attributes namely Category, Rating, Label, Review Text.

Attribute description:

1. **Category:** Various types of products are categorized into the following types .They are

- Clothing, shoes and jewels,
- Toys and games
- Kindle store
- Books
- Pet supplies
- Movies and TV

- Sports and outdoors
- Home and kitchen
- Electronics
- Tools and Home improvements

2. **Rating:** It takes a numeric value ranging between 1 to 5.

3. **Label:** Whole dataset is divided into two classes labelling CG(computer generated fake review), OR(Original Review)

4. **Text:** Review text in the dataset.

4.3 Class Diagram

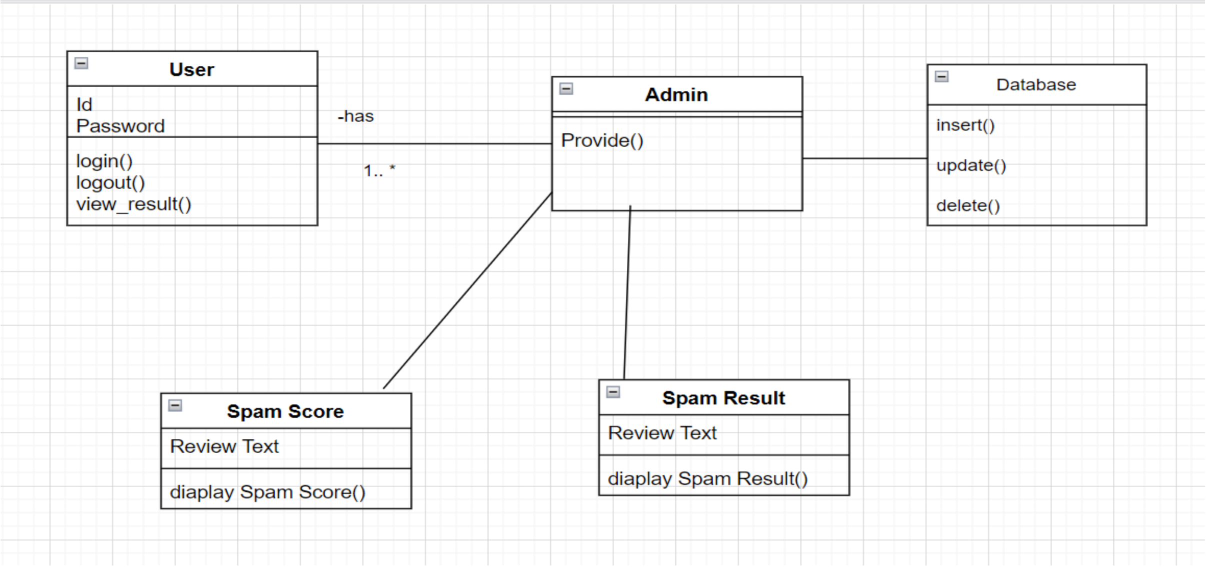


fig 4.3 - Class Diagram

4.4 Activity Diagram

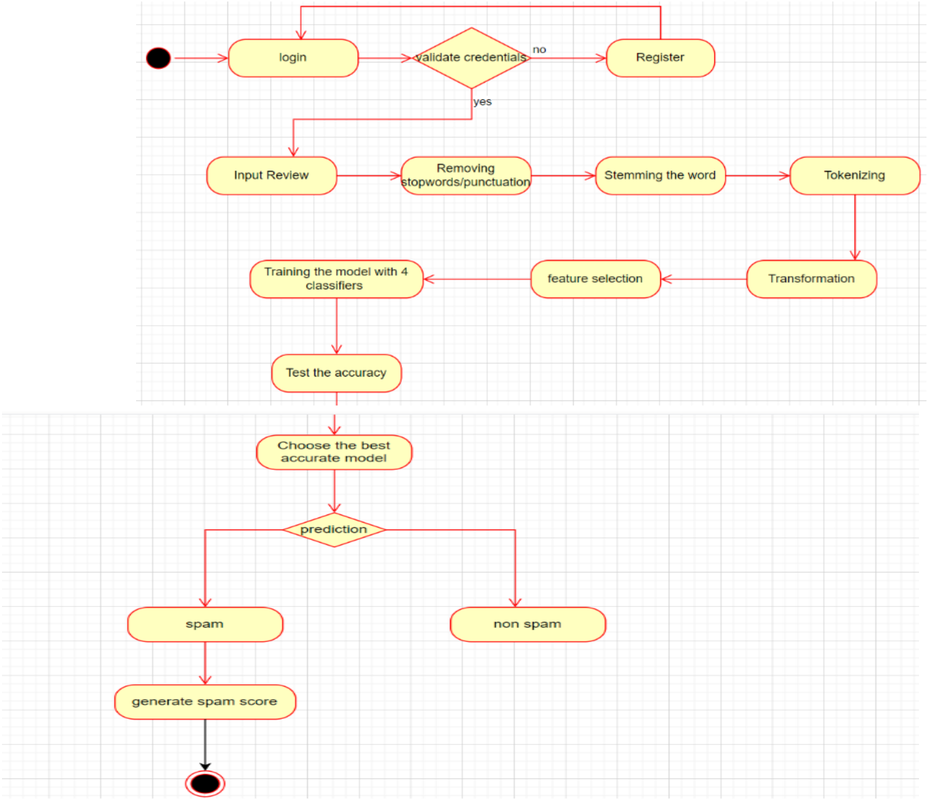


fig 4.4 - Activity Diagram

4.5 Sequence Diagram

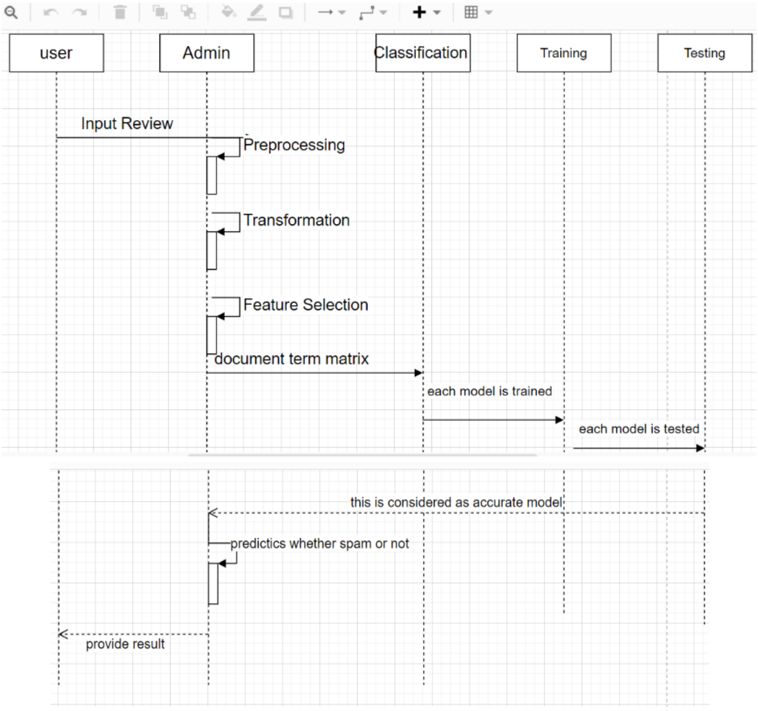


fig 4.5.1 - Sequence Diagram for Linguistic Model

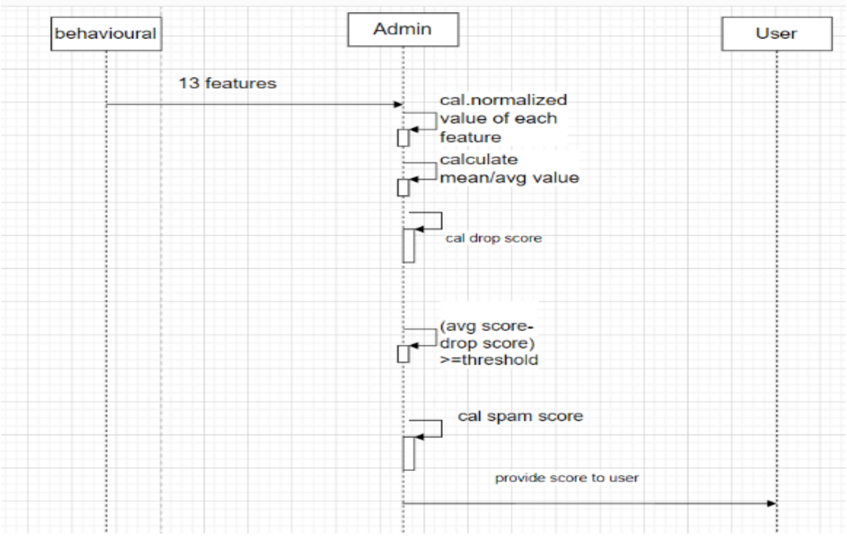


fig 4.5.2 - Sequence Diagram for Behavioral Model

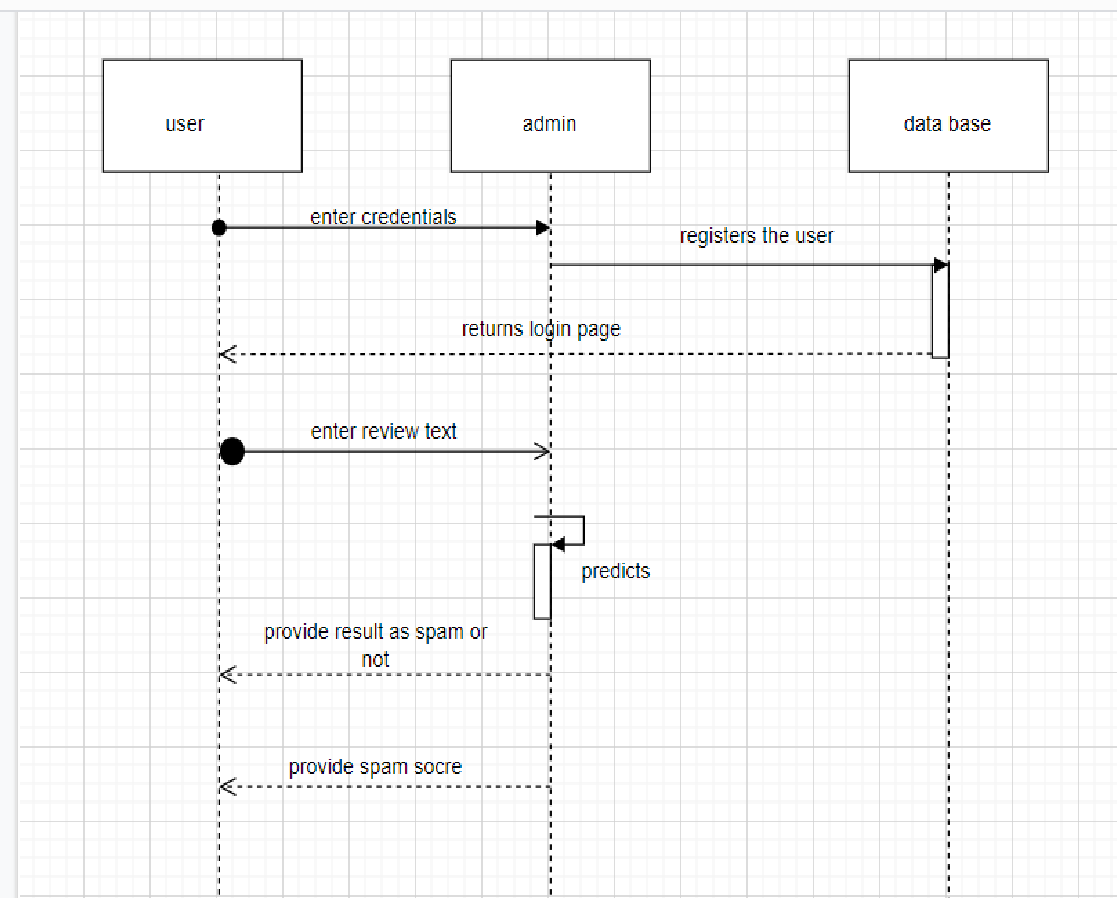


Fig 4.5.3 – sequence Diagram for website

4.6 Collaboration Diagram

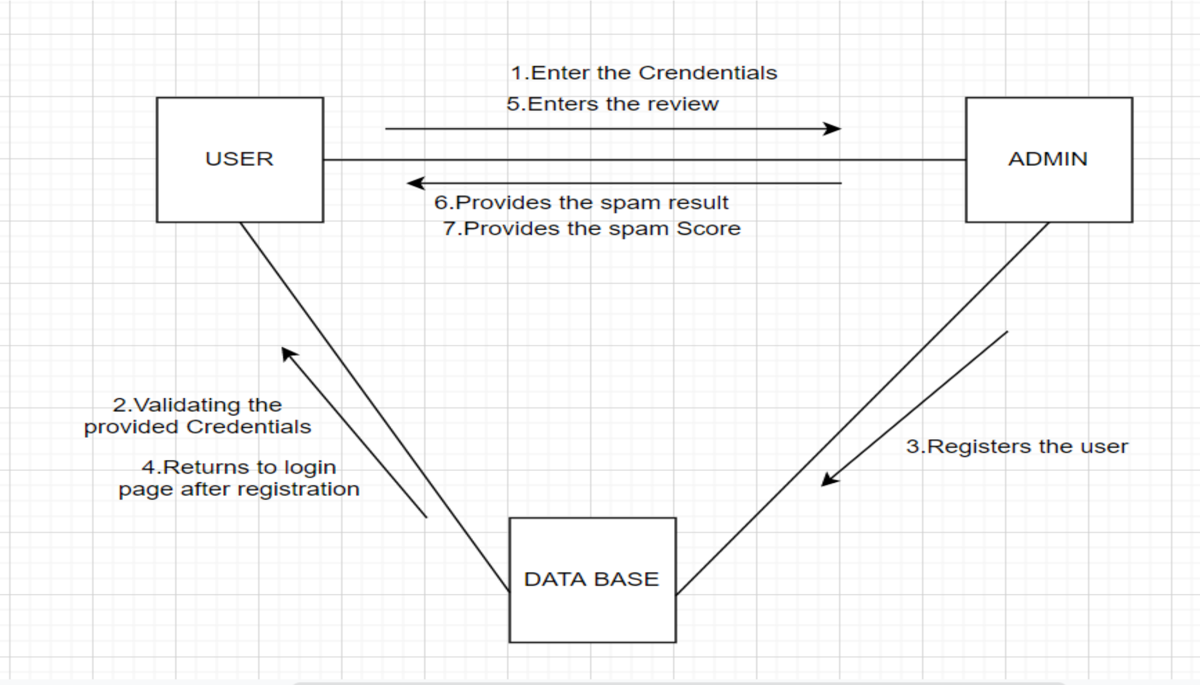

1.Enter the Crendentials
5.Enters the review
USER
ADMIN
6.Provides the spam result
7.Provides the spam Score
2.Validating the provided Credentials
4.Returns to login page after registration
3.Registers the user
DATA BASE

fig 4.6 - Collaboration Diagram

5.IMPLEMENTATION

5.1.Methodology:

DEVELOPMENT TOOLS

5.1.1 **Python:**

Python is a high-level, interpreted, interactive and object-oriented scripting language. Python is designed to be highly readable. It uses English keywords frequently where as other languages use punctuation, and it has fewer syntactical constructions than other languages.

- **Python is Interpreted** − Python is processed at runtime by the interpreter. You do not need to compile your program before executing it. This is similar to PERL and PHP.
- **Python is Interactive** − You can actually sit at a Python prompt and interact with the interpreter directly to write your programs.
- **Python is Object-Oriented** − Python supports Object-Oriented style or technique of programming that encapsulates code within objects.
- **Python is a Beginner's Language** − Python is a great language for the beginner-level programmers and supports the development of a wide range of applications from simple text processing to WWW browsers to games.

5.1.2Getting Python

The most up-to-date and current source code, binaries, documentation, news, etc., is available on the official website of Python https://www.python.org.

Windows Installation

Here are the steps to install Python on Windows machine.

- Open a Web browser and go to https://www.python.org/downloads/.
- Follow the link for the Windows installer python-XYZ.msifile where XYZ is the version you need to install.
- To use this installer python-XYZ.msi, the Windows system must support Microsoft Installer 2.0. Save the installer file to your local machine and then run it to find out if your machine supports MSI.

- Run the downloaded file. This brings up the Python install wizard, which is really easy to use. Just accept the default settings, wait until the install is finished, and you are done.

The Python language has many similarities to Perl, C, and Java. However, there are some definite differences between the languages.

5.1.3.Flask Framework:

Flask is a web application framework written in Python. Armin Ronacher, who leads an international group of Python enthusiasts named Pocco, develops it. Flask is based on Werkzeug WSGI toolkit and Jinja2 template engine. Both are Pocco projects.

Http protocol is the foundation of data communication in world wide web. Different methods of data retrieval from specified URL are defined in this protocol.

The following table summarizes different http methods −

Sr.No	Methods & Description
1	**GET** Sends data in unencrypted form to the server. Most common method.
2	**HEAD** Same as GET, but without response body
3	**POST** Used to send HTML form data to server. Data received by POST method is not cached by server.

4	**PUT** Replaces all current representations of the target resource with the uploaded content.
5	**DELETE** Removes all current representations of the target resource given by a URL

Table 5.1.3 Methods & Description

By default, the Flask route responds to the **GET** requests. However, this preference can be altered by providing methods argument to **route()** decorator.

In order to demonstrate the use of **POST** method in URL routing, first let us create an HTML form and use the **POST** method to send form data to a URL.

Save the following script as login.html

```
<html>
<body>
<formaction="http://localhost:5000/login"method="post">
<p>Enter Name:</p>
<p><inputtype="text"name="nm"/></p>
<p><inputtype="submit"value="submit"/></p>
</form>
</body>
</html>
```

Now enter the following script in Python shell.

```
from flask importFlask, redirect,url_for, request
app=Flask(__name__)
@app.route('/success/<name>')
def success(name):
return'welcome %s'% name
@app.route('/login',methods=['POST','GET'])
def login():
ifrequest.method=='POST':
user=request.form['nm']
return redirect(url_for('success',name= user))
else:
user=request.args.get('nm')
return redirect(url_for('success',name= user))
```

```
if __name__ =='__main__':
app.run(debug =True)
```

After the development server starts running, open **login.html** in the browser, enter name in the text field and click **Submit**.

Form data is POSTed to the URL in action clause of form tag.

http://localhost/login is mapped to the **login()** function. Since the server has received data by **POST** method, value of ‘nm’ parameter obtained from the form data is obtained by −

```
user = request.form['nm']
```

It is passed to **‘/success’** URL as variable part. The browser displays a **welcome** message in the window.

Change the method parameter to **‘GET’** in **login.html** and open it again in the browser. The data received on server is by the **GET** method. The value of ‘nm’ parameter is now obtained by −

```
User = request.args.get(‘nm’)
```

Here, **args** is dictionary object containing a list of pairs of form parameter and its corresponding value. The value corresponding to ‘nm’ parameter is passed on to ‘/success’ URL as before.

5.1.4.Python Install

Many PCs and Macs will have python already installed.

To check if you have python installed on a Windows PC, search in the start bar for Python or run the following on the Command Line (cmd.exe):

```
C:\Users\Your Name>python --version
```

To check if you have python installed on a Linux or Mac, then on linux open the command line or on Mac open the Terminal and type:

```
python --version
```

If you find that you do not have python installed on your computer, then you can download it for free from the following website: https://www.python.org/

Python Quickstart

Python is an interpreted programming language, this means that as a developer you write Python (.py) files in a text editor and then put those files into the python interpreter to be executed.

The way to run a python file is like this on the command line:

```
C:\Users\Your Name>python helloworld.py
```

Where "helloworld.py" is the name of your python file.

Let's write our first Python file, called helloworld.py, which can be done in any text editor.

helloworld.py

```
print("Hello, World!")
```

Simple as that. Save your file. Open your command line, navigate to the directory where you saved your file, and run:

```
C:\Users\Your Name>python helloworld.py
```

The output should read:

```
Hello, World!
```

Congratulations, you have written and executed your first Python program.

The Python Command Line

To test a short amount of code in python sometimes it is quickest and easiest not to write the code in a file. This is made possible because Python can be run as a command line itself.

Type the following on the Windows, Mac or Linux command line:

```
C:\Users\Your Name>python
```

From there you can write any python, including our hello world example from earlier in the tutorial:

```
C:\Users\Your Name>python
```

```
Python 3.6.4 (v3.6.4:d48eceb, Dec 19 2017, 06:04:45) [MSC v.1900 32 bit (Intel)] on win32
```

```
Type "help", "copyright", "credits" or "license" for more information.
>>> print("Hello, World!")
```

Which will write "Hello, World!" in the command line:

```
C:\Users\Your Name>python
Python 3.6.4 (v3.6.4:d48eceb, Dec 19 2017, 06:04:45) [MSC v.1900 32 bit (Intel)] on win32
Type "help", "copyright", "credits" or "license" for more information.
>>> print("Hello, World!")
Hello, World!
```

Whenever you are done in the python command line, you can simply type the following to quit the python command line interface:

```
exit()
```

Execute Python Syntax

As we learned in the previous page, Python syntax can be executed by writing directly in the Command Line:

```
>>> print("Hello, World!")
Hello, World!
```

Or by creating a python file on the server, using the .py file extension, and running it in the Command Line:

```
C:\Users\Your Name>python myfile.py
```

Python Indentations

Where in other programming languages the indentation in code is for readability only, in Python the indentation is very important.

Python uses indentation to indicate a block of code.

Example

```
if 5 > 2:
  print("Five is greater than two!")
```

Python will give you an error if you skip the indentation:

Example

```
if 5 > 2:
print("Five is greater than two!")
```

Comments

Python has commenting capability for the purpose of in-code documentation.

Comments start with a #, and Python will render the rest of the line as a comment:

Example

Comments in Python:

```
#This is a comment.
print("Hello, World!")
```

Docstrings

Python also has extended documentation capability, called docstrings.

Docstrings can be one line, or multiline.

Python uses triple quotes at the beginning and end of the docstring:

Example

Docstrings are also comments:

```
"""This is a
multiline docstring."""
print("Hello, World!")
```

5.2.ALGORITHMS

Logistic regression: Logistic regression is one of the most popular Machine Learning algorithms, which comes under the Supervised Learning technique. It is used for predicting the categorical dependent variable using a given set of independent variables. Logistic regression predicts the output of a categorical dependent variable.

Random forest: Random forest is a Supervised Machine Learning Algorithm that is used widely in Classification and Regression problems. It builds decision trees on different samples and takes their majority vote for classification and average in case of regression.

Decision tree: Decision tree is the most powerful and popular tool for classification and prediction. A Decision tree is a flowchart-like tree structure, where each internal node denotes a test on an attribute, each branch represents an outcome of the test, and each leaf node (terminal node) holds a class label.

Naïve bayes: Naïve Bayes algorithm is a supervised learning algorithm, which is based on Bayes theorem and used for solving classification problems. It is mainly used in text classification that includes a high-dimensional training dataset.

Support vector machine: Support Vector Machine(SVM) is a supervised machine learning algorithm used for both classification and regression. Though we say regression problems as well its best suited for classification. The objective of SVM algorithm is to find a hyperplane in an N-dimensional space that distinctly classifies the data points.

voting classifier: Voting Classifier is a machine-learning algorithm often used by Kagglers to boost the performance of their model and climb up the rank ladder. Voting Classifier can also be used for real-world datasets to improve performance, but it comes with some limitations.

5.3 SYSTEM ARCHITECTURE

The machine learning architecture defines the various layers involved in the machine learning cycle and involves the major steps being carried out in the transformation of raw data into training data sets capable for enabling the decision making of a system.

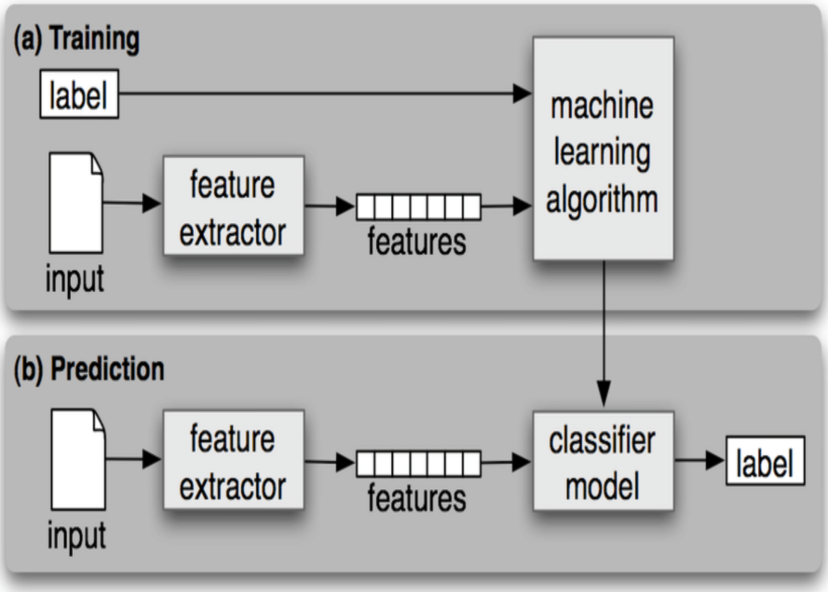


Fig 5.3 Traning & Prediction

6.TESTING

6.1 General

The purpose of testing is to discover errors. Testing is the process of trying to discover every conceivable fault or weakness in a work product. It provides a way to check the functionality of components, sub-assemblies, assemblies and/or a finished product It is the process of exercising software with the intent of ensuring that the Software system meets its requirements and user expectations and does not fail in an unacceptable manner. There are various types of test. Each test type addresses a specific testing requirement.

6.2 Developing Methodologies

The test process is initiated by developing a comprehensive plan to test the general functionality and special features on a variety of platform combinations. Strict quality control procedures are used. The process verifies that the application meets the requirements specified in the system requirements document and is bug free. The following are the considerations used to develop the framework from developing the testing methodologies.

6.3 Types of Tests

6.3.1 Unit testing

Unit testing involves the design of test cases that validate that the internal program logic is functioning properly, and that program input produce valid outputs. All decision branches and internal code flow should be validated. It is the testing of individual software units of the application .it is done after the completion of an individual unit before integration. This is a structural testing, that relies on knowledge of its construction and is invasive. Unit tests perform basic tests at component level and test a specific business process, application, and/or system configuration. Unit tests ensure that each unique path of a business process performs accurately to the documented specifications and contains clearly defined inputs and expected results.

6.3.2 Functional test

Functional tests provide systematic demonstrations that functions tested are available as specified by the business and technical requirements, system documentation, and user manuals.

Functional testing is centered on the following items:

Valid Input : identified classes of valid input must be accepted.
Invalid Input : identified classes of invalid input must be rejected.
Functions : identified functions must be exercised.
Output : identified classes of application outputs must be exercised.
Systems/Procedures : interfacing systems or procedures must be invoked.

6.3.3 System Test

System testing ensures that the entire integrated software system meets requirements. It tests a configuration to ensure known and predictable results. An example of system testing is the configuration oriented system integration test. System testing is based on process descriptions and flows, emphasizing pre-driven process links and integration points.

6.3.4 Performance Test

The Performance test ensures that the output be produced within the time limits, and the time taken by the system for compiling, giving response to the users and request being send to the system for to retrieve the results.

6.3.5 Integration Testing

Software integration testing is the incremental integration testing of two or more integrated software components on a single platform to produce failures caused by interface defects.

The task of the integration test is to check that components or software applications, e.g. components in a software system or – one step up – software applications at the company level – interact without error.

6.3.6 Acceptance Testing

User Acceptance Testing is a critical phase of any project and requires significant participation by the end user. It also ensures that the system meets the functional requirements.

Acceptance testing for Data Synchronization:

- The Acknowledgements will be received by the Sender Node after the Packets are received by the Destination Node
- The Route add operation is done only when there is a Route request in need

- The Status of Nodes information is done automatically in the Cache Updation process

6.3.7 Build the test plan

Any project can be divided into units that can be further performed for detailed processing. Then a testing strategy for each of this unit is carried out. Unit testing helps to identity the possible bugs in the individual component, so the component that has bugs can be identified and can be rectified from errors.

6.3.8 Unit Testing:

Unit testing is usually conducted as part of a combined code and unit test phase of the software lifecycle, although it is not uncommon for coding and unit testing to be conducted as two distinct phases.

Test strategy and approach

Field testing will be performed manually and functional tests will be written in detail.

Test objectives

- All field entries must work properly.
- Pages must be activated from the identified link.
- The entry screen, messages and responses must not be delayed.

Features to be tested

- Verify that the entries are of the correct format
- No duplicate entries should be allowed
- All links should take the user to the correct page.

6.3.9 Integration Testing

Software integration testing is the incremental integration testing of two or more integrated software components on a single platform to produce failures caused by interface defects.

The task of the integration test is to check that components or software applications, e.g. components in a software system or – one step up – software applications at the company level – interact without error.

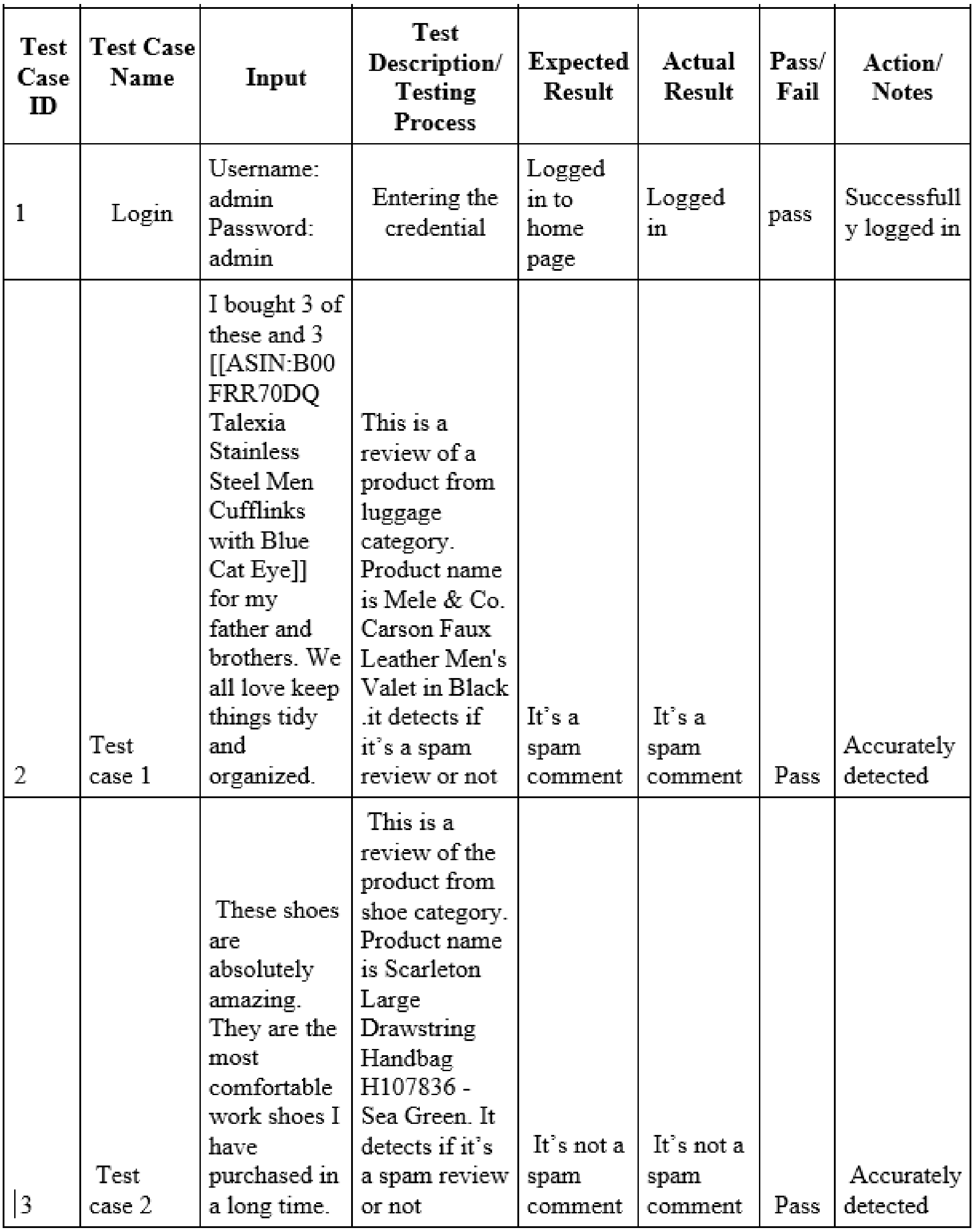
Test Results: All the test cases mentioned above passed successfully. No defects encountered.

6.3.10 Acceptance Testing

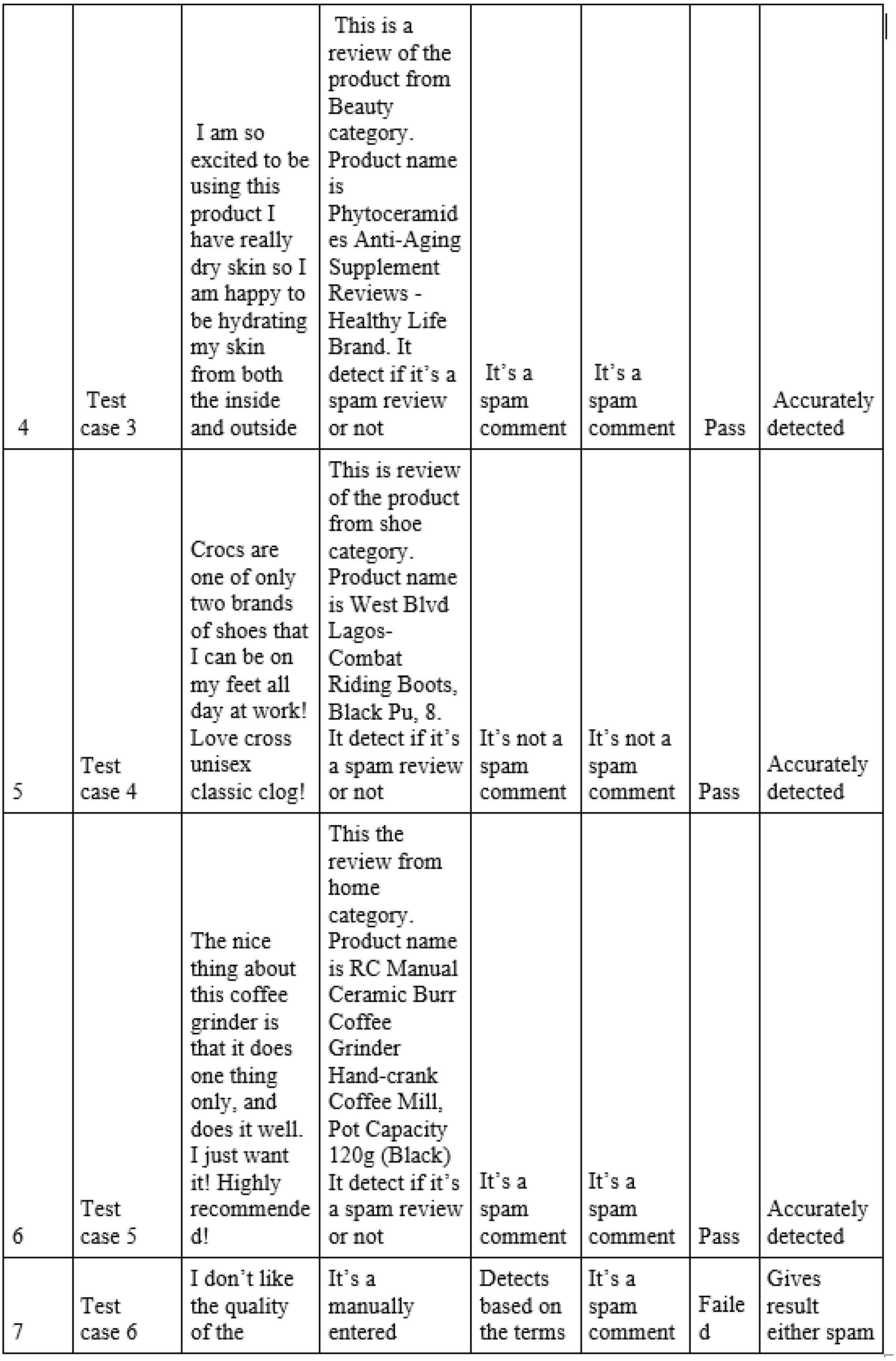
User Acceptance Testing is a critical phase of any project and requires significant participation by the end user. It also ensures that the system meets the functional requirements.

Test Results: All the test cases mentioned above passed successfully. No defects encountered.

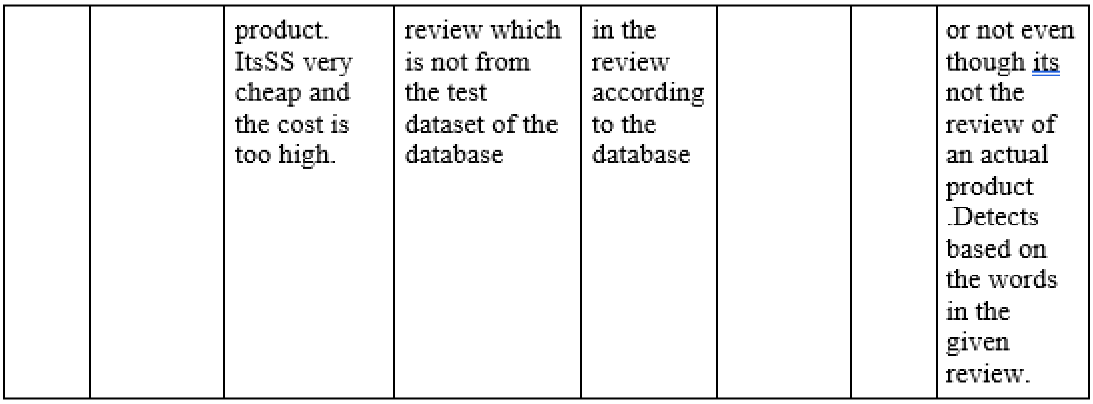
6.4 Testcases



Test Case ID	Test Case Name	Input	Test Description/ Testing Process	Expected Result	Actual Result	Pass/ Fail	Action/ Notes
1	Login	Username: admin Password: admin	Entering the credential	Logged in to home page	Logged in	pass	Successfully logged in
2	Test case 1	I bought 3 of these and 3 [[ASIN:B00FRR70DQ Talexia Stainless Steel Men Cufflinks with Blue Cat Eye]] for my father and brothers. We all love keep things tidy and organized.	This is a review of a product from luggage category. Product name is Mele & Co. Carson Faux Leather Men's Valet in Black .it detects if it's a spam review or not	It's a spam comment	It's a spam comment	Pass	Accurately detected
|3	Test case 2	These shoes are absolutely amazing. They are the most comfortable work shoes I have purchased in a long time.	This is a review of the product from shoe category. Product name is Scarleton Large Drawstring Handbag H107836 - Sea Green. It detects if it's a spam review or not	It's not a spam comment	It's not a spam comment	Pass	Accurately detected



4	Test case 3	I am so excited to be using this product I have really dry skin so I am happy to be hydrating my skin from both the inside and outside	This is a review of the product from Beauty category. Product name is Phytoceramides Anti-Aging Supplement Reviews - Healthy Life Brand. It detect if it's a spam review or not	It's a spam comment	It's a spam comment	Pass	Accurately detected
5	Test case 4	Crocs are one of only two brands of shoes that I can be on my feet all day at work! Love cross unisex classic clog!	This is review of the product from shoe category. Product name is West Blvd Lagos-Combat Riding Boots, Black Pu, 8. It detect if it's a spam review or not	It's not a spam comment	It's not a spam comment	Pass	Accurately detected
6	Test case 5	The nice thing about this coffee grinder is that it does one thing only, and does it well. I just want it! Highly recommended!	This the review from home category. Product name is RC Manual Ceramic Burr Coffee Grinder Hand-crank Coffee Mill, Pot Capacity 120g (Black) It detect if it's a spam review or not	It's a spam comment	It's a spam comment	Pass	Accurately detected
7	Test case 6	I don't like the quality of the	It's a manually entered	Detects based on the terms	It's a spam comment	Failed	Gives result either spam



		product. ItsSS very cheap and the cost is too high.	review which is not from the test dataset of the database	in the review according to the database			or not even though its not the review of an actual product .Detects based on the words in the given review.

Table 6.4 Testcases

7.RESULT AND CONCLUSION

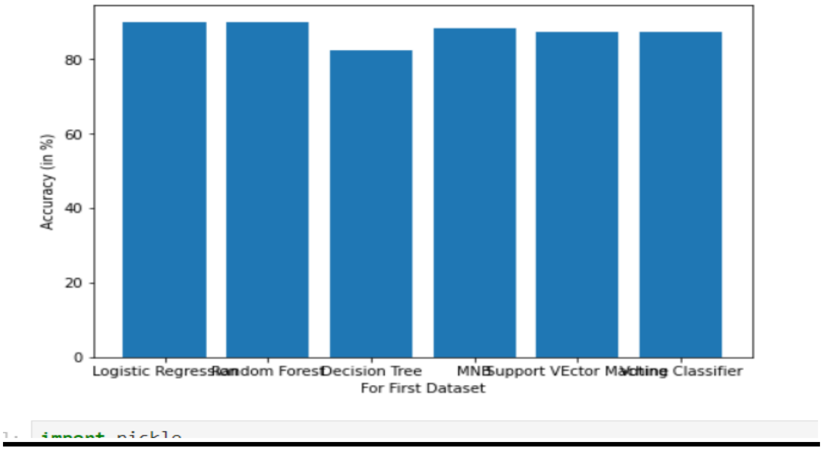
7.1 General

This project is implements like web application using COREJAVA and the Server process is maintained using the SOCKET & SERVERSOCKET and the Design part is played by Cascading Style Sheet. It

7.1.1 Various Snapshots

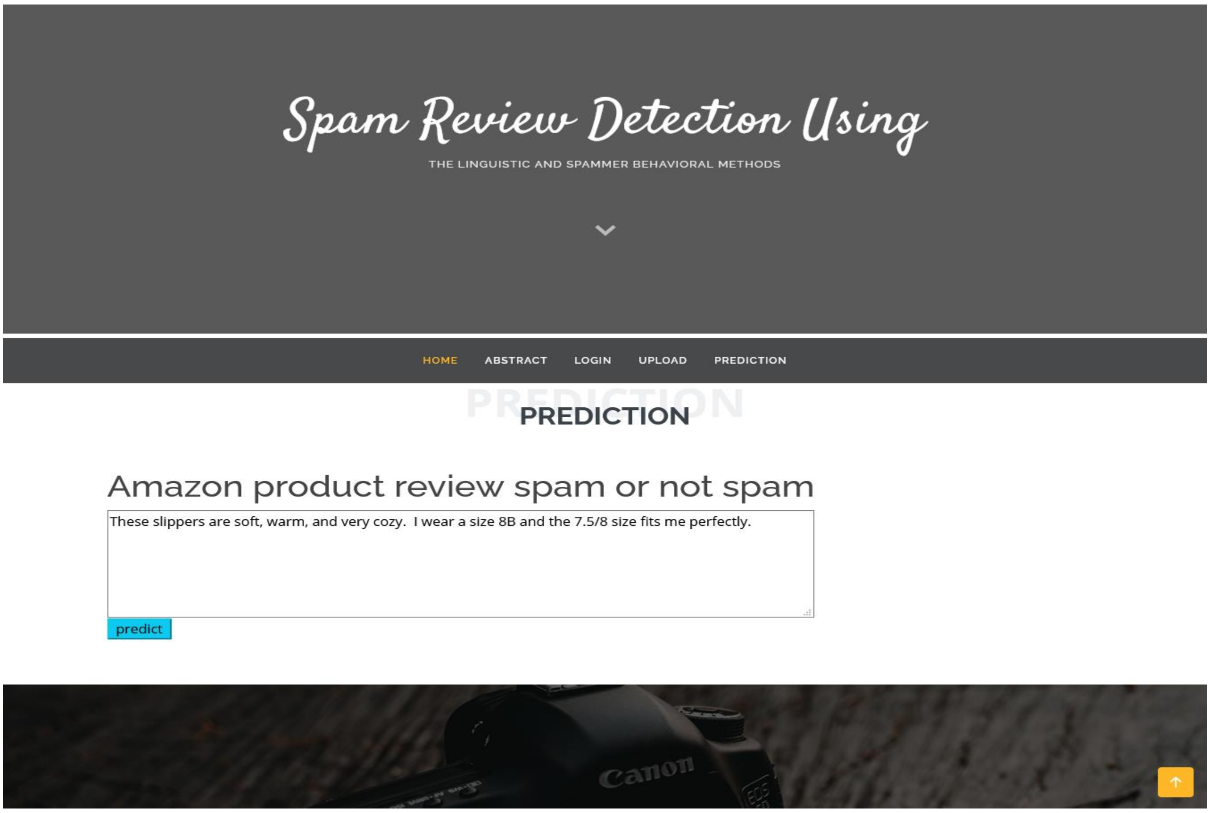
Classifiers	**Accuracy**
Logistic Regression	91
Support Vector Machine	88
Navie Bayes	85
Random Forest	90

Table 7.1.1 Classifiers with Accuracy



```
import pickle
```

Fig 7.1.1.1 Bar Graph showing Accuracy of each classifier



Spam Review Detection Using

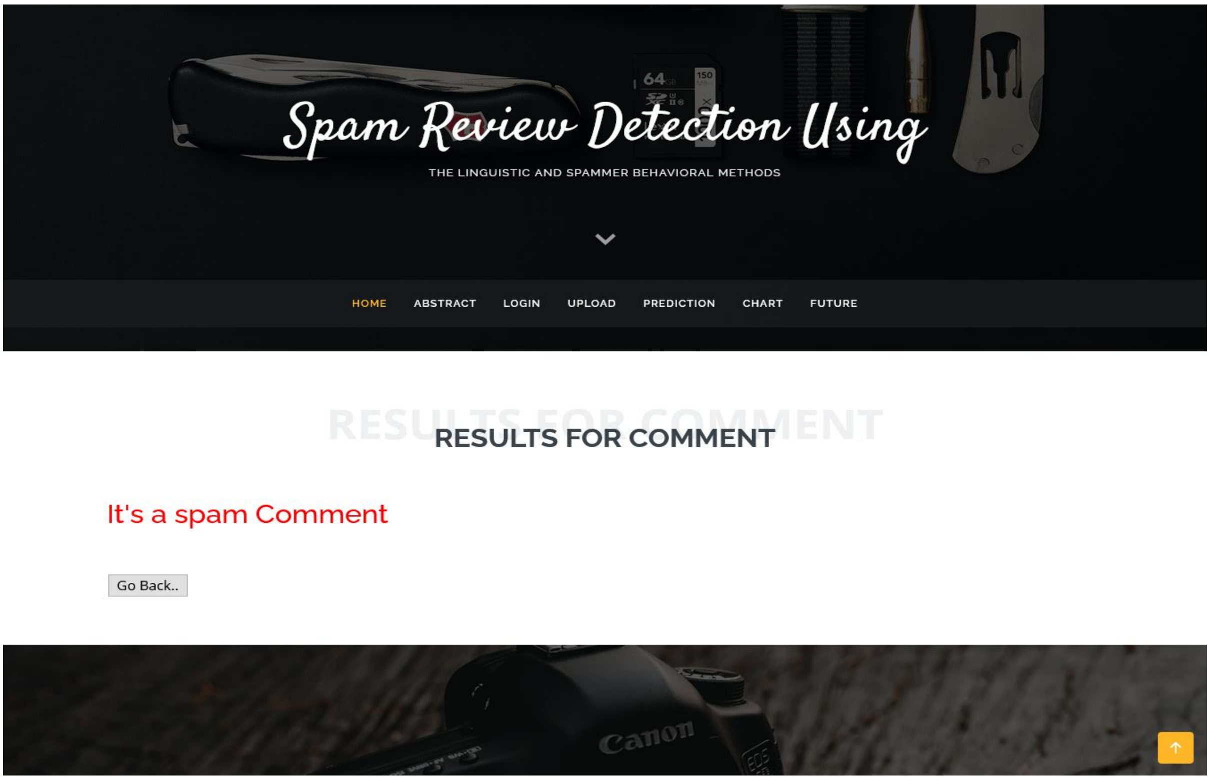
THE LINGUISTIC AND SPAMMER BEHAVIORAL METHODS

PREDICTION

Amazon product review spam or not spam

These slippers are soft, warm, and very cozy. I wear a size 8B and the 7.5/8 size fits me perfectly.

It describes the text field where the review text is given for prediction



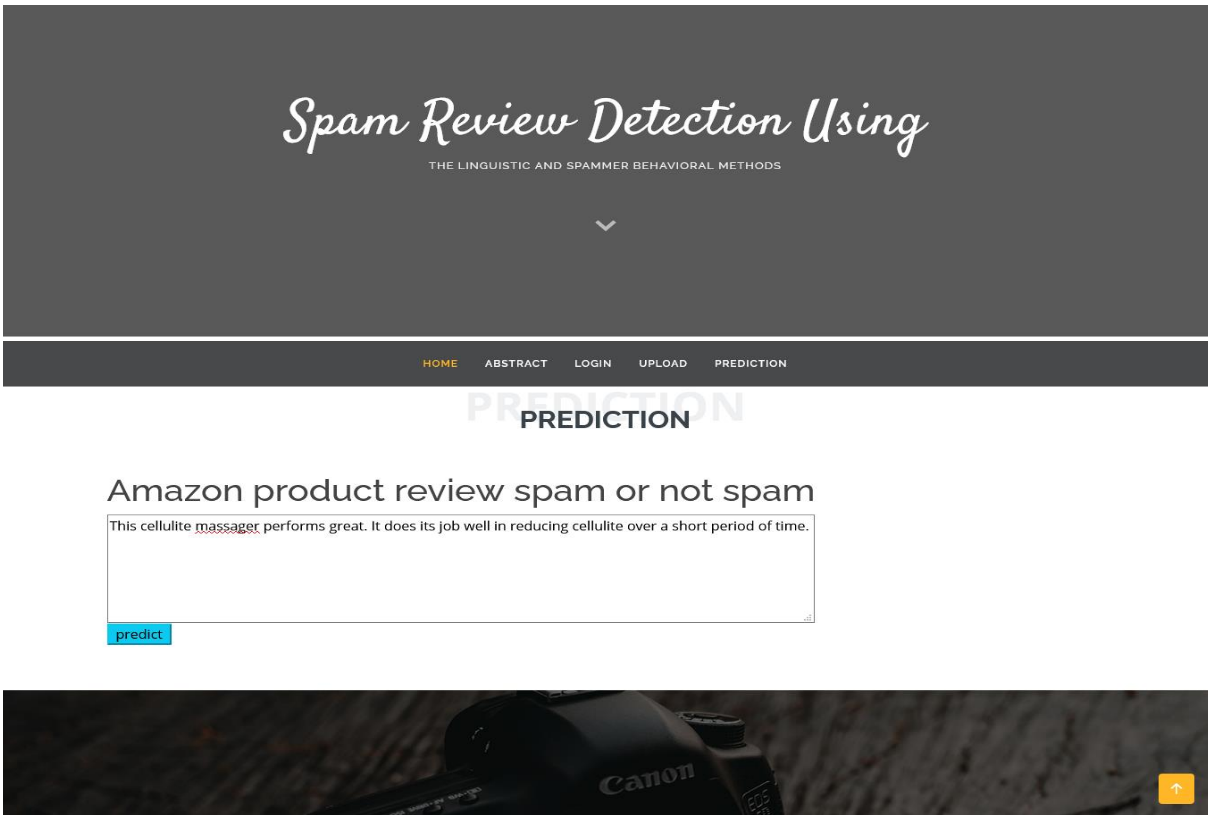
Spam Review Detection Using

THE LINGUISTIC AND SPAMMER BEHAVIORAL METHODS

RESULTS FOR COMMENT

It's a spam Comment

The prediction of the result for a review whether it is a spam or not, the result predicted as It’s a spam comment .



Spam Review Detection Using

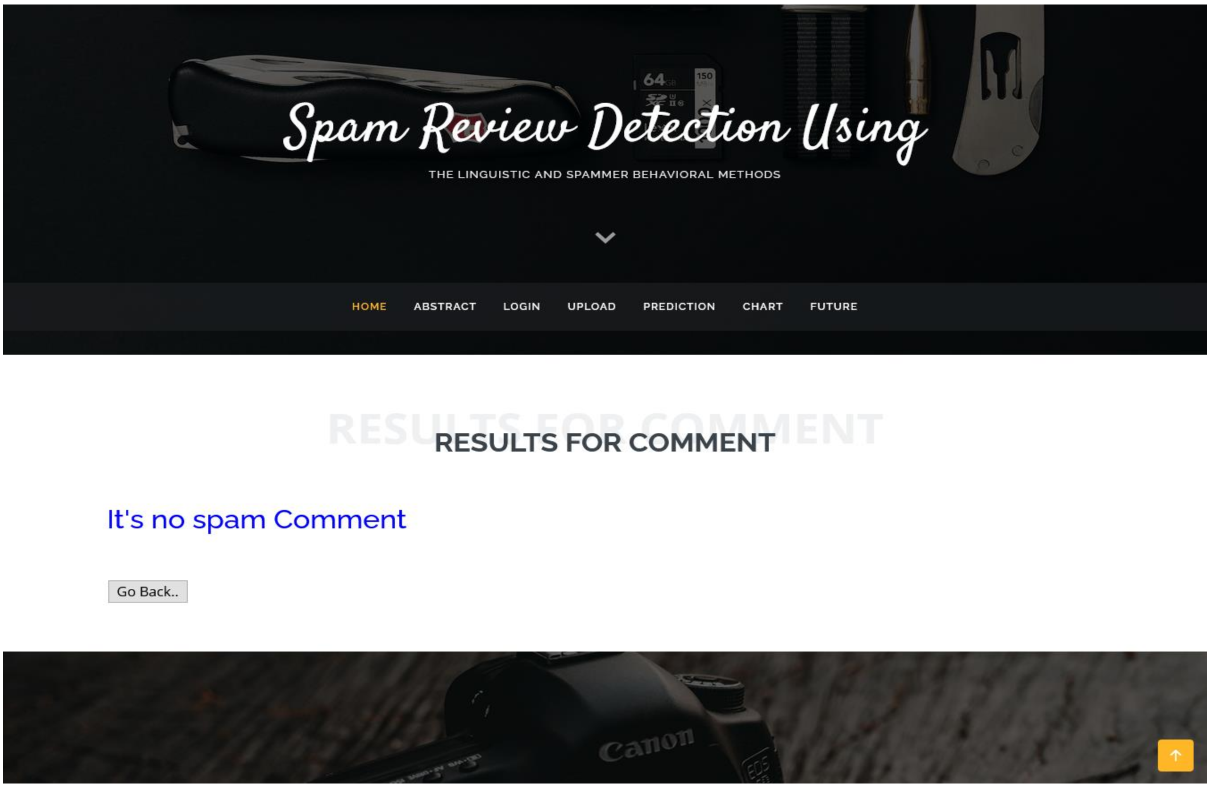
THE LINGUISTIC AND SPAMMER BEHAVIORAL METHODS

PREDICTION

Amazon product review spam or not spam

This cellulite massager performs great. It does its job well in reducing cellulite over a short period of time.

It describes the text field where the review text is given for prediction



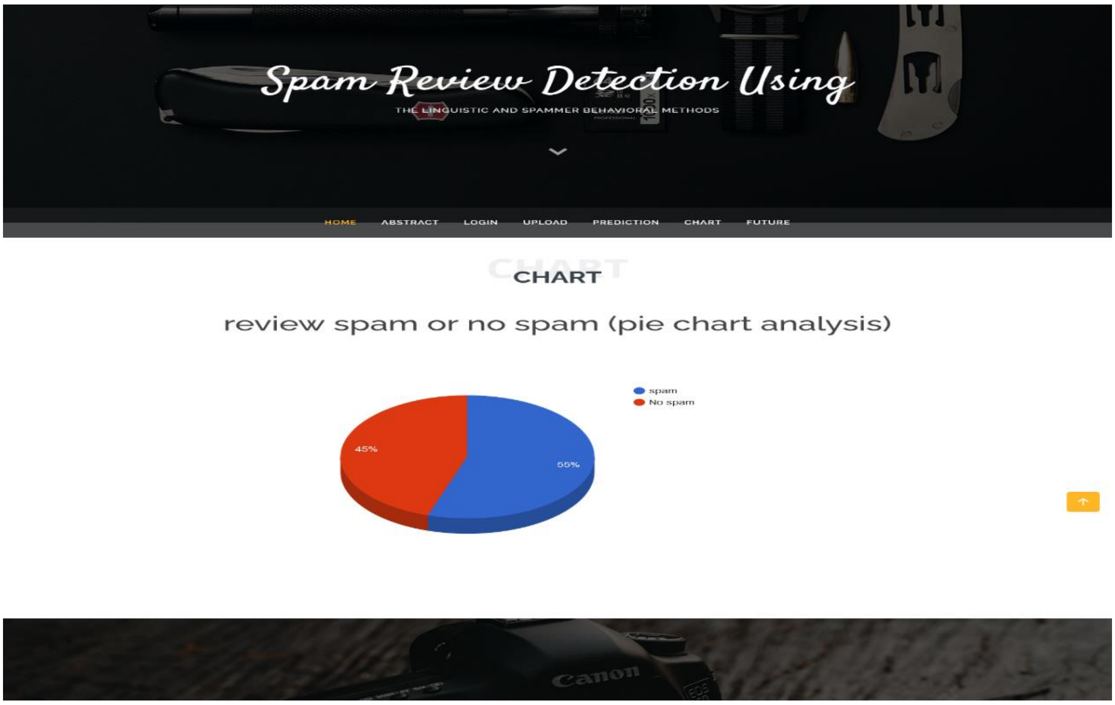
Spam Review Detection Using

THE LINGUISTIC AND SPAMMER BEHAVIORAL METHODS

RESULTS FOR COMMENT

It's no spam Comment

The prediction of the result for a review whether it is a spam or not, the result predicted as It’s no spam comment



Spam Review Detection Using

THE LINGUISTIC AND SPAMMER BEHAVIORAL METHODS

CHART

review spam or no spam (pie chart analysis)

In the above screenshot spam score is displayed for the dataset

7.2 Conclusion & Future Scope

7.2.1 Conclusion

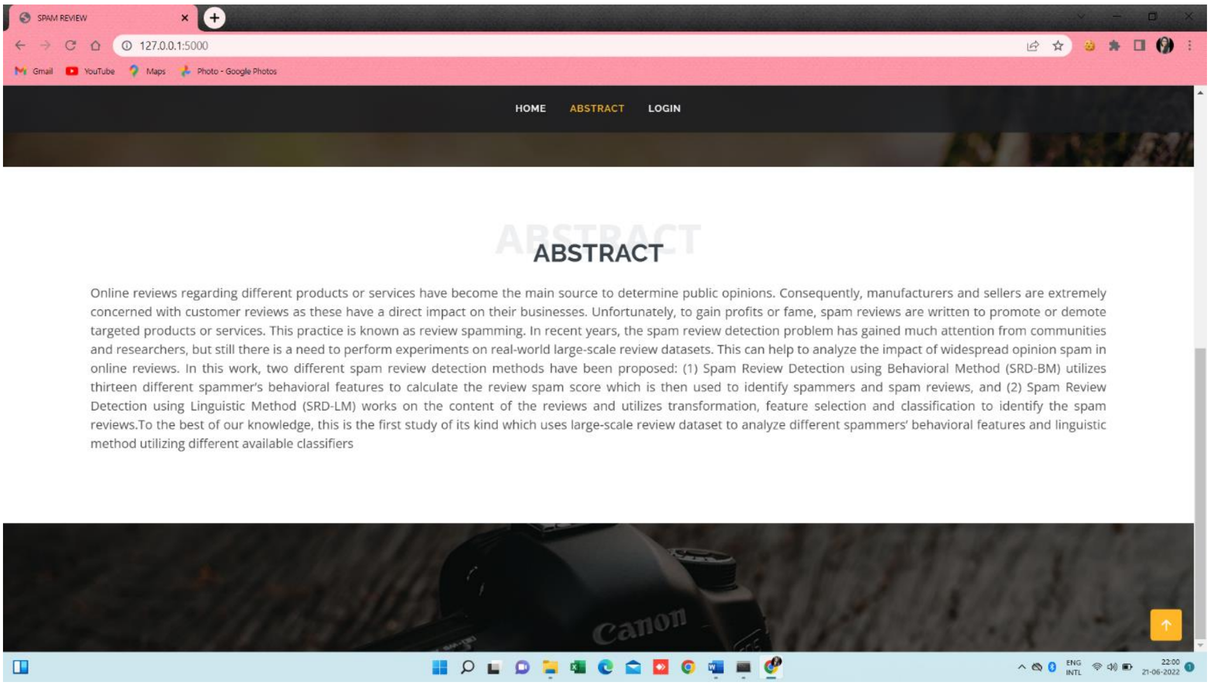
Online review spamming is a rapidly growing problem. Spam Review Detection (SRD) is a significant but challenging task as it is very difficult to differentiate the spam review from not-spam reviews. So far, many research works have attempted to identify the spammer and spam reviews, but these works have not been able to fully solve the spam review detection problem. This work performed an in-depth investigation of Amazon real-world dataset using the spammers' behavioral features and proposed SRD-BM and SRD-LM methods to detect spam reviews using behavioral and linguistic approaches respectively. To the best of the researcher's knowledge, this is the first study that analyzed and applied a rich set of spammers' behavioral features on a large-scale real-world review dataset. Furthermore, the experimental evaluation showed that the behavioral feature like content similarity, maximum number of reviews, review count, ratio of positive review, review of single product, activity window and review length features significantly improved the accuracy of the proposed SRD-BM. On the other hand, the proposed linguistic method SRD-LM, used N-gram techniques, transformation and feature selection, and different classification algorithms to further analyze the dataset for spam review detection. Through performance evaluation of each classifier, it is observed that the Logistic Regression performed better than the Support Vector Machine, Naïve Bayes and Random Forest. The comparison of the two proposed models indicated that the SRD-BM achieved better accuracy than the SRD-LM because SRD-BM uses behavioral attributes of dataset such as time stamps and ratings which provides additional support to identify spammers and thus spam reviews.The findings of this study provide a practical implication for improving the trustworthiness of online product and service review platforms. The applications of the study include spam review detection in product/services reviews on e-commerce websites, product/services websites e.g. Amazon, Yelp, TripAdvisor, Daraz.pk, foodpanda.pk, etc.

7.2.2 Future Scope

Future research will be focused on the availability of standard labelled datasets to train the classifiers. Further additional attributes will be added to the dataset to improve the accuracy and reliability of the spam review detection models. These may include an IP address of the spammer, registered an email address and signed-in location of the reviewer. Future directions may be to identify spam reviews in multilingual review dataset and recognizing the spammer by feedback analysis of other users on their written reviews. A significant future direction of this work is to implement this problem utilizing deep-learning classifiers.

APPENDIX

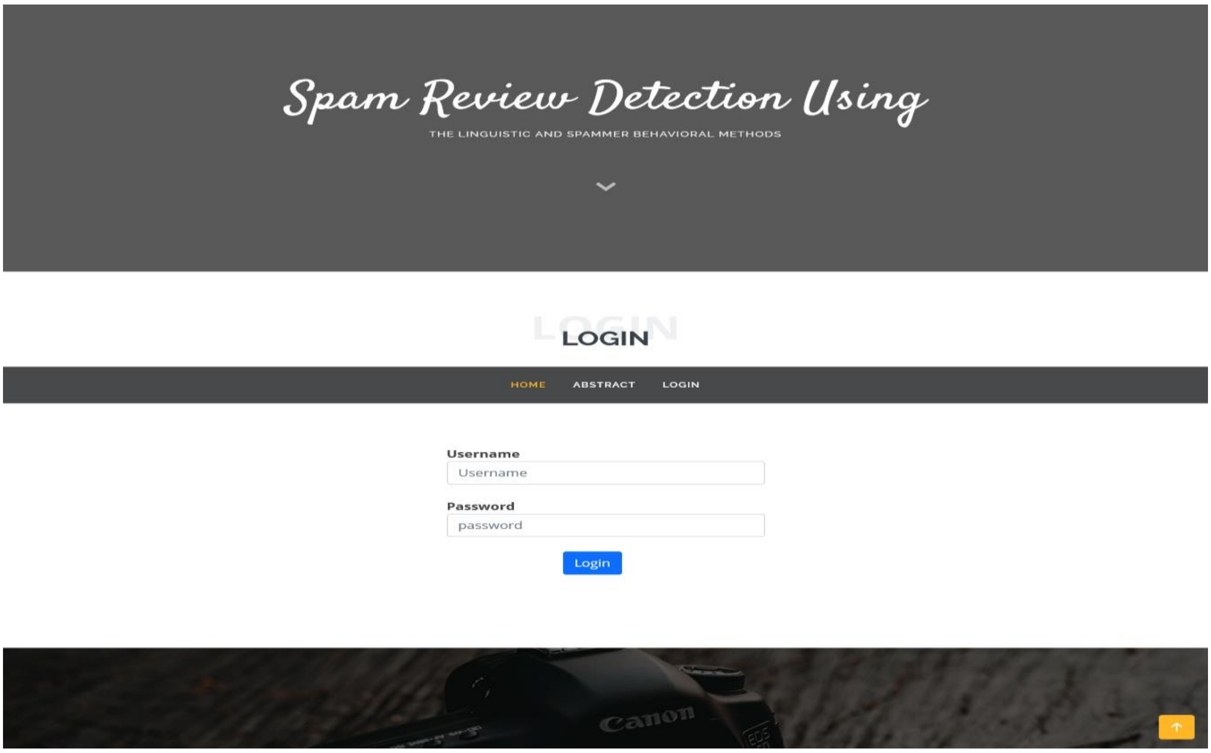
1.Screenshots



ABSTRACT

Online reviews regarding different products or services have become the main source to determine public opinions. Consequently, manufacturers and sellers are extremely concerned with customer reviews as these have a direct impact on their businesses. Unfortunately, to gain profits or fame, spam reviews are written to promote or demote targeted products or services. This practice is known as review spamming. In recent years, the spam review detection problem has gained much attention from communities and researchers, but still there is a need to perform experiments on real-world large-scale review datasets. This can help to analyze the impact of widespread opinion spam in online reviews. In this work, two different spam review detection methods have been proposed: (1) Spam Review Detection using Behavioral Method (SRD-BM) utilizes thirteen different spammer's behavioral features to calculate the review spam score which is then used to identify spammers and spam reviews, and (2) Spam Review Detection using Linguistic Method (SRD-LM) works on the content of the reviews and utilizes transformation, feature selection and classification to identify the spam reviews.To the best of our knowledge, this is the first study of its kind which uses large-scale review dataset to analyze different spammers' behavioral features and linguistic method utilizing different available classifiers

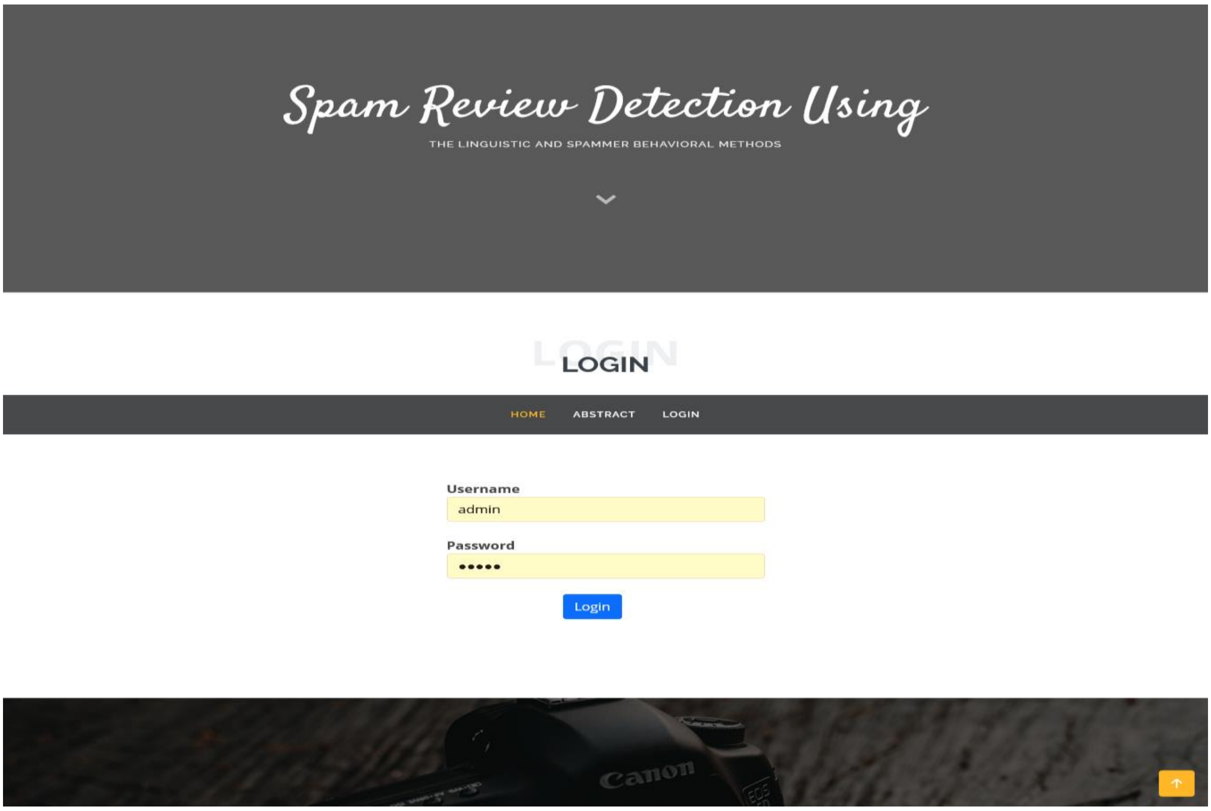
This is the home page which displays the abstract of the project


Spam Review Detection Using
THE LINGUISTIC AND SPAMMER BEHAVIORAL METHODS
LOGIN
HOME
ABSTRACT
LOGIN
Login

Username: Username

Password: password

This is the login page for the users.

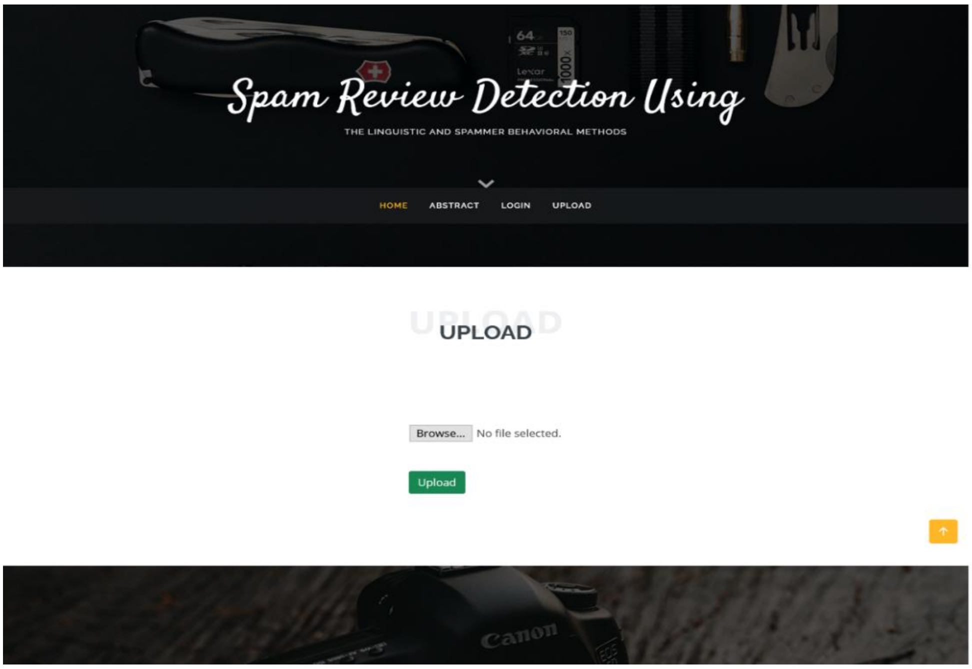


Spam Review Detection Using
THE LINGUISTIC AND SPAMMER BEHAVIORAL METHODS
LOGIN
HOME
ABSTRACT
LOGIN
Login

Username: admin

Password: •••••

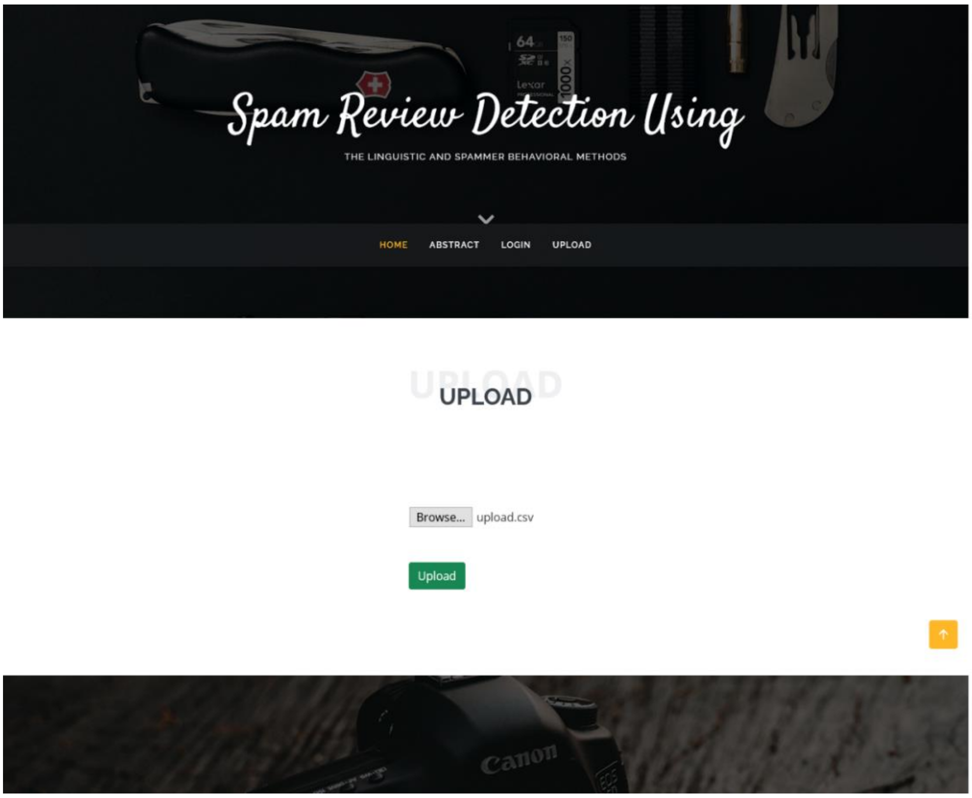
This page describes the user login .Admin entering his user name and password.



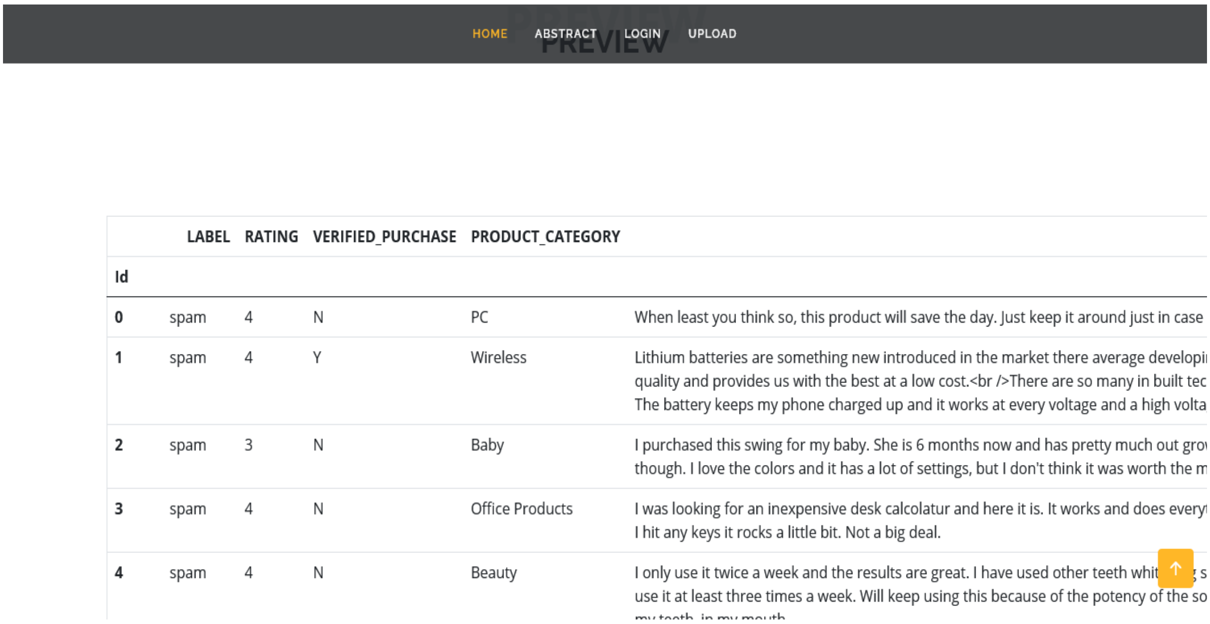
Spam Review Detection Using
THE LINGUISTIC AND SPAMMER BEHAVIORAL METHODS
HOME
ABSTRACT
LOGIN
UPLOAD
UPLOAD
Browse...
Upload

Browse... No file selected.

The page shows the upload button to upload the dataset.A dataset file need to be selected and uploaded to train and test.

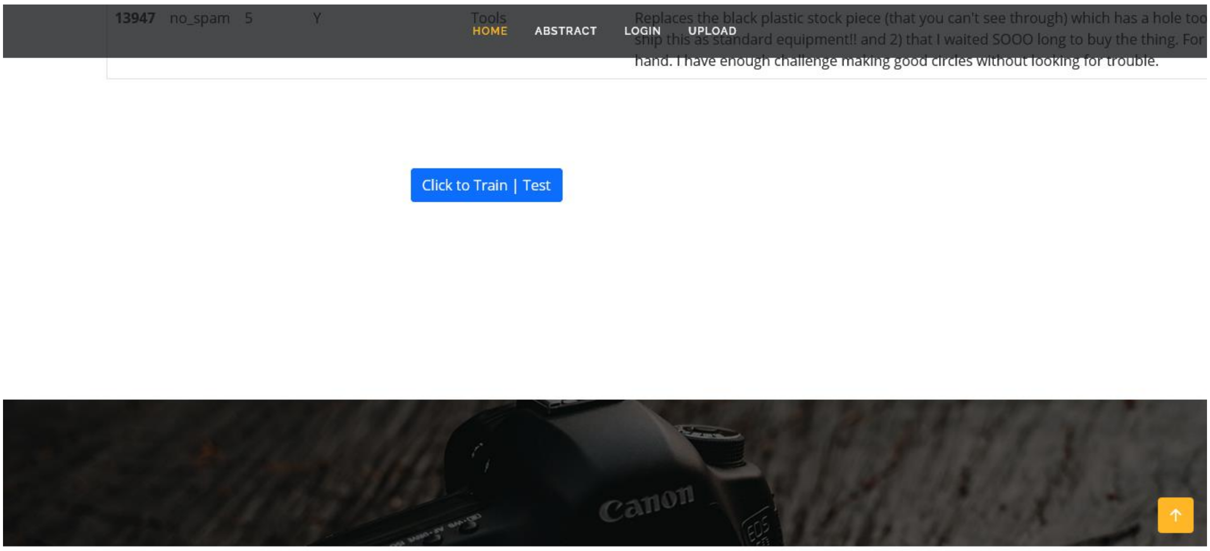

Spam Review Detection Using
THE LINGUISTIC AND SPAMMER BEHAVIORAL METHODS
HOME
ABSTRACT
LOGIN
UPLOAD
UPLOAD
Browse...
upload.csv
Upload
Canon

This page shows the document that uploaded is upload.csv for training



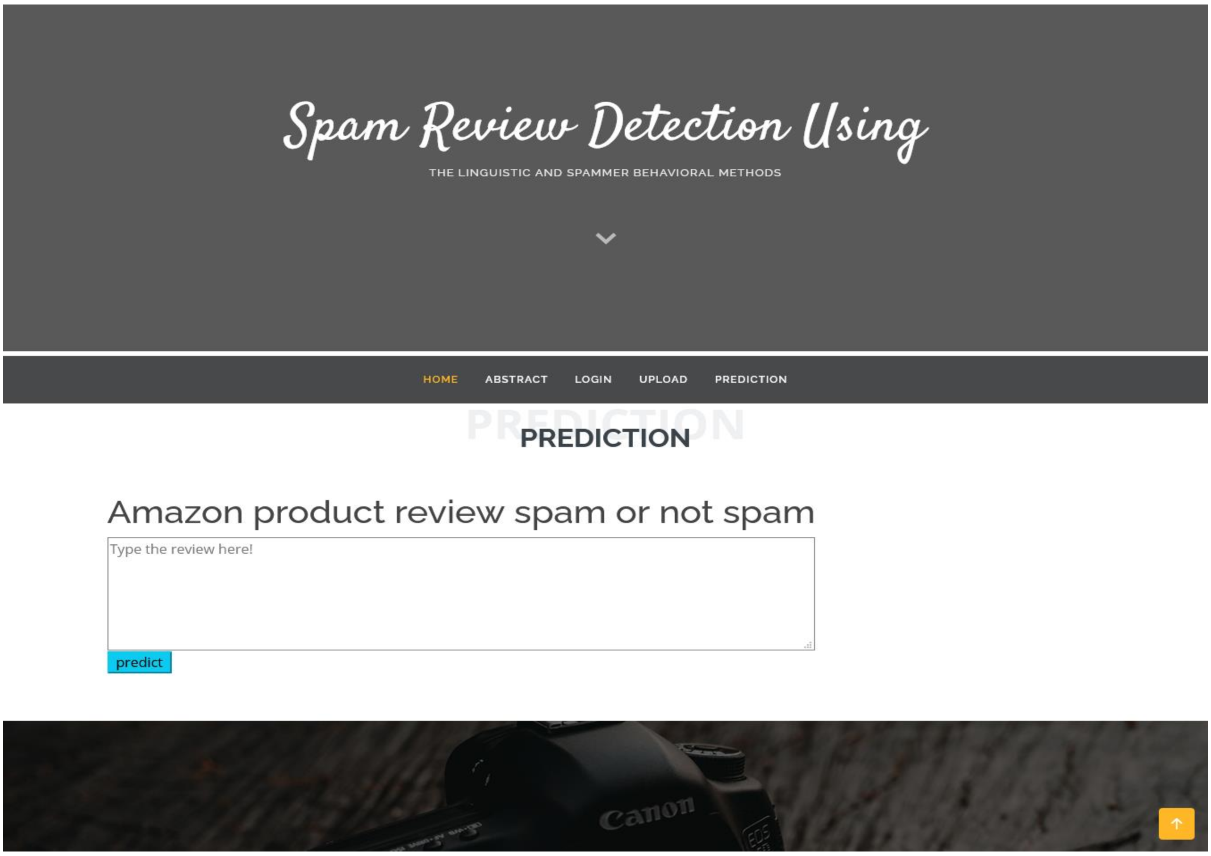
	LABEL	RATING	VERIFIED_PURCHASE	PRODUCT_CATEGORY	
Id					
0	spam	4	N	PC	When least you think so, this product will save the day. Just keep it around just in case
1	spam	4	Y	Wireless	Lithium batteries are something new introduced in the market there average developi quality and provides us with the best at a low cost.<br />There are so many in built tec The battery keeps my phone charged up and it works at every voltage and a high volta
2	spam	3	N	Baby	I purchased this swing for my baby. She is 6 months now and has pretty much out gro though. I love the colors and it has a lot of settings, but I don't think it was worth the m
3	spam	4	N	Office Products	I was looking for an inexpensive desk calcolatur and here it is. It works and does every I hit any keys it rocks a little bit. Not a big deal.
4	spam	4	N	Beauty	I only use it twice a week and the results are great. I have used other teeth whit s use it at least three times a week. Will keep using this because of the potency of the so my teeth, in my mouth

Uploaded dataset is displayed in this page .



13947 no_spam 5 Y Tools Replaces the black plastic stock piece (that you can't see through) which has a hole too ship this as standard equipment!! and 2) that I waited SOOO long to buy the thing. For hand. I have enough challenge making good circles without looking for trouble.

This screenshot shows the button of Train/test which by clicking it trains the model with the uploaded dataset

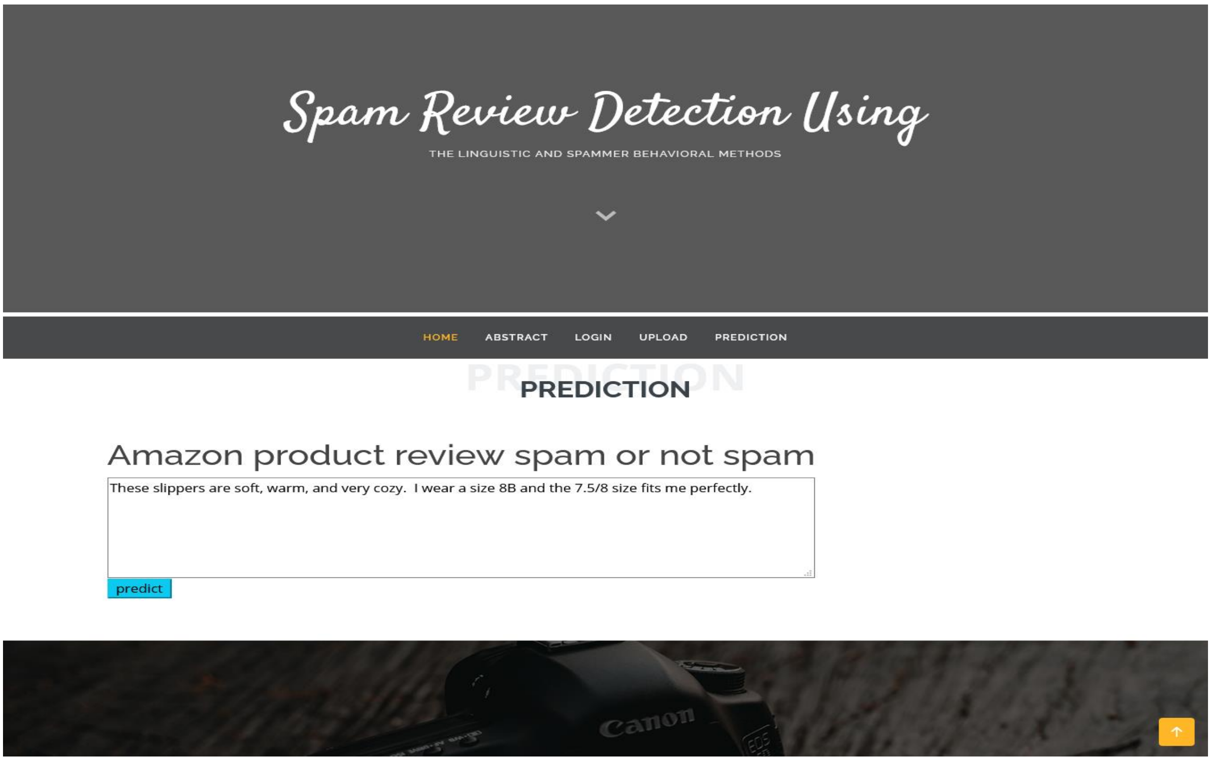

Spam Review Detection Using

THE LINGUISTIC AND SPAMMER BEHAVIORAL METHODS

PREDICTION

Amazon product review spam or not spam

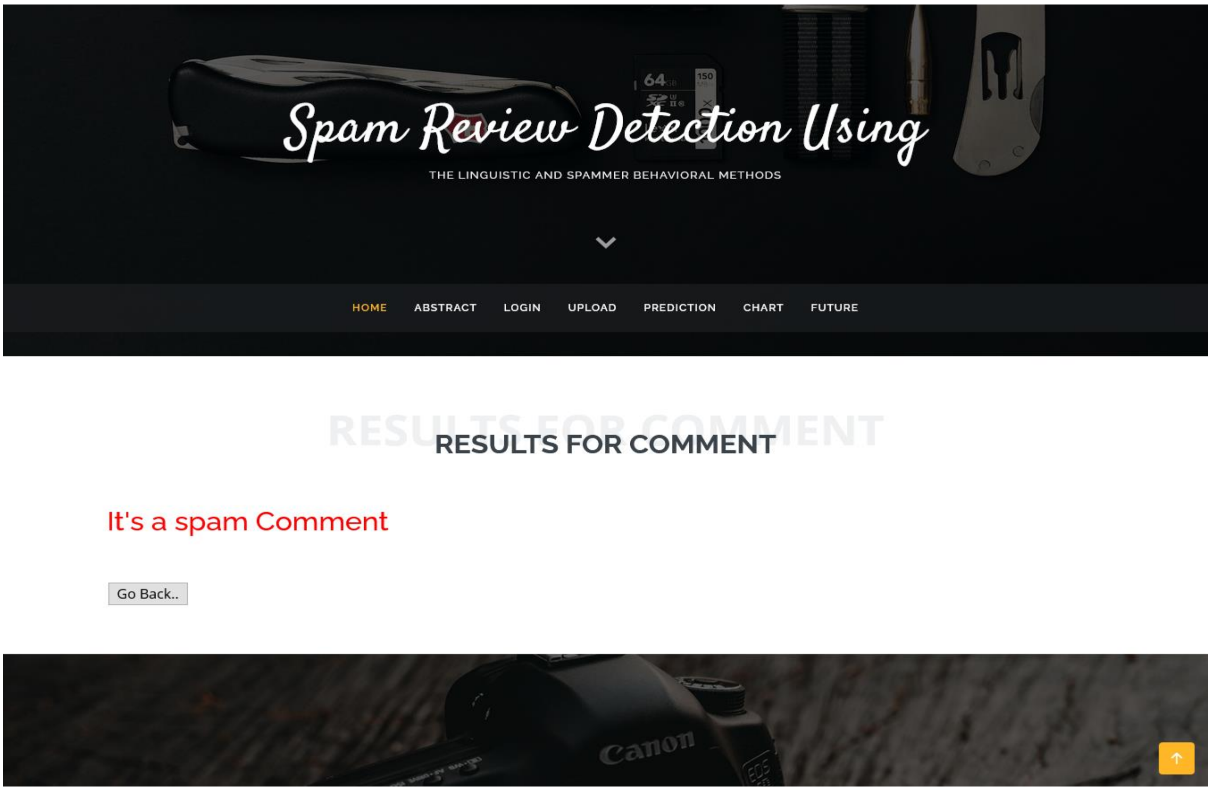
It describes the text field where the review text is given for prediction


Spam Review Detection Using
THE LINGUISTIC AND SPAMMER BEHAVIORAL METHODS
HOME
ABSTRACT
LOGIN
UPLOAD
PREDICTION
PREDICTION
predict
Canon

Amazon product review spam or not spam

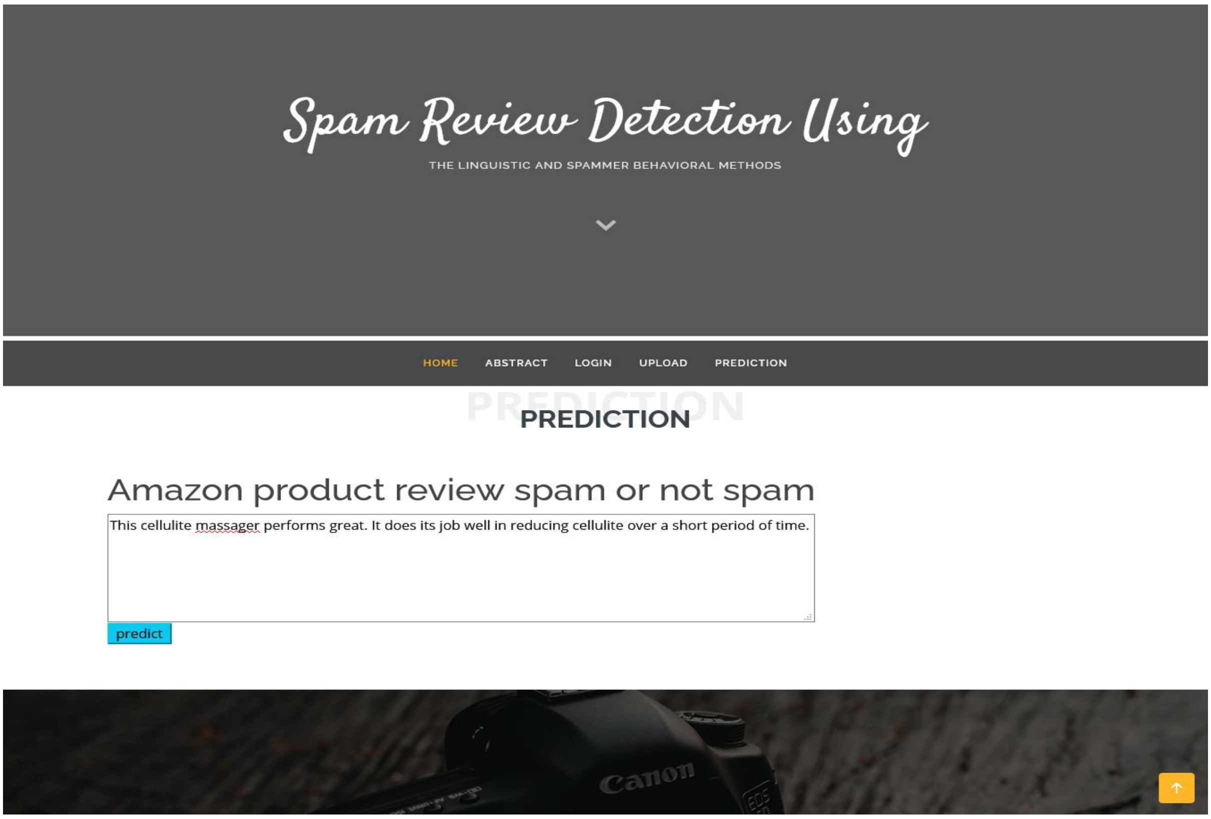
These slippers are soft, warm, and very cozy. I wear a size 8B and the 7.5/8 size fits me perfectly.

For testing, the review text is given to predict after training the model



It's a spam Comment

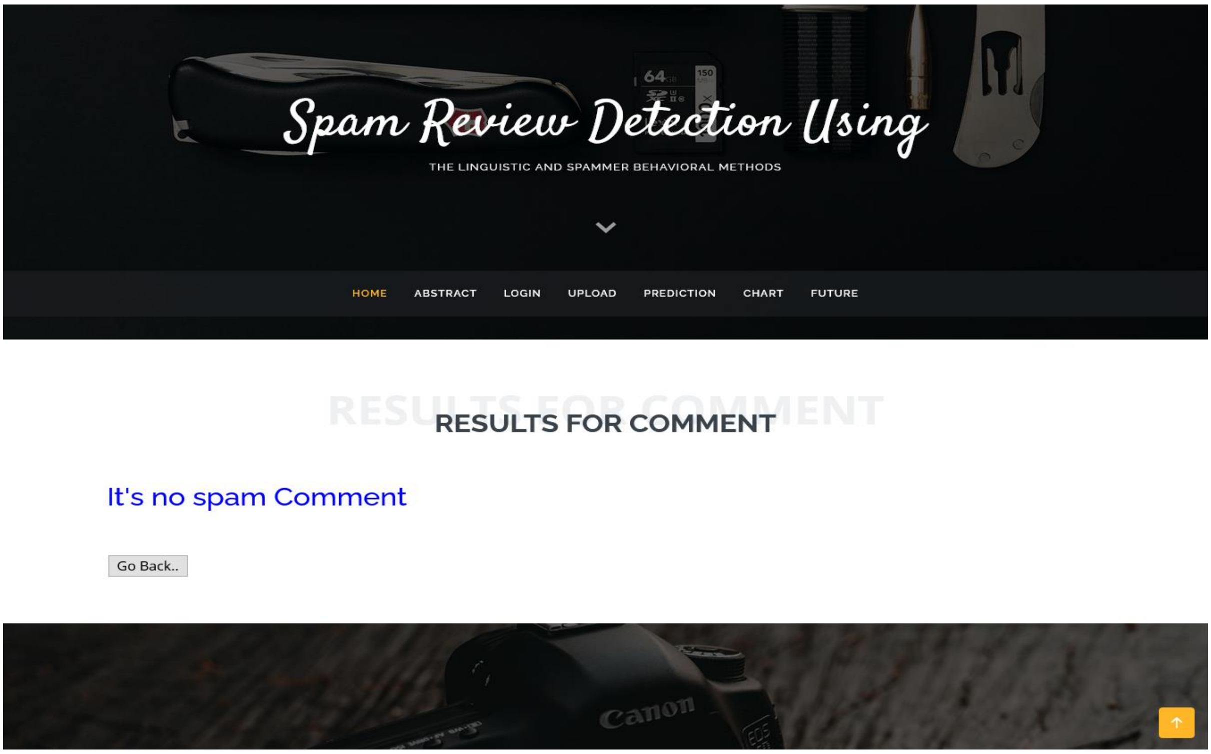
The prediction of the result for a review whether it is a spam or not, the result predicted as It’s spam comment


Spam Review Detection Using
THE LINGUISTIC AND SPAMMER BEHAVIORAL METHODS
HOME
ABSTRACT
LOGIN
UPLOAD
PREDICTION
PREDICTION

Amazon product review spam or not spam

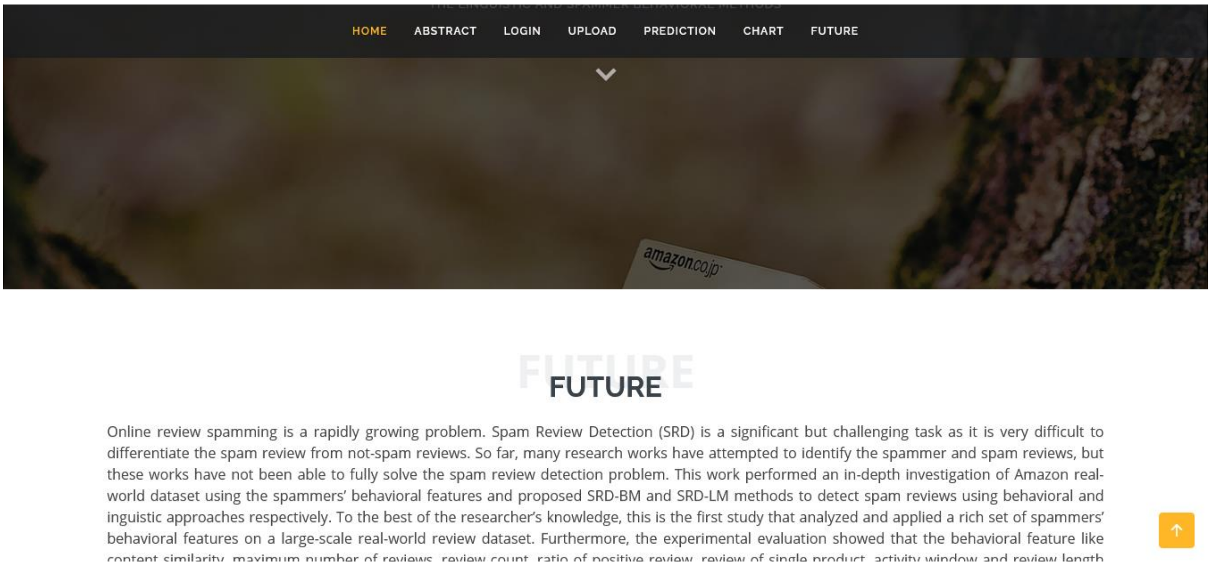
This cellulite massager performs great. It does its job well in reducing cellulite over a short period of time.

For testing, the review text is given to predict after training the model


Spam Review Detection Using
THE LINGUISTIC AND SPAMMER BEHAVIORAL METHODS
HOME
ABSTRACT
LOGIN
UPLOAD
PREDICTION
CHART
FUTURE
RESULTS FOR COMMENT

It's no spam Comment

The prediction of the result for a review whether it is a spam or not, the result predicted as It’s no spam comment.



FUTURE

Online review spamming is a rapidly growing problem. Spam Review Detection (SRD) is a significant but challenging task as it is very difficult to differentiate the spam review from not-spam reviews. So far, many research works have attempted to identify the spammer and spam reviews, but these works have not been able to fully solve the spam review detection problem. This work performed an in-depth investigation of Amazon real-world dataset using the spammers' behavioral features and proposed SRD-BM and SRD-LM methods to detect spam reviews using behavioral and inguistic approaches respectively. To the best of the researcher's knowledge, this is the first study that analyzed and applied a rich set of spammers' behavioral features on a large-scale real-world review dataset. Furthermore, the experimental evaluation showed that the behavioral feature like content similarity, maximum number of reviews, review count, ratio of positive review, review of single product, activity window and review length

Describing the future scope of the Project and its extension

II. Code

Data Loading & Cleaning:

```
import sqlite3
import pandas as pd
import numpy as np
import nltk
import string
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.feature_extraction.text import TfidfTransformer
from sklearn.feature_extraction.text import TfidfVectorizer

from sklearn.feature_extraction.text import CountVectorizer
from sklearn.metrics import confusion_matrix
from sklearn import metrics
from sklearn.metrics import roc_curve, auc
from nltk.stem.porter import PorterStemmer
# using the SQLite Table to read data.
con = sqlite3.connect('data/database.sqlite')

# filtering only positive and negative reviews i.e.
# not taking into consideration those reviews with Score=3
filtered_data = pd.read_sql_query("""
SELECT *
FROM Reviews
WHERE Score != 3
""", con)
# Give reviews with Score>3 a positive rating,
# and reviews with a score<3 a negative rating.
def partition(x):
    if x < 3:
        return 'negative'
    return 'positive'
#changing reviews with score less than 3 to be positive and vice-versa
actualScore = filtered_data['Score']
positiveNegative = actualScore.map(partition)
filtered_data['Score'] = positiveNegative
#Sorting data according to ProductId in ascending order
sorted_data=filtered_data.sort_values('ProductId',
   axis=0, ascending=True, inplace=False, kind='quicksort', na_position='last')

#Deduplication of entries
final=sorted_data.drop_duplicates(subset={
    "UserId","ProfileName","Time","Text"}, keep='first', inplace=False)
final.shape
final=final[final.HelpfulnessNumerator<=final.HelpfulnessDenominator]
final.shape
```

Data Pre-Processing:

```
import re
import string
from nltk.corpus import stopwords
from nltk.stem import PorterStemmer
from nltk.stem.wordnet import WordNetLemmatizer

stop = set(stopwords.words('english')) #set of stopwords
sno = nltk.stem.SnowballStemmer('english') #initialising the snowball stemmer

def cleanhtml(sentence): #function to clean the word of any html-tags
    cleanr = re.compile('<.*?>')
    cleantext = re.sub(cleanr, ' ', sentence)
    return cleantext
#function to clean the word of any punctuation or special characters
def cleanpunc(sentence):
    cleaned = re.sub(r'[?|!|\'|"|#]',r'',sentence)
    cleaned = re.sub(r'[.|,|)|(|\|/]',r' ',cleaned)
    return  cleaned
i=0
str1=' '
final_string=[]
all_positive_words=[] # store words from +ve reviews here
all_negative_words=[] # store words from -ve reviews here.
s=''
for sent in final['Text'].values:
    filtered_sentence=[]
    #print(sent);
    sent=cleanhtml(sent) # remove HTMl tags
    for w in sent.split():
        for cleaned_words in cleanpunc(w).split():
            if((cleaned_words.isalpha()) & (len(cleaned_words)>2)):
                if(cleaned_words.lower() not in stop):
                    s=(sno.stem(cleaned_words.lower())).encode('utf8')
                    filtered_sentence.append(s)
                    if (final['Score'].values)[i] == 'positive':
                #list of all words used to describe positive reviews
                        all_positive_words.append(s)
                    if(final['Score'].values)[i] == 'negative':
               #list of all words used to describe negative reviews reviews
                        all_negative_words.append(s)
                else:
                    continue
            else:
                continue
    #print(filtered_sentence)
    str1 = b" ".join(filtered_sentence) #final string of cleaned words

    final_string.append(str1)
    i+=1
final['CleanedText']=final_string
print(final['CleanedText'].head(3))
```

```
import numpy as np
import pickle
import itertools
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.feature_extraction.text import TfidfVectorizer
from sklearn.linear_model import PassiveAggressiveClassifier
from sklearn.metrics import accuracy_score, confusion_matrix
train = pd.read_csv('amazon.csv')
df_x=train['REVIEW_TEXT']
df_y=train['LABEL']
from sklearn.model_selection import train_test_split
x_train, x_test, y_train, y_test = train_test_split(df_x,df_y,test_size=0.3, random_state=9)
print(x_train.shape)
print(x_test.shape)
tfidf_vectorizer = TfidfVectorizer(stop_words='english', max_df=0.7)
from sklearn.feature_extraction.text import TfidfVectorizer
tfidf_vectorizer= TfidfVectorizer(min_df=1,stop_words='english')
tfidf_train = tfidf_vectorizer.fit_transform(x_train)
pac = PassiveAggressiveClassifier(max_iter=50)
pac.fit(tfidf_train,y_train)
pac.score(tfidf_train,y_train)
tfidf_test = tfidf_vectorizer.transform(x_test)
from sklearn.metrics import accuracy_score
y_pred = pac.predict(tfidf_test )
accuracy_score(y_pred,y_test)
```

Logistic Regression:

```
from sklearn.linear_model import LogisticRegression
LogisticR = LogisticRegression()
LogisticR.fit(X_train, y_train)
y_pred = LogisticR.predict(X_test)
print(confusion_matrix(y_test,y_pred))
print(classification_report(y_test,y_pred))
print(accuracy_score(y_test, y_pred))
LR = accuracy_score(y_test, y_pred)
```

Random Forest:

```
from sklearn.ensemble import RandomForestClassifier
RandomForest = RandomForestClassifier(n_estimators=10, random_state=0)
RandomForest.fit(X_train, y_train)
y_pred = RandomForest.predict(X_test)
print(confusion_matrix(y_test,y_pred))
print(classification_report(y_test,y_pred))
print(accuracy_score(y_test, y_pred))
```

```
RF = accuracy_score(y_test, y_pred)
```

Naive Bayes:

```
from sklearn.naive_bayes import MultinomialNB
MultinomialNB = MultinomialNB()
MultinomialNB.fit(X_train, y_train)
y_pred = MultinomialNB.predict(X_test)
print(confusion_matrix(y_test,y_pred))
print(classification_report(y_test,y_pred))
print(accuracy_score(y_test, y_pred))
MNB = accuracy_score(y_test, y_pred)
```

Support Vector Machine:

```
from sklearn.svm import SVC
model3 = SVC()
model3.fit(X_train, y_train)
y_pred = model3.predict(X_test)
print(confusion_matrix(y_test,y_pred))
print(classification_report(y_test,y_pred))
print(accuracy_score(y_test, y_pred))
SVM = accuracy_score(y_test, y_pred)
```

```
accuracies = [LR, RF,MNB,SVM]
for i in range(0, len(accuracies)):
    accuracies[i] = accuracies[i]*100
models = ['Logistic Regression' , 'Random Forest','MNB', 'Support Vector Machine']
import matplotlib.pyplot as plt
fig = plt.figure()
ax = fig.add_axes([0,0,1,1])
ax.bar(models,accuracies)
plt.xlabel("For First Dataset")
plt.ylabel("Accuracy (in %)")
```

III. Bibliography / References

1. Spam Review Detection Using the Linguistic and Spammer Behavioral Methods Author: Hamid Turab Mirza ,IEEE Paper.
2. ”Learning to detect deceptive opinion spam: A survey," IEEE Access, vol. 7, pp. 4293442945, Y. Ren and D. Ji, 2019.
3. ”A framework for fake review detection in online consumer electronics retailers," Inf. Process. Manage., vol. 56, no. 4, pp. 12341244, R. Barbado, O. Araque, and C. A. Iglesias, Jul. 2019.
4. ”Spam review detection using spiral cuckoo search clustering method," Evol. Intell., vol. 12, no. 2, pp. 147164, A. C. Pandey and D. S. Rajpoot, Jun. 2019.

5. ‘‘Detecting review manipulation on online platforms with hierarchical supervised learning,’’ J. Manage. Inf. Syst., vol. 35, no. 1, pp. 350–380, N. Kumar, D. Venugopal, L. Qiu, and S. Kumar, Jan. 2018.

6. ‘‘Effective sentimental analysis and opinion mining of Web reviews using rule-based classifiers,’’ in Computational Intelligence in Data Mining, vol. 1. S. Ahmed and A. Danti, New Delhi, India: Springer, 2016, pp. 171–179.