

Low Level Design

Spam Ham Classifier

Written By	Geetha S Akanksha Venkatesh V B Ashok Kumar S M C Shravan
Document Version	0.1
Last Revised Date	29 - September - 2021

Document Control**Change Record:**

Version	Date	Author	Comments
0.1	29 –September - 2021	Geetha S Akanksha Venkatesh V B Ashok Kumar S M C Shravan	Introduction & Architecture

Reviews:

Version	Date	Reviewer	Comments

Approval Status:

Version	Review Date	Reviewed By	Approved By	Comments

Contents

1. Introduction	1
1.1. What is Low-Level design document?	1
1.2. Scope	1
2. Architecture	2
3. Architecture Description	3
3.1. Data Description	3
3.2. Data insertion into Database	3
3.3. Export Data from Database	3
3.4. Data Pre-processing	3
3.5 Machine Learning Techniques	3
3.6 Model Building	3
3.7 Evaluation	4
3.8 Cloud Setup	4
3.9 Pushing to Cloud	4
3.10. Application Start	4
3.11. Data from User	4
3.12. Data Validation	4
3.13. Saving Output at Database	4
3.14 Result Prediction	4
4. Unit Test Cases	5

1. Introduction

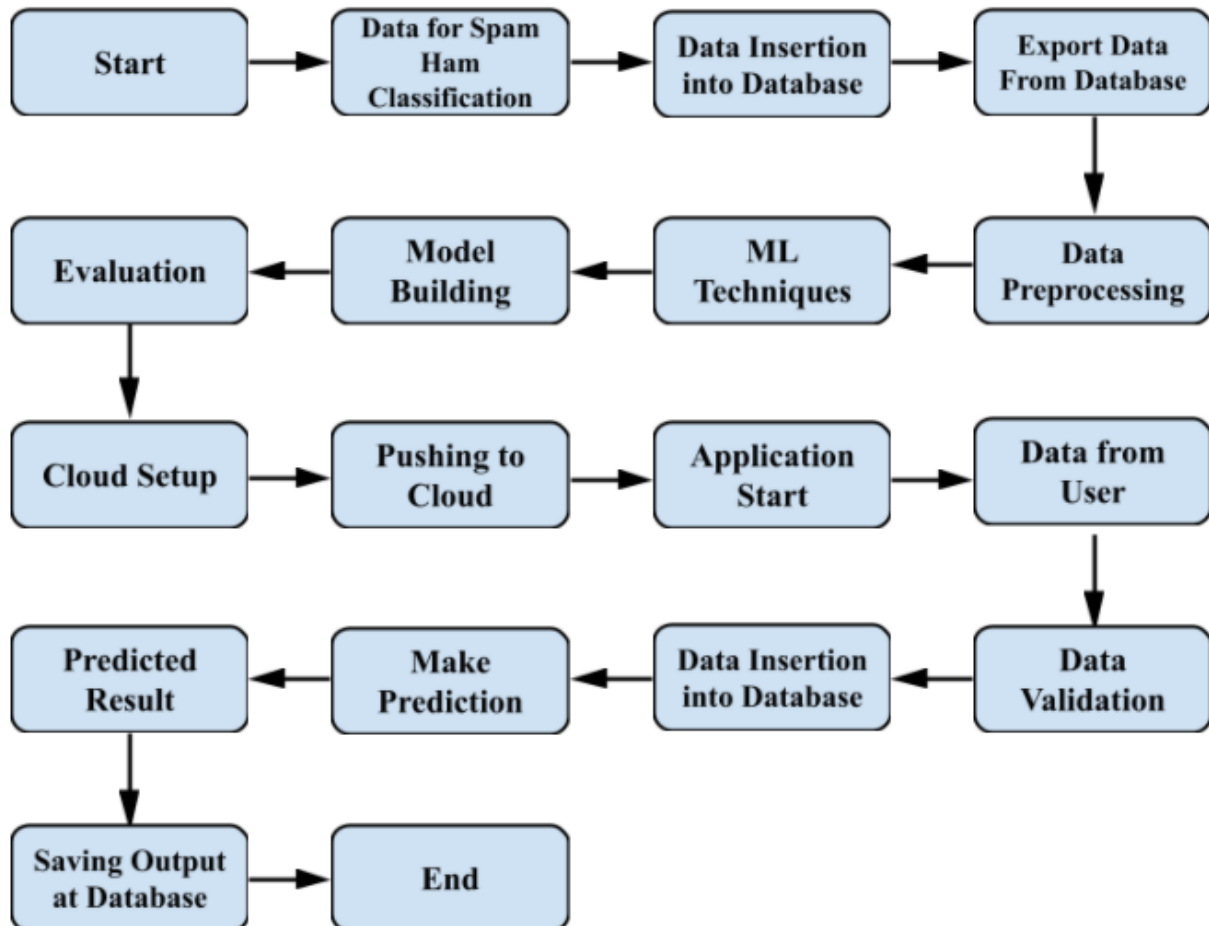
1.1. What is Low-Level design document?

The goal of LLD or a low-level design document (LLDD) is to give the internal logical design of the actual program code for Spam Ham Classifier. LLD describes the class diagrams with the methods and relations between classes and program specs. It describes the modules so that the programmer can directly code the program from the document.

1.2. Scope

Low-level design (LLD) is a component-level design process that follows a step-by-step refinement process. This process can be used for designing data structures, required software architecture, source code and ultimately, performance algorithms. Overall, the data organization may be defined during requirement analysis and then refined during data design work

2. Architecture



3. Architecture Description

3.1. Data Description

The collection is composed of just one text file, where each line has the correct class followed by the raw message.

The identification of the text of spam messages in the claims is a very hard and time-consuming task, and it involves carefully scanning hundreds of web pages.

The 425 SMS spam messages are manually extracted from the Grumble text Web site and a subset of 3,375 SMS have been randomly chosen as the ham messages of the NUS SMS Corpus (NSC), which is a dataset of about 10,000 legitimate.

These messages were collected from volunteers who were made aware that their contributions were going to be made publicly available.

3.2. Data Insertion into Database

- a. In this project we have used MySQL database.
- b. spam_ham database has been created .
- c. Using that database we have created a table called as data.
- d. For that table we have inserted the data .

3.3. Export data from Database

- a. We connect to spam_ham database.
- b. From that database we export the data

3.4. Data Pre-processing

Data Pre-processing steps we could use are data cleaning, data integration, data reduction and data transformation.

3.5. Machine Learning Techniques

Based on the problem statement and requirements we can use a supervised classification technique that fits the project.

3.6. Model Building

Depending on the data type of the target variable we are either going to be building a classification model. The main aspect of machine learning model building is to obtain actionable insights and in order to achieve that it is important to be able to select a subset of important features from the vast number.

4 Low Level Design

3.7. Evaluation

The Evaluation of accuracy can be done using the test data. Confusion Matrix, Accuracy Score and Classification Report can be found using test data and Prediction data.

3.8. Cloud Setup

Using Heroku as the cloud deployment platform, the platform is set up for deploying the virtual app.

3.9. Pushing to Cloud

Once the cloud is set up, the virtual app created will be pushed to the cloud and will finally be deployed into the cloud

3.10. Application Start

Once the virtual app is deployed into the cloud we can open the web application using any web browser.

3.11. Data from user

Using a web browser we open the web application and provide the necessary information as the input for prediction.

3.12. Data Validation

Once the input is provided and we click on the submit button, the system will provide the output based on its requirements.

3.13. Saving Output at Database

For a message given by the user ,the output of the data will be saved in the database as spam or ham.

3.14. Result Prediction

Once the data validation is completed the prediction will be done for the Spam and Ham Messages in Spam Ham Classifier provided in the input.

4. Unit Test Cases

Test Case Description	Pre-Requisite	Expected Result
Verify whether the Application URL is accessible to the user	1. Application URL should be defined	Application URL should be accessible to the user
Verify whether the Application loads completely for the user when the URL is accessed	1. Application URL is accessible 2. Application is deployed	The Application should load completely for the user when the URL is accessed
Verify whether user is able to edit all input fields	1. Application is accessible 2. User can open the application.	User should be able to edit all input fields
Verify whether user gets Submit button to submit the inputs	1. Application is accessible 2. User can edit the inputs.	User should get Submit button to submit the inputs
Verify whether user is presented with suitable results on clicking submit	1. Application is accessible	User should be presented with suitable results on clicking submit
Verify whether the suitable results are in accordance to the user of the selection made	1. Application is accessible	The suitable results should be in accordance to the selections user made
Verify whether KPIs modify as per the user inputs for the prediction	1. Application is accessible	KPIs should modify as per the user inputs for the prediction