

Low Level Design

Mushroom Classification System

Written By	Vishal Kumar Mahato, Sumit Kumar
Document Version	0.2
Last Revised Date	06-April-2024

Document Control**Change Record:**

Version	Date	Author	Comments
0.1	06-April-2024	Vishal Kumar Mahato	Introduction
0.2	07-April-2024	Sumit Kumar	Architecture, Architecture Description

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	2
3.1. Data Description	2
3.6. Data Pre-processing	2
3.10. Model Building	2
3.11. Data from User	2
3.12. Data Validation	3
3.17. Deployment	3
4. Unit Test Cases	3

1. Introduction

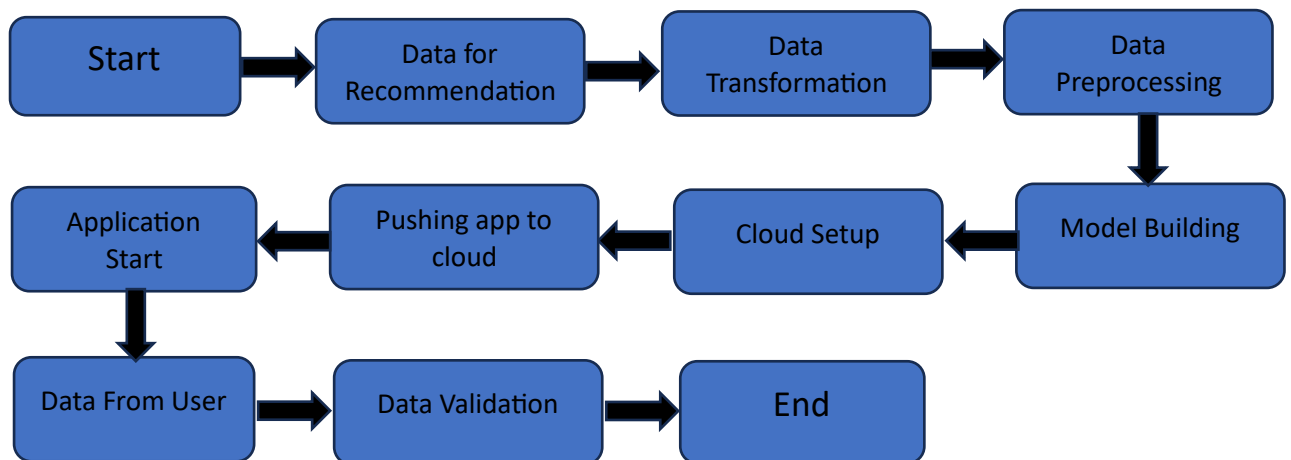
1.1. What is Low-Level design document?

The goal of LLD or a low-level design document (LLDD) is to give the internal logical design of the actual program code for Mushroom Classification System. LLD describes the class diagrams with the methods and relations between classes and program specification. 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 mushroom dataset contains information about various mushroom species, particularly focusing on features that help determine whether a mushroom is edible or poisonous. The dataset is organized into a single CSV (Comma-Separated Values) file. I also provide the feature names and their corresponding possible values as described in the provided mapping JSON file.

3.2. Data Pre-processing

Perform data cleaning and preprocessing tasks such as handling missing values, encoding categorical variables, and scaling numerical features if necessary.

3.3. Model Building

We will find the best model for classification. For each Classification Model, algorithms will be passed with the best parameters derived from Grid-Search. We will calculate the AUC scores for models and select the model with the best score.

3.4. Data from User

The user is expected to submit mushroom characteristics data, adhering to predefined formats and preprocessing guidelines. Upon submission, the system will provide binary classification predictions indicating whether the mushrooms are edible or poisonous.

3.5. Data Validation

Here Data Validation will be done, given by the user

3.6. Deployment

We will be deploying the model to AWS.

4. Unit Test Cases

Test Case Description	Pre-Requisite	Expected Result
Verify application URL accessibility	Application URL defined	Application URL should be accessible to the user
Verify application loads completely	Application deployed	The application should load completely for the user
Verify input field visibility	Application accessible	User should be able to see input fields upon application launch
Verify editability of input fields	Application accessible	User should be able to edit all input fields
Verify submit button presence	Application accessible	User should get a Submit button to submit the inputs
Verify displaying recommended results	Application accessible	User should be presented with recommended results on clicking submit
Verify accuracy of recommendations	Application accessible	The recommended results should be in accordance with the selections user made
Verify filtering options for recommendations (Optional)	Application functionality designed for filtering	User should have options to filter the recommended results (if implemented)
Verify model prediction updates based on user input	Application accessible and trained model	KPIs (if applicable) should modify as per the user inputs for mushroom classification
Verify result details for the classification	Application accessible and trained model	The results should indicate details of the predicted mushroom class (edible or poisonous)

