

Low Level Design (LLD)

Mushroom Type Classifier

Revision No: 2.0

Last Date of Revision: 29/10/2023.

Document Version Control-

<u>Date Issued</u>	<u>Version</u>	<u>Description</u>	<u>Author</u>
26/05/2023	1.0	LLD – V- 1.0	Mahesh. A
29/10/2023	2.0	LLD – V- 2.0	Mahesh. A

Contents -

Description	Page No.
Document Version Control	2
1. Introduction	4
1.1 What is Low-Level design document	4
1.2 Scope	4
2. Architecture	4
3. Architecture Description	5
3.1 Data Accessing	5
3.2 Data Pre-Processing	5
3.3 Splitting the Data	5
3.4 Model Building	5
3.5 Create Front End User Module using flask	5
3.6 Testing the Model	6

1 Introduction

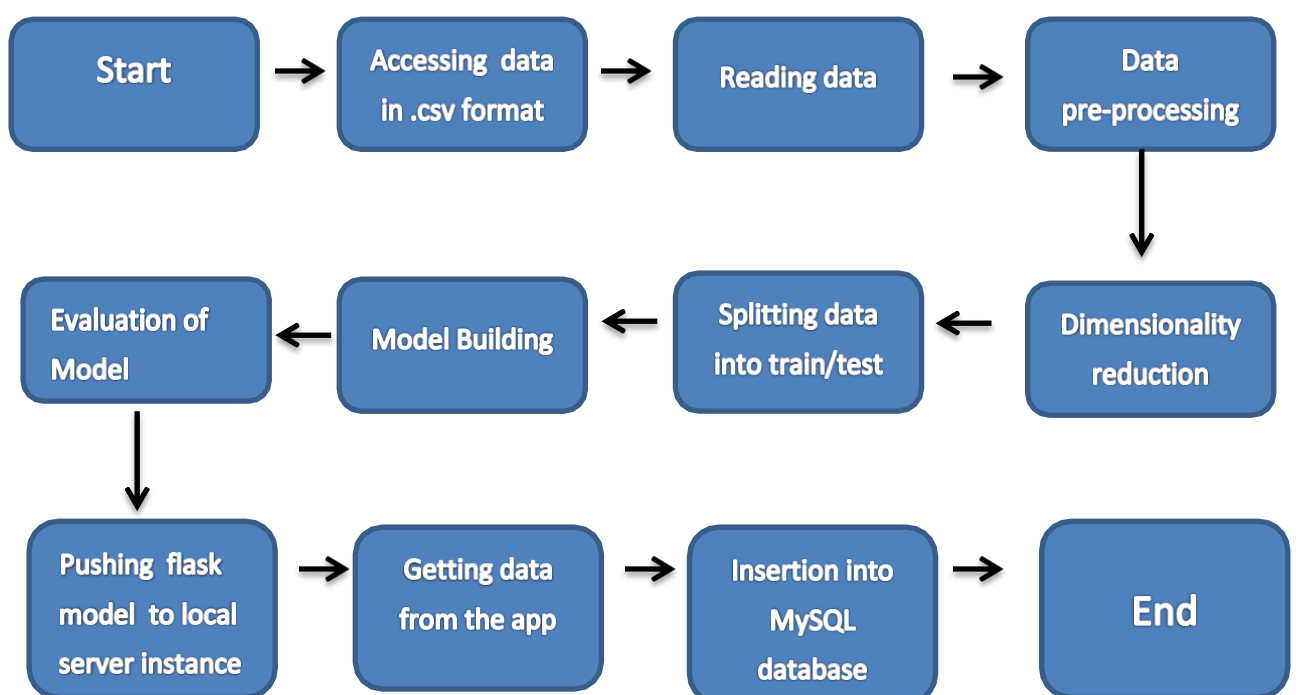
1.1. What is Low-Level design document?

The goal of LLD or a low-level design document (LLDD) is to give the internal logical design of the actual programmed code for Mushroom classifier. LLD describes the class relations with predictors. 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 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 Accessing

We can access the data in the from the download link as available in the project.

We load the data to the framework using the pandas read function.

3.2 Data Pre-Processing

By the usage of the different data manipulation techniques we will remove unwanted features

Also we use the dimensionality reduction and make all the features in numerical data type

3.3 Splitting the Data

We use train test split Sklearn function to split the data for training and validation

3.4 Model Building

We will deploy as many models as possible and fine tune them using GridsearchCV select the model which are performing with highest accuracy and select those models and ensemble them using voting classifier for the best model. Perform the model evaluation.

3.5 Create Front End User Module using flask

Once the model is created download and save the model and now we create GUI for front end user using the flask incorporated with HTML, CSS. Align and map the user data to the data base created. From user data create the data frame and load it to the model for the prediction the same prediction is send back to the user GUI and well saved in the data base (MySQL).

3.6 Testing the Model

- ✓ Verify whether the application is the loading on the local server instance.
- ✓ Verify whether the user can access the application.
- ✓ Verify the user can access the different fields for selection and can be visible
- ✓ Once the user selection the fields and made the submit
- ✓ Check the user can get the result or prediction.
- ✓ Once he gets the prediction.
- ✓ Check the data form the user and prediction from the model is loaded into the local MySQL and Cassandra database
- ✓ Check the database and download the data...