Low Level Design

Mushroom Classification

Written by	Chandru M
Document version	1.0
Last Revision Date	11/02/2023

Document Control

Date	Version	Author
20/06/24	Version-1.0	

Contents

Page no.

1. Introduction

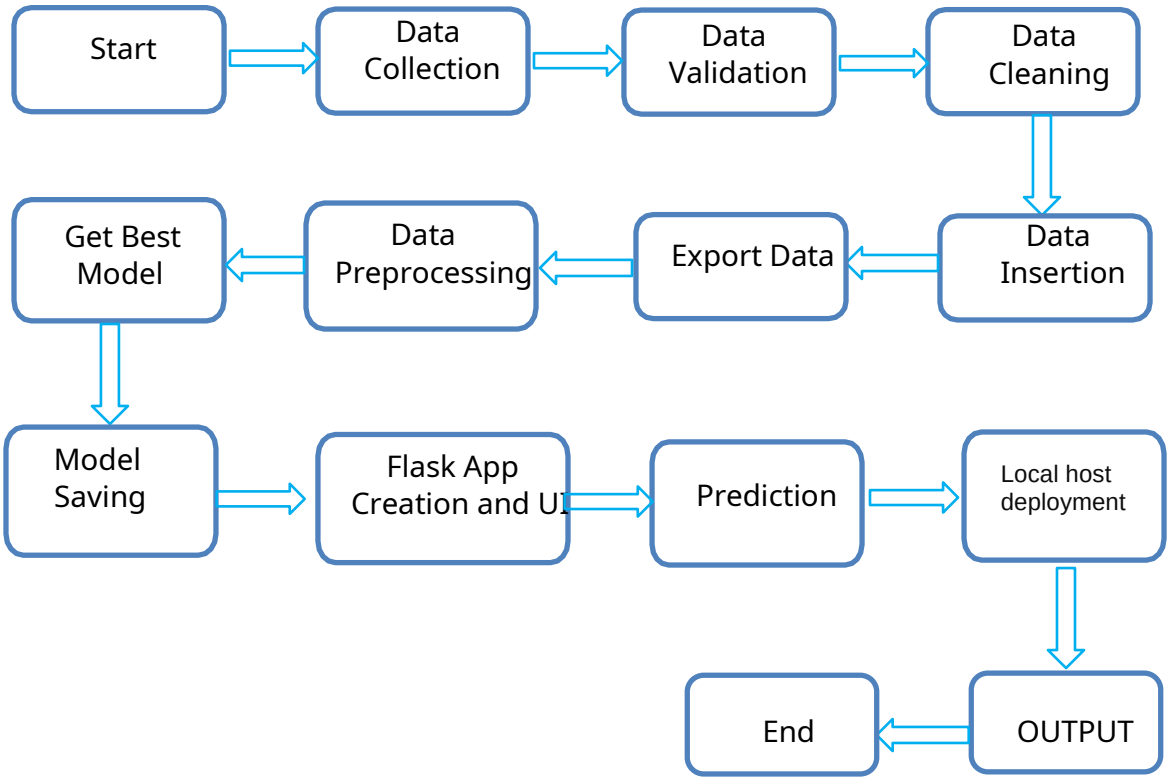
1.1 What is Low Level Design Document?

Giving the internal logical design of the actual programme code is the aim of a low-level design document (LLD). Based on the high-level design, the low-level design is produced. LLD explains class diagrams that show the relationships ar methods between classes and programme specifications. In order for the programmer to create the programme directly from the document, it describe the modules.

1.2 Scope

Low-level design (LLD) is a component-level design method that incorporates constant iteration and improvement. Data structures, necessary software architecture, source code, and finally performance algorithms can all be designed using this method. Overall, during requirement analysis, the data organisation may be created, and then refined, during data design work. Each component is precisely specified after the build.

2 Architecture



3 Architecture Description

3.1 Data Description

From The Audubon Society Field Guide to North American Mushrooms, this dataset comprises descriptions of hypothetical samples belonging to 23 species of gilled mushrooms in the Agaricu and Lepiota Family Mushroom (1981). Each species is classified as either unquestionably edible, unquestionably poisonous, or maybe edible but not advised.

3.2 Import Data

Data Import - The data in a stored database is imported as a CSV file to be used for Data Pre-processing and Model Training.

3.3 Data Cleaning

There are no null values in the data and all the variables are categorical, some of the observation meaningless and they are converted into meaningful observation.
Example: "?" is converted into letter "m" (Missing)

3.4 Exploratory Data Analysis

Every independent variable in the dataset is displayed by a multiple bar plot as it relates to the dependent variable's classes of poisonous and edible mushrooms as part of the EDA process.

3.5 Data Preprocessing

Data preprocessing steps are converting categorical variables into numerical variables using lab encoding method and train and test split of the data etc.

3.6 Model Building

After Data preprocessing split the data train and test (Simple Random Sampling) and implemented different Classification Machine Learning Algorithm. XGBoost model gives the better accuracy.

3.7 Model Dump

I developed a model and used the pickle module to dump the model in a pickle file format after comparing all accuracy levels and determining the optimal model for the dataset.

3.8 Data from User

Here, using the UI interface, the user must input all the feature values in the proper order before sending it to the model. The model, which will determine whether the feature set accura represents the desired features, will be fed the data.

3.9 Data Validation

Here Data Validation will be done, which has given by the user.

3.10 Model Call for specific input

Based on the user's input, the data will be handled in the backend with a variable format before being converted into a NumPy array and provided to an ML model. The model will check the inp against the required criteria after loading the pickle file by sending the output to our html page.

3.11 User Interface

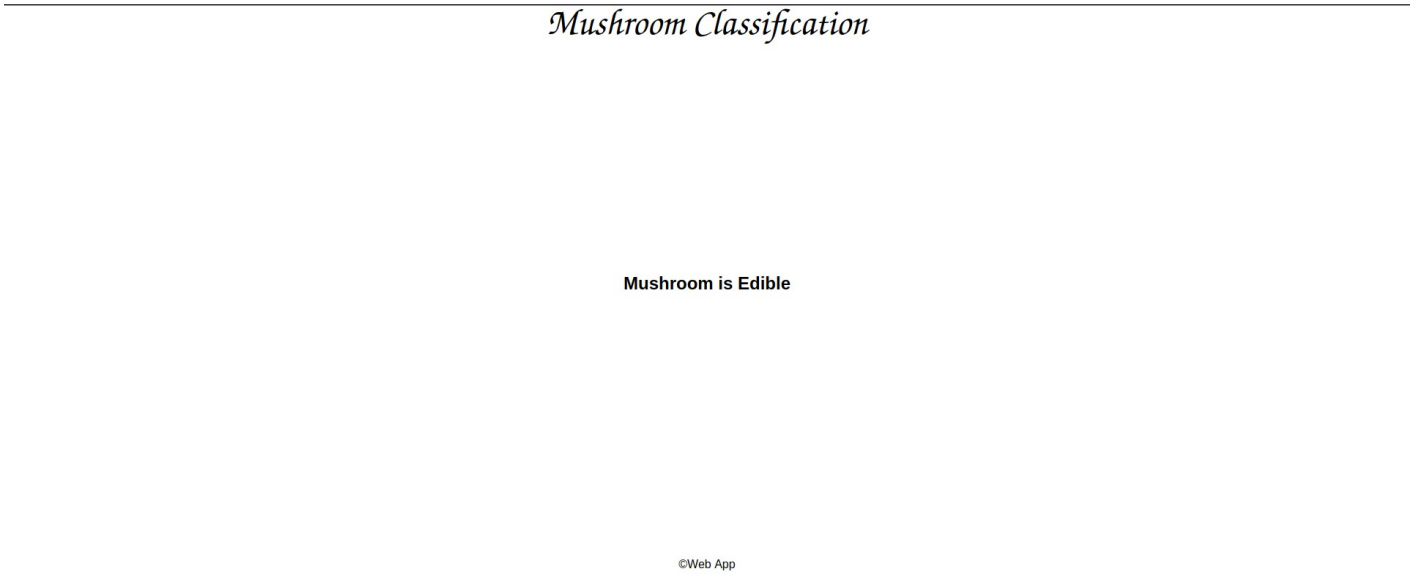
I created a user-interactive page for the frontend where users can input values for the programme on th frontend page. a website with a visually appealing layout created with CSS. This HTML user input data is sent in a decoupled format in a variable form to the backend.

Input Page:


Mushroom Classification
flat
smooth
gray
true
anise
attached
crowded
narrow
chocolate
enlarging
missing
silky
silky
red
gray
white
two

Output Page: If mushroom is edible

If mushroom is poisonous



3.12Deployment

The deployment of the model is done in local host.

4.Technology Stack

Front End	HTML/CSS
Back End	Flask, Pandas, NumPy, Sci-kit learn, etc
Deployment	Local host