Mushroom Classifier (LLD)

Low Level Document (LLD)

Mushroom Classifier (LLD)

Written by	Mohammad Iqbal
Document Version	0.1
Last Revise Date	30/12/2023

Document Version Control

Change Record:

Version	Date	Author	Comments
0.1	30-Dec-23	Suraj Mali	Initial LLD

Reviews:

Approval Status:

Version	Review Date	Review by	Approved by	Comments

1. Introduction

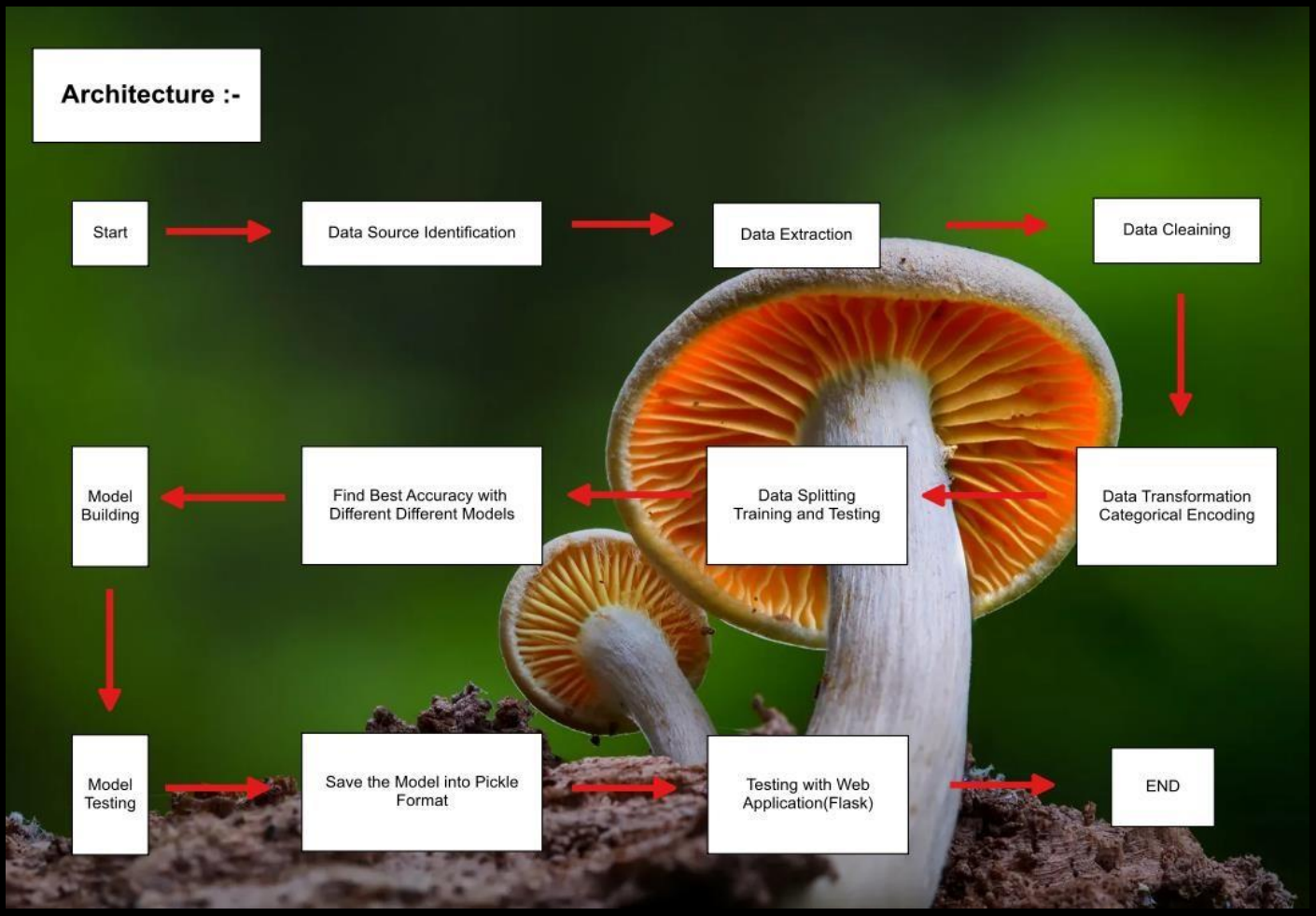
1.1 What is Low-Level Design Document?

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 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 stepby-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

Mushroom toxicity is the open-source dataset available on the internet. Each row contains mushroom class along with its categories of its features in the form of Comma Separated Variable (CSV) file. This dataset contains 8124 different mushrooms including 22 different features.

3.2 Data ingestion

Data is injected with reading csv format by using Pandas Data Frame.

3.3 Data Preprocessing

Data is pre-processed by handling null values, duplicates values, checking balance of data etc.

3.4 Data Transformation

Data is transformed into scaler form by applying encoding techniques & feature engineering.

3.5 Model Building

Once, Data is transformed, it is trained with different categories of classifier models like Logistic Regressor, Decision Tree, CV etc. After results, we will compare with accuracy score on the testing set.

3.6 Model Evaluation

Once, model is tested, model with highest accuracy score will be store in database & will predict output based on user given data.

3.7 Data from User

As a trained model, it is capable of predicting the type of mushroom based on the data provided by the user. It will work in the background and provide the user with the corresponding output.

3.8 Model Deployment

Model can deploy on local host server or cloud platform like AWS, Azure etc.

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 see input fields on logging in	1. Application is accessible 2. User is log in the application	User should be able to see input fields on logging in
Verify whether user is able to edit all input fields	1. Application is accessible 2. User is log in to 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 is log in to application	User should get submit button to submit the inputs
Verify whether user is able to get the output on submitting the results	1. Application is accessible 2. User is log in to application 3. User has submitted the results	User should get the output after submit has been sent.