

**Mushroom Classification**

**Low Level Design**

***Domain : Machine Learning***

***Created By : Biswajit Swain***

***Date : 01.03.2023***

**Document Version Control**

|  |  |  |  |
| --- | --- | --- | --- |
| Date | Version | Description | Author |
| March 1, 2023 | 1.1 | First Draft | Biswajit Swain |
| March 5, 2023 | 1.2 | Added unit test cases | Biswajit Swain |

**CONTENTS**

**[Introduction 4](file:///C:\\Users\\user\\Downloads\\LLD.docx" \l "_Toc110433758)**

**[What is Low-Level Design Document? 4](file:///C:\\Users\\user\\Downloads\\LLD.docx" \l "_Toc110433759)**

**[Scope 4](file:///C:\\Users\\user\\Downloads\\LLD.docx" \l "_Toc110433760)**

**[Architecture 5](file:///C:\\Users\\user\\Downloads\\LLD.docx" \l "_Toc110433761)**

**[Architecture Description 6](file:///C:\\Users\\user\\Downloads\\LLD.docx" \l "_Toc110433762)**

**[Data Preparation 6](file:///C:\\Users\\user\\Downloads\\LLD.docx" \l "_Toc110433763)**

**[Data Description 6](file:///C:\\Users\\user\\Downloads\\LLD.docx" \l "_Toc110433764)**

**[Data Preprocessing 6](file:///C:\\Users\\user\\Downloads\\LLD.docx" \l "_Toc110433765)**

**[Exploratory Data Analysis 6](file:///C:\\Users\\user\\Downloads\\LLD.docx" \l "_Toc110433766)**

**[Feature Engineering 6](file:///C:\\Users\\user\\Downloads\\LLD.docx" \l "_Toc110433767)**

**[Model Development 7](file:///C:\\Users\\user\\Downloads\\LLD.docx" \l "_Toc110433768)**

**[Model implementation 7](file:///C:\\Users\\user\\Downloads\\LLD.docx" \l "_Toc110433769)**

**[Model Evaluation 7](file:///C:\\Users\\user\\Downloads\\LLD.docx" \l "_Toc110433771)**

**[Deployment 7](file:///C:\\Users\\user\\Downloads\\LLD.docx" \l "_Toc110433772)**

**[Designing UI 7](file:///C:\\Users\\user\\Downloads\\LLD.docx" \l "_Toc110433773)**

**[Designing a server 7](file:///C:\\Users\\user\\Downloads\\LLD.docx" \l "_Toc110433774)**

**[Code deployment on cloud 8](file:///C:\\Users\\user\\Downloads\\LLD.docx" \l "_Toc110433775)**

**[Deployment Process 8](file:///C:\\Users\\user\\Downloads\\LLD.docx" \l "_Toc110433776)**

**[Unit cases 9](file:///C:\\Users\\user\\Downloads\\LLD.docx" \l "_Toc110433777)**

INTRODUCTION

# What is Low Level Design Document?

The goal of LLD or a low-level design document is to give the internal logical of the actual program code for Mushroom Type Prediction. It will explain the purpose and features of the system, the interfaces of the system, what the system will do, the constraints under which it must operate and how the system will react to external stimuli.

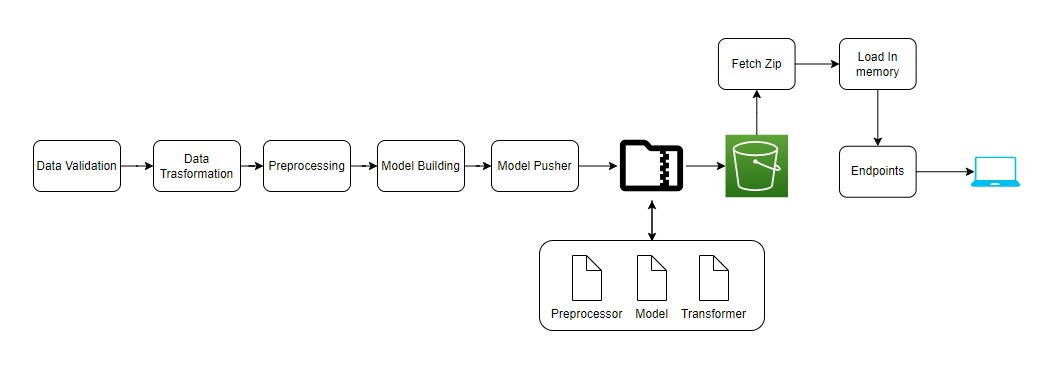
The main objective of this project is to predict the mushroom type wheather it is edible or poisonous.

# Scope

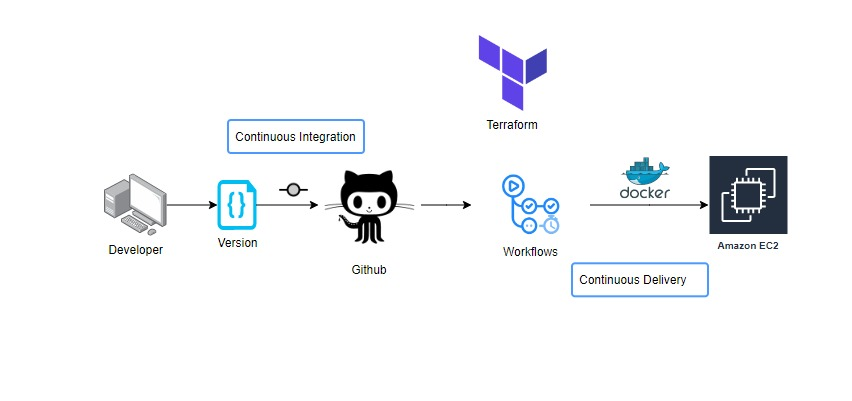
Low-level design (LLD) is a component-level design process that follows a step-by-step refinement process. This process can be used for designing data structures, required software architecture, source code and ultimately, performance algorithms. Overall, the data organization may be defined during requirement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

# *Architecture*

# Project Architecture



# Deployment Architecture



# *Architecture Description*

# Data Preparation

# Data Description

The Audubon Society Field Guide to North American Mushrooms contains descriptions of hypothetical samples corresponding to 23 species of gilled mushrooms in the Agaricus and Lepiota Family Mushroom (1981). Each species is labelled as either definitely edible, definitely poisonous, or maybe edible but not recommended. This last category was merged with the toxic category. The Guide asserts unequivocally that there is no simple rule for judging a mushroom's edibility, such as "leaflets three, leave it be" for Poisonous Oak and Ivy.The main goal is to predict which mushroom is poisonous & which is edible.

# Data Preprocessing

In data preprocessing step, we check if there missing data, duplicate values, and datatypes of each feature. In our dataset, there was not any null and duplicate values .

# Exploratory Data Analysis

This step includes bivariate and univariate analysis of features. Checking outliers using boxplots, and outlier treatment is carried out as well. Distribution of the features are plotted to see to what extent our data is skewed.

# Feature Engineering

In this part, the datatypes of the features were checked whether it belongs same datatypes or different datatypes. Outliers were checked using boxplot but there is no such major outliers in the dataset.

# *Model Development*

# Model Implementation

After train and test splitting, pipeline containing Standard Scaler and Label Encoder was fitted to several models such as RandomForest Classifier,XGB Classifier, KNeighbors Classifier, etc. Their f1 score were obtained and it was determined that KNeighbors Classifier performs better than other models.

# Model Evaluation

Test dataset is used to evaluate the model. 20% of dataset was separated for testing. Predicted results of the model are compared with the actual data to check the amount of error. As there was no considerable change after , it helped us to overcome overfitting and perform better on new data.

# Deployment

# Designing UI

For this project, a user interface is built using Apache Air Flow.It is a open-source platform for developing, scheduling, and monitoring batch-oriented workflows.

# Designing a server

A server should be created to run the UI application continuously. Amazon Web Service EC2 instance is used to create a virtual server for the application.Elastic Compute Cloud (EC2) is a virtual server in AWS for running applications on the AWS infrastructure.

**Code deployment on cloud**

The codes for this machine learning model should be deployed to the cloud, so that when data is entered into the application, our code runs, and a user gets the result online.

# Deployment Process

In this stage, we containerized the code using Docker and will be deploying the model to AWS.This is a workflow diagram for the Recipe Recommendation..

# Unit Cases

|  |  |  |
| --- | --- | --- |
| **Test Case Description** | **Pre-Requisite** | **Expected Result** |
| Verify whether the Application URL is accessible to the user | 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. | Application is accessible | User should be able to see input fields |
| Verify whether user is able to edit all input fields | Application is accessible | User should be able to edit all input fields |
| Verify whether user gets Submit button to submit the inputs | Application is accessible | User should get Submit button to submit the inputs |
| Verify whether user is presented with results on clicking submit | Application is accessible | User should be presented with results on clicking submit |
| Verify whether the results are in accordance to the selections user made | Application is accessible | The results should be in accordance to the selections user made |