High-Level Design (HLD)

Mushroom Classification

Written By	Ayush Wase
Document Version	HLD - V.1.1
Last Revised Date	03-June-2023

Document Version Control

Version	Date	Author	Comments
HLD - V.1.0	24-May-2023	Ayush Wase	Initial HLD - V.1.0
HLD - V.1.1	03-June-2023	Ayush Wase	Document updated

Contents

ABSTRACT

In the age of the internet, the unprecedented volume of data surpasses human capacity for processing. To address this challenge, machine learning techniques have emerged as powerful tools. Our project focuses on the classification of mushrooms as either edible or not edible using machine learning algorithms. By implementing various classification techniques, we aimed to determine the best algorithm for accurately predicting the edibility of mushrooms. Through extensive experimentation, we found that decision tree algorithms provided accurate and reliable results in classifying mushrooms. This work contributes to the field of mushroom classification, highlighting the effectiveness of machine learning algorithms in automating the identification of edible and non-edible mushrooms.

1. Introduction

1.1 Why this High-Level Design Document?

The purpose of this High-level Design (HLD) document is to add the necessary detail to the current project description to represent a suitable model for coding. This document is also intended to help detect contradictions prior to coding and can be used as a reference manual for how the modules interact at a high level.

The HLD will:

- Present all of the design aspects and define them in detail
- Describe the user interface being implemented
- Describe the performance requirements
- Include design features and the architecture of the project
- List and describe the non-functional attributes like:
    - Security
    - Relatability
    - Maintainability
    - Portability
    - Reusability
    - Application Compatibility
    - Resource utilization
    - Serviceability

1.2 Scope

The HLD documentation presents the structure of the system, such as the database architecture, application architecture (layers), application flow (navigation), and technology architecture. The HLD uses non-technical to mildly-technical terms which should be understandable to the administrators of the system.

2. General Description

2.1 Product Perspective

Mushroom Classification is a solution that accurately predicts the edibility of mushrooms based on a provided dataset, using advanced machine learning algorithms.

2.2 Problem Statement

The accurate identification of mushroom edibility is crucial for ensuring food safety and preventing potential health risks. However, manually distinguishing between edible and non-edible mushrooms can be challenging, as it requires expert knowledge and can be time-consuming. There is a need for an automated solution that can efficiently classify mushrooms based on their edibility, providing a reliable and convenient tool for consumers, mushroom enthusiasts, and professionals in the field.

2.3 Proposed Solution

our proposed solution for mushroom classification will involve an intuitive interface where users can input the characteristics of a mushroom. These characteristics may include cap shape, colour, odour, and other relevant attributes. Leveraging a trained machine learning model, specifically a decision tree algorithm, the system will analyse the provided features and promptly classify the mushroom as either edible or non-edible.

2.4 Data Requirements

The data required for the building of the project is already available on the dashboard. The Store Sales Prediction data recorded many product descriptions along with past sales quantity. For building the ml model we will use the dataset that is given. The data consists of 8523 rows and various information about products like product id, product category, store id, store location, etc.

2.4 Further Improvements

In future developments, we can utilize user input history to improve the accuracy of the mushroom classification product. By analysing users' past inputs and feedback, we can gather valuable insights to enhance our dataset and refine the classification model. Additionally, exploring the possibility of incorporating image classification capabilities would offer users a more intuitive and user-friendly experience. These potential enhancements aim to continuously improve the reliability and effectiveness of the mushroom classification solution.

2.5 Technical Requirements

The proposed solution for the mushroom classification can be deployed either as a cloud-based application, hosted on an internal server, or run on a local machine. To access the application, users will require a stable internet connection and a standard web browser.

For training the classification model, the following system requirements are recommended:

- +4 GB RAM (or higher) for efficient processing and training.
- Operating System: Windows, Linux, or Mac, providing flexibility for various platforms.
- Development Environment: Visual Studio Code or Jupyter Notebook, which are popular and user-friendly code editors suitable for machine learning tasks.

By meeting these minimum system requirements, users can effectively utilize the mushroom classification solution and train the model to achieve accurate and reliable results.

2.6 Data requirements

Data requirements completely depend on our problem statement.

For the mushroom classification model, we have utilized a dataset in CSV(comma separated value) format with 4874 instances and 5 columns. The dataset includes the following attributes for each mushroom:

- Gill Size: Describes the size of the mushroom's gills, indicating whether they are broad or narrow.
- Gill Colour: Specifies the colour of the mushroom's gills, indicating various color options such as white, brown, or black.
- Stalk Root: Describes the type of root structure of the mushroom's stalk, which can be categorized as bulbous, club, equal, rooted, or missing.
- Spore Print Colour: Indicates the colour of the mushroom's spore print, representing colours such as black, brown, purple, or white.
- Population: Describes the population of mushrooms in a specific area, indicating various levels such as abundant, clustered, numerous, scattered, or several.

By training the mushroom classification model on this dataset and utilizing these specific attributes, we aim to accurately classify mushrooms as either edible or non-edible based on their gill size, gill colour, stalk root type, spore print colour, and population characteristics.

2.7 Tools Used

Python Programming language and frameworks such as Numpy, Pandas, Scikit-learn, Microsoft Azure are used to build the whole model.

- VS code and Google Colab is used as IDE.
- For visualization of the plots Matplotlib, and Seaborn are used.
- Flask is used for the deployment of the model.
- Front end Development is done using HTML/CSS.
- Python is used for backend development.
- Github is used as a version control system.

2.8 Constraints

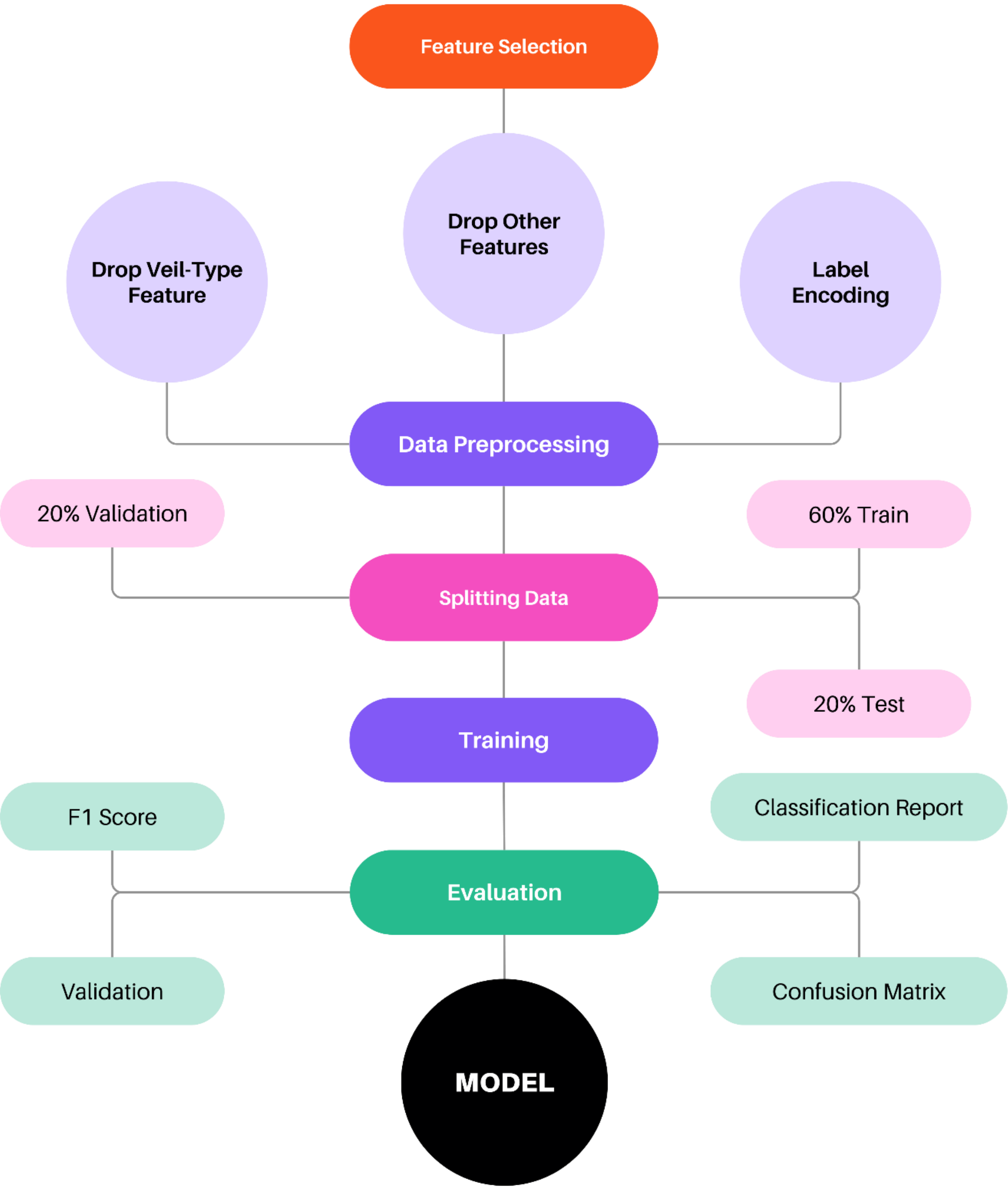
The Mushroom Classification system must be user-friendly, and as automated as possible and users should not be required to know any of the workings.

3. Design Details

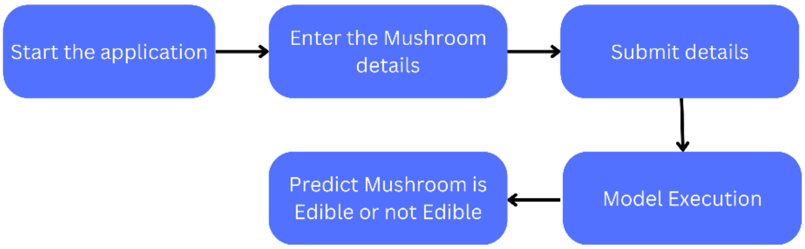
3.1 Process Flow

For Identifying the different types of anomalies, we will use a machine learning model. Below is the process flow diagram as shown below.

3.2.1 Model Training and Evaluation



3.2.2 Deployment Process


Start the application
Enter the Mushroom details
Submit details
Model Execution
Predict Mushroom is Edible or not Edible

4. Performance

The Performance of the model depends on the dataset. We have done in-depth pre-processing of the dataset for greater accuracy and much closer prediction with less error.

4.1 Reusability

The code and modules utilized for the mushroom classification project follow coding guidelines and maintain a modular structure, ensuring reusability. The system provides flexibility by functioning effectively from any location. It includes robust input validation mechanisms to handle improper user inputs and deliver meaningful error messages for corrective actions. The system's versatility enables it to be used with various types of input values, leveraging its comprehensive training. These features enhance the overall usability and effectiveness of the solution.

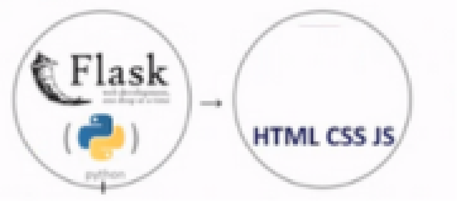
4.2 Application Compatibility

This Project will be using Python as an interface between them. Each Component will have its own task to perform, and it is the job of the python to ensure proper transfer of information.

4.3 Resource Utilization

When any task is performed, it will likely use all the processing power available until that function is finished

4.4 Deployment



5. Conclusion

The mushroom classification Model offers a versatile and reusable system that accurately predicts the edibility of mushrooms based on their distinct attributes.