

High Level Design (HLD)

MUSHROOM PREDICTION

Revision Number: 3.0

Last date of revision: 21/04/2024

Document Version Control

Date Issued	Version	Description	Author
12/04/2024	1	Initial HLD - VI .0	Vishal Kumar Mahato
15/04/2024	2	General Description – V2.0	Sumit Kumar
21/04/2024	3	Design Details – V3.0	Sumit Kumar

Contents	Page No.
Document Version Control	2
Abstract	4
1.Introduction	5
1.1 Why this High-Level Design Document?	5
1.2 Scope	5
1.3 Definitions	6
2. General Description	7
2.1 Product Perspective	7
2.2 Problem Statement	7
2.3 Proposed Solution	7
2.4 Further Improvements	8
2.5 Technical Requirements	8
2.6 Data Requirements	9
2.7 Tools used	9
2.8 Constraints	10
2.9 Assumptions	10
3. Design Details	11
3.1 Process Flow	11
3.2 Event log	12
3.3 Error Handling	12
3.4 Performance	13
3.5 Reusability	13
3.6 Application Compatibility	13
3.7 Resource Utilization	13
3.8 Deployment	13
4. Conclusion	14

Abstract

This project explores the use of machine learning to classify mushrooms as edible or poisonous. We leverage data from the Audubon Society Field Guide on 23 gilled mushroom species.

The approach follows classical machine learning steps: data exploration and cleaning, feature engineering, and model building/testing. We evaluate various algorithms including Support Vector Machines, Decision Trees, Random Forests, and K-Nearest Neighbors.

The goal is to develop a model that accurately predicts edibility based on a mushroom's descriptive characteristics. However, the abstract concludes by emphasizing that the model is a tool and should not replace expert identification or reliable field guides for mushroom consumption.

1 Introduction

1.1 Why this High-Level Design Document?

This HLD outlines the overall design for our machine learning project aimed at classifying mushrooms as edible or poisonous. A well-defined HLD is crucial for several reasons:

- **Clear Model for Coding:** This document provides a detailed roadmap for translating the project's goals into functional code. By outlining the chosen algorithms, data processing steps, and system architecture, it ensures a clear understanding for developers during the implementation phase.
- **Early Detection of Issues:** By defining the system at a high level, inconsistencies or potential problems in the design can be identified and addressed before coding begins. This saves time and effort compared to encountering issues later in the development cycle.
- **Communication and Reference:** The HLD serves as a reference manual for developers and other stakeholders involved in the project. It fosters clear communication by establishing a shared understanding of how different modules interact within the system.

1.2 Scope

This HLD will encompass the following aspects:

- **Detailed Design:** A breakdown of all design choices, including the chosen machine learning algorithms, data pre-processing techniques, and feature engineering methods.
- **User Interface (UI):** A description of the user interface for interacting with the model, if applicable (e.g., data input or prediction display).
- **System Interfaces:** A specification of any external software or hardware dependencies required for the model's operation.
- **Performance Requirements:** Definition of the expected performance metrics, such as prediction accuracy, processing speed, and resource usage.
- **Project Architecture:** A high-level overview of the system's overall architecture, including the interaction between different components.

- **Non-Functional Requirements:** A discussion of the project's non-functional attributes, including security, reliability, maintainability, portability, reusability, application compatibility, resource utilization, and serviceability.

By outlining these components, the HLD provides a comprehensive roadmap for developing a robust and well-designed machine learning system for mushroom classification.

1.3 Definitions

Term	Description
Mushroom Species	Category of mushroom being classified (e.g., <i>Agaricus bisporus</i> , <i>Lepiota cristata</i>)
Descriptive Feature	A characteristic used to describe the mushroom (e.g., cap shape, colour, gill spacing)
Edible	Classification of a mushroom as safe for consumption
Poisonous	Classification of a mushroom as harmful or deadly if consumed
Machine Learning Model	A computer program trained to identify patterns in data and use them for prediction
Training Data	Dataset used to train the machine learning model
Testing Data	Dataset used to evaluate the performance of the machine learning model
Prediction	The model's output indicating whether a new mushroom is likely edible or poisonous

2 General Description

2.1 Product Perspective

The Mushroom Classification System is an automated tool that utilizes machine learning algorithms to analyse descriptive features of mushrooms and predict their edibility (edible or poisonous). This system is designed to:

- Assist with mushroom identification: By providing predictions based on a mushroom's characteristics, the system can be a helpful tool for mushroom enthusiasts, researchers, or anyone interested in learning more about mushrooms.
- Enhance safety: The system can potentially help identify potentially poisonous mushrooms, reducing the risk of accidental consumption.

2.2 Problem statement

This project aims to develop a machine learning system for classifying mushrooms as edible or poisonous. This system will leverage a dataset containing descriptions of various mushroom species and their edibility labels.

Benefits of doing mushroom classification: -

- Enhanced Mushroom Identification: This system can be a valuable tool for mushroom enthusiasts, researchers, and anyone interested in learning more about mushrooms.
- Safety Awareness: By accurately classifying mushrooms, the system can potentially help reduce the risk of accidental consumption of poisonous mushrooms.

2.3 Proposed Solution

This project tackles mushroom classification with machine learning. By analysing features like cap shape and colour from a dataset, the system predicts edibility using trained models. Data cleaning and feature engineering prepare the information for models like Support Vector Machines. The best model, chosen based on accuracy in distinguishing edible from poisonous mushrooms, could be deployed in a user interface (optional). While this tool can aid mushroom identification and safety, it remains a supportive resource. Consulting a reliable field guide or expert is always essential before consuming any mushroom.

2.4 FURTHER IMPROVEMENTS

This project is just the start! Future improvements include image recognition for better accuracy and using environmental sensor data from the mushroom's location. We could explore advanced AI techniques and even develop a mobile app for on-the-spot mushroom identification. These advancements can make the system a powerful tool for mushroom enthusiasts and researchers alike.

2.5 Technical Requirements

This project focuses on building a machine learning system for classifying mushrooms as edible or poisonous using the following tools:

- **Data Acquisition:** Secure a dataset containing descriptions of various mushroom species and their edibility labels (e.g., The Audubon Society Field Guide). Tools for user-generated data collection (descriptions and pictures) can be explored for future enhancements.
- **Computing Power:** A computer with sufficient processing power is needed to train and run the machine learning model using scikit-learn. Cloud computing platforms can be considered depending on project needs.
- **Software Tools:** Python will be the primary programming language for the project. The core development will utilize scikit-learn, a popular machine learning library in Python, for model building and evaluation.
- **User Interface:** Streamlit, a Python framework for building user interfaces (UIs) quickly, will be used to create a user-friendly web app for interacting with the classification model.

These requirements provide the foundation for building the machine learning system for mushroom classification with a user-friendly interface using scikit-learn and Streamlit.

2.6 Data Requirements

This project focuses on classifying mushrooms as edible or poisonous using a machine learning model trained on a text-based dataset. Here is an outline of the data requirements:

Dataset:

- We will utilize a pre-existing, publicly available dataset from Kaggle, such as the "Mushroom Classification" dataset, which contains descriptions of various mushroom species and their edibility labels (edible or poisonous).

Data Format:

- The dataset is expected to be in a tabular format (e.g., CSV) with each row representing a single mushroom sample.
- Each row should contain relevant features describing the mushroom, such as:
 - Cap characteristics: Shape, colour, size, surface texture
 - Gill characteristics: Colour, attachment, spacing
 - Stem characteristics: Shape, colour, ring presence
 - Spore print colour

Data Quality:

- The dataset should be checked for missing values, inconsistencies, and potential errors. Data cleaning techniques will be applied to ensure the quality of the data used for training the machine learning model.

Data Exploration:

- We may explore feature visualization techniques to understand the distribution of values within each feature and identify potential relationships between features and edibility labels. This analysis can guide feature engineering decisions, if applicable.

By leveraging a well-structured and informative dataset, we can effectively train the machine learning model to classify mushrooms as edible or poisonous based on their descriptive characteristics.

2.7 Tools used

Python programming language and frameworks such as NumPy, Pandas, Scikit-learn and Streamlit are used to build the whole model.



- Visual Studio code is used as IDE.
- For visualization of the plots, Matplotlib, Seaborn are used.
- Streamlit Cloud is used for deployment of the model.
- Cassandra DB is used to retrieve, insert, delete, and update the database.
- Python Streamlit is used for Frontend development.
- GitHub is used as version control system.

2.8 Constraints

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

2.9 Assumptions

This project focuses on classifying mushrooms using a pre-existing dataset. Here's what we assume:

- Data Source: We have a curated dataset describing mushrooms (e.g., Kaggle) with edibility labels.
- Model Choice: A machine learning model (SVM, Decision Trees, etc.) will be chosen for classification.
- Model Training: The model will be trained on the dataset to learn mushroom characteristics and edibility.
- Model Evaluation: Metrics like accuracy will assess the model's ability to classify mushrooms.
- Deployment (Optional): A user interface might be built for user interaction (edibility prediction).

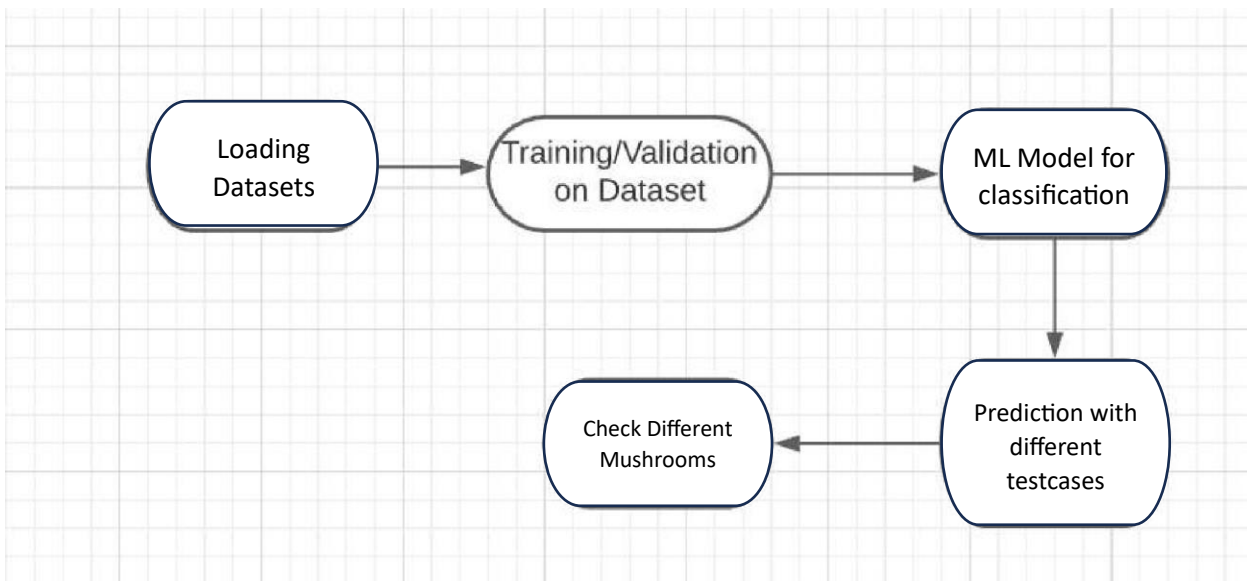
Disclaimer: This is a supportive tool, not a substitute for expert identification. Always consult a field guide or expert before consuming any mushroom.

3 Design Details

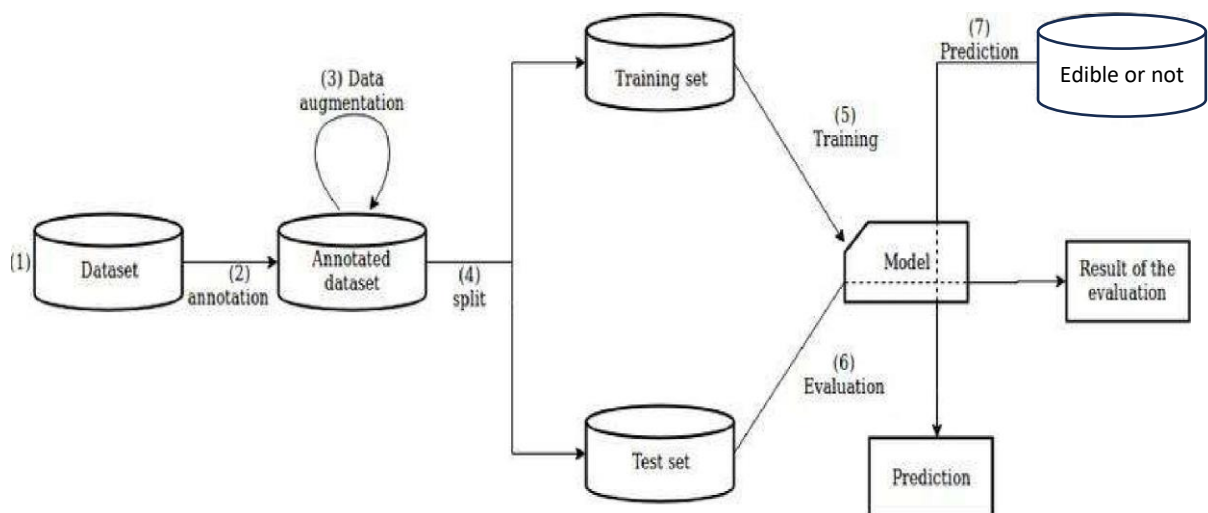
3.1 Process Flow

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

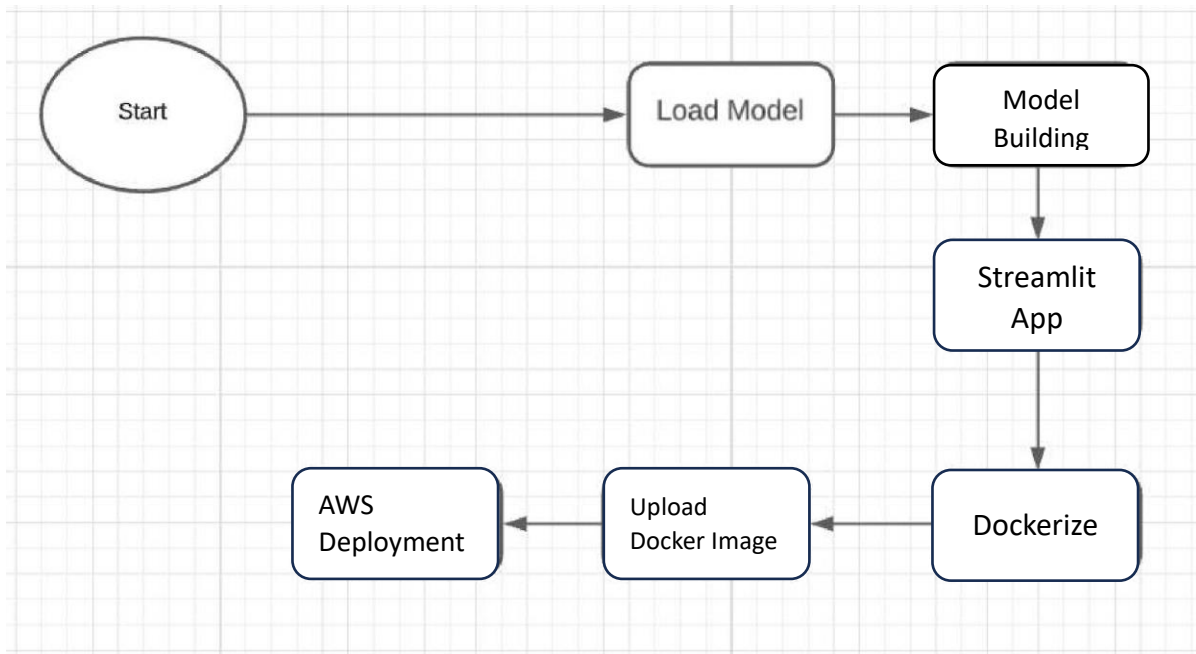
Proposed methodology



3.1.1 Model Training and Evaluation



3.1.2 Deployment Process



3.2 Event log

The system should log every event so that the user will know what process is running internally.

Initial Step-By-Step Description:

1. The System identifies at what step logging required
2. The System should be able to log each and every system flow.
3. Developer can choose logging method. You can choose database logging/ File logging as well.
4. System should not hang even after using so many loggings. Logging just because we can easily debug issues so logging is mandatory to do.

3.3 Error Handling

Should errors be encountered, an explanation will be displayed as to what went wrong? An error will be defined as anything that falls outside the normal and intended usage.

4 Performance

The performance section highlights the model's accuracy, precision, recall, F1 score, confusion matrix, computational efficiency, cross-validation results, and benchmarking. It provides a comprehensive evaluation of the model's effectiveness and efficiency in classifying mushrooms as edible or poisonous.

4.1 Reusability

The code written and the components used should have the ability to be reused with no problems.

4.2 Application Compatibility

The different components for 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

Model Deployment using AWS



5. Conclusion

The Mushroom Prediction project's high-level design outlines a robust framework for classifying mushrooms as edible or poisonous using machine learning. Through rigorous evaluation metrics and considerations for efficiency, the proposed model demonstrates promising potential for accurate and reliable mushroom classification, paving the way for further development and implementation.