Mushroom Classification

High-Level Document (HLD)

Mrinal Kanti Sikdar

Version 1.0, Date: 17/07/2023

ABSTRACT

Mushrooms have been consumed for centuries and are highly regarded for their nutritional value and medicinal properties. With their low-calorie, low-carbohydrate, and cholesterol-free nature, mushrooms have gained popularity as a valuable food source. They provide essential nutrients such as selenium, potassium, riboflavin, niacin, Vitamin D, proteins, and fiber. The historical significance of mushrooms as a staple food further emphasizes their importance. Additionally, mushrooms have demonstrated healing capabilities and have been utilized in traditional medicine, offering potential health benefits and treatment options for certain diseases. Notably, mushrooms exhibit various nutraceutical properties, including anti-cancer and anti-tumor attributes, while acting as antibacterial agents, immune system enhancers, and cholesterol-lowering agents. They also serve as a rich source of bioactive compounds.

In this machine learning project, we aim to develop a model that classifies mushrooms into two categories: poisonous and edible, based on their characteristic features. By employing state-of-the-art machine learning techniques, we will investigate the key features that contribute to accurately predicting whether a mushroom is toxic or safe for consumption.

The dataset used in this study encompasses a comprehensive collection of mushroom samples, encompassing a wide range of species and their associated features. These features include physical characteristics, chemical composition, and other relevant attributes that help distinguish between edible and poisonous mushrooms.

Through the utilization of machine learning algorithms, feature selection, and model training, we will identify the most significant features for accurate classification. Evaluation metrics such as accuracy, precision, recall, and F1-score will be employed to assess the model's performance and robustness.

The outcomes of this project will provide valuable insights into the essential features that influence the toxicity of mushrooms. This knowledge will contribute to improved safety practices, aiding in the identification of poisonous mushrooms and reducing the risks associated with their consumption. Furthermore, this research will lay the foundation for future investigations into the relationship between mushroom characteristics and their potential health benefits.

Keywords: Mushroom classification, machine learning, feature selection, toxicity prediction, nutritional composition, medicinal properties, nutraceuticals.

Contents

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 before 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 hardware and software interfaces
- Describe the performance requirements
- Include design features and the architecture of the project
- List and describe the non-functional attributes like: Security, Reliability, Maintainability, Portability, Reusability, Application compatibility, Resource utilization, and 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 Problem Perspective

Our project Mushroom Classification is a Machine Learning based model that classifies mushrooms into 2 classes: Poisonous and Edible.

2.2 Problem Statement

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 and which is edible.

2.3 Proposed Solution

To solve the problem, we have created a User Interface for selecting the input from the user to predict whether the mushroom is poisonous or edible using our trained ML model. After processing the input, the last output (predicted value) from the model is communicated to the User.

2.4 Further Improvements

We analyzed the given data and extracted features that are important in predicting whether a mushroom is poisonous or edible. We can consider all other available features as well.

However, it will result in a slower response on the web app that we have created. Also, since some of the characteristics of mushroom are same for poisonous and edible mushrooms, we recommend that User should also take help from someone who is expert in identifying edible mushrooms.

2.5 Technical Requirements

For technical requirements, we don't need any specialized hardware for virtualization of the application. The user should have a device that has the access to the web and a fundamental understanding of providing the input. And for the backend, we need a server to run all the required packages to process the input and predict desired output.

2.6 Data Requirements

A public dataset from Kaggle is used for our analysis purpose. This dataset was originally donated to the UCI Machine Learning repository. This dataset includes descriptions of hypothetical samples corresponding to 23 species of gilled mushrooms in the Agaricus and Lepiota Family Mushroom.

Dataset URL: https://www.kaggle.com/datasets/uciml/mushroom-classification

2.7 Constraints

The Mushroom Classification prediction answer should be user friendly, as automatic as attainable and also the user should not be needed to understand any of the operations.

2.8 Tools used

Python 3.8 is used as a programming language, Jupyter Notebook as IDE, Numpy, Pandas, seaborn, matplotlib, scikit learn for data preprocessing, data visualization and model building, HTML, CSS, Flask for creating app and deploying model.

Figure 1: Tools used for the project.

3 Design Details

For identifying different types of anomalies in our data and for data preprocessing, we will use a machine learning base model. Below are the different process diagrams explaining the various steps that are involved in complete execution of this project.

3.1 Model Training and Evaluation

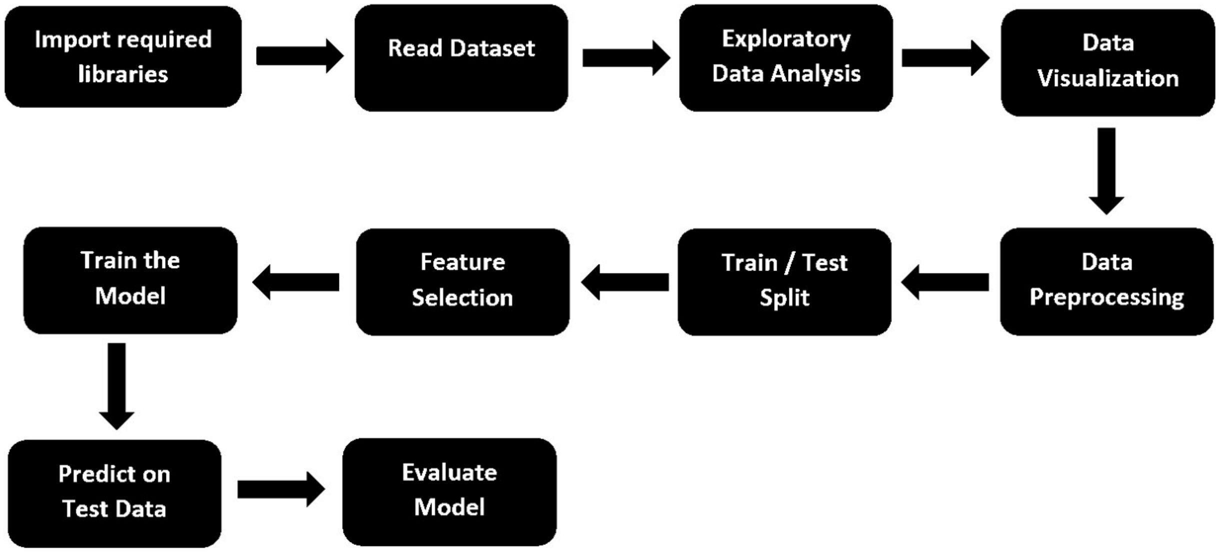

Figure 2: Model training and evaluation process.

3.2 Deployment Process

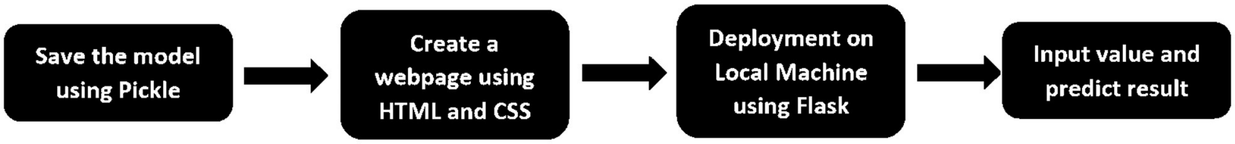
Save the model using Pickle
Create a webpage using HTML and CSS
Deployment on Local Machine using Flask
Input value and predict result

Figure 3: Deployment process.

3.3 Logging

In logging, each time an error or an exception occurs, the event is logged into the system log file with reason and timestamp. This helps the developer to debug the system bugs and rectify the error.

3.4 Error handling

Once the error occurs, the reason is logged into the log file with timestamp to rectify and handle it.

4 Performance

The Mushroom Classification solution is used for detecting whether a mushroom is poisonous or not, so it should be as accurate as possible. It is advised that a person should also take help from someone who is expert in identifying edible mushrooms since some of the characteristics of mushroom are same for poisonous and edible mushrooms.

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 parts of the system are communicating or using Python as an interface between them. All the components have its own tasks to perform and it is the job of a Python to ensure proper transfer of data.

4.3 Resource Utilization

When a task is performed, it'll doubtless use all the process power offered till the process is finished.

4.4 Deployment

The model can be deployed using any cloud services such as Microsoft Azure, Amazon web services, Heroku, Google cloud, etc.

5 Conclusions

The designed Mushroom Classification system will detect whether a mushroom is poisonous or not based on various features that are present in the data. Hence, we can easily identify whether a mushroom is poisonous or not and will certainly help people in selecting right type of mushroom to consume.

Mushroom Classification

Low-Level Document (LLD)

Mrinal Kanti Sikdar

Version 1.0, Date: 17/07/2023

ABSTRACT

Mushrooms have been consumed for centuries and are highly regarded for their nutritional value and medicinal properties. With their low-calorie, low-carbohydrate, and cholesterol-free nature, mushrooms have gained popularity as a valuable food source. They provide essential nutrients such as selenium, potassium, riboflavin, niacin, Vitamin D, proteins, and fiber. The historical significance of mushrooms as a staple food further emphasizes their importance. Additionally, mushrooms have demonstrated healing capabilities and have been utilized in traditional medicine, offering potential health benefits and treatment options for certain diseases. Notably, mushrooms exhibit various nutraceutical properties, including anti-cancer and anti-tumor attributes, while acting as antibacterial agents, immune system enhancers, and cholesterol-lowering agents. They also serve as a rich source of bioactive compounds.

In this machine learning project, we aim to develop a model that classifies mushrooms into two categories: poisonous and edible, based on their characteristic features. By employing state-of-the-art machine learning techniques, we will investigate the key features that contribute to accurately predicting whether a mushroom is toxic or safe for consumption.

The dataset used in this study encompasses a comprehensive collection of mushroom samples, encompassing a wide range of species and their associated features. These features include physical characteristics, chemical composition, and other relevant attributes that help distinguish between edible and poisonous mushrooms.

Through the utilization of machine learning algorithms, feature selection, and model training, we will identify the most significant features for accurate classification. Evaluation metrics such as accuracy, precision, recall, and F1-score will be employed to assess the model's performance and robustness.

The outcomes of this project will provide valuable insights into the essential features that influence the toxicity of mushrooms. This knowledge will contribute to improved safety practices, aiding in the identification of poisonous mushrooms and reducing the risks associated with their consumption. Furthermore, this research will lay the foundation for future investigations into the relationship between mushroom characteristics and their potential health benefits.

Keywords: Mushroom classification, machine learning, feature selection, toxicity prediction, nutritional composition, medicinal properties, nutraceuticals.

Contents

1 Introduction

1.1 Why this 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 Food Recommendation System. 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 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.

2 Architecture

This project is designed to make an interface for the user to predict whether a mushroom is poisonous or not.

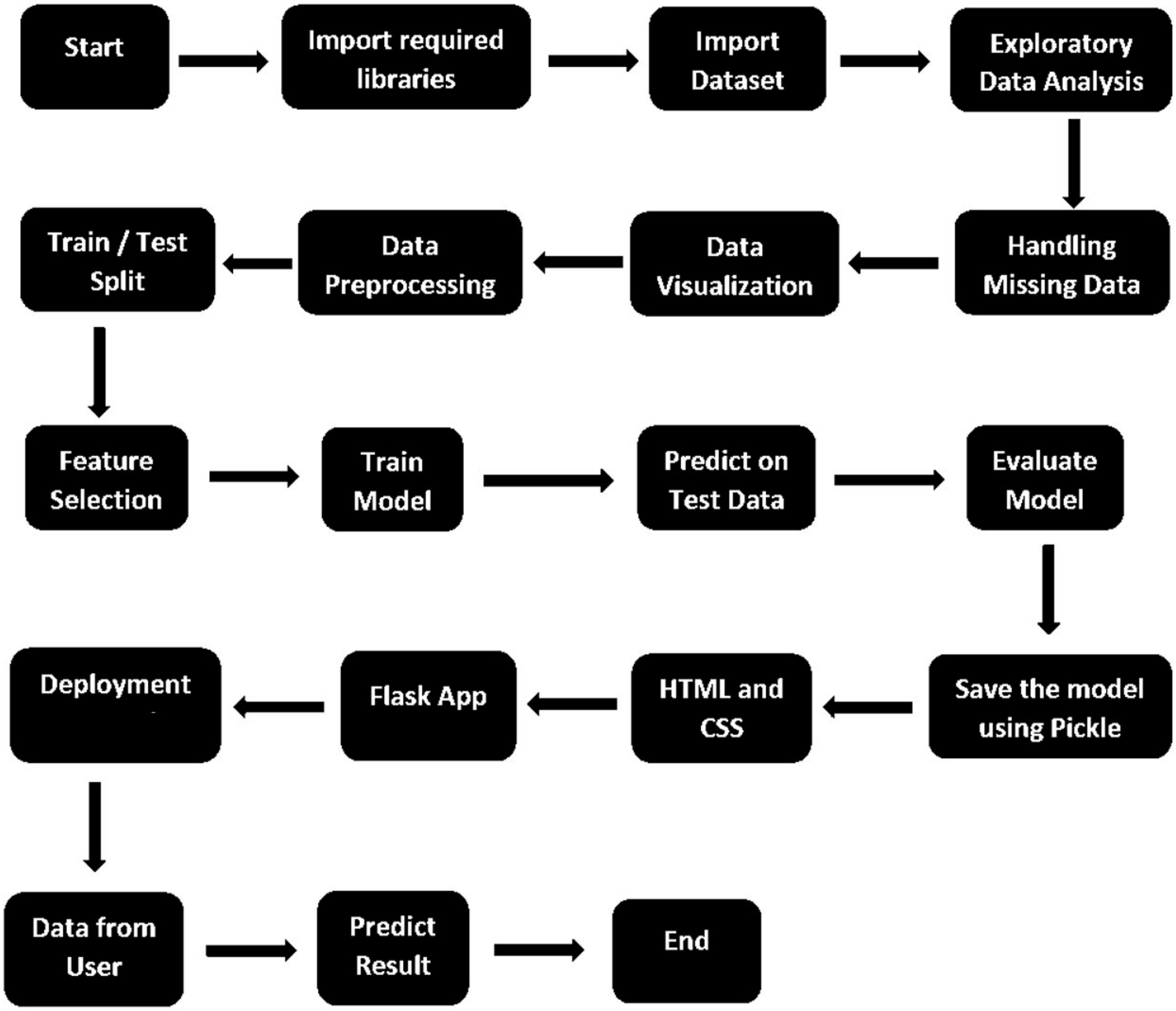

Figure 1: Architecture of the project.

2.1 Data Collection

The data for this project is collected from the Kaggle Dataset, the URL for the dataset is given below: Dataset: https://www.kaggle.com/datasets/uciml/mushroom-classification

2.2 Data Description

This dataset includes descriptions of hypothetical samples corresponding to 23 species of filled mushrooms in the Agaricus and Lepiota Family Mushroom drawn from The Audubon Society Field Guide to North American Mushrooms (1981). Each species is identified as definitely edible, definitely poisonous, or of unknown edibility and not recommended. This latter class was combined with the poisonous one.

2.3 Exploratory Data Analysis

The dataset is a csv file. There are 8124 rows and 23 columns in this data. All the columns are of categorical type. There are two classes present in our target column which are 'p' - poisonous and 'e' - edible. Also, we have nearly equal counts for poisonous and edible classes in our data. Hence, we can say that our data is balanced.

2.4 Handling Missing Data

At first, we observed that there no missing/null values in the dataset. However, if you go through the data description (check the link) you will find that the missing values in one column is replaced with "?". There are 2480 missing values in 'stalk-root' column. First, we will replace these values with np.nan so that we can handle missing data. we will impute the missing values in 'stalk-root' column using sklearn SimpleImputer with strategy='most frequent'.

2.5 Data Visualization

For data visualization, we have used only those columns which we found most relevant to the target column in the feature selection stage. We analyzed various count plots and gathered as much insights as we can about the target column.

2.6 Data Preprocessing

In this step, first we have dropped the column 'veil-type' as it has only one value throughout the data. So, it won't give us much information regarding the class of the mushroom. Next, we mapped our target column to 0 (poisonous) and 1 (edible) values. We used Label Encoder to convert categorical values to numerical then we scaled our data to bring them to same class.

2.7 Feature Selection

After splitting the data into train and test set, we used SelectKBest method with score func=chi2 to find out which features are most relevant to target column and we found that there are 12 columns out of 21 which we needed for training our model.

2.8 Model Training and Evaluation

We used XGBClassifier as a model for model training it was very fast compared to the other models and it produced 100% accuracy on train data as well as on test data which is a very good for our project.

2.9 Model Deployment

We created a webpage using HTML and CSS. We created a Flask web app and first tested in on our local machine. We used different combinations of input and predicted the output and the results were accurate. The app was working fine and there were no issues found. The model is finally deployed as a web application using AWS Elastic Beanstalk.

3 Unit Test Cases

Test Case Description	Pre - Requisites	Expected Results
Verify whether the Webpage is accessible to the User or not.	Webpage URL should be defined.	Webpage should be accessible to the User.
Verify whether the webpage loads completely for the User or not.	1. Webpage URL is accessible. 2. Webpage is deployed.	The Webpage should be able to load completely for the User when accessed.
Verify whether the user is able to select data in input fields or not.	1. Webpage URL is accessible. 2. Webpage is deployed. 3. Webpage input fields are editable.	The User is able to select data in input fields.
Verify whether the user is able to submit details or not.	1. Webpage URL is accessible. 2. Webpage is deployed. 3. Webpage input fields are editable.	The User is able to submit details to process.
Verify whether the user gets recommended results on submitting the details or not.	1. Webpage URL is accessible. 2. Webpage is deployed. 3. Webpage input fields are editable.	The User gets recommended results on submitting the details.