MUSHROOM CLASSIFICATION

High Level Design Document

Prepared by
GOURAB MAHAPATRA

Abstract

The Mushroom Classifier Prediction Project aims to develop a robust machine learning model capable of accurately identifying and classifying mushroom species based on their visual features. This project is motivated by the critical need for reliable tools to differentiate between edible and poisonous mushrooms, considering the potential life-threatening consequences of consuming toxic varieties.

Table of Contents

1. Introduction

1.1. What Is HLD?

The High-Level Design (HLD) Document serves several important purposes for the project. Its primary objective is to provide additional detail to the current project description, making it suitable for coding. By doing so, the document helps identify any contradictions or inconsistencies before the actual coding phase begins. Moreover, it serves as a valuable reference manual, illustrating how different modules interact with each other at a high level within the project. Overall, the HLD Document ensures a clear and well-defined structure for the coding process and enhances the project's efficiency and accuracy.

1.2. Scope

Scope of the HLD Documentation:

- Presents the system structure, including:
    - Database architecture.
    - Application architecture (layers).
    - Application flow (Navigation).
    - Technology architecture.
- Uses non-technical to mildly-technical terms for better understanding by system administrators.

2. General Description

2.1. Problem Statement

In this project, looking at the various properties of a mushroom, we will predict whether the mushroom is edible or poisonous.

2.2. Proposed Solution

- Objective: Develop a predictive model using advanced machine learning to classify mushrooms (edible vs. poisonous) based on diverse features.
- Features Considered: Cap shape, colour, gill attachment, Odor, and other relevant characteristics, including cap attributes, gill characteristics, stalk properties, and spore print colour.
- Approach: Move beyond traditional rule-based methods, ensuring the model efficiently processes diverse mushroom data for real-time accurate predictions.
- Design Focus: Create a robust and scalable solution seamlessly integrating with existing systems, prioritizing generalizability and resilience to species and environmental variations.
- Key Impact: Improve public safety by providing a reliable tool for mushroom enthusiasts, foragers, and researchers.

2.3. Data Requirements

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

The dataset includes descriptions of hypothetical samples corresponding to 23 species of gilled 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. The Guide clearly states that there is no simple rule for determining the edibility of a mushroom; no rule like "leaflets three, let it be" for Poisonous Oak and Ivy.

There are 23 variables and their unique values:

- class: edible, poisonous
- cap_shape: bell, conical, convex, sunken, flat, knobbed
- cap_surface: fibrous, grooves, scaly, smooth
- cap_color: brown, buff, cinnamon, gray, green, pink, purple, red, white, yellow
- bruises: no, yes
- odor: almond, anise, creosote, fishy, foul, musty, none, pungent, spicy
- gill_attachment: attached, free
- gill_spacing: close, crowded
- gill-size: broad, narrow
- gill_color: black, brown, buff, chocolate, gray, green, orange, pink, purple, red, white, yellow

- stalk_shape: enlarging, tapering
- stalk_root: bulbous, club, equal, rooted
- stalk-surface-above-ring: fibrous, scaly, silky, smooth
- stalk-surface-below-ring: fibrous, scaly, silky, smooth
- stalk-color-above-ring: brown, buff, cinnamon, gray, orange, pink, red, white, yellow
- stalk-color-below-ring: brown, buff, cinnamon, gray, orange, pink, red, white, yellow
- Veil-color: brown, orange, white, yellow
- ring-number: none, one, two
- ring-type: evanescent, flaring, large, none, pendant
- spore-print-color: black, brown, buff, chocolate, green, orange, purple, white, yellow
- population: abundant, clustered, numerous, scattered, several, solitary
- habitat: grasses, leaves, meadows, paths, urban, waste, woods

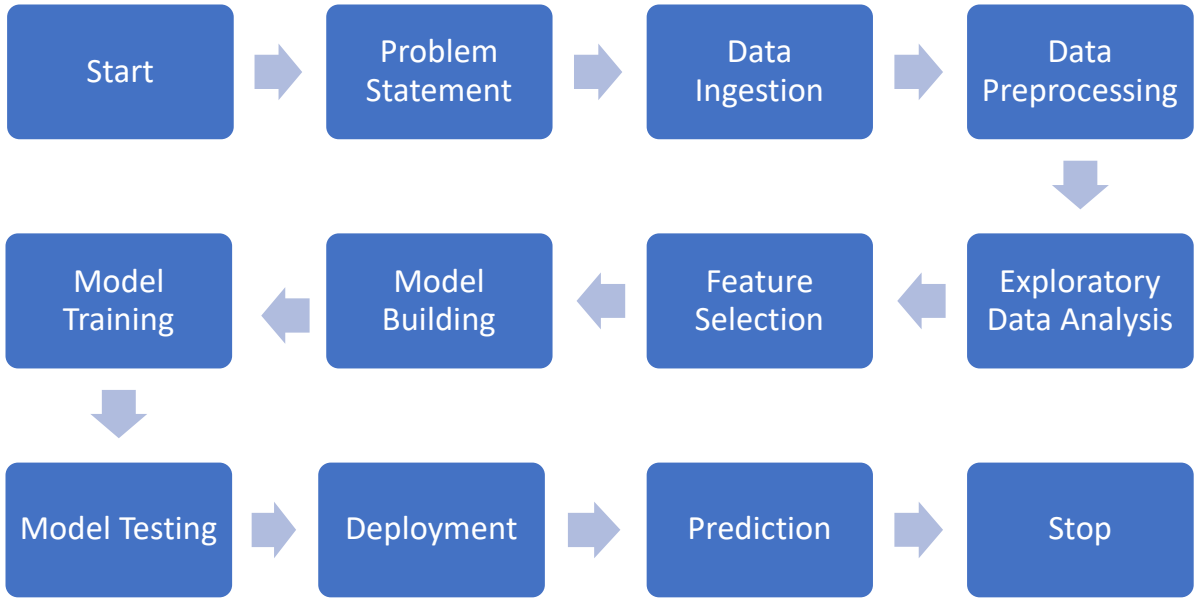
2.4. Tools Used

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

- Jupyter Notebook is used as IDE.
- For visualization of the plots, Matplotlib and Seaborn are used.
- AWS Elastic Beanstalk is used for deployment of the model.
- Front end development is done using HTML/CSS
- Python is used for backend development.
- GitHub is used as version control system.

3. Design Details

3.1. Process Flow



3.2. Application Compatibility & Resource Utilization

- **Application Compatibility**: In this project, all the various components will communicate with each other using Python as their interface. Each component has a specific role to play, and Python is responsible for ensuring seamless information exchange among them.
- **Resource Utilization:** Whenever a task is executed, it tends to utilize the available processing power to complete that task efficiently. This approach optimizes the utilization of system resources during task execution.

3.3. Data Ingestion

- Raw mushroom data is sourced from various datasets, encompassing a comprehensive array of mushroom features such as cap characteristics, gill properties, stalk attributes, spore print details, and other relevant identifiers. The data is collected from diverse locations, ensuring a representative sample of mushroom species, and is gathered in a format that allows for comprehensive analysis.

3.4. Preprocessing

- Data preprocessing is an integral step to ensure the quality and consistency of the mushroom dataset. Techniques are applied to handle missing values, address inconsistencies, and mold the data into a format suitable for subsequent analysis and modelling. Feature engineering is employed to create new features that capture meaningful information, enhancing the dataset's richness and improving the model's predictive capabilities.

3.5. Data Analysis

- Exploratory Data Analysis (EDA) is conducted to delve into the distribution of mushroom data, identify potential outliers, and unveil relationships between different features. Visualizations, statistical summaries, and data profiling techniques are utilized to gain a profound understanding of the characteristics inherent in the dataset. Insights obtained during this phase contribute to informed decisions in subsequent model development

3.6. Feature Selection

- Building on the findings from data analysis, a thoughtful feature selection process is executed. This step involves identifying and choosing the most relevant features (such as cap shape, color, gill attachment, odor, etc.) that significantly contribute to the classification of mushrooms as edible or poisonous. The goal is to enhance model efficiency and interpretability by focusing on the most informative attributes, ensuring the resulting predictive models are robust and effective.

3.7. Logging & Exception Handling

- **Logging:** Think of logging as a digital diary for our application. It diligently records significant events and actions, enabling us to monitor the system's behaviour, identify issues, and gain insights into its performance. Logging is a valuable tool for debugging, real-time monitoring, and enhancing the application's dependability.
- **Exception Handling:** Exception handling ensures that our application gracefully responds to unexpected problems or errors, preventing complete breakdowns. It offers informative error messages for users, safeguards their data, and enhances the overall user experience. This feature adds a layer of resilience to the application, making it more user-friendly and reliable.

3.8. Model Building

- Machine learning algorithms (logistic regression, decision trees, SVM) are applied to build individual base models using the selected features and the historical data.
- Ensemble techniques (bagging, boosting, stacking) are utilized to construct ensemble models, combining multiple base models to enhance predictive accuracy and robustness.

3.9. Model Training

- The individual base models and ensemble models are evaluated.
- Evaluation metrics, such as accuracy, precision, recall, & F1-score are calculated to assess the models' effectiveness in predicting credit card defaults.
- The predictions of the base models and ensemble models are combined using the specific rules of the ensemble method (e.g., averaging, voting, stacking).
- Ensemble combination creates a more accurate and robust prediction compared to individual base models.

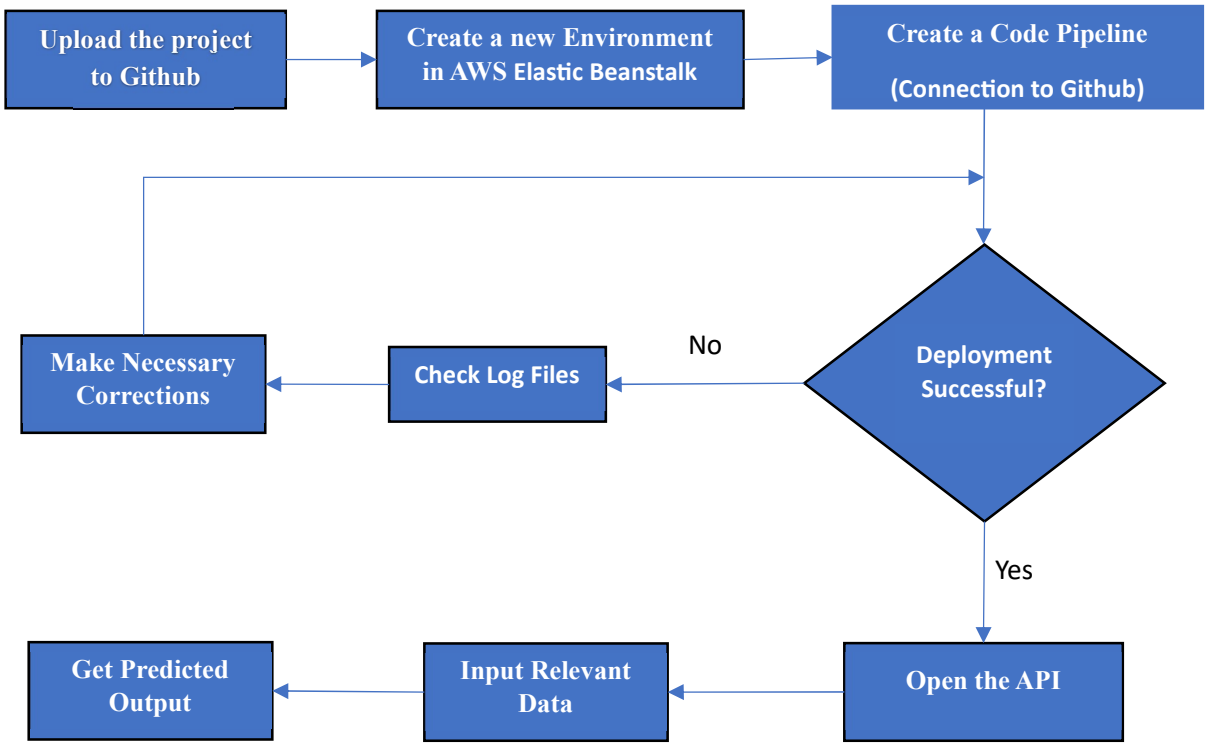
3.10. Model Testing

- The best-performing model is selected as the final predictive model for credit card default prediction.
- The Test dataset is evaluated using the best performing model.
- Evaluation metrics, such as accuracy, precision, recall, F1-score are calculated to assess the prediction credit card defaults.

3.11. Performance & Reusability

- **Performance:** The Mushroom Classification Model is developed to predict whether a given mushroom is edible or poisonous based on its features. The forecast relies on characteristics such as cap shape, color, gill attachment, odor, and other relevant factors. This tool empowers users to proactively assess the edibility of mushrooms, facilitating informed decisions for enthusiasts, foragers, and researchers.
- **Reusability:** The code and components employed in this project are designed with reusability in mind. They are structured to be easily adaptable and reusable without encountering any complications or issues.

3.12. Deployment



The project is uploaded to a GitHub repository for version control and collaboration.

- It is connected to AWS Elastic Beanstalk through a CodePipeline, streamlining the deployment process.
- The model is then deployed on AWS Elastic Beanstalk, making it accessible for use.

3.13. Prediction

- To interact with the deployed Flask API, users can access it via the generated IP address.
- After accessing the API, user needs to fill the fields to get the predicted result.

4. Conclusion

This application serves as a crucial tool for mushroom enthusiasts, foragers, and researchers to predict whether a particular mushroom is edible or poisonous. It accomplishes this by analysing a diverse set of mushroom features, including cap shape, colour, gill attachment, Odor, and other relevant characteristics. By making these predictions, the application empowers users to proactively assess the edibility of mushrooms, enabling informed decisions and enhancing safety in mushroom-related activities. This not only aids in ensuring the well-being of users but also contributes to the broader field of mycology by automating the classification process and promoting a better understanding of fungi-related risks.