Low-Level Design Document

Mushroom Classification

Document Control

Change Record:

|  |  |  |  |
| --- | --- | --- | --- |
| Version | Date | Author | Comments |
| 0.0.0 | 31-May-2023 | Anees Aro | Introduction& Architecture defined |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

Table of Contents

1. Introduction

2. Architecture

2. Component Design

2.1 User Interface

2.2 Data Pre-processing

2.3 Feature Extraction

2.4 Machine Learning Model

2.5 Classification Engine

2.6 Database

3. Data Structures and Algorithms

4. System Integration

5. Deployment and Scalability

6. Error Handling and Logging

7. Conclusion

1. Introduction

The Mushroom Classification System is a software application that aims to classify different types of mushrooms based on their attributes. This Low-Level Design document provides detailed information about the components, data structures, algorithms, system integration, deployment, security implementation, error handling, and logging aspects of the system.

2. Architecture

The Mushroom Classification Project follows a modular and layered architecture to ensure separation of concerns and maintainability. The project architecture consists of the following layers:

A picture containing text, post-it note, screenshot, font

Description automatically generated

2. Component Design

The Mushroom Classification System consists of the following components:

2.1 User Interface

The user interface component is responsible for interacting with the user and providing a means to input mushroom attributes and display classification results. The design considerations include:

- Input form design: Provides an intuitive interface for users to enter mushroom attributes.

- Result display design: Presents the classification results along with confidence levels or probabilities.

2.2 Data Pre-processing

The data pre-processing component handles the cleaning and transformation of input data to prepare it for further processing. The design considerations include:

- Handling missing values: Implements strategies such as imputation.

- Categorical data handling: Converts categorical attributes into numerical representations using ordinal encoding techniques.

2.3 Feature Extraction

The feature extraction component extracts relevant features from the pre-processed data that will be used for classification. The design considerations include:

- Feature selection: Determines the most informative features using techniques chi-square test

2.4 Machine Learning Model

The machine learning model component is responsible for training and deploying the model for mushroom classification. The design considerations include:

- Algorithm selection: Evaluates different algorithms suitable for classification tasks, such as decision trees and Logistic Regression.

- Model training: Uses labelled mushroom samples to train the model, considering techniques like cross-validation or stratified sampling.

- Best Model: Saves the best trained model to disk for future use.

2.5 Classification Engine

The classification engine component applies the trained model to new, unseen mushroom samples and generates classification results. The design considerations include:

- Input validation: Verifies the input data for completeness and correctness before classification.

- Prediction generation: Uses the trained model to predict the class labels for each mushroom sample.

- Confidence estimation: Calculates the confidence level or probability associated with each classification result.

2.6 Database

The Mushroom Classification System incorporates a database to store and retrieve mushroom data, including labelled samples and their attributes. The database component plays a vital role in data management and persistence. Here are some considerations for the database design:

- Database Management System (DBMS): Select an appropriate DBMS that suits the system's requirements, such as MySQL, PostgreSQL, MongoDB, or SQLite.

- Database Schema Design: Define the structure of the database tables to store mushroom attributes and labels. Consider the following entities:

- Data Retrieval: Implement mechanisms to retrieve data from the database for various purposes, such as training the machine learning model.

Proper design and management of the database component are essential for the Mushroom Classification System to store and retrieve mushroom data efficiently and reliably.

3. Data Structures and Algorithms

The Mushroom Classification System utilizes the following data structures and algorithms:

- Data Structures:

- Mushroom Data: A structure or class to represent a single mushroom sample, containing attributes such as cap shape, cap color, gill color, etc.

- Dataset: A collection of Mushroom Data instances representing the labelled mushroom samples used for training the machine learning model.

- Feature Set: A collection of features extracted from the pre-processed data for training and classification.

- Model: A structure or class representing the trained machine learning model, including its parameters and weights.

- Algorithms:

- Data Pre-processing Algorithms: These algorithms handle missing value imputation, categorical data transformation.

- Feature Extraction Algorithms: Algorithms like correlation analysis, feature importance, or dimensionality reduction techniques such as PCA or LDA.

- Machine Learning Algorithms: Various classification algorithms such as decision trees, Logistic Regression can be used for training and classification tasks.

- Evaluation Metrics: Algorithms to calculate evaluation metrics such as accuracy, precision, recall, or F1-score to assess the performance of the classification model.

4. System Integration

The Mushroom Classification System may integrate with external components or services. Some integration points to consider are:

- Database Integration: Integrating with a database management system (DBMS) allows storing and retrieving mushroom data, including labelled samples and their attributes.

- Machine Learning Framework Integration: Integration with machine learning frameworks or libraries, such as scikit-learn may be required to train and deploy the machine learning model.

Proper integration with these components should be established using well-defined interfaces, protocols, or APIs to ensure seamless communication and interoperability.

5. Deployment and Scalability

The Mushroom Classification System can be deployed using a scalable infrastructure to handle increased user demands and future growth. Considerations for deployment and scalability include:

- Cloud Deployment: Deploying the system on a cloud platform provides scalability, flexibility, and easy resource allocation.

7. Error Handling and Logging

Proper error handling and logging mechanisms are crucial for the Mushroom Classification System. Here are some considerations:

Error Handling:

- Identify potential error scenarios, such as invalid input, data pre-processing failures, model training errors, or classification engine failures.

- Implement robust error handling routines to gracefully handle exceptions and provide meaningful error messages to the user.

- Utilize try-catch blocks or exception handling techniques to catch and handle errors at appropriate levels of the system.

- Implement error codes or error objects to classify and categorize different types of errors for easier troubleshooting and debugging.

Logging:

- Implement logging mechanisms to record important system events, errors, warnings, and informational messages.

- Use logging frameworks or libraries to facilitate structured and configurable logging.

- Log critical information such as user actions, system responses, input data, pre-processing steps, model training progress, classification results, and errors encountered.

- Include timestamps, severity levels, and relevant contextual information in log entries for easier troubleshooting and debugging.

- Log messages should follow a consistent format and adhere to the logging best practices.

Logging helps in diagnosing system issues, identifying patterns, monitoring system performance, and auditing user actions.

8. Conclusion

The Mushroom Classification System is designed to classify mushrooms based on their attributes using machine learning techniques. This Low-Level Design document provides insights into the component design, data structures, algorithms, system integration, deployment, security implementation, error handling, and logging aspects of the system.

By following the design guidelines outlined in this document, the system can efficiently pre-process data, extract relevant features, train and deploy machine learning models, and generate accurate classification results. Proper system integration, scalability, security measures, error handling, and logging mechanisms ensure the system's reliability, security, and ease of maintenance.

The Mushroom Classification System has the potential to assist users in accurately identifying different types of mushrooms, promoting safety and knowledge in the domain of mushroom classification and identification.