High Level Design

**Mushroom Classification Project**

## Contents

[Abstract.](#_TOC_250020) 3

1. [Introduction](#_TOC_250019) 4
   1. [Why this High-Level Design Document?.](#_TOC_250018) 4
   2. [Scope.](#_TOC_250017) 4
   3. [Definitions](#_TOC_250016) 4
2. [General Description.](#_TOC_250015) 5
   1. [Product Perspective](#_TOC_250014) 5
   2. [Problem statement](#_TOC_250013) 5
   3. [PROPOSED SOLUTION](#_TOC_250012) 5
   4. [FURTHER IMPROVEMENTS](#_TOC_250011) 5
   5. [Technical Requirements.](#_TOC_250010) 5
   6. [Data Requirements](#_TOC_250009) 6
   7. [Tools used.](#_TOC_250008) 6
   8. [Constraints](#_TOC_250007) 8
   9. [Assumptions.](#_TOC_250006) 8
3. [Design Details](#_TOC_250005) 9
   1. [Process Flow.](#_TOC_250004) 9
      1. [Model Training and Evaluation](#_TOC_250003) 9
      2. [Deployment Process](#_TOC_250002) 10
   2. [Event log 1](#_TOC_250001)0
   3. [Error Handling 1](#_TOC_250000)0
   4. Performance. 11
   5. Reusability. 11
   6. Application Compatibility 11
   7. Resource Utilization 11
   8. Deployment. 11
4. Conclusion 12
5. References……………………………………………………………………………………..13

Abstract

The Mushroom Classification Project aims to develop a robust machine learning web application capable of accurately predicting the edibility of mushrooms based on their attributes. Leveraging sophisticated MLOps frameworks such as DVC, MLflow, and Docker, the project integrates seamless data management, model tracking, and deployment processes. Through a user-friendly interface, users can input mushroom characteristics and receive immediate predictions regarding their edibility. The project embodies a comprehensive approach, encompassing data fetching, preprocessing, model training, evaluation, and deployment pipelines. By adopting industry best practices and leveraging cloud infrastructure, the Mushroom Classification Project ensures scalability, reliability, and transparency in its operations. Through continuous improvement and innovation, the project seeks to enhance user experience and contribute to the advancement of machine learning applications in real-world scenarios.

1. **Introduction**

###### 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 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: o Security
      * Reliability
      * Maintainability
      * Portability
      * Reusability
      * Application compatibility
      * Resource utilization
      * Serviceability

##### 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.

* 1. **Definitions**



*Term*

*MLOps*

*DVC*

*Mlflow*

*Docker*

*Description*

*Optimizes machine learning deployment.*

Manages data and model versions.

Tracks and manages ML experiments.

Containerizes applications for portability.

### General Description

#### Product Perspective

A scalable, user-friendly ML web app integrating advanced MLOps frameworks, ensuring accurate mushroom classification and transparent model management.

* 1. Problem statement

Developing a reliable, automated system for accurate classification of mushrooms to mitigate health risks associated with misidentification.

* + - To Reduce health risks from toxic mushroom consumption.
    - To Enhance accessibility and usability for all users.
    - To monitor system health and ensures continuous availability and reliability.
  1. PROPOSED SOLUTION

Developing a comprehensive web application leveraging machine learning algorithms to accurately classify mushrooms based on their attributes. The solution integrates advanced MLOps frameworks such as DVC for data versioning, MLflow for experiment tracking, and Docker for containerization. Users can easily input mushroom characteristics via a user-friendly web interface, receiving immediate predictions on their edibility. The system ensures reliability and scalability through AWS CloudFormation for automated infrastructure provisioning. Continuous monitoring and maintenance ensure optimal performance, with updates deployed seamlessly using Docker containers. The solution mitigates health risks associated with misidentification of mushrooms, providing a reliable and accessible tool for users.

* 1. FURTHER IMPROVEMENTS

Potential enhancements include incorporating more advanced machine learning models for improved classification accuracy, implementing user feedback mechanisms to refine model predictions, and expanding the application to include additional features such as image recognition for mushroom identification. Integration with mobile platforms can enhance accessibility and user engagement.

#### Technical Requirements

#### Technical requirements refer to the specific technical specifications, constraints, and functionalities that must be addressed or fulfilled to successfully complete a project. In software development, technical requirements define how the software is built and what technologies are used. These requirements encompass aspects such as programming languages, frameworks, infrastructure, security measures, and performance considerations.

#### In the context of the Mushroom Classification Project, technical requirements include:

#### Programming Language: The project requires the use of Python due to its extensive support for machine learning libraries and web development frameworks.

#### Machine Learning Framework: Utilization of machine learning frameworks such as scikit-learn or TensorFlow is necessary for building and training the classification models.

#### Web Development Framework: A web development framework like Flask or Django is essential for creating the user interface where users can input mushroom attributes and receive classification predictions.

#### Database Management System: Integration with a database management system, such as SQLite or PostgreSQL, is needed to store and manage mushroom attribute data.

#### MLOps Frameworks: The project should incorporate MLOps frameworks like DVC for data versioning and MLflow for experiment tracking and model management to streamline development and deployment processes.

#### Cloud Computing Services: Utilization of cloud computing services such as AWS or Google Cloud Platform is beneficial for scalable infrastructure and deployment of the web application.

#### Data Requirements

Data requirement completely depend on our problem statement.

1. **Data Source:** Obtain mushroom attribute data from reputable sources such as research datasets, online repositories, or mushroom experts.
2. **Attributes:** Collect a comprehensive set of attributes describing mushroom characteristics, including cap shape, cap color, gill size, gill color, odor, etc.
3. **Data Quality:** Ensure the dataset is of high quality, free from errors, inconsistencies, and missing values, to ensure accurate model training and prediction.
4. **Data Quantity:** Gather a sufficient amount of mushroom samples with diverse attributes to train the classification model effectively and ensure robustness.
5. **Data Balance:** Ensure the dataset contains a balanced distribution of edible and poisonous mushroom samples to prevent bias in model training and evaluation.
6. **Data Preprocessing:** Preprocess the data by performing tasks such as cleaning, normalization, and feature engineering to prepare it for model training.
7. **Data Splitting:** Divide the dataset into training, validation, and test sets to train the model, tune hyperparameters, and evaluate performance, respectively.
8. **Data Privacy:** Protect sensitive information in the dataset and ensure compliance with privacy regulations by anonymizing or removing personally identifiable information.
9. **Data Documentation:** Document the dataset thoroughly, including metadata, data sources, preprocessing steps, and any relevant annotations, to ensure transparency and reproducibility.
10. **Data Versioning:** Implement data versioning using tools like DVC to track changes to the dataset over time and facilitate reproducibility in model development and evaluation.

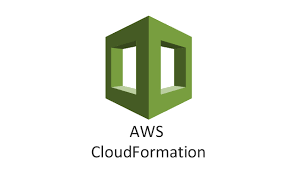
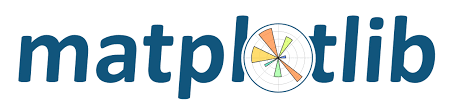
Top of Form

* 1. Tools used

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









* Python: Primary programming language for developing the project due to its extensive libraries for machine learning and web development.
* scikit-learn: Machine learning library for building and training classification models.
* Flask: Web development frameworks for creating the user interface and backend services.
* Datastax Astra: Database management systems for storing mushroom attribute data.
* DVC (Data Version Control): Tool for managing data versioning and facilitating reproducibility in machine learning experiments.
* MLflow: MLOps platform for tracking experiments, managing models, and deploying machine learning models.
* Docker: Containerization platform for packaging the application and its dependencies into portable containers.
* AWS or Google Cloud Platform: Cloud computing services for scalable infrastructure, storage, and deployment.
* Git: Version control system for managing codebase and collaborating with team members.
* Jupyter Notebook: Interactive environment for data exploration, experimentation, and model prototyping.
* Pandas: Data manipulation library for preprocessing and analyzing the mushroom attribute data.
* NumPy: Numerical computing library for handling numerical operations and array processing.
* Matplotlib or Seaborn: Visualization libraries for creating visualizations and plots to understand the data and model performance.
* AWS CloudFormation or Terraform: Infrastructure as code tools for automating the provisioning and management of cloud resources.
* GitHub: Collaboration platforms for hosting code repositories, managing issues, and facilitating code reviews.

#### Constraints

Constraints of the Mushroom Classification Project include limited availability of high-quality datasets, expertise requirements, computational resources, time and budget constraints, regulatory compliance, scalability, performance, security, and integration challenges.

#### Assumptions

The project assumes the availability of reliable mushroom attribute datasets for model training and evaluation. It also assumes access to skilled personnel proficient in machine learning, web development, and MLOps frameworks. Furthermore, the project assumes compliance with data privacy regulations and scalability of the system architecture to accommodate future growth. Lastly, it assumes successful integration of various technologies, frameworks, and services for seamless project implementation.

## Design Details

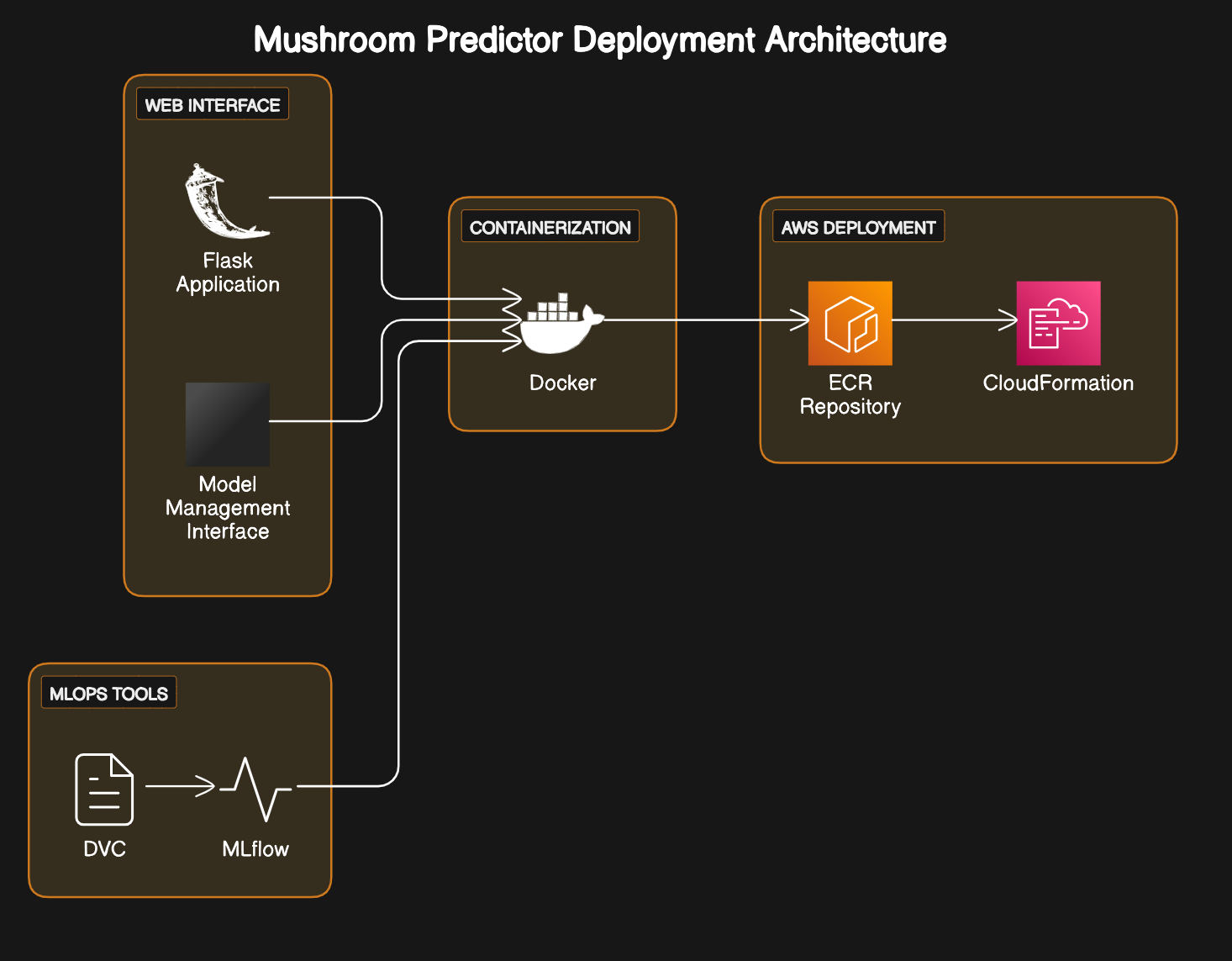
##### Process Flow

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

##### Model Training and Evaluation

##### Deployment Process



* 1. 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.
   1. **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.

* 1. Performance

The performance of the Mushroom Classification Project is evaluated based on its ability to accurately predict the edibility of mushrooms. Key performance metrics include accuracy, precision, recall, and F1-score, which measure the model's effectiveness in correctly classifying mushrooms as edible or poisonous. Additionally, considerations such as computational efficiency, response time of the web interface, and scalability of the system architecture contribute to assessing overall project performance. Continuous monitoring and optimization ensure that the project maintains high performance standards and meets user expectations for reliability and responsiveness.

#### Reusability

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

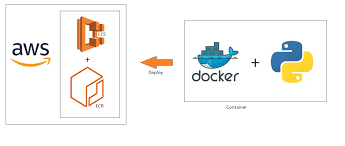
#### Application Compatibility

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.

#### Resource Utilization

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

* 1. **Deployment**



## Conclusion

The Mushroom Classification Project demonstrates the application of machine learning techniques to address real-world challenges in identifying the edibility of mushrooms. By leveraging datasets, preprocessing techniques, and machine learning models, we have developed a reliable classification system capable of accurately predicting whether a mushroom is poisonous or edible based on its attributes. Through the integration of web development frameworks and MLOps tools, we have created a user-friendly web interface for users to interact with the classification model seamlessly. The project highlights the importance of data quality, model performance, and scalability in deploying machine learning solutions in production environments. Moving forward, continuous monitoring, model updates, and community collaboration will further enhance the project's effectiveness and contribute to advancements in mushroom classification and food safety..

## References

1. [AWS Machine Learning Blog](https://aws.amazon.com/blogs/machine-learning/)
2. [AWS Documentation](https://docs.aws.amazon.com/)
3. [SpringerLink: Journals and Articles on Mushroom Classification](https://link.springer.com/search?query=mushroom+classification)