# 

**Mushroom Classification**

**High Level Design**

***Domain : Machine Learning***

***Created By : Biswajit Swain***

***Date : 01.03.2023***

**Document Version Control**

|  |  |  |  |
| --- | --- | --- | --- |
| **Date** | **Version** | **Description** | **Author** |
| **March 01 , 2023** | **1.1** | **First draft** | **Biswajit Swain** |
| **March 05 , 2023** | **1.2** | **Added unit test cases** | **Biswajit Swain** |

**ABSTRACT**

Mushroom is one of the important ingredients in our food that has good nutrients. Most type of mushroom are poisonous (inedible), and because of its importance, we need to identify poisonous from eatable mushrooms.

The goal of this project is to build a prediction model using multiple machine learning techniques, which will classify the mushroom into edible or inedible. We're trying to forecast the category of mushroom with Mushroom Classification dataset downloaded from UCI repository.

**CONTENTS**

**[Introduction..................................................................................5](file:///C:\\Users\\user\\Downloads\\High%20Level%20Document.docx" \l "_Toc110595994)**

**[What is High-Level Design Document? 4](file:///C:\\Users\\user\\Downloads\\High%20Level%20Document.docx" \l "_Toc110595995)**

**[Scope 5](file:///C:\\Users\\user\\Downloads\\High%20Level%20Document.docx" \l "_Toc110595996)**

**[General Description 6](file:///C:\\Users\\user\\Downloads\\High%20Level%20Document.docx" \l "_Toc110595997)**

**[Definitions 6](file:///C:\\Users\\user\\Downloads\\High%20Level%20Document.docx" \l "_Toc110595998)**

**[Product Description 6](file:///C:\\Users\\user\\Downloads\\High%20Level%20Document.docx" \l "_Toc110595999)**

**[Problem Statement 6](file:///C:\\Users\\user\\Downloads\\High%20Level%20Document.docx" \l "_Toc110596000)**

**[Proposed solution 6](file:///C:\\Users\\user\\Downloads\\High%20Level%20Document.docx" \l "_Toc110596001)**

**[Further improvements 7](file:///C:\\Users\\user\\Downloads\\High%20Level%20Document.docx" \l "_Toc110596002)**

**[Data requirements 7](file:///C:\\Users\\user\\Downloads\\High%20Level%20Document.docx" \l "_Toc110596003)**

**[Tools used 8](file:///C:\\Users\\user\\Downloads\\High%20Level%20Document.docx" \l "_Toc110596004)**

**[Hardware Requirements 9](file:///C:\\Users\\user\\Downloads\\High%20Level%20Document.docx" \l "_Toc110596005)**

**[Constraints 9](file:///C:\\Users\\user\\Downloads\\High%20Level%20Document.docx" \l "_Toc110596006)**

**[Assumptions 9](file:///C:\\Users\\user\\Downloads\\High%20Level%20Document.docx" \l "_Toc110596007)**

**[Design Details 8](file:///C:\\Users\\user\\Downloads\\High%20Level%20Document.docx" \l "_Toc110596008)**

**[Process Flow 8](file:///C:\\Users\\user\\Downloads\\High%20Level%20Document.docx" \l "_Toc110596009)**

**[Event Log 9](file:///C:\\Users\\user\\Downloads\\High%20Level%20Document.docx" \l "_Toc110596010)**

**[Error Handling 10](file:///C:\\Users\\user\\Downloads\\High%20Level%20Document.docx" \l "_Toc110596011)**

**[Performance 1](file:///C:\\Users\\user\\Downloads\\High%20Level%20Document.docx" \l "_Toc110596012)1**

**[Reusability 1](file:///C:\\Users\\user\\Downloads\\High%20Level%20Document.docx" \l "_Toc110596013)1**

**[Application Compatibility 1](file:///C:\\Users\\user\\Downloads\\High%20Level%20Document.docx" \l "_Toc110596014)1**

**[Resource Utilization 1](file:///C:\\Users\\user\\Downloads\\High%20Level%20Document.docx" \l "_Toc110596015)1**

**[Dashboards.................................................................................1](file:///C:\\Users\\user\\Downloads\\High%20Level%20Document.docx" \l "_Toc110596016)2**

**[KPIs (Key Performance Indicators) 1](file:///C:\\Users\\user\\Downloads\\High%20Level%20Document.docx" \l "_Toc110596017)2**

**[Conclusion...................................................................................1](file:///C:\\Users\\user\\Downloads\\High%20Level%20Document.docx" \l "_Toc110596018)3**

INTRODUCTION

# What is High Level Design document?

The goal of this HLD or a high-level design 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 design aspects and define them in detail
* Describe all user interfaces being implemented
* Describe the hardware and software interfaces
* Describe the performance requirements
* Include design features and architecture of the project
* List and describe the non-functional attributes such as security, reliability, maintainability, portability, re-usability, application compatibility. resource utilization, serviceability .

Scope

The HLD documentation presents the structure of the system, such as 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.

# *General Description*

# *Definition*

|  |  |
| --- | --- |
| Term | Description |
| Database | Collection of the information |
| Cloud | A data center full of services connected to the internet performing service |
| IDE | Integreted Development Environment |
| UI | User Interface |
| AWS | A cloud service |

# *Product Description*

Mushroom Classification is a Machine Learning based classification model which helps us to do predictive analysis of the mushroom type using certain features and parameters.

# *Problem Statement*

The aim is to create a ML based model which will predict the type of mushroom.

# *Proposed Solution*

Using all the standard techniques used in the life cycle of a Data Science project starting from Data Exploration, Data Cleaning, Feature Engineering, Model Selection, Model Building and Model Testing and also building a frontend where a user can fill their information in the form input and get the output instantly.

# *Further Improvements*

This Mushroom Classification can be easily embedded inside any website or an application and everybody can get quick answer by inputting required data on friendly user interface.

This can be further improved by training more data in the model. Data can be acquired from Audubon Society Field.

# *Data Requirements*

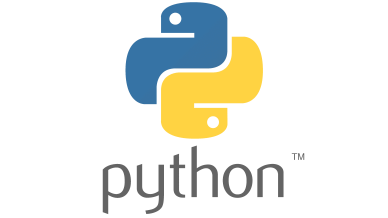
Data requirement completely depend on our problem statement. Required dataset should contain the following features:

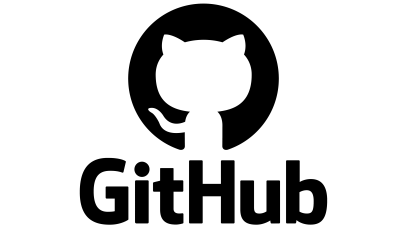
* Color and shape related informations about the mushroom.
* We should know the habitat of the mushroom.
* The veil type and veil color.
* Information related stalk color , stalk surface, stalk root.
* All information related to cap and gill. i.e. color ,size ,surface , shape , spacing ,etc.

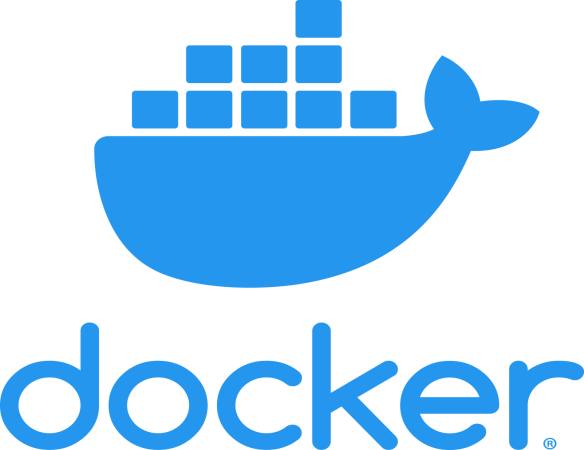
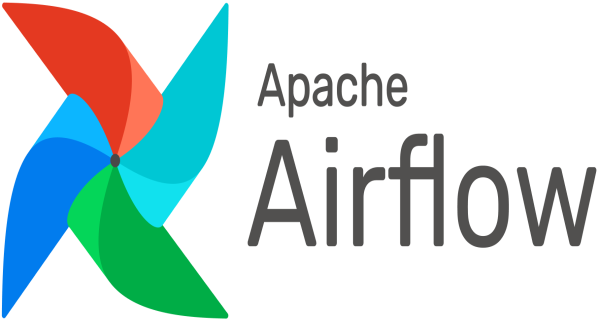
These are the required parameters to feed into model.

*Tools Used*

Python programming language and frameworks such as NumPy, Pandas, Scikit-learn, Air Flow, Docker and a few other libraries were used to build the whole model.





* For visualization tasks, matplotlib, seaborn and plotly were used.
* AWS is used for building the web application and server to run the code.
* Mongodb was used to storage and retrieval of data.
* GitHub is used as version control system.
* NumPy and Pandas were used to clean and interpret data.
* Scikit-learn was used to cross validate and compare different models.

# *Hardware Requirements*

* Minimum 1.10 GHz processor or equivalent.
* Between 1-2 GB of free storage
* Minimum 512 MB of RAM
* 3 GB of hard-disk space

# *Constraints*

The front-end must be user friendly and should not need any one to have any prior knowledge in order to use it.

# *Assumptions*

It is assumed that all aspects of this project have the ability to work together as the designer is expecting and also the data on which our model is trained is as correct as possible.

# *Design Details*

# *Process Flow*

For accomplishment of the task, we will use a trained Machine Learning model. The process flow diagram is shown below:

# *Project Architecture*

# *Deployment Architecture*

# *Event Logs*

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 level logging is required.
2. The system should be able to log each and every system flow.
3. Developer can choose logging method.
4. You can choose database logging/file logging as well.
5. System should not hand even after so much logging.

# *Error Handling*

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

# *Performance*

The Mushroom Classification tool is used to predict whether the mushroom is edible or poisonous. It can be used by various governmental/ non-governmental/ private agencies then it is supposed to be as accurate as possible. So that it doesn’t mislead authorities. Also, model retraining is very important to further enhance its performance.

# *Reusebility*

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 Python to ensure proper transfer of information.

# *Resource Utilization*

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

# *Dashboards*

As and when, the system starts to capture the historic/ periodic data for a user, the dashboards will be included display charts over time with progress on various indicators or factors.

# *KPIs (Key Performance Indicators)*

* Key Performance Indicators of Mushroom Classification.
* Latency or the amount of time the application takes to display results for some specific input.
* The processing power our application takes to run.
* The memory and RAM our application takes to run on a web server.

***Conclusion***

All in all, overall project architecture, design details, used technologies and performance were explained in detail. The Mushroom Classification will give the mushroom type predictions instantly and has the potential to help various government organizations, agencies and etc.