LLD Document

Campus Placement

Name: Abhishek Rajesh Saste

**Contents**

[**List of Figures** 2](#_Toc120613610)

[**1.** **Introduction** 3](#_Toc120613611)

[**1.1** **Why low-level design document?** 3](#_Toc120613612)

[**2.** **Architecture** 3](#_Toc120613613)

[**3.** **Architecture Description** 4](#_Toc120613614)

[**4** **Unit test case** 6](#_Toc120613615)

# **List of Figures**

[**Figure 1: System Architecture** 3](#_Toc120613538)

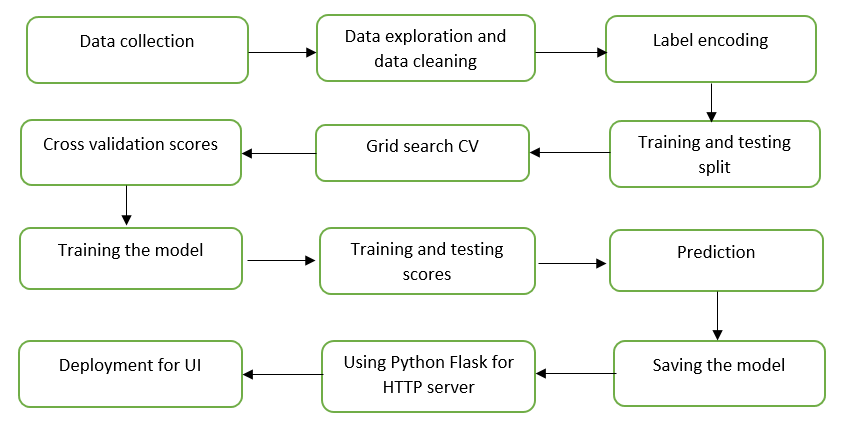
[**Figure 2: confusion matrix for the observation of the predicted values** 6](#_Toc120613539)

# **Introduction**

## **Why low-level design document?**

The main aim of creating a low-level document is to provide internal logic and construction of the process of building and using the machine learning model for the deployment. This document will be beneficial for any programmer to use and rewrite or update the code from scratch. It contains the system architecture and its explanation to understand the internal working of the project.

# **Architecture**



**Figure 1: System Architecture**

# **Architecture Description**

1. Data collection

The dataset on which the entire project is based has been collected from the provided dataset from Kaggle. It contains various features from which the prediction factor is dependent on the required solution to the given problem statement. Using these features from the dataset a machine-learning model is made for the prediction.

1. Data Exploration and Data cleaning

* The dataset on which the model is to be built has some faults or false values which are needed to be cleaned. These junks are cleaned by first exploring the dataset it includes, and learning about various features present in the dataset.
* After discovering the flaws and learning the data's features, it is time to clean the data and make it available for further processing. Cleaning the data includes finding out unconditional values that make the model unstable during the prediction which directly affects the accuracy of the machine learning model.

1. Label Encoding

Label encoding is an important pre-processing part for structured datasets in supervised learning while building any machine learning model. It converts all the data or string values into the numeric form so that the machine can read the information.

1. Training and testing split

The validation process which helps to reveal the working performance of the model is said to be a training and testing split. The model gets trained on a certain amount of data and that trained model is then tested by providing a part of the data which is never seen by the model to check its working accuracy.

1. Grid Search Cross-validation

Many machine learning models with different parameters can be used for a similar problem. Grid search CV helps determine the best of the many algorithms users can feed. It helps to check and compare every model’s accuracy on the same data at the same time. Also, the accuracies can be tried to increase by tuning their hyper-parameters.

1. Cross-Validation scores

The main objective of cross-validation is checking and testing the ability of the machine learning model for predicting new data. This also shows insights into how the model will react to an independent dataset, identifying the problems like selection bias or overfitting.

1. Training the model

The model with the highest accuracy obtained from the grid search CV is then can be used for the training of the machine learning model.

1. Training and testing scores

Using these scores, the performance of the model can be observed knowing how the model performs on seen and unseen data.

1. Prediction

Using the features on which the model has been trained, the required output can be checked by writing code or visualizing the predictions with the help of a confusion matrix.

1. Saving the model

This machine-learning model can be saved in the pickle file to let the user use the trained model or share it for the deployment and it can be easily reloaded without training the dataset over and over.

1. Python FLASK for HTTP server

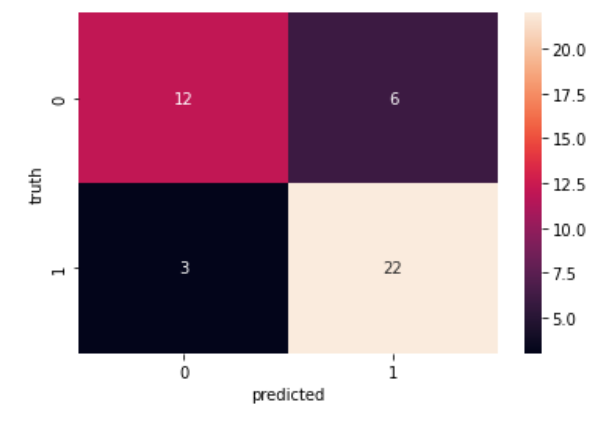
Python FLASK is a web framework that helps to connect the trained machine learning model with the front-end application. A user interfaces wherein this ML model can be used for the prediction with the help of scripting languages can be developed and made available for the user to use its prediction power.

1. Deployment

Finally, this webpage with the trained ML model and a healthy user interface can be made available to the public by deploying it on cloud platforms. This makes it easy for the users to use and access this webpage and make its use to the fullest. There are several cloud platforms where these webpages can be deployed and made available for the user e.g. Azure, Google Cloud, AWS, etc.

# **Unit test case**

The performance of the model was evaluated using the confusion matrix which shows the right predicted count to be 34 out of 43 which is almost 79% of the whole test data and it can be seen in



**Figure 2: confusion matrix for the observation of the predicted values**