

**Air Quality Index Prediction**

Low Level Design

Domain: Machine Learning

Creator: Shridatta Patil

Date: 25.10.2023

|  |  |  |  |
| --- | --- | --- | --- |
| Date issued | Version | Description | Author |
| Oct 25, 2023 | 1.1 | First Draft | Shridatta Patil |

**Document Version Control**

Contents

[Introduction 3](#_Toc110433758)

[What is Low-Level Design Document? 3](#_Toc110433759)

[Scope 3](#_Toc110433760)

[Architecture 4](#_Toc110433761)

[Architecture Description 5](#_Toc110433762)

[Data Preparation 5](#_Toc110433763)

[Data Description 5](#_Toc110433764)

[Data Preprocessing 5](#_Toc110433765)

[Exploratory Data Analysis 5](#_Toc110433766)

[Feature Engineering 5](#_Toc110433767)

[Model Development 6](#_Toc110433768)

[Model implementation 6](#_Toc110433769)

[Hyper-parameter Tuning 6](#_Toc110433770)

[Model Evaluation 6](#_Toc110433771)

[Deployment 6](#_Toc110433772)

[Designing UI 6](#_Toc110433773)

[Designing a server 6](#_Toc110433774)

[Code deployment on cloud 6](#_Toc110433775)

[Deployment Process 6](#_Toc110433776)

# Introduction

## What is Low-Level Design Document?

The goal of LLD or a low-level design document is to give the internal logical of the actual program code for Air Quality Index Prediction. It will explain the purpose and features of the system, the interfaces of the system, what the system will do, the constraints under which it must operate and how the system will react to given data. The main objective of the project is to predict the air quality index on particular date.

## Scope

Low-level design (LLD) is a component-level design process that follows a step-by-step refinement process. This process can be used for designing data structures, required software architecture, source code and ultimately, performance algorithms. Overall, the data organization may be defined during requirement analysis and then refined during data design work.

# Architecture

**Data Preparation**

**Model**

**Development**

**Deployment**

**Deployment**

# Architecture Description

## Data Preparation

### Data Description

Data will contain the city names and other 10 columns of different parameters which includes PM2.5, PM10, NO etc. The last column will have the value of air quality index. The goal of this project is to build a prediction model using multiple machine learning techniques and to use a template to document the end-to-end stages. We're trying to forecast the value of a continuous variable with the Air Quality dataset, which is a regression issue.

### Data Preprocessing

In data preprocessing step, we check if there missing data, duplicate values, and datatypes of each feature. In this dataset missing values are present.

### Exploratory Data Analysis

This step includes analysis of features. Pair plot is used to check the relationships between variables within a dataset. Distribution of numerical values is plotted to see to what extent our data is skewed.

### Feature Engineering

In this part, the missing values in the dataset are handled. The columns ‘City’, ‘Date’ and ‘AQI Bucket’ are dropped from the data.

## Model Development

### Model implementation

After train and test splitting, standardization is done using Standard Scaler and fitted to several models such as Linear Regressor, Random Forest Regressor. Their R2 score were obtained. It was determined that Random Forest Regressor performs better than other models.

### Hyper-parameter Tuning

The best model is chosen, and Grid Search with Cross Validation is applied on that model to get the best parameters. Those parameters are then used on the model to get better result.

### Model Evaluation

Test dataset is used to evaluate the model. 20% of dataset was separated for testing. Predicted results of the model are compared with the actual data to check the amount of error. As there was no considerable change after hyperparameter tuning, it helped us to overcome overfitting and perform better on new data.

## Deployment

### Designing UI

For this project, a user interface is built using simple HTML and CSS code.

### Designing a server

A server should be created to run the UI application continuously. Flask server is built.

### Code deployment on cloud

The codes for this machine learning model should be deployed to the cloud, so that when data is entered into the application, our code runs, and a user gets the result online.

## Deployment Process

In this stage, we establish a server using Flask to run our code, where our model is executed. We will post the hole after execution or asynchronous execution. Git and GitHub are used to code in the Heroku cloud.