LOW LEVEL DESIGN (LLD)

Prediction of LC50 value using Quantitative Structure Activity Relationship models

Written By	Abhilash S Bharadwaj
Document Version	2
Last Revised Date	09-Feb-2024

Document Control Version

Date Issued	Version	Description	Author
5/02/2024	1	Introduction & Architecture defined	Abhilash S Bharadwaj
9/02/2024	2	Unit Test Cases defined and appended	Abhilash S Bharadwaj

Approval Status:

Version	Review Date	Reviewed by	Approved by	Comments

Contents

Introduction

What is a Low-Level Design Document?

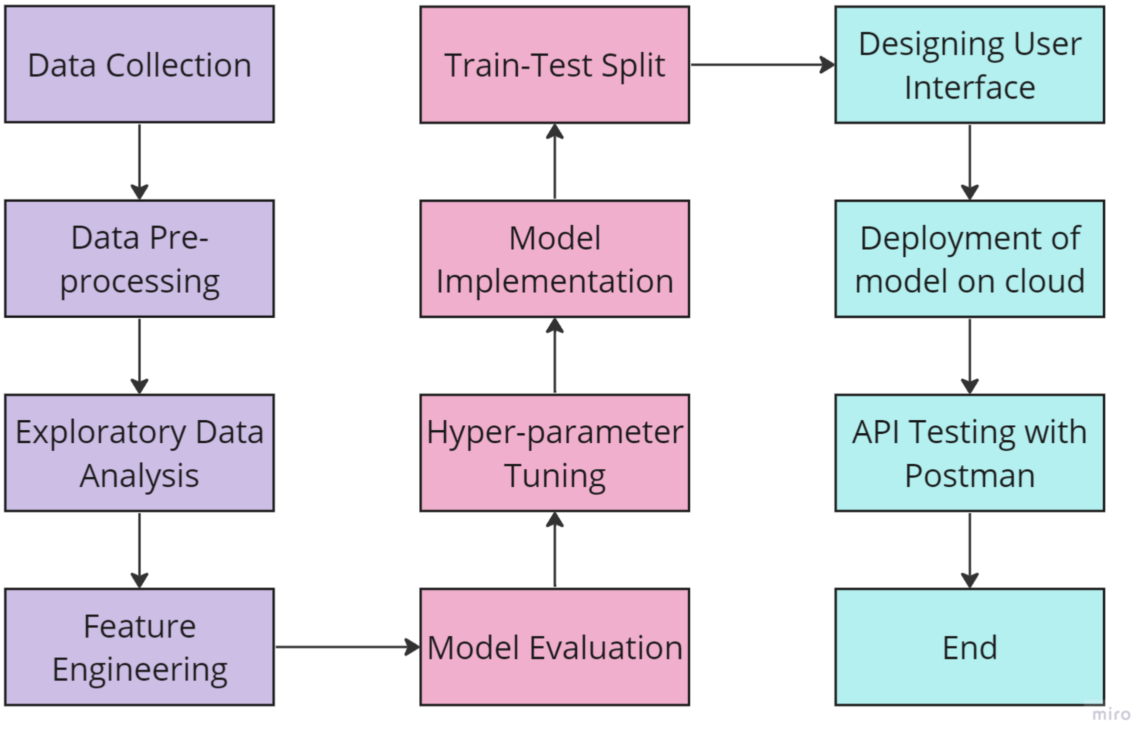
The goal of LLD or a low-level design document is to give the internal logic of the actual program code for Metro Interstate Traffic Volume Prediction. It will explain the purpose and features of the system, the interfaces of the system, what the system will do, the constraints under which it must operate and how the system will react to external stimuli.

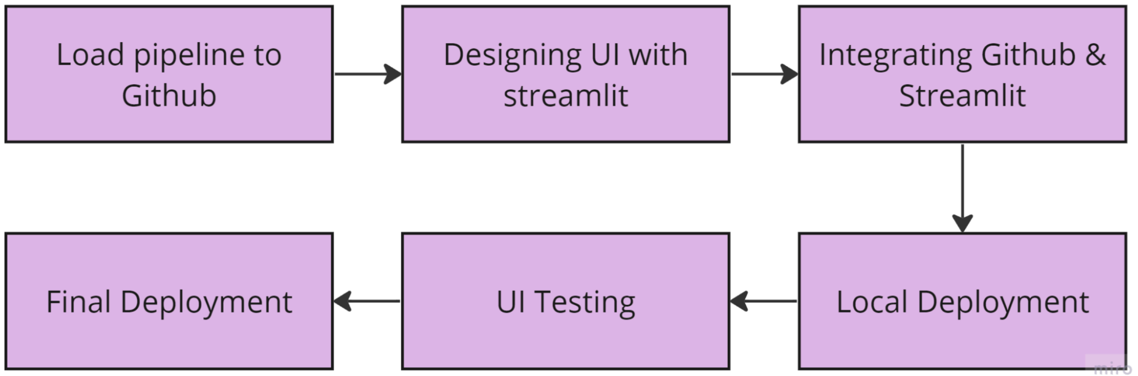
The main objective of the project is to predict if traffic volume is high or low on a particular date. Weather circumstances, special days like holidays, daytime (morning, afternoon, night and etc.), a temperature, a weekday, a numeric percentage of cloud cover are vital attributes for predicting traffic volume.

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





Architecture Description

Data Description

A comprehensive dataset comprising 908 chemicals was meticulously analyzed and modeled. The focus of the investigation was the LC50 data, a pivotal metric representing the concentration of which 50% of test fish experience mortality over a 96 hour period. The development of quantitative regression models hinged on 6 molecular descriptors, each chosen for its unique contribution for unraveling the chemical complexities. We are trying to predict acute toxicity towards the fathead minnow (Pimephales promelas'). This comes under a regression specific problem.

Data pre-processing

In this step, majorly we analyzed if there is any missing data or instances of data in the dataset. Also checked the occurrence of any duplicate values and analyzed the data types of each and every feature.
We came to conclusion that there wasn't any missing value nor any duplicates and all the features were either float or int datatype

Exploratory Data Analysis

In this step, it included the correlation between the features. Basically, correlation refers to a statistical measure that quantifies the degree to which 2 variables change together. That is between features (independent variables) and the target variable (dependent variable). Later we also analyzed the bivariate and univariate analysis of the features. We also checked where any outliers present in the dataset using boxplot etc. At last, the distribution of numerical values were conducted to analyze the plot to understand the degree of skewness in our data.

Feature Engineering

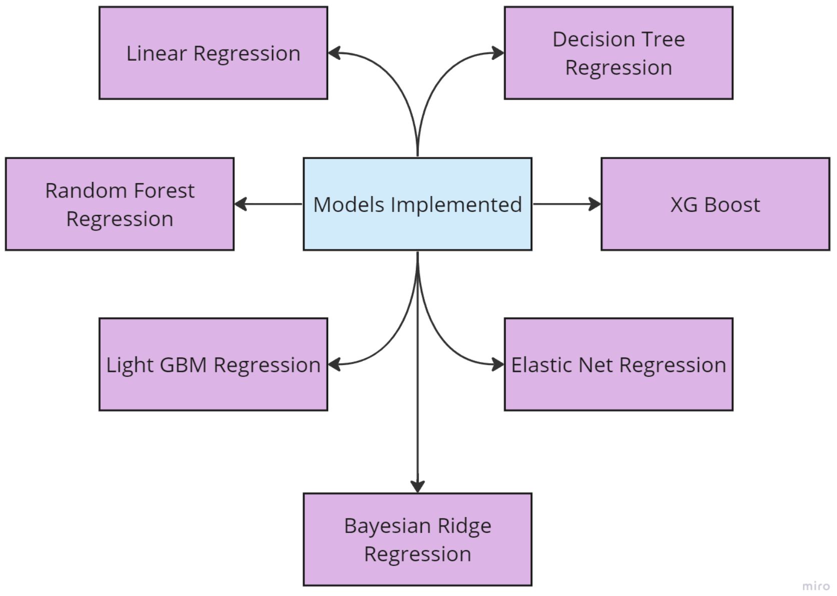
In this step the datatypes of two features such as 'NdsCH' & 'NdssC' were changed. Because, at first, the datatypes of these two features were 'int' but as our domain or the dataset end goal is based on regression type, we converted these int datatype to 'float' datatype for the betterment of the models.

Train Test split

It is essential to effectively partition the available dataset into training and testing sets. This separation enables the model to learn patterns from the training data and assess its performance on unseen, independent data. A commonly employed practice is the 75-25 train-test split, where 75% of the data is allocated to the training set, and the remaining 25% is reserved for testing.

Model Implementation

After train and test splitting, a pipeline which consisted of Standard Scaler was fitted to several regression models such as Linear Regression, Decision Tree Regression, Random Forest Regression, XGBoost Regression, Light Gradient Boosting model, Adaboost regression, Elastic net regression and Bayesian Ridge regression.



Hyper-parameter tuning

For a few models, the best parameters were chosen using Grid Search, Stratified Search etc. These parameters resulted in getting better results.

Model evaluation

Model Evaluation is a crucial step in assessing the performance of machine learning models, and it involves utilizing various metrics to gauge their effectiveness. The R-squared (R2) metric provides an indication of how well the model captures the variance in the target variable, with higher values signifying better ifr. Root Mean Square Error (RMSE) quantifies the average magnitude of the model's errors, providing insight into the precision of its predictions. Mean Absolute Error (MAE) measures the average absolute difference between predicted and actual values, offering a robust evaluation of the model's overall accuracy. These metrics collectively contribute to a comprehensive understanding of the model's strengths and areas for improvement, guiding practitioners in refining their models for optimal performance.

Unit cases

Test Case Description	Prerequisite	Expected Result
Verify whether the Application URL is accessible to the user	Application URL should be defined	Application URL should be accessible to the user
Verify whether the Application loads completely for the user when the URL is accessed	1. Application URL is accessible 2. Application is deployed	The Application should load completely for the user when theURL is accessed
Verify whether the user is able to see input fields.	Application is accessible	User should be able to see input fields
Verify whether user is able to edit all input fields	Application is accessible	User should be able to edit all input fields
Verify whether user gets Submit button to submit the inputs	Application is accessible	User should get Submit button to submit the inputs
Verify whether user is presented with results on clicking submit	Application is accessible	User should be presented with results on clicking submit
Verify whether the results are in accordance to the selections user made	Application is accessible	The results should be in accordance to the selections user made