Prediction of LC50 value using Quantitative Structure Activity Relationship models

ARCHITECTURE

Document Control Version

Date Issued	Version	Description	Author
5/02/2024	1	Introduction & Architecture defined	Abhilash S Bharadwaj
7/02/2024	2	Unit Test Cases defined and appended	Abhilash S Bharadwaj

Contents

Introduction

What is a Low-Level Design Document?

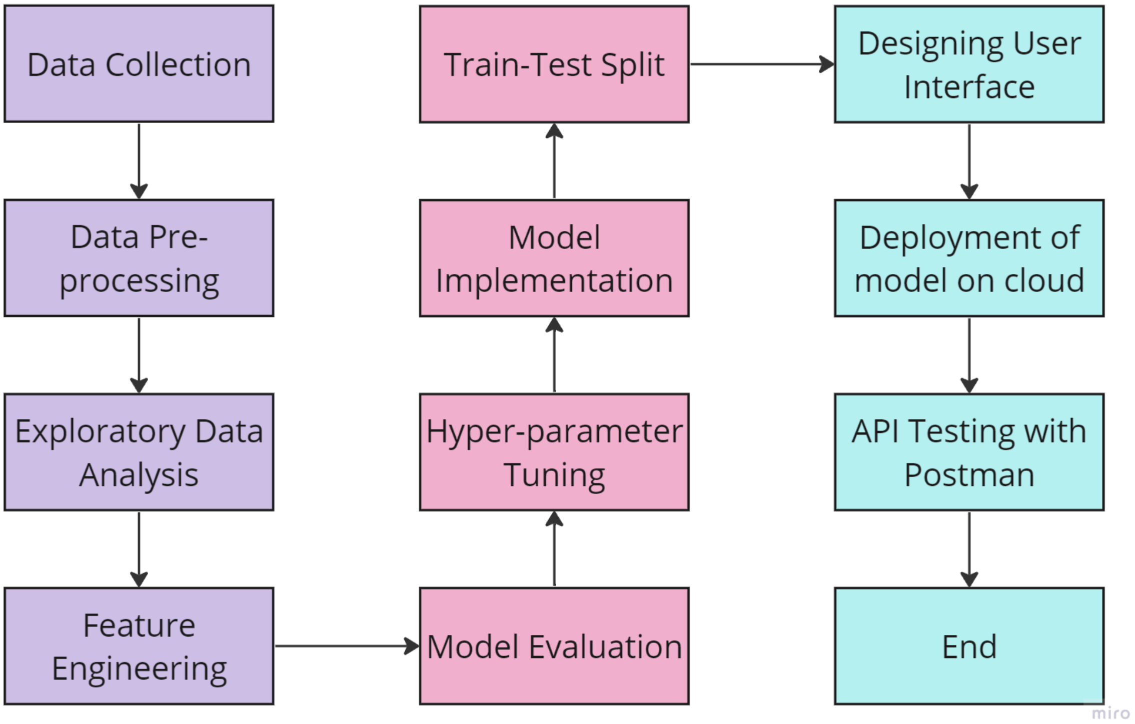
The goal of LLD or a low-level design document is to give the internal logic of the actual program code for Metro Interstate Traffic Volume Prediction. It will explain the purpose and features of the system, the interfaces of the system, what the system will do, the constraints under which it must operate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

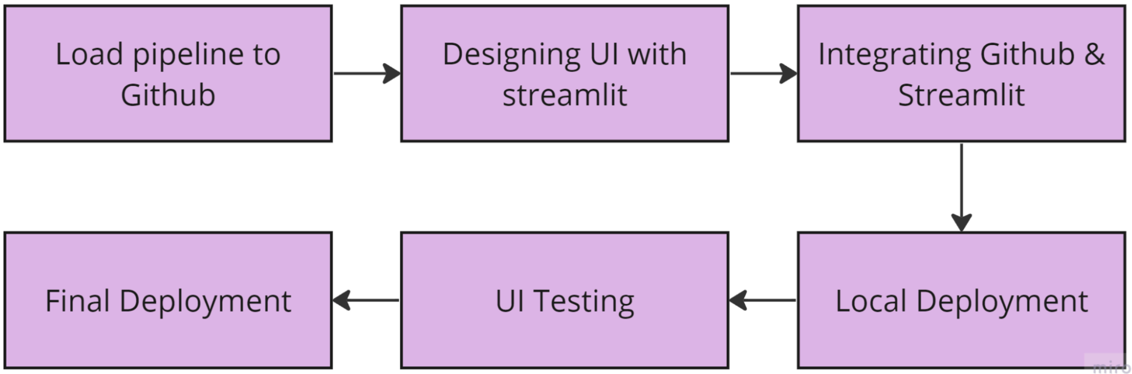
The main objective of the project is to predict if traffic volume is high or low on a particular date. Weather circumstances, special days like holidays, daytime (morning, afternoon, night and etc.), a temperature, a weekday, a numeric percentage of cloud cover are vital attributes for predicting traffic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

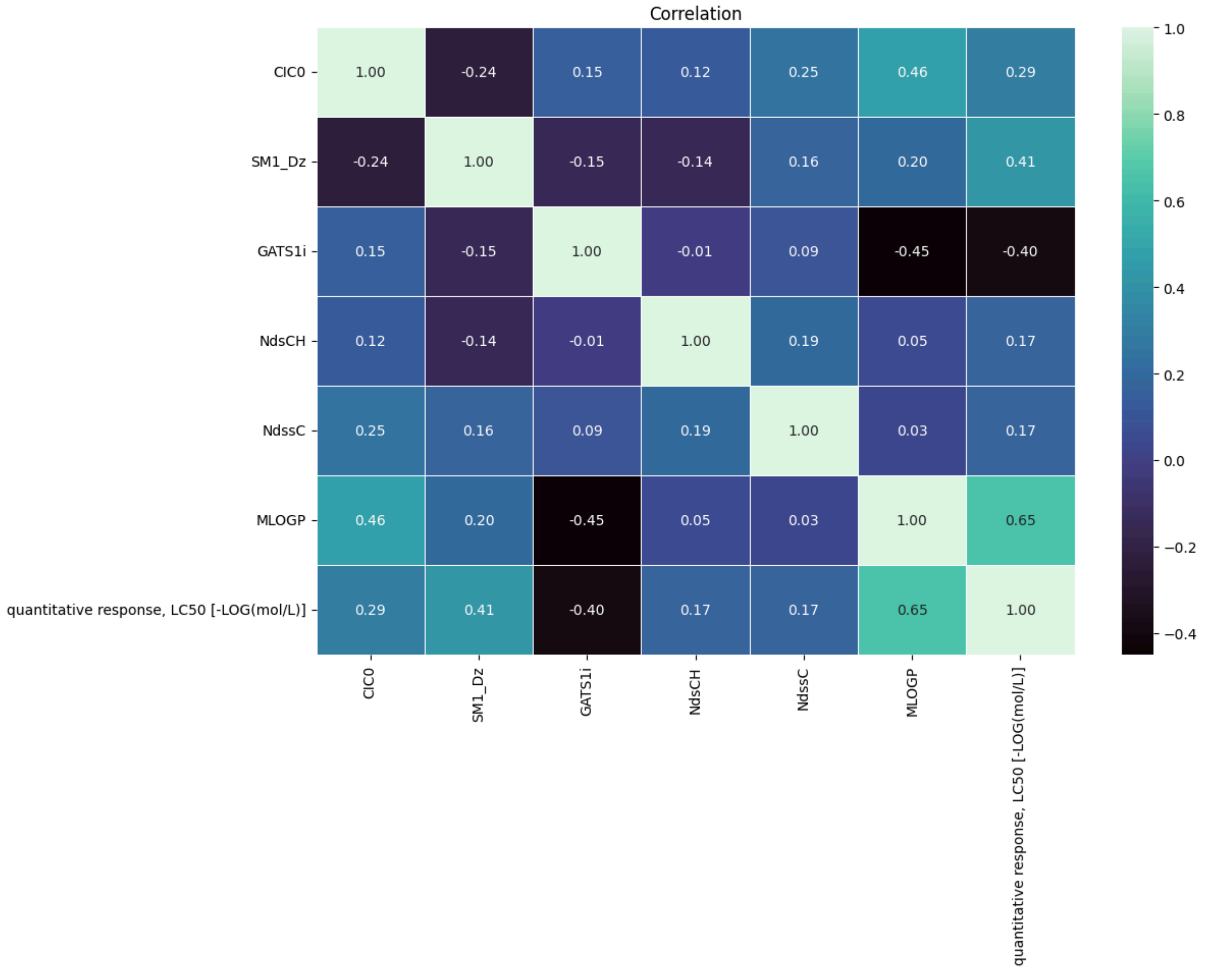
The dataset employed for this analysis encompasses a diverse array of 908 chemicals, each meticulously scrutinized to unveil insights into their acute toxicity towards the fathead minnow (Pimephales promelas). The central metric under examination is the LC50 data, a crucial parameter denoting the concentration at which 50% of the test fish succumb to mortality within a 96-hour period. This specific temporal window is essential for capturing the short-term effects and immediate responses of the fathead minnow to various chemical exposures. In the pursuit of developing robust predictive models, a sophisticated approach was adopted, focusing on quantitative regression modeling. The intricacies of chemical behavior and toxicity were dissected through the incorporation of six carefully selected molecular descriptors. Each descriptor was chosen for its unique contribution to unraveling the chemical complexities associated with acute toxicity. These molecular descriptors act as quantitative representations of various chemical properties, providing a structured basis for the regression models. The meticulous selection of descriptors aims to capture the nuanced relationships between the molecular features of chemicals and their corresponding LC50 values. This process ensures that the predictive models are not only accurate but also capable of discerning the specific molecular attributes that contribute significantly to the observed toxicity in fathead minnows.

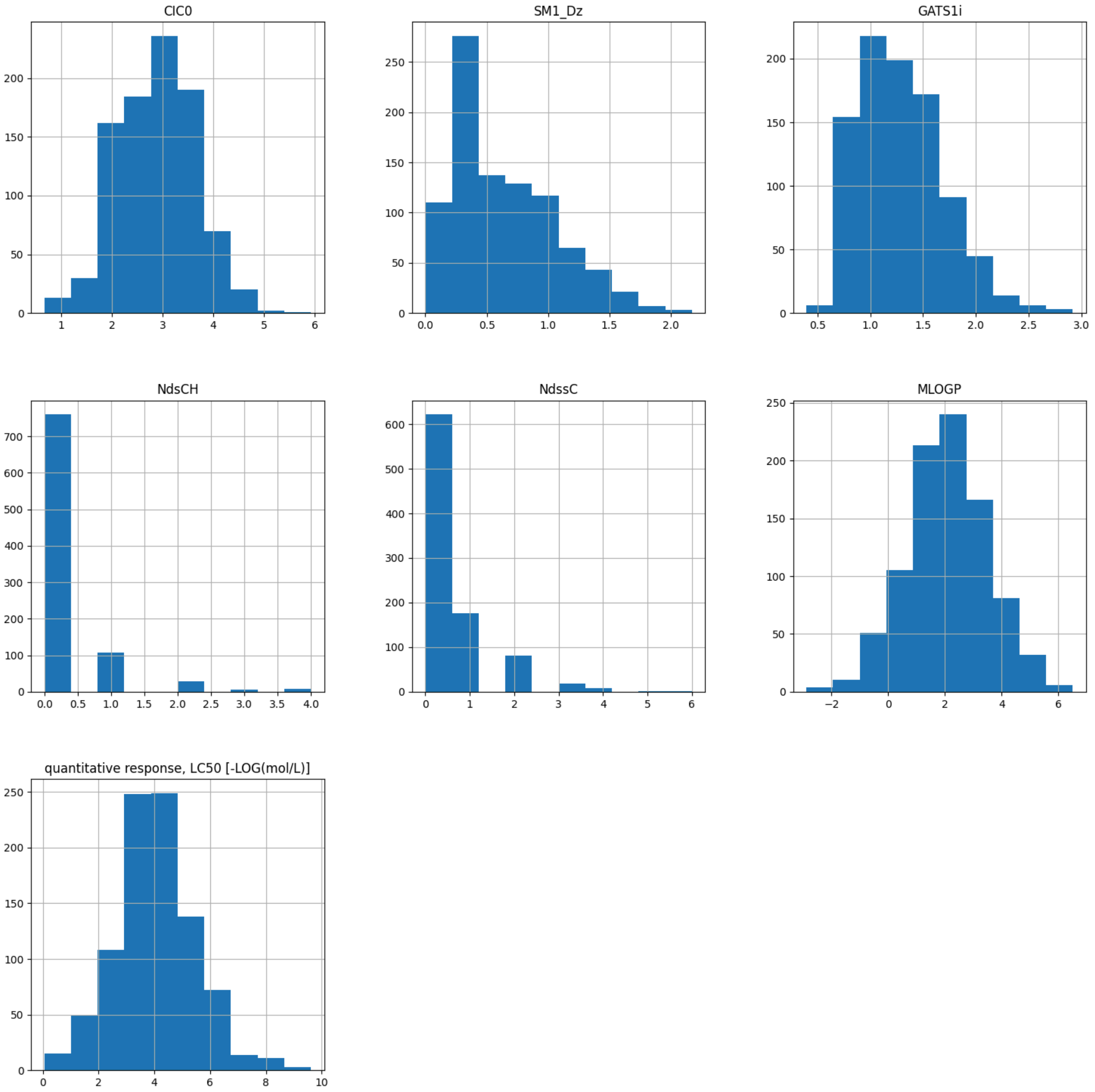
Data pre-processing

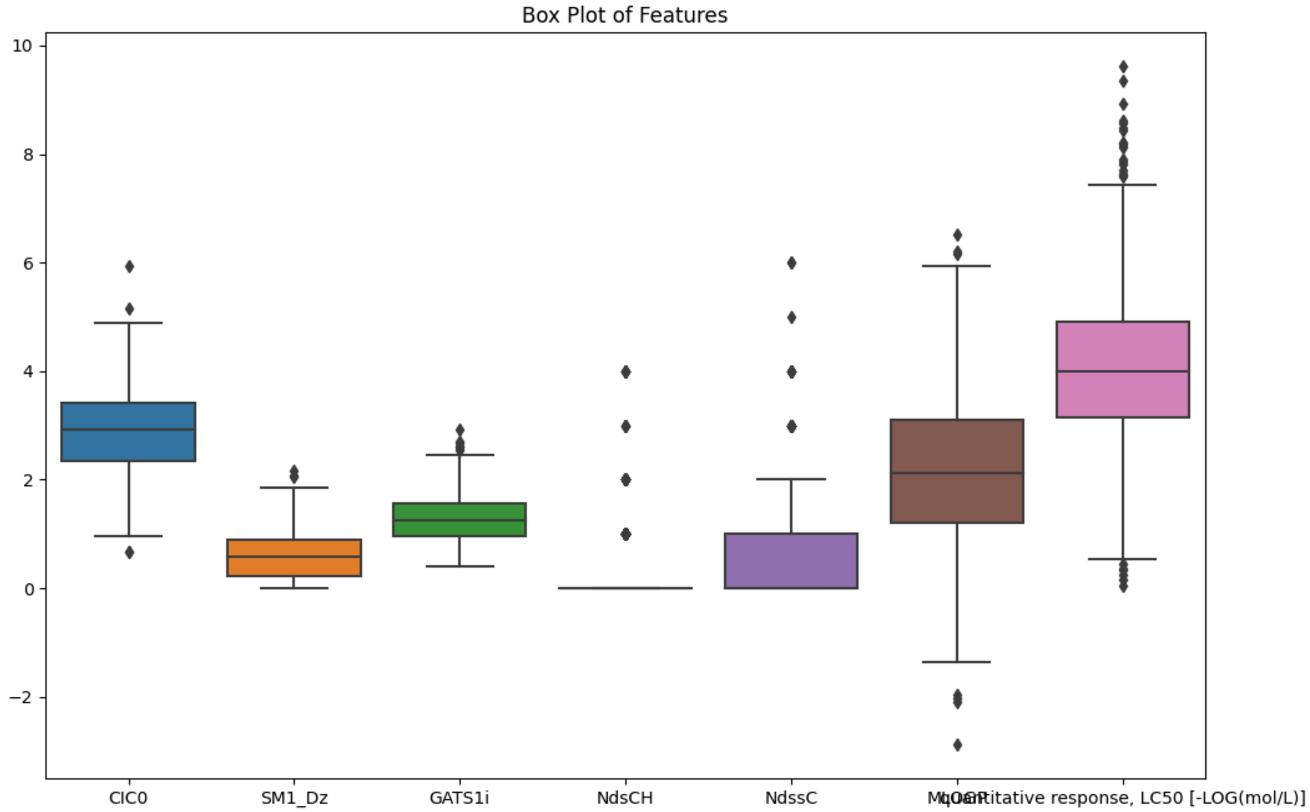
During this crucial step in our analysis, we systematically investigated the integrity of our dataset to ensure its reliability and suitability for modeling. The primary objectives included scrutinizing for missing data, identifying any instances of duplications, and comprehensively understanding the data types associated with each feature. Firstly, we meticulously examined the dataset to ascertain the presence of missing values. It is essential to ensure that all required information is available for meaningful analysis and modeling. Fortunately, our investigation revealed that the dataset was complete, devoid of any missing values that could potentially compromise the accuracy of our findings. In addition to handling missing data, we conducted a thorough evaluation to identify and address any duplicate values within the dataset. Duplicate entries can skew analyses and model training, potentially leading to biased results. Fortunately, our scrutiny confirmed the absence of any duplicate instances, providing confidence in the dataset's uniqueness and reliability. Furthermore, we delved into the data types of each feature to gain insights into their inherent characteristics. This step is crucial for understanding the nature of the variables and ensuring compatibility with the chosen modeling techniques.

Exploratory Data Analysis

In this pivotal step of our analysis, we delved into the interrelationships among the features through the exploration of correlation. Correlation, as a statistical measure, provides insight into the extent to which two variables, whether independent features or the target variable (dependent variable), change in tandem. This analysis allows us to discern patterns and potential dependencies that can significantly impact our modeling efforts. The first facet of our exploration involved assessing the correlation between the independent features and the target variable. Understanding these relationships is crucial as it helps identify which features may have a more pronounced impact on predicting the target variable,




CIC0
0
50
100
150
200
1
2
3
4
5
6
SM1_Dz
0
50
100
150
200
250
0.0
0.5
1.0
1.5
2.0
GATS1i
0
50
100
150
200
0.5
1.0
1.5
2.0
2.5
3.0
NdsCH
0
100
200
300
400
500
600
700
0.0
0.5
1.0
1.5
2.0
2.5
3.0
3.5
4.0
NdssC
0
100
200
300
400
500
600
0
1
2
3
4
5
6
MLOGP
0
50
100
150
200
250
−2
0
2
4
6
quantitative response, LC50 [-LOG(mol/L)]
0
50
100
150
200
250
0
2
4
6
8
10



Box plots are effective visual tools for identifying outliers within a dataset. These graphical representations provide a succinct summary of the distribution of data, showcasing key statistical measures such as the median, quartiles, and potential outliers. In a box plot, outliers are typically depicted as individual points beyond a certain range from the upper or lower quartile. By visually isolating these points, analysts can pinpoint instances where data points deviate significantly from the overall distribution.

Feature Engineering

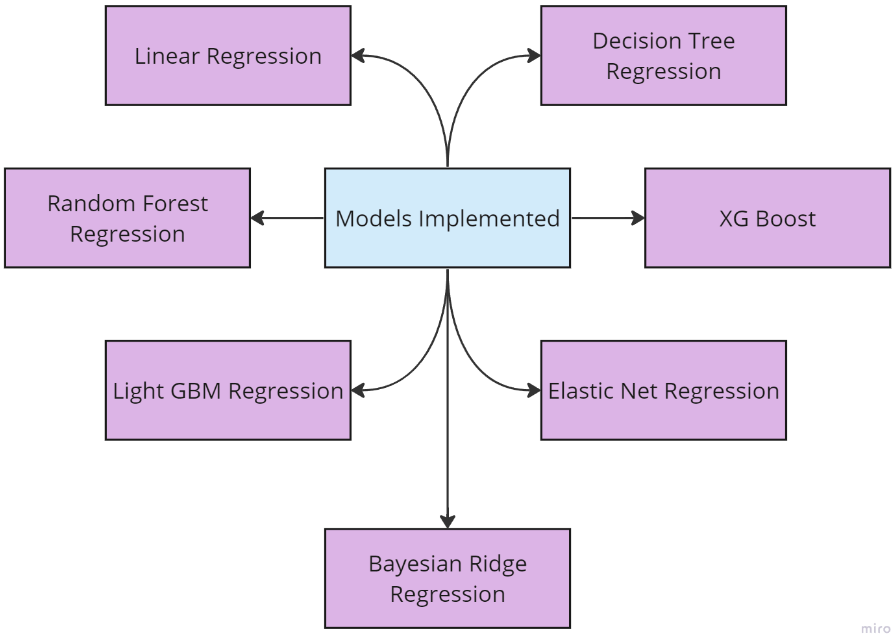
In this step the datatypes of two features such as ‘NdsCH’ & ‘NdssC’ were changed. Because, at first, the datatypes of these two features were ‘int’ but as our domain or the dataset end goal is based on regression type, we converted these int datatype to ‘float’ datatype for the betterment of the models.

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.



A total of 7 different regression models were implemented for this project. They are:

1. **Linear Regression:** Linear regression is a simple yet powerful technique that establishes a linear relationship between input features and the target variable, making it a go-to choice for straightforward modeling with interpretable results.

2. **Decision Tree Regression:** Decision tree regression partitions the input space into regions and predicts the target variable based on the average of training points within each region, offering flexibility and the ability to capture complex relationships in data.

3. **Random Forest (RF):** Random Forest combines multiple decision trees to enhance predictive performance and mitigate overfitting, making it robust for various datasets and particularly effective in handling high-dimensional feature spaces.

4. **LightGBM:** LightGBM, a gradient boosting framework, excels in handling large datasets and high-dimensional features, utilizing a leaf-wise growth strategy and gradient-based optimization to achieve fast and efficient training.

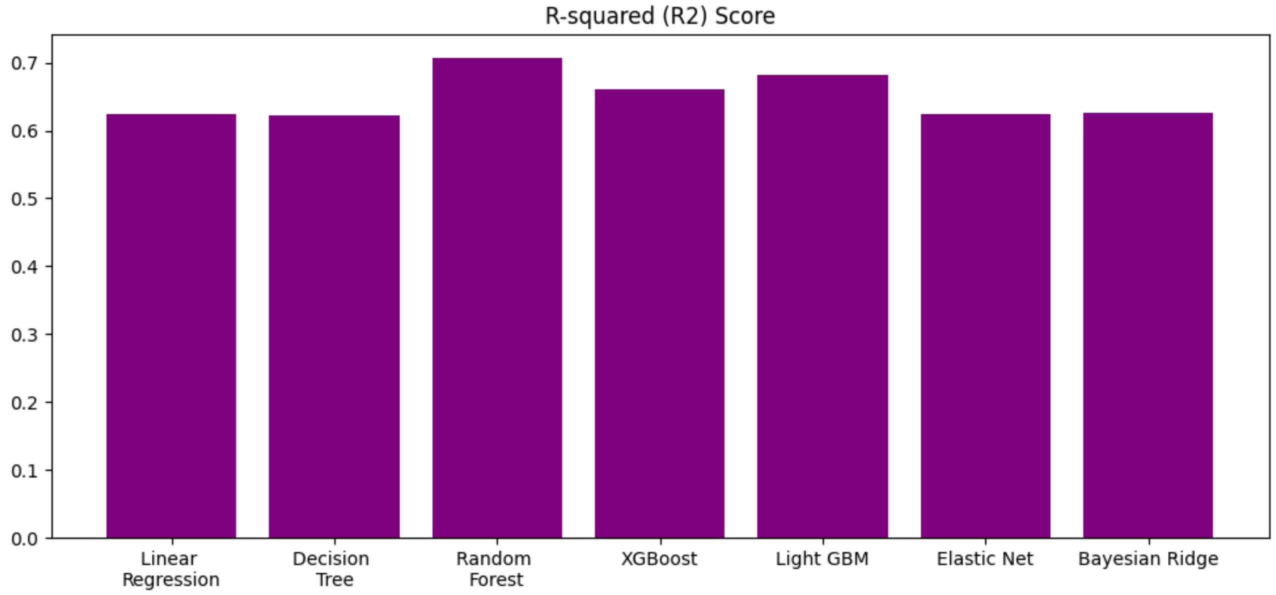
5. **XGBoost:** XGBoost is an optimized gradient boosting algorithm that delivers high accuracy and speed through parallel computation, regularization techniques, and a tree-pruning strategy, making it a popular choice in data science competitions.

6. **Bayesian Ridge Regression:** Bayesian Ridge Regression introduces Bayesian principles to linear regression, providing a probabilistic framework for parameter estimation, which helps handle multicollinearity and uncertainty in the data.

7. **Elastic Net Regression:** Elastic Net Regression combines L1 (Lasso) and L2 (Ridge) regularization, striking a balance between feature selection and model stability, making it effective in scenarios where collinearity and variable importance are both considerations.

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.



The model evaluation results showcase the performance metrics of various regression algorithms applied to the dataset. Linear Regression and Decision Tree models exhibit comparable R-Squared values, indicating their ability to capture a portion of the variance in the target variable. However, Random Forest emerges as the top performer with a significantly higher R-Squared value of 0.706, suggesting its superior ability to explain the variance in the data.

In terms of Mean Squared Error (MSE), Random Forest outshines other models, demonstrating the smallest error in predicting the target variable. This lower MSE aligns with the superior predictive accuracy of Random Forest, highlighting its robustness in minimizing prediction errors. Similarly, the Root Mean Squared Error (RMSE) and Mean

Absolute Error (MAE) reinforce the dominance of Random Forest, as it consistently achieves the lowest values across both metrics.

XGBoost and Light GBM, two gradient boosting techniques, also exhibit competitive performance, showcasing their effectiveness in capturing complex patterns within the data. Elastic Net and Bayesian Ridge, despite yielding slightly lower R-Squared values, demonstrate commendable performance, striking a balance between model simplicity and predictive accuracy.

In summary, the comprehensive evaluation of these regression models provides valuable insights into their strengths and weaknesses. While Random Forest stands out as the top-performing model based on various metrics, the choice of the most suitable algorithm ultimately depends on specific project requirements, interpretability needs, and computational considerations. This analysis lays a foundation for informed model selection, guiding practitioners in choosing the optimal regression technique for their specific use case.