Machine Learning Parameters:

In our research paper, we employed several regression models to investigate and analyze the given data. These models included Decision Tree Regression, K Nearest Neighbors Regression, Linear Regression, Random Forest Regressor and XGBoost Regressor. Each model was configured with specific parameters that contribute a crucial role in their performance.

The Decision Tree Regression model was constructed using the 'squared\_error' criterion, which minimized the sum of squared errors during the tree construction process. The maximum depth of the decision tree was not restricted ('None'), allowing the tree to grow until all the nodes were pure or until a specified stopping criterion was met. Additionally, the best split strategy ('best') was utilized to determine the optimal splitting point at each node.

For the K Nearest Neighbors Regression model, the number of neighbors considered was set to 5 ('n\_neighbors: 5'), and the distance metric used for measuring similarity between data points was the Euclidean distance ('metric: minkowski'). The weights assigned to the neighbors were uniform ('weights: uniform'), meaning that all neighboring points had equal influence on the prediction.

In the case of Linear Regression, no specific parameter values were provided. The default settings were employed, including the option for fitting the intercept term ('fit\_intercept: True') and allowing the model to utilize multiple CPU cores if available ('n\_jobs: None').

The Random Forest Regressor was built with 100 decision trees ('n\_estimators: 100') and employed the squared error criterion for node splitting ('criterion: squared\_error'). The maximum number of features considered at each split was set to 1.0, indicating that all features were eligible for selection ('max\_features: 1.0'). The bootstrap sampling technique was applied to create each tree ('bootstrap: True'), and no restrictions were imposed on the maximum depth of the trees ('max\_depth: None').

Lastly, the XGBoost Regressor utilized the objective function of 'reg:squarederror' to optimize the squared error loss during training. It employed 100 gradient boosted trees ('n\_estimators: 100') and had no specified random seed for reproducibility ('random\_state: None'). Other parameters were left with their default values.