Here we try to improve the r2 value for different types of kernel by passing hyper tuning parameter and then to find the best model out of it.

1. **MULTIPLE LINEAR REGRESSION**

R2 – 0.93

1. **SUPPORT VECTOR MACHINE**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **S.NO** | **HYPERTUNING PARAMETER** | | | **KERNEL TYPE** | | | |
| **C** | **Max\_iter** | **EPSILON** | **LINEAR**  **(r score)** | **RBF**  **(r score)** | **POLY**  **(r score)** | **SIGMOID**  **(r score)** |
| 1 | 1.0 | -1.0 | 0.1 | 0.8950 | -0.0574 | -0.0571 | -0.0572 |
| 2 | 10 | -1.0 | 0.5 | -2.4371 | -0.0568 | -0.0536 | -0.0547 |
| 3 | 100 | -1.0 | 1.0 | -357.07 | -0.0507 | -0.0198 | -0.0304 |
| 4 | 500 | 1000 | 0.1 | 0.6158 | -0.0243 | 0.1146 | 0.0705 |
| 5 | 1000 | 2000 | 0.5 | -7.3638 | 0.0067 | 0.2661 | 0.1850 |
| 6 | 2000 | 4000 | 1.0 | 0.6176 | 0.0675 | 0.4810 | 0.3970 |
| 7 | 3000 | 6000 | 0.1 | 0.5113 | 0.1232 | 0.6370 | 0.5913 |
| 8 | 4000 | 8000 | 0.5 | -7.5174 | 0.1723 | 0.7326 | 0.6282 |
| 9 | 6000 | 12000 | 1.0 | 0.5000 | 0.2452 | 0.8226 | 0.7972 |
| 10 | 10000 | 20000 | 0.1 | -7.8333 | 0.3718 | 0.8129 | 0.8535 |

Support Vector machine with linear kernel type and hyper parameter

(C=1.0, max\_iter=-1.0, Epsilon=0.1) has value as 0.8950

**C-Regularization Parameter**:

The strength of the regularization is inversely proportional to C, It must be strictly positive.

High C value 🡪 Overfitting

Low C value **🡪**reduce overfitting and Training accuracy

**Max\_iter-**

Hard limit on iterations within solver, or -1 for no limit. This value prevent the algorithm from running indefinitely, especially in cases where convergence is difficult to achieve

. High max\_iter value 🡪 Better accuracy

Low max\_iter value 🡪 Less accurate

**Epsilon**

Epsilon specifies a margin around the true target values within which predictions are considered acceptable. If the predicted value lies within this margin, no penalty is assigned during trainin, Must be non-negative

High epsilon value🡪better handle unseen data but could reduce precision

Low epsilon value🡪 Improve accuracy but may lead to overfitting

1. **DECISION TREE:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **S.No** | **CRITERION** | **MAX FEATURES** | **SPLITTER** | **R SCORE** |
| 1 | Squared\_error | None | Best | 0.9042 |
| 2 | Squared\_error | None | random | 0.2729 |
| 3 | Squared\_error | Sqrt | Best | 0.8134 |
| 4 | Squared\_error | Sqrt | random | 0.5575 |
| 5 | Squared\_error | log2 | Best | 0.5837 |
| 6 | Squared\_error | log2 | random | 0.3710 |
| 7 | friedman\_mse | None | Best | 0.9114 |
| 8 | friedman\_mse | None | random | 0.8972 |
| 9 | friedman\_mse | Sqrt | Best | 0.8178 |
| 10 | friedman\_mse | Sqrt | random | 0.3956 |
| 11 | friedman\_mse | log2 | Best | 0.4322 |
| 12 | friedman\_mse | log2 | random | 0.5461 |
| 13 | absolute\_error | None | Best | 0.9486 |
| 14 | absolute\_error | None | random | 0.9129 |
| 15 | absolute\_error | Sqrt | Best | 0.5174 |
| 16 | absolute\_error | Sqrt | random | 0.2755 |
| 17 | absolute\_error | log2 | Best | 0.5986 |
| 18 | absolute\_error | log2 | random | 0.3732 |
| 19 | Poisson | None | Best | 0.9161 |
| 20 | Poisson | None | random | 0.9100 |
| 21 | Poisson | Sqrt | Best | 0.7009 |
| 22 | Poisson | Sqrt | random | 0.8058 |
| 23 | Poisson | log2 | Best | 0.6698 |
| 24 | Poisson | log2 | random | 0.5628 |

Decision Tree with (Criterion=’absolute\_error’, splitter=’best’,max\_features=None) have value as 0.9486

**Criterion:**

The function is to measure the quality of a split.

**Squared Error:**

* It ensures that the tree prioritizes split that create groups with closer predicted values to the actual values
* It minimizes the L2 loss using the mean of each terminal node

**Absolute Error:**

* It minimizes the L1 loss using the median of each terminal node

**Friedman\_mse:**

* It is based on Friedman's improvement score, which evaluates potential splits by considering both the reduction in mean squared error and the weighted difference between the means of the left and right regions after the split

**Poisson:**

* The poisson criterion in decision trees is used for regression tasks, particularly when the target variable represents count data
* It evaluates the quality of a split by reducing the Poisson deviance, which is a measure of how well the model predicts the count data.

**Max\_Features:** It determines the maximum number of features to consider when looking for the best split at each node.

* **If set to an integer**, it specifies the exact number of features to consider.
* **If set to a float**, it represents a fraction of the total number of features (e.g., 0.5 means 50% of the features).
* **If set to "auto" or "sqrt"**, it uses the square root of the total number of features (common for classification tasks).
* **If set to "log2"**, it uses the base-2 logarithm of the total number of features.
* **If set to** None, all features are considered.

**Splitter:** The strategy used to select the feature and threshold for splitting each node.

* Supported strategies are “best” to choose the best split and “random” to choose the best random split.