**Machine Learning Regression Using R2 Value Method**

1. **Multiple Linear Regression –> R2 Value is 0.7894**
2. **Support Vector Machine:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **S.No** | **kernel** | **C** | **Hyper Tuned Parameters** | **r\_Score** |
| 1 | linear | 10 | regressor = Support Vector Regression (kernel = 'linear', C = 10) | 0.4625 |
| 2 | linear | 100 | regressor = Support Vector Regression (kernel = 'linear', C = 100) | 0.4625 |
| 3 | linear | 500 | regressor = Support Vector Regression (kernel = 'linear', C = 500) | 0.7631 |
| 4 | linear | 1000 | regressor = Support Vector Regression (kernel = 'linear', C = 1000) | 0.7649 |
| 5 | linear | 2000 | regressor = Support Vector Regression (kernel = 'linear', C = 2000) | 0.7440 |
| 6 | linear | 5000 | regressor = Support Vector Regression (kernel = 'linear', C = 5000) | 0.7414 |
| 7 | linear | 10000 | regressor = Support Vector Regression (kernel = 'linear', C = 10000) | 0.7414 |
| 8 | rbf | 10 | regressor = Support Vector Regression (kernel = 'rbf', C = 10) | -0.0323 |
| 9 | rbf | 100 | regressor = Support Vector Regression (kernel = 'rbf', C = 100) | 0.3200 |
| 10 | rbf | 500 | regressor = Support Vector Regression (kernel = 'rbf', C = 500) | 0.6643 |
| 11 | rbf | 1000 | regressor = Support Vector Regression (kernel = 'rbf', C = 1000) | 0.8102 |
| 12 | rbf | 2000 | regressor = Support Vector Regression (kernel = 'rbf', C = 2000) | 0.8548 |
| 13 | rbf | 5000 | regressor = Support Vector Regression (kernel = 'rbf', C = 5000) | 0.8748 |
| **14** | **rbf** | **10000** | **regressor = Support Vector Regression (kernel = 'rbf', C = 10000)** | **0.8780** |
| 15 | sigmoid | 10 | regressor = Support Vector Regression (kernel = 'sigmoid', C = 10) | 0.0393 |
| 16 | sigmoid | 100 | regressor = Support Vector Regression (kernel = 'sigmoid', C = 100) | 0.5276 |
| 17 | sigmoid | 500 | regressor = Support Vector Regression (kernel = 'sigmoid', C = 500) | 0.4446 |
| 18 | sigmoid | 1000 | regressor = Support Vector Regression (kernel = 'sigmoid', C = 1000) | 0.2875 |
| 19 | sigmoid | 2000 | regressor = Support Vector Regression (kernel = 'sigmoid', C = 2000) | -0.5940 |
| 20 | sigmoid | 5000 | regressor = Support Vector Regression (kernel = 'sigmoid', C = 5000) | -7.5300 |
| **21** | **sigmoid** | **10000** | **regressor = Support Vector Regression (kernel = 'sigmoid', C = 10000)** | **-34.1515** |
| 22 | poly | 10 | regressor = Support Vector Regression (kernel = 'poly', C = 10) | 0.0387 |
| 23 | poly | 100 | regressor = Support Vector Regression (kernel = 'poly', C = 100) | 0.6180 |
| 24 | poly | 500 | regressor = Support Vector Regression (kernel = 'poly', C = 500) | 0.8264 |
| 25 | poly | 1000 | regressor = Support Vector Regression (kernel = 'poly', C = 1000) | 0.8566 |
| 26 | poly | 2000 | regressor = Support Vector Regression (kernel = 'poly', C = 2000) | 0.8606 |
| 27 | poly | 5000 | regressor = Support Vector Regression (kernel = 'poly', C = 5000) | 0.8596 |
| 28 | poly | 10000 | regressor = Support Vector Regression (kernel = 'poly', C = 10000) | 0.8592 |

(**The Highest R2 Value is Highlighted in Color – Green** and **The Lowest R2 Value is Highlighted in Color – Red**)

1. **Decision Tree :**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **S.No** | **criterion** | **splitter** | **max\_features** | **Hyper Tuned Parameters** | **r\_Score** |
| 1 | squared\_error | best | NA | regressor = DecisionTreeRegressor (criterion = 'squared\_error', splitter = 'best') | 0.6818 |
| 2 | squared\_error | random | NA | regressor = DecisionTreeRegressor (criterion = 'squared\_error', splitter = 'random') | 0.6996 |
| 3 | friedman\_mse | best | NA | regressor = DecisionTreeRegressor (criterion = 'friedman\_mse', splitter = 'best') | 0.6962 |
| 4 | friedman\_mse | random | NA | regressor = DecisionTreeRegressor (criterion = 'friedman\_mse', splitter = 'random') | 0.6942 |
| 5 | absolute\_error | best | NA | regressor = DecisionTreeRegressor (criterion = 'absolute\_error', splitter = 'best') | 0.6825 |
| 6 | absolute\_error | random | NA | regressor = DecisionTreeRegressor (criterion = 'absolute\_error', splitter = 'random') | 0.7354 |
| 7 | poisson | best | NA | regressor = DecisionTreeRegressor (criterion = 'poisson', splitter = 'best') | 0.7181 |
| 8 | poisson | random | NA | regressor = DecisionTreeRegressor (criterion = 'poisson', splitter = 'random') | 0.7010 |
| 9 | squared\_error | best | sqrt | regressor = DecisionTreeRegressor (criterion = 'squared\_error', splitter = 'best', max\_features = 'sqrt') | 0.6849 |
| 10 | squared\_error | random | sqrt | regressor = DecisionTreeRegressor (criterion = 'squared\_error', splitter = 'random', max\_features = 'sqrt') | 0.6777 |
| 11 | friedman\_mse | best | sqrt | regressor = DecisionTreeRegressor (criterion = 'friedman\_mse', splitter = 'best', max\_features = 'sqrt') | 0.6693 |
| 12 | friedman\_mse | random | sqrt | regressor = DecisionTreeRegressor (criterion = 'friedman\_mse', splitter = 'random', max\_features = 'sqrt') | 0.7138 |
| 13 | absolute\_error | best | sqrt | regressor = DecisionTreeRegressor (criterion = 'absolute\_error', splitter = 'best', max\_features = 'sqrt') | 0.7343 |
| 14 | absolute\_error | random | sqrt | regressor = DecisionTreeRegressor (criterion = 'absolute\_error', splitter = 'random', max\_features = 'sqrt') | 0.7208 |
| 15 | poisson | best | sqrt | regressor = DecisionTreeRegressor (criterion = 'poisson', splitter = 'best', max\_features = 'sqrt') | 0.7323 |
| 16 | poisson | random | sqrt | regressor = DecisionTreeRegressor (criterion = 'poisson', splitter = 'random', max\_features = 'sqrt') | 0.7224 |
| 17 | squared\_error | best | log2 | regressor = DecisionTreeRegressor (criterion = 'squared\_error', splitter = 'best', max\_features = 'log2') | 0.7047 |
| 18 | squared\_error | random | log2 | regressor = DecisionTreeRegressor (criterion = 'squared\_error', splitter = 'random', max\_features = 'log2') | 0.6951 |
| 19 | friedman\_mse | best | log2 | regressor = DecisionTreeRegressor (criterion = 'friedman\_mse', splitter = 'best', max\_features = 'log2') | 0.6516 |
| **20** | **friedman\_mse** | **random** | **log2** | **regressor = DecisionTreeRegressor (criterion = 'friedman\_mse', splitter = 'random', max\_features = 'log2')** | **0.6375** |
| **21** | **absolute\_error** | **best** | **log2** | **regressor = DecisionTreeRegressor (criterion = 'absolute\_error', splitter = 'best', max\_features = 'log2')** | **0.7470** |
| 22 | absolute\_error | random | log2 | regressor = DecisionTreeRegressor (criterion = 'absolute\_error', splitter = 'random', max\_features = 'log2') | 0.6686 |
| 23 | poisson | best | log2 | regressor = DecisionTreeRegressor (criterion = 'poisson', splitter = 'best', max\_features = 'log2') | 0.6951 |
| 24 | poisson | random | log2 | regressor = DecisionTreeRegressor (criterion = 'poisson', splitter = 'random', max\_features = 'log2') | 0.7379 |

(**The Highest R2 Value is Highlighted in Color – Green** and **The Lowest R2 Value is Highlighted in Color – Red**)

1. **Random Forest :**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **S.No** | **n\_estimators** | **random\_state** | **criterion** | **max\_features** | **Hyper Tuned Parameters** | **r\_Score** |
| 1 | 50 | 0 | squared\_error | None | regressor = RandomForestRegressor (n\_estimators = 50, random\_state = 0, criterion = 'squared\_error', max\_features = None) | 0.8496 |
| 2 | 100 | 0 | squared\_error | None | regressor = RandomForestRegressor (n\_estimators = 100, random\_state = 0, criterion = 'squared\_error', max\_features = None) | 0.8536 |
| 3 | 50 | 0 | friedman\_mse | None | regressor = RandomForestRegressor (n\_estimators = 50, random\_state = 0, criterion = 'friedman\_mse', max\_features = None) | 0.8497 |
| 4 | 100 | 0 | friedman\_mse | None | regressor = RandomForestRegressor (n\_estimators = 100, random\_state = 0, criterion = 'friedman\_mse', max\_features = None) | 0.8538 |
| 5 | 50 | 0 | absolute\_error | None | regressor = RandomForestRegressor (n\_estimators = 50, random\_state = 0, criterion = 'absolute\_error', max\_features = None) | 0.8536 |
| 6 | 100 | 0 | absolute\_error | None | regressor = RandomForestRegressor (n\_estimators = 100, random\_state = 0, criterion = 'absolute\_error', max\_features = None) | 0.8527 |
| **7** | **50** | **0** | **poisson** | **None** | **regressor = RandomForestRegressor (n\_estimators = 50, random\_state = 0, criterion = 'poisson', max\_features = None)** | **0.8493** |
| 8 | 100 | 0 | poisson | None | regressor = RandomForestRegressor (n\_estimators = 100, random\_state = 0, criterion = 'poisson', max\_features = None) | 0.8528 |
| 9 | 50 | 0 | squared\_error | sqrt | regressor = RandomForestRegressor (n\_estimators = 50, random\_state = 0, criterion = 'squared\_error', max\_features = 'sqrt') | 0.8695 |
| 10 | 100 | 0 | squared\_error | sqrt | regressor = RandomForestRegressor (n\_estimators = 100, random\_state = 0, criterion = 'squared\_error', max\_features = 'sqrt') | 0.8710 |
| 11 | 50 | 0 | friedman\_mse | sqrt | regressor = RandomForestRegressor (n\_estimators = 50, random\_state = 0, criterion = 'friedman\_mse', max\_features = 'sqrt') | 0.8705 |
| 12 | 100 | 0 | friedman\_mse | sqrt | regressor = RandomForestRegressor (n\_estimators = 100, random\_state = 0, criterion = 'friedman\_mse', max\_features = 'sqrt') | 0.8712 |
| 13 | 50 | 0 | absolute\_error | sqrt | regressor = RandomForestRegressor (n\_estimators = 50, random\_state = 0, criterion = 'absolute\_error', max\_features = 'sqrt') | 0.8715 |
| 14 | 100 | 0 | absolute\_error | sqrt | regressor = RandomForestRegressor (n\_estimators = 100, random\_state = 0, criterion = 'absolute\_error', max\_features = 'sqrt') | 0.8713 |
| 15 | 50 | 0 | poisson | sqrt | regressor = RandomForestRegressor (n\_estimators = 50, random\_state = 0, criterion = 'poisson', max\_features = 'sqrt') | 0.8632 |
| 16 | 100 | 0 | poisson | sqrt | regressor = RandomForestRegressor (n\_estimators = 100, random\_state = 0, criterion = 'poisson', max\_features = 'sqrt') | 0.8680 |
| 17 | 50 | 0 | squared\_error | log2 | regressor = RandomForestRegressor (n\_estimators = 50, random\_state = 0, criterion = 'squared\_error', max\_features = 'log2') | 0.8695 |
| 18 | 100 | 0 | squared\_error | log2 | regressor = RandomForestRegressor (n\_estimators = 100, random\_state = 0, criterion = 'squared\_error', max\_features = 'log2') | 0.8710 |
| 19 | 50 | 0 | friedman\_mse | log2 | regressor = RandomForestRegressor (n\_estimators = 50, random\_state = 0, criterion = 'friedman\_mse', max\_features = 'log2') | 0.8705 |
| 20 | 100 | 0 | friedman\_mse | log2 | regressor = RandomForestRegressor (n\_estimators = 100, random\_state = 0, criterion = 'friedman\_mse', max\_features = 'log2') | 0.8712 |
| **21** | **50** | **0** | **absolute\_error** | **log2** | **regressor = RandomForestRegressor (n\_estimators = 50, random\_state = 0, criterion = 'absolute\_error', max\_features = 'log2')** | **0.8715** |
| 22 | 100 | 0 | absolute\_error | log2 | regressor = RandomForestRegressor (n\_estimators = 100, random\_state = 0, criterion = 'absolute\_error', max\_features = 'log2') | 0.8713 |
| 23 | 50 | 0 | poisson | log2 | regressor = RandomForestRegressor (n\_estimators = 50, random\_state = 0, criterion = 'poisson', max\_features = 'log2') | 0.8632 |
| 24 | 100 | 0 | poisson | log2 | regressor = RandomForestRegressor (n\_estimators = 100, random\_state = 0, criterion = 'poisson', max\_features = 'log2') | 0.8680 |

(**The Highest R2 Value is Highlighted in Color – Green** and **The Lowest R2 Value is Highlighted in Color – Red**)

**Overall Validation of Regression**

|  |  |  |  |
| --- | --- | --- | --- |
| **S.No** | **Type of Regression** | **r\_Score (Highest)** | **r\_Score (Lowest)** |
| **1** | **Multiple Linear Regression** | **0.7895** | **NA** |
| **2** | **Support Vector Regression** | **0.8780** | **-34.1515** |
| **3** | **Decision Tree Regressor** | **0.7470** | **0.6375** |
| **4** | **Random Forest Regressor** | **0.8715** | **0.8493** |

|  |  |
| --- | --- |
| **Result** | The **Highest R2 Value - 0.8780** gotten for **Support Vector Regression** from across all the Regressions done. |