**Machine Learning Algorithms and their r2 values model wise.**

1. Simple Linear Algorithm (r score value = 0.974099)
2. Multiple Linear Algorithm (r score value = 0.935868)
3. Support Vector Machine:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| S.No | HYPER PARAMETER | LINEAR  (r value) | RBF(NON LINEAR)  (r value) | POLY  (r value) | SIGMOID  (r value) |
| 1 | C10 | -2.43721 | -0.05580 | 0.02531 | -0.05761 |
| 2 | C100 | -357.07951 | -0.03023 | 0.46566 | -0.05878 |
| 3 | C500 | -8996.86064 | 0.05001 | 0.62077 | -0.06401 |
| 4 | C1000 | higher minus val | 0.16060 | 0.64032 | -0.07070 |
| 5 | C2000 | higher minus val | 0.28839 | 0.67174 | -0.08453 |
| 6 | C3000 | higher minus val | 0.39514 | 0.69099 | -0.09898 |

The SVM Regression use R2 value (non-linear (Rbf) and hyper parameter (C3000) = 0.8609

1. Decision Tree:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| S.No | Criterion value | Splitter value | R value | Notes |
| 1 | ***squared\_error*** | Best | 0.93730, 0.94027 | While re-running r value getting increased. |
| 2 | ***squared\_error*** | random | 0.85613 |  |
| 3 | ***friedman\_mse*** | Best | 0.89715 |  |
| 4 | ***friedman\_mse*** | random | 0.93740 |  |
| 5 | ***absolute\_error*** | Best | 0.95546 |  |
| 6 | ***absolute\_error*** | random | 0.89100 |  |
| 7 | ***poisson*** | Best | 0.93215 |  |
| 8 | ***poisson*** | random | 0.91728 |  |

Using DecisionTreeRegressor, with Criterion = absolute\_error, splitter = best we got high accuracy as 0.95546

Note: For each and every run the r\_score values getting changed and not constant.