

### **SVM Regression:**

kernel	Hyper parameter	rbf(r2 value)	linear (r2 value)	poly (r2 value)	sigmoid(r2 value)
1	C=10	-0.055800	-2.43721	0.025312	-0.057615
2	C=100	-0.030235	-357.0795	0.465662	-0.058780
3	C=500	0.050018	Took long time	0.620773	-0.064016
4	C=1000	0.160600	Took long time	0.64032	-0.070701
5	C=2000	0.288395	Took long time	0.671747	-0.084533
6	C=2500	0.334722	Took long time	0.699168	-0.09168
7	C=3000	0.395140	Took long time	0.690998	-0.098982

### **SVM with Standardisation:**

kernel	Hyper parameter	rbf(r2 value)	linear (r2 value)	poly (r2 value)	sigmoid(r2 value)
1	C=10	-0.05680	-0.039644	-0.053667	-0.054719
2	C=100	-0.05072	0.106468	-0.019802	-0.030453
3	C=500	-0.024323	0.592897	0.1146848	0.070572
4	C=1000	0.006768	0.780283	0.266163	0.18506
5	C=2000	0.067515	0.876772	0.481002	0.397065
6	C=2500	0.096004	0.890814	0.566134	0.503673
7	C=3000	0.123227	0.895674	0.637006	0.591363

**SVM gives low scores compared to Multilinear, So it's not a good model to save.**

**Multilinear r2 value:0 . 93586**

## DECISION TREE Regression

Without any parameter r2 value for Decision Tree: 0.90831

	criterion	splitter	R value
1	squared_error	best	0.91623
2	squared_error	random	0.96013
3	friedman_mse	best	0.89566
4	friedman_mse	random	0.73037
5	absolute_error	best	0.96987
6	absolute_error	random	0.852880
7	poisson	best	0.92322
8	poisson	random	0.92592

**R\_score value: 0.96** nearly 1 good model before save the model checking with RandomForest algorithm

## RandomForestRegressor

Without any parameter r2 value for RandomForest Regressor : 0.93428

	criterion	n_estimators	R value
1	squared_error	50	0.9446
2	squared_error	100	0.9460
3	friedman_mse	50	0.9388
4	friedman_mse	100	0.9412
5	absolute_error	50	0.9401
6	absolute_error	100	0.94590
7	poisson	50	0.9463
8	poisson	100	0.9413

We get 0.94 r2 value in Randomforest. It's a good model but the decision tree r2 value is better than this.

**Best model:**

Gonna Save Decision Tree model **0.96987**