

# Hyper turning Parameter

for ML Algorithm

Dataset = 50\_Startup

1. Multiple Linear Regression -  $R^2$  value = 0.935

2. Support Vector Machine

C-Support Vector Classification.

**C:** float, default=1.0

**Kernel:** {'linear', 'poly', 'rbf', 'sigmoid', 'precomputed'} or callable, default='rbf'

S. No	Hyper parameter 'C'	Linear $R^2$ value	rbf value	$R^2$	Poly $R^2$ value	Sigmoid $R^2$ value
1	10	-9774	-0.057		-7.853	-0.052
2	100	-7624	-0.057		-7.853	-0.018
3	1000	-2028	-0.059		-7.372	-0.322
4	2000	-2795	-0.055		-2.446	-1.729
5	3000	-2381	-0.050		-4.164	-3.489

Here  $R^2$  value is not upto mark.

This model is **Not Good** for this data.

3. Decision Tree

**Criterion:** {"squared\_error", "friedman\_mse", "absolute\_error", "poisson"}, default="squared\_error"

**Splitter:** {"best", "random"}, default="best"

**max\_features :** int, float or {"sqrt", "log2", "auto"}, default=None

S. No	Criterion	Splitter	max_features	R <sup>2</sup> value
1				0.916
2	squared_error	best		0.882
3	squared_error	best	sqrt	0.234
4	squared_error	best	Log2	0.638
5	squared_error	best	auto	0.903
6	squared_error	random		0.845
7	squared_error	random	sqrt	0.473
8	squared_error	random	Log2	0.216
9	squared_error	random	auto	0.281
10	friedman_mse	best		0.926
11	friedman_mse	best	sqrt	0.925
12	friedman_mse	best	Log2	-0.236
13	friedman_mse	best	auto	0.921
14	friedman_mse	random		0.904
15	friedman_mse	random	sqrt	0.746
16	friedman_mse	random	Log2	0.458
17	friedman_mse	random	auto	0.861
18	absolute_error	best		0.935
19	absolute_error	best	sqrt	0.560
20	absolute_error	best	Log2	0.561
21	absolute_error	best	auto	0.941
22	absolute_error	random		0.934
23	absolute_error	random	sqrt	0.713
24	absolute_error	random	Log2	0.629
25	absolute_error	random	auto	0.925
26	poisson	best		0.712
27	poisson	best	sqrt	0.896
28	poisson	best	Log2	0.701
29	poisson	best	auto	0.926
30	poisson	random		0.879
31	poisson	random	sqrt	0.847
32	poisson	random	Log2	0.156
33	poisson	random	auto	0.933

Here **R<sup>2</sup> value** is upto mark.

This model (**absolute\_error, best, auto**) is **Good** for this data.

#### 4. Random Forest

**n\_estimators: int, default=100**

**Criterion: {"squared\_error", "absolute\_error", "friedman\_mse", "poisson"}, default="squared\_error"**

**max\_features : {"sqrt", "log2", None, auto}, int or float, default=1.0**

**random\_state: int, RandomState instance or None, default=None**

S. No	n_estimators	Criterion	max_features	R <sup>2</sup> value
1	100	squared_error		0.941
2	100	squared_error	sqrt	0.762
3	100	squared_error	Log2	0.787
4	100	squared_error	auto	0.948
5	50	squared_error		0.934
6	50	squared_error	sqrt	0.817
7	50	squared_error	Log2	0.826
8	50	squared_error	auto	0.945
9	20	squared_error		0.944
10	20	squared_error	sqrt	0.713
11	20	squared_error	Log2	0.784
12	20	squared_error	auto	0.935
13	100	absolute_error		0.942
14	100	absolute_error	sqrt	0.840
15	100	absolute_error	Log2	0.811
16	100	absolute_error	auto	0.942
17	50	absolute_error		0.944
18	50	absolute_error	sqrt	0.777
19	50	absolute_error	Log2	0.829
20	50	absolute_error	auto	0.935
21	20	absolute_error		0.933
22	20	absolute_error	sqrt	0.819
23	20	absolute_error	Log2	0.783
24	20	absolute_error	auto	0.927
25	100	friedman_mse		0.940
26	100	friedman_mse	sqrt	0.802
27	100	friedman_mse	Log2	0.798
28	100	friedman_mse	auto	0.943
29	50	friedman_mse		0.941
30	50	friedman_mse	sqrt	0.820
31	50	friedman_mse	Log2	0.782
32	50	friedman_mse	auto	0.934
33	20	friedman_mse		0.926
34	20	friedman_mse	sqrt	0.861
35	20	friedman_mse	Log2	0.678
36	20	friedman_mse	auto	0.909
37	100	poisson		0.934
38	100	poisson	sqrt	0.833
39	100	poisson	Log2	0.781
40	100	poisson	auto	0.939
41	50	poisson		0.937
42	50	poisson	sqrt	0.695
43	50	poisson	Log2	0.757
44	50	poisson	auto	0.944
45	20	poisson		0.928
46	20	poisson	sqrt	0.825
47	20	poisson	Log2	0.764

48	20	poisson	auto	0.941
----	----	---------	------	-------

Here **R<sup>2</sup> value** is upto mark. 0.948

This model (100, squared\_error, auto) is **Good** for this data.