Regression Assignment

1. Problem Statement

Predict the Insurance Charges on the several parameters.

2. Dataset Information

Dataset contains 6 columns and 1340 rows of data values including header

3. Data Preprocessing

In the given dataset, there are two columns contains string values. Since Python AI models cannot directly handle categorical data, the "Smoker" and “Sex”columns values will be converted into meaningful numerical data.

Eg: after removed the first column

sex_male	smoker_yes
1	0
0	1
1	0

4. R&D on Model Selection

a. Multi-Linear Regression

The R-score value obtained in the Multi-Linear Regression model is **0.789**

b. Support Vector Machine (SVM)

Hyper Parameter	linear	rbf	poly	sigmoid
0	-0.11	-0.08	-0.06	-0.08
100	0.54	-0.12	-0.09	-0.11
500	0.62	-0.12	-0.08	-0.47
1000	0.63	-0.11	-0.05	-1.71
3000	0.75	-0.09	0.04	-12.54
5000	**0.76**	-0.07	0.14	-32.69

c. Decision Tree

Sl. No	Criterion	Max Features	Splitter	R Value
1	*squared_error*	None	*best*	0.65
2	*squared_error*	None	random	0.71
3	friedman_mse	None	*best*	0.69
4	friedman_mse	None	random	0.71
5	absolute_error	None	best	0.67

6	absolute_error	None	random	0.72
7	poisson	None	best	0.68
8	poisson	None	random	0.69
9	*squared_error*	sqrt	*best*	0.74
10	*squared_error*	sqrt	*random*	0.71
11	*squared_error*	Log2	best	0.70
12	*squared_error*	log2	random	0.61
13	friedman_mse	sqrt	*best*	0.67
14	friedman_mse	sqrt	random	0.54
15	friedman_mse	Log2	best	0.72
16	friedman_mse	log2	random	0.68
17	absolute_error	sqrt	best	0.64
18	absolute_error	sqrt	random	0.69
19	absolute_error	Log2	best	0.67
20	absolute_error	log2	random	0.66
21	poisson	sqrt	best	0.73
22	poisson	sqrt	random	0.71
23	poisson	Log2	best	0.65
24	poisson	Log2	random	0.62

d. Random Forest

Sl. No	Criterion	Max Features	n-estimators	R Value
1	*squared_error*	None	*100*	0.85
2	*squared_error*	*None*	*50*	0.85
3	*squared_error*	None	*80*	0.84
4	absolute_error	None	100	0.85
5	absolute_error	None	50	0.84
6	friedman_mse	None	50	0.84
7	poisson	None	50	0.85
8	poisson	None	80	0.85
9	*squared_error*	sqrt	*100*	0.86
10	*squared_error*	log2	100	0.86
11	friedman_mse	sqrt	*50*	0.86
12	friedman_mse	log2	80	0.86
13	absolute_error	sqrt	70	0.86
14	**absolute_error**	**log2**	**100**	**0.87**
15	poisson	sqrt	80	0.86
16	poisson	log2	50	0.86

In the above experiment, based on the R² score, the **Random Forest** (absolute, log2) model appears to be the best-performing model.