Assignment - Regression

Identify your problem statement

The objective is to develop a **machine learning** model that predicts insurance charges based on various input features such as age, sex, smoking status, number of children, and other relevant attributes. This is a **supervised learning regression problem**, and the project can be titled **"Insurance Charge Prediction."**

Tell basic info about the dataset (Total number of rows, columns)

The dataset consists of **1,338 rows** and **6 columns**. The input features include: **age, BMI, children, sex, and smoker**, while the target variable is **charges**, which represents the insurance cost.

Mention the pre-processing method if you're doing any (like converting string to number – nominal data)

The columns **'sex'** and **'smoker'** are categorical variables without any sequential order, so I applied a **nominal encoding** method. Specifically, I used **one-hot encoding** with the get_dummies() function to convert these categories into numerical format.

Hyperparameter tuning for the models

Multiple Linear Regression: (insurance_pre.csv - Dataset)

Using Multiple Linear Regression the model achieved an **R2 Score : 0.7894**

SVM: (insurance_pre.csv - Dataset)

SNO.	C Value	LINEAR (R value)	RBF (R value)	POLY (R value)	SIGMOID (R value)
1	10	0.4624	-0.0322	0.0387	0.0393
2	100	0.6288	0.3200	0.6179	0.5276
3	1000	0.7649	0.8102	0.8566	0.2874
4	2000	0.7440	0.8547	0.8605	-0.5939
5	3000	0.7414	0.8663	0.8598	-2.1244

Using Support Vector Machine (SVM) with parameters **C = 3000** and **kernel = 'rbf'**, the model achieved an **R^2 score of 0.8663**.

Decision Tree: (insurance_pre.csv - Dataset)

SNO	CRITERION	MAX_FEATURES	SPLITTER	R Value
1	squared_error	sqrt	*best*	0.6944
2	squared_error	sqrt	*random*	0.6209
3	squared_error	log2	*best*	0.6263
4	squared_error	log2	*random*	0.5291
5	squared_error	None	*best*	0.6972
6	squared_error	None	*random*	0.6543
7	*friedman_mse*	sqrt	*best*	0.7472
8	*friedman_mse*	sqrt	*random*	0.7441
9	*friedman_mse*	log2	*best*	0.7506
10	*friedman_mse*	log2	*random*	0.7358
11	*friedman_mse*	None	*best*	0.6891
12	*friedman_mse*	None	*random*	0.6932
13	*absolute_error*	sqrt	*best*	0.6668
14	*absolute_error*	sqrt	*random*	0.6429
15	*absolute_error*	log2	*best*	0.6703
16	*absolute_error*	log2	*random*	0.6909
17	*absolute_error*	None	*best*	0.6890
18	*absolute_error*	None	*random*	0.7571
19	*poisson*	sqrt	*best*	0.7281
20	*poisson*	sqrt	*random*	0.5762
21	*poisson*	log2	*best*	0.7398
22	*poisson*	log2	*random*	0.6667
23	*poisson*	None	*best*	0.7210
24	*poisson*	None	*random*	0.8152

Using Decision Tree model with parameters **criterion = 'poisson'**, **max_features = None**, and **splitter = 'random'**, achieved an **R^2 score of 0.8152**.

Random Forest: (insurance_pre.csv - Dataset)

SNO	CRITERION	MAX_FEATURES	n_estimators	R Value
1	squared_error	sqrt	10	0.8520
2	squared_error	sqrt	*50*	0.8695
3	squared_error	sqrt	*100*	0.8710
4	squared_error	log2	10	0.8520
5	squared_error	log2	*50*	0.8695
6	squared_error	log2	*100*	0.8710
7	squared_error	None	*10*	0.8330
8	squared_error	None	*50*	0.8498
9	squared_error	None	*100*	0.8538
10	*friedman_mse*	sqrt	10	0.8502
11	*friedman_mse*	sqrt	50	0.8702
12	*friedman_mse*	sqrt	*100*	0.8710

13	*friedman_mse*	log2	10	0.8502
14	*friedman_mse*	log2	50	0.8702
15	*friedman_mse*	log2	*100*	0.8710
16	*friedman_mse*	None	10	0.8331
17	*friedman_mse*	None	50	0.8500
18	*friedman_mse*	None	*100*	0.8540
19	*absolute_error*	sqrt	10	0.8574
20	*absolute_error*	sqrt	50	0.8708
21	*absolute_error*	sqrt	*100*	0.8710
22	*absolute_error*	log2	10	0.8574
23	*absolute_error*	log2	50	0.8708
24	*absolute_error*	log2	*100*	0.8710
25	*absolute_error*	None	10	0.8350
26	*absolute_error*	None	50	0.8526
27	*absolute_error*	None	*100*	0.8520
28	*poisson*	sqrt	10	0.8544
29	*poisson*	sqrt	50	0.8632
30	*poisson*	sqrt	*100*	0.8680
31	*poisson*	log2	10	0.8544
32	*poisson*	log2	50	0.8632
33	*poisson*	log2	*100*	0.8680
34	*poisson*	None	10	0.8313
35	*poisson*	None	50	0.8491
36	*poisson*	None	*100*	0.8526

The best model performance, with an R^2 score of 0.8710, was achieved using multiple parameter combinations. These include the criteria **'squared_error'**, **'friedman_mse'**, and **'absolute_error'**, each paired with **max_features** set to either **'sqrt'** or **'log2'**, and **n_estimators** fixed at **100**.

Conclusion:

Although multiple parameter combinations yielded the same highest **R^2 score of 0.8710**, the configuration with **criterion = 'squared_error'**, **max_features = 'sqrt'**, and **n_estimators = 100** is a commonly preferred choice due to its balance of performance and model simplicity.