Assignment-Regression Algorithm

1) Problem Statement 1:

Client requirement is very clear they wants to predict the insurance charge based on the input parameters. So this will comes under **Machine Learning** since this is number based output needs to be the call to action.

Problem Statement 2:

Client has provided the clear dataset, what is their requirement and also has clear input and output. So then it falls under **Supervised Learning.**

Problem Statement 3:

Dataset which customer provides is based on numbers and output expectations also to be in number. So this will falls under **Regression**.

2) Total number of rows: 1338

Total number of Columns: 6

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

- Since the dataset has the nominal data, so need to convert into ordinal data. So used **dataset=pd.get_dummies(dataset,dtype=int)**

4) Develop a good model with r2_score. You can use any machine learning algorithm; you can create many models. Finally, you have to come up with final model

- As per all the algorithms such as Simple Linear, Multiple Linear, Support Vector Machine, Decision Tree and Random forest has been used in this assignment.

5) All the research values (r2_score of the models) should be documented. (You can make tabulation or screenshot of the results.)

- Yes has been documented and added the results under the word format.

6) Mention your final model, justify why u have chosen the same.

- Out of all the algorithms have found the random forest provided the result which is nearing to 1 and the R2 value is **0.8366**

R2 Scores for different Alogorithms

1) **Simple Linear Regression:**

- R2 value of Simple Linear Regression is **0.7587**

2) **Multiple Linear Regression**:

- R2 value of Simple Linear Regression is **0.7587**

3) **Support Vector Machine**:

S No	Kernel	C	R_score
1	Linear	-	-0.1218
2	Linear	0.01	-0.0536
3	Linear	100	0.5255
4	Linear	1000	0.6169
5	Rbf	0.01	-0.0579

6	Rbf	100	-0.0794
7	Rbf	1000	-0.1224
8	Poly	1000	-0.0805
9	Sigmoid	1000	-0.2274

4) **Decision Tree**:

S No	**Criterion**	**Splitter**	**R_score**
1	Squared Error	Best	0.6802
2	Squared Error	Random	0.7074
3	friedman_mse	Best	0.6835
4	friedman_mse	Random	0.6703
5	absolute_error	Best	0.7144
6	absolute_error	Random	0.6668
7	poisson	Best	0.7037
8	poisson	Random	0.6460

5) **Random Forest**:

S No	**Criterion**	**N_estimators**	**R_score**
1	Squared Error	50	0.8308
2	Squared Error	100	0.8307
3	Absolute_error	50	0.8349
4	Absolute_error	5000	0.8366
5	friedman_mse	50	0.8306
6	friedman_mse	100	0.8312
7	poisson	50	0.8326
8	poisson	100	0.8324