Assignment-Regression Algorithm

1. **Identify your problem statement:**
   - Machine Learning
   - Supervised Learning
   - Regression
2. **Basic information:**
   a) Total Number of rows: 1338
   b) Total Number of column: 6
   c) Dependent_Variable(Output): Charges("Have to predict insurance charges")
   d) Independent_Varaible(Input): Age, bmi, children, sex_male, smoker_yes
3. **Preprocessing Method:**
   I. Convert categorical data columns **(sex, smoker)** into valuable numerical data using dummies function get_dummies(). So that, Sex and Smoker column converted into **(sex_male, sex_female, smoker_yes, smoker_no)** and that we drop the two column which gives the same information without changing the data info using drop_first parameter.

```
Dataset=pd.get_dummies(dataset, drop_first=True)
```

   II. In SVM regression algorithm, used standardization for better result. And also, transform() function used in this model.

```
X_train=sc.fit_tranform(X_train)
```

   III. In Decision tree and random forest algorithms, used hyper tuning parameters for better result.(Criterion, max_features)
4. **Research Values for all regression algorithm:**
   - ***Multiple Linear Regression(R2_Score)=0.7891%***
   - ***Support vector Machine Regression***

S.no	Hyper Parameter	Linear (r value)	RBF (r value)	Poly (r value)	Sigmoid (r value)
1	C100	0.6289	0.3196	0.6164	0.5268
2	C1000	0.7648	0.8107	0.8546	0.21204
3	C2000	0.7439	0.8547	0.8583	-0.6216
4	C5000	0.7413	0.8737	0.8587	-8.1606
5	C10000	0.7413	0.8736	0.8572	-28.341

The best R2_Score value in Support Vector Machine**("kernel='rbf', C=5000) is 0.8737%.**

- ***Decision Tree Regressor***

S.NO	CRITERION	MAX_FEATURES	SPLITTER	R VALUE
1	squared_error	sqrt	best	0.7275
2	squared_error	sqrt	random	0.6878

3	squared_error	log2	best	0.7202
4	squared_error	log2	random	0.6711
5	friedman_mse	sqrt	best	0.7334
6	friedman_mse	sqrt	random	0.6544
7	friedman_mse	log2	best	0.7101
8	friedman_mse	log2	random	0.6111
9	absolute_error	sqrt	best	0.6559
10	absolute_error	sqrt	random	0.7208
11	absolute_error	log2	best	0.7308
12	absolute_error	log2	random	0.7032
13	poisson	sqrt	best	0.7261
14	poisson	sqrt	random	0.6768
15	poisson	log2	best	0.6492
16	poisson	log2	random	0.6831

The best R2_Score in Decision Tree Regression Algorithm **("criterion='friedman_mse', max_features='sqrt', splitter='best'")is 0.7334%.**

- ***Random Forest Regression***

S.NO	CRITERION	MAX_FEATURES	R VALUE
1	squared_error	sqrt	0.8662
2	squared_error	log2	0.8662
3	friedman_mse	sqrt	0.8662
4	friedman_mse	log2	0.8662
5	absolute_error	sqrt	0.8835
6	absolute_error	log2	0.8709
7	poisson	sqrt	0.8659
8	poisson	log2	0.8659

The best R2_Score in Random forest regression algorithm **("criterion=friedman_mse, max_features=sqrt") is 0.8835%**

5. Conclusion:

For this model, I have Chosen the **Random forest Regression** as it gives the highest R2_score**(0.8835)** among all other types of regression using various hyper tuning parameters like n_estimators, criterion, max_features, min_samples_spli, random state.