Assignment-Regression Algorithm

1.Identify your problem statement

The problem statement in this scenario is to develop a predictive model for estimating insurance charges. This means creating a machine learning model that can take various parameters or features as input and provide an estimate of the insurance charges as output. The goal is to build a model that can accurately predict insurance costs based on the available data, which can be useful for pricing insurance policies and assisting clients in understanding their potential insurance costs.

Approach:

1) Machine Learning → Supervisor learning → Regression

Reason:

The output of the predict value is Numerical. So we go with **Regression.**

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

Total Number of columns: 6

Total Number of Rows: 1339

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

In this dataset, two categorical data such as **sex, smoker** and this data belongs to nominal. So we need to convert into numerical data by using **One Hot Encoding**.

1) Multiple Linear Regression:

R score value = 0.7894790349867009

2) Support Vector Machine:

S.No	Hyper Parameter	Linear **r value**	RBF (NON LINEAR) **r value**	Poly **r value**	Sigmod **r value**
1	C=10	0.462468414233968	-0.032273293906710	0.0387162227602314	0.039307143782743
2	C=100	0.62887928573203	0.3200317832050831	0.6179569624059795	0.5276103546510407
3	C=500	0.763105805389725	0.6642984645143138	0.8263683541269009	0.44460610338694795
4	C=1000	0.764931173859741	0.8102064851758545	0.8566487675946572	0.28747069486976173
5	C=2000	0.74404183081078	0.8547766425392979	0.8605579258597704	-0.5939509731283505
6	C=3000	0.74142365992498	0.8663393953081687	0.8598930084494356	-2.1244194786689854
7	C=4000	0.74141988030664	**0.8717407875653337**	0.860004958058775	-5.510333547108607

The Support Vectore Machine's best **R Score** is (**rbf, C4000**) = **0.8717407875653337**

3.Decision Tree

S.No	splitter	max_features	R Value
Criterion = squared_error			
1	random	default=None	0.7164575052352842
2	random	sqrt	0.7457866999136783
3	random	log2	0.5851950094276736
4	best	default=None	0.6791277757686889
5	best	sqrt	0.7334132321457576
6	best	log2	0.6617535574495297
Criterion = *friedman_mse*			
1	random	default=None	0.6839314568742743
2	random	sqrt	0.5955752061986902
3	random	log2	0.7157492778771246
4	best	default=None	0.6971552795429837
5	best	sqrt	0.6994782745863823
6	best	log2	0.7281316744082819
Criterion = absolute_error			
1	random	default=None	0.7280457454670881
2	random	sqrt	0.6760798086954077
3	random	log2	0.6363546442165247
4	best	default=None	0.6505928019763672
5	best	sqrt	0.7118539605732017
6	best	log2	0.6606929759262083
Criterion = absolute_error			
1	random	default=None	**0.7438042902083184**
2	random	sqrt	0.7021096521795435
3	random	log2	0.6269235432777364
4	best	default=None	0.6661619625746076
5	best	sqrt	0.649793847129976
6	best	log2	0.6893887240933031

The **Decision Tree** best R Score is **0.7438042902083184** (**Criterion = absolute_error, splitter=random, max_features=None**)

4.Random Forest:

S.No	max_features	n_estimators	R Value
Criterion = ***squared_error***			
1	1.0	50	0.8498329315421834
2	1.0	100	0.8538307913484513
3	sqrt	50	0.8695836787761578
4	sqrt	100	0.8710271903471005
5	log2	50	0.8695836787761578
6	log2	100	0.8710271903471005
Criterion = ***absolute_error***			
1	1.0	50	0.8526655993519747
2	1.0	100	0.8520093621081837
3	sqrt	50	0.8708144250343052
4	sqrt	100	0.8710685856341518
5	log2	50	0.8708144250343052
6	log2	100	**0.8710685856341518**
Criterion = ***friedman_mse***			
1	1.0	50	0.8500716139332296
2	1.0	100	0.8540518935149612
3	sqrt	50	0.8702417511198071
4	sqrt	100	0.8710544015500664
5	log2	50	0.8702417511198071
6	log2	100	0.8710544015500664
Criterion = ***poisson***			
1	1.0	50	0.8491075958392151
2	1.0	100	0.8526334258892607
3	sqrt	50	0.8632391369285537
4	sqrt	100	0.8680156984764337
5	log2	50	0.8632391369285537
6	log2	100	0.8680156984764337

The Random Forest best R Score is **0.8710685856341518** (**Criterion = absolute_error, n_estimators =100, max_features=log2**)

The Best method of Regression is :

1) The Support Vectore Machine's best R Score is **0.8717407875653337** (rbf, C4000)

(OR)

2) The Random Forest best R Score is **0.8710685856341518** (Criterion = absolute_error, n_estimators =100, max_features=log2)