Solution:

1. Here the dataset is provided and requirements are clear, with input and output in numbers, So, we choose Machine Learning, Supervised, Regression

2. Dataset contains 1338 rows and 5 columns

3. Converting Categorical columns to nominal data

4. Comparison of R score Value of different Machine Linear Regression algorithms for same dataset

Multiple Linear Regression- *r_score = 0.789*

Support Vector Machine

S.NO	Hyper Parameter "c" value	r_score /r_score standardized linear	r_score /r_score standardized rbf	r_score /r_score standardized poly	r_score /r_score standardized sigmoid
1	10	-0.001/ 0.462	-0.081/ -0.032	-0.093/ 0.038	-0.09/ 0.039
2	100	0.543/ 0.628	-0.124/ 0.32	-0.099/ 0.617	-0.118/ 0.527
3	1000	0.634/ 0.764	-0.117/ 0.810	-0.055/ 0.856	-1.66/ 0.287
4	2000	0.689/ 0.744	-0.107/ 0.854	-0.002/ 0.860	-5.616/ -0.593
5	5000	0.764/ 0.741	-0.073/ 0.874	0.146/ 0.859	-31.568/ -7.53

Best Model, Parameter= "linear" with c= 5000, r_score = 0.764

After Standardization

Best Model, Parameter= "rbf" with c= 5000, r_score = 0.874

Decision tree

S.no	Criterion	Splitter	Max_features	r_score
1	squared_error	best	None	0.689
2	squared_error	best	sqrt	0.698
3	squared_error	best	log2	0.678
4	squared_error	random	None	0.687
5	squared_error	random	sqrt	0.575

6	squared_error	random	log2	0.638
7	friedman_mse	best	None	0.695
8	friedman_mse	best	sqrt	0.718
9	friedman_mse	best	log2	0.679
10	friedman_mse	random	None	0.720
11	friedman_mse	random	sqrt	0.733
12	friedman_mse	random	log2	0.687
13	absolute_error	best	None	0.652
14	absolute_error	best	sqrt	0.684
15	absolute_error	best	log2	0.646
16	absolute_error	random	None	0.746
17	absolute_error	random	sqrt	0.737
18	absolute_error	random	log2	0.580
19	poisson	best	None	0.722
20	poisson	best	sqrt	0.676
21	poisson	best	log2	0.717
22	poisson	random	None	0.710
23	poisson	random	sqrt	0.653
24	poisson	random	log2	0.653

Best Model, Criterion= absolute_error, Splitter= random, *Max_features= None, r_score = 0.746*

Random Forest

S.no	Criterion	n_estimators	Max_features	r_score
1	squared_error	50	None	0.849
2	squared_error	50	sqrt	0.869
3	squared_error	50	log2	0.869
4	squared_error	100	None	0.853
5	squared_error	100	sqrt	0.87102
6	squared_error	100	log2	0.87102
7	friedman_mse	50	None	0.850
8	friedman_mse	50	sqrt	0.870
9	friedman_mse	50	log2	0.870
10	friedman_mse	100	None	0.854
11	friedman_mse	100	sqrt	0.87105
12	friedman_mse	100	log2	0.87105
13	absolute_error	50	None	0.852
14	absolute_error	50	sqrt	0.870
15	absolute_error	50	log2	0.870
16	absolute_error	100	None	0.852
17	absolute_error	100	sqrt	0.87106
18	absolute_error	100	log2	0.87106
19	poisson	50	None	0.849
20	poisson	50	sqrt	0.8630
21	poisson	50	log2	0.8632

22	poisson	100	None	0.8526
23	poisson	100	sqrt	0.868
24	poisson	100	log2	0.868

Best Model, Criterion= absolute_error, n_estimators=100, Max_features= sqrt/log2,
r_score = 0.8712

Final Model:

For the given input dataset, the best model for Insurance charges prediction is **Random Forest** with *r_score = 0.8712* using the following parameters, ***Criterion= absolute_error, n_estimators=100, Max_features= sqrt/log2, random_state=0.***