Problem Statement: Insurance charges Prediction

Read the dataset and Retrieve the columns present in the dataset:

	age	sex	bmi	children	smoker	charges
0	19	female	27.900	0	yes	16884.92400
1	18	male	33.770	1	no	1725.55230
2	28	male	33.000	3	no	4449.46200
3	33	male	22.705	0	no	21984.47061
4	32	male	28.880	0	no	3866.85520
...	...	...	...	...	...	...
1333	50	male	30.970	3	no	10600.54830
1334	18	female	31.920	0	no	2205.98080
1335	18	female	36.850	0	no	1629.83350
1336	21	female	25.800	0	no	2007.94500
1337	61	female	29.070	0	yes	29141.36030

1338 rows × 6 columns

Data set has 1338 rows ,and 6 columns

Note: By using shape attribute we can able to get the rows and columns count

Performed One hot Encoding:

In our dataset we found that the some columns values are in string .So we performed the One hot encoding using get_dummies function.

	age	bmi	children	charges	sex_female	sex_male	smoker_no	smoker_yes
0	19	27.900	0	16884.92400	1	0	0	1
1	18	33.770	1	1725.55230	0	1	1	0
2	28	33.000	3	4449.46200	0	1	1	0
3	33	22.705	0	21984.47061	0	1	1	0
4	32	28.880	0	3866.85520	0	1	1	0
...	...	...	...	...	...	...	...	...
1333	50	30.970	3	10600.54830	0	1	1	0
1334	18	31.920	0	2205.98080	1	0	1	0
1335	18	36.850	0	1629.83350	1	0	1	0
1336	21	25.800	0	2007.94500	1	0	1	0
1337	61	29.070	0	29141.36030	1	0	0	1

Machine Learning Algorithm:

Multiple Linear Regression:

R2_score value : 0.78947

SVR :

S.No	linear(r2_scrore)	poly(r2_scrore)	rbf(r2_scrore)	sigmoid(r2_scrore)	c
1	0.06034	-0.0623	-0.08188	-0.07204	
2	-0.08808	-0.08943	-0.08963	-0.08953	0.01
3	-0.07349	-0.08694	-0.08892	-0.08793	0.1
4	0.56651	0.15939	-0.0181	0.07305	10
5	0.63595	0.75081	0.3906	0.52756	100
6	0.74409	0.86058	0.82835	0.14377	1000
7	0.74142	0.86018	0.86073	-2.58403	2000
8	0.74142	0.86001	0.86853	-6.82618	3000
9	0.74142	0.8594	0.87073	-12.29827	4000
10	0.74142	0.85886	0.87357	-17.55418	5000
11	0.74142	0.85864	0.87506	-29.02913	6000
12	0.74142	0.8586	0.87609	-36.07464	7000
13	0.74142	0.85836	0.87698	-47.87925	8000
14	0.74142	0.8583	0.87736	-68.38431	9000

SVR (RBF with C=9000) has the R2_score value : 0.87736

Decision Tree

S.No	Criterion	splitter	max_feature	r2_score
1	squared_error	best	None	0.69272
2	squared_error	best	auto	0.70364
3	squared_error	best	sqrt	0.74346
4	squared_error	best	log2	0.67017
5	squared_error	random	None	0.67017
6	squared_error	random	auto	0.66551
7	squared_error	random	sqrt	0.72295
8	squared_error	random	log2	0.65844
9	friedman_mse	best	None	0.69651
10	friedman_mse	best	auto	0.68674
11	friedman_mse	best	sqrt	0.71677
12	friedman_mse	best	log2	0.65512
13	friedman_mse	random	None	0.69138
14	friedman_mse	random	auto	0.67748
15	friedman_mse	random	sqrt	0.74327
16	friedman_mse	random	log2	0.65428
17	absolute_error	best	None	0.67051
18	absolute_error	best	auto	0.65667
19	absolute_error	best	sqrt	0.74474
20	absolute_error	best	log2	0.71037
21	absolute_error	random	None	0.74344
22	absolute_error	random	auto	0.70013
23	absolute_error	random	sqrt	0.5748
24	absolute_error	random	log2	0.62252
25	poisson	best	None	0.67714
26	poisson	best	auto	0.66436
27	poisson	best	sqrt	0.62662
28	poisson	best	log2	0.60937
29	poisson	random	None	0.61354
30	poisson	random	auto	0.67081
31	poisson	random	sqrt	0.65407
32	poisson	random	log2	0.63868

Decision Tree (Criterion: Squared Error,Splitter:best,max_feature:sqrt) has the R2_Score :0.74474

Random Forest:

S.No	n_estimat	criterion	Random_S	max_featu	r2_score
1	50	squared_error	None	None	0.84783
2	50	squared_error	None	sqrt	0.86547
3	50	squared_error	None	log2	0.86704
4	50	absolute_error	None	None	0.85247
5	50	absolute_error	None	sqrt	0.86761
6	50	absolute_error	None	log2	0.86936
7	50	friedman_mse	None	None	0.8534
8	50	friedman_mse	None	sqrt	0.8706
9	50	friedman_mse	None	log2	0.87227
10	50	poisson	None	None	0.83646
11	50	poisson	None	sqrt	0.83222
12	50	poisson	None	log2	0.83078
13	100	squared_error	None	None	0.85529
14	100	squared_error	None	sqrt	0.86559
15	100	squared_error	None	log2	0.87052
16	100	absolute_error	None	None	0.85397
17	100	absolute_error	None	sqrt	0.87331
18	100	absolute_error	None	log2	0.86779
19	100	friedman_mse	None	None	0.85029
20	100	friedman_mse	None	sqrt	0.87041
21	100	friedman_mse	None	log2	0.8718
22	100	poisson	None	None	0.83601
23	100	poisson	None	sqrt	0.82625
24	100	poisson	None	log2	0.82703

The Random Forest Regression has the R2_Score value :0.87331

Conclusion:

comparison of Multiple Linear Regression, SVR, Decision Tree, Random Forest:

- The Multiple Linear Regression and Decision Tree models are not performed well. Because R2_score value is lesser than 80%.

- Even though Support vector Machine, Random Forest has the r2score value below 90%. I have tuned these models , but the r2_score is not reached 90% or above.

- The R2_Score value= 0.87 are same in both Support vector Machine, Random Forest. So that I am considering as these two models performed good. Based on this we can choose either SVM or Random Forest.