Problem Definition: To predict Insurance charges based on various input parameters given in the dataset.

1. **3 type of stage analysis:** Machine Learning, Supervised Learning, Regression (Categorical Nominal col (Sex and Smoker) expand to more than one col using one hot encoder).
2. Basic info about dataset

```
In [5]:
1  dataset = pd.get_dummies(dataset,drop_first=True)
2  dataset
```

Out[5]:

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

1338 rows × 6 columns

3. **Multiple Linear Regression**: Using MLR the predicted R Value is 0.78 (Both before and after Standardisation applied)

4. **Support Vector Machine**: Using SVM the predicted R Value is 0.70 (Kernel ='rbf', Gama= auto , C=100000).

				Non-Linear		
S#	**Hyper Tuning Param**	**Playable Param**	**Linear**	**RBF**	**Poly**	**Sigmoid**
1		Gama = Auto	-1.43	-0.08	-12266.20	-0.07
2	C=1	Gama = Auto	-1.43	-0.08	-12266.20	-0.07
3	C=10	Gama = Auto	-113.04	-0.08	-1163348.23	0.01
4	C=100	Gama = Auto	-146.14	-0.05	-32979013.98	-0.54
5	C=1000	Gama = Auto	154.87	0.00	-10543590.22	-4.54
6	C=10000	Gama = Auto	-156.76	0.37	-1066334.83	-759.86
7	C=100000	Gama = Auto	-156.76	0.70	-537189.39	-135200.91
8	C=1	Gama = Scale	-1.43	-0.08	-12266.20	-0.07
9	C=10	Gama = Scale	-113.04	-0.08	-1163348.23	0.01
10	C=100	Gama = Scale	-146.14	-0.02	-32979013.98	-0.54
11	C=1000	Gama = Scale	-154.87	-0.01	-10543590.22	-4.54

12	C=10000	Gama = Scale	-156.75	-0.11	-1066334.83	-759.86
13	C=100000	Gama = Scale	-156.75	-0.21	-537189.39	-135200.91

4. Decision Tree: Using DT the predicted R Value is 0.76 (Criterion = 'friedman_mse', Splitter = best, Max_features = log2).

SI#	criterion	splitter	max_features	R Value
1				0.70
2	squared_error			0.68
3	friedman_mse			0.68
4	absolute_error			0.68
5	poisson			0.65
6	squared_error	best	auto	0.67
7	absolute_error	Random		0.74
8	absolute_error	Random	log2	0.69
9	absolute_error	Random	auto	0.68
10	absolute_error	Random	sqrt	0.63
11	squared_error	best	sqrt	0.69
12	squared_error	best	log2	0.69
13	squared_error	best		0.69
14		best		0.69
15	friedman_mse	best	auto	0.69
16	friedman_mse	best	sqrt	0.73
17	friedman_mse	best	log2	0.76
18	squared_error	Random	log2	0.70
19	squared error	Random	sqrt	0.73
20			log2	0.63
21			sqrt	0.49
22			auto	0.69
23	absolute_error	best	log2	0.66
24	absolute_error	best	auto	0.69
25	absolute_error	best	sqrt	0.69
26	absolute_error	best		0.68
27	poisson	best	auto	0.66
28	poisson	best	sqrt	0.66
29	poisson	best	log2	0.57
30	poisson			0.65

5. Random Forest: Using RF the predicted R Value is 0.85 (with any hyper tuning & playable parameter)

SI#	estimators	Random_state	R Value
1			0.85
2	50		0.84

3	50	0	0.84
4	10	0	0.82
5	100	0	0.84

Conclusion

Models **R- Score values**

1. MLR = R Value is 0.78
2. SVM = R Value is 0.70
3. DT = R Value is 0.76
4. RF = R Value is 0.85

Since "Random Forest" gives higher R_score (0.85) than that of the other models, thus it gives lesser error. Hence, I will save and deploy this model to the production environment so as to allow my business to input age, Bmi, Children, Sex, Smoker and able to get output or predict insurance charges.