BOOSTING ALGORITHM ASSIGNMENT

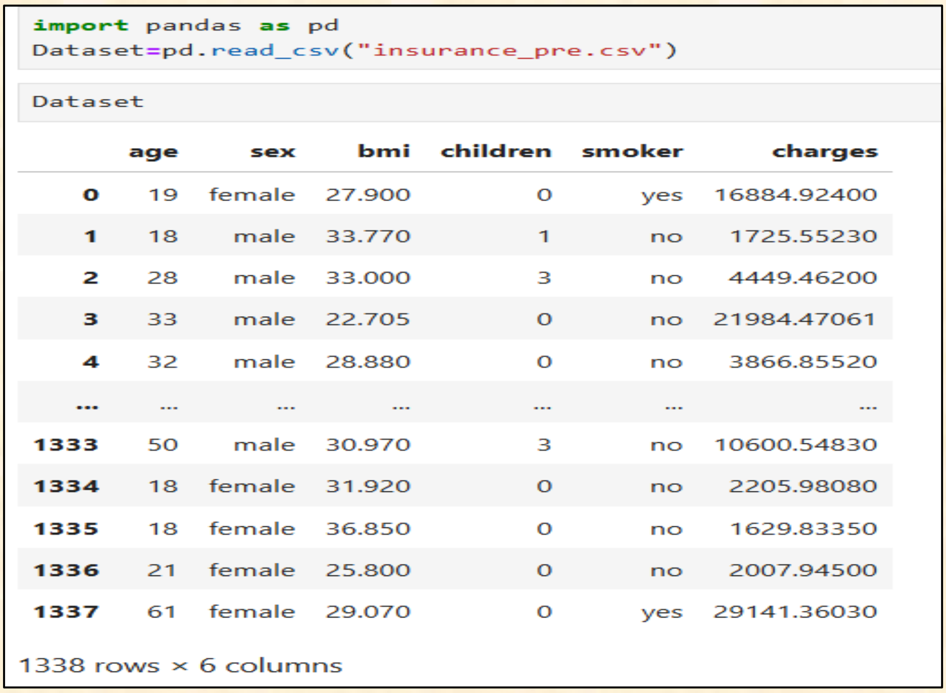
PROBLEM STATEMENT

As Data Scientists, we must develop a model to predict insurance charges when age, sex, BMI, children and smoker values are given as input parameters

DATASET INFORMATION

Total No of rows: 1338 (including column name)

Total No of columns: 6



```python
import pandas as pd
Dataset=pd.read_csv("insurance_pre.csv")
```

```python
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

Column Name: age, sex, bmi, children, smoker, charges

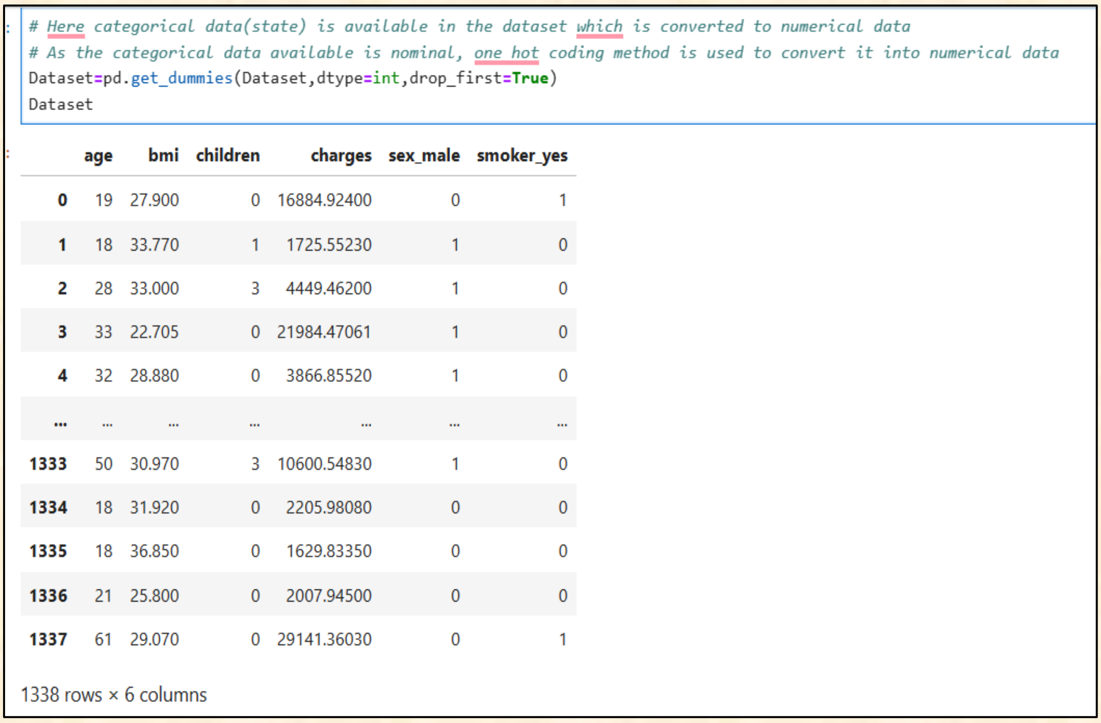
Input Variables: age, sex, bmi, children, smoker

Output Variable: Charges

PREPROCESSING METHOD:

Since the Dataset has categorical columns sex and smoker, it should be converted to numbers

Here the categorical data is nominal, so one hot encoding method is used to convert the categorical data (string) to numerical data (numbers 1 or 0)



```
# Here categorical data(state) is available in the dataset which is converted to numerical data
# As the categorical data available is nominal, one hot coding method is used to convert it into numerical data
Dataset=pd.get_dummies(Dataset,dtype=int,drop_first=True)
Dataset
```

	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

1. ADABOOST REGRESSOR:

S.no	n_estimators	learning_rate	loss	random_state	R2 score
1	50	1.0	linear	0	0.8447
2	20	1.0	linear	None	0.8704
3	50	2.0	linear	None	0.8753
4	20	2.0	linear	0	0.8809
5	50	1.0	square	0	0.5185
6	20	1.0	square	None	0.5837
7	50	2.0	square	None	0.5516
8	20	2.0	square	0	0.5083
9	50	1.0	exponential	0	0.6292
10	20	1.0	exponential	None	0.7472
11	50	2.0	exponential	None	0.5350
12	20	2.0	exponential	0	0.6522

Adaboost Regressor has a highest *R^2value as 0.8809* for n_estimators=20, learning_rate=2.0, loss=’linear’ and random_state=0

2. XGBOOST REGRESSOR:

S.no	eta	Max_depth	Subsample	Colsample_bytree	Tree_method	R Score
1	0.05	3	0.7	0.7	exact	0.8932
2	0.07	3	0.9	0.9	Exact	0.8896
3	0.1	3	0.9	0.9	Exact	0.8869
4	0.05	3	0.7	0.7	auto	0.8907
5	0.07	3	0.9	0.9	auto	0.8912
6	0.1	3	0.9	0.9	auto	0.8871
7	0.05	3	0.7	0.7	hist	0.8787
8	0.07	3	0.9	0.9	hist	0.8882
9	0.1	3	0.9	0.9	hist	0.8859
10	0.05	3	0.7	0.7	approx	0.8815
11	0.07	3	0.9	0.9	approx	0.8945
12	0.1	3	0.9	0.9	approx	0.8914

XGboost Regressor has a highest R^2*value as 0.8945* for eta=0.07,max_depth=3,subsample=0.9,colsample_bytree=0.9,tree_method='approx'

3. LGBM REGRESSOR:

S.NO	Boosting	n_estimators	num_ leaves	max_ depth	early_ stopping_ round	baggin_ freq	bagged_ Fraction	metric	R score
1	gbdt	100	10	3	50	1	0.9	rmse	0.8934
2	gbdt	200	40	5	100	2	0.8	L1	0.8883
3	gbdt	300	70	6	150	1	0.7	L2	0.8863
4	gbdt	400	200	8	200	2	0.6	rmse	0.8839
5	gbdt	500	900	10	250	1	0.5	L1	0.8872
6	dart	100	10	3	NA	1	0.9	rmse	0.8787
7	dart	200	40	5	NA	2	0.8	L1	0.8846
8	dart	300	70	6	NA	1	0.7	L2	0.8810
9	dart	400	200	8	NA	2	0.6	rmse	0.8763
10	dart	500	900	10	NA	1	0.5	L2	0.8772
11	rf	100	10	3	50	1	0.9	rmse	0.8818
12	rf	200	40	5	100	2	0.8	L1	0.8861
13	rf	300	70	6	150	1	0.7	L2	0.8895
14	rf	400	200	8	200	2	0.6	rmse	0.8858
15	rf	500	900	10	250	1	0.5	L2	0.8866

LGBM Regressor has highest R^2*value as 0.8934* for boosting='gbdt', n_estimators=100, num_leaves=10, max_depth=3, early_stopping_rounds=50, bagging_freq=1, bagged_fraction=0.9 and metric='rmse'