Regression Assignment

1.) Problem statement

Stage – 1 :

- Client requirement is to predict Insurance Charges which involves numerical calculations
- also the given input and output datasets also contains numerical values,

*so I prefer for **Machine Learning domain***

Stage – 2 :

Dataset (insurance_pre.csv) have clarity in input datasets and output dataset,

*so its **Supervised Learning***

Stage – 3 :

- Input and output values are Numerical,
- also it has clarity in input datasets and output datasets in a Linear way

*so its best to go for **Regression Analysis***

2.) Information about the dataset (Total number of rows, columns)

Dataset insurance_pre.csv

- 5 columns named as age, sex, bmi, children, smoker are train & test set input values,

- 1- column named as charges are output values to predict and calculate Insurance charges from the given Input values.

- Totally dataset has 1338-rows and 6-columns where,

5-columns are input values and 1 column were output values.

3.) Pre-processing method of converting string to number – nominal data

Dataset insurance_pre.csv has toltal of 6 columns where,

*- 5- columns named as **age, sex, bmi, children, smoker**, are input values for train and test set values*

*- 1- column named as **charges** are output values.*

- *In the given dataset column named as **sex** and **smoker**, has string(text) values which needs to convert as integer(number) – nominal data like*

  ***sex** column can be split as sex_female, sex_male,*

  ***smoker** column can be split as smoker_no, smoker_yes*

4.) Github Model Creation links and R2-score values .

a. Multiple Linear Regression Model creation – Github Link

https://github.com/ksenthilvelan/Regression_Assignment/blob/main/1.1_MultipleLinearRegresssion_Model_Rscore.ipynb

R2 Score value using Multiple Linear Regression is ***0.7895***

b. Support Vector Machine Model creation – Github Link

https://github.com/ksenthilvelan/Regression_Assignment/blob/main/2.1_SupportVectorMachine_Model_Rscore.ipynb

R2 Score value using Support Vector Machine (SVM) is ***0.87747***

c. Decision Tree Model creation – Github Link

https://github.com/ksenthilvelan/Regression_Assignment/blob/main/3.1_DecisionTree_Model_Rscore.ipynb

R2 Score value using Decision Tree is ***0.7588***

d. Random Forest Model creation – Github Link
https://github.com/ksenthilvelan/Regression_Assignment/blob/main/4.1_RandomForest_Model_Rscore.ipynb

R2 Score value using Random Forest is ***0.8706***

Comparing R2_Score and also Insurance charges –
Support Vector Machine model attains **R2-score** *is* **0.8775**, **insurance charges** *is* **18082.05**
Random Forest attains **R2-score** *is* **0.8706**, **insurance charges** *is* **5146.25**

So Random Forest is the best Final Model.

Random Forest Deployment – Github Link

https://github.com/ksenthilvelan/Regression_Assignment/blob/main/4.2_RandomForest_Deployment.ipynb

5.) All the R2-score of the models with tabulation of the results.

1. Mulitple Linear Regression

from sklearn.linear_model import LinearRegression

Sl.No	copy_X	fit_intercept	R2 Score
1	TRUE	TRUE	**0.7895**
2	FALSE	FALSE	0.7895

R2 Score value using Multiple Linear Regression is ***0.7895***

2. Support Vector Machine (SVM)

Epsilon Support Vector Regression - SVR

from sklearn.svm import SVR

Sl.No	C (Regularisation parameter)	R2 Score			
		kernel is ' rbf '	kernel is ' linear '	kernel is ' poly '	kernel is ' sigmoid '
1	1	-0.08188	0.06034	-0.06230	-0.07204
2	10	-0.01811	0.56651	0.15939	0.07305
3	100	0.39060	0.63595	0.75081	0.52756
4	1000	0.82835	0.74409	0.86058	0.14377
5	10000	**0.87747**	0.74142	0.85821	-82.19023

Note *- kernel value given as ' precomputed ' & ' callable ' parameters not supporting*

R2 Score value using Support Vector Machine (SVM) is ***0.87747***

3. Decision Tree

DecisionTreeRegressor

from sklearn.tree import DecisionTreeRegressor

Sl.No	criterion	splitter	max_features	R2 Score
1	squared_error *also known as mse - mean squared error*	best		0.6971
2		random		0.6715
3		best	sqrt	0.7090
4		random	sqrt	0.6589
5		best	log2	0.7120
6		random	log2	**0.7588**
7	friedman_mse *also known as mean squared error with Friedman's*	best		0.6983
8		random		0.7207
9		best	sqrt	0.6487
10		random	sqrt	0.6707
11		best	log2	0.6890
12		random	log2	0.7136
13	absolute_error *also known as mae - mean absolute error*	best		0.6711
14		random		0.7376
15		best	sqrt	0.6966
16		random	sqrt	0.6884
17		best	log2	0.7352
18		random	log2	0.6788

R2 Score value using Decision Tree is ***0.7588***

4. Random Forest

DecisionTreeRegressor

from sklearn.tree import DecisionTreeRegressor

Sl.No	criterion	max_features	max_depth	n_estimators	random_state	R2 Score
1	squared_error *also known as mse - mean squared error*	1.0	None	50	0	0.8496
2		sqrt	None	50	0	0.8704
3		log2	None	50	0	0.8704
4			None	50	0	0.8496
7	friedman_mse *also known as mean squared error with Friedman's*		None	50	0	0.8501
8		1.0	None	50	0	0.8501
9		sqrt	None	50	0	0.8703
10		log2	None	50	0	0.8703
13	absolute_error *also known as mae - mean absolute error*		None	50	0	0.8522
14		1.0	None	50	0	0.8522
15		sqrt	None	50	0	0.8706
16		log2	None	50	0	0.8706

R2 Score value using Random Forest is ***0.8706***

6.) Final Conclusion

Comparing R2_Score and also Insurance charges –

Multiple Linear Regression Model : **R2-score** *is* **0.7895**, **insurance charges** *is* **3056.35**
Support Vector Machine Model : **R2-score** *is* **0.8775**, **insurance charges** *is* **18082.05**
Decision Tree Model : **R2-score** *is* **0.7588**, **insurance charges** *is* **3213.60**
Random Forest Model : **R2-score** *is* **0.8706**, **insurance charges** *is* **5146.25**

So Random Forest is the best Final Model.