Assignment-Regression Algorithm

Problem Statement or Requirement:

A client’s requirement is, he wants to predict the insurance charges based on the several parameters. The Client has provided the dataset of the same.

1. **Problem Statement Identification:**
   Develop a model that predicts the insurance charges.
2. **Dataset Information:**
   The dataset has a total number of 6 columns- Age, Gender, BMI, Children, Smoker and Charges. Out of these, 5 columns are taken as inputs (Age, Gender, BMI, Children, Smoker) and one as output (Charges).
3. **Pre-processing:**
   Two columns (Gender, and Smoker) from the provided dataset has the values stored in string type. Hence to further process these data, the string values should be converted into integer value.

Domain Selection: Machine Learning

Learning Method: Supervised – Regression

Algorithm-1: Multiple Linear Regression:

The performance of Multiple Linear Regression Algorithm for the given dataset is evaluated using R2 score. This algorithm predicts the output with 78% accuracy.

R2 Score=0.7891

Algorithm-2: Support Vector Machine:

On hyper tuning the parameters with different combinations, the end results are given in the following table. Default parameter values are: *kernel=’rbf’, C=1.0*

C	Linear	Polynomial	RBF	Sigmoid
0.0001	-	-	-0.0897	-0.0897
1000	0.6898	-0.0428	-0.1089	-1.14
2000	0.765	0.0233	-0.09	-3.78
3000	0.7648	0.0859	-0.0692	-7.777

4000	0.7644	0.1462	-0.0488	-13.377
5000	0.744	0.2028	-0.0288	-20.406
100000	-	0.7834	0.5926	-
200000	-	0.8201	0.661	-
300000	-	0.8329	0.7036	-

The best R2_Score is: ***(Polynomial, C=300000) = 0.8329***

Algorithm-3: Decision Tree Regression:

On hyper tuning the parameters with different combinations, the end results are given in the following table. Default parameter values are: *criterion='squared_error', splitter='best'*

Splitter	Squared error	Friedman_mse	Absolute error	Poisson
Best	0.6632	0.6576	0.6788	0.6620
Random	0.7394	0.7292	0.7558	0.6833

The best R2_Score is: ***(Absolute error, Random) = 0.7558***

Algorithm-4: Random Forest Regression

On hyper tuning the parameters with different combinations, the end results are given in the following table. Default parameter values are: *n_estimators='100', criterion='squared_error'*

N_Estimators	Squared Error	Absolute Error	Friedman_mse	Poisson
10	0.8443	0.8427	0.8337	0.8373
100	0.8529	0.8608	0.8489	0.8531
200	0.8533	0.8554	0.8536	0.8531
300	0.8522	0.8561	0.8532	0.8532
400	0.8523	0.8565	0.8537	0.8525

500	0.8527	0.8574	0.8523	0.8534
1000	0.8531	0.8564	0.8527	0.8525

The best R2_Score is: ***(Absolute error, n_estimators= 100) = 0.8608***

Comparing the results from all the 4 algorithms, Support Vector Machine and Random Forest Regression provides better results.

Support Vector Machine:

On hyper tuning the kernel value to ‘poly’ and C=300000, we get R2_score of 0.8329. On further increasing the value of C=10000000, the R2_score is 0.8597. But the drawback of hyper tuning this c value is, the execution time delays up to 40 sec.

Random Forest Regression:

On hyper tuning the criterion value to ‘absolute_error’ and n_estimators=100, we get the R2_score of 0.8608. Increasing the n_estimators value doesn’t positively affect the score. The drawback of this algorithm is, each time we run the model, the R2_score value ranges from 0.85 to 0.86. That is, The R2_score is not stable.

Conclusion:

The model that fits best for this project is **Support Vector Regressor**. The accuracy of the Random forest varies each time we run the model, as the result the output value differs, which makes the model inconsistent. On Hyper tuning the C value of SVR further to 1000000, the **R2_score increases to 0.8513** with the run time 2.5 sec. This SVR model would provide same results as Random forest, but works with consistency.