**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.

As a data scientist, you must develop a model which will predict the insurance charges.

1.) Identify your problem statement

2.) Tell basic info about the dataset (Total number of rows, columns)

3.) Mention the pre-processing method if you are doing any (like converting string to number – nominal data)

4.) Develop a good model with r2\_score. You can use any machine learning algorithm; you can create many models. Finally, you must come up with final model.

5.) All the research values (r2\_score of the models) should be documented. (You can make tabulation or screenshot of the results.)

6.) Mention your final model, justify why u have chosen the same.

**Data modelling notes**:

1.) Problem Statement: Using client data sheet columns (age, sex, bmi, children, smoker) we need to find the charges.

Data sheet columns:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| age | sex | bmi | children | smoker | charges |

2.) The client data sheet has 1339 row values including header and 6 column values.

3.) Pre-processing methods we are using here are:

4.) Develop a good model with r2\_score. You can use any machine learning algorithm; you can create many models. Finally, you must come up with final model.

Final model is: RandomForestRegressor(criterion='poisson', n\_estimators=50

6.) Mention your final model, justify why u have chosen the same.:

When compared to the r\_score value for all the models, got high accuracy value with the RandomForestRegressor model.

**Multiple Linear Algorithm** (*r\_score* value = 0.789479)

**Support Vector Machine:**

With by default SVR(), the r\_score = -0.088427

With SVR(C=10), the r\_score = -0.081969

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| S.No | HYPER PARAMETER | LINEAR  (r value) | RBF(NON-LINEAR)  (r value) | POLY  (r value) | SIGMOID  (r value) |
| 1 | C10 | -0.001617 | -0.081969 | -0.09311 | -0.09078 |
| 2 | C100 | 0.543281 | -0.124803 | -0.09976 | -0.11814 |
| 3 | C500 | 0.627046 | -0.124641 | -0.08202 | -0.45629 |
| 4 | C1000 | 0.634036 | -0.117490 | -0.05550 | -1.66590 |
| 5 | C2000 | 0.689326 | -0.107787 | -0.00270 | -5.61643 |
| 6 | C3000 | **0.759089** | -0.096212 | 0.04892 | -12.01904 |

The SVM Regression use R2 value (linear and hyper parameter (C3000) = 0.759089

**Decision Tree**:

With direct DecisionTreeRegressor(), got r\_score = 0.68448

|  |  |  |  |
| --- | --- | --- | --- |
| S.No | Criterion value | Splitter value | R value |
| 1 | ***squared\_error*** | Best | 0.68854 |
| 2 | ***squared\_error*** | random | 0.69230 |
| 3 | ***friedman\_mse*** | Best | 0.70240 |
| 4 | ***friedman\_mse*** | random | 0.73995 |
| 5 | ***absolute\_error*** | Best | 0.65999 |
| 6 | ***absolute\_error*** | random | 0.73604 |
| 7 | ***poisson*** | Best | 0.73506 |
| 8 | ***poisson*** | random | 0.72053 |

Using DecisionTreeRegressor, with Criterion = ***friedman\_mse***, splitter = random we got high accuracy as 0.73995

**Random Forest**:

With direct RandomForestRegressor(), got r\_score = 0.85000

|  |  |  |  |
| --- | --- | --- | --- |
| S.No | Criterion value | n\_estimators | R value |
| 1 | ***squared\_error*** | 100 | 0.85562 |
| 2 | ***squared\_error*** | 50 | 0.84821 |
| 3 | ***friedman\_mse*** | 100 | 0.85207 |
| 4 | ***friedman\_mse*** | 50 | 0.85034 |
| 5 | ***absolute\_error*** | 100 | 0.85424 |
| 6 | ***absolute\_error*** | 50 | 0.85597 |
| 7 | ***poisson*** | 100 | 0.85403 |
| 8 | ***poisson*** | 50 | 0.85665 |

Using RandomForestRegressor, with Criterion = ***poisson***, n\_estimators = 50 we got high accuracy as 0.94635

**\*\*\* Thank you, mam. \*\*\***