**S.RAYEN SUBHIKSHA**

**ASSIGNMENT-REGRESSION ALGORITHM**

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. Identify your problem statement

We need to predict the insurance charges from the given datas as inputs.

3 stages:

Stage-1: Machine Learning

Stage-2: Supervised learning

Stage-3: Regression

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

For output:

Column Name: charges

No. of rows: 1338

For inputs:

Columns Names: age, sex, bmi, children & smoker

No. of columns: 5

No. of rows: 1338

Total no. of rows: 1338

Total no. of columns:6

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

From the given data, smoker column is given as categorical data.

So, we need to convert categorical data to numerical data

The data is in Ordinal data—we can convert by using the technique Label Encoder

1. Develop a good model with r2\_score. You can use any machine learning algorithm; you can create many models. Finally, you have to come up with final model.
2. All the research values (r2\_score of the models) should be documented.

**SVM**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| HYPER TUNING PARAMETER  C VALUE | LINEAR  (r\_score) | RBF  (r\_score) | POLY  (r\_score) | SIGMOID  (r\_score) |
| 0.01 | -0.08 | -0.089 | -0.089 | -0.089 |
| 0.1 | -0.08 | -0.089 | -0.088 | -0.088 |
| 10 | 0.462 | -0.032 | 0.036 | 0.039 |
| 100 | 0.628 | 0.319 | 0.616 | 0.526 |
| 1000 | 0.764 | 0.810 | 0.854 | 0.212 |
| 2000 | 0.743 | 0.854 | 0.858 | -0.621 |
| 2500 | 0.741 | 0.859 | 0.858 | -1.266 |
| 3000 | 0.741 | 0.864 | 0.858 | -2.141 |
| 3500 | 0.741 | 0.868 | 0.858 | -3.198 |
| 4000 | 0.741 | 0.870 | 0.858 | -5.466 |
| 4500 | 0.741 | 0.872 | 0.858 | -6.437 |
| 5000 | 0.741 | 0.873 | 0.858 | -8.160 |
| 5500 | 0.741 | 0.875 | 0.858 | -10.397 |
| 6000 | 0.741 | 0.876 | 0.858 | -11.374 |
| 6500 | 0.741 | 0.876 | 0.857 | -13.355 |
| 7000 | 0.741 | 0.876 | 0.857 | -17.171 |
| 7500 | 0.741 | 0.877 | 0.857 | -19.987 |
| 8000 | 0.741 | 0.877 | 0.857 | -19.005 |
| 8500 | 0.741 | 0.877 | 0.857 | -24.925 |
| 9000 | 0.741 | 0.877 | 0.857 | -27.962 |
| 10000 | 0.741 | 0.878 | 0.857 | -28.341 |
| 20000 | 0.741 | 0.878 | 0.856 | -135.875 |
| 40000 | 0.741 | 0.877 | 0.856 | -482.833 |
| 50000 | 0.741 | 0.876 | 0.856 | -697.111 |
| 60000 | 0.741 | 0.874 | 0.856 | -1052.633 |
| 70000 | 0.741 | 0.873 | 0.856 | -1599.714 |
| 80000 | 0.741 | 0.873 | 0.856 | -1911.953 |
| 90000 | 0.741 | 0.873 | 0.856 | -2808.024 |
| 100000 | 0.741 | 0.873 | 0.856 | -2818.199 |

**DECISION TREE**

|  |  |  |  |
| --- | --- | --- | --- |
| CRITERION | SPLITTER | MAX\_FEATURES | R\_SCORE |
| squared\_error | best | sqrt | 0.743 |
| squared\_error | random | sqrt | 0.728 |
| friedman\_mse | best | sqrt | 0.652 |
| friedman\_mse | random | sqrt | 0.620 |
| poisson | best | sqrt | 0.716 |
| poisson | random | sqrt | 0.684 |
| squared\_error | best | log2 | 0.722 |
| squared\_error | random | log2 | 0.660 |
| friedman\_mse | best | log2 | 0.727 |
| friedman\_mse | random | log2 | 0.670 |
| poisson | best | log2 | 0.690 |
| poisson | random | log2 | 0.761 |

**RANDOM FOREST**

|  |  |  |  |
| --- | --- | --- | --- |
| CRITERION | N\_ESTIMATORS | MAX\_FEATURES | R\_score |
| squared\_error | 10 | sqrt | 0.845 |
| squared\_error | 20 | sqrt | 0.860 |
| squared\_error | 30 | sqrt | 0.863 |
| squared\_error | 40 | sqrt | 0.860 |
| squared\_error | 50 | sqrt | 0.864 |
| squared\_error | 60 | sqrt | 0.861 |
| squared\_error | 70 | sqrt | 0.860 |
| squared\_error | 80 | sqrt | 0.868 |
| squared\_error | 90 | sqrt | 0.864 |
| squared\_error | 100 | sqrt | 0.866 |
| squared\_error | 10 | log2 | 0.854 |
| squared\_error | 20 | log2 | 0.862 |
| squared\_error | 30 | log2 | 0.864 |
| squared\_error | 40 | log2 | 0.864 |
| squared\_error | 50 | log2 | 0.862 |
| squared\_error | 60 | log2 | 0.862 |
| squared\_error | 70 | log2 | 0.863 |
| squared\_error | 80 | log2 | 0.863 |
| squared\_error | 90 | log2 | 0.864 |
| squared\_error | 100 | log2 | 0.863 |
| absolute\_error | 10 | sqrt | 0.853 |
| absolute\_error | 20 | sqrt | 0.861 |
| absolute\_error | 30 | sqrt | 0.865 |
| absolute\_error | 40 | sqrt | 0.867 |
| absolute\_error | 50 | sqrt | 0.867 |
| absolute\_error | 60 | sqrt | 0.870 |
| absolute\_error | 70 | sqrt | 0.865 |
| absolute\_error | 80 | sqrt | 0.872 |
| absolute\_error | 90 | sqrt | 0.869 |
| absolute\_error | 100 | sqrt | 0.869 |
| absolute\_error | 10 | log2 | 0.862 |
| absolute\_error | 20 | log2 | 0.849 |
| absolute\_error | 30 | log2 | 0.867 |
| absolute\_error | 40 | log2 | 0.872 |
| absolute\_error | 50 | log2 | 0.867 |
| absolute\_error | 60 | log2 | 0.868 |
| absolute\_error | 70 | log2 | 0.865 |
| absolute\_error | 80 | log2 | 0.869 |
| absolute\_error | 90 | log2 | 0.868 |
| absolute\_error | 100 | log2 | 0.869 |
| friedman\_mse | 10 | sqrt | 0.848 |
| friedman\_mse | 20 | sqrt | 0.857 |
| friedman\_mse | 30 | sqrt | 0.862 |
| friedman\_mse | 40 | sqrt | 0.858 |
| friedman\_mse | 50 | sqrt | 0.860 |
| friedman\_mse | 60 | sqrt | 0.866 |
| friedman\_mse | 70 | sqrt | 0.864 |
| friedman\_mse | 80 | sqrt | 0.863 |
| friedman\_mse | 90 | sqrt | 0.863 |
| friedman\_mse | 100 | sqrt | 0.862 |
| friedman\_mse | 10 | log2 | 0.840 |
| friedman\_mse | 20 | log2 | 0.850 |
| friedman\_mse | 30 | log2 | 0.857 |
| friedman\_mse | 40 | log2 | 0.860 |
| friedman\_mse | 50 | log2 | 0.860 |
| friedman\_mse | 60 | log2 | 0.861 |
| friedman\_mse | 70 | log2 | 0.861 |
| friedman\_mse | 80 | log2 | 0.865 |
| friedman\_mse | 90 | log2 | 0.867 |
| friedman\_mse | 100 | log2 | 0.866 |
| poisson | 10 | sqrt | 0.846 |
| poisson | 20 | sqrt | 0.856 |
| poisson | 30 | sqrt | 0.859 |
| poisson | 40 | sqrt | 0.861 |
| poisson | 50 | sqrt | 0.868 |
| poisson | 60 | sqrt | 0.867 |
| poisson | 70 | sqrt | 0.864 |
| poisson | 80 | sqrt | 0.861 |
| poisson | 90 | sqrt | 0.863 |
| poisson | 100 | sqrt | 0.861 |
| poisson | 10 | log2 | 0.836 |
| poisson | 20 | log2 | 0.859 |
| poisson | 30 | log2 | 0.857 |
| poisson | 40 | log2 | 0.860 |
| poisson | 50 | log2 | 0.864 |
| poisson | 60 | log2 | 0.863 |
| poisson | 70 | log2 | 0.861 |
| poisson | 80 | log2 | 0.863 |
| poisson | 90 | log2 | 0.860 |
| poisson | 100 | log2 | 0.861 |

For Multiple Linear Regression, r\_score is 0.789

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

I have chosen SVM algorithm for this problem statement to predict the insurance charges. Because I have made different combinations contains hyper tuning parameters like C value, kernel=”linear”,”rbh”,”poly”,”sigumoid”.

From this tabulations, I have chosen C=20000 and kernel=”rbf” which gives r value as 0.878 which is the highest among other model’s r value.

This will provide good performance and quality of the model.