**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’re 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 have to 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.

**1.Identify your problem statement**

Machine learning –for this problem I am choosing machine learning. Because is Insurance charge prediction. It is number based result.

**Supervised learning**

1. we have both input and output.

2. Requirements are clear

**Regression**

This model is continuous prediction.

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

1338- Total number of rows

8 - columns

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

**One Hot encoding method**

Converting categorial data into numerical data using get\_dummies

**4.) 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.**

**Regression:**

Multi regression R2\_score value is 0.710092

**Decision tree**

|  |  |  |  |
| --- | --- | --- | --- |
| **CRITERION** | **SPLITTER** | **MAX\_FEATURES** | **R2** |
| Squared\_error | best | sqrt | **0.71037** |
| Squared\_error | best | Log2 | **0.73169** |
| Squared\_error | best | none | **0.74331** |
| Squared\_error | random | sqrt | **0.69161** |
| Squared\_error | random | Log2 | **0.66053** |
| Squared\_error | random | none | **0.70806** |
| Friedman\_mse | best | sqrt | **0.76690** |
| Friedman\_mse | best | Log2 | **0.66909** |
| Friedman\_mse | best | none | **0.7368** |
| Friedman\_mse | random | sqrt | **0.72863** |
| Friedman\_mse | random | Log2 | **0.6929** |
| Friedman\_mse | random | none | **0.72821** |
| absolute\_error | best | sqrt | **0.74262** |
| absolute\_error | best | Log2 | **0.70580** |
| absolute\_error | best | none | **0.69486** |
| absolute\_error | random | sqrt | **0.76065** |
| Absolute\_error | random | Log2 | **0.76359** |
| Absolute\_error | random | none | **0.7100** |
| Poission | best | sqrt | **0.79869** |
| Poission | best | Log2 | **0.72112** |
| Poission | best | none | **0.763549** |
| poission | random | sqrt | **0.68285** |
| poission | random | Log2 | **0.68080** |
| poission | random | none | **0.72698** |
|  |  |  |  |

**SUPPORT VECTOR MACHINE**

SVM accuracy value for Hyperparameter,Linear,rbf,Poly,Sigmoid

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Hyper Parameter | Linear | (Non – Linear) RBF | Poly | Sigmoid |
| C10 | -7.1613 | -412.124 | -19.3429 | -48891.4 |
| C100 | -0.2253 | -15.0466 | -12.7505 | -466.76 |
| C500 | 0.04073 | -11.8860 | -11.6839 | -15.4132 |
| C1000 | 0.34098 | -11.4903 | -10.5717 | -3.1464 |
| C1500 | 0.66161 | -11.155 | -9.6539 | -1.5095 |
| C2000 | 0.73598 | -10.8952 | -8.7319 | -0.9582 |
| C3000 | 0.73758 | -10.4738 | -7.0677 | -0.5291 |

Random Forest

|  |  |  |  |
| --- | --- | --- | --- |
| Max\_Feature | n\_estimators | criterion | R2 score |
| squared\_error | 50 | log2 | 0.8375 |
| squared\_error | 100 | log2 | 0.8529 |
| squared\_error | 50 | Sqrt | 0.85138 |
| squared\_error | 100 | sqrt | 0.85025 |
| squared\_error | 50 | none | 0.8535 |
| squared\_error | 100 | none | 0.85048 |
| absolute\_error | 50 | log2 | 0.84858 |
| absolute\_error | 100 | log2 | 0.85006 |
| absolute\_error | 50 | Sqrt | 0.83527 |
| absolute\_error | 100 | sqrt | 0.85205 |
| absolute\_error | 50 | none | 0.84946 |
| absolute\_error | 100 | none | 0.84464 |
| poission | 50 | log2 | 0.84559 |
| poission | 100 | log2 | 0.84559 |
| poission | 50 | Sqrt | 0.84077 |
| poission | 100 | sqrt | 0.84862 |
| poission | 50 | none | 0.85039 |
| poission | 100 | none | 0.84909 |
| friedman\_mse | 50 | log2 | 0.84943 |
| friedman\_mse | 100 | log2 | 0.85025 |
| friedman\_mse | 50 | Sqrt | 0.85089 |
| friedman\_mse | 100 | sqrt | 0.84781 |
| friedman\_mse | 50 | none | 0.84488 |
| friedman\_mse | 100 | none | 0.84765 |

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

I am choosing the random forest algorithm it giving the highest accuracy of the model.

The best model is created.

Random forest algorithm (criterion="absolute\_error",max\_features="log2",n\_estimators=100)

R2\_score= 0.852056