**Problem Statement:**

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.

**3 stages of problem Identification:**

1.Machine Learning

2.Supervised Learning

3.Regression

**Total No. of rows=**1338 rows

**Total No. of Columns=** 6 columns

**Pre-processing method:**

* Converts categorical data into numerical data.
* Create binary columns and avoid redundancy.

**Multiple Linear Regression:**

In MLR R\_score value is 0.789479

**Support Vector Machine:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Hyperparameter** | **Linear** | **Non Linear (rbf)** | **Poly** | **Sigmoid** |
| **C10** | **-0.001617** | **-0.081969** | **-0.093116** | **-0.090783** |
| **C100** | **0.543281** | **-0.124803** | **-0.099761** | **-0.118145** |
| **C500** | **0.627046** | **-0.124641** | **-0.082028** | **-0.456294** |
| **C1000** | **0.634036** | **-0.117490** | **-0.055505** | **-1.665908** |
| **C1500** | **0.639421** | **-0.112389** | **-0.028732** | **-3.316374** |
| **C2000** | **0.639421** | **-0.107787** | **-0.002702** | **-5.616431** |
| **C3000** | **0.759089** | **-0.096212** | **0.048928** | **-12.019048** |

**Decision Tree:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **S. No** | **criterion** | **Splitter** | **Max Features** | **R-Score Value** |
| **1** | **Squared\_error** | **best** | **sqrt** | **0.729714** |
| **2** | **Squared\_error** | **random** | **sqrt** | **0.698976** |
| **3** | **Squared\_error** | **best** | ***log2*** | **0.66234** |
| **4** | **Squared\_error** | **random** | ***log2*** | **0.621155** |
| **5** | **Squared\_error** | **best** | **none** | **0.683209** |
| **6** | **Squared\_error** | **random** | **none** | **0.687126** |
| **7** | ***friedman\_mse*** | **best** | **sqrt** | **0.710683** |
| **8** | ***friedman\_mse*** | **random** | **sqrt** | **0.614436** |
| **9** | ***friedman\_mse*** | **best** | ***log2*** | **0.759087** |
| **10** | ***friedman\_mse*** | **random** | ***log2*** | **0.693754** |
| **11** | ***friedman\_mse*** | **best** | **none** | **0.695046** |
| **12** | ***friedman\_mse*** | **random** | **none** | **0.673971** |
| **13** | ***absolute\_error*** | **best** | **sqrt** | **0.678780** |
| **14** | ***absolute\_error*** | **random** | **sqrt** | **0.646763** |
| **15** | ***absolute\_error*** | **best** | ***log2*** | **0.709199** |
| **16** | ***absolute\_error*** | **random** | ***log2*** | **0.685710** |
| **17** | ***absolute\_error*** | **best** | ***none*** | **0.681307** |
| **18** | ***absolute\_error*** | **random** | **none** | **0.723519** |
| **19** | ***poisson*** | **best** | **sqrt** | **0.762531** |
| **20** | ***poisson*** | **random** | **sqrt** | **0.633440** |
| **21** | ***Poisson*** | **best** | ***log2*** | **0.768063** |
| **22** | ***poisson*** | **random** | ***log2*** | **0.696448** |
| **23** | ***poisson*** | **best** | **none** | **0.713012** |
| **24** | ***poisson*** | **random** | **none** | **0.724698** |

**Random Forest:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **S. No** | **criterion** | **N\_estimators** | **Max Features** | **R-Score Value** |
| **1** | **Squared\_error** | **50** | **sqrt** | **0.865473** |
| **2** | **Squared\_error** | **100** | **sqrt** | **0.869122** |
| **3** | **Squared\_error** | **50** | ***log2*** | **0.864763** |
| **4** | **Squared\_error** | **100** | ***log2*** | **0.872957** |
| **5** | **Squared\_error** | **50** | **none** | **0.860093** |
| **6** | **Squared\_error** | **100** | **none** | **0.848843** |
| **7** | ***friedman\_mse*** | **50** | **sqrt** | **0.864763** |
| **8** | ***friedman\_mse*** | **100** | **sqrt** | **0.867405** |
| **9** | ***friedman\_mse*** | **50** | ***log2*** | **0.866099** |
| **10** | ***friedman\_mse*** | **100** | ***log2*** | **0.871165** |
| **11** | ***friedman\_mse*** | **50** | **none** | **0.853235** |
| **12** | ***friedman\_mse*** | **100** | **none** | **0.854254** |
| **13** | ***absolute\_error*** | **50** | **sqrt** | **0.869466** |
| **14** | ***absolute\_error*** | **100** | **sqrt** | **0.873541** |
| **15** | ***absolute\_error*** | **50** | ***log2*** | **0.868338** |
| **16** | ***absolute\_error*** | **100** | ***log2*** | **0.874488** |
| **17** | ***absolute\_error*** | **50** | **none** | **0.850908** |
| **18** | ***absolute\_error*** | **100** | **none** | **0.851986** |
| **19** | ***poisson*** | **50** | **sqrt** | **0.867668** |
| **20** | ***poisson*** | **100** | **sqrt** | **0.868369** |
| **21** | ***poisson*** | **50** | ***log2*** | **0.865622** |
| **22** | ***poisson*** | **100** | ***log2*** | **0.872102** |
| **23** | ***poisson*** | **50** | **none** | **0.853900** |
| **24** | ***poisson*** | **100** | **none** | **0.856602** |

**Final Model:**

* **Random Forest** by using the hypertuning parameters (criterion= *‘absolute\_error’,*n\_estimators=100**,** Max\_features=log2).
* **R\_score** =**0.874488**
* This model is closest to the actual value.
* S0, I select this model for the further deployment process.