**REGRESSION ASSIGNMENT**

**Problem statement**

The client is from an insurance company and he wants help in predicting the insurance charges based on the parameters which he has given.

**Dataset**

The dataset contains 1338 rows and 6 columns of data.

**Data pre-processing**

The data is a mix of categorical and numerical value. Here the categorical data is converted into numerical data using get\_dummies function from pandas.

**Creating a good model**

The goal is to create a model with good r\_score using machine learning algorithm. Here the output is numerical so “regressor” is used. It’s a supervised learning as the input and outputs is clearly defined

**1.MULTIPLE LINEAR REGRESSION**

**(r\_score Value) = 0.7894790349867009**

**2.SUPPORT VECTOR MACHINE**

**(r\_score value) using SVM (C=3000, kernel=’rbf’) = 0.86633939530**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **S.NO** | **HYPER PARAMETER** | **LINEAR**  **(r\_score value)** | **RBF(NL)**  **(r\_score value)** | **POLY (r\_score value)** | **SIGMOID**  **(r\_score value)** |
| **1** | **C10** | **0.46246841423** | **-0.0322732939** | **0.03871622276** | **0.039307143782** |
| **2** | **C100** | **0.62887928573** | **0.32003178320** | **0.61795696240** | **0.527610354651** |
| **3** | **C500** | **0.76310580538** | **0.66429846451** | **0.82636835412** | **0.444606103386** |
| **4** | **C1000** | **0.76493117385** | **0.81020648517** | **0.85664876759** | **0.287470694869** |
| **5** | **C2000** | **0.74404183081** | **0.85477664253** | **0.86055792585** | **-0.59395097312** |
| **6** | **C3000** | **0.74142365992** | **0.86633939530** | **0.85989300844** | **-2.12441947866** |

**3.DECISION TREE**

**(r\_score value) using Decision Tree (Criterion = ’squared\_error’, Splitter =’best’, Max Features = ‘log2’) = 0.7622634222198909**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **S.NO** | **CRITERION** | **SPLITTER** | **MAX FEATURES** | **r\_SCORE VALUE** |
| **1** | **squared\_error** | **best** | **sqrt** | **0.7419518791417663** |
| **2** | **squared\_error** | **random** | **sqrt** | **0.6604531205806926** |
| **3** | **squared\_error** | **best** | **log2** | **0.7622634222198909** |
| **4** | **squared\_error** | **random** | **log2** | **0.7086085535707132** |
| **5** | **squared\_error** | **best** | **None** | **0.6917019235660604** |
| **6** | **squared\_error** | **random** | **None** | **0.6745835407426458** |
| **7** | **friedman\_mse** | **best** | **sqrt** | **0.7474521020283519** |
| **8** | **friedman\_mse** | **random** | **sqrt** | **0.7071763616113942** |
| **9** | **friedman\_mse** | **best** | **log2** | **0.7123827095059756** |
| **10** | **friedman\_mse** | **random** | **log2** | **0.6687016485593593** |
| **11** | **friedman\_mse** | **best** | **None** | **0.7039605455966418** |
| **12** | **friedman\_mse** | **random** | **None** | **0.7163760982338949** |
| **13** | **absolute\_error** | **best** | **sqrt** | **0.632362651551901** |
| **14** | **absolute\_error** | **random** | **sqrt** | **0.5723011489529024** |
| **15** | **absolute\_error** | **best** | **log2** | **0.6837227876950855** |
| **16** | **absolute\_error** | **random** | **log2** | **0.7120830476824139** |
| **17** | **absolute\_error** | **best** | **None** | **0.6743568812053272** |
| **18** | **absolute\_error** | **random** | **None** | **0.7461349658006042** |
| **19** | **poisson** | **best** | **sqrt** | **0.7578205867225241** |
| **20** | **poisson** | **random** | **sqrt** | **0.7522688054881514** |
| **21** | **poisson** | **best** | **log2** | **0.7551368188344012** |
| **22** | **poisson** | **random** | **log2** | **0.7134405358764765** |
| **23** | **poisson** | **best** | **None** | **0.7202532638179917** |
| **24** | **poisson** | **random** | **None** | **0.6287647750598568** |

**4.Random Forest**

**(r\_score value) using Random Forest (estimators =’100’, Criterion = ’absolute \_error’,**

**Max Features = ‘sqrt’) = 0.8751225531082324**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **S.NO** | **n\_Estimators** | **CRITERION** | **MAX FEATURES** | **r\_SCORE VALUE** |
| **1** | **50** | **squared\_error** | **sqrt** | **0.8556825535116126** |
| **2** | **100** | **squared\_error** | **sqrt** | **0.8707618436170819** |
| **3** | **50** | **squared\_error** | **log2** | **0.8710049364286552** |
| **4** | **100** | **squared\_error** | **log2** | **0.8676168571557872** |
| **5** | **50** | **squared\_error** | **None** | **0.855584085885751** |
| **6** | **100** | **squared\_error** | **None** | **0.8538691375323095** |
| **7** | **50** | **friedman\_mse** | **sqrt** | **0.8657549896442687** |
| **8** | **100** | **friedman\_mse** | **sqrt** | **0.8711198030271314** |
| **9** | **50** | **friedman\_mse** | **log2** | **0.8652808465700027** |
| **10** | **100** | **friedman\_mse** | **log2** | **0.8682120180799379** |
| **11** | **50** | **friedman\_mse** | **None** | **0.8528541027329369** |
| **12** | **100** | **friedman\_mse** | **None** | **0.8507193188602692** |
| **13** | **50** | **absolute\_error** | **sqrt** | **0.8701295431631535** |
| **14** | **100** | **absolute\_error** | **sqrt** | **0.8752694458966623** |
| **15** | **50** | **absolute\_error** | **log2** | **0.8735148442923265** |
| **16** | **100** | **absolute\_error** | **log2** | **0.8720485835554923** |
| **17** | **50** | **absolute\_error** | **None** | **0.847796528876458** |
| **18** | **100** | **absolute\_error** | **None** | **0.8547079328682656** |
| **19** | **50** | **poisson** | **sqrt** | **0.8631073900547437** |
| **20** | **100** | **poisson** | **sqrt** | **0.8702182211856742** |
| **21** | **50** | **poisson** | **log2** | **0.868963722401279** |
| **22** | **100** | **poisson** | **log2** | **0.8703030253128295** |
| **23** | **50** | **poisson** | **None** | **0.8518697716699726** |
| **24** | **100** | **poisson** | **None** | **0.8510379237984672** |

**Final Model**

The final model is **Machine learning>>Regression>>Random Forest**

Justification: The r\_score value is high comparing to other algorithms.