**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.

**Identify your problem statement**

* I need to predict the charges columns
* The data is given with clear data so its comes under supervised learning
* And we are going to predict the number --🡪 ML (Regression)

**Total number of rows, columns**

* Rows : 1338
* Columns : 6

**The pre-processing method**

Here in this dataset we have categorical data , When its comes to machine learning we need numbers because most ML algorithms doesn’t work with strings so here in this data set it is nominal data set so now we are going to convert this into numerical ***[One Hot Encoding]***

**Models With R2\_Scores**

**ML-Regression**

***MLR- R2\_score* : 0.7891**

**SVM**

|  |  |  |  |
| --- | --- | --- | --- |
| ***S.no*** | ***kernel*** | ***C*** | ***R2\_SCORE*** |
| 1 | linear | 100 | 0.5950 |
| 2 | Linear | 1000 | 0.6898 |
| 3 | linear | 10000 | 0.7413 |
| 4 | rbf | 100 | -0.12 |
| 5 | rbf | 1000 | -0.10 |
| 6 | rbf | 10000 | 0.06 |
| 7 | poly | 100 | -0.097 |
| 8 | poly | 1000 | -0.042 |
| 9 | poly | 10000 | 0.4098 |
| 10 | sigmoid | 100 | -0.11 |
| 11 | sigmoid | 1000 | -1.14 |
| 12 | sigmoid | 10000 | -77.275 |

**BEST 0.7213**

**WORST : -77.275**

**Decision Tree**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **S.no** | **Criterion** | **Splitter** | **Max** | **R2\_score** |
| **1** | squared\_error | none | None | 0.67 |
| **2** | squared\_error | best | 5 | 0.86 |
| **3** | squared\_error | random | 5 | 0.87 |
| **4** | friedman\_mse | none | none | 0.65 |
| **5** | friedman\_mse | best | 5 | 0.8814 |
| **6** | friedman\_mse | random | 5 | 0.8761 |
| **7** | absolute\_error | none | none | 0.72 |
| **8** | absolute\_error | Best | 5 | 0.8827 |
| **9** | absolute\_error | random | 5 | 0.85 |
| **10** | poisson | none | none | 0.66 |
| **11** | poisson | Best | 5 | 0.8823 |
| **12** | Poisson | random | 5 | 0.8669 |

**BEST : 0.8827**

**WORST : 0.66**

**Random Forest**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **S.no** | **n\_estimators** | **criterion** | **max\_depth** | **R2\_score** |
| 1 | Default = 100 | squared\_error | none | **0.8525** |
| 2 | 1000 | squared\_error | 5 | **0.888325** |
| 3 | 10000 | squared\_error | 5 | **0.88809** |
| 4 | 100 | absolute\_error | 5 | **0.88770** |
| 5 | 1000 | absolute\_error | 5 | **0.888261** |
| 6 | 100 | friedman\_mse | 5 | **0.88806** |
| 7 | 1000 | friedman\_mse | 5 | **0.888196** |
| 8 | 100 | poisson | 5 | **0.888879** |
| 9 | 1000 | poisson | 5 | **0.88820** |

**BEST : 0.888879**

**WORST : 0.8525**

**Final model**

*Using Random Forest algorithm gave me the best results when I compare to other algorithms*