**Insurance Prediction Assignment Using ML**

**1.)Identify your problem statement.**

Stage 1- Machine Learning

Based on client data and requirement it is Machine Learning.Because Datas also in excel datas.

Stage 2-Supervised Learning

Because input and output was very clear both are present in datas.

Stage 3-Regression

The output is insurance charges so definitely it will be numbers.so it is regression.

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

dataset.shape=(1338, 6)

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

In python we cant import categorical data(Text) so we need to convert text as number.So we use dataset=pd.get\_dummies(dataset,drop\_first=True).

**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.**

R\_Score: 0.872325076

**5.) All the research values (r2\_score of the models) should be documented. (You can make tabulation or screenshot of the results.)**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **S.No** | **Algorithm\_Name** | **criterion** | **splitter** | **Max\_Depth** | **R\_Score** |
| **1** | Multiple\_Linear\_Regression |  |  |  | 0.789479035 |
| 2 | Decision Tree Algorithm | squared\_error | best | None | 0.710560544 |
| 3 | Decision Tree Algorithm | squared\_error | best | auto | 0.709596634 |
| 4 | Decision Tree Algorithm | squared\_error | Best | sqrt | 0.701335256 |
| 5 | Decision Tree Algorithm | squared\_error | Best | log2 | 0.701335256 |
| 6 | Decision Tree Algorithm | squared\_error | random | None | 0.729841955 |
| 7 | Decision Tree Algorithm | squared\_error | random | auto | 0.701479133 |
| 8 | Decision Tree Algorithm | squared\_error | random | Sqrt | 0.656961308 |
| 9 | Decision Tree Algorithm | squared\_error | random | log2 | 0.653326606 |
| 10 | Decision Tree Algorithm | friedman\_mse | best | None | 0.691452388 |
| 11 | Decision Tree Algorithm | friedman\_mse | best | auto | 0.687521008 |
| 12 | Decision Tree Algorithm | friedman\_mse | best | sqrt | 0.730934887 |
| 13 | Decision Tree Algorithm | friedman\_mse | best | log2 | 0.702290407 |
| 14 | Decision Tree Algorithm | friedman\_mse | random | None | 0.677453406 |
| 15 | Decision Tree Algorithm | friedman\_mse | random | auto | 0.699134839 |
| 16 | Decision Tree Algorithm | friedman\_mse | random | sqrt | 0.705846817 |
| 17 | Decision Tree Algorithm | friedman\_mse | random | log2 | 0.654913716 |
| 18 | Decision Tree Algorithm | absolute\_error | best | None | 0.656304798 |
| 19 | Decision Tree Algorithm | absolute\_error | best | auto | 0.664424721 |
| 20 | Decision Tree Algorithm | absolute\_error | best | sqrt | 0.752777092 |
| 21 | Decision Tree Algorithm | absolute\_error | best | log2 | 0.734779452 |
| 22 | Decision Tree Algorithm | absolute\_error | random | None | 0.671174481 |
| 23 | Decision Tree Algorithm | absolute\_error | random | auto | 0.709308369 |
| 24 | Decision Tree Algorithm | absolute\_error | random | sqrt | 0.729003378 |
| 25 | Decision Tree Algorithm | absolute\_error | random | log2 | 0.666011918 |
| 26 | Decision Tree Algorithm | poisson | best | None | 0.718618715 |
| 27 | Decision Tree Algorithm | poisson | best | auto | 0.732192292 |
| 28 | Decision Tree Algorithm | poisson | best | sqrt | 0.708789212 |
| 29 | Decision Tree Algorithm | poisson | best | log2 | 0.748676306 |
| 30 | Decision Tree Algorithm | poisson | random | None | 0.750757926 |
| 31 | Decision Tree Algorithm | poisson | random | auto | 0.757263065 |
| 32 | Decision Tree Algorithm | poisson | random | sqrt | 0.618913937 |
| 33 | Decision Tree Algorithm | poisson | random | log2 | 0.6872858 |
| 34 | Random\_Forest | squared\_error | random\_state=0 |  | 0.853830791 |
| 35 | Random\_Forest | squared\_error |  | None | 0.854030092 |
| 36 | Random\_Forest | squared\_error |  | auto | 0.855886598 |
| 37 | Random\_Forest | squared\_error |  | sqrt | 0.871665644 |
| 38 | Random\_Forest | squared\_error |  | log2 | 0.869721558 |
| 39 | Random\_Forest | friedman\_mse |  | None | 0.85401739 |
| 40 | Random\_Forest | friedman\_mse |  | auto | 0.852641252 |
| 41 | Random\_Forest | friedman\_mse |  | sqrt | 0.868670482 |
| 42 | Random\_Forest | friedman\_mse |  | log2 | 0.868136223 |
| 43 | Random\_Forest | absolute\_error |  | None | 0.853846787 |
| 44 | Random\_Forest | absolute\_error |  | auto | 0.854350481 |
| 45 | Random\_Forest | absolute\_error |  | sqrt | 0.871194845 |
| 46 | Random\_Forest | absolute\_error |  | log2 | 0.871142129 |
| 47 | Random\_Forest | poisson |  | None | 0.853823396 |
| 48 | Random\_Forest | poisson |  | auto | 0.850348426 |
| **49** | **Random\_Forest** | **poisson** |  | **sqrt** | **0.872325076** |
| 50 | Random\_Forest | poisson |  | log2 | 0.86843979 |
| 51 | Random\_Forest | n\_estimators=50 | random\_state=0 |  | 0.853830791 |

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

0.872325

Above 0.9 is the good model but using three model I got 0.872325 is maximum and nearer to 0.9, that’s why choose this value as final model.