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 :**

Machine Learning – Supervised – Regression

**2.) Tell basic info about the dataset (Total number of rows, columns) :** 1338 rows × 6 columns

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

**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.**

Final model is Decision Tree as it has the highest score of 85%.

To find following the machine learning regression method using in r2 value

1.**MULTIPLE LINEAR REGRESSION** (R2 value) = 0.7100

2. **SUPPORT VECTOR MACHINE:**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **S.No** | **Hyper Parameter (r score)** | **Linear (r score)** | **Rbf(Non-Linear) (r score)** | **Poly  (r score)** | **Sigmoid (r score)** |
| 1. | C=10 | -9.49 | -345.6 | -16.58 | -33699.7 |
| 2. | C=100 | -0.62 | -14.6 | -12.42 | -319.3 |
| 3. | C=1000 | 0.04 | -11.6 | -10.71 | -1.98 |
| 4. | C=3000 | 0.66 | -10.9 | -7.83 | -0.36 |

**3.DECISION TREE:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **S.No** | **criterion** | **splitter** | **max\_features** | **R Score** |
| 1. | *squared\_error* | *best* | *sqrt* | 0.60 |
| 2. | *squared\_error* | best | log2 | 0.69 |
| 3. | *squared\_error* | random | *sqrt* | 0.65 |
| 4. | *squared\_error* | random | log2 | 0.72 |
| 5. | Friedman\_mse | *best* | *sqrt* | 0.70 |
| 6. | Friedman\_mse | best | log2 | 0.72 |
| 7. | Friedman\_mse | random | *sqrt* | 0.71 |
| 8. | Friedman\_mse | random | log2 | 0.64 |
| 9. | Absolute\_error | *best* | *sqrt* | 0.71 |
| 10. | Absolute\_error | best | log2 | 0.72 |
| 11. | Absolute\_error | random | *sqrt* | 0.61 |
| 12. | Absolute\_error | random | log2 | 0.66 |
| 13. | poisson | *best* | *sqrt* | 0.75 |
| 14. | poisson | best | log2 | 0.76 |
| 15. | poisson | random | *sqrt* | 0.69 |
| 16. | poisson | random | log2 | 0.68 |
| 17. | *squared\_error* | best | Default=none | 0.72 |
| 18. | *squared\_error* | random | Default=none | 0.73 |
| 19. | Friedman\_mse | best | Default=none | 0.74 |
| 20. | Friedman\_mse | random | Default=none | 0.75 |
| 21. | poisson | best | Default=none | 0.75 |
| 22. | poisson | random | Default=none | 0.74 |

**4. RANDOM FOREST:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **S.NO** | **criterion** | **n\_estimators** | **max\_features** | **R2\_score** |
| 1. | Square\_root | 10 | *sqrt* | 0.84 |
| 2. | Square\_root | 10 | log2 | 0.85 |
| 3. | Square\_root | 100 | *sqrt* | 0.84 |
| 4. | Square\_root | 100 | log2 | 0.84 |
| 5. | Absolute\_error | 10 | *sqrt* | 0.83 |
| 6. | Absolute\_error | 10 | log2 | 0.82 |
| 7. | Absolute\_error | 100 | *sqrt* | 0.84 |
| 8. | Absolute\_error | 100 | log2 | 0.85 |
| 9. | Friedman\_mse | 10 | *sqrt* | 0.83 |
| 10. | Friedman\_mse | 10 | log2 | 0.80 |
| 11. | Friedman\_mse | 100 | *sqrt* | 0.85 |
| 12. | Friedman\_mse | 100 | log2 | 0.84 |
| 13. | poisson | 10 | *sqrt* | 0.84 |
| 14. | poisson | 10 | log2 | 0.83 |
| 15. | poisson | 100 | *sqrt* | 0.84 |
| 16. | poisson | 100 | log2 | 0.84 |