**Assignment-Regression Algorithm**

**Problem Statement or Requirement:**

A client’s requirement is, that he wants to predict the insurance charges based on several parameters. The Client has provided the dataset of the same.

1. **Identify your problem statement**

Predict the insurance charges.

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

Total number of rows = 1338, Total number of rows Columns = 6.

In Columns Headers are (Age, Sex, BMI, Children, Smoker, And Charges)

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

The Client Given Dataset, I convert the string to a number

I use the pre-processing method **Nominal – One Hot Encoding.**

1. **Develop a good model with r2\_score. You can use any machine learning algorithm; you can create many models. All the research values are in tabulations**.
2. Simple Linear Regression R2 Score Value = 0.78913454847886.
3. Multiple Linear Regression R2 Score Value = 0.7891345484788599.
4. SVM R2 Score Values.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **S.No.** | **C** | **kernel** | **gamma** | **R-Value** |
| 1 | 100 | linear | scale | 0.543221029625509 |
| 2 | 300 | rbf | scale | -0.12603141946425 |
| 3 | 1000 | poly | scale | -0.054656238441107785 |
| 4 | 3000 | sigmoid | scale | -12.544529622891135 |
| 5 | 3000 | linear | scale | 0.7590272859363828 |
| 6 | 6000 | linear | scale | 0.7648493015935964 |

The **SVM Regression** uses R2 Value ((C-6000),(kernel-linear),(gamma-scale))= 0.7648493015935964.

1. Decision Tree

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **S.No.** | **criterion** | **splitter** | **max\_features** | **min\_samples\_leaf** | **R-Value** |
| 1 | friedman\_mse | best | None | **1** | 0.6780033766726528 |
| 2 | squared\_error | random | log2 | 1 | 0.7157641702589957 |
| 3 | squared\_error | random | log2 | 0.01 | 0.76248628663105 |
| 4 | squared\_error | random | sqrt | 0.01 | 0.7242675613867426 |
| 5 | absolute\_error | random | sqrt | 0.01 | 0.7007141051013618 |
| 6 | absolute\_error | random | sqrt | 1 | 0.731275314944811 |
| 7 | friedman\_mse | random | sqrt | 1 | 0.6890590923739732 |
| 8 | friedman\_mse | random | log2 | 1 | 0.6399352993770677 |
| 9 | friedman\_mse | random | log2 | 0.01 | 0.7476318060737843 |
| 10 | absolute\_error | best | sqrt | 1 | 0.7808576848991753 |
| 11 | absolute\_error | best | sqrt | 0.01 | 0.7438310455611499 |
| 12 | absolute\_error | best | log2 | 1 | 0.7049922788250593 |
| 13 | absolute\_error | best | log2 | 0.01 | 0.8768181657455923 |
| 14 | friedman\_mse | best | log2 | 0.01 | 0.7914940626464114 |
| 15 | friedman\_mse | best | sqrt | 0.01 | 0.8535810077955392 |
| 16 | squared\_error | best | sqrt | 0.01 | 0.8284212007173694 |

The **Decision Tree Regression** use R2Value (criterion='absolute\_error', splitter='best', max\_features='log2', min\_samples\_leaf=0.01)= 0.8768181657455923.

1. Random Forest

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **S.No.** | **n\_estimators** | **criterion** | **oob\_score** | **min\_samples\_leaf** | **R-Value** |
| 1 | 600 | squared\_error | False | 1 | 0.8511373933094158 |
| 2 | 3000 | absolute\_error | False | 0.01 | 0.8861369761670236 |
| 3 | 3000 | squared\_error | False | 0.01 | 0.8854064872395006 |
| 4 | 3000 | friedman\_mse | False | 0.01 | 0.8854668422210327 |
| 5 | 3000 | poisson | False | 0.01 | 0.885106933323623 |

The **Random Forest Regression** use R2Value (n\_estimators=3000, criterion='absolute\_error', oob\_score=False, min\_samples\_leaf=0.01)= 0.8861369761670236.

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

The Final model is **Random Forest Regression**.

The Parameter are (n\_estimators=3000, criterion=’absolute\_error’, oob\_score=False, min\_samples\_leaf=0.01)=

The R2Value is 0.8861369761670236.

The **Random Forest Regression** R2Value is **higher compared** to simple linear regression, multiple linear regression, SVM, and Decision tree, that’s why the finalized model comes to Random Forest Regression.