**Regression Assignment**

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

Stage 1: Machine Learning

Stage 2: Supervised Learning

Stage 3: Regression

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

Total number of rows: 1339

Total number of columns: 6

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

Ordinal Data

1. **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.**

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

**or screenshot of the results.)**

**Regression Assignment - Model Creation with R2 Value**

**1.Multiple Linear Regression:**

R2 Value: 0.7895

**2.SVMR: (Without Standardization)**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **S.no** | **Hyper parameter** | **linear** | **poly** | **rbf** | **sigmoid** |
| 1 | default | -0.1117 | -0.0642 | -0.0884 | -0.0899 |
| 2 | 0.01 | -0.0797 | -0.0893 | -0.0896 | -0.0897 |
| 3 | 0.1 | -0.1220 | -0.0862 | -0.0895 | -0.0897 |
| 4 | 1 | -0.1116 | -0.0642 | -0.0884 | -0.0899 |
| 5 | 10 | -0.0016 | -0.0931 | -0.0819 | -0.0907 |
| 6 | 100 | 0.5432 | -0.0997 | -0.1248 | -0.1181 |
| 7 | 1000 | 0.6340 | -0.0555 | -0.1175 | -1.6659 |
| 8 | 10000 | 0.7444 | 0.3529 | -0.0172 | -119.5185 |

**SVMR: (With Standardization)**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **S.no** | **Hyper parameter** | **linear** | **poly** | **rbf** | **sigmoid** |
| 1 | default | -0.0101 | -0.0756 | -0.0833 | -0.0754 |
| 2 | 0.01 | -0.0888 | -0.0895 | -0.0896 | -0.0895 |
| 3 | 0.1 | -0.0809 | -0.0883 | -0.0890 | -0.0882 |
| 4 | 1 | -0.0101 | -0.0756 | -0.0833 | -0.0754 |
| 5 | 10 | 0.4624 | 0.0387 | -0.0322 | 0.0393 |
| 6 | 100 | 0.6288 | 0.6179 | 0.3200 | 0.5276 |
| 7 | 1000 | 0.7649 | 0.8566 | 0.8102 | 0.2874 |
| 8 | 10000 | 0.7414 | 0.8591 | 0.8779 | -34.1515 |

**SVMR R2 Value: 0.8779**

**3.Decision Tree:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **S.no** | **criterion** | **splitter** | **max\_features** | **R2 Value** |
| 1 | default | - | - | 0.6887 |
| 2 | squared\_error | best | sqrt | 0.7192 |
| 3 | squared\_error | random | sqrt | 0.7001 |
| 4 | squared\_error | best | log2 | 0.7402 |
| 5 | squared\_error | random | log2 | 0.6990 |
| 6 | friedman\_mse | best | sqrt | 0.7232 |
| 7 | friedman\_mse | random | sqrt | 0.6951 |
| 8 | friedman\_mse | best | log2 | 0.6131 |
| 9 | friedman\_mse | random | log2 | 0.6175 |
| 10 | absolute\_error | best | sqrt | 0.5790 |
| 11 | absolute\_error | random | sqrt | 0.6302 |
| 12 | absolute\_error | best | log2 | 0.6427 |
| 13 | absolute\_error | random | log2 | 0.7227 |
| 14 | poisson | best | sqrt | 0.7150 |
| 15 | poisson | random | sqrt | 0.6736 |
| 16 | poisson | best | log2 | 0.7290 |
| 17 | poisson | random | log2 | 0.6878 |
| 18 | squared\_error | best | Int(5) | 0.6874 |
| 19 | squared\_error | random | Int(5) | 0.7202 |
| 20 | squared\_error | best | Float(10) | 0.6854 |
| 21 | squared\_error | random | Float(10) | 0.6753 |
| 22 | friedman\_mse | best | Int(5) | 0.6952 |
| 23 | friedman\_mse | random | Int(5) | 0.7212 |
| 24 | friedman\_mse | best | Float(10) | 0.7108 |
| 25 | friedman\_mse | random | Float(10) | 0.7000 |
| 26 | absolute\_error | best | Int(5) | 0.7014 |
| 27 | absolute\_error | random | Int(5) | 0.6699 |
| 28 | absolute\_error | best | Float(10) | 0.6790 |
| 29 | absolute\_error | random | Float(10) | 0.7201 |
| 30 | poisson | best | Int(5) | 0.7293 |
| 31 | poisson | random | Int(5) | 0.7034 |
| 32 | poisson | best | Float(10) | 0.7248 |
| 33 | poisson | random | Float(10) | 0.6732 |

Decision Tree R2 Value: 0.7402

**4.Random Forest:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **S.no** | **n\_estimators** | **criterion** | **max\_features** | **R2 Value** |
| 1 | default | **-** | **-** | 0.8498 |
| 2 | 100 | squared\_error | sqrt | 0.8705 |
| 3 | 100 | squared\_error | log2 | 0.8723 |
| 4 | 100 | absolute\_error | sqrt | 0.8707 |
| 5 | 100 | absolute\_error | log2 | 0.8739 |
| 6 | 100 | friedman\_mse | sqrt | 0.8421 |
| 7 | 100 | friedman\_mse | log2 | 0.8669 |
| 8 | 100 | poisson | sqrt | 0.8540 |
| 9 | 100 | poisson | log2 | 0.8710 |

**Random Forest R2 Value: 0.8739**

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

Best model for Machine Learning is **SVMR: (With Standardization) - 0.8779** and **Random Forest - 0.8739.**