**Regression Assignment:**

**1.) Identify your problem statement**

**Problem Statement**: Predict a best model using AI predictions to find the insurance charges based on a person’s age, BMI, children, Sex and smoker/nonsmoker.

**Problem Identification**: Machine Learning, Supervised Learning and Regression

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

The input parameters (columns) are

1. age
2. sex (male/female)
3. bmi (body mass index)
4. children (no. of children the person has)
5. smoker or not

The output parameters (columns) are

1. charges (insurance charges)

The dataset consists of 1338 (rows) data of insurance charges for each person for various input parameters

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

Here we are converting two string data to nominal data

(i)male/female to nominal data 1/0

(ii)smoker/nonsmoker to nominal data 1/0

Syntax: dataset=pd.get\_dummies(dataset,dtype=int, 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.**

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

**Model1: Multiple regression Model:**

r\_score=0.7277

**Model2: Support Vector Machine Model:**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| S.No | C Value | Kernel:linear | Kernel:rbf | Kernel:poly | Kernel:sigmoid |
| 1 | 10 | 0.0668 | -0.3447 | -0.0480 | -0.1193 |
| 2 | 100 | 0.5524 | -0.1254 | -0.5198 | -0.1504 |
| 3 | 500 | 0.6122 | -0.1279 | -0.0291 | -0.5383 |
| 4 | 1000 | 0.6549 | -0.1224 | -0.0006 | -1.915 |
| 5 | 2000 | 0.7300 | -0.1108 | 0.0555 | -6.6213 |
| 6 | 3000 | 0.7103 | -0.0966 | 0.1102 | -14.620 |

**Model3: Decision Tree Algorithm: r\_score with hyper tuning parameters**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| S.No | Criterion | splitter | max\_features | R\_score |
| 1 | Squared\_error | random | sqrt | 0.5591 |
| 2 | Squared\_error | random | log2 | 0.5907 |
| 3 | Squared\_error | best | sqrt | 0.5267 |
| 4 | Squared\_error | best | log2 | 0.5789 |
| 5 | Friedman\_mse | random | sqrt | 0.4935 |
| 6 | Friedman\_mse | random | log2 | 0.6248 |
| 7 | Friedman\_mse | best | sqrt | 0.5913 |
| 8 | Friedman\_mse | best | log2 | 0.5668 |
| 9 | Absolute\_error | random | sqrt | 0.6033 |
| 10 | Absolute\_error | random | log2 | 0.6359 |
| 11 | Absolute\_error | best | sqrt | 0.2863 |
| 12 | Absolute\_error | best | log2 | 0.6044 |
| 13 | poisson | random | sqrt | 0.6412 |
| 14 | poisson | random | log2 | 0.5713 |
| 15 | poisson | best | sqrt | 0.6240 |
| 16 | poisson | best | log2 | 0.6244 |

**Model: Random Forest Model:r\_score with random\_state=0 and other hyper tuning parameters**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| S.No | n-estimator | criterion | max\_features | r\_score |
| 1 | 50 | squared\_error | sqrt | 0.8482 |
| 2 | 100 | squared\_error | sqrt | 0.8501 |
| 3 | 200 | squared\_error | sqrt | 0.8499 |
| 4 | 50 | squared\_error | log2 | 0.8451 |
| 5 | 100 | squared\_error | log2 | 0.8469 |
| 6 | 200 | squared\_error | log2 | 0.8488 |
| 7 | 50 | friedman\_mse | sqrt | 0.8497 |
| 8 | 100 | friedman\_mse | sqrt | 0.8499 |
| 9 | 200 | friedman\_mse | sqrt | 0.8518 |
| 10 | 50 | friedman\_mse | log2 | 0.8520 |
| 11 | 100 | friedman\_mse | log2 | 0.8533 |
| 12 | 200 | friedman\_mse | log2 | 0.8527 |
| 13 | 50 | absolute\_error | sqrt | 0.8473 |
| 14 | 100 | absolute\_error | sqrt | 0.8537 |
| 15 | 200 | absolute\_error | sqrt | 0.8516 |
| 16 | 50 | absolute\_error | log2 | 0.8521 |
| 17 | 100 | absolute\_error | log2 | 0.8477 |
| 18 | 200 | absolute\_error | log2 | 0.8506 |
| 19 | 50 | poisson | sqrt | 0.8496 |
| 20 | 100 | poisson | sqrt | 0.8511 |
| 21 | 200 | poisson | sqrt | 0.8504 |
| 22 | 50 | poisson | log2 | 0.8505 |
| 23 | 100 | poisson | log2 | 0.8520 |
| 24 | 200 | poisson | log2 | 0.8510 |

Output:

**Best Model: Random Forest Model**

**hyper parameters :**

1. **n-estimator=100**
2. **criterion=absolute\_error**
3. **max\_features=sqrt**

**r\_score value= 0.8537**