**Machine Learning – Regression**

**Problem Statement / Requirements :**

1. Insurance Charges Prediction
2. Dataset – ‘insurance \_pre.csv’

Stage 1: Machine Learning

Stage 2: Supervised Learning

Stage 3: Regression

1. **Multiple Linear Regression**
   1. Dataset – insurance\_pre.csv
   2. One or more Inputs
      1. age
      2. bmi
      3. children
      4. sex\_male
      5. smoker\_yes
   3. Output
      1. charges Prediction

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| age  (X) | bmi  (X) | children (X) | sex\_male (X) | smoker\_yes (X) | Dependent  (Y)  charges |
|  |  |  |  |  |  |
|  |  |  |  |  |  |

weight - 257.8006705 , 321.06004271, 469.58113407, -41.74825718,23418.6671912

bias - -12057.244846

r2\_score = 0.789

1. **Support Vector Machine**
2. Dataset – insurance\_pre.csv
3. One or more Inputs
   * 1. age
     2. bmi
     3. children
     4. sex\_male
     5. smoker\_yes
4. Output
   * 1. charges Prediction

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Support Vector Machine | kernel | C  penalty | Standardization | r2\_score Accuracy |
| SVM() |  |  |  | 0.088 |
|  | linear |  |  | -0.111 |
|  | poly |  |  | -0.064 |
|  | sigmoid |  |  | -0.089 |
|  |  | C=10.0 |  | -0.0819 |
|  |  |  |  |  |

* Bias and support vectors are calculated in SVM.
* Sklearn.svm.SVC
* Hyper tuning parameter
* **kernel*{‘linear’, ‘poly’, ‘rbf’, ‘sigmoid’, ‘precomputed’} or callable, default=’rbf’***
* **C*float, default=1.0 Penalty***

1. **Decision Tree** 
   1. Dataset – insurance\_pre.csv
   2. One or more Inputs
      1. age
      2. bmi
      3. children
      4. sex\_male
      5. smoker\_yes
   3. Output
      1. charges Prediction

|  |  |  |  |
| --- | --- | --- | --- |
|  | criterion | splitter | r2\_score |
| DT() |  |  | 0.717 |
|  | squared\_error |  | 0.680 |
|  | absolute\_error |  | 0.660 |
|  | friedman\_mse |  | 0.693 |
|  | poisson |  | 0.731 |
|  |  | best | 0.698 |
|  |  | random | 0.744 |
|  | squared\_error | best | 0.712 |
|  | absolute\_error | best | 0.668 |
|  | friedman\_mse | best | 0.687 |
|  | poisson | best | 0.710 |
|  | **friedman\_mse** | **random** | **0.769** |
|  | absolute\_error | random | 0.739 |
|  | squared\_error | random | 0.718 |
|  | **poisson** | **random** | **0.769** |

* Graphs are generated in decision tree using matplotlib
* sklearn.tree.DecisionTreeRegressor
* hyper tuning parameter
* **criterion*{“squared\_error”, “friedman\_mse”, “absolute\_error”, “poisson”}, default=”squared\_error”***
* **splitter*{“best”, “random”}, default=”best”***

1. **Random Forest**
2. Dataset – insurance\_pre.csv
3. One or more Inputs
   * 1. age
     2. bmi
     3. children
     4. sex\_male
     5. smoker\_yes
4. Output
   * 1. charges Prediction

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | n\_estimators | random\_state | criterion | r2\_score |
| **RF()** |  |  |  | **0.856** |
|  | **100** |  |  | **0.858** |
|  | 50 |  |  | 0.848 |
|  | 150 |  |  | 0.856 |
|  | 100 | 0 |  | 0.853 |
|  | 100 | 0 | squared\_error | 0.853 |
|  | 100 | 0 | absolute\_error | 0.852 |
|  | 100 | 0 | friedman\_mse | 0.854 |
|  | 100 | 0 | poisson | 0.852 |
|  |  |  |  |  |
|  |  |  |  |  |

* Sklearn.ensemble.RandomForestRegressor
* Hyper tuning parameter
* **n\_estimators: *int, default=100* -** The number of trees in the forest.
* **random\_state: *int, RandomState instance or None, default=None***
* **criterion*{“squared\_error”, “absolute\_error”, “friedman\_mse”, “poisson”}, default=”squared\_error”***

|  |  |  |
| --- | --- | --- |
| S.No | Algorithm | r2\_score - Accuracy |
| 1 | Multiple Linear Regression | 0.789 |
| 2 | Support Vector Machine | 0.088 |
| 3 | Decision Tree | 0.717 |
| 4 | **Random Forest** | **0.856** |

After Hyper Tuning Parameter

|  |  |  |
| --- | --- | --- |
| S.No | Algorithm | r2\_score - Accuracy |
| 1 | Multiple Linear Regression | 0.789 |
| 2 | Support Vector Machine | 0.088 |
| 3 | Decision Tree | 0.769 |
| **4** | **Random Forest** | **0.858** |

Using the dataset, ‘insurance\_pre.csv’, **Random Forest** gives the best accuracy r2 value