**Assignment-Regression Algorithm**

1. **Problem statement:**

* Client wants to predict the insurance charges based on Age, BMI, Sex, No. of children and smoking criteria in the client’s dataset.

1. **Basic information about dataset:**

Machine Learning 🡪 Supervised learning 🡪 Regression 🡪 Support vector Machine 🡪 “poly’’, C=3000 🡪 Highest value **r2 value** = 0.839

* Rows=1338
* Columns=6

1. **Pre-processing method:**

Nominal data – String to number

* Sex – if male means true, female means false.
* Smoker – if yes means true, No means false.

1. **Final model:**

Machine Learning

Supervised learning

Regression

Support vector Machine

“poly’’, C=3000

Highest value **r2 value = 0.839**

1. **The research values(r2\_score) is documented as tabulation as below:**

* ***MULTIPLE LINEAR REGRESSION***: (r2 value),
* When (random\_state= 42) is **0.69**
* When (random\_state= 0) is **0.71**
* ***SUPPORT VECTOR MACHINE***: (r2 value),

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| SI.NO | HYPER  PARAMETER | RBF (NON LINEAR)  (r 2value) | LINEAR  (r2 value) | POLY  (r 2value) | SIGMOID  (r 2value) |
| 1. | NIL(without standardization) | -314 | -10.7 | -147 | -280 |
| 2. | NIL(with standardization) | -500 | -328 | -163 | -121 |
| 3. | C10 | -481 | -1.5 | -149 | -108 |
| 4. | C100 | -4.4 | 0.03 | -0.34 | -0.79 |
| 5. | C500 | 0.124 | 0.71 | 0.73 | -0.26 |
| 6. | C1000 | 0.68 | 0.74 | 0.81 | -0.05 |
| 7. | C2000 | 0.79 | 0.74 | 0.833 | -0.04 |
| 8. | C3000 | 0.81 | 0.74 | 0.839 | -0.005 |

The SVM Regression use **r2 value** Poly (with standardization) and hyper parameter is (C=3000) = **0.839**

* ***DECISION TREE***: (r2 value),

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| SI.NO | CRITERION | MAX FEATURES | SPLITTER | With Random\_state=42  (r2 value) |
| 1. | () | () | () | 0.76 |
| 2. | Mse | None | Best | 0.768 |
| 3. | Mse | None | Random | 0.725 |
| 4. | Mse | Sqrt | Best | 0.725 |
| 5. | Mse | Sqrt | Random | 0.55 |
| 6. | Mse | Log2 | best | 0.69 |
| 7. | Mse | Log2 | random | 0.55 |
| 8. | Mae | None | Best | 0.78 |
| 9. | Mae | None | Random | 0.74 |
| 10. | Mae | Sqrt | Best | 0.68 |
| 11. | Mae | Sqrt | random | 0.64 |
| 12. | Mae | Log2 | Best | 0.68 |
| 13. | Mae | Log2 | Random | 0.64 |
| 14. | friedman\_mse | None | Best | 0.76 |
| 15. | friedman\_mse | None | Random | 0.727 |
| 16. | friedman\_mse | Log2 | Best | 0.69 |
| 17. | friedman\_mse | Log2 | random | 0.55 |
| 18. | friedman\_mse | Sqrt | Best | 0.69 |
| 19. | friedman\_mse | Sqrt | Random | 0.55 |

The Decision Tree Regression use **r2 value** (Mean absolute\_value\_None\_best) with (random\_state as 42) = **0.78**

* ***RANDOM FOREST***: (r2 value),

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| SI.NO | CRITERION | MAX FEATURES | N-ESTIMATORS | r2 value |
| 1. | MSE | None | 50 | 0.835 |
| 2. | MSE | None | 100 | 0.838 |
| 3. | MSE | Sqrt | 50 | 0.834 |
| 4. | MSE | Sqrt | 100 | 0.831 |
| 5. | MSE | Log2 | 50 | 0.834 |
| 6 | MSE | Log2 | 100 | 0.831 |
| 7. | MAE | None | 50 | 0.832 |
| 8. | MAE | None | 100 | 0.836 |
| 9. | MAE | Sqrt | 50 | 0.825 |
| 10. | MAE | Sqrt | 100 | 0.826 |
| 11. | MAE | Log2 | 50 | 0.825 |
| 12. | MAE | Log2 | 100 | 0.826 |

The Random Forest Regression **r2 value** (Mse\_None\_100) = **0.838**

1. **Final model justification:**

The final model of the Insurance charges predicting project is Machine Learning as supervised learning by regression method and Support vector Machine with hyper-tuning by “poly” and C=3000, based on the selection of highest r2\_score that is **0.839**.

***Import pickle***

***loaded\_model=pickle.load(open('finalized\_assignment\_svmcharges.sav','rb'))***

***result=loaded\_model.predict([[]])***

***result***

**Github:** [***https://github.com/DrAjitha1AI/Regression-Assignment***](https://github.com/DrAjitha1AI/Regression-Assignment)

AdaBoost Algorithm

|  |  |  |  |
| --- | --- | --- | --- |
| SI.NO | n\_estimators | loss | r2 value |
| 1. | 50 | ***linear*** | 0.82 |
| 2. | 100 | ***linear*** | 0.82 |
| 3. | 50 | ***square*** | 0.24 |
| 4. | 100 | ***square*** | 0.18 |
| 5. | 50 | ***exponential*** | 0.42 |
| 6. | 100 | ***exponential*** | 0.28 |

AdaBoost = 0.82

XG boosting = 0.837

LG boosting = 0.846