**Machine Learning Regression Assignment**

Dataset: insurance\_pre.csv

**Problem Statement or Requirement:**

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

As a data scientist, you must develop a model which will predict the insurance charges.

1.) Identify your problem statement

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

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

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.)

6.) Mention your final model, justify why u have chosen the same.

**Problem Statement:**

* A client’s requirement is, he wants to predict the insurance charges based on the several parameters. The Client has provided the dataset of the same.
* As a data scientist, you must develop a model which will predict the insurance charges.

**Requirement:**

* To predict the insurance charges from given dataset insurance\_pre.csv based on the input and output column variables.

**Problem Identification:**

**The 3 stages of identifying the problem:**

**Stage 1 – Domain Selection**

* Here the dataset contains numbers. So we for the domain **Machine Learning.**

**Stage 2 – Learning Selection**

* In this dataset, both inputs and output variables are present and the requirement is clear. So that it comes under **Supervised** **Learning**.

**Stage 3 – Classification or Regression**

* The Prediction Value “charges” would be a numerical data. So that, it comes under **Regression**.

**Machine Learning**

**Supervised Learning**

**Regression**

**Basic information about the dataset:**

|  |  |
| --- | --- |
| Dataset Name | **insurance\_pre.csv** |
| Total Number of Rows | **1339** |
| Total Number of Columns | **6** |
| Input / Independent Variables | **age**  **sex**  **bmi**  **children**  **smoker** |
| Output / Dependent Variable | **charges** |

**Preprocessing Method:**

* Here we have two columns ‘sex’ and ‘smoker’ is containing string values. So that, we convert the string value into numerical value by using “**One Hot Encoding Method**”.

**MODELS**

**Regression – r2\_score Value**

To create and select best model for the following dataset using regression method by finding r2\_score value.

**Dataset**: 50\_Startups.csv

**1. Multiple Linear Regression**

r2\_score Value : 0.78913454847886

**2. Support Vector Machine**

**Reference :** **https://scikit-learn.org/1.5/modules/generated/sklearn.svm.SVR.html**

|  |  |  |  |
| --- | --- | --- | --- |
| **S.No** | **Hyper Parameter** | | **r2\_score Value** |
| **Kernel** | **C** |
| 1 | rbf (default) | C = 1.0 | -0.08340516096481387 |
| 2 | rbf (default) | C = 10.0 | -0.032380600328872244 |
| 3 | rbf (default) | C = 100.0 | 0.31966454505291175 |
| 4 | rbf (default) | C = 1000.0 | 0.8107195705218606 |
| 5 | linear | C = 1.0 | -0.010195463359872203 |
| 6 | linear | C = 10.0 | 0.4624263375382691 |
| 7 | linear | C = 100.0 | 0.6289632029980753 |
| 8 | linear | C = 1000.0 | 0.764839476863155 |
| 9 | poly | C = 1.0 | -0.07571733771584865 |
| 10 | poly | C = 10.0 | 0.03862518745153265 |
| 11 | poly | C = 100.0 | 0.6164698351109898 |
| 12 | poly | C = 1000.0 | 0.8546515591431771 |
| 13 | sigmoid | C = 1.0 | -0.07544638548539218 |
| 14 | sigmoid | C = 10.0 | 0.03944012147190745 |
| 15 | sigmoid | C = 100.0 | 0.5268415404135779 |

**3. Decision Tree**

**Reference :**

[**https://scikit-learn.org/dev/modules/generated/sklearn.tree.DecisionTreeRegressor.html**](https://scikit-learn.org/dev/modules/generated/sklearn.tree.DecisionTreeRegressor.html)

**Default** (Without Parameters):0.6846282594725315

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **S.No** | **criterion** | **splitter** | **max\_features** | **r2\_score Value** |
| 1 | squared\_error | best | None | 0.69195879050275 |
| 2 | squared\_error | best | sqrt | 0.719292156518875 |
| 3 | squared\_error | best | log2 | 0.7092130939767978 |
| 4 | squared\_error | random | None | 0.7649414113413403 |
| 5 | squared\_error | random | sqrt | 0.7377763364085246 |
| 6 | squared\_error | random | log2 | 0.6637193505936877 |
| 7 | friedman\_mse | best | None | 0.6935878088583727 |
| 8 | friedman\_mse | best | sqrt | 0.5380034348199008 |
| 9 | friedman\_mse | best | log2 | 0.6146538167577326 |
| 10 | friedman\_mse | random | None | 0.7543256379317644 |
| 11 | friedman\_mse | random | sqrt | 0.7337325328115014 |
| 12 | friedman\_mse | random | log2 | 0.5744924313991377 |
| 13 | absolute\_error | best | None | 0.7331724170845562 |
| 14 | absolute\_error | best | sqrt | 0.720981911774397 |
| 15 | absolute\_error | best | log2 | 0.6888844226446749 |
| 16 | absolute\_error | random | None | 0.7056160688438438 |
| 17 | absolute\_error | random | sqrt | 0.7011523781712932 |
| 18 | absolute\_error | random | log2 | 0.7085419230429022 |
| 19 | poisson | best | None | 0.6841271720441577 |
| 20 | poisson | best | sqrt | 0.7599216058242008 |
| 21 | poisson | best | log2 | 0.7152874588664129 |
| 22 | poisson | random | None | 0.7345984825042098 |
| 23 | poisson | random | sqrt | 0.6831015946647707 |
| 24 | poisson | random | log2 | 0.7152874588664129 |

**4.Random Forest**

|  |  |  |
| --- | --- | --- |
| n\_estimators | random\_state | **r2\_score Value** |
| 100 | 0 | 0.8520141592957491 |
| 50 | 0 | 0.8515423872655268 |
| 10 | 0 | 0.8247283492665464 |

* **n\_estimators**=100 and **random\_state**=0 gives best r2\_score value. Furthermore, we kept changing the hyper tuning parameters to get best results.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **S.No** | n\_estimators | **criterion** | **max\_features** | **r2\_score Value** |
| 1 | 100 | squared\_error | None | 0.8520141592957491 |
| 2 | 100 | squared\_error | 1 | 0.8520141592957491 |
| 3 | 100 | squared\_error | sqrt | 0.8674254724623967 |
| 4 | 100 | squared\_error | log2 | 0.8674254724623967 |
| 5 | 100 | friedman\_mse | None | 0.8520141592957491 |
| 6 | 100 | friedman\_mse | 1 | 0.8520141592957491 |
| 7 | 100 | friedman\_mse | sqrt | 0.8674254724623967 |
| 8 | 100 | friedman\_mse | log2 | 0.8674254724623967 |
| 9 | 100 | absolute\_error | None | 0.8554677996406439 |
| 10 | 100 | absolute\_error | 1 | 0.8512114616806963 |
| 11 | 100 | absolute\_error | sqrt | 0.8673380028099178 |
| 12 | 100 | absolute\_error | log2 | 0.8673380028099178 |
| 13 | 100 | poisson | None | 0.8507131558335165 |
| 14 | 100 | poisson | 1 | 0.846842375826689 |
| 15 | 100 | poisson | sqrt | 0.8606756785132448 |
| 16 | 100 | poisson | log2 | 0.8606756785132448 |

**Conclusion:**

Finally, Random Forest Algorithm is used to create a model to predict the insurance charges**.**

Based on the analysis, best r2\_score value would be **0.8674254724623967**.