# **MACHINE LEARNING-DECISION TREE**

**Statement: [REQUIREMENT]**

To predict the insurance Charges based on the various parameters factors (US)

**Input:**

Age, Sex, BMI, Children – NOMINAL, Smoker – NOMINAL

1338 ROWS & 6 COLUMNS

***("insurance\_pre.csv")***

**Output:**

Insurance Charges details

**Profit Predicted with percentage – R2 value**

**Analysis:**

**Machine Learning/Supervised Learning/Regression (Categorical – Nominal (Hot E-coding)**

* Machine Learning – Regression (R2 Value) = 78% (0-1)
* Support Vector Machine – Standardisation (R2 Value) = 87%
  + Kernel – precomputed (N/A for this file -> rows and columns s/b equal)
  + ALGORITHM:
    - E.g.

regressor=SVR(kernel="linear",gamma="auto",C=1)

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| ***BEFORE STANDARDISATION - SUPPORT VECTOR MACHINE*** | | | | | |
| **Hyper Tuning Parameter** | **Playable Parameter** | **Kernal='linear'** | **Kernal='rbf'** | **Kernal='poly'** | **Kernal= 'sigmoid'** |
| C=1 |  | -0.01 | -0.08 | -0.07 | -0.01 |
| C=10 |  | 0.46 | -0.03 | 0.03 | 0.03 |
| C=100 |  | 0.62 | 0.32 | 0.61 | 0.52 |
| C=1000 |  | 0.76 | 0.81 | 0.85 | 0.28 |
| C=10000 |  | 0.74 | **0.87** | 0.85 | -34.15 |
| C=1 | gamma = 'auto' | -0.01 | -0.08 | -0.07 | -0.01 |
| C=10 | gamma = 'auto' | 0.46 | -0.03 | 0.03 | 0.03 |
| C=100 | gamma = 'auto' | 0.62 | 0.32 | 0.61 | 0.52 |
| C=1000 | gamma = 'auto' | 0.76 | 0.81 | 0.85 | 0.28 |
| C=10000 | gamma = 'auto' | 0.74 | **0.87** | 0.85 | -34.15 |
| C=1 | gamma = 'scale' | -0.01 | -0.08 | -0.07 | -0.01 |
| C=10 | gamma = 'scale' | 0.46 | -0.03 | 0.03 | 0.03 |
| C=100 | gamma = 'scale' | 0.62 | 0.32 | 0.61 | 0.52 |
| C=1000 | gamma = 'scale' | 0.76 | 0.81 | 0.85 | 0.28 |
| C=10000 | gamma = 'scale' | 0.74 | **0.87** | 0.85 | -34.15 |

* Decision Tree - As per the given Data file
  + ***insurance\_pre.csv***
  + ALGORITHM:

Eg:

criterion='absolute\_error', splitter='best', max\_features='sqrt'

|  |  |  |  |
| --- | --- | --- | --- |
| ***AFTER STANDARDISATION (Non-Linear)*** | | | |
| ***CRETERION*** | ***SPLITTER*** | ***max\_features*** | ***r\_score(%)*** |
| **friedman\_mse** |  |  | 0.69 |
| **squared\_error(Default)** |  |  | 0.69 |
| **absolute\_error** |  |  | 0.67 |
| **poisson** |  |  | 0.71 |
| **friedman\_mse** | random |  | 0.70 |
| **squared\_error(Default)** | random |  | 0.73 |
| **absolute\_error** | **random** |  | **0.74** |
| **poisson** | random |  | 0.72 |
| **friedman\_mse** | Best |  | 0.69 |
| **squared\_error(Default)** | Best |  | 0.68 |
| **absolute\_error** | **Best** |  | 0.66 |
| **poisson** | Best |  | 0.72 |
| **friedman\_mse** | random | log2 | 0.53 |
| **squared\_error(Default)** | random | log2 | 0.69 |
| absolute\_error | random | log2 | 0.73 |
| **poisson** | random | log2 | 0.61 |
| **friedman\_mse** | random | Sqrt | 0.73 |
| **squared\_error(Default)** | random | Sqrt | 0.63 |
| **absolute\_error** | random | Sqrt | 0.68 |
| **poisson** | random | Sqrt | 0.65 |
| **friedman\_mse** | Best | log2 | 0.67 |
| **squared\_error(Default)** | Best | log2 | 0.66 |
| **absolute\_error** | Best | log2 | 0.68 |
| **poisson** | Best | log2 | 0.70 |
| **friedman\_mse** | Best | Sqrt | 0.72 |
| **squared\_error(Default)** | Best | Sqrt | 0.65 |
| **absolute\_error** | Best | Sqrt | 0.60 |
| **poisson** | Best | Sqrt | 0.65 |

**4. Decision Tree**: Using DT the predicted R Value is 0.74 (Criterion = ‘absolute\_error’, Splitter = ‘random’)

**5. Random Forest**: Using RF the predicted R Value is 0.87

Example:

regressor=RandomForestRegressor(n\_estimators = 50, criterion = 'squared\_error', max\_features='log2', random\_state = 0)

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| ***STANDARDISATION*** | | | | | |
| **criterion** | **n\_estimators** | **max\_features** | **random\_state** | **R2 Value** |
| squared\_error(Default) | 10 | sqrt | 0 | 0.85 |
| absolute\_error | 10 | sqrt | 0 | 0.85 |
| poisson | 10 | sqrt | 0 | 0.85 |
| friedman\_mse | 10 | sqrt | 0 | 0.85 |
| squared\_error(Default) | 50 | sqrt | 0 | 0.86 |
| absolute\_error | 50 | sqrt | 0 | 0.87 |
| poisson | 50 | sqrt | 0 | 0.86 |
| friedman\_mse | 50 | sqrt | 0 | 0.86 |
| squared\_error(Default) | 100 | sqrt | 0 | 0.87 |
| absolute\_error | 100 | sqrt | 0 | 0.87 |
| poisson | 100 | sqrt | 0 | 0.86 |
| friedman\_mse | 100 | sqrt | 0 | 0.87 |
| squared\_error(Default) | 10 | log2 | 0 | 0.85 |
| absolute\_error | 10 | log2 | 0 | 0.85 |
| poisson | 10 | log2 | 0 | 0.85 |
| friedman\_mse | 10 | log2 | 0 | 0.85 |
| squared\_error(Default) | 50 | log2 | 0 | 0.86 |
| absolute\_error | 50 | log2 | 0 | 0.87 |
| poisson | 50 | log2 | 0 | 0.86 |
| friedman\_mse | 50 | log2 | 0 | 0.87 |
| squared\_error(Default) | 100 | log2 | 0 | 0.87 |
| absolute\_error | 100 | log2 | 0 | 0.87 |
| poisson | 100 | log2 | 0 | 0.86 |
| friedman\_mse | 100 | log2 | 0 | 0.87 |
| squared\_error(Default) | 10 |  |  | 0.83 |
| absolute\_error | 10 |  |  | 0.84 |
| poisson | 10 |  |  | 0.82 |
| friedman\_mse | 10 |  |  | 0.84 |
| squared\_error(Default) | 50 |  |  | 0.85 |
| absolute\_error | 50 |  |  | 0.85 |
| poisson | 50 |  |  | 0.86 |
| friedman\_mse | 50 |  |  | 0.85 |
| squared\_error(Default) | 100 |  |  | 0.85 |
| absolute\_error | 100 |  |  | 0.85 |
| poisson | 100 |  |  | 0.85 |
| friedman\_mse | 100 |  |  | 0.85 |

**Conclusion**

**Models R- Score values**

1. MLR = R Value is 0.78
2. SVM = R Value is 0.87
3. DT = R Value is 0.74
4. RT = R Value is 0.87

As per the conclusion of all parameters of three (LINEAR, SVM, Decision Tree). “Random Forest” and “SVM” gives higher R\_score (0.87) than that of the other models. Hence, I will save and deploy this model to the production environment