# Insurance Charges Prediction

## Problem Statement:

- **Machine Learning →** Here the input is in excel format, so we can consider the **Input as Numbers.** So in AI if input is Number we can give solution with Machine Learning.
- **Supervised →** In this problem we have clear idea about the requirement and the dataset, so we can continue with Supervised learning.
- **Regression →** Here the output is in Numeric type (Charges), so we can continue with regression.

**Machine Learning → Supervised Learning → Regression**

## Basic Information about dataset:

- It has 6 columns and 1338 rows.
- This dataset includes the clients personal details as input like Age, Sex, BMI, No. of children's, Smoker.
- We need to predict the output as Insurance Charges and the output details are given in the dataset.

## Pre-processing Method:

- In this model Pre-processing method is done on two columns which is Sex (Female/Male) to (0/1) by ordinal data and Smoker (Yes/No) to (0/1) ordinal data.

## Machine Learning Algorithms:

1. Simple Linear Regression
2. Multiple Linear Regression
3. Support Vector Machine
4. Decision Tree
5. Random Forest

1. **Simple Linear Regression**:

   In simple Linear Regression we should have only one Input and one output, but in client's dataset we have multiple inputs so this algorithm will not work for this dataset.

2. **Multiple Linear Regression:**

   ML regression use R_score value is **0.78913**

## 3. Support Vector Machine:

| Sl.no | C value | Linear(r_score value) | RBF (r_score value) | Poly(r_score value) | Sigmoid(r_score value) |
|---|---|---|---|---|---|
| 1 | C=0.1 | -0.1221 | -0.0895 | -0.0862 | -0.0897 |
| 2 | C=1.0 | -0.1115 | -0.0884 | -0.0645 | -0.0899 |
| 3 | C=10 | -0.0017 | -0.0818 | -0.0930 | -0.0909 |
| 4 | C=100 | 0.5432 | -0.1245 | -0.0992 | -0.1185 |
| 5 | C=500 | 0.6269 | -0.1246 | -0.0817 | -0.4735 |
| 6 | C=1000 | 0.6338 | -0.1176 | -0.0546 | -1.7112 |
| 7 | C=2000 | 0.6898 | -0.1078 | -0.0016 | -5.8190 |
| 8 | C=3000 | 0.7590 | -0.0962 | 0.0494 | -12.5445 |

From above list , SVM Regression use R_score value(Linear, C=3000 ) = **0.7590**

## 4. Decision Tree:

If parameters are not passed then r_score value is 0.6789

| **SL.NO** | **criterion** | **splitter** | **max_features** | **R_score** |
|---|---|---|---|---|
| 1 | squared_error | Best | None | 0.6854 |
| 2 | squared_error | Random | None | 0.7111 |
| 3 | squared_error | Random | Sqrt | 0.6945 |
| 4 | squared_error | Random | Log2 | 0.6307 |
| 5 | squared_error | Best | Log2 | 0.7202 |
| 6 | squared_error | Best | Sqrt | 0.5904 |
| 7 | friedman_mse | Best | Sqrt | 0.6263 |
| 8 | friedman_mse | Random | Sqrt | 0.6727 |
| 9 | friedman_mse | Random | Log2 | 0.6407 |
| 10 | friedman_mse | Best | Log2 | 0.7373 |
| 11 | friedman_mse | Best | None | 0.7010 |
| 12 | friedman_mse | Random | None | 0.6351 |
| 13 | absolute_error | Random | None | 0.6523 |
| 14 | absolute_error | Best | None | 0.7258 |
| 15 | absolute_error | Best | Log2 | 0.7635 |
| 16 | absolute_error | Random | Log2 | 0.6994 |
| 17 | absolute_error | Random | Sqrt | 0.6910 |
| 18 | absolute_error | Best | Sqrt | 0.7387 |
| 19 | Poisson | Best | Sqrt | 0.7297 |
| 20 | Poisson | Random | Sqrt | 0.7304 |
| 21 | Poisson | Random | Log2 | 0.7108 |
| 22 | Poisson | Best | Log2 | 0.7009 |
| 23 | Poisson | Best | None | 0.6816 |
| 24 | Poisson | Random | None | 0.7008 |

From the above list, DecisionTreeRegressor use R_score value (criterion="absolute_error", splitter="best", max_features="log2") = **0.7635**

## 5. Random Forest:

If parameters are not passed then r_score value 0.8454

| SL.NO | n_estimators | criterion | max_features | R_score |
|---|---|---|---|---|
| 1 | 100 | squared_error | None | 0.8535 |
| 2 | 50 | squared_error | None | 0.8512 |
| 3 | 100 | squared_error | Sqrt | 0.8661 |
| 4 | 100 | squared_error | Log2 | 0.8692 |
| 5 | 50 | squared_error | Log2 | 0.8645 |
| 6 | 50 | squared_error | Sqrt | 0.8652 |
| 7 | 100 | friedman_mse | Sqrt | 0.8654 |
| 8 | 50 | friedman_mse | Sqrt | 0.8652 |
| 9 | 100 | friedman_mse | Log2 | 0.8642 |
| 10 | 50 | friedman_mse | Log2 | 0.8623 |
| 11 | 100 | friedman_mse | None | 0.8484 |
| 12 | 50 | friedman_mse | None | 0.8480 |
| 13 | 100 | absolute_error | None | 0.8539 |
| 14 | 50 | absolute_error | None | 0.8558 |
| 15 | 100 | absolute_error | Log2 | 0.8679 |
| 16 | 50 | absolute_error | Log2 | 0.8688 |
| 17 | 100 | absolute_error | Sqrt | 0.8680 |
| 18 | 50 | absolute_error | Sqrt | 0.8659 |
| 19 | 100 | Poisson | Sqrt | 0.8637 |
| 20 | 50 | Poisson | Sqrt | 0.8675 |
| 21 | 100 | Poisson | Log2 | 0.8633 |
| 22 | 50 | Poisson | Log2 | 0.8651 |
| 23 | 100 | Poisson | None | 0.8508 |
| 24 | 50 | Poisson | None | 0.8492 |

From the above list “RandomForestRegressor” use R_score value (n_estimators=100, criterion=”squad_error”, max_features=”log2”) = **0.8692**

Also I have tried with (min_samples_split,bootstrap) parameters, R_score value was around 0.84 to 0.88 range only.

## Final Model of Regression in Machine Learning:

From all the algorithm we could see Random Forest Regression algorithm only as Max R_score value, So we choose Random Forest is the best model.

Random Forest R_score value (n_estimators=100, criterion="squad_error", max_features="log2") = **0.8692**