**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

**Dataset:**

Git Hub Link: <https://raw.githubusercontent.com/RamishaRaniK/dataset/main/insurance_pre.csv>

**Identified Problem Statement:**

To build a regression model that predicts the insurance charges incurred by individuals based on their health related parameters

**Basic Info about the Dataset:**

 Total number of rows: 1338

 Total number of columns: 6

**Pre-processing Methods Used**

Converted categorical 'smoker' and ‘Sex’ column to one-hot encoding

**Model Development & Evaluation (R² Score)**

1. **Multiple Linear Regression:** R² Score is **1.0**.
2. **SVM – Support Vector Machine**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **S.NO** | **HYPER PARAMETER** | **LINEAR**  **(r value)** | **RBF, NON- LINEAR**  **(r value)** | **POLY**  **(r value)** | **SIGMOID**  **(r value)** |
| 1 | C=1.0 | 0.10079181731024034 | 0.08084991977485756 | - 0.041360824146346786 | -0.06367999597761353 |
|  | C=10 | 0.8529518860715233 | 0.00826025752083348 | 0.33513589722806036 | 0.14863572894269383 |
| 2 | C=100 | 0.9999999999613921 | 0.4621572041789693 | 0.9089257539178448 | 0.7959944218698409 |
| 3 | C=500 | 0.9999999999799407 | 0.9063302906422341 | 0.9424490176810378 | 0.6042724777640163 |
| 4 | C=1000 | 0.9999999999818759 | 0.9632237066983724 | 0.9543274283782202 | -0.17269131026361317 |

The best model is **SVM with Poly kernel and C=1000**, achieving the highest R² score of **0.9543**,

1. **Decision Tree**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **S.NO** | **criterion** | **splitter** | **max\_features** | **R value** |
| 1 | squared\_error | best | sqrt | 0.9879912037940415 |
| 2 | squared\_error | random | sqrt | 0.9353586441847493 |
| 2 | friedman\_mse | random | sqrt | 0.9697736747515894 |
| 3 | friedman\_mse | Best | sqrt | 0.9793381190827237 |
| 4 | absolute\_error | Best | sqrt | 0.9323216259947916 |
| 5 | absolute\_error | random | Sqrt | 0.9644264011770957 |
| 6 | poisson | random | Sqrt | 0.9431570718778388 |
| 7 | poisson | Best | Sqrt | 0.9840006029861501 |
| 8 | squared\_error | best | log2 | 0.9789905701792334 |
| 9 | squared\_error | random | log2 | 0.9817373403644927 |
| 10 | friedman\_mse | random | log2 | 0.9314687564898771 |
| 11 | friedman\_mse | Best | log2 | 0.9821577409960018 |
| 12 | absolute\_error | Best | log2 | 0.9715442492701435 |
| 13 | absolute\_error | random | log2 | 0.9609801053643781 |
| 14 | poisson | random | log2 | 0.9356668499345531 |
| 15 | poisson | Best | log2 | 0.9245028387259718 |

The best model is **Decision Tree Regressor** with parameters criterion='squared\_error', splitter='best', and max\_features='sqrt', achieving the highest R² value of **0.98799**

1. **Random Forest**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **S.NO** | **criterion** | **n\_estimators** | **max\_features** | **R value** |
| 1 | squared\_error | 10 | sqrt | 0.9939881404889959 |
| 2 | squared\_error | 100 | sqrt | 0.9969413625560966 |
| 2 | friedman\_mse | 10 | sqrt | 0.9939881404889959 |
| 3 | friedman\_mse | 100 | sqrt | 0.9969413625560966 |
| 4 | absolute\_error | 10 | sqrt | 0.9933658511323451 |
| 5 | absolute\_error | 100 | Sqrt | 0.995843798617714 |
| 6 | poisson | 10 | Sqrt | 0.9927870711240302 |
| 7 | poisson | 100 | Sqrt | 0.9957182959834402 |
| 8 | squared\_error | 10 | log2 | 0.9939881404889959 |
| 9 | squared\_error | 100 | log2 | 0.9969413625560966 |
| 10 | friedman\_mse | 10 | log2 | 0.9939881404889959 |
| 11 | friedman\_mse | 100 | log2 | 0.9969413625560966 |
| 12 | absolute\_error | 10 | log2 | 0.9933658511323451 |
| 13 | absolute\_error | 100 | log2 | 0.995843798617714 |
| 14 | poisson | 10 | log2 | 0.9927870711240302 |
| 15 | poisson | 100 | log2 | 0.9957182959834402 |

The best model is **Random Forest Regressor** with parameters criterion='squared\_error',

n\_estimators=100,

and max\_features='sqrt' (or 'log2'),

achieving the highest R² value of **0.9969413625560966**.

**The Final Machine Learning Best Method of Regression**

**Multiple Linear Regression is selected as the best model** for predicting insurance charges due to its perfect performance (R² = 1.0)