📝 Insurance Charges Prediction using Regression



*Submitted by: Madhusudhanan  
Submitted to: Hope AI  
Date: June 27, 2025*

[**1. Problem Statement 3**](#_kidbnloxnadm)

[**2. Dataset Overview 3**](#_59o3jybxpw)

[3. Data Preprocessing 4](#_7a2jupphya7r)

[4. Model Building and Evaluation 4](#_mvtgewc6lt0w)

[5. Model Comparison (R² Scores) 8](#_3zl5xoh28cs6)

[**6. Final Model and Justification 9**](#_r1peoaq7nfdt)

[Why Random Forest (RT)? 9](#_dxdd852e2xcw)

[**7. Conclusion 9**](#_fdifd4m510zs)

[**8. Appendix 10**](#_pdxn8u3sd8ko)

# 1. Problem Statement

A health insurance company aims to estimate the medical insurance charges of its clients using demographic and health-related attributes. The company wants to build a predictive model that can be used for cost estimation.

This is a supervised regression problem where the goal is to predict a continuous numeric value.

# 2. Dataset Overview

Data Source:  
 [Insurance Dataset (GitHub CSV)](https://github.com/madhusudhanan-jayaram/2.Machine-Learning/tree/main/Regression%20Assignment/data/insurance_pre.csv)

Total Records: 1338

Total Features: 7

| **Column** | **Description** | **Type** |
| --- | --- | --- |
| Age | Age of the individual | Numeric |
| Sex | Gender (male, Female) | Categorical |
| BMI | Body Mass Index | Numeric |
| Children | Number of children covered by insurance | Numeric |
| Smoker | Smoking Status (Yes, No) | Categorical |
| Charges | Medical insurance charges (target) | Numeric |

## 

## 

# 3. Data Preprocessing

The following preprocessing steps were applied:

* Label Encoding was applied to categorical columns:  
  + sex, smoker
* Train-Test Split:  
  + 80% training set, 20% testing set  
    - Target Column:  
      * charges — continuous variable to be predicted

# 4. Model Building and Evaluation

In this section, we train and evaluate three regression models to predict insurance charges:

* Multiple Linear Regression
* Decision Tree Regressor
* Random Forest Regressor

Each model is evaluated using the R² Score to measure how well it fits the test data.

**1. 🔵 Multiple Linear Regression:**

R² Value: 0.7895

**2. 🟠 Support Vector Machine (SVM Regression)**

Type: Non-Linear Regression  
 Kernel Used: RBF (Radial Basis Function)

R² Value: -0.0834

**3. 🟢 Decision Tree Regression**

| **S.No** | **CRITERION** | **SPLITTER** | **MAX FEATURE** | **R VALUE** |
| --- | --- | --- | --- | --- |
| 1 | squared\_error | best | None | 0.6673 |
| 2 | squared\_error | best | Int - 5 | 0.6673 |
| 3 | squared\_error | best | Float -0.80 | 0.6545 |
| 4 | squared\_error | best | sqrt | 0.6513 |
| 5 | squared\_error | best | log2 | 0.6513 |
| 6 | squared\_error | random | None | 0.6510 |
| 7 | squared\_error | random | Int - 5 | 0.6510 |
| 8 | squared\_error | random | Float - 0.80 | 0.6616 |
| 9 | squared\_error | random | sqrt | 0.6009 |
| 10 | squared\_error | random | log2 | 0.6009 |
| 11 | friedman\_mse | best | None | 0.6572 |
| 12 | friedman\_mse | best | int | 0.6572 |
| 13 | friedman\_mse | best | float | 0.6539 |
| 14 | friedman\_mse | best | sqrt | 0.6519 |
| 15 | friedman\_mse | best | log2 | 0.6519 |
| 16 | friedman\_mse | random | None | 0.6419 |
| 17 | friedman\_mse | random | int | 0.6419 |
| 18 | friedman\_mse | random | float | 0.6599 |
| 19 | friedman\_mse | random | sqrt | 0.6108 |
| 20 | friedman\_mse | random | log2 | 0.6108 |
| 21 | absolute\_error | best | None | 0.6254 |
| 22 | absolute\_error | best | int | 0.6254 |
| 23 | absolute\_error | best | float | 0.6581 |
| 24 | absolute\_error | best | sqrt | 0.6260 |
| 25 | absolute\_error | best | log2 | 0.6260 |
| 26 | absolute\_error | random | None | 0.6689 |
| 27 | absolute\_error | random | Int - 5 | 0.6689 |
| 28 | absolute\_error | random | float | 0.6542 |
| 29 | absolute\_error | random | sqrt | 0.6291 |
| 30 | absolute\_error | random | log2 | 0.6291 |
| 31 | poisson | best | None | 0.6449 |
| 32 | poisson | best | int | 0.6449 |
| 33 | poisson | best | float | 0.6626 |
| 34 | poisson | best | sqrt | 0.6491 |
| 35 | poisson | best | log2 | 0.6491 |
| 36 | poisson | random | None | 0.6727 |
| 37 | poisson | random | int | 0.6727 |
| 38 | poisson | random | float | 0.6522 |
| 39 | poisson | random | sqrt | 0.5799 |
| 40 | poisson | random | log2 | 0.5799 |

# 

# 

# 

# 

# 

# 

# 

# 

# 

# 

**4. 🟢 Random Forest Regression**

| **S.No** | **MAX Depth** | **Max Features** | **Min sample leaf** | **Min sample split** | **N Estimators** | **R2n Score** |
| --- | --- | --- | --- | --- | --- | --- |
| 1 | 5 | None | 2 | 2 | 50 | 0.8380 |
| 2 | 5 | None | 2 | 5 | 50 | 0.8380 |
| 3 | 5 | None | 2 | 2 | 100 | 0.8379 |
| 4 | 5 | None | 2 | 5 | 100 | 0.8378 |
| 5 | 5 | None | 1 | 5 | 100 | 0.8326 |

# 

# 

# 

# 5. Model Comparison (R² Scores)

| **Model** | **R2 Score** |
| --- | --- |
| RT (Random Forest) | 0.8380 |
| DT (Decision Tree) | 0.6673 |
| MLM (Multiple Linear Regression | 0.7895 |
| SVM (Support Vector Machine) | -0.0834 |

📝 **Interpretation:**The Random Forest (RT) model achieved the highest R² score (0.8380), demonstrating the best performance in capturing the variance of the target variable among all models tested.

On the other hand, Support Vector Machine (SVM) performed poorly in this regression setting with a negative R² score, indicating it failed to fit the data effectively.

📌 **Note:** The **R² score** (coefficient of determination) reflects the proportion of variance in the dependent variable explained by the model. A higher score implies better predictive accuracy.

# 6. Final Model and Justification

**RT (Random Forest)**

Why Random Forest (RT)?

Random Forest achieved the **highest R² score** among all evaluated models, reaching **0.8380**, clearly outperforming other approaches such as Decision Tree, Multiple Linear Regression, and SVM.

Key reasons for selection:

✅ **High Predictive Accuracy:** Captures complex, non-linear relationships effectively.  
✅ **Robustness:** Reduces overfitting by combining multiple decision trees (ensemble averaging).  
✅ **Feature Importance:** Automatically ranks feature contributions, aiding model explainability.  
✅ **Ease of Use:** Performs well even with default parameters, reducing tuning overhead.  
✅ **Scalability & Stability:** Works efficiently with large datasets and provides consistent predictions.

# 7. Conclusion

Based on comprehensive model evaluation, **Random Forest (RT)** was selected as the **most suitable model** for predicting insurance charges. It delivered **accurate and consistent predictions**, demonstrating **strong generalization** capabilities and **minimal overfitting**. These qualities make Random Forest an **ideal candidate for integration** into real-world systems such as:

* 📊 **Pricing systems** for premium calculation
* 📈 **Analytical dashboards** for insurers and analysts
* 🛠️ **Risk assessment tools** for underwriting decisions

Its robust performance and interpretability ensure **reliable insights and better business decisions** in the insurance domain.

# 8. Appendix