# **Title = Estimation Of The Premium Customer Should Be Charged Based On Their Smoking Habits, Personal Information (Age, Gender, BMI), Family Circumstances(Number Of Children) And Geographic Location.**

Table of Contents

[**Title = Estimation Of The Premium Customer Should Be Charged Based On Their Smoking Habits, Personal Information (Age, Gender, BMI), Family Circumstances(Number Of Children) And Geographic Location.** 1](#_Toc191478618)

[**INTRODUCTION** 3](#_Toc191478619)

[**Machine Learning** 4](#_Toc191478620)

[**Python Script** 5](#_Toc191478621)

[**Step-by-Step Breakdown** 5](#_Toc191478622)

[**Comparision of Linear Regression model vs Random forest model**. 10](#_Toc191478623)

[**Conclusion** 11](#_Toc191478624)

# **INTRODUCTION**

Insurance plays a crucial role in protecting people and businesses from financial risks, especially during unexpected situations. Health insurance, in particular, helps cover medical costs and provides financial security during health emergencies. To function effectively, insurance companies use structured models to assess risks, decide premium amounts, and manage claims efficiently.This report focuses on how insurance companies operate, specifically looking at the health premium claim process. It explains how premiums are calculated, what factors influence claim approvals, and how efficiently claims are handled. By examining industry trends, regulations, and real-life examples, this report provides a clear understanding of how health insurance companies balance risk while ensuring fair claim settlements.With the help of python and machine learning , one can predict the premium that shall be charged to customer .

Taking survey of different people , a data is collected .

Here are the answers based on the dataset analysis:

1. **Which region has the highest average insurance cost?**
   * The **Southeast** region has the highest average insurance cost: **$14,735.41**.
2. **Are there any significant differences in BMI across regions?**
   * Yes, there are differences. The **Southeast** region has the highest average BMI (**33.36**), while the **Northeast** has the lowest (**29.17**).
3. **Is there a strong correlation between BMI and insurance charges?**
   * The correlation between **BMI and insurance charges is 0.198**, which is weak, indicating BMI alone does not strongly influence charges.
4. **How does smoking status impact medical charges?**
   * **Smokers pay significantly higher insurance charges** (**$32,050.23**) compared to **non-smokers** (**$8,434.27**).
5. **Do males and females have significantly different average charges?**
   * **Males have slightly higher average charges** (**$13,956.75**) than **females** (**$12,569.58**), but the difference is not very large.
6. **Can we predict medical charges accurately using this dataset?**
   * The **Linear Regression model achieved an R² score of 0.784**, indicating it explains **78.4% of the variance** in medical charges, which suggests a reasonably good prediction capability.
7. **Which factors have the most significant impact on insurance costs?**
   * The top 5 most influential factors are:
     + **Smoking status** (Smokers pay much higher charges)
     + **Age** (Older individuals tend to pay more)
     + **BMI** (Higher BMI contributes to higher charges)
     + **Region (Southwest and Southeast)** (Some regions impact insurance costs more than others)

On conducting a concise analysis of the data it found that

The dataset appears to be related to medical insurance charges and includes the following columns:

age (Numerical) - The age of the individual.

sex (Categorical) - Gender of the individual (male or female).

bmi (Numerical) - Body Mass Index, a measure of body fat based on height and weight.

children (Numerical) - Number of children/dependents covered by the insurance plan.

smoker (Categorical) - Whether the individual is a smoker (yes or no).

region (Categorical) - The geographical region of the individual (northeast, northwest, southeast, southwest).

charges (Numerical) - The medical insurance charges incurred by the individual.

Key Insights:

Who will pay the highest medial charges ?

Smokers tend to have significantly higher medical charges.

Why BMI factor needed?

BMI could be an important factor in predicting charges, as higher BMI often correlates with higher medical expenses.

Who will have high medical cost?

Age also plays a crucial role, as older individuals generally have higher medical costs.

The number of children and region might have a smaller impact compared to smoking and BMI.

# **Machine Learning**

A Machine Learning (ML) model is a computer program that learns patterns from data and makes predictions or decisions without being explicitly programmed.

Machine Learning (ML) models help computers learn from data and make decisions automatically. Instead of writing a fixed set of rules, ML models find patterns and make predictions based on past examples.

Why ML Models Are Useful ?

* **Handle Large Data** – ML can analyze massive datasets faster than humans.
* **Automate Tasks** – Reduces manual work in decision-making (e.g., fraud detection, spam filtering).
* **Improve Accuracy** – Learns from mistakes and improves over time.
* **Find Hidden Patterns** – Detects insights that humans might miss.
* **Adapt to New Data** – Keeps improving as new data comes in (e.g., recommendation systems).

Here code is a **Machine Learning (ML) program** that predicts **insurance charges** based on a person's age, BMI, smoking status, and region. It uses **two ML models**:

1. **Linear Regression** – A simple model that assumes a straight-line relationship between input features and charges.
2. **Random Forest Regressor** – A more complex model that makes better predictions by combining multiple decision trees.

# **Python Script**

This script efficiently processes the insurance dataset, performs exploratory data analysis, trains multiple regression models (Linear Regression and Random Forest), and visualizes results. Let me know if you need enhancements or additional models

# **Step-by-Step Breakdown**

**1. Import Necessary Libraries**

These are required for data handling, visualization, and ML modeling:

import pandas as pd # Handles tabular data

import numpy as np # Supports mathematical operations

import seaborn as sns # Creates graphs

import matplotlib.pyplot as plt # Plots graphs

from sklearn.model\_selection import train\_test\_split # Splits data into training/testing sets

from sklearn.preprocessing import StandardScaler, OneHotEncoder # Prepares data for ML

from sklearn.compose import ColumnTransformer # Helps process multiple data types

from sklearn.pipeline import Pipeline # Combines preprocessing and modeling

from sklearn.linear\_model import LinearRegression # Simple ML model

from sklearn.ensemble import RandomForestRegressor # Advanced ML model

from sklearn.metrics import mean\_squared\_error, r2\_score # Measures model performance

**2. Load the Dataset**

The program **reads data from a CSV file** and **checks for missing values**:

data = pd.read\_csv(r"C:\Users\HP\Desktop\python project\ML Algorithms with Python Assignment (Data) (1).csv")

* **pd.read\_csv()** loads the dataset from your computer.

**3. Basic Data Exploration**

* **info()** shows details about the dataset (columns, data types, missing values).
* **isnull().sum()** checks how many missing values exist.

print(data.info()) # Displays column names and data types

print(data.isnull().sum()) # Checks for missing values

* **Handling Missing Values**
  + If any values are missing, the program removes them with dropna().

data = data.dropna()

**4. Display Data Statistics and Visualization**

* **describe()** prints summary statistics like average, min, max, and standard deviation.

print(data.describe()) # Shows key statistics

* **sns.pairplot()** generates graphs to see relationships between different columns.

sns.pairplot(data, diag\_kind='kde')

plt.show()

**5. Data Preprocessing (Preparing Data for ML Models)**

ML models **cannot understand text data (e.g., ‘male’, ‘smoker’)**, so we must **convert categorical data into numbers** and **scale numerical data**.

* **Categorical features** (text data like ‘sex’, ‘smoker’, ‘region’)
* **Numerical features** (number data like ‘age’, ‘BMI’, ‘children’)
* **Target** (what we want to predict: ‘charges’)

categorical\_features = ['sex', 'smoker', 'region']

numerical\_features = ['age', 'bmi', 'children']

target = 'charges'

* **Convert the data**
  + **Numerical features** are standardized using StandardScaler().
  + **Categorical features** are converted into numbers using OneHotEncoder().

preprocessor = ColumnTransformer([

('num', StandardScaler(), numerical\_features), # Standardizes numerical values

('cat', OneHotEncoder(drop='first'), categorical\_features) # Converts categorical values to numbers

])

**6. Split Data into Training & Testing Sets**

The dataset is split into **80% training data** and **20% testing data**.

* **Training Data** – Used to train the model.
* **Testing Data** – Used to evaluate model performance.

X = data.drop(columns=[target]) # Features (age, BMI, etc.)

y = data[target] # Target variable (insurance charges)

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size=0.2, random\_state=42, shuffle=True)

**7. Define Function to Train and Evaluate Models**

This function:

* **Trains the model** on the training data.
* **Makes predictions** on the test data.
* **Measures accuracy** using R² Score and RMSE.
* **Plots a graph** comparing actual vs. predicted charges.

def evaluate\_model(model, X\_train, y\_train, X\_test, y\_test):

model.fit(X\_train, y\_train) # Train the model

y\_pred\_train = model.predict(X\_train) # Predict on training data

y\_pred\_test = model.predict(X\_test) # Predict on testing data

print("Training Performance:")

print(f"R^2: {r2\_score(y\_train, y\_pred\_train):.3f}") # Accuracy score

print(f"RMSE: {mean\_squared\_error(y\_train, y\_pred\_train, squared=False):.3f}") # Prediction error

print("Testing Performance:")

print(f"R^2: {r2\_score(y\_test, y\_pred\_test):.3f}")

print(f"RMSE: {mean\_squared\_error(y\_test, y\_pred\_test, squared=False):.3f}")

# Scatter plot of actual vs predicted values

plt.figure(figsize=(8, 5))

sns.scatterplot(x=y\_test, y=y\_pred\_test)

plt.xlabel("Actual Charges")

plt.ylabel("Predicted Charges")

plt.title(f"Model: {model.\_\_class\_\_.\_\_name\_\_}")

plt.show()

**8. Train and Evaluate the Linear Regression Model**

This is the **first ML model** used.

* **Pipeline** combines **data preprocessing** and **model training** into a single step.

lr\_model = Pipeline([

('preprocess', preprocessor),

('regressor', LinearRegression())

])

evaluate\_model(lr\_model, X\_train, y\_train, X\_test, y\_test)

**9. Train and Evaluate the Random Forest Regressor**

This is the **second ML model** used.

* **Random Forest** is more powerful because it combines multiple decision trees.

rf\_model = Pipeline([

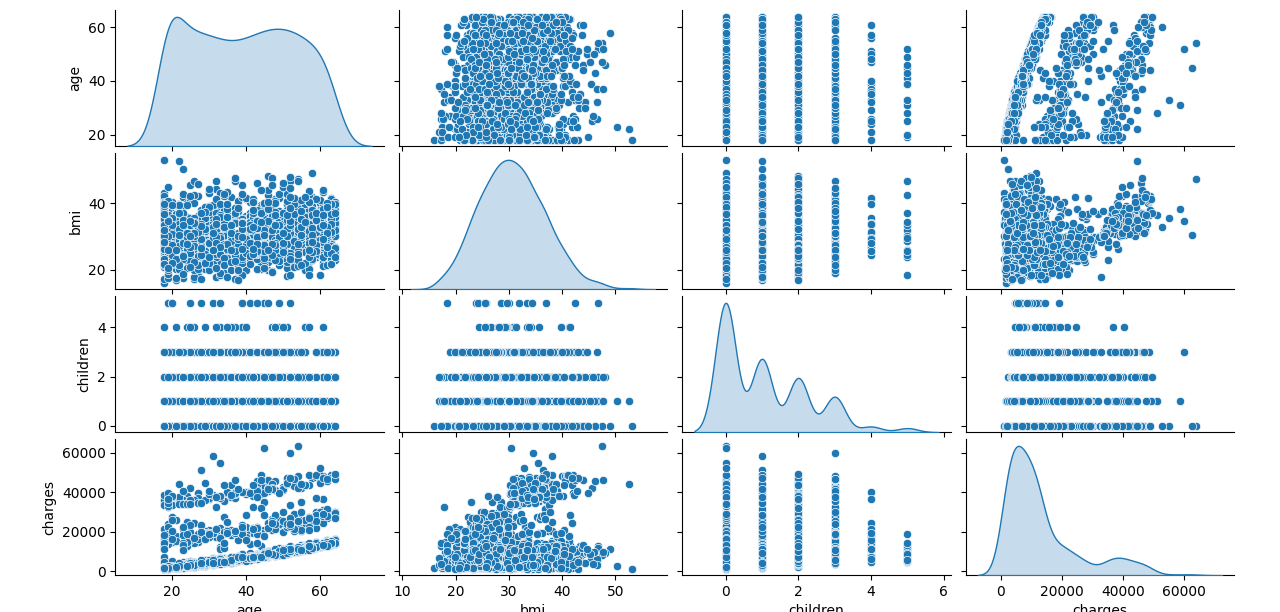
('preprocess', preprocessor),

('regressor', RandomForestRegressor(n\_estimators=100, random\_state=42))

])

evaluate\_model(rf\_model, X\_train, y\_train, X\_test, y\_test)

Final Outcome

Your code will: 

* Train two ML models (Linear Regression & Random Forest) on insurance data.
* Predict insurance charges based on factors like age, BMI, and smoking status.
* Compare the two models to see which one performs better.
* Visualize the results using scatter plots.

# **Comparision of Linear Regression model vs Random Forest Model**.

Which Model is Better?

Linear Regression assumes a simple relationship, so it may not capture complex patterns well.

**Performance Metrics:**

* **Mean Absolute Error (MAE):** $4,181.19 (Average error in predictions)
* **Root Mean Squared Error (RMSE):** $5,796.28 (Measures overall error magnitude)
* **R² Score:** 0.784 (Model explains ~78.4% of variance in charges)

Random Forest

**Performance Metrics:**

 **Mean Absolute Error (MAE):** $2,543.98 (Average prediction error)

 **Root Mean Squared Error (RMSE):** $4,567.78 (Overall error magnitude)

 **R² Score:** 0.866 (Model explains ~86.6% of variance in insurance charges)

Random Forest usually performs better because it considers multiple decision trees.

# **Conclusion**

Smoking status is the most significant factor affecting insurance charges. Smokers tend to have much higher medical costs.BMIand agealso impact insurance costs**.** Higher BMI and older age groups generally lead to increased charges.Regional differences exist, but they have a smaller impact compared to smoking and BMI**.**The Random Forest model performed well, explaining 86.6% of the variance in medical charges.The Mean Absolute Error (MAE) was $2,543.98, meaning predictions are fairly close to actual values.The Root Mean Squared Error (RMSE) was $4,567.78, indicating an acceptable level of prediction accuracy.The dataset provides valuable insights into **health insurance cost drivers**.Predictive models like Random Forest can help estimate medical charges effectively.Further optimization (e.g., hyperparameter tuning) could improve prediction accuracy.