Title Report on Estimation Of The Premium of Customer

Table of Contents

Introduction

Insurance plays a critical role in protecting people and businesses entities from financial risks, especially during unexpected event or 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 mainly focuses on how insurance companies operate, specifically looking at the health premium claim process. It explains how premiums are calculated, what factors influence the 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 .

After analysing the dataset one can answer the following questions-

Which region has the highest average insurance cost?

The Southeast region has the highest average insurance cost: **$14,735.41**.

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**).

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.

How does smoking status impact medical charges?

Smokers pay significantly higher insurance charges ($32,050.23) compared to **non-smokers** (**$8,434.27**).

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.

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.

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 1: Importing Libraries

python

```
import pandas as pd
import numpy as np

import seaborn as sns

import matplotlib.pyplot as plt

from sklearn.model_selection import train_test_split

from sklearn.preprocessing import StandardScaler, OneHotEncoder

from sklearn.compose import ColumnTransformer

from sklearn.pipeline import Pipeline

from sklearn.linear_model import LinearRegression

from sklearn.ensemble import RandomForestRegressor

from sklearn.metrics import mean_squared_error, r2_score
```

- pandas (pd): Handles data manipulation and loading.
- numpy (np): Provides numerical operations.
- seaborn (sns): Used for data visualization.
- matplotlib.pyplot (plt): Another visualization tool.
- sklearn.model_selection.train_test_split: Splits data into training and testing sets.
- sklearn.preprocessing.StandardScaler: Standardizes numerical features.
- sklearn.preprocessing.OneHotEncoder: Encodes categorical variables.
- sklearn.compose.ColumnTransformer: Applies transformations to different columns.
- sklearn.pipeline.Pipeline: Creates machine learning pipelines.
- sklearn.linear_model.LinearRegression: Implements Linear Regression.
- sklearn.ensemble.RandomForestRegressor: Implements Random Forest Regression.
- sklearn.metrics: Measures model performance.

Step 2: Loading Data

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

- **Loads the dataset from a CSV file into a Pandas DataFrame.**

Step 3: Exploring Data

python

```
print(data.info())

print(data.isnull().sum())
```

- **data.info(): Displays column names, data types, and non-null values.**
- **data.isnull().sum(): Checks for missing values in each column.**

Step 4: Handling Missing Data

python

```
data = data.dropna()
```

- **Drops rows with missing values (if any exist).**

Step 5: Statistical Summary

```
print(data.describe())
```

- **Displays summary statistics (mean, standard deviation, min, max, etc.).**

Step 6: Data Visualization

6.1 Histogram of Insurance Charges

python

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

sns.histplot(data['charges'], bins=30, kde=True)

plt.title("Distribution of Insurance Charges")

plt.xlabel("Charges")

plt.ylabel("Frequency")

plt.show()
```

- **Visualizes distribution of charges (target variable).**

6.2 Boxplot of Charges by Smoking Status

python

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

sns.boxplot(x='smoker', y='charges', data=data)

plt.title("Boxplot of Charges by Smoking Status")

plt.xlabel("Smoker (No = 0, Yes = 1)")

plt.ylabel("Charges")

plt.show()
```

- **Shows how smoking affects insurance charges.**

6.3 Correlation Heatmap

python

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

sns.heatmap(data.corr(), annot=True, cmap='coolwarm', fmt='.2f')

plt.title("Correlation Heatmap")

plt.show()
```

- **Displays correlation between numerical variables.**

6.4 Scatter Plot of BMI vs. Charges

python

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

sns.scatterplot(x='bmi', y='charges', hue='smoker', data=data, alpha=0.7)

plt.title("Scatter Plot of BMI vs. Charges")

plt.xlabel("BMI")

plt.ylabel("Charges")

plt.show()
```

- **Shows relationship between bmi and charges.**

6.5 Bar Plot of Average Charges by Region

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

sns.barplot(x='region', y='charges', data=data, estimator=np.mean)

plt.title("Average Insurance Charges by Region")

plt.xlabel("Region")

plt.ylabel("Average Charges")

plt.show()
```

- **Compares average charges across different regions.**

Step 7: Feature Engineering

7.1 Identifying Features

```
categorical_features = ['sex', 'smoker', 'region']

numerical_features = ['age', 'bmi', 'children']

target = 'charges'
```

- **Categorical features: sex, smoker, region**
- **Numerical features: age, bmi, children**

- **Target variable: charges**

7.2 Preprocessing Pipeline

python

```
preprocessor = ColumnTransformer([
   ('num', StandardScaler(), numerical_features),
   ('cat', OneHotEncoder(drop='first'), categorical_features)
])
```

- **StandardScaler(): Standardizes numerical columns.**
- **OneHotEncoder(drop='first'): Converts categorical columns into numerical values.**

Step 8: Splitting Data

```
X = data.drop(columns=[target])
y = data[target]
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42, shuffle=True)
```

- **Splits data into training (80%) and testing (20%) sets.**

Step 9: Model Evaluation Function

```
def evaluate_model(model, X_train, y_train, X_test, y_test):
   model.fit(X_train, y_train)
   y_pred_train = model.predict(X_train)
   y_pred_test = model.predict(X_test)

   print("Training Performance:")
```

```
    print(f"R^2: {r2_score(y_train, y_pred_train):.3f}")
    print(f"RMSE: {mean_squared_error(y_train, y_pred_train, squared=False):.3f}")


    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}")


    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()
```

- **Fits the model, predicts values, and evaluates performance using:**
  - **R² Score (higher is better)**
  - **Root Mean Squared Error (RMSE) (lower is better)**
  - **Scatter plot of actual vs. predicted values**

Step 10: Training Models

10.1 Linear Regression

```
lr_model = Pipeline([
    ('preprocess', preprocessor),
    ('regressor', LinearRegression())
])
```

```
evaluate_model(lr_model, X_train, y_train, X_test, y_test)
```

- **Creates a Linear Regression model with the preprocessing pipeline.**
- **Trains & evaluates the model.**

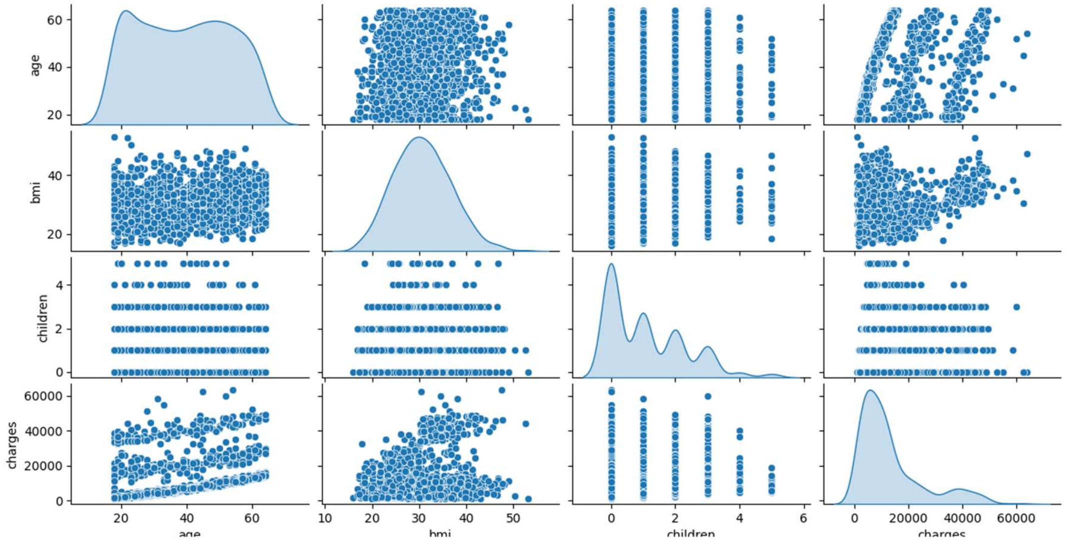
10.2 Random Forest Regressor

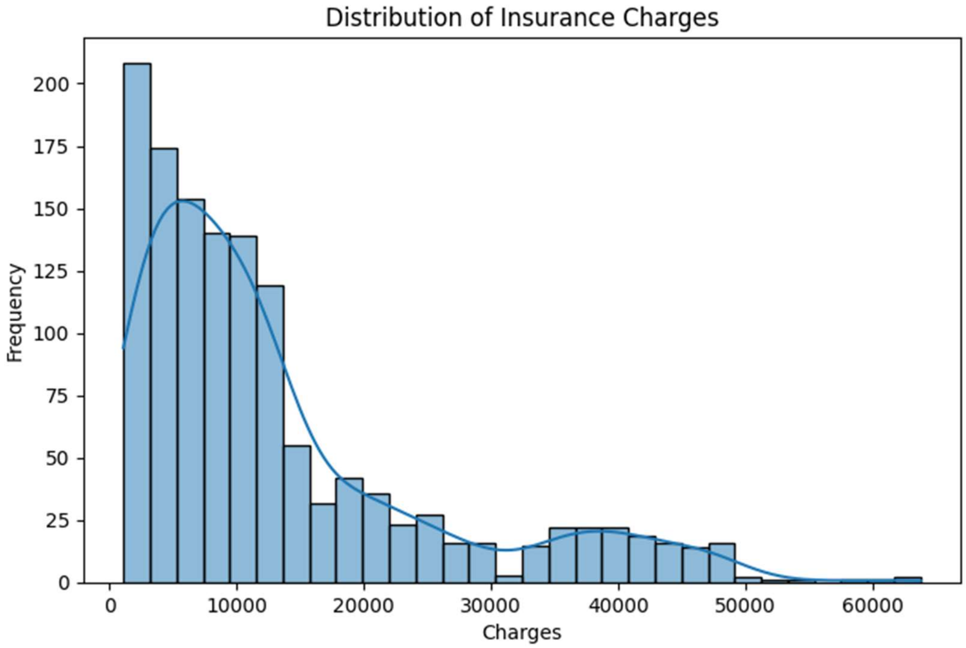
```
rf_model = Pipeline([
    ('preprocess', preprocessor),
    ('regressor', RandomForestRegressor(n_estimators=100, random_state=42))
])
evaluate_model(rf_model, X_train, y_train, X_test, y_test)
```

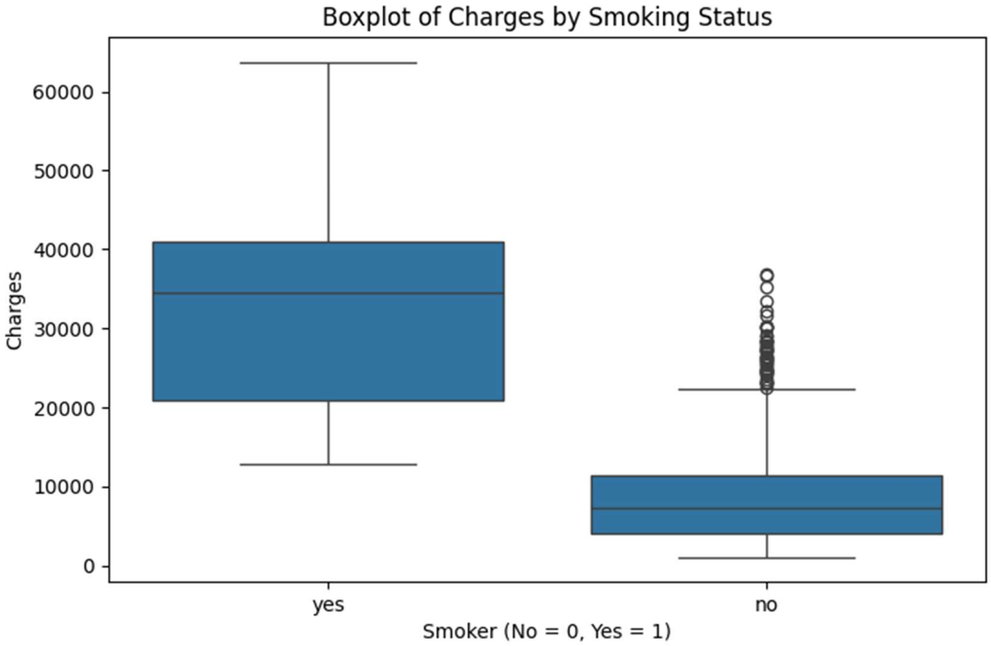
- **Uses Random Forest Regressor (ensemble learning technique).**
- **Evaluates its performance.**

Final Outcome

Your code will:







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.BMI and age also 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.