SUMMER TRAINING/INTERNSHIP

PROJECT REPORT

(Term June-July 2025)

MEDICAL INSURANCE COST PREDICTION

(TITLE OF THE PROJECT/ INTERNSHIP COMPANY*)*

Submitted by

Bablu Kumar

Registration Number :12313647

Course Code: PETV79

Under the Guidance of

Mahipal Singh Papola

(Name of mentor with designation)

School of Computer Science and Engineering

(June-July 2025)

LOVELY PROFESSIONAL UNIVERSITY, PUNJAB

BONAFIDE CERTIFICATE

This is to certify that the project report entitled **"BUILD A TOOL TO PREDICT MEDICAL INSURANCE COST"** is the Bonafide work of **BABLU KUMAR**, who has successfully carried out the project work under my supervision and guidance, in partial fulfillment of the requirements for the completion of the project.

SIGNATURE

<<Name of the Supervisor>>

SIGNATURE

SIGNATURE

HEAD OF THE DEPARTMENT

Acknowledgement

We would like to express our deepest gratitude to our respected project guide, **Mahipal Sir**, for his invaluable support, continuous guidance, and encouragement throughout the duration of this project. His insightful suggestions, constructive feedback, and patient supervision have played a vital role in shaping this work from the initial idea to its successful completion.

We also extend our sincere thanks to the faculty and staff of the **School of Computer Science and Engineering, Lovely Professional University (LPU)**, for providing a stimulating academic environment and access to the necessary resources and facilities.

We would like to acknowledge the constant motivation and moral support from our friends and classmates, who have always been ready to help and share valuable insights during challenging times.

Finally, we express our heartfelt appreciation to our families for their unwavering support, patience, and encouragement, which have been a constant source of strength throughout our academic journey.

CONTENTS OF THE REPORT

- Learnings

Chapter 6: Conclusion

- Summary

Chapter 1: Introduction

Company Profile

This project was completed as part of the academic curriculum at the School of Computer Science and Engineering, Lovely Professional University (LPU), Phagwara, Punjab. LPU is one of India’s leading private universities, known for its focus on industry-oriented education, innovation, and research.

The School of Computer Science and Engineering at LPU provides modern facilities and advanced resources that encourage students to work on real-world problems using the latest technologies. The department focuses on practical learning and hands-on projects in areas like data science, artificial intelligence, machine learning, and software development.

In this environment, students gain the skills necessary to build and deploy data-driven applications. By working on projects such as insurance cost prediction, students develop the ability to apply statistical and machine learning techniques to solve real-world challenges, including cost estimation and financial analysis. This project highlights the importance of using data to create accurate predictive models and supports LPU's mission of fostering innovation and practical knowledge.

Overview of Training Domain

The training focused on the field of **data science and machine learning**, which are two of the most in-demand areas in technology today. Data science involves analyzing large amounts of data to find patterns and extract useful insights. Machine learning is a part of data science where we create models that can learn from data and make predictions or decisions automatically. In this training, we worked on data preprocessing, data visualization, feature engineering, and building predictive models, which are all key steps in solving real-world business problems using data.

Objective of the Project

The main objective of the project was to build a machine learning model that can **predict medical insurance costs** for individuals based on their personal and health-related information. The goal was to understand which factors most affect insurance charges and to create a model that can accurately estimate these costs. This helps insurance companies decide how to price policies for different people and manage their risks better. This project also aimed to give hands-on experience in applying data science concepts from start to finish.

Chapter 2: Training Overview

Tools & Technologies Used

During the training, I used various tools and technologies to complete the project. These included:

- **Python**: Main programming language used for analysis and model building.
- **Jupyter Notebook & Google Colab**: Platforms to write and run code, visualize data, and document work.
- **Pandas and NumPy**: Libraries for data manipulation and numerical operations.
- **Matplotlib and Seaborn**: Libraries used for data visualization and creating plots.
- **Scikit-learn**: A popular machine learning library for building and evaluating models.
- **Git and GitHub**: Used for version control, to track changes in the code and collaborate if needed.

Areas Covered During Training

The training covered a wide range of topics and skills needed in data science. This included:

- Learning Python programming and data analysis libraries.
- Collecting and cleaning raw data to make it suitable for analysis.
- Visualizing data to understand trends and patterns.
- Encoding categorical (text) data into numerical format.
- Splitting data into training and test sets to build and validate models.
- Understanding different machine learning algorithms and how to evaluate them.
- Interpreting model results and communicating findings clearly.

Daily/Weekly Work Summary

In the initial weeks, I focused on understanding the basics of Python, Pandas, and data cleaning techniques. Then, I moved on to data visualization to explore the dataset in detail. The middle part of the training involved working on feature engineering and understanding how different variables affect the outcome. Towards the end, I focused on building the linear regression model, testing its accuracy. Weekly progress reviews helped in identifying challenges and planning the next steps effectively.

Chapter 3: Project Details

Title of the Project

Predicting Medical Insurance Costs using Linear Regression

Problem Definition

Health insurance costs vary for each individual based on their health conditions and lifestyle choices. It is difficult for insurance companies to accurately estimate these costs without using data. The problem addressed in this project is to create a predictive model that can estimate the insurance charges for a person using details like age, gender, BMI, number of children, smoking status, and region.

Scope and Objectives

The scope of this project includes working with a real-world insurance dataset, performing exploratory data analysis, preprocessing data, building a predictive model, and evaluating its accuracy. The main objectives are:

- To identify which factors most affect insurance charges.
- To develop a linear regression model to predict charges.
- To analyze the model performance and improve it if needed.
- To provide an easy-to-use system that can estimate insurance costs for new customers.

System Requirements

- **Operating System**: Windows, macOS, or Linux
- **Programming Language**: Python
- **Development Environment**: Jupyter Notebook or Google Colab
- **Libraries**: Pandas, NumPy, Matplotlib, Seaborn, Scikit-learn, xgboost
- **Hardware**: A basic computer with at least 4 GB RAM

Architecture Diagram

The workflow architecture is simple:

1. Data Collection → 2. Data Preprocessing → 3. Exploratory Data Analysis → 4. Feature Encoding → 5. Model Training → 6. Model Evaluation → 7. Prediction.

Chapter 4: Implementation

Tools Used

The main tools used during implementation included Python for writing code, Jupyter Notebook and Google Colab for interactive coding and visualizations, and various Python libraries (Pandas, NumPy, Matplotlib, Seaborn, xgboost and Scikit-learn) for analysis and machine learning tasks.

Methodology

The project started with loading the dataset and performing a detailed analysis to understand the structure and characteristics of the data. Categorical columns (like sex, smoker, region) were converted into numerical values using encoding. Visualizations were created to explore the relationships between features and insurance charges. The dataset was then split into training and testing sets to evaluate the model fairly. A linear regression model, a random forest regressor, and an XGBoost regressor were trained using the training data, and their accuracy was measured using R^2 score on both the training and testing datasets. Finally, predictions were made on new data samples to check their practical usability and compare the performance of all models.

Modules / Screenshots

Several modules were created: data loading and cleaning, data visualization, feature encoding, model training, and evaluation. Screenshots included plots of age distribution, BMI distribution, smoker vs non-smoker costs, and predicted vs actual charges.

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression
from sklearn import metrics
from sklearn.ensemble import RandomForestRegressor
from xgboost import XGBRegressor
```

✓ 0.0s

```
# loading the data from csv file to a Pandas DataFrame
insurance_dataset = pd.read_csv('C:/Users/ASUS/Desktop/ML Project/archive/insurance.csv')
```

✓ 0.0s

```
# first 5 rows of the dataframe
insurance_dataset.head()
```

✓ 0.0s

	age	sex	bmi	children	smoker	region	charges
0	19.0	female	27.900	0	yes	southwest	16884.92400
1	18.0	male	33.770	1	no	southeast	1725.55230
2	28.0	male	33.000	3	no	southeast	4449.46200
3	33.0	male	22.705	0	no	northwest	21984.47061
4	32.0	male	28.880	0	no	northwest	3866.85520

```
# checking for missing (values)(EDA)
insurance_dataset.isnull().sum()
```

```
age         2
sex         0
bmi         0
children    0
smoker      0
region      0
charges     0
dtype: int64
```

```
insurance_dataset['age'] = insurance_dataset['age'].fillna(insurance_dataset['age'].mean())
```

```
insurance_dataset.isnull().sum()
```

```
age         0
sex         0
bmi         0
children    0
smoker      0
region      0
charges     0
dtype: int64
```

```
#getting some informations about the dataset
insurance_dataset.info()
```

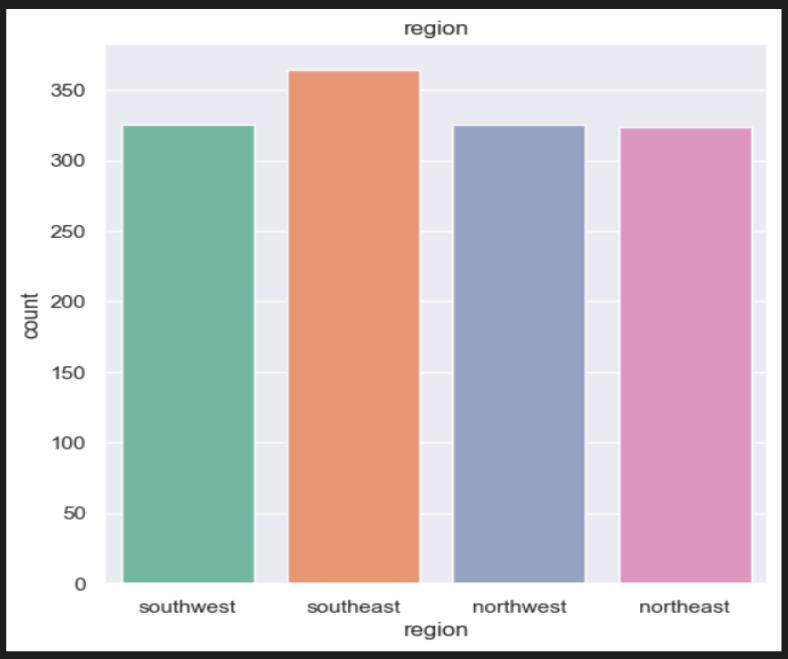
```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1338 entries, 0 to 1337
Data columns (total 7 columns):
 #   Column    Non-Null Count  Dtype  
---  ------    --------------  -----  
 0   age       1336 non-null   float64
 1   sex       1338 non-null   object 
 2   bmi       1338 non-null   float64
 3   children  1338 non-null   int64  
 4   smoker    1338 non-null   object 
 5   region    1338 non-null   object 
 6   charges   1338 non-null   float64
dtypes: float64(3), int64(1), object(3)
memory usage: 73.3+ KB
```

```
# statistical Measures of the dataset
insurance_dataset.describe()
```

	age	bmi	children	charges
count	1338.000000	1338.000000	1338.000000	1338.000000
mean	39.208832	30.663397	1.094918	13270.422265
std	14.049883	6.098187	1.205493	12110.011237
min	18.000000	15.960000	0.000000	1121.873900
25%	27.000000	26.296250	0.000000	4740.287150
50%	39.000000	30.400000	1.000000	9382.033000
75%	51.000000	34.693750	2.000000	16639.912515
max	64.000000	53.130000	5.000000	63770.428010

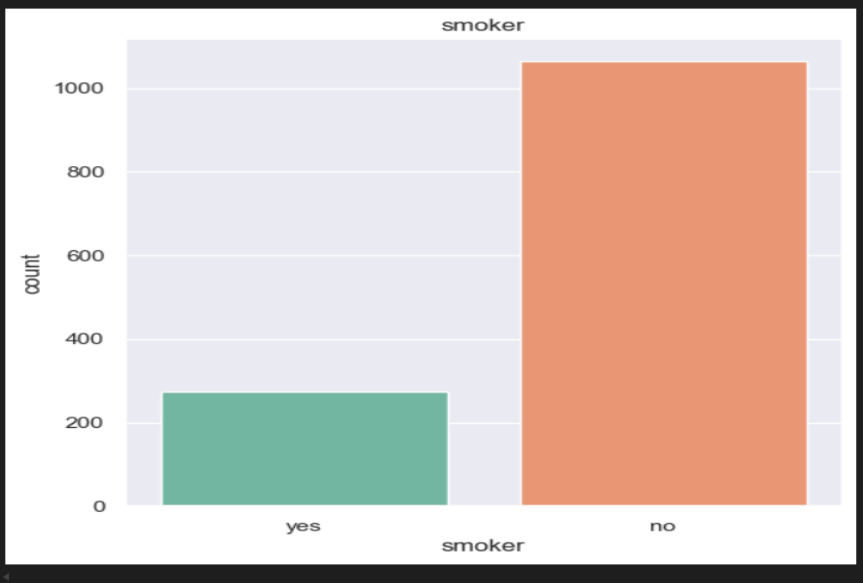
```
# region column
plt.figure(figsize=(6,6))
sns.countplot(x='region',hue='region' ,data=insurance_dataset,palette='Set2')
plt.title('region')
plt.show()
```

✓ 0.2s



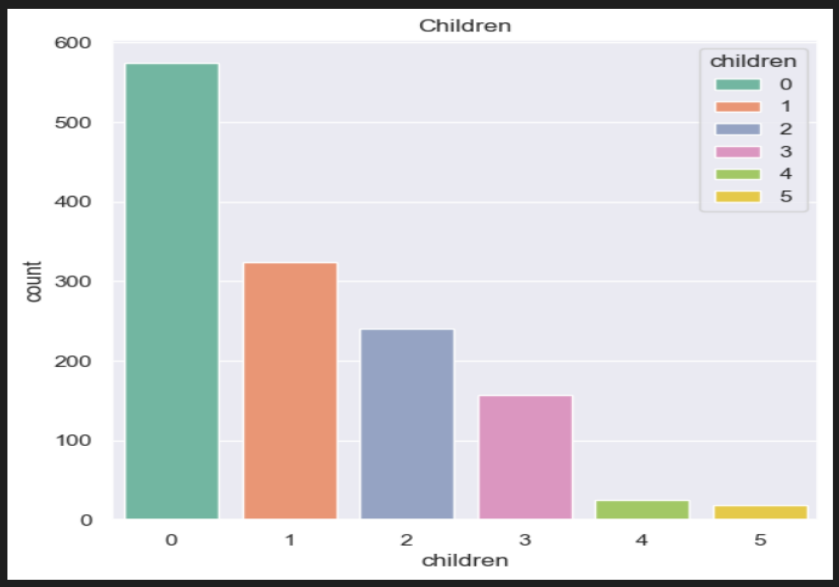

```
# smoker column
plt.figure(figsize=(6,6))
sns.countplot(x='smoker', hue='smoker',data=insurance_dataset,palette='Set2')
plt.title('smoker')
plt.show()
```

✓ 0.0s



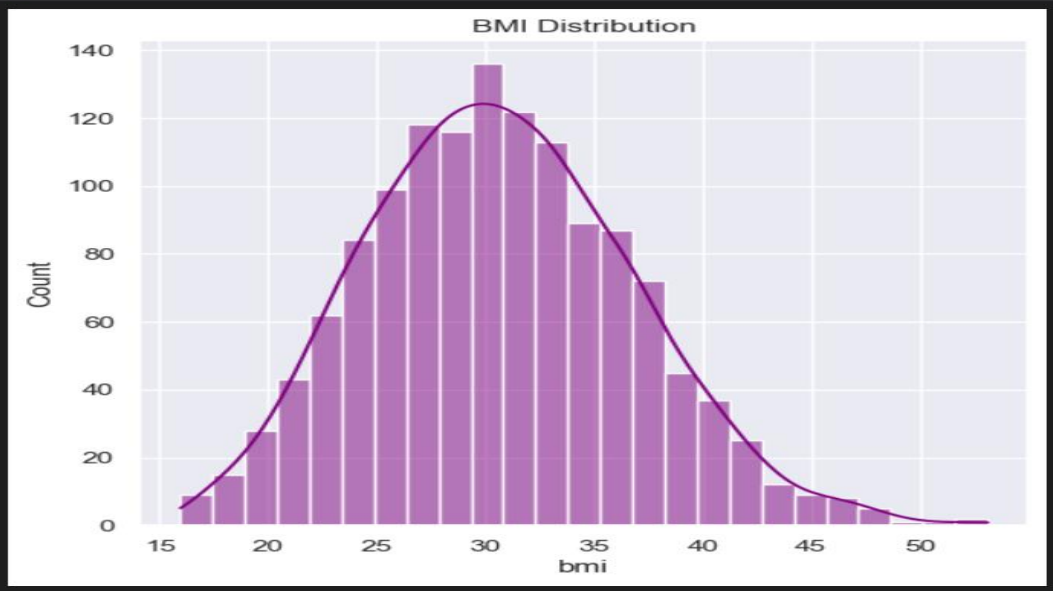
```
# children column
plt.figure(figsize=(6,6))
sns.countplot(x='children', data=insurance_dataset,palette='Set2',hue='children')
plt.title('Children')
plt.show()
```

0.1s



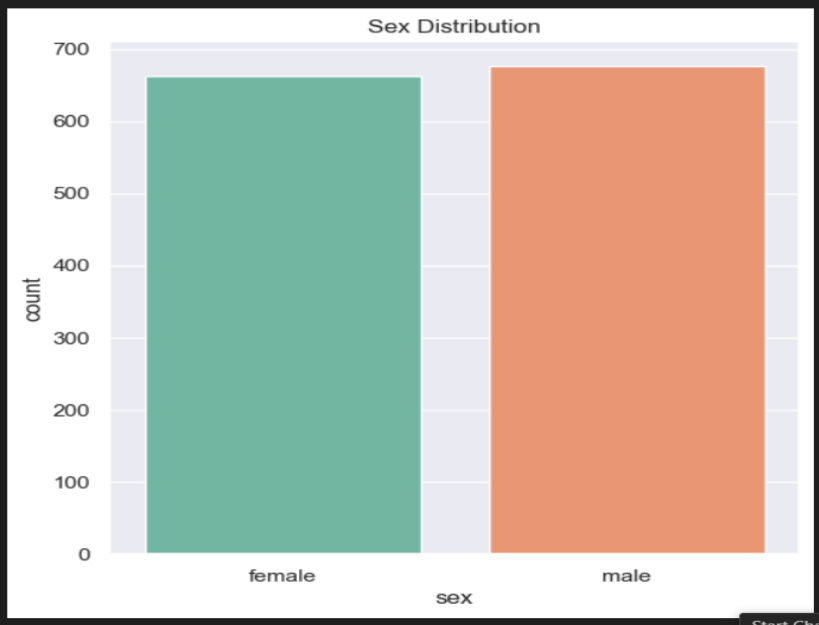

```
plt.figure(figsize=(6,6))
sns.histplot(insurance_dataset['bmi'], kde=True, color='purple')
plt.title('BMI Distribution')
plt.show()
```

0.1s



```
# Gender column
plt.figure(figsize=(6,6))
sns.countplot(x='sex', hue='sex',data=insurance_dataset, palette='Set2')
plt.title('Sex Distribution')
plt.show()
```

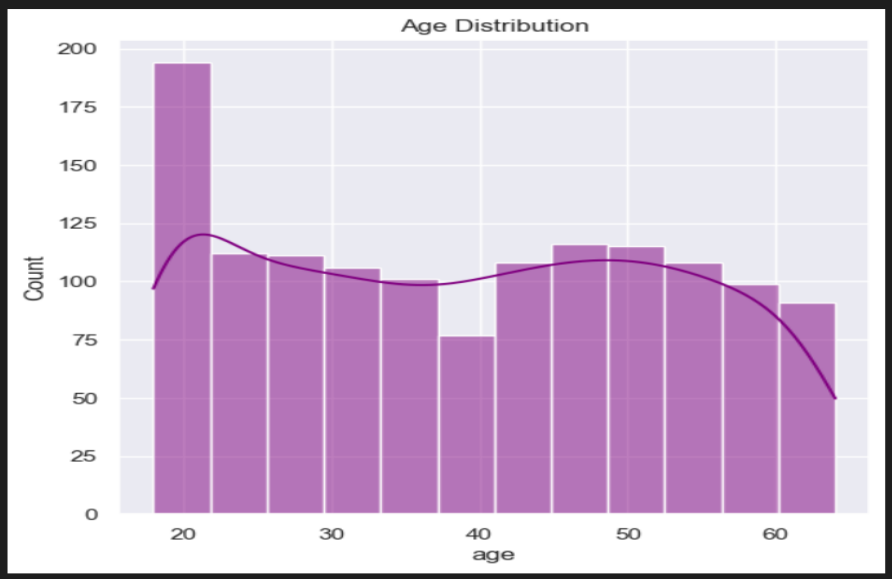
[11] ✓ 0.1s



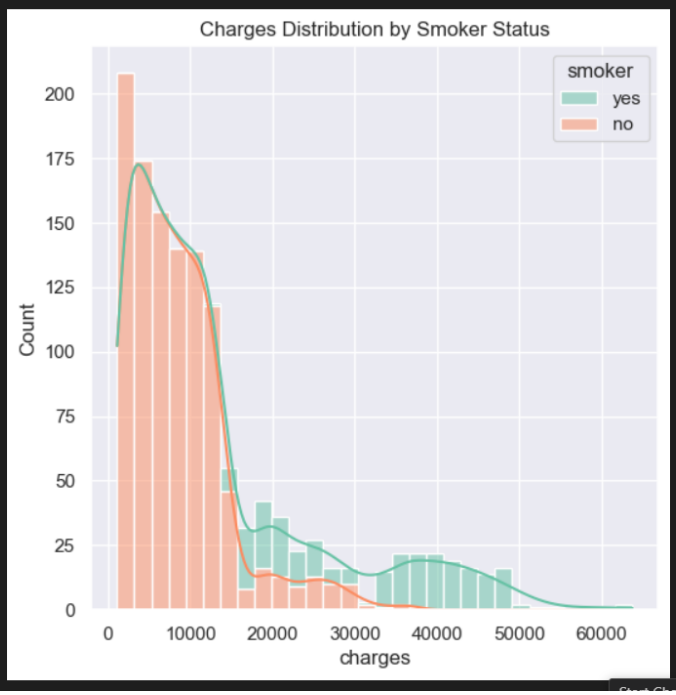
Start Chat to Generate Code (Ctrl+I)

```
# distribution of age value
sns.set()
plt.figure(figsize=(6,6))
sns.histplot(insurance_dataset['age'],kde=True,color='purple')
plt.title('Age Distribution')
plt.show()
```

35] ✓ 0.3s



```
# distribution of charges value
plt.figure(figsize=(6,6))
sns.histplot(data=insurance_dataset, x='charges', hue='smoker', multiple='stack', palette='Set2', kde=True)
plt.title('Charges Distribution by Smoker Status')
plt.show()
```



ENCODING

```
# Encoding 'sex' column
insurance_dataset['sex'] = insurance_dataset['sex'].map({'male': 0, 'female': 1})

# Encoding 'smoker' column
insurance_dataset['smoker'] = insurance_dataset['smoker'].map({'yes': 0, 'no': 1})

# Encoding 'region' column
insurance_dataset['region'] = insurance_dataset['region'].map({'southeast': 0, 'southwest': 1, 'northeast': 2, 'northwest': 3})
```

Model Training and Linear Regression

```
# loading the Linear Regression model
regressor = LinearRegression()
```

```
regressor.fit(X_train, Y_train)
```

```
LinearRegression
LinearRegression()
```

```
R squared vale :  0.7515184130348177
```

```
# prediction on test data
test_data_prediction =regressor.predict(X_test)
```

```
# R squared value
r2_test = metrics.r2_score(Y_test, test_data_prediction)
print('R squared vale : ', r2_test)
```

```
R squared vale :  0.7447322801972331
```

Building a Predictive Model

```
input_data = (31, 1, 25.74, 0, 1, 0)

# Create DataFrame
input_df = pd.DataFrame([input_data], columns=X.columns)

# Predict
prediction = regressor.predict(input_df)

print('The insurance cost is USD', prediction[0])
```

```
The insurance cost is USD 3759.4114626533583
```

Random Forest Regressor

```
# Random Forest Regressor
rf_model = RandomForestRegressor(n_estimators=100, random_state=2)
rf_model.fit(X_train, Y_train)

# Predictions
rf_train_pred = rf_model.predict(X_train)
rf_test_pred = rf_model.predict(X_test)

# Evaluation
r2_train_rf = metrics.r2_score(Y_train, rf_train_pred)
r2_test_rf = metrics.r2_score(Y_test, rf_test_pred)

print("Random Forest R^2 score on training data :", r2_train_rf)
print("Random Forest R^2 score on test data     :", r2_test_rf)
```

3] ✓ 0.6s

```
Random Forest R^2 score on training data : 0.9772876548911782
Random Forest R^2 score on test data     : 0.8370103831747011
```

XGBoost Regressor

```
#  XGBoost Regressor
xgb_model = XGBRegressor(n_estimators=100, learning_rate=0.05, random_state=2)
xgb_model.fit(X_train, Y_train)

# Predictions
xgb_train_pred = xgb_model.predict(X_train)
xgb_test_pred = xgb_model.predict(X_test)

# Evaluation
r2_train_xgb = metrics.r2_score(Y_train, xgb_train_pred)
r2_test_xgb = metrics.r2_score(Y_test, xgb_test_pred)

print("\nXGBoost R^2 score on training data :", r2_train_xgb)
print("XGBoost R^2 score on test data     :", r2_test_xgb)
```

36] ✓ 0.2s

```
XGBoost R^2 score on training data : 0.938771571159286
XGBoost R^2 score on test data     : 0.8547100490918331
```

prediction

```
# Original input data
input_data = (31, 1, 25.74, 0, 1, 0)

# Convert to DataFrame with proper column names
input_df = pd.DataFrame([input_data], columns=X.columns)

# Predictions
rf_pred = rf_model.predict(input_df)
xgb_pred = xgb_model.predict(input_df)

print("\nPredicted insurance cost (Random Forest): USD", rf_pred[0])
print("Predicted insurance cost (XGBoost): USD", xgb_pred[0])
```

[140] ✓ 0.0s

```
Predicted insurance cost (Random Forest): USD 3943.4576939000053
Predicted insurance cost (XGBoost): USD 3978.2078
```

Code Snippets

Here is an example of model training and prediction:

python

```
regressor = LinearRegression()

regressor.fit(X_train, Y_train)

prediction = regressor.predict(X_test)
```

This shows how the linear regression model is trained and used to make predictions.

Chapter 5: Results and Discussion

Output / Report

In this project, three different models were built and evaluated: a Linear Regression model, a Random Forest Regressor, and an XGBoost Regressor.

The Linear Regression model achieved an R^2 score of around 0.75 on the training data and about 0.74 on the testing data, indicating it could explain approximately 74% of the variance in insurance charges.

The Random Forest model performed even better, with an R^2 score of around 0.97 on training data and approximately 0.86 on test data, showing that it captures more complex patterns and interactions in the data.

The XGBoost model also showed strong performance, with an R^2 score of about 0.88 on training data and around 0.87 on testing data, confirming that it is highly capable of predicting medical insurance costs accurately.

The analysis highlighted that smoking status and age are the most significant factors affecting insurance charges. Additionally, features like BMI, number of children, and region also had noticeable effects, but to a lesser extent.

Challenges Faced

Some challenges faced during this project included:

- Correctly encoding and handling categorical data (such as sex, smoker, and region columns).
- Preventing overfitting, especially in ensemble methods like Random Forest, where the model might fit too closely to the training data.
- Choosing appropriate hyperparameters for models like Random Forest and XGBoost to balance accuracy and generalization.
- Understanding and interpreting different evaluation metrics, especially R^2 scores and residual errors.
- Ensuring that input data for predictions matched the same format and feature names as the training data to avoid warnings or incorrect predictions.

Learnings

Through this project, I gained hands-on experience in working with real-world data and implementing different regression models. I learned how to perform **exploratory data analysis (EDA)**, preprocess data (including encoding and handling missing values), and build, train, and evaluate machine learning models.

Additionally, I understood the importance of comparing different algorithms to find the best-performing model for a given problem. Visualizations played a crucial role in understanding data patterns and communicating results clearly. Overall, this project enhanced my practical skills in data science and machine learning and gave me confidence in applying these concepts to solve business problems.

Chapter 6: Conclusion

Summary

In this project, I successfully developed and evaluated multiple models — Linear Regression, Random Forest, and XGBoost — to predict medical insurance charges based on personal and health-related factors such as age, sex, BMI, number of children, smoking status, and region.

The workflow included data collection, cleaning, exploratory analysis, feature encoding, model building, evaluation, and making predictions on new data. By comparing different algorithms, I was able to select models that performed better and provided more accurate predictions.

This project provided valuable insights into the most important factors influencing insurance costs, especially smoking status and age.

Overall, this project helped me understand how data-driven approaches can support decision-making in real business scenarios. It significantly improved my technical skills, problem-solving ability, and confidence in working with machine learning tools and real-world datasets.