**School of Engineering and Technology**

**PRACTICAL FILE**

**OF**

**Machine Learning and Pattern Recognition**

**ENSP202**



|  |  |
| --- | --- |
| **Submitted to:** | **Submitted by:** |
| **Mr. Kunal Rai** | **Amit Mohanty** |
| **Samatrix** | **2301730325** |
|  | **B.Tech CSE (AI/ML)-E** |

**Index**

|  |  |  |
| --- | --- | --- |
| **S.No** | **Title** | **Page No.** |
| **1.** | **Introduction** | **1** |
| **2.** | **Objective** | **2** |
| **3.** | **Project Pipeline** | **3** |
| **4.** | **Conclusion** | **17** |

**Introduction**

**Insurance claim prediction** involves estimating the financial compensation a person may require from an insurance provider based on their personal information and medical history. By analyzing key factors such as age, sex, body mass index (BMI), smoking status, and region, we can predict the likely claim amount. This analysis is crucial for insurance companies to assess risk, calculate premiums, and maintain a balanced portfolio.

Given the variety of data and the potential complexity in the relationships between these factors, developing an **Automated Machine Learning Model** is essential to accurately predict insurance claims. The dataset includes various features, each of which has a significant impact on the claim amount. Preprocessing and feature engineering are important steps, as the presence of noisy or irrelevant data can hinder the effectiveness of the models. Therefore, cleaning, encoding, and scaling are key processes before implementing the machine learning algorithms to achieve reliable predictions.

**Objective**

The objective of this project is to predict the claim amounts in health insurance based on the user's profile and medical history. We aim to develop a Machine Learning model by implementing two regression algorithms:

1. **Linear Regression (LR)** – To understand the linear relationship between features and insurance charges.
2. **Random Forest Regressor (RFR)** – To capture non-linear relationships and interactions among features for more accurate predictions.

Additionally, we use **feature scaling** to standardize the input data, ensuring that the models perform optimally. The performance of these models is then evaluated using metrics such as **Mean Absolute Error (MAE)**, **Mean Squared Error (MSE)**, **Root Mean Squared Error (RMSE)**, and **R² score** to assess prediction accuracy and model reliability.

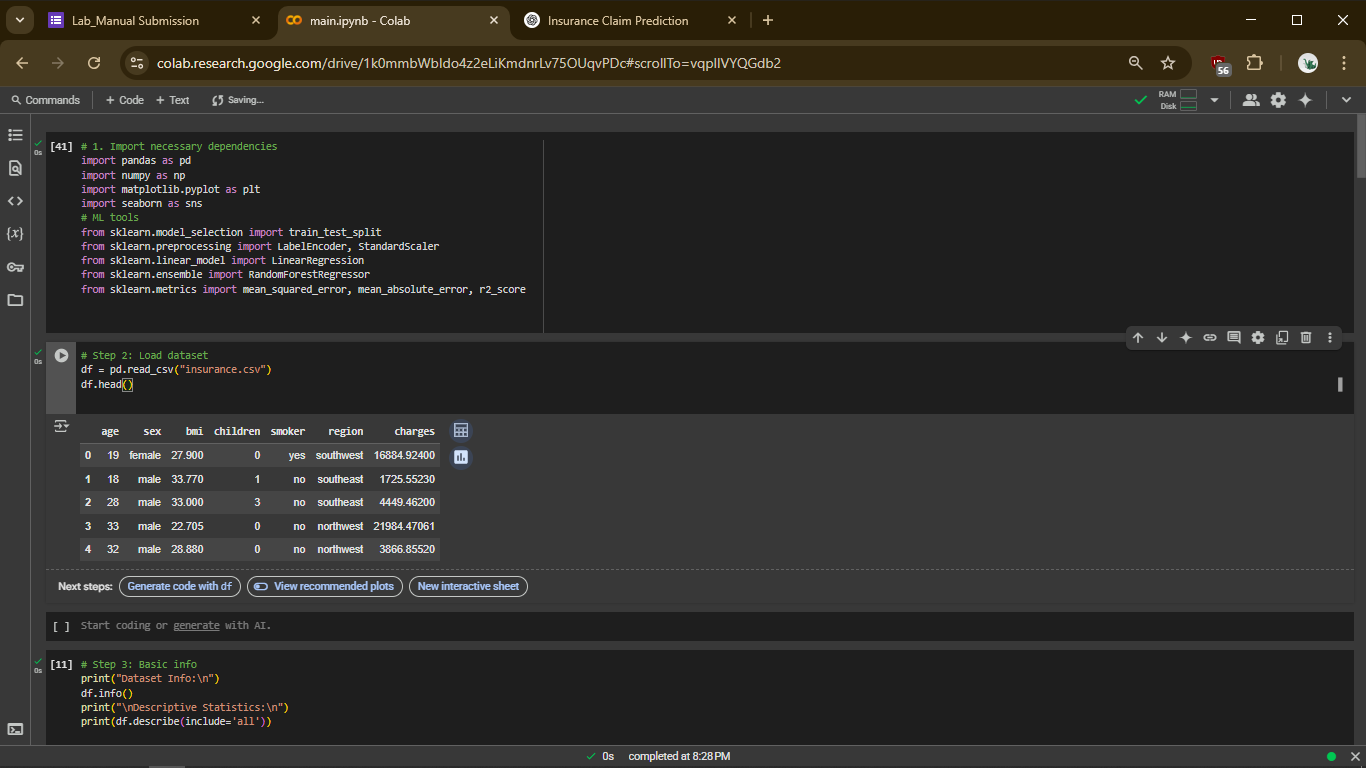
**Project Pipeline**

The various steps involved in the Machine Learning Pipeline are :

1. Import necessary dependencies.
2. Read and load the dataset.
3. Exploratory data analysis.
4. Data visualization of Target Variables.
5. Data preprocessing.
6. Splitting our data into Train and Test subset.
7. Transforming dataset using TF-IDF Vectorizer.
8. Function for model evaluation.
9. Model building.
10. Conclusion

**Importing the necessary dependencies:**

Here in this part, we import all the necessary libraries that we will use in our project. The choice of libraries depends on the approach we will follow.

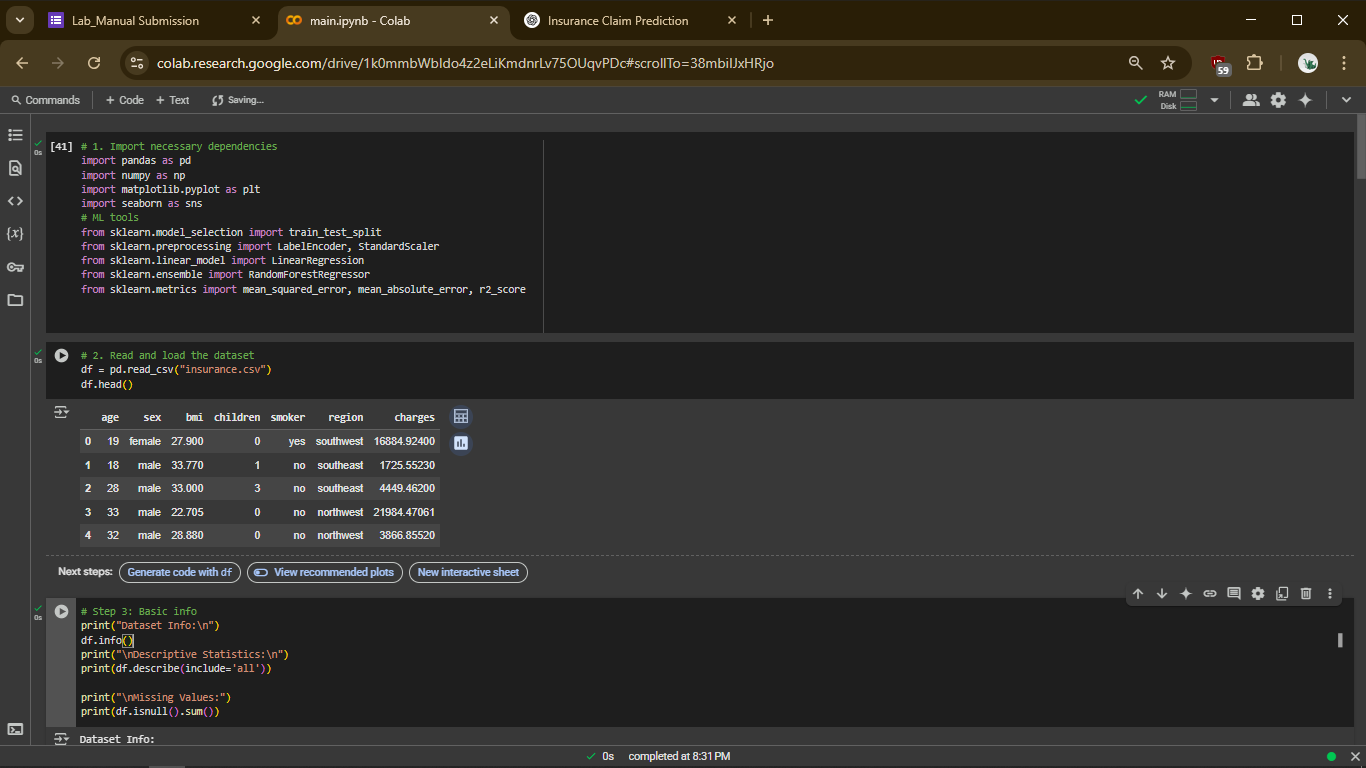


**Reading and loading the dataset:**

In any project related to the manipulation and analysis of data, we always start by collecting the data on which we are going to work. In our case, we will import our data from a .csv file.

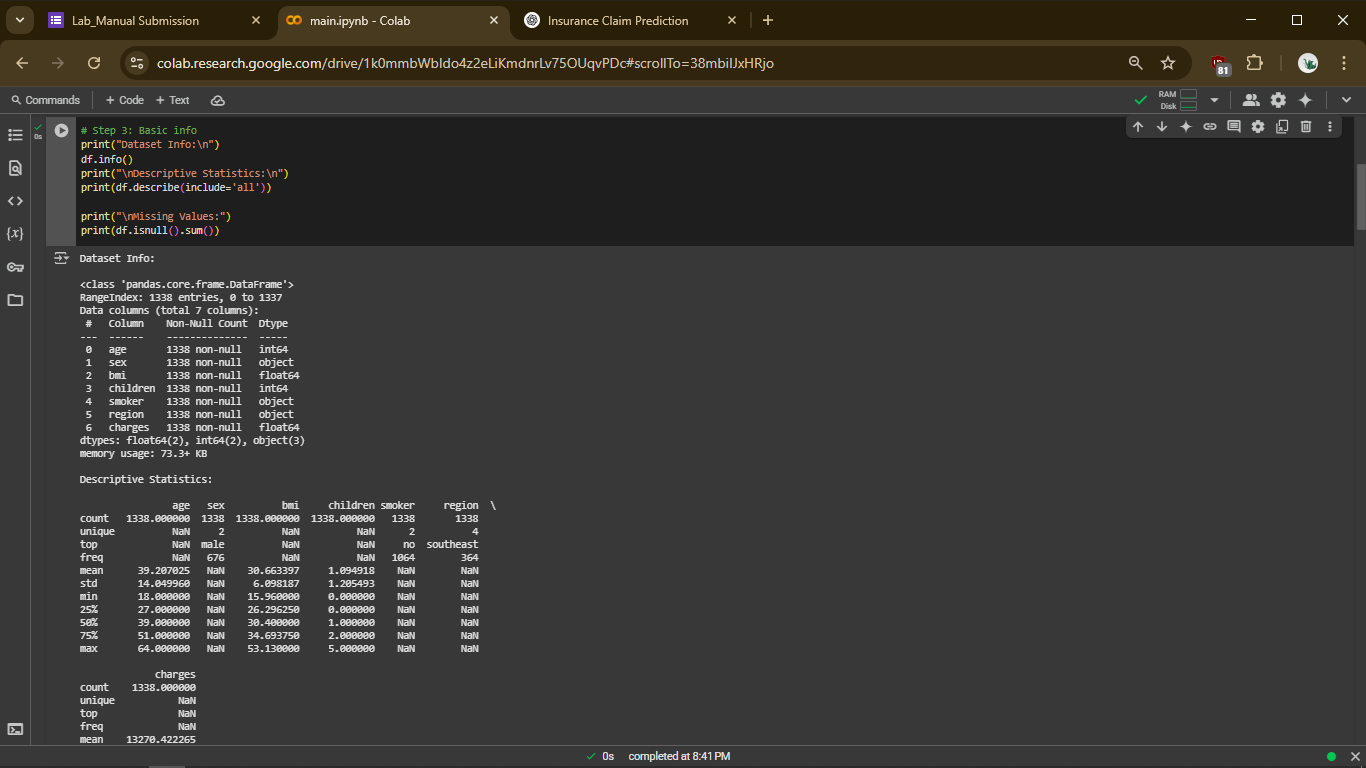
The various columns present in the dataset are:

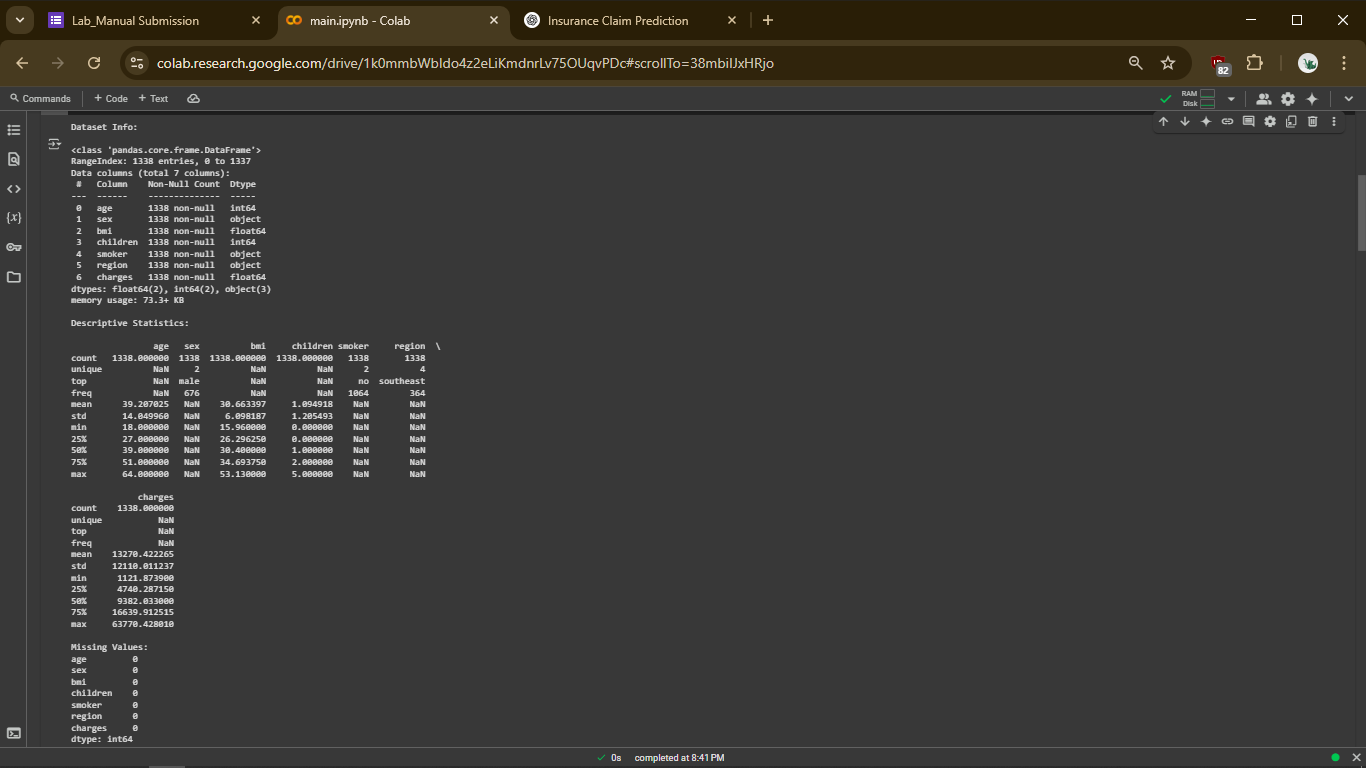
* age: The age of the policyholder.
* sex: The gender of the policyholder (male or female).
* bmi: Body Mass Index (BMI) of the policyholder.
* children: The number of children or dependents covered by the insurance.
* smoker: Whether the policyholder is a smoker (yes or no).
* region: The geographical region in which the policyholder resides.
* charges: The medical insurance charges (the target variable).



**Exploratory data analysis:**

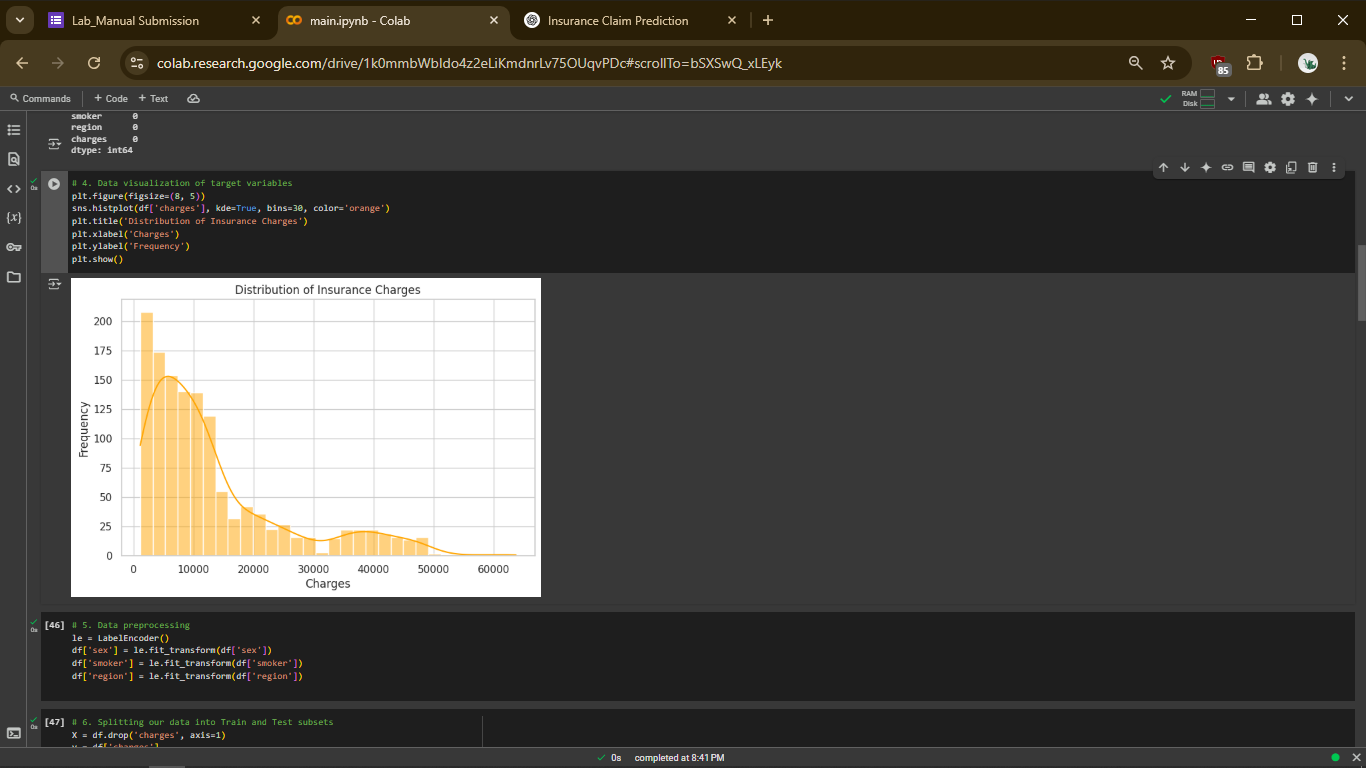
In this part, the objective is to understand the imported data as much as possible. We begin by analyzing a sample of the data, checking the shape of the dataset, and reviewing the column names. We also examine the data types of each column, identify any null values, and perform other initial data checks. Essentially, we process the data to gain insights and, most importantly, target the columns that are relevant for our analysis, such as those affecting the prediction of **insurance charges**.

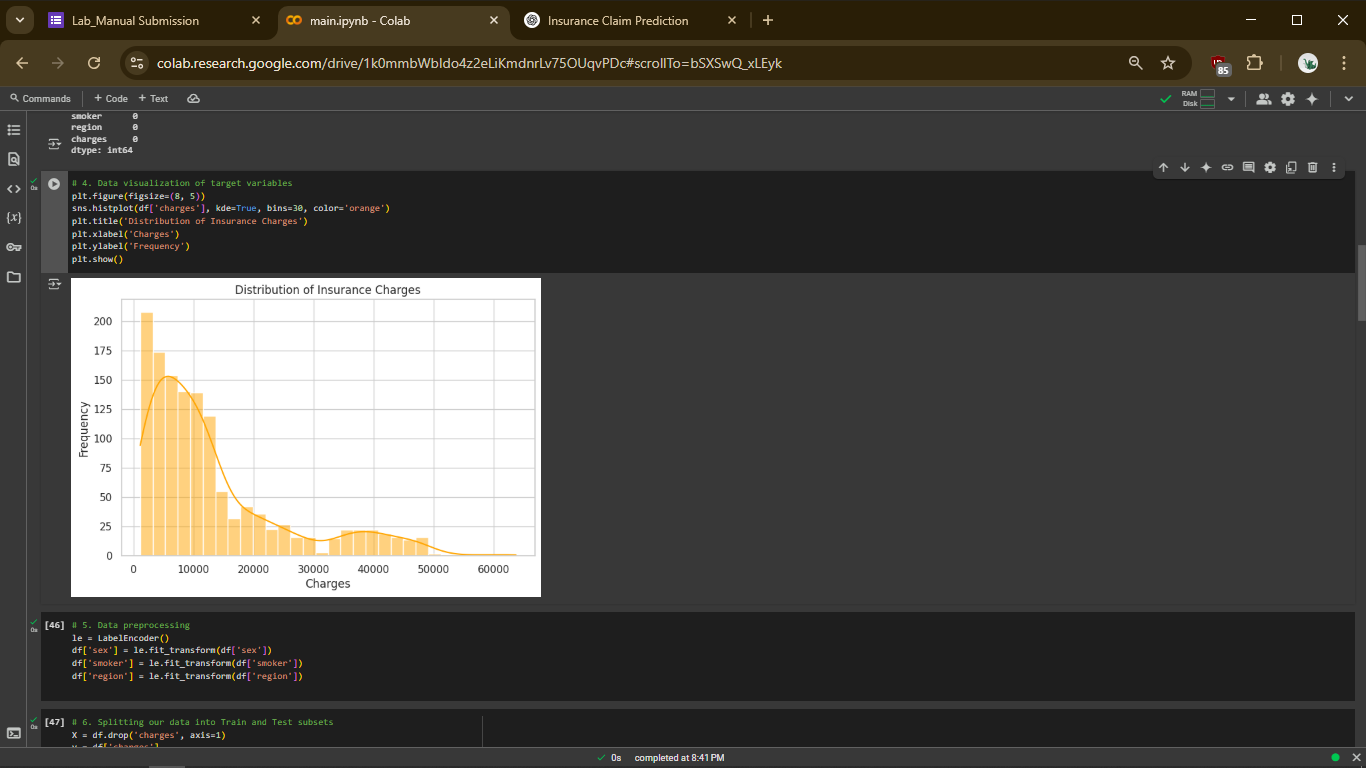




**Data visualization of target variables:**

After processing our data and identifying the columns we are interested in, the next step is to visualize the data using informative plots. The reason for using visualizations is that they allow us to interpret patterns, trends, and relationships more effectively. Plots make the data more intuitive and help uncover insights that may not be immediately obvious through raw numbers—especially in understanding how features like **age**, **BMI**, or **smoking status** influence the **insurance charges**.





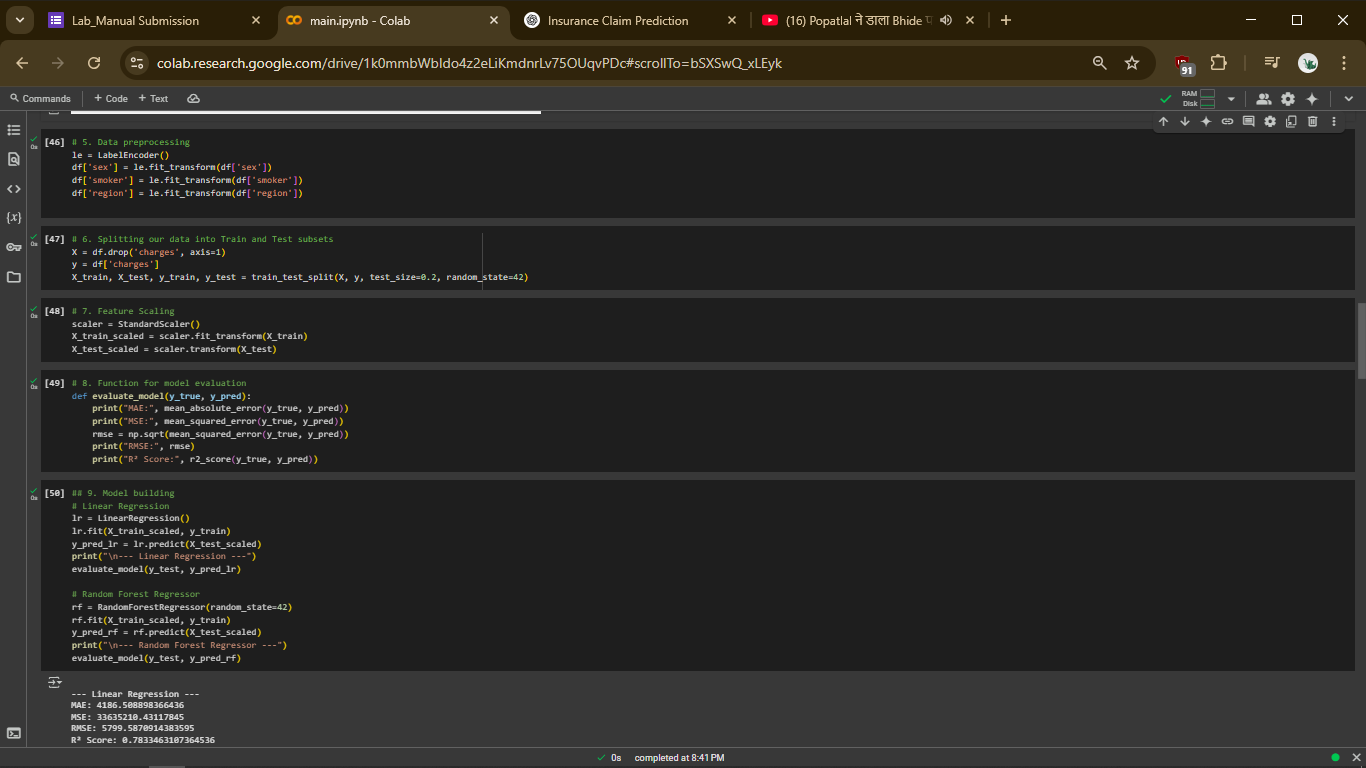
**Data preprocessing:**

**Data Preprocessing:**

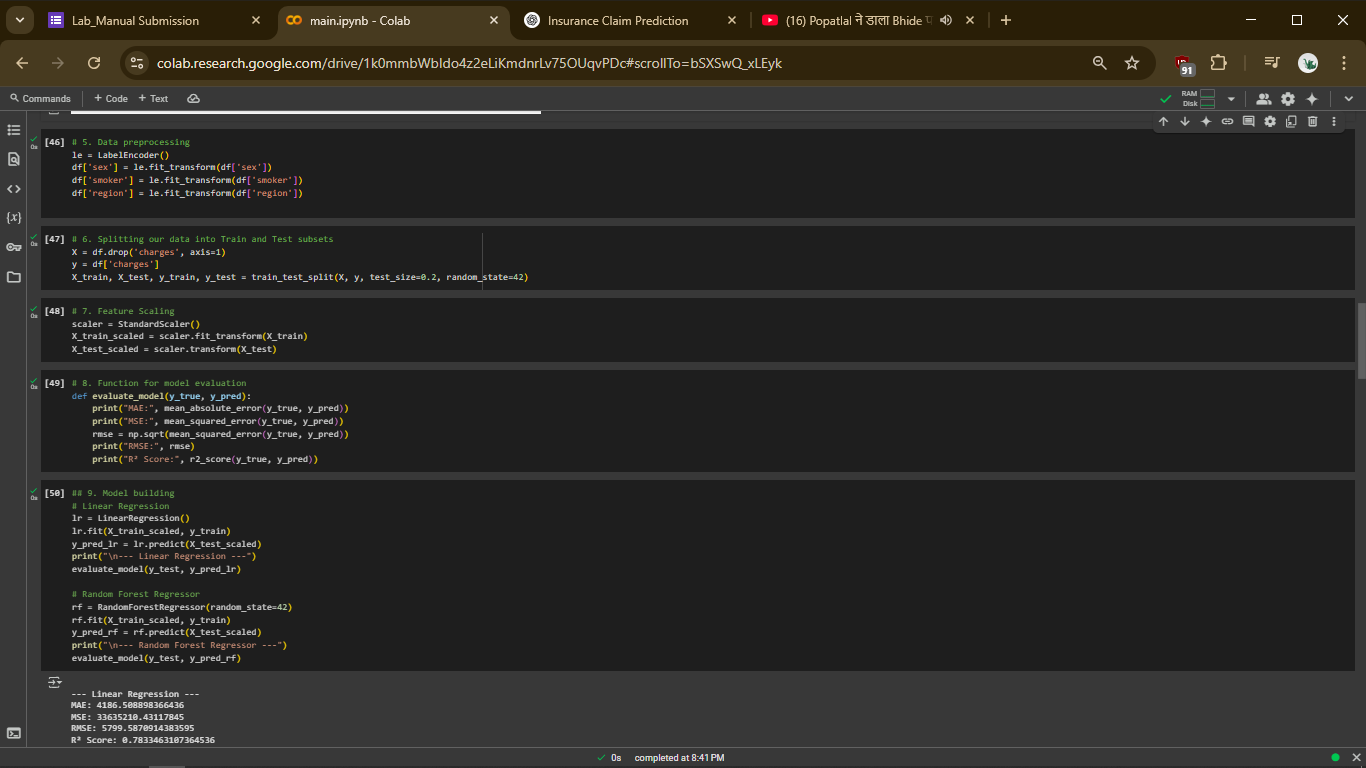
Before training the model, we will perform several preprocessing steps on the dataset to prepare it for analysis and modeling. These include:

* **Encoding categorical variables** such as sex, smoker, and region to numerical values.
* **Checking for and handling missing values**, if any.
* **Standardizing numerical features** like age, bmi, and children using feature scaling techniques to ensure all variables are on the same scale.
* **Ensuring correct data types** for each column to avoid processing issues.

These preprocessing steps are essential for improving model performance and ensuring accurate predictions. We will go through each of these steps in detail during the implementation phase..



**Splitting our data into Train and Test subset:**



-  **random\_state** is used to ensure reproducibility of the results. By setting a random\_state value in **train\_test\_split**, we make sure that the train and test datasets remain the same across multiple runs, which is especially helpful for debugging and consistency.

 X contains the **independent features**: age, sex, bmi, children, smoker, region.

 y contains the **target variable**: charges.

 X\_train contains 80% of the data used to train the model.

 X\_test contains 20% of the data used to test the model.

 y\_train contains 80% of the corresponding charges values for training.

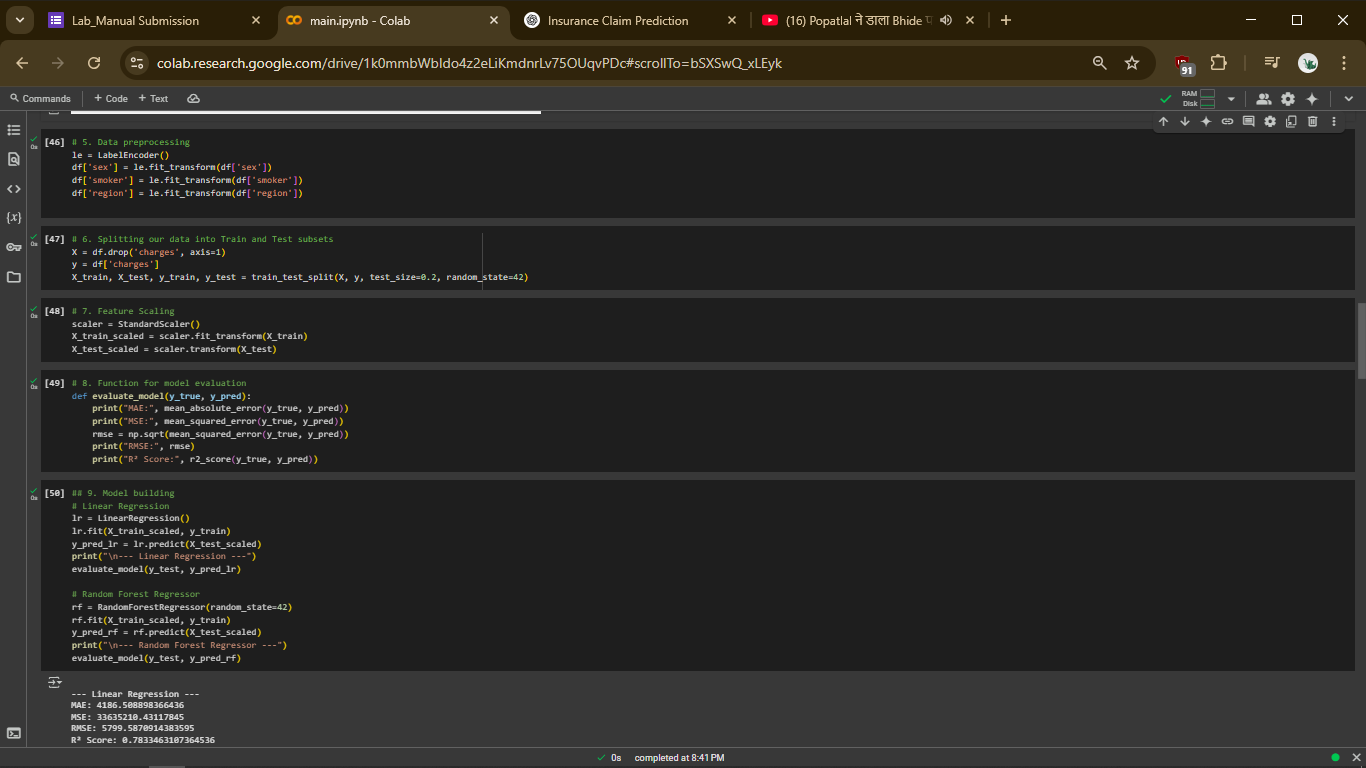
 y\_test contains 20% of the corresponding charges values for testing.

**Feature Scaling:**

Feature scaling is an essential step before training machine learning models, especially those that rely on distance-based calculations or assume normally distributed data. In our dataset, features like age, bmi, and children have different numerical ranges, which can lead to biased results in certain algorithms.

To address this, we use Standardization (Z-score normalization) to scale the numerical features. This technique transforms the data such that it has a mean of 0 and a standard deviation of 1. It ensures that each feature contributes equally to the model training process.

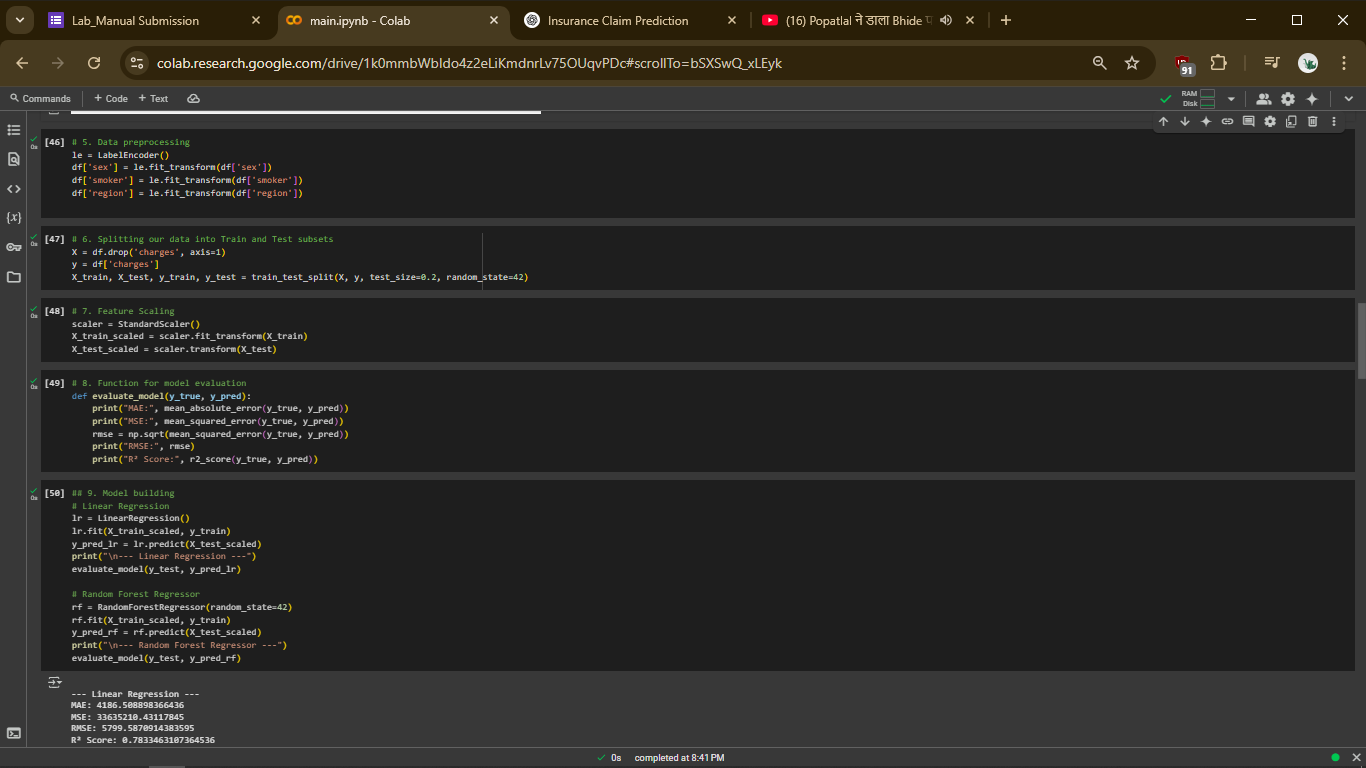
We apply feature scaling after splitting the dataset into training and testing sets to avoid data leakage.



**Function for model evaluation:**

After training the model, we apply evaluation metrics to assess how well it performs in predicting insurance claim amounts. Since this is a **regression problem**, we use the following evaluation metrics:

* **Mean Absolute Error (MAE):** Measures the average magnitude of the errors in a set of predictions, without considering their direction. Lower values indicate better performance.
* **Mean Squared Error (MSE):** Represents the average of the squares of the errors—that is, the average squared difference between the estimated and actual values. It penalizes larger errors more than MAE.
* **Root Mean Squared Error (RMSE):** The square root of MSE, providing error in the same units as the target variable. It’s easier to interpret and compare to actual charges.
* **R² Score (Coefficient of Determination):** Indicates how well the independent variables explain the variability of the target variable. An R² close to 1 means the model explains most of the variability.

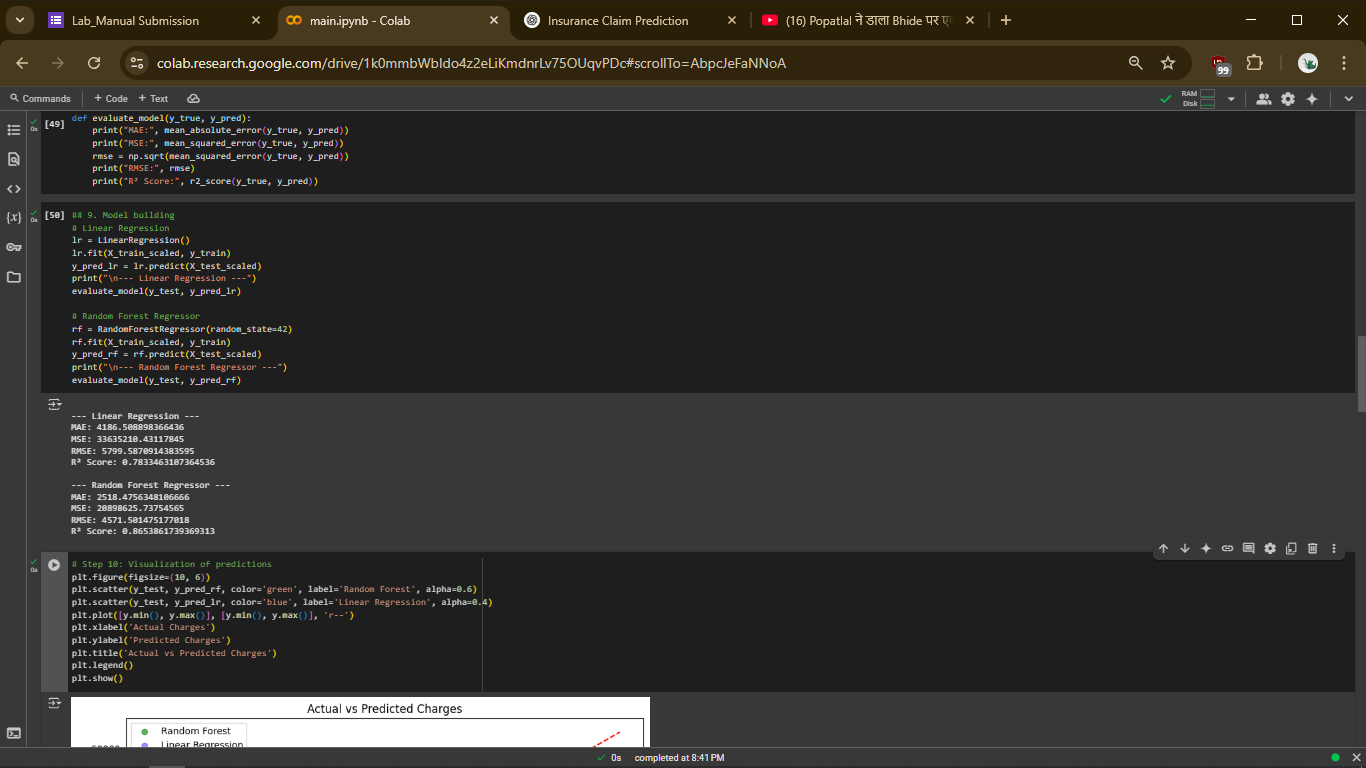


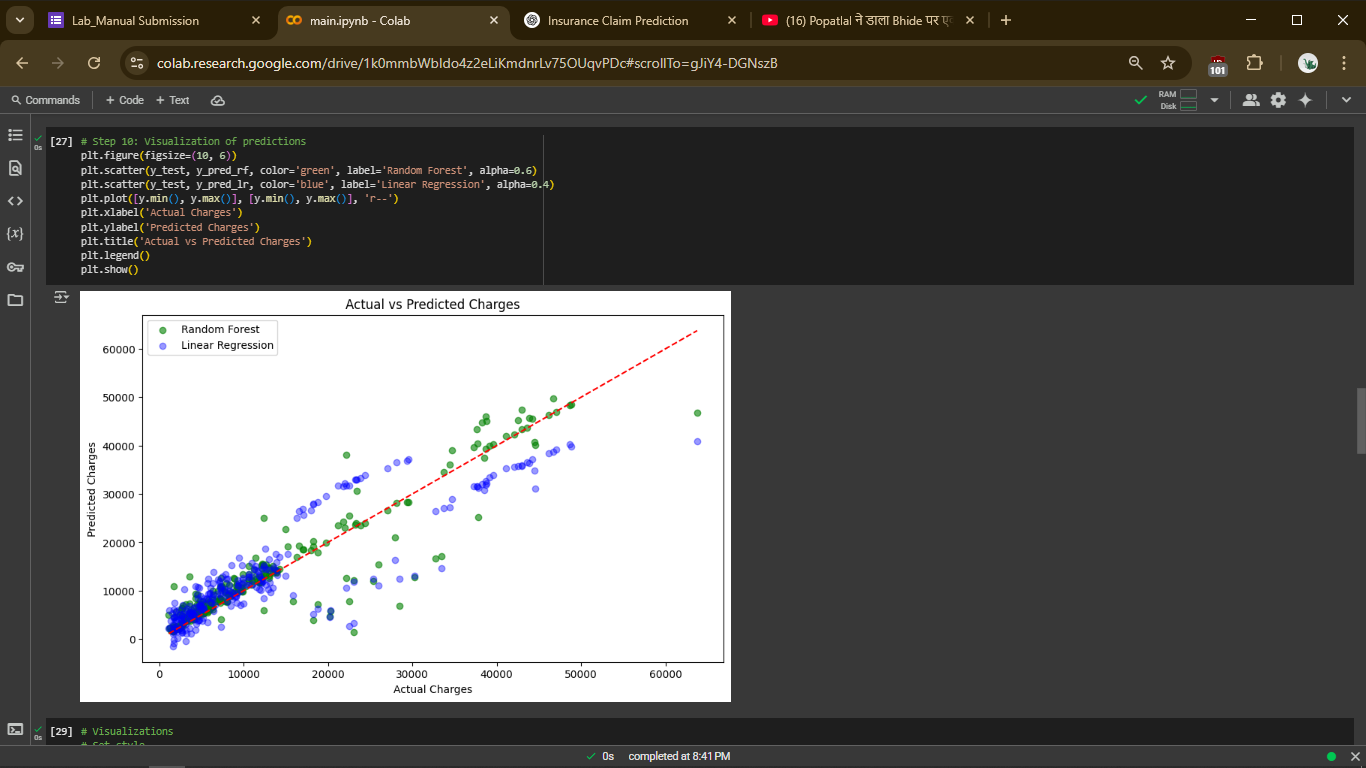
**Model building:**

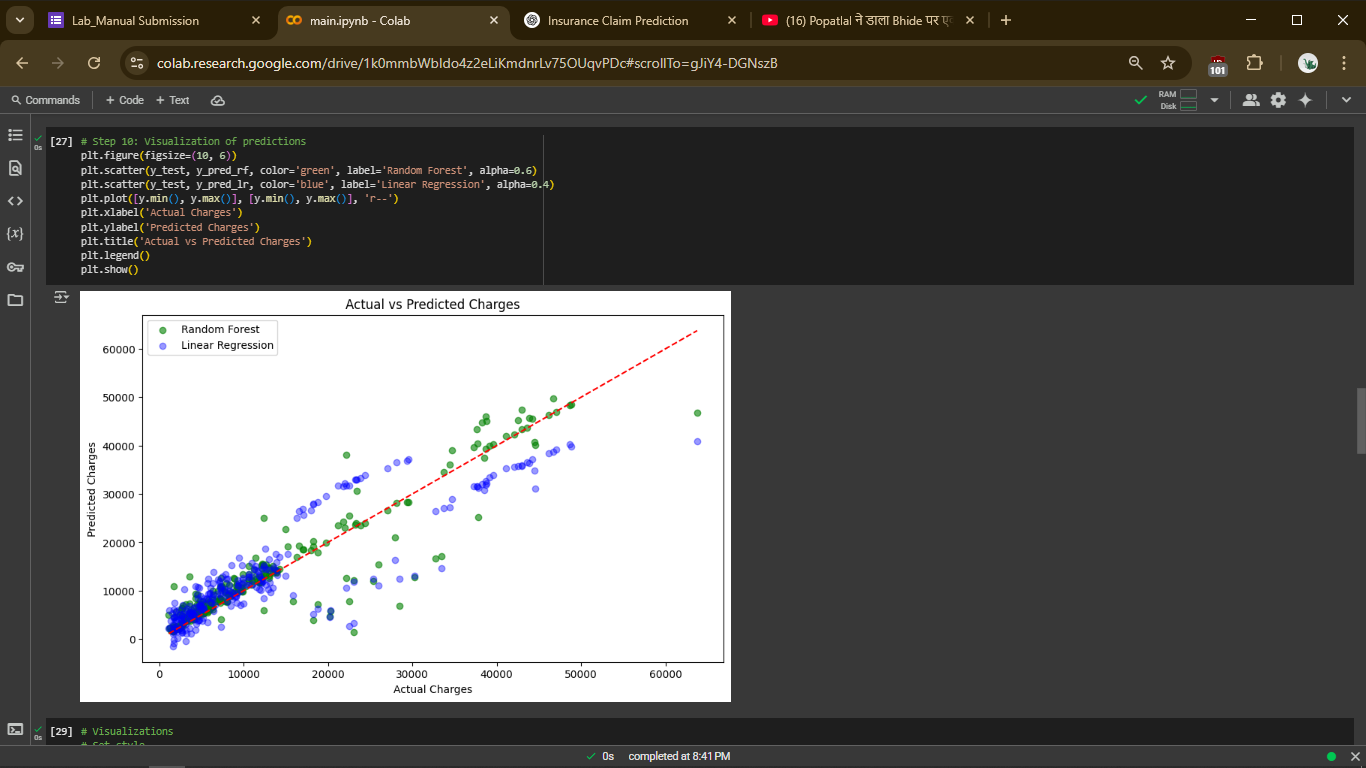
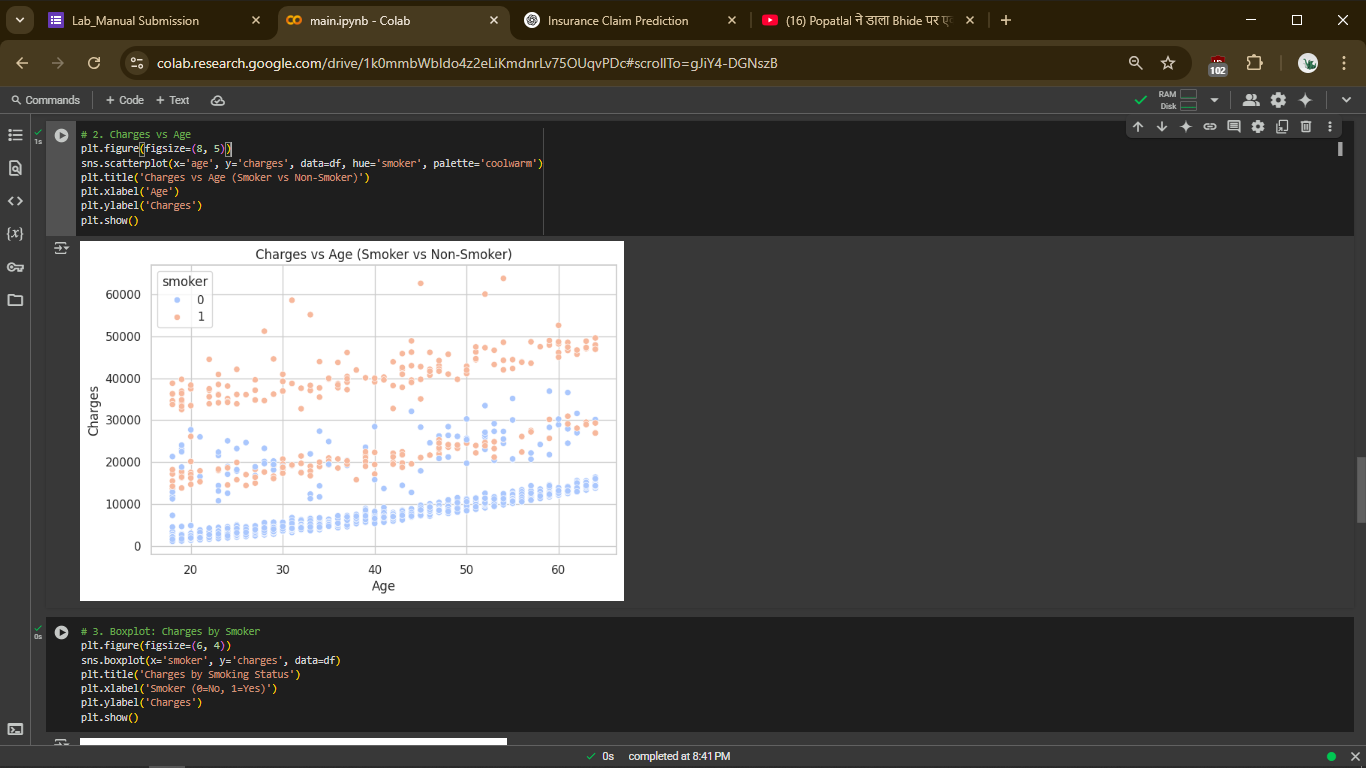
In this project, we have used three different regression models respectively:

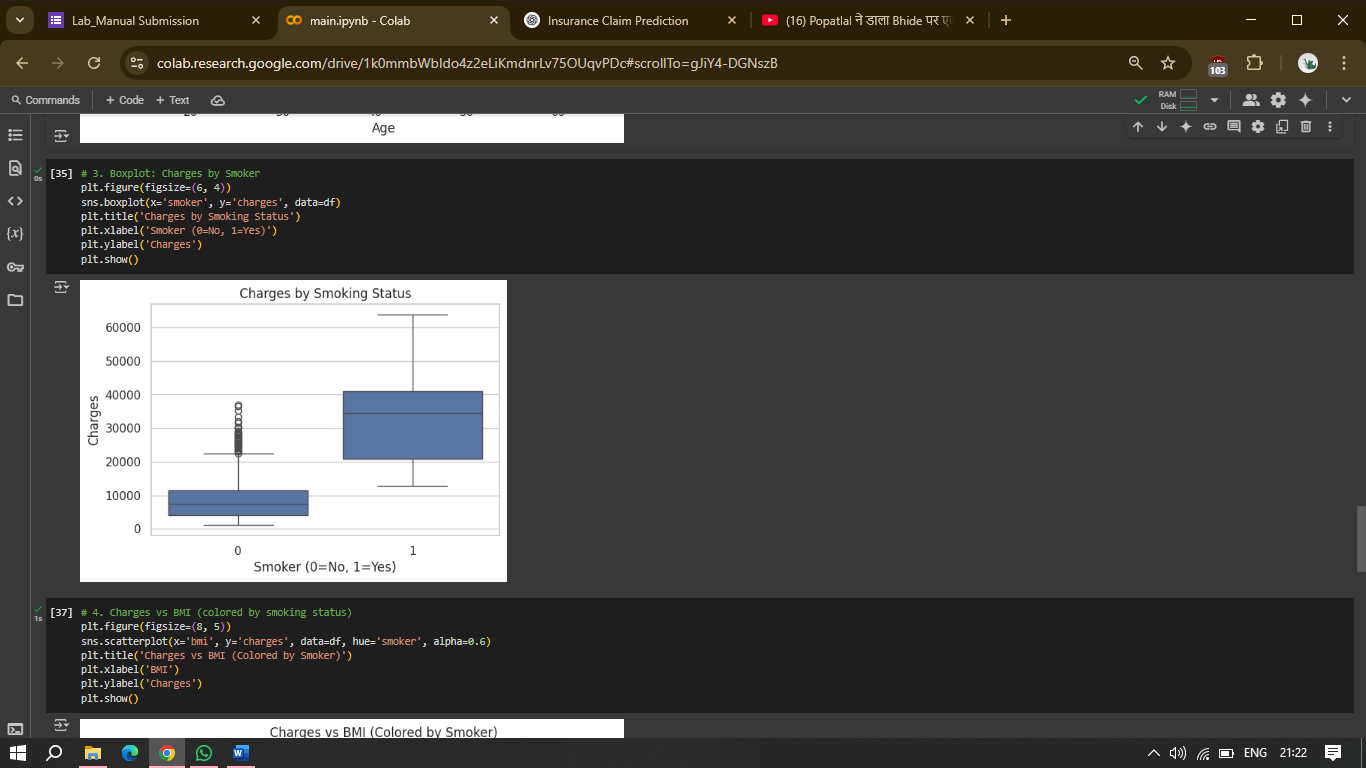
* **Linear Regression**
* **Random Forest Regressor**
* **Support Vector Regressor (SVR)**

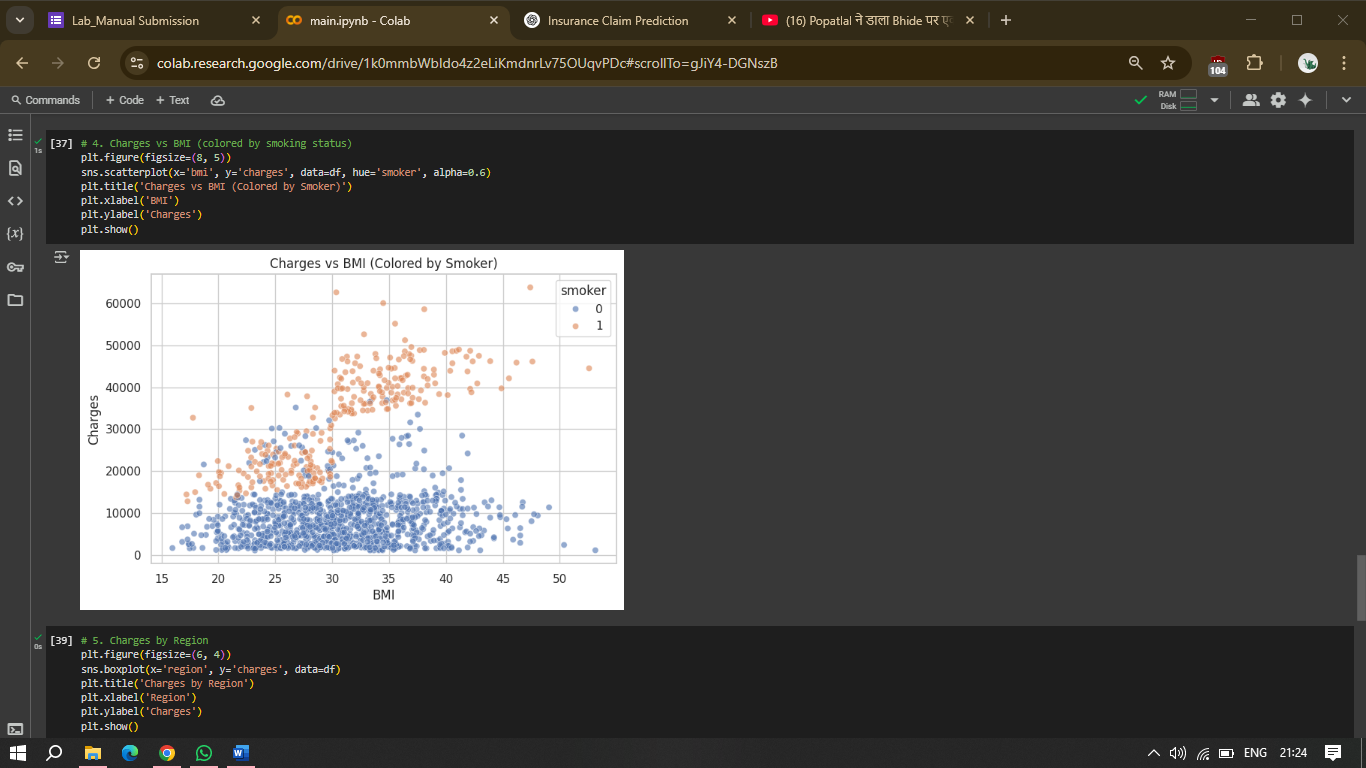
The rationale behind choosing these models is to explore a range of algorithms—from simple linear models to more complex ensemble and kernel-based methods—in order to identify the one that delivers the best performance in estimating health insurance claim amounts based on user profile and medical history.

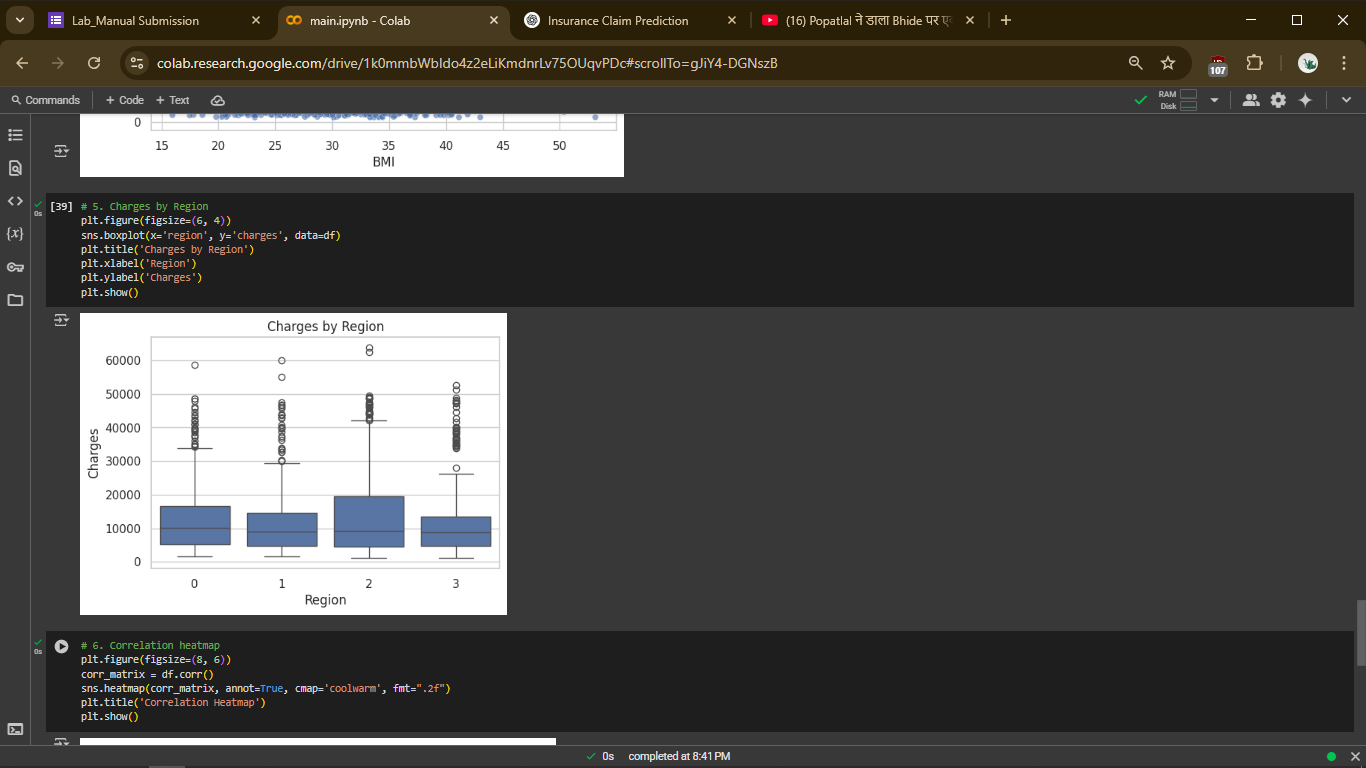


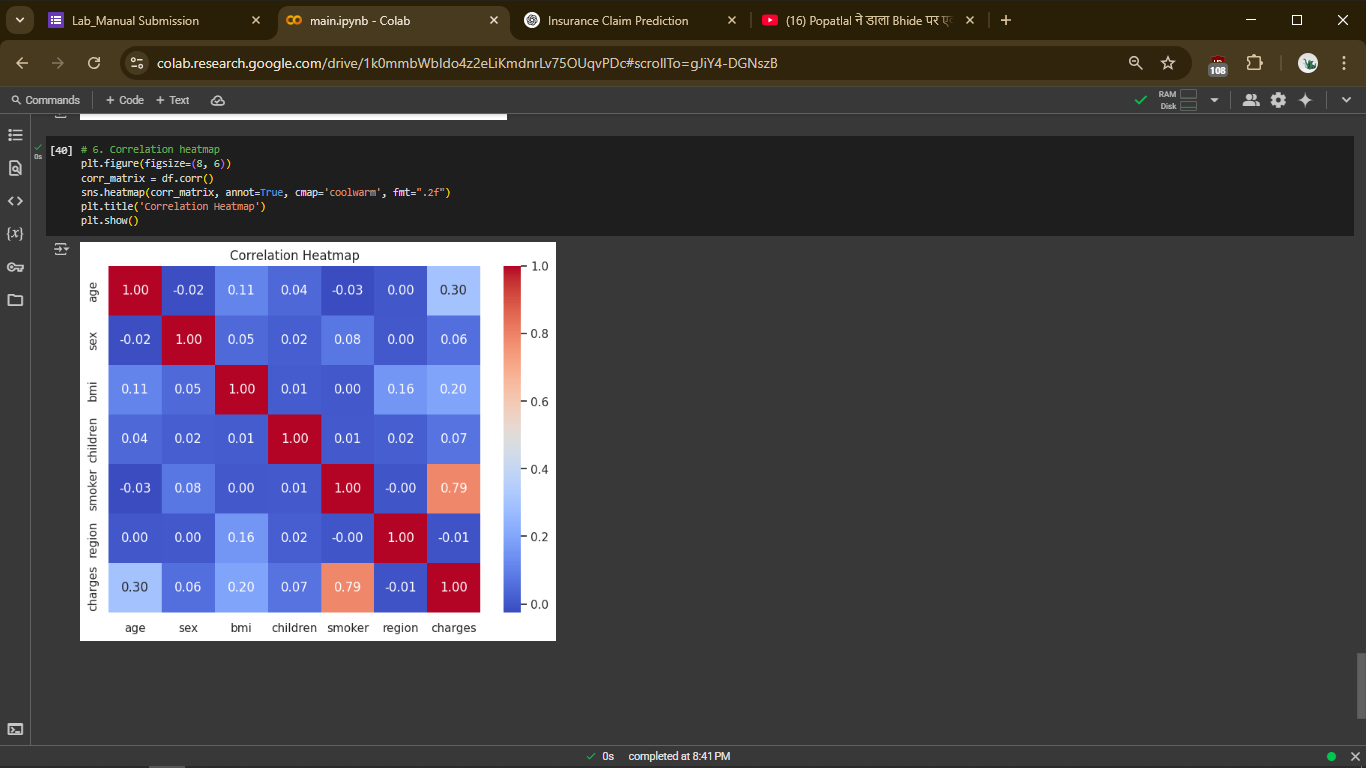












**Conclusion**

* **Execution Time**: When comparing model run-times, Linear Regression is the fastest, followed by Random Forest Regressor, while Support Vector Regressor (SVR) takes the longest to execute due to its computational complexity.
* **Accuracy (R² Score):** In terms of predictive performance:
  + Random Forest Regressor performs the best with the highest R² score, capturing non-linear patterns effectively.
  + Support Vector Regressor comes next with competitive accuracy.
  + Linear Regression performs reasonably well but is less accurate due to its simplicity and linear assumptions.
* **Error Metrics (MAE, MSE, RMSE):**
  + Random Forest shows the lowest error values, indicating better generalization.
  + SVR has moderate errors.
  + Linear Regression exhibits the highest error values, reflecting limited flexibility in modeling complex relationships.

We conclude that the Random Forest Regressor is the most suitable model for predicting insurance claim amounts in our dataset. Despite taking slightly longer to train, its accuracy and lower error rates outweigh the minor increase in computation time.

Furthermore, Linear Regression still holds value due to its simplicity and interpretability, aligning with Occam's Razor—in scenarios with limited data complexity, simpler models can often yield effective results.