**Name:** Niranjan Joglekar

**Roll No:** 281043

**Batch:** A2

Assignment 2

**Problem Statement:**

Perform the following operations using R/Python on the data sets:

1. Compute and display summary statistics for each feature available in the dataset.(e.g. minimum value, maximum value, mean, range, standard deviation, variance and percentiles
2. Illustrate the feature distributions using histogram.
3. Data cleaning, Data integration, Data transformation, Data model building (e.g. Classification)

**Objectives:**

1. To preprocess and clean the dataset for improved model performance.
2. To split the dataset for effective model training and evaluation.
3. To build and apply a Decision Tree classifier for admission prediction.
4. To assess the model’s accuracy and effectiveness.

**Resources used:**

1. Software used: Jupyter Notebook, Anaconda Navigator
2. Libraries used: Pandas, Matplotlib, sklearn, Seaborn

**Theory:**

#### **1. Linear Regression**

#### Linear Regression is a supervised learning algorithm used primarily for regression tasks, where the goal is to predict a continuous output value based on one or more input features. It models the relationship between the dependent variable (target) and one or more independent variables (features) by fitting a straight line through the data points.

#### **The main components of a Linear Regression model are:**

#### **Independent Variables (Features):** Input variables used to predict the output.

#### **Dependent Variable (Target):** The continuous value we aim to predict**.**

#### **Regression Line:** A straight line that best fits the data. In simple linear regression, this line is represented by the equation: y=mx+cy = mx + cy=mx+c **where:**

#### ***y* is the predicted value**

#### ***m* is the slope (coefficient of the feature)**

#### ***x* is the feature value**

#### ***c* is the intercept**

#### **The model works by finding the best-fitting line that minimizes the difference between the actual values and the predicted values. This difference is usually measured using a loss function such as Mean Squared Error (MSE).**

#### The learning process involves:

#### Calculating the error between predicted and actual values.

#### Adjusting the coefficients (m and c) to minimize this error using optimization techniques like Gradient Descent.

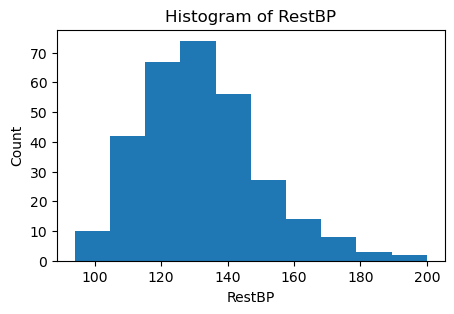
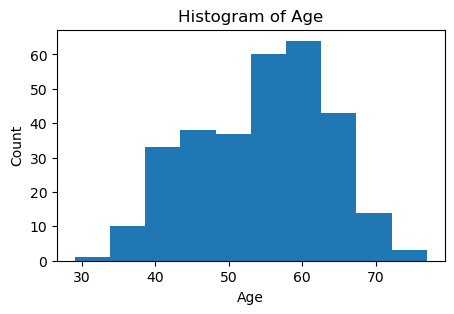
#### **Linear Regression can be of two types:**

#### **Simple Linear Regression:** Involves one independent variable.

#### **Multiple Linear Regression:** Involves two or more independent variables.

#### 

#### **2. Visualizing data**



#### **3. Data Preprocessing**

Data preprocessing ensures the dataset is clean and ready for analysis. Steps include:

* Handling missing values (e.g., imputation, removal).
* Encoding categorical variables (e.g., Label Encoding, One-Hot Encoding).
* Normalization and scaling for numerical stability.

#### **4. Model Evaluation Metrics**

To assess model performance, we use:

* **Accuracy:** Measures correct predictions over total instances.
* **Precision & Recall:** Evaluate positive class performance.
* **F1-score:** Balances precision and recall.
* **Confusion Matrix:** Displays true/false positives and negatives.

**Methodology:**

#### **1. Data Preprocessing**

* Handle missing values, if any, using imputation techniques.
* Perform feature scaling or normalization, if required.
* Encode categorical features if necessary.

df['Ca'] = df['Ca'].fillna(df['Ca'].mean())

df['Thal'] = df['Thal'].fillna(df['Thal'].mode()[0])

df['AHD'] = df['AHD'].map( {'Yes':1 ,'No':0})

#### **2. Data Preparation**

* Split the dataset into training (80%) and testing (20%) subsets using Scikit-Learn’s train\_test\_split() function.

x = df[["RestBP", "Chol", "Fbs", "RestECG", "MaxHR", "ExAng", "Oldpeak", "Slope", "Ca"]]

y = df["AHD"]

# Splitting training

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size=0.20)

#### **3. Model Implementation**

* Implement a Linear regressor using Scikit-Learn’s DecisionTreeClassifier.
* Train the model using the training dataset.
* Predict admission outcomes using the test dataset.

model = LinearRegression()

model.fit(x\_train, y\_train)

#### **4. Model Evaluation**

* Evaluate model performance using metrics such as Accuracy, Precision, Recall, and F1-score.
* Use a Confusion Matrix to analyze model predictions.

Mean absolute error: 0.3048410656626215

Mean squared error: 0.15662976673313642

**Conclusion:**

* Preprocessing ensures that the dataset is clean and suitable for modeling.
* Linear regressor can effectively classify ADH on heart dataset.
* Evaluating performance metrics helps in understanding model accuracy and effectiveness.

