Name: Prachee Prasad
Class: CSE-AI SY A
PRN: 22311748
Roll no.: 281**60**

Assignment 2

Problem Statement

Perform the following operations using R/Python on the data sets:
a) 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
b) Illustrate the feature distributions using histogram.
c) Data cleaning, Data integration, Data transformation, Data model building (e.g. Classification)

Objective

The objective of this assignment is to explore a dataset by computing summary statistics, visualizing feature distributions, and performing essential preprocessing steps such as data cleaning, integration, and transformation. The assignment concludes with building a basic classification model using the cleaned and processed data.

Resources Used

Programming Language: Python

Libraries Used:

pandas – for data manipulation and analysis

numpy – for numerical computations

matplotlib / seaborn – for data visualization

sklearn – for building the classification model

Dataset: heart.csv (heart disease dataset)

Methodology of Solution

Data Collection

Imported the dataset using pandas

Verified the structure and previewed the data using df.head() and df.info()

Summary Statistics

Used df.describe() to get statistical details of each feature

Calculated min, max, mean, range (max - min), standard deviation, and variance using built-in functions

Calculated percentiles using df.quantile()

Feature Distribution Visualization

Plotted histograms using df.hist() and seaborn’s histplot for each numerical feature

Helped to understand the spread, skewness, and outliers in the data

Data Cleaning

Checked for missing values using df.isnull().sum()

Removed or filled missing values using dropna() or fillna()

Removed duplicate records if any

Data Integration

If the dataset was split into multiple parts, combined them using pd.concat() or pd.merge()

Ensured consistency in column names and formats

Data Transformation

Transformed categorical data using label encoding or one-hot encoding

Scaled numerical features using normalization or standardization

Data Model Building (Classification)

Split the dataset into training and test sets using train_test_split

Used RandomForestClassifier or any basic classification algorithm

Trained the model using fit() and evaluated its accuracy using accuracy_score()

Advantages

- Helps understand the data thoroughly before building any machine learning model
- Visualization provides insights into distribution, trends, and potential outliers
- Data cleaning and transformation improve model performance
- Building a classification model helps to apply and test preprocessing knowledge in a real-world scenario

Disadvantages

- If the dataset is too small or too large, certain summary statistics or visualizations might not be meaningful.

- Manual data cleaning can be time-consuming.

- Selecting the wrong model or ignoring preprocessing steps can lead to poor performance.

Applications with Working Examples

Health Diagnosis
Dataset: Patient medical records
Summary stats help doctors understand blood pressure, cholesterol levels, etc.
Classification model predicts diseases like diabetes based on input features

Banking and Finance
Dataset: Customer loan data
Summary statistics identify average income, loan amounts, and risk ranges
Classification used to predict loan default

E-commerce
Dataset: User behavior data
Feature distribution identifies popular products, peak shopping hours
Classification model predicts whether a user will click on an ad or not

Working / Algorithm

Step 1: Load the dataset
Step 2: Explore the data structure
Step 3: Compute summary statistics: mean, min, max, std, percentiles
Step 4: Visualize feature distributions using histograms
Step 5: Clean the data by handling missing values and duplicates
Step 6: Integrate datasets (if multiple)
Step 7: Transform features – encode categorical and scale numerical values
Step 8: Split data into training and testing sets
Step 9: Train a classification model on the training data
Step 10: Predict and evaluate the model on the test set

This algorithm ensures that the dataset is fully prepared and modeled efficiently, following the best practices in data preprocessing and classification.

Conclusion

In this assignment, we performed key steps in the data science pipeline: starting from data loading and cleaning, to statistical analysis and visualizations, and finally building a classification model. Each step played an important role in preparing the data and understanding its behavior, which is essential for creating effective and accurate machine learning models.