Name: Jiya Sharma
Roll no: 218056
PRN: 22311741

Assignment 1

Statement:

Q. In this assignment we have to do
Perform the following operations using R/Python on suitable data sets:
a) read data from different formats (like csv, xls)
b) Find Shape of Data
c) Find Missing Values
d) Find data type of each column
e) Finding out Zero's
f) Indexing and selecting data, sort data,
g) Describe attributes of data, checking data types of each column,
h) counting unique values of data, format of each column, converting variable data type (e.g. from long to short, vice versa)

Objective:

- This assignment aims to introduce the Pandas library and its fundamental functions. Pandas provides powerful tools for reading various file formats such as CSV and Excel, making data handling more efficient.
- It also familiarizes users with essential data cleaning and preprocessing techniques, ensuring data accuracy and consistency.
- Additionally, it enhances our ability to manage and manipulate data in different formats, improving proficiency in data analysis.

Resources Used:

- Software Used: Jupyter Notebook
- Library Used: Pandas

Introduction to Pandas:

Pandas is a widely-used open-source Python library designed for data manipulation and analysis. It offers intuitive data structures and powerful functions, making it an essential tool for working with structured data.
Key components of Pandas:

- **Series:** A one-dimensional labeled array capable of holding any data type.
- **DataFrame:** A two-dimensional labeled data structure with columns of potentially different types.

These structures enable users to perform various data operations, such as loading data from different file formats (CSV, Excel, SQL databases), manipulating data (sorting, filtering, grouping), and executing statistical and analytical tasks.

Basic Functions Used in the Program:

- pd.read_csv() – Reads data from a CSV file into a DataFrame.
- head() – Displays the first few rows of the DataFrame for a quick overview.
- sort_values() – Sorts the DataFrame by a specified column (e.g., 'Age').
- describe() – Generates descriptive statistics, including count, mean, standard deviation, minimum, and maximum values.
- nunique() – Returns the number of unique values in a column, useful for identifying distinct categories.
- info() – Prints detailed information about the DataFrame, such as the number of columns, data types, memory usage, and non-null values.

Methodology:

- **Data Collection and Exploration:**

1. Obtain a heart attack prediction dataset containing relevant features such as age, gender, blood pressure, cholesterol levels, etc.
2. Load the dataset into a Pandas DataFrame and examine its structure, including the number of samples, features, and missing or erroneous values.

Data Preprocessing:

1. Handling Missing Values: Identify and manage missing values using methods such as imputation (mean, median, mode) or removing rows/columns with significant missing data.
2. Data Cleaning: Remove duplicates, correct errors, and ensure consistent data formatting.

Feature Engineering:

1. Feature Selection: Identify important features for heart attack prediction through correlation analysis or feature importance scores.
2. Feature Encoding: Convert categorical variables into numerical format using one-hot encoding or label encoding to make them suitable for machine learning algorithms.

Advantages of Pandas:

1. Ease of Use: Pandas is user-friendly and widely recognized for its intuitive functionalities.
2. Powerful Data Structures: Provides efficient data handling through Series and DataFrame structures.
3. Extensive Functionality: Supports a wide range of data manipulation techniques, making it a versatile tool for data analysis.

Disadvantages of Pandas:

1. High Memory Consumption: Working with large datasets may require significant memory, which can impact performance.
2. Limited Interoperability: Pandas is deeply integrated into the Python ecosystem, which may restrict its compatibility with other programming languages or environments.

Conclusion:

This assignment introduced the Pandas library as a fundamental tool for data manipulation and analysis in Python. We explored its core functions, including reading and organizing data, handling missing values, and performing basic statistical operations. Through practical exercises, we developed a solid foundation in using Pandas for data analysis, making complex tasks more efficient and accessible. These skills will serve as a strong starting point for more advanced data analysis projects in the future.