Name: Pathan Firdos Maheraj
Roll no: 281073
Batch: A1

Assignment 1

Statement:

Q. 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 Zeros
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:

1. This assignment aims to introduce the Pandas library and its basic functions, which provide functionality for reading different file formats such as CSV and Excel.
2. Additionally, it familiarizes users with data cleaning and preprocessing techniques.
3. Enhance our skills in handling data in various formats, improving our proficiency in data analysis and manipulation.

Resources used:

1. Software used: Google Colab
2. Library used: Pandas

Introduction to Pandas:

1. Pandas is a powerful and widely-used open-source Python library for data manipulation and analysis.
2. It provides easy-to-use data structures and functions, making it an essential tool for working with structured data.
3. At the core of Pandas are two main data structures: Series and DataFrame.
4. A Series is a one-dimensional labeled array capable of holding any data type.
5. A DataFrame is a two-dimensional labeled data structure with columns of potentially different types.
6. These data structures allow users to perform a wide range of operations on data, including loading data from various file formats (such as CSV, Excel, SQL databases), manipulating data (e.g., sorting, filtering, grouping), and performing statistical and analytical tasks.

Some basic functions that we used in the program:

1. pd.read_csv(): This function is used to read data from a CSV file into a DataFrame.
2. shape: Returns the number of rows and columns in the dataset.
3. isnull().sum(): Identifies missing values in the dataset.
4. dtypes: Returns the data type of each column in the dataset.
5. (df == 0).sum(): Identifies the number of zeros in each column.

6. sort_values(): Sorts the DataFrame by the values of a specified column, allowing data to be arranged in ascending order.
7. describe(): Generates descriptive statistics for numerical columns in the DataFrame, such as count, mean, standard deviation, minimum, and maximum values.
8. unique(): Returns an array of unique values in a column of the DataFrame, useful for identifying distinct categories or groups in categorical data.

Methodology:

1. **Data Collection and Exploration:**
    - Collect Data: Obtain a relevant dataset ensuring it contains key features.
    - Explore Data: Load the dataset into a Pandas DataFrame and analyze its structure, including the number of samples, features, data types, and any missing or erroneous values.
2. **Data Preprocessing:**
    - Handle Missing Values: Identify and manage missing values using techniques such as imputation or removal.
    - Data Cleaning: Remove duplicates, correct erroneous entries, and ensure consistency in data formatting.
3. **Feature Engineering:**
    - Feature Selection: Select relevant features using domain knowledge and statistical techniques.
    - Feature Encoding: Convert categorical variables into numerical format using one-hot encoding or label encoding for better processing.

Advantages:

1. Pandas is an easy-to-use library, making it widely popular.
2. It provides powerful data structures like Series and DataFrame.
3. It offers extensive functionality for data manipulation.

Disadvantages:

1. Pandas can consume significant memory when working with large datasets.
2. It is highly integrated with the Python ecosystem, limiting interoperability with other programming languages.

Conclusion:

In summary, this assignment provided an introduction to the Pandas library, a crucial tool for data manipulation and analysis in Python. We explored its basic functions, such as reading various data formats, organizing and describing data, and handling missing values. Through practical exercises, we gained a better understanding of how Pandas can simplify complex data tasks, making data analysis more accessible and efficient. These foundational skills with Pandas will serve as a strong base for more advanced data analysis projects in the future.