Page No.

Date :

Assignment No. 1

Problem Statement

Data Handling - Locate any open source data. Load data into data frame. Perform dataframe operations, perform basic statistical operations like mean, median, standard deviation etc.

Objectives

1. To explore various data sources & data repositories.
2. To explore the operations on a dataset file using df with basic statistical operations in Python.

Theory

1. Identify and study various data sources (e.g. "IRIS, Features of Dataset)

- Public

  Freely available datasets from various public platforms like Kaggle, UCI Machine Learning Repository, and government portals. E.g. The IRIS dataset is publicly available and commonly used for learning and testing algorithms.

- Private

  Data that is proprietary or confidential, often owned by organizations. Access to such data might require permission, and its usage is usually governed by legal agreements.

- Government

  Data released by government agencies and institutions, often available through portals like data.gov. This data is typically used for policy-making, research, and public services.

2. To study a dataset with various operations using Python.

- Load dataset

  Import necessary libraries like pandas and load the dataset.

- Reading CSV files.

```
df = pd.read_csv("file.csv")
```

- Display:
  - Head

    Display the first few rows of the dataset using 'df.head()'.

  - Tail

    Display the last few rows of the dataset using 'df.tail()'.

  - Shape

    The no. of rows and columns using 'df.shape'.

  - Describe

    Use 'df.describe()' to get a statistical summary of numerical columns, including count, mean, standard deviation, min, and max.

  - Summary

    A general summary can be generated using 'df.info()' to get an overview of the data types and non-null counts.

- Handling
  - Remove repeated observations (duplicates)

    Identify and remove duplicate rows using 'df.drop_duplicates()'.

  - Identify and display missing values.

    Use 'df.isnull().sum()' to find the count of missing values in each column.

  - Statistical basic operations to handle missing values

    Fill missing values with a statistical value like mean, median, or mode using 'df.fillna(df.mean())'.

Drop rows with missing values using 'df.dropna()'.

CONCLUSION

Basic operations were performed on the .csv data file using Python.

FAQs

1. State the significance of handling missing values in a dataset.

- Data Integrity: Missing values can lead to inaccurate or biased results in your analysis, as they can distort the overall data distribution.

- Algorithm Performance: Many ML algorithms cannot handle missing data, or their performance may degrade if missing values are present.

- Consistency: Inconsistent handling of missing values can introduce errors and inconsistencies in your analysis, leading to unreliable conclusions.

- Statistical Analysis: Properly handling missing data ensures that statistical analysis like mean, median, and correlations are accurate and representative of the actual dataset.

2. Explain the central tendency measures with examples.

They are statistical measures that represent the center or typical value of a dataset.

i. Mean - The arithmetic average of a set of numbers.
E.g. for [2, 3, 4, 5, 6], the mean is 4.

ii. Median - The middle value in a dataset when the numbers are arranged in ascending or descending order.
E.g. for [3, 5, 7, 9, 11], the median is 7.

iii. Mode - The value that appears most frequently in a dataset.
E.g. for [1, 2, 2, 3, 4], the mode is 2.

3. Describe various methods to handle missing values in a dataset.

i. Deletion

- Remove entire rows containing missing values.
- Analyze datasets w/o any missing values.

ii. Imputation

- Replace with mean, median, or mode.
- Replace with preceding or following value.
- Interpolate (estimate based on nearby values)
- KNN

4. Explain different data types.

i. Numeric

- Int - Whole numbers w/o decimal points. e.g. 5, -1, 0
- Float - Numbers w/ fractional parts. e.g. 3.14, 22.3

ii. Boolean

Represents binary values: True/False, or 0/1.

iii. String

A sequence of characters. E.g. "123ABCxyz", "Hello!".

iv. Date / Time

Represents dates, times, or timestamps. E.g. 2023-08-09, 14:30:00.