18B1WCI674: MACHINE LEARNING LAB
Assignment-2 (Pandas)
Jan 14, 2026

Instructions

1. Use Jupyter Notebook or Google Colab to solve the assignment.

2. Use only the **Pandas** library for data manipulation tasks.

3. Avoid explicit Python loops unless explicitly mentioned.

4. Use any suitable CSV dataset (e.g., students.csv, sales.csv).

5. Submit the notebook within the allotted lab hours.

Overview of Pandas

Pandas is a powerful Python library designed for ***data manipulation, data analysis, and data preprocessing***. It provides high-level data structures such as **Series** and **DataFrame**, which are essential for handling structured and tabular data in machine learning workflows.

Part A: Pandas Basics

1. Install Pandas before starting the experiment.

```
pip install pandas
```

2. Import the Pandas library using standard convention:

```
import pandas as pd
```

3. All indexing, filtering, aggregation, and transformation operations must be performed using Pandas functions only.

1. Series & DataFrame Creation

1. Create a Pandas Series from a list of integers:

```
data = [10, 20, 30, 40, 50]
s = pd.Series(data)
print(s)
```

2. Create a DataFrame from a dictionary:

```
data = {
    "Name": ["Alice", "Bob", "Charlie"],
    "Age": [23, 25, 22],
    "City": ["Delhi", "Mumbai", "Chennai"]
}

df = pd.DataFrame(data)
```

3. Create a DataFrame with custom row and column labels:

```
data = [[85, 90, 88],
        [78, 82, 80],
        [92, 95, 94]]

df = pd.DataFrame(
    data,
    index=["Student1", "Student2", "Student3"],
    columns=["Maths", "Science", "English"]
)
```

4. Read a CSV file into a DataFrame:

    —

5. Display the first five rows of the DataFrame:

    —

6. Display the last five rows of the DataFrame:

    —

7. Find the shape of the DataFrame:

    —

8. Display column names:

    —

9. Display basic information of the DataFrame:
   —

10. Generate summary statistics:
    —

2. Data Selection & Indexing

1. Select a single column:
   —

2. Select multiple columns:
   —

3. Select rows using `loc`:
   —

4. Select rows using `iloc`:
   —

5. Filter rows based on a condition:
   —

6. Select rows where values are greater than the column mean:
   —

7. Access a specific cell value:
   —

8. Set a column as index:
   —

9. Reset the index:
   —

10. Rename columns:
    —

3. Data Cleaning & Manipulation

1. Detect missing values:

    —

2. Replace missing values with column mean:

    —

3. Drop rows with missing values:

    —

4. Replace negative values with zero:

    —

5. Remove duplicate rows:

    —

6. Sort the DataFrame by a column:

    —

7. Apply a function to a column:

    —

8. Create a new column from existing columns:

    —

9. Change data type of a column:

    —

10. Drop a column:

    —

4. Statistical Operations

1. Compute the mean of a column:

    —

2. Compute median and standard deviation:

    —

3. Find maximum and minimum values:

    —

4. Compute correlation between numerical columns:

    —

5. Count unique values in a column:

    —

6. Compute cumulative sum:

    —

7. Display value counts of a categorical column:

    —

8. Normalize a column using Min–Max normalization:

    —

9. Group data by a column and compute mean:

    —

10. Create a pivot table:

    —

Part B: Advanced Pandas

1. Perform group-wise normalization:

    —

2. Apply rolling mean with window size 3:

    —

3. Detect outliers using the 3σ rule:

    —

4. Merge two DataFrames using inner join:

    —

5. Concatenate multiple DataFrames:

    —

6. Apply conditional logic using `np.where()`:

    —

7. Rank values within each group:

    —

8. Use `apply()` with a lambda function:

    —

9. Resample time-series data:

   —

10. Detect duplicate records based on multiple columns:

   —