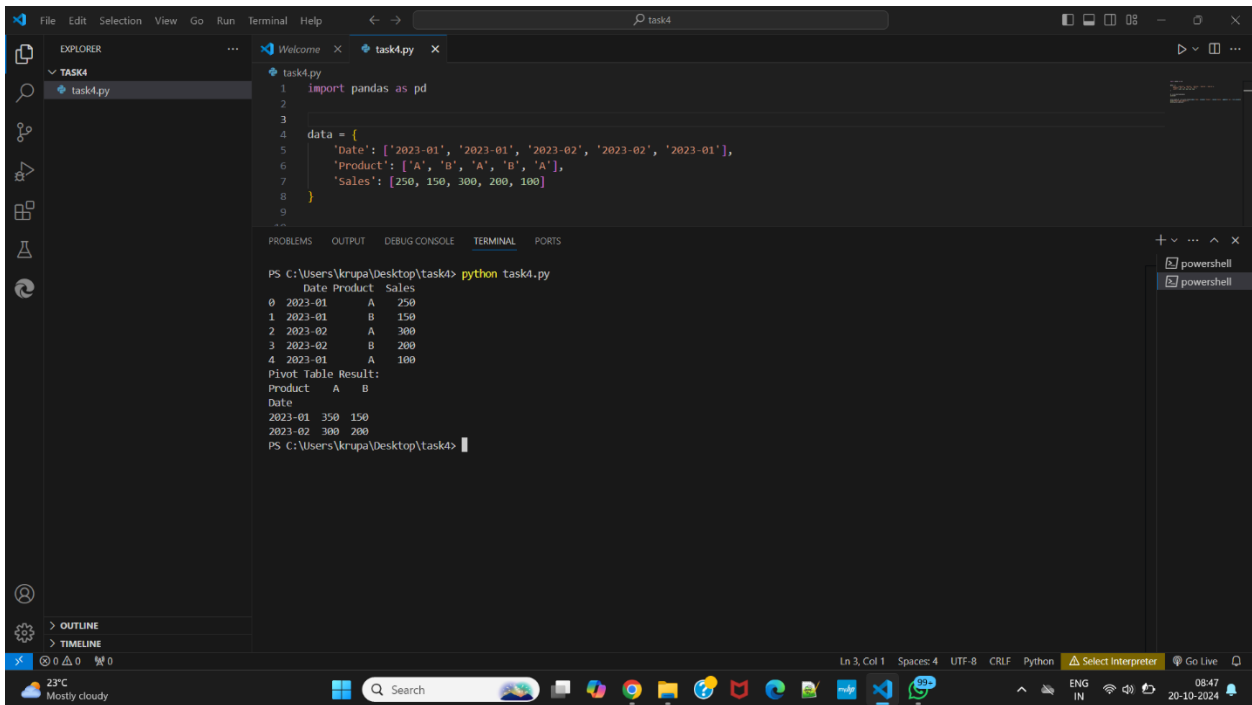


1. Task Description

Task Data manipulation

Use pandas' pivot and pivot_table functions to perform advanced data reshaping operations

2. Task Output Screenshot



The screenshot shows a VS Code editor with a file named `task4.py` open. The code in the file is as follows:

```
1 import pandas as pd
2
3
4 data = {
5     'Date': ['2023-01', '2023-01', '2023-02', '2023-02', '2023-01'],
6     'Product': ['A', 'B', 'A', 'B', 'A'],
7     'Sales': [250, 150, 300, 200, 100]
8 }
9
```

The terminal output shows the execution of the script, displaying the data as a table and the result of a pivot table operation:

```
PS C:\Users\krupa\Desktop\task4> python task4.py
   Date Product  Sales
0 2023-01     A    250
1 2023-01     B    150
2 2023-02     A    300
3 2023-02     B    200
4 2023-01     A    100
Pivot Table Result:
Product A  B
Date
2023-01 350 150
2023-02 300 200
PS C:\Users\krupa\Desktop\task4>
```

3. Algorithms Used In Task :

1) Data Aggregation Algorithms:

Mean/Median Calculation: When using `pivot_table()`, the most common operation is calculating the mean or median of groups. Pandas internally runs an aggregation algorithm to compute these statistics across different groups.

Sum: Summing values in a grouped dataset is another common operation, which is performed efficiently using internal sum algorithms.

2) Sorting Algorithms:

Sorting Columns and Index: When reshaping the dataset using `pivot()`, pandas sorts the unique values of the columns to place them as headers. This uses sorting algorithms internally

3) GroupBy Operations:

- **GroupBy Algorithm:** Pandas implements an efficient algorithm to group rows based on certain keys (columns). This is a key step in both `pivot()` and `pivot_table()` when calculating summary statistics for each group of rows.

4) Sparse Data Handling:

- **Handling Missing Values:** When using `pivot()` or `pivot_table()`, pandas will fill in missing values (e.g., where no data exists for a particular combination of index and column). You can specify how missing values are handled using `fill_value`.

5. Memory Optimization Techniques for Big Datasets:

- **Chunked Processing:** For very large datasets, pandas uses algorithms that allow loading and processing data in chunks rather than all at once, optimizing memory usage.
- **Data Type Conversion:** Converting data types to more memory-efficient formats