**Lab Exercise 7- Implementing Data Transformations from Multiple CSV Files Using Pandas in Metaflow**

**Objective:**

Learn how to create a Metaflow pipeline that reads data from multiple CSV files, merges them, performs data transformations, and outputs the results.

**Step 1: Set Up Your Environment**

Ensure you have Metaflow and pandas installed. If not, you can install them using pip:

pip install metaflow pandas

**Step 2: Prepare Your CSV Files**

Create two CSV files named products.csv and sales.csv.

**File: products.csv**

product\_id,product\_name,category

1,Widget A,Category 1

2,Widget B,Category 1

3,Widget C,Category 2

4,Widget D,Category 2

**File: sales.csv**

sale\_id,product\_id,sale\_amount,sale\_date

101,1,150,2023-01-01

102,2,200,2023-01-02

103,1,100,2023-01-03

104,3,300,2023-01-01

105,4,400,2023-01-02

**Step 3: Create the Data Transformation Flow**

Create a Python file named MultiCSVDataTransformationFlow.py and add the following code:

import pandas as pd

from metaflow import FlowSpec, step

class MultiCSVDataTransformationFlow(FlowSpec):

@step

def start(self):

"""

Start step: Load data from multiple CSV files.

"""

self.products = pd.read\_csv('products.csv')

self.sales = pd.read\_csv('sales.csv')

print("Products Data:")

print(self.products)

print("\nSales Data:")

print(self.sales)

self.next(self.merge\_data)

@step

def merge\_data(self):

"""

Merge the products and sales data on product\_id.

"""

self.merged\_data = pd.merge(self.sales, self.products, on='product\_id')

print("Merged Data:")

print(self.merged\_data)

self.next(self.clean\_data)

@step

def clean\_data(self):

"""

Clean the merged data by removing duplicates and NaN values.

"""

self.cleaned\_data = self.merged\_data.drop\_duplicates().dropna().reset\_index(drop=True)

print("Cleaned Data:")

print(self.cleaned\_data)

self.next(self.feature\_engineering)

@step

def feature\_engineering(self):

"""

Create new features based on the cleaned data.

"""

self.cleaned\_data['total\_sales'] = self.cleaned\_data['sale\_amount']

self.cleaned\_data['sale\_date'] = pd.to\_datetime(self.cleaned\_data['sale\_date'])

self.cleaned\_data['year'] = self.cleaned\_data['sale\_date'].dt.year

self.cleaned\_data['month'] = self.cleaned\_data['sale\_date'].dt.month

print("Features created:")

print(self.cleaned\_data)

self.next(self.end)

@step

def end(self):

"""

End step: Final output.

"""

print("Data transformation flow completed.")

print("Final transformed data:")

print(self.cleaned\_data)

if \_\_name\_\_ == "\_\_main\_\_":

MultiCSVDataTransformationFlow()

**Explanation of the Flow:**

* **start step**: Loads the data from products.csv and sales.csv using pandas and prints the raw data.
* **merge\_data step**: Merges the products and sales data on the product\_id.
* **clean\_data step**: Cleans the merged data by removing duplicates and NaN values.
* **feature\_engineering step**: Creates new features, such as total sales, year, and month extracted from the sale date.
* **end step**: Finalizes the flow and prints the transformed data.

**Step 4: Run the Flow**

Run the flow using the following command:

python MultiCSVDataTransformationFlow.py run

You should see output for each step, displaying the products data, sales data, merged data, cleaned data, and final transformed data.

**Step 5: Modify the Flow**

To further enhance the flow, you can add more transformations or aggregate the data. For example, you could aggregate sales by category. Update the feature\_engineering step:

@step

def feature\_engineering(self):

"""

Aggregate sales by product category.

"""

self.category\_sales = self.cleaned\_data.groupby('category')['total\_sales'].sum().reset\_index()

print("Total sales by category:")

print(self.category\_sales)

self.next(self.end)

**Step 6: Rerun the Flow**

After modifying the flow, rerun it to see the new transformations:

python MultiCSVDataTransformationFlow.py run

**Step 7: Check Your Output**

You should see the updated output with total sales aggregated by product category.

**Conclusion**

In this exercise, you learned how to:

* Create a Metaflow pipeline that reads data from multiple CSV files using pandas.
* Merge datasets, perform cleaning, and create new features.
* Aggregate data and output the final results.