ASSIGNMENT_2:Enhanced Exercise

Project Folder Structure

```
RetailSalesPipeline/
|
├── data/
|   └── sales_data.csv
|
├── scripts/
|   └── sales_data_pipeline.py
|
├── requirements.txt
├── azure-pipelines.yml
```

Folder/File	Description
data/	Stores the input and processed sales data files.
scripts/	Contains the Python script for cleaning + enrichment + upload.
requirements.txt	List of all dependencies required for pipeline execution.
azure-pipelines.yml	YAML configuration for Azure DevOps CI/CD pipeline.

Step 1 — Install Required Libraries

Add the following dependencies to **requirements.txt**:

pandas

azure-storage-blob

Install them locally:

pip install -r requirements.txt

Step 2 — Create sales_data_pipeline.py

This script performs **data cleaning, enrichment, and uploads files to Azure Blob Storage**.

Key Steps in the Script

Step	Task	Description
1	Load CSV	Read the dataset into a Pandas DataFrame.
2	Remove Duplicates	Remove duplicate order_id records.
3	Handle Missing Data	Replace missing region with "Unknown" and missing revenue with 0.
4	Add Calculated Column	profit_margin = (revenue - cost) / revenue.
5	Customer Segmentation	Categorize into **Platinum**, **Gold**, and **Standard** based on revenue.
6	Save Output Files	Save **raw_sales_data.csv** and **processed_sales_data.csv** in data/.
7	Upload to Azure Blob	Use Azure Storage SDK to upload both files.

Step 3 — Set Up Azure Storage Variables

In **Azure DevOps** → Go to:
Pipeline → Variables → Add the following:

Variable Name	**Value**
AZURE_STORAGE_ACCOUNT_NAME	storage account name
AZURE_STORAGE_ACCOUNT_KEY	account access key
AZURE_CONTAINER_NAME	Target container name

Step 4 — Create azure-pipelines.yml

Testing the Pipeline & Viewing Artifacts

Once your pipeline is created:

Step 1: Run the Pipeline

- Go to **Pipelines → New Pipeline**.
- Select **GitHub / Azure Repos**.
- Connect the repository.
- Choose the azure-pipelines.yml file.
- Click **Run Pipeline**.

Step 2: Validate the Output

After the pipeline runs successfully:

- Go to **Pipelines → Runs → Artifacts**.
- You should see:
    - raw_sales_data.csv
    - processed_sales_data.csv

Uploading Files to Azure Blob Storage

Since we already integrated the **Azure SDK**, the cleaned and raw files are automatically uploaded when the script runs.

Steps to Verify Upload:

1. Go to **Azure Portal → Storage Accounts**.
2. Select your **container**.
3. Verify that both:
    - raw_sales_data.csv
    - processed_sales_data.csv
      are uploaded successfully.

Final Deliverables

Deliverable	**Description**
sales_data_pipeline.py	Python script for cleaning, enrichment, and uploading data.
azure-pipelines.yml	YAML configuration for Azure DevOps pipeline.
requirements.txt	Python dependencies file.