PROJECT

Azure Data Factory Pipeline

Create Pipelines in Azure data factory

I. Build 2 Data Pipeline having many activities.

II. Create Mapping Data Flow involving many transformations**.**

An Azure Blob storage account created and uploaded 4 csv files (from Kaggle.com) with different schema architecture, uploaded to the Blob storage container.

Datasets (csv format)

- Beverage Sales Prediction- 3 columns, follows a simple schema architecture.
- Viral social media trend Analysis Dataset- Simple schema structure having 10 columns.
- Student depression Dataset- Moderate schema structure with 18 columns.
- Crash Analysis system Dataset- complex schema structure having many tables

Pipeline 1: Pipeline with Data Movement and other Activities

In this pipeline:

- Move data from Blob Storage to Azure Data Lake Storage.
- Use other activities like Lookup, Wait.

Step 1: Create the Pipeline

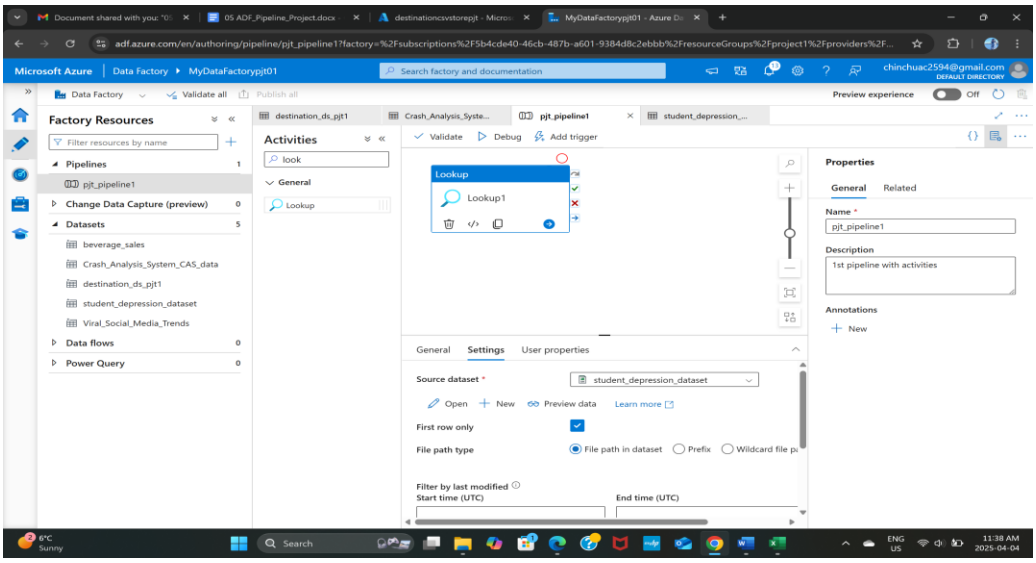
1. Go to Azure Data Factory:
    - Navigate to Azure Data Factory instance in the Azure Portal.
    - Under the Author section, click on + New Pipeline.

Name the pipeline (pjt_pipeline1).

Step 2: Add Activities

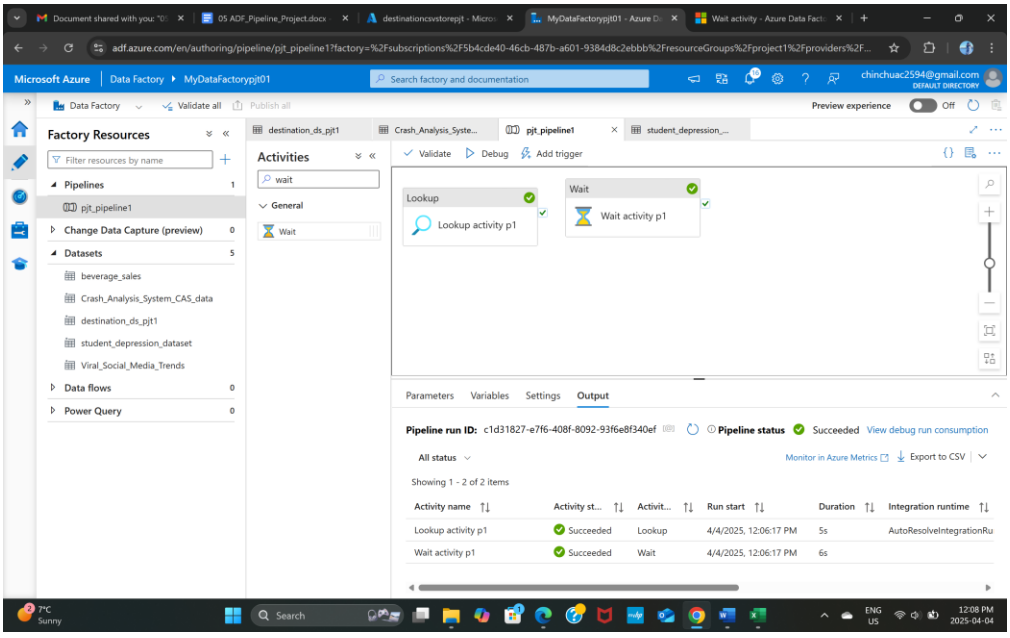
Activity 1: Lookup Activity

- Objective: Fetch metadata or reference data from a source.
- In the Activities pane, drag the Lookup activity onto the canvas.
- Configure the Source: Select a dataset, Blob Storage.
- In the Lookup settings, specified the file path to fetch data.



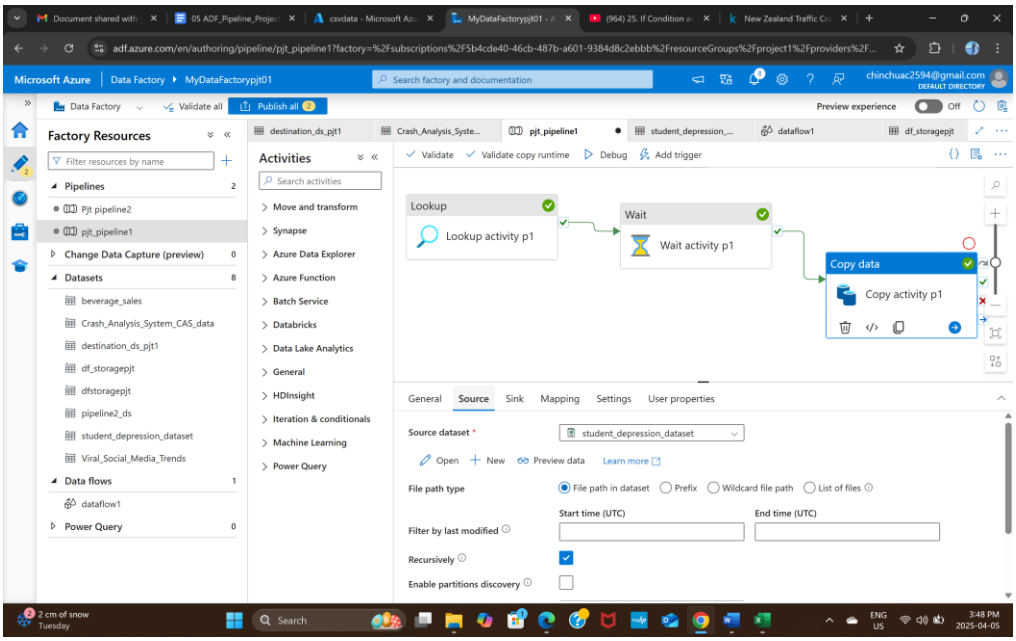
Activity 2: Wait Activity

- Objective: Delay the pipeline execution for a specified time.
- Drag the Wait activity to introduce a delay before continuing with the next activity.
- Set the Wait Time in seconds (3 seconds)



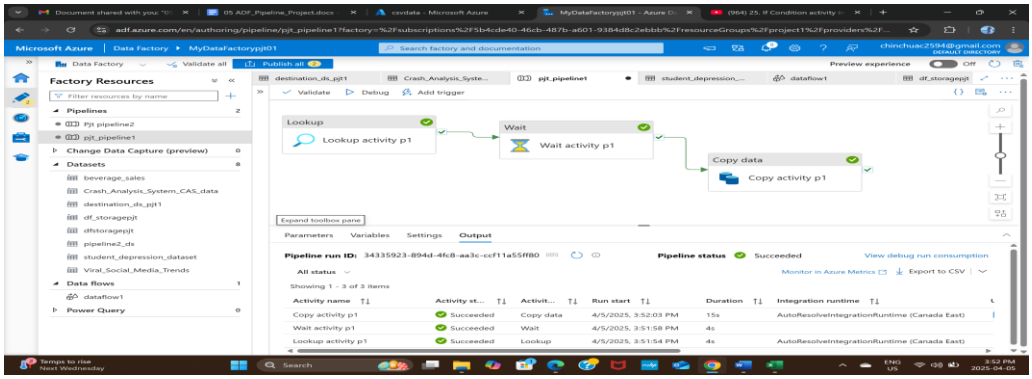
Activity 3: Copy Data Activity

- Objective: Copy data from Blob Storage to a destination Azure Data Lake.
- In the Activities pane, drag the Copy Data activity onto the pipeline canvas.
- Configure the Source dataset to Blob Storage (student_depression_dataset) csv format.
- Configure the Sink to Azure Data Lake (destination_ds_pjt) to store the copied data.



Step 3: Debug and Publish

- Debug the pipeline by clicking on Debug to test the execution.
- After successful debugging, Publish the pipeline.



Pipeline 2: Pipeline with Conditional Logic and ForEach Activity

In this pipeline:

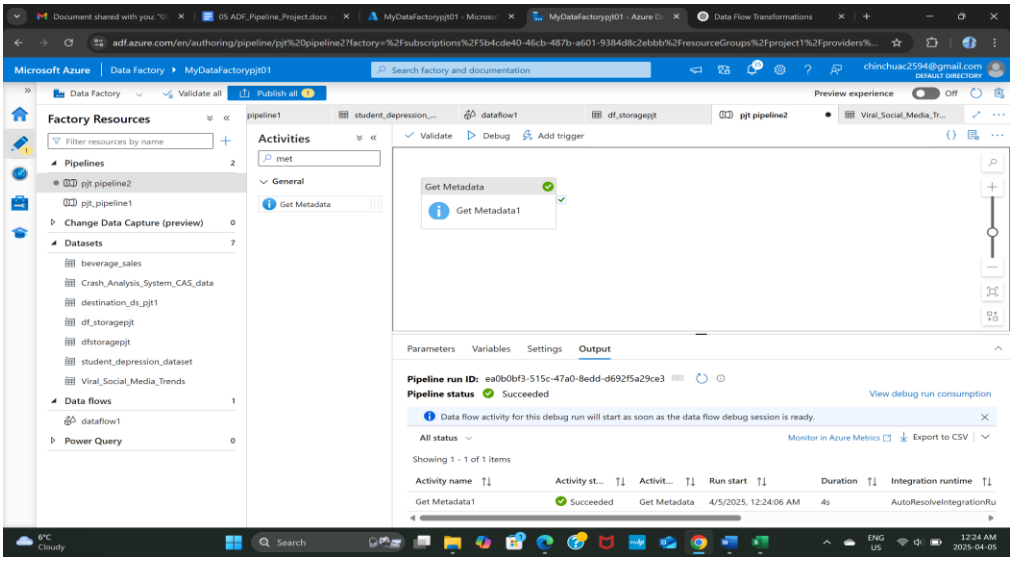
- Process files in Blob Storage using the ForEach activity.
- Perform a copy operation for each file and apply transformations.
- Conditionally process files using the If Condition activity.

Step 1: Create the 2nd Pipeline (project pipeline2)

Step 2: Add Activities

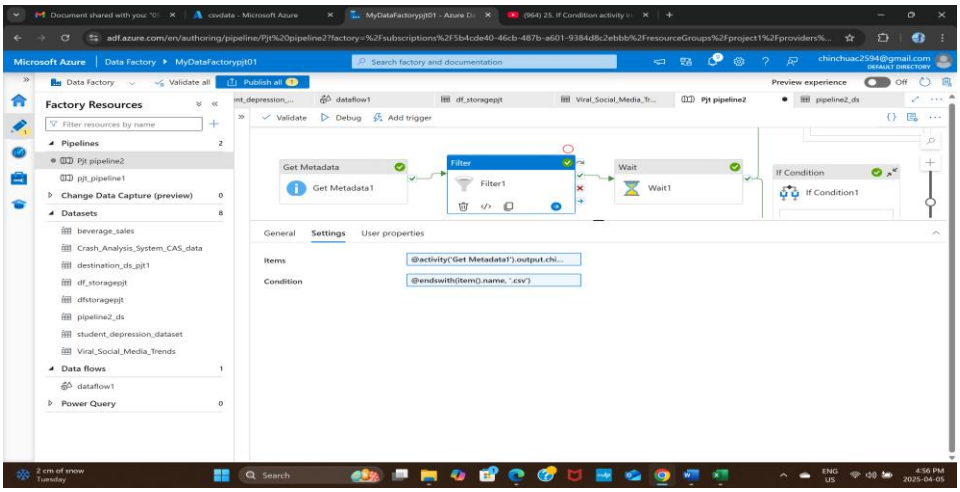
Activity 1: Get Metadata Activity

- Objective: Get metadata about the files stored in a Blob Storage container.
- Drag the Get Metadata activity onto the pipeline canvas.
- Configure the Source to Blob Storage and select a folder (csv files).
- Use the Field list to get metadata about the files, such as File Names and File Size.



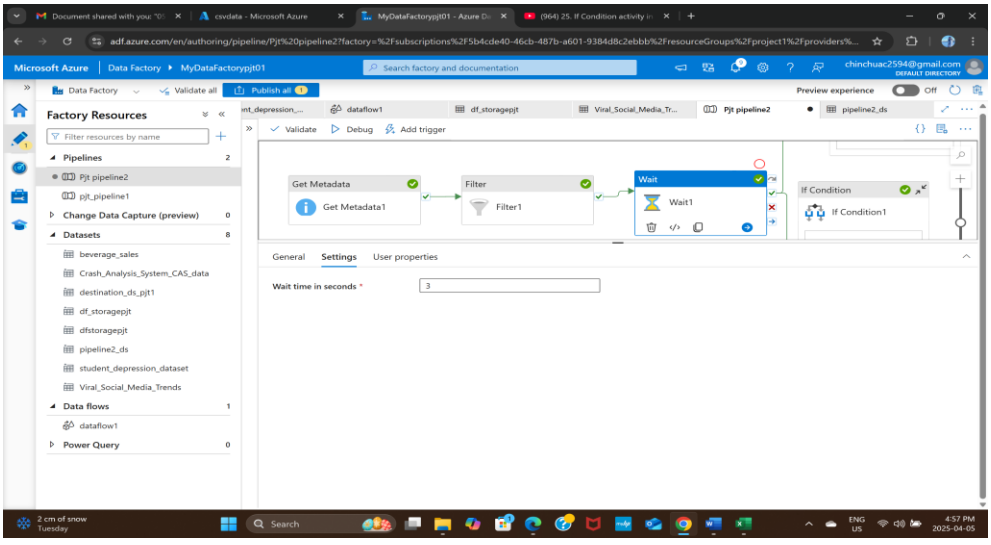
Activity 1: Filter Activity

- Objective: Get filtered data from the output of metadata activity based on condition specified.
- Drag the Filter activity onto the pipeline canvas.
- Configure the settings, write the item and condition.



Activity 3: Wait Activity

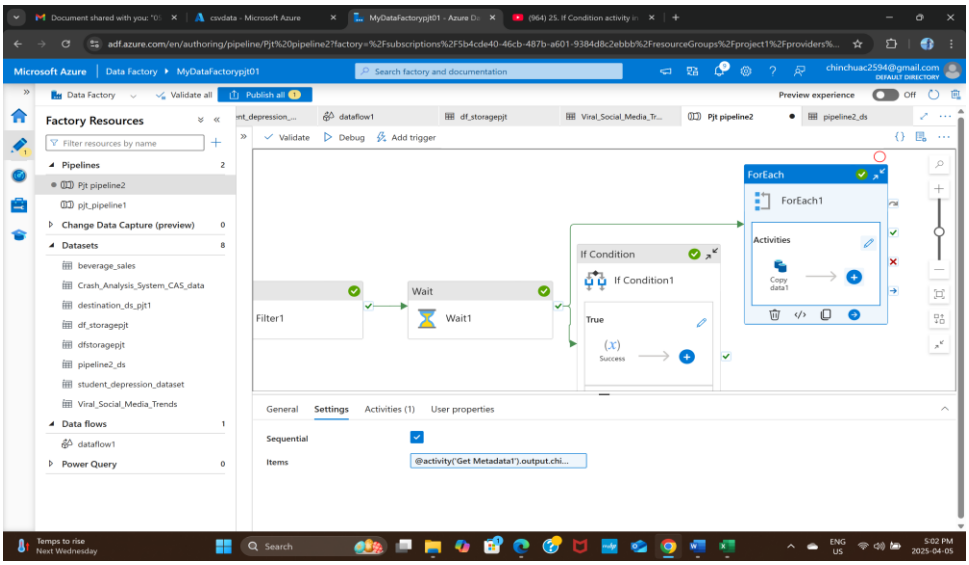
- Objective: Delay the pipeline execution for a specified time.
- Drag the Wait activity to introduce a delay before continuing with the next activity.
- Set the Wait Time in seconds (3 seconds).



Activity 4: ForEach Activity

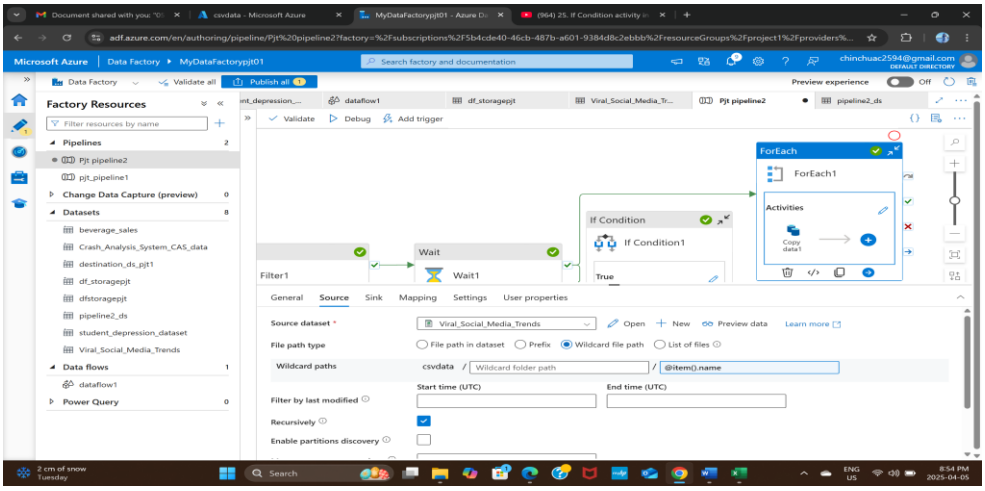
- Objective: Loop over the list of files returned from the Filter activity.

- Drag the ForEach activity onto the pipeline canvas and configure the Items to the output of the Filter activity.



Activity 4.1: Copy Data Activity (Inside ForEach)

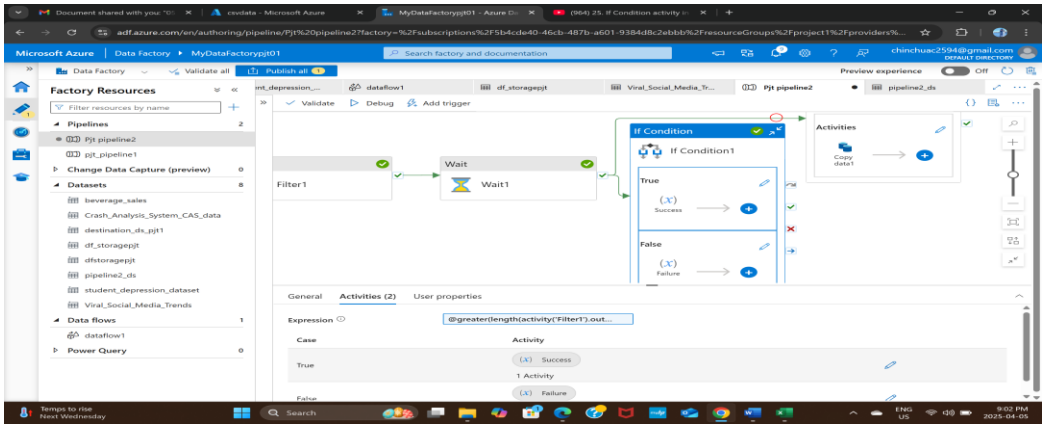
- Objective: Copy each file from Blob Storage to a destination, Azure Data Lake). Inside the ForEach activity, drag the Copy Data activity to copy each file.
- Configure the Source dataset to read each file from Blob Storage.
- Set the Sink dataset, Data Lake where the processed files will be stored



Activity 5: If Condition Activity

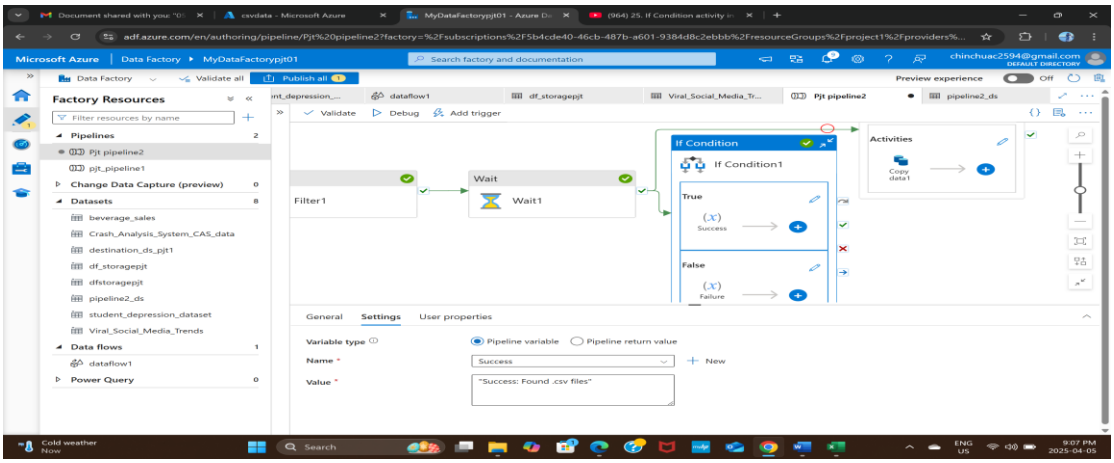
- Objective: Conditionally process files based on filtered data using filter activity.

- Drag the If Condition activity into the pipeline.
- Define the condition to determine which files should be processed.
- Inside the If Condition, add additional activities to perform different actions.



Activity 5.1: Set Variable Activity (inside If condition Activity)

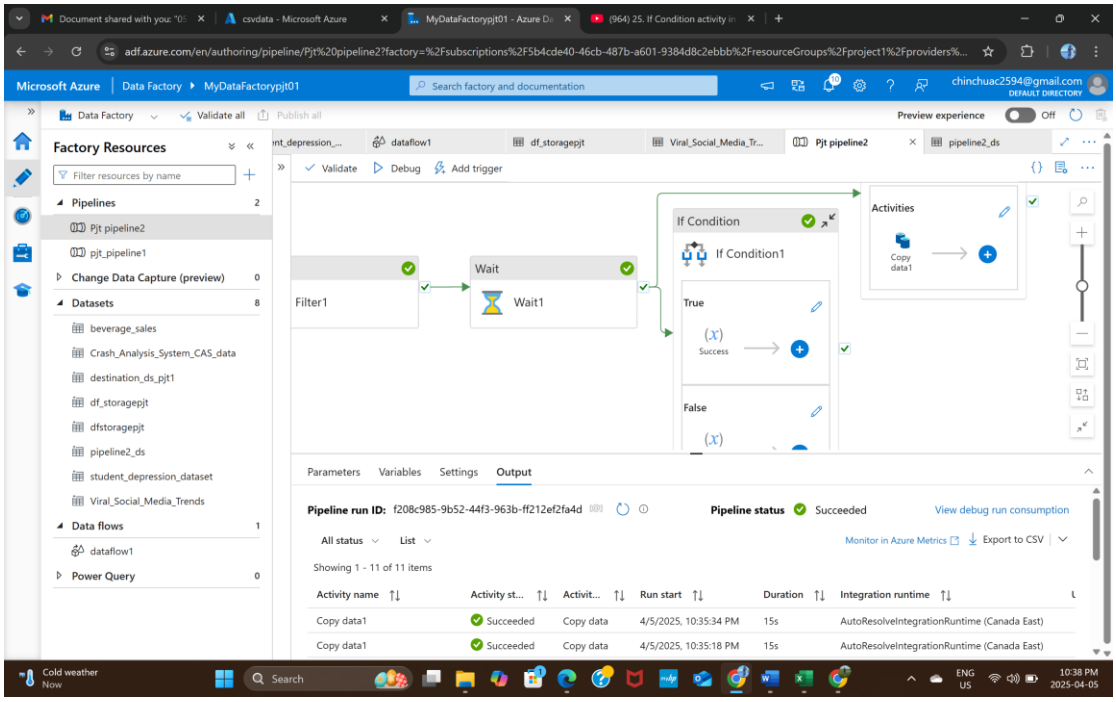
- Objective: To print message based on the if condition, if true show success variable(message) and if False failure variable(message).
- Drag the set variable activities into the If condition Activity.
- Define the variables.



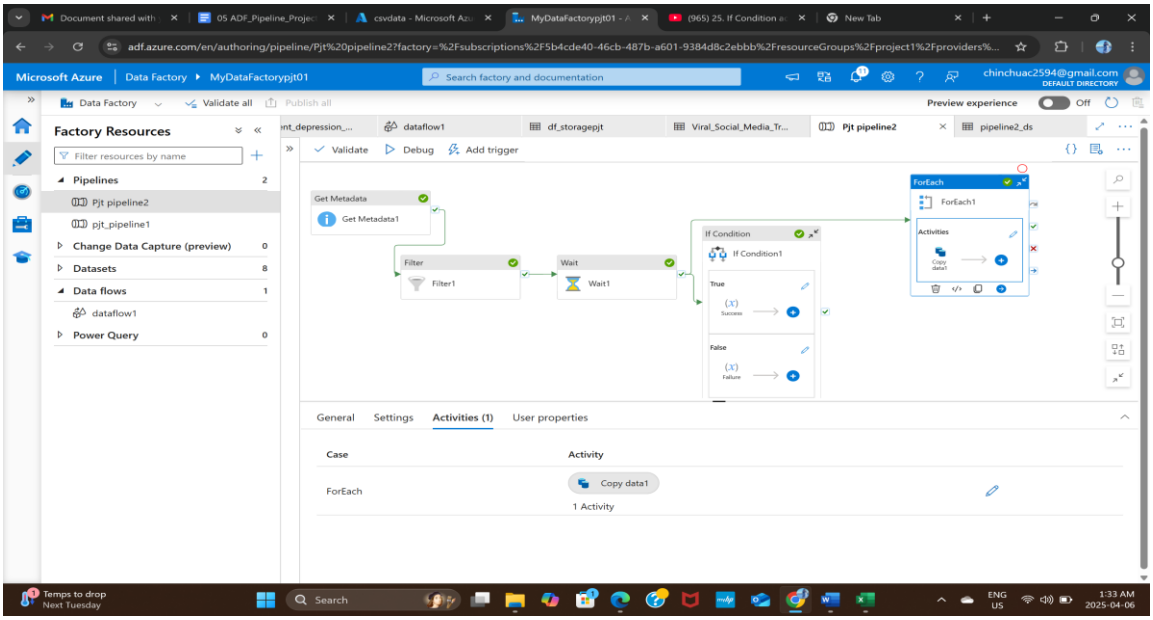
Step 3: Debug and Publish

- Debug the pipeline by clicking Debug to check if the activities run as expected.

- After debugging, click Publish to deploy the pipeline.

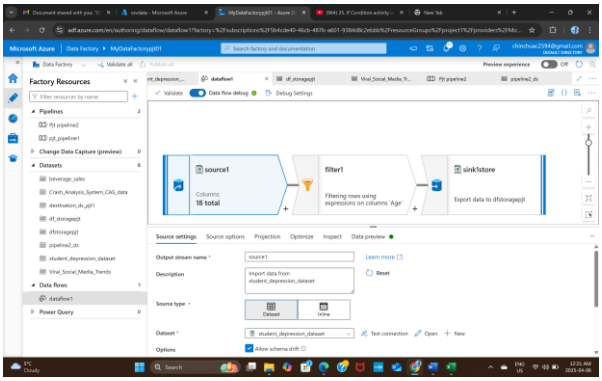


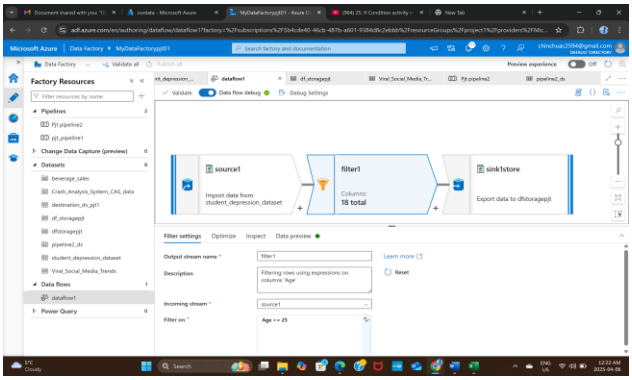
The final Pipeline Looks like:

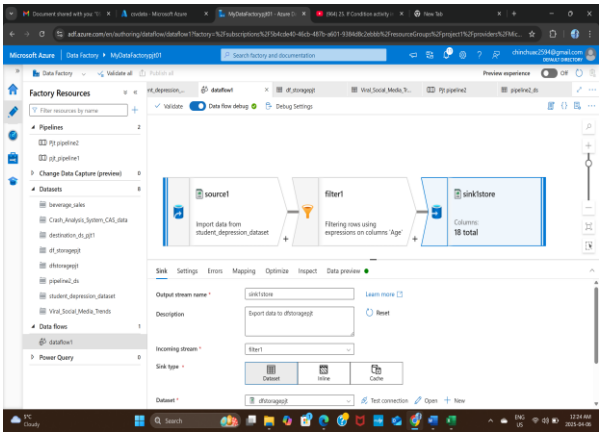


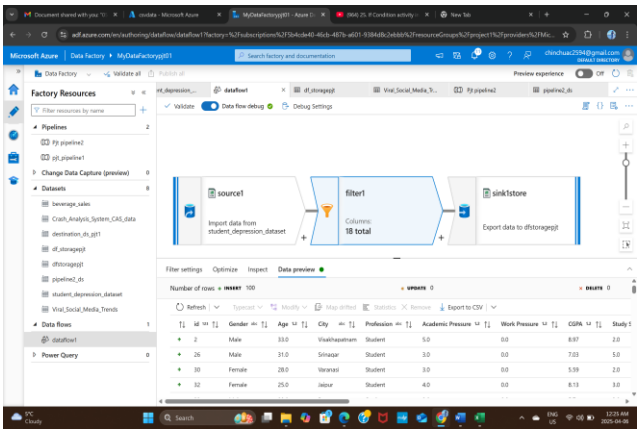
III. **Create Mapping Data Flow involving many transformations.**

- A Dataflow created and, in the source, import a dataset already have and configure the settings.
- Add a Filter transformation activity after the source and configure the Filter settings and write the filter conditions in the ‘Filter on’.
- Store the filtered data result in a sink, for this add a Sink activity and configure the settings by providing incoming stream and storage data set.
- Validate and publish it if no errors.



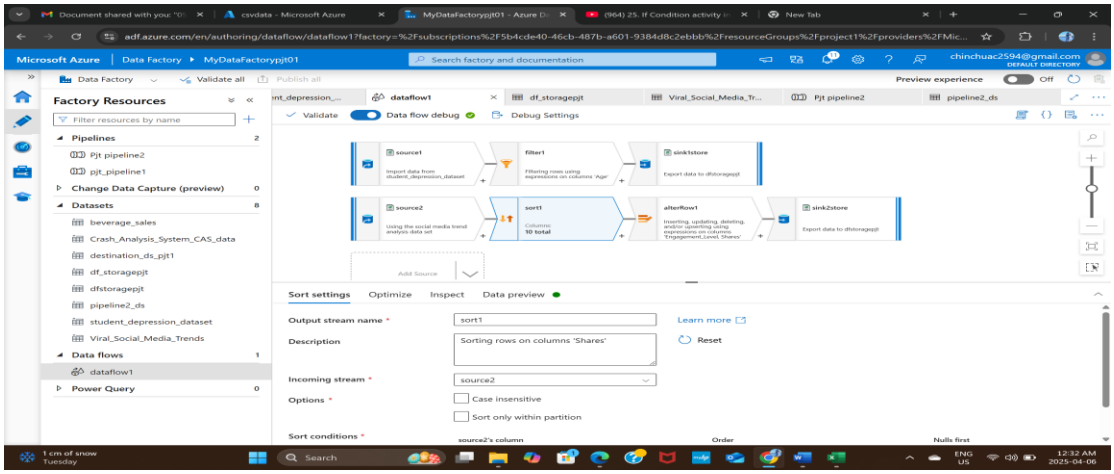






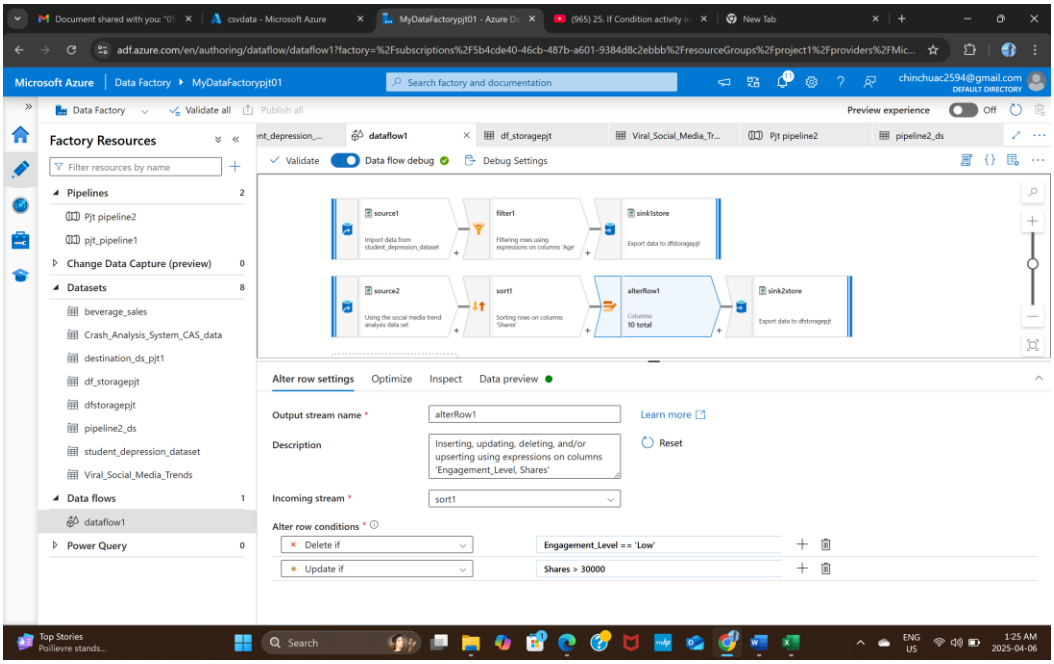
More transformations on the above Dataflow

- Perform some more transformations to the above dataflow such as Sort, alterRow etc. using another source dataset and store the output dataset in a sink.
- Validate the Dataflow and publish.

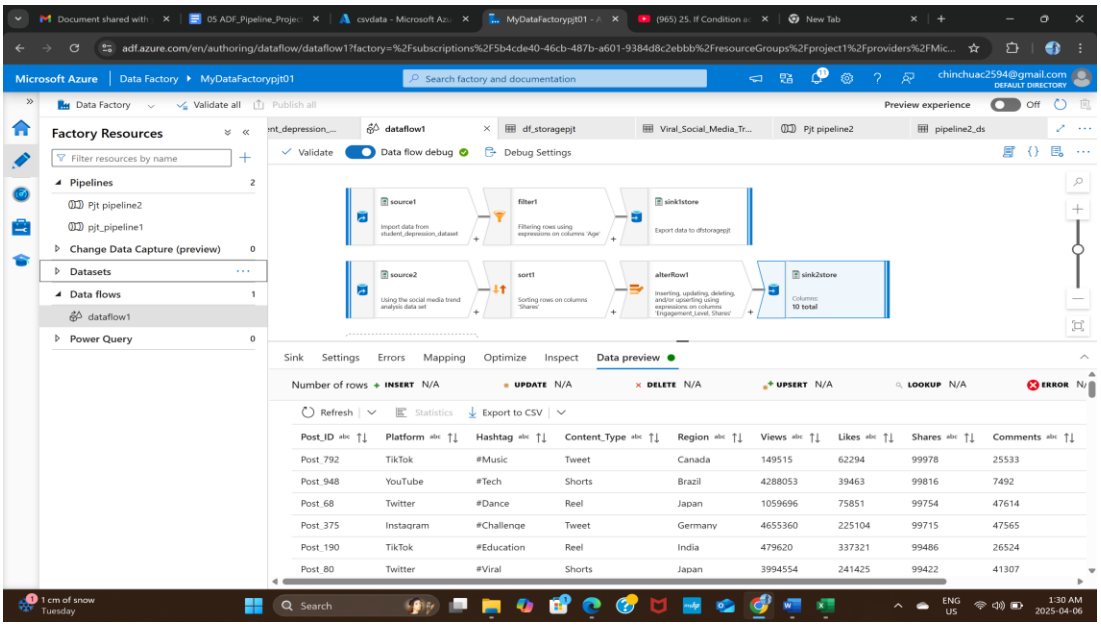


Activity alterRow:

Perform Alter transformation to the input data based on the specified conditions and output result dataset stored in the Sink.



The final Dataflow looks like below and with the Data preview after all the transformations, which is ready to publish.



Challenges Faced during the Project

- To do some operations in the dataset, the selected datasets was not appropriate so changed the selected datasets in some points.
- Stuck with many technical error, syntax errors while writing query conditions and all, referd some online tutorials such as microoft provided to find the solution.
- More errors faced while creating the Dataflow when compared with pipeline.
- Lack of practical knowledge with most of the technologies or operations in activities.
- Building transformations using Mapping Data Flows was difficult to debug.
- Difficulty in reprocessing only failed or partial data.

Time taken to finish the Project

- I took around 2 days to complete the project.
- 3 to 4 hours(starting)- Gone through different videos and self learning topics of Datafactory and pipeline creation.