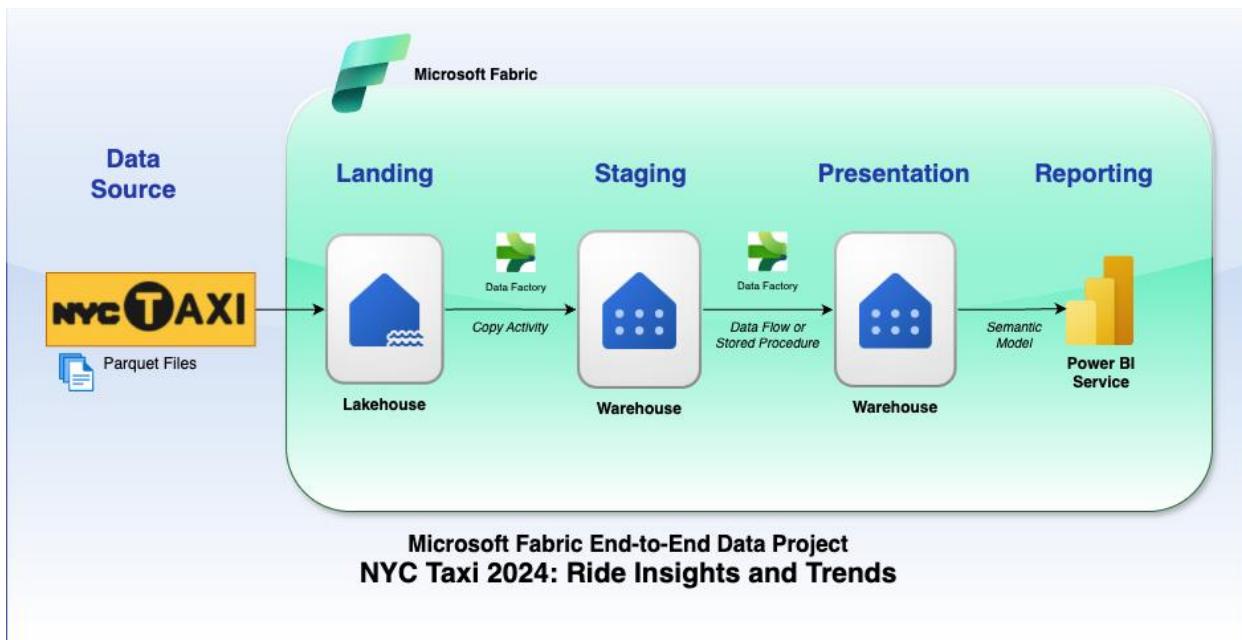


Microsoft Fabric End-to-End Data Engineering Project: NYC Taxi 2024: Ride Insights and Trends

by Randy A. Velasco

Certified Data Engineer

Project Overview



This project leverages **Microsoft Fabric** to build a scalable, automated end-to-end data pipeline for analyzing 2024 NYC Taxi Trip data. The solution ingests raw data, stages it for transformation, prepares it for analysis in a Power BI dashboard, and includes metadata-driven automation for continuous data processing.

Data Sources

- **Taxi Trip Data (Parquet)**
 - Format: Parquet
Frequency: Monthly (Jan - Dec 2024)
Content: Trip records including pickup/drop-off times, passenger count, trip distance, fare, etc.
- **Taxi Zone Lookup Table (CSV)**
 - Contains mapping of location IDs to taxi zones, boroughs, and service zones.

Architecture Layers

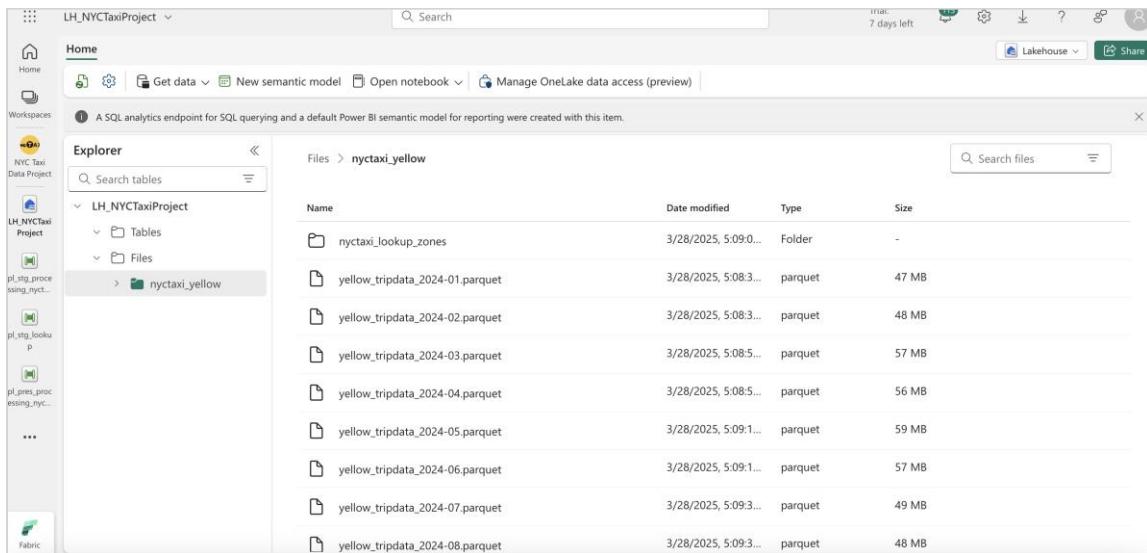
- **Landing Layer: Lakehouse**

Data is uploaded to the **Lakehouse landing zone** in Microsoft Fabric.

Files stored:

- *yellow_tripdata_2024-01.parquet* to *yellow_tripdata_2024-12.parquet*
- *nyctaxi_lookup_zones.csv*

Purpose: Centralized, raw data storage for structured processing downstream.

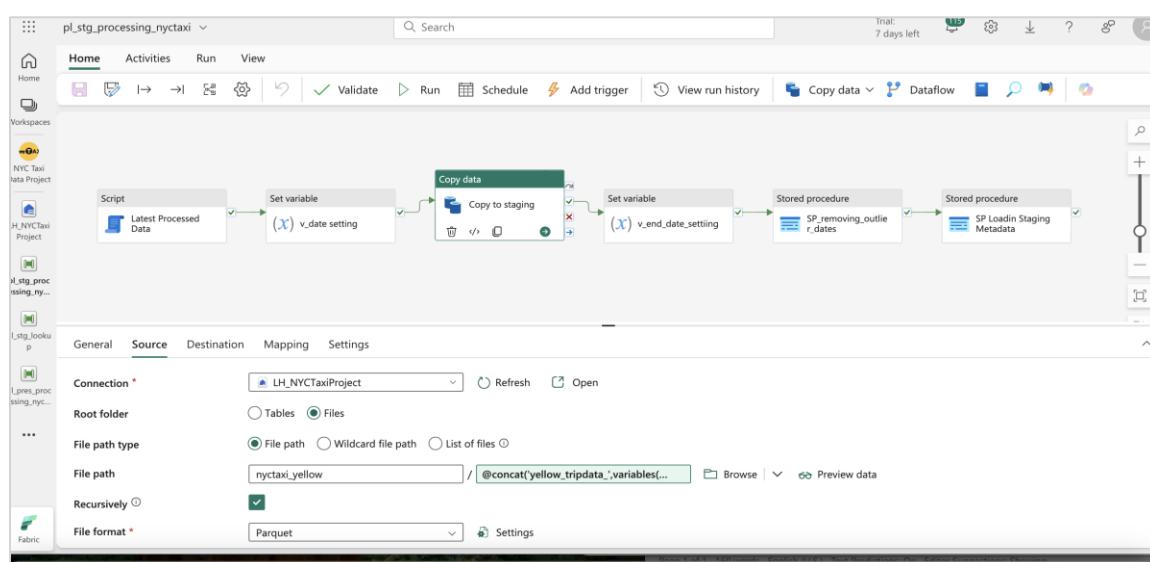


The screenshot shows the Microsoft Fabric Explorer interface. On the left, the navigation pane displays a project structure under 'LH_NYCTaxiProject'. It includes a 'NYC Taxi Data Project' folder containing several sub-folders like 'pl_stg_processing_nyc...', 'pl_stg_lookup_p...', and 'pl_pres_processing_nyc...'. Below these are 'Fabric' and '...' options. The main workspace is titled 'Home' and shows a 'Search' bar at the top. A message indicates there is a SQL analytics endpoint and a Power BI semantic model. The 'Explorer' tab is selected, showing a tree view with 'LH_NYCTaxiProject' expanded, revealing 'Tables' and 'Files' sub-folders. Under 'Tables', there is a single entry: 'nyctaxi_yellow'. Under 'Files', there is a folder named 'nyctaxi_yellow' which contains several parquet files: 'yellow_tripdata_2024-01.parquet', 'yellow_tripdata_2024-02.parquet', 'yellow_tripdata_2024-03.parquet', 'yellow_tripdata_2024-04.parquet', 'yellow_tripdata_2024-05.parquet', 'yellow_tripdata_2024-06.parquet', 'yellow_tripdata_2024-07.parquet', and 'yellow_tripdata_2024-08.parquet'. The table has columns for 'Name', 'Date modified', 'Type', and 'Size'.

Name	Date modified	Type	Size
nyctaxi_lookup_zones	3/28/2025, 5:09:0...	Folder	-
yellow_tripdata_2024-01.parquet	3/28/2025, 5:08:3...	parquet	47 MB
yellow_tripdata_2024-02.parquet	3/28/2025, 5:08:3...	parquet	48 MB
yellow_tripdata_2024-03.parquet	3/28/2025, 5:08:5...	parquet	57 MB
yellow_tripdata_2024-04.parquet	3/28/2025, 5:08:5...	parquet	56 MB
yellow_tripdata_2024-05.parquet	3/28/2025, 5:09:1...	parquet	59 MB
yellow_tripdata_2024-06.parquet	3/28/2025, 5:09:1...	parquet	57 MB
yellow_tripdata_2024-07.parquet	3/28/2025, 5:09:3...	parquet	49 MB
yellow_tripdata_2024-08.parquet	3/28/2025, 5:09:3...	parquet	48 MB

2. Staging Layer: Data Warehouse

- **Pipeline: Load to Staging**
 - **Copy Data Activity:** Loads individual monthly parquet files into the staging table.
 - **Stored Procedure Activity:** Filters rows based on the file's corresponding month using dynamic variables (@start_date, @end_date).
 - **Variable Configuration:** Dynamically sets processing window using metadata table reference.



- **Metadata Table (metadata.processing_log):**
 - Columns: *pipeline_run_id, table_processed, rows_processed, processed_datetime, latest_processed_datetime.*
 - Used to track processed files and determine the next file to load.

The screenshot shows the Databricks interface with the following details:

- Home:** Reporting
- Workspaces:** NYC Taxi Data Project, H_NYCTaxi Project, stg_processing.nyctaxi, stg_lookups, pres_processing.nyctaxi, WH_nyctaxi, ...
- Explorer:** + Warehouses, WH_nyctaxi, Schemas, dbo, INFORMATION_SCHEMA, metadata, Tables, processing
- Table:** processing_log
- Data preview - processing_log:** Showing 1000 rows
- Columns:** pipeline_run_id, table_processed, row_processed, latest_processed_pickup, processed_datetime
- Rows (Sample):**

pipeline_run_id	table_processed	row_processed	latest_processed_pickup	processed_datetime
b8ad71d5-3a59-41a3-bb4f-d564...	dbo.nyctaxi_yellow	41169300	2024-12-31 00:00:00.000000	2025-03-30 08:37:20.126667
bfd145f3-da71-4190-9c67-90fb...	stg.nyctaxi_yellow	3668337	2024-12-31 23:59:58.000000	2025-03-30 08:34:38.443333
70da4414-c211-4868-ab6f-f61c...	dbo.nyctaxi_yellow	37500963	2024-11-30 00:00:00.000000	2025-03-30 08:28:20.626667
c11fec21-fcd2-4d80-88ab-60de...	stg.nyctaxi_yellow	3646319	2024-11-30 23:59:59.000000	2025-03-30 08:25:55.653333
962f9296-83b6-4ae1-9181-cb1d0...	dbo.nyctaxi_yellow	33854644	2024-10-31 00:00:00.000000	2025-03-30 08:21:01.336667
e85f0210-1b85-44ac-904b-17bbd...	stg.nyctaxi_yellow	3833731	2024-10-31 23:59:59.000000	2025-03-30 08:19:08.193333
a35bf449-e76a-4b80-991c-75ebd...	dbo.nyctaxi_yellow	30020913	2024-09-30 00:00:00.000000	2025-03-30 08:08:19.946667
eeb874a2-c08-452d-8224-f524...	stg.nyctaxi_yellow	3632981	2024-09-30 23:59:59.000000	2025-03-30 08:05:42.083333
f6fb70e-5b1e-43fb-9ff4-c96a0b...	dbo.nyctaxi_yellow	26387932	2024-08-31 00:00:00.000000	2025-03-30 07:59:10.553333
8d7c55b7-7c1d-4289-96d3-ce37...	stg.nyctaxi_yellow	2979132	2024-08-31 23:59:59.000000	2025-03-30 07:57:34.576667
0cb6a833-8dad-4e35-836e-cb2c...	dbo.nyctaxi_yellow	23408800	2024-07-31 00:00:00.000000	2025-03-30 07:53:08.860000
5d37423e-13cc-408c-a279-b26ea...	stg.nyctaxi_yellow	3076856	2024-07-31 23:59:59.000000	2025-03-30 07:51:12.270000
c28ae020-d506-41b4-8644-5e82...	dbo.nyctaxi_yellow	20331944	2024-06-30 00:00:00.000000	2025-03-30 07:47:20.300000
0685ad3b-46fe-4515-b37d-0d35...	stg.nyctaxi_yellow	3539142	2024-06-30 23:59:57.000000	2025-03-30 07:44:46.503333
6ea64417-bbf7-4f48-ba32-dc8f...	dbo.nyctaxi_yellow	16792802	2024-05-31 00:00:00.000000	2025-03-30 07:40:27.596667
1c64b586-df3-4a44-831d-1be0f...	stg.nyctaxi_yellow	3723800	2024-05-31 23:59:58.000000	2025-03-30 07:37:44.203333
64663685-5ade-4338-8263-ac36...	dbo.nyctaxi_yellow	13069002	2024-05-30 00:00:00.000000	2025-05-30 07:26:20.756667
- Status:** Succeeded (9 sec 551 ms)
- Free Online Video Downloader - SaveFrom.net**
- Columns:** 5 Rows: 24

3. Presentation Layer: Data Warehouse

- **Table Initialization:**
 - A blank **presentation table** is created to house enriched and final-formatted data.

The screenshot shows the Databricks interface with the following details:

- Home:** Reporting
- Workspaces:** NYC Taxi Data Project, H_NYCTaxi Project, stg_processing.nyctaxi, stg_lookups, pres_processing.nyctaxi, WH_nyctaxi, ...
- Explorer:** + Warehouses, WH_nyctaxi, Schemas, dbo, INFORMATION_SCHEMA, metadata, queryinsights, stg, Tables
- Table:** SQL query 3
- Code:**

```

CREATE TABLE dbo.nyctaxi_yellow
(
    vendor varchar(50),
    tpep_pickup_datetime date,
    tpep_dropoff_datetime date,
    pu_borough varchar(100),
    pu_zone varchar(100),
    do_borough varchar(100),
    do_zone varchar(100),
    payment_method varchar(50),
    passenger_count int,
    trip_distance float,
    total_amount float
);

```
- Preview:** Copilot uses AI. Mistakes can happen. Verify code suggestions before running them. [Review terms](#)

- **Pipeline: Load to Presentation**

- **Dataflow Activity:**

- Reads from staging table
- Performs transformations:
 - Removes unwanted columns
 - Adds new columns (e.g., *vendor*, *payment_method*)
 - Joins with **Taxi Zone Lookup Table** (converted to table in staging)
- Merges output into presentation table for analysis.

The screenshot displays two Microsoft Power BI interfaces: the Data Flow interface at the top and the Pipeline interface below it.

Data Flow Interface (Top):

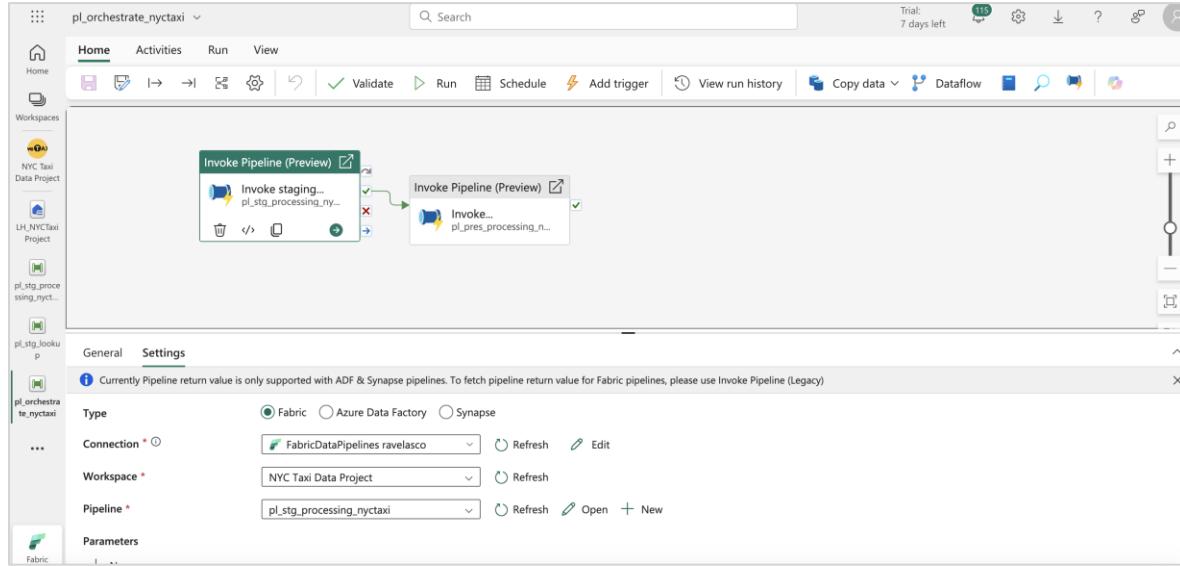
- Title:** DF_pres_processing_nyctaxi
- Home tab:** Shows the flow of data from 'stg nyctaxi_yellow' and 'stg taxi_zone_lookup' through a 'Merge' step and another 'Merge (2)' step, resulting in a query table.
- Properties Panel (Right):**
 - Query settings:** Properties for 'Merge (2)'.
 - Applied steps:** Shows the steps taken: Source, Expanded s..., Renamed c..., Reordered ..., and Removed c... (highlighted).
 - Data destination:** Set to 'Warehouse'.
- Table Preview:** Shows a sample of the merged data with columns: vendor, tpep_pickup_datetime, tpep_dropoff_datetime, pu_borough, pu_zone, do_zone, do_borough, pa...

Pipeline Interface (Bottom):

- Title:** pl_pres_processing_nyctaxi
- Activities Tab:** Shows a 'Dataflow' activity followed by a 'Stored procedure' activity.
- Output Tab:**
 - Pipeline run ID:** 35f22928-2137-4dbd-b914-71d91981fd36
 - Pipeline status:** Succeeded
 - Table:** Shows the history of pipeline runs with columns: Activity name, Activity status, Run start, Duration, Input, and Output.

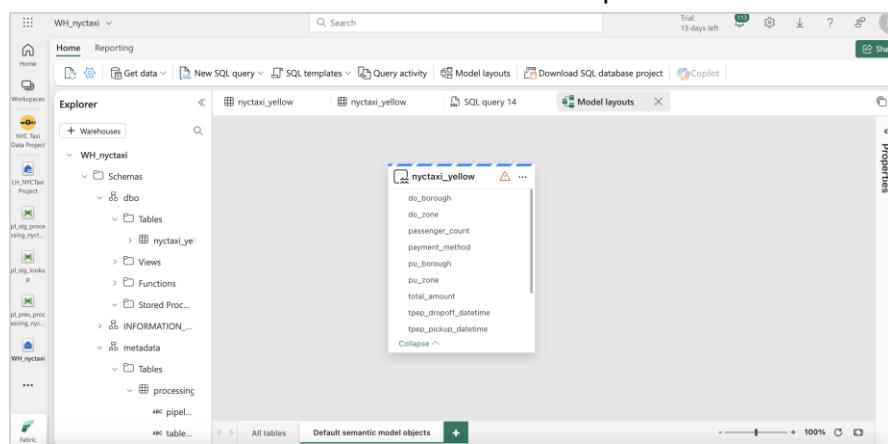
4. Pipeline Orchestration

- A **master pipeline** orchestrates:
 - The **staging pipeline**
 - The **presentation pipeline**



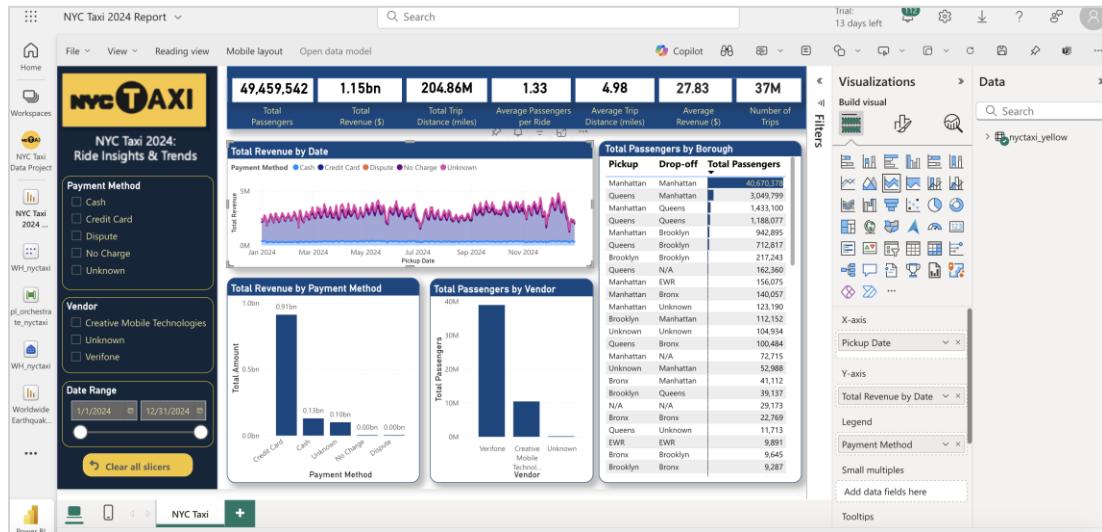
5. Power BI Dashboard

- Built on the **default semantic model** from the presentation table.

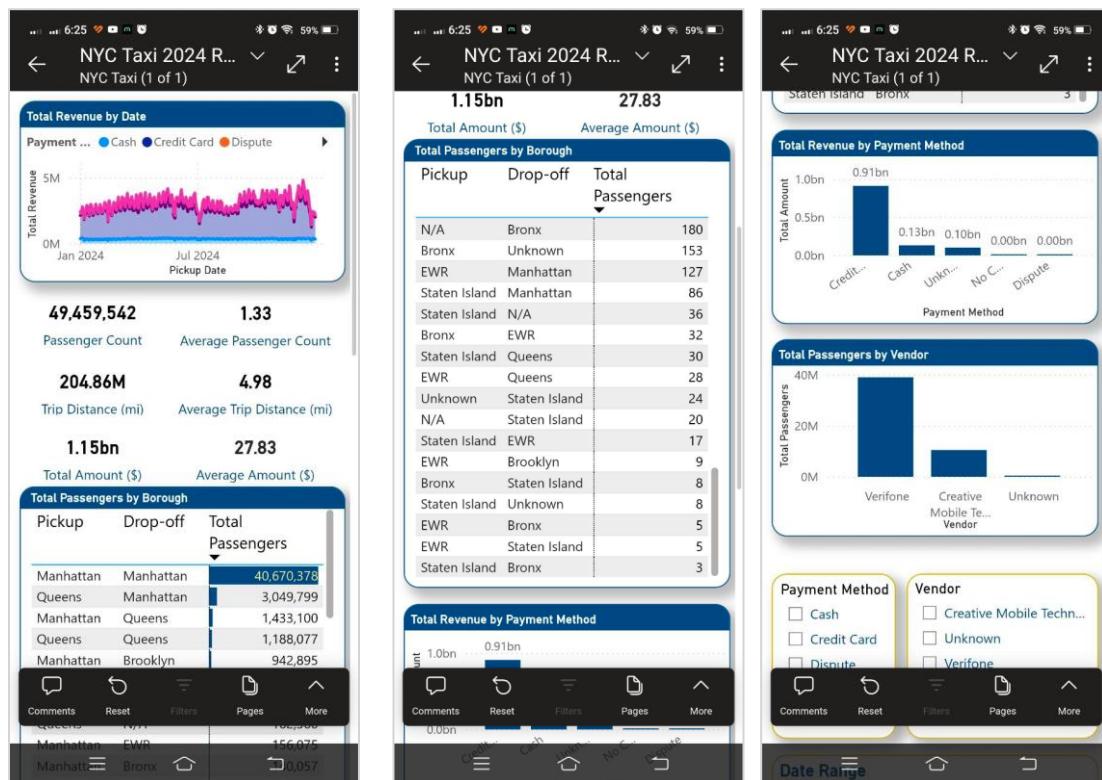


- Metrics visualized:

- Revenue Trends** (monthly/yearly)
- Total and Average Passengers**
- Total and Average Revenue**
- Trip Distance Totals & Averages**



Power BI Report Desktop Version



Power BI Report Mobile Version



Tools and Skills Utilized in Creating This Project

This project leveraged a modern data architecture built on Microsoft Fabric, utilizing a diverse set of tools and skills to ensure efficient, scalable, and automated data processing from ingestion to visualization.

To store and manage raw and curated data, I implemented a Lakehouse architecture within Microsoft Fabric, providing the flexibility of both structured and unstructured data storage. For more structured reporting and querying needs, I also utilized Microsoft Fabric Warehouse, enabling robust data warehousing capabilities.

The ETL pipelines were built using Microsoft Fabric Data Pipelines, which facilitated seamless data movement across systems. Within these pipelines, Dataflows were used to perform data transformation, ensuring clean, structured, and enriched data ready for consumption.

We employed T-SQL stored procedures for advanced scripting requirements, allowing for efficient querying and data manipulation directly within the warehouse environment. For managing complex workflows, pipeline orchestration with variables was implemented, enabling conditional logic and dynamic execution paths throughout the data process.

A custom metadata table was created to handle metadata management, with dynamic logic incorporated to drive scalable and reusable pipeline behaviors. This helped to minimize hardcoding and ensured flexibility as data sources and structures evolved.

For data visualization, Power BI was integrated directly with the semantic model, providing rich, interactive dashboards and reports that reflect real-time data insights. This allowed stakeholders to derive meaningful conclusions and make data-driven decisions.

Lastly, the project emphasized automation and scalability, with dynamic logic built in to support monthly file processing. This ensured that the system could adapt to changing data loads and growing datasets without the need for constant manual intervention.

Overall, the combination of these tools and skills created a powerful, end-to-end data solution that is robust, scalable, and aligned with modern data platform best practices.

How This Project Enhanced My Skills as a Data Engineer

Working on this project has significantly sharpened and expanded my skill set as a Data Engineer, particularly in designing and building modern, scalable data solutions using Microsoft Fabric.

Throughout the project, I gained hands-on experience with **Lakehouse architecture**, allowing me to effectively manage both structured and semi-structured data within a unified platform. This gave me a deeper understanding of how to balance flexibility with performance in data storage strategies.

I also deepened my proficiency in **data warehousing** through the use of **Microsoft Fabric Warehouse**, optimizing query performance and storage efficiency for large-scale data reporting.

Building **ETL pipelines** using **Microsoft Fabric Data Pipelines** taught me how to efficiently orchestrate and automate data movement, transformation, and integration processes. I implemented **Dataflows** for seamless transformation logic, while also enhancing my scripting skills through **T-SQL stored procedures**, solving complex business logic scenarios with precision and clarity.

One of the key skills I developed was **pipeline orchestration with variables**, which allowed for dynamic and intelligent workflow management. This required a solid understanding of logic design and helped me build systems that are resilient, adaptive, and easy to maintain.

By designing a **custom metadata management layer**, I introduced a reusable, scalable approach to data ingestion and transformation. This not only minimized redundancy but also made the overall architecture more future-proof—a critical skill for any Data Engineer working in agile environments.

I also got to work closely with **Power BI**, connecting my backend data model to create impactful dashboards. This full-cycle experience—from data ingestion to visualization—gave me a holistic perspective on how data engineering directly supports business insights and decision-making.

What truly set this project apart was the opportunity to build **automated and scalable systems**, such as dynamic monthly file processing logic. This taught me how to design solutions that minimize manual intervention and gracefully handle growing data volumes—skills that are essential for enterprise-level data platforms.

About Me



I am **Randy A. Velasco**, a seasoned Professional Electronics Engineer with a strong foundation in data governance, data engineering, data analysis, and database management. I am a certified Data Engineer and Data Analyst. I am also certified in Python, SQL, Data Literacy, Microsoft Azure AI Fundamentals (AI-900) and Microsoft Azure Fundamentals (AZ-900).

By working on end-to-end projects like this, I mimic real-world workflows, such as designing pipelines, ensuring data quality, and presenting insights. It solidifies my understanding of tools, technologies and techniques, such as ETL processes, data pipelines, and visualization tools like Power BI. I gained experience handling issues like missing data, data cleaning, and integrating multiple sources into cohesive datasets. Tackling real-world scenarios helps you bridge the gap between theory and practice, which is invaluable in mastering concepts like SQL queries, cloud technologies, and Python data manipulation.