Data Analytics Team

PARK MANAGEMENT DASHBOARD

Langara.
THE COLLEGE OF HIGHER LEARNING.

CITY OF VANCOUVER

Meet the Team

Angeli Delosreyes

Team Lead and Data Modeler

angeli.delosreyes@gmail.com

Nay Zaw Lin

Cloud Solutions Architect

nayzawlin07@gmail.com

Javier Merino

Data Visualization

javier.merino.maldonado@gmail.com

Meyliani Sanjara

Machine Learning

meyliani.sanjaya@gmail.com

Understanding Park Usage

Objectives

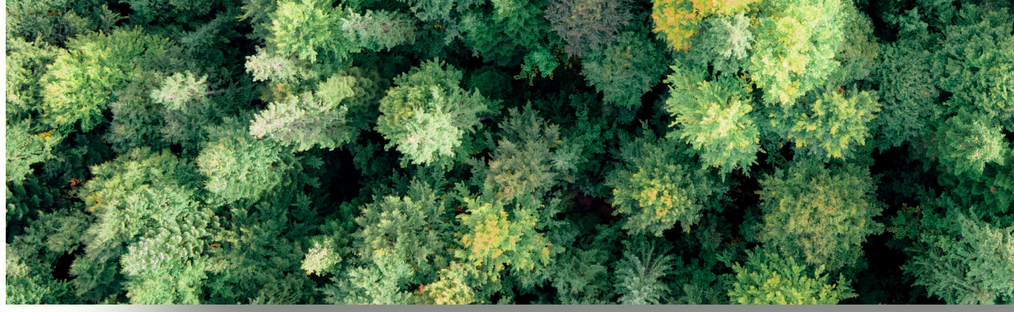
Uncover meaningful patterns in park usage to support:

- Smarter Resource Allocation
- Improved Access & Visitor Flow
- Sustainable, Enjoyable Park Experiences
- Strategic Planning & Investment

Challenges

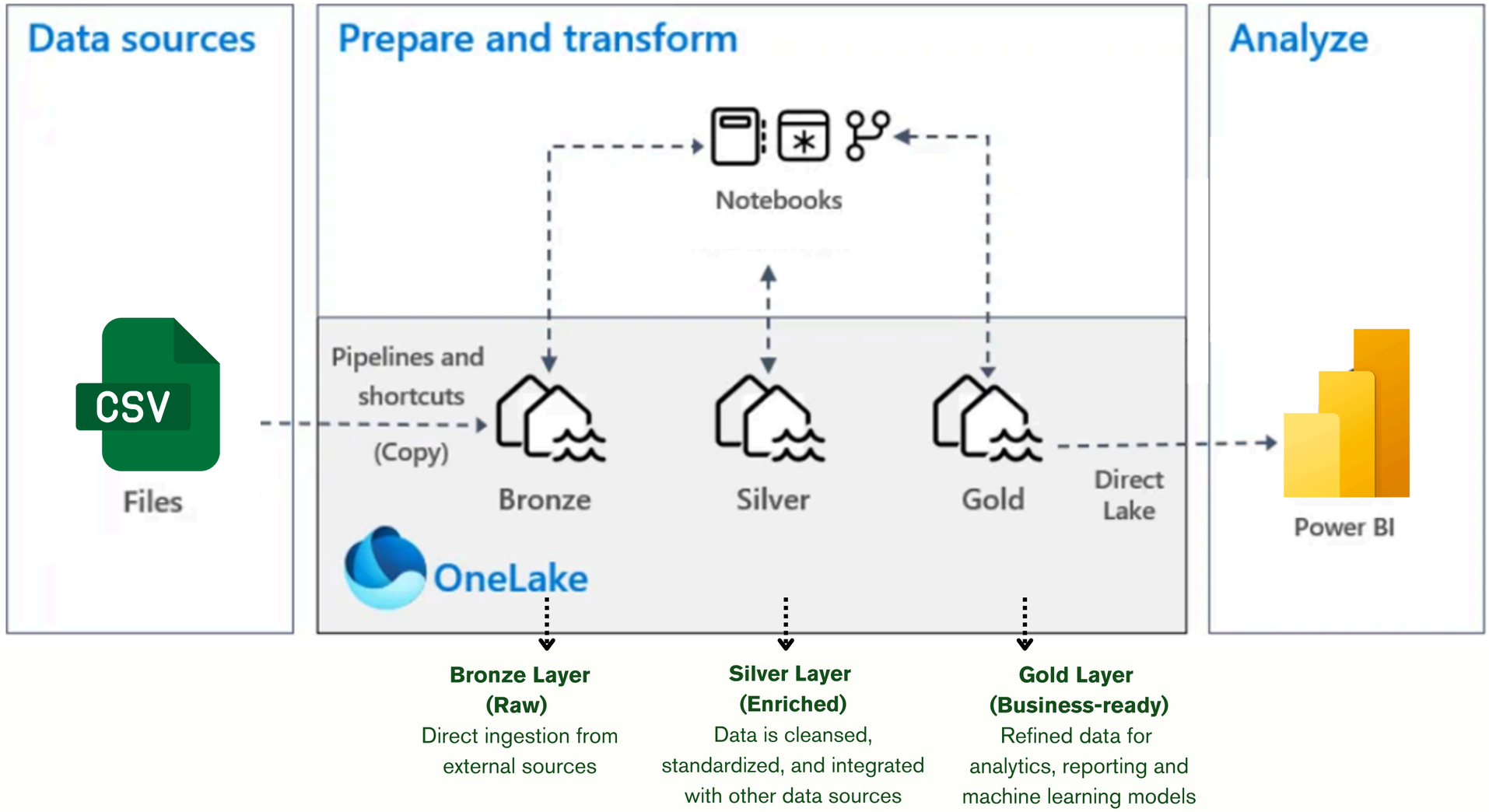
- Limited visibility into when and how many people visit parks
- Sparse observational data (~300 counts, mostly weekdays)
- Google Popular Times ≠ exact user counts
- Tech alternatives (e.g., sensors, cameras) have barriers: privacy, cost, setup

Data Sources

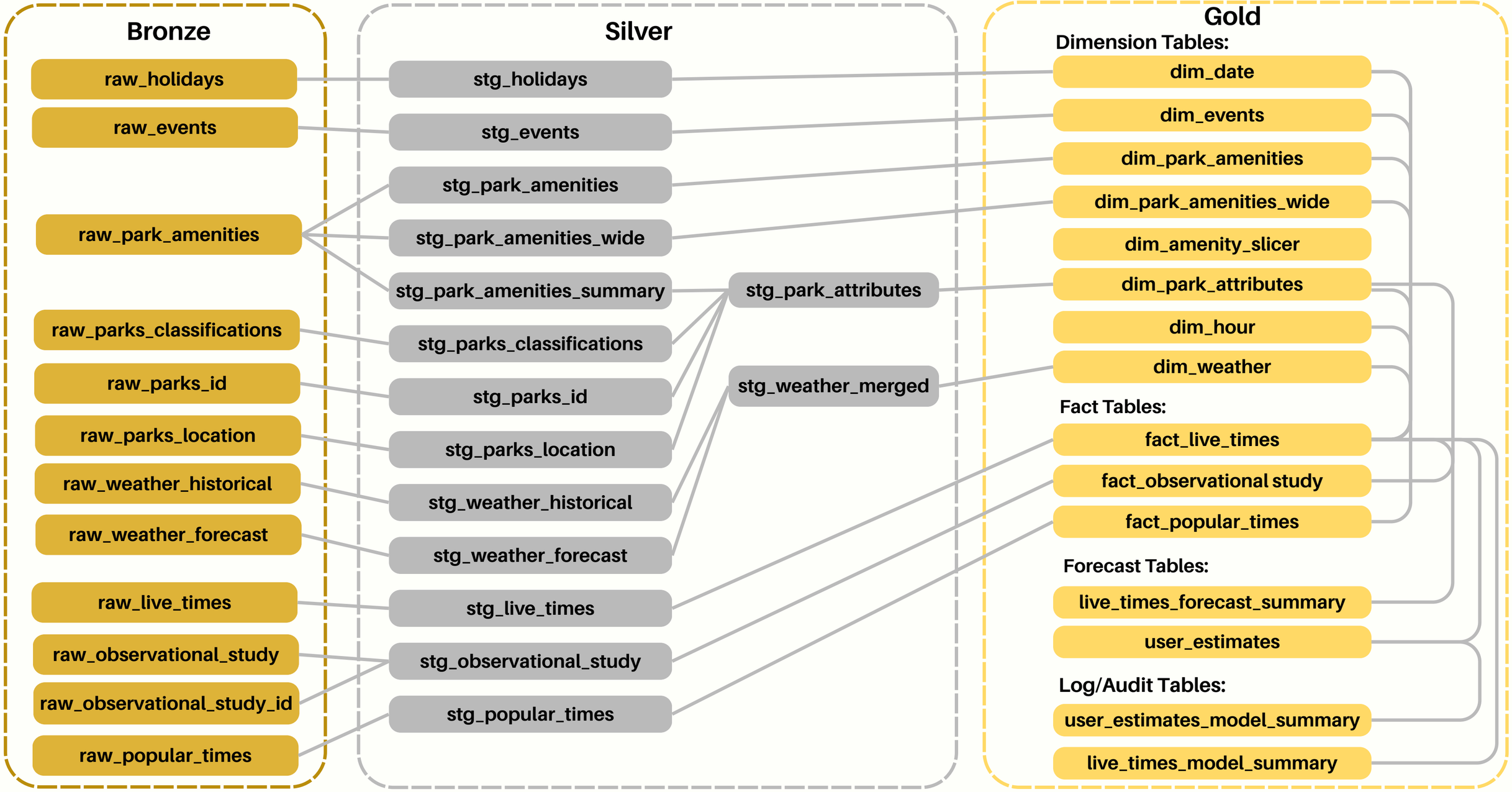


Data Type	Description	Status
Time Series Data	Google Popular Times (Usual & Live Times)	Provided
	Observational Study & Observational Study ID	Provided
	Weather Data (Historical & Forecasted)	Provided (and externally sourced)
Static Data	Parks Amenities	Provided
	Parks Classifications	Provided
	Parks ID	Provided
	Park Events	Provided
	Parks Map	Not Provided (externally sourced)
	BC Holidays	Not Provided (externally sourced)

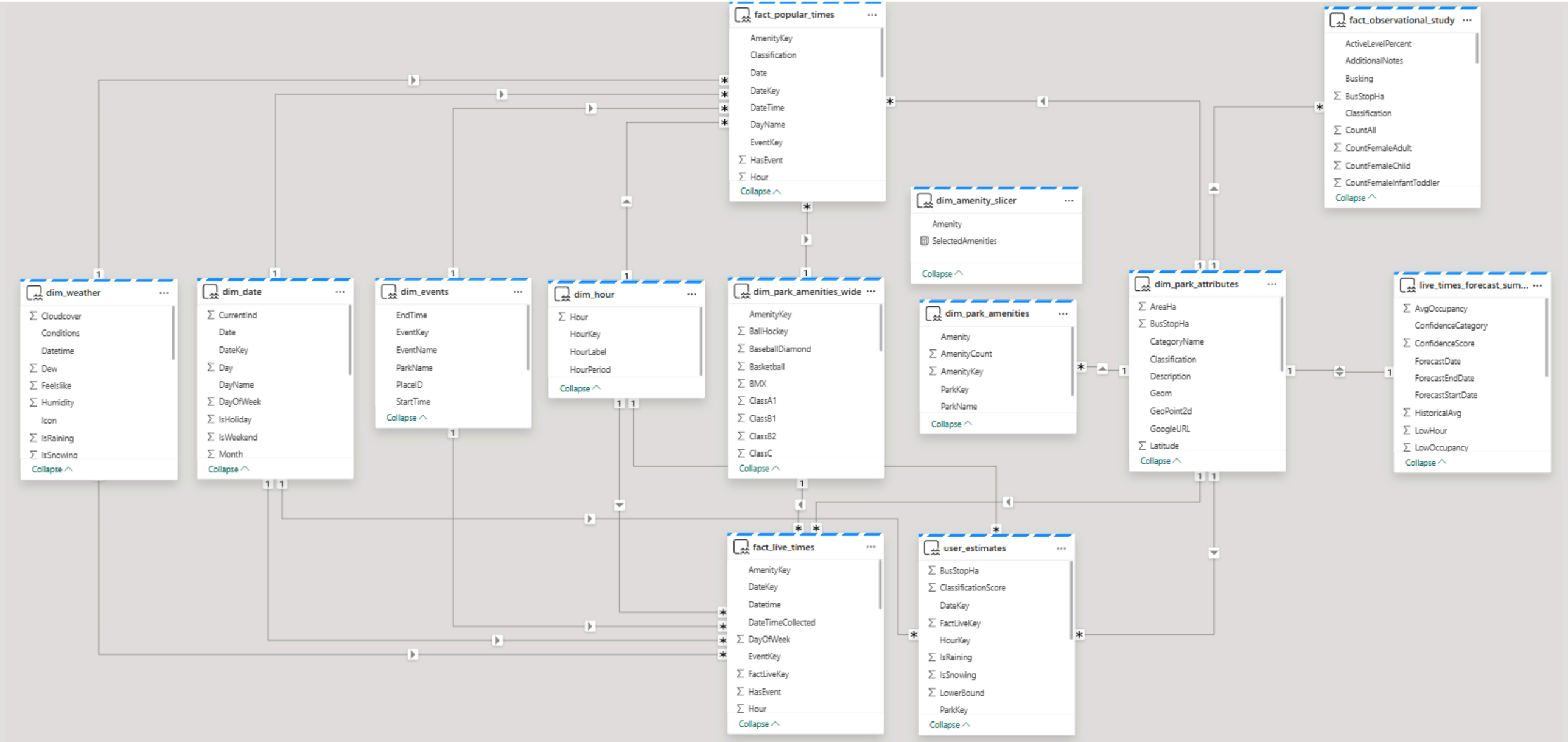
Lakehouse Medallion Architecture



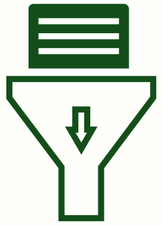
Lakehouse Medallion Architecture



Semantic Model

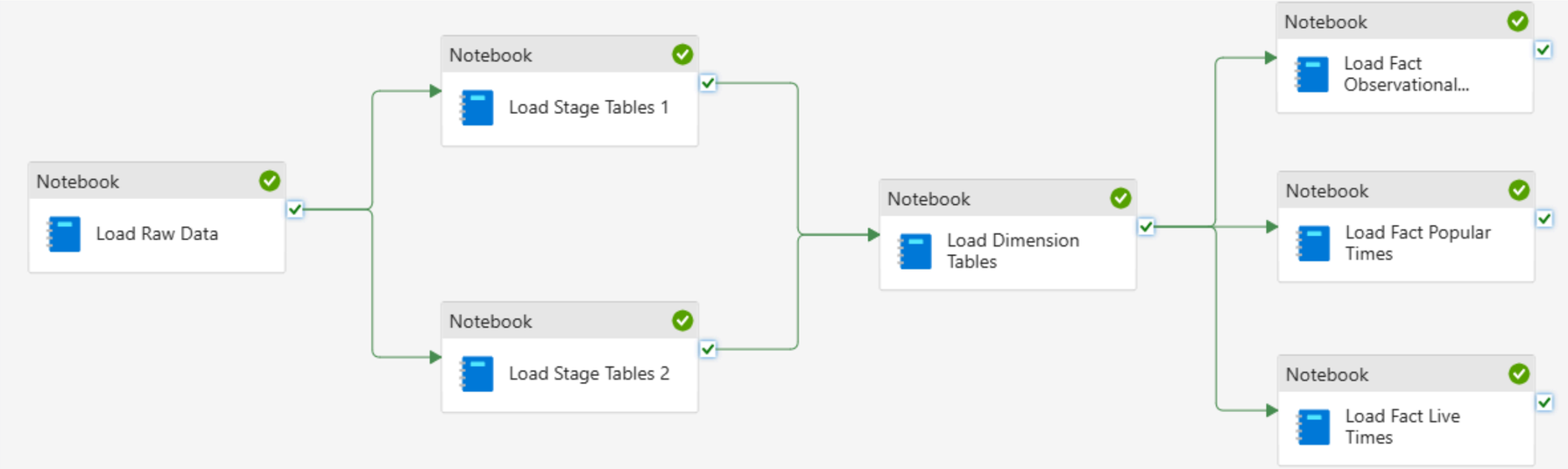


Data Pipelines (in Microsoft.Fabric)

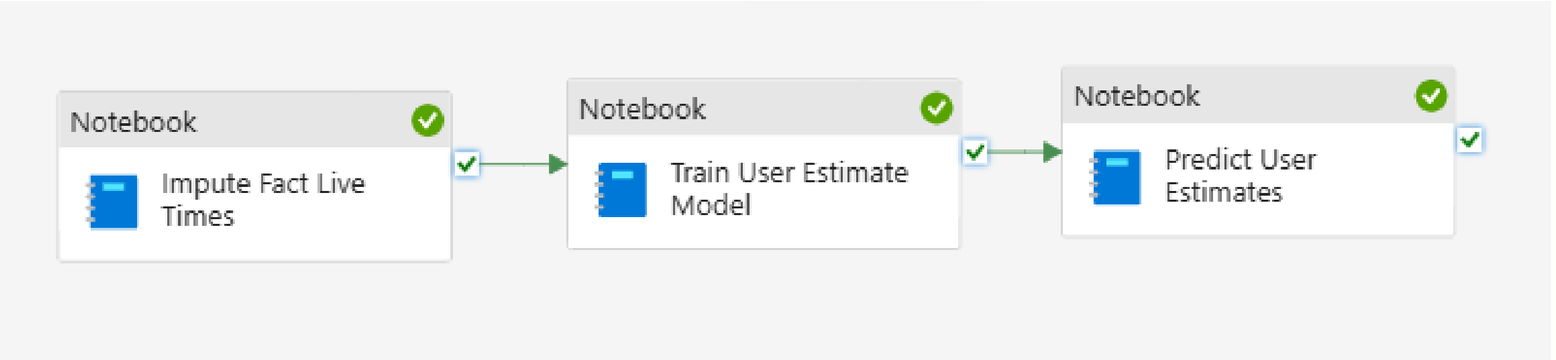


	Data Ingestion + Transformation	User Estimation	Train Forecasting Model	Predict Park Occupancy
Purpose	Extracts and transforms raw data to generate clean dimension and fact tables	Predicts park-level daily user counts using ML model	Builds a park-specific 7-day time series forecasting model	Generates 7-day hourly occupancy forecasts per park
Schedule	Daily	Daily (post Pipeline 1)	Quarterly or as needed	Weekly
Key Features	Full refresh from raw → stage → dims & facts using client-provided CSVs	XGBoost model using 9 features (6 static, 3 weather); outputs 1 park/day estimate	Trained with LightGBM; incorporates weather forecast + park events	Outputs 2-week view: 7-day historical + 7-day forecast with confidence intervals

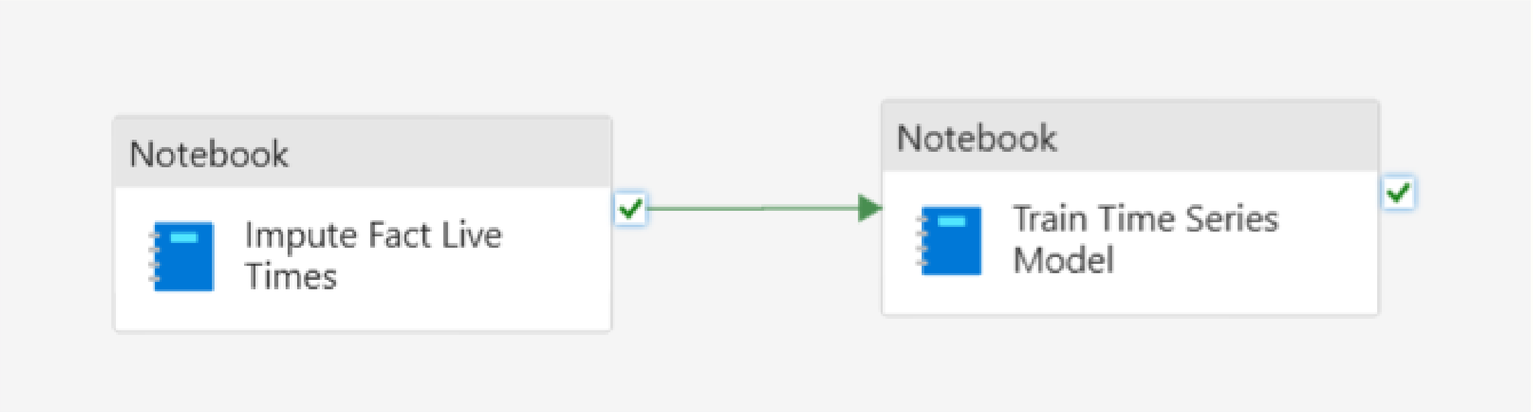
Pipeline 1: Data Ingestion + Transformation



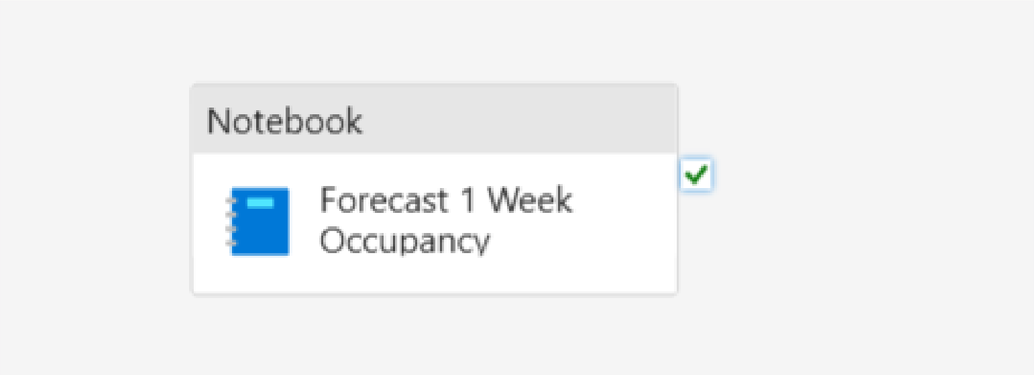
Pipeline 2: User Estimation



Pipeline 3: Train Forecasting Model



Pipeline 4: Predict Park Occupancy (1 week)



Key Features

Descriptive Analytics

- Park Occupancy patterns.
  - Data: Live and Average Busy Times, Weather, Events.
  - Granularity: Hour, Day of Week, Month, Year.
  - Filters: Park classification, Maintenence Area, Amemities, and Geolocation.
  - KPI:
    - Peak Hour, Off-Peak Hour, Live vs Average proportion.
    - Pop. per Ha, Bus Stop per Ha.

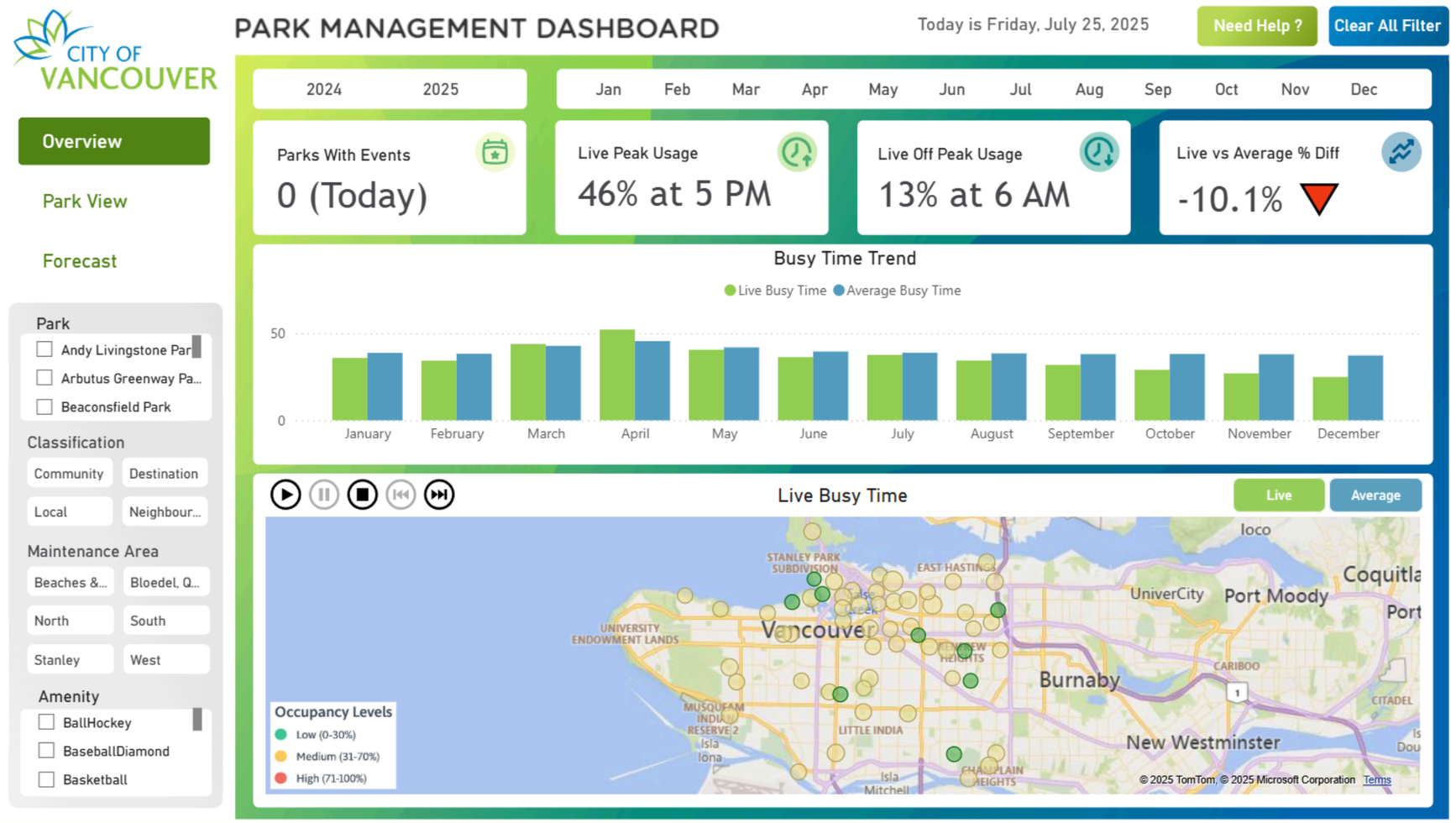
Predictive Analytics

- Weekly occupancy forecast for resource planning
  - Granularity: Operational Hours, Day of the Week.
- Estimated Users
  - Granularity: Day of the week.

Demo (in Power BI)

Go to Demo page

Click Here

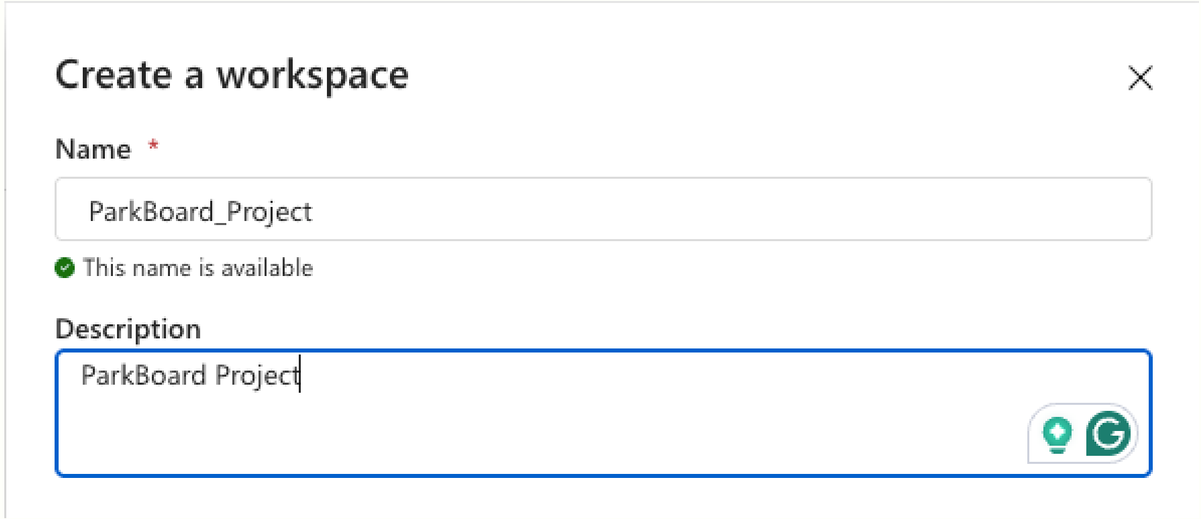


Migration Plan (Overview)

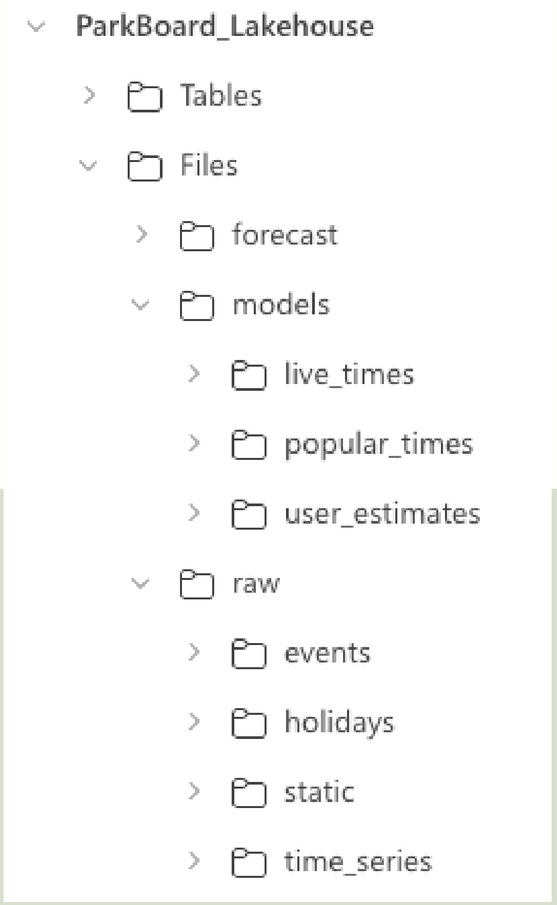
- Create the new Workspace (ParkBoard_Project) in Microsoft Fabric
- Set up the OneLake folder hierarchy and copy the forecast, models, and raw data
- Import Fabric Notebooks
- Rebuild Data Pipelines & Run Ingestion-to-Prediction Flow
- Rebuild the Semantic Model
- Connect dashboard visuals to the semantic model and publish the Power BI Report

Migration Plan Steps

1. Create the new Workspace (ParkBoard_Project) in Fabric

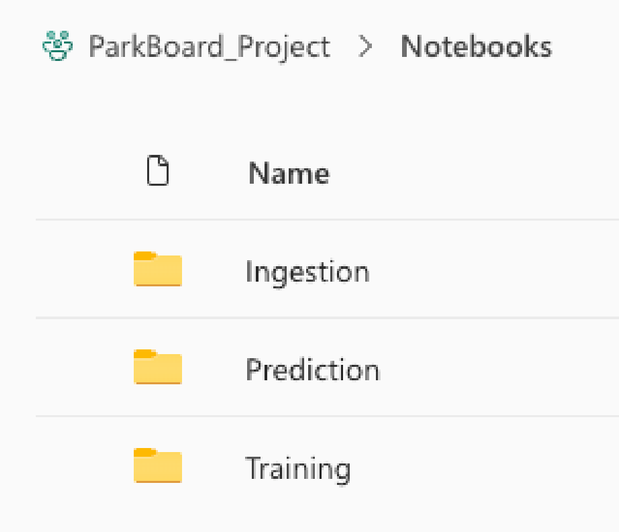


2. Set up the OneLake folder hierarchy and copy the forecast, models, and raw data packages using OneLake File Explorer



Migration Plan Steps

3. Import Fabric Notebooks

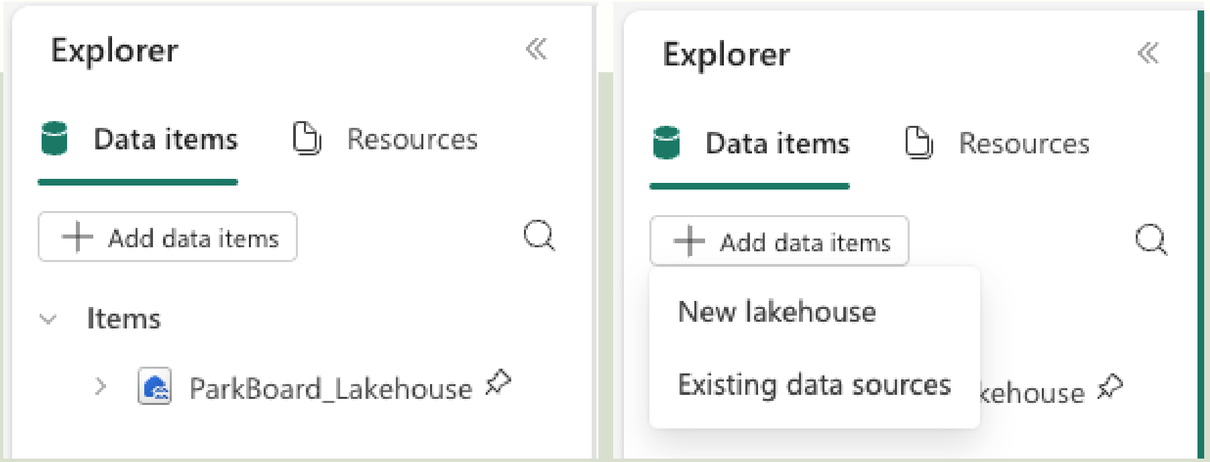


Note: Update ingest_01_load_raw.*ipynb* hardcoded file paths (e.g., abfss://) to match with your workspace.

```
# Parks Location (CSV)
spark.read.option("header", True).option("sep", ";") \
    .csv("Files/raw/static/parksMap/parks-polygon-representation.csv") \
    .write.format("delta").mode("overwrite").saveAsTable("raw_parks_location")

# Park amenities (Excel)
amenity_df = pd.read_excel("/lakehouse/default/Files/raw/static/parksAmenities/parkAmenityTable.xlsx", header=0)
amenity_spark_df = spark.createDataFrame(amenity_df)
amenity_spark_df.write.format("delta").mode("overwrite").saveAsTable("raw_park_amenities")
```

Note: Verify that each notebook is correctly linked to its intended Lakehouse data source. Otherwise, change the correct data source from Existing data sources.

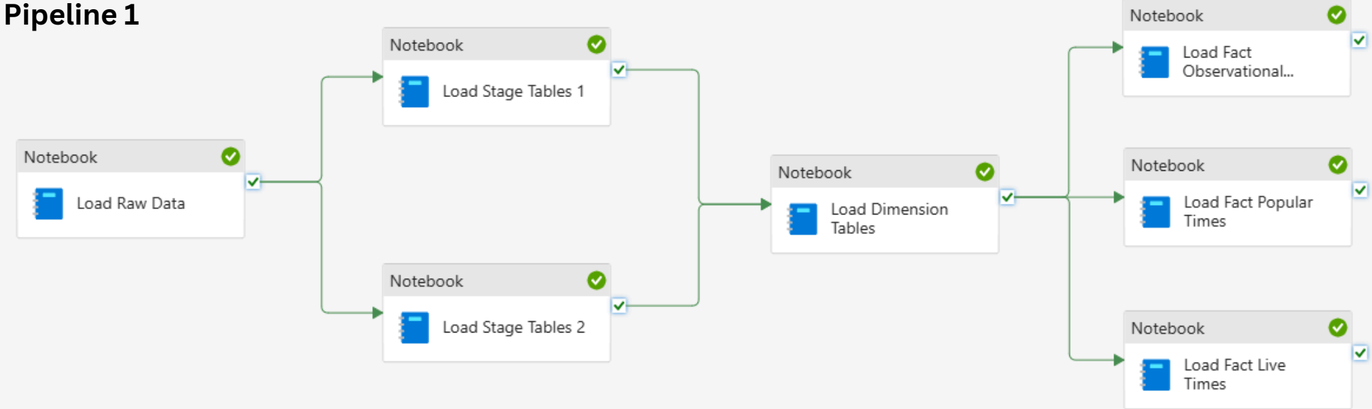


Migration Plan Steps

4. Rebuild Data Pipelines & Run Ingestion-to-Prediction Flow

- Pipeline 1: Data Ingestion + Transformation
- Pipeline 2: User Estimation
- Pipeline 3: Train Forecasting Model
- Pipeline 4: Predict Park Occupancy (1 week)

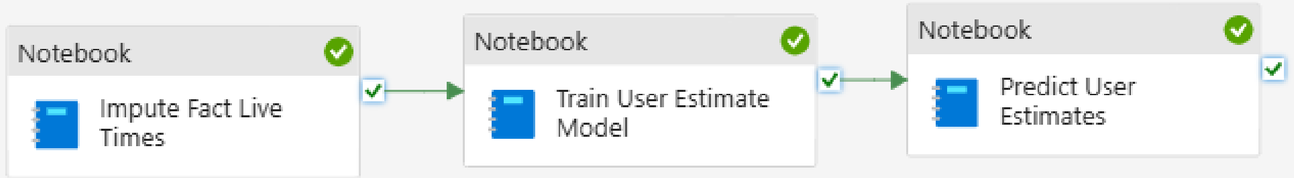
Pipeline 1



Pipeline 3

Pipeline 4

Pipeline 2

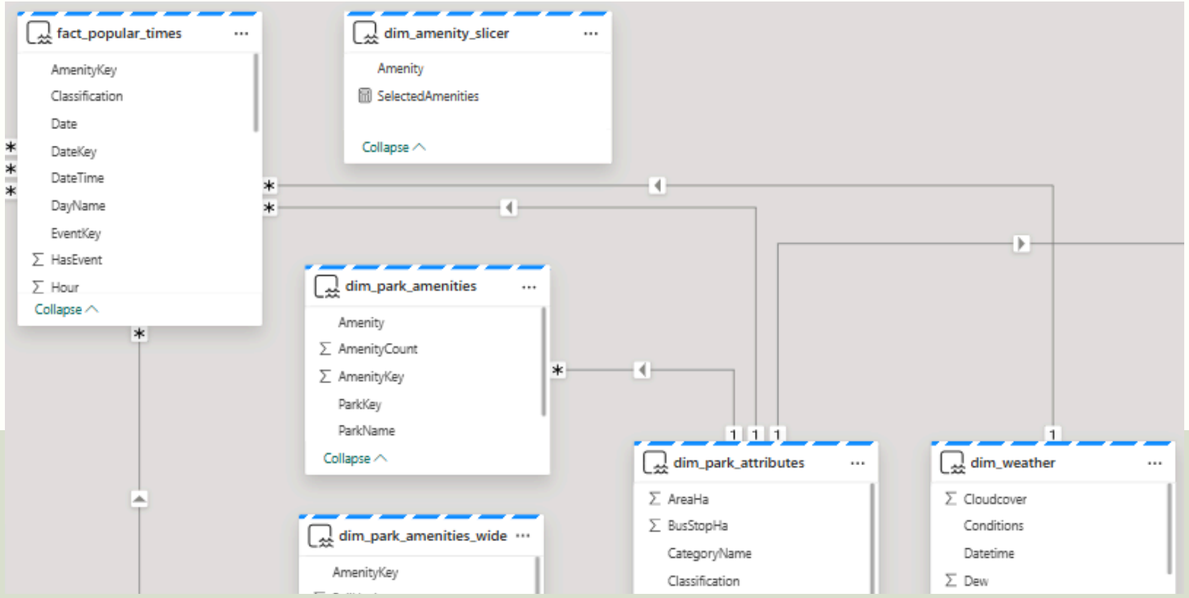


Migration Plan Steps

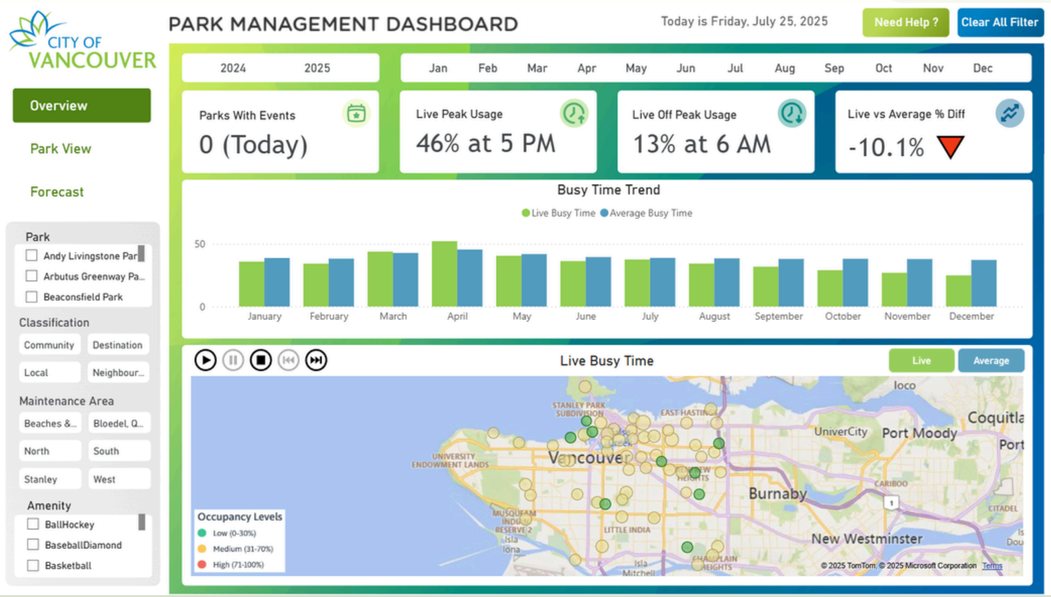
5. Rebuild the Semantic Model

Pre-req: Data pipeline must be run to populate tables

Due to current limitations in Fabric Web, measures must be recreated through code. We are actively exploring alternative export options for the semantic model.



6. Connect dashboard visuals to the semantic model and publish the Power BI Report



Note: As part of the integration, we used GitHub repository to host the generated forecast plot images from Fabric, allowing us to access and display them directly within the Power BI dashboard.

THANK YOU