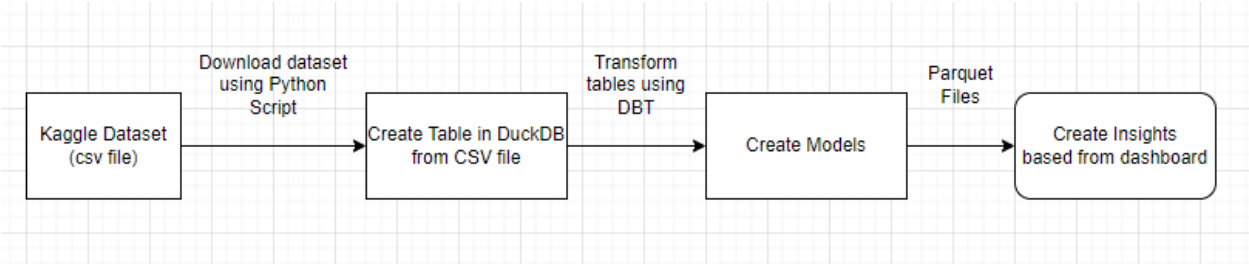
A. Methodology



Overview:
This section contains the high level

1. We write a script called ***extract-kaggle-dataset.py*** where we download the csv file from Kaggle using a library called ***opendatasets***. The input that we need is the url of the dataset. We shall name this csv file ***ecommerce.csv***
2. Open DuckDB. By default, the database is in memory. To create a new database, type .open and then type the name that you want on your database. In this case, let's use the command ***.open ecommerce-transactions.duckdb***
3. After extracting the dataset (in csv format), we create a table in DuckDB using a command ***CREATE TABLE ecommerce AS FROM read_csv('ecommerce.csv')*** (Note: make sure that the csv file is in the same path as duckdb.exe)
4. Load database in an IDE (I used VSCode). Before proceeding, make sure that there's a dedicated environment for our Python packages
    a. The dependencies used include pandas, opendatasets, duckdb, dbt, and dbt-duckdb
5. Initialize dbt project by typing ***dbt init***
6. Configure ***dbt_project.yml*** and ***profile.yml*** (can be found in the root directory of dbt)
7. Create intermediate, staging, and mart models using dbt
8. Once satisfied with the end result of the models, export the tables as parquet file
9. Connect the parquet file to Power BI. We can create custom measures in Power BI

B. Model Lineage

Figure 1: Model Lineage (using dbt docs generate and dbt docs serve)

The plan is as follows

- Do most of the data modeling and transformation in dbt.
    - Clean the raw table and materialize it as ephemeral first (an intermediate table
    - Once cleaned, we referenced that intermediate table to a staging table
    - Create mart models from the staging model. Mart models are based on what we are trying to answer based from the dataset
- Create custom measures on PowerBI
    - As much as possible, we do less data transformation on Power BI to avoid lagging issues

Here are the description of the models

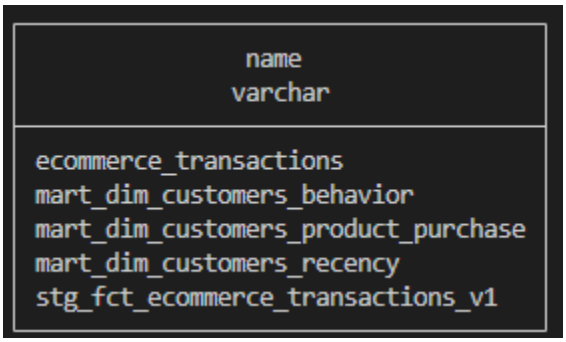


Figure 2: Models seen in DuckDB

- Intermediate models
    - **Int_fct_ecommerce_transactions_cleaned_v1**
        - We treat this as a base model. This is the model where we clean the raw table **ecommerce_transactions** by filtering out CustomerIds that are null, any anomalous StockCode, and other computations that we can consider. We materialized this as **ephemeral** (can be view) so that we cannot alter the present logic in this model
    - **Stg_fct_commerce_transactions_v1**

- This is where the intermediate model is being referenced to. We use this later on to be referenced by different mart models. It is also materialized as incremental model
- The reason why this is materialized as an incremental model is because we don't want to load the entire table everytime we run it. We only load the data that we need based on a specific timeframe

- **Mart_dim_customers_behavior**
  - This is a mart model that can be used to answer questions about customer's purchase behavior
  - A dimension table and materialized as an incremental model
- **Mart_dim_customers_product_purchase**
  - This is a mart model that can be used to answer questions about customer's history of product purchase
  - A dimension table and materialized as an incremental model
- **Mart_dim_customers_recency**
  - This is a mart model that can be used to answer questions about the last time the customer is seen active in making transactions
  - A dimension table and materialized as an incremental model

C. Model in PowerBI

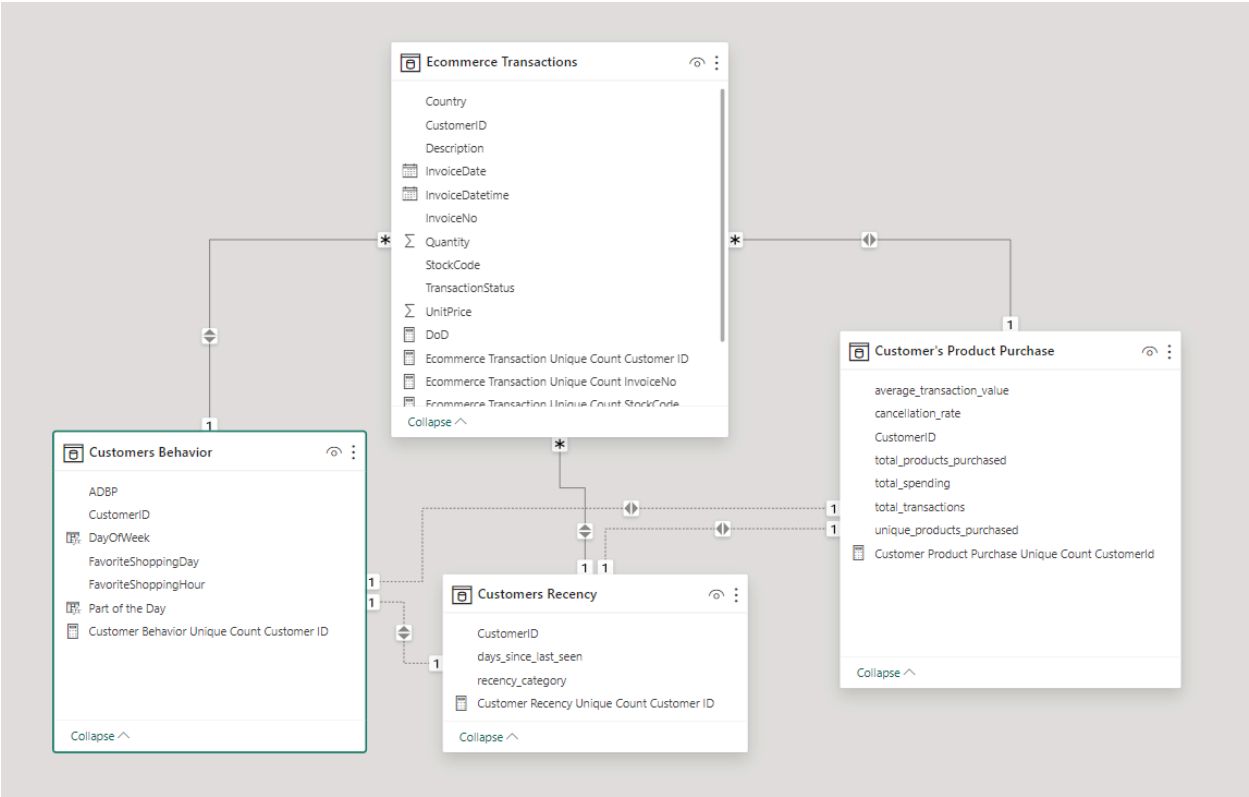


Figure 3: Table View in Power BI

This figure shows how we relate the models in PowerBI. Ecommerce transaction (the staging model) is connected to the mart models on many-to-one relationships. For each mart models, they are related on one another by one-to-one relationship.

D. Dashboard

The dashboard contains the following

- Ecommerce Transaction Overview
    - This page shows the overall performance of the ecommerce company. It has metrics include how many unique number transactions are made on a given time period as well as its DoD, WoW, and MoM change. It also tells us what country has the most number of transactions

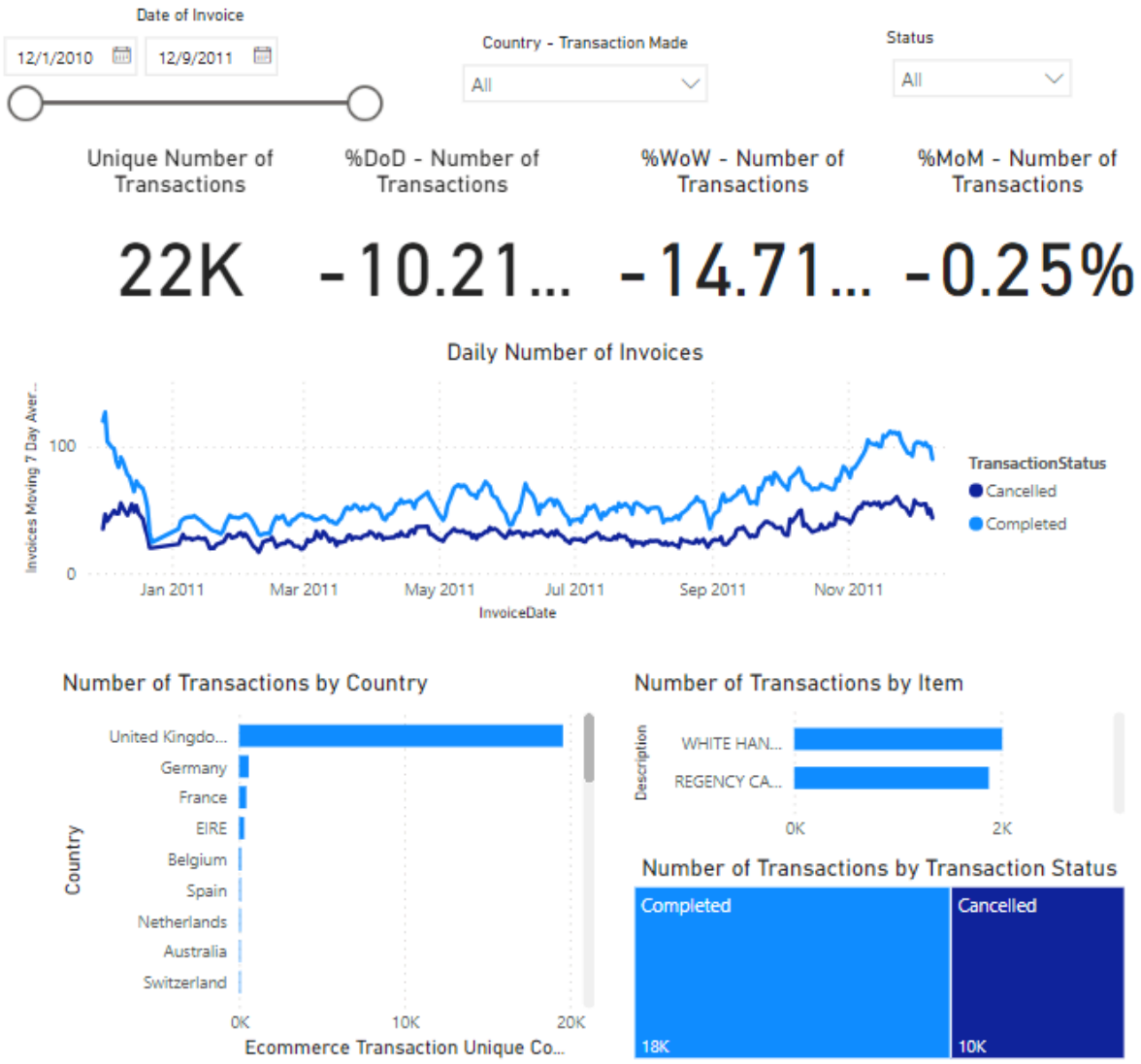
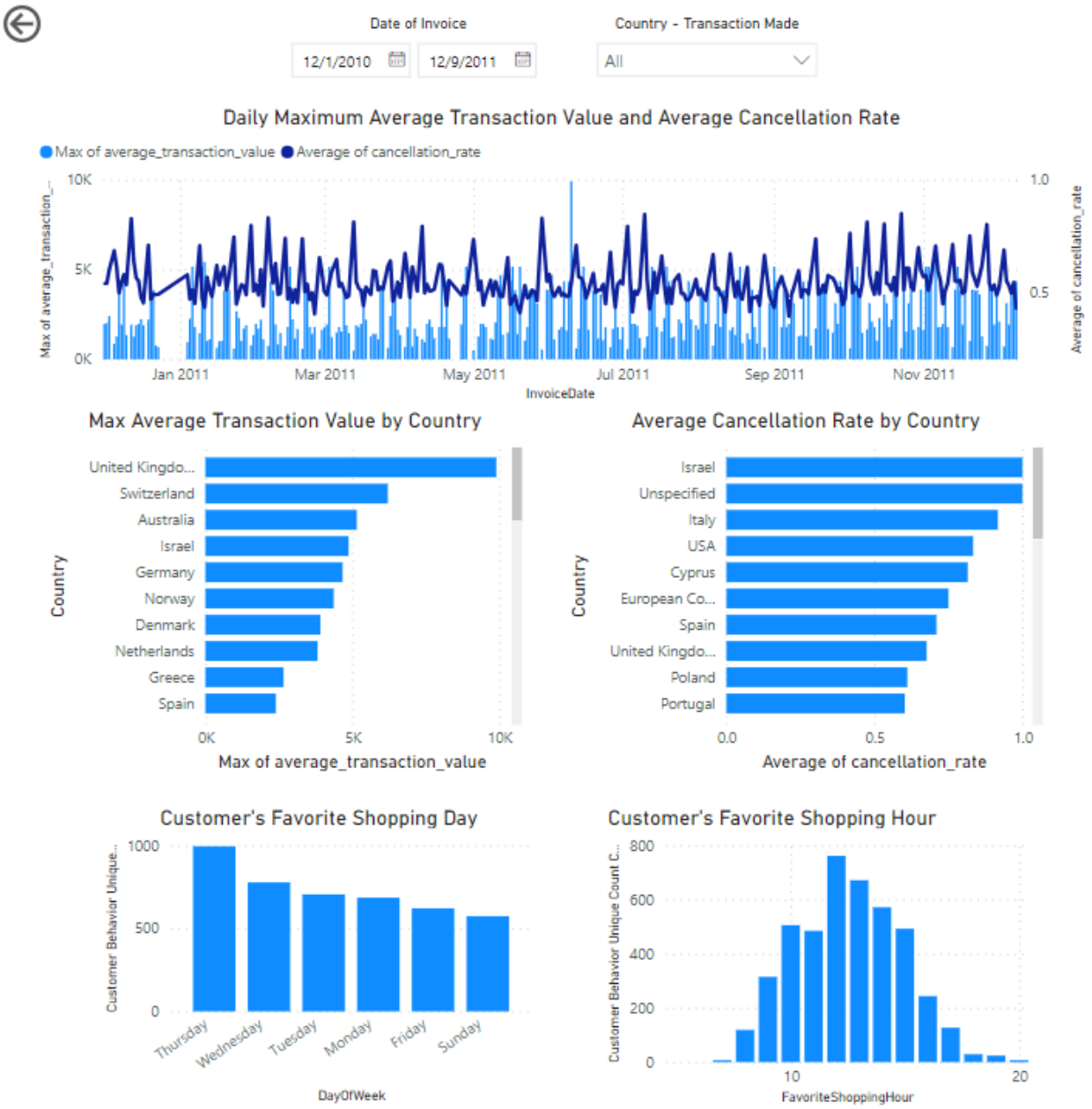
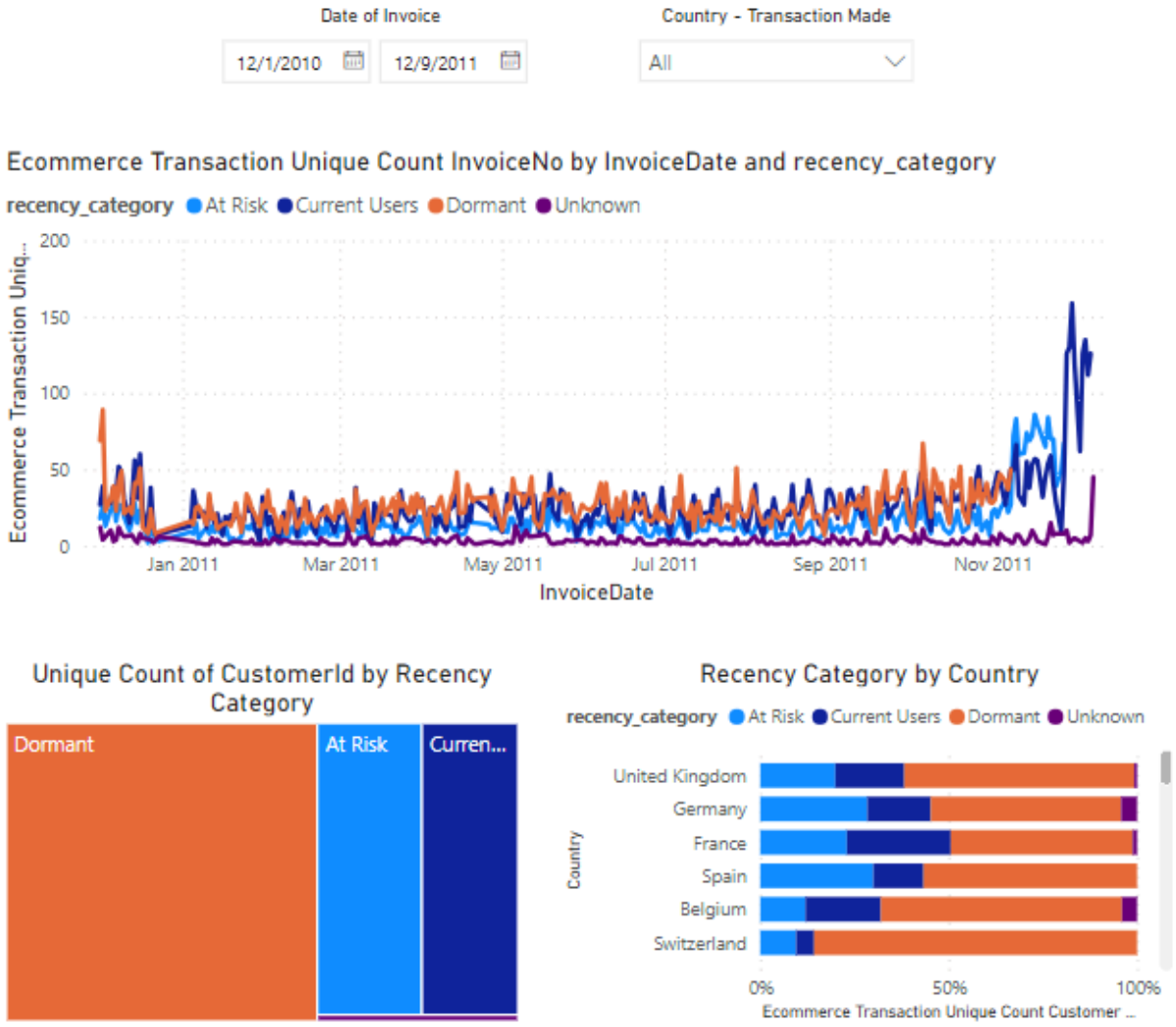


Figure 4: Ecommerce Transaction Page

- Customer Behavior
    - This page shows what is the customer's behavior (favorite part of the day to create transactions (also what hour). It also tells us some metrics about customer's purchase history (cancellation rate and average transaction value)

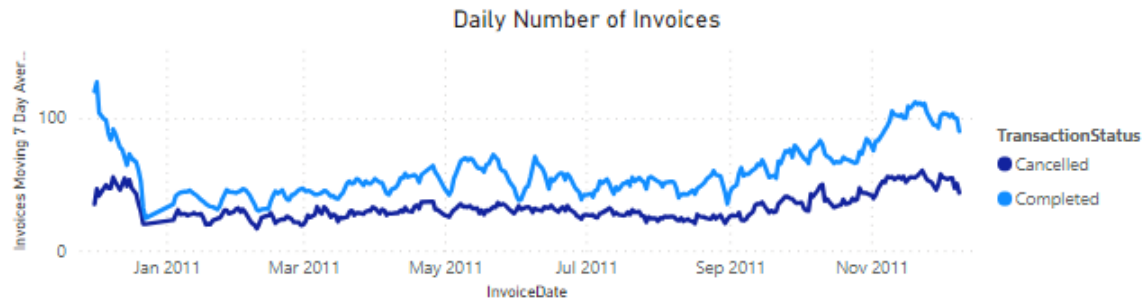


- Customer Retention
    - How many users are still currently active, how many are at risk of being dormant, and how many are dormant users



E. Insight

- From this line chart



We can say that from the beginning of the transaction, there's a sharp decline of number of transactions. At around October 2011 (approximately), there's a steady increase of transactions

- There's an alarming number of dormant users for all the countries recorded in this dataset
- The United Kingdom has the most number of transactions, while RSA has the least. Also UK has the highest average transaction value - meaning it has the highest average amount of customer spends on a given purchase
- Israel has the highest average cancellation rate but has pretty fair average transaction value
- Customers usually create transactions in the morning. The peak number of transaction is recorded at 12 noon
- Thursday has the most recorded number of transactions
- Many buys 'WHITE HANGING HEART T-LIGHT HOLDER'

F. Recommendations

Dataset

- If given more time, I could've enrich the dataset by providing the category of each item, latitude and longitude of the country, and first seen date of each customer

Tech stack used

- Use Docker containing all dependencies used in the project
- Can use cloud based data warehouses such as Google BigQuery or Snowflake
- If possible, can create more mart models
- Use Github Actions