94693 Big Data Engineering - Spring 2023

AT3

Done by:

Name: Tahmidul Islam

ID: 24587139

1. Overview

Airbnb's innovative online platform has revolutionized the lodging industry, enabling individuals to rent out their properties and travelers to secure accommodations for varying durations. Airbnb has disrupted the traditional hospitality sector, amassing over 150 million guest users and 6+ million listings worldwide by 2019. This substantial data trove, encompassing rental details, pricing information, and regional listing distribution, forms the foundation of this project, focusing on Sydney, a global tourism hotspot.

Complementing Airbnb data, the project integrates 2016 census information from the Australian Bureau of Statistics. This census data provides context for Airbnb listings, enabling insights into the neighborhoods where these rentals are concentrated. The project's primary objective is to establish production-ready pipelines utilizing Airflow. These pipelines process and cleanse input datasets before loading them into a data warehouse through Extract, Load, Transform (ELT) pipelines. Once the data resides in the warehouse, it undergoes further transformations and is subsequently transferred to a data mart for analytical purposes..

2. Overall Architecture

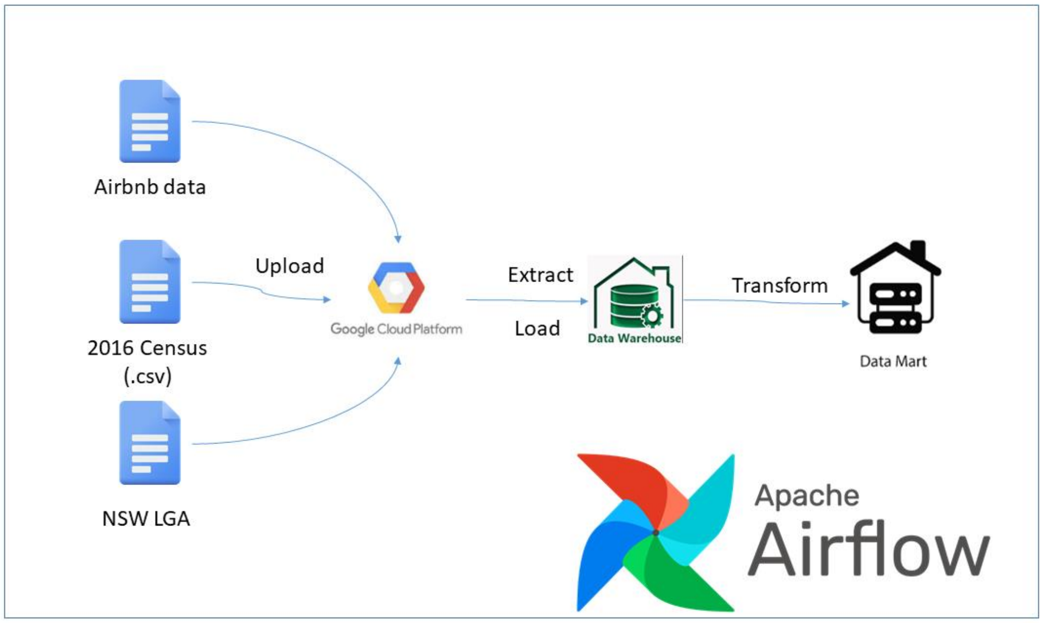


Fig: Overall Architecture of the Project

The project was executed in 4 basic steps:

1. GCP: in the first stage, we imported the data (.csv) files into the cloud storage.
2. Airflow: The data pipeline was constructed using Airflow to automate the data loading procedures. This implementation pushed the raw files within Google Cloud Storage into the data warehouse.

3. Data Warehouse: Upon uploading the data files to Google Cloud Storage, the next step involved extracting and saving them in the data warehouse's raw layer. These files were initially stored as raw tables. Subsequently, they were moved into the staging layer, where they transitioned into views. Following this, the data underwent a transformation, forming a star schema, and ultimately found their place in the warehouse layer as both dimension and fact tables.
4. Leveraging the dimension and fact tables sourced from the warehouse layer, three tables containing Key Performance Indicators (KPIs) were created and archived in the data mart layer of the Data Warehouse.

3. Data Ingestion and Preparation

Ideally, the project aimed to ingest data by making use of Azure Data Cloud Storage before mounting to Databricks File Storage. However, the dataset was extremely large and upload errors were coming up repeatedly after waiting for a long time. Thus, the parquet files were directly mounted to the DBFS from google drive, provided by the Course Lecturer.

Data Ingestion:

- 12 months of Airbnb data were obtained from the following source: Airbnb Data.
- Census data was retrieved from the following source: Census Data.
- New South Wales (NSW) LGA data was acquired from the following source: NSW LGA Data.
- Following the data acquisition, all the data files were uploaded to Google Cloud Storage, and organized into three distinct folders:
    - Airbnb data was placed in the "data/listings" folder within the Google Cloud Storage bucket.
    - Census data was stored in the "data/Census_LGA" folder within the same Google Cloud Storage bucket.
    - NSW LGA data was situated in the "data/NSW_LGA" folder within the Google Cloud Storage bucket.

- Subsequently, in Postgres (DBeaver), a database was established.
- Within that database, a schema named "raw" was created.
- Finally, subsequent schemas of staging, warehouse and datamart were created.

4. Data Warehouse Design

Raw schema

Adhering to the ELT methodology, all data files from the three datasets were loaded into Snowflake as external tables. The external tables are as follows:

- raw.listings: Holds Airbnb data spanning 12 months

- raw.g01_census: Encompasses census data pertaining to individual characteristics categorized by sex
- raw.g02_census: Comprises census data related to medians and averages
- raw.nsw_lga_code: Contains information on NSW LGA codes and their corresponding LGA names
- raw.nsw_lga_suburb: Provides details on NSW suburbs and their respective LGA names
- host_snapshot: Snapshot of the host dimension
- listing_snapshot: Snapshot of the listing dimension
- property_type_snapshot: Snapshot of the property type dimension

Staging schema
In the staging schema, the data, especially the snapshots were loaded into the staging tables. Not all columns from the census data were included in the load, as some were considered non-essential for the project's analysis.

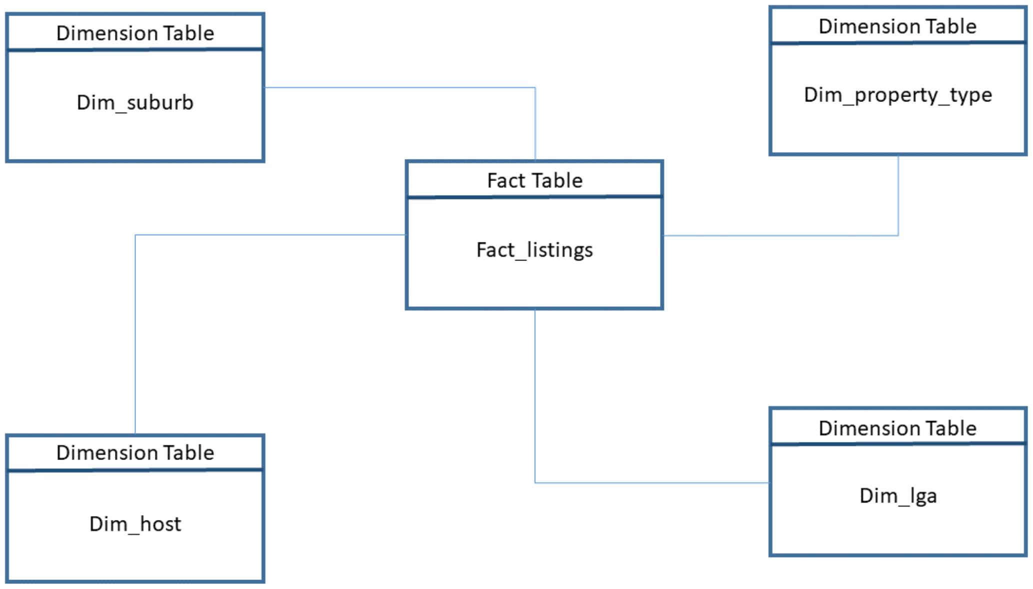
- census_g01_stg
- census_g02_stg
- facts_listing_stg
- host_stg
- lga_stg
- listing_stg
- property_type_stg

Datawarehouse schema
The tables from the staging schema underwent a transformation process and were subsequently loaded into the data warehouse schema as both fact and dimension tables. These fact and dimension tables were designed to establish a star schema, where a central fact table connects to various dimension tables using foreign keys.

- Fact_listings
- Dim_property_type
- Dim_lga
- Dim_host
- Dim_suburb

Star Schema

The star schema was built by having the Fact_listings as the fact table and dim_suburb, dim_property_type, dim_host and dim_lga as the dimensions tables.



5. Ad-hoc Analysis

Differences between best and worst performing neighbourhoods from a population point of view

An explaination of how this was achieved in the coding is explained below:

First, the code calculates various statistics on a monthly basis for different characteristics of Airbnb listings within specific neighborhoods (listing_neighbourhoods). The code retrieves data from the datawarehouse.fact_listings table and joins it with several dimension tables (e.g., dim_lga, dim_date, and dim_listings) to gather relevant information. The calculated statistics include the median age of people, total population, population of people aged 0-4 years, 5-14 years, 15-19 years, 20-24 years, and 25-34 years. Additionally, it computes the average estimated revenue per active listing, considering factors like availability and price. All these statistics are aggregated on a monthly basis for each listing_neighbourhood.

In the second part of the code, the previously calculated monthly statistics are aggregated further to summarize data on a per-neighborhood basis. It computes the following aggregated statistics for each listing_neighbourhood:

Median age of people, averaged over the months, total population and averaged over the months, combined population of people aged 0-34 years, total estimated revenue per active listing.
These final neighborhood-level statistics are generated from the data obtained in the first step and provide insights into neighborhood characteristics and Airbnb listing performance. The neighborhoods are then sorted in descending order based on their estimated revenue per active listing.

In summary, this code processes Airbnb listing data to derive neighborhood-level statistics and insights, facilitating analysis and comparison of different listing neighborhoods in terms of population, age demographics, and estimated revenue.

Listing type with highest number of stays for the top 5 listing neighbourhoods
This SQL for this part is designed to perform an in-depth analysis of Airbnb listing data, with a specific focus on individual listing neighborhoods. It is split into two main sections that work together to provide insights and rankings based on various criteria.

In the first part of the code, referred to as "Listing Neighborhood Revenue Analysis," a subquery named listing_neighbourhood_monthly_stats is utilized. This subquery calculates the average estimated revenue per active listing for each listing neighborhood and month. This calculation is quite comprehensive, taking into account factors such as the availability of listings and their corresponding prices. The data for this analysis is sourced from the datawarehouse.fact_listings table and is enriched through joins with dimension tables, specifically dim_lga and dim_listings.

Following this subquery, the main part of the code processes and aggregates the results from listing_neighbourhood_monthly_stats. This summary encompasses the total estimated revenue per active listing for each neighborhood. Furthermore, a row number, rk_est_revenue, is assigned to each neighborhood based on the descending order of estimated revenue. This row number will be instrumental in ranking neighborhoods according to their estimated revenue, allowing us to identify which neighborhoods generate the highest revenue, as they will have lower row numbers.

In the second part of the code, titled "Listing Neighborhood Stays and Accommodations Analysis," the focus shifts to analyzing data related to stays and accommodations within each listing neighborhood. Here, we encounter another subquery named listing_neighbourhood_stays. This subquery computes the total stays, property types, and room types associated with each neighborhood. The total stays are calculated based on listing availability, and a row number, rank_stays, is assigned to combinations of neighborhood, property type, room type, and accommodation criteria. This ranking is based on the total stays in descending order.

The final part of the code combines the results of the two preceding sections. It matches the listing neighborhoods analyzed for revenue with those analyzed for stays and accommodations.

The code selects the top five neighborhoods based on estimated revenue (determined by rk_est_revenue values less than or equal to 5) and neighborhoods where rank_stays is equal to 1. This dual filtering approach allows us to identify neighborhoods that excel both in generating revenue and accommodating guests, making them highly desirable.

The output of this analysis contains several informative columns, including listing_neighbourhood, property_type, room_type, accommodates, total_stays, and estimated_revenue_per_active_listing. The final list of neighborhoods is sorted in descending order of estimated revenue. This SQL code, therefore, serves to provide valuable insights into neighborhoods that are top performers in terms of revenue generation and accommodation popularity within the context of Airbnb listings.

6. Issues and Solutions

To address a discrepancy in the LGA (Local Government Area) codes between the LGA table and the census tables, a specific issue was identified. The census tables included an extra 'LGA' string as a prefix to the actual code, causing an inconsistency. To rectify this, the solution involved removing the 'LGA' prefix from the LGA codes within the census tables. After this correction, the census tables were seamlessly integrated with the LGA table through a join operation, aligning the LGA codes for accurate data synchronization.

7. Conclusion

The data pipeline constructed using Airflow yielded valuable insights into the Airbnb listings' data, providing substantial relevance to the stakeholders. Nevertheless, it's important to note that the dataset used for this analysis was somewhat outdated. To provide stakeholders with a more current and comprehensive perspective on the performance of these rental listings, it is recommended to incorporate an updated dataset. This updated dataset should encompass the latest 2021 census data and the most recent Airbnb listings' data, offering a more current and accurate portrayal of the situation.