94693 Big Data Engineering - Assignment 3

Building ELT Data Pipelines with Airflow and dbt for Airbnb and Census Data: A Comprehensive Report

Github Repo: https://github.com/kirandas-dev/bde_airbnb

Executive Summary:

The assignment's primary objective was to build an ELT (**Extract, Load, Transform**) data pipeline using Apache Airflow and dbt (data build tool) to process, analyze, and present insights from Airbnb and Census data. The assignment successfully accomplished this goal, and this executive summary provides an overview of the key highlights and outcomes of the report.

Introduction:

We have designed and implemented an Apache Airflow DAG (Directed Acyclic Graph) to manage the extraction and loading of various datasets. This report aims to give a brief walkthrough of how data flow is handled within this Airflow workflow.

Overall Data Flow

Data is initially extracted from CSV files residing in Google Drive directories. These files are then uploaded to a GCP bucket, making them accessible in a centralized location. The Airflow environment was set up to manage the flow of data from GCP to a SQL PostgreSQL instance. This instance was remotely managed by DBeaver, installed on our local macOS machine. The extracted and transformed data was then efficiently loaded into the PostgreSQL database, ensuring data integrity and consistency.

Our aim is to shed light on the significance of each step and its role in ensuring high-quality data for analysis.

Key Components of the DAG

The DAG consists of the following key components:

Data Extraction and Upload: Data extraction tasks are executed by PythonOperators, responsible for loading data from different CSV files. The **import_listings** task extracts **listing** data, the **import_2016Census_G01_NSW_LGA** task extracts **2016 Census demographic** data, and additional tasks focus on extracting other datasets related to Local Government Areas (LGA)

and suburbs. After extraction, the data is uploaded to a GCP bucket, facilitating centralized data storage.

Data Transformation: Each extraction task processes the extracted data, which includes data type conversion and cleaning. Data types of numeric columns are cast to integers or floats to ensure data integrity, and in one specific task (**import_NSW_LGA_SUBURB**), columns are selectively chosen for data loading.

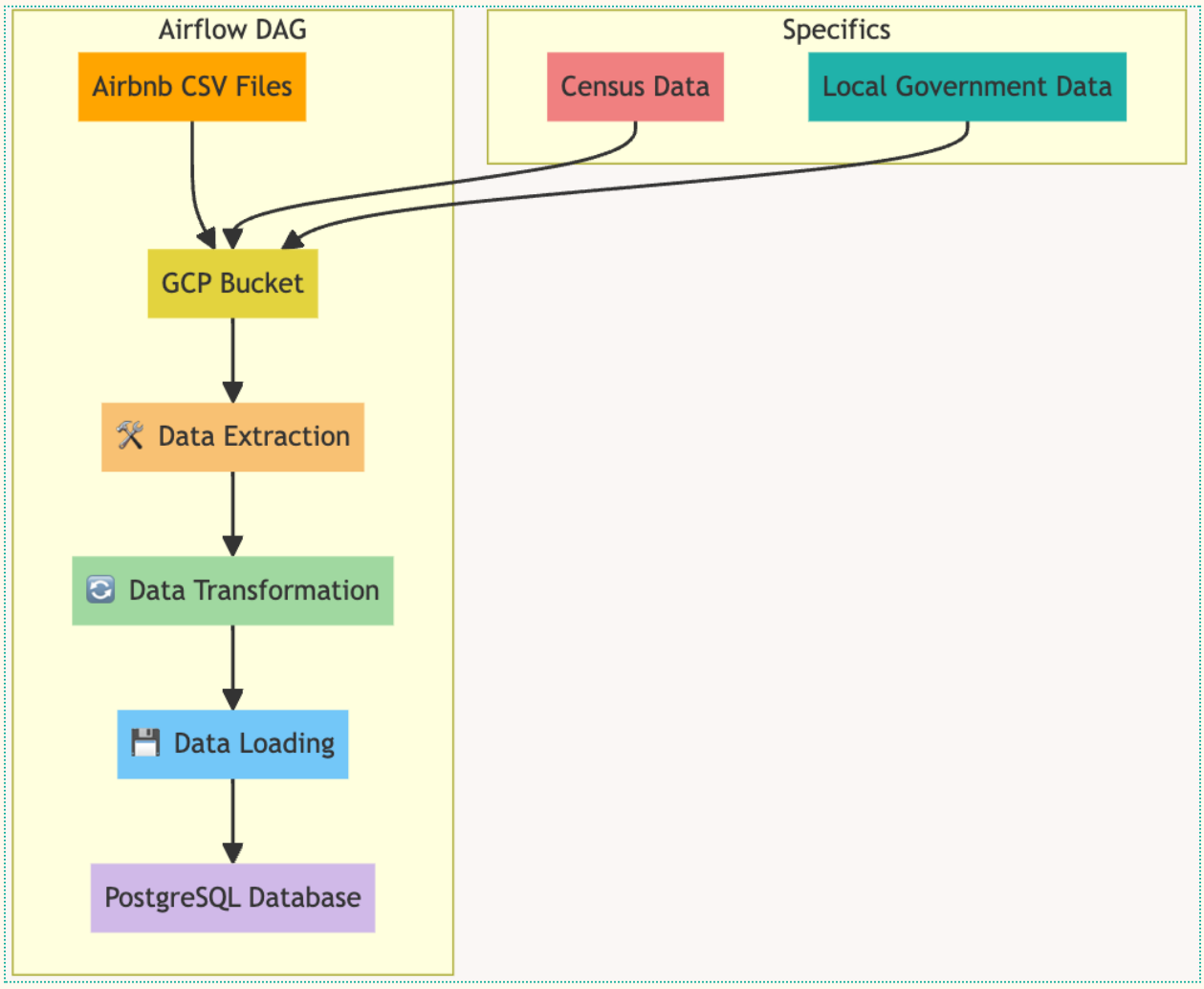


Figure 1

Data Loading: Data is loaded into the PostgreSQL database using the execute_values function, ensuring efficient and high-performance data insertion.

Automation and Consistency: The DAG's automation of data extraction, upload, transformation, and loading establishes a consistent and reliable workflow, reducing the potential for human errors.

Data Consistency and Integrity: Transformation processes, like data type conversion, maintain data consistency and integrity by casting columns into appropriate data types.

Selective Data Loading: Multiple tasks were set up to enable data export from GCP, for eg **import_NSW_LGA_SUBURB** task selectively loads specific columns, ensuring the insertion of only relevant data into the database, aligning with data relevance principles.

Detailed Logging: The DAG includes detailed logging, offering insights into the data extraction and loading process, which aids in monitoring and troubleshooting.

Data Source Diversity: The fact that DAG can accommodate data from various sources, is interesting. Although we had a unified data source, DAG can seamlessly integrate data from multiple touch points.

Snapshot Creation and SCD Handling for Staging Layer

After loading data from GCP into the raw schema of the PostgreSQL database, the process involves creating snapshots of key tables, including the host, property_type, and room_type tables. These snapshots serve as historical records and are essential for tracking changes over time.

Data Cleaning in 'staging' Schema: Data cleaning and standardization are carried out in the 'staging' schema to ensure data quality and consistency. Specifically, data type casting is applied to columns like **'lga_code_2016'** in the **'2016Census_G01_NSW_LGA'** and **'2016Census_G02_NSW_LGA'** tables. This consistency is crucial for reliable analysis and reporting. Null or NaN values are thoughtfully addressed to enhance data completeness and accuracy.

Slowly Changing Dimensions (SCD) Handling: Slowly Changing Dimensions (SCD) are managed to monitor changes over time. This is observed in the **'staging'** schema update of the **'raw.host_snapshot'** table, where **'dbt_valid_to'** values are set based on the lead date, and the

'dbt_scd_id' is used to track changes. SCD handling is vital for maintaining a historical record of data while ensuring data consistency.

Data Cleaning came with some challenges: Within the 'staging' schema, an important aspect of data cleaning involves standardizing the Local Government Area (LGA) names and codes in the **'nsw_lga_code'** data. This standardization process is crucial as it ensures that proper joins and data integration can be carried out effectively in the subsequent data warehousing layer. Consistent LGA names and codes are essential for maintaining data integrity and facilitating seamless analysis and reporting in the data mart environment.

Standardizing 'host_since' Column: The **'host_since'** date column presented issues with invalid values, including NaN entries. To address this, we employed a transformation that checked for null or non-standard date formats and standardized them. The transformation can be described as:

```
CASE

           WHEN host_since IS NULL OR host_since !~ '^[0-9]{1,2}/[0-9]{1,2}/[0-9]{4}$'
THEN '01/01/1900' ELSE host_since END as host_since,

       Scraped_date,

--Next Step was to change it to 'YYYY-MM-DD' format

TO_CHAR(TO_DATE(host_since, 'DD/MM/YYYY'), 'YYYY-MM-DD') as host_since,
```

Figure 2. Code Snippet

Host_ids conundrum: In the data, we noticed cases where multiple listings belong to multiple host_ids.

	host_id	listing_count
1	15,030	2
2		
3		
4	19,082	2
5	20,258	2
6	52,279	3
7	55,948	2
8	57,949	2
9	67,766	2
10	106,591	2
11	113,874	74

listing_count: int8 (Read-only: No corresponding table column)

	listing_id	host_count
1	249,158	2
2	1,051,911	2
3	1,226,451	2
4	1,352,336	2
5	1,765,417	2
6	2,000,728	2
7	2,811,738	2
8	3,055,867	2
9	3,327,165	2
10	3,911,220	2
11	5,041,030	2

Figure 3. Hos_ids vs Listing-ids

Managing Duplicate host_names: Similar to host_ids, we observed instances where the same host_id had multiple associated host_names.

This situation posed potential data integrity concerns. However, given the lack of contextual information, making transformative decisions was a delicate endeavor, as it carried the risk of inadvertently generating inaccurate reports.

Dealing with Missing host_neighbourhood: 'host_neighbourhood,' which represents a suburb, was missing from numerous entries. To address this, we aimed to approximate missing host_neighbourhood values. We initially attempted a more complex approach by estimating missing host_neighbourhoods based on the distribution of the number of host_ids residing in respective suburbs within the 'listing_neighbourhood' (an LGA). This distribution would have allowed us to select the suburb with the maximum number of hosts as a replacement for null host_neighbourhood entries. However, this approach proved computationally expensive, leading us to opt for a simpler solution. In the simpler approach, we selected the first suburb within the 'listing_neighbourhood' (LGA) that had a list of suburbs. This decision helped us maintain consistency in the absence of better reference data.

```
COALESCE(
            CASE
```

```
            WHEN h.host_neighbourhood = '' THEN s.suburb_name
            ELSE h.host_neighbourhood
        END,
        s.suburb_name
    ) AS host_neighbourhood
```

Figure 4. Code Snippet

Transformation of Listing Statistics: In the **'listings'** data, the process involves calculating mean and standard deviation statistics for the 'price' column. The purpose is to identify and address outliers within the dataset. A threshold is set at 1.5 times the standard deviation, ensuring that approximately 80% of the data falls within this range, effectively filtering out extreme values that could compromise data integrity.

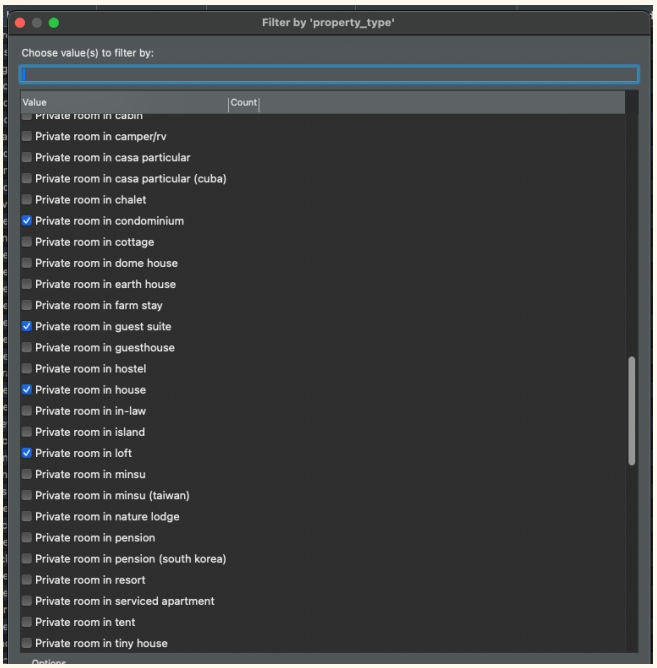
Handling listing_id with different listing_neighbourhood values: In the **'stg_property'** table, a data transformation step was applied to ensure data integrity and consistency. Specifically, the FIRST_VALUE function, combined with the PARTITION BY listing_id ORDER BY SCRAPED_DATE clause, was used to select the first non-null value of the 'listing_neighbourhood' column for each unique 'listing_id,' effectively resolving cases where multiple entries for the same 'listing_id' had different 'listing_neighbourhood' values. This approach eliminated redundancy and enhanced data consistency, contributing to the overall quality of the dataset.

```
stg_property AS (
   SELECT
  -- Cast host_id to integer
      CAST(listing_id AS INT) as listing_id, -- Cast listing_id to integer
          FIRST_VALUE(listing_neighbourhood) OVER (PARTITION BY listing_id ORDER BY
SCRAPED_DATE) AS listing_neighbourhood,


   FROM source…..
```

Figure 5. Code Snippet

Property_type naming discrepancy- The **property_type** naming within the dataset displays inconsistencies when compared to the standardized property types provided by Airbnb's official website(https://help.hostfully.com/en/articles/4823898-airbnb-property-type-room-type-how-to-manage-correctly-from-hostfully-pmp). These discrepancies pose challenges when attempting to match the dataset's property_type values with the officially recognized property types.



One intriguing observation is the property_type "shared room in tiny house," which corresponds to "Entire Home/apt" in the room_type. This inconsistency in the dataset leaves room for interpretation, making it difficult to definitively categorize the listings. To ensure data accuracy, these instances were retained as-is, as contextual information was insufficient to make a clear determination. This highlights the importance of maintaining standardized and consistent naming conventions in data collection and analysis to avoid such ambiguities.

Data Warehouse Layer:

In the data warehouse layer, we bring together the dimensions that have been processed and cleaned in the staging layer to create a comprehensive and integrated view of the data. This is where the actual data integration takes place, allowing for meaningful analyses and reporting

on Airbnb listings, their hosts, and their respective neighborhoods which we are going to discuss in the Datamart section.

Dimension Tables:

These tables contain essential dimensions that provide context to the fact table.

dim_G01: This dimension table includes data from 'stg_G01' and serves as a reference for government statistics in New South Wales.

dim_G02: It includes data from 'stg_G02' and provides additional government statistics relevant to our analysis.

dim_host: This dimension table contains information about Airbnb hosts, such as host names, registration dates, superhost status, and neighborhood affiliations.

dim_lga: It includes data from 'stg_lga' and represents Local Government Area information.

dim_property: This dimension table encompasses data related to Airbnb property listings, including neighborhood, property type, room type, and accommodation details.

dim_suburb: This dimension table offers insights into suburban data, which is crucial for understanding neighborhood dynamics.

Fact Table: The fact table is the centerpiece of the data warehouse layer, where data from various dimensions is joined to create a comprehensive view of Airbnb listings. This fact table is the foundation for analyses and reporting.

Datamart Layer

In the data mart layer, a set of SQL queries was executed to derive insightful metrics from the Airbnb listings. The first query focuses on host neighborhoods, calculating metrics such as the number of distinct hosts, estimated revenue, and estimated revenue per distinct host for various host neighborhoods and month/year combinations.

	host_neighbourhood_lga	month/year	Number of Distinct Hosts	Estimated Revenue	Estimated Revenue per Host (distinct)
1	Northern Beaches	2020-12-01 00:00:00.000 +1100	3,909	1,830,506.5191919192	468.2799997933
2	Northern Beaches	2021-01-01 00:00:00.000 +1100	3,860	1,827,856.1234126984	473.5378558064
3	Northern Beaches	2020-09-01 00:00:00.000 +1000	3,908	1,784,571.0527777778	456.6456122768
4	Northern Beaches	2021-02-01 00:00:00.000 +1100	3,849	1,765,249.9664141414	458.6256083175
5	Northern Beaches	2021-04-01 00:00:00.000 +1100	3,774	1,764,014.5490620491	467.412440133
6	Northern Beaches	2021-03-01 00:00:00.000 +1100	3,819	1,746,501.4862193362	457.3190589734
7	Northern Beaches	2020-05-01 00:00:00.000 +1000	4,079	1,672,874.1404401154	410.1186909635
8	Sydney	2020-09-01 00:00:00.000 +1000	5,481	1,671,726.896031746	305.0039948972
9	Northern Beaches	2020-11-01 00:00:00.000 +1100	3,826	1,668,403.5101010101	436.0699190018
10	Northern Beaches	2020-06-01 00:00:00.000 +1000	4,041	1,654,127.0168831169	409.3360596098
11	Northern Beaches	2020-10-01 00:00:00.000 +1000	3,837	1,647,995.9818542569	429.5011680621
12	Sydney	2020-05-01 00:00:00.000 +1000	5,855	1,611,092.2927128427	275.1652079783
13	Sydney	2020-06-01 00:00:00.000 +1000	5,795	1,600,647.6521645022	276.2118467928
14	Sydney	2021-01-01 00:00:00.000 +1100	5,296	1,508,935.425036075	284.9198310113
15	Northern Beaches	2020-08-01 00:00:00.000 +1000	3,582	1,499,137.8148629149	418.5197696435
16	Sydney	2020-12-01 00:00:00.000 +1100	5,330	1,482,147.3305916306	278.0764222498
17	Sydney	2021-02-01 00:00:00.000 +1100	5,261	1,476,738.0268398268	280.6953101767
18	Sydney	2021-03-01 00:00:00.000 +1100	5,216	1,455,320.2811327561	279.0107900945
19	Northern Beaches	2020-07-01 00:00:00.000 +1000	3,524	1,447,219.3831529582	410.6751938573
20	Sydney	2020-11-01 00:00:00.000 +1100	5,384	1,427,076.7531746032	265.0588323133
21	Sydney	2020-10-01 00:00:00.000 +1000	5,420	1,425,642.3341991342	263.0336409962
22	Sydney	2021-04-01 00:00:00.000 +1100	5,132	1,412,793.8410533911	275.2910836035
23	Sydney	2020-07-01 00:00:00.000 +1000	4,903	1,265,082.0377705628	258.0220350338
24	Sydney	2020-08-01 00:00:00.000 +1000	4,895	1,247,342.1831168831	254.8196492578
25	Waverley	2020-05-01 00:00:00.000 +1000	4,117	1,205,113.8641414141	292.7165081713
26	Waverley	2020-09-01 00:00:00.000 +1000	3,896	1,180,123.7882034632	302.9065164793
27	Waverley	2020-06-01 00:00:00.000 +1000	4,069	1,178,891.7838023088	289.7251864837
28	Overseas	2020-07-01 00:00:00.000 +1000	76	1,123,824.3041125541	14,787.1618962178
29	Waverley	2020-12-01 00:00:00.000 +1100	3,894	1,047,952.1193362193	269.1197019353
30	Randwick	2020-05-01 00:00:00.000 +1000	2,510	1,032,461.51504329	411.3392490212
31	Waverley	2021-01-01 00:00:00.000 +1100	3,859	1,029,517.7287518038	266.7835524104
32	Waverley	2021-02-01 00:00:00.000 +1100	3,838	1,021,946.1106060606	266.2704821798
33	Waverley	2021-03-01 00:00:00.000 +1100	3,799	1,003,929.3962121212	264.2614888687
34	Waverley	2020-10-01 00:00:00.000 +1000	3,868	1,002,649.032972583	259.2163994241
35	Waverley	2020-11-01 00:00:00.000 +1100	3,851	1,000,139.0349206349	259.7089158454
36	Waverley	2021-04-01 00:00:00.000 +1100	3,741	987,098.5795454545	263.8595508007
37	Waverley	2020-08-01 00:00:00.000 +1000	3,443	977,521.0228715729	283.9154873284
38	Waverley	2020-07-01 00:00:00.000 +1000	3,411	961,372.8306637807	281.8448638709

Table 1. Data Snippet

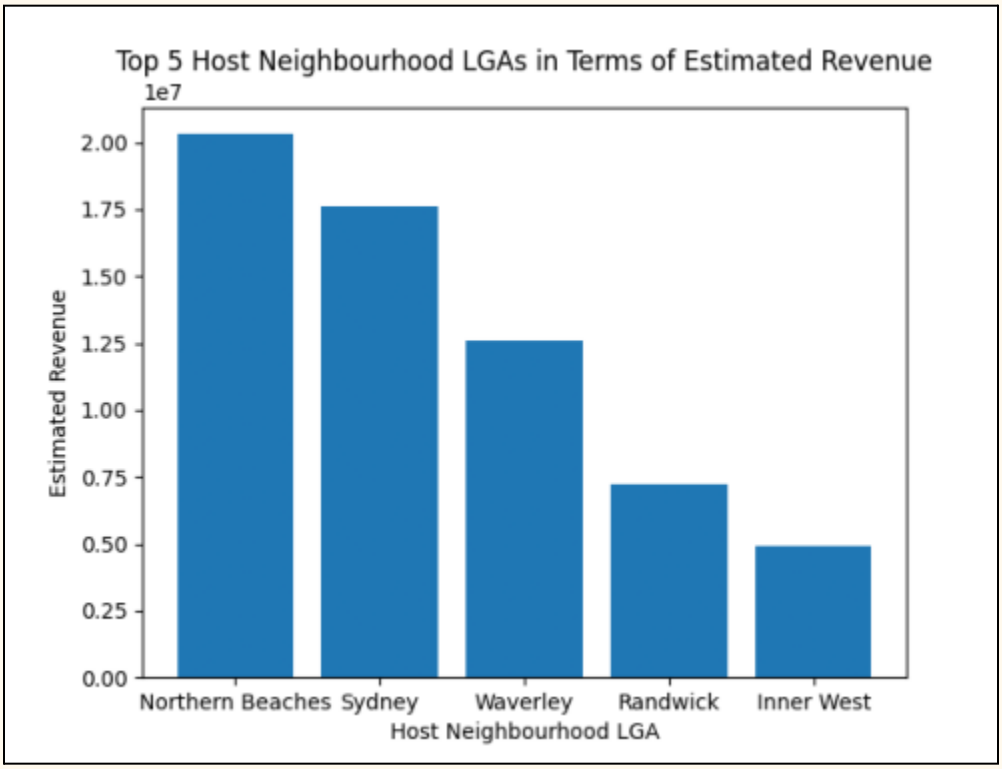


Figure 7

The data mart reveals the top 5 host_neighbourhood_lga areas, showcasing their strengths in the Airbnb market. Northern Beaches stands out with 46,008 distinct hosts and an impressive estimated revenue of $20,308,460, resulting in $5,296 in estimated revenue per host. Sydney, Waverley, Randwick, and Inner West also feature prominently with notable host counts and substantial estimated revenues, making them appealing regions for Airbnb hosting. These insights provide valuable guidance for hosts and industry stakeholders in understanding the performance of different host neighborhoods.

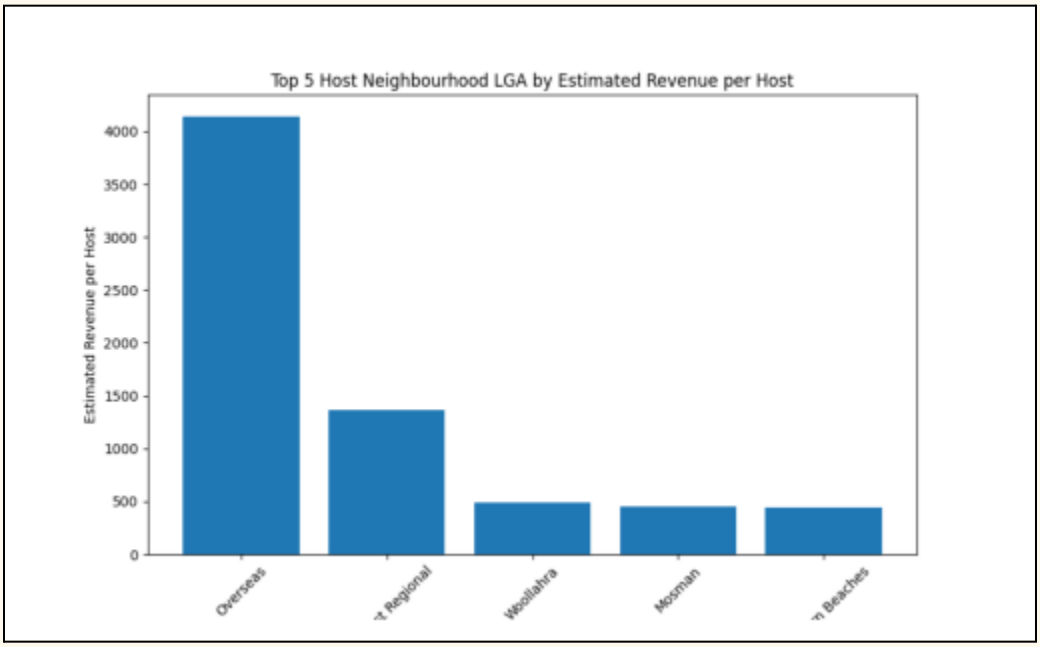


Figure 8

The data reveals the top 5 host_neighbourhood_lga areas with the highest estimated revenue per host, highlighting their lucrative potential for Airbnb hosts. Overseas leads the list with an exceptional revenue per host of $4,140, reflecting the strong earning capacity of these international locations. Bathurst Regional, Woollahra, Mosman, and Northern Beaches also stand out as attractive destinations for hosts, offering competitive revenue per host ratios. This information can guide hosts in selecting locations with high earning potential and optimizing their Airbnb listings.

The second query delves into neighborhood-level metrics, including the active listings rate, minimum and maximum prices, median price, average price, number of distinct hosts, superhost rate, average review scores rating, percentage changes in active and inactive listings, total stays, and average estimated revenue per active listing. These metrics provide a detailed analysis of Airbnb listings' performance in different neighborhoods.

	listing_ne	mont	Active List	Minimum P	Maximum Price	Median P	Average Pric	Number of Distinc	Superhost Rate	Average Review Scores	Percentage cl
	Bayside	-01 00:00:0	100	17.8333333333	3,872.6666666667	85	152.6733862002	1,220	11.5573770492	59.416460396	[NULL
	Bayside	-01 00:00:0	100	17.8333333333	3,872.6666666667	84	151.4845688919	1,201	11.3238967527	59.000630517	-1.856435643
	Bayside	-01 00:00:0	100	17.8333333333	3,872.6666666667	84.7765151515	161.8244128404	1,041	13.2564841499	57	-14.123581336
	Bayside	-01 00:00:0	100	17	2,904	79	111.1071307276	1,051	12.9400570885	58.2724637681	1.321585903
	Bayside	-01 00:00:0	100	17	2,904	78	107.7814222327	1,157	12.1002592913	58.3851937536	25.2898550725
	Bayside	-01 00:00:0	100	17	2,904	78	109.0993842908	1,140	11.6666666667	57.962962963	-12.5506072874
	Bayside	-01 00:00:0	100	15	2,904	78	111.7926655987	1,119	11.528150134	57.8690396239	-1.521164021
	Bayside	-01 00:00:0	100	15	2,904	80	114.7179784184	1,132	11.3074204947	56.5363091272	0.805910006
	Bayside	-01 00:00:0	100	15	2,904	80	127.6352323309	1,133	10.6796116505	55.6302910053	0.732844770
0	Bayside	-01 00:00:0	99.7317236754	15	2,904	80	123.6231404496	1,110	10.592459605	56.3221250841	-1.653439153
1	Bayside	-01 00:00:0	99.6572995202	15	2,904	80	120.8845105109	1,104	10.2795311091	56.5474552957	-2.219233355
2	Bayside	-01 00:00:0	99.1672449688	15	2,904	80	119.2929310853	1,080	10.0917431193	56.6836948915	-1.719394773
3	Blacktown	-01 00:00:0	100	23	881	65	89.516304555	239	12.9707112971	56.8364779874	[NULL
4	Blacktown	-01 00:00:0	100	23	880	65	89.7209477276	234	13.6752136752	55.6872964169	-3.459119496
5	Blacktown	-01 00:00:0	100	23	2,164	66	122.523168959	212	14.6226415094	54.0073800738	-11.726384364
6	Blacktown	-01 00:00:0	100	20	2,000	60	94.5661938534	216	13.4259259259	54.5815602837	4.059040590
7	Blacktown	-01 00:00:0	100	20	2,000	60	92.2978056426	220	13.6363636364	54.8495297806	13.120567375
8	Blacktown	-01 00:00:0	100	18	2,000	61	96.8560060442	216	11.5740740741	54.8647686833	-11.912225705
9	Blacktown	-01 00:00:0	100	18	2,000	61	92.4326900585	221	11.3122171946	53.9438596491	1.423487544
0	Blacktown	-01 00:00:0	99.649122807	22	2,000	61	92.7336267606	217	11.5207373272	55.0598591549	-0.350877193
1	Blacktown	-01 00:00:0	100	22	2,228	61	106.5790935673	213	10.3286384977	55.8947368421	0.352112676
2	Blacktown	-01 00:00:0	99.6376811594	22	2,000	63	93.5847107438	215	10.1851851852	55.12	-3.508771929
3	Blacktown	-01 00:00:0	98.1684981685	22	2,000	65	97.6173507463	214	10.6481481481	54.776119403	-2.545454545
4	Blacktown	-01 00:00:0	96	22	2,000	66	99.9362373737	213	10.0917431193	53.2916666667	-1.492537313
5	Burwood	-01 00:00:0	100	15	2,440	70	127.4624020916	169	11.2426035503	59.7638376384	[NULL
6	Burwood	-01 00:00:0	100	14	2,440	69	109.595072397	165	10.9090909091	59.8897338403	-2.952029520
7	Burwood	-01 00:00:0	100	14	2,164	70	121.7392770809	149	12.0805369128	57.8760683761	-11.026615969
8	Burwood	-01 00:00:0	100	15	530	66	92.2948752834	152	10.5263157895	61.4367346939	4.700854700
9	Burwood	-01 00:00:0	100	15	530	68.5	92.8884353741	158	10.1265822785	63.8197278912	2
0	Burwood	-01 00:00:0	100	15	530	67	90.280098832	162	9.2592592593	61.9396226415	-9.863945578
1	Burwood	-01 00:00:0	100	15	530	68.5	91.4901515152	162	9.2592592593	57.1363636364	-0.377358490
2	Burwood	-01 00:00:0	100	16	530	69.5	95.3030919886	159	9.4339622642	60.3435114504	-0.757575757
3	Burwood	-01 00:00:0	100	16	530	69	94.3773290598	156	9.6153846154	62.0961538462	-0.763358778
4	Burwood	-01 00:00:0	100	16	530	70	95.3619246862	152	9.2105263158	64.870292887	-8.076923076
5	Burwood	-01 00:00:0	100	16	530	70	96.9126811594	150	10	66.4	-3.765690376
6	Burwood	-01 00:00:0	99.5633187773	16	530	75	100.6086744639	149	10.6666666667	66.0701754386	-0.869565217
7	Camden	-01 00:00:0	100	35	406.25	91.1666666667	107.1392084106	41	24.3902439024	57.7755102041	[NULL

Table 2

listing_neighbourhood	Average Review Scores Rating	Median Price	Total Number of Stays
Sutherland Shire	72.192267	139.909091	126649
Hornsby	71.578543	90.000000	94215
Campbelltown	69.033799	80.386364	21320
Sydney	67.699841	120.000000	2227151
Penrith	67.491493	107.687500	28427

Table 3

This table presents a balance between the **top 5 listing_neighbourhoods**, taking into account both high **review scores** and **reasonable pricing** alongside the total number of stays. Sutherland Shire leads with an impressive average review score of 72.19, a median price of $139.91, and a substantial 126,649 stays. Hornsby offers competitive ratings, with a score of 71.58, a median price of $90, and 94,215 stays. Campbelltown, Sydney, and Penrith round out the list with their unique combinations of ratings, prices, and total stays. This information empowers both hosts

and travelers, enabling them to find neighborhoods that strike a balance between quality reviews and budget-friendly accommodations with a range of stay options.

The third query further refines the analysis by considering property type, room type, and accommodates. It evaluates metrics akin to the second query but categorizes the data by these specific dimensions, allowing for a more granular understanding of listing performance based on distinct attributes.

property_type	room_type	accommodates	month/year	Active Listings Rate	Minimum Price	Maximum Price	Median Price	Average Price	Number of Distinct Ho
Entire house	Entire home/apt	5	2021-01-01 00:00:00.000 +1100	430	70	2,000	300	360.9686821705	
Private room in townhou	Private room	6	2021-01-01 00:00:00.000 +1100	1	60	60	60	60	
Private room in house	Private room	2	2021-01-01 00:00:00.000 +1100	2,571	13	28,613	65	93.1112044909	
Entire townhouse	Entire home/apt	6	2021-01-01 00:00:00.000 +1100	143	62	1,657.5	300	347.5049093232	
Entire apartment	Entire home/apt	3	2021-03-01 00:00:00.000 +1100	1,269	33	1,078	135	155.705930463	
Private room in condom	Private room	4	2021-04-01 00:00:00.000 +1100	8	40	4,990	84	756	
Entire apartment	Entire home/apt	4	2021-03-01 00:00:00.000 +1100	4,185	16	10,000	180	219.7375201236	
Private room in apartme	Private room	4	2021-04-01 00:00:00.000 +1100	81	39	5,000	120	205.7952674897	
Private room in townhou	Private room	1	2021-03-01 00:00:00.000 +1100	204	15	9,021	50	109.8664215686	
Private room in townhou	Private room	1	2021-04-01 00:00:00.000 +1100	197	15	9,021	50	111.8819796954	
Shared room in house	Shared room	4	2021-03-01 00:00:00.000 +1100	4	30	99	47.5	56	
Shared room in house	Shared room	4	2021-04-01 00:00:00.000 +1100	4	30	99	47.5	56	
Private room in townhou	Private room	2	2021-03-01 00:00:00.000 +1100	452	23	14,315	69	177.4850663717	
Shared room in apartme	Shared room	2	2021-03-01 00:00:00.000 +1100	87	19	350	65	74.2998084291	
Entire house	Entire home/apt	2	2021-03-01 00:00:00.000 +1100	245	46	900	130	156.9758503401	
Entire apartment	Entire home/apt	1	2021-03-01 00:00:00.000 +1100	101	30	790	98	131.0594059406	
Private room in house	Private room	1	2021-03-01 00:00:00.000 +1100	1,060	18	14,589	50	79.0168082442	
Private room in townhou	Private room	3	2021-03-01 00:00:00.000 +1100	33	30	237	78	81.8787878788	
Private room in apartme	Private room	6	2021-04-01 00:00:00.000 +1100	6	46	280	140.5	156.1666666667	
Private room in guest su	Private room	1	2021-04-01 00:00:00.000 +1100	30	22	250	40	56.5	
Private room in bungalo	Private room	2	2021-03-01 00:00:00.000 +1100	36	25	10,000	60	619.3611111111	
Private room in bungalo	Private room	2	2021-04-01 00:00:00.000 +1100	36	25	10,000	60	620.5277777778	
Private room in house	Private room	2	2021-03-01 00:00:00.000 +1100	2,519	13	28,613	65	93.8095099466	
Private room in house	Private room	6	2021-04-01 00:00:00.000 +1100	28	25	1,150	145	239.1785714286	
Private room in apartme	Private room	1	2021-03-01 00:00:00.000 +1100	1,656	14	14,568	59	81.5152109867	
Private room in apartme	Private room	2	2021-03-01 00:00:00.000 +1100	4,206	13	12,867	70	93.5261810329	
Entire apartment	Entire home/apt	2	2021-03-01 00:00:00.000 +1100	5,072	30	12,000	125	154.4093074873	
Entire house	Entire home/apt	4	2021-04-01 00:00:00.000 +1100	783	40	4,000	210	259.8961094248	
Private room in townhou	Private room	2	2021-04-01 00:00:00.000 +1100	441	23	14,315	68	180.222297808	
Private room in house	Private room	1	2021-04-01 00:00:00.000 +1100	1,044	18	14,589	50	79.6586211734	
Private room in house	Private room	4	2021-04-01 00:00:00.000 +1100	161	24	999	90	123.483436853	
Private room in apartme	Private room	1	2021-04-01 00:00:00.000 +1100	1,643	14	14,568	60	81.7239348752	
Entire house	Entire home/apt	4	2021-03-01 00:00:00.000 +1100	791	40	10,000	210	269.7448377581	
Shared room in apartme	Shared room	2	2021-04-01 00:00:00.000 +1100	83	19	350	68	75.9648594378	
Private room in apartme	Private room	2	2021-01-01 00:00:00.000 +1100	4,274	13	12,902	70	92.4660793121	
Entire apartment	Entire home/apt	1	2021-04-01 00:00:00.000 +1100	101	30	790	98	130.5643564356	
Entire house	Entire home/apt	2	2021-04-01 00:00:00.000 +1100	238	46	900	132.5	161.3256069094	

Refresh · Save · Cancel · Export data · 200 · 200+ · 200 row(s) fetched - 857ms, on 2023-11-07 at 18:32:26

Table 4. Data Snippet

	Top 5 Property Types (Min Price)	Top 5 Room Types (Min Price)	Top 5 Accommodates (Min Price)	Top 5 Minimum Prices	Top 5 Property Types (Max Price)	Top 5 Room Types (Max Price)	Top 5 Accommodates (Max Price)	Top 5 Maximum Prices	Bottom 5 Property Types (Min Price)	Bottom 5 Room Types (Min Price)	Bottom 5 Accommodates (Min Price)	Bottom 5 Minimum Prices	Bottom 5 Property Types (Max Price)	Bottom 5 Room Types (Max Price)	Bottom 5 Accommodates (Max Price)	Bottom 5 Maximum Prices
0	Boat	Entire home/apt	10	23000.000000	Private room in house	Private room	2	26208.000000	Shared room in apartment	Shared room	3	6.555556	Tiny house	Shared room	1	14.954545
1	Entire home/apt	Entire home/apt	9	6633.000000	Boat	Entire home/apt	10	23000.000000	Apartment	Shared room	3	10.025253	Shared room in tiny house	Shared room	1	15.000000
2	Private room in cabin	Private room	6	3453.000000	House	Private room	2	17993.303030	Entire apartment	Entire home/apt	6	11.000000	Barn	Private room	1	16.000000
3	Cabin	Private room	6	3452.972222	Private room in apartment	Private room	3	15000.000000	Apartment	Private room	2	12.972222	Room in bed and breakfast	Hotel room	4	19.777778
4	Private room in boat	Private room	4	3311.440000	Apartment	Private room	3	14999.972222	Private room in apartment	Private room	2	13.000000	Shared room in bed and breakfast	Shared room	6	20.000000

Table 5- top and bottom 5 categories for property types, room types, and accommodates

The table highlights the top and bottom 5 categories for property types, room types, and accommodations based on minimum and maximum prices for active listings. Notable findings

include high minimum prices for boats and entire home/apartment rentals, while shared rooms and tiny houses offer more budget-friendly options. Boats and apartments also command high maximum prices, whereas shared rooms in bed and breakfasts and hotels offer more affordable choices. This information is valuable for both hosts and guests when making pricing and booking decisions.

Collectively, these data mart queries furnish a comprehensive evaluation of Airbnb listings, facilitating data-driven insights and strategic decision-making in the short-term rental market.

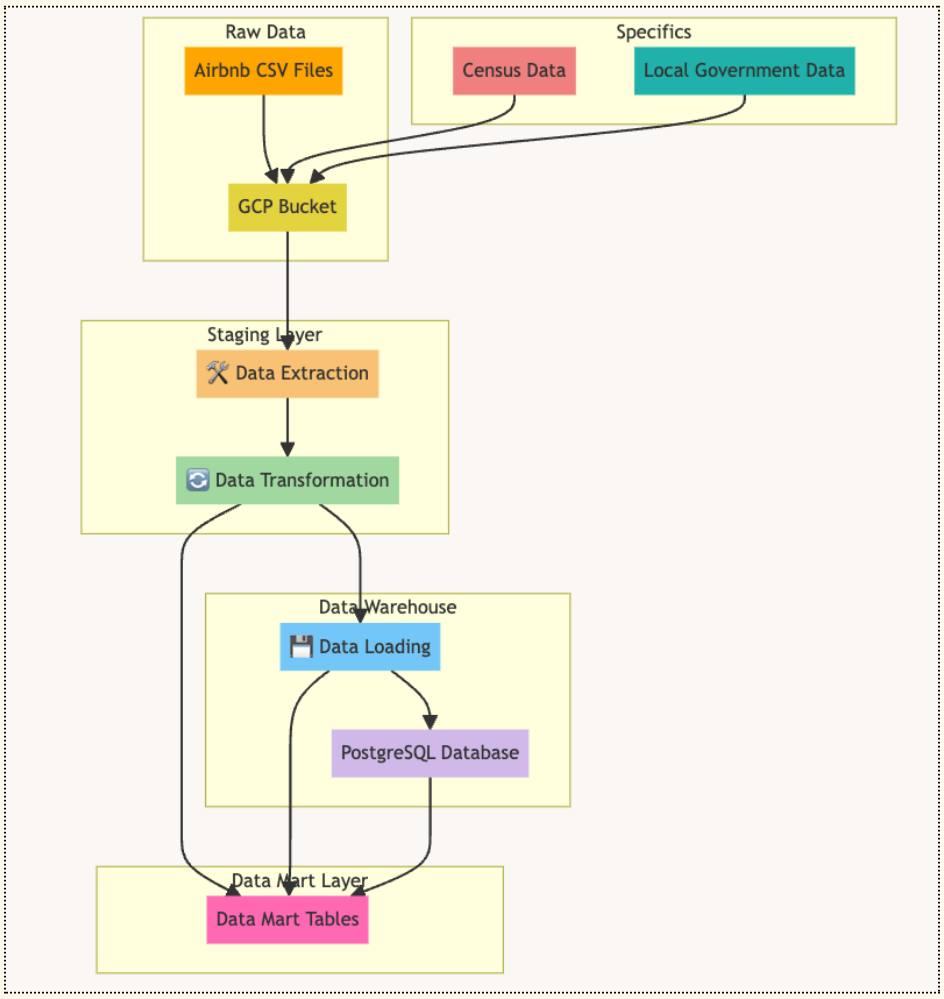


Figure 9

Ad-Hoc Analysis

1. We aimed to extract demographic information for the best and worst-performing "listing_neighbourhoods" in terms of estimated revenue per active listings over the last 12 months.

The analysis reveals intriguing insights into the disparities between the best and worst-performing "listing_neighbourhoods" from an estimated revenue perspective over the past year. While the analysis is based on factual data, it's essential to consider potential factors contributing to the observed differences critically.

In the competition of neighborhoods, **Mosman** takes the lead with a thriving population, age diversity, and youthful energy. Its sizable population and balanced age groups create a broad customer base, while the presence of young adults aged 25-29 fuels Airbnb demand and revenue.

abc listing_neighbourhood	123 lga_code	123 tot_p_m	123 tot_p_f	123 age_0_4_yr_m	123 age_0_4_yr_f	123 age_5_14_yr_m
Mosman	15,350	13,189	15,290	757	732	1,820

Table 6. Best Neighbourhood

Fairfield, on the other hand, is the underdog with a cozier population and age variance. Although it has its share of youthful potential, its smaller size and diverse age groups might be influencing factors, limiting its Airbnb revenue growth.

2. To optimize revenue and stays in the top-performing "listing_neighbourhoods," focusing on specific listing attributes is essential. Our analysis found that the property type, room type, and accommodation capacity can significantly impact the number of stays.The order of results was determined by the average review scores and the number of reviews. These factors are vital in attracting guests and promoting longer stays. A higher average review score and a greater number of reviews indicate guest satisfaction and reliability, driving more bookings and stays.

In the top 5 "**listing_neighbourhoods**" with high estimated revenue per active listing, the following listing attributes stood out: Northern Beaches offers an "Entire floor" property for two guests with a 100 average review score, 134 reviews, and an estimated revenue of $20,114.25

(201 nights of stays).

listing_neighbourhood	property_type	room_type	accommodates	total_stays	total_estimated_revenue	avg_review_scores_rating	total_number_of_reviews
Northern Beaches	Entire floor	Entire home/apt	2	134	20,114.25	100	771
Penrith	Entire guesthouse	Entire home/apt	2	156	18,720	100	611
The Hills Shire	Entire villa	Entire home/apt	4	195	27,300	100	586
Randwick	Entire cabin	Entire home/apt	1	171	12,162	100	395
Northern Beaches	Island	Entire home/apt	2	85	25,500	100	392

Table 6. Data Snippet

Since we lack detailed information about the individual profitability of each listing, taking the average of review scores and the sum of the number of reviews in the order by clause strikes a balance between maximizing profitability and ensuring reliability.

3. In our analysis, we sought to determine whether hosts with multiple listings tend to place those listings within the same Local Government Area (LGA) as their place of residence. To address this question, we identified hosts with more than one listing and examined whether their "host_neighbourhood_lga_name" (the LGA where the host's property is located) matched the "listing_neighbourhood" (the neighborhood where the listing is situated).

The partial results indicated that many hosts with multiple listings indeed placed those listings within the same LGA as their residence. Here are a few sample records that demonstrate this alignment:

host_id	listing_id	host_neighbourhood_lga_name	listing_neighbourhood
15,030	32,124,043	Cumberland	Cumberland
15,030	28,720,008	Cumberland	Cumberland
17,061	73,639	Sydney	Sydney
17,061	12,351	Sydney	Sydney
17,331	26,430,130	Waverley	Waverley
17,331	21,283,213	Waverley	Waverley
19,082	48,673,525	Sydney	Inner West
19,082	7,824,130	Sydney	Sydney
20,258	30,094,181	Sydney	Sydney
20,258	39,581,614	Sydney	Sydney
52,279	24,956,575	Sydney	Sydney
52,279	23,398,484	Sydney	Sydney
52,279	23,525,381	Sydney	Sydney
55,948	4,590,307	Northern Beaches	Northern Beaches
55,948	14,250	Northern Beaches	Northern Beaches
57,949	5,491,278	Sydney	Sydney
57,949	2,426,786	Sydney	Sydney
67,766	21,652,732	Sydney	Sydney
67,766	4,832,426	Sydney	Sydney
106,591	20,559,017	Northern Beaches	Northern Beaches

Table 8. Data Snippet

Calculating the percentage of hosts with multiple listings who choose to have their listings in the same LGA as their residence revealed that approximately **94.94%** of hosts exhibit this behavior. It suggests that most hosts opt for the convenience and familiarity of managing listings in their local area.

4. We wanted to plot a graph to show whether hosts' estimated revenue can cover their listing neighborhood's median mortgage which is looked up from census data. We'll distinguish between 'Yes' and 'No'."

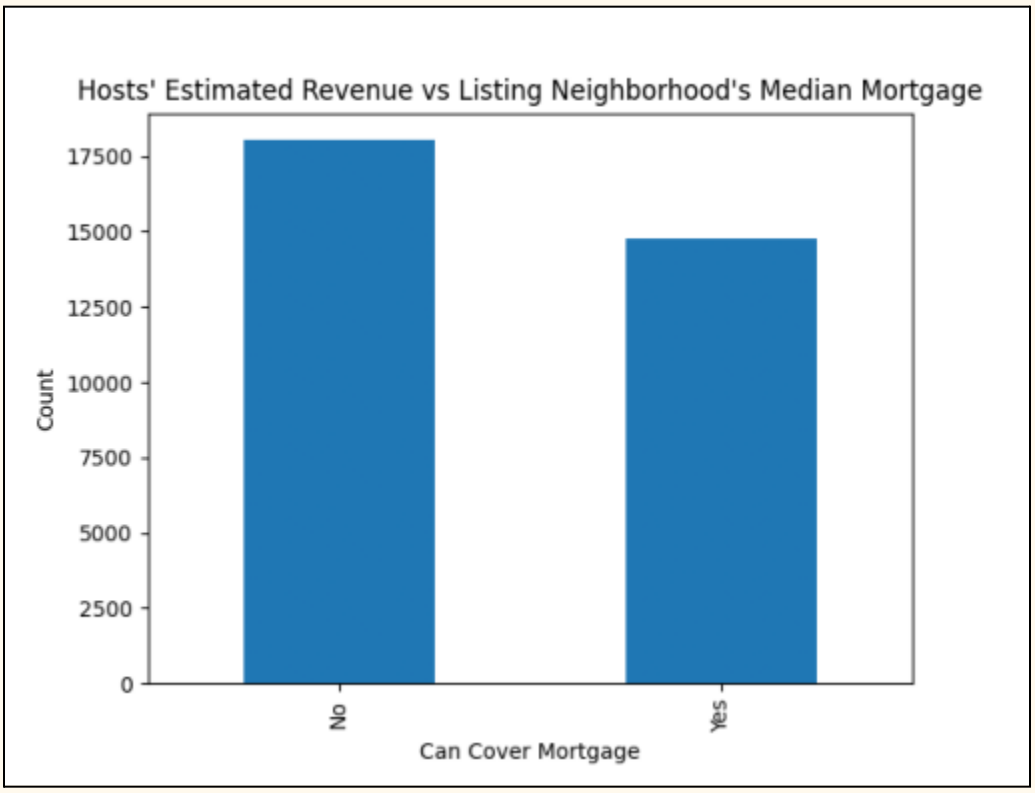


Figure 10. Can Cover Mortgage

The analysis reveals that only 45.02% of hosts' estimated revenue can cover their listing neighborhood's median mortgage. This finding suggests that for a substantial portion of hosts, the revenue they generate from renting out their properties may fall short of covering the annualized median mortgage expenses in their respective listing neighborhoods. Hosts in these areas may be faced with a financial challenge, as they might need to rely on other sources of

income to bridge the gap. Understanding the dynamics of this balance between estimated revenue and mortgage costs is essential for both hosts and potential property investors to make informed decisions about their short-term rental endeavors in various neighborhoods.

Conclusion:

In summary, the development and implementation of an ELT data pipeline using Apache Airflow and dbt have successfully processed and analyzed Airbnb and Census data. This report has covered the data flow, key components, and challenges encountered in the pipeline.

The data pipeline has automated data extraction, transformation, and loading, ensuring data integrity and consistency. Slowly Changing Dimensions (SCD) handling and data cleaning in the 'staging' schema have further enhanced data quality.

The data mart layer has provided valuable insights through SQL queries, offering a comprehensive view of host neighborhoods, neighborhood performance, and listing attributes. The ad-hoc analysis has shed light on the best and worst-performing neighborhoods and the factors influencing host revenue and stays.

While this project has yielded significant insights, it's important to note that data analysis is an evolving process. As market dynamics change, so should the data pipeline and analysis methods. This project underscores the importance of data-driven decision-making in the short-term rental and demographic analysis domains.

Appendix

1. SQL script to calculate the distribution of host_ids per suburb in a given listing neighborhood. .

```sql
WITH max_frequency_per_listing AS (
    SELECT
        listing_neighbourhood,
        host_neighbourhood,
        MAX(frequency) AS max_frequency
    FROM (
        SELECT
            listing_neighbourhood,
            host_neighbourhood,
            COUNT(host_id) AS frequency
        FROM listings
        WHERE host_neighbourhood IS NOT NULL AND host_neighbourhood <> ''
        GROUP BY listing_neighbourhood, host_neighbourhood
    ) AS subquery
    GROUP BY listing_neighbourhood, host_neighbourhood
)

UPDATE public.listings AS t1
SET host_neighbourhood = t2.host_neighbourhood
FROM (
    SELECT
        l.listing_neighbourhood,
        l.host_neighbourhood,
        m.max_frequency
    FROM listings l
    JOIN max_frequency_per_listing m
    ON l.listing_neighbourhood = m.listing_neighbourhood
   WHERE l.host_neighbourhood IS NOT DISTINCT FROM 'NaN' OR l.host_neighbourhood IS NULL
) AS t2;
```

2. Host ids having multiple listing id and vice versa.

```sql
SELECT host_id, COUNT(DISTINCT listing_id) AS listing_count
FROM public.listings
GROUP BY host_id
HAVING COUNT(DISTINCT listing_id) > 1;
```

```sql
SELECT listing_id, COUNT(DISTINCT host_id) AS host_count
FROM public.listings
GROUP BY listing_id
HAVING COUNT(DISTINCT host_id) > 1;
```

3. The code extracts and summarizes the top 5 listing neighborhoods with the highest average review scores, along with their median prices and total stays from a DataFrame.

```
df = dfs[0]
```

```
    grouped_df = df.groupby('listing_neighbourhood').agg({'Average Review Scores Rating': 'mean', 'Median Price': 'median', 'Total Number of Stays': 'sum'})
```

```
    sorted_df = grouped_df.sort_values('Average Review Scores Rating', ascending=False)

    top_5_neighbourhoods = sorted_df.head(5)

    result = top_5_neighbourhoods[['Average Review Scores Rating', 'Median Price', 'Total Number of Stays']]

    return {'type': 'dataframe', 'value': result}
```

4. The code analyzes and presents the top and bottom 5 combinations of property types, room types, and accommodates based on average minimum and maximum prices in a DataFrame.

```
grouped_data = dfs[0].groupby(['property_type', 'room_type', 'accommodates'])

    avg_min_price = grouped_data['Minimum Price'].mean()

    avg_max_price = grouped_data['Maximum Price'].mean()

    top_min_price = avg_min_price.nlargest(5)

    top_max_price = avg_max_price.nlargest(5)

    bottom_min_price = avg_min_price.nsmallest(5)

    bottom_max_price = avg_max_price.nsmallest(5)

    result_df = pd.DataFrame({'Top 5 Property Types (Min Price)': top_min_price.index.get_level_values('property_type'), 'Top 5 Room Types (Min Price)': top_min_price.index.get_level_values('room_type'), 'Top 5 Accommodates (Min Price)': top_min_price.index.get_level_values('accommodates'), 'Top 5 Minimum Prices': top_min_price.values, 'Top 5 Property Types (Max Price)': top_max_price.index.get_level_values('property_type'), 'Top 5 Room Types (Max Price)': top_max_price.index.get_level_values('room_type'), 'Top 5 Accommodates (Max Price)': top_max_price.index.get_level_values('accommodates'), 'Top 5 Maximum Prices': top_max_price.values, 'Bottom 5 Property Types (Min Price)': bottom_min_price.index.get_level_values('property_type'), 'Bottom 5 Room Types (Min Price)': bottom_min_price.index.get_level_values('room_type'), 'Bottom 5 Accommodates (Min Price)': bottom_min_price.index.get_level_values('accommodates'), 'Bottom 5 Minimum Prices': bottom_min_price.values, 'Bottom 5 Property Types (Max Price)': bottom_max_price.index.get_level_values('property_type'), 'Bottom 5 Room Types (Max Price)': bottom_max_price.index.get_level_values('room_type'), 'Bottom 5 Accommodates (Max Price)': bottom_max_price.index.get_level_values('accommodates'), 'Bottom 5 Maximum Prices': bottom_max_price.values})

    return {'type': 'dataframe', 'value': result_df}
```

5. The code calculates the percentage of distinct host IDs with matching "host_neighbourhood_lga_name" and "listing_neighbourhood" values in the DataFrame.

```
total_host_ids = dfs[0]['host_id'].nunique()

    matching_host_ids = dfs[0][dfs[0]['host_neighbourhood_lga_name'] == dfs[0]['listing_neighbourhood']]['host_id'].nunique()
```

```
    percentage = matching_host_ids / total_host_ids * 100

    result = {'type': 'string', 'value': f'The percentage of distinct host_id who have host_neighbourhood_lga_name = listing_neighbourhood_lga_name is {percentage:.2f}%.'}

    return result
```

5.

The code generates a bar chart showing the count of hosts categorized by whether they can cover their listing neighborhood's median mortgage with their estimated revenue. The chart is saved as an image file.

```
grouped = dfs[0].groupby('can_cover_mortgage').count()

    fig, ax = plt.subplots()

    grouped['host_id'].plot(kind='bar', ax=ax)

    ax.set_xlabel('Can Cover Mortgage')

    ax.set_ylabel('Count')

    ax.set_title("Hosts' Estimated Revenue vs Listing Neighborhood's Median Mortgage")

    plt.savefig('/content/temp_chart.png')

    plt.close(fig)

    return {'type': 'plot', 'value': '/content/temp_chart.png'}
```