Analyzing the Polish Housing Market: An End-to-End Data Analytics Project

Efthimios Vlahos

Project Overview

- The project aims to offer an in-depth analysis of the housing market in major Polish cities, providing valuable insights into property pricing, availability, and characteristics. By leveraging an end-to-end data analytics framework, the analysis covers everything from data extraction to transformation and final reporting.
- The primary data source for this study is the Otodom website, a prominent real estate platform in Poland. Data has been scraped using Bright Data and is subsequently stored in a Snowflake cloud-based data warehouse for robust analysis. Additionally, the dataset includes variables like location, number of rooms, size, and advertiser type, among others.
- The objectives are multi-faceted, ranging from understanding the average rental and sale prices in different cities and neighborhoods to investigating the nature of advertisements (private vs. business). The study employs Python for data transformation, SQL for data querying, and Google Sheets for further transformation and visualization, aiming to provide actionable insights for both potential homeowners and real estate stakeholders.

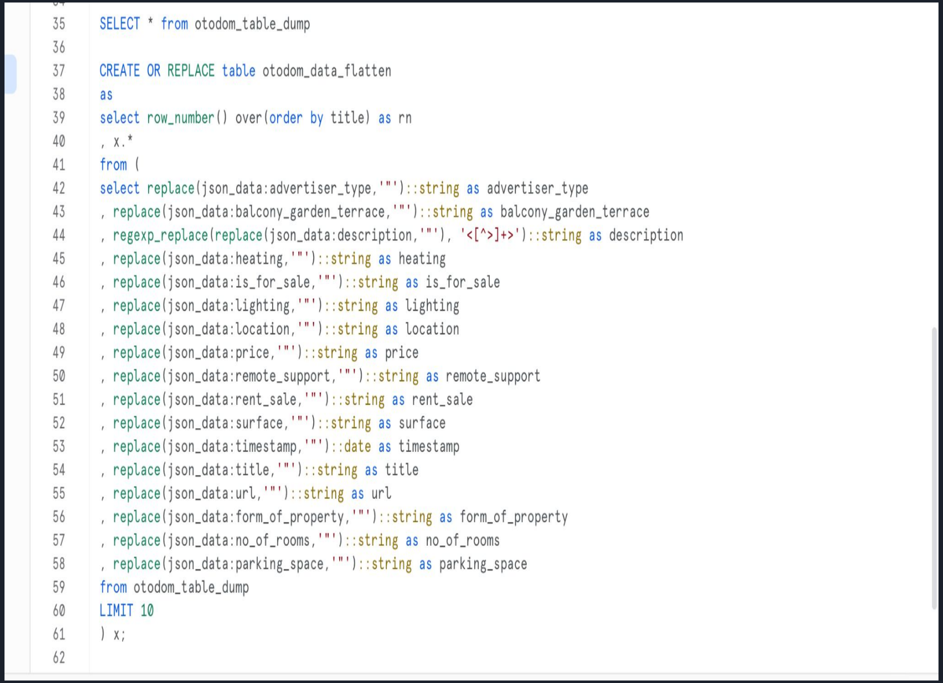
Tools and Technologies

- Python: A versatile programming language used for data extraction, transformation, and initial analysis. Python serves as the backbone for our data manipulation and transformation tasks.
- SQL: Structured Query Language (SQL) is used for querying and manipulating the data stored in Snowflake, our cloud-based data warehouse.
- Google Sheets: This accessible platform acts as both a data transformation and visualization tool. It is integrated with Python and Snowflake for seamless data flow and is also used for some specific transformations using Google Translate API.
- Snowflake: Our cloud-based data warehouse where all extracted and transformed data is stored. Bright Data: A leading data collection platform used for scraping data from the Otodom website. It provides the raw data that fuels our analyses.
- Geocode and Google Translate APIs: These APIs are employed for specific data transformations. The Geocode API helps in standardizing and enriching location data, while the Google Translate API is used to translate Polish text fields into English, making the data more universally accessible.

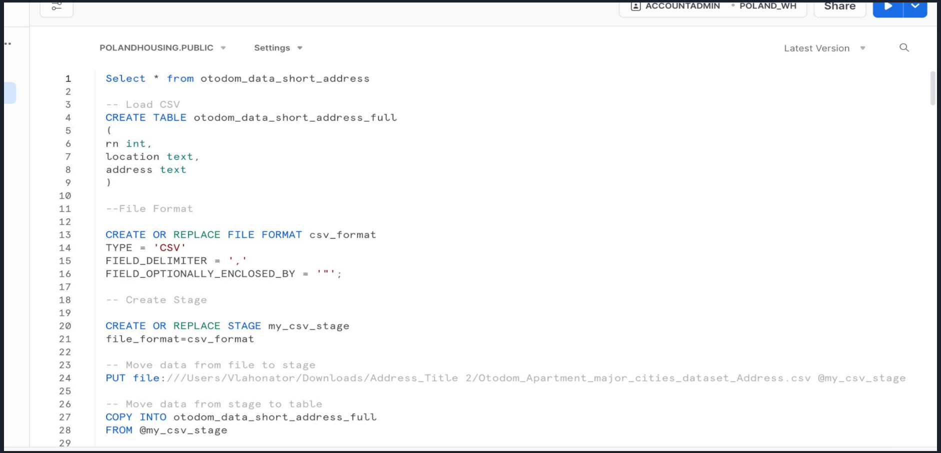
Data Ingestion

- Bright Data is our primary tool for data collection. It enables me to scrape listings and various details such as price, location, size, and other attributes from the Otodom website, a Polish real estate platform.
- Once the data is scraped using Bright Data, it is immediately exported into our cloud-based data warehouse, Snowflake. This ensures that we have a centralized location for all our raw data, making it easier to conduct further transformations and analyses.
- The data scraped comes in a nested JSON format. This complicates matters as relational databases like Snowflake work better with flat tables. Therefore, a critical part of our ingestion process is to 'flatten' the JSON objects to make the data more suitable for SQL-based analysis in Snowflake.

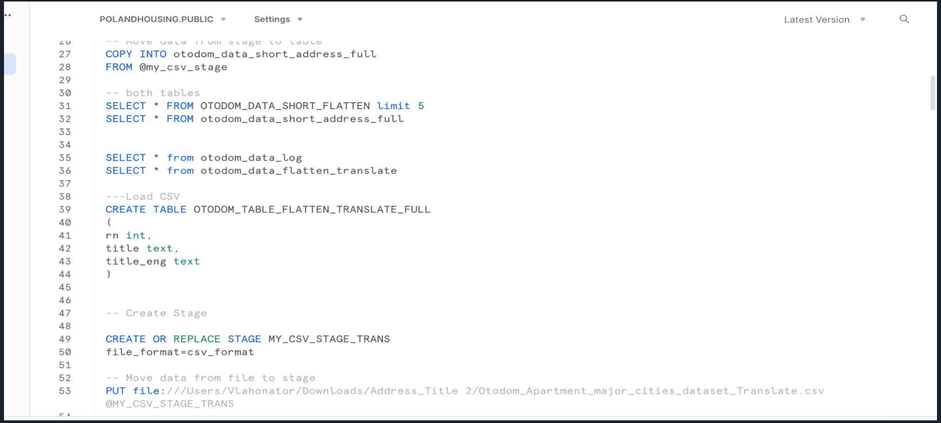
Snippets of Data Ingestion



```sql
SELECT * from otodom_table_dump

CREATE OR REPLACE table otodom_data_flatten
as
select row_number() over(order by title) as rn
, x.*
from (
select replace(json_data:advertiser_type,'"')::string as advertiser_type
, replace(json_data:balcony_garden_terrace,'"')::string as balcony_garden_terrace
, regexp_replace(replace(json_data:description,'"'), '<[^>]+>')::string as description
, replace(json_data:heating,'"')::string as heating
, replace(json_data:is_for_sale,'"')::string as is_for_sale
, replace(json_data:lighting,'"')::string as lighting
, replace(json_data:location,'"')::string as location
, replace(json_data:price,'"')::string as price
, replace(json_data:remote_support,'"')::string as remote_support
, replace(json_data:rent_sale,'"')::string as rent_sale
, replace(json_data:surface,'"')::string as surface
, replace(json_data:timestamp,'"')::date as timestamp
, replace(json_data:title,'"')::string as title
, replace(json_data:url,'"')::string as url
, replace(json_data:form_of_property,'"')::string as form_of_property
, replace(json_data:no_of_rooms,'"')::string as no_of_rooms
, replace(json_data:parking_space,'"')::string as parking_space
from otodom_table_dump
LIMIT 10
) x;
```



```sql
Select * from otodom_data_short_address

-- Load CSV
CREATE TABLE otodom_data_short_address_full
(
rn int,
location text,
address text
)

--File Format

CREATE OR REPLACE FILE FORMAT csv_format
TYPE = 'CSV'
FIELD_DELIMITER = ','
FIELD_OPTIONALLY_ENCLOSED_BY = '"';

-- Create Stage

CREATE OR REPLACE STAGE my_csv_stage
file_format=csv_format

-- Move data from file to stage
PUT file:///Users/Vlahonator/Downloads/Address_Title 2/Otodom_Apartment_major_cities_dataset_Address.csv @my_csv_stage

-- Move data from stage to table
COPY INTO otodom_data_short_address_full
FROM @my_csv_stage
```

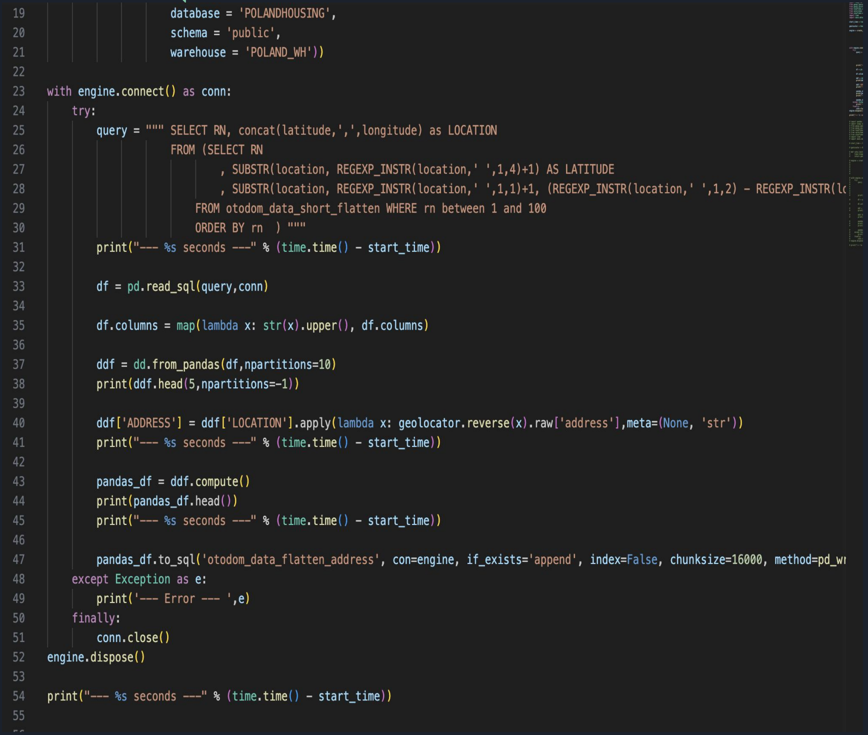


```sql
COPY INTO otodom_data_short_address_full
FROM @my_csv_stage

-- both tables
SELECT * FROM OTODOM_DATA_SHORT_FLATTEN limit 5
SELECT * FROM otodom_data_short_address_full


SELECT * from otodom_data_log
SELECT * from otodom_data_flatten_translate

---Load CSV
CREATE TABLE OTODOM_TABLE_FLATTEN_TRANSLATE_FULL
(
rn int,
title text,
title_eng text
)


-- Create Stage

CREATE OR REPLACE STAGE MY_CSV_STAGE_TRANS
file_format=csv_format

-- Move data from file to stage
PUT file:///Users/Vlahonator/Downloads/Address_Title 2/Otodom_Apartment_major_cities_dataset_Translate.csv
@MY_CSV_STAGE_TRANS
```

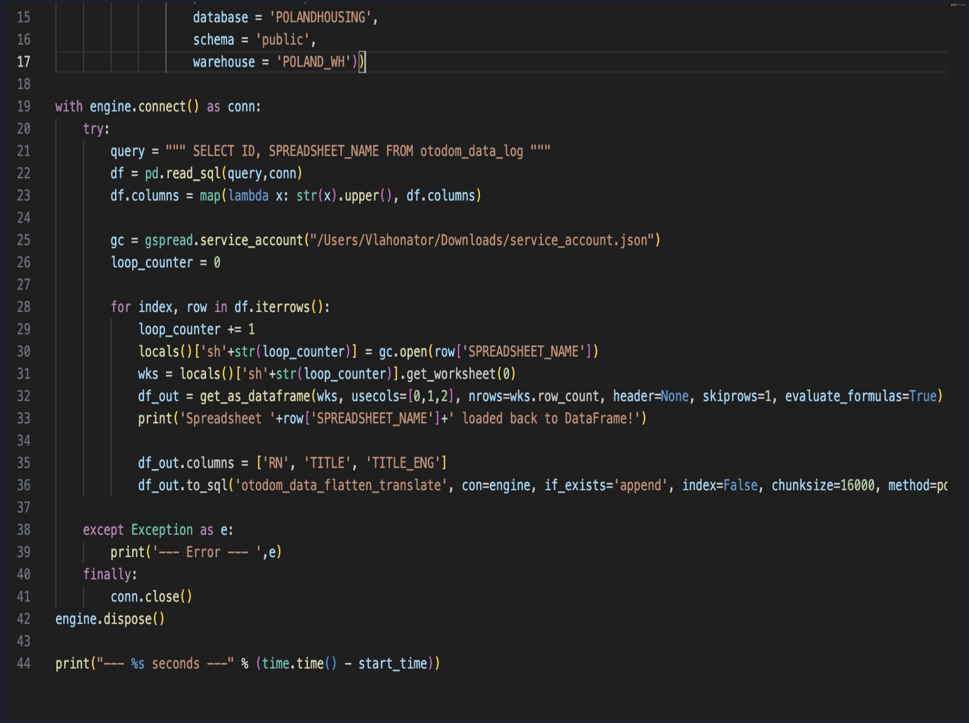
Data Transformation

- In this phase, we leverage Python in conjunction with the Geocode API to transform and enrich the location data. This allows us to convert raw address details into standardized geolocations, thereby improving the quality and usability of the location-based fields in our dataset.
- Transformed 'Title' from Polish to English One unique challenge we faced was that the title text for the listings was primarily in Polish. To make this data universally understandable, we used Google Sheets integrated with the Google Translate API to automatically translate these titles into English.
- After these multiple layers of transformation—spanning geolocation standardization and language translation—the newly transformed data is loaded back into our Snowflake data warehouse. This ensures that all subsequent analyses are based on high-quality, standardized data.

Snippets of Data Transformation



```python
                        database = 'POLANDHOUSING',
                        schema = 'public',
                        warehouse = 'POLAND_WH'))

with engine.connect() as conn:
    try:
        query = """ SELECT RN, concat(latitude,',',longitude) as LOCATION
                    FROM (SELECT RN
                            , SUBSTR(location, REGEXP_INSTR(location,' ',1,4)+1) AS LATITUDE
                            , SUBSTR(location, REGEXP_INSTR(location,' ',1,1)+1, (REGEXP_INSTR(location,' ',1,2) - REGEXP_INSTR(lc
                        FROM otodom_data_short_flatten WHERE rn between 1 and 100
                        ORDER BY rn  ) """
        print("--- %s seconds ---" % (time.time() - start_time))

        df = pd.read_sql(query,conn)

        df.columns = map(lambda x: str(x).upper(), df.columns)

        ddf = dd.from_pandas(df,npartitions=10)
        print(ddf.head(5,npartitions=-1))

        ddf['ADDRESS'] = ddf['LOCATION'].apply(lambda x: geolocator.reverse(x).raw['address'],meta=(None, 'str'))
        print("--- %s seconds ---" % (time.time() - start_time))

        pandas_df = ddf.compute()
        print(pandas_df.head())
        print("--- %s seconds ---" % (time.time() - start_time))

        pandas_df.to_sql('otodom_data_flatten_address', con=engine, if_exists='append', index=False, chunksize=16000, method=pd_wr
    except Exception as e:
        print('--- Error --- ',e)
    finally:
        conn.close()
engine.dispose()

print("--- %s seconds ---" % (time.time() - start_time))
```

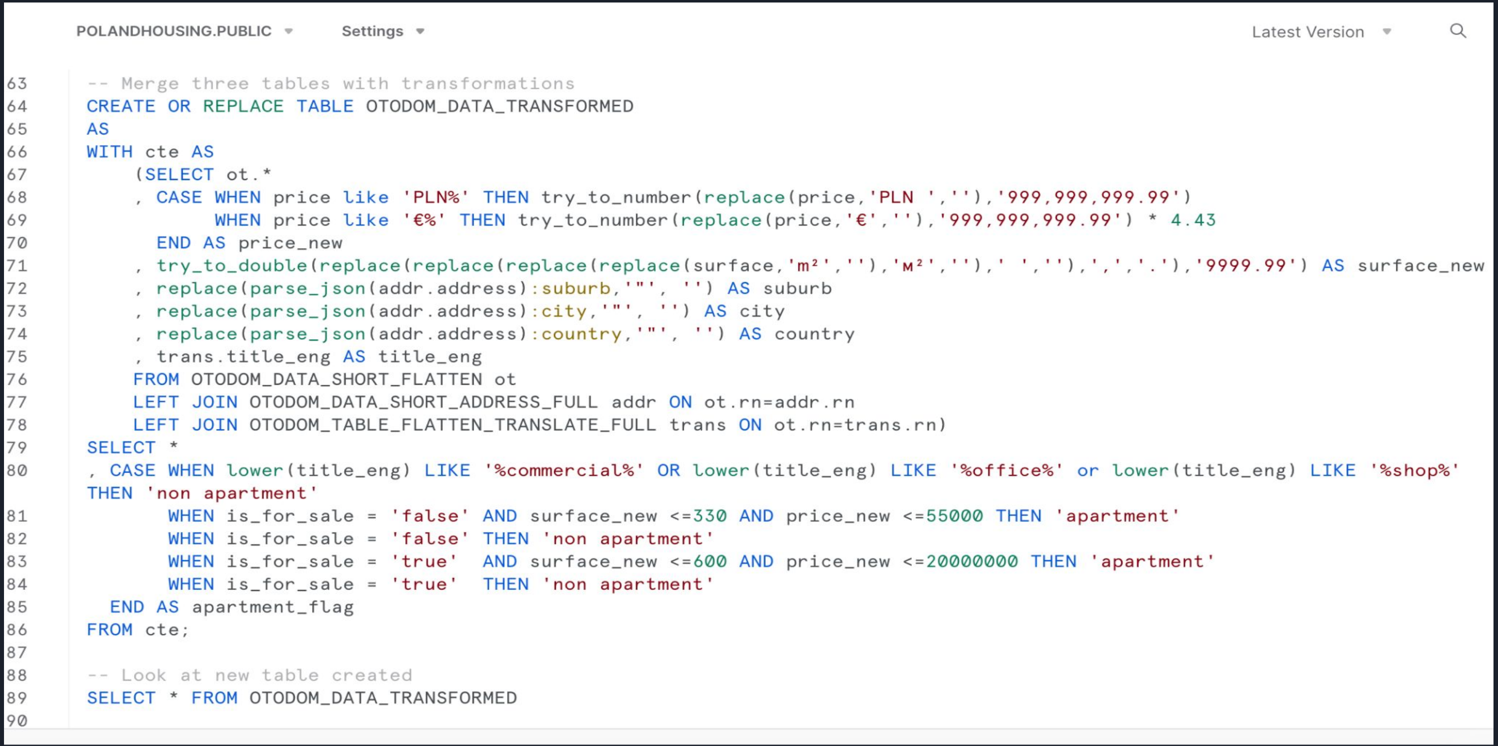


```python
                        database = 'POLANDHOUSING',
                        schema = 'public',
                        warehouse = 'POLAND_WH'))

with engine.connect() as conn:
    try:
        query = """ SELECT ID, SPREADSHEET_NAME FROM otodom_data_log """
        df = pd.read_sql(query,conn)
        df.columns = map(lambda x: str(x).upper(), df.columns)

        gc = gspread.service_account("/Users/Vlahonator/Downloads/service_account.json")
        loop_counter = 0

        for index, row in df.iterrows():
            loop_counter += 1
            locals()['sh'+str(loop_counter)] = gc.open(row['SPREADSHEET_NAME'])
            wks = locals()['sh'+str(loop_counter)].get_worksheet(0)
            df_out = get_as_dataframe(wks, usecols=[0,1,2], nrows=wks.row_count, header=None, skiprows=1, evaluate_formulas=True)
            print('Spreadsheet '+row['SPREADSHEET_NAME']+' loaded back to DataFrame!')

            df_out.columns = ['RN', 'TITLE', 'TITLE_ENG']
            df_out.to_sql('otodom_data_flatten_translate', con=engine, if_exists='append', index=False, chunksize=16000, method=pc

    except Exception as e:
        print('--- Error --- ',e)
    finally:
        conn.close()
engine.dispose()

print("--- %s seconds ---" % (time.time() - start_time))
```

Data Analysis

- Our analysis begins by joining three essential tables to create a comprehensive dataset. These joins are meticulously designed to ensure data integrity, utilizing the 'RN' field as the common linking element among the tables. This consolidated data view enables us to conduct robust, multi-dimensional analyses.
- The core of our analytical work lies in crafting precise SQL queries to address key questions about the Polish real estate market. These questions include analyzing rental and sale prices across major cities, identifying optimal locations for specific needs, and much more.
- We take our analysis one step further by focusing on the joined table. This refined table allows us to dig deeper into the dataset to extract additional insights, including the identification of luxurious neighborhoods, the most and least popular private ads in Warsaw, and various other market dynamics.

Snippet of SQL Join for Data Analysis



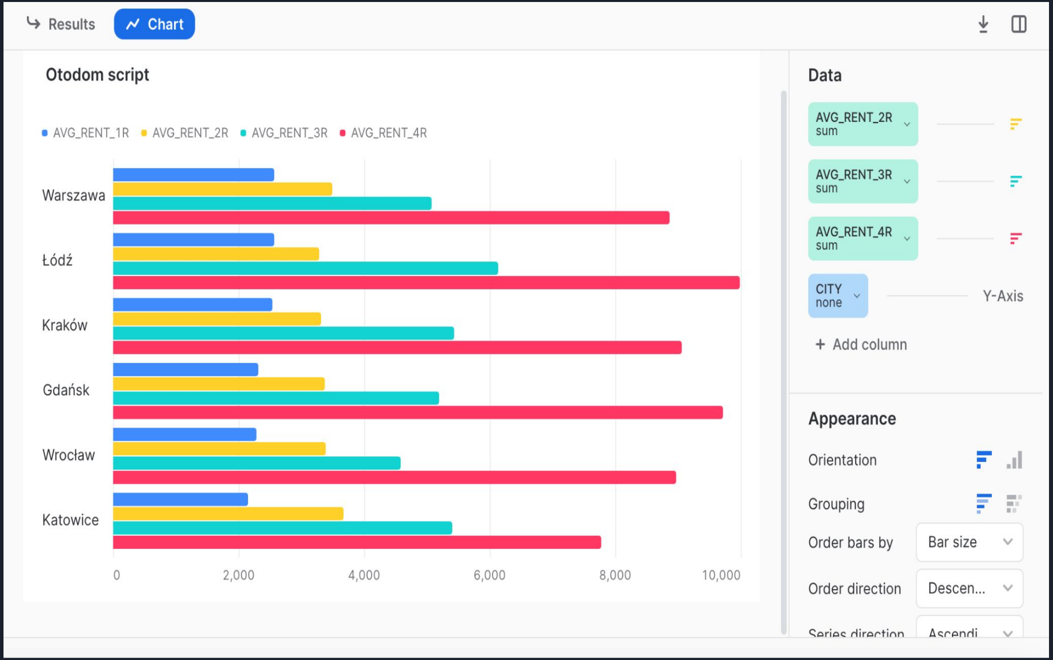
POLANDHOUSING.PUBLIC   Settings   Latest Version

```sql
-- Merge three tables with transformations
CREATE OR REPLACE TABLE OTODOM_DATA_TRANSFORMED
AS
WITH cte AS
    (SELECT ot.*
    , CASE WHEN price like 'PLN%' THEN try_to_number(replace(price,'PLN ',''),'999,999,999.99')
           WHEN price like '€%' THEN try_to_number(replace(price,'€',''),'999,999,999.99') * 4.43
      END AS price_new
    , try_to_double(replace(replace(replace(replace(surface,'m²',''),'м²',''),' ',''),',','.'),'9999.99') AS surface_new
    , replace(parse_json(addr.address):suburb,'"', '') AS suburb
    , replace(parse_json(addr.address):city,'"', '') AS city
    , replace(parse_json(addr.address):country,'"', '') AS country
    , trans.title_eng AS title_eng
    FROM OTODOM_DATA_SHORT_FLATTEN ot
    LEFT JOIN OTODOM_DATA_SHORT_ADDRESS_FULL addr ON ot.rn=addr.rn
    LEFT JOIN OTODOM_TABLE_FLATTEN_TRANSLATE_FULL trans ON ot.rn=trans.rn)
SELECT *
, CASE WHEN lower(title_eng) LIKE '%commercial%' OR lower(title_eng) LIKE '%office%' or lower(title_eng) LIKE '%shop%' THEN 'non apartment'
       WHEN is_for_sale = 'false' AND surface_new <=330 AND price_new <=55000 THEN 'apartment'
       WHEN is_for_sale = 'false' THEN 'non apartment'
       WHEN is_for_sale = 'true'  AND surface_new <=600 AND price_new <=20000000 THEN 'apartment'
       WHEN is_for_sale = 'true'  THEN 'non apartment'
  END AS apartment_flag
FROM cte;

-- Look at new table created
SELECT * FROM OTODOM_DATA_TRANSFORMED
```

Questions and SQL Queries

- The Otodom dataset provides us with a wealth of data about the real estate market in Poland. To transform this raw data into actionable insights, we've posed several key questions that we aim to answer. These questions are carefully chosen to cover a broad range of real estate concerns, from rental rates to buying opportunities.
- All codebase, including the scripts utilized for this analysis, will be made publicly available on my GitHub repository, https://github.com/EfthimiosVlahos/Otodom-Data-Analytics-Project, for further review and collaboration. The subsequent slides will feature a curated selection of questions, accompanied by their respective data visualizations to provide a comprehensive understanding of each query's answer.

Q1) What is the average rental price of 1 room, 2 room, 3 room and 4 room apartments in some of the major cities in Poland?



- Upon examination, we observe a linear relationship in the cost-to-size ratio across these categories. This suggests a consistent pricing strategy in the market, regardless of the apartment size. This consistency in the price-to-room ratio allows us to predict with a certain level of accuracy the potential pricing scale for apartments that fall outside the scope of this current analysis.

Q2) What size of an apartment can I expect with a monthly rent of 3000 to 4000 PLN in different major cities of Poland?

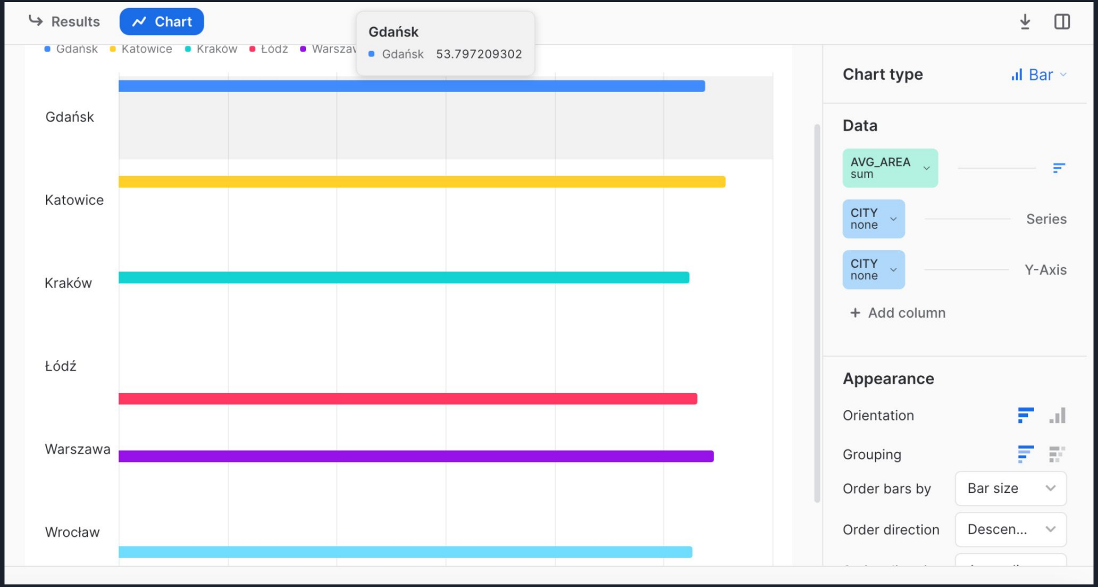


Chart type: Bar

Data

AVG_AREA sum

CITY none — Series

CITY none — Y-Axis

+ Add column

Appearance

Orientation

Grouping

Order bars by: Bar size

Order direction: Descen...

- As a key observation, it is notable that Warszawa (Warsaw) exhibits the highest average rental rates for comparable sizes, thereby serving as a valuable benchmark for evaluating property values in other cities.

Q2) Which are the top 3 most luxurious neighborhoods in Warsaw?

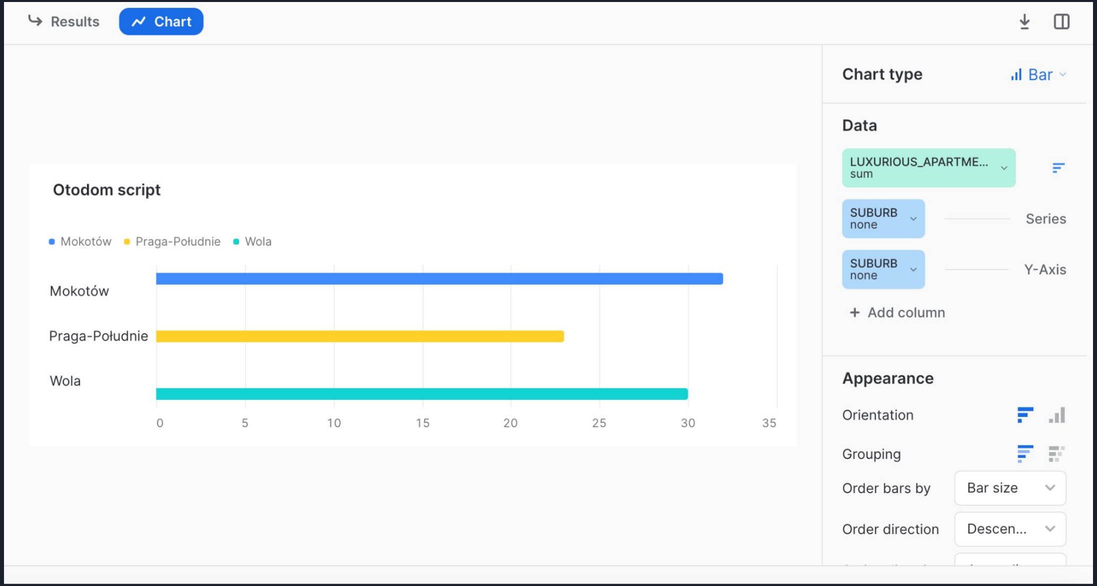


Chart type: Bar

Data: LUXURIOUS_APARTME... sum

SUBURB none — Series

SUBURB none — Y-Axis

Orientation

Grouping

Order bars by: Bar size

Order direction: Descen...

- Utilizing a combination of filtering, aggregation, and ranking mechanisms, the query precisely identifies the top three neighborhoods in Warsaw that are most saturated with luxury apartments.

Conclusion and Future Work

- **Robust Data Ingestion**: Utilized Bright Data for web scraping and efficiently exported raw data into Snowflake.
- **Advanced Transformation**: Leverage Python, Geocode, and Google APIs to transform and enrich the dataset.
- **Comprehensive Analysis**: Performed in-depth SQL queries on Snowflake to derive actionable insights on the Polish housing market.
- **Future Work**: There are two primary avenues for enhancement. First, I plan to augment our existing dataset by integrating additional data sources. This will not only deepen the analysis but also provide a more holistic view of the Polish housing market. Second, I aim to employ machine learning algorithms to forecast emerging trends, thereby offering predictive insights into market behavior.

THANK YOU!

Should you have any questions or require further clarification, please do not hesitate to reach out!
Email: vlahos89@gmal.com
LinkedIn: https://www.linkedin.com/in/efthimios-vlahos/
GitHub: https://github.com/EfthimiosVlahos