



# Relocation Navigator

Insightful Guide for Moving within the  
United States

# Agenda

- Introduction: What, Why and How of Project
- Problems we are trying to Solve
- System Architecture
- Data Modeling and Analysis
- Demo
- References

# Introduction

## What

- A Unified Dashboard for Comprehensive Relocation Insights.
- Detailed Information on Housing prices, Safety, and Demographics in U.S. Regions.
- Easy Comparison of Affordability and Living Conditions Across States and Cities.
- Help to get data driven analysis

## Why

- To Simplify Complex Relocation Decisions for Job Seekers, Students.
- To Empowers Users with Data on Safety and Cost of Living
- Help take informed decision

## How

- Utilized Attom API for Real estate data
- Used USAFacts data for Crime Data
- Built an Interactive Dashboard showing all the details



# Problems we are trying to Solve

- What is the current state of real estate market dynamics?
- How is Neighborhood Safety in particular loacation?
- Which one is better? Comparative Analysis for Better Decision Making
- Which is the optimal choice? Identifying Affordable Living Areas with some conditions
- Overall, Streamlining the Relocation Research Process

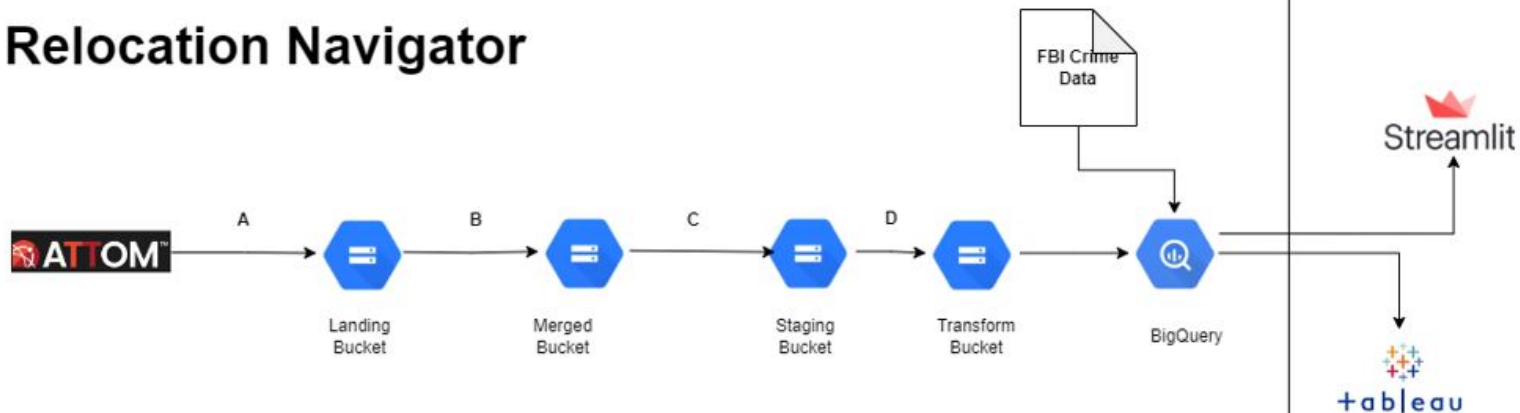




# Architecture



## Relocation Navigator



### List of Python Files

A: extract\_data.py

B: merge\_data.py

C: clean\_data.py

D: transform\_data.py



# DATA COLLECTION



Extracted from the Attom Data API, targeting real estate information across multiple ZIP codes."



Leverages Google Cloud Platform for secure data handling, using service account authentication to ensure data privacy and integrity.



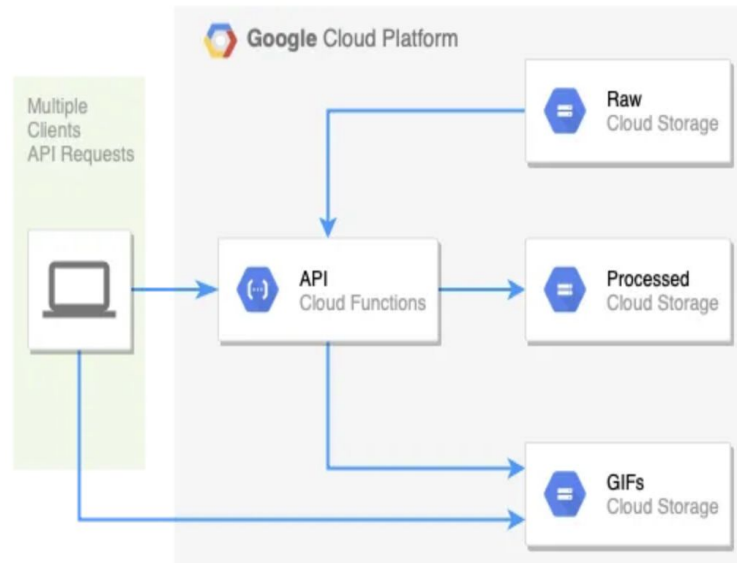
Incorporates advanced error handling to manage varying API response scenarios and adheres to API rate limits to maintain service stability.



Employs a systematic approach to clean and structure the raw data, making it ready for further analysis or integration into data warehouses.



Designed for scalability, the script can handle large data volumes and automatically saves the processed data to Google Cloud Storage



# Merging Data

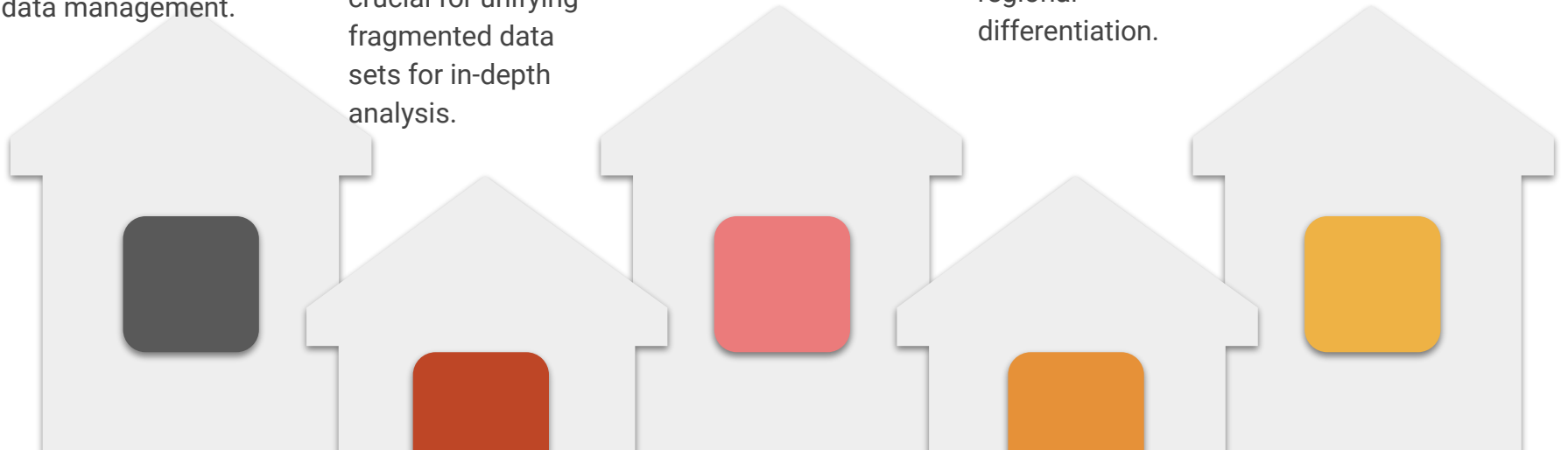
The MergeData class integrates with Google Cloud Storage, enabling efficient reading from and writing to cloud buckets, illustrating effective cloud-based data management.

Automates merging of data from multiple CSV files, representing API response pages, into a single DataFrame, crucial for unifying fragmented data sets for in-depth analysis.

Processes data dynamically for various ZIP codes, handling geographically diverse and voluminous datasets, essential for regional data analysis applications.

Enriches dataset by associating ZIP codes with corresponding states, adding valuable geographic context, vital for analyses needing regional differentiation.

Designed for scalability, efficiently processes and combines large data volumes, demonstrating Python and cloud computing's capabilities in managing big data challenges.

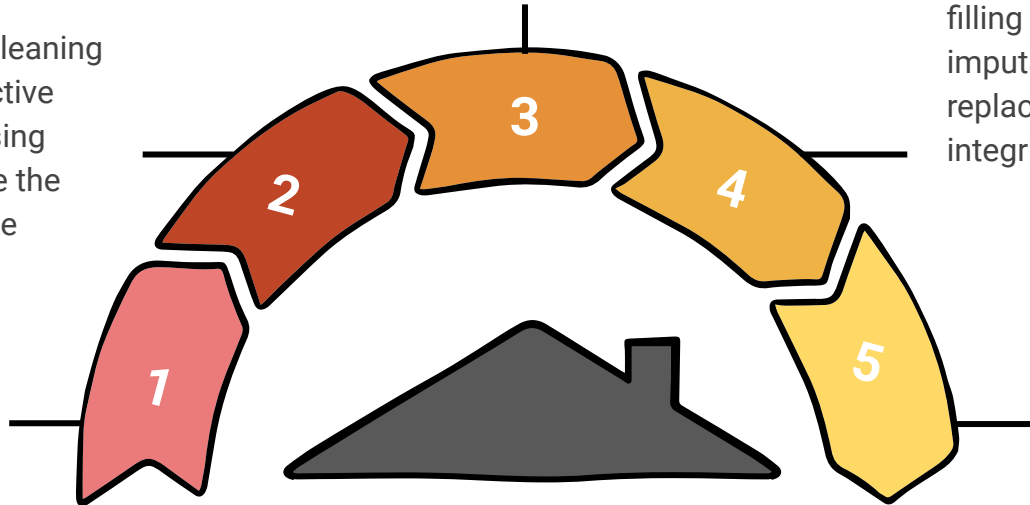


# Data Cleaning

Optimizes data structure through strategic Data Frame reordering and relevant column selection, enhancing data accessibility and analytical readiness.

Implements customized data filling techniques, such as mode imputation and specific replacements, to maintain dataset integrity and utility.

Employs advanced data cleaning strategies, including selective column removal and missing data imputation, to ensure the quality and reliability of the dataset.

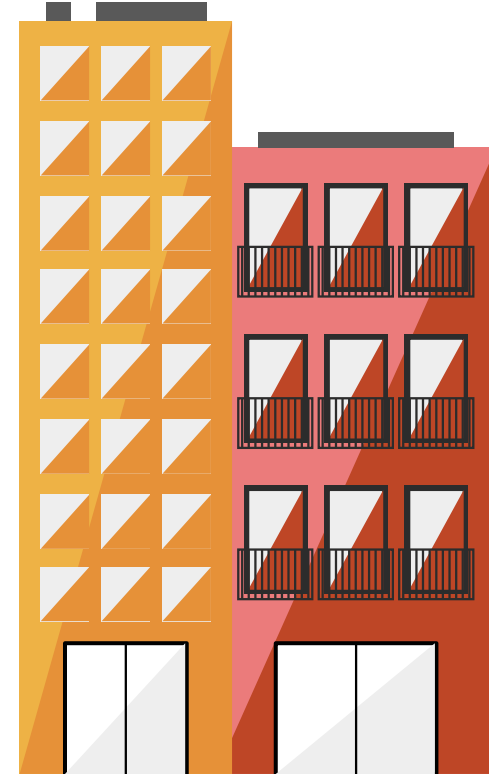
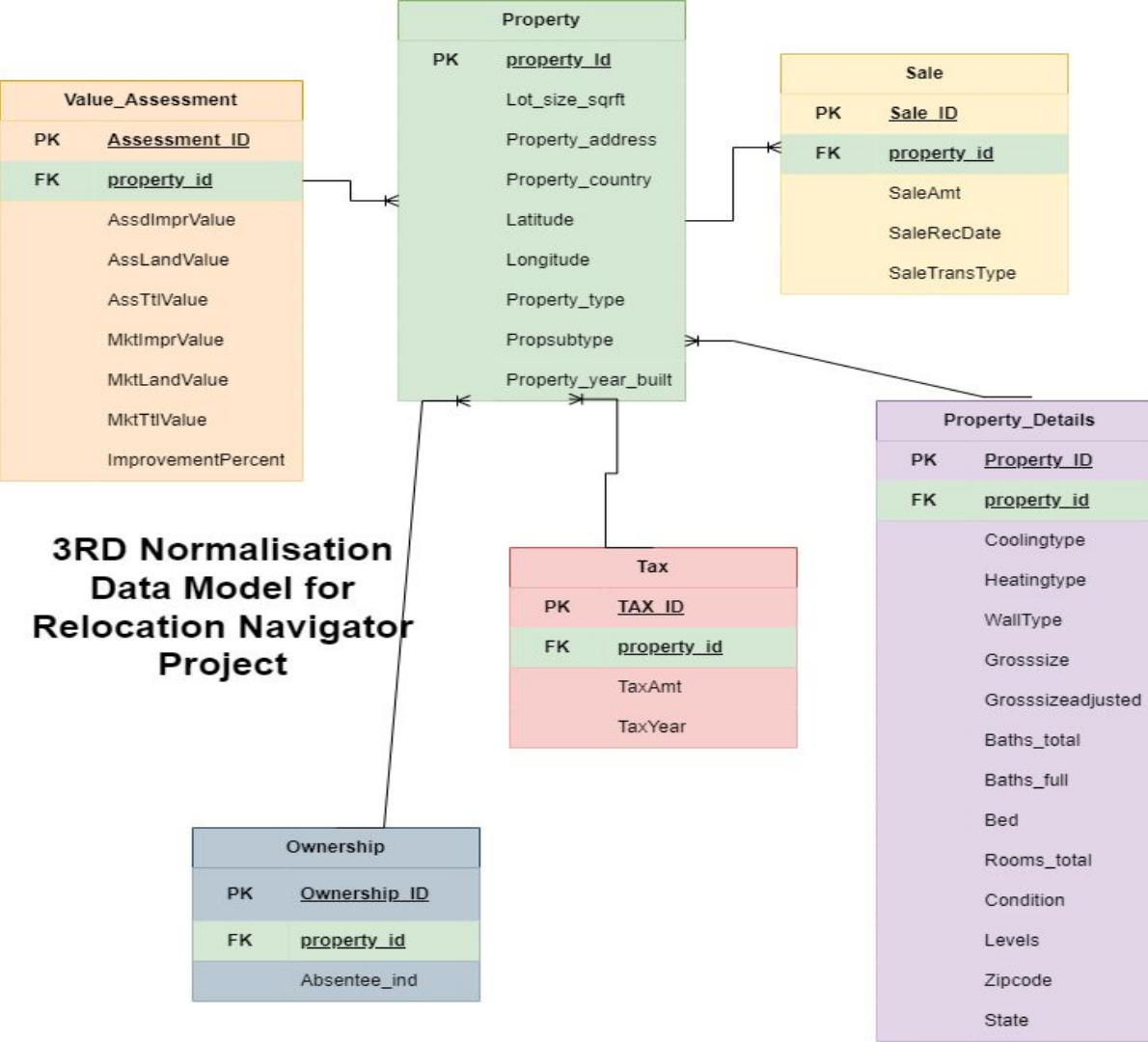


Utilizes Google Cloud Storage to seamlessly aggregate and process CSV files, demonstrating effective cloud-based data management.

Integrates into an automated data pipeline, showcasing the capability to efficiently manage the data workflow from extraction to cleaning and storage in a cloud environment.



# Data Modeling



# Data Transformation



1.

The data processing journey started by ingesting information from the merge bucket and shaping it into the Pandas DataFrame 'df' for efficient manipulation.



2.

To ensure data integrity, the Property DataFrame underwent meticulous deduplication, serving as both a primary and foreign key for related datasets.



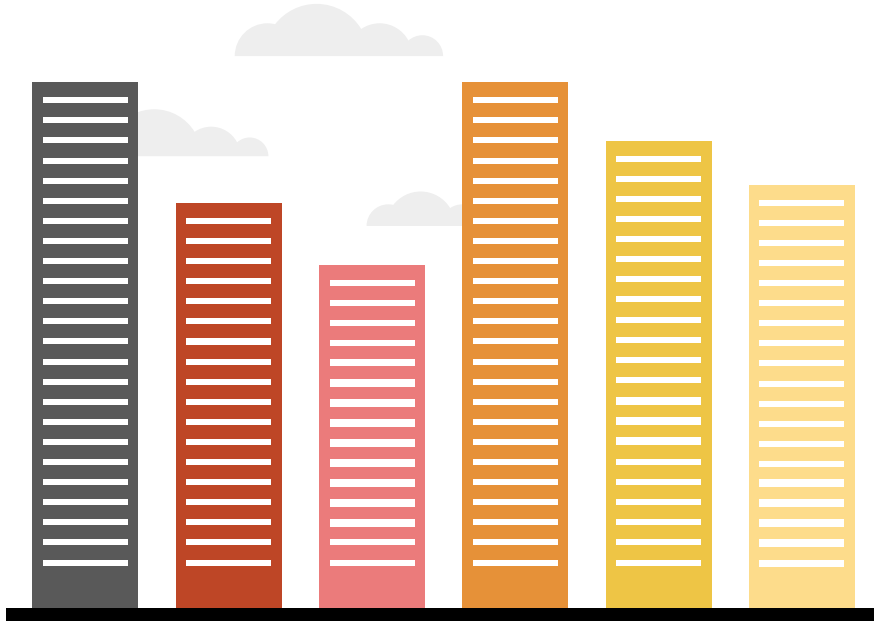
3.

In preparation for future integration with BigQuery, specific adjustments to data types, especially for date columns, were implemented to ensure seamless compatibility.

# Demo



# Future Work



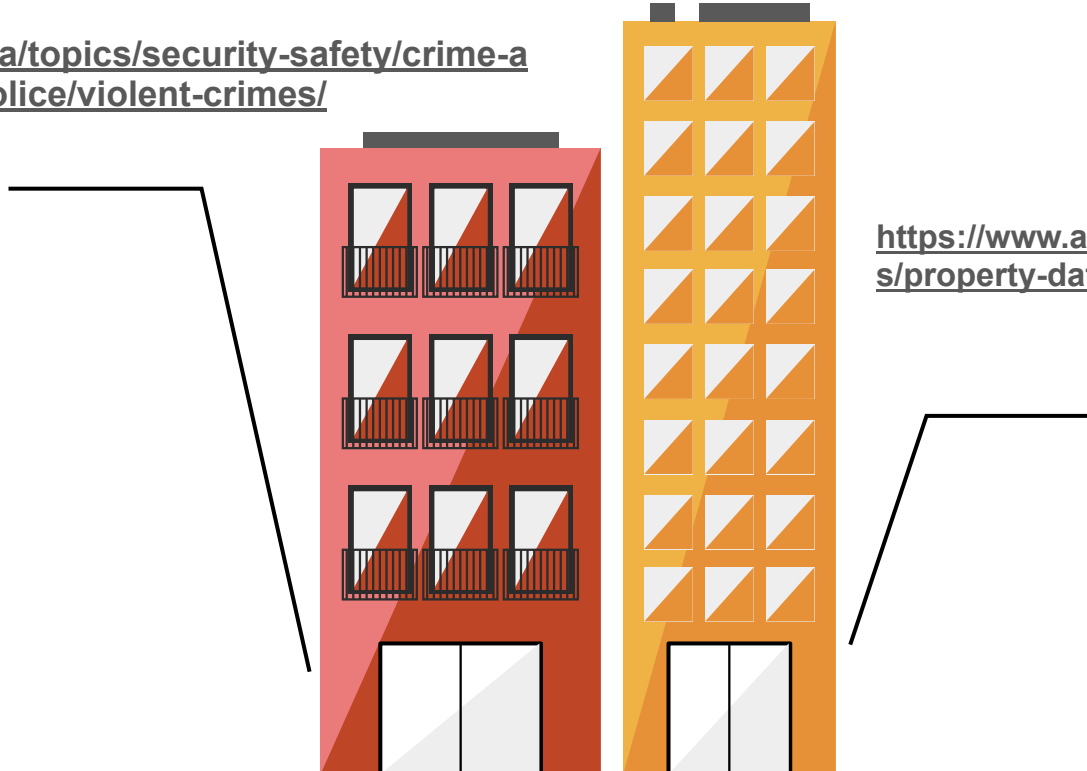
Developed our streamlit based web app further

We can try to use streaming services to fetch real time data

Implement recommendation system using deep learning and hence leverage MLOps.

# References

<https://usafacts.org/data/topics/security-safety/crime-and-justice/crime-and-police/violent-crimes/>



<https://www.attomdata.com/solutions/property-data-api/>

# Thank You!!!

