Crime analysis and Airbnb’s price prediction in the city of Austin

Prepared by:- Binaya Khadka

1. Introduction

This project was focused on investigating whether crime rates have a significant impact on the listing prices of Airbnb properties in the city of Austin, Texas. As Airbnb listings continue to grow in number, it becomes crucial to understand the factors that influence their pricing dynamics. By examining the relationship between crime rates and listing prices in Austin, we can gain insights into how safety concerns affect the rental market and potentially inform both hosts and travelers.

To accomplish this, we will analyze crime data from Austin and combine it with Airbnb listing information, including property attributes, location, and pricing details. By leveraging data analysis techniques and statistical modeling, we aim to uncover any associations between crime rates and Airbnb listing prices. Such insights can help hosts optimize their pricing strategies, provide valuable information to prospective guests, and contribute to the broader understanding of the interplay between crime and the sharing economy.

2. Data Acquisition and Cleaning

Data were acquired from two sources: the Austin open data portal and Airbnb's data archive. The data acquired from Airbnb's archive was nearly clean, with the exception of a few missing values. On the other hand, the crime data contained more columns than we needed, but it was relatively clean.

2.1 Data Wrangling: crime data

This dataset was huge, with nearly 2.4 million rows of data. Out of the 33 columns available, the following columns were selected for analysis:

- Incident Number
- Highest Offense Code
- Highest Offense Description
- Occurred Date Time
- Occurred Date
- Address
- Zip Code
- Location Type
- X-coordinate
- Y-coordinate
- Longitude
- Latitude
- Council District

The following steps performed at this stage:

1. All rows with missing values were dropped.
2. Duplicates were dropped based on ***Incident Number, X-coordinate,*** and ***Y-coordinate.***
3. All values in the ***Highest Offense Description*** column were converted to camel case.
4. Additional columns, including ***month, year, week,*** and ***day of the month***, were created from the ***Occurred Date Time*** column.

5. All listed columns were renamed with underscores (_) replacing white spaces.
6. Similar crime types with different spellings were renamed using a common name.
7. Finally, the cleaned data was exported to a new CSV file for later use.

2.2 Data Wrangling: Airbnb

There were originally 75 columns in the dataset, and out of those, the following 16 columns were used for analysis:

- id
- host_id
- neighbourhood_cleansed
- host_neighbourhood
- property_type
- room_type
- accommodates
- bedrooms
- beds
- minimum_nights
- review_scores_rating
- price
- longitude
- latitude
- number_of_reviews
- availability_365

The following steps performed at this stage:

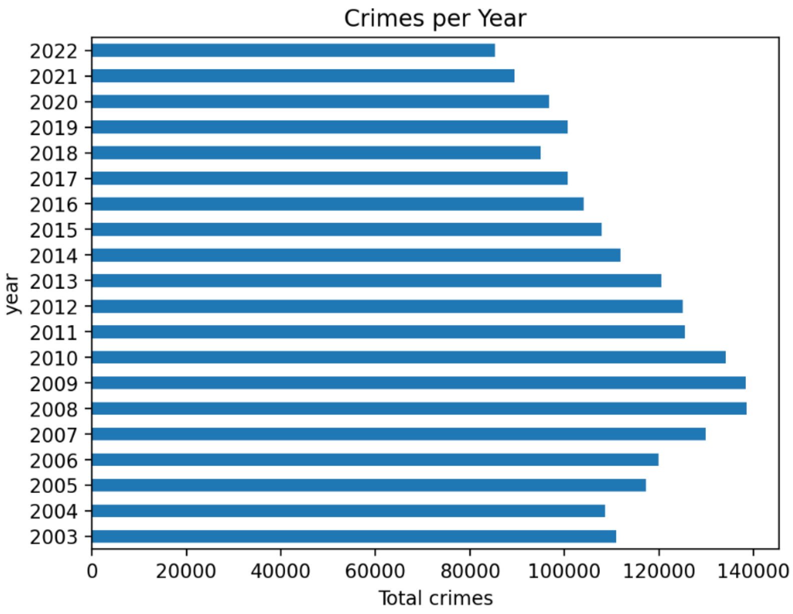
1. The listed columns were renamed to replace white spaces with underscores (_).
2. The duplicated entries were dropped based on ***host_id, longitude,*** and ***latitude***.
3. In the "***price***" column, the dollar ($) sign and comma (,) were removed, and the data type was converted to an integer.
4. Missing values in the "***review_scores_rating***" column were imputed with the median, while missing values in the "***bedrooms***" and "***beds***" columns were imputed with a value of 1, assuming that there would be at least 1 bed or bedroom.
5. The "***host_neighbourhood***" column had missing data, so the missing values were imputed based on the corresponding zip code.
6. Identical names that were spelled differently were renamed using a common name.
7. Finally, the cleaned data was exported to a new CSV file for later use.

3. Exploratory Data Analysis (EDA)

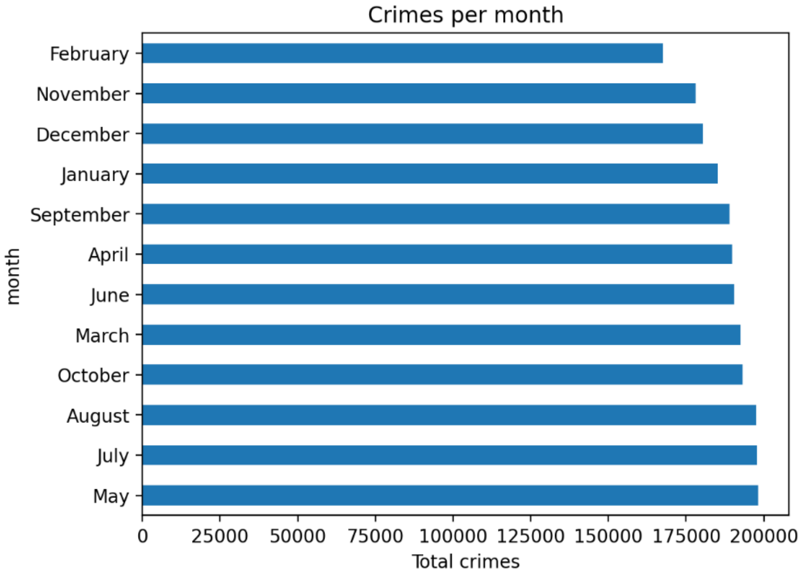
Performing EDA on by merging crime dataset and Airbnb's dataset would be the ideal choice but large computation power would require to merge them so EDA was performed on separate dataset.

3.1 EDA: crime data

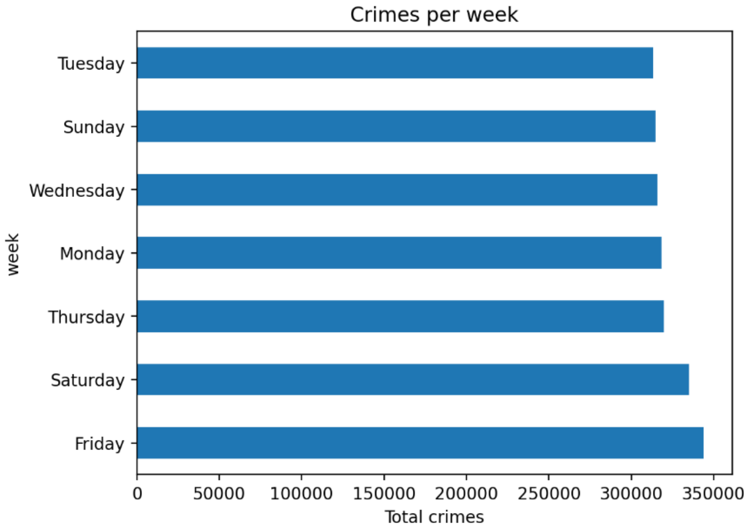
Data was grouped by ***year*** so the graph depicts crimes per each year.



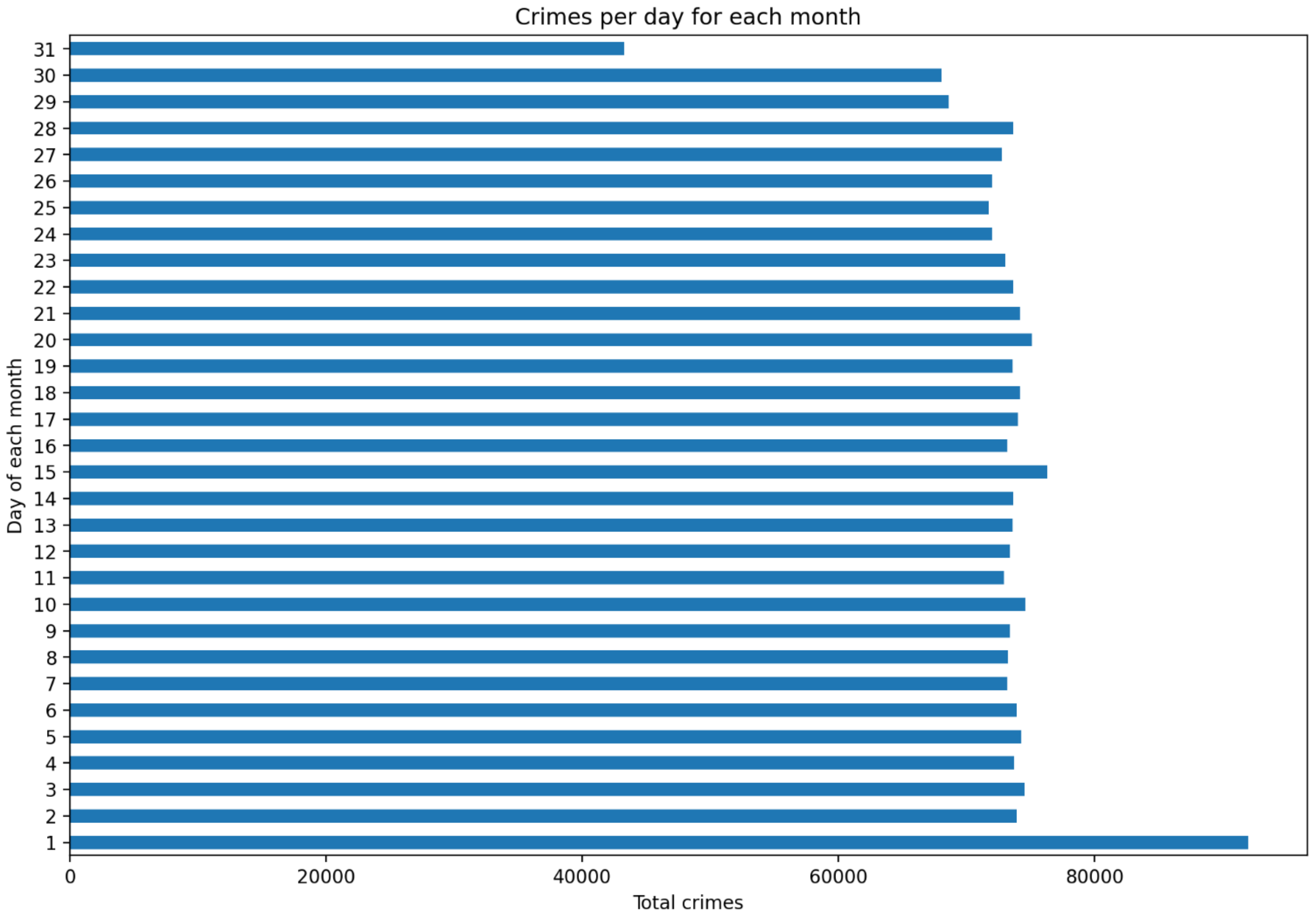
Next, the data was grouped by ***month.***



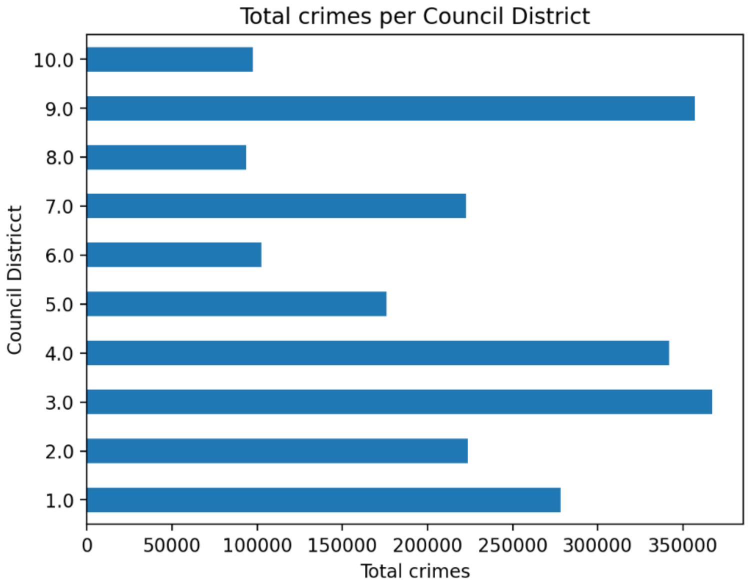
Data grouped by ***week.***



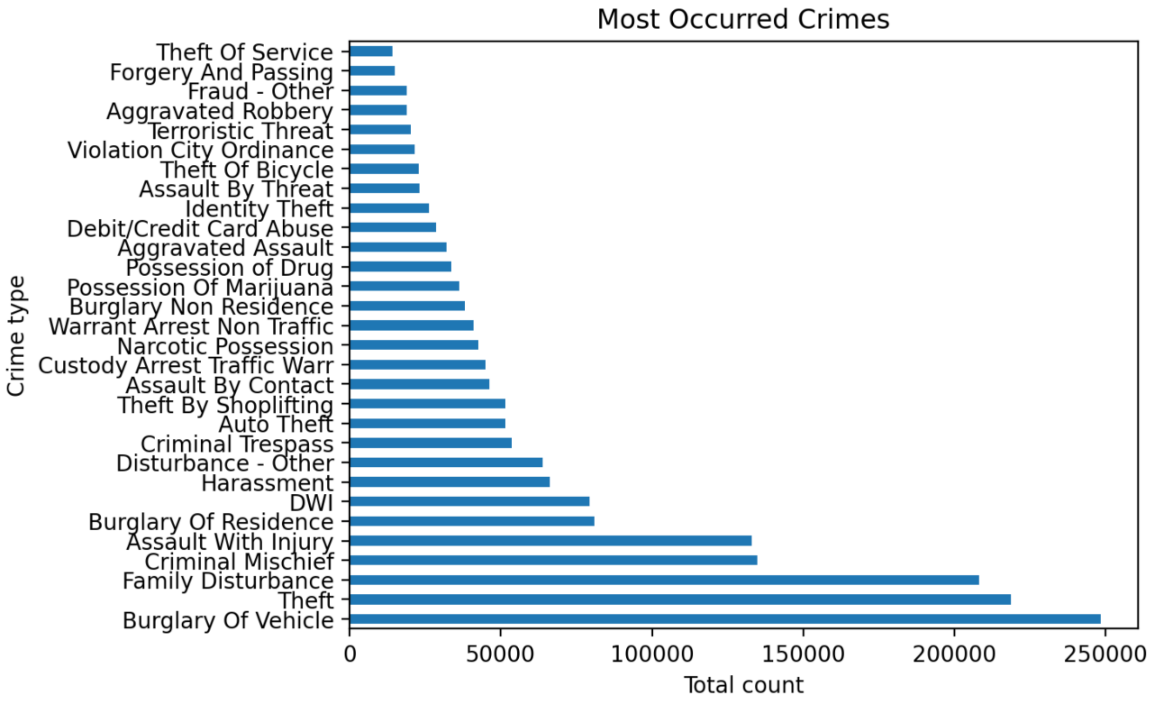
The graph shows how many crimes happened on daily basis from 2003 to 2022.



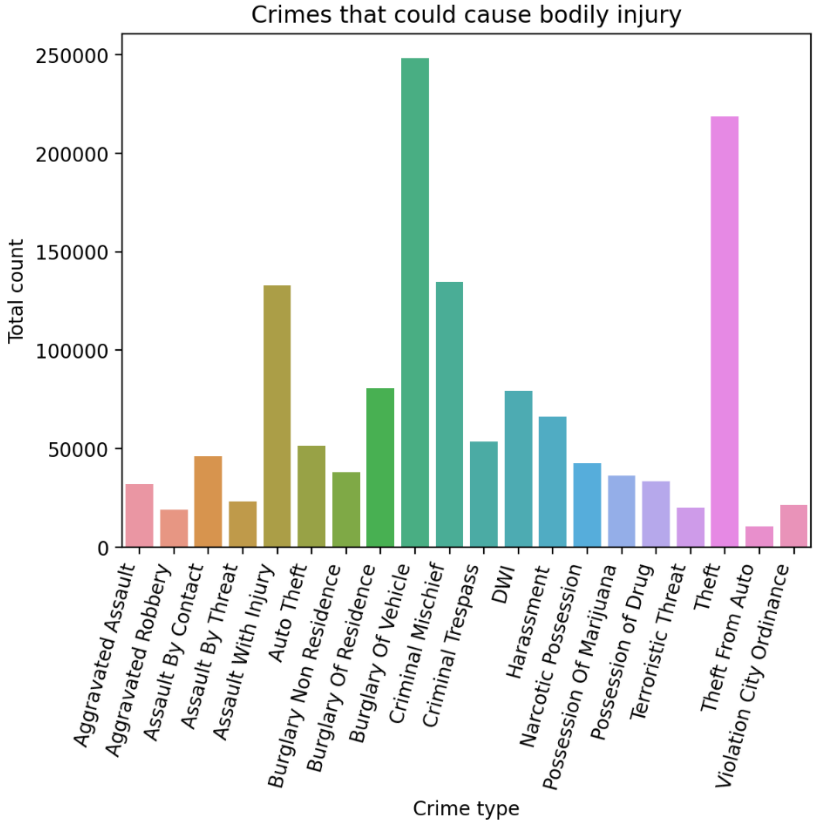
Data grouped by ***council district.***



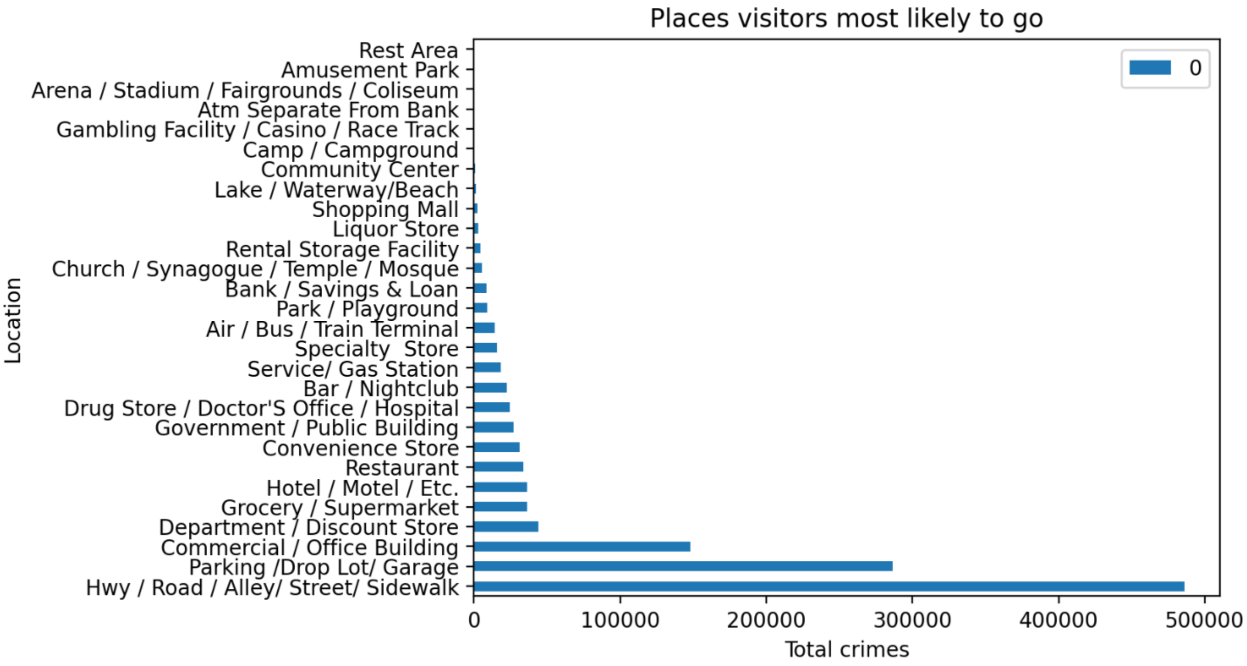
Most occurred crimes



Crimes that could cause bodily injury or death grouped by ***crime_type.***

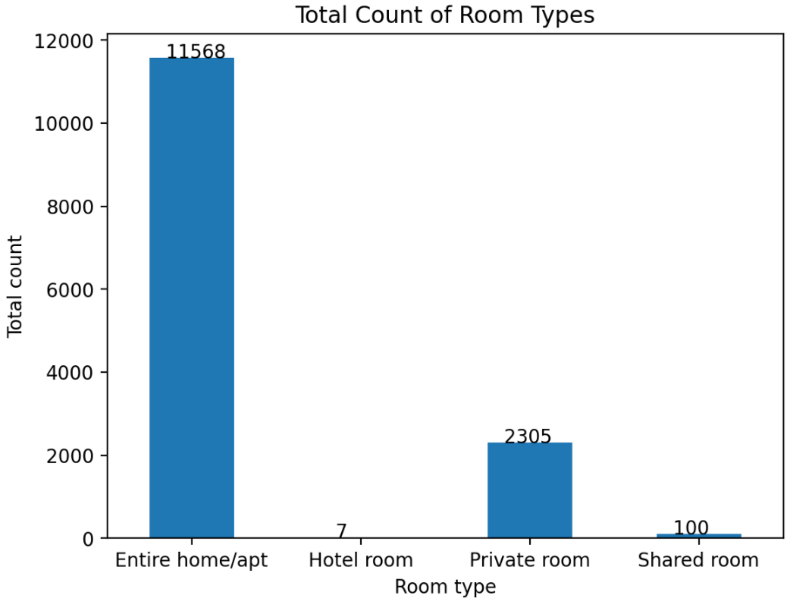


Places visitors were likely to go most.

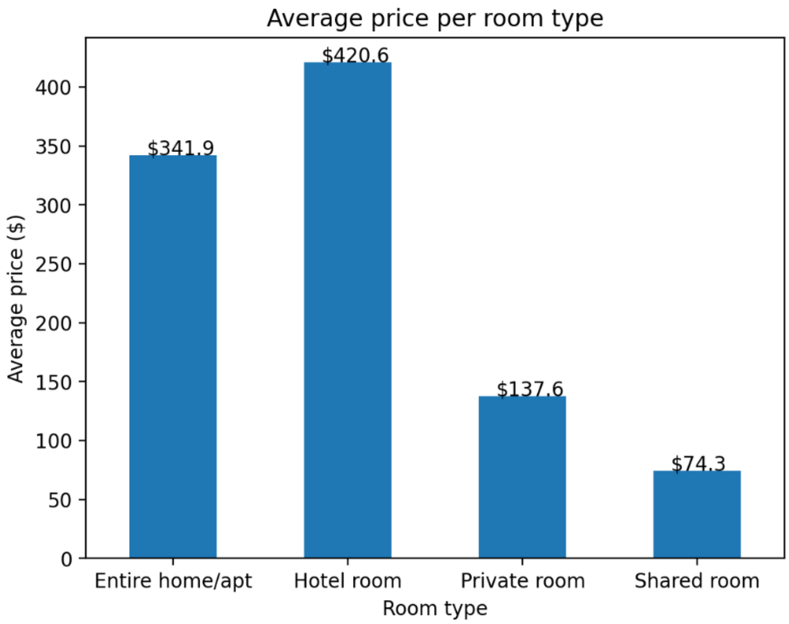


3.2 EDA: Airbnb

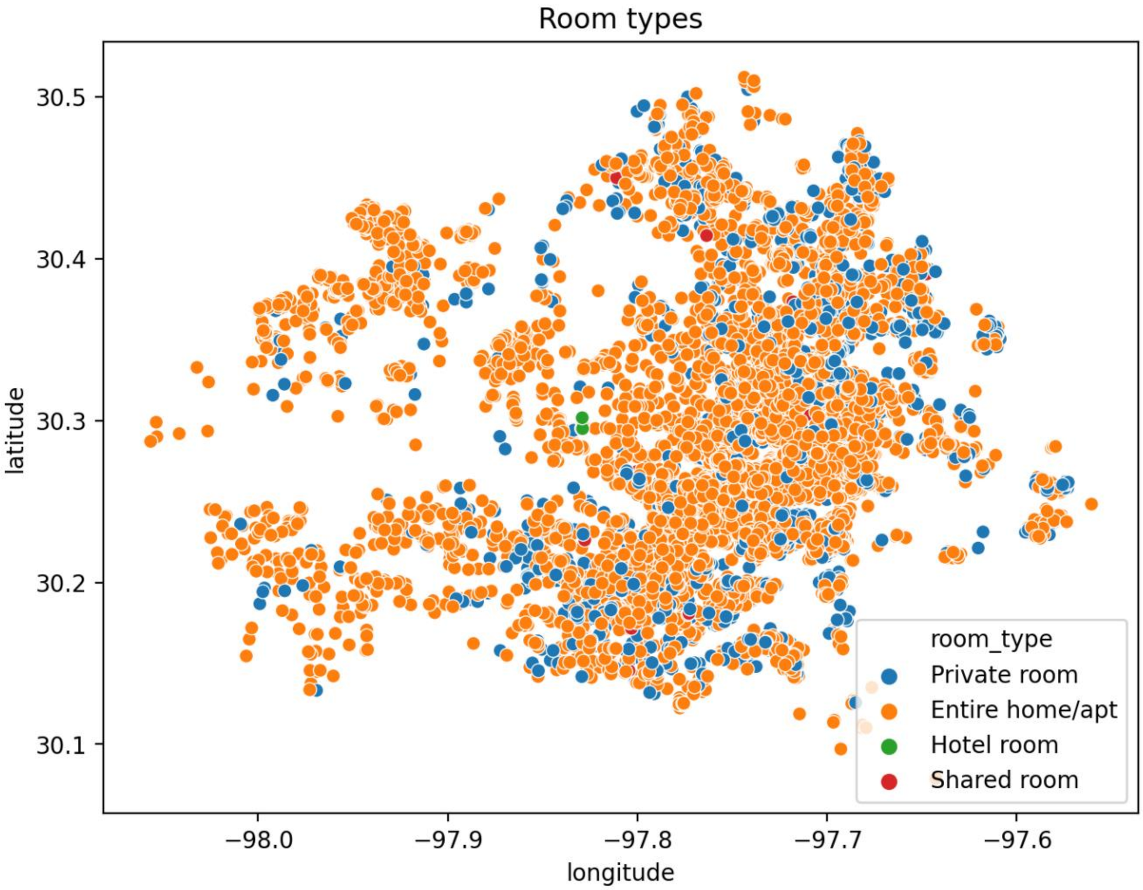
Total room count



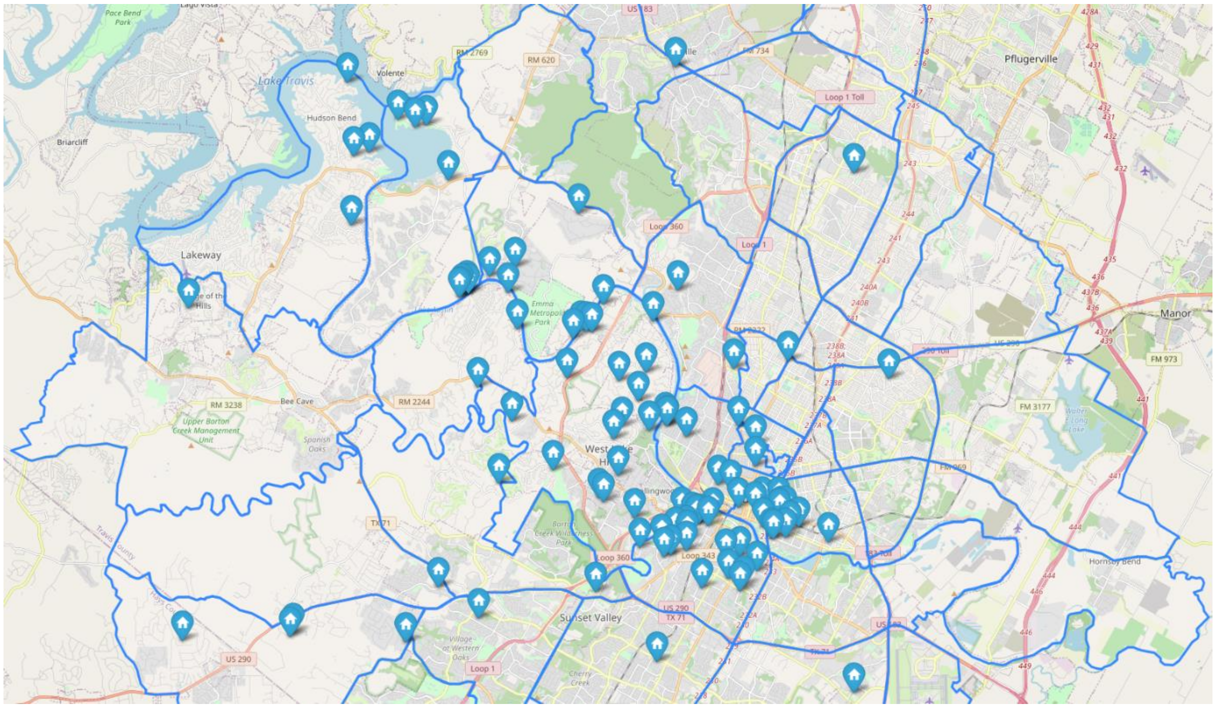
Average price



Room types scattered along the city.

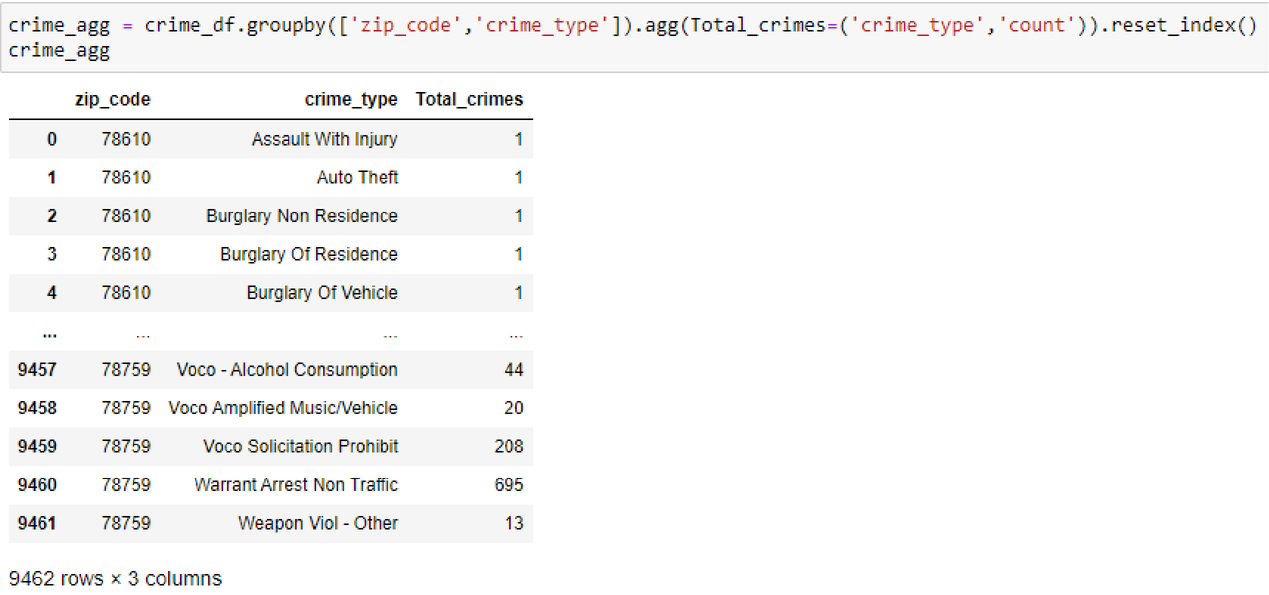


Top 100 expensive places



4. Training and Modeling

Merging the crime data with Airbnb's data would result in an extremely large dataset, estimated to be around 3 billion rows, which exceeds the computer's handling capacity. As a result, the crime types were grouped together based on the zip code, resulting in the formation of a new dataframe as shown below.



```
crime_agg = crime_df.groupby(['zip_code','crime_type']).agg(Total_crimes=('crime_type','count')).reset_index()
crime_agg
```

	zip_code	crime_type	Total_crimes
0	78610	Assault With Injury	1
1	78610	Auto Theft	1
2	78610	Burglary Non Residence	1
3	78610	Burglary Of Residence	1
4	78610	Burglary Of Vehicle	1
...	...	...	...
9457	78759	Voco - Alcohol Consumption	44
9458	78759	Voco Amplified Music/Vehicle	20
9459	78759	Voco Solicitation Prohibit	208
9460	78759	Warrant Arrest Non Traffic	695
9461	78759	Weapon Viol - Other	13

9462 rows × 3 columns

This newly created dataframe was then merged with Airbnb's data to create a new dataset that was utilized for testing and training purposes.

Three regression techniques were used for training and testing the dataset.

1. CatBoost
2. XGBoost
3. LightGBM

To find the best parameters for the models, Tree-Structured Parzen Estimator (TPE) was used for the hyperparamter tuning.

I adopted RMSE (Root Mean Squared Error) as the accuracy metric instead of MAE (Mean Absolute Error) because RMSE gives higher weight to large errors due to the squaring of errors before averaging. This makes RMSE particularly useful when large errors are considered more undesirable. The RMSE is calculated by taking the square root of the average of the squared residual errors from the line of best fit.

A smaller RMSE value indicates a more accurate prediction since it measures the overall magnitude of the errors. By taking the square root, the RMSE provides a measure of the typical difference between the predicted values and the actual values.

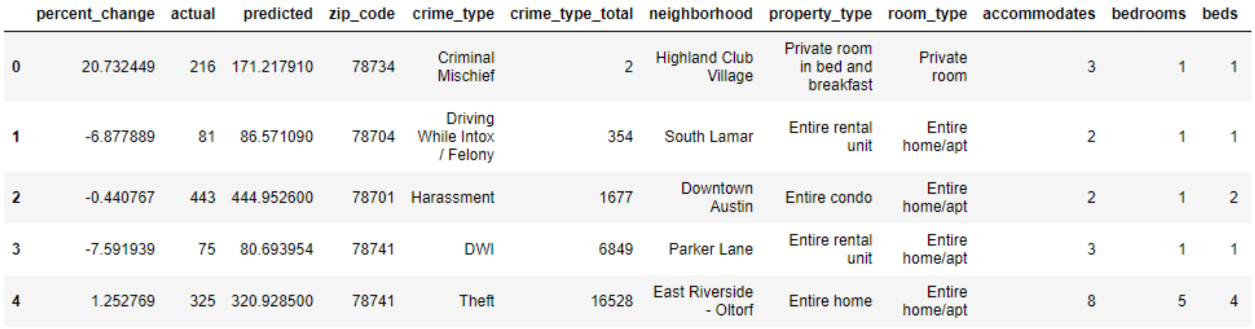
Here are the final RMSE scores after the training and testing were completed.

CatBoost:
Data type: categorical, Training score: 16.66 Testing score: 14.48

XGBoost:
Data type: Label encoded, Training score: 9.31, Testing score: 8.99
Data type: One hot encoded, Training score: 15.01, Testing score: 15.18

LightGBM:
Data type: categorical, Training score: 213.59, Testing score: 220.07
Data type: one hot encoded, Training score: 80.29, Testing score: 80.95

Predictions:



	percent_change	actual	predicted	zip_code	crime_type	crime_type_total	neighborhood	property_type	room_type	accommodates	bedrooms	beds
0	20.732449	216	171.217910	78734	Criminal Mischief	2	Highland Club Village	Private room in bed and breakfast	Private room	3	1	1
1	-6.877889	81	86.571090	78704	Driving While Intox / Felony	354	South Lamar	Entire rental unit	Entire home/apt	2	1	1
2	-0.440767	443	444.952600	78701	Harassment	1677	Downtown Austin	Entire condo	Entire home/apt	2	1	2
3	-7.591939	75	80.693954	78741	DWI	6849	Parker Lane	Entire rental unit	Entire home/apt	3	1	1
4	1.252769	325	320.928500	78741	Theft	16528	East Riverside - Oltorf	Entire home	Entire home/apt	8	5	4

The model predicted that the price could be dropped by 4%.
The scores would improve by a good margin if the outliers were removed. Some of the listings in Airbnb had price tag of over $1,000.

5. Future improvement

The main limitation that prevented the model from utilizing the entire dataset was the RAM (Random Access Memory) constraint. Due to the limited amount of available RAM, the model had to work with a subset of the data. As a result, the model's performance and score could potentially be improved if the entire dataset could be utilized.

Additionally, with higher computational power, better hyperparameter tuning would have been possible. Hyperparameters are parameters that are not learned by the model itself but are set prior to the training process. These hyperparameters can significantly impact the model's performance. With more computational power, it would have been feasible to explore a wider range of hyperparameter values and conduct more extensive hyperparameter tuning, potentially leading to improved model performance and higher scores.