Analyzing Airbnb Data with Machine Learning and Geospatial Modeling

Charlie Zien and Raymond Atta-Fynn

NYC Data Science Academy: Capstone Presentation

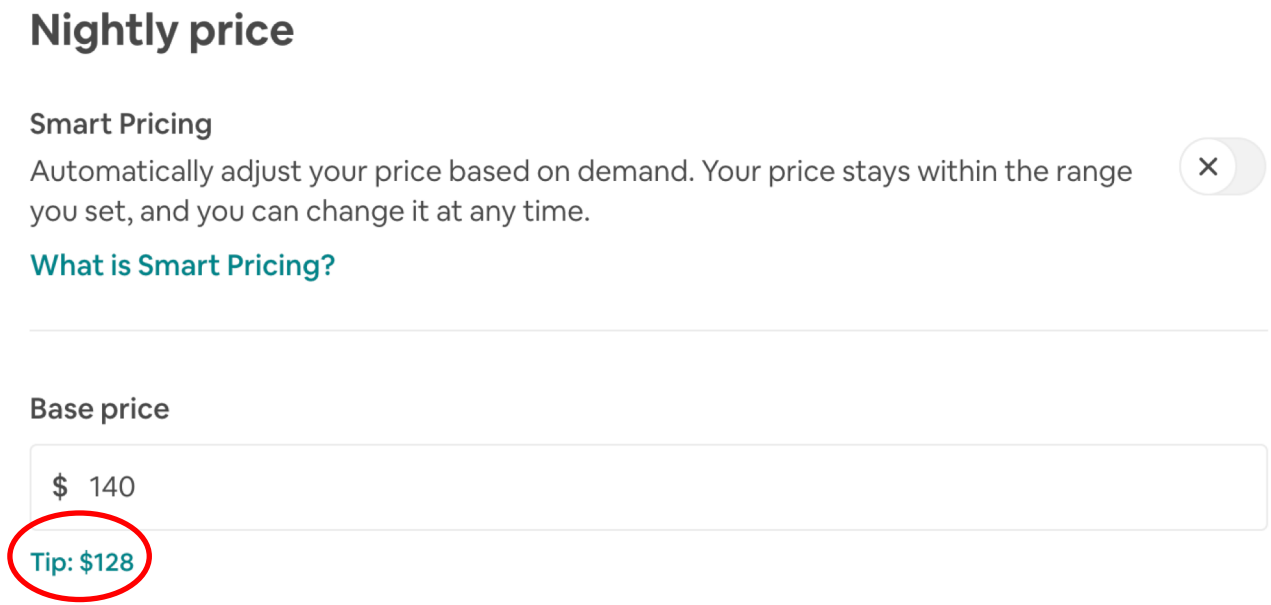
September 19, 2019

Outline

- Motivation and background information
- The data set
- Data Analysis via visualization and geospatial modeling
- Model training and prediction via machine learning
- Concluding remarks

Introduction

- Airbnb generates revenue by charging its guests and host fees for arranging stays (hosts are charged 3% of the value of the booking; guests are charged 6-12% depending on the nature of the booking)
- As an ecosystem, Airbnb transactions generate a lot of data: density of rentals across cities and neighborhoods; price variations across rentals and cities; host-guest interactions in the form of reviews (mainly guest experiences).
- A "smart pricing" algorithm recommends hosts a price to set when posting



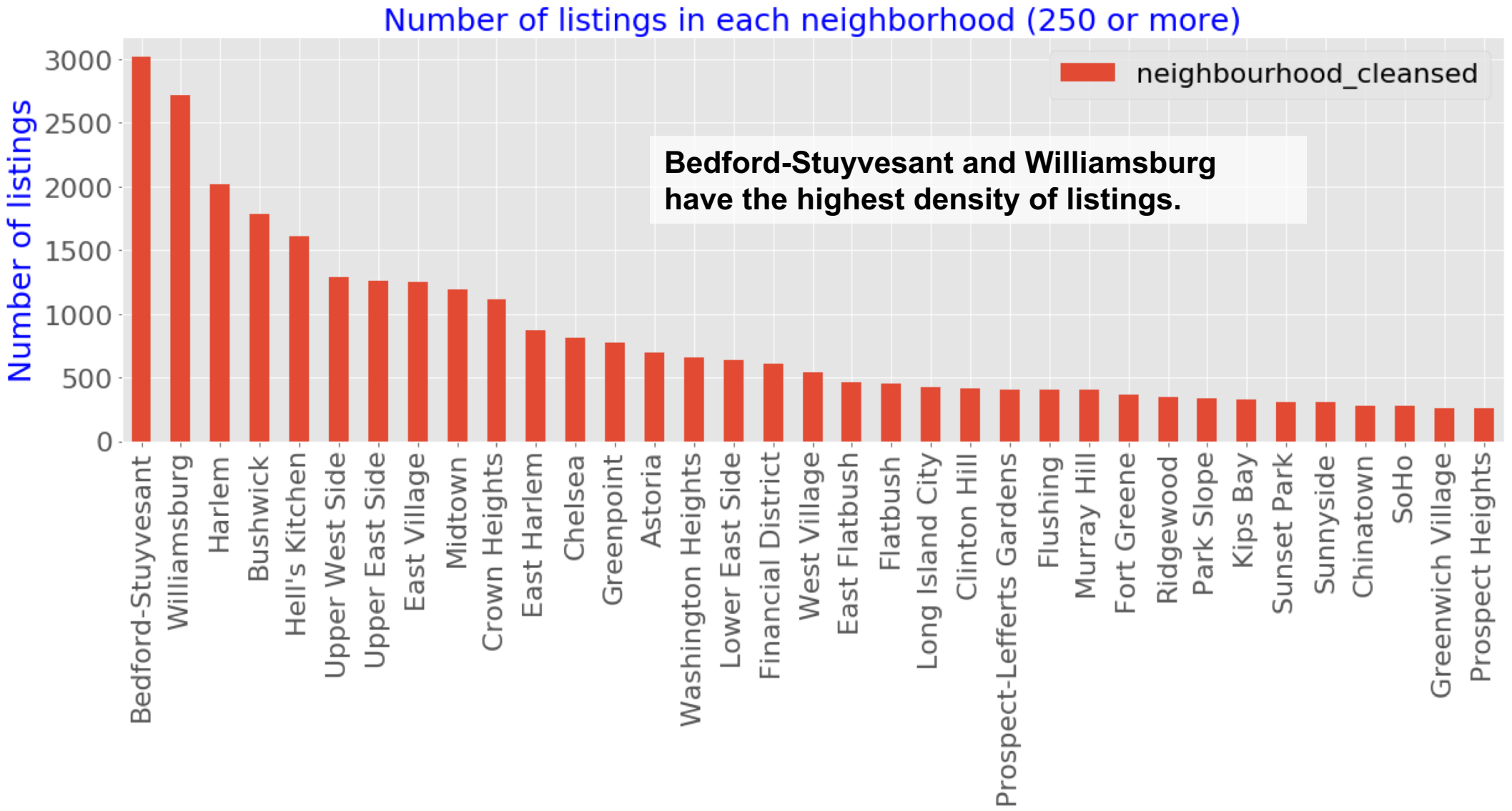
Motivation

- Airbnb's goal is to book as many properties as possible
- Question: Is the algorithm a smart move for hosts, or is it only "smart" with regard to Airbnb's objectives?
- What further information can be gleaned from Airbnb transactions in New York city?
  - How are the rental properties geographically distributed across the city?
  - How do prices vary across properties, neighborhoods and rental amenities?
  - How well can the rental prices be predicted using machine learning methods?

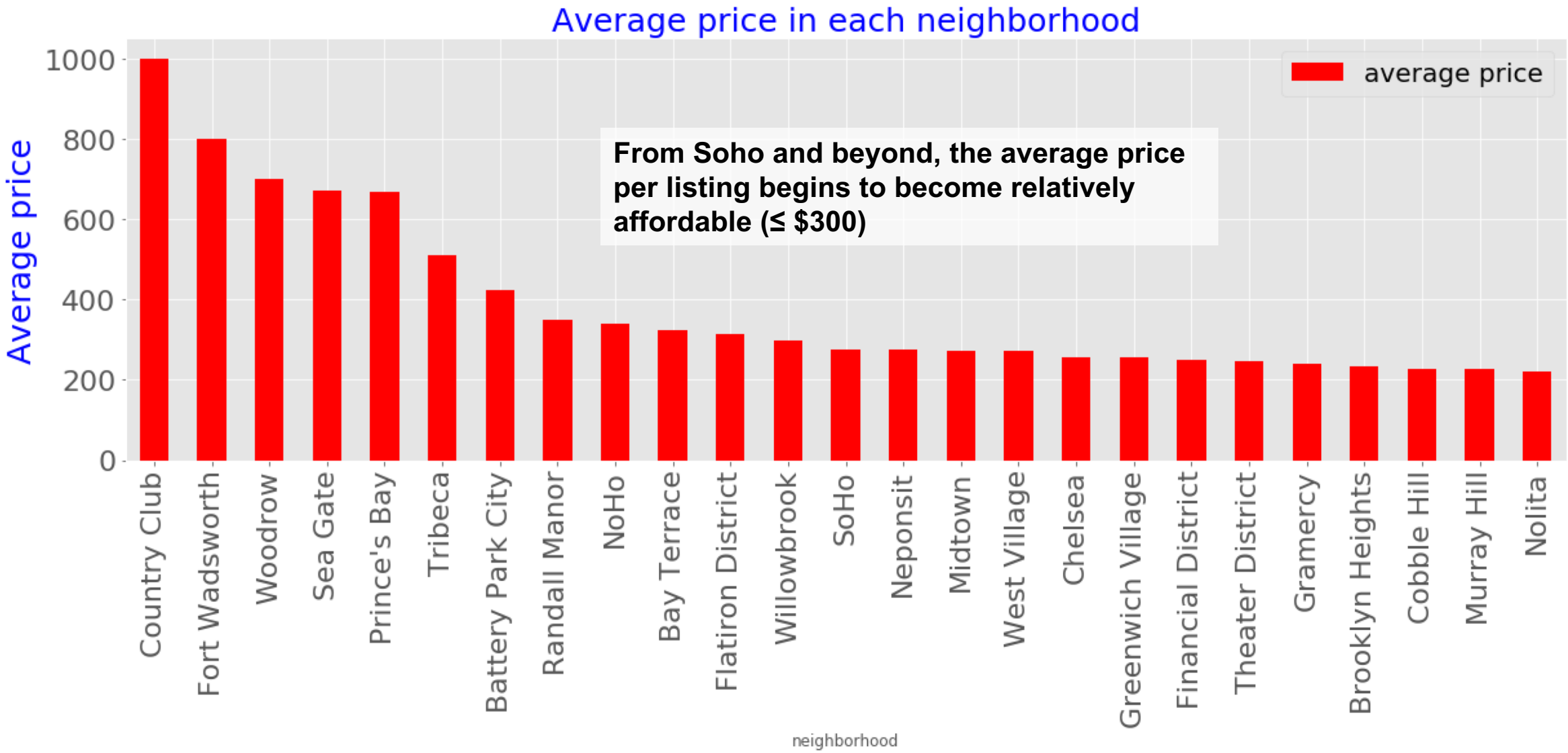
The data

- The data set is broken into three key groups:
  - Listings: a detailed list of rental listings with 106 attributes from Inside Airbnb (independent of Airbnb). A few of the attributes are price, number of beds, property type, neighborhood, cleaning fee, etc.
  - Mapping shapefiles: Inside Airbnb's map of neighborhood boundaries for mapping; US Census shapefiles of census tracts and blocks
  - Property taxes: Primary Land Use Tax Lot Output (PLUTO) – Public register of NYC property values and taxes assessed. Values are based on estimates for 1 year worth of rental income

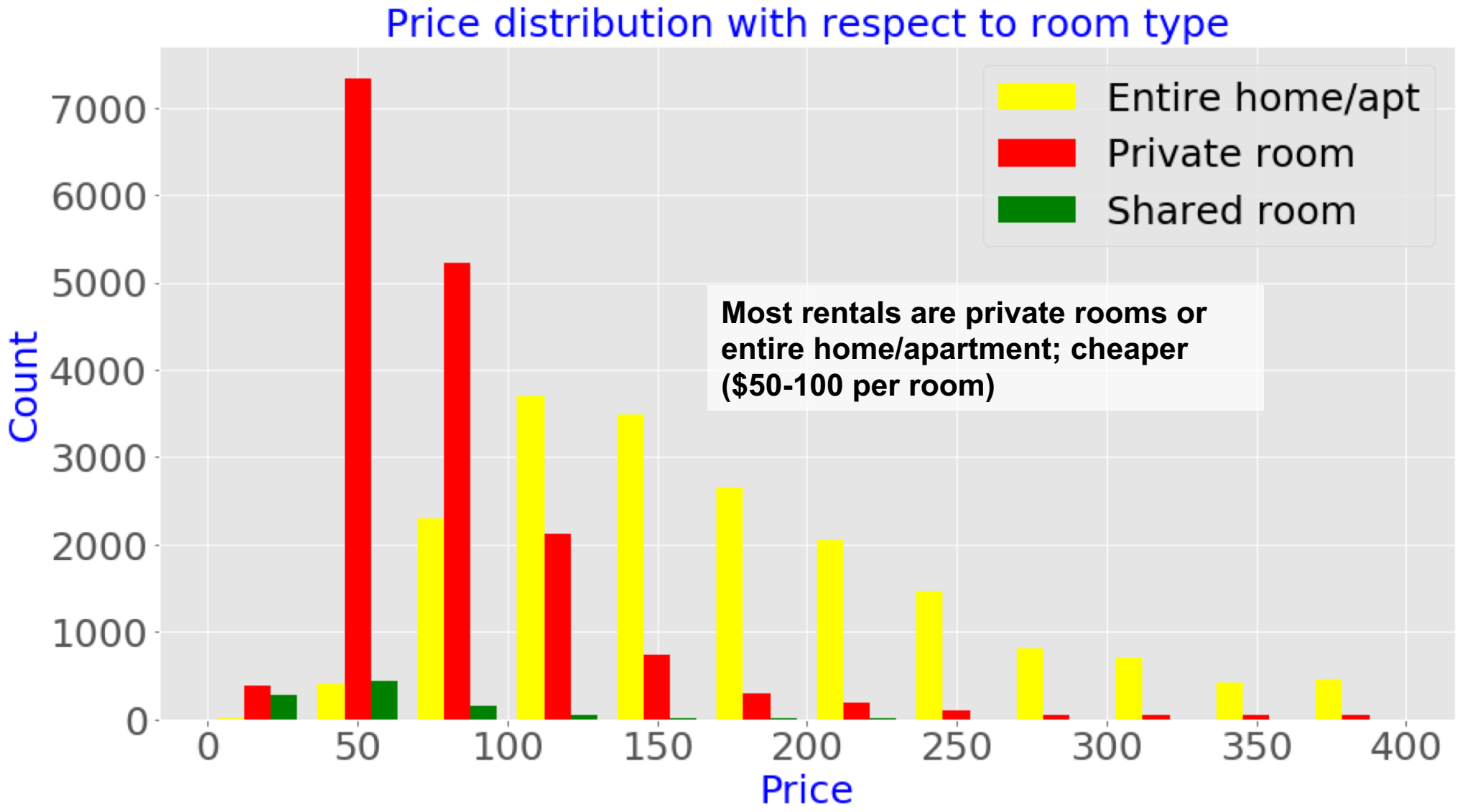
Exploring the data: listings per neighborhood



Exploring the data: price per neighborhood

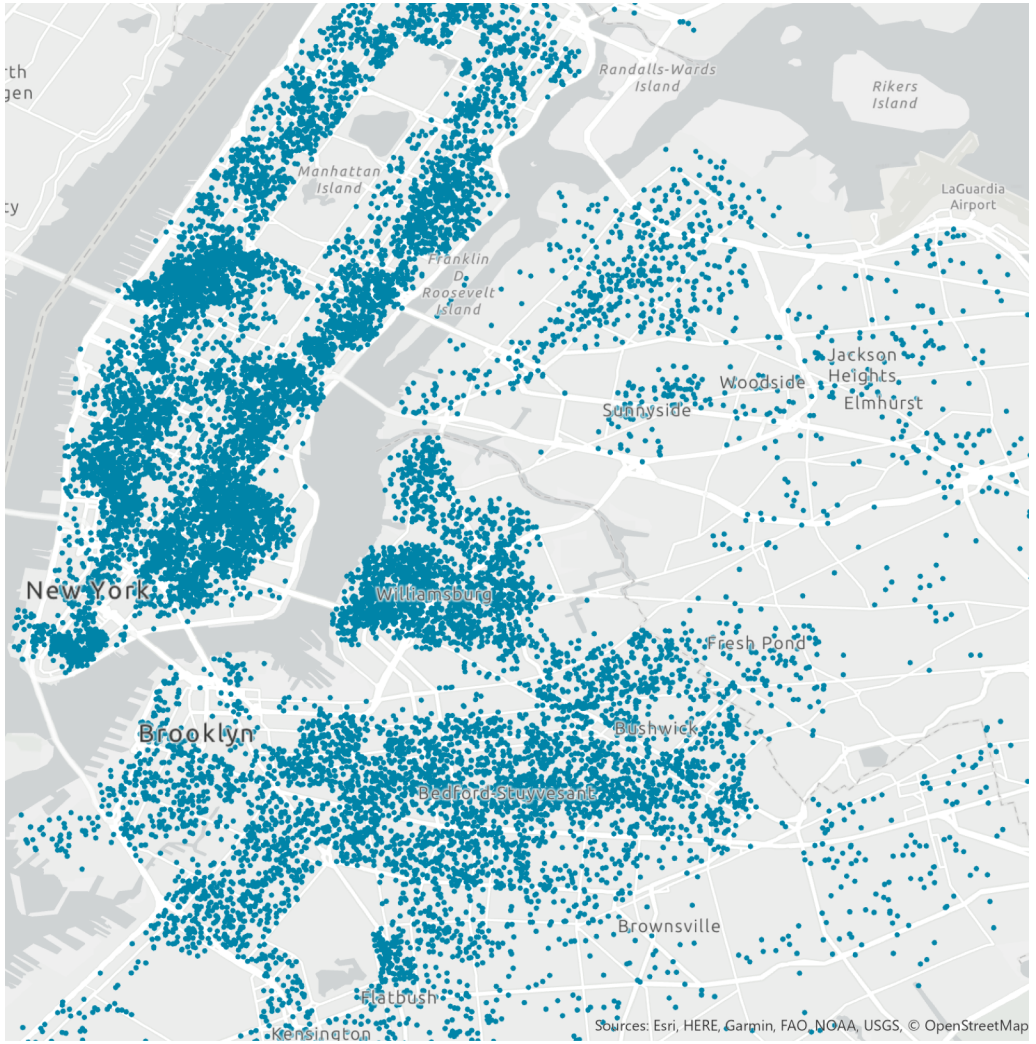


Exploring the data: price distributions



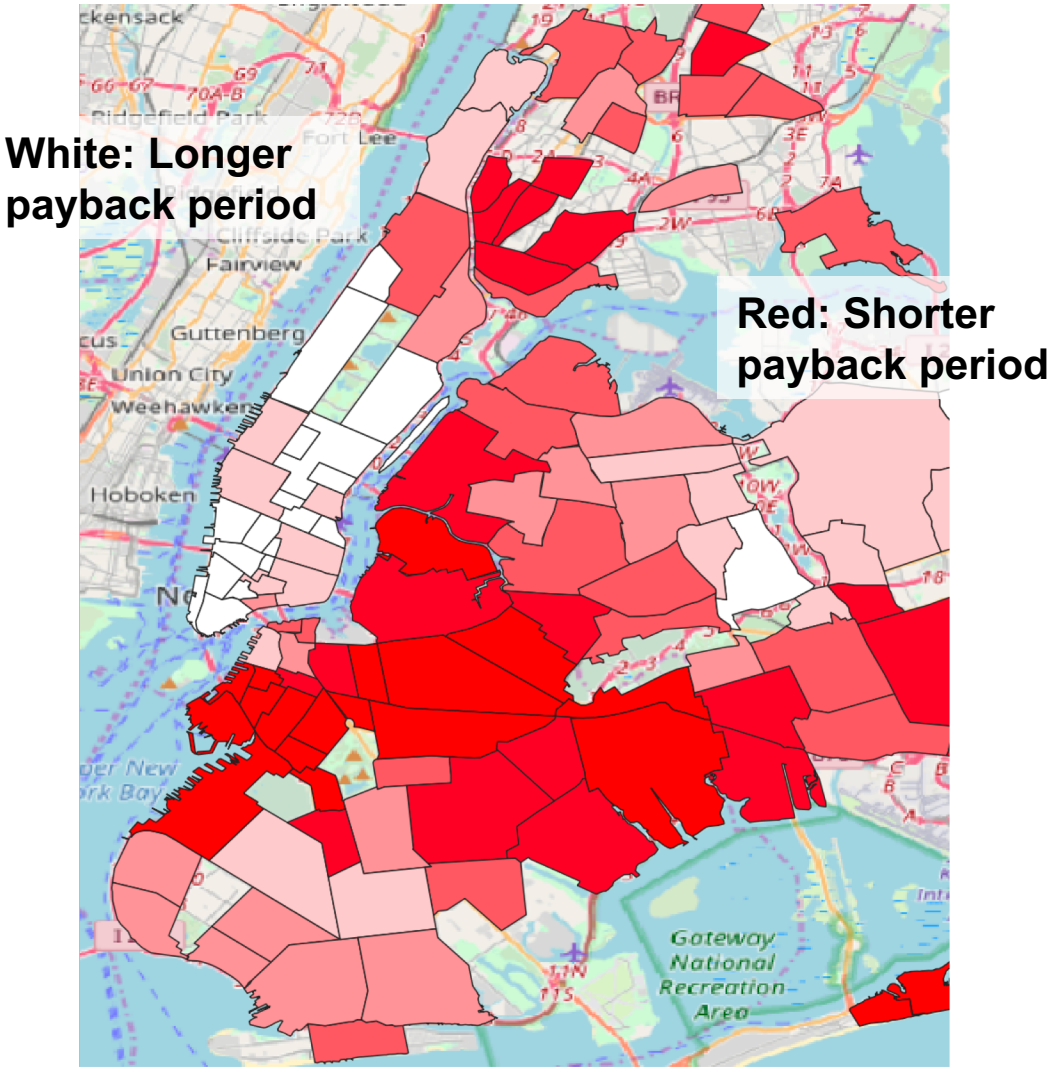
Geospatial modeling

- There are ~19,000 Airbnbs (entire apartments) in New York
- Questions:
  - Are there any geographic patterns in the data?
  - Where to buy property for the best return on Airbnb?
- Methodology:
  - Divide the city and geospatially join sections with Airbnb coordinates (ArcGIS)
  - Compare Airbnb prices to property values to determine where to buy
- Problem: How to subdivide the city?



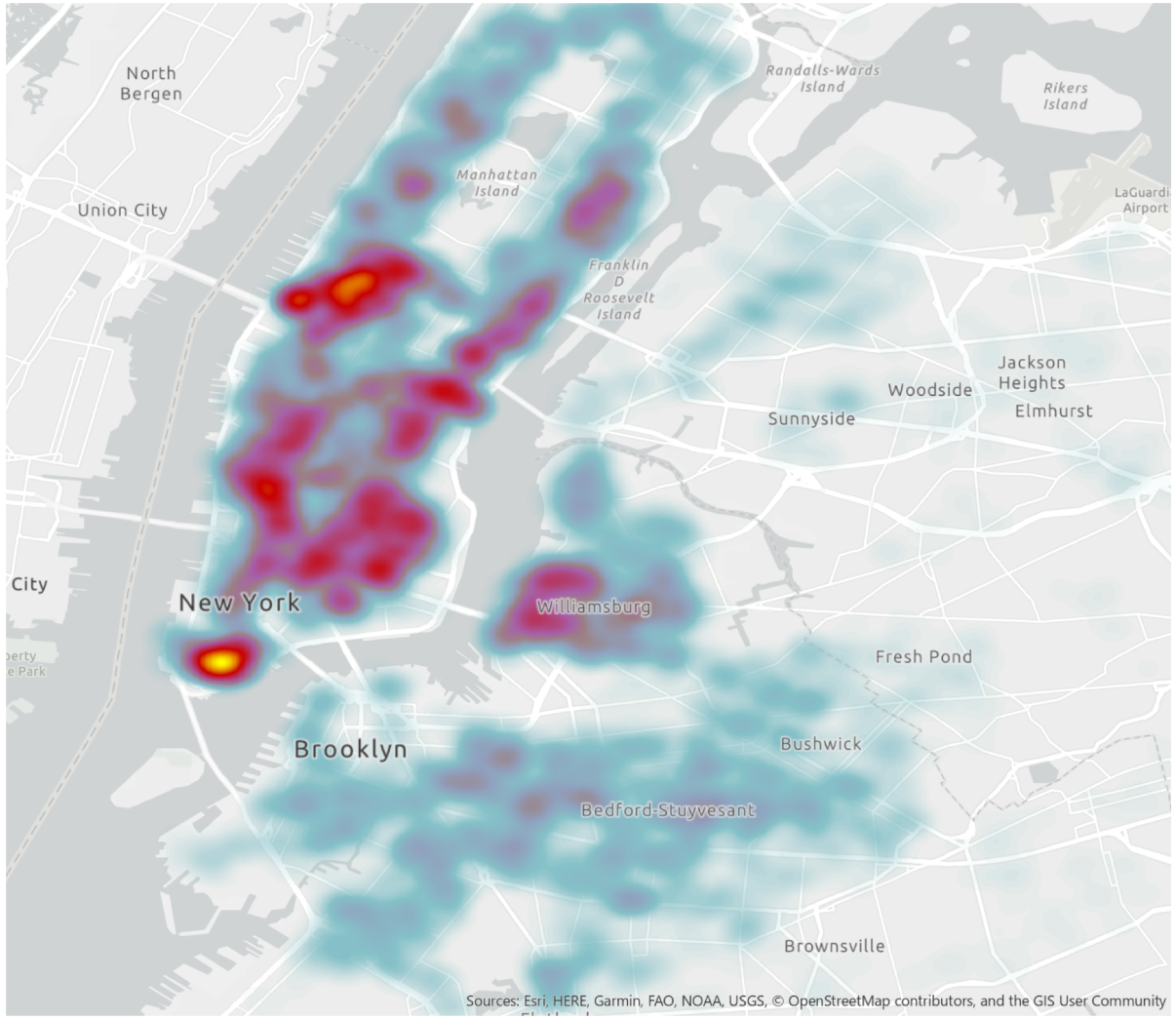
Geospatial modeling

- Problem: The neighborhoods provided by Airbnb give a highly generalized overview
- In reality these neighborhoods are extremely diverse, with luxury high-rises a few blocks away from low-income housing



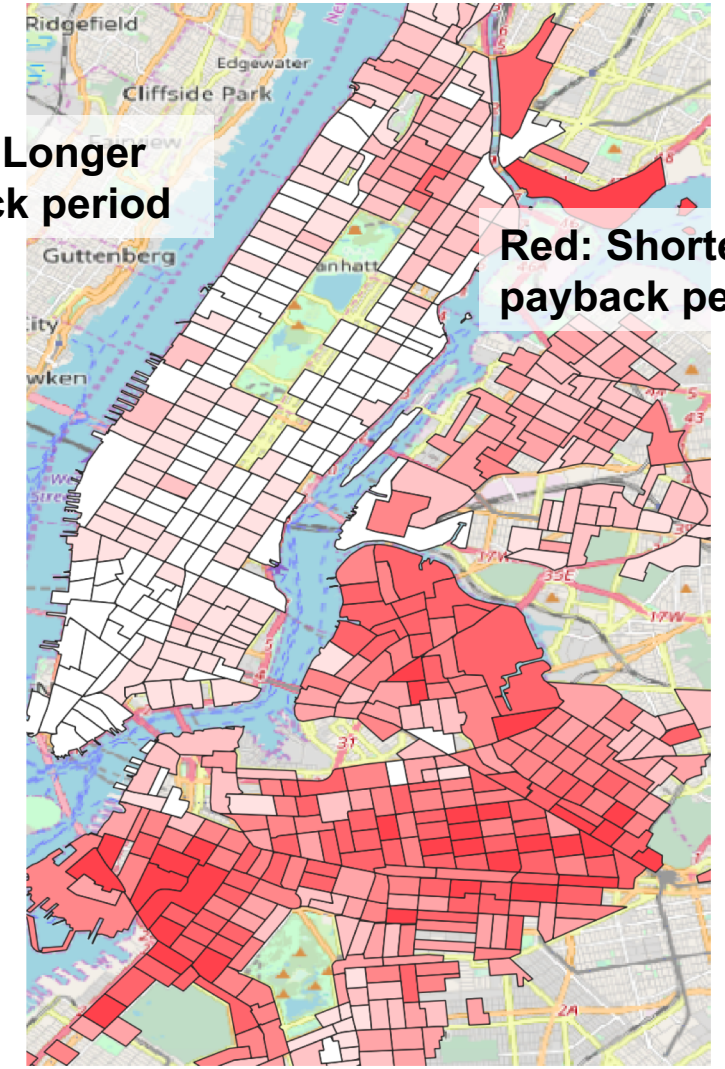
Geospatial modeling

- Idea 1: Show Airbnb prices with a heatmap
- Pros:
  - Easy to build
  - Does not depend on arbitrary boundaries
- Cons:
  - Does not take property values into account
  - Difficult to analyze against other data (tax, census)
  - Too easy



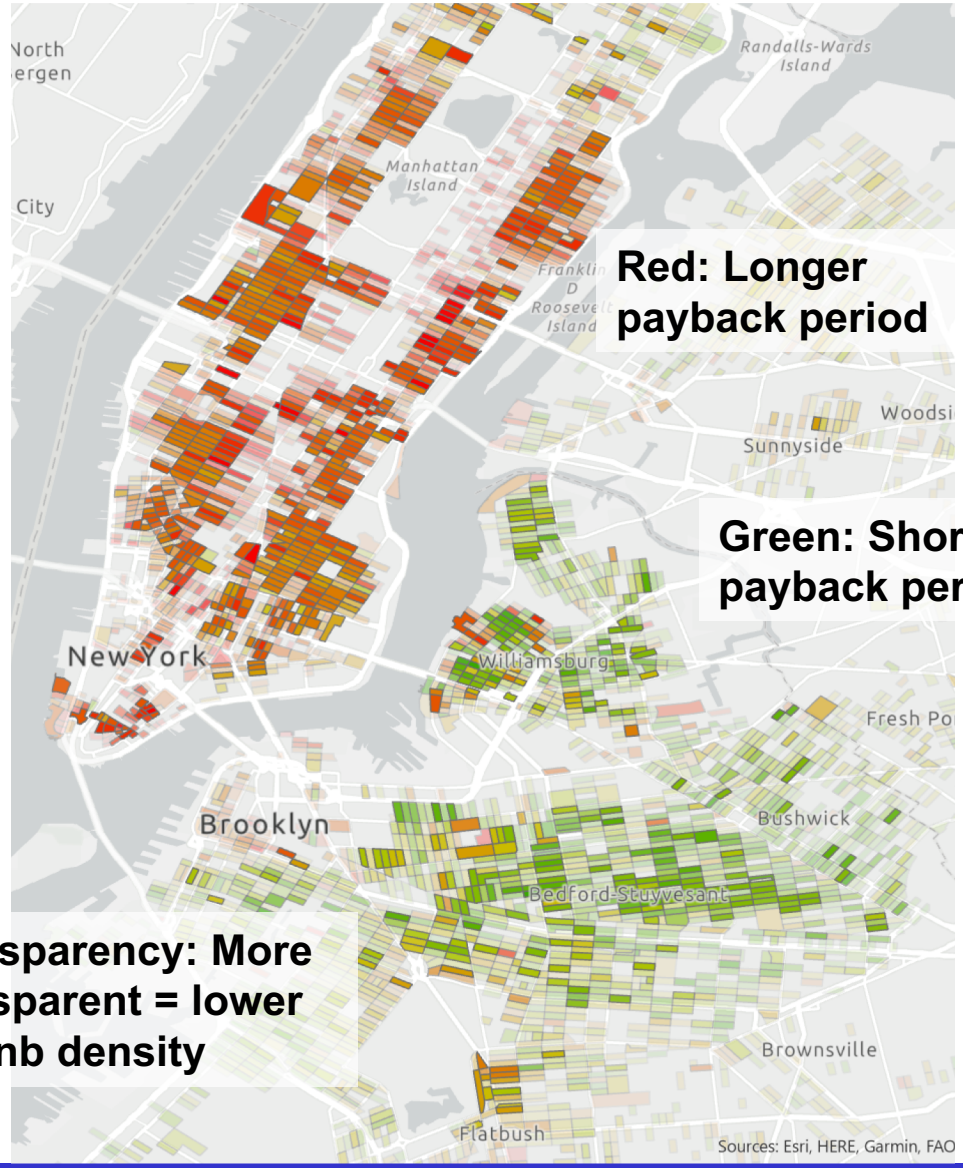
Geospatial modeling

- Idea 2: Divide the city by census tracts
- Pros:
  - Smaller than neighborhoods
  - Easy to define
  - Well documented, easy to digest
- Cons:
  - Still too big – too much price variation



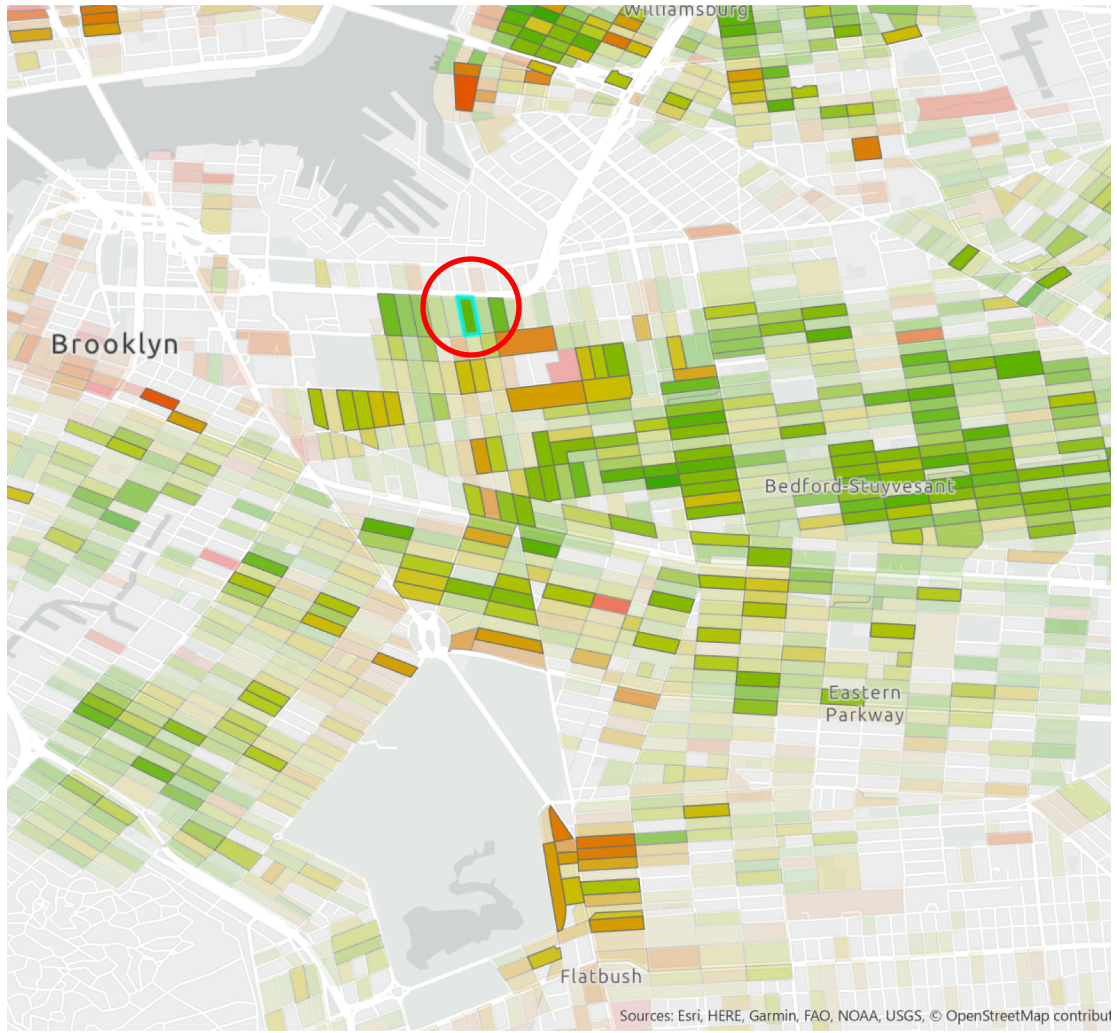
Geospatial modeling

- Idea 3: Divide the city by census blocks
- Pros:
  - Much more granular than tracts
  - Less price variation
  - Can be joined with property tax and and census data
- Cons:
  - Computationally intensive



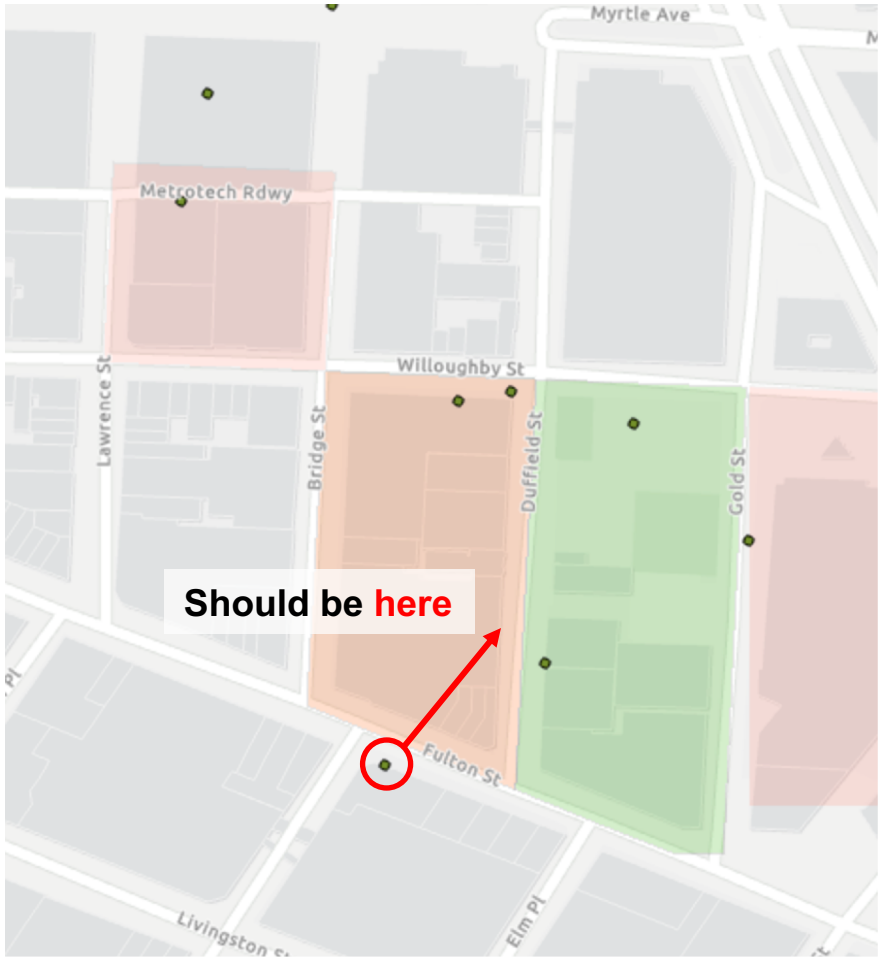
Geospatial modeling

- Example: Fort Greene
- Median rent: $3300 (approx.)
- Median nightly rate: $224
- Median payback period: 176 nights



Geospatial modeling

- Caveats:
  - Property taxes do not always coincide with rents
  - Airbnbs are not necessarily a perfect cross-section of rentals
  - For privacy reasons, Airbnb coordinates are not 100% accurate



Machine learning models for price prediction

- The number one factor that affects a customer's choice of rental is price (followed by the rental property type and the quality of the neighborhood, and other factors).

- Can accurate models be built to reliably predict the price of a rental given features such as: neighborhood, rental type, amenities, etc.?

- To answer this question, we resort to machine learning methodologies.

Data splitting

- Before any preprocessing was done, the data was split in a 80%:20% ratio with 80% comprising the training data and 20% comprising the set data.

- The division between training and test set is an attempt to replicate the situation where you have past information and are building a model which you will test on future as-yet unknown information.

- More specifically, anything done on the training data should not be informed by the test data (the philosophy here is that the future should not affect the past).

Feature selection

- Price was the response/dependent variable; there were 105 features (or predictors) of 63 were categorical and 42 were numerical.

- **Feature selection:** The features were selected with the customer in mind. Features like host picture verification, description of rental property were dropped as they virtually have no effect on price.

- We ended up selecting two sets of features: a small set of 26 features and a large set with 51 features.

- The discussions to follow will be focused on the data with the large set of features.

Feature selection

- Example: small feature set of size 26

11 categorical features=['amenities', 'bed_type', 'cancellation_policy', 'host_has_profile_pic', 'host_identity_verified', 'host_is_superhost', 'instant_bookable', 'neighbourhood_cleansed', 'property_type', 'require_guest_profile_picture', 'room_type']

15 numerical features=['accommodates', 'bathrooms', 'bedrooms', 'beds', 'cleaning_fee', 'extra_people', 'guests_included', 'latitude', 'longitude', 'maximum_nights', 'minimum_nights', 'number_of_reviews', 'price', 'security_deposit', 'square_feet']

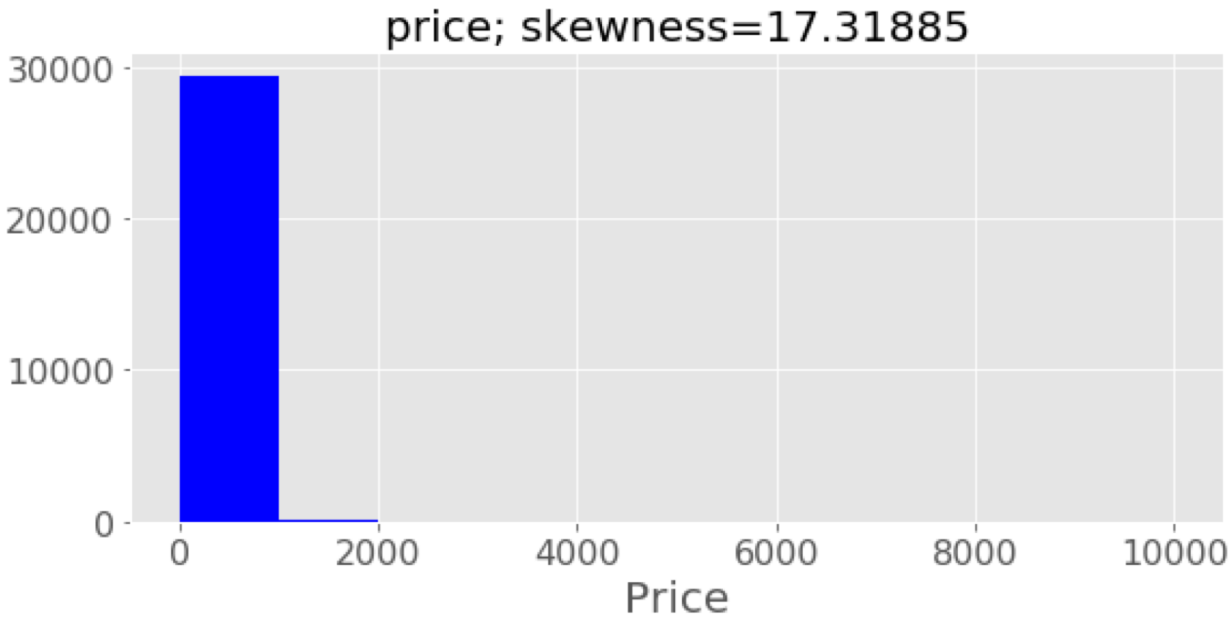
Cleaning the data

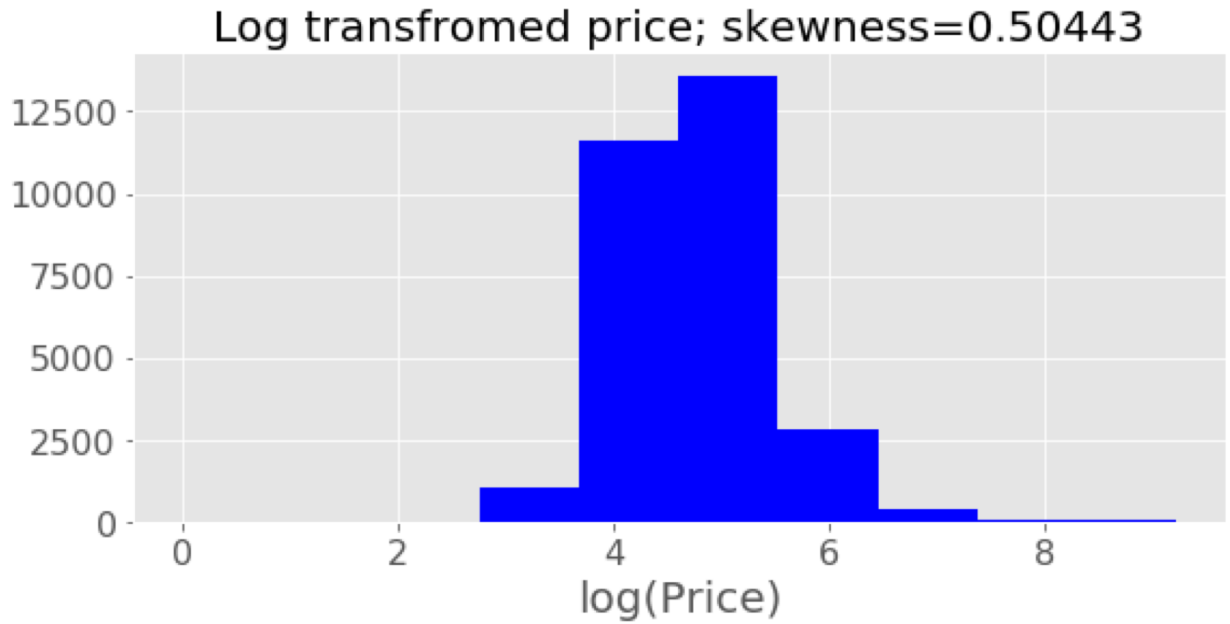
- Missing numerical values were imputed the with zero or the mean value of the feature; missing categorical values were imputed with the mode of the feature.

	Feature	Missing values	Feature type	Percentage missing	First 5 values
0	security_deposit	8479	numerical	28.347431	(0.0, 0.0, 0.0, 600.0, 0.0)
1	cleaning_fee	4613	numerical	15.422420	(10.0, nan, 20.0, 80.0, 15.0)
2	zipcode	265	categorical	0.885962	(10453, 10451, 10002, 10010, 11211)
3	amenities	46	categorical	0.153790	(TV,TV,Internet,Wifi,Air conditioning,Free str...
4	city	44	categorical	0.147103	(Bronx, The Bronx, New York, New York, Brooklyn)
5	bathrooms	27	numerical	0.090268	(1.0, 1.0, 1.0, 1.0, 1.0)
6	bedrooms	16	numerical	0.053492	(1.0, 1.0, 1.0, 1.0, 1.0)
7	beds	15	numerical	0.050149	(1.0, 1.0, 1.0, 1.0, 1.0)
8	host_is_superhost	5	categorical	0.016716	(f, f, f, t, t)
9	host_total_listings_count	5	numerical	0.016716	(8.0, 1.0, 5.0, 1.0, 1.0)
10	host_has_profile_pic	5	categorical	0.016716	(t, t, t, t, t)
11	host_identity_verified	5	categorical	0.016716	(t, f, t, t, f)

Response variable transformation

The skewness of the response variable (price) was corrected using a logarithm transformation

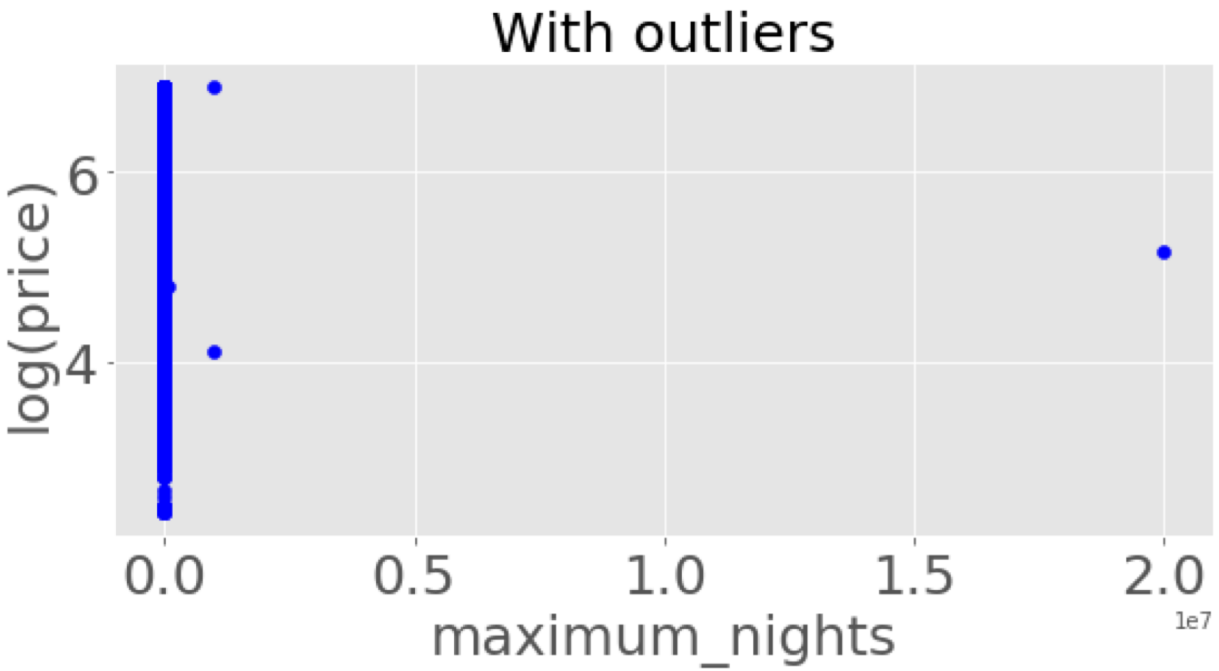


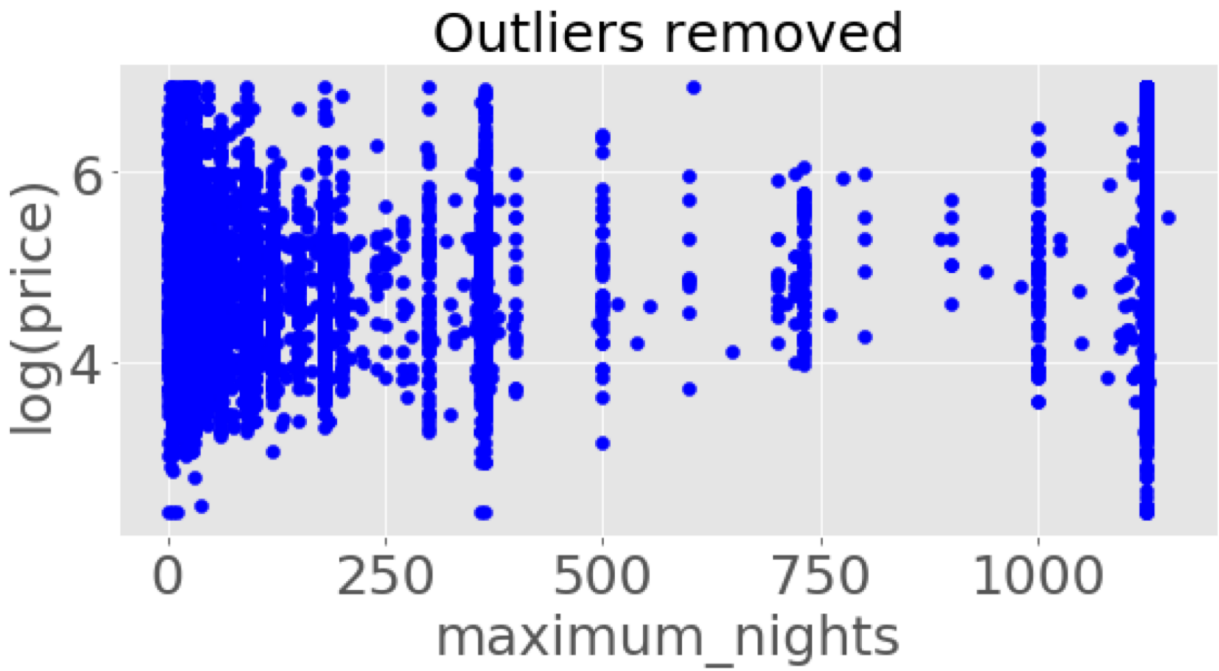


Removal of outliers

Outliers in the numerical features were removed via visual/graphical inspection.

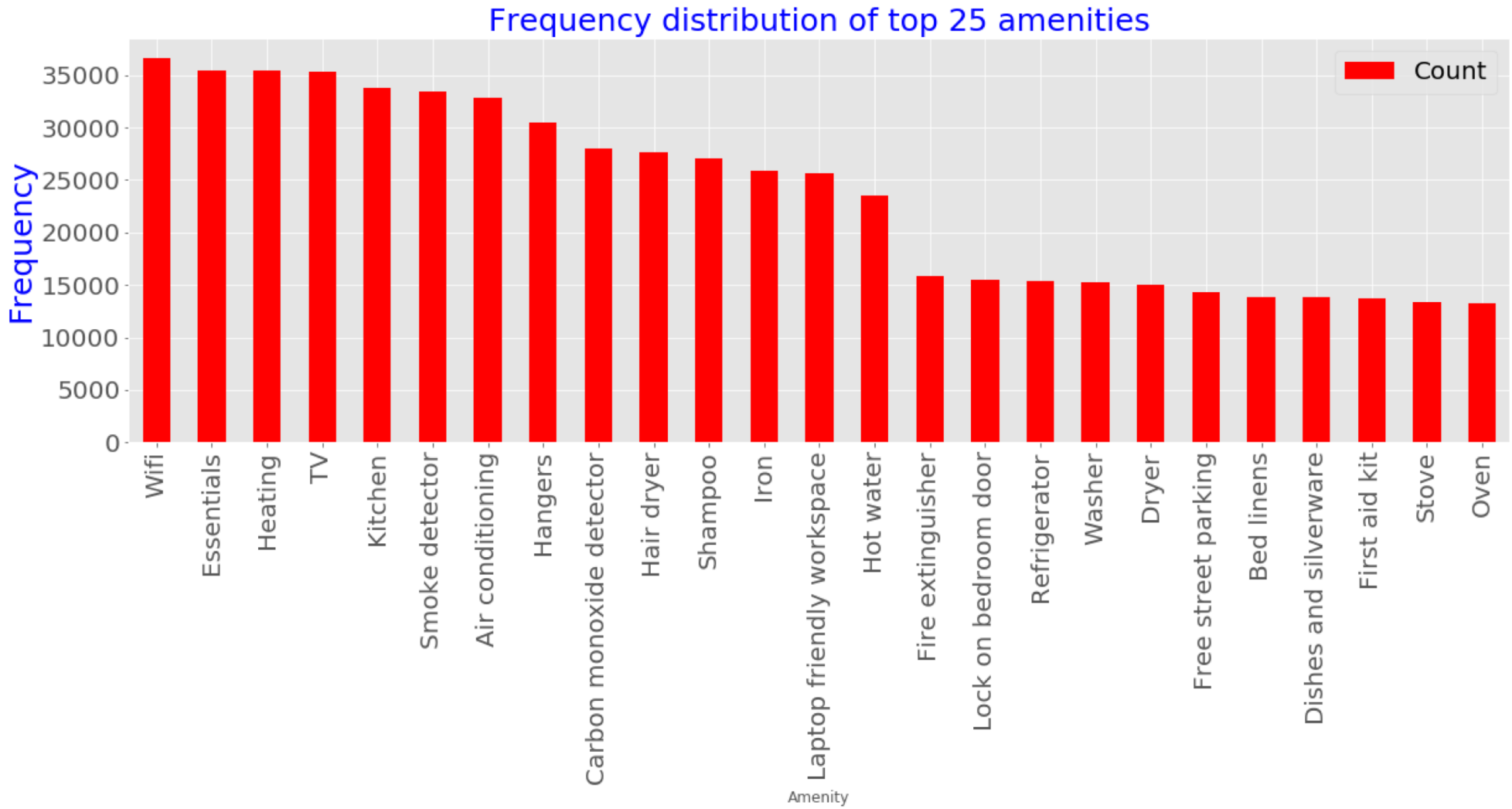
Below is an example of outlier removal in the feature maximum_nights:





Feature engineering

The amenities feature comprises several sub-features such as Wifi, TV, Heating, etc (128 of in total). Some were more frequent that others.



Feature engineering

The amenities feature was engineered as follows:

(i) Each sub-feature $i \in \{\mathbf{Wifi, TV, heating, pool, \ldots.}\}$ was assigned weight *w*(*i*) given by:

$$w(i) = \frac{f(i)}{f_{max}}$$

where $f(i)$ is the frequency of i in the data set and f_{max} is the maximum frequency.

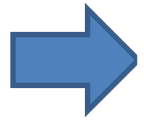
(ii) The sum of the weights $\mathbf{s}$ is used to represent the value of each row of amenities column:

$$s = \sum w(i)$$

Feature engineering

```
train['amenities'].head()
```

```
0    TV,TV,Internet,Wifi,Air conditioning,Free stre...
1    Wifi,Air conditioning,Kitchen,Heating,Smoke de...
2    Wifi,Air conditioning,Kitchen,Heating,Smoke de...
3    TV,TV,Internet,Wifi,Air conditioning,Kitchen,P...
4    Wifi,Pets allowed,Heating,Smoke detector,Carbo...
Name: amenities, dtype: object
```



```
train['amenities'].head()
```

```
0    13.464043
1    10.242527
2    10.817777
3    17.599993
4     7.089850
Name: amenities, dtype: float64
```

- The remaining categorical variables were one-hot encoded (binary categorical variables) or label-encoded.

Regression models

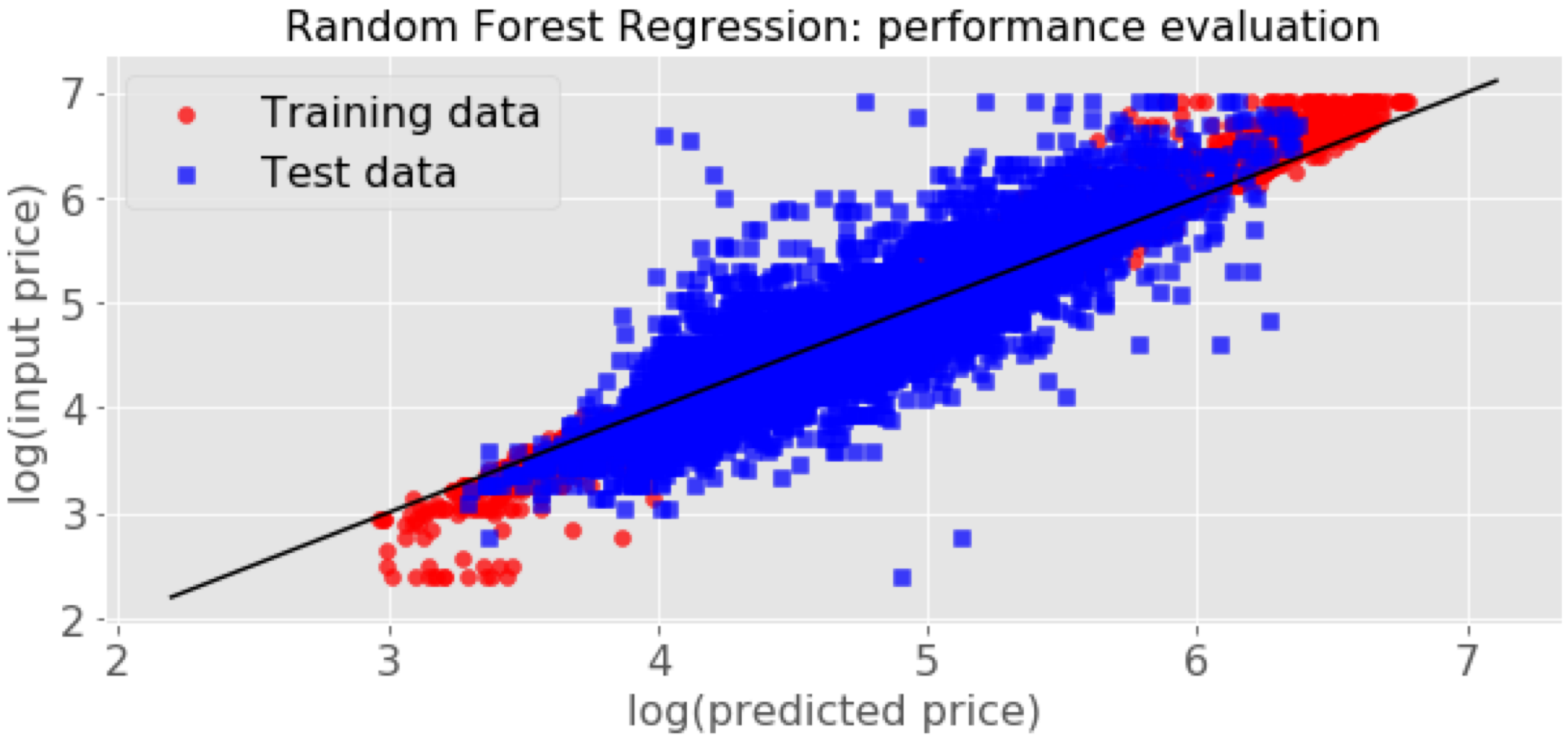
- Lasso regression
- Gradient boosting regression
- Extreme gradient boosting regression
- Random forest regression

Each regression method was hyperparameter-tuned using a grid search and a 10-fold cross-validation

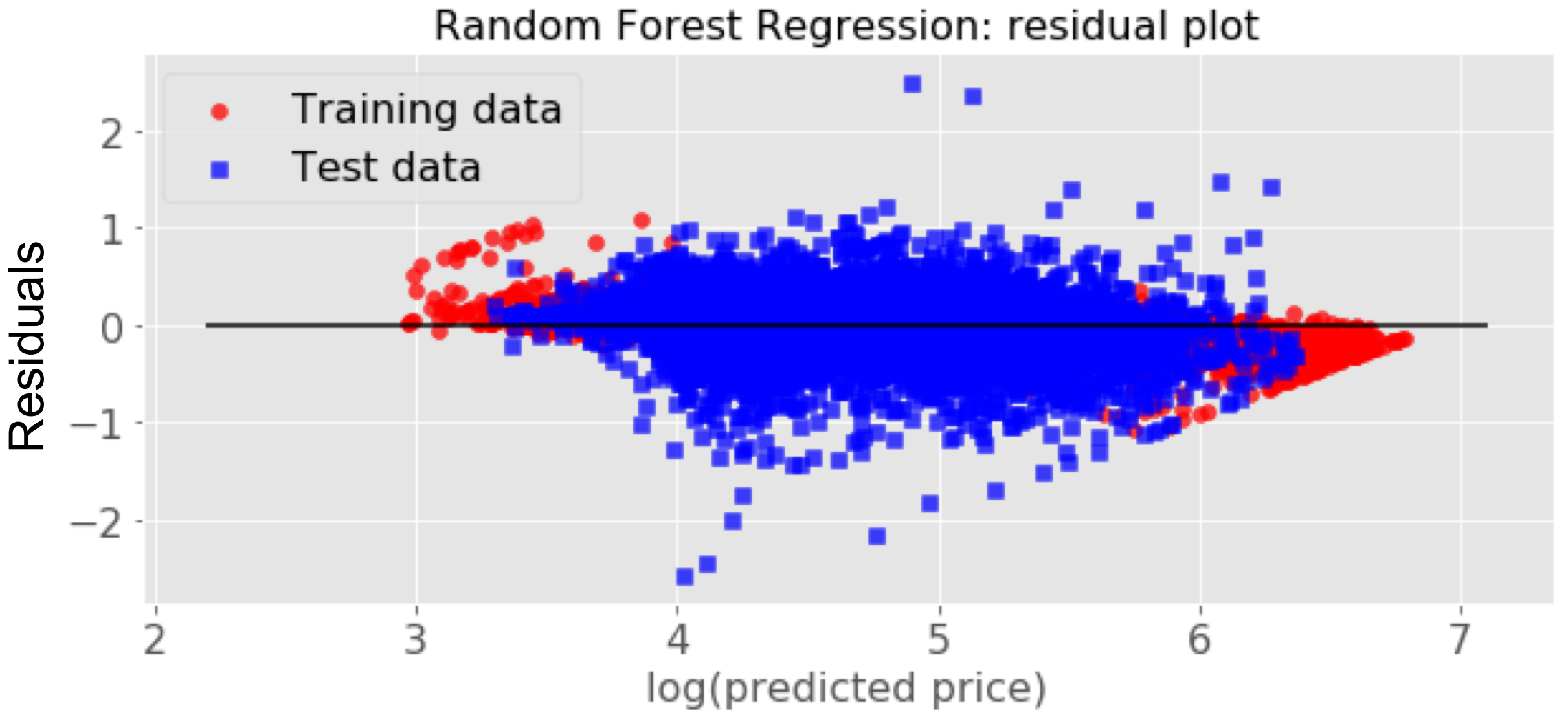
Results

Model	Training Score	Training RMSE	Test RMSE	Training R^2	Test R^2	Training Accuracy	Test Accuracy
Lasso	0.1563	$82	$79	65%	64%	69%	69%
GBR	0.0904	$40	$61	92%	79%	86%	77%
XGBR	0.0918	$35	$62	93%	78%	86%	77%
RF	0.0990	$32	$67	97%	77%	91.6%	76%

Results

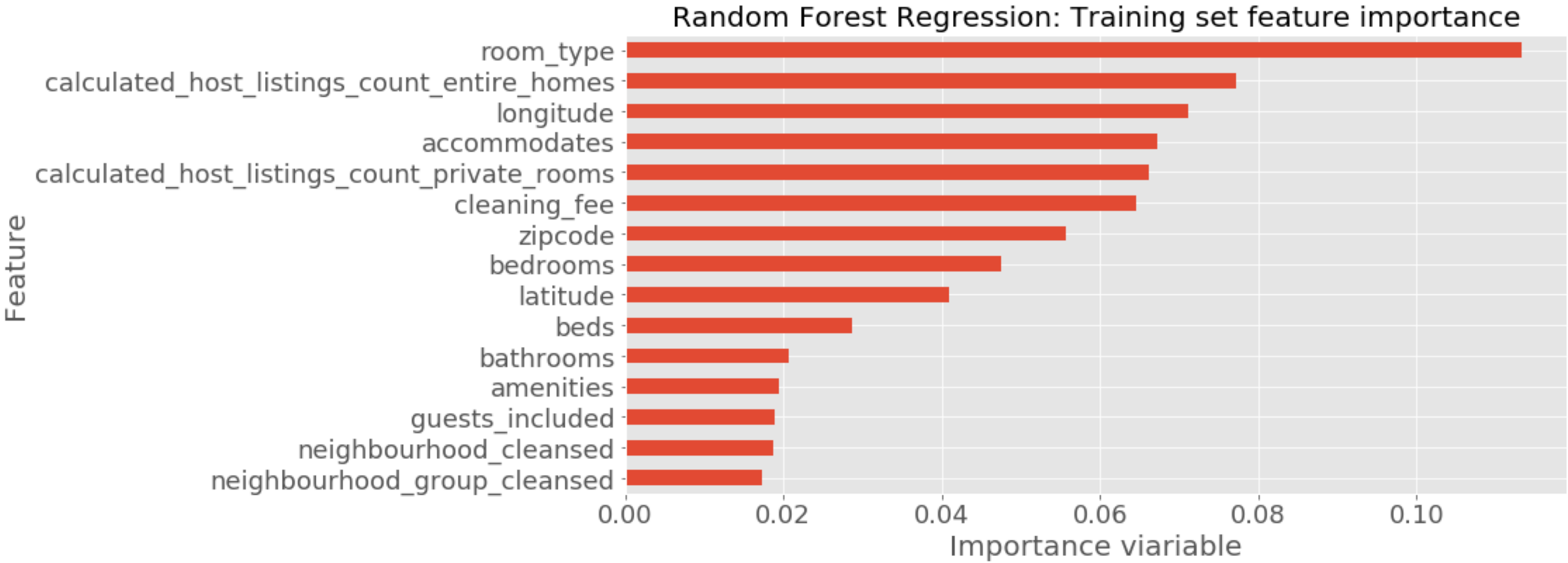

Random Forest Regression: performance evaluation
Training data
Test data
log(input price)
log(predicted price)
2
3
4
5
6
7

Results


Random Forest Regression: residual plot
Training data
Test data
Residuals
2
1
0
−1
−2
2
3
4
5
6
7
log(predicted price)

Results

- Random forest “room_type” was the feature with the most importance



Conclusions

- When it comes to Airbnb pricing, there are definitive numeric and geospatial patterns in pricing
- Understanding these patterns can benefit guests and hosts alike (but not Airbnb!)
- Next steps:
  - Join with census data to extrapolate key attributes of profitable blocks for clustering
  - Integrate better real estate pricing data to identify properties for sale
  - Explore other cities
  - Turn into a product (Flask web app)