אוניברסיטת בר-אילן
Bar-Ilan University

Data Since Project

Berlin Ratings

by Ilya Zutler

2024

Ramat Gan, Israel

Structure of Project

The Goal of the project is to investigate apartments rented in Berlin via Airbnb and the relationship between: apartment details, polarity of comments and ratings with the rental price in data frame:

- https://www.kaggle.com/datasets/thedevastator/berlin-airbnb-ratings-how-hosts-measure-up

The project includes the following stages:

- Data Preparation
- Exploratory Data Analysis
- Data Cleaning & Outliers
- Feature Engineering & Data Transformation
- Model Selection and Feature Importance
- Fine Tuning with Cross-Validation

Data Preparation

The data frame contains information about apartments rented in Berlin via Airbnb and guests' comments about them.

457k

Guest	Comment	ID Aprt	Apartment details	Price	Rating
Mitchell	Great	1	2 badrooms 1 Bathrooms	60	98.5
Cindy	Not bad	1	2 badrooms 1 Bathrooms	60	98.5
Marco	Wonderful	1	2 badrooms 1 Bathrooms	60	98.5
		2	1 badrooms 1 Bathrooms	40	
Martina	So-so	3	3 badrooms 2 Bathrooms	145	95

The data frame has 457k rows. Rows contain:

- Comment of one guest
- Apartment details (location, number of rooms, etc.)
- Rental Price
- Apartment rating (average for all reviews)

The apartment data is repeated in rows many times.

Data Preparation

The data frame contains information about apartments rented in Berlin via Airbnb and guests' comments about them.

457k

Guest	Comment	ID Aprt	Apartment details	Price	Rating
Mitchell	Great	1	2 badrooms 1 Bathrooms	60	98.5
Cindy	Not bad	1	2 badrooms 1 Bathrooms	60	98.5
Marco	Wonderful	1	2 badrooms 1 Bathrooms	60	98.5
		2	1 badrooms 1 Bathrooms	40	
Martina	So-so	3	3 badrooms 2 Bathrooms	145	95

The data frame has 457k rows. Rows contain:

- Comment of one guest
- Apartment details (location, number of rooms, etc.)
- Rental Price
- Apartment rating (average for all reviews)

The apartment data is repeated in rows many times.

The data frame contains data about 23.5k unique apartments, of which 4.1k apartments have no comments and ratings.

23.5k
(4.1k have no Comments and Rating)

ID Aprt	Apartment details	Price	Rating
1	2 badrooms 1 Bathrooms	60	98.5
2	1 badrooms 1 Bathrooms	40	
3	3 badrooms 2 Bathrooms	145	95

Data Preparation

The data frame contains information about apartments rented in Berlin via Airbnb and guests' comments about them.

457k

Guest	Comment	ID Aprt	Apartment details	Price	Rating
Mitchell	Great	1	2 badrooms 1 Bathrooms	60	98.5
Cindy	Not bad	1	2 badrooms 1 Bathrooms	60	98.5
Marco	Wonderful	1	2 badrooms 1 Bathrooms	60	98.5
		2	1 badrooms 1 Bathrooms	40	
Martina	So-so	3	3 badrooms 2 Bathrooms	145	95

The data frame has 457k rows. Rows contain:

- Comment of one guest
- Apartment details (location, number of rooms, etc.)
- Rental Price
- Apartment rating (average for all reviews)

The apartment data is repeated in rows many times.

23.5k (4.1k have no Comments and Rating)

ID Aprt	Apartment details	Price	Rating
1	2 badrooms 1 Bathrooms	60	98.5
2	1 badrooms 1 Bathrooms	40	
3	3 badrooms 2 Bathrooms	145	95

The data frame contains data about 23.5k unique apartments, of which 4.1k apartments have no comments and ratings.

ID Aprt	Comment	Polarity
1	Great	0.9
1	Not bad	0.4
1	Wonderful	1
3	So-so	0.1

added Polatity using TextBlob

The Polarity of comments was analyzed using TextBlob and translation from different languages.

Data Preparation

The data frame contains information about apartments rented in Berlin via Airbnb and guests' comments about them.

The data frame has 457k rows. Rows contain:

- Comment of one guest
- Apartment details (location, number of rooms, etc.)
- Rental Price
- Apartment rating (average for all reviews)

The apartment data is repeated in rows many times.

The data frame contains data about 23.5k unique apartments, of which 4.1k apartments have no comments and ratings.

The Polarity of comments was analyzed using TextBlob and translation from different languages.

Aggregated data on the Polarity of comments for each apartment were added to the Apartments table (min, max, mean, median polarity) .

The resulting Apartments table is a flat file for further analysis.

457k

Guest	Comment	ID Aprt	Apartment details	Price	Rating
Mitchell	Great	1	2 badrooms 1 Bathrooms	60	98.5
Cindy	Not bad	1	2 badrooms 1 Bathrooms	60	98.5
Marco	Wonderful	1	2 badrooms 1 Bathrooms	60	98.5
		2	1 badrooms 1 Bathrooms	40	
Martina	So-so	3	3 badrooms 2 Bathrooms	145	95

23.5k (4.1k have no Comm and Rating)

ID Aprt	Apartment details	Price	Rating
1	2 badrooms 1 Bathrooms	60	98.5
2	1 badrooms 1 Bathrooms	40	
3	3 badrooms 2 Bathrooms	145	95

ID Aprt	Comment	Polarity
1	Great	0.9
1	Not bad	0.4
1	Wonderful	1
3	So-so	0.1

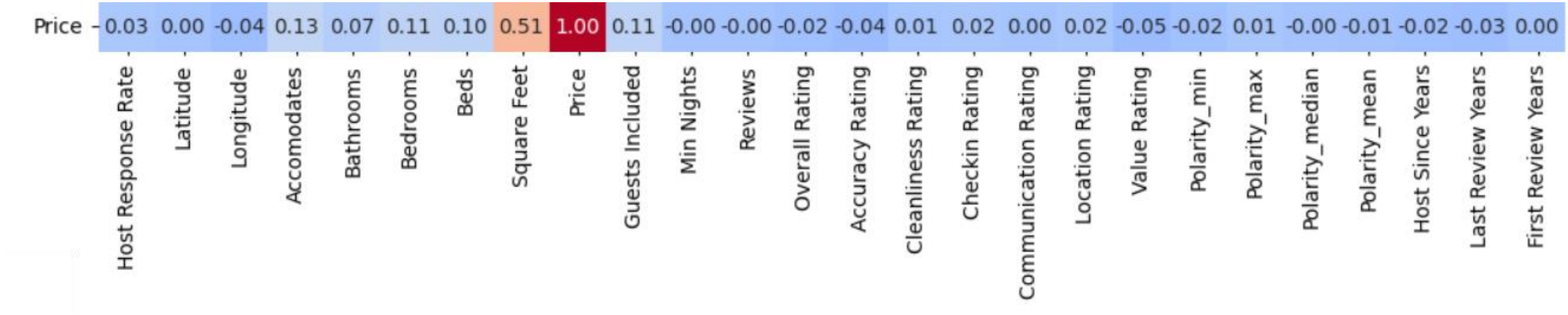
added Polatity using TextBlob

23.5k (4.1k have no Polarity and Rating)

ID Aprt	Apartment details	Price	Rating	Avg Polarity
1	2 badrooms 1 Bathrooms	60	98.5	0.77
2	1 badrooms 1 Bathrooms	40		
3	3 badrooms 2 Bathrooms	145	95	0.1

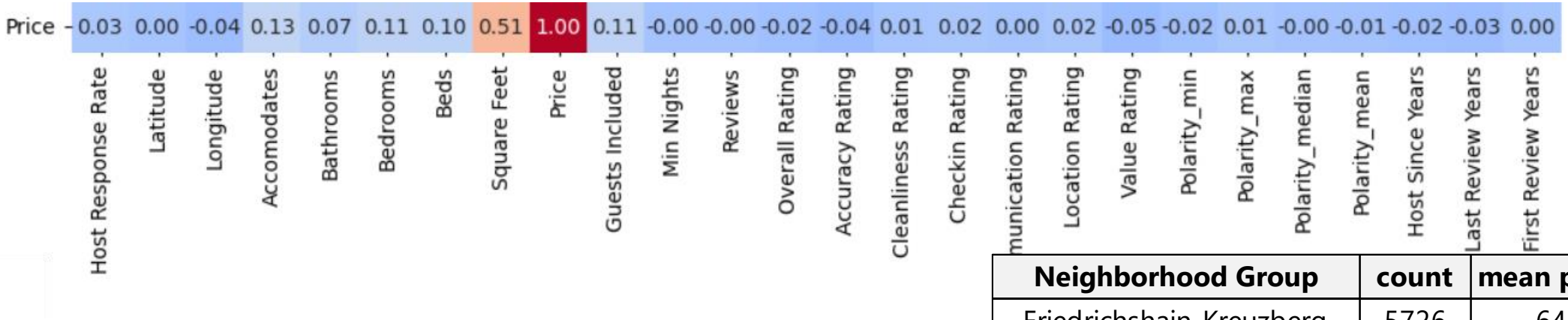
Exploratory Data Analysis

Data Protocol, Descriptive Statistics, visualization of Missing data, Numerical and Categorical features , Cross-correlations and Skewness were created. A low correlation was found between numerical data and accommodation price.



Exploratory Data Analysis

Data Protocol, Descriptive Statistics, visualization of Missing data, Numerical and Categorical features , Cross-correlations and Skewness were created. A low correlation was found between numerical data and accommodation price.



The ANOVA test showed significant differences in Prices in different Berlin districts: *F-value* = 12.88, *p-value* = 0.000.

The significance of paired differences also was clarified using t-test.

Neighborhood Group	count	mean price	std price
Friedrichshain-Kreuzberg	5726	64	131
Mitte	4865	72	90
Pankow	3687	72	139
NeukÃ¶lln	3587	48	35
Charlottenburg-Wilm.	1676	112	511
Tempelhof - SchÃ¶neberg	1586	98	474
Lichtenberg	743	67	346
Treptow - KÃ¶penick	626	54	40
Steglitz - Zehlendorf	468	59	57
Reinickendorf	296	45	30
Marzahn - Hellersdorf	147	60	46
Spandau	129	56	50

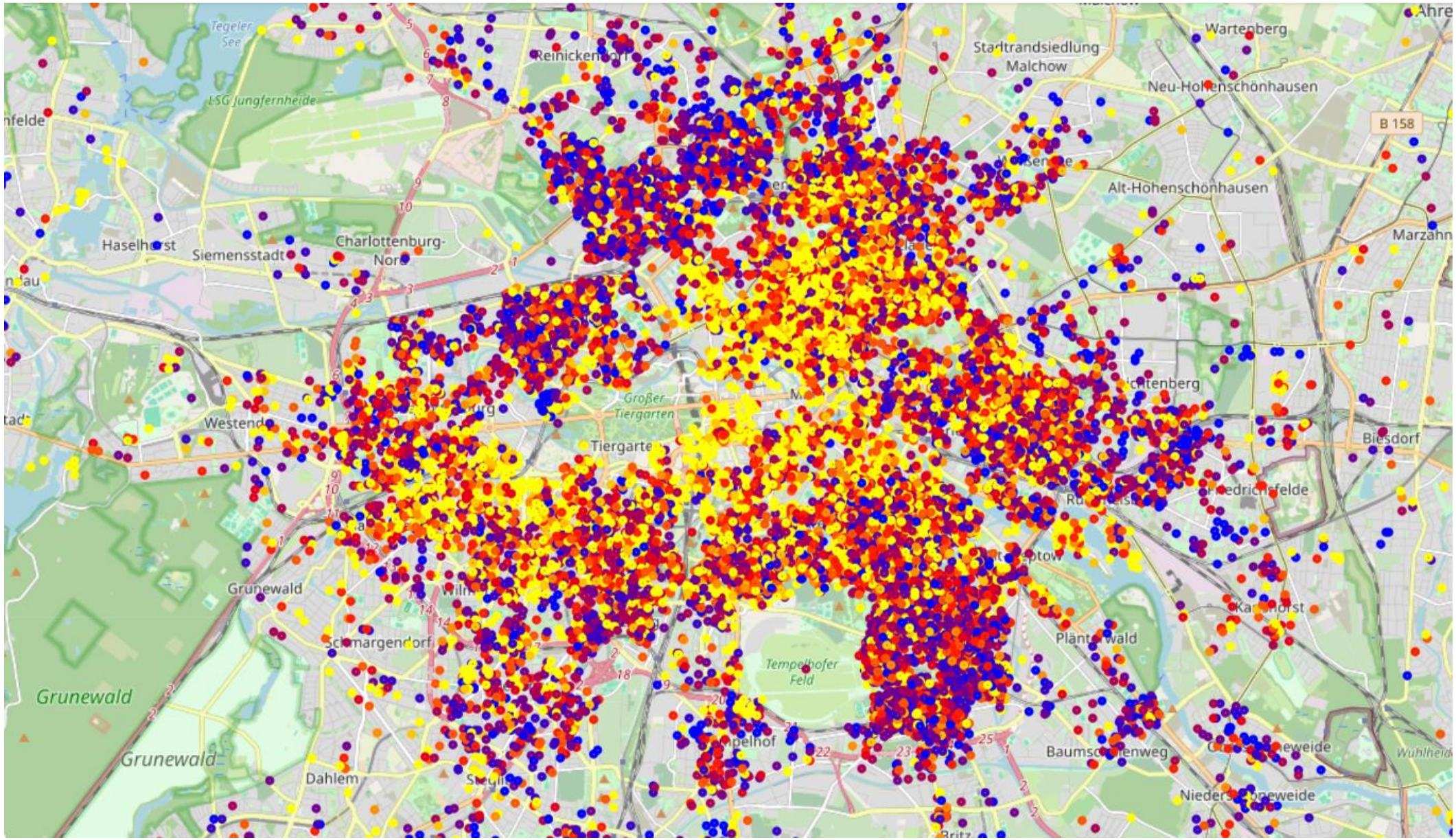
Exploratory Data Analysis

This is a map of apartments, colored according to rental price.

We can see some dependence of prices on neighbors' prices.

Spatial Autocorrelation of Price was calculated using Moran's *I*

Price Moran's *I* : 0.23
p-value = 0.001



Exploratory Data Analysis

Spatial Autocorrelation of Price and Location Rating was calculated using Moran's *I*

Price
Moran's *I* : 0.23
p-value = 0.001

Location Rating
Moran's *I* : 0.22
p-value = 0.001

$$I = \frac{N}{W} \frac{\sum_{i=1}^{N} \sum_{j=1}^{N} w_{ij}(x_i - \bar{x})(x_j - \bar{x})}{\sum_{i=1}^{N} (x_i - \bar{x})^2}$$

where

- N is the number of spatial units indexed by i and j;
- x is the variable of interest;
- $\bar{x}$ is the mean of x;
- w_{ij} are the elements of a matrix of spatial weights;
- and W is the sum of all w_{ij}

NB Computational complexity

To find the closest element to a given one from a set of *N* elements by simple enumeration, *N* operations are required. To find the closest element for each element in the set, about N^2 operations are required.

To reduce the complexity of calculations, it is sometimes possible to divide the set into a grid. How to do it?

For small distances, the angular grid (Latitude-Longitude) can be considered as rectangular (if we are far from the poles, but normally it is) .

This code divides the city into *k* lanes from north to south and *l* lanes from east to west.

```
lat_bin = np.linspace(apartments['Latitude'].min(), apartments['Latitude'].max(), k)
lon_bin = np.linspace(apartments['Longitude'].min(), apartments['Longitude'].max(), l)

apartments['Latitude_Bin'] = pd.cut(apartments['Latitude'], bins=lat_bin, labels=False, include_lowest=True)
apartments['Longitude_Bin'] = pd.cut(apartments['Longitude'], bins=lon_bin, labels=False, include_lowest=True)
```

Data Cleansing & Outliers

Missing ‘Post Code’ we fill according to the nearest apartment with a filled Post Code. It is 1-NN model for haversine distance on the Latitude-Longitude. We use grid to do it quickly.

'Bathrooms', 'Bedrooms', 'Beds’, 'Host Since’ we fill the missing values and outliers using MICE imputation.

Data Cleansing & Outliers

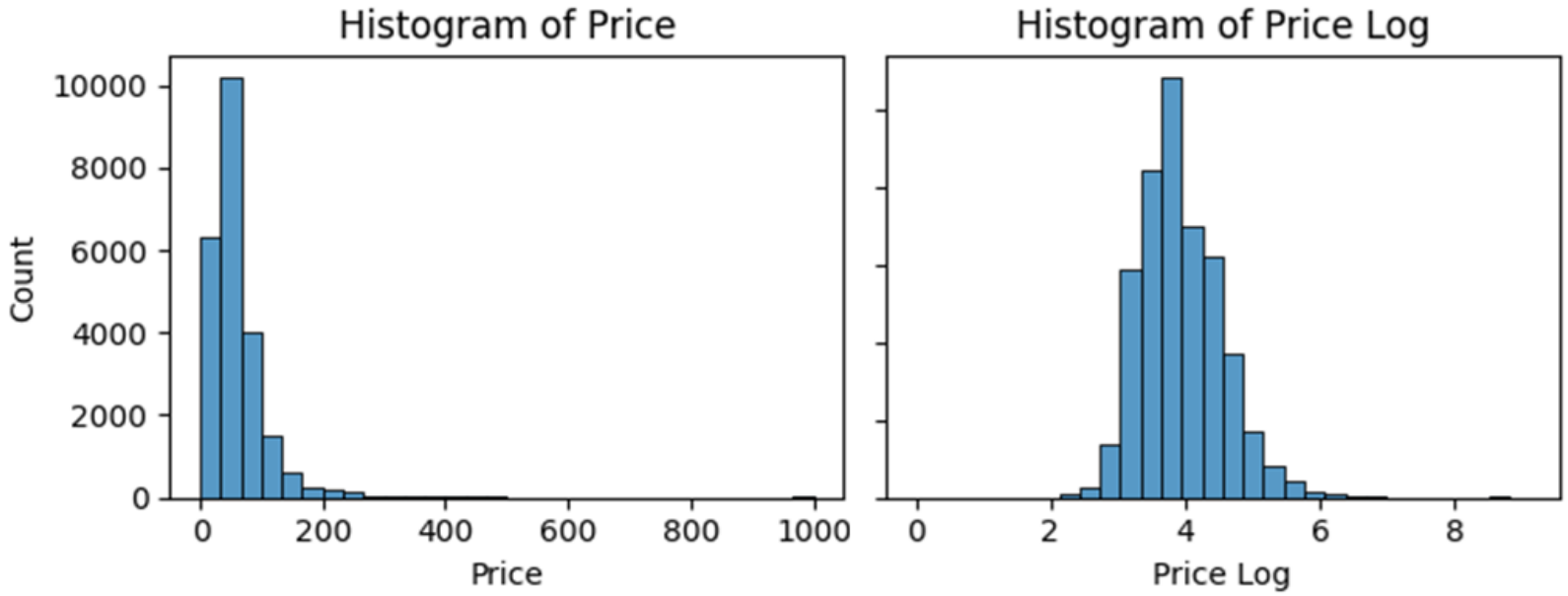
Missing ‘Post Code’ we fill according to the nearest apartment with a filled Post Code. It is 1-NN model for haversine distance on the Latitude-Longitude. We use grid to do it quickly.

'Bathrooms', 'Bedrooms', 'Beds’, 'Host Since’ we fill the missing values and outliers using MICE imputation.

Price’ outliers. Price has a distribution with heavy tails. Log transformation made the distribution more similar to normal.

Here are the outliers boundaries for initial variable and new variable with recalculation them in the original scale.

For the final calculation, I determined the boundaries using more subtle methods. Outliers by 'Price’ - less than 14 € and more than 300€.



Mean Price: 69.61
Standard Deviation Price: 216.66

Calculated Price boundaries based on initial data

3 Sigma Lower: -580.37
3 Sigma Upper: 719.59
1.5 IQR Lower: -32.50
1.5 IQR Upper: 139.50

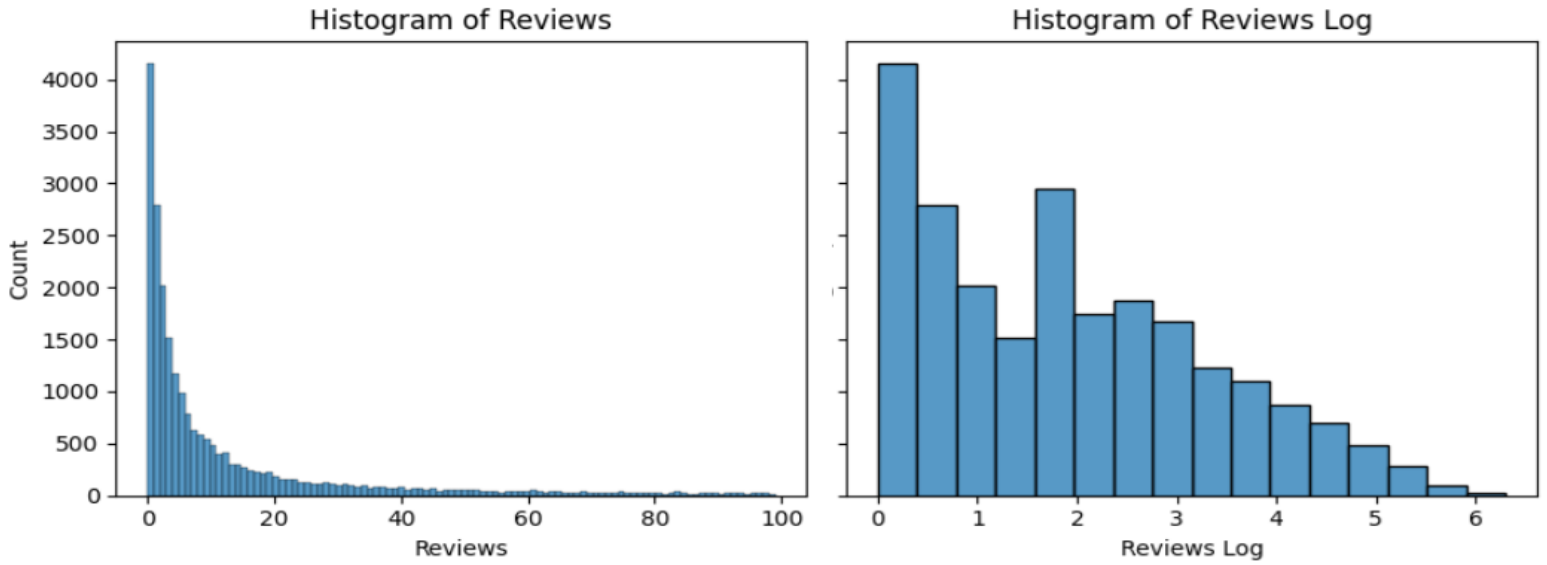
Calculated price boundaries based on transformed data

3 Sigma Lower: 6.66
3 Sigma Upper: 341.98
1.5 IQR Lower: 8.44
1.5 IQR Upper: 264.62

Feature Engineering & Data Transformation

The number of Comments takes values in very wide ranges - from 0 to 500 with an average of 19. It can be inconvenient for linear models.

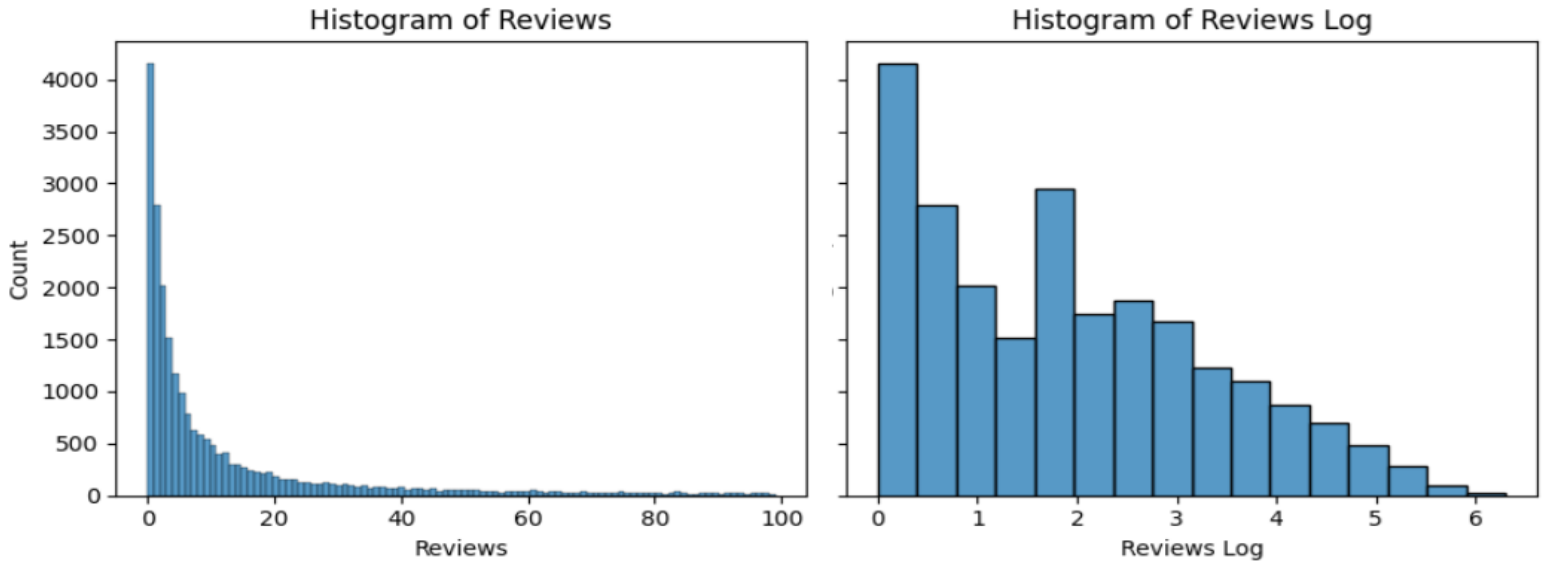
If in the model one comment adds, for example, 0.3 € to the Price, then for some apartments this will immediately add 150 €. Therefor we make Log transformation.



Feature Engineering & Data Transformation

The number of Comments takes values in very wide ranges - from 0 to 500 with an average of 19. It can be inconvenient for linear models.

If in the model one comment adds, for example, 0.3 € to the Price, then for some apartments this will immediately add 150 €. Therefor we make Log transformation.

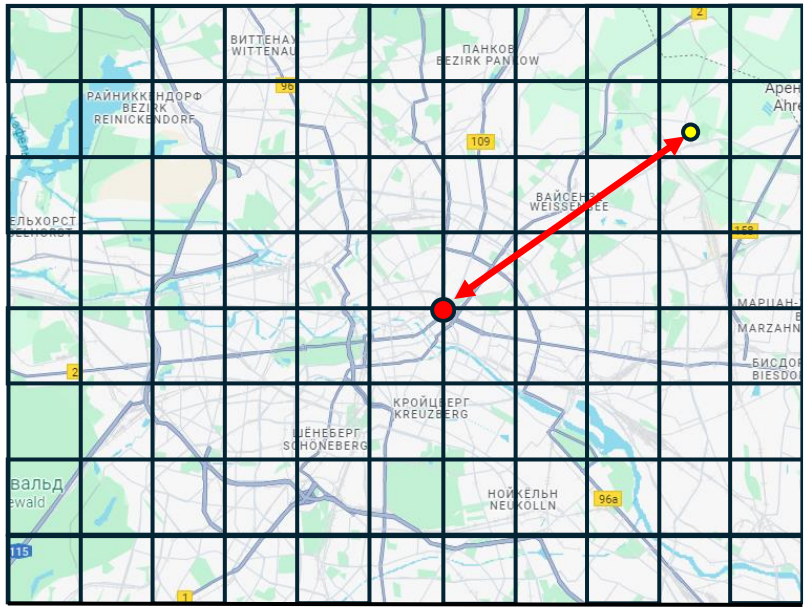


New features are related to the location of apartments.

The city is divided into small squares of 250 x 250 meters and large squares of 500 x 500 meters. Were added new features :

1. index of the small square in which the apartment is located
2. index of the large square in which the apartment is located.

A feature has also been added - distance from the city center. It has a small but reliable correlation with the rental price of the apartment.



Model Selection & Feature Importance

We tested the following regression models in basic settings:

Lasso, Random Forest, CatBoost, LightGBM, XGBoost

for two types of encoding: One-Hot Encoding and Target Encoding.

(in fact, CatBoost and LightGBM do not require preprocessing of categorical features - they do Target Encoding themselves)

Model	Encoding	Train MSE	Test MSE	Train R2	Test R2
Lasso	One-Hot	872,0	919,0	0,485	0,483
Random Forest	One-Hot	103,0	779,9	0,939	0,561
CatBoost	One-Hot	450,8	708,5	0,734	0,602
LightGBM	One-Hot	479,4	713,7	0,717	0,599
XGBoost	One-Hot	364,2	746,0	0,785	0,581
Lasso	Target	785,9	858,6	0,535	0,517
Random Forest	Target	99,1	753,5	0,941	0,576
CatBoost	Target	360,3	709,5	0,787	0,601
LightGBM	Target	442,3	717,5	0,739	0,597
XGBoost	Target	235,5	769,7	0,861	0,567

CatBoost and LightGBM are the best out-of-the-box solution, they show good results without manual settings.

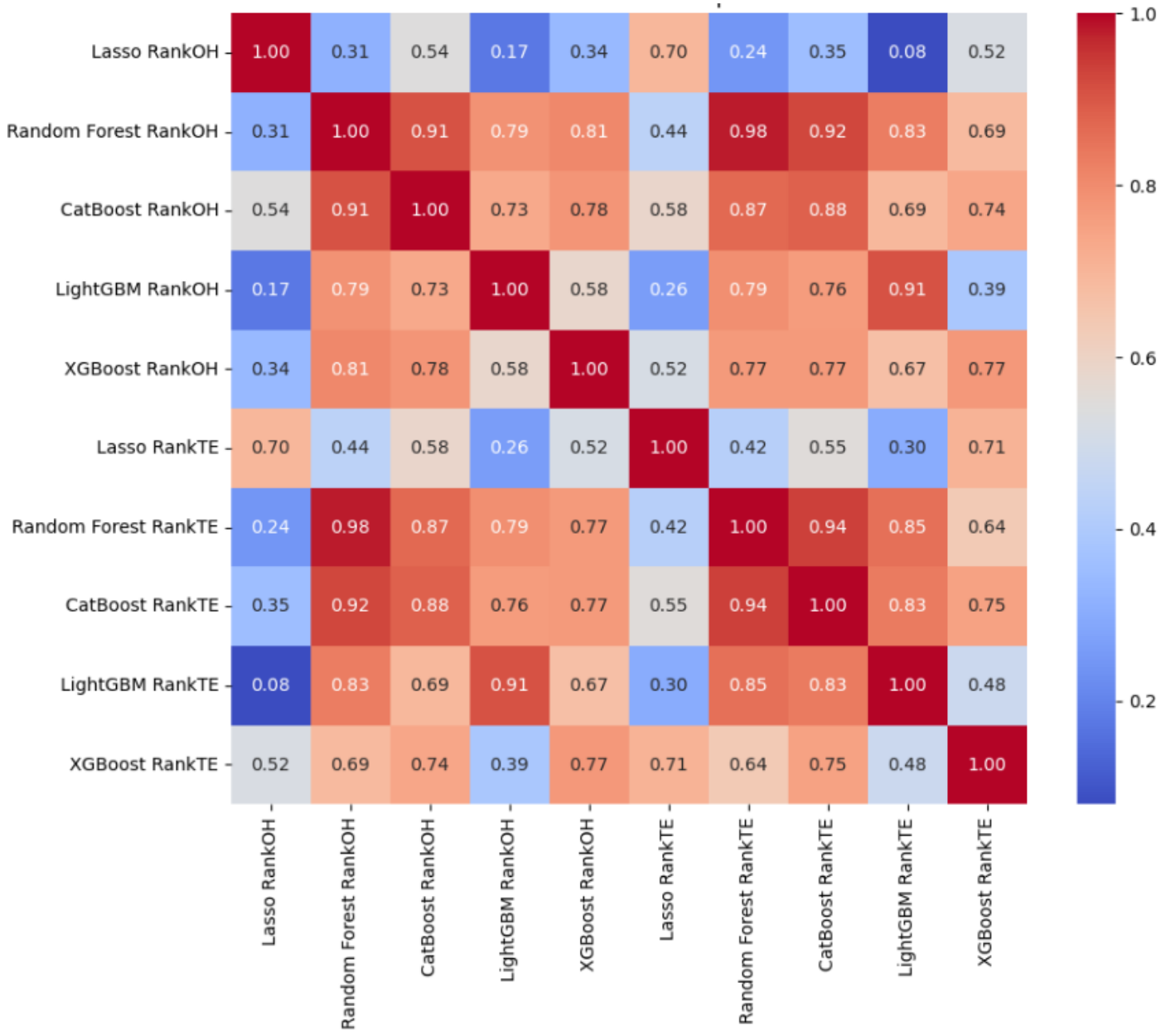
Model Selection & Feature Importance

For each model and encoding method, feature importances were ranked (for One-Hot sums of the importances of derived features).

Here is the correlation matrix of feature importance ranks.

It is clear that Lasso ranks features differently from other models that give good results.

Lasso is a linear regression model and gives poor results in the case of nonlinear dependencies.



Fine Tuning with Cross-Validation

Based on the initial launch of the models, we decided to fine-tune the models CatBoost and XGBoost with Cross-Validation.

Model	Encoding	Train MSE	Test MSE	Train R2	Test R2
CatBoost	built-in TE	486,8	675,5	0,715	0,604
XGBoost	One-Hot	449,9	699,2	0,737	0,591

Fine Tuning with Cross-Validation

Based on the initial launch of the models, we decided to fine-tune the models CatBoost and XGBoost with Cross-Validation.

Model	Encoding	Train MSE	Test MSE	Train R2	Test R2
CatBoost	built-in TE	486,8	675,5	0,715	0,604
XGBoost	One-Hot	449,9	699,2	0,737	0,591

Features can be divided into three groups: location-related, apartment-related, and hospitality-related.

Our models assigned importance to the features in these groups in completely different ways:

feature group	CatBoost importance	XGB importance
Apartmants	50,6	28,7
Location	22,0	66,9
Hospitality	27,4	4,6

feature	CatBoost importance
Room Type	12,50
Accomodates	10,88
Bedrooms	8,97
Property Type	7,00
Bathrooms	5,30
Guests Included	4,81
Postal Code	4,44
Distance_from_center	4,30
neighbourhood	4,09
First Review Years	3,53
Longitude	3,03
Host Since Years	2,77
Host Response Time	2,75
Last Review Years	2,26
Latitude	2,10
Min Nights	1,97
Reviews Log	1,96
Host Response Rate	1,83
Polarity_mean	1,70
Bin2-Bin2	1,49
Bin-Bin	1,39
Polarity_median	1,31
Overall Rating	1,30
Value Rating	1,29
Beds	1,22
Cleanliness Rating	1,09
Polarity_max	1,07
Neighborhood Group	0,92
Polarity_min	0,81
Is Superhost	0,59
Is Exact Location	0,27
Accuracy Rating	0,24
Instant Bookable	0,24
Checkin Rating	0,22
Location Rating	0,22
Communication Rating	0,14

feature	XGB importance
Bin-Bin	27,47
Bin2-Bin2	15,07
Postal Code	10,95
neighbourhood	9,55
Property Type	7,98
Room Type	6,92
Bedrooms	4,51
Bathrooms	3,65
Neighborhood Group	2,81
Accomodates	2,79
Guests Included	1,64
Host Response Time	1,52
Beds	1,17
Distance_from_center	0,53
Is Superhost	0,24
Polarity_mean	0,24
Host Since Years	0,21
Cleanliness Rating	0,19
Reviews Log	0,19
Min Nights	0,19
Last Review Years	0,18
Polarity_max	0,17
Value Rating	0,17
Location Rating	0,17
Latitude	0,16
Polarity_min	0,15
Longitude	0,15
Overall Rating	0,15
Is Exact Location	0,14
Instant Bookable	0,13
First Review Years	0,13
Polarity_median	0,11
Communication Rating	0,11
Host Response Rate	0,10
Checkin Rating	0,08
Accuracy Rating	0,08

Thank you for your attention