Data Science Challenge

Bayut & dubizzle Data Science Case Study V1

Nikita Parfenov

27.03.2022

Project task

Bayut & dubizzle Data Science Case Study V1

Description of the Data

The dataset shared with you is an anonymized dataset of different properties in Dubai. The dataset contains the size, number of bedrooms, number of bathrooms, neighbourhood name, and building name and the listing price for different properties in Dubai. You can find the data under **data_science_challenge_data.csv**

Assignment

1. Build a **Property Price Prediction Model**. Please use the Python programming language.
    a. Explore the dataset, and share with us any insights that you may find which can help you create the price evaluation tool. Please summarise your findings in terms of the relationship between the different features, the price, and feature importance.
    b. Build a model which predicts the listing price of the property based on the property's features
        i. Model Input: Features of property
        ii. Model Output: Predicted Price
    c. How do you evaluate the quality of your results?
    d. What are the possible shortcoming & extensions of your approach?

2. Build a **Property Price Valuation Tool**, which would take as input the features of a property and its listing price and determine whether the property is under-priced, fairly priced or overpriced.
    a. Implement a program to determine whether a property is underpriced, fairly priced or overpriced. This is your chance to show us the process that you would follow to solve this problem, and how you would model the data.
        i. Program Input: Property features and its price
        ii. Program Output: Whether the prediction is Underpriced, Fairly Priced, Overpriced
    b. How do you evaluate the quality of your results?
    c. What are the possible shortcoming & extensions of your approach?

Deliverables

Please share with us

- your program code
- a short presentation (10-15 slides) with the results of your work

within one week.

Project pipeline


.csv data

Exploratory Data Analysis

Baseline Model

Feature Generation

Feature Selection

Model Selection, hyperparams tunning

Property Price Valuation Tool

0. Input data was in .csv format ~3MB, with 67107 rows and 6 parameters.

1. EDA contains data “quick look” using statistics and graphs, looking at correlation between data and cleaning the data from nulls and outliers.

2. Baseline model built with initial data after filling nulls and outliers.

3. New features were generated relying on grouping and aggregating summarizing features. After that squares, square roots and logarithm of numerical features were added.

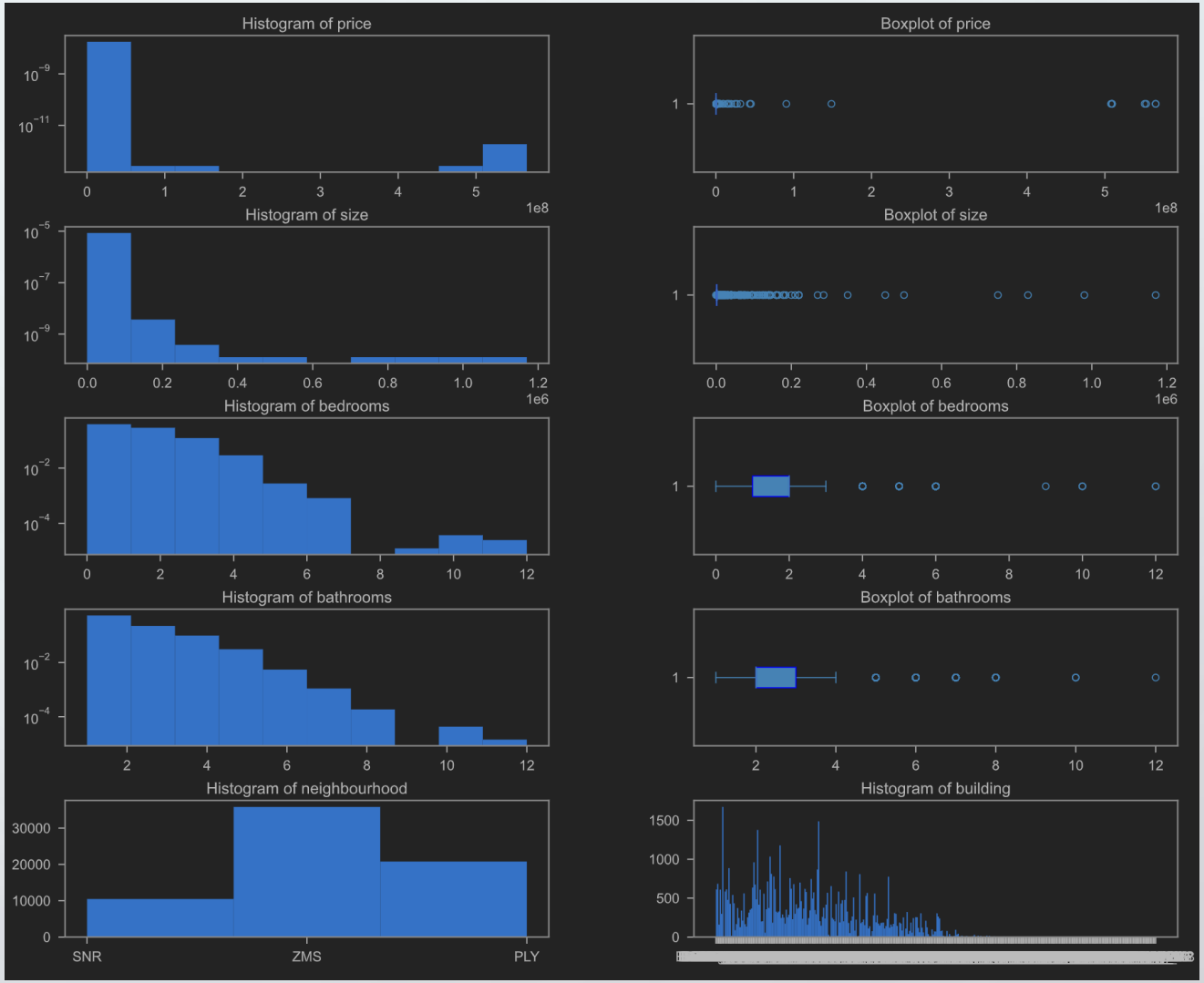
4. Feature selection was held by using permutation importance for linear type models and by shap for gradient boosting type models (the reason of two model use will be explained later in note on slide 10).

5. For auto selection of models and hyperparameters Optuna library was used.

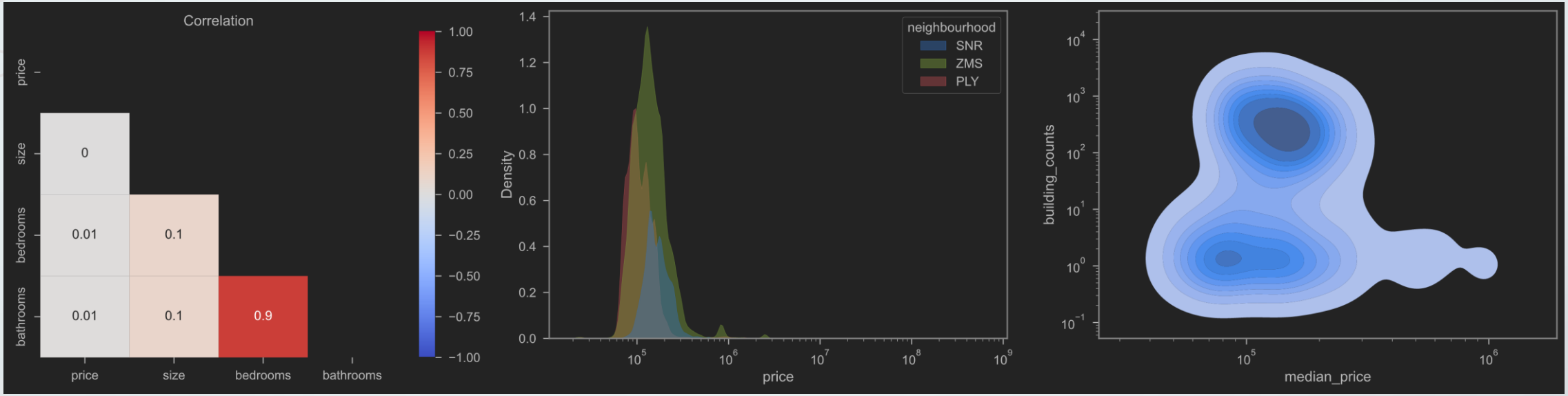
6. The degree of expensiveness was estimated by using standard deviation of trained data.

Exploratory Data Analysis – data ‘quick look’

- The data contains 67107 rows and 6 parameters.
- There 4 numerical and 2 categorical parameters.
- Histograms of numerical parameters show skewed distributions.
- There are some extremely high values in price. It could be connected with huge luxury flat or just outliers.
- There are abnormal size values equals to zero.
- Feature “building” is categorical and contains a lot of categories including both high and low frequent. It’s not good to leave such kind of feature, but it could be changed by numerical frequency.



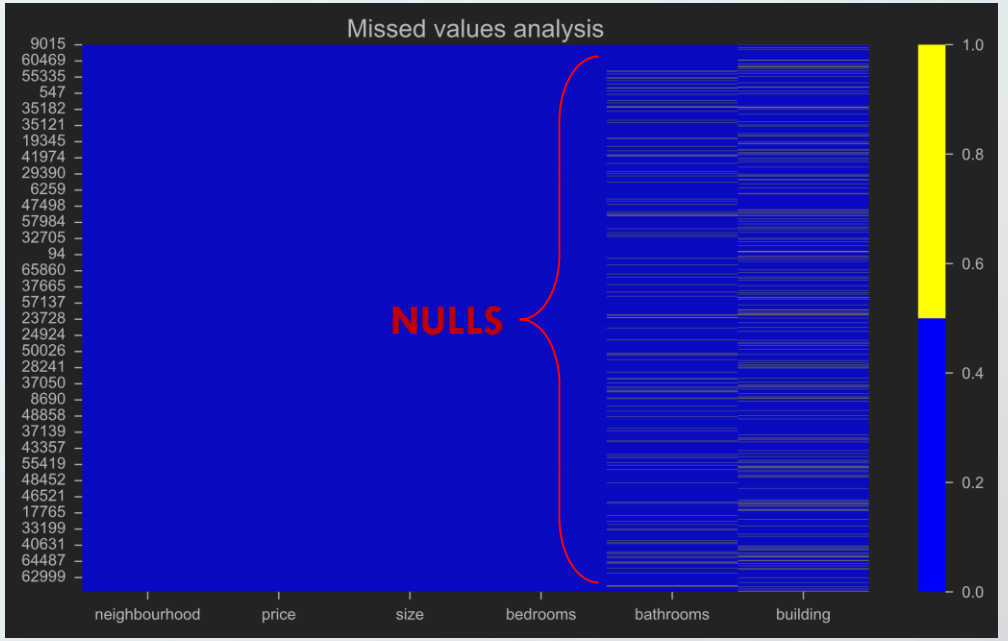
Exploratory Data Analysis – data correlations



- There is a high correlation between bathrooms and bedrooms (this idea will be used later).
- It looks very weird that price doesn't have a correlation with rooms and size. It could be connected with some outliers in price.
- Neighborhood is not a highly important feature but never the less it could help to predict price in SNR and PLY regions.
- The right graph depicts the grouping data by buildings and aggregation of building by counts and price by median value. Hence, this graph depicts the median price in buildings with different flat amount. The differentiation is very weak.

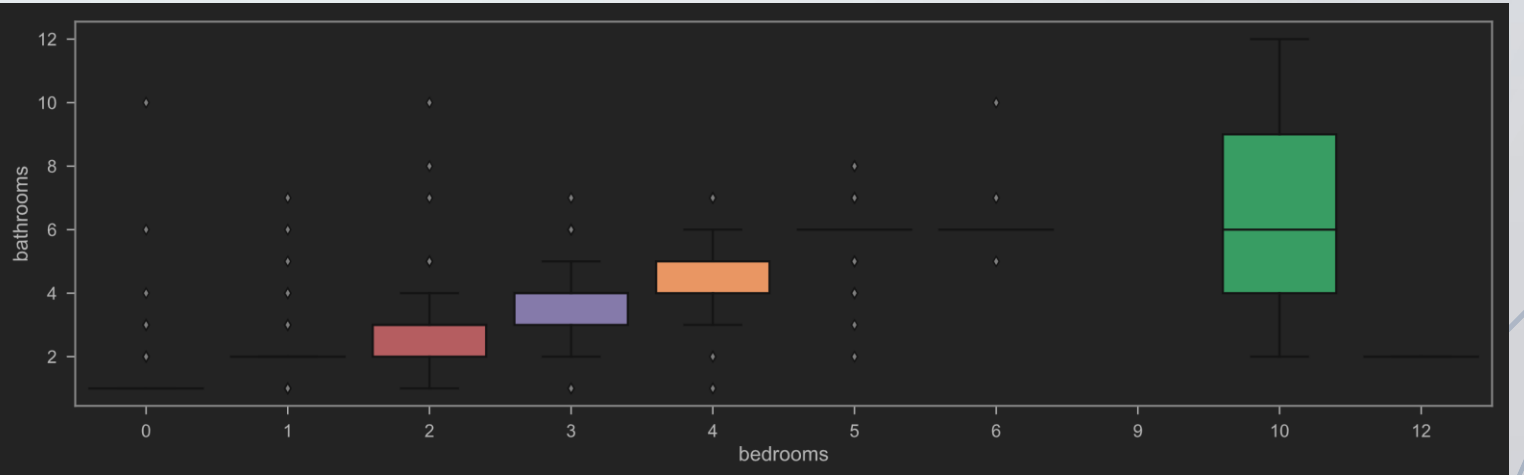
Exploratory Data Analysis – filling nulls

- As the analysis showed there is some nulls in bathrooms and building features.
- Heat map of missing values shows that there is no correlation between nulls of these two features.
- Bathrooms nulls have been filled relying on bedrooms data as median value of bathrooms for each bedrooms group. For bedrooms equals 9 wasn't any statistics thus it fill by nearest values 6 (as for 5, 6, 10 bedrooms).
- Building nulls has been filled by median building value grouped by neighborhood and bathrooms amount.



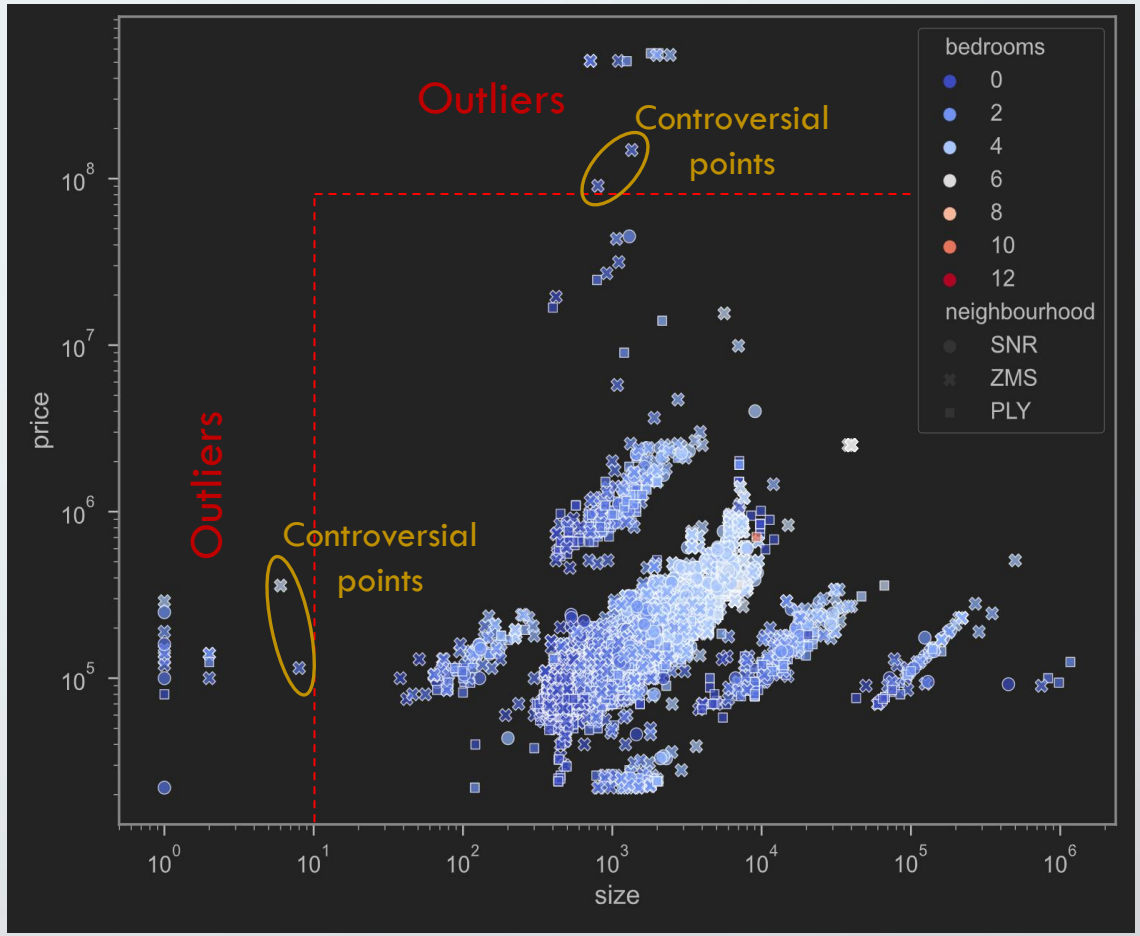

```
Column          Non-Null Count
------          --------------
neighbourhood   67107 non-null
price           67107 non-null
size            67107 non-null
bedrooms        67107 non-null
bathrooms       63657 non-null
building        61068 non-null
```

NULLS



Exploratory Data Analysis – “repair” outliers

- Size values have some extremely low value. It’s hard to imagine zero size or size less than 10 (assumed m2). Outliers were filled as median values for corresponding building.
- Price has some extremely high values but there are no reasons for such values (medium size and bedrooms). Outliers were filled as median values for corresponding building as in the case with size.

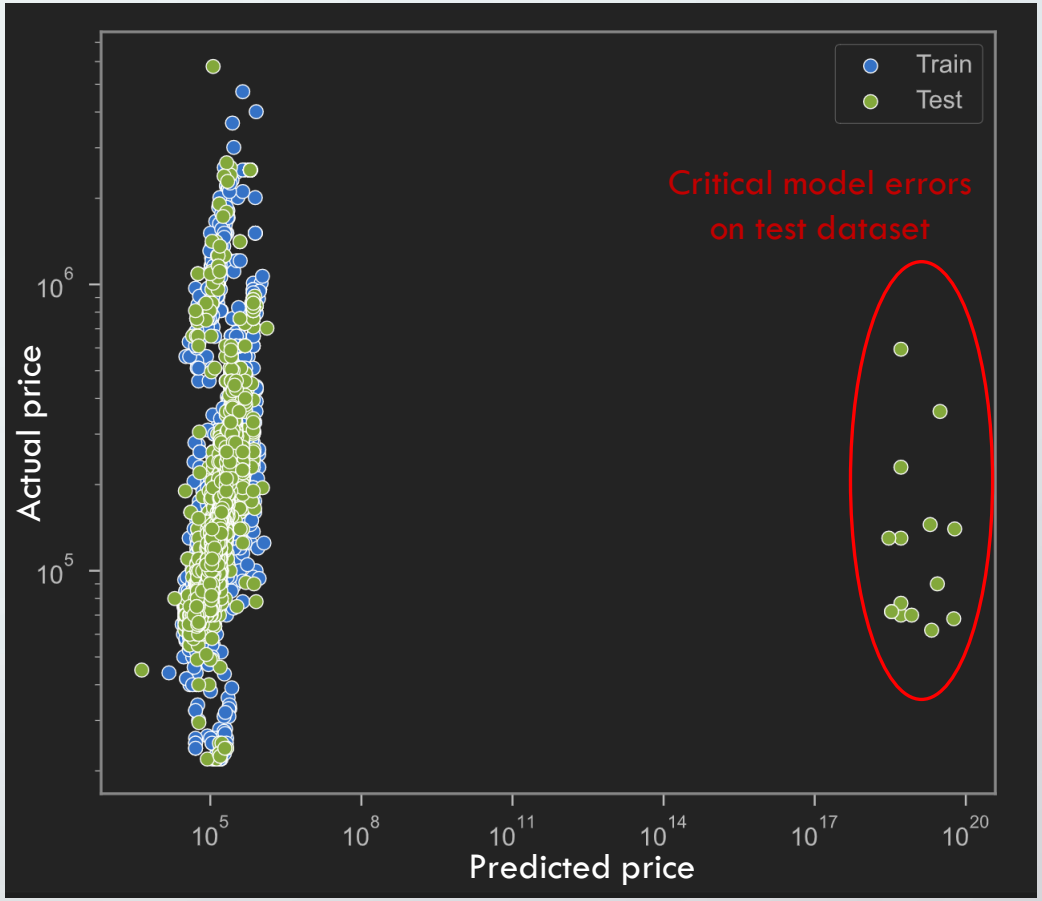


NOTE: For some controversial points the best way is to talk with experts of the market in order to ensure its outlier nature

Baseline estimation

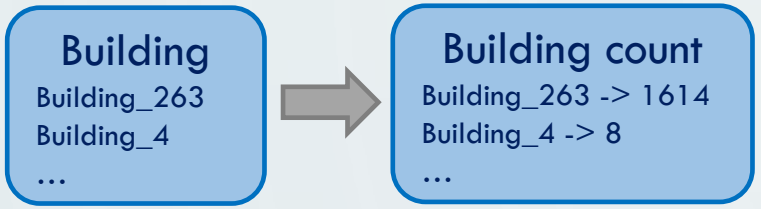
- For baseline model Linear Regression with 5-folds cross validation has been used. The model showed a very low R2 metric on train dataset ~0.35 and abnormal metric on test dataset connected with critical errors of the model.

Train: 0.3547808030826392, Test: -1.1968951460105131e+27
Train: 0.3462804866927746, Test: -1.0031344593718527e+28
Train: 0.3492939287668763, Test: -4.973288248379924e+26
Train: 0.3582387986677622, Test: -8.225591117497716e+26
Train: 0.3614425048075337, Test: -1.1177507657292753e+26

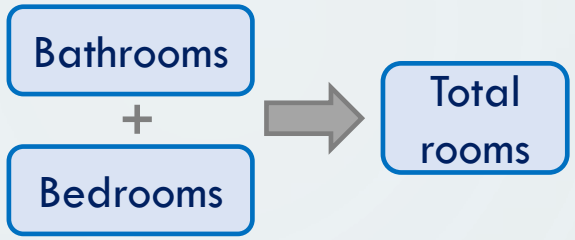


Feature generation – basic generation

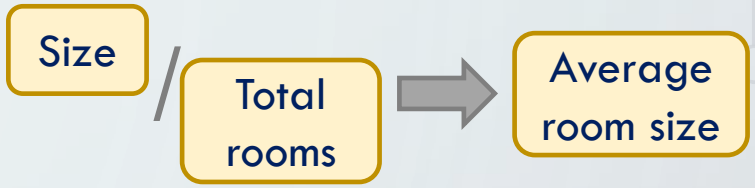
1. Building counts (from cat. feat to num. feat)



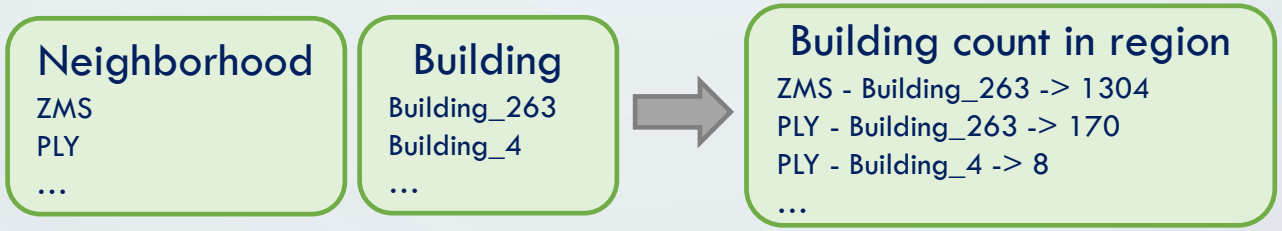
2. Total amount of rooms



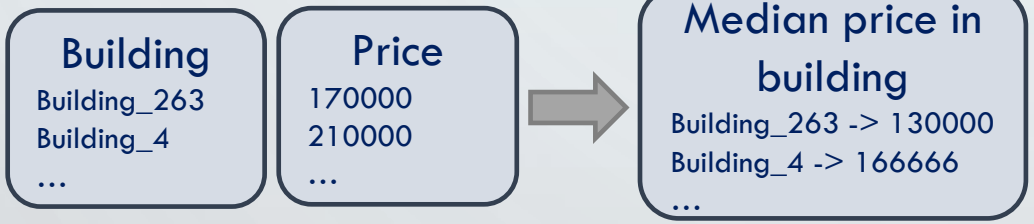
3. Average room size



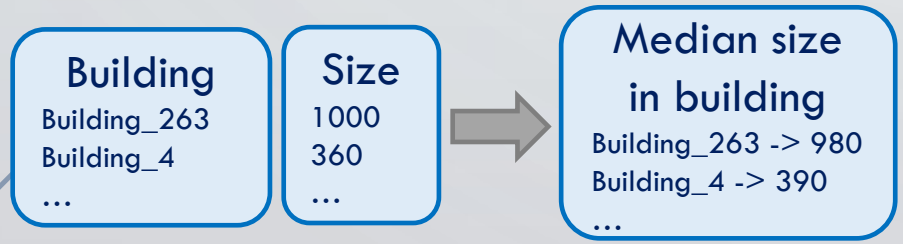
4. Building counts in neighborhoods



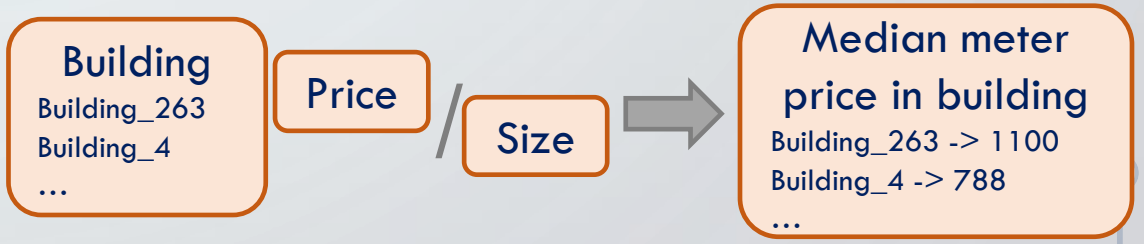
5. Median price in each building



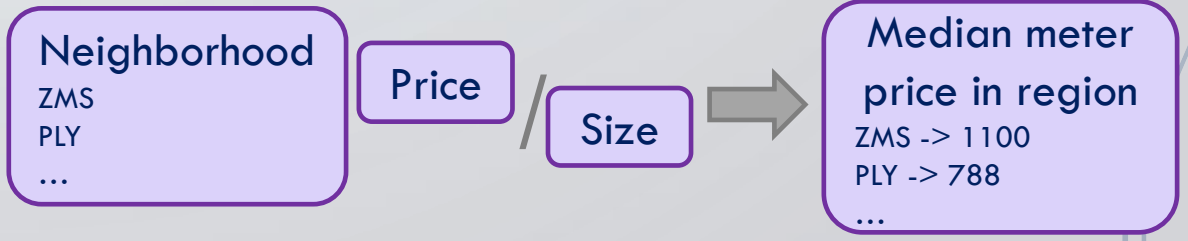
6. Median size in each building



7. Price per squared meter in buildings

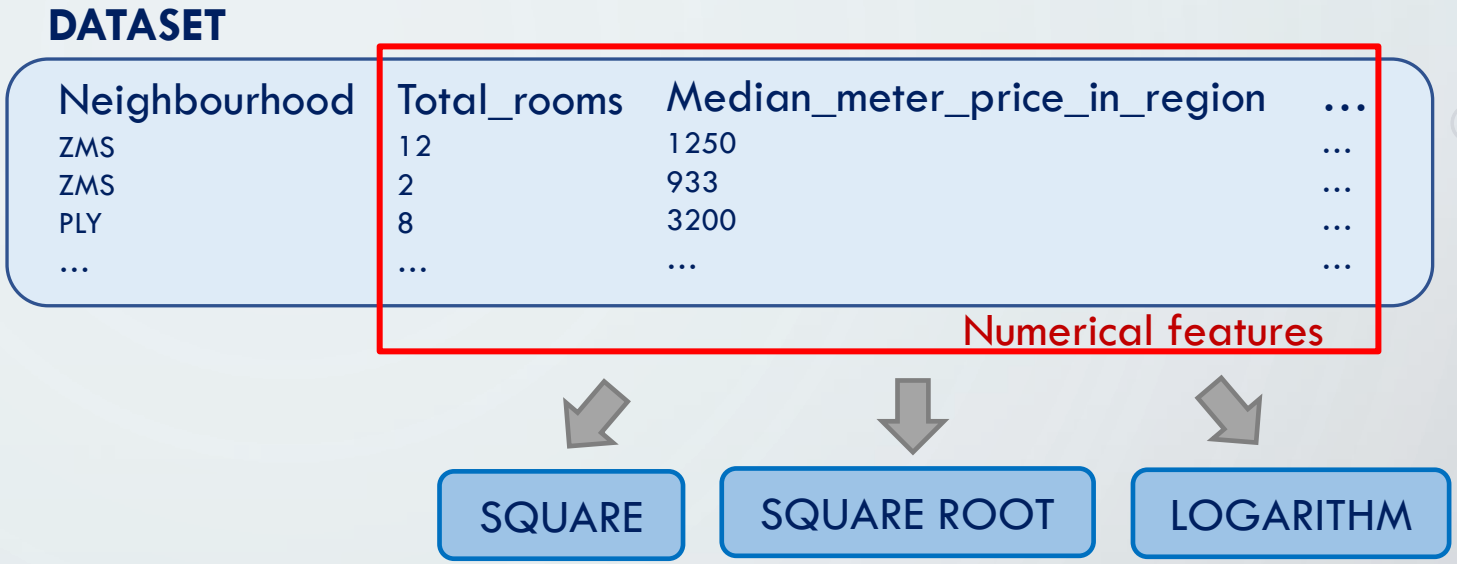


8. Price per squared meter in regions



Feature generation – adding non-linearity

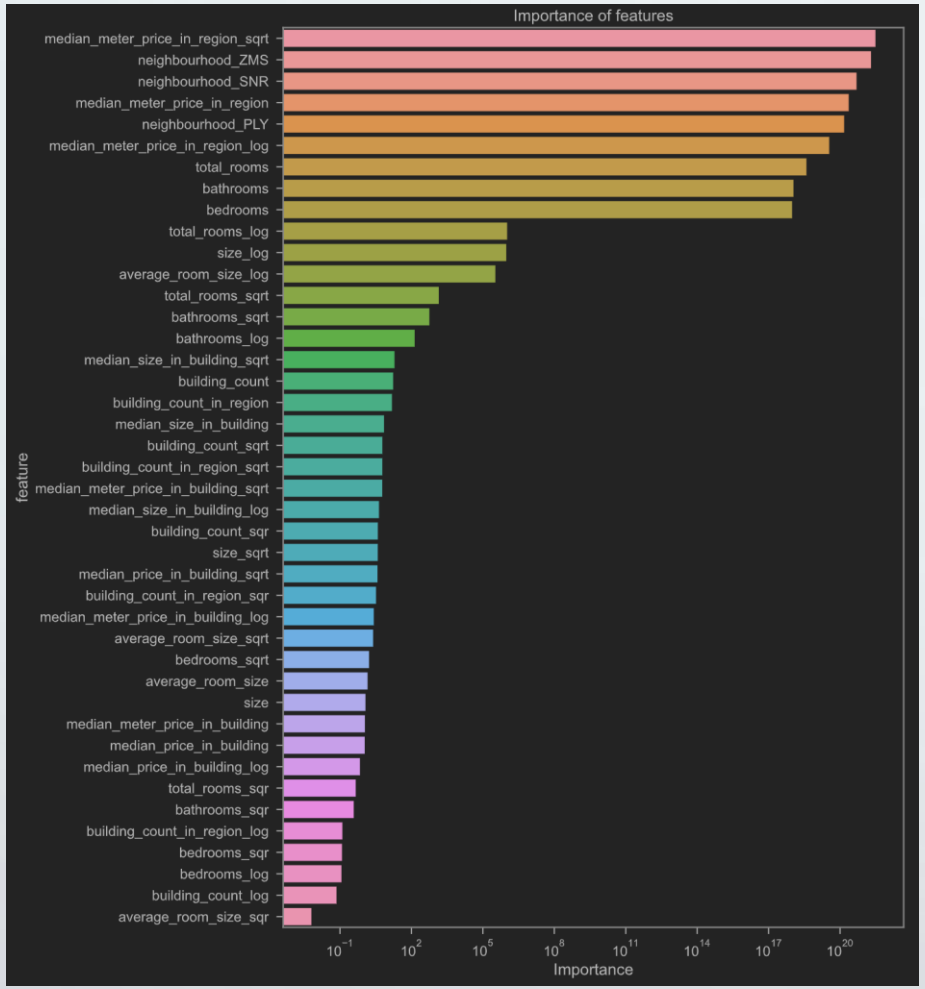
- In order to add non-linearity to models all numerical data has been transformed to squares (excluding features with very high values), square roots and logarithms



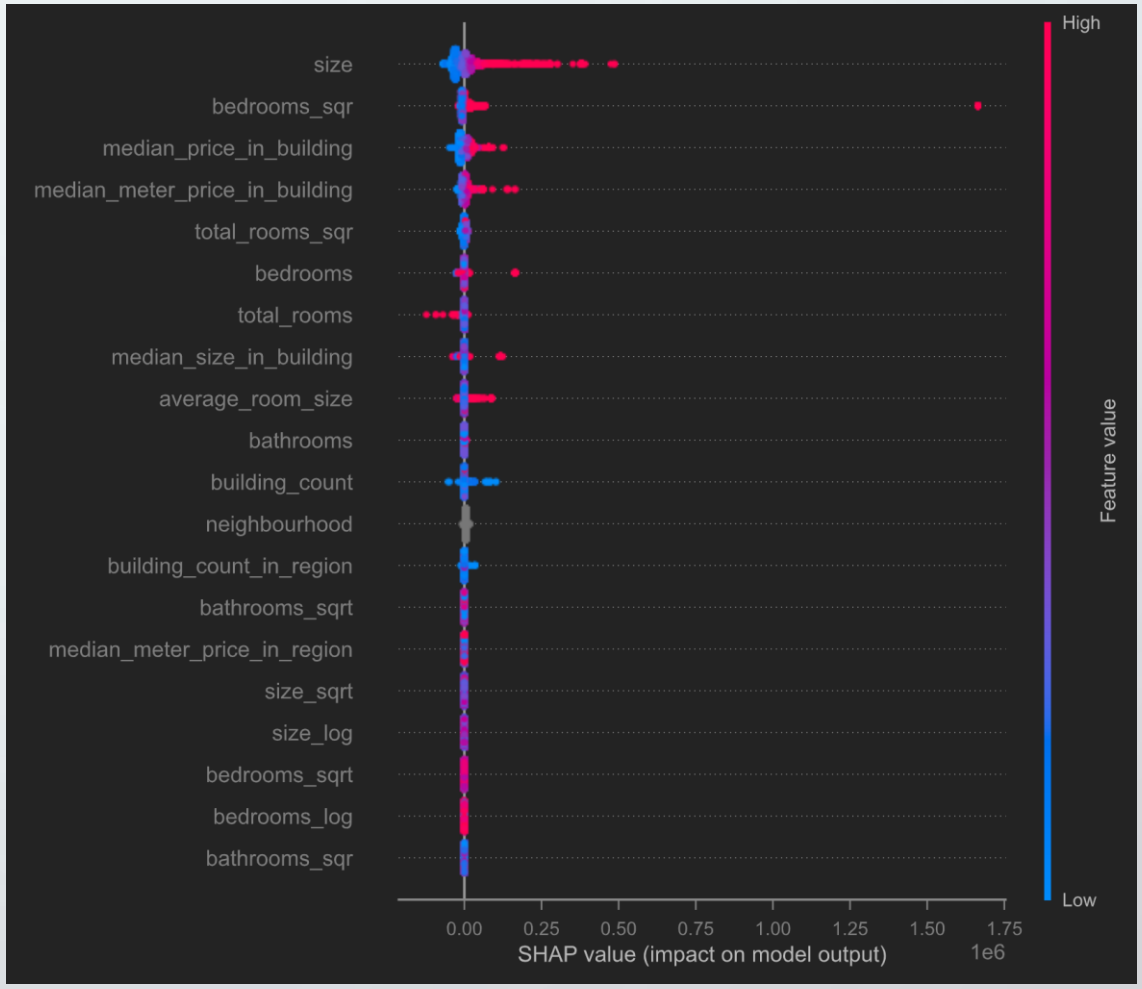
NOTE: In further slides two types of models will be used: linear model (LR) and gradient boosting tree (GB). Boosting type models almost always showed the better quality due to their non-linearity but they are restricted by trained data values range. It means that appearance of a new value (in any feature) outside the previous range will cause constant value prediction. It could be critical for price prediction task (for example, appears a new building or new amount of rooms etc.). Linear models don't have such restriction, that's why there were two type of models has been considered. GB model is good for price prediction on the data like in existing data set and LR model is for a new kind of data.

Feature selection

Linear Model (permutation importance)



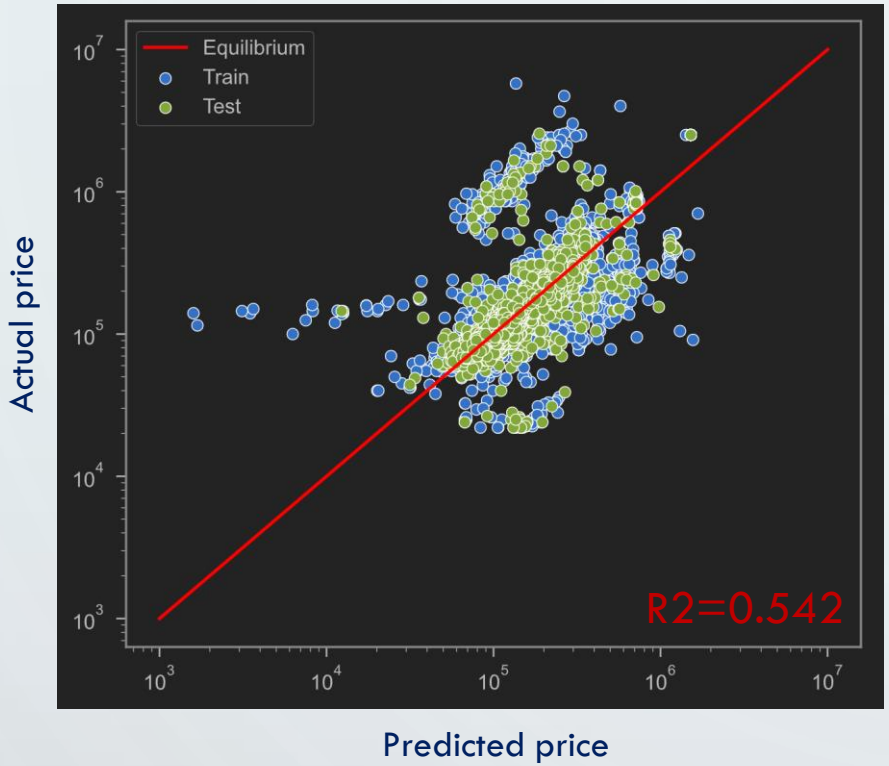
Gradient Boosting Model (shaply)



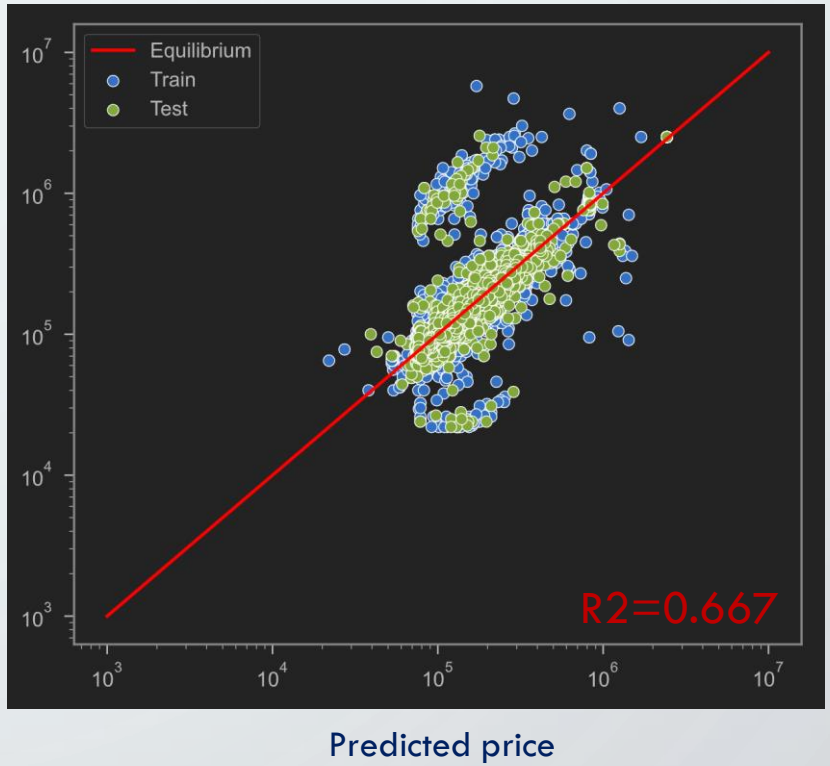
- For linear model all features are important. Never the less, there are some features with very low importance and could be removed to make the model easier.
- For gradient boosting model only 13 features are important, other has been removed

Model selection and hyperparameters tunning

Linear Model (ElasticNet)



Gradient Boosting Model (LightGBM)



- Linear auto model selection was held for LinearRegression, Lasso, Ridge and ElasticNet. The best linear model was ElasticNet model with L1 and L2 regularization. R2 on 20% of test data is about 0.542
- Boosting auto model selection was held for LightGBM, XGBosst and CatBoost. The best gradient boosting model was LightGBM. R2 on 20% of test data is about 0.667

NOTE: The quality of models could be improved by additional feature generation.

Expensiveness evaluation tool

To estimate the expensiveness the standard deviation (ϭ) from average (predicted) cost was used. The suggested method compares a current price with an average (predicted) price and summarize whether current price in range average (predicted) price +/- one standard deviation or not:

- FAIRLY PRICED:

pred_price - ϭ <= price <= pred_price + ϭ

- UNDERPRICED:

price < pred_price – ϭ

- OVERPRICED:

price > pred_price + ϭ