Sberbank House Price Prediction

Hossein Namazian

July 2017

Introduction:

Housing costs demand a significant investment from both consumers and developers. And when it comes to planning a budget—whether personal or corporate—the last thing anyone needs is uncertainty about one of their biggest expenses. Sberbank, Russia's oldest and largest bank, helps their customers by making predictions about realty prices so renters, developers, and lenders are more confident when they sign a lease or purchase a building.

Although the housing market is relatively stable in Russia, the country's volatile economy makes forecasting prices as a function of apartment characteristics a unique challenge. Complex interactions between housing features such as number of bedrooms and location are enough to make pricing predictions complicated. Adding an unstable economy to the mix means Sberbank and their customers need more than simple regression models in their arsenal.

Acquiring Data:

The Sberbank Russian Housing Market dataset has recently been released by Sber Bank, the oldest and the largest Russia's bank (https://www.kaggle.com/c/sberbank-russian-housing-market/data). The datasets was released for kagglers to propose some models to forecast the house prices. The train dataset contains more than 30000 transactions from August 2011 to June 2015 in the Russian House Market for some years, and the macro dataset provides the daily economic parameters in Russia. A test data also published to evaluate the kagglers' models. As the real prices in the test file is hidden by competition holders, therefor for the purpose of this capstone project, only the train and macro data is used.

The train data contains 30471 rows with 391 columns for each data point, which has been collected from August 2011 to June 2015. The dataset has richened with different types of features such as house characteristics, demographics, cultural information, country economical information, etc. The macro data includes 100 economical parameters in Russia for 2484 successive days.

Data Wrangling:

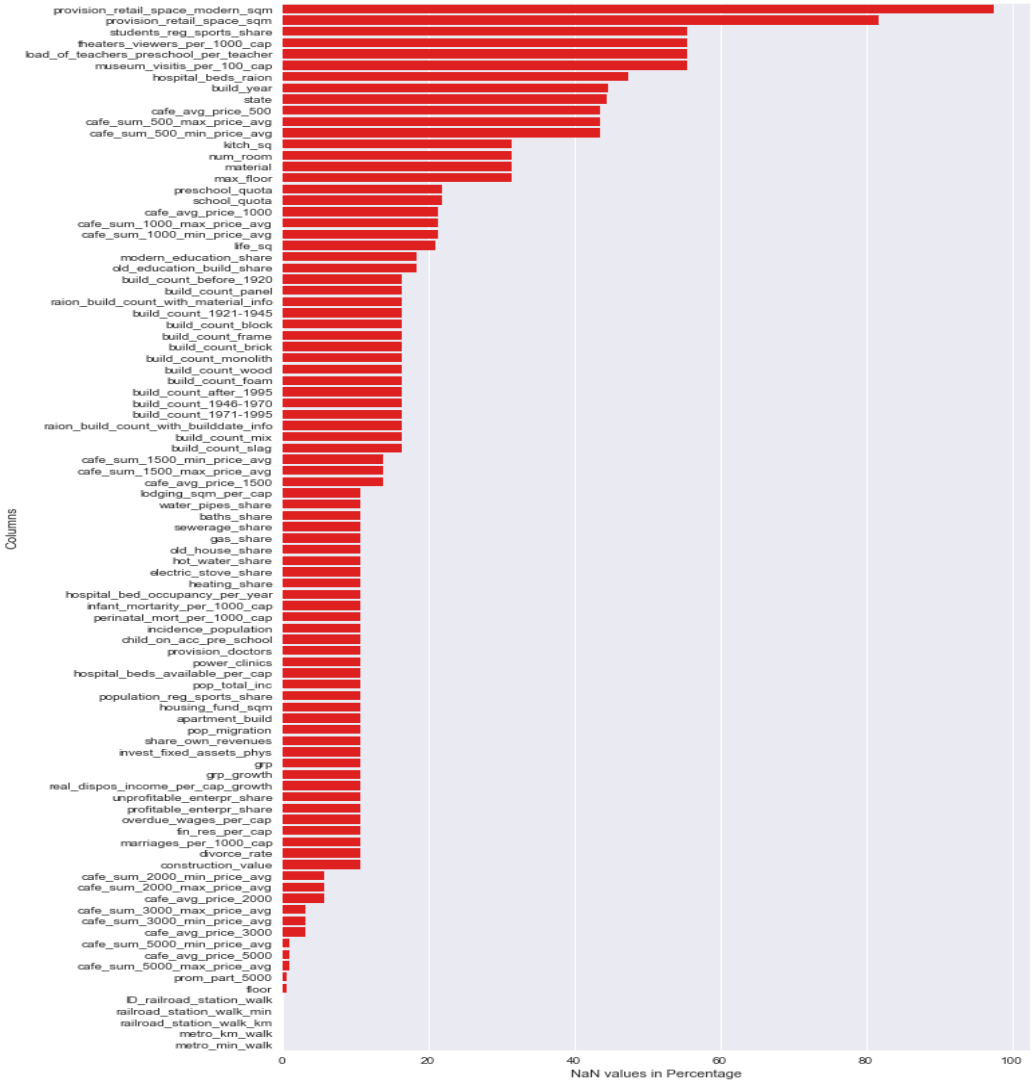
The dataset is a real big data with hundreds of columns and thousands of rows. First of all, it's very beneficial to classify all the features to better understand about the data and the relations

between different columns. I divided the columns of the train data into 13 classes and inserted the related features in each one.

The bigger data provides the more information helps learning better predictive models, on the other hand, the more concerns arise. Here we go through the data finding missing values and seeking outlier and logically wrong values such as bad relationships within each feature and between different features. For each of these cases there would be different strategies to deal with.

Number of missing data

The figure shows the missing value proportions in each column. It could be seen that two features of dataset contain more than 80% missing values and 10 columns includes more than 40%. Overall, there is 93 columns with at least one missing value.



House characteristic features:

The House characteristics might be considered as the most important factor to determine the house price in region. The dataset includes different related features:

- **full_sq**: total area in square meters, including loggias, balconies and other non-residential areas
- **life_sq**: living area in square meters, excluding loggias, balconies and other non-residential areas
- **floor**: for apartments, floor of the building
- **max_floor**: number of floors in the building
- **material**: wall material
- **build_year**: year built
- **num_room**: number of living rooms
- **kitch_sq**: kitchen area
- **state**: apartment condition
- **product_type**: owner-occupier purchase or investment
- **sub_area**: name of the district

The Investigation must be done on features separately and relatively to see some difficulties in the data, missing values, outliers and wrong relations.

Total Area

Some odd things could be seen in the **full_sq** column:

- There are 2 houses with 0 total area.
- There are 26 houses with less than 2.5 square meters total area.
- There is only 1 house with more than 1000 square meters total area.

Total Area vs. House Price

From the plot we can see a data point with more than 5000 square meters at very low price. That could be considered as an outlier.

Life Area vs. House Price

From the plot we can see a data point with more than 5000 square meters, life area at very low price. That could be considered as an outlier.

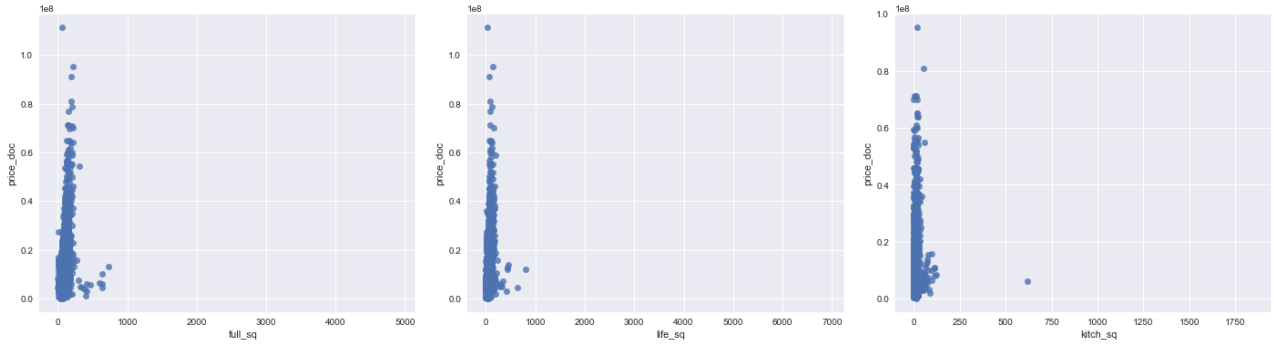
Kitchen Area

Some odd things could be seen in the kitch_sq column:

- There are 1381 houses with 0 total life area.
- The maximum kitchen area is 2014 square meters.
- There are 2 houses with more than 2000 square meters and 4 houses with more than 1000 square meters and 5 houses with 500 square meters kitchen area.
- For indices of [10368, 13117, 21415], it seems that the kitch_sq is in fact the build_year.
- In [11520] the kitch_sq is abnormally large, even much larger than life_sq and full_sq, so it's better to drop this row or alternatively replace the kitch_sq with NaN value.

Kitchen Area vs. House Price

From the plot we can see a data point with more than 500 square meters and 4 data points with around 2000 square meters at very low prices. That could be considered as an outlier.



Relations between Total Area, Life Area and Kitchen Area

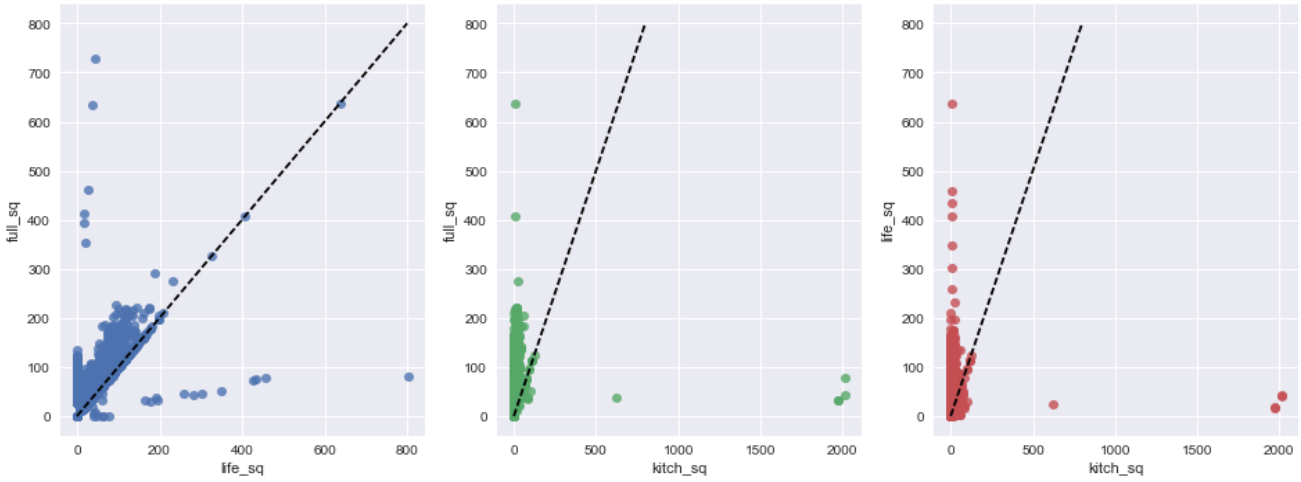
There are some simple relationships between these variables anticipated in the dataset:

- full_sq > life_sq
- full_sq > kitch_sq
- life_sq > kitch_sq (for isolated houses)

However, it could be seen:

- There are 37 data with life_sq > full_sq
- There are 12 data with kitch_sq > full_sq
- There are 56 data with kitch_sq > life_sq

There is a possibility that in these cases the employees had displaced these values. for example, one might be written the house life_sq in the full_sq column.

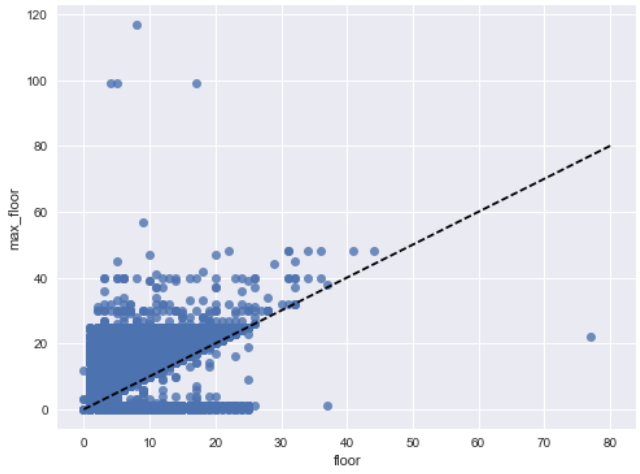


Number of Rooms

- There are 9 houses with more than 7 rooms and 18 houses with more than 6 rooms.
- For houses with more than 6 rooms, mostly the full_sqs are very small in comparison with number of rooms. At least we can put a serious suspicious for those values which have full_sq<100.

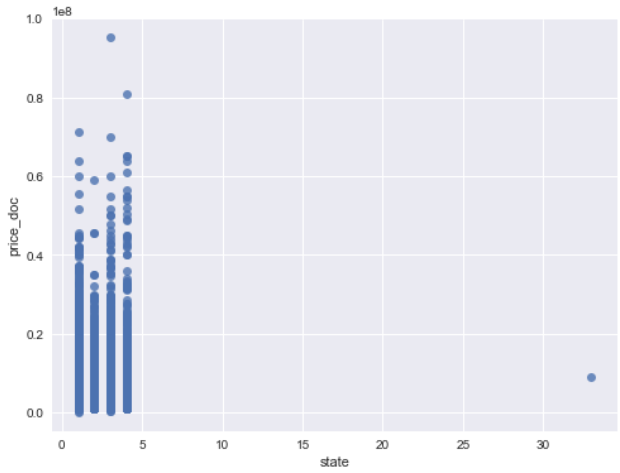
Maximum Floors vs. Floor

There is another logical relationship between house characteristics that must be held in the dataset, in which max_floor > floor. Unfortunately, there are 1493 data violate this relation.



State:

The state could get integer values of 1-4, therefor the state = 36 is not acceptable. The row of data containing this value either must be dropped out or considered as missing value.



Build year:

There are some values of build_year that have big differences to others. Some of them can be dropped out with no concern, however some other must be modified because they include a large proportion of dataset.

- The most important values are the ones with the build_year of 0 and 1, as the dataset includes 898 rows with these values. Fortunately, there is a reasonable assumption to take these values as the house' age instead of its build_year.
- The build_year=20052009 can be considered as a range of 2005-2009 and alternatively replaced with the mid-range.
- Other values including build_year = [3, 20, 71, 215, 4965] can be dropped out or replaced with NaN.

Handling Categorical data

There are 19 categorical columns in the dataset. Most of the machine learning algorithms are not compatible with categorical data. There are different ways to deal with this difficulty:

1- Label Encoding: Encode labels with value between 0 and n_classes-1.

2- One-Hot Encoding: Encode categorical integer features using a one-hot aka one-of-K scheme. This estimator transforms each categorical feature with m possible values into m binary features, with only one active.

Handling Missing data

After all the dataset has a lot of missing values and also we might add more to it. A lot of different strategies are accessible to handle the missing values and outliers:

1- drop-out missing values: When we have a very few outliers or missing values it could be acceptable, but in most of the cases that's a very bad idea.

2- Independent Imputation of columns: We can fill missing values in each column with a function (example; mean, median) of that column independently. It does very fast but not consider the correlations with other variables.

3- Regression Imputation: We can apply some Regression models such as Linear or knn Regression to estimate the missing values with respect to correlated features. It's more precise but very slow method.

4- Decision Tree based Algorithm: Decision tree has the potential to handle with missing values. The best Machine Learning algorithm to do this is the Extreme Gradient boosting which has been recently published. xgboost is a package of libraries that implemented this algorithm.

Feature Engineering

A very effective way to boost up the model performance is to use the art of Feature Engineering. That's to some extent heuristic, but it worth considering. Here I extended the timestamp column into different variables, such as, year, month, quarter, etc. This is because the house price has a periodic characteristic with respect to month, quarter, year, etc. Therefore, these features could guide the model to better learned.

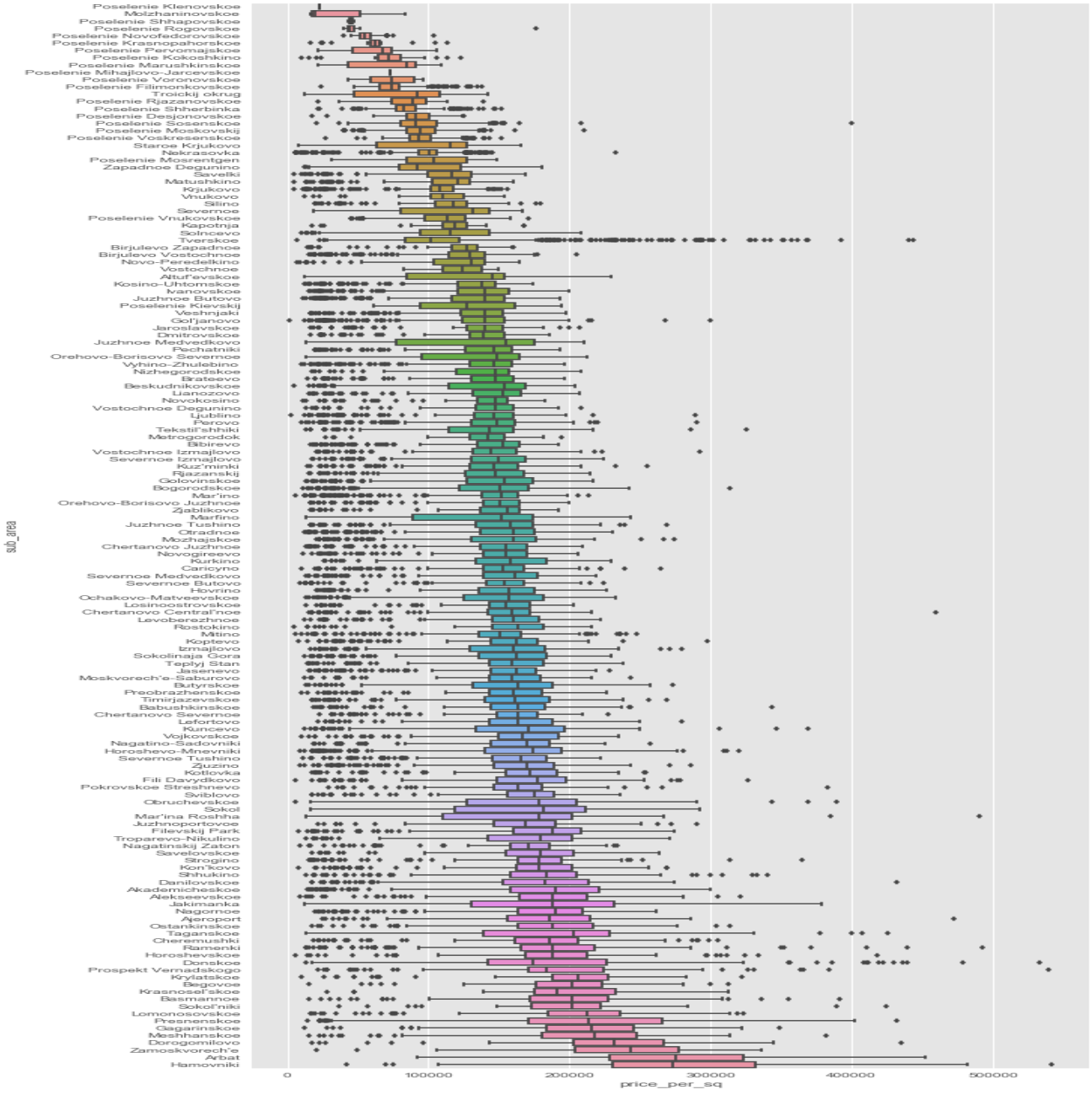
Another important feature that could be extracted from the dataset is the building age. It's very effective factor to determine the house price. This feature is simply computed with the subtraction of transaction year and build year.

Also It could be useful to add some relative features, such as the ratio of full_sq and life_sq, kitch_sq and life_sq and life_sq and num_room.

Exploratory Data Analysis

Region

Maybe the first and major factor in determining a house price is the region located in. In the figure below that could be seen that, the average per squared of houses encompasses a wide range from 20000 RUB in **Poselenie Klenovekoe** to more than 250000 RUB in **Hamovniki**. Notice that here per squared price (total price per total house area) is used instead of total house price to not considering the effect of total house area.



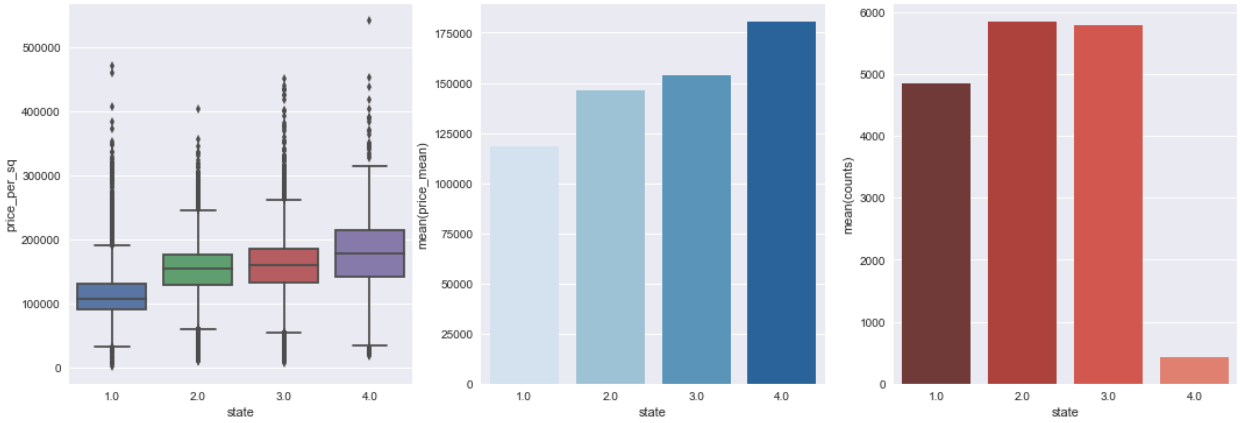
House Characteristics

State

Naturally that's expected for houses with better condition to have higher prices. That could be clearly seen in the figure below. The average per squared house price for the best conditioned apartment is more than 175000 RUB, however this value for worst stated house is less than 125000 RUB.

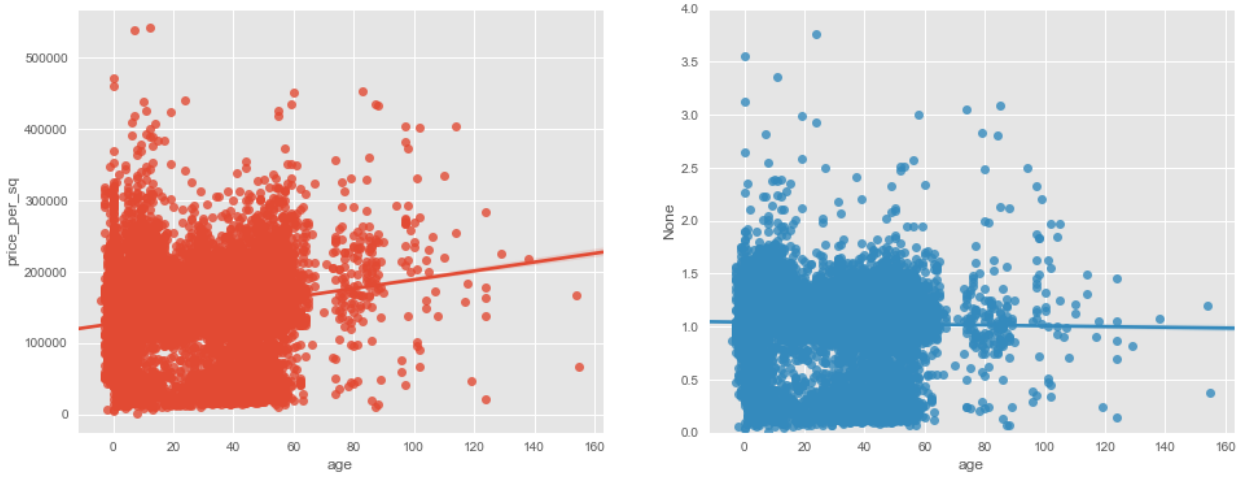
There is another thing that must be noted in the figure that, the number of houses that dealt is much lower than other types. Despite more 5500 house are dealt for house with the states of 1-3,

however, less than 500 transactions are recorded for state-4 houses. That might be related to their much higher costs in comparison with the economic situation of people.



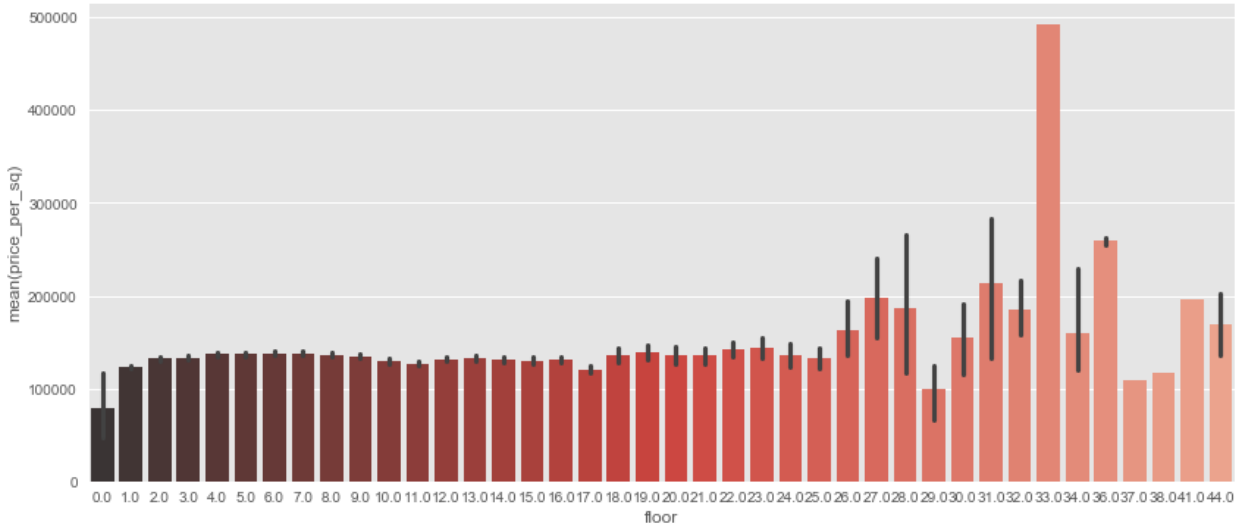
AGE

We expect for older houses to have lower per squared house. However, if they are located in higher level regions, then this assumption is not true. From the figures, the regression line shows that generally, the older houses have higher prices. On the other hand, we can divide each per squared house price to the average per squared prices of the houses in that region, and we name it the **region price ratio**. This parameter tries to compensate the effect of the region in house price. Now the second figure shows that the region price ratio is not effected by the age of the houses.



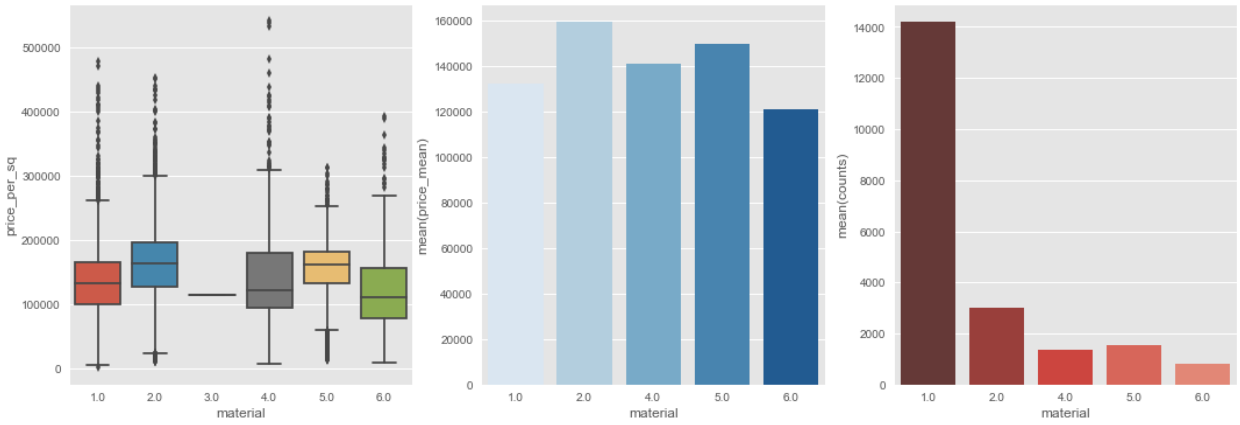
Floor

In general view we can state that the houses in higher floors are more expensive by average. This is what many times heard in real-estate.



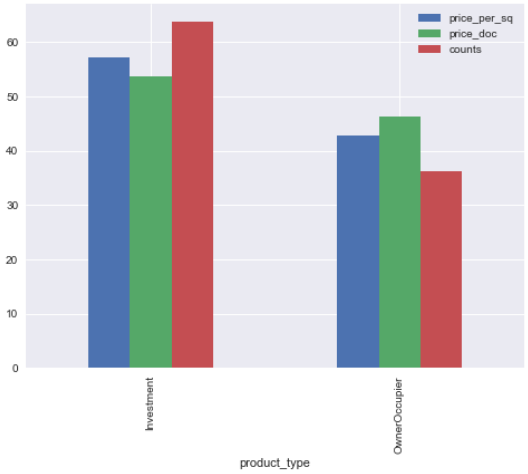
Wall Material

There is no enough information on the meaning of different Wall Material's type. However we can see some differences between them. The most notable thing that could be seen is that, most of the houses (more than 50%) are made by the wall material type-1.

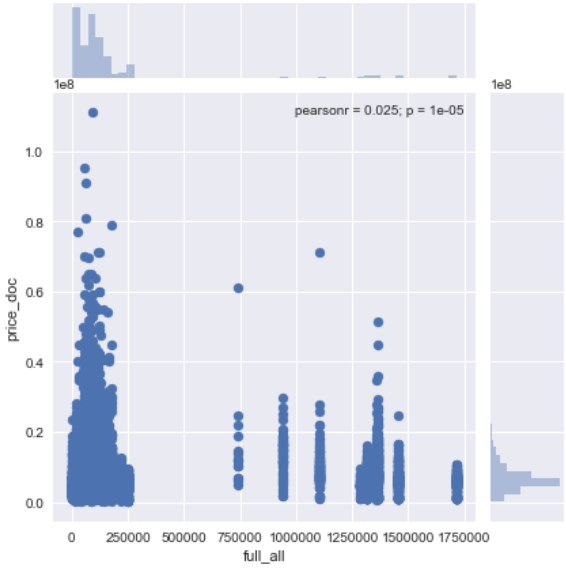


Product type

On the view point of **Product type** (***Investment*** or ***Owner Occupier***), it's concluded that, the investment type houses are more expensive (15% more) and dealt more (30% more).

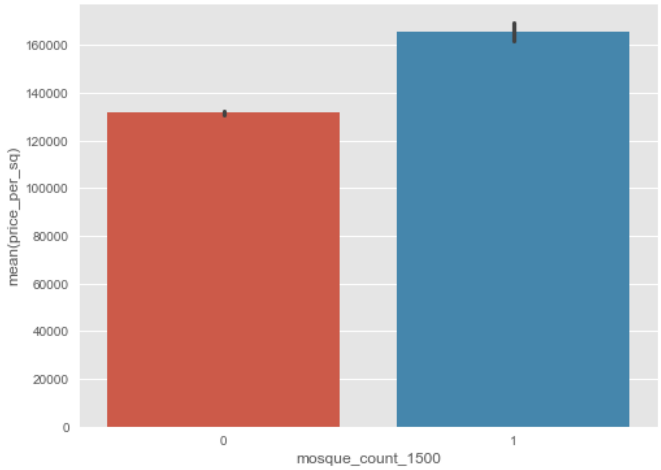


Demographic

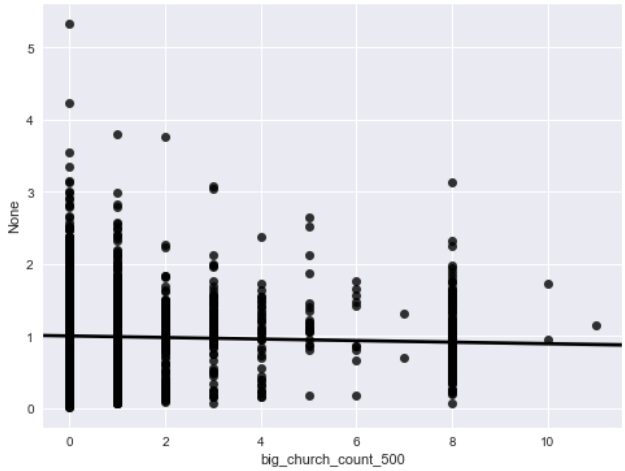


Culture

From the figure below, maybe it's better to say the mosques are built in higher prices region by average. Also, it could be said that the houses nearer to mosques are more valuable averagely.



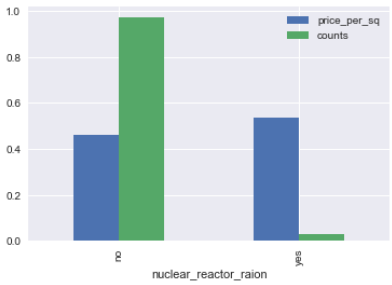
Being near the churches has not a notable proficiency, but also has a slightly contrary effect.

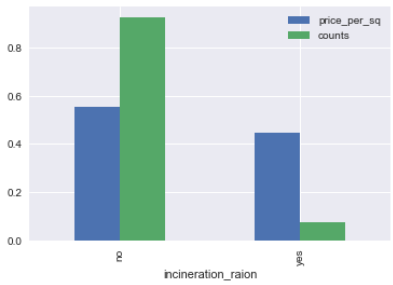


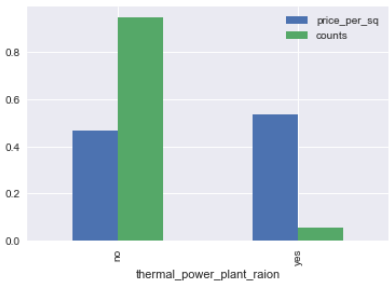
Industry

Nuclear Reactor

Although the figure shows about 10% higher prices for houses nearer to Nuclear Reactor, Thermal Power plant, but we must notice that these are less than 10% of all the cases.

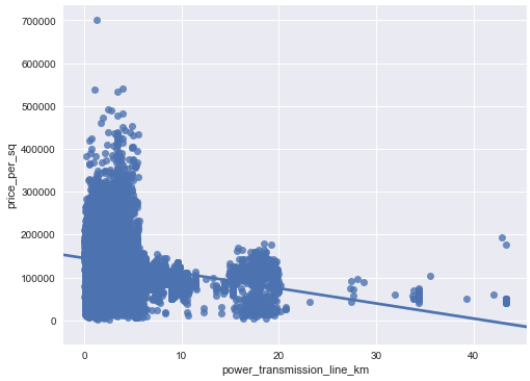






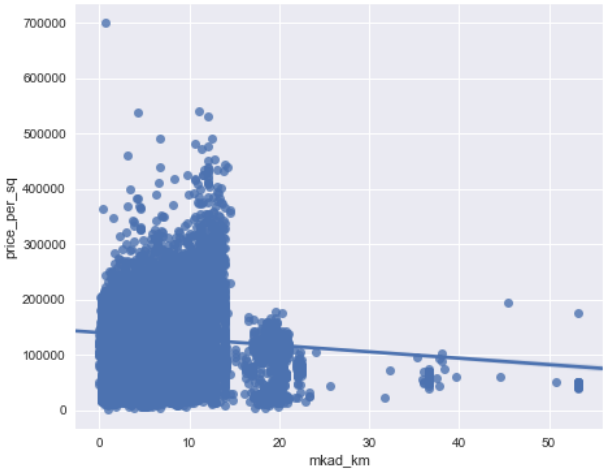
Power Transmission Line

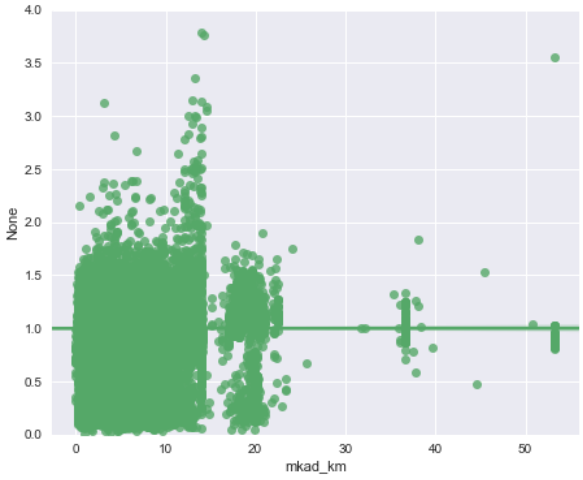
The regression line in the figure below has a negative slope that means, the closer to the Power Transmission Lines the lower house prices.

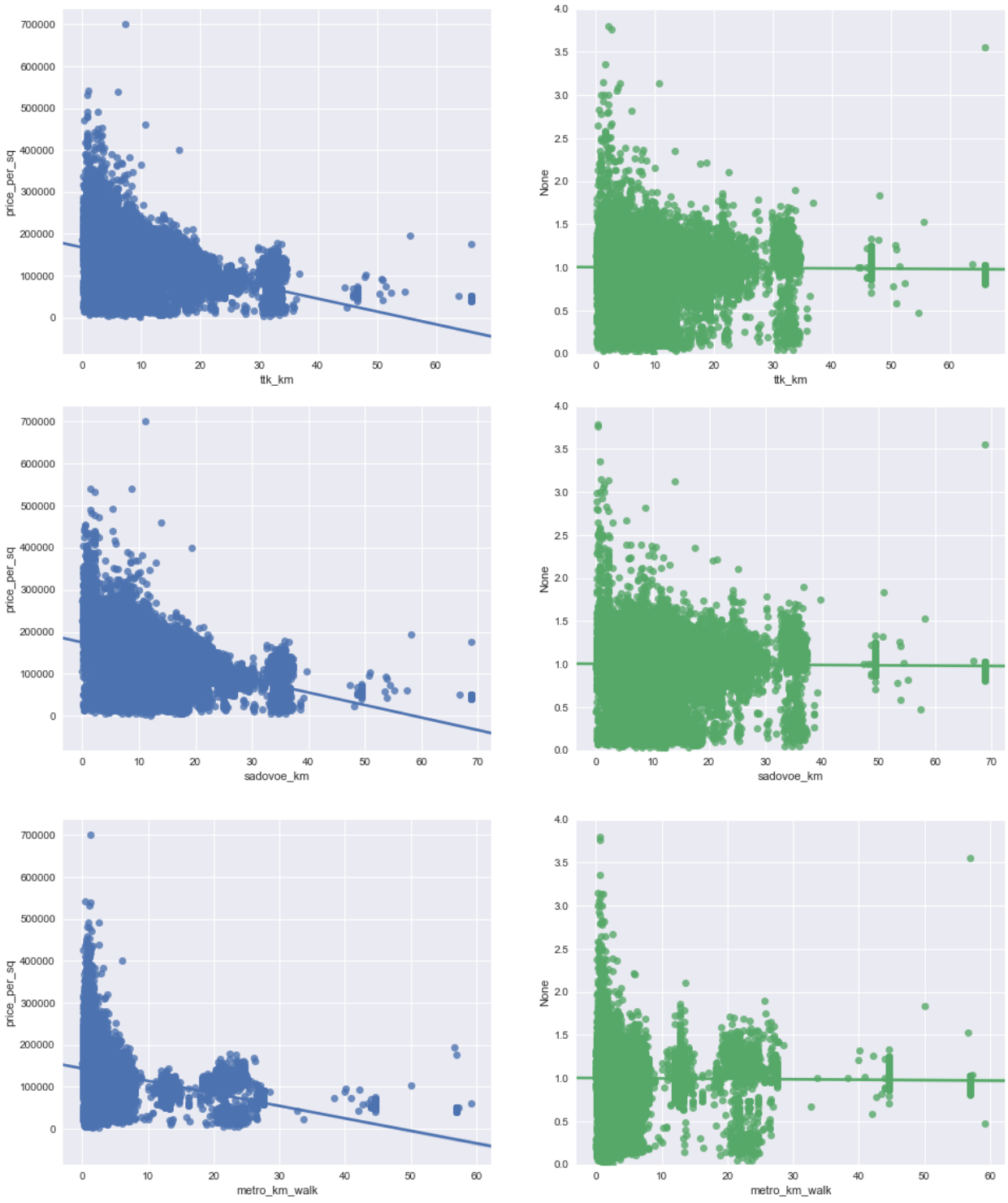


Transportation

In general, there is an obvious effect that being near the Transportation facilities is benefit. It could be seen that the houses closer to TTRs, Subways, etc. are more expensive. These are shown in the figures below.

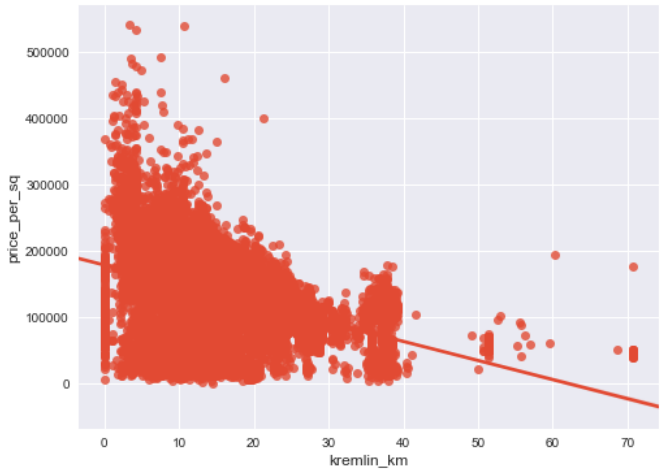






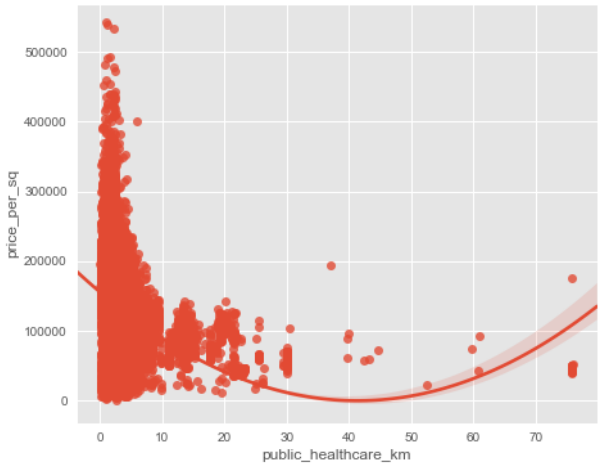
Kremlin

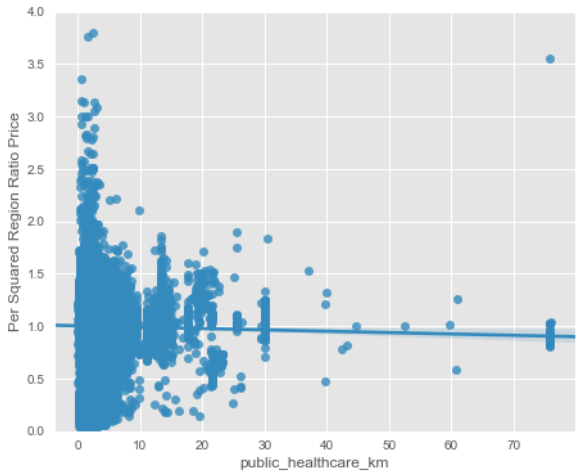
As seen in the figure, with respect to the negative slope of the regression line, the houses closer to the city center are more expensive.



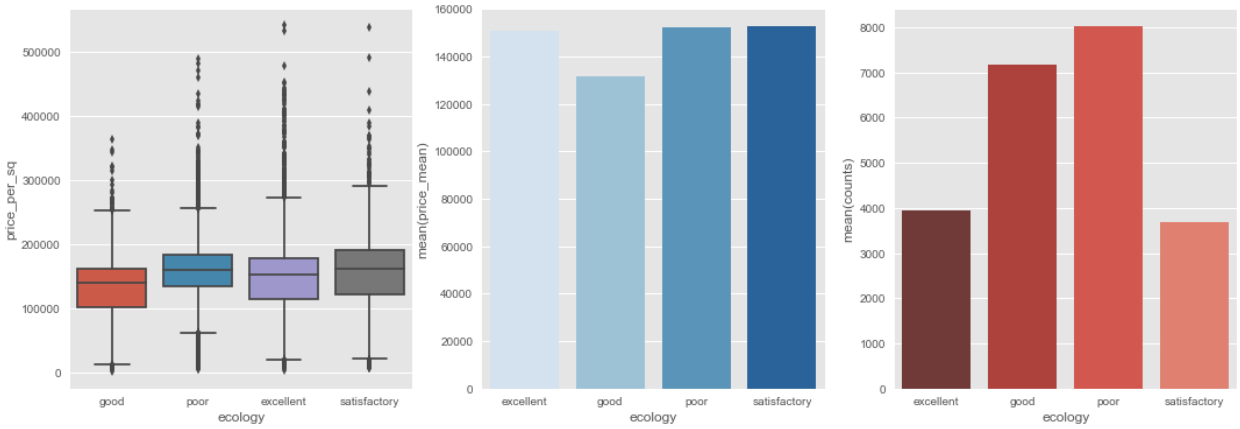
Public Healthcare

Being near to the Public Healthcare is a benefit for neighborhood houses. It could be seen that per-squared price is negatively related to the distance from public healthcare.

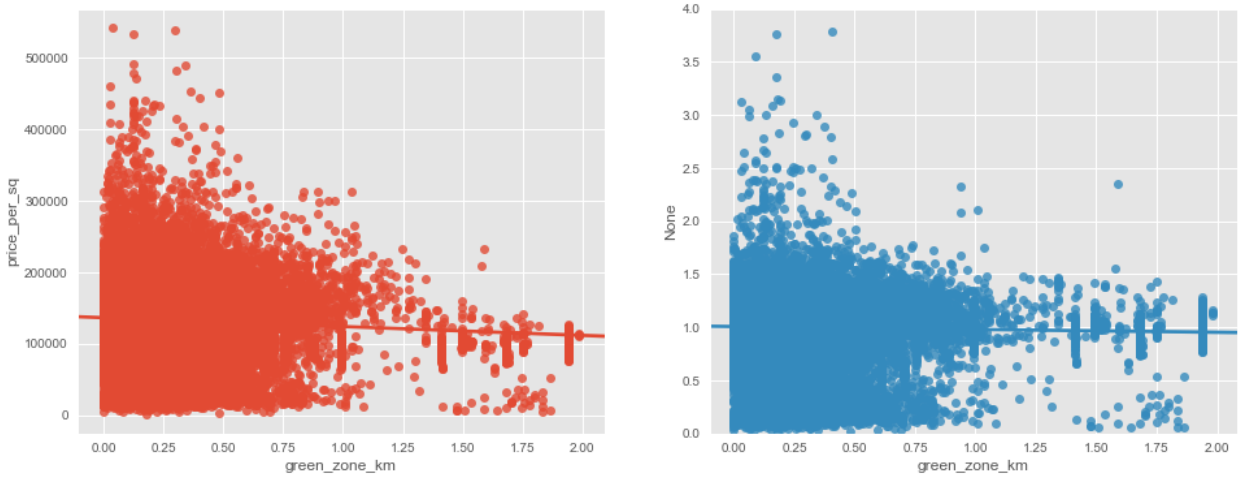




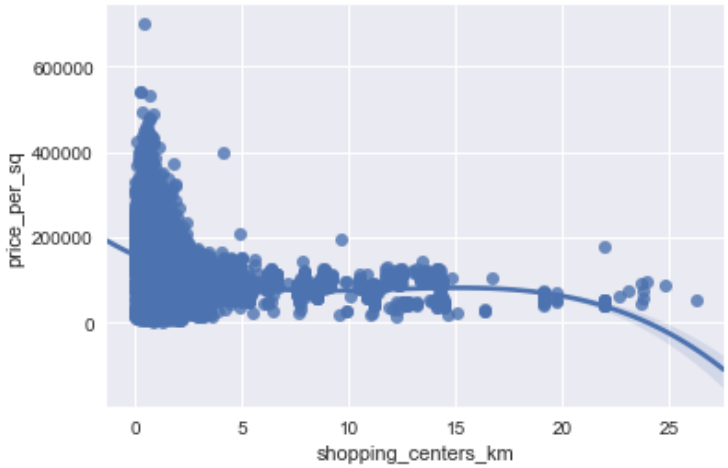
Ecology



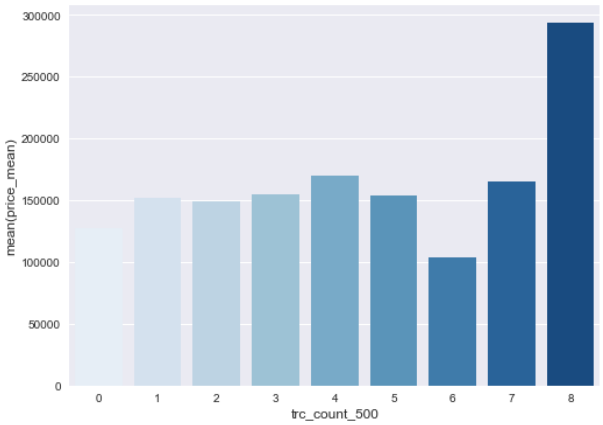
Green Zone

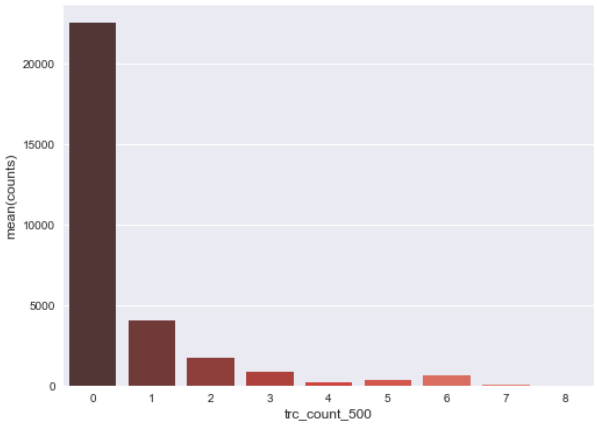


Shopping Center



Shopping Mall

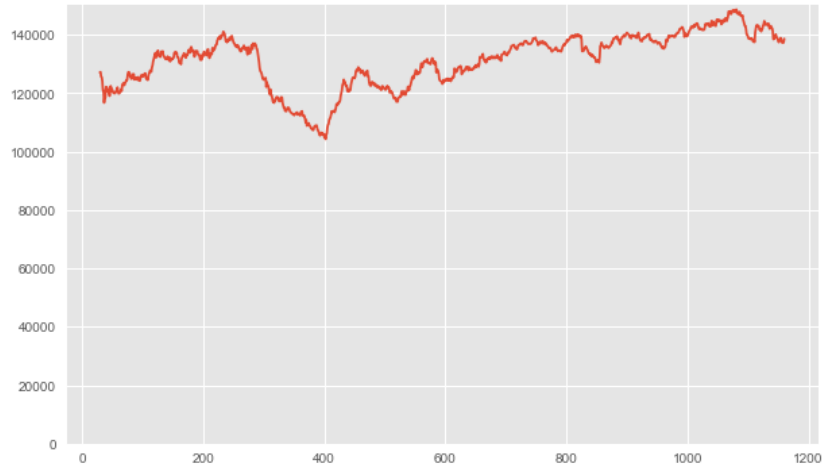


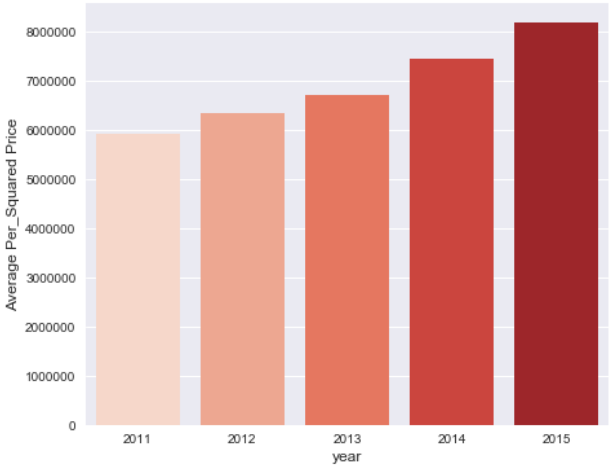


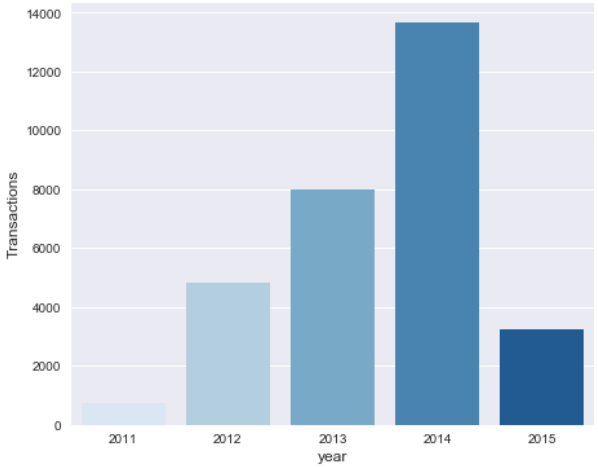
Time series

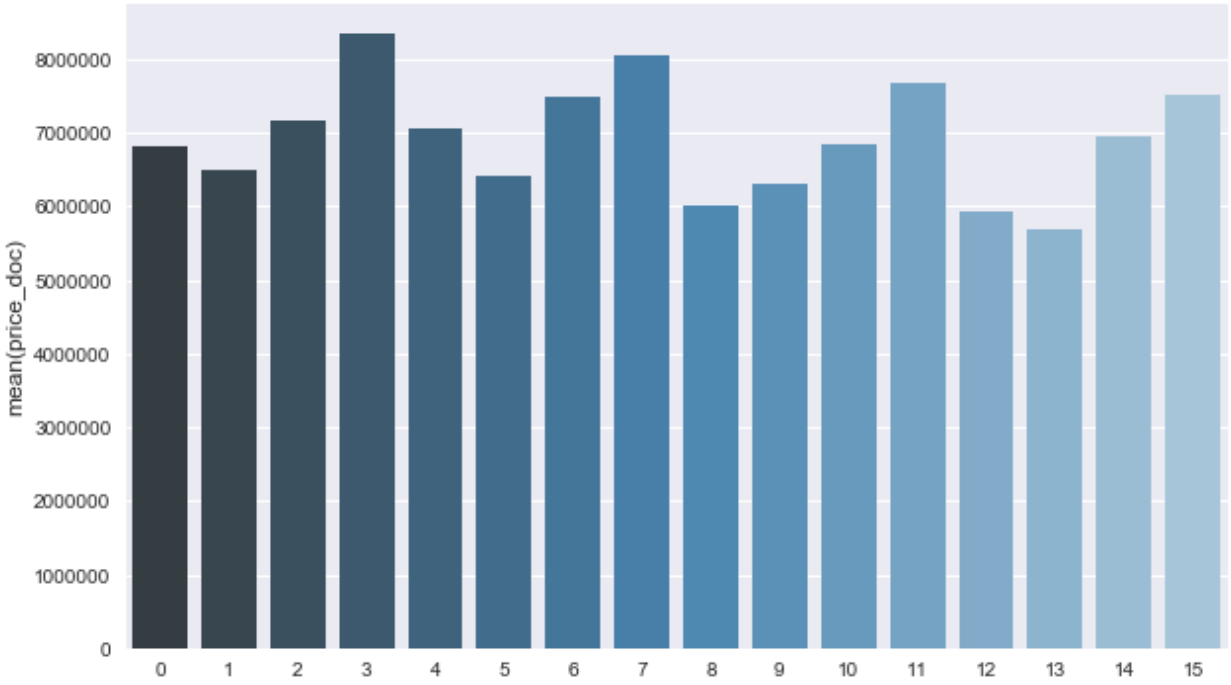
one of the most useful analysis in economic data is the Time series analysis. Here we could seek the economical parameters and the market characteristics through the time. Also we can see some notable changes that could be related to some different happenings.

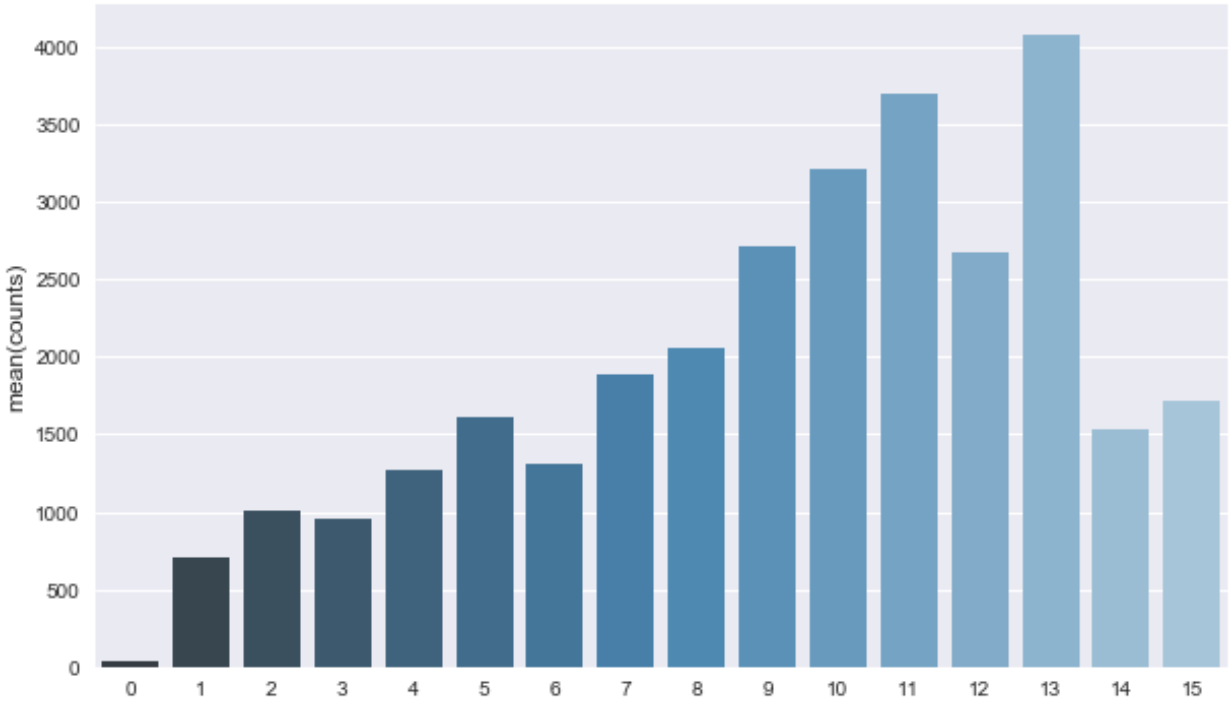
- From the first figure, it seems that, generally, the house prices have increased most of the times, except for a period of time in … that it deeply decreased.
- The average of the house prices has increased annually. In this period the annual average growth in house prices is around 6%.
- On the other hand, in the period of 2012-2014, the Number of Transactions in the House Market has a notable growth, 40% for 201 and 70% for 2014.
- A very interesting thing could be seen in the 4th figure, the quarterly average prices. We can see an orderly seasonal characteristic in the House prices. For 3 successive years, reaches to its highest value in Summers, and after that it goes down. This is a very important thing that could be used in learning the regression model.


140000
120000
100000
80000
60000
40000
20000
0
0
200
400
600
800
1000
1200


Average Per_Squared Price
8000000
7000000
6000000
5000000
4000000
3000000
2000000
1000000
0
2011
2012
2013
2014
2015
year


Transactions
14000
12000
10000
8000
6000
4000
2000
0
2011
2012
2013
2014
2015
year

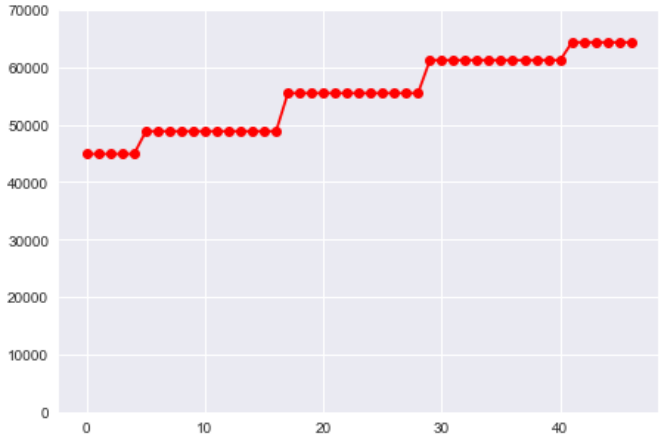


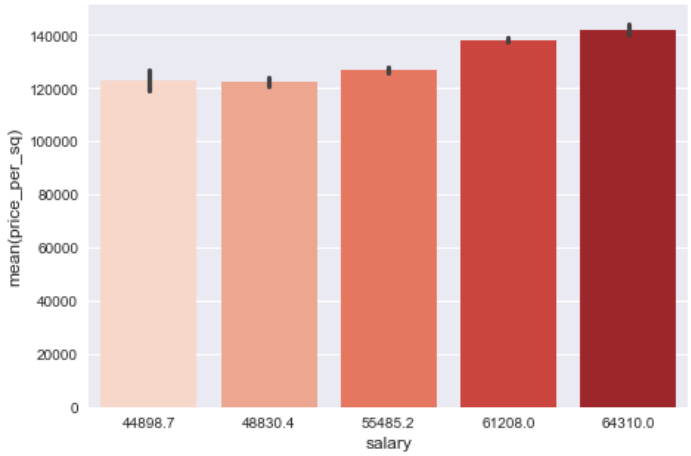


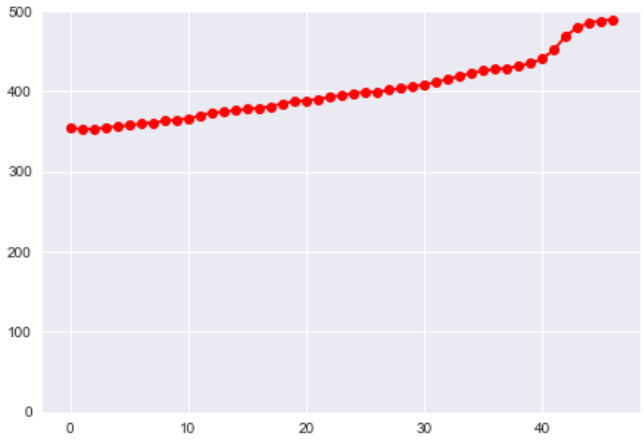
Economical Parameters

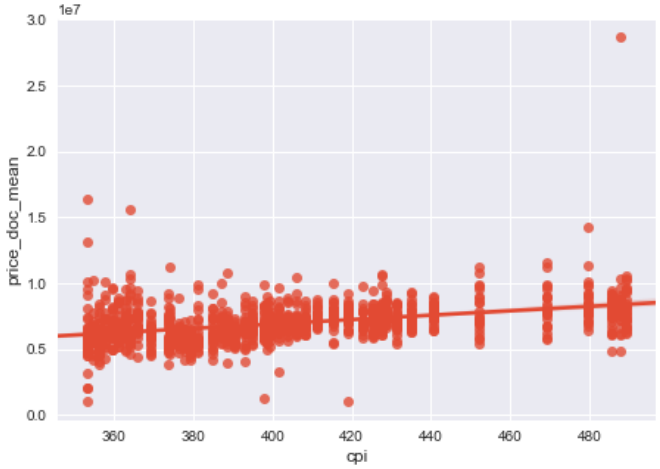
There are a lot of economical parameters, that many of them are correlated to each other, and have an effective impact on variations in markets. The most important parameters, that also people often talk about them, are the exchange rate, Inflation rate, average salary and House Funding. Here we can see the variations of these parameters over the time and also their correlations with average of the house prices.

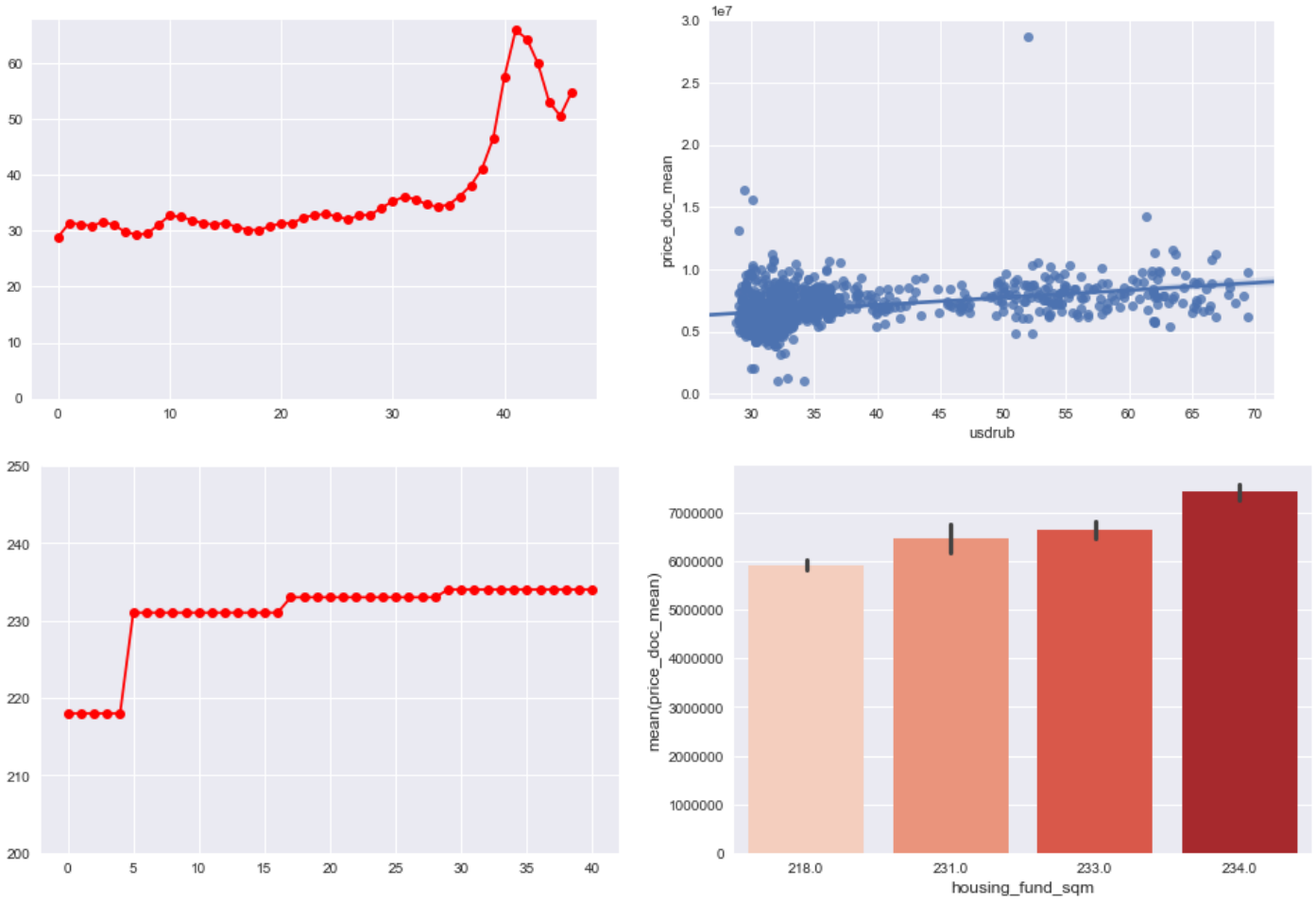
- The exchange rate USD-RUBL has increased gently for a long time, until 2015 that it abruptly jumped up around 100% in a period of 3 months. This might be related to EU and US sanctions against Russia. Despite the USD-RUBL exchange rate has a positive relation on house prices, however, we don't see an abruption in the sale prices.
- Inflation rate is one the most important parameter that prices are expected to change with. It can also show economic stability during the time. As seen in the figure, the Inflation growth rate has been around 6.7% for more than 3 years, until 2015, that increases up to 30% for a period of 4 years. However, it returned back to the 6% rate after 4 months.
- The average salary had been growing for 3 years with around 10% annually. However, we see that the average salary growth is only 5% in 2015.
- Housing Funds are mostly used to shake the house market. This could increase the number of transactions. Maybe this is why a big jump of Housing Funds has been occurred in 2012.







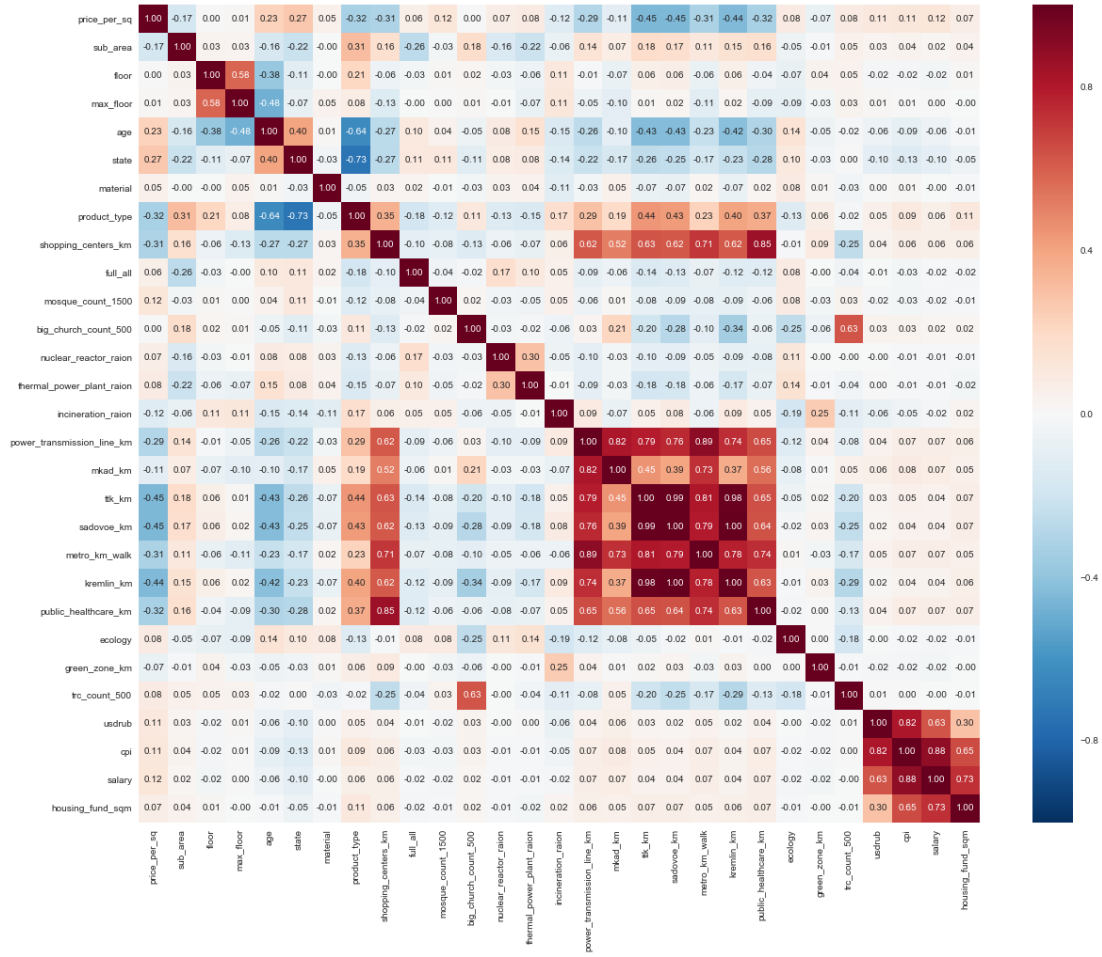


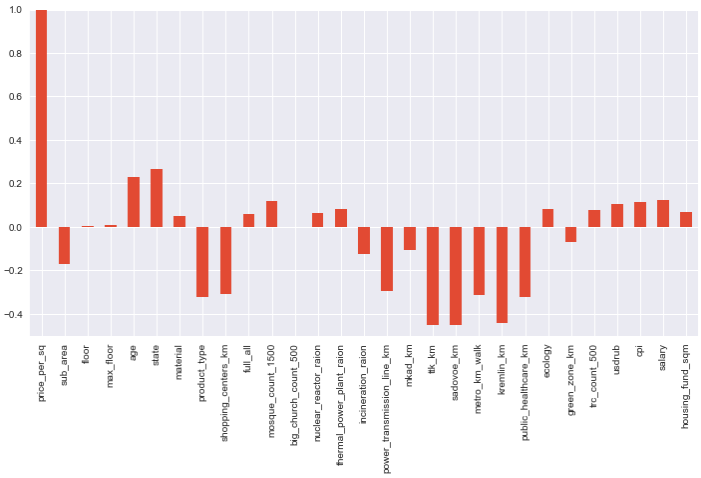


Correlations

The figure below shows the correlations between these parameters using color coding. We can find groups of features most correlated with each other, for example, we see that the transportation features are highly correlated to each other, or also we can see this property for economical features.

In the second figure, the correlation between average per squared price and other parameters is shown. We can find most correlated features, positive or negative relation, to the sale prices. For instance, age of the house, state, region and transportation facilities have an impact on prices. However, floor of the house and nearing to the church have almost no effect.



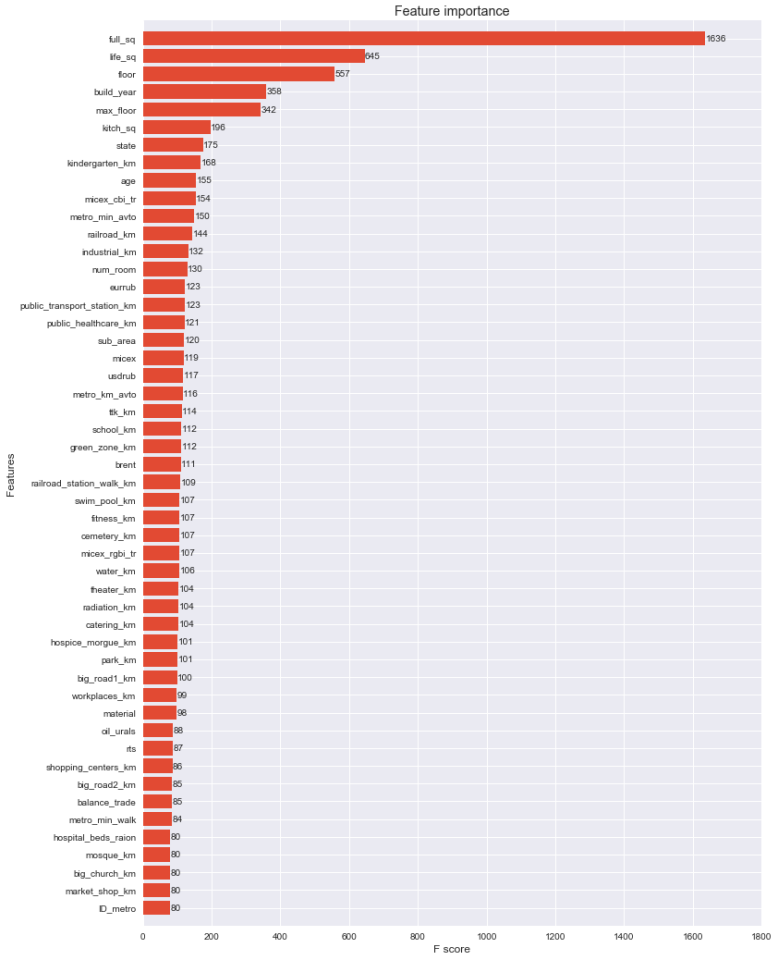


Important Features

There is a very useful method implemented in algorithms, is feature_importance. It shows the most important features in data based their positive effect on some internal scores such as Gini Index measured during the Learning process. Here I used the Extreme Gradient boosting model

(xgboost), and the 50 most important features are shown in the figure below with respect to their scores.

Here, we can see that, total and life area of a house, floor, state and age are the most important features. This is not an absolute measurement, as the collinearity between different parameters especially in high dimensional space, might be concluded to some weird results. As an instance we can see the Kindergarten distance as a very important feature.



Machine Learning:

The House price prediction is a regression problem, that could be seen from 2 different points of view, Interpolation and Forecasting. Naturally in real application the main goal is Forecasting the sale prices for cases in future, not cases in before. However, I evaluated both Regression and Forecasting power of algorithms.

For both of tests, the procedure is generally the same with some minor differences. The first step is preprocessing data that consists of data wrangling. feature engineering and label encoding for all algorithms. The second step is splitting data into train and test set. The train set is used to

learning and validation of the model and the test set is used to evaluating the model on a fresh data. For Interpolation problem I simply split the whole data randomly, using the train_test_split. 30% of the data is dedicated to test set and the remaining as training. However, the Forecasting problem is different. The oldest data must be used as training and the newest one as the test set. Therefore, the data related to the year of 2015 are assigned as test set and the remaining one are held as the training.

In next step, a naive imputation s applied on both training and test dataset. However, this step is ignored for xgboost algorithm which handle the missing values by itself. Finally, I applied five different regression models for both cases, Interpolation and Forecasting. The Algorithms utilized in this project are, Linear Regression, Decision Tree, Random Forest, Gradient Boosting and Extreme Gradient Boosting.
All models are evaluated with five different criteria. The most usual score utilized for Regression models is R2R2 that is a statistical measure of how close the data are to the fitted regression line. Mean Squared Error is another common way to measure the Regression power. In most relevant challenges in Kaggle, another measure, Root Mean Squared of Log Error (RMSLE) is used. I used also Mean Absolute Error (MAE) and the ratio of MAE and average sales prices, that give a practical sense about the error rate of the Regression model.

Regression Power

Linear Model

The simplest choice for a Regression model is Linear Regression. It's very fast and doesn’t need parameter tuning. On the other hand, it works blindly and is not an accurate model especially in the case of Forecasting. As could be seen the model is not over fitted and scores of Validation and test set are very close. There Error ratio of the test set is more than 24%.

Decision Tree

Decision Tree is a non-parametric supervised learning method used for classification and regression. It's very fast and simply understandable. It can handle categorical data and doesn't need data scaling which is required in some algorithms such as SVM. However, it's not powerful in extracting linear combinations of features.

It has different hyper parameters that must be tuned using a grid search and cross validation. The most important parameters are, the maximum depth of the tree (**max_depth**), The minimum number of samples required to split an internal node (**min_samples_split**), The minimum number of samples required to be at a leaf node (**min_samples_leaf**), Threshold for early stopping in tree growth (**min_impurity_split**). All of these parameters must be well tuned to prevent both underfitting and overfitting.

With inspiration from [1], [2], I utilized a stage-wise tuning to optimize the hyper parameters. The full procedure could be seen in [5].

With the selection of the best parameters, I got 23.2% error ration, 1.5% better than the Linear Regression.

Random Forest

As mentioned Decision Tree has various advantages, unless it's not a powerful and accurate model. However, with the use of Ensemble methods we can cover that and boosting its accuracy. Random Forest is a meta estimator that fits a number of classifying decision trees on various sub-samples of the dataset and use averaging to improve the predictive accuracy and control over-fitting.

In addition to the parameters mentioned for Decision Trees, Random Forest has two important features as well. The first one is the number of trees in the forest (**n_estimator**), and the other one is the number of features to consider when looking for the best split (**max_features**). In most cases the best value for **max_features** is the squared number of features. However, the best value for **n_estimator** must be tuned. The higher the **n_estimator** is selected, the better accuracy is gotten, with the penalty of slower training.

Again I used and stage-wise parameter tuning, that could be followed ..., and the best parameters concluded to 19.5% error ratio, 3.7% better than Decision Tree, and 5.2% better than Linear Regression.

Gradient Boosting

Gradient boosting is a machine learning technique for regression and classification problems, which produces a prediction model in the form of an ensemble of weak prediction models, typically decision trees. It builds the model in a stage-wise fashion like other boosting methods do, and it generalizes them by allowing optimization of an arbitrary differentiable loss function.

The most important parameters in Gradient Boosting, are **max_depth**, **min_samples_split**, **min_samples_leaf**, **min_impurity_split**, **n_estimators**, the fraction of observations to be selected for each tree (**subsample**) and the **learning_rate** that determines the impact of each tree on the final outcome.

Again I used a stage-wise hyperparameter tuning in which the complete procedure could be seen [5]. Using the best selection of parameters, concluded 19.5% error ratio the same as the Random Forest.

Extreme Gradient Boosting

Extreme Gradient Boosting is an efficient and scalable implementation of Gradient Boosting framework by Friedman, with several benefits. It's fast as it can automatically do parallel computation, it's optimized for sparse input, and also it can handle missing values in the input data.

The most important parameters are **max_depth**, **learning_rate**, **subsample**, **n_estimator** and the fraction of columns to be randomly samples for each tree (**colsample_bytree**).

The error ratio acquired from this algorithm. using the tuned parameters, is 19% that means 0.5% better than Gradient boosting.

Summary

The best performance is acquired from xgboost with 19% error ratio and the worst one is measured for Linear Regression model with 24.7% error ratio. The table below shows the scores of different models in Regression problem:

Regression Power

Regression Model	R-Squared	RMSLE	Error Ratio
Linear Regression	0.5547	0.5143	0.2474
Decision Tree	0.5834	0.4911	0.2326
Random Forest	0.6844	0.4685	0.1949
Gradient Boosting	0.7007	0.4698	0.1949
Xgboost	0.7104	0.4671	0.1905

Forecasting Power

As mentioned before, the most important problem in price prediction is the Forecasting power of the model. Here again I prepared data, and splitter the newest data as the test set and the remaining as the train data. The algorithms used in the Regression section, utilized in Forecasting power. The optimum parameters for each model are selected as before.

The best performance has been measured for Gradient boosting with 17.9% error ratio and the worst one is for Linear Regression with 45% error ratio. As expected, the Linear Regression had weaker performance in Forecasting problem comparison to Regression problem. The table below shows the scores of different models:

Forecasting Power

Regression Model	R-Squared	RMSLE	Error Ratio
Linear Regression	0.1993	0.6226	0.4516
Decision Tree	0.6247	0.4217	0.2346
Random Forest	0.6247	0.4217	0.2346
Gradient Boosting	0.7505	0.3966	0.1798
Xgboost	0.7424	0.3991	0.1881

Conclusion and Future Works

In this project, a real application in machine learning and data science investigated. We saw that in such a project, it's important to firstly exploring the dataset to find outliers and bad data in addition to missing values. The data wrangling is not only to handling the obvious missing values but also the hidden ones.

Feature engineering is an Art in the data science. It's very important in both EDA and ML. For example, we saw that, despite some feature have a strong effect on House prices, however, they don't have a notable impact on per squared house prices.

Different machine learning algorithms evaluated in two different problems. The Regression and the Forecasting problems. We saw that most of the algorithms have a similar performance in both cases. However, the Linear Regression had a weak performance on Regression problem and much weaker in Forecasting one. The best performance was computed for Gradient boosting and Xgboost, with around 19% error ratio, in both cases.

In contrary to our expectation, a Deep Learning model with 4 Fully connected layers, didn't have a satisfactory result. However, it could be deeply studied in future.

By the way, there is some recommendations for future researches, that have been concluded within this project:

- As seen in Time series analysis, the House Market has a seasonality characteristic. Therefore, it's highly recommended to utilize the TSA for such these problems, especially in the case of forecasting the target in future.
- It might be also useful to learning models in clusters. So, we can firstly split data in some number of clusters, and then learning the regression model on each one. This could be effective because different clusters might have different characteristics.
- Another recommendation is the utilization of a combination of different models with different wrangling options and applying some weights proportion to their accuracies.

References:

[1] https://www.analyticsvidhya.com/blog/2016/02/complete-guide-parameter-tuning-gradient-boosting-gbm-python/

[2] https://www.analyticsvidhya.com/blog/2016/03/complete-guide-parameter-tuning-xgboost-with-codes-python/

[3] http://xgboost.readthedocs.io/en/latest/model.html

[4] https://www.youtube.com/watch?v=ufHo8vbk6g4

[5]https://github.com/hnamzian/Springboard/blob/master/Capstone%20Project/Capston%20project%20-%20ML-DNN.ipynb

[6]https://github.com/hnamzian/Springboard/blob/master/Capstone%20Project/Capston%20project%20-%20ML-DNN.ipynb

[7] https://github.com/hnamzian/Springboard/tree/master/Capstone%20Project