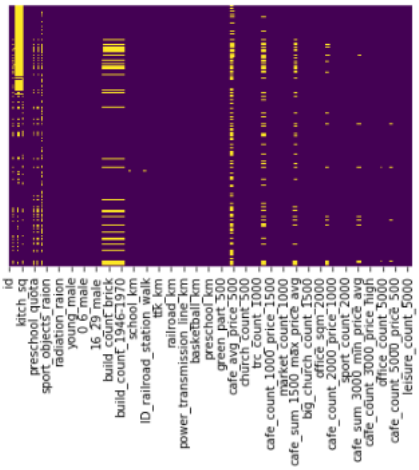
This is a report for the analysis and prediction for Russian housing market price data set which can be found on kaggle. This report details the analysis approach and prediction of the housing market and findings within the data. The dataset includes over 250 features all of which predicts the price of the real estate. This report is spread out into three sections: data inspection, feature engineering and predictions, and feature visualization and analysis. For model implementation Random Forest and XGBoost are applied and compared between the two with cross validation.

Data Inspection

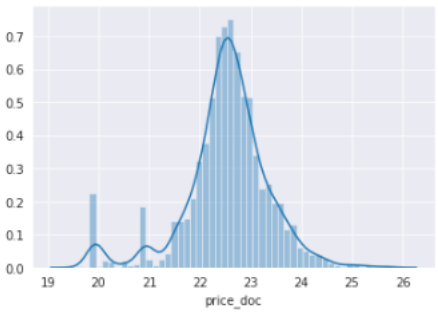
Partial sampling on the data set is done to check the amount of missing data.



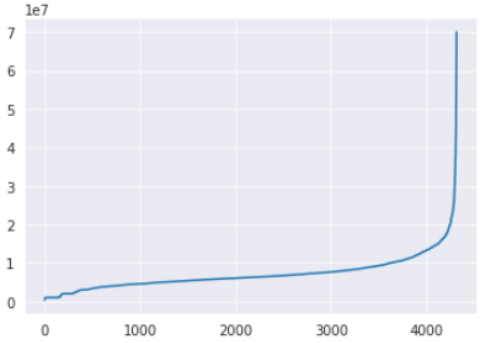
Visualizing the amount of missing data.

There are a few columns that have over 70% of the data missing and are dismissed in the set. For the remaining missing data they are filled with the median values with respect to the columns.

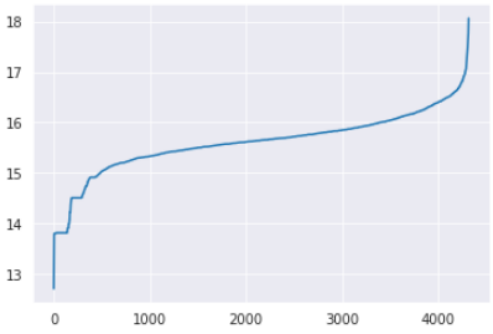
Next the price distribution of the market is observed. The price is transformed under a log scale which have a defined Gaussian curve. There are few exceptions at the lower price range which could be outliers. The price range is also plotted in normal and log scale. In log scale the ends of the plot leads to some extreme values. In order better predictability a set range for the log-price is filtered and used in model prediction. After defining the dataset model training is performed with Random Forest and XGB.



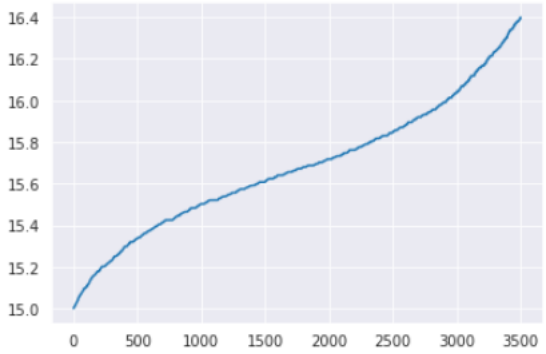
Distribution of Log-price



Normal range of Price



Range of Log-Price



Range of filter Log-Price

Feature engineering and predictions

Feature engineering is performed for the 250 features in the dataset to weed out features that are not significant and could cause overfitting. The approach is tally the feature importance within the model within each kfold cross validation (and not tallying the feature importance after the CV is finished which is the wrong approach since the CV have already seen the dataset). For the first CV run a 10 fold is performed to tally the importance of the features. Afterwards a CV with 5 folds is performed on different numbers of top features, checking how many are ideal. After determining the appropriate number of features another 5 fold CV is performed on those features on a test set to see how it performs. The models used here are XGB and Random Forest.

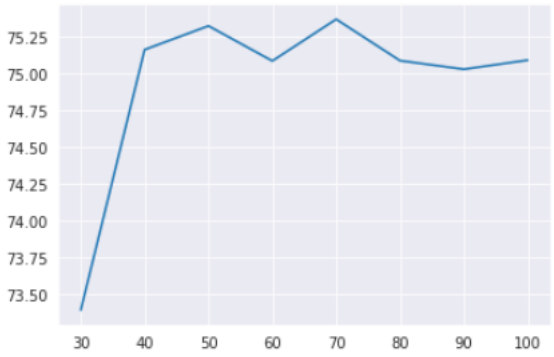
XGB

In this section XGB is implemented. The list below are a few of the top features found within the 10 fold CV. It seems full_sq has the highest importance, not that surprising.

	0	1
61	full_sq	1020
133	year	270
79	max_floor_int_11	190
71	ice_rink_km	150
104	public_healthcare_km	120
113	school_km	100
18	build_year_int_11	100
103	prom_part_5000_int_11	100
92	park_km	100
107	railroad_km	100
13	big_road2_km	90
132	water_treatment_km	80

Top Features from XGB 10 fold CV.

Next, the number of features is determined with the plot below detailing score or performance from the 5 fold CV depending on the number of the top feature used. This is ran several times with and without shuffling of the data and the score does not differ that much and hovers around 74%. Based on this information the top 60 features will be used to train the final model.



Accuracy score from the number of instances used in XGB CV.

Below is the table containing MSE, RMSE, and the R2 score with different inputs. The XGB model is trained with the top 60 features found from feature selection with the exception of Test(full features).

The Training input data is the same set that trained the model and is fitted back in. The MSE, RMSE, and R2 of the Training data are quite low with a good R2 fit of 0.75. The Test data is a separate set used as for testing as stated. The errors a slightly higher than the Training set but its RMSE is still quite low indicating there isn't a lot of points that are far away from the prediction. The R2 score is lowered but overall holding above 70%. The full range price is fitted into the model with Price(full range). As expected the MSE and RMSE increased significantly since does not predict prices that are too high or too

long. The R2 score is below 0.3 indicating the model most likely would not work if it included extreme prices.

Lastly, the full set of features is used in both the model training and testing. For this model the full set of features is used in training and is applied to Test(full features). The results are similar to Test where its MSE and RMSE are slightly higher but with the same R2.

Input data	Mean Square Er	Root Mean Square Er	R2
Training	0.0251	0.159	0.752
Test	0.0277	0.166	0.717
Price (full range)	0.246	0.496	0.297
Test (full features)	0.0280	0.167	0.714

Table of different input data and error from XGB model. Training – is the same training set used to train the model. Test – testing set separate from training. Price(full range) – full range price from test set is used. Test(full features) – full feature is used in both training and test set.

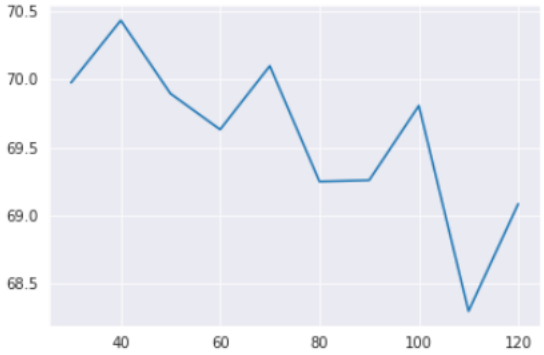
Random Forest

Random Forest model is implemented to compare its performance with XGB. The same approach is done with tallying the important features with a 10 fold CV and checking the performance with those top features. Afterwards checking the fit of the model by observing the errors with a 5 fold CV.

The list below shows some of the top features from the RF model an overall full_sq is still the dominant feature. With the RF model the number of features used did not vary that much similar to XGB but its accuracy is lower at around 69%. In this case the top 50 features will be used.

	0	1
145	full_sq	17.465055
85	cafe_count_3000	6.991286
77	cafe_count_2000	3.154730
252	sport_count_3000	3.102908
98	cafe_count_5000_price_2500	1.120673
216	power_transmission_line_km	1.002635
285	year	0.779053
171	life_sq_int_11	0.472478
160	indust_part	0.444033
198	num_room_int_11	0.398602
143	floor_int_11	0.395921
181	max_floor_int_11	0.362176
52	build_year_int_11	0.361687

Top features from Random Forest 10 fold CV.



Accuracy score from the number of top feature used.

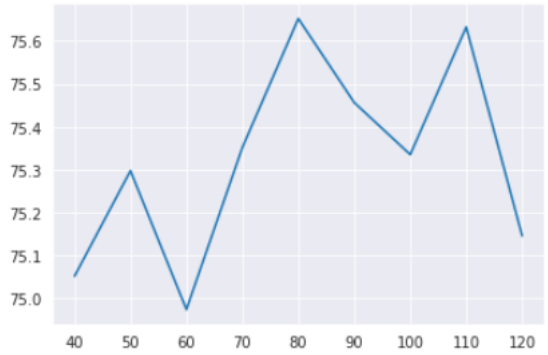
The table below shows the MSE, RMSE, and R2 of the different input data from the trained RF model with the top 50 features. Training data is the same set used for training the RF model. The errors are slightly higher than those of XGB and its R2 score is lower as well. For the Test MSE and RMSE they increase by a bit but stays relatively the same and with a lower R2 score. For Test(full feature) the model is both trained and tested with the full set of features, overall the errors increased along with R2. If one were to compare, the XGB model gives a slightly lower error but the overall fit is better with RF.

Input data	Mean Square Er	Root Mean Square Er	R2
Training	0.0265	0.163	0.743
Test	0.0285	0.169	0.724
Test (full features)	0.0304	0.174	0.706

Score table with different input data from RF model.

Using RF features in XGB

Here I follow a procedure from a publication where it might be possible to improve accuracy by using important RF features in XGB model [1]. By using this idea the performance is first plotted below by following the as process and before but now using the top RF features in XGB. From the performance plot below 80 feature should be used since it has the highest performance.



Accuracy score based on the number of RF features used in XGB model.

The table below shows the different input of the trained model which follows the same procedure as before. For the Training data it has relatively decent fit and for the actual Test set the error and fit is similar to XGB with a slightly better R2 fit. Overall there does not seem to be that much of a difference for the new approach but it does proves it could be a valid solution for other data set.

Input data	Mean Square Er	Root Mean Square Er	R2
Training	0.0262	0.162	0.746
Test	0.0277	0.166	0.732

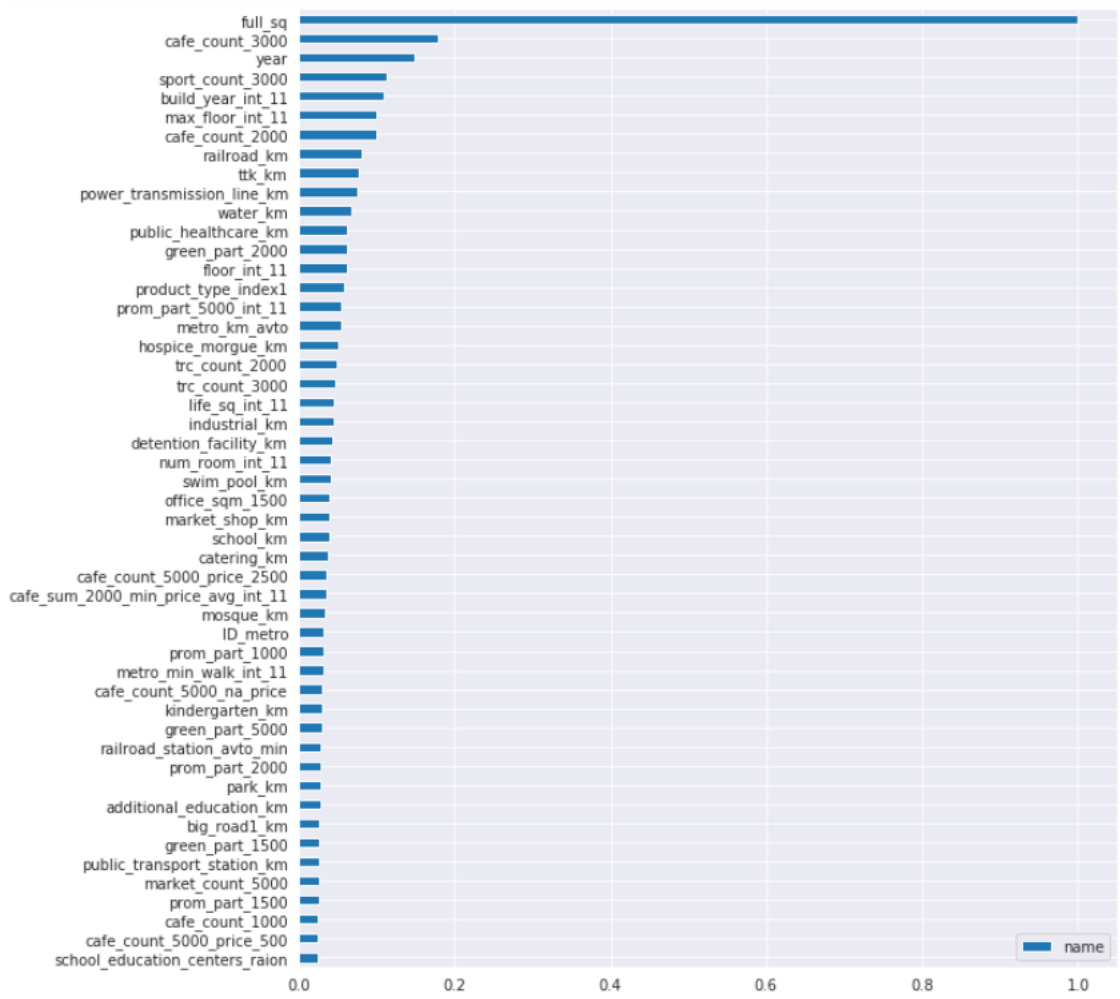
Table of the training and test set used in XGB with RF features.

Overall, XGB performs the best by a slight margin with a good fit over 70%. For Random Forest the model also performed quite well but with fit and error edge given to XGB. If one were to use a particular model for prediction XGB would still be the prefer choice even with the full set of feature.

[1] Galathiya, A. S., A. P. Ganatra, and C. K. Bhensdadia. "Improved decision tree induction algorithm with feature selection, cross validation, model complexity and reduced error pruning." *International Journal of Computer Science and Information Technologies* 3.2 (2012): 3427-3431.

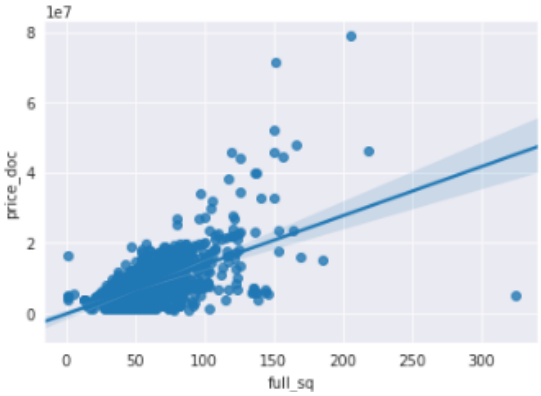
Feature visualization and analysis

Some of the important features related to full_sq and price will be shown here, more visuals can be found within 1) and 2) notebooks. Based on feature selection from RF and XGB model the top important features are visualized with full_sq being the most important.

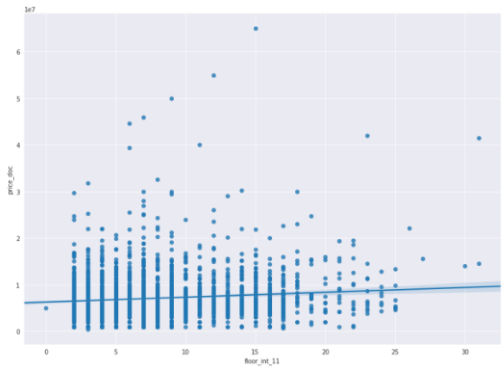


Sum tally of the feature importance found within each model. The score of the feature importance is normalized from each model and summed up and normalized again with respect to full_sq.

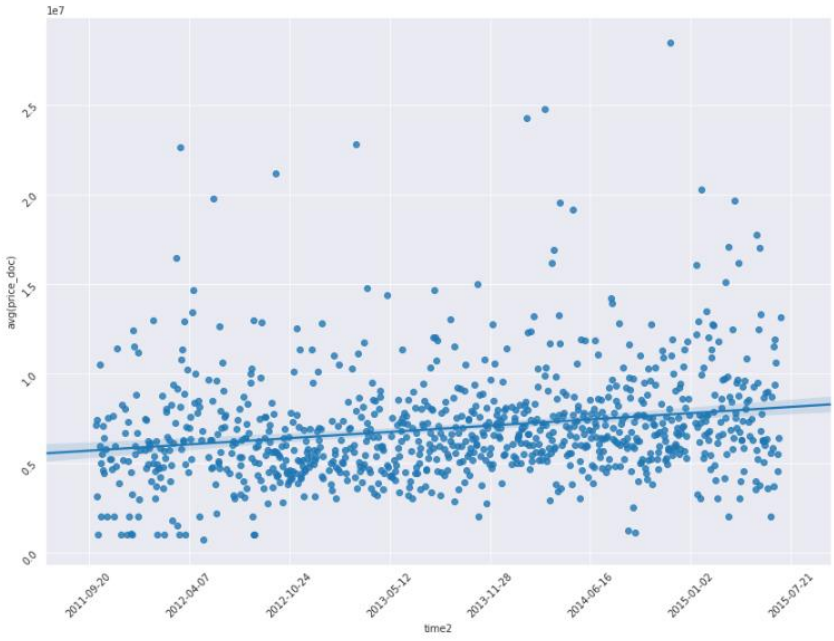
To start off lets look at the price increase for several features that is predicted to go up. Square-meter, time, and floor(how high) are plotted below with a regression line indicating the price direction as each unit is increased. Based on these information the size of the estate definitely causes the most change in price as it has a steep slope.



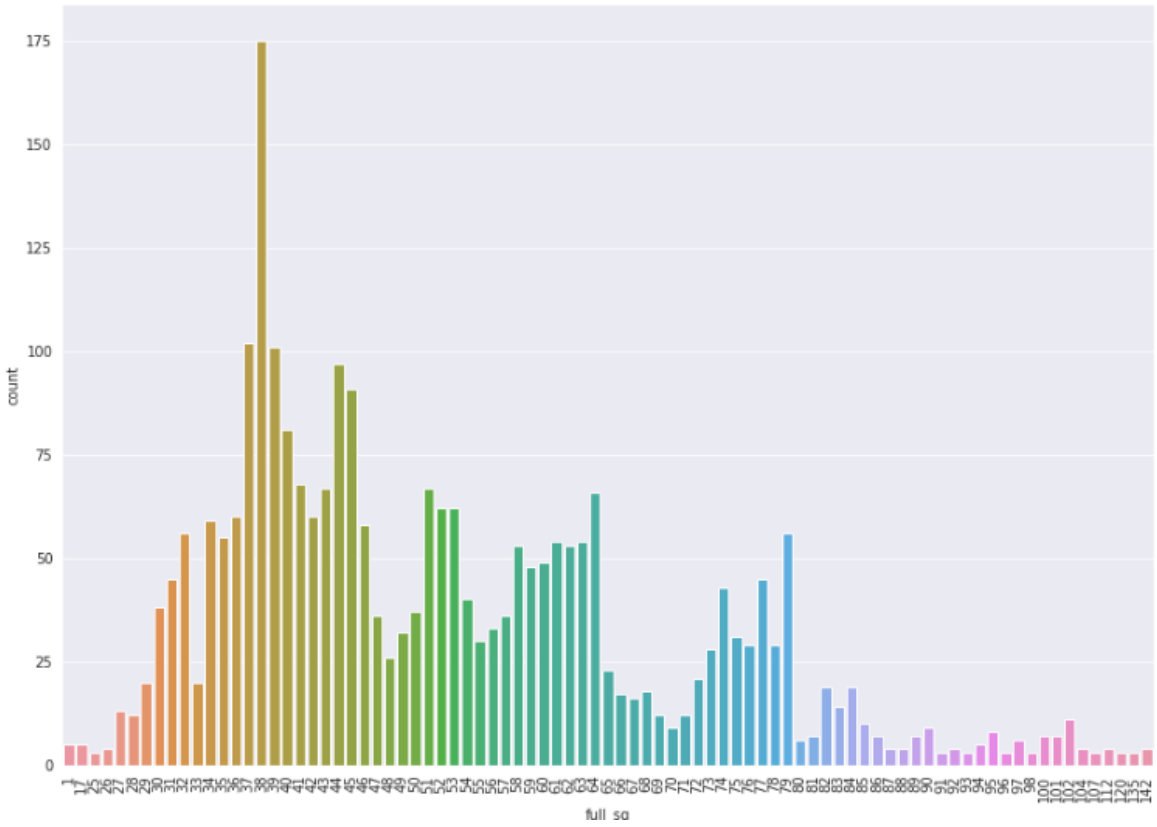
Price vs SQM



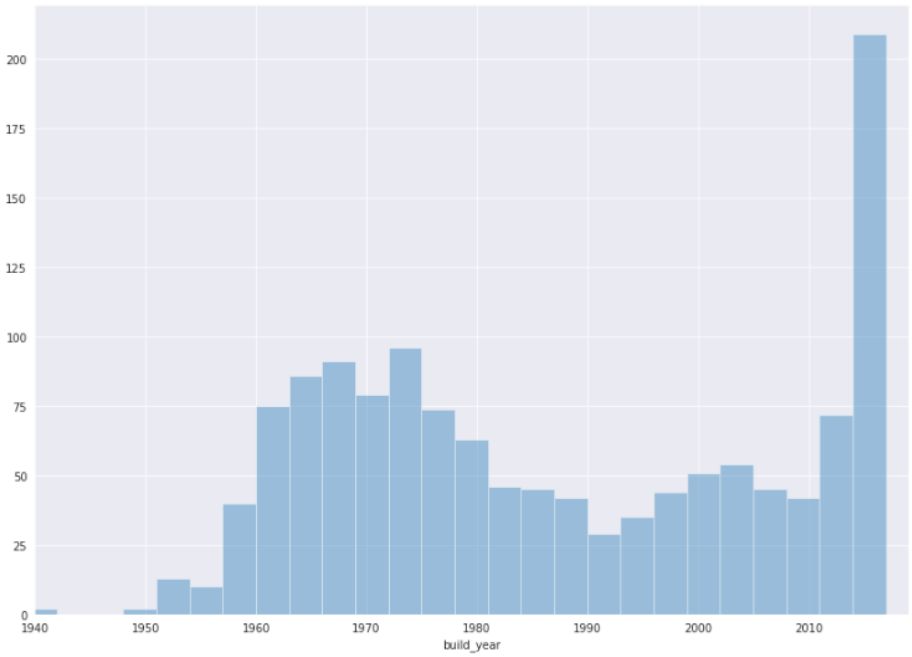
Price vs Floor(0-30)



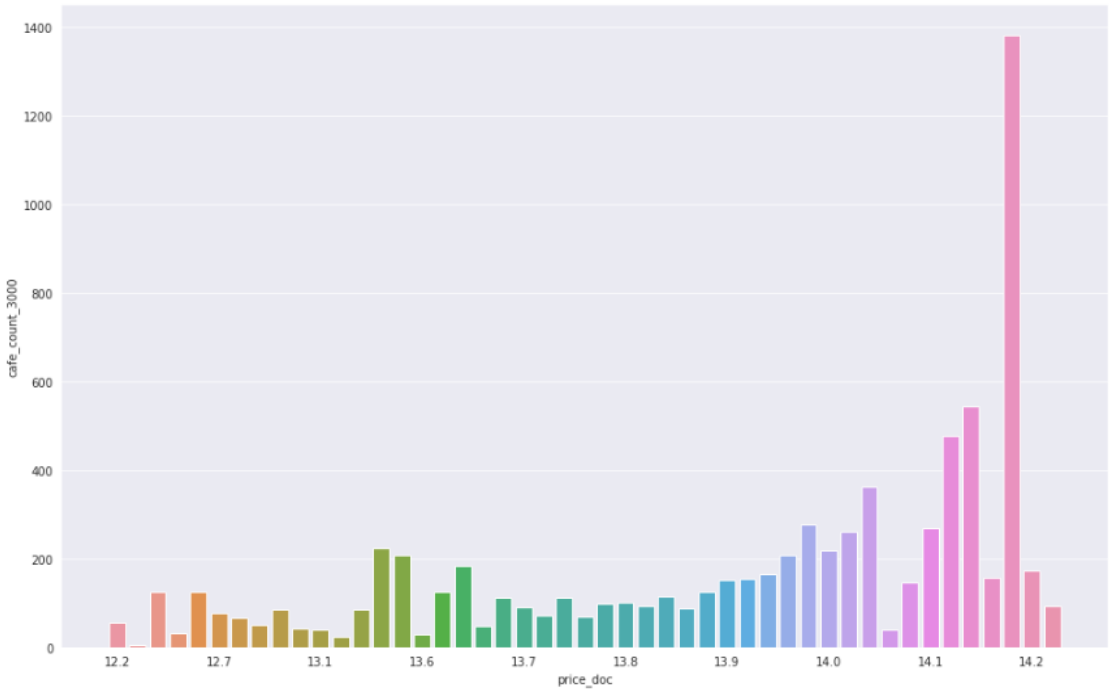
Avg Price vs Time(2011-2015)



Distribution of full_sq



Distribution of year built



Log-Price and the amount of café near them. The price definitely goes up as there are more cafes around them.