Capstone Project

Elias Laura Wiegandt

Sberbank Russian Housing Market Kaggle competition

May 26, 2017

1 Definition

1.1 Domain Background

For many individuals, buying and selling housing will be the biggest financial decision they make during their lifetime. A bad decision, such as buying to dearly or selling too cheap, can cause financial ruin. A model for housing prices can be used as a benchmark to avoid buying overpriced or selling underpriced. When a house is overpriced the seller wins and the buyer loses, and vice versa when it is underpriced. Disregarding payment to an estate agent or bank, buying and selling housing is essentially a zero sum game. It is assumed that there is decreasing marginal utility of purchasing power (the gain from "winning" the housing trade). This is known as "Gossen's law" and is generally accepted in Economics [1]. This causes both the buyer and seller to wish to decrease the likelihood of over- or underpricing, as their downside is bigger than their upside. This is the raison d'être for constructing a model that predicts housing prices.

1.2 Problem Statement

My capstone project is to build a model for predicting Russian housing prices, as part of the Kaggle competition: Sberbank Russian Housing. This is a regression problem, where the inputs are features of the housing, its neighborhood and the macro economic environment. The output of the project is a model that predicts housing prices given the dwelling's features, its neighborhood and the macro economic environment.

1.3 Evaluation Metrics

Root Mean Squared Logarithmic Error (RMSLE) is used for evaluating the models. RMSLE is used as benchmark as it is the metric that Sberbank has chosen for the competition. RMSLE is defined as:

$$\epsilon = \sqrt{\frac{1}{n}\sum_{i=1}^{n}\left(\log\left(p_i+1\right)-\log\left(a_i+1\right)\right)^2} \tag{1}$$

Where ϵ is the RMSLE, n is the total number of observations, p_i is the prediction, a_i is the actual value for observation i and $\log(x)$ is the natural logarithm of x.
RMSLE puts less emphasis on larger errors than other standard metrics such as RMSE, but is still differentiable unlike e.g. the absolute error. The error on large residences can also be expected to be larger. Using RMSE would imply the model put would focus more on on getting the larger housing prices right than the smaller housing prices, relative to RMSLE. As it is likely that a few housing trades will relate to very expensive housing, using RMSLE instead of RMSE is a way to make certain that these expensive residences do not overly dominate the loss function the models seek to minimize.
At the time of writing, the best contributions have an RMSLE of 0.31. I do not expect to be able to match this, but my goal is to achieve an RMSLE lower than 0.5. Further, a Random Forest Regressor with 100 trees "naively" applied to the entire competition dataset will serve as the benchmark for the other models.

2 Analysis

In this section, the dataset Sberbank has provided for the competition will be described and analyzed. This is done to build an understanding of the data. As the entire dataset is very large, only the dwelling specific variables are included in the EDA. They are chosen because they are expected to be the most important variables in determining the price of a given residence.

2.1 Data

Sberbank has made data on housing trades and macro economic data available here: Sberbank Data. The data is also included in the uploaded zip folder. The data consists of three headline categories of data: dwelling specific, neighborhood specific and macro economic.
The dwelling specific dataset describes 14 features of the dwellings at the time they were sold. One of these 14 features is the target variable, the price the dwelling was sold at. The data includes variables for the full size of the residence, the number of rooms and other variables specific to the dwelling itself. The neighborhood specific data includes 208 variables describing the neighborhood, e.g. how many cafs there are within a specific radius of the dwelling and distance to the nearest kindergarten. The macro economic dataset includes 101 macro economic variables such as GDP growth and inflation measures.
In total, this makes for 1 label and 402 features, before categorical variables are transformed to binary variables. In the uploaded zip folder, the file `data_dict.txt` with all variables listed is included.
Sberbank has included both a training dataset for training and validation, and a set of test-features that can be used to test the performance of the final model. To do this, a csv datafile with predictions of the prices for the dwellings in the test set have to be uploaded to the Kaggle website, which then returns the Root Mean Squared Logarithmic Error (RMSLE) for the predictions. As this was cumbersome, it was not done during this project.

2.2 Preprocessing

The data was thoroughly cleaned. This was an iterative process, in which later analyzes gave rise to adjustments of data and rerunning of charts and models. The final result was a quite rough handling of outliers and missing values. The approach is described below. The analysis could be rerun with only the variables singled out in the final and best model. Was further work to be done on this project, this would be an obvious candidate for refining the model and achieve an even better understanding of the data.

2.2.1 Missing values

All missing observations were replaced by their median values (for continuous variables) or mode (for categorical variables). The median is used instead of the mean, as the mean is affected by outliers. Replacing missing values in this way alters the distribution of the data, and is definitely not optimal. The alternative would have been to either drop all rows with missing values or impute the value of the missing value another way. As almost all variables contain missing values, dropping all observations with missing data entails dropping approximately 75 percent of all observations. A different approach could be to use KNN to "infer" missing values. This was dropped, as it is very time consuming. Further, if the missing values of a variable can be inferred from other variables in the dataset, the information in the variable is already in the dataset, and arguably nothing would be achieved by this more advanced imputation of missing values.

2.2.2 Outliers

Clear outliers in the dwelling dataset were identified and removed during the Data Exploration Process. These include values that should not be possible, such as an observation were it is stated a building was was built in "20052009", which is likely to mean 2005-2009. As categorical variables were "onehot encoded", the final dataset ended with well above 400 variables. Cleaning "by hand", as with the dwelling variables during the exploratory data analysis, is not feasible. For a rough automatic cleaning, a function was implemented on all data, which removes an entire row if one of its variables has a z-score above 8 compared to all other observations of this variable. This cleaning left 27,665 observations in the dataset.

2.2.3 Transformations

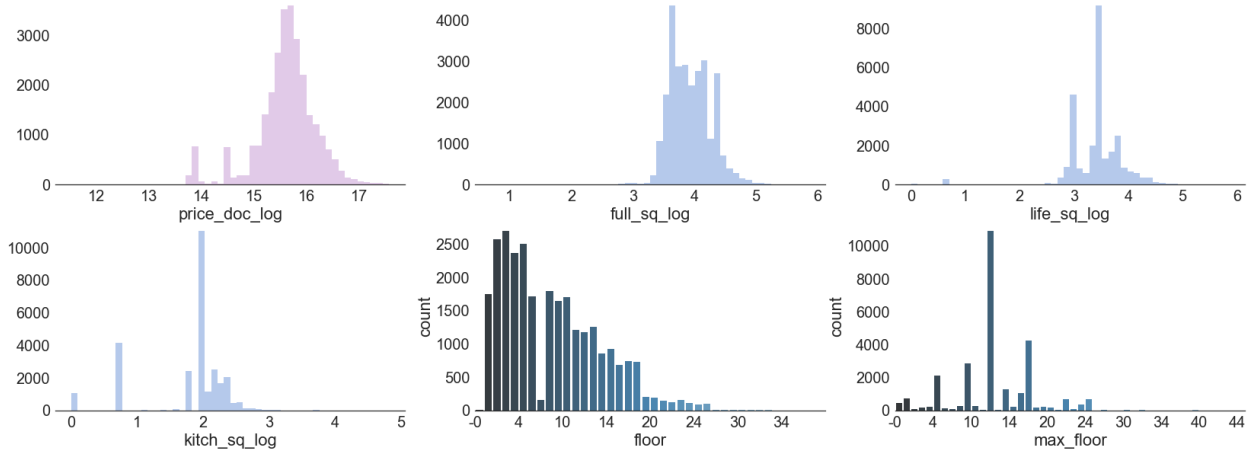
Some variables were log transformed to attain a more Gaussian distribution, see next section. Further, variables which were numbered but where the number denoted categorical values were reformated to categorical values and one hot encoded.

2.3 Exploratory data analysis

2.3.1 Distribution of the data

To attain a less skewed distribution, both the price and all measures of area were transformed using the natural logarithm. As none of them are negative, this pose no problem. As the logarithm of 0 is not defined, a "1" is added to all values before the transformation.

Figure 1: Distribution of numerical variables



Distributions of all variables have been plotted in figure 1 and 2. A description of the variables can be seen in table 1. As seen from the charts, "price_doc_log" appears to be pretty decently distributed. Replacing NaNs with the median or mode has created some spikes in some of the variables, e.g. "build_years". The variable "trade_year" is included here, but was removed from later models, as it will not be available for forecasting trades in a year not included in the dataset.

A few interesting points can be deduced from these plots. Dwellings are generally build from the 1950's and until now. Most of the trades involved residences with two rooms, walls of panel (which is strictly speaking not a material) and in state "2". The volume of trades were increasing from 2011-2014. Only data from January to June 2015 is included in the training dataset, so the sharp fall in trades for this year is partly cosmetic. That said, in the first half of 2015 there were only a fourth as many trades as in all of 2014, which could indicate a fall in the number of trades.

2.3.2 Relationships between dwellings variables

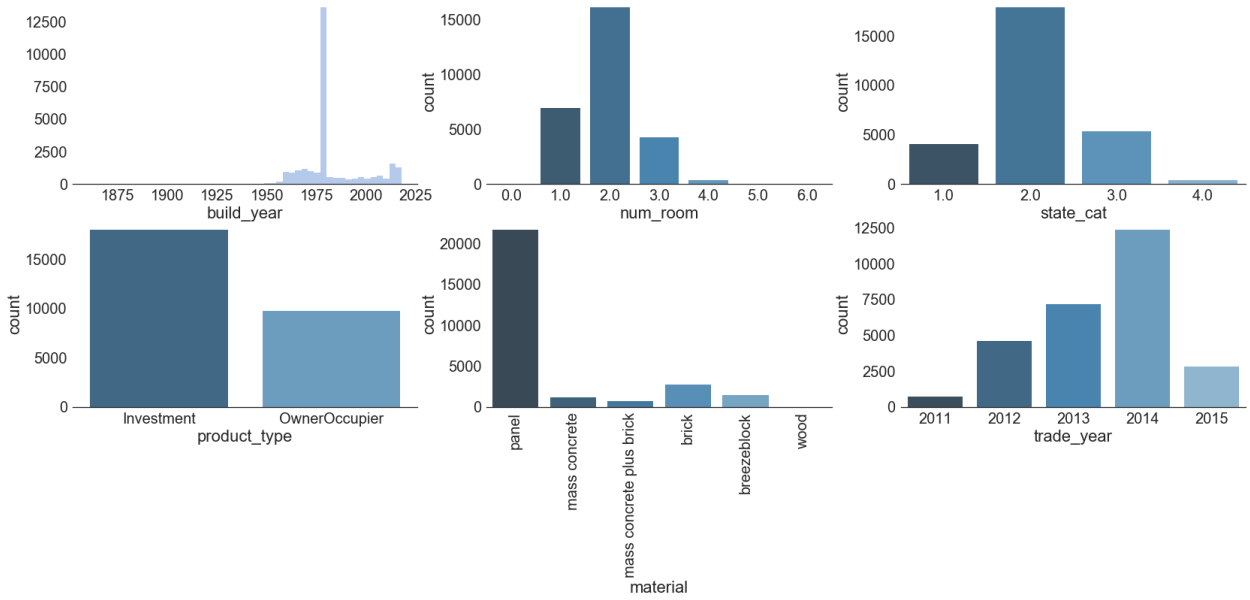
The next step was to construct a correlation matrix. This can be seen in table 1. This reveals that the "full_sq_log", "num_room" and "life_sq_log" have the highest correlation with "price_doc_log". But as these variables are likely to correlate with each other, using only these three variables for building a model to predict the prices might not be optimal.

To better understand the relation between the dwelling specific variables and the price, scatter plots and violin plots are constructed. These can be seen in figure 4 and 5.

The scatter plots shows how price tends to increase as size of the dwelling increases. Further, price appear to increase as both "floor", "max_floor" and "build_year" increases. It seems logical that larger residences command higher prices. Higher floors might allow more sunlight to enter the residence. But it is not intuitively clear if the "max_floor" should cause price to increase. If newer residences are of higher quality, rising prices as "build_year" increases seem plausible.

For the violin plots, it appears as if price increase as "num_room" increases, as "state_cat"

Figure 2: Distribution of categorical variables



(where state is forced to be a categorical variables instead of a numerical) increases and as "trade_year" increases. Except for "trade_year", the rising trends seems intuitively correct. The slight tendency towards higher prices across the years could e.g. be driven by inflation or an economic upturn.

3 Methodology

3.1 Algorithms and Techniques

To solve the problem of predicting house prices, models for regression was needed. Due to the possibility of non-linear relationships in the data, ensemble models for tree regressions were chosen. These models build an ensemble of regression tree and use them for forecasting. They naturally find non-linear relationships in the data, while the ensemble structures imply they have a smaller variance and are more robust than a single tree.
An alternative could e.g. have been linear models with feature selection, such as LASSO, Ridge regression or Elastic Net. The advantage of these models are that they are easy to interpret. But as this task focuses on prediction and not interpretation, the ensemble methods were chosen.

3.2 Splitting data into test and training

25 percent of the preprocessed data is used as a training set, while the remaining 75 percent is used for training and validating models. In the later section on models, all reported errors RMSLEs are the error on the test set. The test set is used to estimate how the different models would perform on new data it has not been fitted to.

Figure 3: Correlation Matrix

Figure 4: Scatter Plots

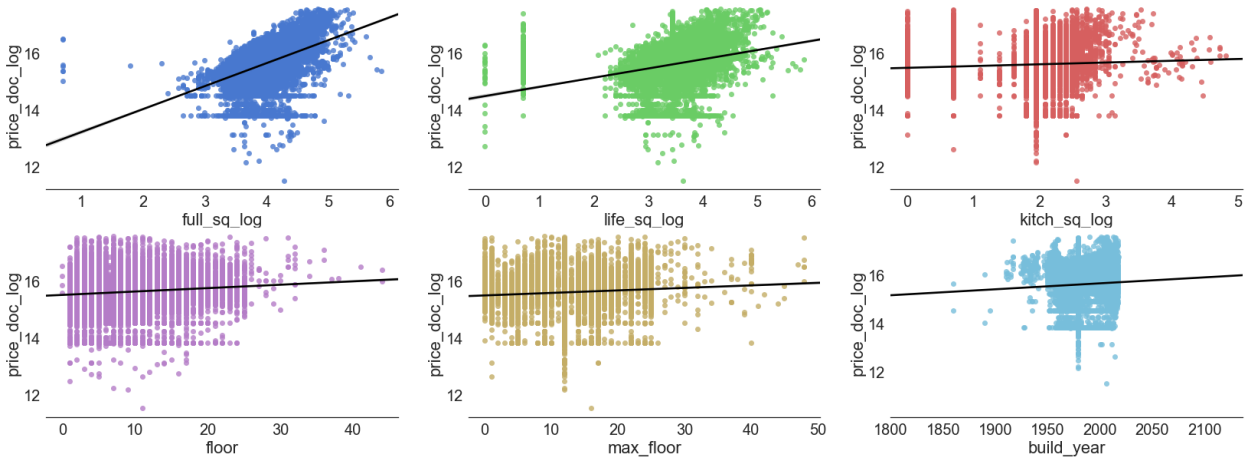


Figure 5: Violin Plots

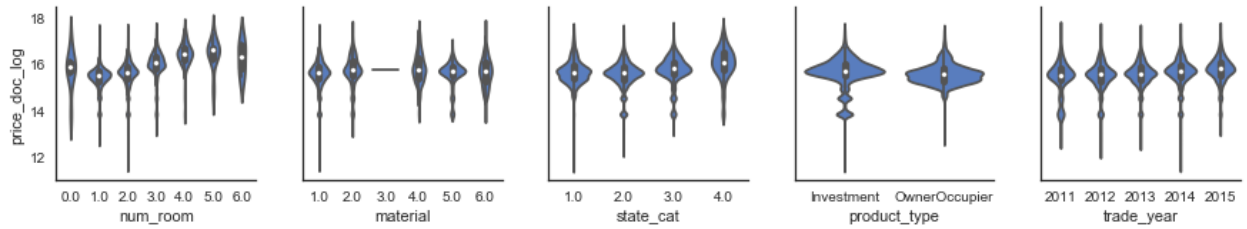


Table 1: Dwelling Specific Variables

Variable	Description
price_doc	sale price (this is the target variable)
full_sq	total area in square meters, including balconies and other non-residential areas
life_sq	living area in square meters, excluding 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 (4 is best, 1 is worst)
product_type	owner-occupier purchase or investment

3.3 Benchmark Model

To have a benchmark for future models, the RMSLE one achieves on the test set by guessing all prices to be equal to the average of the prices in the training set was calculated. This was equal to 0.5967, and can be seen as a "best guess benchmark".
Subsequently, a Random Forrest Regression model with 50 trees was run on all data, to construct a "model benchmark". A Random Forest is appropriate as it can find non-linearities, deal with all data types and can be applied without any tuning. The model's RMSLE is 0.4873. The RMSLE from all models can be seen in figure 13. The 25 most important features can be seen in figure 6. It appears as if the size of the residence is by far the most important variable. Then a few features describing the area follows, and finally a few economic variables.

3.3.1 Complications during coding process

The variables "state" and "material" were formatted as numeric, when it is probably more truthful to describe them as categorical variables. They were reformated as categorical to take this into account.
The data was packed into two dataset, which necessitated a join to construct the dataset for the analysis.
A lot of "build_years" have suspicious observations below 100. These were all removed. Alternatively, it could have been assumed that any "build_year" below 100 designated a year in the 20th century, implying 10 should be treated as 1910. As this is only a guess, these observations were simply removed.
Formatting charts nicely proved a challenge. A combination of Seaborn and Matplotlib commands turned out to be a powerful combination to achieve charts were information can easily be understood.
To achieve both feature selection and tuning of parameters simultaneously, sklearn's pipelines were extensively used.

Figure 6: Feature Importances from RF Benchmark

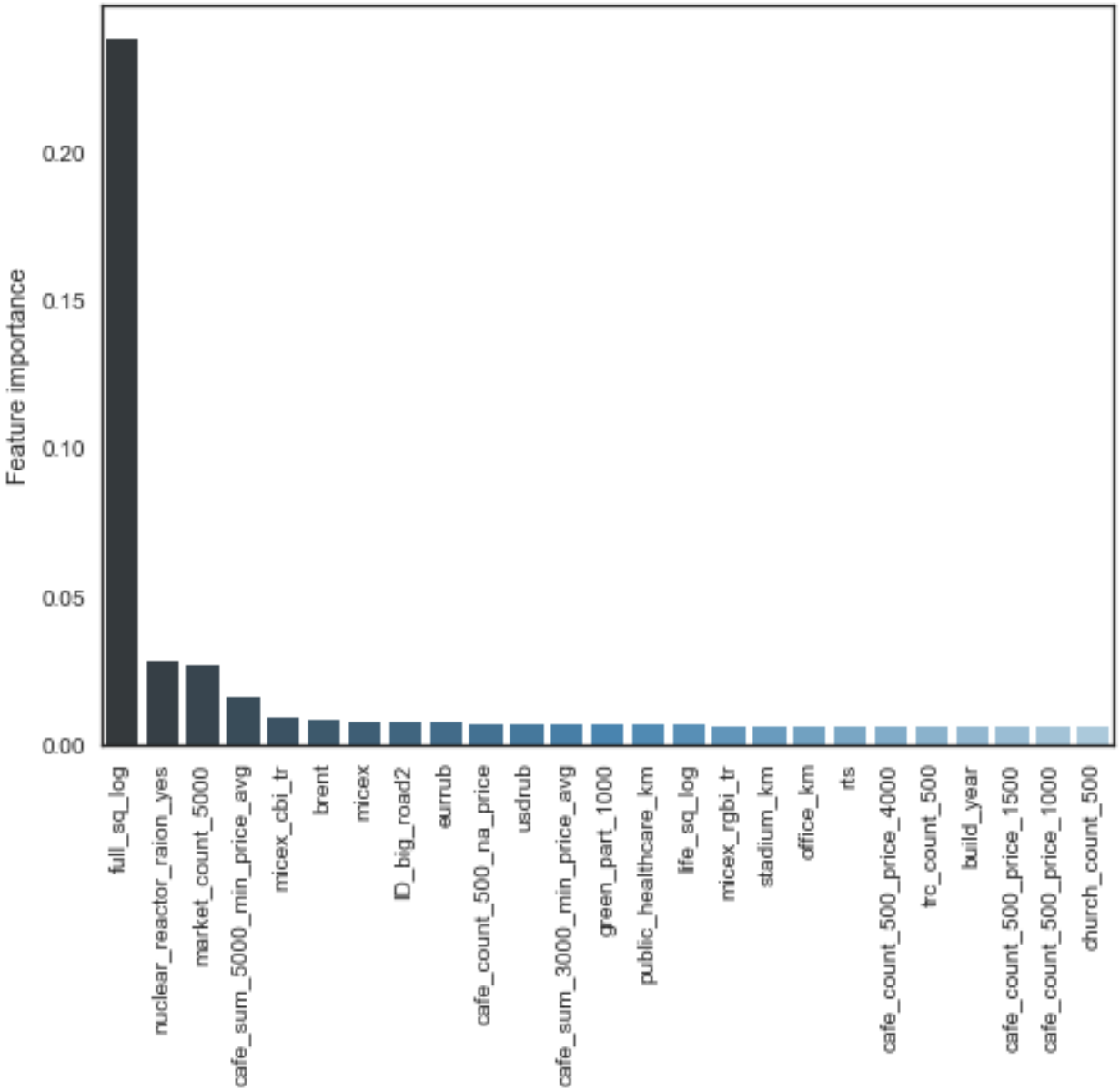
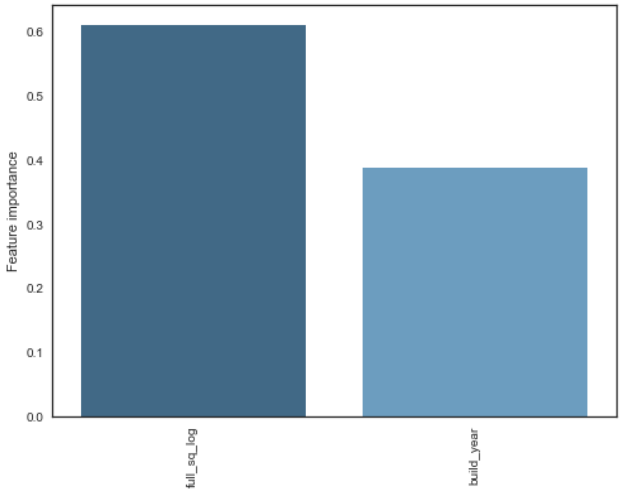


Figure 7: Dwelling Importances



To not constantly retrain models every time new work had to be done on the project, use of the package "pickle" was used throughout the coding process.

4 Results

4.1 Dwelling Model

The next step is to build a model with only the dwelling variables. This is not expected to beat the model benchmark's RMSLE, but is done to achieve a better understanding of the dwelling specific variables. The model used is a Gradient Boosting Regressor (GBR). The learning rate and number of included features is tuned using 5-fold cross validation. Recursive feature elimination is used for choosing features. The best model has a learning rate of 0.1 and only includes two variables. The RMSLE of the best model on the test dataset is 0.515. This is better than the "best guess benchmark" but worse than the "model benchmark". Feature importances can be seen in figure 7. Only two variables have been included in the model: "full_sq_log" and "build_year". This indicates that the other variables related to size of the dwelling are redundant once the full size has been used in the model. The model with only dwelling variables scores worse than the benchmark random forest model. This is likely to be because the benchmark model has a lot more variables available to it.

4.2 Area Model

A model which only uses variables related to the area was also used to predict prices. Again, a GBR is used and learning rate and included features were tuned. The best model includes 152 variables and has an RMSLE 0.5491. The fact that the cross validation chooses such a large subset of the variables is an indication that there is no variable which clearly captures the desirability of an area. The model performs better than the "best guess benchmark"

but worse than the "model benchmark". This indicates that the area variables contain some information on housing prices, but far from all. The 40 most important features can be seen in chart 8. The individual importances appear to decline slowly, and no feature stands clearly out as the most important.

4.3 Macro Economic Model

Another model, only using macro economic variables to predict house prices, was tried next. Again, a GBR is used and learning rate and included features are tuned. The best model includes 32 variables and has an RMSLE 0.5906. This is hardly better than the "best guess benchmark" and indicates that the economic variables are bad at predicting housing prices, at least when only using them. The variable importances can be seen in 9.

The most important variables covers stock indices, exchange rates, inflation measures and the oil price. From an economic point of view, these make sense as they are very indicative of the general economic environment.

4.4 Naive model on all data

This leads to the "total model", which includes all variables. This is the "brute force" way of mining the dataset for information. The model is not tuned, but merely fit to the data. The model achieves and RMSLE of 0.4778, thus beating all other models. The top 25 most important features can be seen in figure 10. Like the Random Forest Benchmark "full_sq_log" is by far the most important model. But the GBM appear to focus more on area variables than the benchmark.

4.5 Tuning the naive model

The final step is to tune the naive model. This is done by selecting the top 100 most important features, and implement a cross validation routine which includes feature selection and tuning of "max_depth", "subsample" size and "min_sample_split". The number of trees and learning rate were not tuned. They had been part of earlier cross validations and found to consistently be optimal at or close to a learning rate of 0.05 and 100 trees. The final model achieves and RMSLE of 0.4854 and feature importances can be seen in 11. The model only includes 10 features, but scores worse than both the Naive Model and the Random Forest Benchmark.

To investigate if a better model can be found, the feature selection was dropped and an extensive cross validation carried out and the top 100 variables from the naive model was kept. A model with 100 trees achieved an RMSLE of 0.4814. Feature importances for the 50 most important features can be seen in 12. This can still not beat the Naive Model, but does outperform the benchmark model

4.6 XGBoost

As an alternative, an XGBoost model was estimated on the preprocessed data. Like the earlier Naive Model, it was simply applied to all data, and actually achieved the best RMSLE

Figure 8: Feature Importances from Area Model

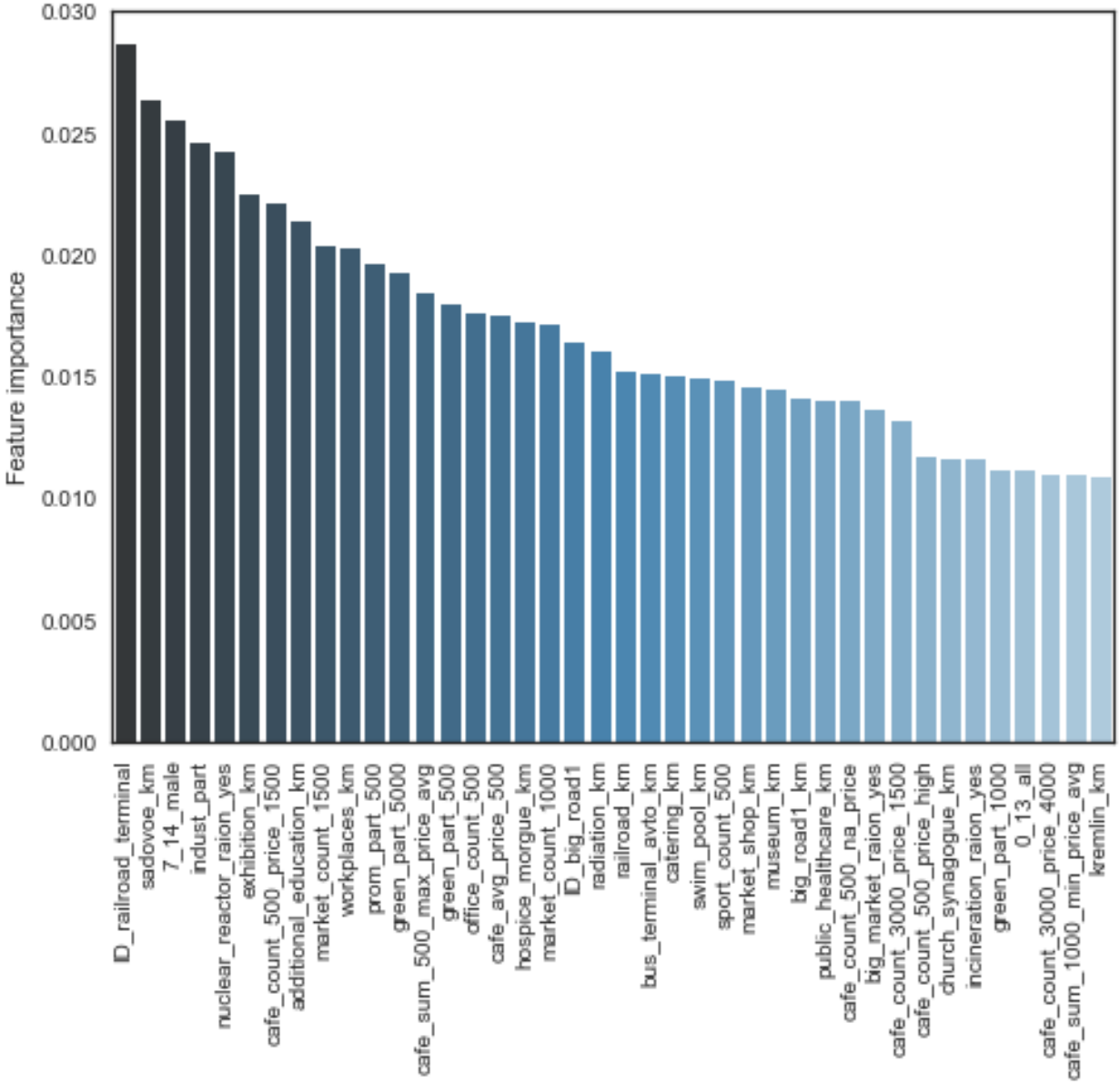


Figure 9: Macro Economic Importances

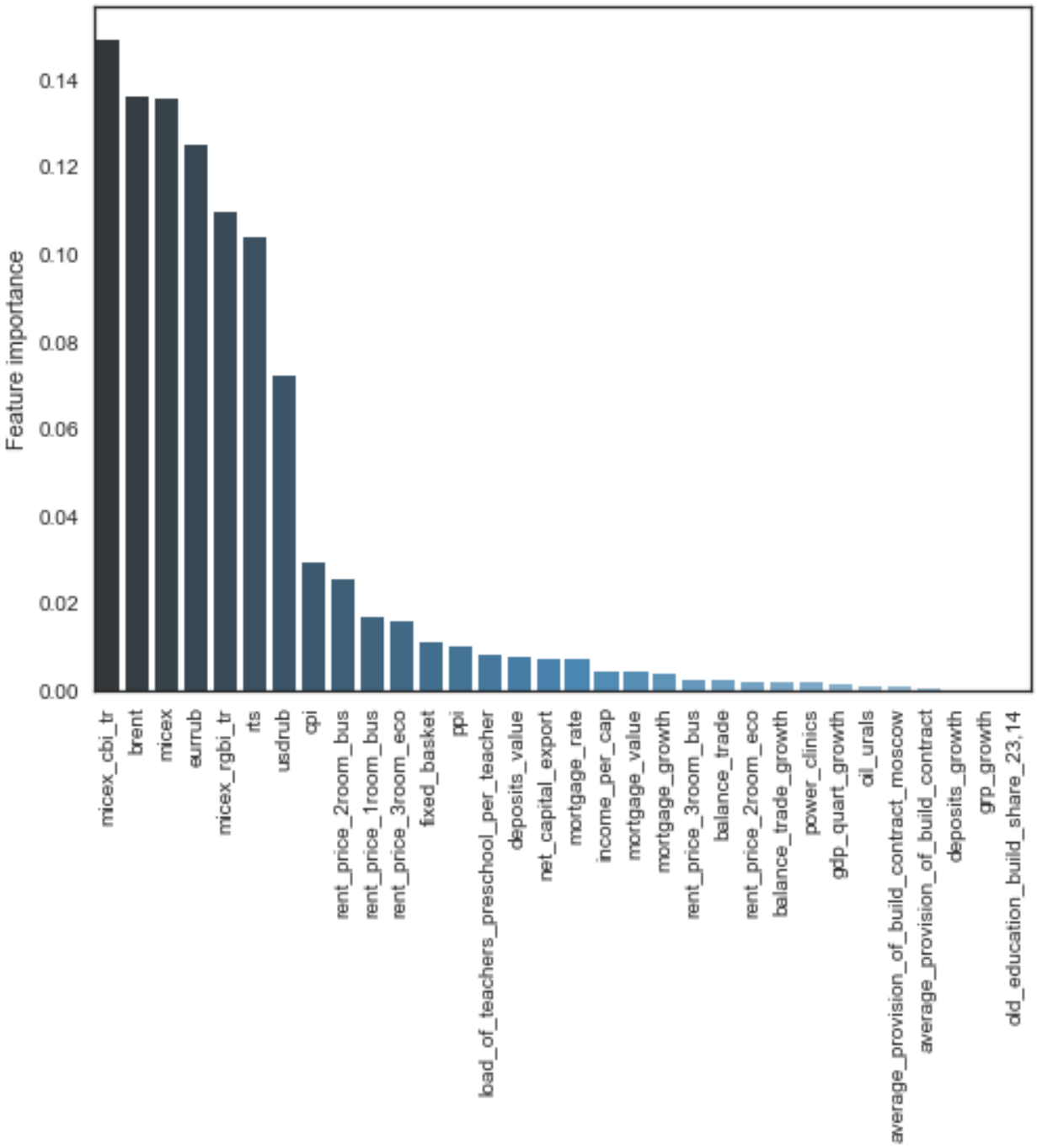


Figure 10: Importances from Naive Model

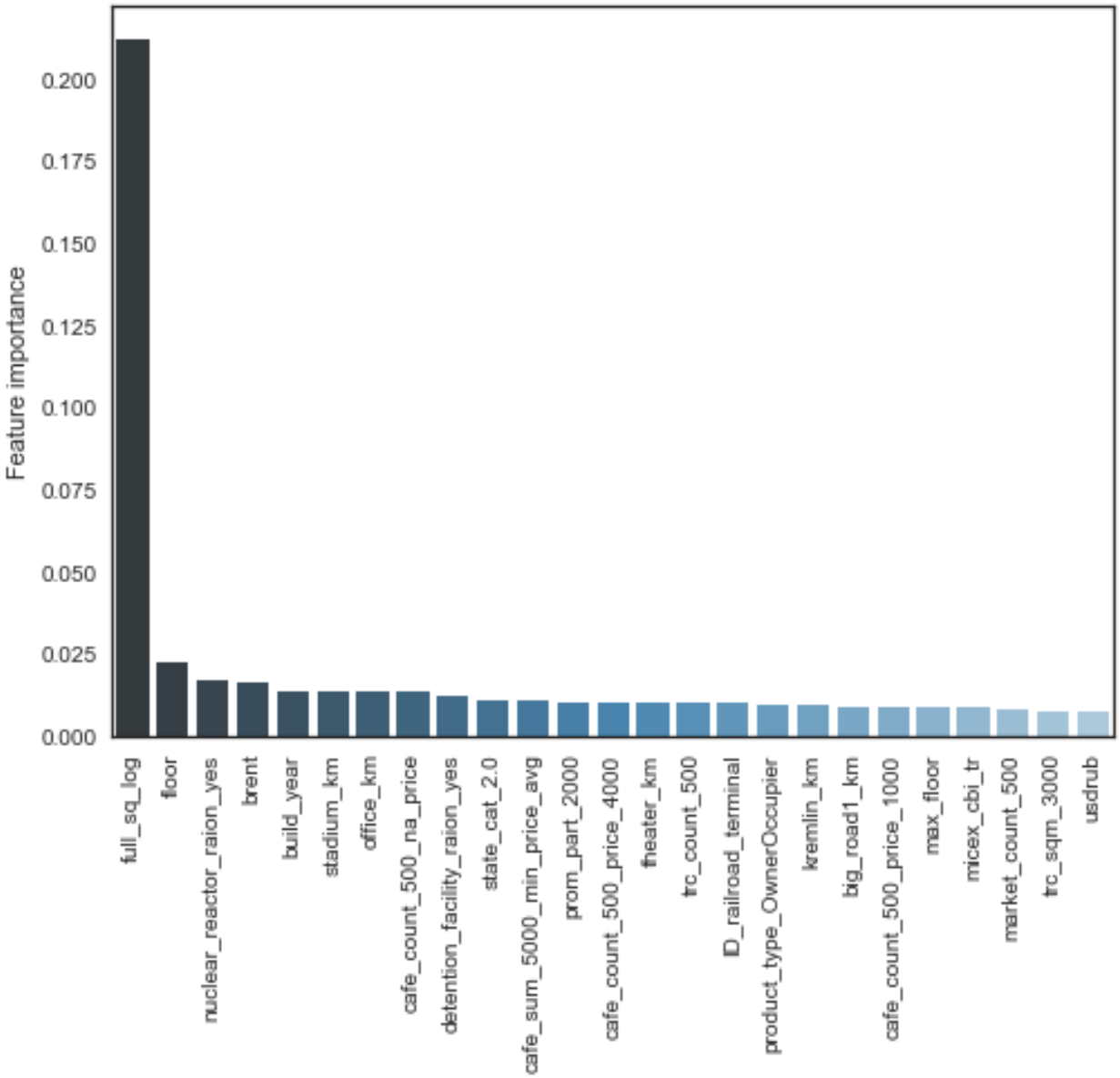


Figure 11: Importances from First Improved Model

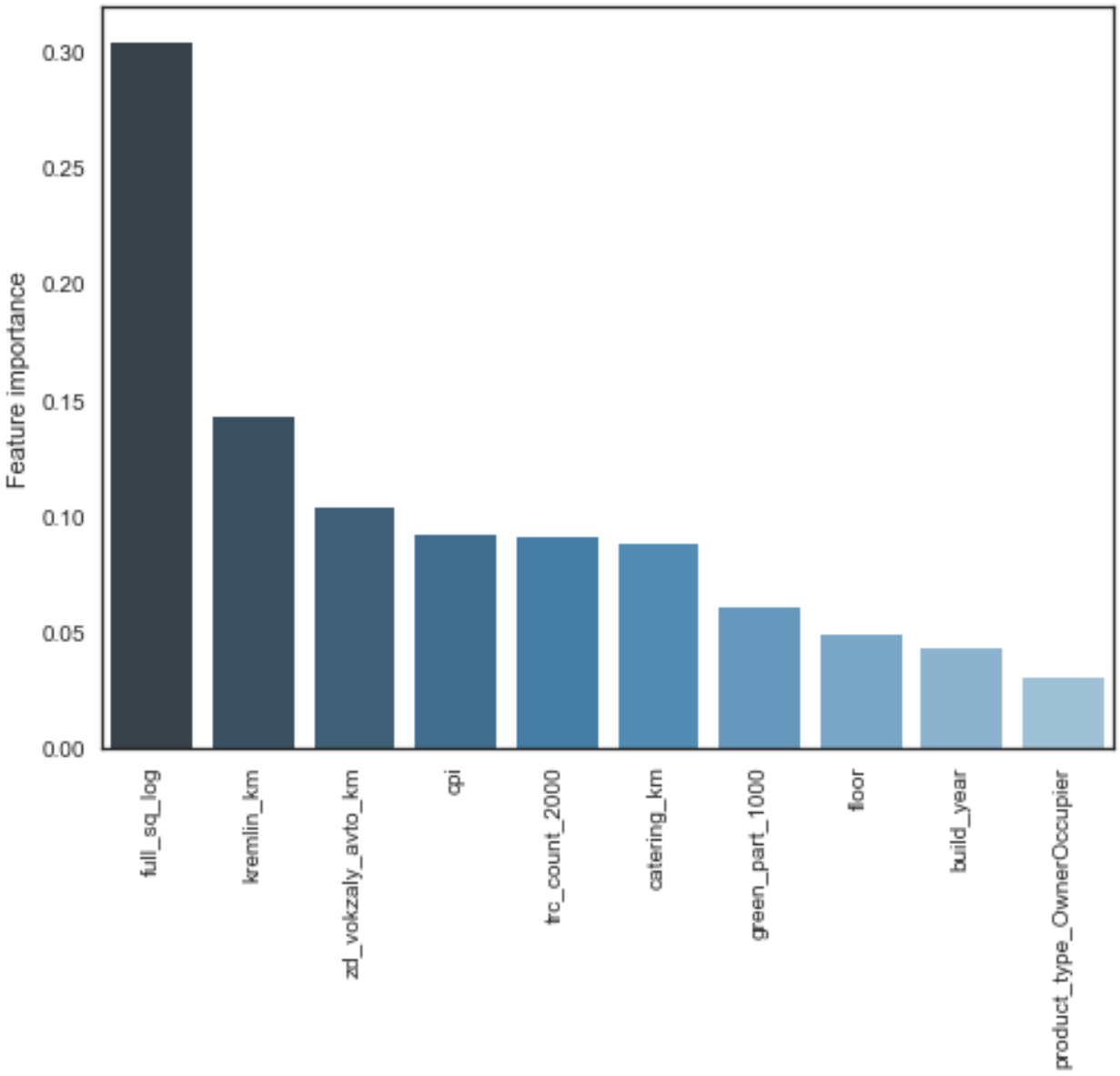
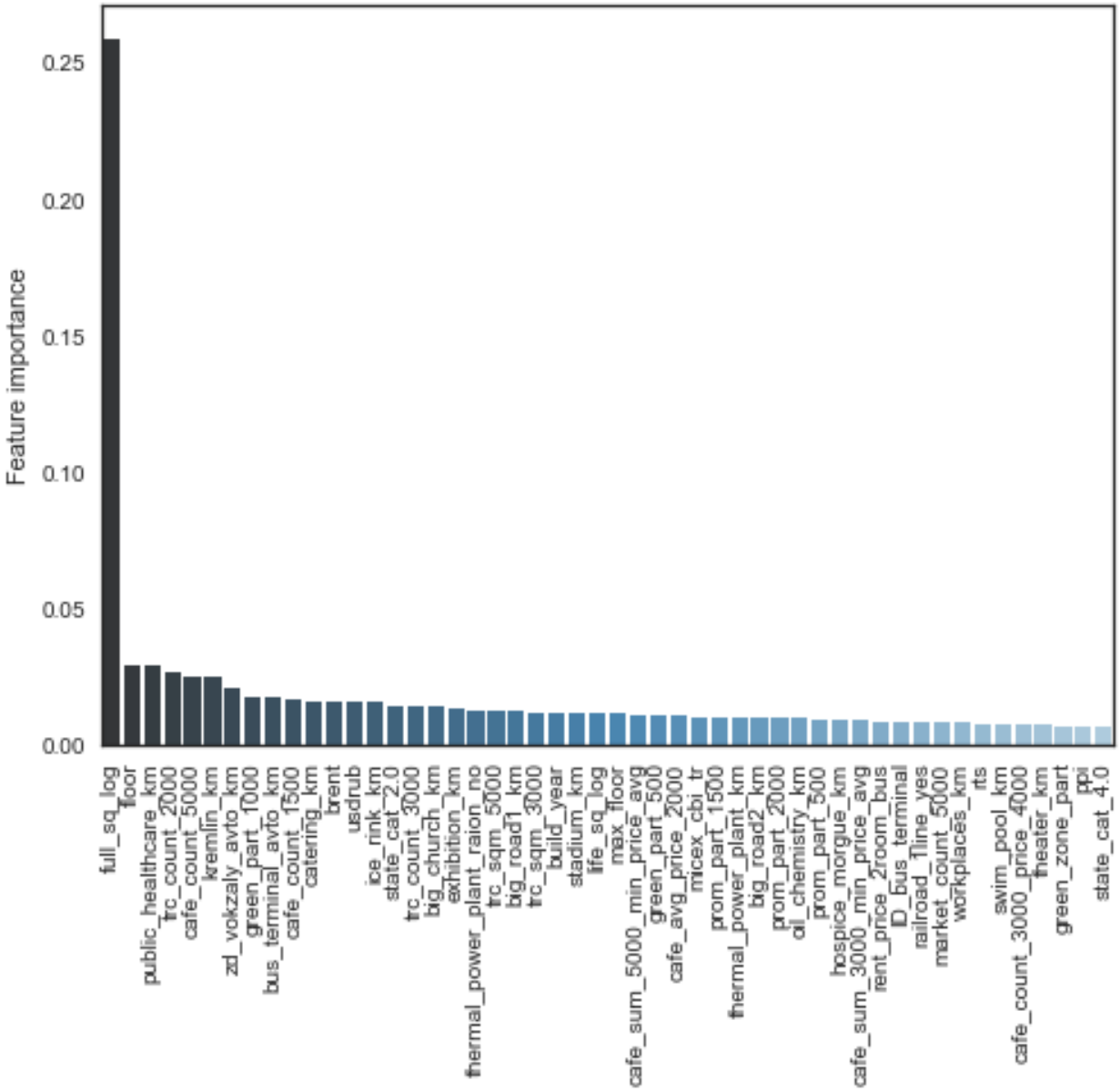


Figure 12: Top 50 importances from Second Improved Model



of all model, at 0,4768. The difference between the RMSLE for the two models is on the scale of 0.001, while their RMSLE's outperform the benchmark mode by approximately 0.01. This indicates the XGBoost Gradient Boosting Regressor and the Sklearn Gradient Boosting Regressor are about equally well at predicting the data. But the training time for XGBoost was significantly faster, taking about half the time of the Sklearn GBM.

5 Conclusion

5.1 Comparison of different models

The RMSLE from all the tested models can be seen in figure 13. It is noticable that the tuned model only just outperforms the benchmark Random Forest model.

The size of the residence denoted by the variable "full_sq_log" appear to be the most important variable in the entire dataset. Of all the other dwelling variables, only "build_year" and "floor" seem to truly be of consequence.

Variables describing the neighborhood are clearly important, but it does not appear as if any small subset of area variables can be used to easily extract the desirability of a neighborhood. The variables related to the number of cafes are clearly important, but also variables describing e.g. distances to cultural institutions or Kremlin show up as important in the models.

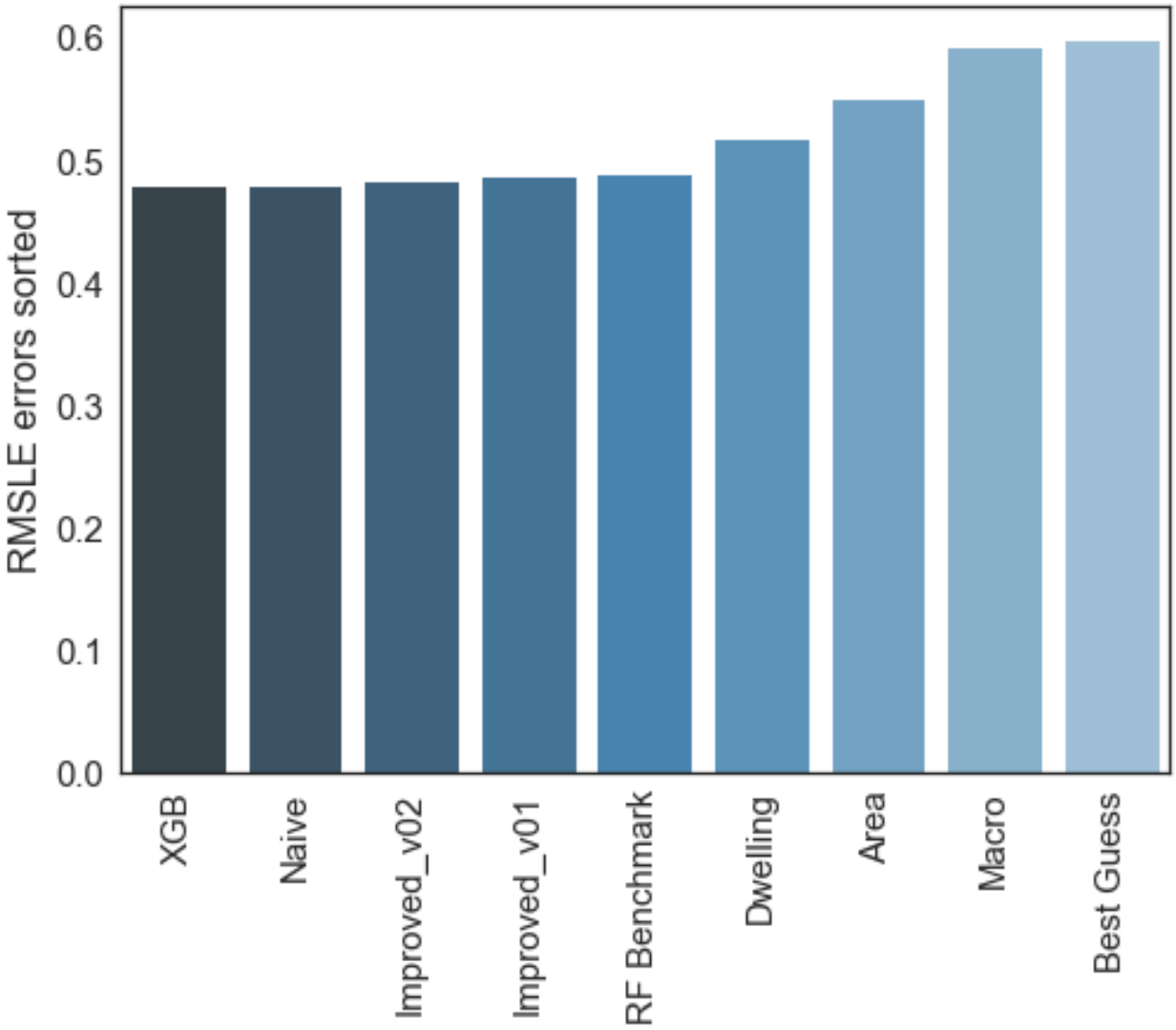
Economic variables appear to be less important than dwelling or area specific variables, but still have some impact. The most important appear to be the oil price, inflation and the volume of household deposits. As these variables are the same for all observations at a given point in time, they are likely to capture the "trend" on housing prices. This is both a strenght and a weakness. On one hand, it makes sense that for a country somewhat dependent on oil prices (note that many gas contracts contain gas prices indexed to the oil price), the oil price can be seen as an indicator of the current and future economic environment. But when you include more than a hundred variables in the dataset were every one is the same for all observations with the same timestamp, any time varying trends in the data is likely to be caught by one or more of the economic variables, simply by pure coincidence. But since it might be a coincidence, it might not generalize to predicting the price of housing in the future.

So to sum up: the best model were the Naive Model which used all variables and achieved an RMSLE on the test dataset of 0.4778 beating the Random Forest Benchmark Model with 2 percent better RMSLE. The best model mainly relies on information about the residence itself, subsequently the area and finally the economic environment.

It is thought provoking that the Random Forrest performs so well. The reason could be that the high number of features, and especially the high number of features needed to adequately describe a neighborhood, are simply handled much more meaningfully by the forced feature randomness of the Random Forrest.

But as the best models still outperform both the "best guess benchmark" by 25 percent and the model benchmark by 2 percent, the conclusion is that the Naive Model satisfactorily solves the problem of predicting Russian housing prices.

Figure 13: RMSLE for all models



5.2 Summary of the end-to-end problem solution

This capstone project set out to build a model for Russian housing prices. First, the available data was thoroughly analyzed and cleaned. Then, models for different subset of the data was created and scrutinized. Following this, models using all the data was implemented. These naive models turned out to perform much better than the models trained on subset of the data. It also appears as if models needs many input to correctly utilize the neighborhood specific information.
The final and best model was a Gradient Boosting Regressor, a tree regression ensemble model with boosting, that naively uses all features. The feature importances from both the final model and the intermediate results all indicates that the most important feature is by far the size of the residence being traded. Both the information specifically about the residence and about the macro economic environment could be summed up easily by quite few variables. But the information describing the area did not appear to be easily summarized by just a few features.

6 Reflections and future improvements

It appears as if the size of the dwelling, its build year and the floor are enough to adequately use the information in all the dwelling specific variables. The macro economic environment appears to be well described by a few variables (stock market, oil price, inflation and exchange rate). Looking into a clever way of describing the desirability of the neighborhood of the dwelling would very likely improve the precision of the model.
Improving the handling of missing observations also seem to be pivotal. Currently, missing values are replaced with the median or the mode. Using a K-Nearest-Neighbors could be a solution. A state of the art solution could be to find the relevant neighbors by measuring which variables in the dataset are most correlated with each other, and use this information to choose the neighbors.
Looking into changing the handling of outliers is also an option. With better handling of missing data and singling out of the most relevant variables, it would be easier to plot and analyze each variable, instead of relying on the automatic outlier removal used in this project. With the important variables singled out, the entire analysis and handling of missing data and outliers could be rerun. Using only this subset of the variables, it would be feasible to clean the data more "by hand", i.e. without relying on automatic scripts for cleaning away most outliers.
During my work on this project, I became aware of the XGBoost package. Using this package would likely improve the runtime of the models used in this project.

References

[1] Laurence S. Moss. The Laws of Human Relations and the Rules of Human Action Derived Therefrom. By Hermann Heinrich Gossen. Translated by Rudolph C. Blitz with an introductory essay by Nicholas Georgescu-Roegen. Cambridge: M.I.T. Press, 1983. Pp. 460. USD 47.50. *The Journal of Economic History*, 44(04):1130–1132, 1984.