Université Paris-Sud
EDF Lab Paris-Saclay

Statistics and Machine Learning Program

Project Report

Prediction of Housing Prices in Russia

Marina Gomtsyan and Hugo Marival

Supervisor: Yannig Goude

Paris
2020

Contents

1 Introduction

Accurate forecast of housing prices is of great interest to different real estate stakeholders such as house owners, buyers, investors, and agents. For house buyers a good house price prediction can help to search for candidate houses that are suitable to their financial capabilities. House owners can use predictions to be aware of the market and find the best opportunity for selling their houses. Additionally, real estate agents can help their customers to find out market trends.

Motivated by the importance of housing price prediction problem, in this project we will use various machine learning and statistical methods to forecast housing prices of Moscow, one of the largest cities in the world.

1.1 Dataset Description

The dataset is taken from a competition in Kaggle, the aim of which is to predict the sale price of each property. The dataset consists of training and testing data. The training data ranges from August 2011 to June 2015, and the testing data from July 2015 to May 2016. Since the testing data does not contain the target variable, in our experiments we use only the training data.

In the training data there are 30471 observations of transactions on the Russian housing market. Each entry contains general information about the transaction (unique identifier id, a timestamp and the target variable price), as well as 290 other variables giving an extensive description of the house/apartment and its neighbourhood. Moreover, there is a dataset called "macro.csv" with information on Russia's macroeconomic indicators for each date. The following link gives access to the full description of the dataset : http://tiny.cc/d40pkz.

In our preliminary analysis, we can break down the variables into four main categories:

- General information about the transaction – unique identifier, timestamp, transaction price (target variable).

- Estate features – description of the property (location, area in square meters, property condition, build year, etc.). Those features seem to be the most relevant to predict the price and only represent 11 columns out of the 290. Hence, we will be able to focus very specifically on them during the feature selection and data cleaning phases. Also, we will be able to conduct analysis to understand their true relevance with respect to price prediction. They will often be referred to as internal features.

- Neighbourhood features – 279 variables giving an extensive description of the neighbourhood of each property. Some features are constant across each district, including information about the population of the district, green zones, industrial zones, transportation, and cultural life. For our exploratory analysis, 279 columns is a lot of information, and it appears that the information they provide is sometimes too fine-grained or heavily correlated. Therefore, we created macro-categories, aggregating some of those columns together to have a better overview of the data.

- Macroeconomic indicators - a series of macroeconomic indicators for each day, covering the time range of the transactions in our dataset. Initially, we thought that they would provide very useful information for the price prediction, as the Russian economy relies very heavily on gas and oil exportation, which could be correlated to housing prices. The Russian crisis of 2014 could also be factored in using those indicators. However, it turns out that these data actually had almost no influence on our predictions.

2 Exploratory Analysis

In the data exploration phase, we analyze the dataset in order to:

- Assess the quality of the data and understand which variables can reliably be used and which ones will simply add noise to the model as they are either wrong (worst case) or have too many missing values.

- Get a rough understanding of the relevant variables with regard to our target.

- Find correlations between variables, which would then allow us to reduce the dimension of the problem by aggregating them into fewer less correlated variables.

We also found it relevant to gather outside and general information about the Russian housing market from sources like economic reports, etc. to get a grasp of the real situation of the market and not only the one described by our data. As we expect our data to reflect some of those behaviours, this can be viewed as a way to assess the quality of our data and get a hint to possible correlations between the covariates. For example, observing the opposite behaviour for some variables can either tell us that the data is very skewed, which we can take advantage of, or that we should be suspicious regarding the correctness of the variables at stake.

2.1 Methodology

Our approach is twofold:

1. As our goal is to predict the price, we started by plotting the evolution of the price depending on some variables of interest according to our basic intuition, and with the help of the data dictionary provided with the dataset that contains a short explanation of each variable, combined to our intuition. Those plots include the price per district, build year per district, average size of a property per district, etc. We also used the descriptive functions of R to compute some basic statistics on the data such as mean values and standard deviations of variables of interest and look for outliers, check the number of missing values etc.

2. To accompany this analysis, we also exploited feature importance and ALE Plots obtained on simple linear models and an untuned xgboost in order to guide our analysis. As the number of explanatory variables is quite high (290), we grouped them into categories by hand and selected the top PCA components in each category as a way to aggregate those variables. We then fit another XGBoost on this new dataset to get a more high level view of the feature importance, while also keeping our previous detailed version. Those plots also helped us perform further checks on the most relevant features, which was a crucial step as it allowed us to detect that some key variable values were actually very misleading.

2.2 Different dynamics per district

Moscow is the capital and the largest city in Russia and 10th most populous city in the world. It is major economic, cultural, and scientific center of Russia and Eastern Europe. Currently, Moscow is divided into 12 administrative district (Figure 1). Until 2012 there were only 10 districts. Troitsky and Novomoskovsky are relatively recently newly joined districts.

We note that the various districts have very different dynamics regarding the housing market parameters depending on the district. First, it can be noted that peripheral districts like Novomoskovksy or Troitski tend to have low transaction prices, whereas we experience high transaction prices in the central ones (mostly named after cardinal points). It is not surprising since the price

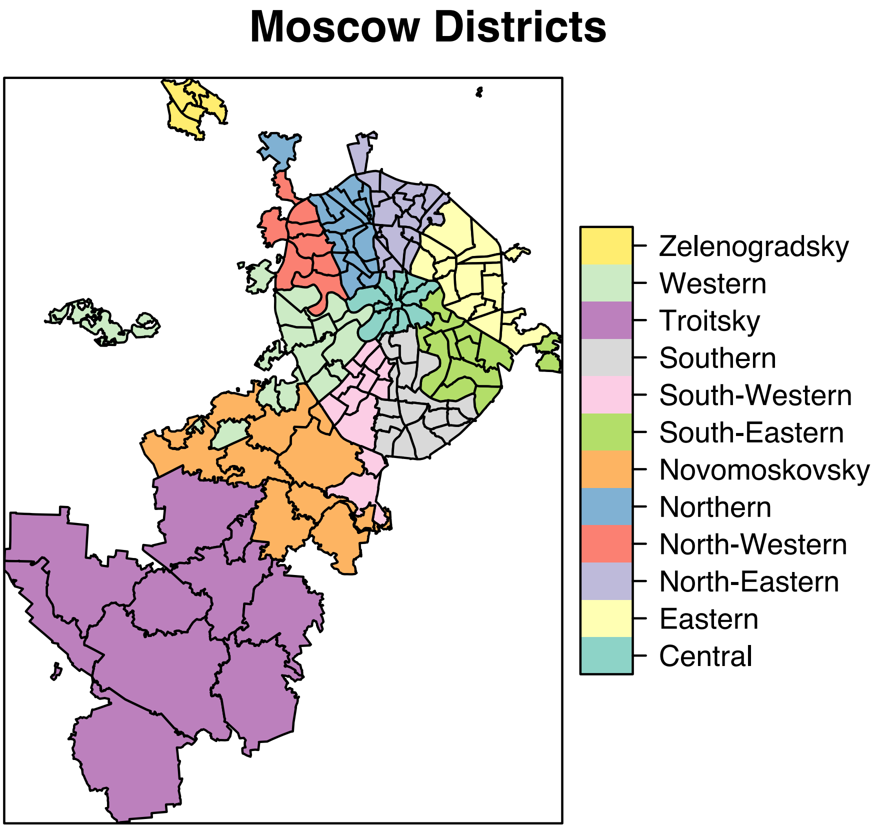


Figure 1: Administrative division of Moscow

of real estate tends to be higher closer to the center in most big cities. Moreover, unlike the old "intra muros" districts, the peripheral districts are new arrivers in the Moscow administrative divisions. We note that Novomoskovsky and Troitski tend to attract owner occupiers and be unattractive for investments. This is coherent with our parisian intuition that an appartment is easier to rent in the center than in the "banlieue". Figure 3b is also interesting since it shows that the size of the properties in those districts is larger than in the more central ones, probably due to the fact that those districts are large and have a low price per square meter. It may indicate that they are more prone to family life than the small and expensive districts in the center, and can be relevant when studying the influence of neighbourhood parameters on the price in those districts : proximity to schools, presence of green spaces, etc. may be of greater importance than in the center, and variables such as the distance to the metro for example may be less relevant as families tend to have their own car.

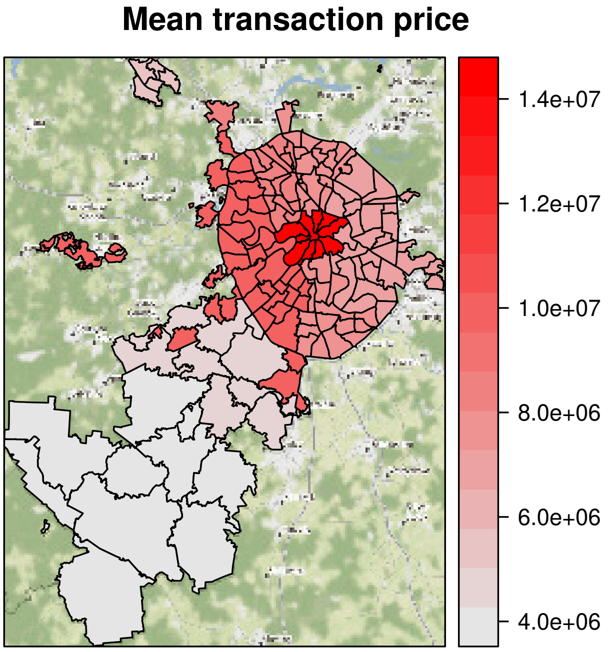
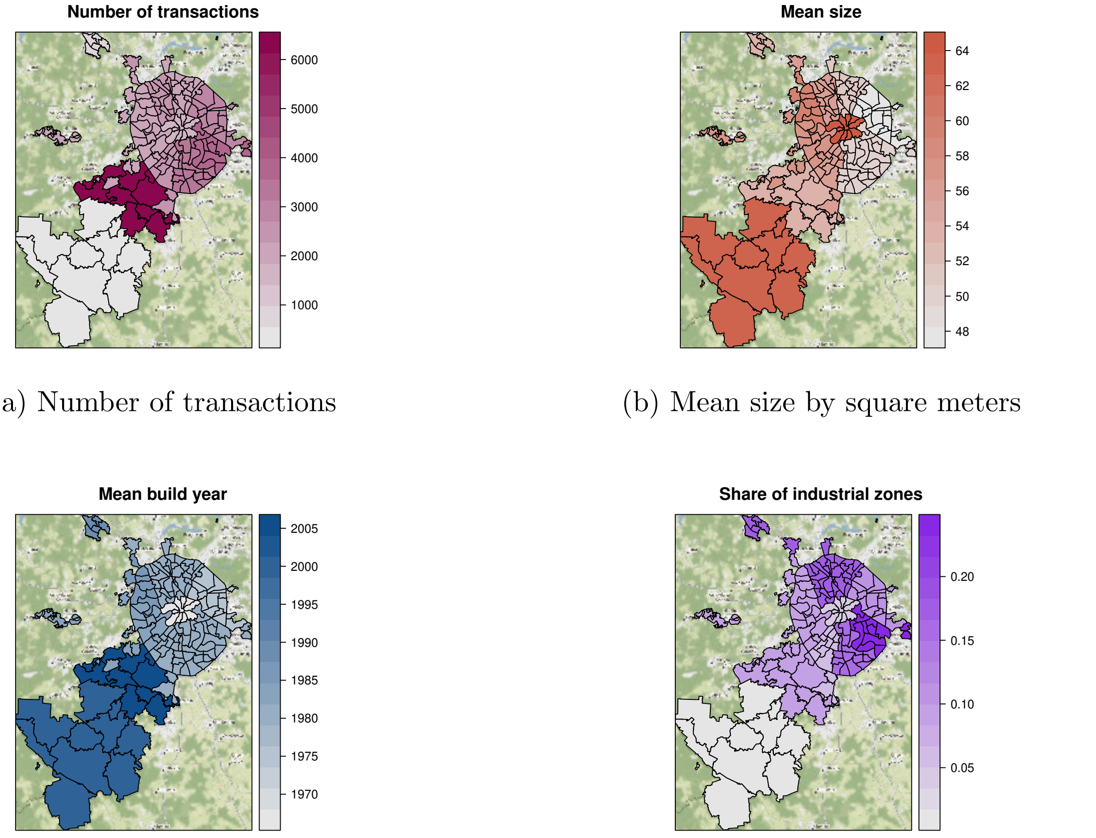


Figure 2: Mean transaction price by district in rubles



(a) Number of transactions

(b) Mean size by square meters

(c) Mean build year

(d) Share of industrial zones by percentage

Figure 3: Illustration of some features by district

Another important pattern that we can see in Figure 2 is that the houses in West of Moscow are more expensive than in East. This can follow from the facts, that currently most economic and scientific centers of the Moscow are located in the Western districts. Additionally, in the western suburbs of Moscow is located a prestigious residential area, where many Russian government officials and successful business people reside. Moreover, during the Soviet period in that residential area were cottage houses of famous Soviet politicians, scientist, musicians, and diplomats. The western suburbs of Moscow are known to have better ecology and higher air quality.

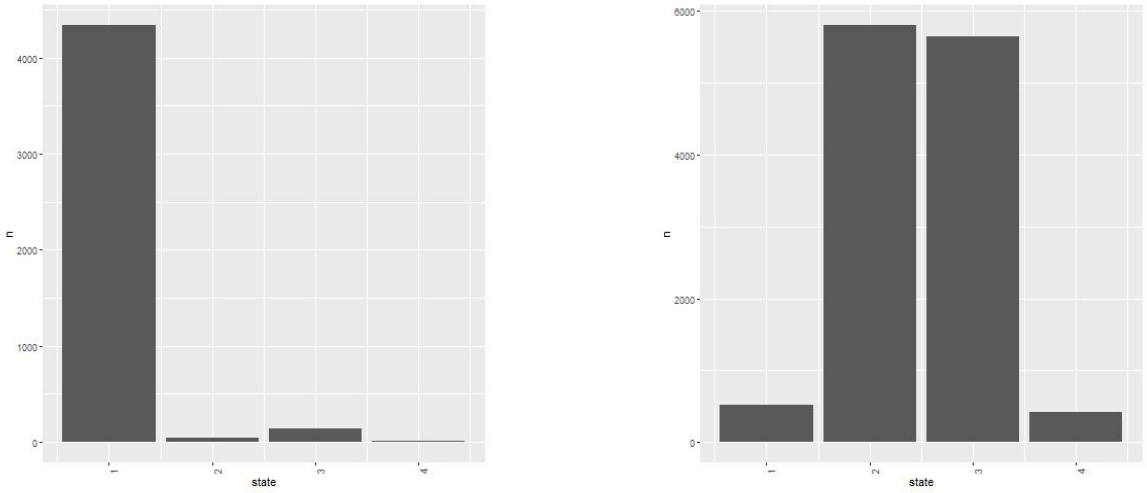
Interestingly, the western areas of many cities are richer than the eastern ones. According to a study[1] conducted in for 70 English cities including London, rich West sides is a consequence of East sides having worse ecology and air. They explain that by the fact that prevailing winds blow from West to East and take industrial smoke with them. As a consequence, industrial zones are more concentrated in East. We think that this might be the case of Moscow too, since according to Figure 3d the share of industrial zones in the eastern part is bigger.

2.3 Development in new districts

As noted in the previous section, the different dynamics between the districts seem to have an influence on the price, and the difference between old and central districts and new and peripheral ones seems particularly relevant. It appears on the heatmap of the number of transactions per district in Figure 3a, that the new district Novomoskovsky is very active in terms of number of transactions, as many new houses are probably being built and sold to the freshly arriving inhabitants.

2.4 Misleading values for product_type

At first, we believed that OwnerOccupier refered to older products owned by private consumers whereas Investment would refer to newer products built by real estate developers and sold to investors. A (too) quick analysis of missing values and data coherency also favored this point of view, as OwnerOccupier type products had more missing values and some suspiciously distributed variables, like the state variable. Indeed, on a scale from 1 to 4, we expect the middle values to be the most common ones, which as we can see from the plot below, is the case for Investment but clearly not for OwnerOccupier.



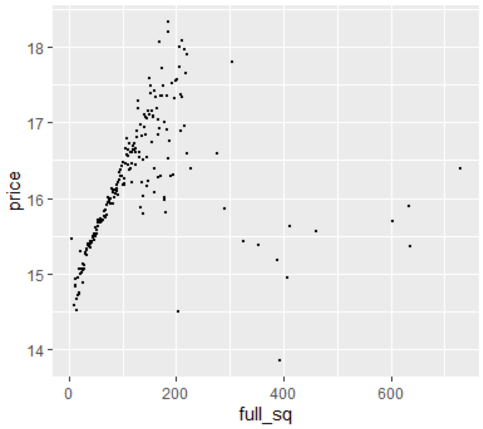
(a) state count for OwnerOccupier

(b) state count for Investment

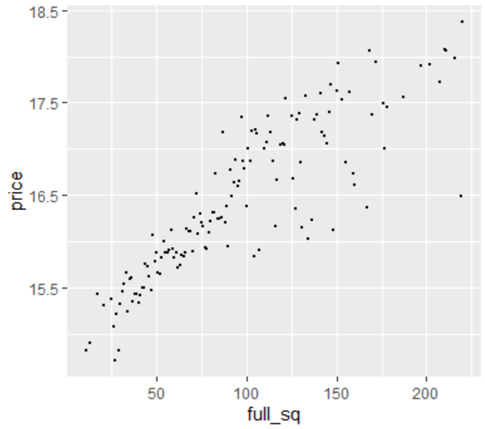
Figure 4: Distribution of the state variable depending on the product type

[1] S. Heblich *et al.*, East Side Story: Historical Pollution and Persistent Neighborhood Sorting, at *Journal of Political Economy*

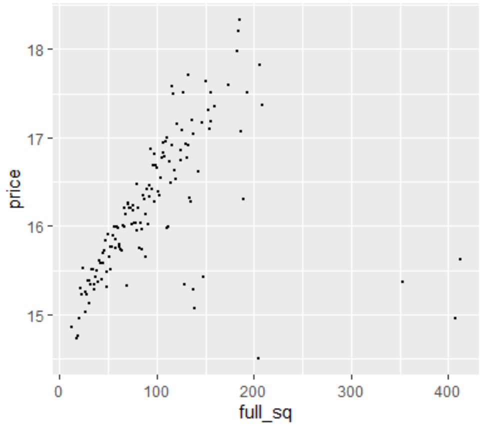
When running our initial models, the feature full_sq indicating the size of the product was deemed the most important feature both for OwnerOccupier and Investment, and the plots below indicate a quite good linear approximation of the price by the full_sq variable :



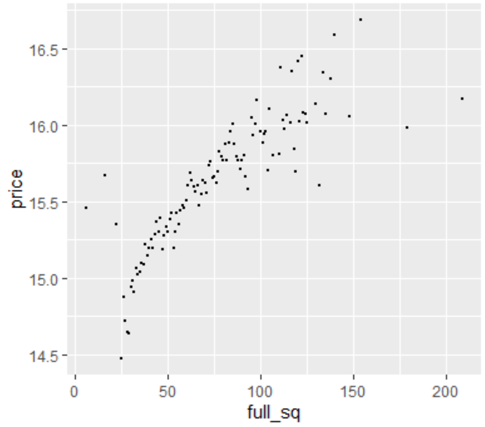
(a) Constant ratio as a first approx.



(b) Central



(c) Western



(d) Novomoskovsky

However, the linear fit seems better for Novomoskovsky and the Western district than for Central, and several other plots confirmed that the new and western districts, towards which the city of Moscow seems to be expanding, have a much better linear relation to the price than the central and eastern ones. This seems counter-intuitive : Investment type products which are more present in the older type of districts are expected to have a more stable price per square meter than OwnerOccupier products. Crossing this with the fact that new districts contain a lot of newly built properties and after searching the internet, we changed our view on those categories : OwnerOccupier actually seems to refer to newly built products sold to private occupiers, whereas Investment corresponds to second or third hand products sold to investors. This explains the owerwhelming majority of ones for the state variable in the OwnerOccupier case, as the buildings are new or very recent. The poor linear fit between the price and full_sq could be explained by :

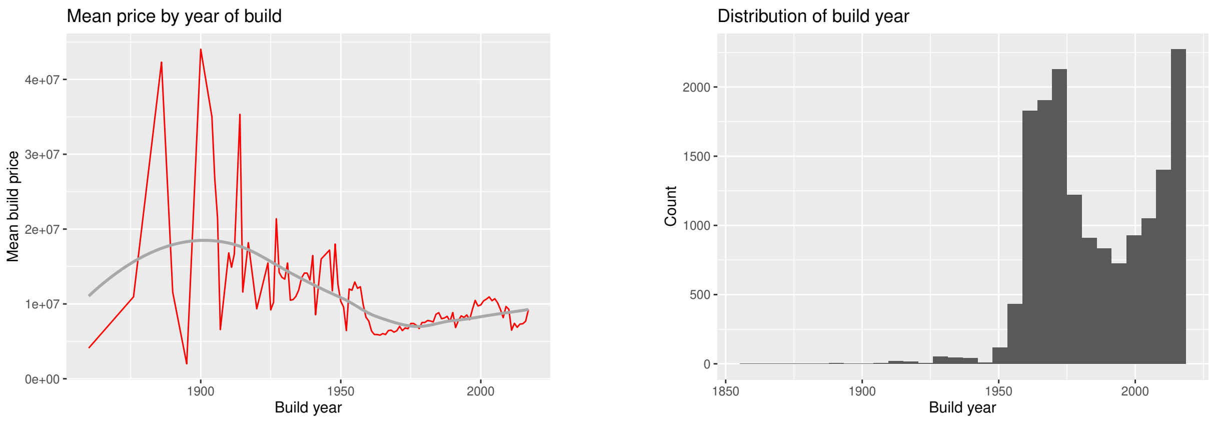
- the fact that more recent buildings have more accurate information,
- due to some over- or under-estimation of full_sq for some Investment type products (tax optimizations for instance)

In fact, the RMSE error for linear models is very different when filtering the data for OwnerOccupier only vs. Investment only. It is also the case for XGBoost, hence we will take this behaviour into account when building our models later on.

2.5 General analysis of the dataset

2.5.1 Build year

In our further analysis we noticed that build_year is an important feature for predicting the price of a property. From Figure 3c the oldest buildings are located in the Central district, while the most recently built houses are located in the Novomoskovksy and Troitski districts. Additionally, the graph of average price of a property by its build year in Figure 6a shows that the prices of older buildings are higher. In the case of houses built before 1950 this can be a consequence of most of the old buildings being located in the city center, as we saw in Figure 3c. Also, from Figure 6b we can tell that very few properties in the dataset were built before 1950. It is important to note, that during the governance of I. Stalin (until 1956) buildings were built using high quality materials and spacious apartments. While after his death there low-cost buildings were built, with smaller apartments and lower quality materials. This can be the reason, why before the period of 1960-1980 properties have higher average price, and after it they start to increase again. We also note that the average price decays in the late 2000s, which can be caused by the fact that houses built during that time are mostly located in peripheral districts.



(a) Mean price by the build year

(b) Distribution of build year

Figure 6: Illustration of some properties of build_year

2.5.2 Size of the appartment

While analyzing the influence of the size, it is obvious that all other parameters being similar, a larger surface will be more expensive than a small one. This can lead to a certain biais in our models since if a district has many transactions for small properties, but expensive with regard to their size, big properties will act like outliers and their price will be underestimated. This can be the case with Troitski for example, which has the lowest average transaction price, but the largest houses. Novokovski, also has very low transaction prices but quite large houses. This lead us to introduce a new variable corresponding to the price per square meters, which can be used as an auxiliary target variable to predict the price. Correcting this biais seems interesting, since Novomoskovski for example totalizes around 20% of the transactions.

2.5.3 Neighbourhood factors

Although each district seems to have its own dynamics, we believe that part of the price is more attached to some neighbourhood features than to the district. Feature importance plots come in quite handy to identify the most relevant neighbourhood indicators driving the transaction prices, as we can see for instance, on the global feature importance graph, that variables such as cafe_count_xxx_price_xxx are part of the top explanatory variables. The PCA feature importance chart confirms this idea since the expensiveness variable is third in the feature importance ranking. Indeed, this variable gives a good indication of the wealth of the neighbourhood, which is quite unsurprisingly linked to the transaction prices : rich people buy expensive houses ! Other factors come up as quite relevant, such as

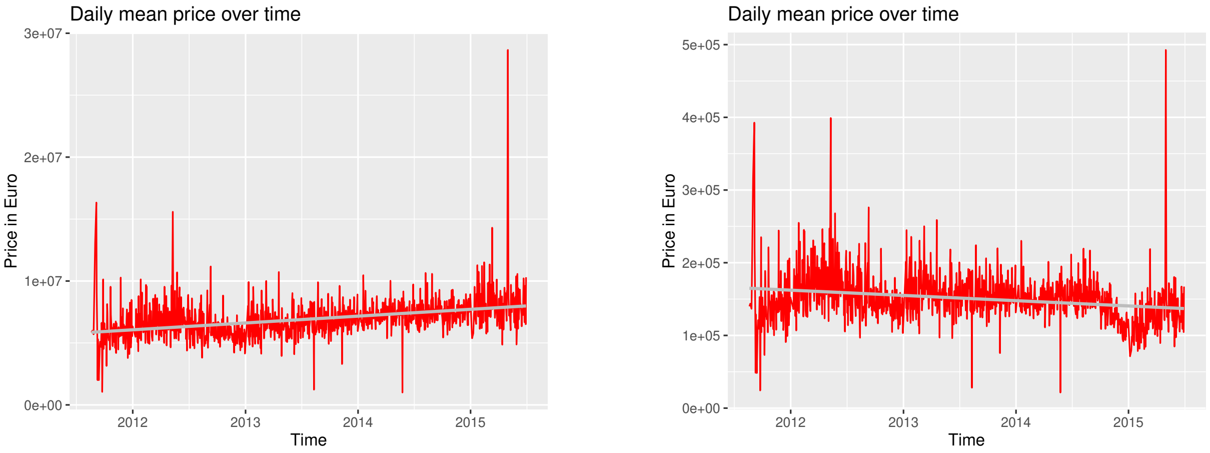
- work : in fact, we will see later on that the product_type variable is very important, not only because of its correlation to the transaction price, but also since the data coming from investment sources is much more detailed and accurate than when it comes from owner occupiers. Work zones tend to have few owner occupiers, leading to an overall higher quality of the data for these zones.
- religion also comes up as a variable of interest. It is probably related to some socio-economic factors linked to certain religious communities. Much like the expensiveness variable, it can act as a proxy for the global wealth of the neighbourhood, as there are links between socio-economical classes and religious minorities. Also, some religions have a more active practising community than others, which can explain the importance of variables linked to the proximity to their place of worship.

We will later on study the influence of neighbourhood factors independently from the district, with the assumption that prices are correlated for similar neighbourhoods. It will also motivate the use of a k-nearest neighbours algorithm to impute missing values.

2.5.4 Influence of the macroeconomic factors

As the internal data combined to the neighbourhood features allows us to understand the "value" of the property on the scale of the housing market, macroeconomic data might be quite relevant to determine this scale. Assuming that economic growth has a positive effect on housing market prices, we can try to identify among those macroeconomic parameters, which are the ones that are the most significantly linked to general economic growth as an initial approach, and then in a second phase try to factor in additional variables linked to the housing market more specifically.

Macroeconomic data can give us some explanations when taking into account the timeseries aspect (as we can quite safely consider that the time correlation for the neighbourhood features and the internal features is negligible). Looking at the evolution of the average transaction price in ruble from 2012 to 2015 in Figure 7a we notice an upward trend. However, if we look at the prices in euro in Figure 7b we see that the curve has a big slump at the second half of 2014. We have this curve due to the Russian financial crises of 2014, which happened in the result of the fall of oil prices. As a consequence, Russian ruble was devaluated.



(a) Daily mean price in ruble

(b) Daily mean price in euro

Figure 7: Daily mean price over time

3 Data preprocessing

3.1 Feature selection

Together with macroeconomic features the training dataset has 390 features. It is important to do feature selection in order to decrease the dimension of the data and get simpler methods.

One of the feature selection approaches that we use is variable importance obtained by XGBoost model, which is presented in Figure 8. As we can see, full_sq is the most important feature for predicting the price of a property. There are several other internal features which are selected to be important, including num_room, life_sq, build_year, and max_floor. There are many neighborhood features as well, several of which are describing the range of prices of cafes and sport centers. It was an expected result, because these features are a good price indicator of a particular area in general. Another important features are distance to the center of Moscow and time to metro by car. The last one can well separate properties in the peripheral districts, because they do not have metro stations.

The only macroeconomic variables that are important according to XGBooost model, are ruble to euro exchange rate and CPI (Consumer Price Index).

We select the top 20 features that are predicted by XGBoost to be important by XGboost as a subdataset, on which we will train our models.

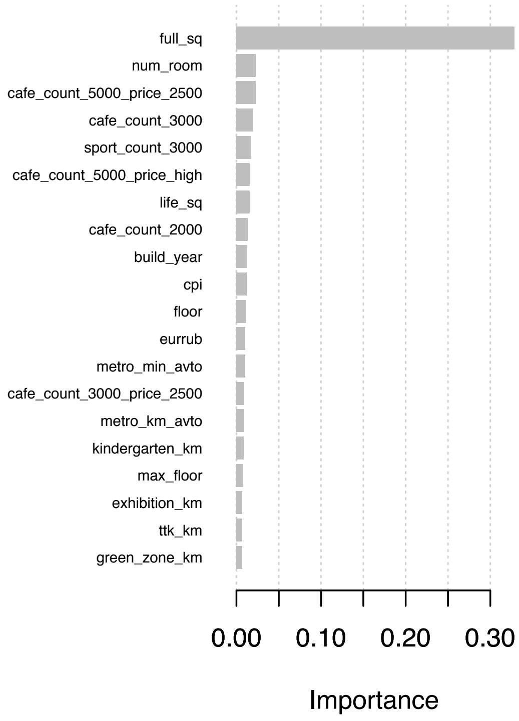


Figure 8: Variable importance by XGBoost

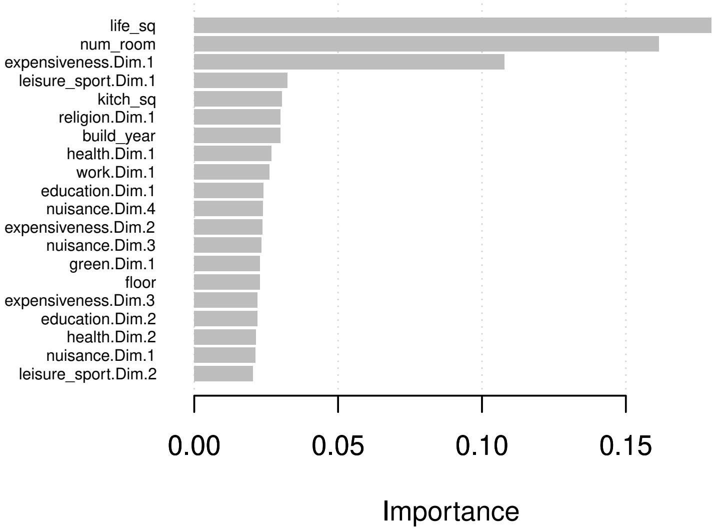


Figure 9: Variable importance by XGBoost on PCA transformed data

3.2 Group-PCA

As we could see from our initial analysis with additive and tree models, the internal features alone are not enough to accurately predict the price, and we need to factor in more variables, namely the neighbourhood and macroeconomic features. Given the relative large number of features (even though), and more importantly, the fact that many of them seemed of little relevance and quite colinear, we performed PCA on hand-selected groups of variables in order to orthogonalize our enriched dataset, then reduce the number of variables by selecting within each group, the minimal amount of variables to explain at least 80% of the variance. This "group-PCA" procedure gives us a good tradeoff between the loss of interpretability of the PCA transformed features in each group, while still keeping track of the group they belong to. We selected 9 groups, namely education, religion, expensiveness, green, health, nuisance, male_fem, work, sports_leisure.

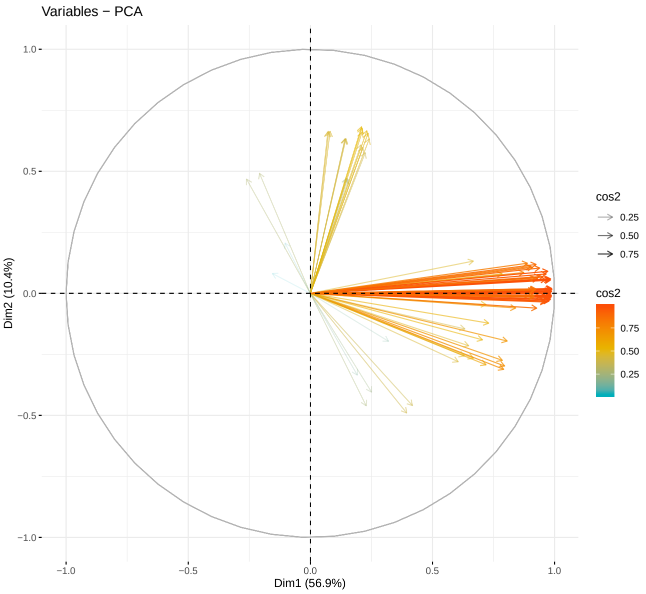


Figure 10: Correlation circle for the expensiveness features - positively correlated features are grouped together

We can observe from the correlation circle of figure 10, that the variables are quite well grouped into five clusters (two of which are relatively small). This confirms that many variables are heavily correlated and that the data can be compressed into 5 components without much loss. In our case, we reduced it even further, down to the three larger components.

3.3 Missing value treatment

In the dataset of internal and neighborhood features 51 out of 290 features have missing values. However, since we have do feature selection, we will focus on missing values in the subdataset with selected features. Figure 11 shows which features among top 20 features selected by XGboost have missing values.

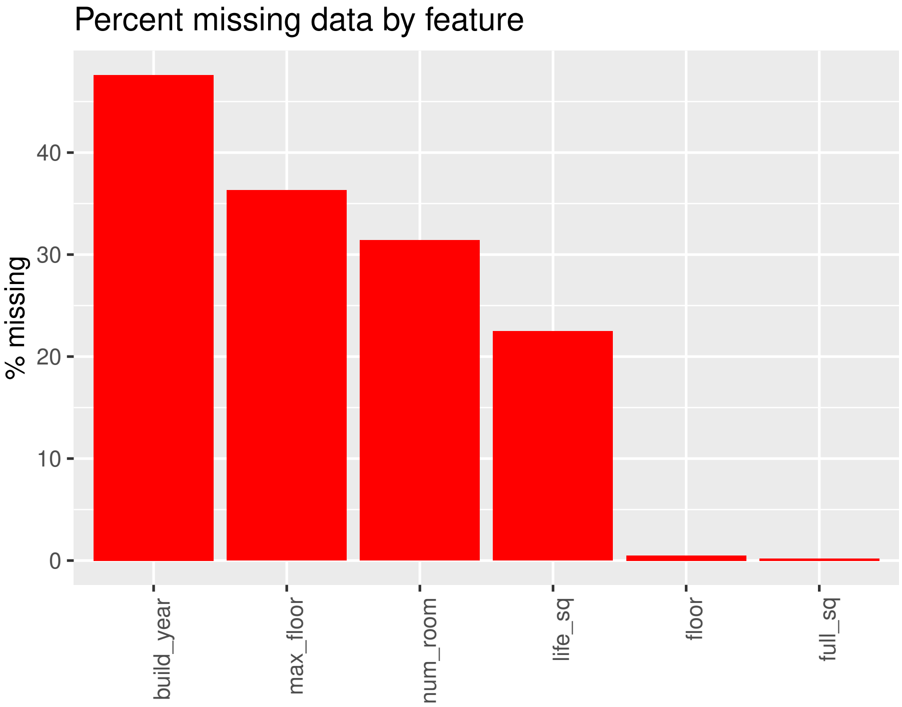


Figure 11: Missing values among selected features

Since some prediction methods do not work with datasets that have missing values, we will do missing value imputation. For that we will use DMwR package in R, that uses kNN method for imputing missing values. This means that we need the parameter of optimal number of neighbors, which we obtain by cross-validation. According to Figure 12 the optimal k is 35.

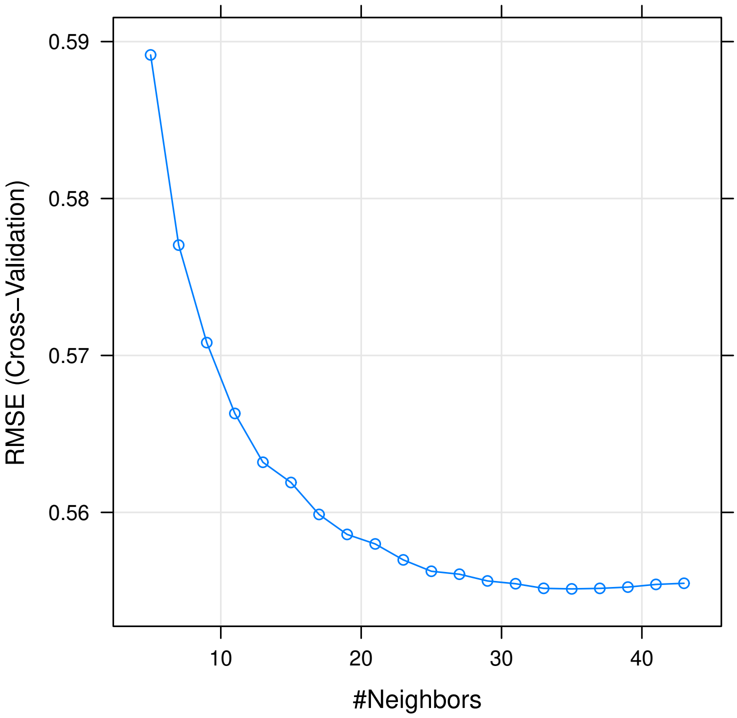


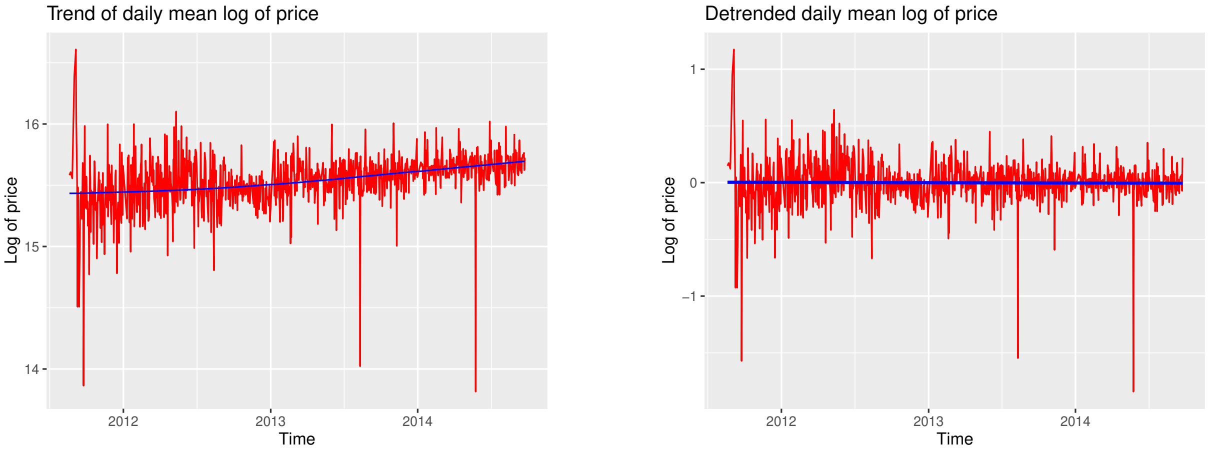
Figure 12: rmse of kNN by number of neighbors

3.4 Work on full_sq and OwnerOccupier vs.Investment splitting

As full_sq is by far our most important variable to predict the price, it crucial to be sure that the values we have are as close to correct as possible. The previous analysis on OwnerOccupier vs.Investment gave us doubts about the correctness of full_sq in the Investment case. To improve our results, we decided to build models on two types of datasets : the first one containing the original values of full_sq and a second one containing infered values of full_sq for the Investment products from selected features of OwnerOccupier type products using k-NN. Also, in our aggregation of models later on, we fit a copy of each model on the datasets filtered by product_type and combined the results to increase our performance.

3.5 Detrending the target variable

Since in Figure 7a we notice a trend in price in rubles over time, we detrend the target variable price. To that end, we fit a GAM of order $k = 4$ on training set to predict price only by timestamp, which is the same as to tell that we predict the trend. The obtained curve is illustrated in Figure 13a. After subtracting the predicted trend from price, we obtain detrended target variable in Figure 13b. In our models we will also use the detrended price as the target variable. This means that for the observations of test data we need to add to the output of the prediction model the trend obtained by GAM.



(a) Fitted trend curve for price

(b) Detrended price

Figure 13: Detrending price with GAMs

4 Additive Models

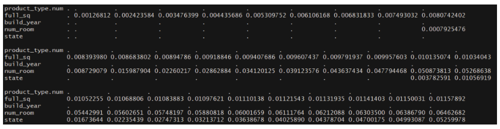
Although getting the most accurate predictions as possible is obviously a key goal of our project, we also believe that it is interesting to get a good understanding of the dataset and of the interactions between the key variables by building humanly interpretable models. Therefore, our focus will be split between :

- **performance** : build a complex model that will yield the best predictions possible
- **interpretation/understanding** : extract interpretable information from come complicated models (such as feature importance from tree based ones) and building simpler models (mostly linear) that will give us understandable information about the data. Moreover, this

step will help us perform feature engineering and data transformations that can then be fed to the more complex models.

4.1 Linear models

In order to build our linear models, we checked for linear relationships, in a generalized sense, between our co-variates (*i.e.* after a possible transform) and the target variable log_price which we will refer to as price from now on. First, we performed some variable selection using a Lasso estimator to extract the most important variables and use them to build linear models. The glmnet package provides such an estimator, and runs the estimation for increasing values of the regularization parameter. This allowed us to rank the variables in order of appearance, which coincides with the importance ranking (Figure 14a). The t-value provided by this method could also have been used to rank the variables (Figure 14b).

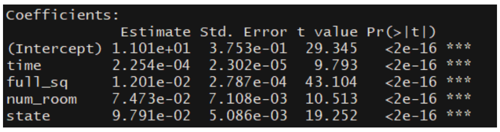


```
product_type.num . .          .           .           .           .           .           .           .           .
full_sq          . 0.00126812 0.002423584 0.003476399 0.004435686 0.005309752 0.006106168 0.006831833 0.007493032 0.0080742402
build_year       . .          .           .           .           .           .           .           .           .
num_room         . .          .           .           .           .           .           .           .           0.0007925476
state            . .          .           .           .           .           .           .           .           .

product_type.num .           .           .          .          .           .           .           .           .           .
full_sq          0.008393980 0.008683802 0.00894786 0.00918846 0.009407686 0.009607437 0.009791937 0.009957603 0.010135074 0.01034043
build_year       .           .           .          .          .           .           .           .           .           .
num_room         0.008729079 0.015987904 0.02260217 0.02862884 0.034120125 0.039123576 0.043637434 0.047794468 0.050873813 0.05268638
state            .           .           .          .          .           .           .           .           0.003782591 0.01056919

product_type.num .          .          .          .          .          .          .          .          .          .
full_sq          0.01052255 0.01068806 0.01083883 0.01097621 0.01110138 0.01121543 0.01131935 0.01141403 0.01150031 0.01157892
build_year       .          .          .          .          .          .          .          .          .          .
num_room         0.05442991 0.05602651 0.05748197 0.05880818 0.06001659 0.06111764 0.06212088 0.06303500 0.06386790 0.06462682
state            0.01673644 0.02235439 0.02747313 0.03213712 0.03638678 0.04025890 0.04378704 0.04700175 0.04993087 0.05259978
```

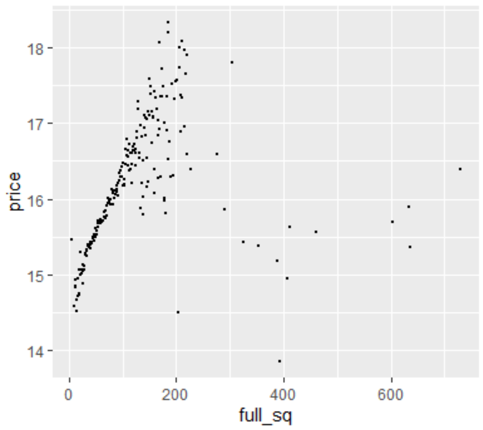
(a) Feature importance by Lasso



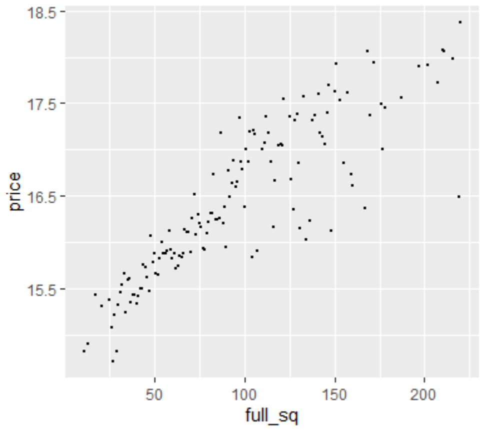
```
Coefficients:
            Estimate Std. Error t value Pr(>|t|)    
(Intercept) 1.101e+01  3.753e-01  29.345   <2e-16 ***
time        2.254e-04  2.302e-05   9.793   <2e-16 ***
full_sq     1.201e-02  2.787e-04  43.104   <2e-16 ***
num_room    7.473e-02  7.108e-03  10.513   <2e-16 ***
state       9.791e-02  5.086e-03  19.252   <2e-16 ***
```

(b) Feature t-value

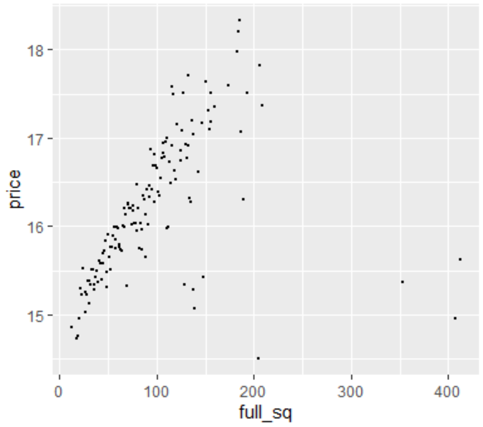
It appears that the size of the property and the number of rooms (in this order) are the two most important variables. Although they can appear to be quite correlated, it can also indicate for example that T2-type apartments are generally more expensive than T1-types. There is a linear trend between the price and the size and between the price and the room number for small values of the co-variates ($< 200\text{m}^2$ for the size, <10 rooms for the room count), which is quite coherent given that these are the more common values of sizes and room counts and are therefore less prone to outlying or erroneous values. This gives us the threshold of 200m^2 for which this univariate linear model price $\sim$ full_sq



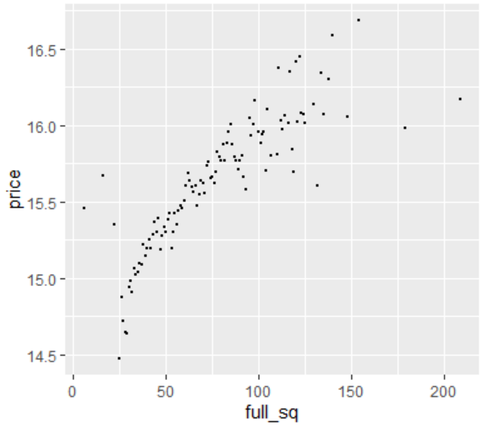
(a) Constant price/m^2 as a first approx.



(b) Central



(c) Western



(d) Novomoskovsky

As the price per square meter is expected to vary depending on the district, we plot the same graphs for each district and also observe the same type of linear behaviour, with varying coefficients corresponding to the price per square meter of the district under study. Hence, we build an initial naive linear model using the size full_sq and dummy variables for district. We get an error of around 0.48 for the rmse, with an R squared of 0.302, which seems acceptable for such a simple model given the spread of the data. Actually, we cannot expect great performances from this model, since multiple prices correspond to each possible combination of the considered explanatory variables (in the sense that multiple transactions with different prices are associated to one combination of full_sq and district). Therefore, we need to complexify our model and factor in additional explanatory variables. Moreover, the interesting linear univariate effect we observe when grouping the data according to one variable and obtaining a linear fit on another one in each case hints towards interaction terms between the different variables. We will also attempt to take those effects into account by adding terms of the form variable1*variable2 in our formulas to account for bivariate effects for instance. In order to visualize the multivariate effects, which would require multi-dimensional plots to be represented in a functional manner, we chose a "univariate" representation in the sense that we predict the price using the multiple variables, but then represent each data point by a 2D-point (var_repr, price) where var_repr is a chosen variable and price the predicted price. If the other variables are relevant in the prediction of the price, this should lead to points more scattered around the plot than simply on a line, since this time for a given value of var_repr, the other variables will allow us to predict different prices.

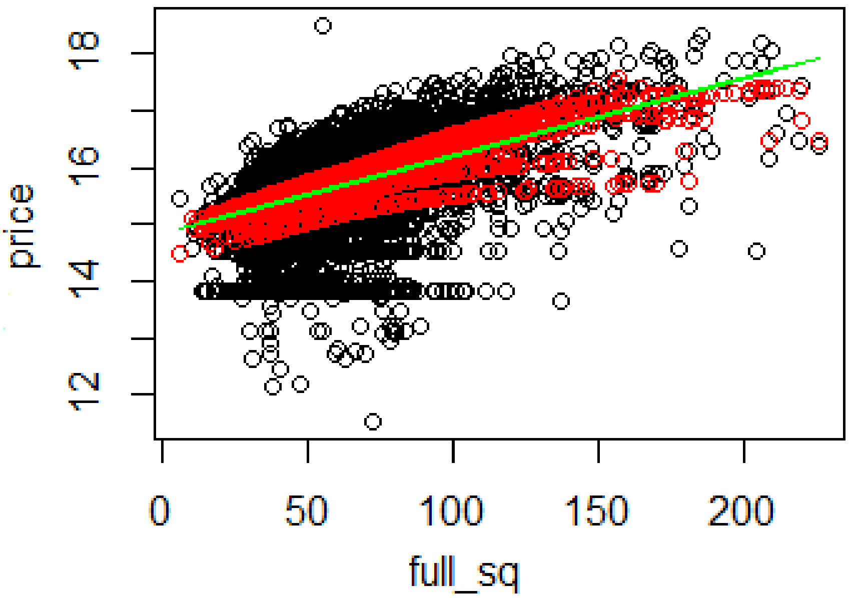


Figure 16: Green : Global price per square meter model
Red : Univariate representation of the 2D-explanatory variable linear model
price ∼ full_sq by district

4.2 Enriching our linear models

Adding linear effects with more explanatory variables

We enrich our linear model by adding more explanatory variables to it :

- baseline model : price ∼ full_sq + district + time.num
- running the model with all the internal features does not significantly improve our prediction compared to the baseline model as the rmse only decreases slightly, from 0.4694 to 0.4686. This suggests that the linear effects of the other internal variables are not relevant to predict the price.
- adding the PCA processed neighbourhood variables : the rmse decreases to 0.4402.

Now that we have a rough understanding of which internal variables to keep and about the impact of the neighbourhood variables, we add regularization and perform variable selection in our models to improve the predictions. For the Lasso penalization, there is no variable selection to do as it is inherently taken care of by the model. The performances are summed up in the table below

Performances

Model	RMSE
All internal	0.4686
Ridge PCA	0.4457
Lasso PCA	0.4402
ElasticNet ($\alpha = 0.5$) PCA	0.4436

Table 1: RMSE of linear models trained on the internal features only and with added PCA features (the best result in red)

We note that Lasso outperforms Ridge regression, and cross validation (performed "by hand") actually shows that for the ElasticNet, the optimal mixing parameter is $\alpha = 1$ which corresponds to Lasso. This means that variable selection is very important whereas variable scaling is not very relevant. As we move on to GAM models in order to include some smooth non linear effects in our model, one of the conclusions of this section is that few internal variables actually have linear effects on the price. We will see in the next section that although they do not have a linear influence, some variables can be exploited for their bivariate effects with the key variables identified in this section. As GAM models allow us to search for linear and non-linear bivariate effects, we will directly move on to those types of models instead of restricting ourselves to adding terms of the form $\mathsf{var1} * \mathsf{var2}$ in linear models. Obviously, we will also search for univariate non-linear effects.

4.3 GAM Models

Non-linear univariate effects

To search efficiently for non-linear univariate effects, we used the approximate significance values given for the smooth terms in summary(my.gam). All the variables involved in our baseline model were considered significant, so we included them.

Searching for bivariate effects

As the number of interaction combinations between variables is exponential, we restricted our analysis to bivariate effects. We mainly used three approaches to search for those effects :

- feature importance : we looked for interactions between the most important variables, given by xgboost and Lasso selection
- cross categorical variables with continuous variables :
  - *Scaling effects* of the categorical variable : dummy encoding for categorical variables gives one coefficient per dummy variable and our model can therefore be adjusted to each value of the categorical variable independently
  - *Level effects* of the categorical variable : we look at the Student test for that variable to see whether the categorical variable should only be used to scale a continuous variable, or whether it's own additive value should be taken into account
- ALE Plots : those plots give a heatmap of the prediction improvement on the 2D space $\mathsf{var1} \times \mathsf{var2}$. Interactions between those two variables are given by the presence of distinct regions. Moreover, a discontinuous transition from one region to another indicates that

non-linear effects occur and should be taken into account. Therefore, these plots can also be exploited in the case of GAM models for instance to identify which smooth non-linear terms should be included.

Scaling and level effects of district

In the section on linear models, we saw that the district was considered unimportant by linear models. However, we identified different average prices per square meter depending on the district in our earlier analysis. This suggests a scaling effect of district on full_sq. It is confirmed by the ALE Plot below, showing distinct heat regions depending on the district. This graph is coherent with our earlier remark on more accurate predictions for the more owner occupied districts. Hence, the following models will use district as a scaling variable. Moreover, the significance of district in the former linear models on the otherside, suggests that we should also take into account the level effects.

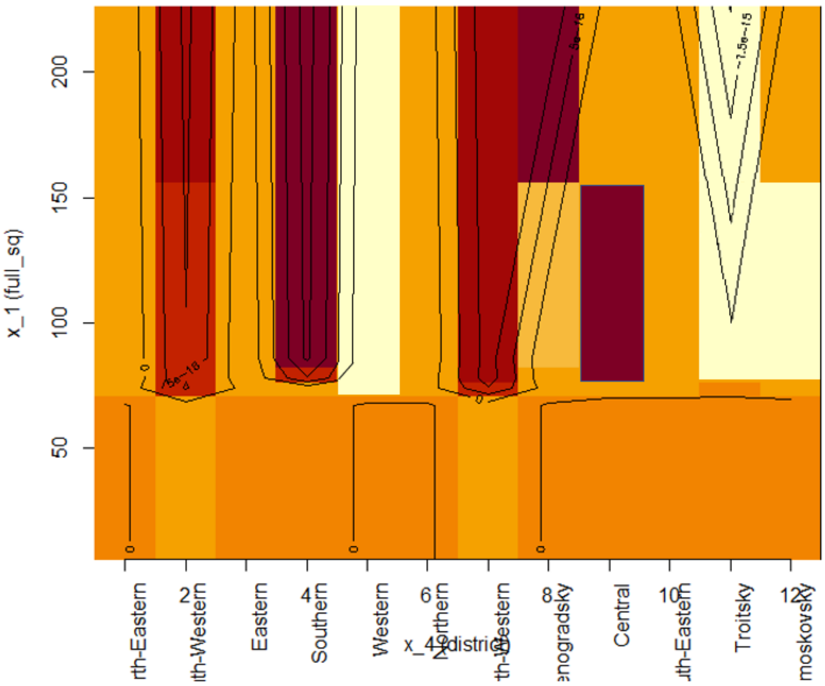


Figure 17: ALE Plot for full_sq × district.
Colors indicate the effect of the covariates on the predictions.
Light : positive effects - Dark : negative effects

state × full_sq

The ALE Plot shows that the state variable has a positive effect on predictions for state >2. It can be because the case of state $=1$ mainly corresponds to OwnerOccupier type products for which full_sq alone is probably more relevant. The interaction state × full_sq does not seem very important, and indeed only leads to marginal improvements of our score. Nonetheless, we chose to keep it in our models to gain some extra digits.

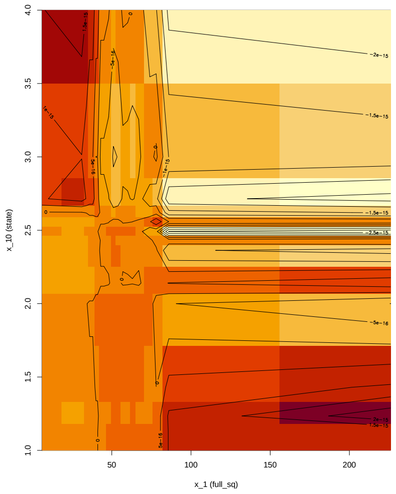


Figure 18: ALE Plot for full_sq × state

Other interactions

Applying this method to other combinations of variables, we came up with the following additional interactions :

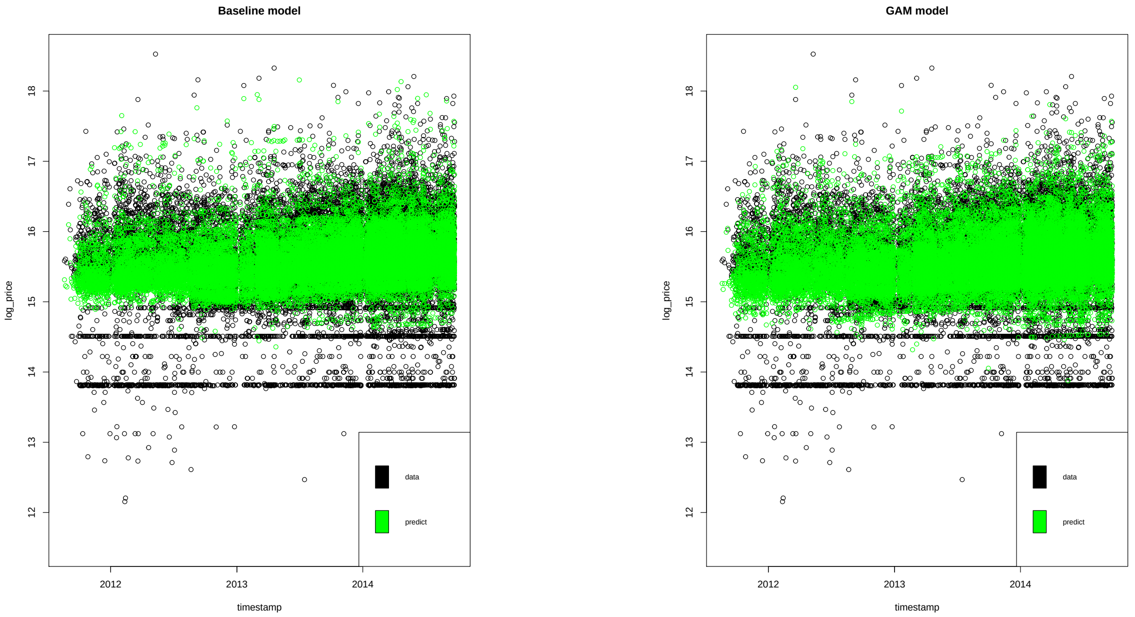
- PCA_variables × district: notably expensiveness, work and sports_leisure as they came up quite high in the XGBoost feature importance graph.
- state × product_type: as mentionned previously, the state variable seems to provide little additional information for OwnerOccupier if we include product_type, as it is almost always equal to one in that case. However, it is useful for Investment.

These are the variables involved in the GAM3 model.

Performances and model fit

Model	Description	RMSE
GAM1	baseline variables	0.441
GAM1	baseline + PCA no select.	0.430
GAM3	see above	0.424
GAM4	baseline + PCA select.	0.429

Table 2: RMSE of GAM models trained on the PCA dataset (the best result in red)

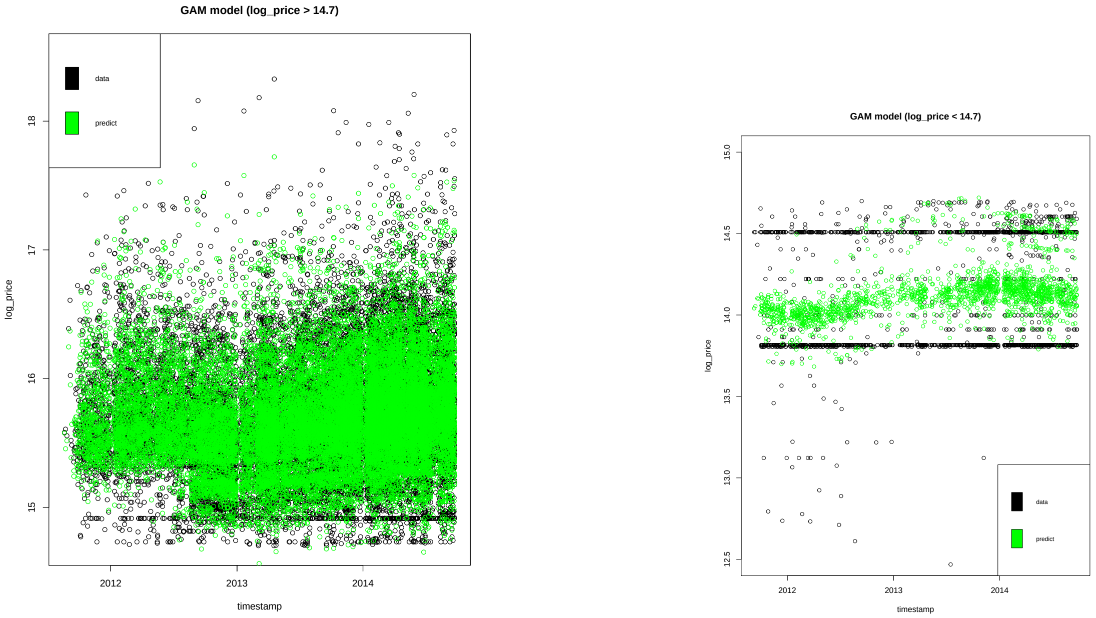


(a) Predictions for our baseline model

(b) Predictions for our best GAM model

Figure 19: Model fit

Residual and outlier analysis



(a) price > 14.7

(b) price < 14.7

Figure 20: GAM fit after filtering on price threshold 14.7

We note from Figure 19b from the performance subsection that the model seems quite performant for prices above 14.7, however it struggles with the values under this threshold. Even though we obviously don't have access to the price for the test data, we tried fitting our models seperately for price < 14.7 and price > 14.7 to see how well it improved our predictions. It turns out that there is a very good fit in the latter case (fig.20a), giving a rmse of 0.228. For the other case (fig.20b), we fit another simpler GAM model, giving a rmse of 0.337. Cumulatively this amounts to a rmse below 0.3 would be is a huge improvement, provided we can correctly identify those outliers.

4.3.1 Further improvements

Investment/ OwnerOccupier split.
When analyzing the dataset filtered for price < 14.7, we notice that 1677 out of the 1814 observations correspond to Investment type products. This echoes with our previous analysis and shows that the distinction between OwnerOccupier and Investment is really relevant. Therefore, we also trained two copies of a GAM model on two separate datasets according to product_type and then grouped the predictions. This lead to a minor improvement of 0.002 on the rmse. Indeed, even though there is a clear majority of Investment products among those outliers, they remain a minority even among the Investment products, and in the end we do not manage to isolate many of them.

k-NN imputation for outliers.
We identified that the outliers share a common feature : they are mostly of Investment type. However, this is not enough to be able to isolate them. Figure 20b shows that while these outliers distributed according to price < 14.7, which unfortunately we don't have access to, there seems to be an even stronger pattern as they are distributed in two groups of prices. This encourages us to believe even more that there are some underlying common features which will allow us to isolate those outliers and k-NN imputation seemed like an appropriate way to do so. We added a flag variable indicating whether or not price < 14.7 and used the knnImputation function from the DMwR package to impute the "missing values" on the test set. This technique allows us to isolate a bit more outliers than previously, as we can see on Figure 21 and improves the rmse to 0.419

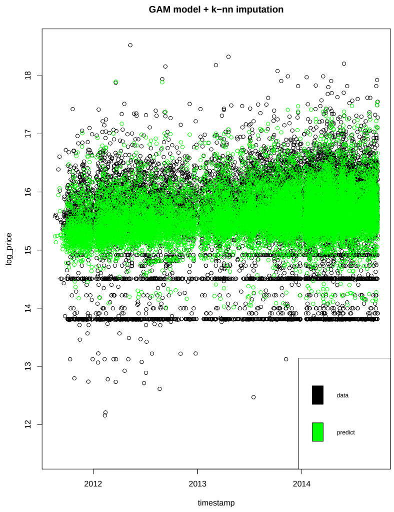


Figure 21: GAM model fit after k-nn imputation - Showing more green dots for low price values now !

k-NN imputation for full_sq.
As all our models seem to give a huge importance to full_sq, it seems crucial to have correct values for this variable. Given that we are not sure it is correct for Investment type products, we thought

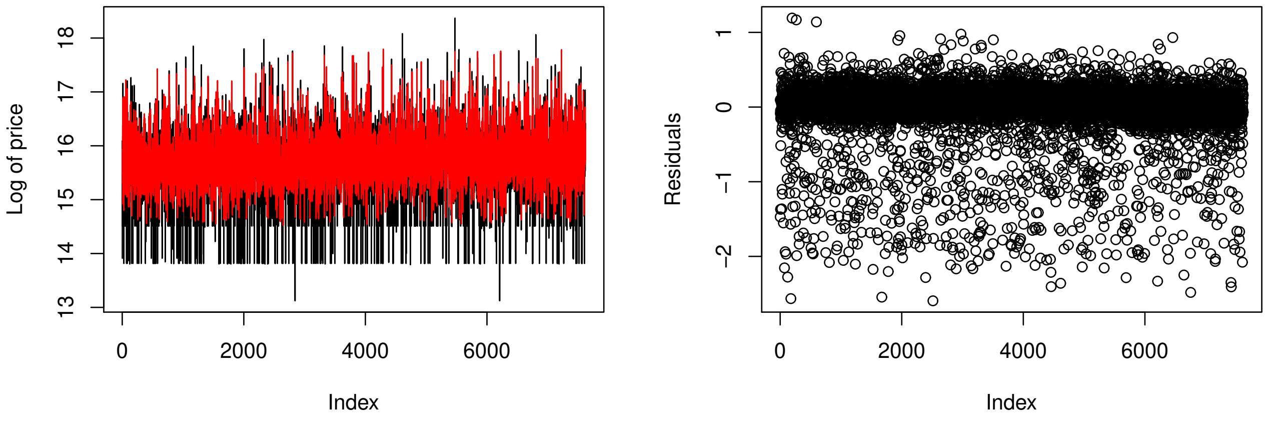
of correcting it through k-nn imputation, keeping only the OwnerOccupier values and putting missing values for Investment. This could then be included in our final aggregation of estimators to possibly increase the performance. Unfortunately, we did not have enough time for this approach.

Better model for outliers.
The outliers seem to be grouped around two prices, we could try to take advantage of this.

5 Tree Models and XGBoost

5.1 XGBoost and Gradient Boosting

We fit XGBoost models on 8 different datasets that we have obtained. We take the initial data, dataset of only internal features, the dataset that we got with feature selection conducted with XGBoost, and the dataset with feature selection and missing value imputation. Also, with each dataset we take as target variable price and detrended price. The results are presented in Table 3. The best performance is obtained on dataset with feature selection and detrended target price. Figure 3 shows the predictions of XGBoost model in red and the actual values of price in black. Mostly, the prediction fits the actual values quite well, however it is not able to predict the prices which are lower. The residual plot in Figure 22b gives the same conclusion: most of the residuals are close to 0, but there are many negative residuals, implying that low prices are not predicted well.



(a) Fitted price

(b) Residuals

Figure 22: XGBoost on seleted variables with price detrending

Dataset	Detrending	RMSE	MAPE
Initial data	No	0.432	1.819
	Yes	0.419	1.521
Internal features	No	0.433	1.868
	Yes	0.426	1.599
Feature selection	No	0.425	1.758
	Yes	0.415	1.513
Feature selection, no missing values	No	0.429	1.831
	Yes	0.416	1.620

Table 3: RMSE and MAPE of XGBoost models trained on different datasets (the best result in red)

In Gradient Boosting Method we use as training the subdataset with selected features by XGBoost and detranded price variable as target. We perform cross-validation in order to choose the optimal number of trees for GBM. The best model has rmse of 0.423. Although the result is not better than XGBoost, GBM is compatible with the best GAMs.

5.2 Decision Tree and Random Forest

On the subdataset with selected features and detrended price we fit Decision tree model. Additionally, we perform cross-validation to choose the optimal depth of the tree and do pruning. In the result rmse reduces from 0.498 to 0.449. The obtained tree is presented in Figure 23. Again, we see that full_sq has huge influence on the prediction.

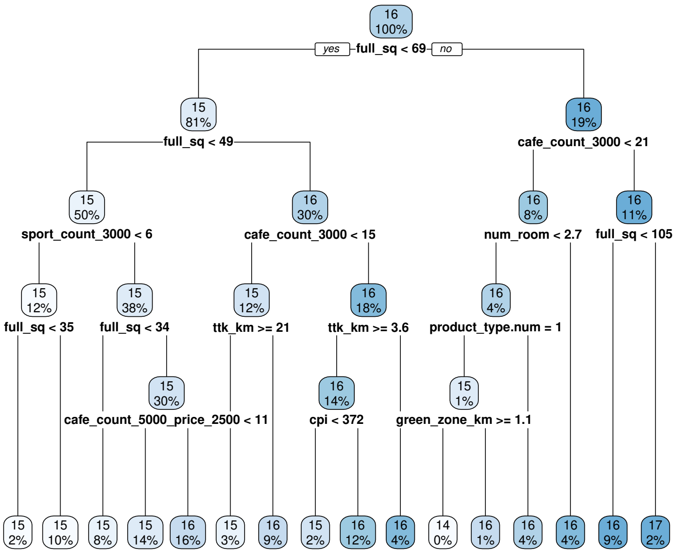


Figure 23: Tree

Additionally, we train Random forest model. The best model yields rmse of 0.441. However, we were not able to perform proper cross-validation to tune the parameters due to computational limitations.

6 Model aggregation

EWA expert aggregation on all our models

Now that we have 9 different models at our disposal, our last improvement is to aggregate them using the EWA aggregation method of the opera package. The average losses suffered by each of our models is summed up in the graph below : As we can see, the resulting estimator, obtained as a convex combination of our individual estimators, is indeed an improvement compared to our best individual estimator. It yields an rmse of 0.409 on the test data. Figure 25 shows that our aggregated model seems to fit the data quite similarly to our best models. Especially, we note that the aggregation did not result in a siginificant improvement for the predictions of the low outlying values, which could be expected since all our models struggle with those data points.

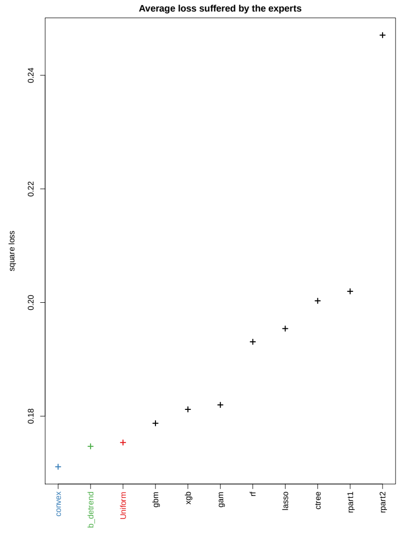


Figure 24: Average loss suffered by each model

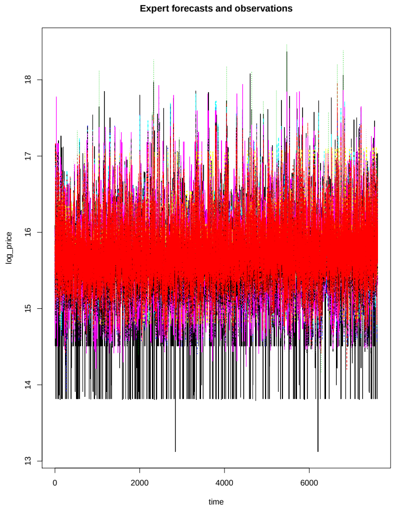
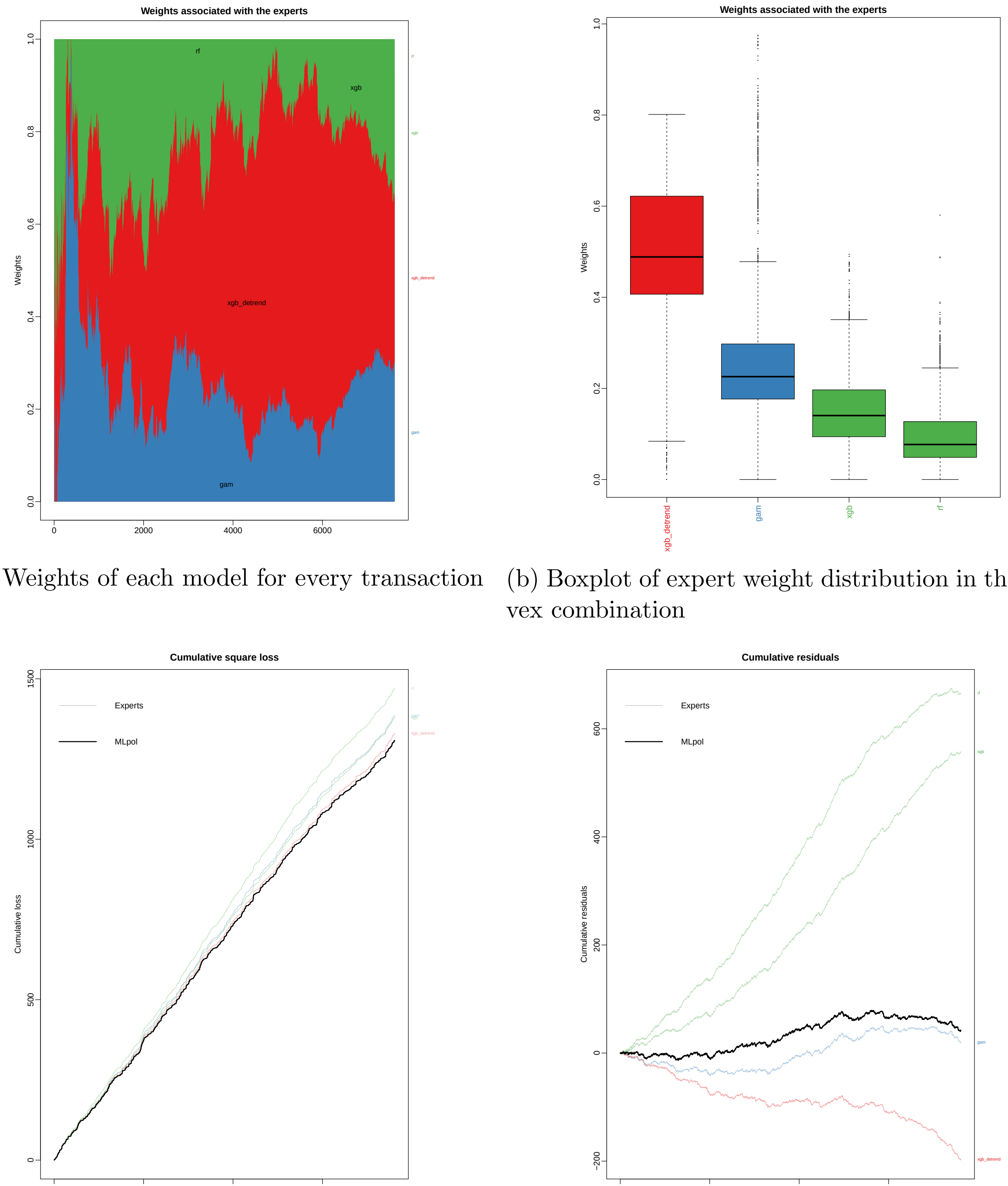


Figure 25: Fit between predictions from the aggregated model and the data (in black)

Looking at the coefficients of the convex combination, we note that lasso, gbm, rpart2, ctree have negligible coefficients (below 10^{-17}). Therefore, they can safely be removed from the aggregation to allow us to visualize the mixture more clearly. Figure 26a shows that the only models with a significant contribution to the prediction are xgb, xgb_detrend and gam. Overall, these three models seem to contribute quite evenly to the prediction, although xgb_detrend seems to slightly have the upper hand.



(a) Weights of each model for every transaction

(b) Boxplot of expert weight distribution in the convex combination

(c) Cumulative square loss

(d) Cumulative residuals

The cumulative square loss curve has some very steep increases, for example around id = 2000, which could indicate the presence of many outliers around this id value. Since transaction IDs are ordered in time, this could maybe be exploited (if it hasn't already been ”done” by our models) to isolate the outliers a bit better.

7 Conclusion and discussion

The best model is XGBoost; nonetheless, additive models perform very well on regular values of the data and it is actually the presence of outliers that hinder their performance. The fact that XGBoost outperforms them for the outliers could indicate that more complex interactions than the bivariate ones we included are required to account for those outliers. Moreover, a lot of data analysis and pre-processing is required in order to improve their performance on those outliers, which is probably innately done by XGBoost.

The good performance of additive models on regular data however, gives information about some key features influencing the price of housing transactions :

- the most important feature by far is the size of the appartment
- neighbourhood features tend to be the next most important features, even more than many of the auxiliary internal features. More specifically, the features serving as a proxy for the wealth of the neighbourhood outweigh those indicating the intrinsic quality of the neighbourhood regarding feature importance.
- outside of the global trend captured by the time.num variable, we were surprised to note that the macroeconomic indicators were quite irrelevant. It is maybe due to the time range of our dataset, during which those indicators were too "stable", or due to a misuse of those features on our side.

This project showed us that exploratory analysis and data checks are key in order to build relevant models. In fact, here the quality of the data is the main burden to our predictions, and identifying the presence of outliers in order to isolate them is in our opinion the key to further improving the scores. Furthermore, the misleading Investment product type taught us that having the wrong understanding of the data can lead to inappropriate models, focusing on the wrong variables or missing some key aspects of the data. It was our case as we spent quite some time before realizing that some Investment values for full_sq most notably were incorrect or at least over or under valued and were not able to handle the outliers correctly (even though in the end our best models still struggle to isolate them).