
PREDICTING LISBON HOUSE PRICES WITH TREE-BASED MODELS

A PREPRINT

Julia Przybytniowska

Faculty of Mathematics and Information Science
Warsaw University of Technology
Warsaw, ul. Koszykowa 75
j.przybytniowska@gmail.com

Hubert Ruczyński

Faculty of Mathematics and Information Science
Warsaw University of Technology
Warsaw, ul. Koszykowa 75
hruczynski21@interia.pl

Kacper Skonieczka

Faculty of Mathematics and Information Science
Warsaw University of Technology
Warsaw, ul. Koszykowa 75
k.skonieczka01@gmail.com

May 26, 2022

ABSTRACT

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetur id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

Keywords Tree-based models · Regression · Lisbon House Prices · XAI

1 Introduction

The main purpose of our study is to analyze the estate situation in Lisbon and provide responsible and explainable models for the task of price estimation. To achieve our goal we conduct a thorough analysis of the Lisbon House Prices data set followed by the creation of the models, which are later explained with the usage of the DALEX package (Add References).

2 Data

Lisbon House Prices is a rather small data set from Kaggle. It contains 250 observations, which represent estates for sale in Lisbon, which are technically described by 17 columns, however, some of them are unnecessary. In the end, we decided to remove *Id* column which gives us no information at all and columns *Country*, *District* and *Municipality*, because all records have the same values there. Despite these columns, the data set doesn't have any corrupted records and NULL values. Additionally, we provide the description of every variable in the data set:

- *Id*: is a unique identifying number assigned to each house
- *Condition*: The house condition (i.e., New, Used, As New, For Refurbishment)
- *PropertyType*: Property type (i.e., Home, Single habitation)

- 2

2.1.3 Map visualization

To get more information about the estate market in Lisbon, we've created maps of the estates presented in the data set. The first thing we saw is the aforementioned Apartment dominance of the market. Moreover, there is a high similarity of distribution between AreaNet and Price of the property. Apparently, the most expensive properties are not located in the city centre, despite the fact, that the highest prices per square meter are obviously in the city centre.

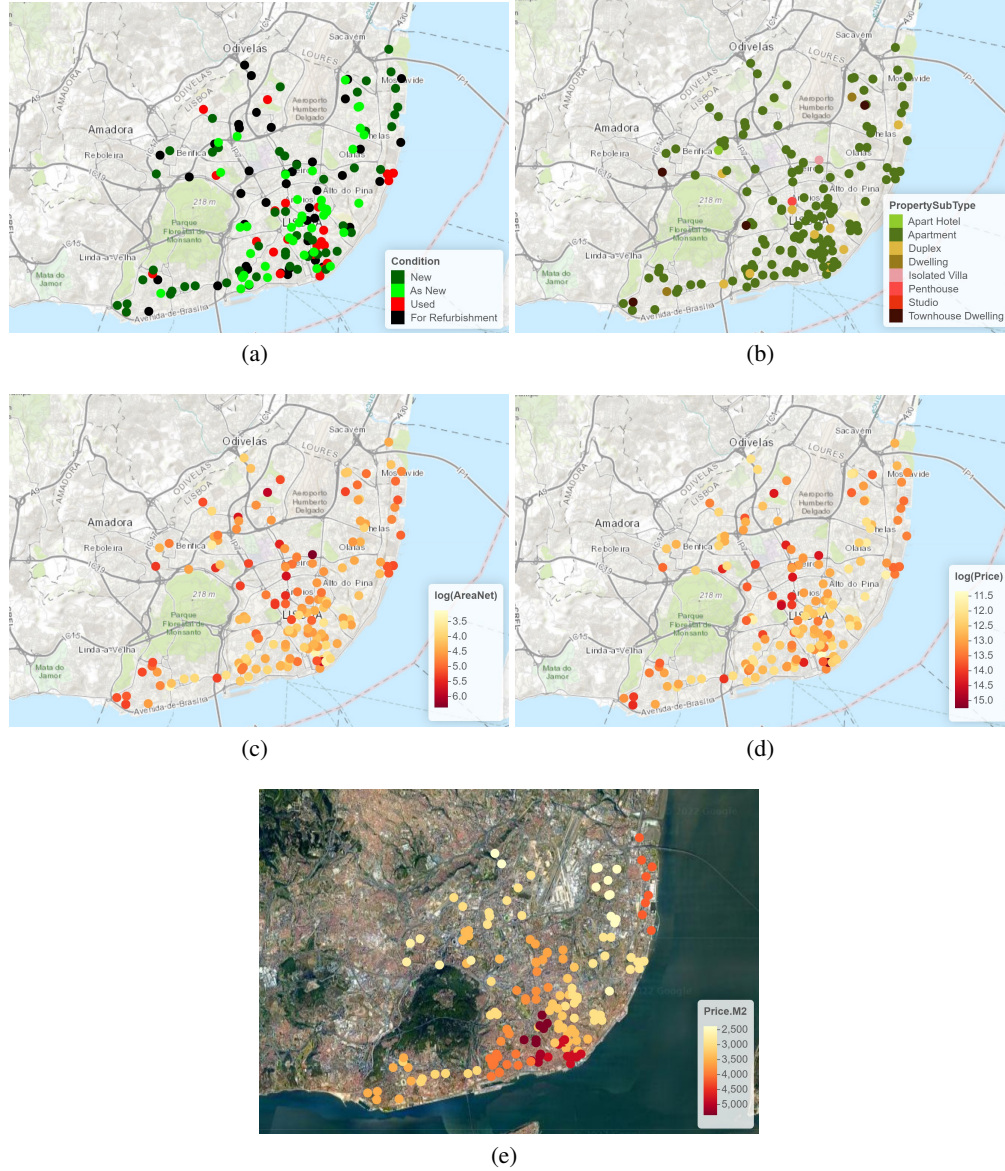


Figure 2: (a) Property Condition (b) Property SubType (c) Property AreaNet (d) Property Price (e) Property Price m^2

This analysis lets us understand the problem we are facing and enables us to use correct information from the data set in the next phase, which is the model training.

3 Models

In this section, we will present the process of the models construction dwelling deeply into the hyperparameter optimization. The regression task of predicting real estate values, which we are facing is a tough one because as

we’ve mentioned before, the data set contains only 250 observations. Because of that, we focus on hyperparameters optimization and we deeply analyse the models created in the process to correctly choose the best ones.

The variables from Lisbon Houses Prices were preprocessed before the training. As it is mentioned in the previous section, we removed columns: Id, Country, District, Municipality and AreaGross, because, after the EDA, we found them unnecessary. We’ve also scaled the AreaNet column and one-hot encoded categorical variables which are: Condition, PropertyType, PropertySubType and Parish. The data set was also split into train and test subsets which are 70% and 30% of the original data set.

3.1 Hyperparameters Optimization

To find the best models we used the Bayesian Optimisation method from the ‘ParBayesianOptimization’ library. In each model our grid was a little different, depending on the models’ specifications. It means that every model used a different set of hyperparameters and even if some of them were present in more than one model, they were set on different values to maximize the outcomes of the given model. A full hyperparameter tuning list is present in the Table 1. During the optimization step, we aimed for the best Mean Absolute Error (MAE) values, because this metric gives more intuition about the prediction correctness than, for example, Rooted Mean Squared Error (RMSE).

Parameter / Model	Random Forest	XGBoost	LightGBM	CatBoost
num_trees	+	-	-	-
mtry	+	-	-	-
min_node_size	+	-	-	-
max_depth	+	+	+	-
eta	-	+	-	-
min_child_weight	-	+	-	-
subsample	-	+	-	-
lambda	-	+	-	-
alpha	-	+	-	-
nrounds	-	+	-	-
min_data_in_leaf	-	-	+	+
num_leaves	-	-	+	-
bagging_fraction	-	-	+	-
depth	-	-	-	+
n_estimators	-	-	-	+
l2_leaf_reg	-	-	-	+
learning_rate	-	-	-	+

Table 1: Hyperparameters’ grid.

3.2 Best models selection

At this point we have to evaluate the models created in the previous step and select the best performing ones. Our selection method consists of three main steps:

- RMSE and MAE test values comparison,
- RMSE and MAE train vs test analysis,
- Split Groups plots analysis.

During the training process, we’ve created about 100 models and narrowed them down to 20. These models were chosen as 4 groups of 5 best performing models of every model type: Random Forest, XGBoost, LightGBM and CatBoost. The easiest method to compare them is via plots present on Figures 3 and 4. These visualizations plot RMSE or MAE train values against equivalent test values on a logarithmic scale. This way, we can detect models that perform the best for each metric or we can find models which are overfitted (far from the $y=x$ line).

Accordingly to this selection method we narrowed our set of 20 models to just 4, which are presented in the Table 2. The pros of every model are presented below:

- Random Forest 3 is the best performing model in terms of RMSE and has second lowest difference between train and test values of RMSE. It is also second the least overfitted model in terms of MAE.

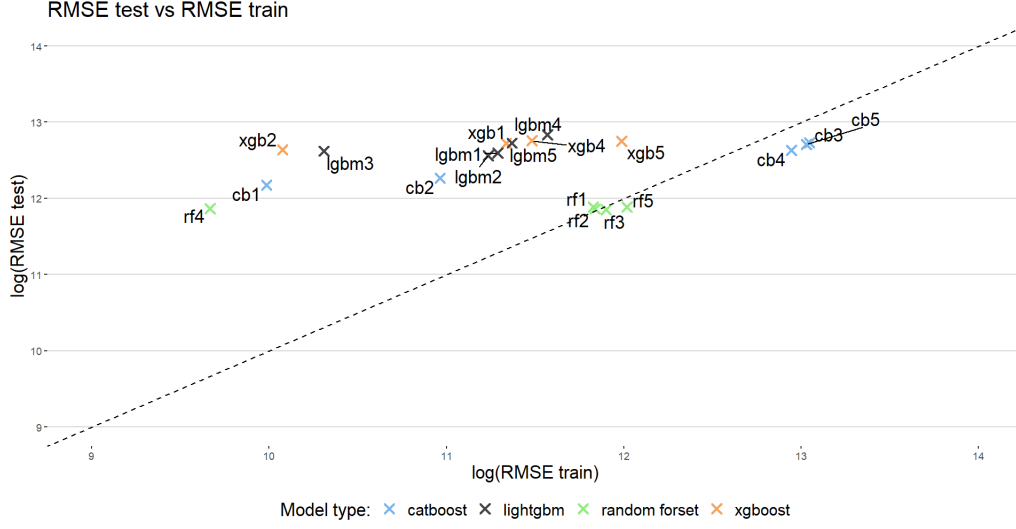


Figure 3: Train and test RMSE values comparison for all trained models in logarithmic scale

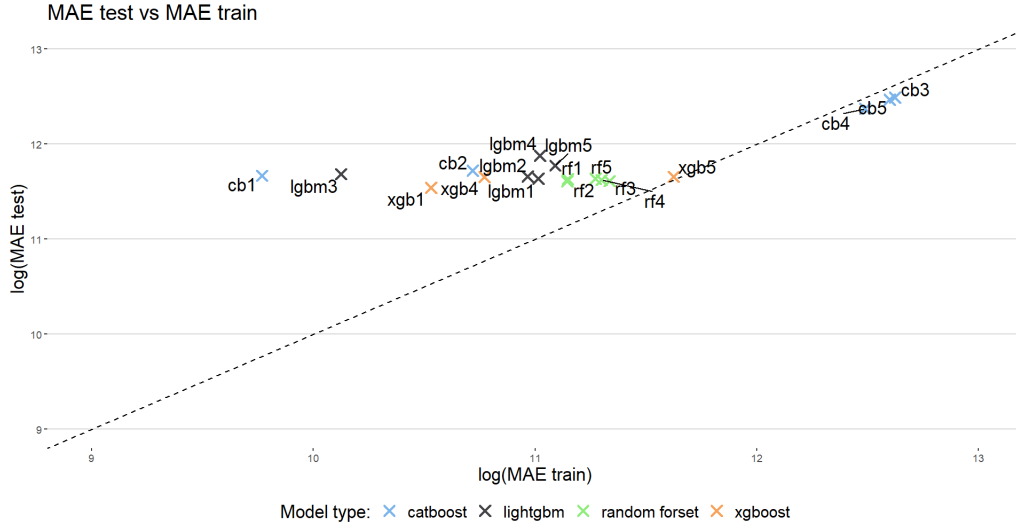


Figure 4: Train and test MAE values comparison for all trained models in logarithmic scale

- Random Forest 2 is second best model in terms of RMSE value and the least overfitted one in terms of RMSE.
- xgboost 1 has the lowest MAE test metric, however it is overfitted
- xgboost 5 is the only model which is not overfitted in terms of MAE.

Model Name	RMSE train	RMSE test	MAE train	MAE test
Random Forest 2	138332	141902	69283	109958
Random Forest 3	147210	140305	83805	110186
xgboost 1	83898	335600	37455	102646
xgboost 5	160925	344837	111682	115334

Table 2: Preselected models

In the end, we created Split Groups plots to see, how our models classify observations. These plots automatically create bins for 8 subgroups of price ranges of the original data set and save where specific observation lands. Later they take

predicted values and do the same thing for them. In the end, they take these observations and compare in which bin they were originally and where did they land after the prediction. The outcomes for our models are presented in Figure 5.

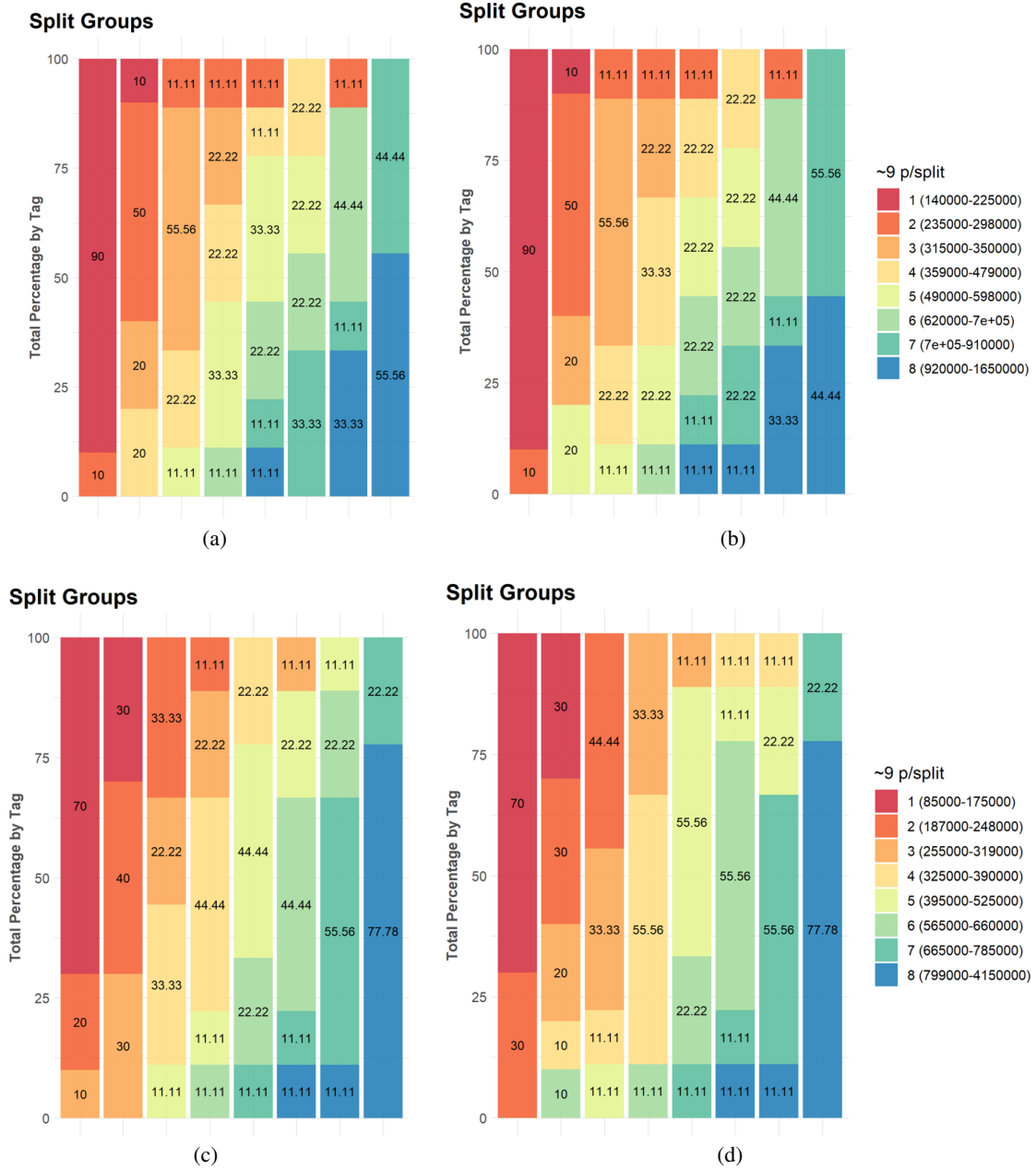


Figure 5: Split groups plots for: (a) Random Forest 2 (b) Random Forest 3 (c) XGBoost 1 (d) XGBoost 5

According to this diagnostic, we decided to choose two models for eXplainable Artificial Intelligence (XAI) analysis, which will be conducted in the next part of this paper. first The model is **Random Forest 3** because despite being one of the two best performing models in terms of the metrics, it was also slightly better than Random Forest 2 in the prediction quality for the expensive houses. The second chosen model is **XGBoost 5** due to its lack of overfitting in terms of MAE and really good prediction of expensive houses, which outperforms even the aforementioned, random forest model.

4 XAI

Creation of well performing models is not the final step in our research, because it only says what the prediction is and it doesn't tell us why it has such value. To obtain and analyse such knowledge, we have conducted explainable AI analysis on the previously selected models Random Forest 3 and xgboost 5. It consists of local (Break Down, SHapley Additive exPlanations (SHAP), Ceteris Paribus (CP)) and global methods (Feature Importance, Partial Dependence Profiles (PDP), Accumulated Local Dependence (ALE)).

4.1 Local methods

4.1.1 Break Down profile and SHAP

The first local method is a Break Down profile which is similar to the method introduced by Robnik-Šikonja and Kononenko (Staniak and Biecek [2018]), which enables the user to track changes in the final prediction with the input of the most important variables. The name of this method is pretty logical because it breaks down the reasoning which stands after the prediction. The first label on the plot is the Intercept, which is the mean prediction of the model, later we can see the effect of the most important variables on the prediction. The red bar means that it has a negative impact on the outcome and the green one means the opposite.

Break Down profile method is highly connected with the second one - SHAP. SHapley Additive exPlanations are based on "Shapley values" developed by Shapley (Lundberg and Lee [2017]). This method, similarly to the Break Down profile, informs us of the impact of the most important variables of the computed prediction. However, instead of focusing on the clear representation of the outcome, it represents the uncertainty of the impact of presented variables. In Figure 7 (b) we can see that in terms of latitude, the uncertainty is quite big and it can even change its sign.

To see how our models work, we've decided to compare two predictions that have different characteristics for each of the models. Break Down profile and SHAP plots for Random Forest observations are presented on Figures 6 and 7 and for XGBoost on Figures 8 and 9.

From the first pair of plots for the Random Forest model (Figure 6), we can see that the biggest impact on the outcome has the AreaNet variable, however, for the first observation it makes the price of the estate go up and in the second case, it makes it go down. We can also see that having only one bathroom instead of three or one bedroom instead of five, make the price go down too. From the first plot, we can also see that the estate being a duplex with a property subtype apartment contributed over 250 thousand to the price and is one of the most important variables for the first object.

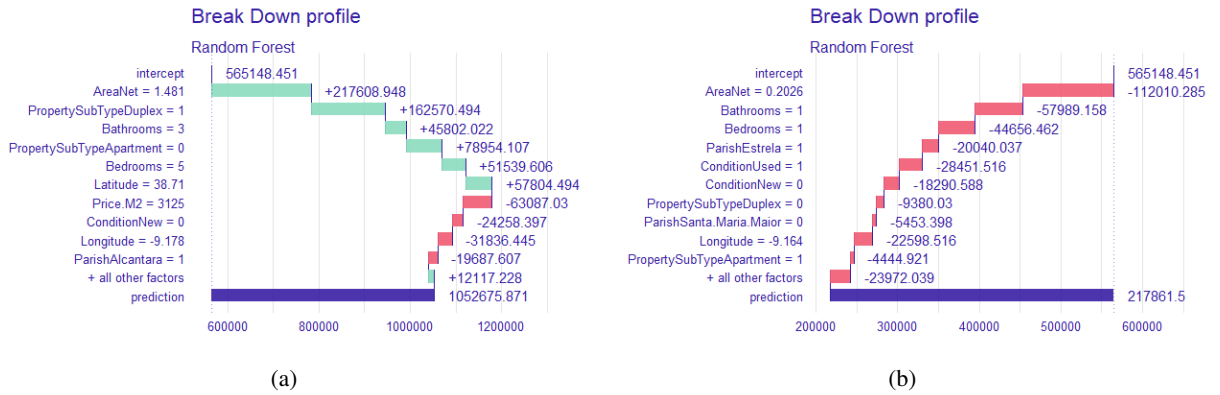


Figure 6: Random Forest Break Down profiles for: (a) expensive estate (b) cheap estate

These observations are highlighted by the pair of SHAP plots (Figure 7). We can see that the top 5 most important variables for the two observations differ. For the first estate, these are AreaNet, PropertySubTypeDuplex, PropertySubTypeApartment, Bedrooms and Bathrooms, whereas for the second one these are: AreaNet, Bathrooms, Bedrooms, ParishEstrela and ConditionUsed.

The first pair of plots for the XGBoost model (Figure 8) presents us an interesting example where both real estates have a pretty small area, which contributes negatively to the final value, however, the second one, due to its good localization and great condition manages to finally exceed average price, whereas the first ones value only goes down with other variables.

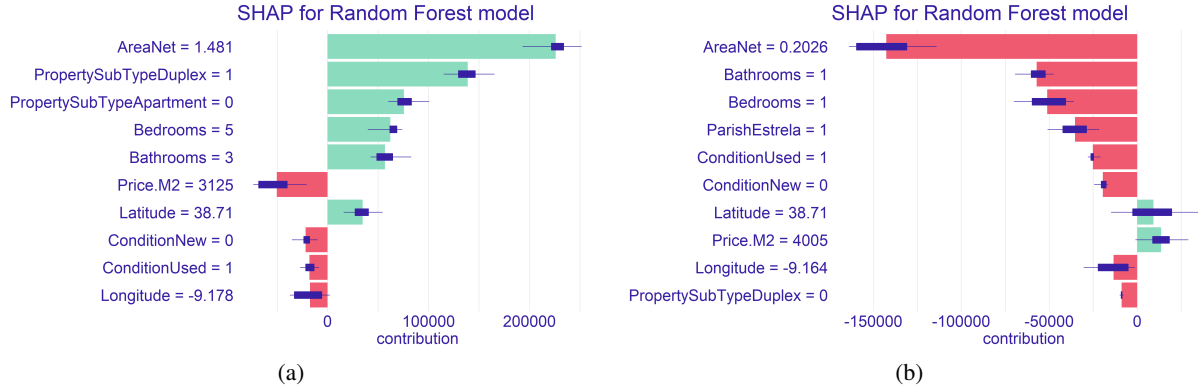


Figure 7: Random Forest SHAP profiles for: (a) expensive estate (b) cheap estate

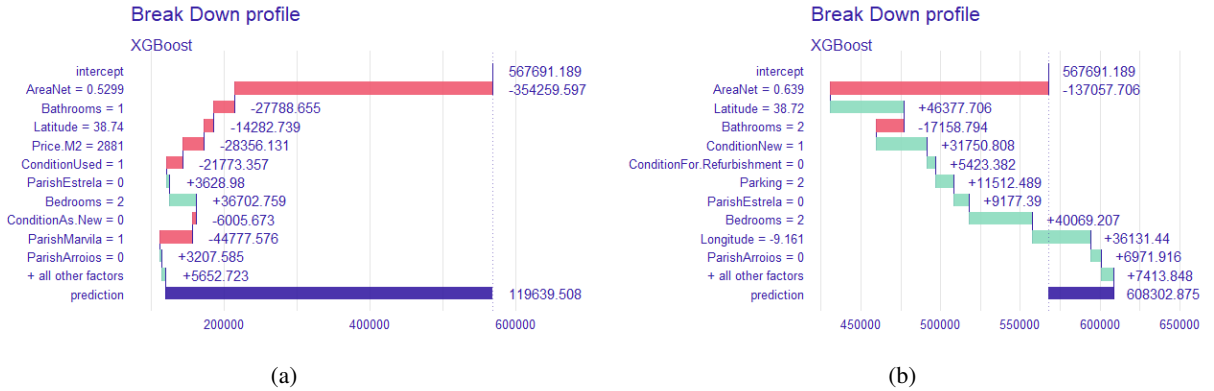


Figure 8: XGBoost Break Down profiles for: (a) cheap estate (b) regular estate

In this case, from SHAP plots (Figure 9) we can see that the top 5 most important variables for both observations are a permutation of one another, however they have a dramatically different outcome in the end. From the analysis of these plots we can say, that due to the better condition of the second house, its price per square meter goes up, which makes the whole price go up too.

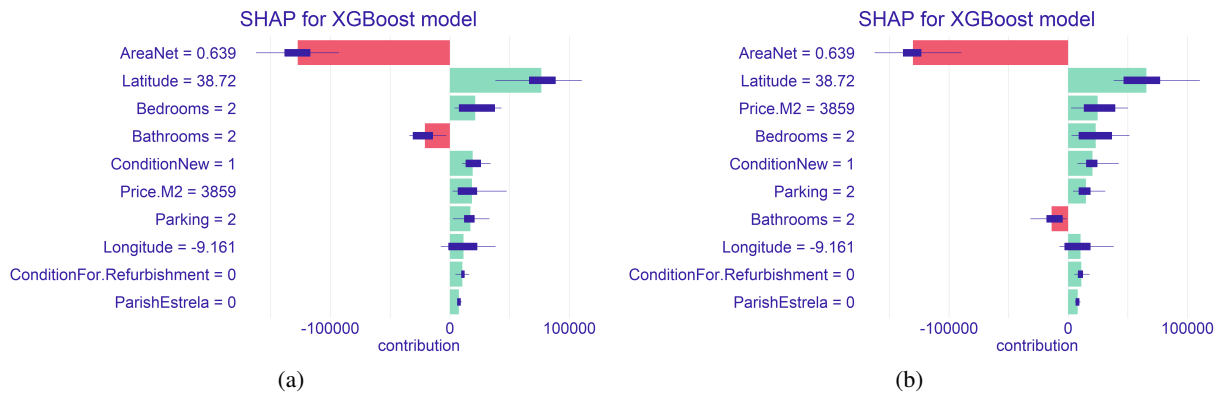


Figure 9: XGBoost SHAP profiles for: (a) cheap estate (b) regular estate

4.1.2 Ceteris Paribus

Another local method used in our work is the Ceteris Paribus profile introduced by (van Benthem [2008]). This method represents how the prediction will change if we take a single variable and change its value (with other variables staying the same), this kind of visualization lets us answer some what-if questions.

In the example of CP plot for Random Forest model (Figure 10) we will analyse how prediction will change with the difference in AreaNet, ConditionFor.Refurbishment, Latitude and Longitude variables. The aforementioned analysis is conducted on two apartments, the cheap one and an expensive one. As we can see with the growth of AreaNet, the price also grows for both estates. We can also see that if the condition of the expensive house was worse, then its price would go down a bit. Another interesting observation is that if the cheaper estate was more in the north, which would be further from the city centre, its value would go down. From the last subplot, we can see that for the expensive house, localization more on the east is beneficial, because of being in the city centre.

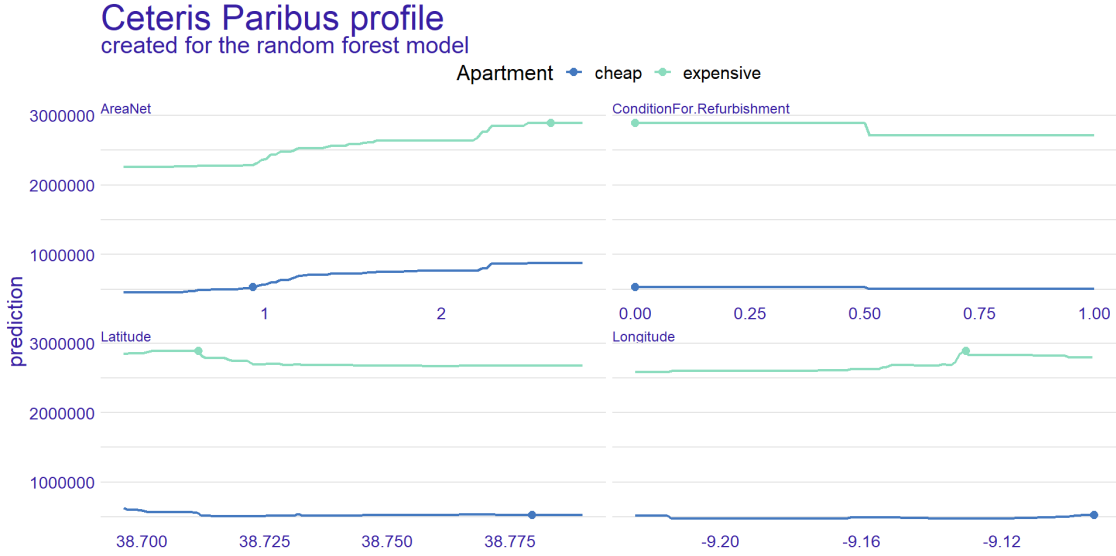


Figure 10: Ceteris Paribus profiles comparison for cheap and expensive real estates, created for Random Forest model

In the example of CP plot for XGBoost model (Figure 11) we will focus on comparisons with the first model and its behaviours, because general trends like price growth with bigger AreaNet or better condition, meaning higher price stays the same.

The interesting fact here is the limit of price prediction according to the AreaNet. The price grows for both estates with the growth in AreaNet, however, its limit is much smaller for the cheap estate than for the expensive one. It is due to a lack of observations for the model that has a huge AreaNet value and other conditions similar to our observation. We must remember that when the AreaNet value grows, the other variables, like the number of bedrooms stay the same which makes observations with unnatural values. Moreover, the flattening there exists in the xgboost model, because, both models were preprocessed by different pipelines, dedicated to their model, so they also have different observations in the training and testing sets and that's why the X-axis ends here with 4 values and for the previous example it ended around 3.

Another interesting observation is that the XGBoost model lowers the prices on the outskirts of the city, which is visible in the Longitude analysis.

4.2 Global methods

4.2.1 Feature Importance

After local analysis, we conduct the first global method which is Feature Importance (Musolf [2021]). It measures the importance of the variable V . To measure that we will compare the initial model with the model in which the effect of the variable V is removed. We obtain such an effect through a random reshuffling of the data. We take the original data, then we permute it and get new data, on which we calculate the prediction. If a variable is important in a model, then after its permutation the model prediction should be less precise, so the RMSE would be bigger.

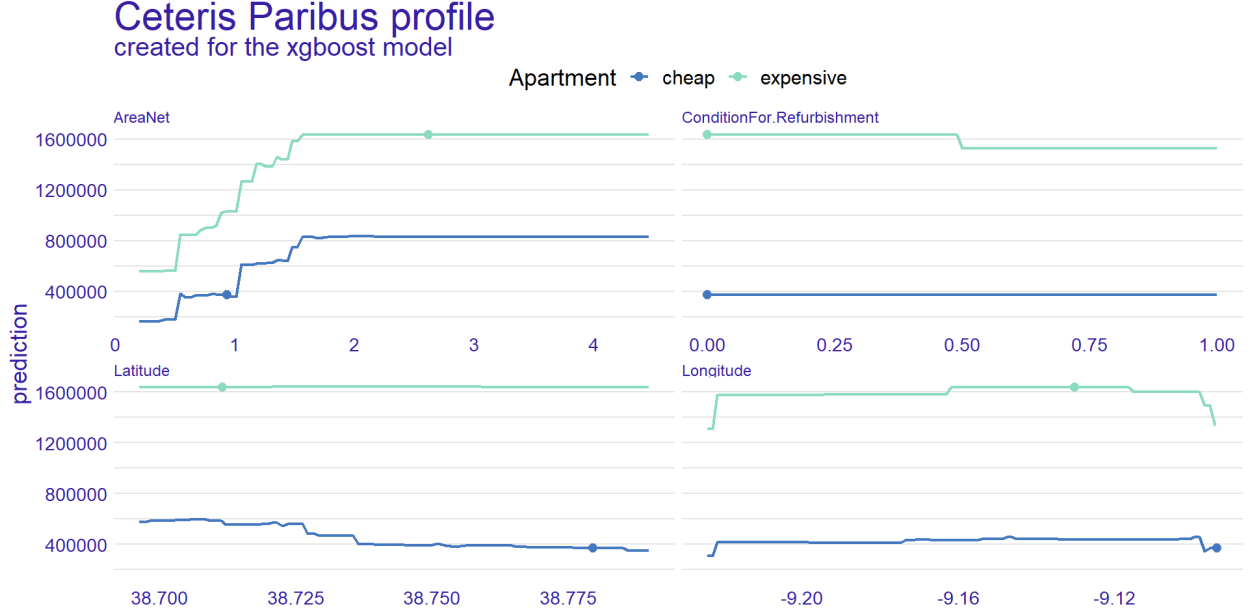


Figure 11: Ceteris Paribus profiles comparison for cheap and expensive real estates, created for XGBoost model

For our chosen models presented in Figure 12 we can see that the most important variables are quite similar, yet slightly different. For both of them the most important are AreaNet and Bathrooms, however for Random Forest the next ones are Bedrooms, ConditionNew and Longitude, whereas for XGBoost these are Latitude, Longitude and Price.M2. We shouldn't be surprised by the occurrences of these variables, because they also were really important during SHAP analysis for particular observations.

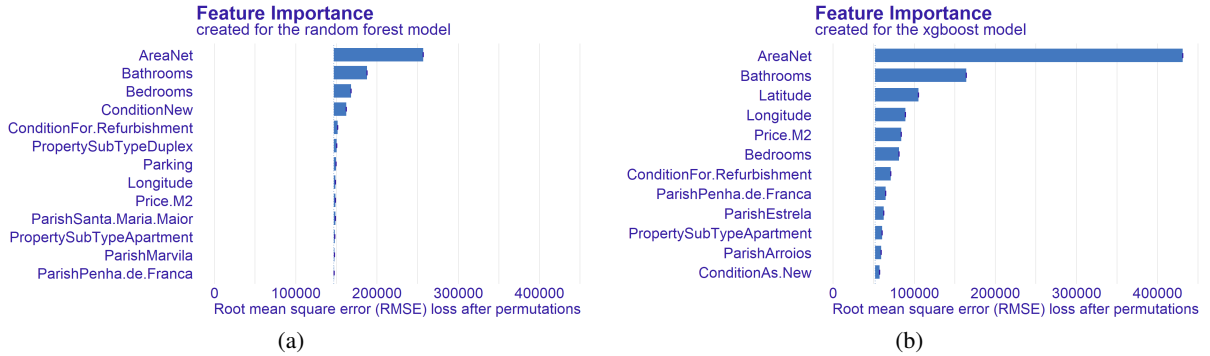


Figure 12: Feature Importance for: (a) Random Forest model (b) XGBoost model

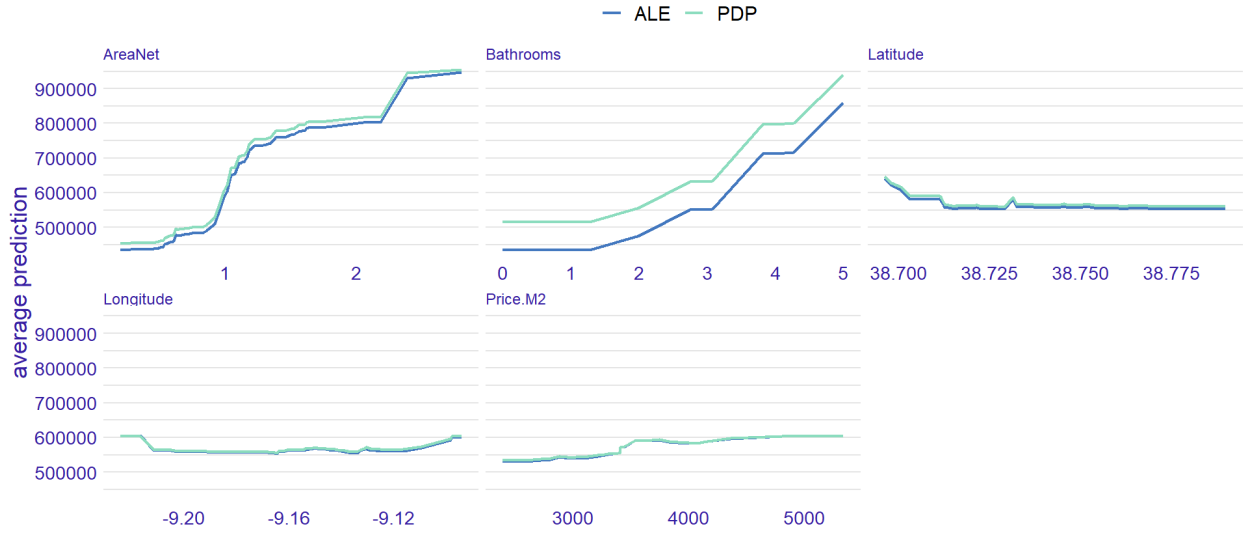
4.2.2 PDP and ALE

The last part of our explainable AI analysis is PDP (Amizadeh et al. [2019]) and ALE plots. These two are similar to the Ceteris Paribus profile, so they describe how the mean prediction would change if we set a different value of the selected variable. A partial Dependence profile is calculated as a mean of several CP profiles, however, this method has problems where some of the variables are correlated (like it is in our data set). That's why we decided to compare PDP plots with the ALE plots, which eliminate the effect of correlated variables.

In Figure 13 we can see that in our case, these plots are parallel to each other, which suggests that the model is additive for these explanatory variables. From these plots, we can see the difference between the split data sets, because X-axis for the AreaNet subplot, has a bigger range for xgboost. We can also see that number of bathrooms is indeed an

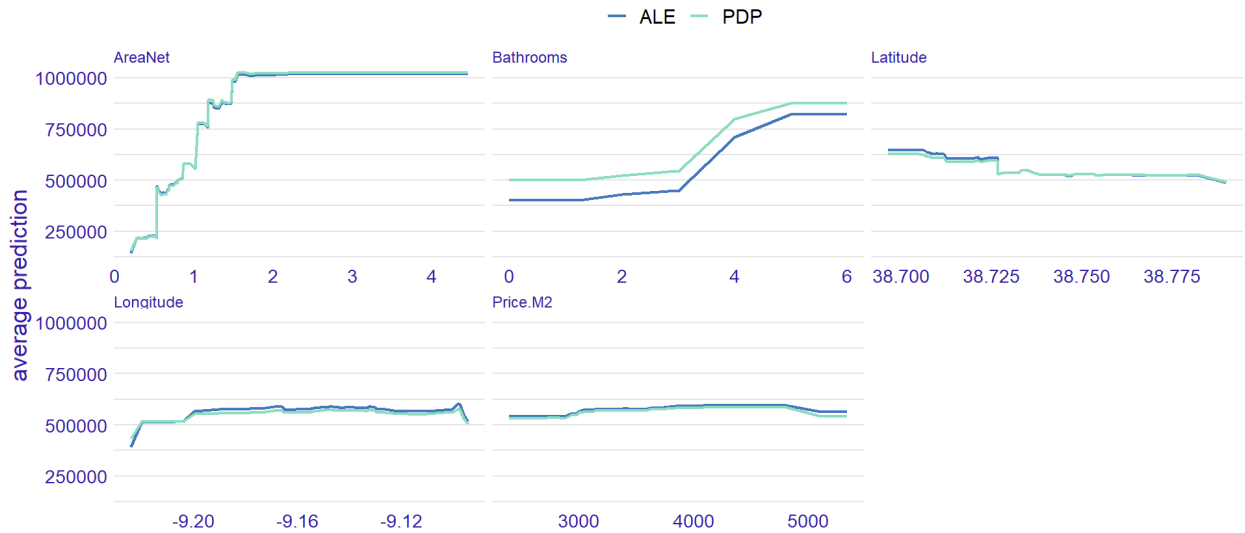
important price factor for both of the models and southern locations in the city, which are closer to the city centre are clearly preferred. A surprising fact is that price per meter squared is relatively not important and it doesn't change the overall value so much. The only visible and really interesting difference is that the Random Forest model gives higher values for the estates on the west or east of the city, whereas the XGBoost model prefers ones in the middle of the city.

Comparison of Partial Dependence Profiles (PDP) and Accumulated Local Dependence (ALE) profiles for random forest



(a)

Comparison of Partial Dependence Profiles (PDP) and Accumulated Local Dependence (ALE) profiles for xgboost



(b)

Figure 13: PDP and ALE plots for: (a) Random Forest model (b) XGBoost model

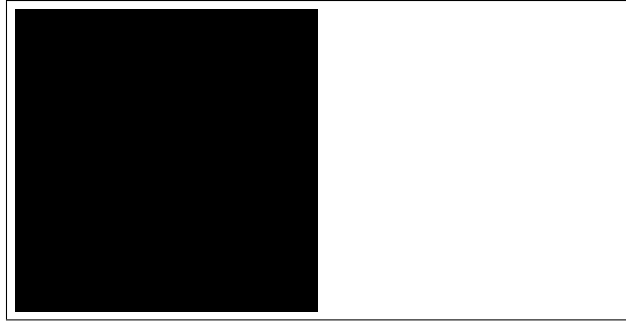


Figure 14: Sample figure caption.

Table 3: Sample table title

Part		
Name	Description	Size (μm)
Dendrite	Input terminal	~ 100
Axon	Output terminal	~ 10
Soma	Cell body	up to 10^6

5 Examples of citations, figures, tables, references

5.1 Citations

Citations use natbib. The documentation may be found at

<http://mirrors.ctan.org/macros/latex/contrib/natbib/natnotes.pdf>

Here is an example usage of the two main commands (`citet` and `citep`): Some people thought a thing [??] but other people thought something else [?]. Many people have speculated that if we knew exactly why ? thought this. . .

5.2 Figures

Suspendisse vitae elit. Aliquam arcu neque, ornare in, ullamcorper quis, commodo eu, libero. Fusce sagittis erat at erat tristique mollis. Maecenas sapien libero, molestie et, lobortis in, sodales eget, dui. Morbi ultrices rutrum lorem. Nam elementum ullamcorper leo. Morbi dui. Aliquam sagittis. Nunc placerat. Pellentesque tristique sodales est. Maecenas imperdiet lacinia velit. Cras non urna. Morbi eros pede, suscipit ac, varius vel, egestas non, eros. Praesent malesuada, diam id pretium elementum, eros sem dictum tortor, vel consectetur odio sem sed wisi. See Figure 14. Here is how you add footnotes. ¹ Sed feugiat. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Ut pellentesque augue sed urna. Vestibulum diam eros, fringilla et, consectetur eu, nonummy id, sapien. Nullam at lectus. In sagittis ultrices mauris. Curabitur malesuada erat sit amet massa. Fusce blandit. Aliquam erat volutpat. Aliquam euismod. Aenean vel lectus. Nunc imperdiet justo nec dolor.

5.3 Tables

See awesome Table 3.

The documentation for booktabs (‘Publication quality tables in LaTeX’) is available from:

<https://www.ctan.org/pkg/booktabs>

5.4 Lists

- Lorem ipsum dolor sit amet

¹Sample of the first footnote.

- consectetur adipiscing elit.
- Aliquam dignissim blandit est, in dictum tortor gravida eget. In ac rutrum magna.

References

- Mateusz Staniak and Przemysław Biecek. Explanations of Model Predictions with live and breakDown Packages. *The R Journal*, 10(2):395–409, 2018. URL <https://doi.org/10.32614/RJ-2018-072>.
- Scott M Lundberg and Su-In Lee. A unified approach to interpreting model predictions. In I. Guyon, U. Von Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017. URL <https://proceedings.neurips.cc/paper/2017/file/8a20a8621978632d76c43dfd28b67767-Paper.pdf>.
- Girard P. Roy O. van Benthem, J. Everything else being equal: A modal logic for ceteris paribus preferences. 2008. doi:<https://doi.org/10.1007/s10992-008-9085-3>.
- Holzinger E.R. Malley J.D. et al Musolf, A.M. What makes a good prediction? feature importance and beginning to open the black box of machine learning in genetics. 2021. URL <https://doi.org/10.1007/s00439-021-02402-z>.
- Saeed Amizadeh, Sergiy Matushevych, and Markus Weimer. Pdp: A general neural framework for learning constraint satisfaction solvers, 2019. URL <https://arxiv.org/abs/1903.01969>.