PREDICTING LISBON HOUSE PRICES WITH TREE-BASED MODELS

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

June 1, 2022

ABSTRACT

This paper presents the effects of the study describing the real estate situation in Lisbon. This topic includes data analysis, the creation of tree-based models and their explanation. These goals were obtained by conducting thorough Exploratory Data Analysis, and training of Random Forest, XGBoost, LightGBM and CatBoost models with the Bayesian Optimization method and explaining these models with DALEX. Best Random Forest and XGBoost models achieve satisfactory results with decent RMSE and MAE values and are not overfitted. XAI analysis on the other hand showed us the importance of the area net, the number of bathrooms and bedrooms and the localization of the property for predicting its price. This paper provides a detailed analysis of this problem and demonstrates the machine learning and explainable AI methods, which lead us to excellent results.

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 (Rodrigues [2021]) data set followed by the creation of the models, which are later explained with the usage of the DALEX package (Biecek [2018]).

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),

- **PropertySubType**: Property Sub Type (i.e., Apartment, duplex, etc.),
- **Bedrooms**: Number of Bedrooms,
- **Bathrooms**: Number of Bathrooms,
- **AreaNet**: Net area of the house,
- **AreaGross**: Gross area of the house,
- **Parking**: Number of parking places,
- **Latitude**: Geographical Latitude,
- **Longitude**: Geographical Longitude,
- **Country**: Country where the house is located,
- **District**: District where the house is located,
- **Municipality**: Municipality where the house is located,
- **Parish**: Parish where the house is located,
- **Price Sq. M.**: Price per m² in the location of the house,
- **Price**: This is our training variable and target. It is the homes price.

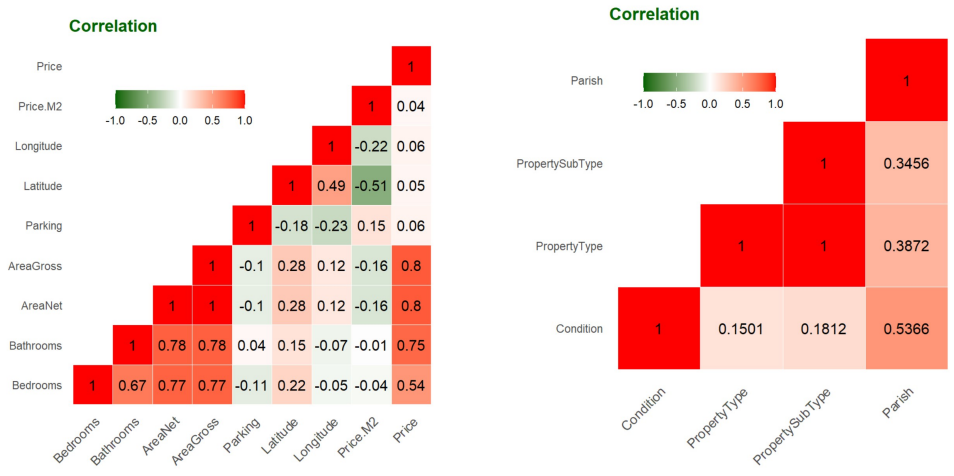
2.1 Exploratory Data Analysis

2.1.1 Dependence of numeric variables

To know more about the regression task that we are facing, we decided to conduct an Exploratory Data Analysis. We've created correlation heatmaps for numeric and categorical variables separately (Figure 1). Thanks to this method we were able to see the high dependence indexes between *Bathrooms* and *Area Net*, *Bathrooms* and *Area Gross* and most importantly, an indistinguishable differences between *AreaNet* and *AreaGross*. The latter ones give us the same information (correlation index equal 1), so we decided to truncate *AreaGross* for the training process.

2.1.2 Dependence of categorical variables

The set of categorical variables is much smaller and consists of 4 variables only, however, we've managed to see another important dependence in the data set. The creation of Crammers V Correlation Heatmap (method introduced in Cramér [1946]) showed us a strict dependence between *PropertyType* and *PropertySubType* variables. A deeper analysis showed us that it is caused by the absolute dominance of Apartment in *PropertySubType* variable.

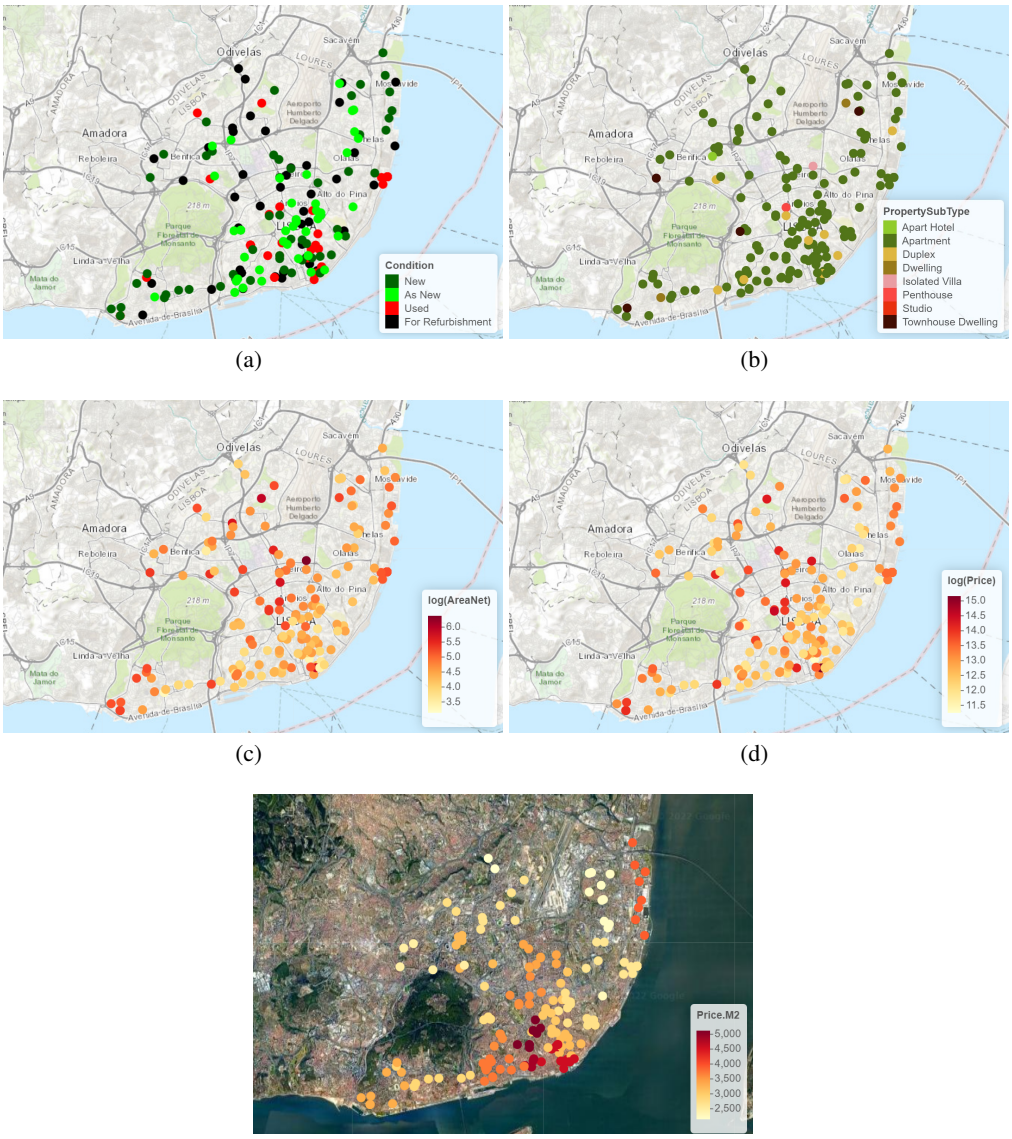


(a) (b)

Figure 1: Correlation Heatmap of: (a) numerical variables (b) categorical variables

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 (Figure 2). 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 there.



(a) (b) (c) (d) (e)

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

This analysis gives us a basic understanding of 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' package (Wilson [2021]). 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 hundreds of 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 Breiman [2001], XGBoost Tianqi Chen [2016], LightGBM Ke et al. [2017] and CatBoost Liudmila Prokhorenkova [2017]. 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.

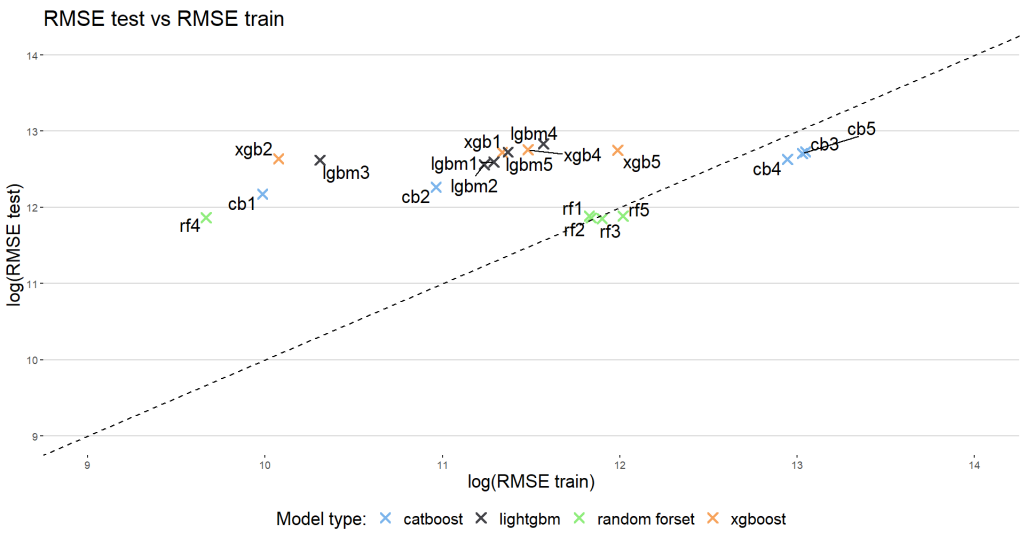


Figure 3: Train and test RMSE values comparison for all trained models in logarithmic scale

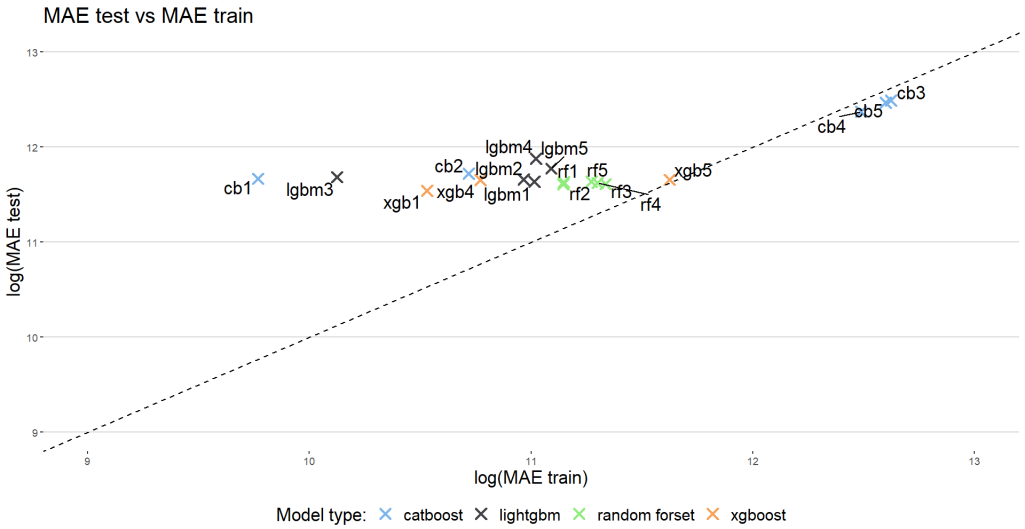


Figure 4: Train and test MAE values comparison for all trained models in logarithmic scale

- 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.

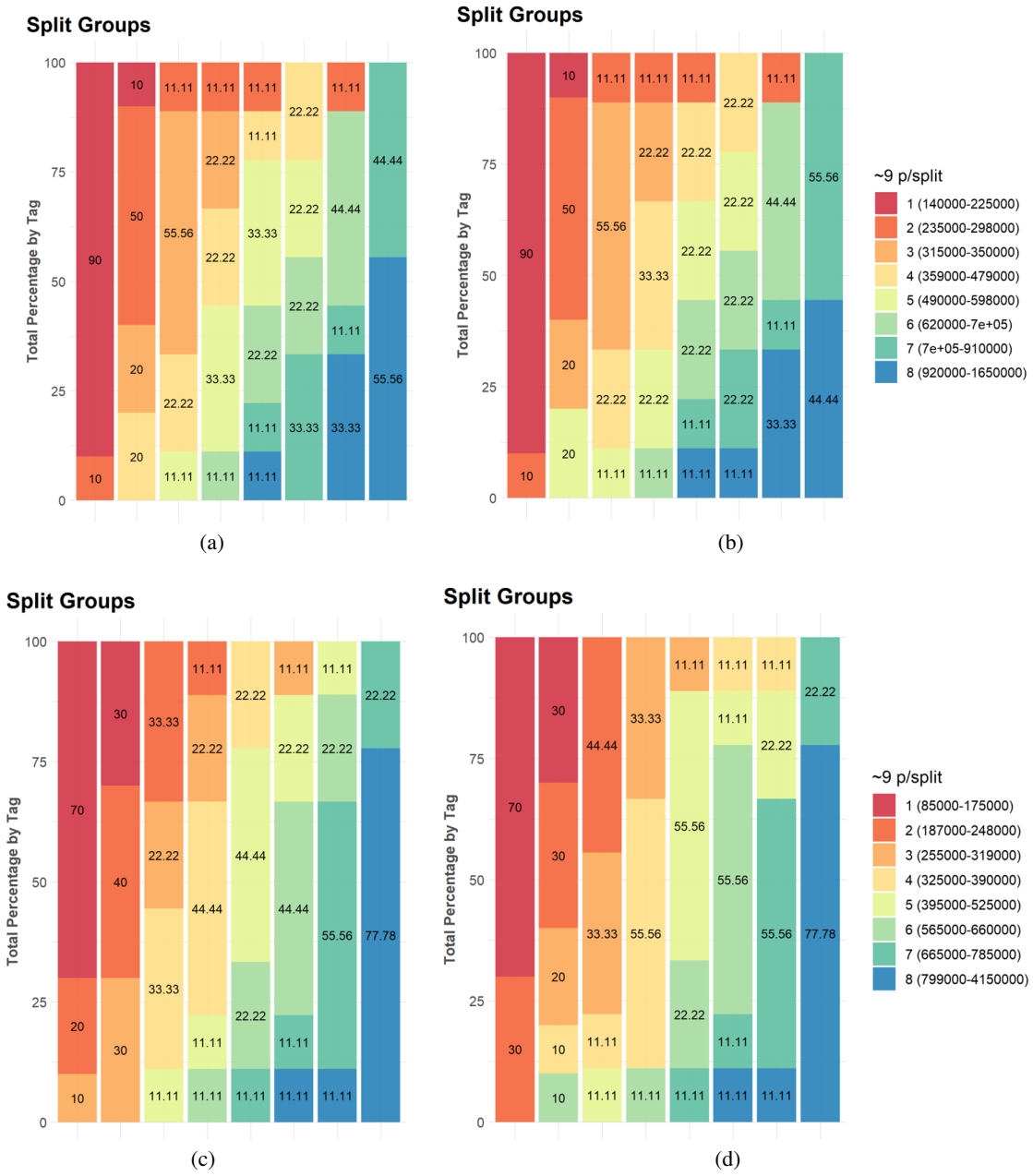


Figure 5: Split groups plots for: (a) Random Forest 2 (b) Random Forest 3 (c) XGBoost 1 (d) XGBoost 5

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. The first 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

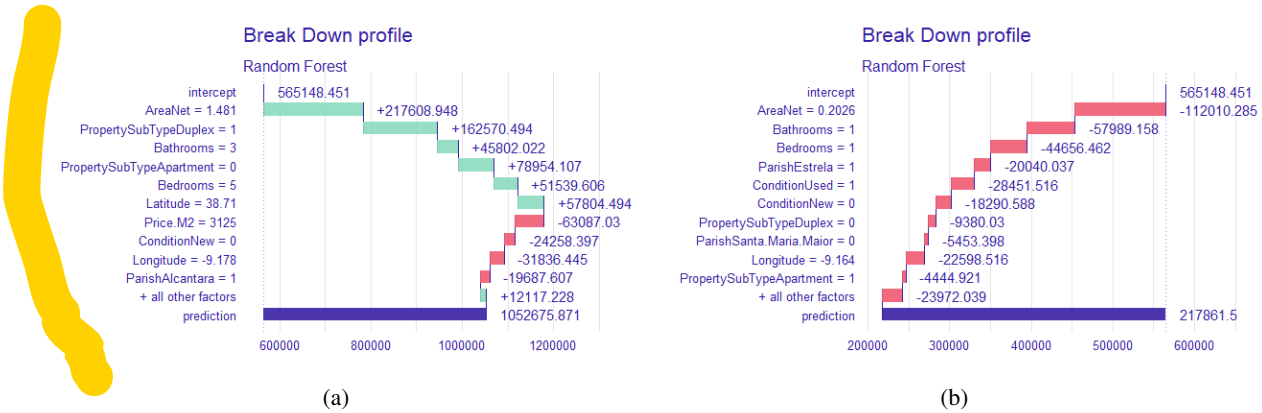
4.1.1 Break Down profile and SHAP

The first local method is a Break Down profile which is similar to the method (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 (Lundberg and Lee [2017]) are based on "Shapley values". 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 sub type apartment contributed over 250 thousand to the price and is one of the most important variables for the first object.

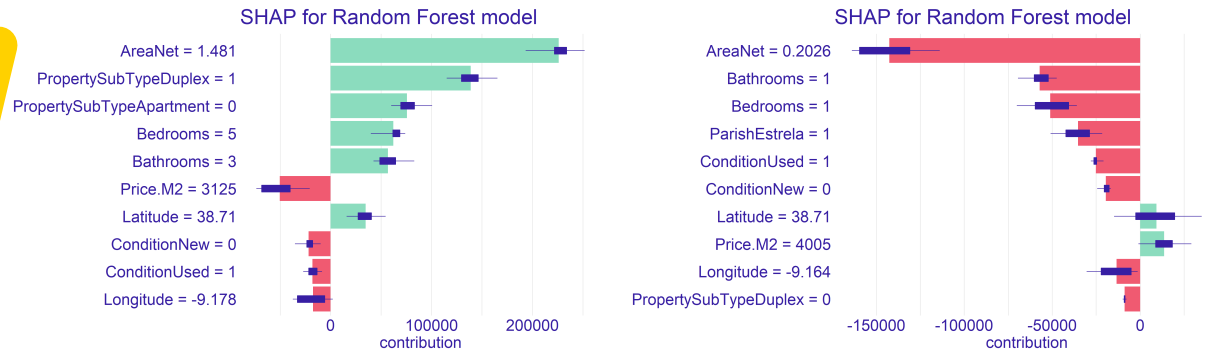


(a) (b)

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

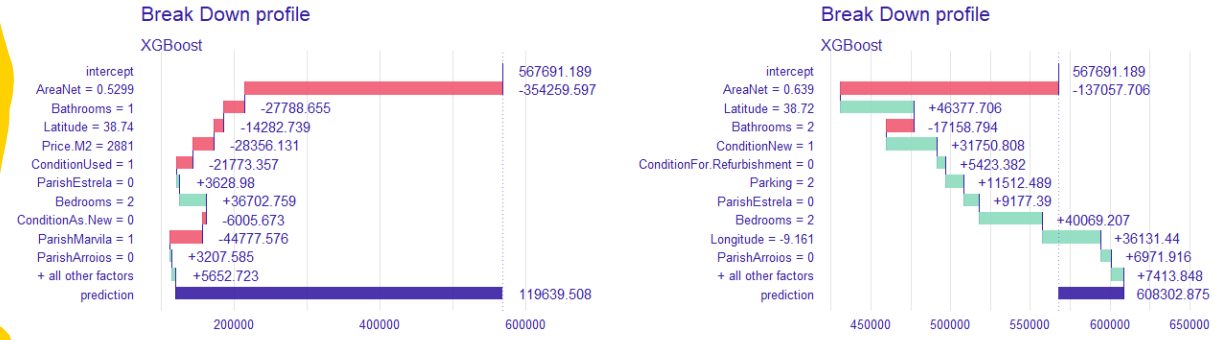
These observations are highlighted by the pair of SHAP plots (Figure 7). We can see that the top 5 most important variables for the two observations differ. For the first estate, these are *AreaNet*, *PropertySubTypeDuplex*, *PropertySubTypeApartment*, *Bedrooms* and *Bathrooms*, whereas for the second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



(a) (b)

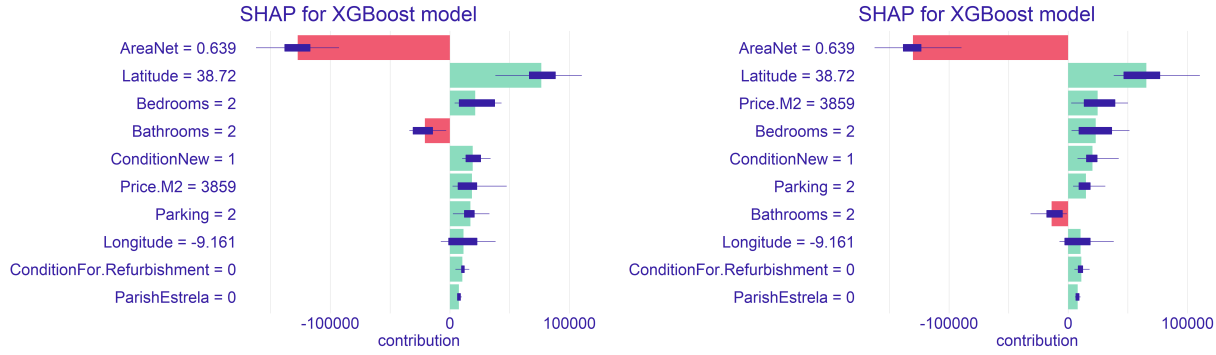
Figure 7: Random Forest SHAP profiles for: (a) expensive estate (b) cheap estate



(a) (b)

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.



(a) (b)

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.

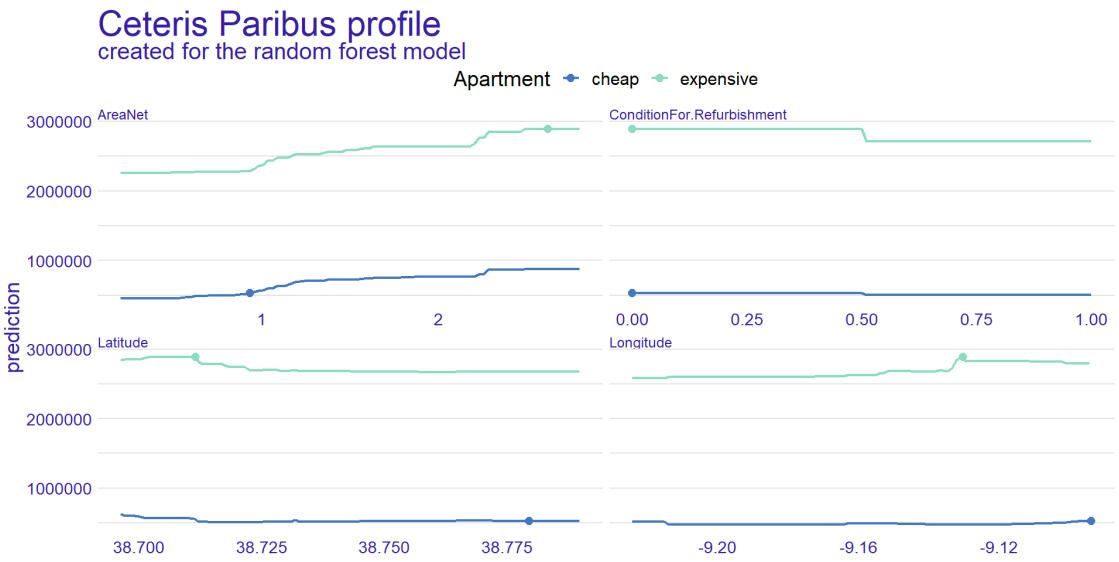


Figure 10: Ceteris Paribus profiles comparison for cheap and expensive real estates, created for Random Forest model

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.

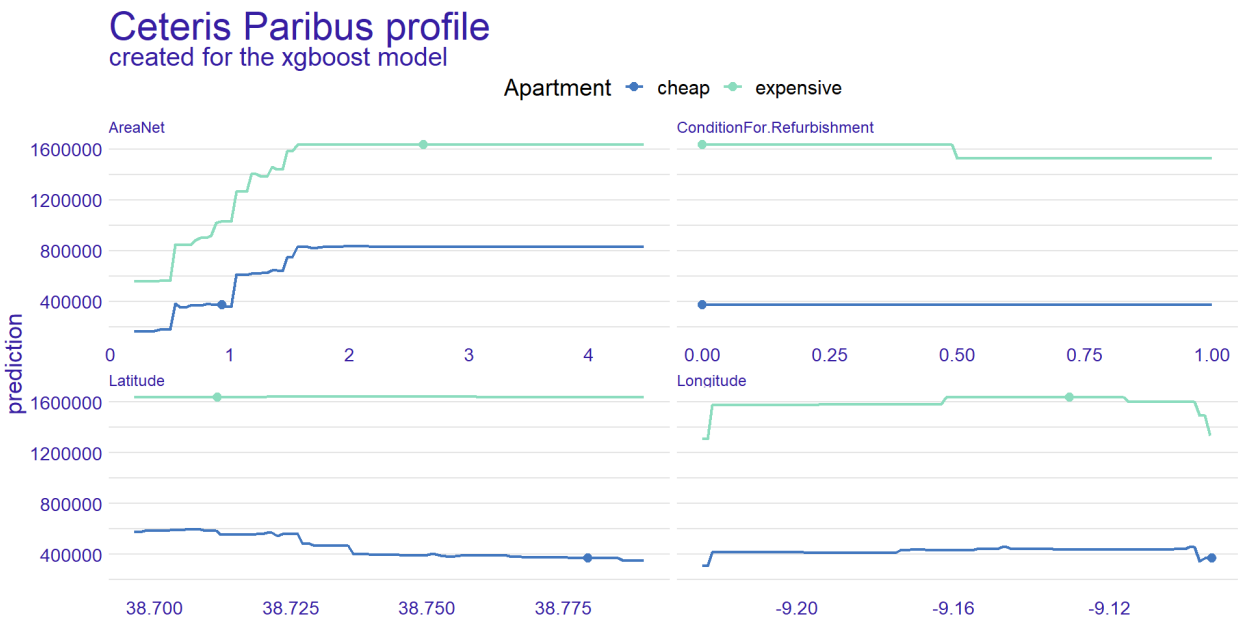
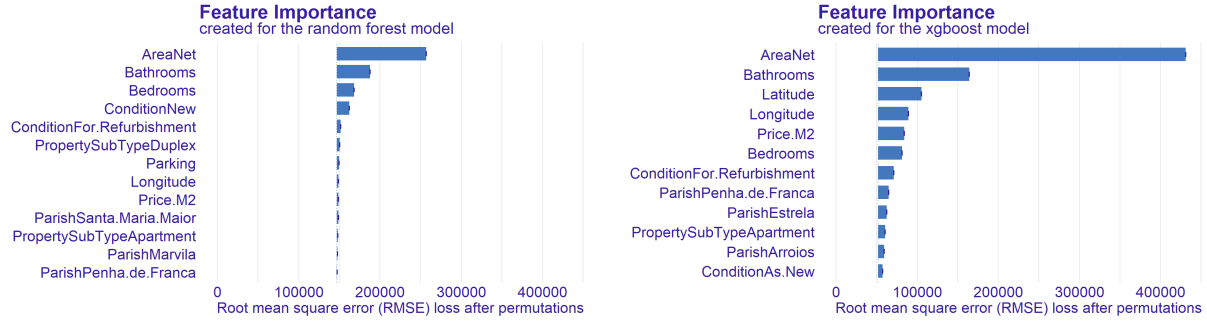


Figure 11: Ceteris Paribus profiles comparison for cheap and expensive real estates, created for XGBoost model

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.



(a) (b)

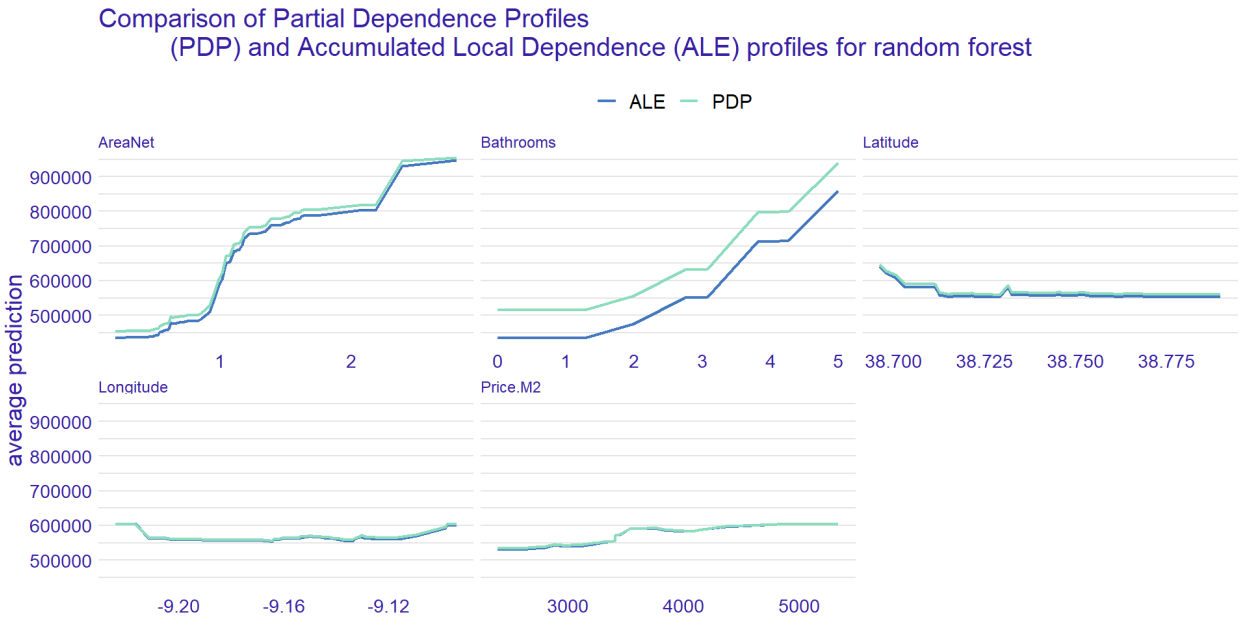
Figure 12: Feature Importance for: (a) Random Forest model (b) XGBoost model

4.2.2 PDP and ALE

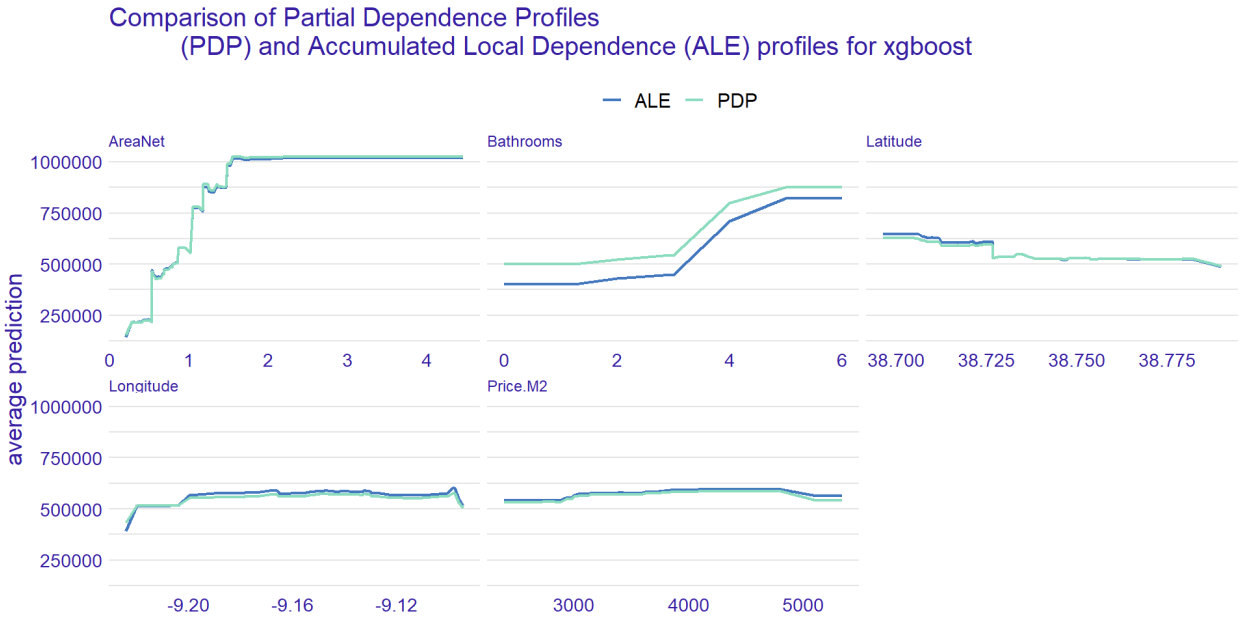
The last part of our explainable AI analysis is PDP (Amizadeh et al. [2019]) and ALE plots. These two are similar to the Ceteris Paribus profile, so they describe how the mean prediction would change if we set a different value of the selected variable. A Partial Dependence profile is calculated as a 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.



(a)



(b)

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

4.2.3 Summary

In the end, we can confront XAI analysis with the EDA, conducted at the beginning of this article. As we identified *AreaNet*, *Bathrooms*, *Bedrooms*, general condition of the apartment and most importantly, its localization as the most important factors, we should focus on Figure 2 during this comparison.

From PDP and ALE profile present in Figure 13, we can see that estate prices go up if they are more south and closer to the city centre, which is reflected via map (e), picturing price per square meter. However, the lack of resemblance in the visualization regarding the final *Price* is finally understandable. Our models highlight that the *AreaNet* is indisputably the most important variable and if we look at maps (c) and (d) we observe that they are quite similar due to the aforementioned relation. We can also link big *Bathrooms*' impact on the price, because in general, the bigger the house area is, the larger the number of bathrooms.

5 Conclusions

In this paper, we've thoroughly analysed the problem of house price prediction in Lisbon. We've created hundreds of tree models via Bayesian Optimization to obtain the final two well-performing models. These models enable us to correctly estimate Lisbon property prices. We've also explained these models with DALEX visualizations, which showed us the importance and dependence of houses price on particular variables. With the help of EDA, we can characterize this market in a few sentences. Real estate prices in Lisbon depend mostly on their area, which is also indicated by the number of bedrooms and bathrooms. Properties condition and localization closer to the city centre also have a big impact on the final price, however, it scales directly with the estate's area.

References

Carlos Garces Rodrigues. Lisbon house prices, 2021. URL `https://www.kaggle.com/datasets/cgrodrigues/lisbon-house-prices`.

Przemyslaw Biecek. Dalex: Explainers for complex predictive models in r. *Journal of Machine Learning Research*, 19 (84):1–5, 2018. URL `https://jmlr.org/papers/v19/18-416.html`.

Harald Cramér. *Mathematical Methods of Statistics*. Princeton University Press, 9 edition, 1946. doi:10.1515/9781400883868.

Samuel Wilson. Parallelizable bayesian optimization, 2021. URL `https://www.rdocumentation.org/packages/ParBayesianOptimization/versions/1.2.4`.

Leo Breiman. Random forests. 2001. doi:10.1023/A:1010933404324.

Carlos Guestrin Tianqi Chen. Xgboost: A scalable tree boosting system. pages 785–794, 2016. doi:10.1145/2939672.2939785.

Guolin Ke, Qi Meng, Thomas Finley, Taifeng Wang, Wei Chen, Weidong Ma, Qiwei Ye, and Tie-Yan Liu. Lightgbm: A highly efficient gradient boosting decision tree. 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/6449f44a102fde848669bdd9eb6b76fa-Paper.pdf`.

Aleksandr Vorobev Anna Veronika Dorogush Andrey Gulin Liudmila Prokhorenkova, Gleb Gusev. Catboost: unbiased boosting with categorical features. 2017. doi:10.48550/arXiv.1706.09516.

Mateusz Staniak and Przemysław Biecek. Explanations of Model Predictions with live and breakDown Packages. *The R Journal*, 10(2):395–409, 2018. doi:10.32614/RJ-2018-072.

Scott M Lundberg and Su-In Lee. A unified approach to interpreting model predictions. 30, 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: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. doi:10.1007/s00439-021-02402-z.

Saeed Amizadeh, Sergiy Matusevych, and Markus Weimer. Pdp: A general neural framework for learning constraint satisfaction solvers. 2019. URL `https://arxiv.org/abs/1903.01969`.