PREDICTING REAL ESTATE PRICES IN MONTREAL USING MACHINE LEARNING METHODS

Maria Kędzierska
Faculty of Mathematics and Information Technology
Warsaw University of Technology
Warsaw, Poland
https://github.com/kaluskam

Szymon Gut
Faculty of Mathematics and Information Technology
Warsaw University of Technology
Warsaw, Poland
https://github.com/Szymon-Gut

Wiktor Jakubowski
Faculty of Mathematics and Information Technology
Warsaw University of Technology
Warsaw, Poland
https://github.com/WJakubowsk

Maciej Orsłowski
Faculty of Mathematics and Information Technology
Warsaw University of Technology
Warsaw, Poland
https://github.com/maciejors

May 26, 2022

ABSTRACT

Machine Learning. The indispensable part of twenty-first century technology's development. Methods of ML are helpful in many fields of our daily live. We decided to utilize ML powerful methods in analyzing real estate market in Montreal. Based on collected data, we want to predict the value of real assets in one of the biggest and most popular cities in Canada.

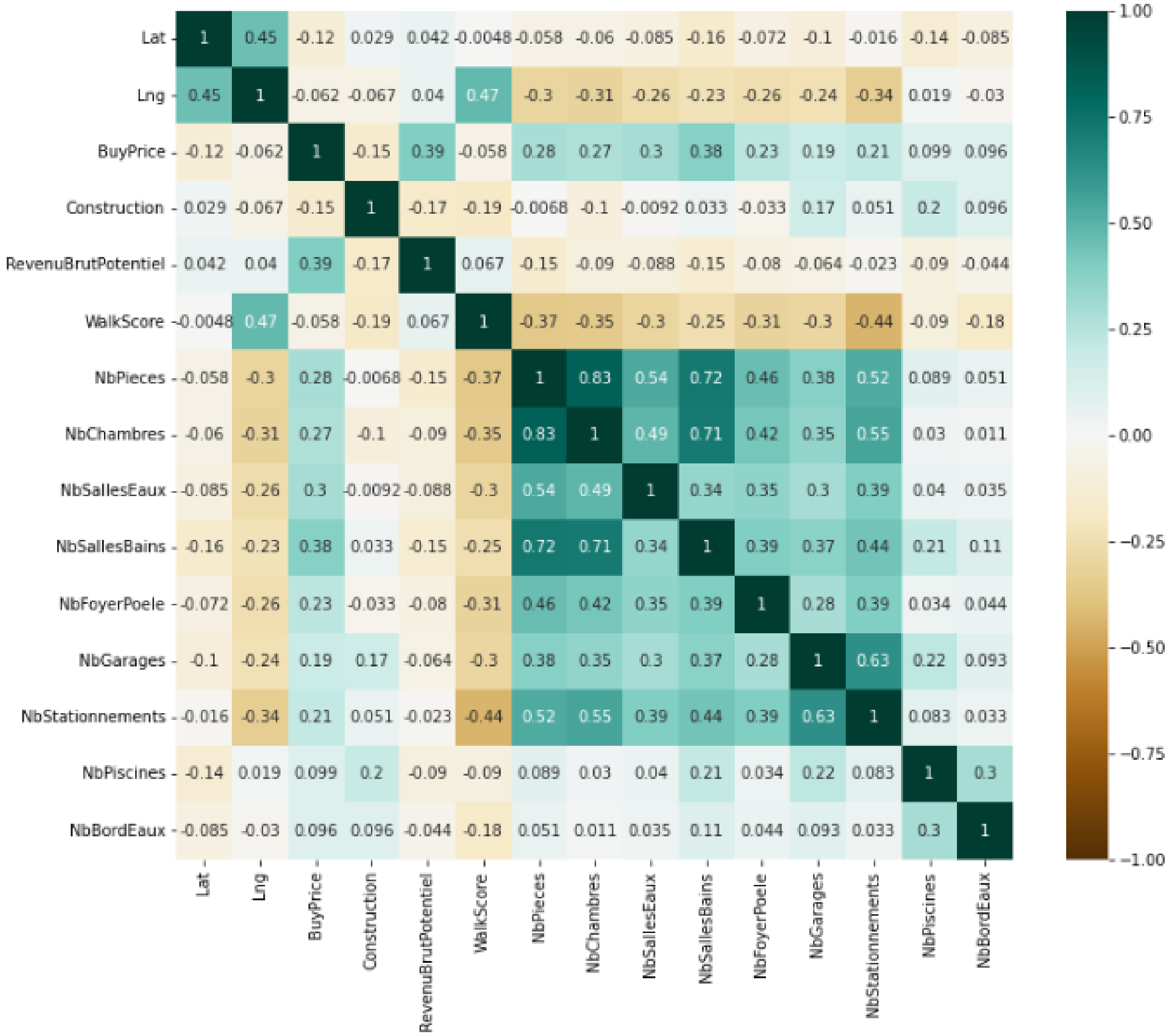
Keywords Machine Learning · Artificial Intelligence · ML models · housing market · real estate · business · predicting · Data Science

1 Introduction

2 Data

Our dataset consists of six data frames. The core data frame is 'listings.csv', as it contains key attributes for real estates sold in Montreal. During data exploratory analysis we have discovered that some columns contain largely nulls, so we decided to drop them. Additionally, columns with the same value for all records were also rejected as they did not provide any information. The variable we want to predict is BuyPrice, we observed that some records contained the price per foot or square meter while the rest contained the total price of the property. In addition, we have calculated that the purchase price is mostly correlated with the potential gross income, the number of bathrooms, bedrooms, and rooms.

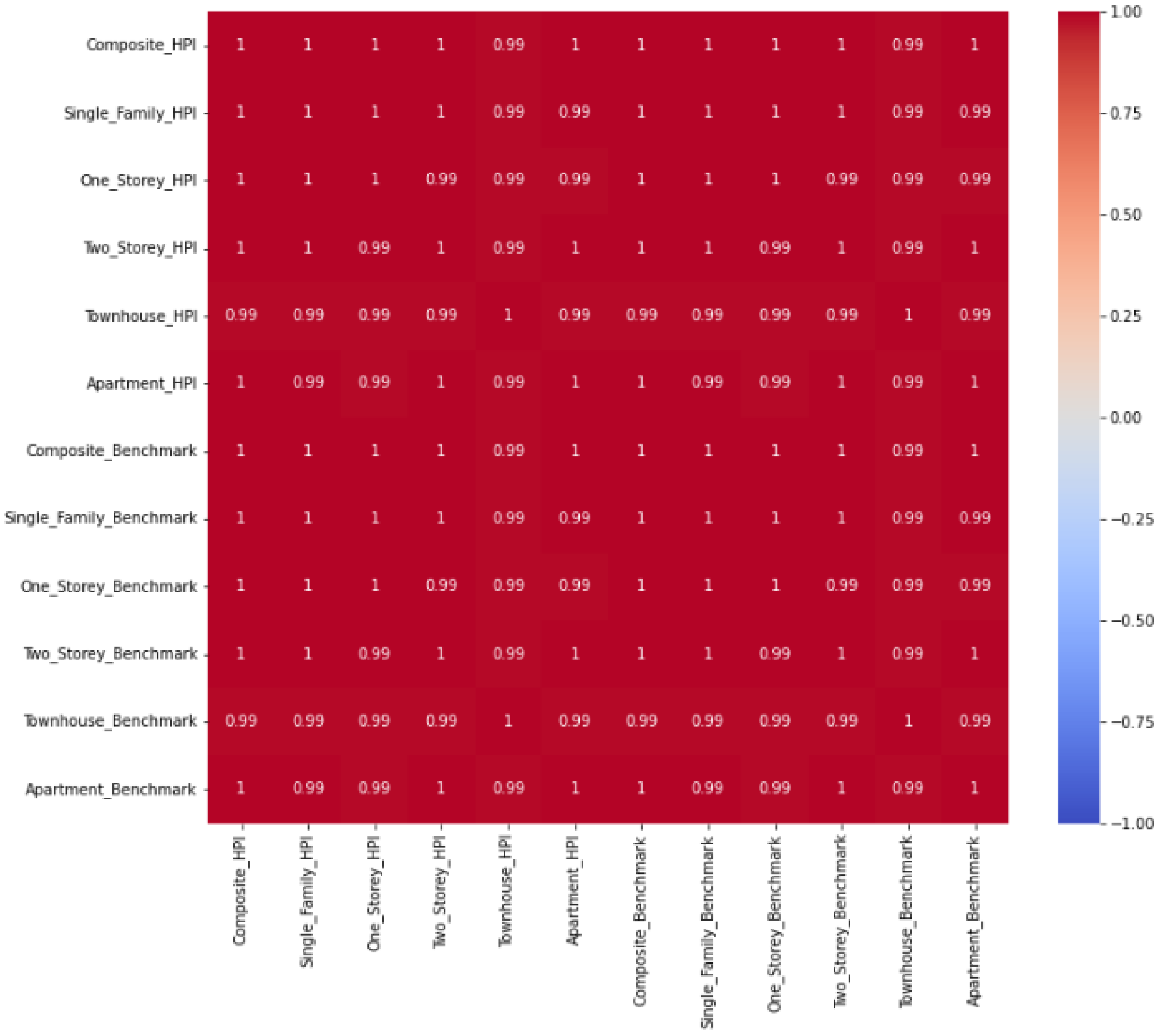
Figure 1: Correlation matrix for columns from listings data frame.



The next data frames that were analyzed were "policeCoord.csv" and "firestations.csv". These data frames only contained the coordinates of the police and fire departments. It seems useful to calculate the distance to the nearest police and fire station for each property.

Next, we examined the file "montreal_hpi.csv" which contained the house price index measures. Due to the significant correlation (greater than 0.99) between the variables, we concluded that only one column should be used from the mentioned dataframe.

Figure 2: Correlation matrix for montreal hpi data frame.



In order to add socio-demographic data, we have analyzed 'sociodemo.csv'. A data frame with information about population, nearby schools, average income, etc.

The last data frame was 'extra_data.csv' which provided additional information for some properties in 'listings.csv', unfortunately this frame was contained vast majority of null values. However, some columns, such as LivingArea, found themselves truly relevant.

3 Selected models

After exact EDA (exploratory data analysis) we started next step which was building and testing different models. We have built several RandomForest, XGBoost, LightGBM and CatBoost models. Our models were trained using Grid Search and Random Search. Then were selected only the best models based on RMSE. For each alghorithm mentioned before we have selected five models with different hyperparameters which have reached the best scores. Then we have tried PCA to reduce dimension of the dataset and this significantly improved scores of our models. Below is a table with the best scores on test data sets from the four selected algorithms.

RMSE	RandomForest	XGBoost	LightGBM	CatBoost
PCA	87 740.23	360.06	74 396.47	46 023.39
WITHOUT PCA	334 091.74	228890.90	282 358.90	246 424.10

All hyperparameters were set to avoid overfitting and underfitting so models presented above have similar scores on training and test data sets. Then we selected the best algorithm for our models, which, as we can see in the table above, turned out to be XGBoost.

Below we can see the best 3 models built on XGboost's algorithm and their hyperparameters

Model	RMSE train	RMSE test	booster	eta	max_depth	gamma	min_child_weight	subsample	colsample_bytree
xgboost 1	101.90	360.07	gbtree	0.05	7	0	1	1	1
xgboost 2	430.97	881.18	gbtree	0.09	8	4	1	1	1
xgboost 3	808.91	1 477.11	gbtree	0.12	6	6	1	1	1

Next, we created residual plots to check how our predictions looked like. What is really satisfying is that almost all predictions are very accurate. Residual plots are presented below.

Figure 3: First model

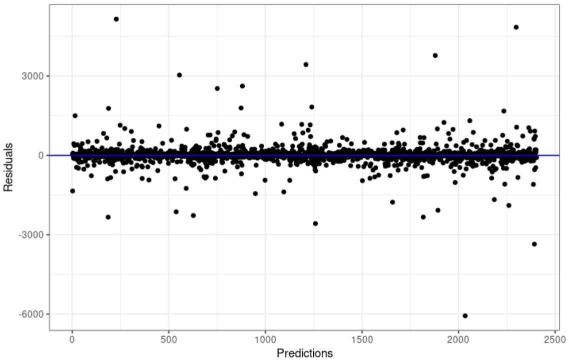


Figure 4: Second model

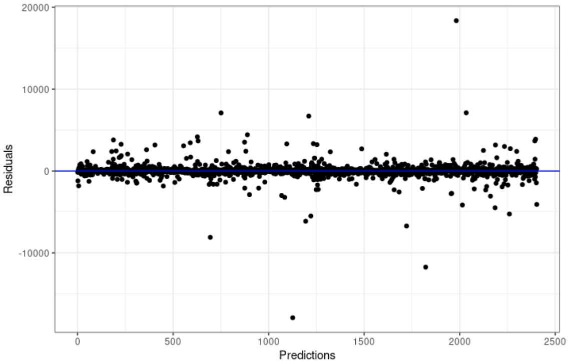
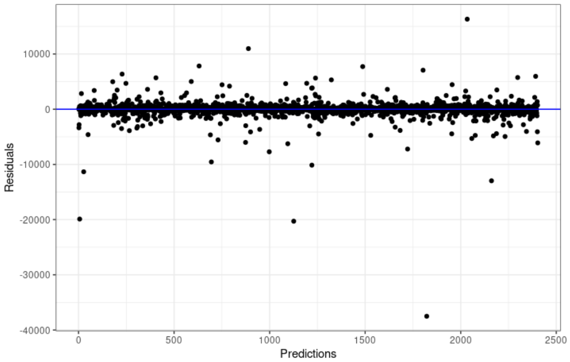


Figure 5: Third model



So as we can see these three XGBoost models were very accurate and reached very low RMSE. But how about the other metrics? That was our next step in choosing accurate model. We have measured the predictions using MAE, R^2 and the mentioned before RMSE. In the following formulas, y_i stands for real value of i-th observation in dataset and $\hat{y_i}$ stands for the model's prediction of this value.

MAE is defined as $\sum_{i=1}^{n} |y_i - \hat{y_i}|$,

R^2 is defined as $\frac{\sum_{i=1}^{n}(\hat{y_i} - \bar{y})}{\sum_{i=1}^{n}(y_i - \bar{y})}$,

and the RMSE is defined as $\sqrt{\frac{\sum_{i=1}^{n}(\hat{y_i} - y_i)^2}{n}}$

So now we can present our results:

Model name	**RMSE test**	**MAE test**	**R2**
xgboost1	360.07	135.74	0.9998625
xgboost2	881.18	289.28	0.9995856
xgboost3	1 477.11	520.25	0.9978164

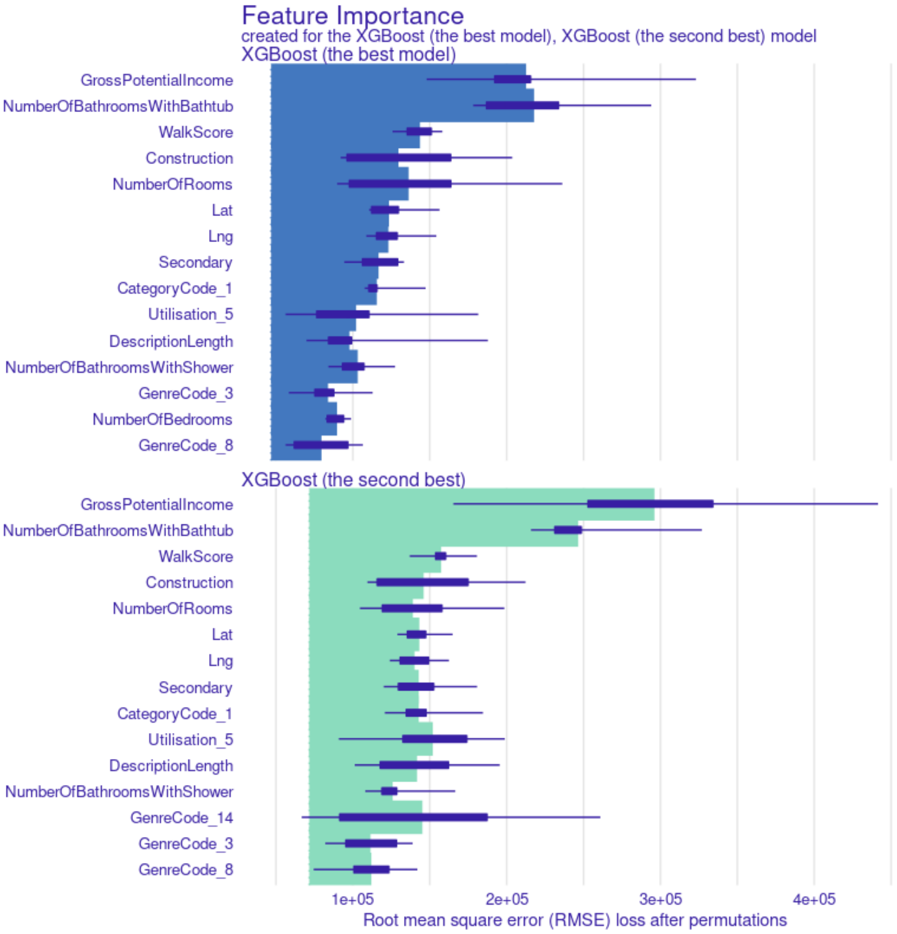
All three models achieve very good metrics and will also be further analyzed using XAI methods. But there is one change that must be performed. As we have used the PCA method to reduce dimension of our data sets we have lost the interpretability of our data. So we need to decide what is more important: increasing the accuracy of the prediction or preserving the interpretability and an insight into the influence of individual variables on the predictions. In the next steps, where we are analyzing our models using XAI methods, we are using the data set without PCA.

4 Explainability of models - XAI

In this section, we will take a closer look at the inside of two of our best XGBoost models. In order to do so, we are going to use several methods to interpret the predictions of our models. All the visualizations were generated with Vivo XAI package in R language. For start, let us focus on the global overview of our algorithms, then we will go on to the local perspective.

4.1 Permutational feature importance

The very first thins that comes to mind, analyzing the model, is inspecting, which variables have the greatest impact on model's outcome. Not only should this give us a sense of understanding our model, but also by seeing it we can detect some anomalies, that will just look strange for humans. Below, there are two plots, for xgboost1 and xgboost2 respectively, showing features' importance to the model's prediction.

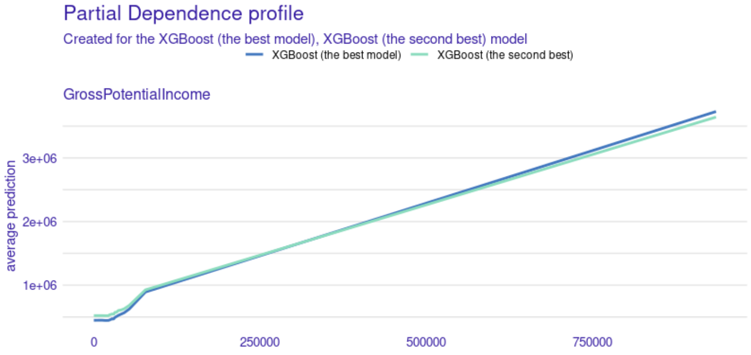


We can see, that both of the inspected models have quite similar order of the most important features. There are two variables that stand out from the rest: *GrossPotentialIncome*, which is maximum income the property can produce and *NumberOfBathroomsWithBathHub* which is pretty self-explainable. Other significant features are also *WalkScore* - metric describing how well-located the property is (and not car-dependent to go somewhere) as well as *Construction* (year of construction) and *NumberOfRooms*. From what we see, we can deduce, that these features could in fact be substantial to the predictions. It makes sense to take number of bathrooms, location or potential income the property can produce when we estimate its value. Now that we learned about significant features for our models, let us find out, how these variables affect the average prediction. In order to do so, we will use two approaches - Partial Dependence Profile (PDP) and Accumulated Local Effects (ALE).

4.2 Partial Dependence Profile

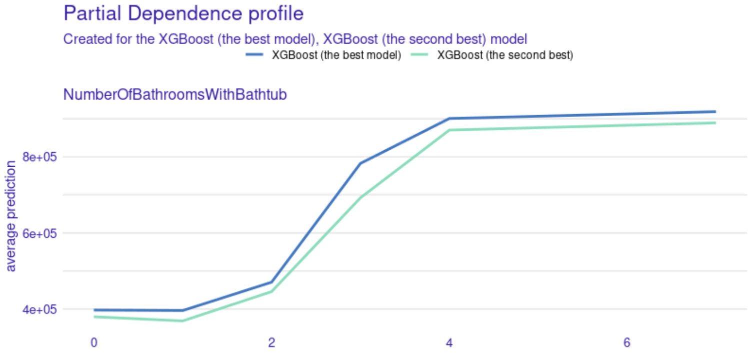
Partial Dependence Plots are used to depict the functional relationship between a small number of input variables and predictions. They show how the predictions partially depend on values of the input variables of interest. Let's see the impact of three variables: *GrossPotentialIncome*, *NumberOfBathroomsWithBathHub* and *Construction*.

4.2.1 Gross Potential Income



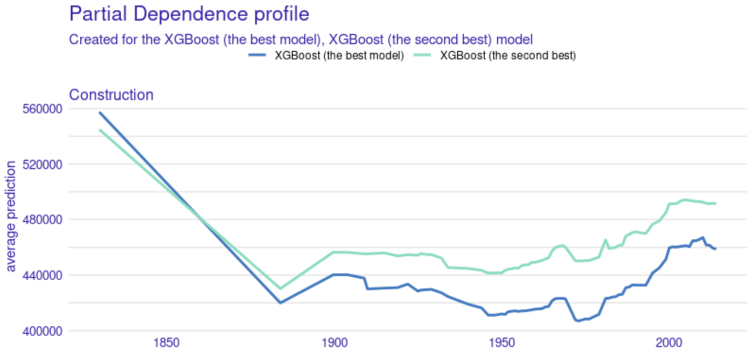
The result should not worry us and it is not surprising - properties that more profitable, according to GPI index, are better priced according to the models. There is a clear, linear correlation between average prediction of price, and the GPI.

4.2.2 Number of Bathrooms With Bathtub



The plot shows us, that in fact the more bathrooms property has, the more expensive it is. In our opinion, not only because it is purely more comfortable to have more than one bathroom, but also it indicates, that the property is bigger - you can not imagine a bungalow with an area of $100m^2$ having 4 bathrooms in it.

4.2.3 Year of Construction

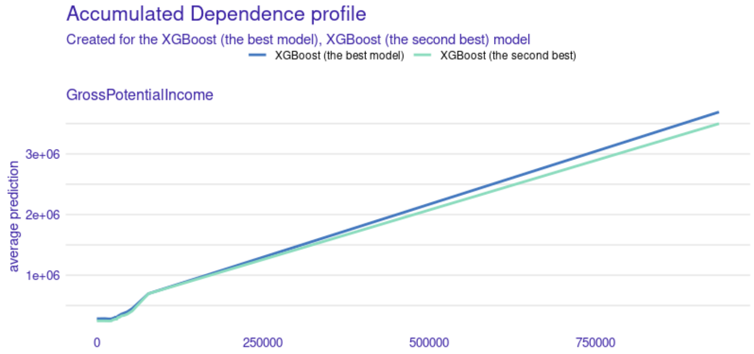


We can observe a steady fall in preditced value of property, when the year of construction of a real estate was later up until 1880. It can be caused by small amount of data in these years, so the outcome is not that accurate. From 1880 onwards, the price is fluctuating, but keeps growing, which should be a good sign as mainly newer properties are getting sold quicker and there should be a sign of inflation.

4.3 Accumulated Local Effects

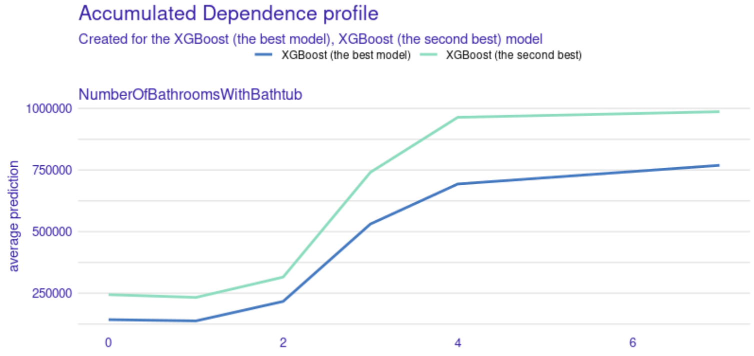
ALE plots describe how features influence the prediction of a machine learning model on average. They are a more reliable and unbiased alternative to partial dependence plots (PDPs).Again, we will analyze the influcene of three variables: *GrossPotentialIncome*, *NumberOfBathroomsWithBathHub* and *Construction* on the models' average prediction .

4.3.1 Gross Potential Income



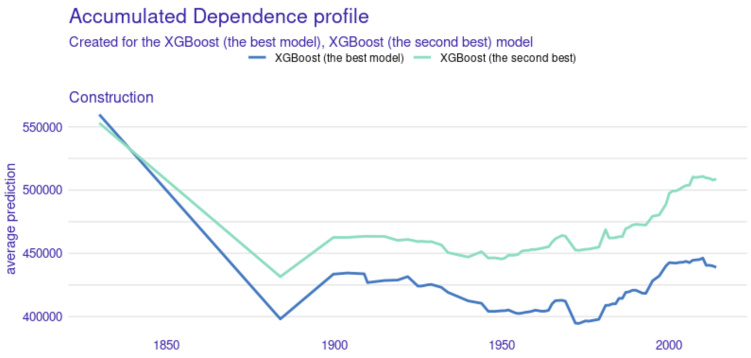
The result and insights are similar to the ones from PDP plot - properties that more profitable, according to GPI index, are better priced according to the models. There is a clear, linear correlation between average prediction of price, and the GPI.

4.3.2 Number of Bathroms With BathTub



The plot shows us, that in fact the more bathrooms property has, the more expensive it is - just as in the PDP plot. However, the difference of price between real estates with fewer bathrooms and properties with greater number of them is significantly greater - from under 250.000$ with one bathroom we jump to 750.000$ so the difference is bigger than in the PD plot.

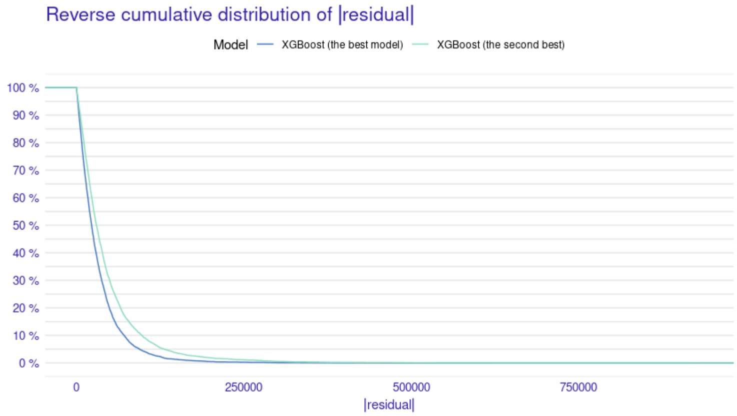
4.3.3 Year of Construction



This one is interesting. The second best model (the lime one on the plot) shows similar behaviour to the PD plot. But, the best xgboost model, predicted that on average, the price of real estate decreases with the increase of construction year, in the period from 1900 until 1980.

4.4 Models' performance

The last part of our models' global analysis is their performance. Let's take a look at the distribution of the residuals, that is the absolute value of real price of property minus predicted value.



We can see that generally the model fits well to the data. There are less than 5% of records with an error greater than 100.000$ for both models. For the record, the average price of our models lay between 450.000$ and 600.000$.

We have analyzed our models globally. Now it is time to see, what's going on in the smaller scale. For local inspection, we will use Break Down and Ceteris Paribus plots.

4.5 Break Down profiles

Break Down plots show how the contributions attributed to individual explanatory variables change the mean value of prediction. We will use two different observations from our data set, and we will explore the behaviour of our two models in these two cases. Let's start with first xgboost model and see how it performs on our two observations.

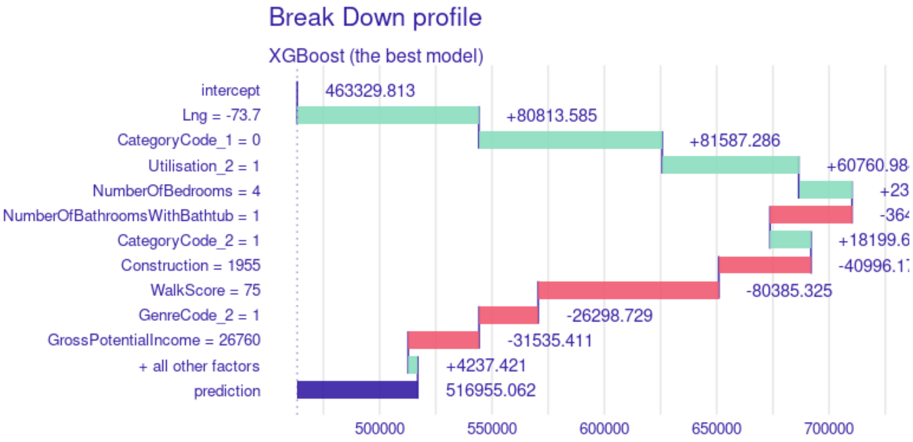


Figure 6: First observation's profile

The most important features for this observation are *Longitude* coordinates and *CategoryCode1* which corresponds to the category of the property - here it is a flag of small condominium which should lower the price.

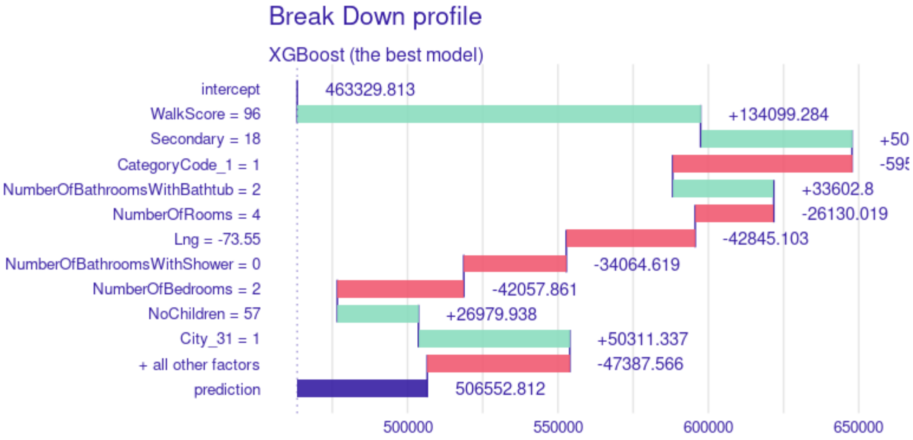


Figure 7: Second observation's profile

In this example, the most significant variables are *WalkScore* and *Secondary* - the metric describing the presence of secondary schools in near location.

We can see that in the first case, *CategoryCode1* increases the prediction while it decreases the outcome in the second observation. Also, we can see that *WalkScore* metric influence both results differently. If its slight lower - 75 - it decreases the prediction drastically. On the other hand, it hugely elevates the prediction in the other example when it is considerably higher - 96 out of 100.

Now, let's move on to the second model analysis.

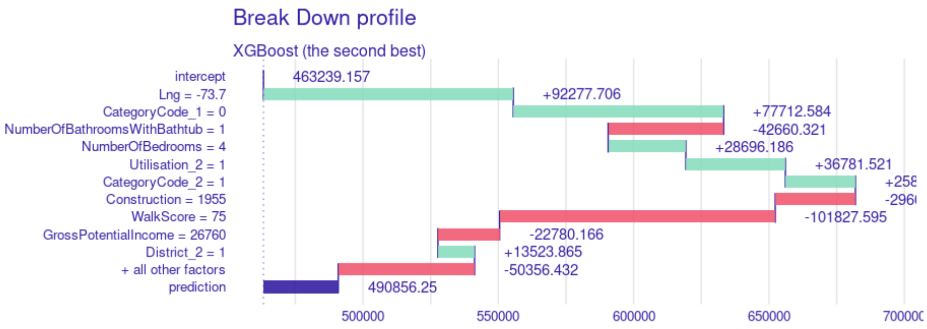


Figure 8: First observation's profile

The most important features for this observation are *Longitude* coordinates and *WalkScore*.

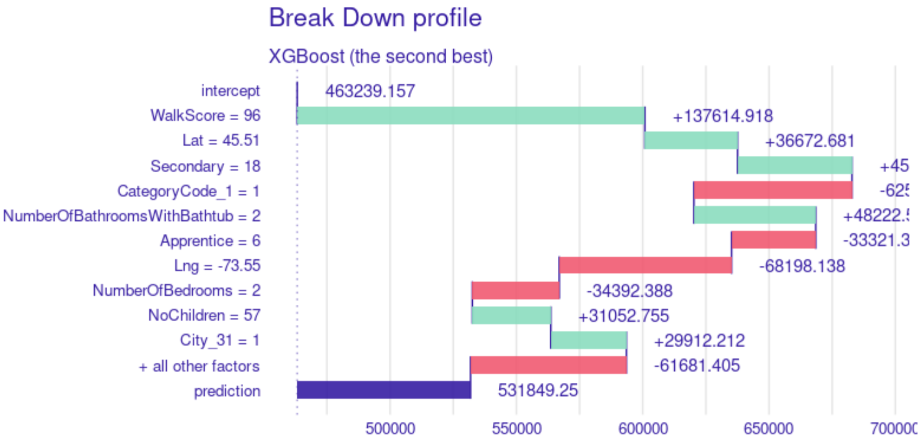


Figure 9: Second observation's profile

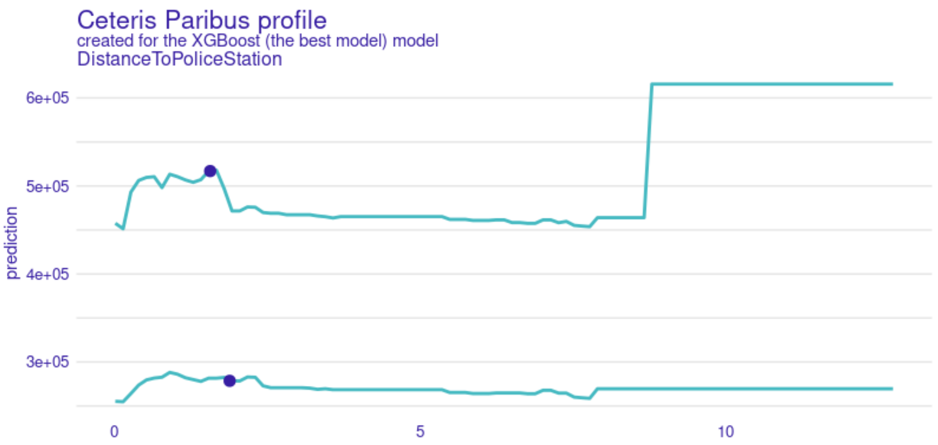
In this example, the most significant variables are *WalkScore* and *Longitute*.

We can see that in the second case, small change in *Longitute* can shift the prediction considerably In the first observation, the longitude of -73.7 increases the prediction while the -73.55 value in second observation - shrinks the value of prediction noticeably. Also, we can see that *WalkScore* metric influence both results differently. If its slight lower - 75 - it decreases the prediction drastically. On the other hand, it hugely elevates the prediction in the other example when it is considerably higher - 96 out of 100 - just as in the first example.

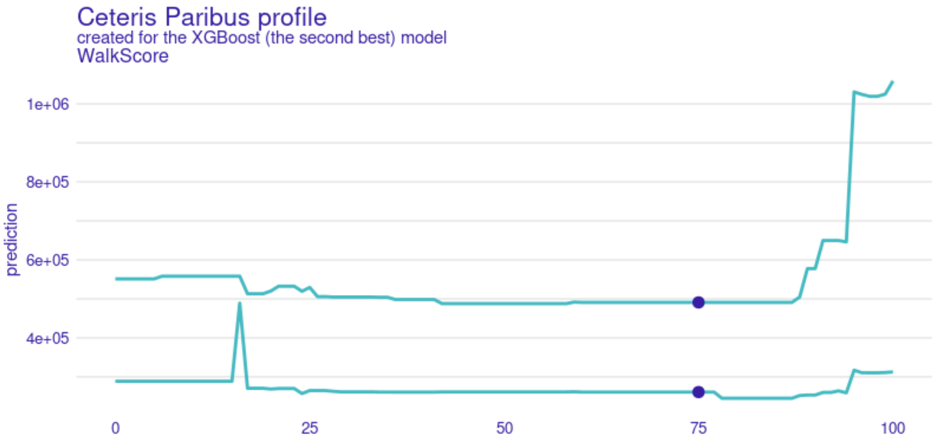
4.6 Ceteris Paribus profile

Ceteris Paribus decomposition profiles are designed to show model response around a single point in the feature space. Essentially, we take the observation and analyze how the prediction is changing provided that other features' values in this observation stay untouched.

The first figure shows the analysis of the first model based on aforementioned two observations. In this part, we will examine the influence of *DistanceToPoliceStation* variable on the model's outcome.



The second CP plot concerns the behaviour of the second model based on two observations. Here, we will inspect the influence of *WalkScore* feature on the model's outcome.



We can see that the first observation behaves differently from the second one in both models. It reacts more abruptly with the increase of variables' value. The second one does not show any significant reaction.

References