High Performance Computing I

WS 2017/2018

Decision Tree Model for the Graz Housing Market

Sabrina-Sigrid Spiegel

Inhaltsverzeichnis

Introduction

In this paper a decision tree Machine Learning (ML) model using a database on housing transactions in Graz is built. The goal is to get a better insight into which variables – or variable combinations – are most important in determining transaction prices of apartments. The outcome of this project will provide a good comparison to more traditional hedonic housing models and will be the first step to the application of more complex ML techniques with this data set.

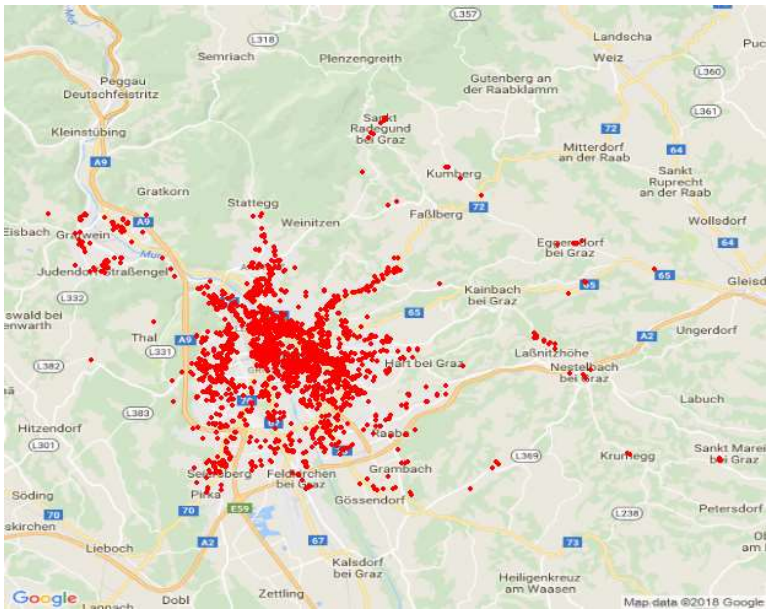
There is a variety of ML techniques to choose from. An advantage of tree-based ML models is that they are easy to interpret and easy to implement. They can handle many types of predictors. Also, these models do not require the user to specify the form of the predictors´ relationship to the response like a linear regression model requires. Their outputs are easy to understand and visually appealing. However, models based on single trees have two well-known weaknesses: model instability and less-than-optimal predictive performance.

Ensemble methods – like Random Forests - that combine many trees into one model combat both of these problems. In Random Forests several trees are generated on the different bootstrapped samples from training data and decorrelated. The variance is reduced by averaging the trees. Building a lot of trees makes the correlation between the trees smaller. At a split on the training data, only a random sample of predictors is considered. This improves the predictive performance (Kuhn and Johnson, 2016).

Dataset and pre-computation activities

The dataset is provided by the ZTdatenforum which collects all transactions that are entered into the Austrian land registry and describes each transaction with up to 34 characteristics. For this research paper I focus on the transactions of apartments in Graz and some of the surrounding area. The data set is based on actually transacted properties and consists of over 6000 observations for the 2014 to2017 period. Figure 1 gives an overview of the data and how the transactions are located in Graz and the surrounding area.

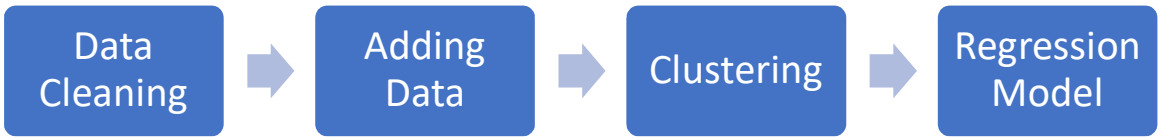
Figure 1: Apartment transactions Graz 2014-2017



The first necessary step is cleaning the dataset (for more on data cleaning see next section). Luckily, decision tree models are quite insensitive to missing data or skewness issues and can handle many different types of predictors without the need to pre-process them. However, decision tree models can react negatively to autocorrelation between predictors. I will consider this in the selection of predictor variables. Also, in contrast to other ML techniques, decision tree models work best with a limited number of strong predictor variables rather than a large number of input variables. A further reason for a careful selection (and transformation) of predictor variables is that decision tree models have an inbuilt preference for continuous variables and will tend to choose them over more granular predictors (Loh and Shih, 1997, Strobl et al. 2007, Loh 2010, Kuhn and Johnson 2016). Transforming continuous variables into level parameters should resolve this issue.

Code structure

The code for this model was written in R and consists of four main parts. In the first part the dataset is cleaned, in part two and three the dataset is extended with new variables by different methods. The dataset that is built by running the first three parts of code, is the database of the essential part. This is the dataset that is used as input for the ML models.



Data Cleaning

After reading the dataset into R, several variables have to be modified that they can be used in R for the analysis. As some integer variables are changed by the code into factor variables, other variables need to be determined as numeric or as character variables. Some numeric variables like floor space are divided into groups to factorize them.

```
data1a$Kat <-  cut(data1a$NutzFl, breaks = c(20, 35, 50, 70, 90, 110, 140, 170))
```

Next, reasonable thresholds are set to clean the dataset from outliers. One variable for example informs whether a transaction took place under special circumstances (bankruptcy, estate or a transaction between related people). These special transactions are excluded because their prices do generally not reflect the market price. A cut-off for longitude and latitude has to be determined to exclude sales outside the Graz area. Apartments smaller than 20m^2 are also excluded. With respect to price outliers the following steps were taken: Apartments costing less than €800 per m^2, or used apartments costing more than €6000 per m^2 were removed from the data-set. The effect of this outlier rule is similar to imposing the robust-3σ-rule to the price of the apartment per square meter of living space.

```
data1a <- dplyr::filter(data1a, trimws(Verwandtschaft) == "FALSCH")

data1a <- dplyr::filter(data1a, trimws(Konkurs) == "FALSCH")

data1a <- dplyr::filter(data1a,(preissqm > 800))

data1a <- dplyr::filter(data1a,(preissqm < 6000))

data1a <- dplyr::filter(data1a, trimws(longitude) > 15.3)

data1a <- dplyr::filter(data1a, trimws(latitude) > 47.0)
```

The code also tries to handle missing values by several steps. Where possible, missing values are filled in with information of other variables. E.g. a missing postal code can fill up the name of the district or vice versa.

```
subset1$Postleitzahl[subset1$GBName =='Lend'] <- '8020'

subset1$Postleitzahl[subset1$GBName =='Algersdorf'] <- '8020'

subset1$Postleitzahl[subset1$GBName =='Andritz'] <- '8045'

subset1$Postleitzahl[subset1$GBName =='Engelsdorf'] <- '8041'
```

These data cleaning steps leave us with a large number of variables and information. Of these a limited set of characteristics – those that seem to have the best predictive power – are chosen for inclusion in the ML models:

- Geographical location: name of district, postal code, cadastral community, longitude and latitude (missing values are excluded)
- Year of transaction: 2014, 2015, 2016 and 2017 (missing values are excluded)
- interior size: Excluding areas below 20m² and above 200m², as well as those with missing values, creating 8 to 11 levels for this size parameter.

- Age of building or age of parification:
    - Unknown age
    - Before 1950 and no new parification
    - Before 1950 with new parification
    - 1950 to 1979
    - 1050 to 1979 with new parification
    - 1980 to 1999
    - 2000 to 2017
- Equipment of apartment: terrace, balcony, garden, cellar
- Category of zoning area as well as maximum allowed building density
- Total costs
- Price per square meters
- Seller is property developer or not

The adjusted and cleaned data set then is exported into a CVS file which is the data base for the next part of the code.

Adding New Data Sources

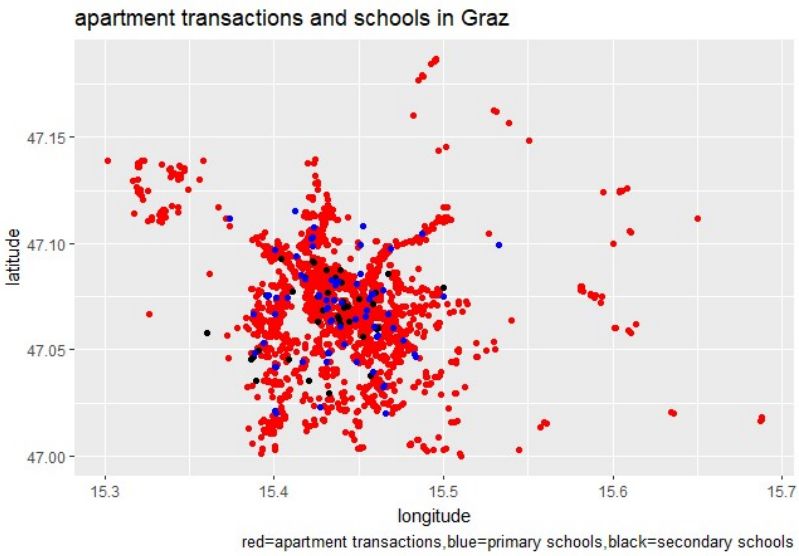
The data base (the cleaned transaction database which is the outcome of the first part of the code) is then expanded by new data sources to allow for more detailed research results. the cleaned transaction database is now extended by additional characteristics. In particular, publicly available information on location characteristics are included as these most likely influence the quality of the property and thus the sales price:

- Longitude and latitude of different types of schools (Volksschule, Neue Mittelschule, Gymnasium etc.) are included in the data set. Then the distances between each of the transacted apartments to each of these schools is calculated and recorded in the dataset.

- Similarly, information about the location of kindergartens and nurseries and their distance to the transacted apartment are included.
- Park and Ride positions and the distance to the transacted apartment are included.
- Pharmacies' longitudes and latitudes as well as their distances to the transacted apartment are included as well.

To link the new information with the existing data set, the location is introduced into the code through longitude and latitude. The type of the school is specified, so that primary schools and higher-tier schools can be investigated separately. Figure 2 illustrates the effect of linking these two data sets: the location of transacted apartments (red) and schools (blue: primary schools and black: higher schools).

Figure 2: Apartment transactions and school location



In order to calculate the distance of the transacted apartments to these type of schools, first subsets of data-frames with just longitude and latitude coordinates are built to be able to create a distance function between these two data sets.

```
Schooldistance2 <- distm(data1b_coord,data_Schulen_coord, fun=distHaversine )

Schooldistance2[1,1]

Schooldistance2[1,2]

min(Schooldistance2[1,])
```

```
data1b$nearestSchool<- data_Schulen$NAME[max.col(-Schooldistance2)]

X <- Schooldistance2

X <- as_data_frame(X)

colnames(X) <- data_Schulen$NAME

rownames(X)

resultA <- (sapply(seq(nrow(X)), function(i) { j <- which.min(X[i,]) }))

resultB <- (sapply(seq(nrow(X)), function(i) { j <- min(X[i,]) }))

data1b$nearestSchool_meter <- resultB

data1b$nearest_KAT3 <- data_Schulen$KAT3[max.col(-Schooldistance2)]

head(data1b$nearest_KAT3)

Schooldistance2_NMS <-

distm(data1b_coord,data_Schulen_NMS_coord,fun=distHaversine )

head(Schooldistance2_NMS)

Schooldistance2_NMS[1,1]
```

As a result, the nearest primary school and secondary school and its distances to the transacted apartment can be created as new variables in the data set.

```
data1b$nearestNMS<- data_Schulen_NMS$NAME[max.col(-Schooldistance2_NMS)]
```

For pharmacies, kindergartens and nurseries the procedure is the same. Distances to the next pharmacy, the next kindergarten, the next nursery and the next park and ride position from the transacted apartment are new variables in the data set.

Another price influencing location parameter can be created by calculating the distance from the transacted apartment to the city center. The data set is extended also with this variable.

```
Hauptplatz_coord <- c(15.438391, 47.070794)

Hauptplatz_coord

HauptplatzDist <- distm(data1c_coord,Hauptplatz_coord,fun=distHaversine )

X <- HauptplatzDist
```

```
X <- as_data_frame(X)

rownames(X)

resultC <- (sapply(seq(nrow(X)), function(i) {

 j <- min(X[i,]) }))

data1c$HauptplatzDist <- result
```

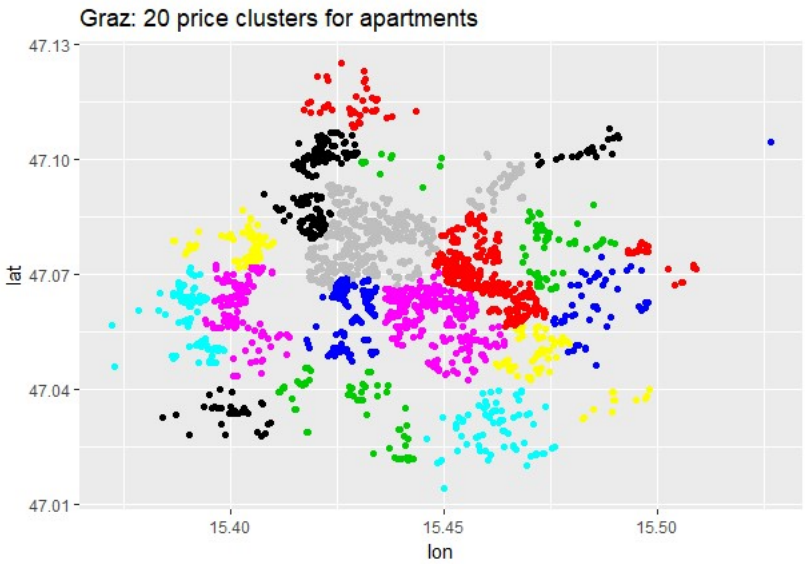
The improved data set builds the data basis of the next part of the code.

Cluster Analysis

After optimizing and extending the original dataset a cluster analysis is performed. This will group the transactions into clusters such that the transactions in one cluster are more similar than in another cluster. The cluster analysis extends the data set with another location parameter so that the area of Graz is not only divided into districts but also in clusters. First 20 clusters are calculated where only longitude and latitude matter. Every transaction then will belong to one of these clusters.

```
d= as.matrix(dist(cbind(data_new$lon_1,data_new$lat_1)))

d=ifelse(d<5,d,0) d=as.dist(d)

hc <- hclust(dist(data_new))

clust <- cutree(hc, 20)

plot(hc)

graz_clust <- cutree(hc,k=20)
```

figure 3: Clustering Graz into 20 clusters (based on location)



The mean floor space for every cluster is calculated and also saved as new variable.

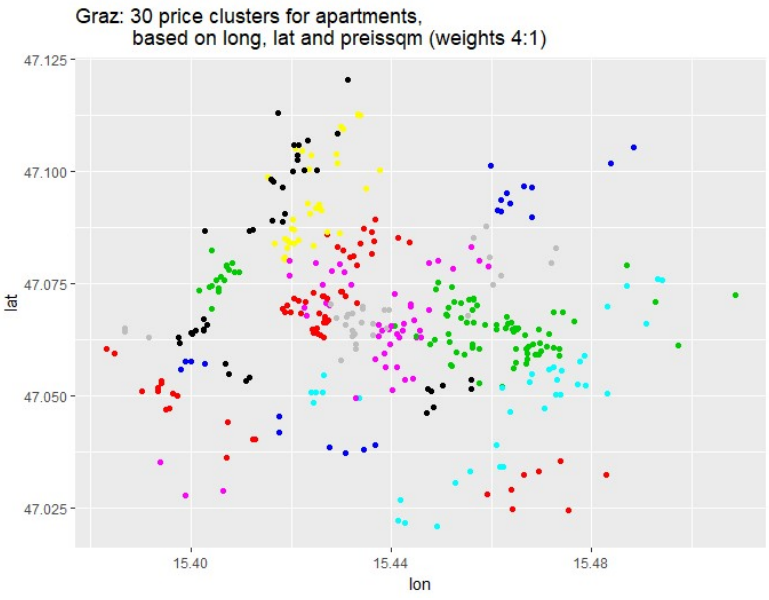
```
data1c <- data1c %>%

 group_by(graz_clust) %>%

 mutate(NutzFl_mean = mean(NutzFl, na.rm=TRUE))
```

Because in the first cluster analysis only the location matters, another cluster analysis should assess the price per square meter together with the location.

```
daisy1 <- daisy(data_sample, metric = c("gower"),
        stand = FALSE, type = list(), weights=c(4,4,1))
…
d= as.matrix(dist(daisy1))
d=ifelse(d< 0.001,d,0) d=as.dist(d)
hc <- hclust(dist(daisy1)) # hierarchical clustering
…
graz_clust_30 <- cutree(hc,k=30)
```

Since the goal of the clustering analysis is to come up with locational clusters, longitude and latitude of the transactions are still given a higher weight than the price information (relative weights are 4:1). Figure 4 illustrates the result of this step.

Figure 4: Clustering Graz into 30 clusters (based on location and price)



As output of this analysis the cluster the transacted apartment belongs to is saved as a new variable and added to the data base.

Regression Model

In the last part of the code a random forest model is applied. In a first round the model is applied in order to decide which variables are most important in determining transaction prices of apartments in Graz.

When computing a random forest model in R it is important to avoid variables with more than 53 categories. These variables have to be deselected or their factor number reduced. Variables with too many missing values make the entire model instable and are therefore eliminated (This applies to the variables “maximum building density” and “Widmung”). Colinear variables are reduced to increase the significance of the data set. Also, apartments that are sold by property developers are cut from the data set to have a more homogenous market.

Then the resulting data set and its variables are checked for skewness. An un-skewed distribution is roughly symmetric. A distribution with a large number of points on the left side of the distribution than on the right side is right-skewed. This means there is a greater concentration of data points at relatively small values and a small number of large values (Kuhn and Johnson, 2016) .

All numeric variables of the data set are checked for skewness:

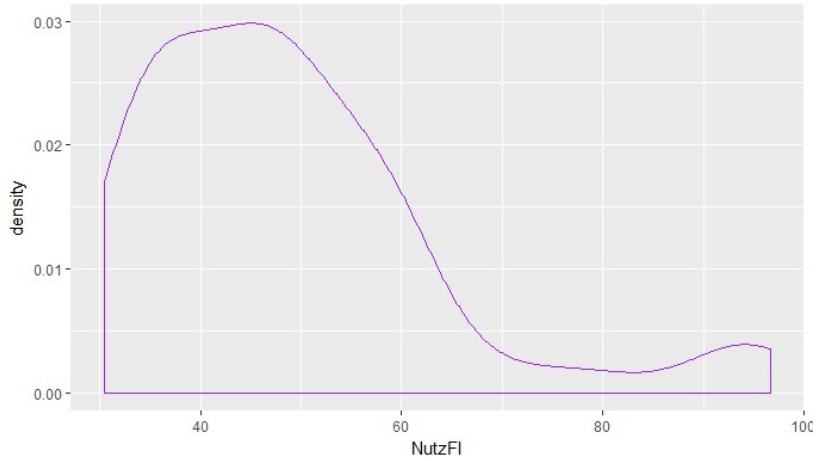
```
skew <- sapply(numeric_columns,function(x){skewness(data4[[x]],na.rm = T)})
```

This leads to following result:

```
                PLZ                GstGroesse                    NutzFl 
         -0.3189388                  1.9376709                  1.4463829 
          LNGesamtFl                   BauFlGeb                GesamtPreis 
                 NaN                  2.2435940                  0.1132175 
       InventarPreis                  PkwApPreis              SonstigesPreis 
           5.7996658                  0.9745437                        NaN 
         Steuersatz       ParifizierungsJahr                    KellerFL 
                 NaN                 -4.6853011                  0.5984918 
              PKWFl                  TerasseFL                    BalkonFl 
          1.6400022                  2.0329194                  2.0611306 
           GartenFl                 Gerichtsnr                   latitude 
          2.3761587                  1.3276495                 -0.6576707 
          longitude                    preissqm                 logpreissqm 
         -1.1520965                  0.2674659                 -0.2319192 
          preissqm2             loggesamtpreis            nearestNMS_meter 
          0.7740510                 -0.4397061                  0.1989057 
   nearestVS_meter    nearestApotheke_meter                HauptplatzDist 
          0.2275280                  1.2512482                 -0.7072884 
     Personen_total              Prakt_to_Fach                 Aerzte_total 
          0.9993442                  1.1522077                  4.6350577 
      Aerzte_ratio       Kindergarten_meter          Kinderkrippen_meter 
          0.3230532                  0.4635999                  0.9652354 
  ParkandRide_meter              NutzFl_mean 
         -0.4259278                        NaN 
```

Looking at the skewness parameter of the variable NutzFl, which is floor space, one can see that many more apartments with small floor space are sold apartments sold than apartments with large floor space. The distribution is right skewed (skewness greater than 0).

Figure 5: Floor space of sold apartments

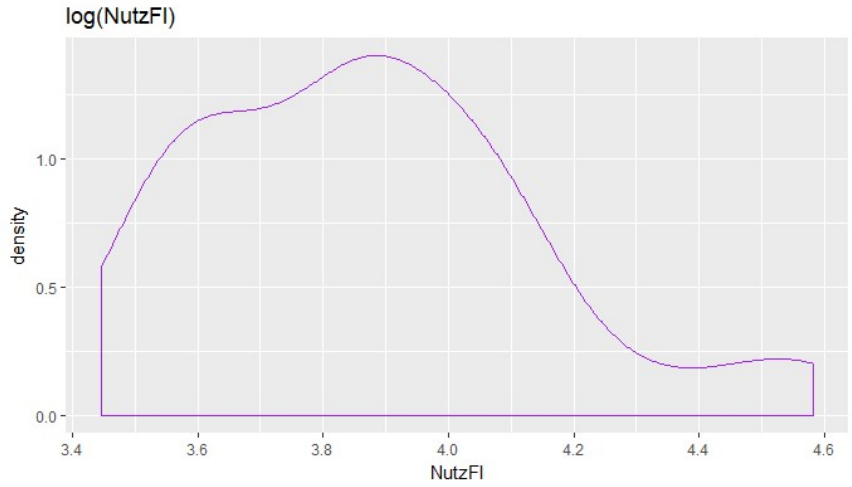


Replacing the data with the log, square root, or inverse can remove the skew (Kuhn and Johnson, 2016). A threshold for skewness is determined and all variables above this threshold are transformed with log(x+1).

```
skew <- skew[skew > 0.75]

for(x in names(skew))

{data4[[x]] <- log(data4[[x]] + 1)}
```

Looking at the variable NutzFl, the floor space, replacing it with the log has following effect:

Figure 6: Loged floor space of sold apartments



Random Forest Model

After these data pre-processing steps, a random forest model is applied. As the output variable the price of the transacted apartment is used – however as mentioned above, it is used in its log form to resolve the skewness. The distribution of this variable is then not entirely symmetric but it´s better behaved than in natural units. For the predictor variables different combinations of the data set variables are used to check which variable combinations lead to the highest explained variance. Below is the list of variables that lead to the highest explanatory power (psydo R^2):

```
data5 <- select(data4, c("GesamtPreis", "NutzFl", "longitude", "latitude","Parkplatz", "Keller", "Balkon",
                         "Garten","Postleitzahl","AlterKategorie","logpreissqm", "jahrdatum", "landverkaeufer",
                         "landkaeufer", "statusverkaeufer", "statuskaufer"))
```

The data is separated into test set and training set and then the Random Forest model is applied to the training data set:

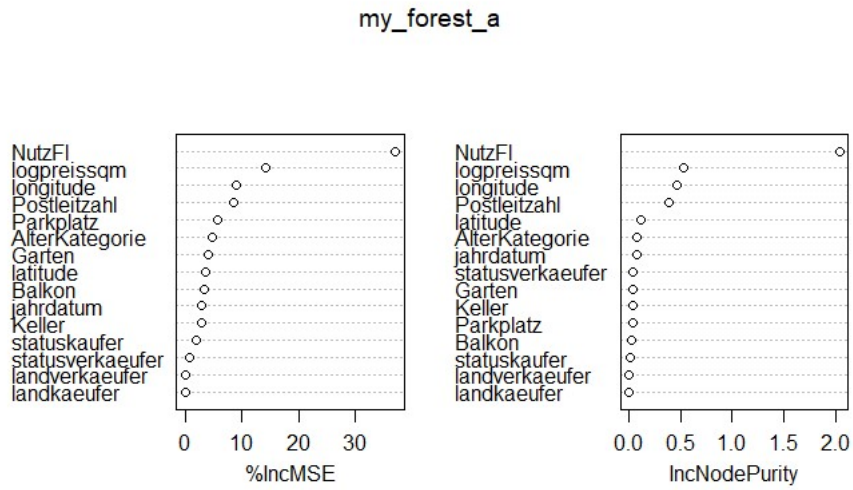
```
set.seed(1234)
id <- sample(2, nrow(data5), prob=c(0.9,0.1), replace=TRUE)
data_train_a <- data5[id==1,]
data_test_a <- data5[id==2,]
…
my_forest_a <- randomForest(log(GesamtPreis)~., data = data_train_a,
                            ntree = 501,
                            na.action=na.omit, mtry = 8,
                            importance = TRUE, proximity = TRUE)
```

The two main arguments in the formula are mtry – for the numbers of predictors that are randomly sampled as candidates for each split – and ntrees – the number of bootstrap samples. Running the code leads to following outcome:

```
Type of random forest: regression
Number of trees: 501
No. of variables tried at each split: 8
Mean of squared residuals: 0.01613551
% Var explained: 79.11
```

The % Var explained variance measures how well out-of-bag predictions explain the target variance of the training set. In this case we get 79.11 % variance explained with 501 created trees.

Using the formula importance in our random forest model indicates the increase of the Mean Squared Error when the given variable is randomly permuted. This analysis shows that the most important variable influencing the price of the apartment are the floor space, the price per square meter, the longitude and the postal code. These variables increase the mean squared error by more than 10 %. Information about the buyer and the seller of the apartment have hardly influence.



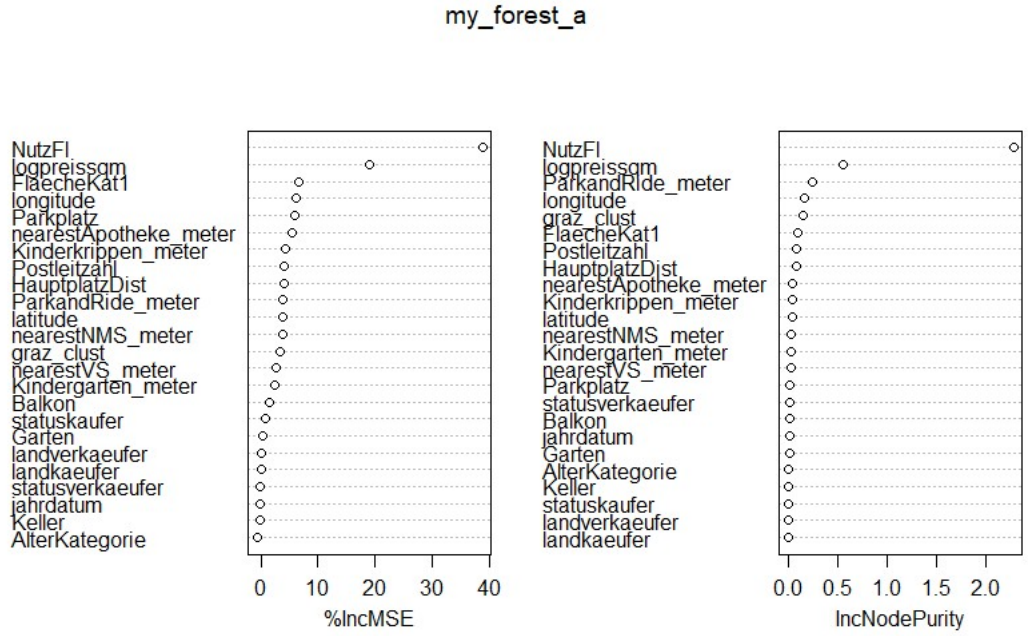
These results are the base model. Next, the model is applied to the extended data set (see code B and C) to check whether more information influences the explained variance and how important these new variables are for the price determination. The newly included variables are: nearest primary school, nearest secondary school, nearest pharmacy, distance to city center, nearest kindergarten, nearest nursery and nearest park&ride station. No old variables have been dropped at this point.

Adding these variables increases the % Var increased variable up to 81.03 percent.

```
Type of random forest: regression
Number of trees: 501
No. of variables tried at each split: 24
Mean of squared residuals: 0.01465113
% Var explained: 81.03
```

Looking at the importance of the new variables shows that these new variables all have influence on the price determination. Environmental features like the distance to schools and

childcare facilities have an influence between 2 and 5 %, the distance to the next pharmacy over 5 %.



For comparison to the other results variables with no or hardly influence on the price determination are dropped. The variables that give information about buyer and seller, garden, year of sale, cellar and the category of age are dropped to see whether this increases the explained variance or not.

```
Type of random forest: regression
Number of trees: 501
No. of variables tried at each split: 17

Mean of squared residuals: 0.01350384
% Var explained: 82.52
```

The result shows that excluding unimportant variables for the price determination increases the explained variance.

Conclusion

The random forest model applied here is able to give a good price estimate for apartments in the Graz market. It is able to do so while using predominantly location features of the apartments. Excluding variables with less explanatory power can improve the price estimation. The next step for this model will be the inclusion of data to better model the internal quality of apartments. The goal is to raise the explanatory power of the model.

Reference List

Kuhn M., Johnson K. (2016), Applied Predictive Modeling, Springer, New York.

Loh W.Y., Shih Y.S. (1997), Split Selection Methods for Classification Trees, Statistica Sinica, 7, 815-840.

Loh W.Y. (2010), Tree-Structured Classifiers, Wiley Interdisciplinary Reviews: Computational Statistics, 2, 364-369.

Strobl C., Carolin C., Boulesteix A.-L., Augustin T (2007), Unbiased Split Selection for Classification Trees Based on the Gini Index, Computational Statistics and Data Analysis, 52(1), 483-501