Decision Tree Model for the Graz Housing Market

High Performance Computing I&II
WS 2017/2018 & SS 2018

Sabrina-Sigrid Spiegel

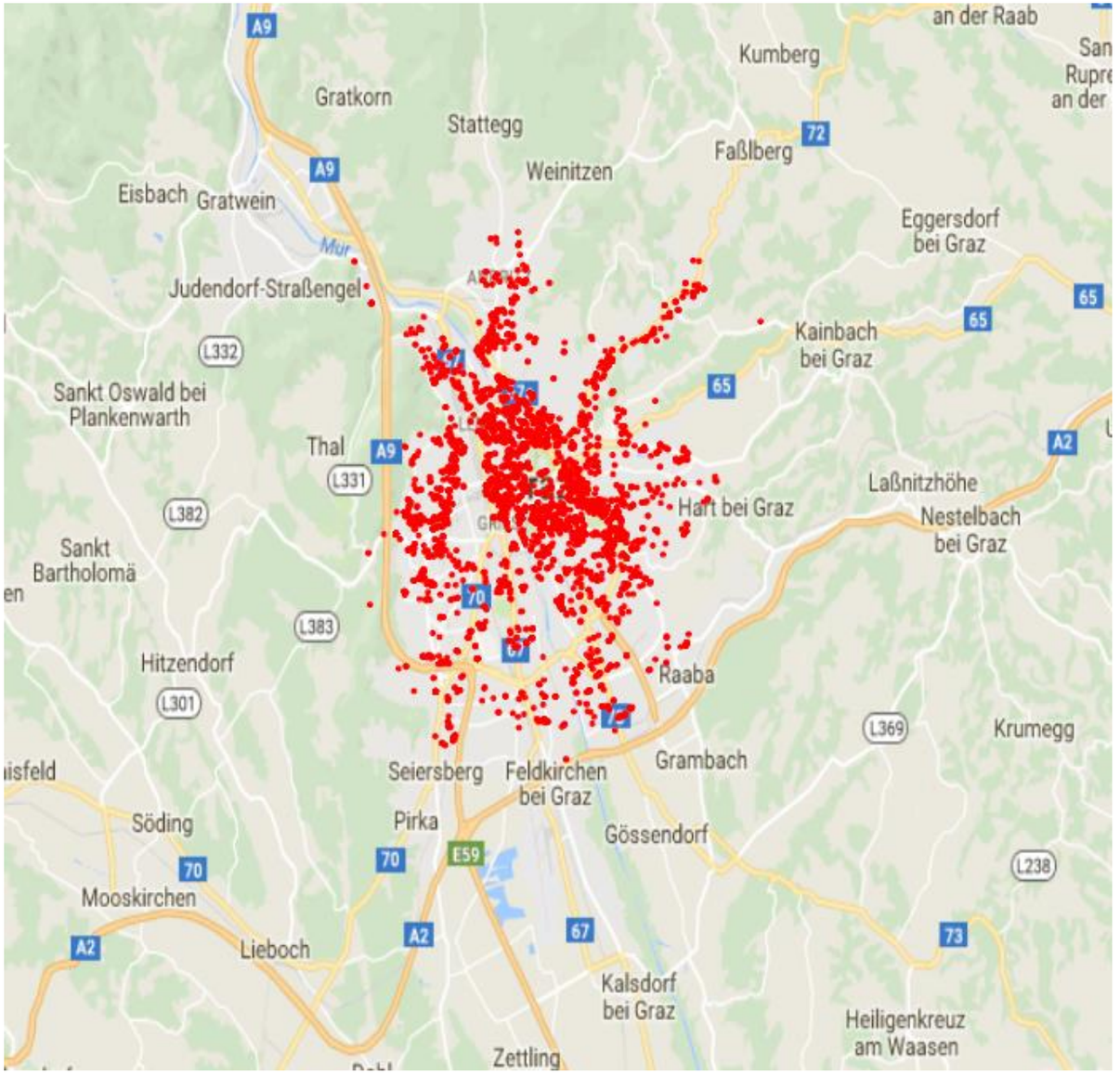
immo netZT

Introduction

- ✓ decision tree Machine Learning model is built
- ✓ database on housing transactions in Graz
- ✓ dataset is provided by the Ztdatenforum: collection of all transactions that are entered into the Austrian land registry
- ✓ 6000 observations of transacted apartments in Graz from 2014-2017
- ✓ goal: better insight which variables or variable combinations are most important in determing transaction prices of apartments

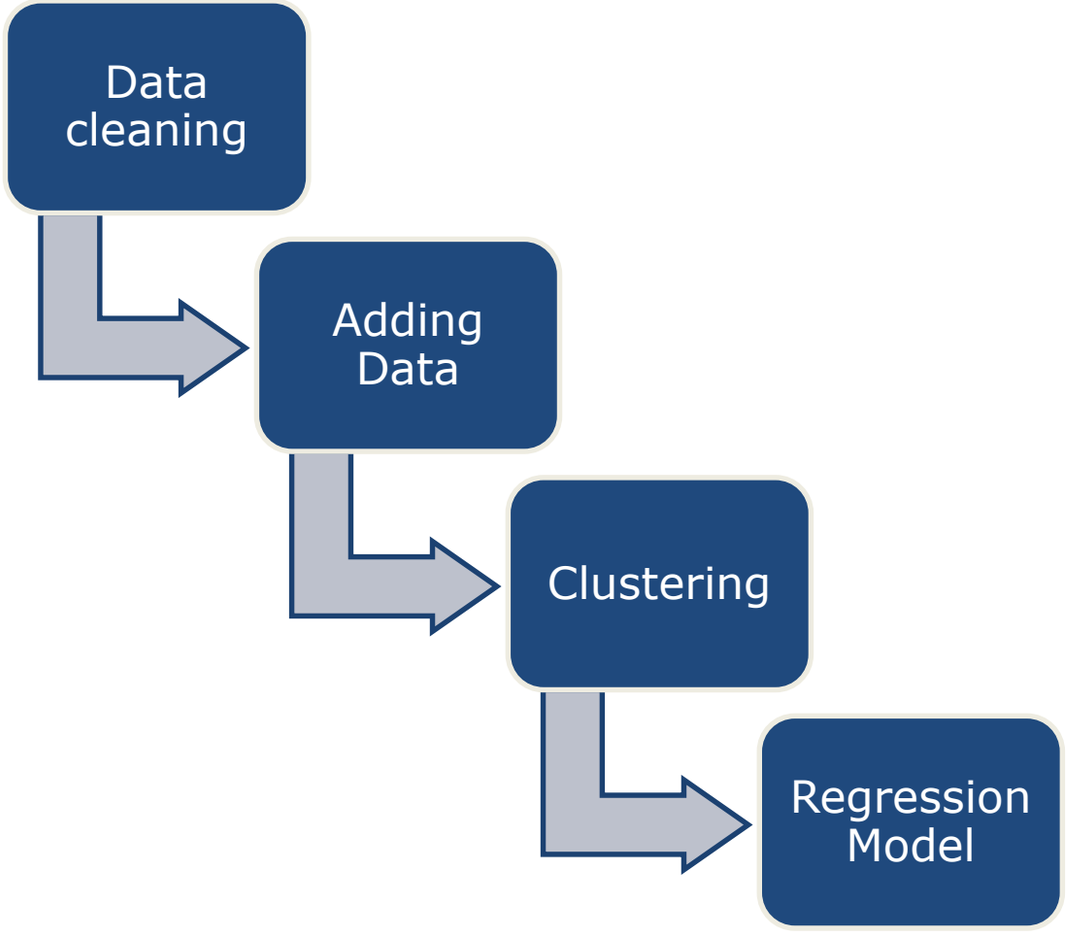
immo netZT

data base


A9
Gratkorn
Stattegg
Kumberg
an der Raab
Faßlberg
72
Weinitzen
Eisbach
Gratwein
Mur
Eggersdorf bei Graz
Judendorf-Straßengel
65
Kainbach bei Graz
L332
Sankt Oswald bei Plankenwarth
Thal
L331
A2
Laßnitzhöhe
Haft bei Graz
Nestelbach bei Graz
L382
Sankt Bartholomä
L383
Hitzendorf
Raaba
L301
L369
Krumegg
Seiersberg
Feldkirchen bei Graz
Grambach
Söding
Pirka
Gössendorf
E59
L238
Mooskirchen
Lieboch
73
Kalsdorf bei Graz
Heiligenkreuz am Waasen
Zettling

code structure

- ✓ code is written in R and consists of 4 main parts



immo netZT

Data Cleaning (I)

- ✓ modifying variables: integer variables are changed by the code into factor variables, others are determined as numeric or character variables
- ✓ dividing numeric variables into groups
- ✓ thresholds are set to detect outliers
- ✓ special transactions (e.g. transaction between related people) are excluded that do not reflect the market price

Data Cleaning (II)

- ✓ Where possible, missing values are filled in with information of other variables
- ✓ Outcome: large number of variables and information
- ✓ Limited set of characteristics (with best predictive power) is chosen:
  - ***Geographical location**: name of district, postal code, cadastral community, longitude and latitude*
  - ***Year of transaction**: 2014, 2015, 2016 and 2017*
  - ***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 Widmung and maximum building density***
  - ***Total costs***
  - ***Price per square meters***
  - ***Seller is property developer or not***

- ✓ adjusted and cleaned dataset is then exported into a CVS file as data base for the next part of the code

immo netZT

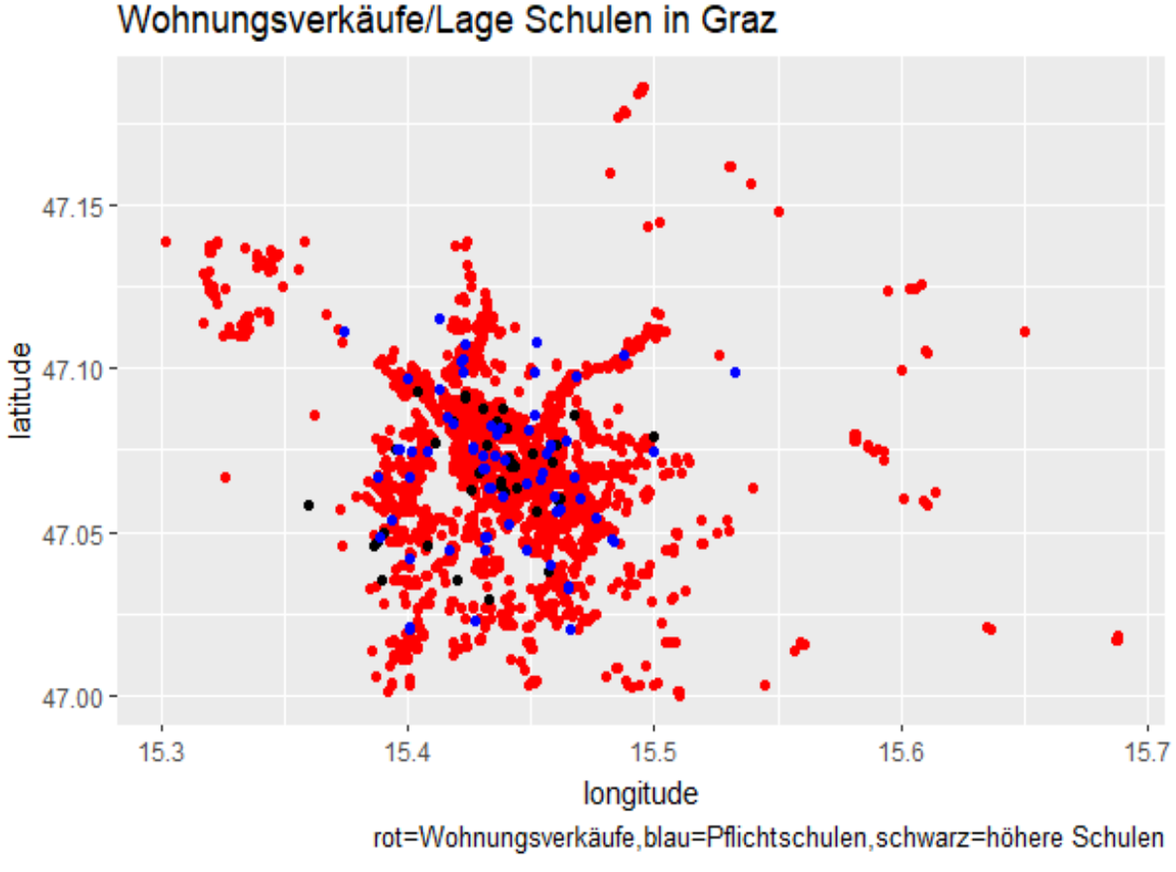
Adding New Data Sources (I)

- ✓ data base is expanded by publicly available information on location characteristics
- ✓ E.g.: different types of schools in Graz
- ✓ new information is linked with existing data set by introducing longitude and latitude of these schools into the code
- ✓ distances of the transacted apartments to these schools are calculated
- ✓ nearest primary school and secondary school and its distances are created as new variables in the data set

immo netZT

Adding New Data Sources (II)

- ✓ appartment transactions and schools in Graz



Adding New Data Sources (III)

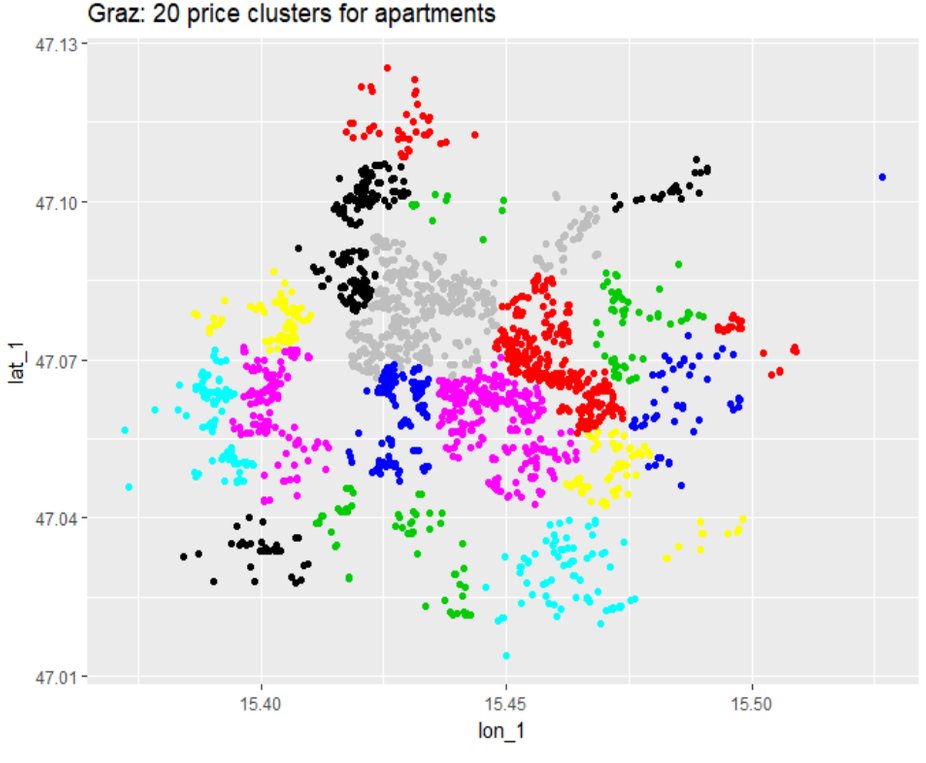
- ✓ Similarly, information about the location of other institutions and the distance to the transacted apartment are included:
  - Kindergarten
  - Nurseries
  - Pharmacies
  - Park&Ride stations

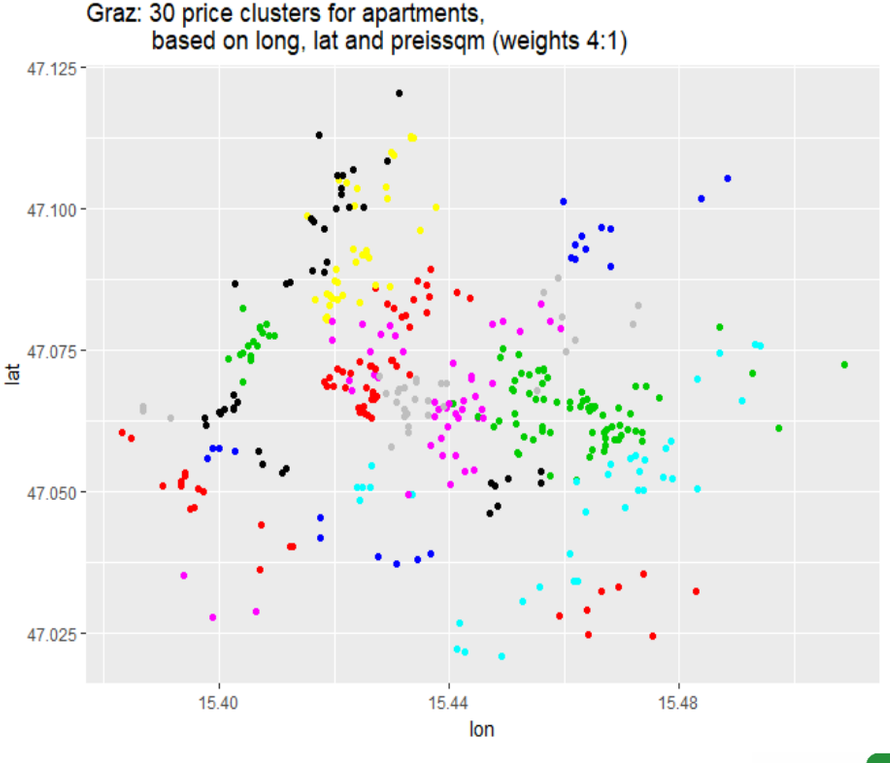
- ✓ The distance to the city centre (Hauptplatz) ist calculated for each apartment and saved as new variable

immo netZT

Cluster Analysis

- ✓ Grouping transactions into clusters such that the transactions in one cluster are more similar than in another one
- ✓ Dividing Graz not only into districts, but also in clusters
- ✓ Calculating 30 clusters depedning on location (longitude and latitude) and the price information (relative weights of 4:1)





- ✓ the cluster the transacted apartment belongs to is saved as a new variable

Regression Model

- ✓ Random Forest Model is applied
- ✓ tree based ML models are easy to interpret and easy to implement
- ✓ they can handle many types of predictors
- ✓ their outputs are easy to understand and visually appealing
- ✓ single trees have two weaknesses: model instability and less-than-optimal predictive performance
- ✓ Random Forests combine many trees into one model and can combat both of these problems
- ✓ several trees are generated on the different bootstrapped samples from training data and decorrelated
- ✓ averaging the trees reduces the variance
- ✓ the builiding of many trees makes the correlation between the trees smaller
- ✓ at a split on the training data, only a random sample of predictor is considered
- ✓ this improves the predictive performance

Random Forest Model (I)

- ✓ first round: decide which variables are most important in determing transaction prices of apartments in Graz
- ✓ variables with more than 53 categories have to be avoided
- ✓ variables with too many missing values have to be eliminated (instability of model)
- ✓ colinear variables have to be reduced to increase the significance ot the data set
- ✓ checking the data set for skewness (an un-skewed distribution is roughly symmetric)

Figure 5: Floor space of sold apartments

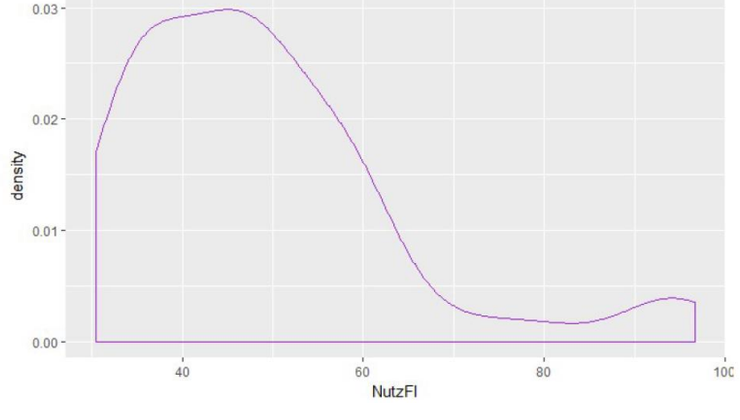
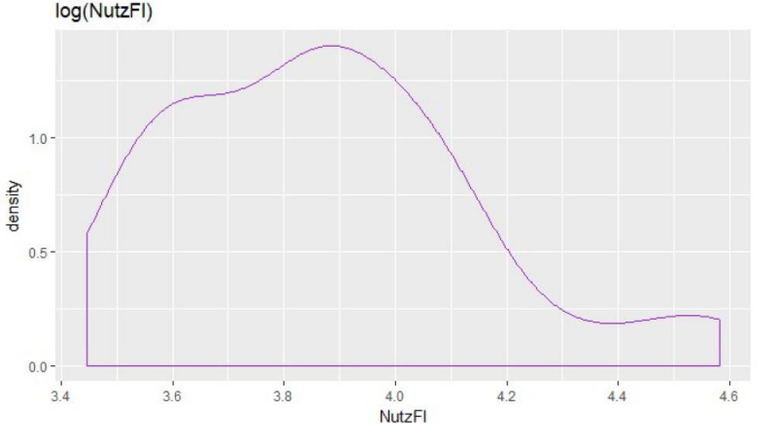


Figure 6: Loged floor space of sold apartments



Random Forest Model (II)

- ✓ output variable: price of transacted variable (used in log form to resolve skewness)
- ✓ predictor variable: different combinations of the data set variables
- ✓ check which variable combinations lead to highest explained variance (measures of how well out-of-bag predictions explain the target variance of the training set)
- ✓ 1) chosing variable combination:

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

- ✓ 2)seperating into test set and training 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)
```

immo netZT

Random Forest Model (III)

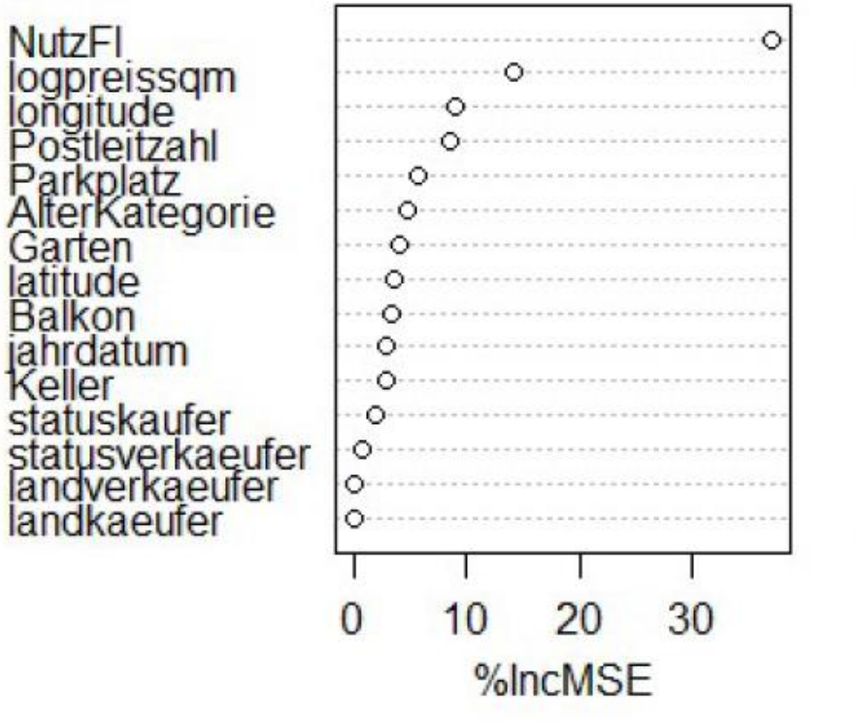
- ✓ mtry: when forming each split a different random set of 8 variables is selected within which the best split point is chosen -> all variables might be used at some point when searching for split points while growing the tree

- ✓ 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

Random Forest Model (IV)

✓setting importance TRUE shows the most important variable influencing the price of the apartment (importance indicates the increase of the Mean Squared Error when the given variable is randomly permuted)



Random Forest Model (V)

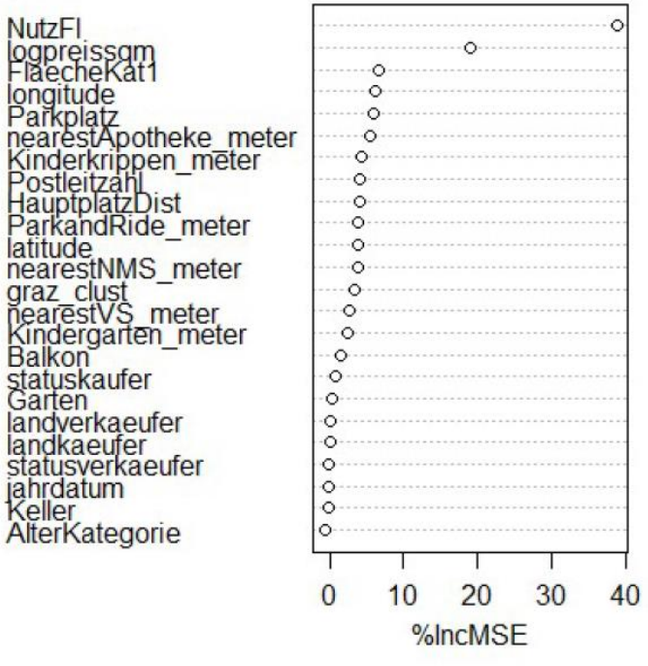
- ✓ model is applied to extended data set: Which influence does more information have on the price determination?
- ✓ new variables: nearest primary school, nearest secondary school, nearest pharmacy, distance to city centre, nearest kindergarten, nearest nursery and nearest park&ride station
- ✓ adding new variables increases %Var 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

immo netZT

Random Forest Model (V)

- ✓ importance of new variables: all variables have influence on the price determination



Random Forest Model (VI)

- ✓ for comparison: unimportant variables are dropped
- ✓ dropping information about buyer and seller, garden, year of sale, cellar and the category of age increaes the explained variance

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

Conclusions

- ✓ random forest model: able to give good price estimate for apartments in the Graz market
- ✓ able to do so while using predominantly location features of the apartments
- ✓ excluding variables with less explanatory power can improve the price estimation
- ✓ next step: inclusion of data to better model the internal quality of apartments

immo netZT