Predicting House Prices

Oliver Tomondy, Friedrich Winkelbauer

28/12/2021

Contents

1 Ziele

- Das Ziel dieser Arbiet is es, einen Datensatz mit Seattle Häuser zu analysieren und mit verschiedenen Machine Learning Modellen den Preis der Häuser in der Stadt Seattle vorherzusagen.
- Anschließend wird das beste Modell auch als Webservice deployed.

2 Libraries

```
#install.packages("corrplot")
#install.packages("plumber","rjson")
##install.packages(c("cowplot", "googleway", "ggplot2", "ggrepel",
##"ggspatial", "libwgeom", "sf", "rnaturalearth", "rnaturalearthdata"))
library(zoo, quietly = TRUE)
library(corrplot, quietly = TRUE)
library(tidyverse, quietly = TRUE)
library(tidygraph, quietly = TRUE)
library(igraph, quietly = TRUE)
library(ggplot2, quietly = TRUE)
library(ggraph, quietly = TRUE)
library(rnaturalearth, quietly = TRUE)
library(rnaturalearthdata, quietly = TRUE)
```

```
library(caret, quietly = TRUE)
library(randomForest, quietly = TRUE)
library(nnet, quietly = TRUE)
library(e1071, quietly = TRUE)
```

3 Datenaufbereitung

Zuerst lesen wir die Daten ein. Wir verwenden dafür read_delim anstatt read_csv um den Spaltentyp zu schätzen.

```
data = read_delim("data/house_sales.csv", delim=",")
data = data %>% as_tibble()
```

Wir entfernen einen Ausreißer, der wahrscheinlich nur eine Fehleingabe war.

```
data = data %>% subset(bedrooms != 33)
```

4 Explorative Datenanalyse

Unsere Datensatz enthält Informationen über 21.613 Häuser in der US-amerikanischen Stadt Seattle. Jedes Haus ist durch eine ID gekennzeichnet und ist durch 19 Merkmale beschrieben. Unten findet man einen Überblick dieser Merkmale.

```
summary(data)
```

```
##       id                 date                          price
##  Length:21612       Min.   :2014-05-02 00:00:00   Min.   :  75000
##  Class :character   1st Qu.:2014-07-22 00:00:00   1st Qu.: 321838
##  Mode  :character   Median :2014-10-16 00:00:00   Median : 450000
##                     Mean   :2014-10-29 04:46:26   Mean   : 540084
##                     3rd Qu.:2015-02-17 00:00:00   3rd Qu.: 645000
##                     Max.   :2015-05-27 00:00:00   Max.   :7700000
##     bedrooms        bathrooms      sqft_living       sqft_lot
##  Min.   : 0.000   Min.   :0.000   Min.   :  290   Min.   :    520
##  1st Qu.: 3.000   1st Qu.:1.750   1st Qu.: 1426   1st Qu.:   5040
##  Median : 3.000   Median :2.250   Median : 1910   Median :   7619
##  Mean   : 3.369   Mean   :2.115   Mean   : 2080   Mean   :  15107
##  3rd Qu.: 4.000   3rd Qu.:2.500   3rd Qu.: 2550   3rd Qu.:  10688
##  Max.   :11.000   Max.   :8.000   Max.   :13540   Max.   :1651359
##      floors        waterfront           view          condition
##  Min.   :1.000   Min.   :0.000000   Min.   :0.0000   Min.   :1.000
##  1st Qu.:1.000   1st Qu.:0.000000   1st Qu.:0.0000   1st Qu.:3.000
##  Median :1.500   Median :0.000000   Median :0.0000   Median :3.000
##  Mean   :1.494   Mean   :0.007542   Mean   :0.2343   Mean   :3.409
##  3rd Qu.:2.000   3rd Qu.:0.000000   3rd Qu.:0.0000   3rd Qu.:4.000
##  Max.   :3.500   Max.   :1.000000   Max.   :4.0000   Max.   :5.000
##      grade          sqft_above   sqft_basement       yr_built
##  Min.   : 1.000   Min.   : 290   Min.   :   0.0   Min.   :1900
##  1st Qu.: 7.000   1st Qu.:1190   1st Qu.:   0.0   1st Qu.:1951
##  Median : 7.000   Median :1560   Median :   0.0   Median :1975
##  Mean   : 7.657   Mean   :1788   Mean   : 291.5   Mean   :1971
##  3rd Qu.: 8.000   3rd Qu.:2210   3rd Qu.: 560.0   3rd Qu.:1997
##  Max.   :13.000   Max.   :9410   Max.   :4820.0   Max.   :2015
##   yr_renovated        zipcode           lat             long
##  Min.   :   0.00   Min.   :98001   Min.   :47.16   Min.   :-122.5
##  1st Qu.:   0.00   1st Qu.:98033   1st Qu.:47.47   1st Qu.:-122.3
##  Median :   0.00   Median :98065   Median :47.57   Median :-122.2
##  Mean   :  84.41   Mean   :98078   Mean   :47.56   Mean   :-122.2
##  3rd Qu.:   0.00   3rd Qu.:98118   3rd Qu.:47.68   3rd Qu.:-122.1
##  Max.   :2015.00   Max.   :98199   Max.   :47.78   Max.   :-121.3
##  sqft_living15    sqft_lot15
##  Min.   : 399   Min.   :   651
##  1st Qu.:1490   1st Qu.:  5100
##  Median :1840   Median :  7620
##  Mean   :1987   Mean   : 12769
##  3rd Qu.:2360   3rd Qu.: 10083
##  Max.   :6210   Max.   :871200
```

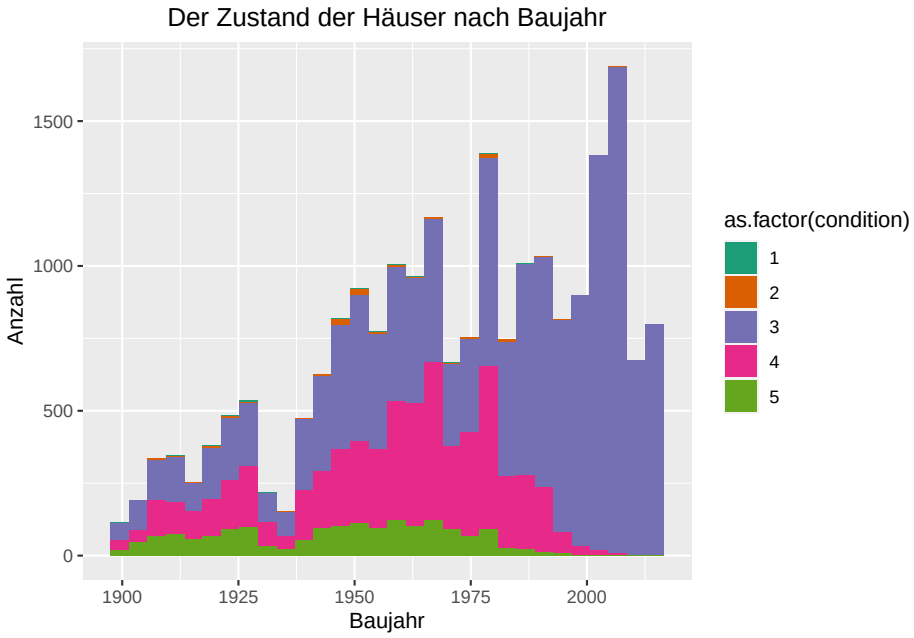
- Der durschnittliche Preis eines Häuses im Datensatz beträgt 540.084 US-Dollar. Das teuerste Haus kostet 7.7 Millionen US-Dollar.
- Die Wohnfläche beträgt durchschnittlich 2.080 Quadraftfuß, was ca. 193 Quadratmeter ist.
- Die Median Größe eines Grundstücks beträgt 7.618 Quadratfuß, wobei das größte Gründstück 1.651.359 Quadratfuß hat.
- Die Häuser in unserem Datensatz haben außerdem durchschnittlich 3.4 Zimmer und 2.25 Badezimmer.
- Von Mehr als 20 Tausend Häuser liegen nur 163 am Wasser.
- Das älterste Haus wurde im Jahr 1900 gebaut. Der durschnittliche Alter der Häuser im Datensatz beträgt 50

Schauen wir uns nun weitere Statistiken graphisch an. Da das Ziel dieser Arbeit die Erstellung mehrerer Modelle für die Vorhersage der Hauspreise ist, wird der Fokus dieser visuellen Datenanalyse auf der Variable **Preis** liegen.

4.0.1 Zustand der Häuser nach Baujahr

```
data %>%
  ggplot(aes(x=yr_built, fill=as.factor(condition))) +
  geom_histogram() +
  ggtitle("Der Zustand der Häuser nach Baujahr") +
  xlab("Baujahr") + ylab("Anzahl") +
  scale_fill_brewer(palette = "Dark2") +
  theme(plot.title = element_text(hjust = 0.5))
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```

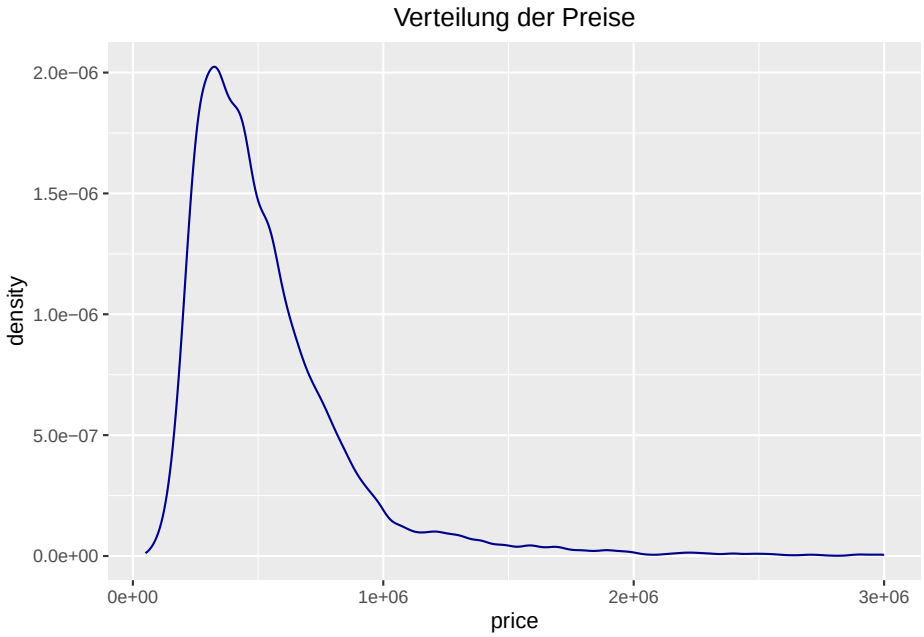


- Auf dem Diagramm ist zu sehen, dass

4.0.2 Verteilung der Preise

```
data %>%
  ggplot(aes(x=price)) +
  geom_line(stat="density", color="darkblue") +
  xlim(50000, 3000000) +
  ggtitle("Verteilung der Preise") +
  theme(plot.title = element_text(hjust = 0.5))
```

```
## Warning: Removed 45 rows containing non-finite values (stat_density).
```



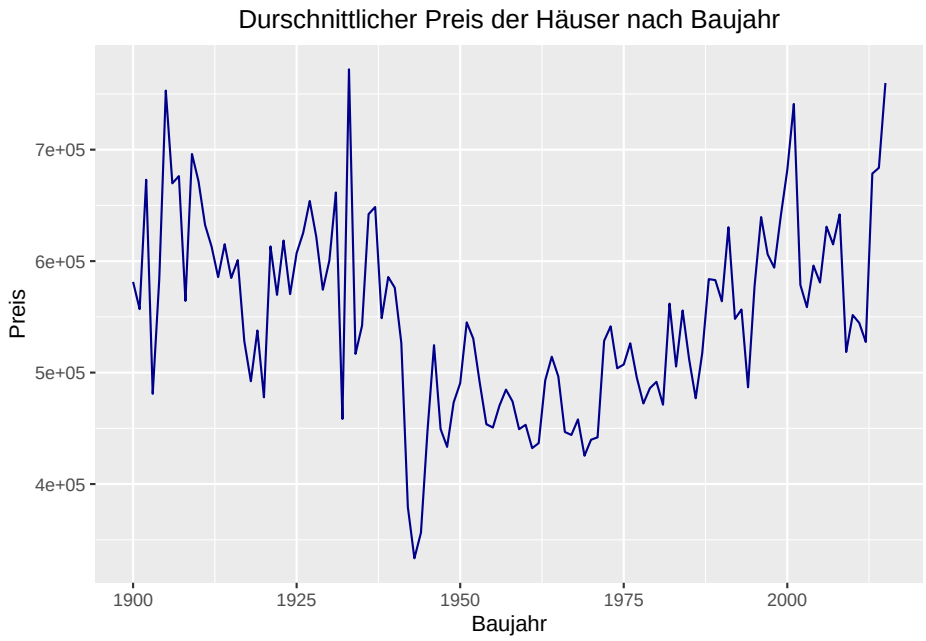
- Auf dem Diagramm sehen wir die Verteilung der Preise für Häuser in unserem Datensatz.
- Die Verteilung folgt einer ungefähren F-Verteilung.
- Die Mehrheit der Häuser kostet zwischen 320.000 und 645.000 US-Dollar.

```
summary(data$price)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   75000  321838  450000  540084  645000 7700000
```

4.0.3 Durschnittlicher Preis der Häuser nach Baujahr

```
data %>%
  ggplot(aes(x=yr_built,y=price)) +
  geom_line(stat = "summary", fun = "mean", color="darkblue") +
  ggtitle("Durschnittlicher Preis der Häuser nach Baujahr") +
  xlab("Baujahr") + ylab("Preis") +
  scale_fill_brewer(palette = "Dark2") +
  theme(plot.title = element_text(hjust = 0.5))
```

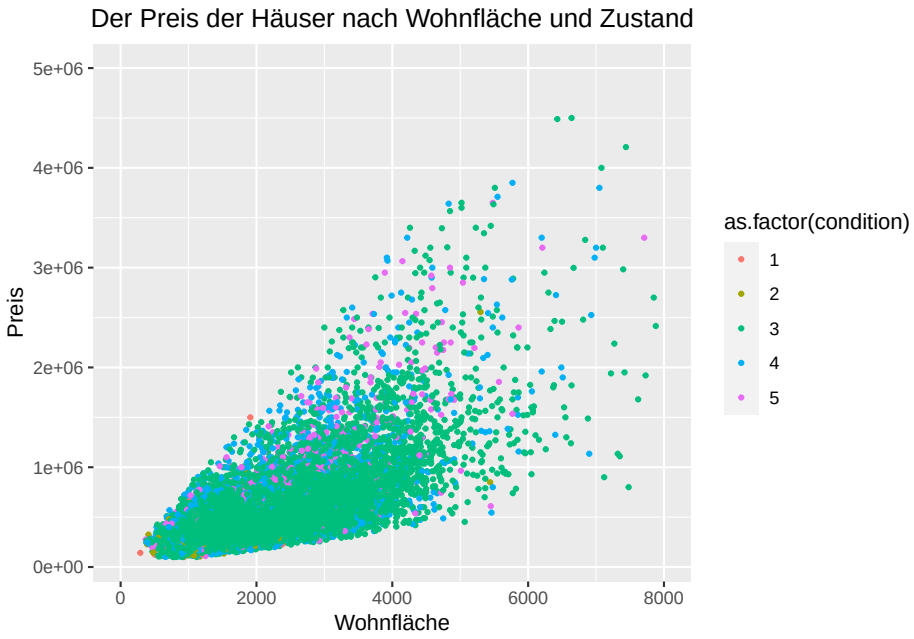


- Auf dem Diagramm sehen wir den durschnittlichen Verkaufspreis der Häuser nach dem Baujahr.
- Aufällig ist, dass Häuser,die zwischen 1900 und 1930 gebaut wurden, durschnittlich einen höheren Preis haben als Häuser die zwischen den Jahren 1945 und 1980 gebaut wurden. Erst ganz junge Häuser, die am Ende des 20. Jahrhunderts und am Anfang des 21. Jahrhunderts gebaut wurden, sind wieder teuerer.

4.0.4 Preis der Häuser nach Wohnfläche und Zustand

```
data %>%
  ggplot(aes(x=sqft_living,y=price, colour =as.factor(condition))) +
  geom_point(size=0.8) +
  ggtitle("Der Preis der Häuser nach Wohnfläche und Zustand") +
  xlab("Wohnfläche") +
  ylab("Preis") +
  scale_fill_brewer(palette = "Dark2") +
  theme(plot.title = element_text(hjust = 0.5)) +
  ylim(100000, 5000000) +
  xlim(0, 8000)
```

```
## Warning: Removed 36 rows containing missing values (geom_point).
```

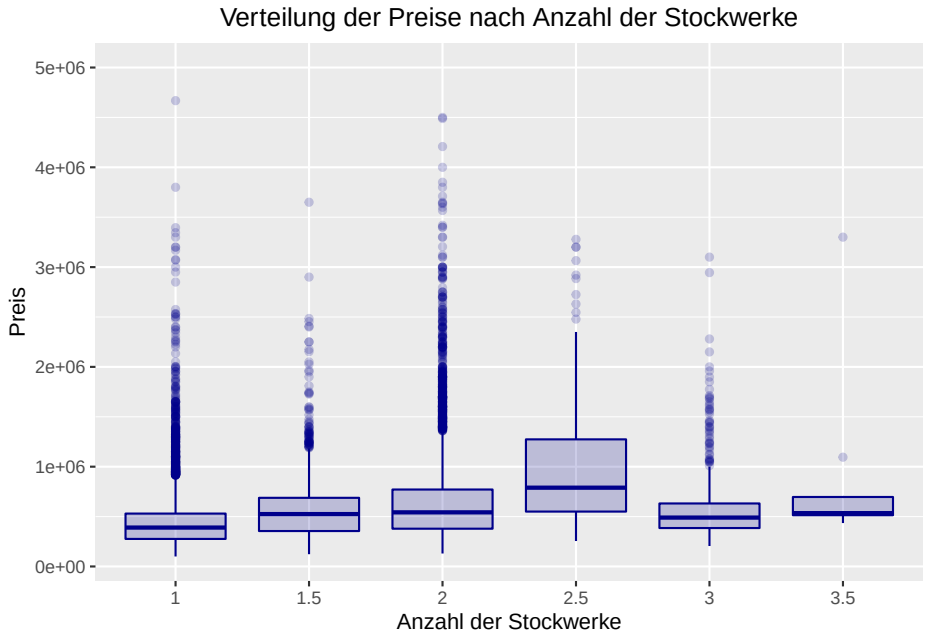


- Das Diagramm zeigt den Preis der Häuser nach Wohnfläche und Zustand.
- Es lässt sich deutlich erkennen, dass mit steigender Wohnfläche auch der Preis für ein Haus steigt.
- Leider kann man nicht deutlich sehen, ob der Zustand auch eine Rolle beim Preis des Hauses spielt. Es ist lediglich zu beobachten, dass die Mehrheit der Häuser in einem mittleren Zustand sind.

4.0.5 Verteilung der Preise nach Anzahl der Stockwerke

```
data %>%
  ggplot(aes(x=as.factor(floors), y=price)) +
  geom_boxplot(color="darkblue", fill="darkblue", alpha=0.2) +
  ylim(100000, 5000000) +
  ggtitle("Verteilung der Preise nach Anzahl der Stockwerke") +
  xlab("Anzahl der Stockwerke") + ylab("Preis") +
  scale_colour_brewer(palette = "Dark2") +
  theme(plot.title = element_text(hjust = 0.5))
```

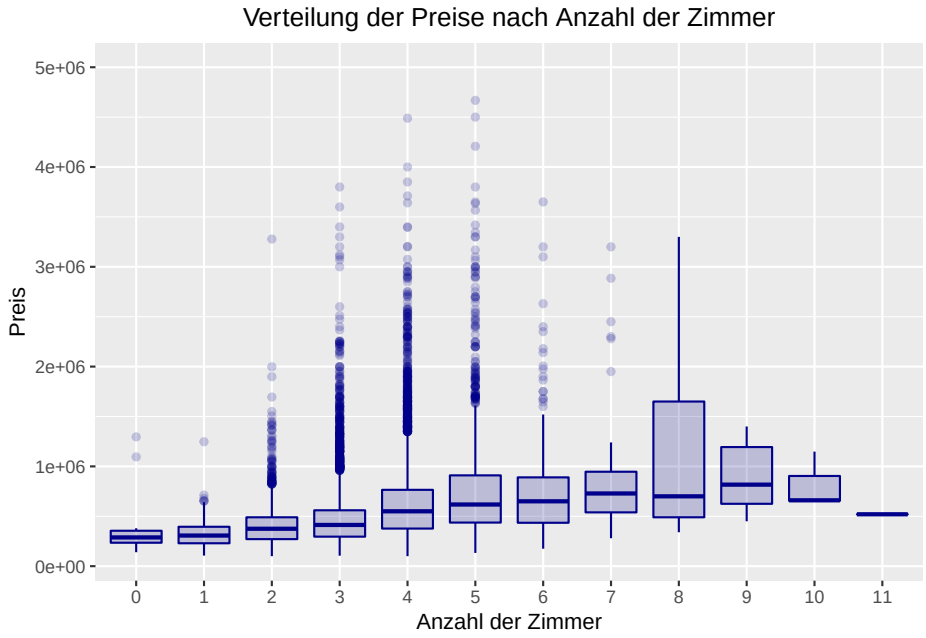
```
## Warning: Removed 32 rows containing non-finite values (stat_boxplot).
```



- Das Diagramm zeigt die Verteilung der Hauspreise nach der Anzahl der Stockwerke.
- Es lässt sich erkennen, dass je mehr Stockwerke das Haus hat, desto höher der Preis ist. Interessenterweise gilt dieser Trend nur bis zu 2.5 Stockwerken. Wenn ein Haus 3 oder 3.5 Stockwerke hat, ist der Preis durschnittlich niedriger als bei Häusern mit nur 2.5 Stockwerken.
- Vielleicht lässt sich eine deutlichere Tendenz bei der Anzahl der Zimmer festellen.

4.0.6 Verteilung der Preise nach Anzahl der Zimmer

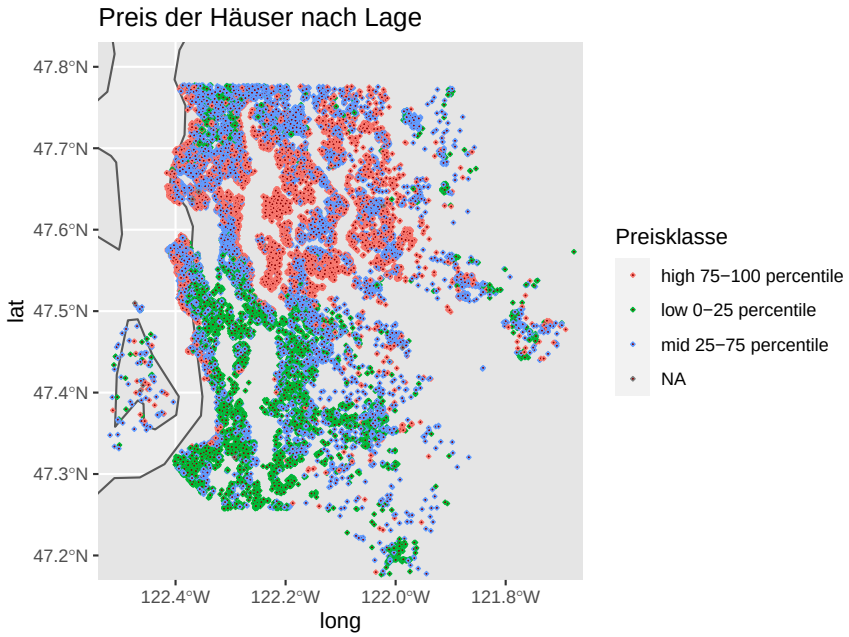
```
data %>%
  ggplot(aes(x=as.factor(bedrooms), y=price)) +
  geom_boxplot(color="darkblue", fill="darkblue", alpha=0.2) +
  ylim(100000, 5000000) +
  ggtitle("Verteilung der Preise nach Anzahl der Zimmer") +
  xlab("Anzahl der Zimmer") + ylab("Preis") +
  scale_colour_brewer(palette = "Dark2") +
  theme(plot.title = element_text(hjust = 0.5))
```



- Auf der Grafik können wir die Verteilung der Hauspreise nach der Anzahl der Zimmer beobachten.
- Es gibt einen klaren aufsteigenden Trend: Also je mehr Zimmer ein Haus hat, desto mehr wird er wahrscheinlich kosten.

4.0.7 Preis der Häuser nach Lage

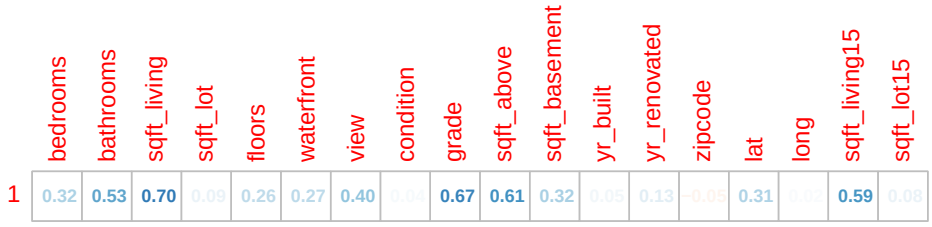
```
data = data %>% mutate(pricecat = case_when(
  price < 321950  ~ 'low 0-25 percentile',
  price < 645000  ~ 'mid 25-75 percentile',
  price > 645000  ~ 'high 75-100 percentile'
))

world <- ne_countries(scale = "medium", returnclass = "sf")
ggplot(data = world) +
    geom_sf() +
    geom_point(data = data, aes(x = long, y = lat, col=as.factor(pricecat)), size = 0.5,
     shape = 23, fill = "darkred") +
    ggtitle("Preis der Häuser nach Lage")+
    coord_sf(xlim = c(-122.5, -121.7), ylim = c(47.20, 47.8)) +
   labs(color="Preisklasse")
```



- Auf der geograpfischen Karte können wir die Lage der Häuser im Datensatz sehen, gefärbt nach Preisklasse.
- Wir können beobachten, dass die teuersten Häuser (rot) näher zum Stadtzentrum liegen, als billigere Häuser (grün). Häuser, die preismäßig in der Mitte liegen (blaue), sind in der Stadt ungefähr regelmäßig verteilt.

4.0.8 Korrelation der einzelnen Merkmale mit Preis

```
datacor = data %>% select(-c("id", "date", "pricecat"))
corrplot(cor(datacor$price, datacor), method="number",diag = FALSE, tl.cex = 1,
   number.cex=0.75, cl.pos = "n")
```



5 Modelierung

Zuerst wird ein fester Seed gesetzt, sodass die Ergebnisse gleich bleiben.

```
set.seed(3000)
```

```
data = data %>% select(-c(id,date,condition,sqft_living15,sqft_lot15,zipcode,lat,long,pricecat,sqft_lot
data
```

```
## # A tibble: 21,612 x 10
##       price bedrooms bathrooms sqft_living floors waterfront  view grade
##       <dbl>    <dbl>     <dbl>       <dbl>  <dbl>      <dbl> <dbl> <dbl>
##  1  221900        3      1           1180      1          0     0     7
##  2  538000        3      2.25        2570      2          0     0     7
##  3  180000        2      1            770      1          0     0     6
##  4  604000        4      3           1960      1          0     0     7
##  5  510000        3      2           1680      1          0     0     8
##  6 1225000        4      4.5         5420      1          0     0    11
##  7  257500        3      2.25        1715      2          0     0     7
##  8  291850        3      1.5         1060      1          0     0     7
##  9  229500        3      1           1780      1          0     0     7
## 10  323000        3      2.5         1890      2          0     0     7
## # ... with 21,602 more rows, and 2 more variables: sqft_above <dbl>,
## #   sqft_basement <dbl>
```

Die Daten werden zuerst in Test- und Trainingsdaten aufgeteilt.

```
part = createDataPartition(data$price, times = 2, p = 2/3)
train = data[part$Resample1,]
test  = data[-part$Resample1,]
train
```

```
## # A tibble: 14,409 x 10
##       price bedrooms bathrooms sqft_living floors waterfront  view grade
##       <dbl>    <dbl>     <dbl>       <dbl>  <dbl>      <dbl> <dbl> <dbl>
##  1  221900        3      1           1180      1          0     0     7
##  2  538000        3      2.25        2570      2          0     0     7
##  3  180000        2      1            770      1          0     0     6
##  4  604000        4      3           1960      1          0     0     7
##  5  510000        3      2           1680      1          0     0     8
##  6 1225000        4      4.5         5420      1          0     0    11
##  7  291850        3      1.5         1060      1          0     0     7
##  8  662500        3      2.5         3560      1          0     0     8
##  9  468000        2      1           1160      1          0     0     7
## 10  400000        3      1.75        1370      1          0     0     7
## # ... with 14,399 more rows, and 2 more variables: sqft_above <dbl>,
## #   sqft_basement <dbl>
```

Die Test- und Trainingsdaten werden abgespeichert.

```
write.table(train,
            file = "data/train.csv",
            row.names = FALSE)
write.table(test,
            file = "data/test.csv",
            row.names = FALSE)
```

Das erste Modell, das verwendet wird, ist eine lineare Regression. Die Funktion RMSE berechnet den Root Mean Square Error, und liefert damit die Vergleichbarkeit der Güte des Modells im Vergleich zu den anderen Varianten.

```
RMSE = function(true, pred) {
    ret <- sqrt(crossprod(true - pred)/length(pred))
    cat("RMSE: ", round(ret, 2))
    ret
}
model_r_linearModel = lm(price ~ . , data = train)
pred_r_linearModel = predict(model_r_linearModel, test)
```

```
## Warning in predict.lm(model_r_linearModel, test): prediction from a rank-
## deficient fit may be misleading
```

```
stats_r_linearModel = RMSE(test$price, pred_r_linearModel)
```

```
## RMSE:  238345.5
```

Die zweite Methode ist ein Random Forest:

```
model_r_randomForest = randomForest(price ~ ., data = train)
pred_r_randomForest = predict(model_r_randomForest, test)
stats_r_randomForest = RMSE(test$price, pred_r_randomForest)
```

```
## RMSE:  218043.6
```

Das dritte Modell ist ein Neural Network:

```
model_r_nnet = nnet(price ~ ., data = train,
                    size = 100, MaxNWts = 10000, trace = FALSE, maxit = 150)
pred_r_nnet = predict(model_r_nnet, test)
stats_r_nnet = RMSE(test$price, pred_r_nnet)
```

```
## RMSE:  653079.1
```

Als zusätzliche Features werden außerdem eine Poison-Regression als auch die Regression mittels Support Vector Machine (SVM) durchgeührt. Poisson-Regression:

```
model_r_poissonReg = glm(price ~ ., data = train, family = poisson)
pred_r_poissonReg = predict(model_r_poissonReg, test)
```

```
## Warning in predict.lm(object, newdata, se.fit, scale = 1, type = if (type == :
## prediction from a rank-deficient fit may be misleading
```

```
stats_r_poissonReg= RMSE(test$price, pred_r_poissonReg)
```

```
## RMSE:  653069
```

SVM-regression:

```
model_r_svm = svm(price ~ ., data = train)
pred_r_svm = predict(model_r_svm, test)
stats_r_svm = RMSE(test$price, pred_r_svm)
```

```
## RMSE:  237218.9
```

Vergleich der Modelle mittels RMSE-Werten:

```
res = t(t(sort(c(`Lineare Regression` = stats_r_linearModel,
                 `Poisson-Regression` = stats_r_poissonReg,
                 `Support Vector Machine` = stats_r_svm,
                 `Random Forest` = stats_r_randomForest,
                 `Neural Network` = stats_r_nnet))))
colnames(res) = "RMSE"
round(res, 3)
```

```
##                            RMSE
## Random Forest          218043.6
## Support Vector Machine 237218.9
## Lineare Regression     238345.5
## Poisson-Regression     653069.0
## Neural Network         653079.1
```

Alle Modelle weisen einen sehr hohen RMSE auf und sind daher eigentlich ungeeignet, um sichere/gute Vorhesagen zu treffen. Unter der Annahme, dass nur die hier gezeigten Methoden und Modelle zur Verfügung stehen, ist das beste Modell: Random Forest.

5.1 ...