DataChallenge_ShapeMyCity

Irma Glatt

27 11 2020

Introduction

In this sub part of the challenge, I want to find out if the answers given in the question “Wie engagiert sind Sie in der Gestaltung des städtischen Lebens?” can be found to be correlated with any of the main demographic factors. Therefore, I use the following variables for my analysis:

Dependent variable: - v_311: “Wie engagiert sind Sie in der Gestaltung des städtischen Lebens?”

Independent variables: - v_6: Geschlecht (Sex), - v_22: Altersgruppe (Altersgruppe), - v_7: Beziehungsstand (Beziehungsstatus), - v_113: Quartier (Quartier), - v_99: Lebensdauer in Luzern (Zugzug), - v_142: höchster Bildungsabschluss (Bildung), - v_14: Art des Haushalts (HH), - v_158: Schulpflichtige Kinder (Kinder_Schule), - v_15: Erwerbstätigkeit (Erwerb), - v_66: Arbeitsplatz in Luzern (ArbeitsplatzLuzern)

Load packages

```r
library(nnet) # for multinomial logistic regression
library(ggplot2)
library(MASS) # for ordinal logistic regression
library(tree)
library(plyr)
library(naniar)
library(ggpubr)
library(ipred)
library("xlsx")
library(randomForest)
```

Load and prepare data

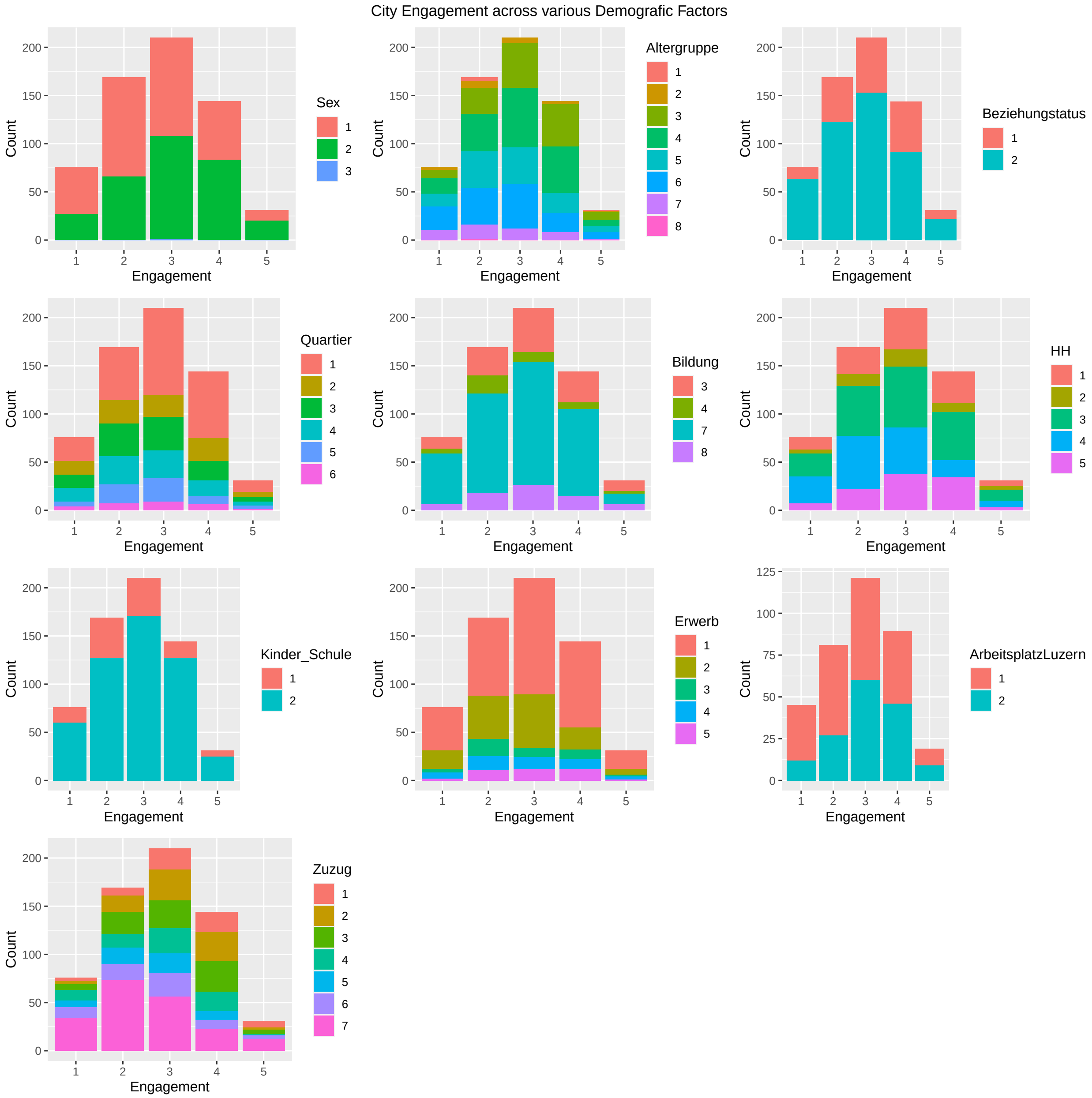
```r
df <- read.csv2("./project/HackdaysLucerne-LifeQualityAnalysis/cleaned_data.CSV")

# Excluding all observations with no information about engagement
df <- df[!(df$v_311==-77),]

# Change unknown datapoints to NA
df$ArbeitsplatzLuzern[df$ArbeitsplatzLuzern == -77] <- NA
```

```r
# Convert the variables to factors and ordered factors, respectively.
df$Engagement <- ordered(df$v_311)
df$Sex <- as.factor(df$Sex)
df$Altergruppe <- ordered(df$Altergruppe)
df$Beziehungstatus <- as.factor(df$Beziehungstatus)
df$Quartier <- as.factor(df$Quartier)
df$Bildung <- as.factor(df$Bildung)
df$HH <- as.factor(df$HH)
df$Erwerb <- as.factor(df$Erwerb)
df$ArbeitsplatzLuzern <- as.factor(df$ArbeitsplatzLuzern)
df$Kinder_Schule <- as.factor(df$Kinder_Schule)
df$Zuzug <- ordered(df$Zuzug)
```

Plotting the data

```r
df1 <- data.frame(table(df$Engagement, df$Sex))
names(df1) <- c("Engagement", "Sex", "Count")

p1 <- ggplot(data = df1, aes(x=Engagement, y=Count, fill=Sex)) +
        geom_bar(stat = "identity")

df2 <- data.frame(table(df$Engagement, df$Altergruppe))
names(df2) <- c("Engagement", "Altergruppe", "Count")

p2 <- ggplot(data = df2, aes(x=Engagement, y=Count, fill=Altergruppe)) +
        geom_bar(stat = "identity")

df3 <- data.frame(table(df$Engagement, df$Beziehungstatus))
names(df3) <- c("Engagement", "Beziehungstatus", "Count")

p3 <- ggplot(data = df3, aes(x=Engagement, y=Count, fill=Beziehungstatus)) +
        geom_bar(stat = "identity")

df4 <- data.frame(table(df$Engagement, df$Quartier))
names(df4) <- c("Engagement", "Quartier", "Count")

p4 <- ggplot(data = df4, aes(x=Engagement, y=Count, fill=Quartier)) +
        geom_bar(stat = "identity")

df5 <- data.frame(table(df$Engagement, df$Bildung))
names(df5) <- c("Engagement", "Bildung", "Count")

p5 <- ggplot(data = df5, aes(x=Engagement, y=Count, fill=Bildung)) +
        geom_bar(stat = "identity")

df6 <- data.frame(table(df$Engagement, df$HH))
names(df6) <- c("Engagement", "HH", "Count")

p6 <- ggplot(data = df6, aes(x=Engagement, y=Count, fill=HH)) +
        geom_bar(stat = "identity")
```

```r
df7 <- data.frame(table(df$Engagement, df$Kinder_Schule))
names(df7) <- c("Engagement", "Kinder_Schule", "Count")

p7 <- ggplot(data = df7, aes(x=Engagement, y=Count, fill=Kinder_Schule)) +
        geom_bar(stat = "identity")

df8 <- data.frame(table(df$Engagement, df$Erwerb))
names(df8) <- c("Engagement", "Erwerb", "Count")

p8 <- ggplot(data = df8, aes(x=Engagement, y=Count, fill=Erwerb)) +
        geom_bar(stat = "identity")

df9 <- data.frame(table(df$Engagement, df$ArbeitsplatzLuzern))
names(df9) <- c("Engagement", "ArbeitsplatzLuzern", "Count")

p9 <- ggplot(data = df9, aes(x=Engagement, y=Count, fill=ArbeitsplatzLuzern)) +
        geom_bar(stat = "identity")

df10 <- data.frame(table(df$Engagement, df$Zuzug))
names(df10) <- c("Engagement", "Zuzug", "Count")

p10 <- ggplot(data = df10, aes(x=Engagement, y=Count, fill=Zuzug)) +
        geom_bar(stat = "identity")

figure <- ggarrange(p1, p2, p3, p4, p5, p6, p7, p8, p9, p10, nrow=4, ncol=3)
annotate_figure(figure, top = "City Engagement across various Demografic Factors")
```



Interpretation

Just by looking at the plots, there is no obvious irregularity one can observe, that indicates a special correlation between one of the independent variables and the dependent variable.

Regression Model

As the dependent variable is an ordered factor, I apply an ordered logicstic regression model. Due to the fact, that the variable *ArbeitsplatzLuzern* has many NA-values, I skip this variable for the following analysis.

```
m_olr <- polr(Engagement ~ Sex + Altergruppe + Beziehungsstatus +
                        Quartier + Zuzug + Bildung  + HH + Kinder_Schule +
                        Erwerb, data = df, Hess = TRUE)
```

```r
summary(m_olr)
```

```
## Call:
## polr(formula = Engagement ~ Sex + Altergruppe + Beziehungstatus +
##     Quartier + Zuzug + Bildung + HH + Kinder_Schule + Erwerb,
##     data = df, Hess = TRUE)
##
## Coefficients:
##                       Value Std. Error   t value
## Sex2               0.5707773     0.1571  3.633254
## Sex3               0.8551236     1.5633  0.547010
## Altergruppe.L      0.2792491     1.1017  0.253467
## Altergruppe.Q     -1.3292895     1.0067 -1.320401
## Altergruppe.C      0.5693041     0.8105  0.702386
## Altergruppe^4     -0.0516425     0.6134 -0.084191
## Altergruppe^5      0.0295543     0.4419  0.066883
## Altergruppe^6      0.0540618     0.2840  0.190388
## Altergruppe^7     -0.0396840     0.1826 -0.217353
## Beziehungstatus2 -0.2283932     0.2207 -1.034960
## Quartier2          0.0846153     0.2440  0.346786
## Quartier3         -0.1921088     0.2191 -0.876906
## Quartier4         -0.3294613     0.2386 -1.380948
## Quartier5         -0.2165736     0.2642 -0.819583
## Quartier6         -0.1267453     0.3899 -0.325051
## Zuzug.L           -1.0985604     0.2495 -4.403303
## Zuzug.Q            0.0627499     0.2293  0.273609
## Zuzug.C           -0.0001359     0.2277 -0.000597
## Zuzug^4           -0.1623990     0.2220 -0.731692
## Zuzug^5           -0.2562797     0.2233 -1.147843
## Zuzug^6            0.0432902     0.2232  0.193913
## Bildung4          -0.9424492     0.3482 -2.706368
## Bildung7          -0.7230385     0.1972 -3.666565
## Bildung8          -0.0931333     0.2883 -0.323028
## HH2                0.1267504     0.3631  0.349100
## HH3                0.0864050     0.2595  0.332921
## HH4               -0.3425237     0.3107 -1.102558
## HH5                0.0495053     0.2743  0.180467
## Kinder_Schule2    -0.1581316     0.2674 -0.591433
## Erwerb2           -0.3073529     0.1940 -1.584278
## Erwerb3           -0.1687054     0.4430 -0.380827
## Erwerb4            0.3650437     0.4054  0.900410
## Erwerb5           -0.2843057     0.3336 -0.852144
##
## Intercepts:
##     Value   Std. Error t value
## 1|2 -2.7217  0.4756    -5.7229
## 2|3 -1.0205  0.4646    -2.1967
## 3|4  0.5984  0.4633     1.2917
## 4|5  2.7368  0.4886     5.6014
##
## Residual Deviance: 1730.369
## AIC: 1804.369
```

Interpretation

As there were only used factors in this model, this result gets really hard to interpret… to reduce the complexitiy of the modelinterpretation, I try a logistic regression for binary data by grouping the answers to the “Engagement-Question”:

```r
# Create a second dataset in which the answers to the "Engagement-question"
# are put togehter:
# 1: "eher engagiert" and "sehr engagiert"
# 0: "weniger engagiert" and "nicht engagiert"
# Answers with
df2 <- df
df2$Engagement <- as.integer(df2$Engagement)
df2$Engagement[df2$Engagement == 2] <- 1
df2$Engagement[df2$Engagement == 3] <- 4
df2$Engagement[df2$Engagement == 4] <- 0
df2 <- df2[!(df2$Engagement==5),]


glm.df_new <- glm(Engagement ~ Sex + Altergruppe + Beziehungstatus +
                    Quartier + Zuzug + Bildung  + HH + Kinder_Schule + Erwerb,
                  family = "binomial", data = df2)

summary(glm.df_new)
```

```
##
## Call:
## glm(formula = Engagement ~ Sex + Altergruppe + Beziehungstatus +
##     Quartier + Zuzug + Bildung + HH + Kinder_Schule + Erwerb,
##     family = "binomial", data = df2)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.9304  -0.9178  -0.6166   1.0495   2.1902
##
## Coefficients:
##                     Estimate Std. Error z value Pr(>|z|)
## (Intercept)          2.67307  202.12501   0.013  0.98945
## Sex2                -0.59528    0.19679  -3.025  0.00249 **
## Sex3               -15.99676 1455.39758  -0.011  0.99123
## Altergruppe.L       -0.68254  873.27735  -0.001  0.99938
## Altergruppe.Q       17.22977  873.27719   0.020  0.98426
## Altergruppe.C       -0.65761  696.63425  -0.001  0.99925
## Altergruppe^4        8.55293  456.05430   0.019  0.98504
## Altergruppe^5        0.09214  242.20389   0.000  0.99970
## Altergruppe^6        1.74741   99.51968   0.018  0.98599
## Altergruppe^7       -0.11997   27.60257  -0.004  0.99653
## Beziehungsstatus2    0.05280    0.28349   0.186  0.85225
## Quartier2            0.14164    0.29840   0.475  0.63504
## Quartier3            0.29154    0.27310   1.068  0.28572
## Quartier4            0.48273    0.29083   1.660  0.09694 .
## Quartier5            0.19743    0.33765   0.585  0.55874
## Quartier6            0.19710    0.48189   0.409  0.68252
## Zuzug.L              1.13533    0.31950   3.553  0.00038 ***
## Zuzug.Q             0.16729    0.30389   0.550  0.58199   
## Zuzug.C             0.26340    0.29354   0.897  0.36955   
## Zuzug^4             0.22056    0.28295   0.780  0.43569   
## Zuzug^5             0.08465    0.27842   0.304  0.76110   
## Zuzug^6             0.11437    0.27282   0.419  0.67505   
## Bildung4            1.27201    0.43909   2.897  0.00377 **
## Bildung7            0.67113    0.25230   2.660  0.00781 **
## Bildung8           -0.09869    0.38428  -0.257  0.79732   
## HH2                -0.13677    0.45972  -0.298  0.76608   
## HH3                 0.14621    0.32570   0.449  0.65350   
## HH4                 0.42115    0.37708   1.117  0.26405   
## HH5                -0.18583    0.36115  -0.515  0.60688   
## Kinder_Schule2      0.08987    0.32531   0.276  0.78235   
## Erwerb2             0.36524    0.24005   1.522  0.12813   
## Erwerb3             0.52064    0.54946   0.948  0.34336   
## Erwerb4            -0.04700    0.50775  -0.093  0.92624   
## Erwerb5             0.48816    0.41742   1.169  0.24222   
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## 
## (Dispersion parameter for binomial family taken to be 1)
## 
##     Null deviance: 810.44  on 598  degrees of freedom
## Residual deviance: 708.34  on 565  degrees of freedom
## AIC: 776.34
## 
## Number of Fisher Scoring iterations: 14
```

```
# As link functions are used the interpretation of these coefficients must be adapted.
# In particular, we can interpret the conc coefficient by applying the
# exponential function.
exp(coef(glm.df_new))
```

```
##      (Intercept)             Sex2             Sex3    Altergruppe.L 
##     1.448433e+01     5.514064e-01     1.129000e-07     5.053338e-01 
##    Altergruppe.Q    Altergruppe.C    Altergruppe^4    Altergruppe^5 
##     3.039436e+07     5.180866e-01     5.181908e+03     1.096518e+00 
##    Altergruppe^6    Altergruppe^7 Beziehungstatus2        Quartier2 
##     5.739742e+00     8.869478e-01     1.054217e+00     1.152157e+00 
##        Quartier3        Quartier4        Quartier5        Quartier6 
##     1.338492e+00     1.620497e+00     1.218267e+00     1.217869e+00 
##          Zuzug.L          Zuzug.Q          Zuzug.C          Zuzug^4 
##     3.112187e+00     1.182096e+00     1.301350e+00     1.246773e+00 
##          Zuzug^5          Zuzug^6         Bildung4         Bildung7 
##     1.088336e+00     1.121171e+00     3.568000e+00     1.956443e+00 
##         Bildung8              HH2              HH3              HH4 
##     9.060245e-01     8.721705e-01     1.157437e+00     1.523712e+00 
##              HH5   Kinder_Schule2          Erwerb2          Erwerb3 
##     8.304182e-01     1.094035e+00     1.440856e+00     1.683102e+00 
##          Erwerb4          Erwerb5 
##     9.540840e-01     1.629313e+00
```

Interpretation

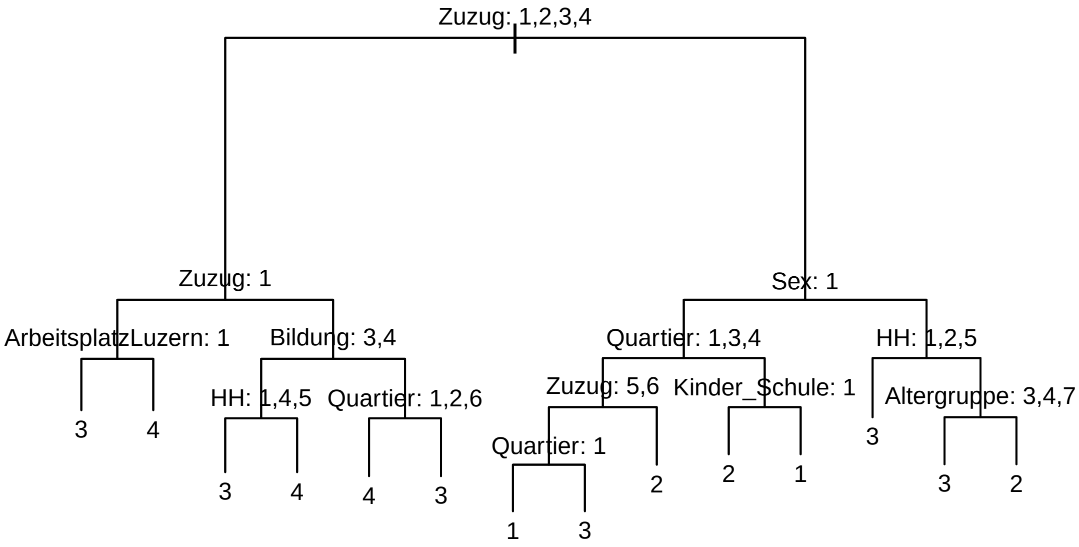
With this new model, the factors *Zuzug* and *Bildung* are slightly significant. This is in agreement to further sup parts of this challenge.

Classification Tree

```
set.seed(12)

# define model
tree.classification.df <- tree(Engagement ~ Sex + Altergruppe + Beziehungstatus +
                                 Quartier + Zuzug + Bildung + HH + Kinder_Schule +
                                 Erwerb + ArbeitsplatzLuzern, data = df)

summary(tree.classification.df)
```

```
##
## Classification tree:
## tree(formula = Engagement ~ Sex + Altergruppe + Beziehungstatus +
##     Quartier + Zuzug + Bildung + HH + Kinder_Schule + Erwerb +
##     ArbeitsplatzLuzern, data = df)
## Variables actually used in tree construction:
## [1] "Zuzug"              "ArbeitsplatzLuzern" "Bildung"
## [4] "HH"                 "Quartier"           "Sex"
## [7] "Kinder_Schule"      "Altergruppe"
## Number of terminal nodes:  14
## Residual mean deviance:  2.456 = 837.4 / 341
## Misclassification error rate: 0.5437 = 193 / 355
```

```
plot(tree.classification.df)
text(tree.classification.df, pretty=1, cex=0.75)
```



```r
# Setup training and test set
ratio <- 0.8
total <- nrow(df)
train <- sample(1:total, as.integer(total * ratio))
test = df[-train, ]

tree.classification.df.pred <- predict(tree.classification.df, df[train,], type="class")

# confusion table to determine classification error on *train data*
(tree.classification.df.pred.ct <- table(tree.classification.df.pred,
                                          df[train,]$Engagement))
```

```
##
## tree.classification.df.pred  1  2  3  4  5
##                            1 19 14 11  8  2
##                            2 18 48 28  7  6
##                            3 18 47 80 46 10
##                            4  8 18 50 60  6
##                            5  0  0  0  0  0
```

```r
tree.classification.df.pred.correct <- 0
tree.classification.df.pred.error <- 0
for (i1 in 1:5) {
  for (i2 in 1:5) {
    if (i1 == i2) {
      tree.classification.df.pred.correct <- tree.classification.df.pred.correct +
        tree.classification.df.pred.ct[i1,i2]
    }else{
     tree.classification.df.pred.error <- tree.classification.df.pred.error +
       tree.classification.df.pred.ct[i1,i2]
    }
  }
}
(tree.classification.df.pred.rate <- tree.classification.df.pred.correct/
    sum(tree.classification.df.pred.ct))
```

```
## [1] 0.4107143
```

```
# portion of correctly classified observations 41.1%
(tree.classification.df.pred.error <- 1 - tree.classification.df.pred.rate)
```

```
## [1] 0.5892857
```

```
# train error (pruned): 58.9%

# and on test data --> test error
tree.classification.df.pred.test <- predict(tree.classification.df,
                                              df[-train,], type="class")
# confusion table to determine classification error on *test data*
(tree.classification.df.pred.test.ct <- table(tree.classification.df.pred.test,
                                                df[-train,]$Engagement))
```

```
##
## tree.classification.df.pred.test  1  2  3  4  5
##                                 1  3  7  4  0  0
##                                 2  5 12  9  3  3
##                                 3  4 13 18  9  3
##                                 4  1 10 10 11  1
##                                 5  0  0  0  0  0
```

```
tree.classification.df.pred.correct <- 0
tree.classification.df.pred.error <- 0
for (i1 in 1:5) {
  for (i2 in 1:5) {
    if (i1 == i2) {
      tree.classification.df.pred.correct <- tree.classification.df.pred.correct +
        tree.classification.df.pred.test.ct[i1,i2]
    }else{
      tree.classification.df.pred.error <- tree.classification.df.pred.error +
        tree.classification.df.pred.test.ct[i1,i2]
    }
  }
}
(tree.classification.df.pred.rate <- tree.classification.df.pred.correct/
    sum(tree.classification.df.pred.test.ct))
```

```
## [1] 0.3492063
```

```
# portion of correctly classified observations 34.9%
(tree.classification.df.pred.error <- 1 - tree.classification.df.pred.rate)
```

```
## [1] 0.6507937
```

```
# test error (pruned): 65.1%
```

Interpretation

The classification tree uses the variable *Zuzug* as the main separator. In the left main branch - for people living between 0-15 years in Lucerne, the final branches result in responses 4 (nicht engagiert) and 3 (eher weniger engagiert). In the right main branch - representing people that live more than 15 years in Lucerne - the final branches result in the responses 1 (sehr engagiert), 2 (eher engagiert) and 3 (eher weniger engagiert).

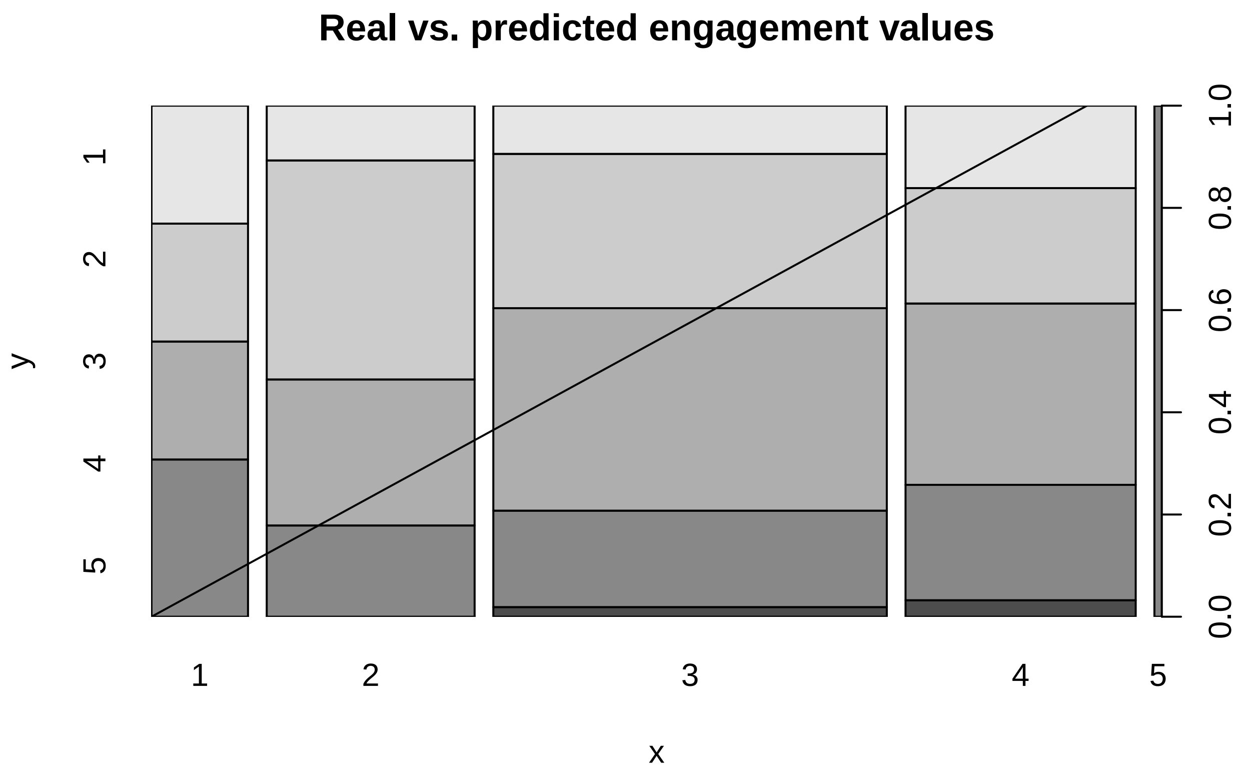
So this might be an indication, that the time of living in the city of Lucerne plays a role in the measure of engagement.

However, as the misclassification rate of the tree is quite high (over 50%), the significance of the model is rather low.

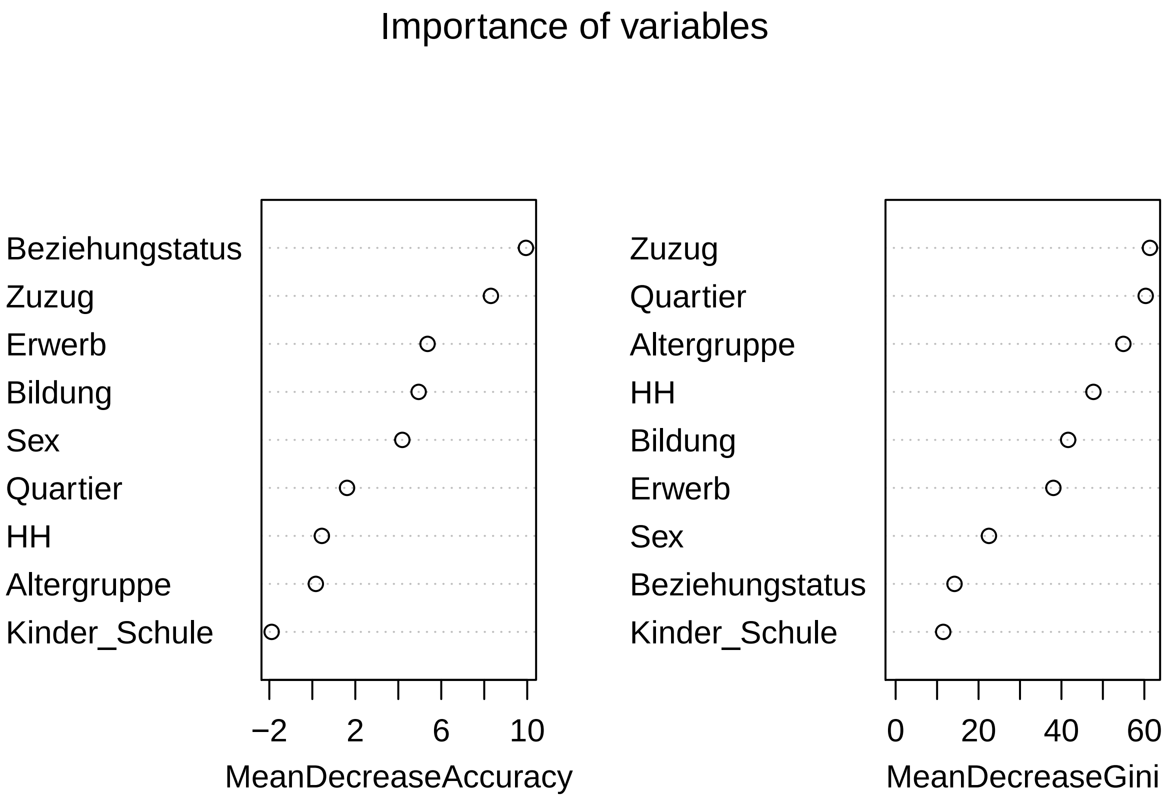
```
# Setup training and test set
set.seed (1)

ratio <- 0.8
total <- nrow(df)
train <- sample(1:total, as.integer(total * ratio))
test = df[-train, ]

# mtry = 12 means that we should use all 9 predictors for each split of the tree,
# hence, do bagging (not randomForrest).
bag.df=randomForest(Engagement ~ Sex + Altergruppe + Beziehungsstatus + Quartier +
                      Zuzug + Bildung + HH + Kinder_Schule +
                      Erwerb, data = df,
                    subset=train, mtry=9, importance =TRUE)

print(bag.df)
```

```
##
## Call:
##  randomForest(formula = Engagement ~ Sex + Altergruppe + Beziehungsstatus +      Quartier + Zuzug + B
##                Type of random forest: classification
##                      Number of trees: 500
## No. of variables tried at each split: 9
##
##         OOB estimate of  error rate: 69.05%
## Confusion matrix:
##    1  2  3  4 5 class.error
## 1  4 26 21  8 1   0.9333333
## 2 15 43 46 24 3   0.6717557
## 3 13 50 65 36 3   0.6107784
## 4  8 25 39 43 2   0.6324786
## 5  2  5 11 10 1   0.9655172
```

```r
# How well does the bagged model perform on the test set?
yhat.bag = predict(bag.df,newdata=df[-train,])
plot(yhat.bag, test[,"Engagement"], "main"="Real vs. predicted engagement values")
abline(0,1)
```



```r
# Investigating variable importance
# importance(bag.df)
varImpPlot (bag.df, "main"="Importance of variables")
```



Interpretation

To improve the previous tree and to make a more reliable prediction I tried bagging. However, with an out-of-bag estimate of error rate of nearly 70%, the model the significance of the model is still rather low. Interestingly however, one of the most important variables is still *Zuzug*.

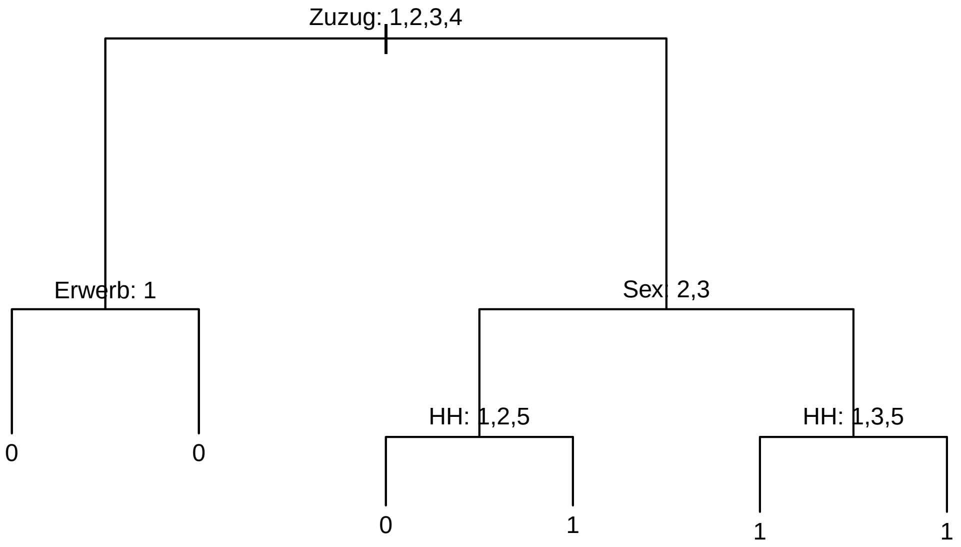
Finally, lets try again with the summarised responses used already in the binary logistic regression above:

```
set.seed(12)

# define the column Engagement as factor
df2$Engagement <- as.factor(df2$Engagement)

ratio <- 0.8
total <- nrow(df2)
train <- sample(1:total, as.integer(total * ratio))
test = df[-train, ]

# define model
tree.classification.df <- tree(Engagement ~ Sex + Altergruppe + Beziehungstatus +
                                 Quartier + Zuzug + Bildung + HH + Kinder_Schule +
                                 Erwerb, data = df2)

summary(tree.classification.df)
```

```
##
## Classification tree:
```

```
## tree(formula = Engagement ~ Sex + Altergruppe + Beziehungstatus +
##     Quartier + Zuzug + Bildung + HH + Kinder_Schule + Erwerb,
##     data = df2)
## Variables actually used in tree construction:
## [1] "Zuzug"  "Erwerb" "Sex"    "HH"
## Number of terminal nodes:  6
## Residual mean deviance:  1.217 = 721.9 / 593
## Misclassification error rate: 0.3322 = 199 / 599
```

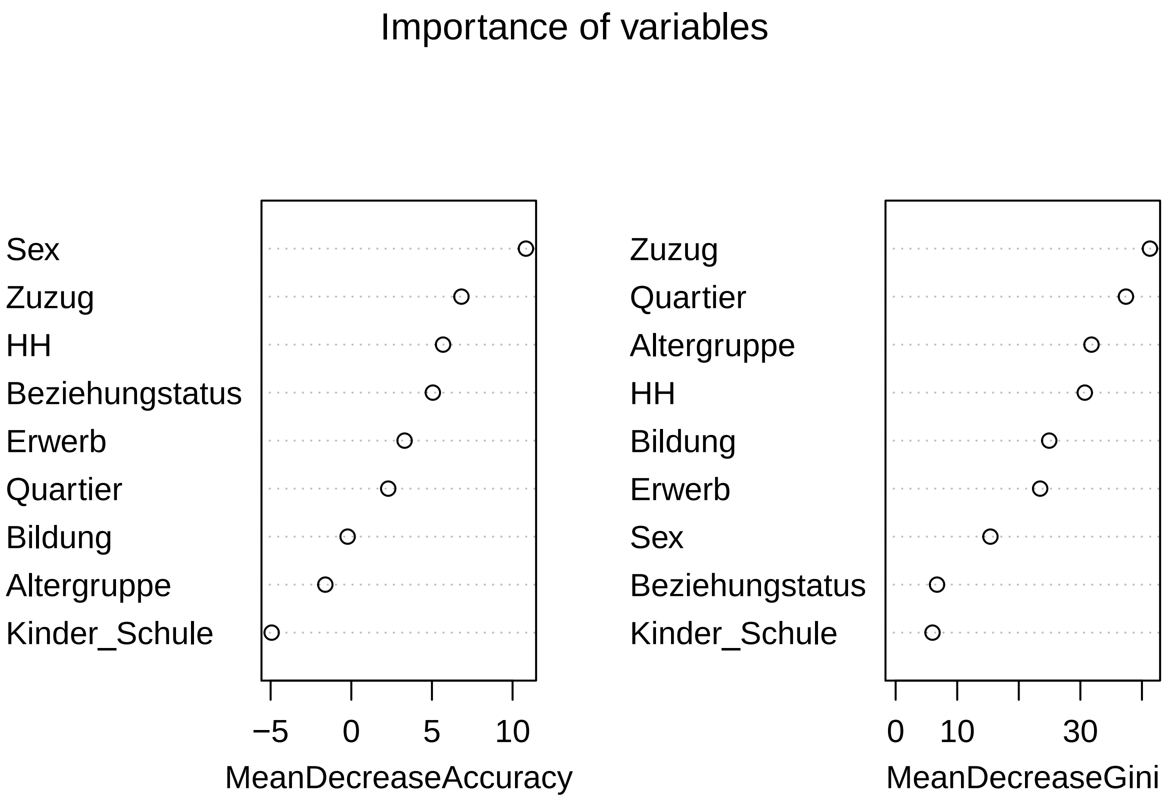
```r
plot(tree.classification.df)
text(tree.classification.df, pretty=1, cex=0.75)
```



```r
# bagging
bag.df=randomForest(Engagement ~ Sex + Altergruppe + Beziehungstatus +
                      Quartier + Zuzug + Bildung + HH + Kinder_Schule +
                      Erwerb, data = df2,
                    subset=train, mtry=8, importance =TRUE)

print(bag.df)
```

```
##
## Call:
##  randomForest(formula = Engagement ~ Sex + Altergruppe + Beziehungstatus +      Quartier + Zuzug + Bi
##                Type of random forest: classification
##                      Number of trees: 500
## No. of variables tried at each split: 8
##
##         OOB estimate of  error rate: 40.92%
## Confusion matrix:
##     0  1 class.error
## 0 194 85   0.3046595
## 1 111 89   0.5550000
```

```
# Investigating variable importance
# importance(bag.df)
varImpPlot (bag.df, "main"="Importance of variables")
```



Interpretation

With the summarised answers, the tree shows only 6 terminal nodes. The main criteria is again the factor *Zugzug*. By applying bagging, one can reduce the out-of-bag estimate of error rate to 40.1%.