Appendix with the Code - Classification model predicting High Value Properties

Damian Trzcinski

2022-06-05

```
library(MASS)
library(ROCR)
library(nnet)
library(st514)
library(readxl)
```

#A brief summary of the data (among other things: background information, attributes (which can be served as categorical and which can be potential classifiers)) etc.

Check data types, based on the data description and familiarizing myself with the dataset

- Numerical (they are all INT - dont change anything):

1. id_num, 2. price, 3. area, 4. bedrooms, 5. bathrooms, 7. garage, 9. year, 12. lot,

- Categorical:

6. aircon, 8. pool, 11. style, 13. highway

- Ordinal:

10. quality,

```
# loading the file, transformed from txt to csv in Excel
realestate = read.table("realestate.txt",header=FALSE)

# adding column names
colnames(realestate) = c("id_num", "price", "area", "bedrooms", "bathrooms", "aircon", "garrage", "pool

column_names = colnames(realestate)

for (i in 1:length(realestate)){
  print(paste(column_names[i],": ",typeof(realestate[,i])))
}
```

```
## [1] "id_num :  integer"
## [1] "price :  integer"
## [1] "area :  integer"
## [1] "bedrooms :  integer"
## [1] "bathrooms :  integer"
## [1] "aircon :  integer"
## [1] "garrage :  integer"
## [1] "pool :  integer"
## [1] "year :  integer"
## [1] "quality :  integer"
## [1] "style :  integer"
## [1] "lot :  integer"
## [1] "highway :  integer"
```

```
summary(realestate)
```

```
##      id_num          price             area         bedrooms
##  Min.   :  1.0   Min.   : 84000   Min.   : 980   Min.   :0.000
##  1st Qu.:131.2   1st Qu.:180000   1st Qu.:1701   1st Qu.:3.000
##  Median :261.5   Median :229900   Median :2061   Median :3.000
##  Mean   :261.5   Mean   :277894   Mean   :2261   Mean   :3.471
##  3rd Qu.:391.8   3rd Qu.:335000   3rd Qu.:2636   3rd Qu.:4.000
##  Max.   :522.0   Max.   :920000   Max.   :5032   Max.   :7.000
##    bathrooms         aircon          garrage          pool
##  Min.   :0.000   Min.   :0.0000   Min.   :0.0   Min.   :0.00000
##  1st Qu.:2.000   1st Qu.:1.0000   1st Qu.:2.0   1st Qu.:0.00000
##  Median :3.000   Median :1.0000   Median :2.0   Median :0.00000
##  Mean   :2.642   Mean   :0.8314   Mean   :2.1   Mean   :0.06897
##  3rd Qu.:3.000   3rd Qu.:1.0000   3rd Qu.:2.0   3rd Qu.:0.00000
##  Max.   :7.000   Max.   :1.0000   Max.   :7.0   Max.   :1.00000
##       year         quality          style             lot
##  Min.   :1885   Min.   :1.000   Min.   : 1.000   Min.   : 4560
##  1st Qu.:1956   1st Qu.:2.000   1st Qu.: 1.000   1st Qu.:17205
##  Median :1966   Median :2.000   Median : 2.000   Median :22200
##  Mean   :1967   Mean   :2.184   Mean   : 3.345   Mean   :24370
##  3rd Qu.:1981   3rd Qu.:3.000   3rd Qu.: 7.000   3rd Qu.:26787
##  Max.   :1998   Max.   :3.000   Max.   :11.000   Max.   :86830
##     highway
##  Min.   :0.00000
##  1st Qu.:0.00000
##  Median :0.00000
##  Mean   :0.02107
##  3rd Qu.:0.00000
##  Max.   :1.00000
```

```
# all data are now INT, we need to assign them the right datatypes

## CHANGING TYPES for categorical
### 6. aircon, 8. pool, 11. style, 13. highway

realestate$aircon = factor(realestate$aircon)
realestate$pool = factor(realestate$pool)
realestate$style = factor(realestate$style)
```

```r
realestate$highway = factor(realestate$highway)

## CHANGING TYPES for ordinal
### 10. quality

summary(realestate$quality)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   1.000   2.000   2.000   2.184   3.000   3.000
```

```r
# assigning ordinal type to quality variable and defining levels in the order 3, 2, 1, as 3 indicates t
realestate$quality = factor(realestate$quality, ordered = TRUE, levels = c(3,2,1))

summary(realestate$quality)
```

```
##   3   2   1
## 164 290  68
```

```r
# results are printed, ordered from the low quality score to high quality score (3,2,1)

# SUMMARY of all data after assigning the right data types to each column
summary(realestate)
```

```
##      id_num          price             area         bedrooms
##  Min.   :  1.0   Min.   : 84000   Min.   : 980   Min.   :0.000
##  1st Qu.:131.2   1st Qu.:180000   1st Qu.:1701   1st Qu.:3.000
##  Median :261.5   Median :229900   Median :2061   Median :3.000
##  Mean   :261.5   Mean   :277894   Mean   :2261   Mean   :3.471
##  3rd Qu.:391.8   3rd Qu.:335000   3rd Qu.:2636   3rd Qu.:4.000
##  Max.   :522.0   Max.   :920000   Max.   :5032   Max.   :7.000
##
##    bathrooms     aircon     garrage    pool         year      quality
##  Min.   :0.000   0: 88   Min.   :0.0   0:486   Min.   :1885   3:164
##  1st Qu.:2.000   1:434   1st Qu.:2.0   1: 36   1st Qu.:1956   2:290
##  Median :3.000           Median :2.0           Median :1966   1: 68
##  Mean   :2.642           Mean   :2.1           Mean   :1967
##  3rd Qu.:3.000           3rd Qu.:2.0           3rd Qu.:1981
##  Max.   :7.000           Max.   :7.0           Max.   :1998
##
##      style          lot         highway
##  1      :214   Min.   : 4560   0:511
##  7      :136   1st Qu.:17205   1: 11
##  3      : 64   Median :22200
##  2      : 58   Mean   :24370
##  5      : 18   3rd Qu.:26787
##  6      : 18   Max.   :86830
##  (Other): 14
```

Check if ID column present, it should not be considered for modeling

YES, column 1

Check for missing data

Based on the summary of the dataset above, there are no missing values in the data set. If there were any missing values in any of the columns, the function would give an information on how many "NA" values there are for a particular column.

What is the target value (Y - what we wanna predict)

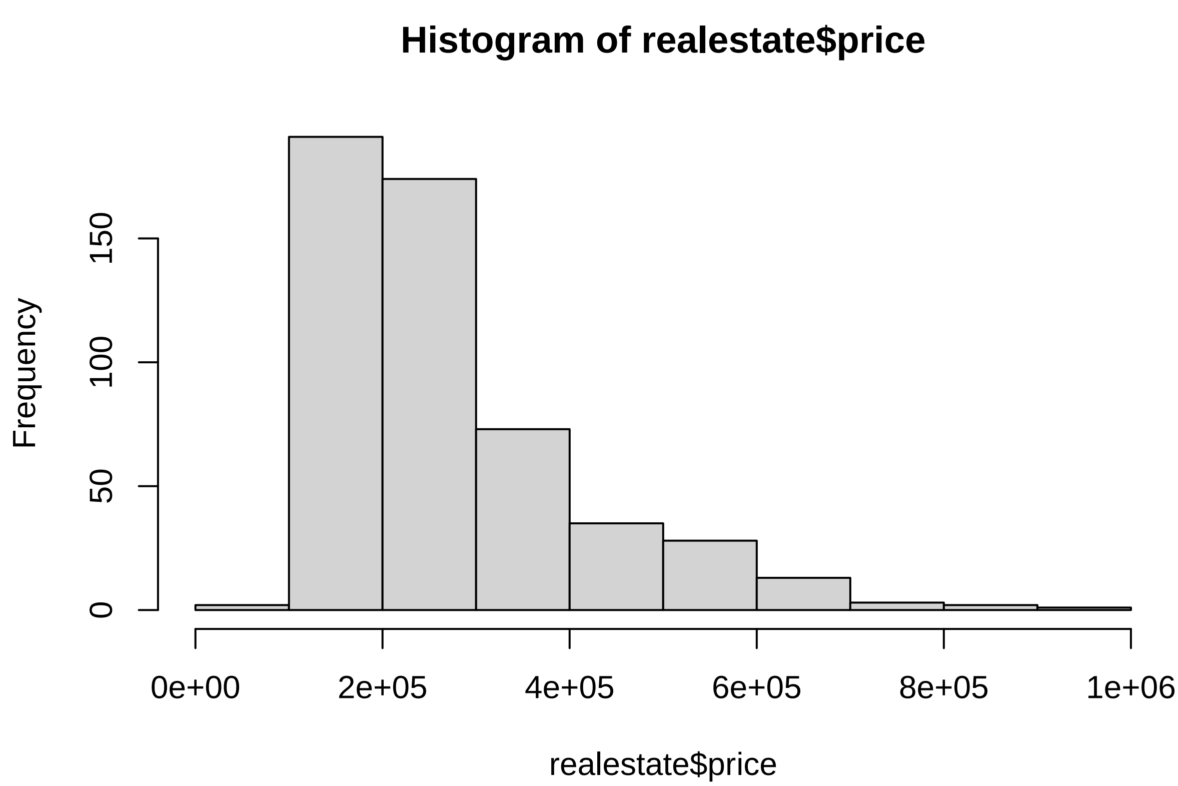
According to the description of the dataset attached to the Description of the Final project for DS805 , our target value should be the column 2 - Sales price of a house in US dollars. The remaining columns (apart from the id_num column) will be considered as potential predictors (11 in total). However, that would indicate that we will build a regression model.

I understand that the description of the Final Project asks for building a Classification model. Therefore, if I am to use Sales Price as a target value, I would need to transform it into a categorical variable

```
summary(realestate$price)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   84000  180000  229900  277894  335000  920000
```

```
hist(realestate$price)
```



Based on the histogram, mean and median of the variable Sales Price (price in the data frame), I have selected 70th percentile as a threshold for the transformation of the numerical variable into a categorical variable which is 300.000 USD. The new categorical variable "high_value" will indicate if a price of a real estate is above the given threshold, which we can interpret as a high value real estate.

```
# checking value for the 70th / 71tha percentile
quantile(realestate$price,0.7)
```

```
##    70%
## 299900
```

```
quantile(realestate$price,0.71)
```

```
##    71%
## 307000
```

```
high_value_thereshold = 300000

# our categorical variable high_value considers homes above 70th percentile, hence thereshold 300000

# adding the new variable high_value
realestate$high_value = factor(ifelse(realestate$price > high_value_thereshold,1,0))

# removing the numerical variable "price"
realestate = realestate[,-c(2)]

# summary
summary(realestate$high_value)
```

```
##   0   1
## 367 155
```

#Check necessary assumptions, e.g. normality, homogeneity of covariance matrices

```
# creating a copy of the same data set and applying jitter function to make the overlapping data points
realestate_pairs = realestate
realestate_pairs$high_value = jitter(as.numeric(realestate$high_value),0.5)

# plotting scatterplots of pairs of columns only for the numerical variables + categorical target varia
pdf("realestate_pairs_1.pdf")
pairs(realestate_pairs[,-c(1,2,3,4,5,6)])
#pairs(realestate[,-c(1,7,8,9,10,11,12)])
#pairs(realestate[,-c(1,5,7,9,10,12)])
dev.off()
```

```
## pdf
##   2
```

```
pdf("realestate_pairs_2.pdf")
#pairs(realestate_pairs[,-c(1,2,3,4,5,6)])
pairs(realestate_pairs[,-c(1,7,8,9,10,11,12)])
#pairs(realestate[,-c(1,5,7,9,10,12)])
dev.off()
```

```
## pdf
##   2
```

```
mvec <- colMeans(realestate[,-c(1,5,7,9,10,12,13)]) #sample mean vector#
mvec
```

```
##         area     bedrooms    bathrooms      garrage         year          lot
##  2260.626437     3.471264     2.641762     2.099617  1966.904215 24369.704981
```

```
covM <- cov(realestate[,-c(1,5,7,9,10,12,13)])      #sample covariance matrix#
corM <- cor(realestate[,-c(1,5,7,9,10,12,13)])  #sample correlation matrix#

##### plot association between variables high_value and area, as area seems to have the highest correla

pdf("./centrality_price_sale_realestate.pdf")
plot(realestate$area, as.numeric(as.character(realestate$high_value)), col='blue', lwd=2)
dev.off()
```

```
## pdf
##   2
```

As we have 11 variables in our data set, checking all the necesairly assumption of eg. normality and homogenity for all variables and then pairs of 2 variables etc would be time consuming. In order to avoid unnecessary computations, I will concentrate on the variables that are the most promising from the initial analysis of the scatter plots for the pairs of variables.

Based on that, I will focus on 4 variables: area, quality, highway and year

After conducing the initial analysis, I will verify if adding some of the eliminated variables from the initial analysis would improve the accuracy of predictions in the classification model.

```
# Assessing univariate normality for area  #

x <- sort(realestate$area)
q <- qnorm((1:522-0.5)/522)

pdf('qqplot_area.pdf')
plot(q,x,xlab="Standard normal quantile",ylab="Ordered data")
qqline(x)
title("(a)")
dev.off()
```

```
## pdf
##   2
```

```
cor_area <- cor(q,x)
```

According to the table 4.2 from the book (ref in report), for the population 300 the critical value is 0.9953 for significance level 0.95 (alpha = 0.05), so for the population size 522 it would be even higher.

The correlation between the theoretical quarantines and ordered values of variable *area* is lower than the critical value ($0.9568 < 0.9953$), therefore we reject the null hypothesis that the data are normally distributed.

```
# Assessing univariate normality for area  #

x <- sort(realestate$year)
q <- qnorm((1:522-0.5)/522)

pdf('qqplot_year.pdf')
plot(q,x,xlab="Standard normal quantile",ylab="Ordered data")
qqline(x)
title("(a)")
dev.off()
```

```
## pdf
##   2
```

```
cor_year <- cor(q,x)
```

Checking normality for multivariate normallity of year and area

```
##### find critical value for chi-square distribution ####

##set up the proper n and p, different datasets different n and p of course#

n0 <- 522

p0 <- 2

alpha1 <- 0.05  #upper quantile/significant level#


N0 <- 1000 #iteration 1000 times, you could increase it, in fact should do it for different N until inc



FindcrikChi <- function(n=n0, p=p0, alpha=alpha1, N=N0){

    cricvec <- rep(0, N)  #vector for the rQ result collection#

    for(i in 1:N){
        #iteration to estimate rQ#
        numvec <- rchisq(n, p)  #generate a data set of size n, degree of freedom=p#
        d <- sort(numvec)
        q <- qchisq((1:n-0.5)/n, p)
        cricvec[i] <- cor(d,q)
    }

    scricvec <- sort(cricvec)
    cN <- ceiling(N* alpha) #to be on the safe side I use ceiling instead of floor(), take the 'worst' 
    cricvalue <- scricvec[cN]
    result <- list(cN, cricvalue, scricvec)
    return(result)
}
```

```r
result1 <- FindcrikChi(n0,p0,alpha1,N0)

val1 <- result1[[1]]

val2 <- result1[[2]]
```

```r
#
# Evaluate bivariate normality: scatterplot with 0.25, 0.50 and 0.75 probability ellipse
#

bivar_norm_eval <- data.frame(area = realestate$area, year = realestate$year)
summary(bivar_norm_eval)
```

```
##       area           year
##  Min.   : 980   Min.   :1885
##  1st Qu.:1701   1st Qu.:1956
##  Median :2061   Median :1966
##  Mean   :2261   Mean   :1967
##  3rd Qu.:2636   3rd Qu.:1981
##  Max.   :5032   Max.   :1998
```

```r
m <- colMeans(bivar_norm_eval)

pdf("scatter_bivar_norm_eval.pdf")
    plot(bivar_norm_eval$area,bivar_norm_eval$year,xlab="Area",ylab="Year",xlim=c(900,5500),ylim=c(1880
    lines(c(m[1],m[1]),c(-10,m[2]),lty=2)  #dotted line for showing the center, i.e. sample mean vector
    lines(c(-10,m[1]),c(m[2],m[2]),lty=2) #dotted line for showing the center, i.e. sample mean vector#
    points(m[1],m[2],pch=4)
dev.off()
```

```
## pdf
##   2
```

```r
c <- cov(bivar_norm_eval)
cinv <- solve(c)  #the inverse matrix of covariance matrix#

##### prepare to draw the contours ###

x1 <- seq(900,5500,40)
x2 <- seq(1885,2000,1)

n <- length(x1)

f <- matrix(0,n,n)

for (i in 1:n){
    for (j in 1:n){

        xv <- c(x1[i],x2[j])

        f[i,j] <- t(xv-m)%*%cinv%*%(xv-m)  #quadrtic form#
```

```
    }
}

pdf("bivar_norm_eval_DataContour.pdf")
    chiq <- qchisq(seq(.25,.75,.25),2)
    contour(x1,x2,f,levels=chiq,xlab="Area",ylab="Year")
    points(bivar_norm_eval$area,bivar_norm_eval$year)
    lines(c(m[1],m[1]),c(-10,m[2]),lty=2)
    lines(c(-10,m[1]),c(m[2],m[2]),lty=2)
    points(m[1],m[2],pch=4)
dev.off()
```

```
## pdf
##   2
```

```
### comparison  ###

# #
# # Evaluate bivariate normality: count the number of points
# # in the 0.25, 0.50 and 0.75 probability ellipse
# #

d <- rep(0,522)
for (i in 1:522){
    xv <- t(bivar_norm_eval[i,])
    d[i] <- t(xv-m)%*%cinv%*%(xv-m)  ##squared mahalanobis distance, the quadratic form for this data s
}

count1 <- sum(d < qchisq(.25,2))
count2 <- sum(d < qchisq(.5,2))
count3 <- sum(d < qchisq(.75,2))

exp_count1 = round(0.25*522)
exp_count2 = round(0.5*522)
exp_count3 = round(0.75*522)

#EXP VS ACTUAL is different for exp_count 2 and 3 which could indicate the data is not normally distrib

#
# Evaluate bivariate normality: Q-Q plot of squared Mahalanobis distances,
# note that theoretical quantile should be chi-square
#

pdf("QQplotchisquare_bivar_norm_eval.pdf")
d <- sort(d)
q <- qchisq((1:522-0.5)/522,2)  #p=2 here as the 2nd parameter, but can be applicable for even higher d
plot(q,d,xlab="Chi-square quantiles",ylab="Ordered squared Mahalanobis distances")
abline(0,1)
dev.off()
```

```
## pdf
##   2
```

```r
rQcor <- cor(d,q)

# rQcor is lower than the critical value, calculated using FindChiCrik function 0.9446 < 0.9915
# we reject the H0 that the bivariate distribution is normal
```

Neither Univariate nor Bivariate distributions are normal, therefore I will now verify if Box Cox transformation will help to transform those distributions to normal distributions.

```r
# create function to loop over and get plot + correlation value

qq_function <- function(x, name_variable){
  qq_plot <- qqnorm(x, plot.it = F)
  my_cor <- cor(qq_plot$x, qq_plot$y) # step 4
  plot(qq_plot,
     main = paste0("Q-Q plot for ", name_variable),
     ylab = "Observed quantiles",
     xlab = "Theoretical quantiles") # plot the data
  legend('topleft', paste0("r = ", round(my_cor,4))) # add the correlation value to the chart
}

qq_norm_box_cox <- function(x, name){
  boxcoxTransc <- boxcox(x~1,
                         lambda=seq(-.5,1.5,.01),
                         plotit = F)
  flagidx <- which(boxcoxTransc$y==max(boxcoxTransc$y))
  # uses the index value of the max to get the corresponding value
  optlam <- boxcoxTransc$x[flagidx]
  transvec <- (x^optlam-1)/optlam
  qq_function(transvec, name)
}

pdf("QQplots_post_boxcox.pdf", width=10)
par(mfrow=c(1,2))
area_boxcox = qq_norm_box_cox(bivar_norm_eval$area, "area")
year_boxcox = qq_norm_box_cox(bivar_norm_eval$year, "year")
dev.off()
```

```
## pdf
##   2
```

The correlation value from the qq plot for the variable *area* has improved significantly from 0.9568 to 0.9934 but it's still below the critical value of 0.9953, indicating that the variable is not normally distributed, considering critical value for alpha 0.05. The correlation from QQ plot for variable *year* is negligible.

As Box Cox transformations for Univariate distributions did not help us to achieve normal distribution for neither of the variable, in the next step I am going to check if removing outliers for the the bivariate distribution of area and year helps to increase the correlation value hence bring us closer to the normal distribution of those variables.

```r
bivar_norm <- function(x1, x2, alpha, name, remove_outlier = FALSE) {
  df <- data.frame(x1,x2) # create dataframe
  n <- nrow(df) # obersvations
  p <- ncol(df) # number of variables
```

```
  D2 <- mahalanobis(df,
                    center  = colMeans(df),
                    cov = cov(df)) # generalized squared distance
  if(remove_outlier == TRUE){
    D2 <- D2[-which.max(D2)]
  }
  chi_plot <- qqplot(qchisq(ppoints(n, a = .5), df = p), D2,
                     plot.it = F) # chi square plot values.
                      # ppoints: j-1/2/n = 1:length(x)-1/2/length(x)
  my_cor <- cor(chi_plot$x, chi_plot$y) # correlation value
  critical_value <- qchisq(p = alpha,
                           df = p,
                           lower.tail = F) # calculate critical value
  prop_within_contour <- round(length(D2[D2 <= critical_value]) / length(D2),4)
  plot(chi_plot,
       ylab = 'Mahalanobis distances',
       xlab = 'Chi-square quantiles',
       main = paste0(name, ' alpha = ',alpha)) # plot chi square plot
  legend("topleft",
         paste0("r = ", round(my_cor,4), "\n",
                    "% D2 <= c^2: ", prop_within_contour, "\n",
                    "Expected if normal: ", 1-alpha),
         cex = 0.75,
         bty = "n") # add legend to plot
}


pdf("QQplots_comparison_bivariate_outliers.pdf", width=10)
par(mfrow=c(1,2))
bivar_norm(bivar_norm_eval$area,bivar_norm_eval$year, .05, "Area & Year", F)
bivar_norm(bivar_norm_eval$area,bivar_norm_eval$year, .05, "Area & Year (removed outlier)", T)
dev.off()
```

```
## pdf
##   2
```

```
###Test Homogeneous covariance matrices###

g <- 2 #we have groups above or below high value price thereshold
p <- 2 #two attributes, year, area

estate_group1 <- realestate[realestate$high_value==0, c(2,8)]

estate_group2 <- realestate[realestate$high_value==1, c(2,8)]

s1 <- cov(estate_group1)
s2 <- cov(estate_group2)

n1 <- nrow(estate_group1)
n2 <- nrow(estate_group2)
n <- n1+n2

w <- (n1-1)*s1+(n2-1)*s2  #Within matrix#
```

```r
spooled <- w/(n-g)

#
# Compute M (6-50)
#

M <- (n-g)*log(det(spooled))-(n1-1)*log(det(s1))-(n2-1)*log(det(s2))

#
# Compute correction factor (6-51)
#

u <- (1/(n1-1)+1/(n2-1)-1/(n-g))*(2*p^2+3*p-1)/(6*(p+1)*(g-1))

#
# Test statistic
#

C <- (1-u)*M

#
# critical value
#

critvalue <- qchisq(.95,p*(p+1)*(g-1)/2)   #v=p*(p+1)*(g-1)/2#


### final decision ####

decisionflag <- (C > critvalue)   #TRUE, therefore we should REJECT the H0, which means these are not h
```

The covariances are not homogenous for the numerical variables area and year.

Selection of optimal classification rule

Logistic Regression is an appropriate classification alghoritm, as we do not normally distributed variables

```r
n <- nrow(realestate)

# fit logistic regression model

result <- glm(high_value~area+year+quality+highway,realestate,family=binomial(link="logit"))
# fitted values are predictions for the full data set

summary(result)
```

```
##
## Call:
## glm(formula = high_value ~ area + year + quality + highway, family = binomial(link = "logit"),
##     data = realestate)
##
```

```
## Deviance Residuals:
##      Min        1Q    Median        3Q       Max
## -1.69961  -0.26360  -0.10419   0.00001   2.95872
##
## Coefficients:
##               Estimate Std. Error z value Pr(>|z|)
## (Intercept) -1.018e+02  3.233e+02  -0.315 0.752952
## area         4.458e-03  5.195e-04   8.582  < 2e-16 ***
## year         4.875e-02  1.276e-02   3.820 0.000133 ***
## quality.L    1.335e+01  6.837e+02   0.020 0.984420
## quality.Q    7.611e+00  3.947e+02   0.019 0.984617
## highway1     7.877e-01  1.153e+00   0.683 0.494519
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 635.01  on 521  degrees of freedom
## Residual deviance: 203.54  on 516  degrees of freedom
## AIC: 215.54
##
## Number of Fisher Scoring iterations: 18
```

```r
# only 2 variables seem to be statistically significant (area and year)

p <- result$fitted.values
# returns vector of probabilities if an observation belongs to class 1

y <- realestate$high_value
# actual classes of observations

# compute the apparent error rate

#setting up a thereshold for assigning fitted observations to 0/1 classes
pr <- as.numeric( p >= .5)

# table of misclassifications
table(y,pr)
```

```
##    pr
## y     0   1
##   0 349  18
##   1  27 128
```

```r
pred <- prediction(p,y)
# preparing data for ROCR package to use it

perf <- performance(pred,"tpr","fpr")
# based on the data of predictions and actual data (labels), we check how ration between True Positive
# False Positive Rate is changing, when we change a threshold used for the classification

pdf("Realestate_LogiRoc.pdf")
    plot(perf,colorize=F,lwd=3)
dev.off()
```

```
## pdf
##   2
```

```r
# compute the area under the curve


AUC_realestate <- performance(pred, measure="auc")
AUC_realestate_1 <- AUC_realestate@y.values #0.9679

#area under curve - probability when we present the model with 2 obs, one with class 1 and one with cla

# cross-validation

prcv <- rep(0,n)
for (i in 1:n){
    realestate1 <- realestate[-i,]
    res1 <- glm(high_value~area+year+quality+highway,realestate1,family=binomial(link="logit"))
    xc <- realestate[i,c(2,8,9,12)]
    prcv[i] = predict(res1,xc,type="response")
    #lp <- predict(res1,xc)
    #prcv[i] <- exp(lp)/(1+exp(lp))
    #prcv2[i] = predict(res1,xc,type="response")
}

# odds ratio for a particular observation is ratio of a probability that's is positive (class 1) divide

# compute the CV error rate

pr <- as.numeric( prcv >= .5)
table(y,pr)
```

```
##    pr
## y     0   1
##   0 349  18
##   1  29 126
```

```r
#
# Plot the CV ROC curve
#

pred1 <- prediction(prcv,y)
perf1 <- performance(pred1,"tpr","fpr")
pdf("realestate_LogiROCcv.pdf")
    plot(perf,colorize=T,lwd=3)
    plot(perf1,colorize=F,lwd=3, add=T)
dev.off()
```

```
## pdf
##   2
```

```r
# compute the area under the curve
```

```r
AUC_realestate_CV <- performance(pred1, measure="auc")
AUC_realestate_CV1 <- AUC_realestate_CV@y.values

#looks like the coloful one is better 0.9679 > 0,9580, with a bigger area under the curve#
```

An additional classification rule for further comparison

```r
x = realestate$area
boxcoxTransc <- boxcox(x~1,
                       lambda=seq(-.5,1.5,.01),
                       plotit = F)
flagidx <- which(boxcoxTransc$y==max(boxcoxTransc$y))
# uses the index value of the max to get the corresponding value
optlam <- boxcoxTransc$x[flagidx]
area_transformed_boxcox <- (x^optlam-1)/optlam

realestate_boxcox = realestate
realestate_boxcox$area = area_transformed_boxcox

n_2 <- nrow(realestate_boxcox)

# fit logistic regression model

result_2 <- glm(high_value~area+year,realestate_boxcox,family=binomial(link="logit"))
# fitted values are predictions for the full data set

summary(result_2)
```

```
##
## Call:
## glm(formula = high_value ~ area + year, family = binomial(link = "logit"),
##     data = realestate_boxcox)
##
## Deviance Residuals:
##      Min        1Q    Median        3Q       Max
## -1.89011  -0.25390  -0.06717   0.16216   3.01914
##
## Coefficients:
##               Estimate Std. Error z value Pr(>|z|)
## (Intercept) -1.217e+03  1.196e+02 -10.172  < 2e-16 ***
## area         5.656e+02  5.843e+01   9.679  < 2e-16 ***
## year         5.475e-02  1.085e-02   5.045 4.54e-07 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 635.01  on 521  degrees of freedom
## Residual deviance: 229.43  on 519  degrees of freedom
## AIC: 235.43
```

```
##
## Number of Fisher Scoring iterations: 7
```

```
# only 2 variables seem to be statistically significant (area and year)

p_2 <- result_2$fitted.values
# returns vector of probabilities if an observation belongs to class 1

y_2 <- realestate_boxcox$high_value
# actual classes of observations

# compute the apparent error rate

#setting up a thereshold for assigning fitted observations to 0/1 classes
pr_2 <- as.numeric( p_2 >= .5)

# table of misclassifications
table(y_2,pr_2)
```

```
##    pr_2
## y_2   0   1
##   0 343  24
##   1  26 129
```

```
pred_2 <- prediction(p_2,y_2)
# preparing data for ROCR package to use it

perf_2 <- performance(pred_2,"tpr","fpr")
# based on the data of predictions and actual data (labels), we check how ration between True Positive 
# False Positive Rate is changing, when we change a threshold used for the classification

pdf("Realestate_LogiRoc_mod2.pdf")
    plot(perf,colorize=F,lwd=3)
dev.off()
```

```
## pdf
##   2
```

```
#
# compute the area under the curve
#


AUC_realestate_mod2 <- performance(pred_2, measure="auc")
AUC_realestate_1_mod2 <- AUC_realestate_mod2@y.values #0.9630

#area under curve - probability when we present the model with 2 obs, one with class 1 and one with cla

# cross-validation

prcv_2 <- rep(0,n_2)
for (i in 1:n_2){
```

```r
    realestate1_boxcox_2 <- realestate_boxcox[-i,]
    res1 <- glm(high_value~area+year,realestate1_boxcox_2,family=binomial(link="logit"))
    xc <- realestate_boxcox[i,c(2,8)]
    prcv_2[i] = predict(res1,xc,type="response")
    #lp <- predict(res1,xc)
    #prcv[i] <- exp(lp)/(1+exp(lp))
    #prcv2[i] = predict(res1,xc,type="response")
}

# odds ratio for a particular observation is ratio of a probability that's is positive (class 1) divide
# that something is negative (class 0)

# compute the CV error rate

pr_2 <- as.numeric( prcv_2 >= .5)
table(y_2,pr_2)
```

```
##    pr_2
## y_2   0   1
##   0 343  24
##   1  26 129
```

```r
#
# Plot the CV ROC curve
#

pred1_mod2 <- prediction(prcv_2,y_2)
perf1_mod2 <- performance(pred1_mod2,"tpr","fpr")
pdf("realestate_LogiROCcv_mod2.pdf")
    plot(perf,colorize=T,lwd=3)
    plot(perf1,colorize=F,lwd=3, add=T)
dev.off()
```

```
## pdf
##   2
```

```r
# compute the area under the curve

AUC_realestate_CV_mod2 <- performance(pred1_mod2, measure="auc")
AUC_realestate_CV1_mod2 <- AUC_realestate_CV_mod2@y.values

#looks like the coloful one is better 0.9630 > 0,9610, with a bigger area under the curve#
```

Conclusion: BoxCox transformation of variable *area* did not improve the accuracy of the model, it requires data transformation that makes the interpretation of the coefficients of the model more difficult.

3rd model - additional model with fitting all variables

Fitting the model with ALL variables in the DF realestate. It is possible as Logistic Regression doesn't rely on the normality assumption, hence I do not need to check it for all numerical variables to use the model

I eliminate from the original DF column 1 - ID - as it is not a variable but an identification number for the observations. I will also eliminate variable "style" as there are some styles that occur only once, therefore it causes troubles when doing cross validation of the fitted model (alternatively we could also remove problematic observations)

```r
# fit logistic regression model
realestate_no_id = realestate[,c(2,3,4,5,6,7,8,9,11,12,13)]

result <- glm(high_value~.,realestate_no_id,family=binomial(link="logit"))
# fitted values are predictions for the full data set

n_3 <- nrow(realestate_no_id)

summary(result)
```

```
##
## Call:
## glm(formula = high_value ~ ., family = binomial(link = "logit"),
##     data = realestate_no_id)
##
## Deviance Residuals:
##      Min        1Q    Median        3Q       Max
## -2.25809  -0.21873  -0.05707   0.00002   2.88831
##
## Coefficients:
##               Estimate Std. Error z value Pr(>|z|)
## (Intercept) -1.474e+02  3.356e+02  -0.439   0.6604
## area         4.555e-03  6.707e-04   6.791 1.11e-11 ***
## bedrooms     8.213e-03  2.533e-01   0.032   0.9741
## bathrooms    4.373e-02  3.366e-01   0.130   0.8966
## aircon1      1.433e+00  8.467e-01   1.692   0.0906 .
## garrage      3.847e-01  3.638e-01   1.057   0.2903
## pool1        4.575e-01  7.218e-01   0.634   0.5261
## year         6.935e-02  1.711e-02   4.053 5.05e-05 ***
## quality.L    1.227e+01  7.082e+02   0.017   0.9862
## quality.Q    7.063e+00  4.089e+02   0.017   0.9862
## lot          7.404e-05  1.707e-05   4.337 1.44e-05 ***
## highway1     1.471e-01  1.311e+00   0.112   0.9107
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 635.01  on 521  degrees of freedom
## Residual deviance: 177.12  on 510  degrees of freedom
## AIC: 201.12
##
## Number of Fisher Scoring iterations: 18
```

```r
# only 2 variables seem to be statistically significant (area and year)

p_3 <- result$fitted.values
# returns vector of probabilities if an observation belongs to class 1
```

```r
y_3 <- realestate$high_value
# actual classes of observations

# compute the apparent error rate

#setting up a thereshold for assigning fitted observations to 0/1 classes
pr_3 <- as.numeric( p_3 >= .5)

# table of misclassifications
table(y_3,pr_3)
```

```
##    pr_3
## y_3   0   1
##   0 352  15
##   1  23 132
```

```r
pred_3 <- prediction(p_3,y_3)
# preparing data for ROCR package to use it

perf_3 <- performance(pred_3,"tpr","fpr")
# based on the data of predictions and actual data (labels), we check how ration between True Positive 
# False Positive Rate is changing, when we change a threshold used for the classification

pdf("Realestate_LogiRoc_mod3.pdf")
    plot(perf_3,colorize=F,lwd=3)
dev.off()
```

```
## pdf
##   2
```

```r
# compute the area under the curve

AUC_realestate_mod3 <- performance(pred_3, measure="auc")
AUC_realestate_1_mod3 <- AUC_realestate_mod3@y.values #0.9679

#area under curve - probability when we present the model with 2 obs, one with class 1 and one with cla

# cross-validation

prcv_3 <- rep(0,n_3)
for (i in 1:n_3){
    realestate_no_id1 <- realestate_no_id[-i,]
    res1 <- glm(high_value~.,realestate_no_id1,family=binomial(link="logit"))
    xc <- realestate_no_id[i,1:10]
    prcv_3[i] = predict(res1,xc,type="response")
    #lp <- predict(res1,xc)
    #prcv[i] <- exp(lp)/(1+exp(lp))
    #prcv2[i] = predict(res1,xc,type="response")
}

# odds ratio for a particular observation is ratio of a probability that's is positive (class 1) divide
```

```r
# compute the CV error rate

pr_3 <- as.numeric( prcv_3 >= .5)
table(y_3,pr_3)
```

```
##    pr_3
## y_3   0   1
##   0 351  16
##   1  25 130
```

```r
#
# Plot the CV ROC curve
#

pred1_mod3 <- prediction(prcv_3,y_3)
perf1_mod3 <- performance(pred1_mod3,"tpr","fpr")
pdf("realestate_LogiROCcv_mod3.pdf")
    plot(perf_3,colorize=T,lwd=3)
    plot(perf1_mod3,colorize=F,lwd=3, add=T)
dev.off()
```

```
## pdf
##   2
```

```r
# compute the area under the curve

AUC_realestate_CV_mod3 <- performance(pred1_mod3, measure="auc")
AUC_realestate_CV1_mod3 <- AUC_realestate_CV_mod3@y.values

#looks like the coloful one is better 0.9679 > 0,9580, with a bigger area under the curve#
```

Selection of classifier

Based on the proposed 3 fitted models it seems like the first model would be prefered, due to its high accuracy value and relatively simple interepration due to low number of variables. However, during the process of fitting the model of 4 variables we have discovered that the 2 variables *quality* and *highway* were not statistically significant in contrary to high statistical significance of the variables *area* and *year*.

I would suggest proceeding with the model consisting of only 2 statistically significant variables *area* and *year*

```r
n_4 <- nrow(realestate)

# fit logistic regression model

result <- glm(high_value~area+year,realestate,family=binomial(link="logit"))
# fitted values are predictions for the full data set

summary(result)
```

```
##
## Call:
## glm(formula = high_value ~ area + year, family = binomial(link = "logit"),
##     data = realestate)
##
## Deviance Residuals:
##      Min        1Q    Median        3Q       Max
## -1.88684  -0.27856  -0.10513   0.08219   2.93265
##
## Coefficients:
##               Estimate Std. Error z value Pr(>|z|)
## (Intercept) -1.269e+02  2.216e+01  -5.725 1.04e-08 ***
## area         4.649e-03  4.721e-04   9.848  < 2e-16 ***
## year         5.822e-02  1.112e-02   5.237 1.63e-07 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 635.01  on 521  degrees of freedom
## Residual deviance: 230.09  on 519  degrees of freedom
## AIC: 236.09
##
## Number of Fisher Scoring iterations: 7
```

```
# only 2 variables seem to be statistically significant (area and year)

p_4 <- result$fitted.values
# returns vector of probabilities if an observation belongs to class 1

y_4 <- realestate$high_value
# actual classes of observations

# compute the apparent error rate

#setting up a thereshold for assigning fitted observations to 0/1 classes
pr_4 <- as.numeric( p_4 >= .5)

# table of misclassifications
table(y_4,pr_4)
```

```
##    pr_4
## y_4   0   1
##   0 347  20
##   1  27 128
```

```
pred_4 <- prediction(p_4,y_4)
# preparing data for ROCR package to use it

perf_4 <- performance(pred_4,"tpr","fpr")
# based on the data of predictions and actual data (labels), we check how ration between True Positive
# False Positive Rate is changing, when we change a threshold used for the classification
```

```
pdf("Realestate_LogiRoc_mod4.pdf")
    plot(perf_4,colorize=F,lwd=3)
dev.off()
```

```
## pdf
##   2
```

```
#
# compute the area under the curve
#


AUC_realestate_mod4 <- performance(pred_4, measure="auc")
AUC_realestate_1_mod4 <- AUC_realestate_mod4@y.values #0.9679 with 4 variables, 0.9610 with 2 variables

#area under curve - probability when we present the model with 2 obs, one with class 1 and one with cla
```