Projektbericht

Ferdinand Bubeck und Johannes Bubeck

19.01.2021

## Data Exploration

#import libraries  
library(readr)  
library(dplyr)  
library(ggplot2)  
library(tidyverse)  
  
#import CSVs  
datTest <- read\_csv("aug\_test.csv")  
  
datTrain <- read\_csv("aug\_train.csv")  
  
#join of Train & Test Data to have full Dataset for Exploration  
dat <- full\_join(datTest, datTrain)   
  
#first Impressions of Data  
head(dat)

## # A tibble: 6 x 14  
## enrollee\_id city city\_developmen~ gender relevent\_experi~ enrolled\_univer~  
## <dbl> <chr> <dbl> <chr> <chr> <chr>   
## 1 32403 city~ 0.827 Male Has relevent ex~ Full time course  
## 2 9858 city~ 0.92 Female Has relevent ex~ no\_enrollment   
## 3 31806 city~ 0.624 Male No relevent exp~ no\_enrollment   
## 4 27385 city~ 0.827 Male Has relevent ex~ no\_enrollment   
## 5 27724 city~ 0.92 Male Has relevent ex~ no\_enrollment   
## 6 217 city~ 0.899 Male No relevent exp~ Part time course  
## # ... with 8 more variables: education\_level <chr>, major\_discipline <chr>,  
## # experience <chr>, company\_size <chr>, company\_type <chr>,  
## # last\_new\_job <chr>, training\_hours <dbl>, target <dbl>

tail(dat)

## # A tibble: 6 x 14  
## enrollee\_id city city\_developmen~ gender relevent\_experi~ enrolled\_univer~  
## <dbl> <chr> <dbl> <chr> <chr> <chr>   
## 1 29754 city~ 0.92 Female Has relevent ex~ no\_enrollment   
## 2 7386 city~ 0.878 Male No relevent exp~ no\_enrollment   
## 3 31398 city~ 0.92 Male Has relevent ex~ no\_enrollment   
## 4 24576 city~ 0.92 Male Has relevent ex~ no\_enrollment   
## 5 5756 city~ 0.802 Male Has relevent ex~ no\_enrollment   
## 6 23834 city~ 0.855 <NA> No relevent exp~ no\_enrollment   
## # ... with 8 more variables: education\_level <chr>, major\_discipline <chr>,  
## # experience <chr>, company\_size <chr>, company\_type <chr>,  
## # last\_new\_job <chr>, training\_hours <dbl>, target <dbl>

summary(dat)

## enrollee\_id city city\_development\_index gender   
## Min. : 1 Length:21287 Min. :0.4480 Length:21287   
## 1st Qu.: 8554 Class :character 1st Qu.:0.7390 Class :character   
## Median :16967 Mode :character Median :0.9030 Mode :character   
## Mean :16874 Mean :0.8285   
## 3rd Qu.:25162 3rd Qu.:0.9200   
## Max. :33380 Max. :0.9490   
##   
## relevent\_experience enrolled\_university education\_level major\_discipline   
## Length:21287 Length:21287 Length:21287 Length:21287   
## Class :character Class :character Class :character Class :character   
## Mode :character Mode :character Mode :character Mode :character   
##   
##   
##   
##   
## experience company\_size company\_type last\_new\_job   
## Length:21287 Length:21287 Length:21287 Length:21287   
## Class :character Class :character Class :character Class :character   
## Mode :character Mode :character Mode :character Mode :character   
##   
##   
##   
##   
## training\_hours target   
## Min. : 1.00 Min. :0.0000   
## 1st Qu.: 23.00 1st Qu.:0.0000   
## Median : 47.00 Median :0.0000   
## Mean : 65.33 Mean :0.2493   
## 3rd Qu.: 88.00 3rd Qu.:0.0000   
## Max. :336.00 Max. :1.0000   
## NA's :2129

dim(dat)

## [1] 21287 14

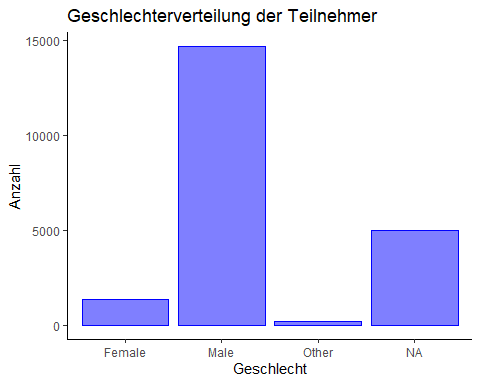
str(dat)

## tibble [21,287 x 14] (S3: spec\_tbl\_df/tbl\_df/tbl/data.frame)  
## $ enrollee\_id : num [1:21287] 32403 9858 31806 27385 27724 ...  
## $ city : chr [1:21287] "city\_41" "city\_103" "city\_21" "city\_13" ...  
## $ city\_development\_index: num [1:21287] 0.827 0.92 0.624 0.827 0.92 0.899 0.624 0.92 0.878 0.624 ...  
## $ gender : chr [1:21287] "Male" "Female" "Male" "Male" ...  
## $ relevent\_experience : chr [1:21287] "Has relevent experience" "Has relevent experience" "No relevent experience" "Has relevent experience" ...  
## $ enrolled\_university : chr [1:21287] "Full time course" "no\_enrollment" "no\_enrollment" "no\_enrollment" ...  
## $ education\_level : chr [1:21287] "Graduate" "Graduate" "High School" "Masters" ...  
## $ major\_discipline : chr [1:21287] "STEM" "STEM" NA "STEM" ...  
## $ experience : chr [1:21287] "9" "5" "<1" "11" ...  
## $ company\_size : chr [1:21287] "<10" NA NA "10/49" ...  
## $ company\_type : chr [1:21287] NA "Pvt Ltd" "Pvt Ltd" "Pvt Ltd" ...  
## $ last\_new\_job : chr [1:21287] "1" "1" "never" "1" ...  
## $ training\_hours : num [1:21287] 21 98 15 39 72 12 11 81 2 4 ...  
## $ target : num [1:21287] NA NA NA NA NA NA NA NA NA NA ...

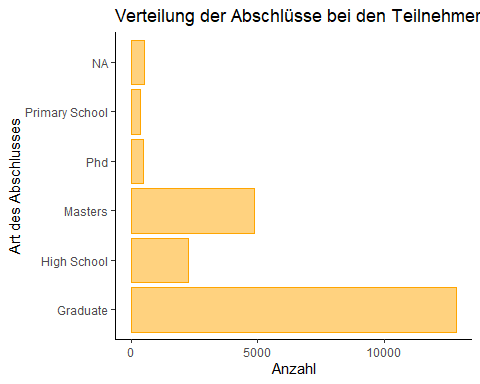
#count NAs in column gender  
sum(is.na(dat$gender))

## [1] 5016

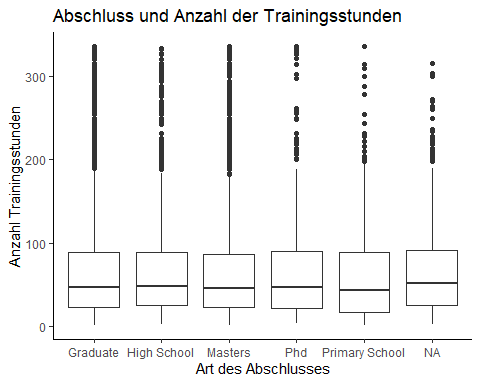
#Plot zu Geschlechterverteilung  
ggplot(data=dat, aes(x=gender))+  
 geom\_bar(color='blue', fill='blue', alpha=.5)+  
 theme\_classic()+  
 labs(title="Geschlechterverteilung der Teilnehmer")+  
 xlab("Geschlecht")+  
 ylab("Anzahl")



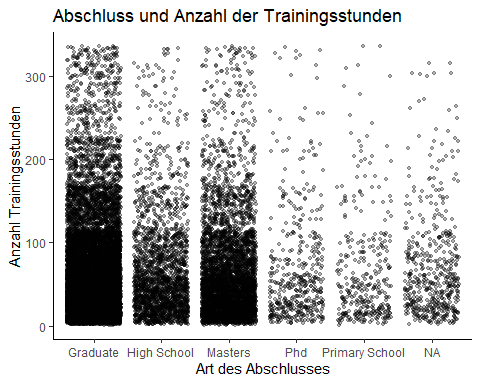
#Plot der Abschlüsse  
ggplot(data=dat, aes(x=education\_level))+  
 geom\_bar(color='orange', fill='orange', alpha=.5)+  
 coord\_flip()+  
 theme\_classic()+  
 labs(title="Verteilung der Abschlüsse bei den Teilnehmer")+  
 xlab("Art des Abschlusses")+  
 ylab("Anzahl")



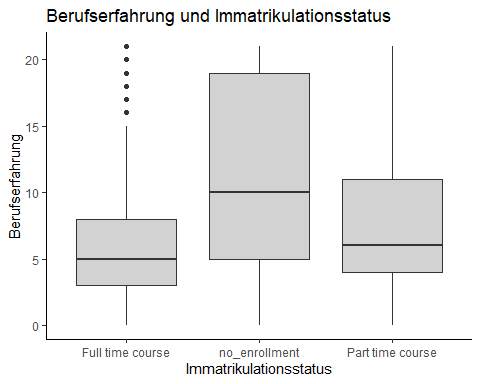
#Plot Anzahl der Trainngsstunden und Abschluss (Boxplot)  
ggplot(data=dat, aes(x=education\_level, y=training\_hours))+  
 geom\_boxplot()+  
 theme\_classic()+  
 labs(title="Abschluss und Anzahl der Trainingsstunden")+  
 xlab("Art des Abschlusses")+  
 ylab("Anzahl Trainingsstunden")



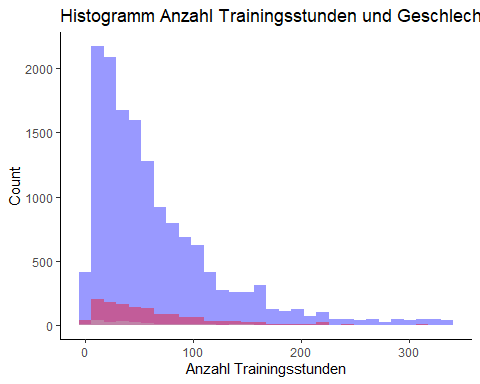
#Plot Anzahl der Trainngsstunden und Abschluss (Point/Jitter)  
ggplot(data=dat, aes(x=education\_level, y=training\_hours))+  
 geom\_jitter(color='black', alpha=.3, size=.9)+  
 theme\_classic()+  
 labs(title="Abschluss und Anzahl der Trainingsstunden")+  
 xlab("Art des Abschlusses")+  
 ylab("Anzahl Trainingsstunden")



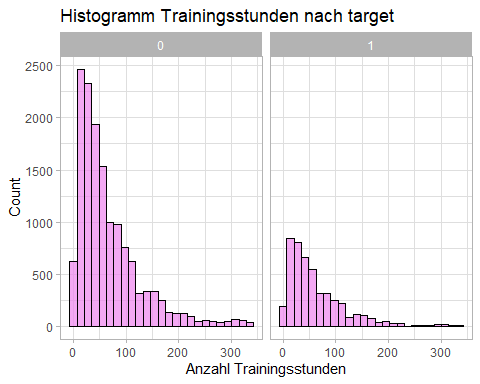
#ersetze < und > durch feste numerische Werte um typecast durchzuführen  
dat$experience <- sub("<1", "0", dat$experience)  
dat$experience <- sub(">20", "21", dat$experience)  
dat$experience <- as.numeric(dat$experience)  
  
dat %>%   
 select(enrolled\_university, experience) %>%  
 filter(!is.na(enrolled\_university),!is.na(experience)) %>%  
 ggplot(data=., aes(x=enrolled\_university, y=experience))+  
 geom\_boxplot(fill="grey", alpha=.7)+  
 theme\_classic()+  
 labs(title="Berufserfahrung und Immatrikulationsstatus")+  
 xlab("Immatrikulationsstatus")+  
 ylab("Berufserfahrung")



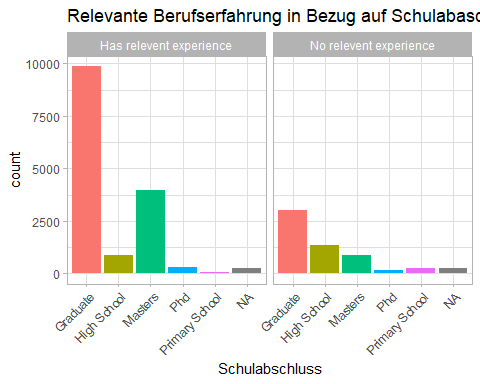
###  
ggplot()+  
 geom\_histogram(data=subset(dat, gender == "Male"), aes(x=training\_hours), fill="blue", alpha=.4)+  
 geom\_histogram(data=subset(dat, gender == "Female"), aes(x=training\_hours), fill="red", alpha=.4)+  
 geom\_histogram(data=subset(dat, gender == "Other"), aes(x=training\_hours), fill="grey", alpha=.4)+  
 theme\_classic()+  
 labs(title="Histogramm Anzahl Trainingsstunden und Geschlecht")+  
 xlab("Anzahl Trainingsstunden")+  
 ylab("Count")



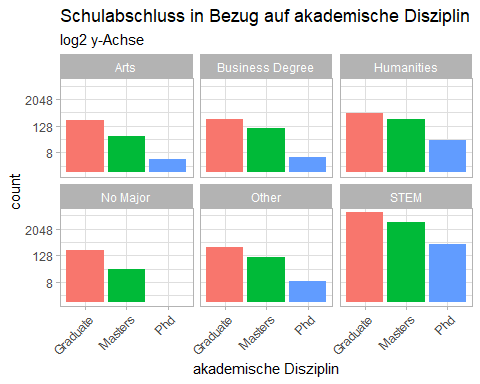
dat %>%  
 select(training\_hours, target) %>%  
 filter(!is.na(training\_hours),!is.na(target)) %>%  
 ggplot(data=., aes(x = training\_hours)) +  
 geom\_histogram(bins = 25, color = "black", fill = "violet", alpha=.7) +  
 theme\_light() +  
 facet\_wrap(vars(target))+  
 labs(title="Histogramm Trainingsstunden nach target")+  
 xlab("Anzahl Trainingsstunden")+  
 ylab("Count")



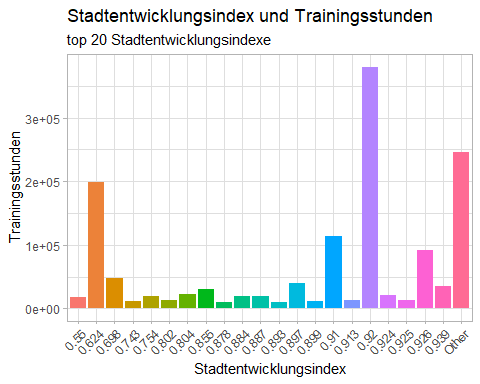
ggplot(data=dat, aes(x=as.factor(education\_level)))+  
 geom\_bar(data=dat, aes(fill=as.factor(education\_level)))+  
 facet\_wrap(vars(relevent\_experience))+  
 theme\_light()+  
 theme(legend.position = "none")+  
 theme(axis.text.x = element\_text(angle = 45, hjust = 1))+  
 labs(title="Relevante Berufserfahrung in Bezug auf Schulabaschluss")+  
 xlab("Schulabschluss")



dat %>%   
 select(major\_discipline, education\_level) %>%  
 filter(!is.na(major\_discipline),!is.na(education\_level)) %>%  
 ggplot(data=.,aes(x=education\_level))+  
 geom\_bar(aes(fill=education\_level))+  
 facet\_wrap(vars(major\_discipline))+  
 theme\_light()+  
 theme(legend.position = "none")+  
 theme(axis.text.x = element\_text(angle = 45, hjust = 1))+  
 labs(title="Schulabschluss in Bezug auf akademische Disziplin", subtitle = "log2 y-Achse")+  
 xlab("akademische Disziplin")+  
 scale\_y\_continuous(trans = 'log2')



dat %>%   
 select(city\_development\_index, training\_hours)%>%  
 mutate(city\_development\_index = fct\_lump(as.factor(city\_development\_index), n = 20)) %>%  
 filter(!is.na(city\_development\_index)) %>%  
 ggplot(data= ., aes(x=city\_development\_index, y=training\_hours, fill = city\_development\_index))+  
 geom\_col()+  
 theme\_light()+  
 theme(axis.text.x = element\_text(angle = 45, hjust = 1))+  
 theme(legend.position = "none")+  
 labs(title="Stadtentwicklungsindex und Trainingsstunden", subtitle = "top 20 Stadtentwicklungsindexe")+  
 xlab("Stadtentwicklungsindex")+  
 ylab("Trainingsstunden")



library(caret)  
  
#Dealing with NAs  
#detect variables with more than 5% (threshold) of NAs  
detectNA <- function(x){sum(is.na(x))/length(x)\*100}  
apply(datTrain, 2, detectNA)

## enrollee\_id city city\_development\_index   
## 0.0000000 0.0000000 0.0000000   
## gender relevent\_experience enrolled\_university   
## 23.5306399 0.0000000 2.0148241   
## education\_level major\_discipline experience   
## 2.4010857 14.6831611 0.3392839   
## company\_size company\_type last\_new\_job   
## 30.9948846 32.0492745 2.2079549   
## training\_hours target   
## 0.0000000 0.0000000

#Data Preparation  
datTrain\_casted <- datTrain %>%   
 select(-enrollee\_id, -city) %>%  
 mutate(city\_development\_index = as.numeric(city\_development\_index),  
 target = as.factor(target)) %>%   
 as.data.frame() %>%  
 mutate\_if(is.character, as.factor) #converting all variables with type=char to factor  
  
   
  
#Data Imputation  
library(mice)  
  
impute <- mice(datTrain\_casted, m = 1, remove.constant = FALSE) #takes time :)

##   
## iter imp variable  
## 1 1 gender enrolled\_university education\_level major\_discipline experience company\_size company\_type last\_new\_job  
## 2 1 gender enrolled\_university education\_level major\_discipline experience company\_size company\_type last\_new\_job  
## 3 1 gender enrolled\_university education\_level major\_discipline experience company\_size company\_type last\_new\_job  
## 4 1 gender enrolled\_university education\_level major\_discipline experience company\_size company\_type last\_new\_job  
## 5 1 gender enrolled\_university education\_level major\_discipline experience company\_size company\_type last\_new\_job

datTrain\_imputed <- complete(impute)  
  
apply(datTrain\_imputed, 2, detectNA)

## city\_development\_index gender relevent\_experience   
## 0 0 0   
## enrolled\_university education\_level major\_discipline   
## 0 0 0   
## experience company\_size company\_type   
## 0 0 0   
## last\_new\_job training\_hours target   
## 0 0 0

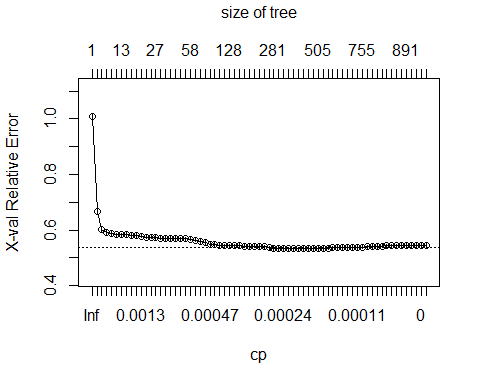
#Data Balancing  
library(groupdata2)  
  
target.balancing <- datTrain\_imputed %>%   
 group\_by(target) %>%  
 summarise(no\_rows = length(target))  
  
datTrain\_balanced <- upsample(datTrain\_imputed, cat\_col = "target")  
  
datTrain\_balanced %>%   
 group\_by(target) %>%  
 summarise(no\_rows = length(target))

## # A tibble: 2 x 2  
## target no\_rows  
## <fct> <int>  
## 1 0 14381  
## 2 1 14381

#Test / Training Split  
set.seed(3024)  
trainingRows <- sort(sample(nrow(datTrain\_balanced), nrow(datTrain\_balanced)\*.7))  
  
Train <- datTrain\_balanced[trainingRows,]  
Test <- datTrain\_balanced[-trainingRows,]  
  
###Decision tree  
library(rpart)  
library(rpart.plot)  
  
##1st tree  
set.seed(34)  
dt.default <- rpart(target~ ., data=Train, cp=-1)  
  
printcp(dt.default)

##   
## Classification tree:  
## rpart(formula = target ~ ., data = Train, cp = -1)  
##   
## Variables actually used in tree construction:  
## [1] city\_development\_index company\_size company\_type   
## [4] education\_level enrolled\_university experience   
## [7] gender last\_new\_job major\_discipline   
## [10] relevent\_experience training\_hours   
##   
## Root node error: 10036/20133 = 0.49849  
##   
## n= 20133   
##   
## CP nsplit rel error xerror xstd  
## 1 3.3191e-01 0 1.00000 1.00967 0.0070689  
## 2 3.3778e-02 1 0.66809 0.66809 0.0066633  
## 3 9.2666e-03 3 0.60054 0.60054 0.0064750  
## 4 3.0391e-03 4 0.59127 0.59197 0.0064482  
## 5 2.1921e-03 8 0.57732 0.58808 0.0064358  
## 6 2.0925e-03 9 0.57513 0.58290 0.0064191  
## 7 1.9430e-03 12 0.56815 0.58290 0.0064191  
## 8 1.5943e-03 15 0.56198 0.58370 0.0064217  
## 9 1.4946e-03 17 0.55879 0.58131 0.0064139  
## 10 1.4448e-03 19 0.55580 0.58051 0.0064113  
## 11 1.1957e-03 21 0.55291 0.57822 0.0064038  
## 12 1.0961e-03 22 0.55171 0.57513 0.0063935  
## 13 1.0462e-03 24 0.54952 0.57493 0.0063929  
## 14 9.9641e-04 26 0.54743 0.57413 0.0063902  
## 15 8.9677e-04 30 0.54344 0.57174 0.0063822  
## 16 8.8016e-04 37 0.53577 0.57084 0.0063792  
## 17 8.4695e-04 45 0.52760 0.56925 0.0063738  
## 18 7.9713e-04 47 0.52591 0.56935 0.0063742  
## 19 7.7222e-04 49 0.52431 0.56885 0.0063725  
## 20 6.9749e-04 54 0.51963 0.56895 0.0063728  
## 21 6.4767e-04 57 0.51754 0.56736 0.0063674  
## 22 5.9785e-04 59 0.51624 0.56387 0.0063555  
## 23 5.4803e-04 67 0.51146 0.55929 0.0063397  
## 24 4.9821e-04 77 0.50578 0.55580 0.0063275  
## 25 4.4839e-04 95 0.49611 0.54912 0.0063038  
## 26 4.3842e-04 108 0.49004 0.54853 0.0063017  
## 27 4.3178e-04 113 0.48784 0.54454 0.0062873  
## 28 4.2348e-04 121 0.48426 0.54324 0.0062826  
## 29 4.1849e-04 127 0.48117 0.54354 0.0062837  
## 30 4.1280e-04 134 0.47798 0.54384 0.0062848  
## 31 3.9857e-04 157 0.46353 0.54434 0.0062866  
## 32 3.4874e-04 179 0.45476 0.54235 0.0062794  
## 33 3.3878e-04 189 0.45118 0.54185 0.0062775  
## 34 3.3214e-04 195 0.44878 0.54235 0.0062794  
## 35 3.2739e-04 198 0.44779 0.54085 0.0062739  
## 36 2.9892e-04 210 0.44360 0.53996 0.0062706  
## 37 2.6571e-04 248 0.43165 0.53657 0.0062581  
## 38 2.4910e-04 280 0.42168 0.53398 0.0062485  
## 39 2.4527e-04 331 0.40743 0.53388 0.0062481  
## 40 2.4199e-04 346 0.40335 0.53388 0.0062481  
## 41 2.3250e-04 378 0.38840 0.53278 0.0062440  
## 42 2.1921e-04 400 0.38133 0.53278 0.0062440  
## 43 2.1589e-04 409 0.37904 0.53288 0.0062444  
## 44 1.9928e-04 420 0.37635 0.53378 0.0062478  
## 45 1.7714e-04 485 0.36220 0.53378 0.0062478  
## 46 1.7437e-04 494 0.36060 0.53408 0.0062489  
## 47 1.6607e-04 504 0.35821 0.53438 0.0062500  
## 48 1.4946e-04 523 0.35472 0.53448 0.0062504  
## 49 1.4234e-04 598 0.34028 0.53547 0.0062541  
## 50 1.3950e-04 605 0.33928 0.53607 0.0062563  
## 51 1.3286e-04 626 0.33579 0.53727 0.0062607  
## 52 1.2455e-04 644 0.33340 0.53687 0.0062592  
## 53 1.2178e-04 663 0.33061 0.53697 0.0062596  
## 54 1.1388e-04 672 0.32951 0.53697 0.0062596  
## 55 9.9641e-05 679 0.32872 0.53816 0.0062640  
## 56 7.4731e-05 754 0.32084 0.53926 0.0062681  
## 57 7.1172e-05 769 0.31965 0.54026 0.0062717  
## 58 6.6428e-05 776 0.31915 0.54085 0.0062739  
## 59 6.2276e-05 797 0.31776 0.54115 0.0062750  
## 60 5.6938e-05 805 0.31726 0.54215 0.0062786  
## 61 4.9821e-05 812 0.31686 0.54324 0.0062826  
## 62 4.2703e-05 861 0.31437 0.54364 0.0062841  
## 63 3.9857e-05 868 0.31407 0.54374 0.0062844  
## 64 3.3214e-05 878 0.31327 0.54484 0.0062884  
## 65 2.4910e-05 890 0.31287 0.54494 0.0062888  
## 66 1.6607e-05 902 0.31257 0.54424 0.0062862  
## 67 1.4234e-05 908 0.31248 0.54414 0.0062859  
## 68 0.0000e+00 915 0.31238 0.54454 0.0062873  
## 69 -1.0000e+00 1402 0.31238 0.54454 0.0062873

plotcp(dt.default)



#Prediction & Performance  
#Train Data  
train.default.pred <- predict(dt.default, newdata = Train, type = "class")  
train.default.confMatrix <- table(train.default.pred, Train[, 12])  
print(train.default.confMatrix)

##   
## train.default.pred 0 1  
## 0 8459 1497  
## 1 1638 8539

train.default.accuracy <- sum(diag(train.default.confMatrix))/sum(train.default.confMatrix)  
print(train.default.accuracy)

## [1] 0.8442855

#Test Data  
test.default.pred <- predict(dt.default, newdata = Test, type = "class")  
test.default.confMatrix <- table(test.default.pred, Test[, 12])  
print(test.default.confMatrix)

##   
## test.default.pred 0 1  
## 0 3072 1085  
## 1 1212 3260

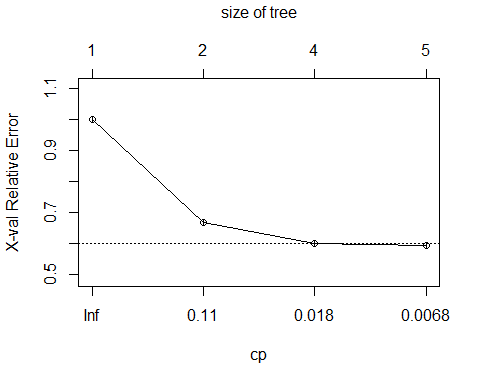
test.default.accuracy <- sum(diag(test.default.confMatrix))/sum(test.default.confMatrix)  
print(test.default.accuracy)

## [1] 0.7338046

##2nd tree with cp limitation  
set.seed(39)  
dt.fit <- rpart(target~ ., data=Train, cp=0.005)  
  
#summary(dt.fit)  
  
printcp(dt.fit)

##   
## Classification tree:  
## rpart(formula = target ~ ., data = Train, cp = 0.005)  
##   
## Variables actually used in tree construction:  
## [1] city\_development\_index education\_level relevent\_experience   
##   
## Root node error: 10036/20133 = 0.49849  
##   
## n= 20133   
##   
## CP nsplit rel error xerror xstd  
## 1 0.3319051 0 1.00000 1.00000 0.0070691  
## 2 0.0337784 1 0.66809 0.66809 0.0066633  
## 3 0.0092666 3 0.60054 0.60054 0.0064750  
## 4 0.0050000 4 0.59127 0.59346 0.0064529

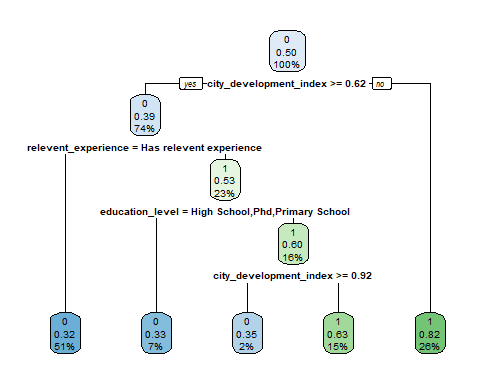
plotcp(dt.fit)



#search cp with lowest Cross-Validation Error  
xerror.min <- dt.fit$cptable[which.min(dt.fit$cptable[,4]),]  
cp.best <- xerror.min[1]  
cp.best

## CP   
## 0.005

#Pruning with best cp  
dt.pruned <- prune(dt.fit, cp=cp.best)  
  
#summary(dt.pruned)  
  
rpart.plot(dt.pruned)



#Prediction & Performance  
#Train Data  
train.pred <- predict(dt.pruned, newdata = Train, type = "class")  
train.confMatrix <- table(train.pred, Train[, 12])  
print(train.confMatrix)

##   
## train.pred 0 1  
## 0 8036 3873  
## 1 2061 6163

train.accuracy <- sum(diag(train.confMatrix))/sum(train.confMatrix)  
print(train.accuracy)

## [1] 0.70526

#Test Data  
test.pred <- predict(dt.pruned, newdata = Test, type = "class")  
test.confMatrix <- table(test.pred, Test[, 12])  
print(test.confMatrix)

##   
## test.pred 0 1  
## 0 3351 1676  
## 1 933 2669

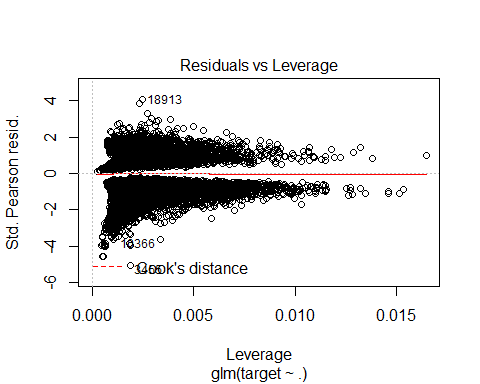
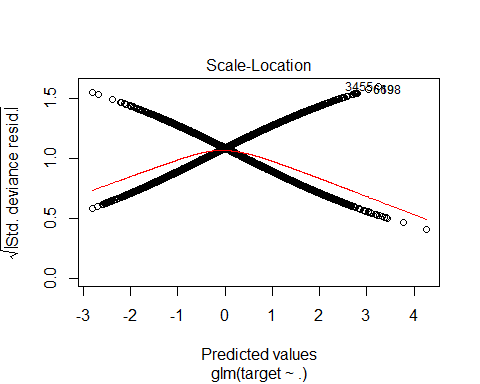
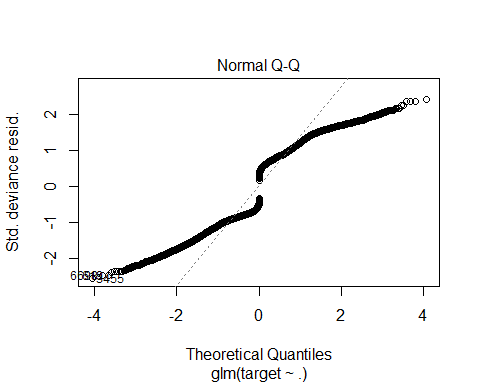
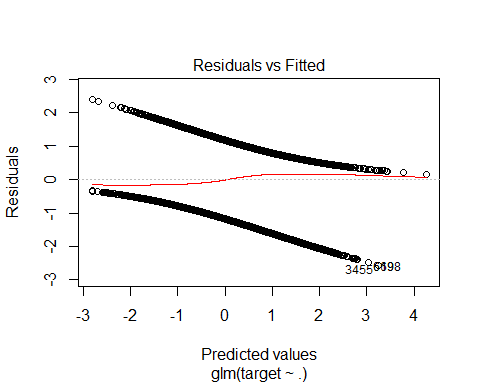
test.accuracy <- sum(diag(test.confMatrix))/sum(test.confMatrix)  
print(test.accuracy)

## [1] 0.6976475

###glm  
#dummy variable  
  
dummy\_target <- as.numeric(Train[12] == 1)  
dat.dummy\_target <- cbind(Train,dummy\_target)  
  
set.seed(4)  
glm.fit <- glm(target ~ ., family = binomial, Train)  
#summary(glm.fit)  
  
varImp(glm.fit, scale = FALSE)

## Overall  
## city\_development\_index 43.366402223  
## genderMale 3.823392089  
## genderOther 1.551394783  
## relevent\_experienceNo relevent experience 13.792498794  
## enrolled\_universityno\_enrollment 8.967853887  
## enrolled\_universityPart time course 6.040801741  
## education\_levelHigh School 14.221896600  
## education\_levelMasters 7.585863508  
## education\_levelPhd 5.879687712  
## education\_levelPrimary School 9.431778384  
## major\_disciplineBusiness Degree 1.805668231  
## major\_disciplineHumanities 0.727087875  
## major\_disciplineNo Major 0.214236322  
## major\_disciplineOther 1.432192650  
## major\_disciplineSTEM 0.670894360  
## experience>20 4.586478115  
## experience1 1.012757706  
## experience10 5.391019675  
## experience11 2.519500448  
## experience12 5.425618988  
## experience13 4.823916065  
## experience14 5.357207805  
## experience15 4.797839909  
## experience16 6.251196892  
## experience17 4.803179189  
## experience18 4.216986971  
## experience19 2.626760470  
## experience2 4.968574183  
## experience20 2.396137507  
## experience3 3.023633594  
## experience4 3.802033007  
## experience5 4.820513513  
## experience6 4.524707197  
## experience7 3.598315287  
## experience8 5.021753210  
## experience9 5.492451682  
## company\_size10/49 4.803078683  
## company\_size100-500 0.025729777  
## company\_size1000-4999 0.902234413  
## company\_size10000+ 5.885234874  
## company\_size50-99 1.100557695  
## company\_size500-999 0.932290007  
## company\_size5000-9999 2.253687292  
## company\_typeFunded Startup 2.347536203  
## company\_typeNGO 0.648084681  
## company\_typeOther 4.134865834  
## company\_typePublic Sector 6.191622355  
## company\_typePvt Ltd 0.004261375  
## last\_new\_job1 2.001095417  
## last\_new\_job2 0.100660043  
## last\_new\_job3 0.025309804  
## last\_new\_job4 0.705020246  
## last\_new\_jobnever 4.927382695  
## training\_hours 3.629262783

glm.fit.predicted <- predict(   
 object = glm.fit,   
 data = Test,   
 type = "response"  
)  
  
#glm.fit.predicted  
plot(glm.fit)



plot(  
 x = dat.dummy\_target$city\_development\_index,   
 y = dat.dummy\_target$dummy\_target,   
 col = "red"  
)  
lines(Train$city\_development\_index, glm.fit.predicted, col="blue")

