```
############################################################
#ALY 6040 Data Mining Applications
#Final Project R markdown code
#Default Risk Detection
#Minyi Chen
#July 3, 2018
############################################################


#open dataset train.csv, and save it in R as train
train <- read.csv(file.choose())
# read the first 2 rows
head(train,2)
str(train)

### I found that 1) target is not factor, and 2) some variables I don't need anymore,
### We can see the 3) days_birth, and days_employed are negative,
### which means the days happen in the past.
### For example, if the days is -15750, that means the person was born for 15750 days
### before he submited the application
### ALSO need to change the birth days and employed days to year

# Change target to factor variable
train$TARGET<- as.factor(train$TARGET)
# delete three columns
# remove the columns and store variable into another df.
df = subset(train, select = -c(AMT_REQ_CREDIT_BUREAU_HOUR, NAME_TYPE_SUITE, OWN_CAR_AGE) )
#dplyr is a powerful R-package to transform and summarize tabular data with rows and columns.
#install.packages("dplyr")
library(dplyr)
# use mutate to create new columns that change the days to years for days_birth, and days_employed
df1 = df %>% mutate(age = DAYS_BIRTH / (-365), work_years= DAYS_EMPLOYED/ (-365) ) %>% subset(select = -c(DAYS_BIRTH, DAYS_EMPLOYED) )
head(df1,2)
str(df1)
summary(df1)


### finally, I get the dataset i want
### Next, I use expss package to label all the variables



################################      table visulization      ########################

#table(df1$NAME_EDUCATION_TYPE)



#install.packages("expss")
library(expss)

df1 = apply_labels(df1,
TARGET = "default risk",
NAME_CONTRACT_TYPE= "contract",

CODE_GENDER= "sex",
FLAG_OWN_CAR = "owncar",
FLAG_OWN_REALTY = "ownhouse",

CNT_CHILDREN = "children#",

AMT_INCOME_TOTAL = "income",
AMT_CREDIT = "loan credit",
AMT_ANNUITY = "loan annuity",
AMT_GOODS_PRICE = "goods price",

NAME_INCOME_TYPE = "income type",
NAME_EDUCATION_TYPE = "education",
NAME_FAMILY_STATUS = "family status",
NAME_HOUSING_TYPE ="housing",

REGION_POPULATION_RELATIVE ="region population index",
OCCUPATION_TYPE="occupation",
CNT_FAM_MEMBERS="family member",
ORGANIZATION_TYPE="organization",
age="age",
work_years="working years",
AMT_REQ_CREDIT_BUREAU_QRT="enquiries/qrt",
AMT_REQ_CREDIT_BUREAU_YEAR="enquiries/yr"
)

# crosstabulation, similar to base R 'table' function, but this one includes labels

# there are 10 categorial independent variables, so, I draw 10 tables for them to see
# their relationship with target variale
#cro(df1$NAME_CONTRACT_TYPE, df1$TARGET)
#cro(df1$CODE_GENDER, df1$TARGET)
#cro(df1$FLAG_OWN_CAR, df1$TARGET)
#cro(df1$FLAG_OWN_REALTY, df1$TARGET)

#cro(df1$NAME_INCOME_TYPE, df1$TARGET)
#cro(df1$NAME_EDUCATION_TYPE, df1$TARGET)
#cro(df1$NAME_FAMILY_STATUS, df1$TARGET)
#cro(df1$NAME_HOUSING_TYP, df1$TARGET)

#cro(df1$OCCUPATION_TYPE, df1$TARGET)
#cro(df1$ORGANIZATION_TYPE, df1$TARGET)

#from the graphs we can see
# default risk across different types of organization/occupation/housing/family status
# default risk also across both type of contracts, both genders, own cars or not, own house or not,
# For income type and education type, we can see that all businessmen and students pay back their debts
# maternity leave and unemployed has high default risk

# for education type, people who get academic degree has much lower default risk.
# This makes sense that people with academic degree might find better jobs to get higher pay and be able to pay back money
```

```
# also they might younger that they don't have much responsibilites on family
# This lead to see the graphs about education type vs. age/childen number


# build a contingency table of the row percent.
cro_rpct(df1$NAME_CONTRACT_TYPE, df1$TARGET)
cro_rpct(df1$CODE_GENDER, df1$TARGET)
cro_rpct(df1$FLAG_OWN_CAR, df1$TARGET)
cro_rpct(df1$FLAG_OWN_REALTY, df1$TARGET)

cro_rpct(df1$NAME_INCOME_TYPE, df1$TARGET)
cro_rpct(df1$NAME_EDUCATION_TYPE, df1$TARGET)
cro_rpct(df1$NAME_FAMILY_STATUS, df1$TARGET)
cro_rpct(df8$NAME_HOUSING_TYP, df8$TARGET)

cro_rpct(df8$OCCUPATION_TYPE, df8$TARGET)
cro_rpct(df8$ORGANIZATION_TYPE, df8$TARGET)



#####################################  graph visulization  ############################

# 11 continous independent variables
hist(df1$CNT_CHILDREN, main=" children number")
cro(df2$CNT_CHILDREN, df2$TARGET)


# Check outliers, remove them
#For a given continuous variable, outliers are those observations
# that lie outside 1.5*IQR, where IQR, the 'Inter Quartile Range' is
#the difference between 75th and 25th quartiles. Look at the points outside
#the whiskers in below box plot.


#The boxplot.stats function; is a ancillary function that produces statistics
#for drawing boxplots. It returns among other information a vector stats with
#five elements: the extreme of the lower whisker, the lower 'hinge', the median,
#the upper 'hinge' and the extreme of the upper whisker, the extreme of the whiskers
#are the adjacent values (last non-missing value, i.e. every value beyond is an outlier.

boxplot(df1$CNT_CHILDREN, main="children number", boxwex=0.1)
boxplot.stats(df1$CNT_CHILDREN)

#delete outliers rows of children number
df2 <-df1[df1$CNT_CHILDREN <5,]
cro(df2$CNT_CHILDREN, df2$TARGET)
boxplot(df2$CNT_CHILDREN, main="children number", boxwex=0.1)
filter(df2, CNT_CHILDREN >3 )
hist(df2$CNT_CHILDREN, main=" children number")



#income
hist(df2$AMT_INCOME_TOTAL, main="income ")

boxplot.stats(df2$AMT_INCOME_TOTAL) # get the upper whisker value
df3 <- df2[df2$AMT_INCOME_TOTAL <337501,]
boxplot(df3$AMT_INCOME_TOTAL, main="Income", boxwex=0.1)
filter(df2, AMT_INCOME_TOTAL >9000000)
hist(df3$AMT_INCOME_TOTAL, main="income")

ggplot(df3, aes(x = TARGET, y = AMT_INCOME_TOTAL)) + geom_boxplot()

boxplot(AMT_INCOME_TOTAL ~ TARGET, data=df3, xlab="Target", ylab="income")



#loan credit
hist(df3$AMT_CREDIT, main=" loan credit")

boxplot.stats(df3$AMT_CREDIT) # get the upper whisker value
df4 <- df3[df3$AMT_CREDIT <1490000,]
boxplot(df4$AMT_CREDIT, main="loan credit", boxwex=0.1)
hist(df4$AMT_CREDIT, main="loan credit")
boxplot(AMT_CREDIT ~ TARGET, data=df4, xlab="Target", ylab="loan credit")




hist(df4$AMT_ANNUITY, main=" loan annuity")
boxplot.stats(df4$AMT_ANNUITY) # get the upper whisker value
df5 <- df4[df4$AMT_ANNUITY <56772,]
boxplot(df5$AMT_ANNUITY, main="loan annuity", boxwex=0.1)
hist(df5$AMT_ANNUITY, main="loan annuity")
boxplot(AMT_ANNUITY ~ TARGET, data=df5, xlab="Target", ylab="loan annuity")



hist(df5$AMT_GOODS_PRICE, main=" goods price")

boxplot.stats(df5$AMT_GOODS_PRICE) # get the upper whisker value
df6 <- df5[df5$AMT_GOODS_PRICE <1350001,]
boxplot(df6$AMT_GOODS_PRICE, main="goods price", boxwex=0.1)
hist(df6$AMT_GOODS_PRICE, main="goods price")
boxplot(AMT_GOODS_PRICE ~ TARGET, data=df6, xlab="Target", ylab="goods price")


filter(df6, AMT_GOODS_PRICE >1200000)

# By specifying a single variable, qplot() will by default make a histogram.
# Here we make a histogram if the age data and stratify on the education type.
# So technically this is three histograms overlayed on top of each other.
qplot(AMT_GOODS_PRICE, data = df6, fill = TARGET, binwidth = 50000, xlab="goods price") +labs(fill = "Target")
```

```
hist(df6$REGION_POPULATION_RELATIVE, main=" region population index")

boxplot.stats(df6$REGION_POPULATION_RELATIVE) # get the upper whisker value
df7 <- df6[df6$REGION_POPULATION_RELATIVE <0.05,]
boxplot(df7$REGION_POPULATION_RELATIVE, main="region population index", boxwex=0.1)
hist(df7$REGION_POPULATION_RELATIVE, main="region population index")
boxplot(REGION_POPULATION_RELATIVE ~ TARGET, data=df7, xlab="Target", ylab="region population index")
qplot(REGION_POPULATION_RELATIVE, data = df7, fill = TARGET, binwidth = 0.01, xlab="region population index") +labs(fill = "Target")




hist(df7$CNT_FAM_MEMBERS, main="family member ")
summary(df7$CNT_FAM_MEMBERS)
cro_rpct(df7$CNT_FAM_MEMBERS, df7$TARGET)



hist(df7$age, main="age" )



hist(df7$work_years, main="working years")
df8 <- df7[df7$work_years >0,]
hist(df8$work_years, main="working years")
boxplot.stats(df8$work_years)
boxplot(df8$work_years, main="working years")


hist(df8$AMT_REQ_CREDIT_BUREAU_QRT, main="enquiries/qrt")
filter(df8, AMT_REQ_CREDIT_BUREAU_QRT >5)
qplot(AMT_REQ_CREDIT_BUREAU_QRT, data = df8, fill = TARGET, binwidth = 10, xlab="enquiries to Credit Bureau/qrt") +labs(fill = "Target")
filter(df8, AMT_REQ_CREDIT_BUREAU_QRT >12)
cro(df8$AMT_REQ_CREDIT_BUREAU_QRT, df8$TARGET)

hist(df8$AMT_REQ_CREDIT_BUREAU_YEAR, main="enquiries to Credit Bureau/yr")
cro(df8$AMT_REQ_CREDIT_BUREAU_YEAR, df8$TARGET)






# 10 categorial independent variables
pie(table(df8$NAME_CONTRACT_TYPE), main="contract")
pie(table(df8$CODE_GENDER),main="sex")
pie(table(df8$FLAG_OWN_CAR),main="owncar")
pie(table(df8$FLAG_OWN_REALTY),main="ownhouse")

barplot(table(df8$NAME_INCOME_TYPE),main="incometype")

# Histogram on a Categorical variable
g <- ggplot(df8, aes(NAME_INCOME_TYPE))
g + geom_bar(aes(fill=TARGET), width = 0.5) +
  theme(axis.text.x = element_text(angle=65, vjust=0.6)) +
  labs(title="Histogram on Income Type",
       subtitle="Target across Income Type", x = "Income type")


barplot(table(df8$NAME_EDUCATION_TYPE), main="education")
g <- ggplot(df8, aes(NAME_EDUCATION_TYPE))
g + geom_bar(aes(fill=TARGET), width = 0.5) +
  theme(axis.text.x = element_text(angle=65, vjust=0.6)) +
  labs(title="Histogram on Education Type",
       subtitle="Target across Education Type", x = "Education type")



pie(table(df8$NAME_FAMILY_STATUS), main="familystatus")

cro_rpct(df8$NAME_FAMILY_STATUS, df8$TARGET)

cro_rpct(df8$NAME_HOUSING_TYP, df8$TARGET)

cro_rpct(df8$OCCUPATION_TYPE, df8$TARGET)


################################################################################################
write.csv(df9, "C:/Users/minyi/Desktop/FINAL/df9.csv")

df0 =df9[df9$TARGET == '0',]
df0sample = sample_n(df0, 16674)
summary(df9$TARGET)
df1 = df9[df9$TARGET == '1',]
df10 = rbind(df0sample,df1)
write.csv(df10, "C:/Users/minyi/Desktop/FINAL/df10.csv")


set.seed(1230) #random number generator

ind = sample(2, nrow(df10), replace=TRUE, prob=c(0.7, 0.3))

df10.train = df9[ind==1,] #the training data set

df10.test = df9[ind==2,] #the test data set
summary(df10$TARGET)
summary(df10.train$TARGET)
summary(df10.test$TARGET)


zero =df10.train[df10.train$TARGET == '0',]
zerosample = sample_n(zero, 19923)
zero1 = df10.train[df10.train$TARGET == '1',]
df11.train=rbind(zerosample,zero1)
```

```
summary(df11.train$TARGET)


zero2 =df10.test[df10.test$TARGET == '0',]
zerosample2 = sample_n(zero2, 6259)
zero12 = df10.test[df10.test$TARGET == '1',]
df11.test=rbind(zerosample2,zero12)
summary(df11.test$TARGET)




############################### Logistic Rregression ################################
#Logistic regression for the training dataset
logitmodel <- glm(TARGET ~ NAME_CONTRACT_TYPE + CODE_GENDER + FLAG_OWN_CAR +
                    CNT_CHILDREN+ AMT_INCOME_TOTAL +AMT_CREDIT+AMT_ANNUITY+AMT_GOODS_PRICE
                    +NAME_EDUCATION_TYPE+NAME_HOUSING_TYPE+REGION_POPULATION_RELATIVE+age+work_years, data = df11.train, family = "binomial")

summary(logitmodel)




################################     decision tree    ################################
library(rpart)
install.packages("rpart.plot")
library(rpart.plot)


df9 = na.omit(df8) #delete the observations with missing values

#rpart.control(minsplit = 20, minbucket = round(minsplit/3), cp = 0.01,
#              maxcompete = 4, maxsurrogate = 5, usesurrogate = 2, xval = 10,
#              surrogatestyle = 0, maxdepth = 30, ...)

# build a decision tree use rpart function
tree1 <- rpart(TARGET ~ NAME_CONTRACT_TYPE + CODE_GENDER + FLAG_OWN_CAR +
                 CNT_CHILDREN+ AMT_INCOME_TOTAL +AMT_CREDIT+AMT_ANNUITY+AMT_GOODS_PRICE
               +NAME_EDUCATION_TYPE+NAME_HOUSING_TYPE+REGION_POPULATION_RELATIVE+age+work_years,
               data=df11.train, method = "class", cp=0.003)
summary(tree1)
print(tree1)
printcp(tree1)



#Each node shows
#- the predicted class (died or survived),
#- the predicted probability of survival,
#- the percentage of observations in the node

rpart.plot(tree1)


print(tree1$cptable)

tree1$variable.importance


opt1 <- which.min(tree1$cptable[,"xerror"])
cp1 <- tree1$cptable[opt1, "CP"]
df11train_prune1 <- prune(tree1, cp = cp1)
rpart.plot(df11train_prune1)


# Model evalutaiton
# predict on test data
predictions0 <- predict(tree1, df11.test, type="class")
# check prediction result
table(df11.test$TARGET, predictions0)


#################################################################################
tree3<- rpart(TARGET ~ CODE_GENDER+AMT_INCOME_TOTAL +AMT_CREDIT + age + work_years, data=df11.train,
              method = "class",control=rpart.control(cp=0.001))

print(tree3)
printcp(tree3)
rpart.plot(tree3)
print(tree3$cptable)

opt3 <- which.min(tree3$cptable[,"xerror"])
cp3 <- tree3$cptable[opt3, "CP"]
df11train_prune3 <- prune(tree3, cp = cp3)
rpart.plot(df11train_prune3)




# Model evalutaiton
# predict on test data
predictions3 <- predict(tree3, df11.test, type="class")
# check prediction result
table(df11.test$TARGET, predictions3)




tree3<- rpart(TARGET ~ CODE_GENDER+AMT_INCOME_TOTAL +AMT_CREDIT + age + work_years, data=df11.train,
              method = "class",control=rpart.control(cp=0.001))

print(tree3)
printcp(tree3)
rpart.plot(tree3)
print(tree3$cptable)
```

```
opt3 <- which.min(tree3$cptable[,"xerror"])
cp3 <- tree3$cptable[opt3, "CP"]
df11train_prune3 <- prune(tree3, cp = cp3)
rpart.plot(df11train_prune3)




# Model evalutaiton
# predict on test data
predictions3 <- predict(tree3, df11.test, type="class")
# check prediction result
table(df11.test$TARGET, predictions3)



tree3<- rpart(TARGET ~  NAME_EDUCATION_TYPE+ age + work_years, data=df11.train,
              method = "class",control=rpart.control(cp=0.001))

print(tree3)
printcp(tree3)
rpart.plot(tree3)
print(tree3$cptable)

opt3 <- which.min(tree3$cptable[,"xerror"])
cp3 <- tree3$cptable[opt3, "CP"]
df11train_prune3 <- prune(tree3, cp = cp3)
rpart.plot(df11train_prune3)




# Model evalutaiton
# predict on test data
predictions3 <- predict(tree3, df11.test, type="class")
# check prediction result
table(df11.test$TARGET, predictions3)




##############################################################################################
tree4<- rpart(TARGET ~ AMT_INCOME_TOTAL +AMT_CREDIT + age + work_years, data=df11.train,
              method = "class", minsplit = 2, minbucket = 1, cp=0.001)
rpart.plot(tree4)

opt4 <- which.min(tree4$cptable[,"xerror"])
cp4 <- tree4$cptable[opt4, "CP"]
df11train_prune4 <- prune(tree4, cp = cp4)
rpart.plot(df11train_prune4)


predictions4 <- predict(tree4, df11.test, type="class")
# check prediction result
table(df11.test$TARGET, predictions4)


#####################################    random forest  ##############################
install.packages("randomForest")
library(randomForest)
# ntree: number of trees grown


# Fitting model
ran1 <- randomForest(TARGET ~ NAME_EDUCATION_TYPE  + AMT_INCOME_TOTAL +AMT_CREDIT + NAME_HOUSING_TYPE+ age + work_years, data=df11.train,ntree=100)

importance(ran1)
print(ran1)
# Model evalutaiton
# predict on test data
predictions1 <- predict(ran1, df11.test, type="class")
# check prediction result
table(df11.test$TARGET, predictions1)




ran2 <- randomForest(TARGET ~ NAME_CONTRACT_TYPE + CODE_GENDER + FLAG_OWN_CAR +
                       CNT_CHILDREN+ AMT_INCOME_TOTAL +AMT_CREDIT+AMT_ANNUITY+AMT_GOODS_PRICE
                     +NAME_EDUCATION_TYPE+NAME_HOUSING_TYPE+REGION_POPULATION_RELATIVE+age+work_years, data=df11.train,ntree=100)
importance(ran2)
print(ran2)
# Model evalutaiton
# predict on test data
predictions2 <- predict(ran2, df11.test, type="class")
# check prediction result
table(df11.test$TARGET, predictions2)



ran3 <- randomForest(TARGET ~ CODE_GENDER+AMT_INCOME_TOTAL +AMT_CREDIT + age + work_years, data=df11.train,ntree=100)
importance(ran3)
print(ran3)
# Model evalutaiton
# predict on test data
predictions3 <- predict(ran3, df11.test, type="class")
# check prediction result
table(df11.test$TARGET, predictions3)
```