```
## random forest code
library(mice)
dat = read.csv('aug_train.csv')

dat$city <- as.factor(dat$city)
dat$gender <- as.factor(dat$gender)
dat$relevent_experience <- as.factor(dat$relevent_experience)
dat$enrolled_university <- as.factor(dat$enrolled_university)
dat$education_level <- as.factor(dat$education_level)
dat$major_discipline <- as.factor(dat$major_discipline)
dat$experience <- as.factor(dat$experience)
dat$company_size <- as.factor(dat$company_size)
dat$company_type <- as.factor(dat$company_type)
dat$last_new_job <- as.factor(dat$last_new_job)
dat$target <- as.factor(dat$target)

summary(dat)

# In R, missing data are not displayed as NA, but as ''. Thus, we
first replace these empty strings with 'NA'
dat[dat==''] <- NA
summary(dat)

# First we look at the numerical values:
mean(is.na(dat$city_development_index))
mean(is.na(dat$training_hours))

#No missing values for numerical variables.

#Next we look at the categorical variables
jobdata <- dat
## 2nd missing data imputation
library(missForest)
imputed_Data <- missForest(jobdata[,c(-1,-2)])
jobdata <- imputed_Data$ximp

# jobdata = jobdata[,-1]
# jobdata = jobdata[,-1]
# jobdata

library(VIM)
mice_plot <- aggr(dat, col=c('navyblue','orange'),
                  numbers=TRUE, sortVars=TRUE,
                  labels=names(dat), cex.axis=.7,
```

```
                gap=3, ylab=c("Missing data","Pattern"))
# There are more than 30% of the data in company_type and
company_size is missing.
set.seed(12)
n <- nrow(jobdata)
train.id <- sample(1:n,round(0.8*n))
train <- jobdata[train.id,]
test <- jobdata[-train.id,]

library(data.table)
library(rpart)
library(rpart.plot)
library(randomForest)
setDT(jobdata)[,.N/nrow(train),target]
setDT(train)[,.N/nrow(train),target]
setDT(test)[,.N/nrow(test),target]

Kfold_CV_rf <- function(K, train, ntree,mtry){
  fold_size = floor(nrow(train)/K)
  cv_error = rep(0,K)

  for(i in 1:K){
    # iteratively select K-1 folds as training data in CV procedure,
remaining as test data.
    if(i!=K){
      CV_test_id = ((i-1)*fold_size+1):(i*fold_size)
    }else{
      CV_test_id = ((i-1)*fold_size+1):nrow(train)
    }
    CV_train = train[-CV_test_id,]
    CV_test = train[CV_test_id,]
    set.seed(12)
    rf_mod = randomForest(target ~ ., data = CV_train, ntree = ntree,
mtry = mtry,
                      importance = TRUE)

    rf_test_pred = predict(rf_mod, newdata = CV_test)
    cv_error[i] = mean(rf_test_pred!=CV_test$target)
  }
  # Calculate CV error by taking averages
  return(mean(cv_error))
}

ntree.list = c(100, 200, 500)
```

```
mtry.list = c(2, floor(sqrt(ncol(train))), 4, 5)
errorTable <- matrix(ncol=4, nrow=3)

for(i in 1:length(ntree.list)){
  for(j in 1:length(mtry.list)){
    print(c(ntree.list[i], mtry.list[j]))
    errorTable[i,j]     <-     Kfold_CV_rf(5,     train[,c(-1,-2)],
ntree.list[i],mtry.list[j])
  }

}
errorTable
which.min(errorTable)

trainup[,c(-1,-2)]
set.seed(12)
rf = randomForest(target ~ ., data = train,
                  mtry = 2,
                  ntree=200,
                  importance = TRUE)

rf_pred = predict(rf, newdata = test)
rf_err = mean(rf_pred!=test$target)
1-rf_err

library(tidyverse)
imp <- as.data.frame(importance(rf))
imp <- cbind(vars=rownames(imp), imp)
imp <- imp[order(imp$MeanDecreaseAccuracy),]
imp$vars=factor(imp$vars,levels=unique(imp$vars))
barplot(imp$MeanDecreaseAccuracy, names.arg=imp$vars)

imp %>%
  pivot_longer(cols=matches("Mean")) %>%
  ggplot(aes(value, vars)) +
  geom_col() +
  geom_text(aes(label=round(value),      x=0.5*value),      size=3,
colour="white") +
  facet_grid(. ~ name, scales="free_x") +
  scale_x_continuous(expand=expansion(c(0,0.04))) +
  theme_bw() +
  theme(panel.grid.minor=element_blank(),
        panel.grid.major=element_blank(),
        axis.title=element_blank())
```