MA679 Midterm Exam

Simu Huang

3/18/2021

Context

A company wants to hire data scientists from pool of people enrolled in the courses conduct by the company. The company wants to know which of these candidates are looking to change their job. Information related to demographics, education, experience are in hands from candidates sign up and enrollment. In this exam, your goal is to predict if the candidate is looking for a new job or will work for the current company.

- uid : Unique ID for candidate
- city: City code
- city_dev_index : Development index of the city (scaled)
- gender: Gender of candidate
- relevant_experience: Relevant experience of candidate
- enrolled_university: Type of University course enrolled if any
- education_level: Education level of candidate
- major_discipline :Education major discipline of candidate
- experience_years: Candidate total experience in years
- company_size: No of employees in current employer's company
- company_type : Type of current employer
- lastnewjob: Difference in years between previous job and current job
- training_hours: training hours completed
- change_job: 0 – Not looking for job change, 1 – Looking for a job change

---

Details of your work to be submitted.

Data Processing

```
summary(train)
```

```
##        V1            uid          city_id          city_dev_index  
##  Min.   :   1   Min.   :    1   Length:8000        Min.   :0.4480  
##  1st Qu.:2001   1st Qu.: 8295   Class :character   1st Qu.:0.7430  
##  Median :4000   Median :16660   Mode  :character   Median :0.9030  
##  Mean   :4000   Mean   :16734                      Mean   :0.8293  
##  3rd Qu.:6000   3rd Qu.:25081                      3rd Qu.:0.9200  
##  Max.   :8000   Max.   :33377                      Max.   :0.9490  
##     gender          relevant_experience enrolled_university education_level   
##  Length:8000        Length:8000         Length:8000         Length:8000       
##  Class :character   Class :character    Class :character    Class :character  
##  Mode  :character   Mode  :character    Mode  :character    Mode  :character  
## 
## 
## 
```

```
##  major_discipline   experience_years   company_size       company_type
##  Length:8000        Length:8000        Length:8000        Length:8000
##  Class :character   Class :character   Class :character   Class :character
##  Mode  :character   Mode  :character   Mode  :character   Mode  :character
##
##
##
##  last_new_job       training_hours     change_job
##  Length:8000        Min.   :  1.00   Min.   :0.0000
##  Class :character   1st Qu.: 23.00   1st Qu.:0.0000
##  Mode  :character   Median : 47.00   Median :0.0000
##                     Mean   : 65.02   Mean   :0.2432
##                     3rd Qu.: 88.00   3rd Qu.:0.0000
##                     Max.   :336.00   Max.   :1.0000
```

```r
# unique(train$gender)
# unique(train$relevant_experience)
# unique(train$enrolled_university)
# unique(train$experience_years)
# unique(train$city_id)
# unique(train$company_size)

df <- data.frame(train)
df %<>% separate(city_id, c("city_x", "city_id_n"), sep = "_" )
df <- df[, -3]

df$gender[which(df$gender == "")] <- "unknown"

# table(df[, 9])
# table(df[, 11])
# table(df[, 7])
# table(df[, 8])
# table(df[, 11])
# table(df[, 12])

df$major_discipline[which(df$major_discipline == "")] <- "STEM"
df$enrolled_university[which(df$enrolled_university == "")] <- "no_enrollment"
df$education_level[which(df$education_level == "")] <- "Graduate"
df$company_size[which(df$company_size == "")] <- "unknown"
df$company_type[which(df$company_type == "")] <- "Pvt Ltd"

df$city_id_n <- as.numeric(df$city_id_n)

names_fac <- c(5:13)
df[,names_fac] <- lapply(df[,names_fac], as.factor)

# sum(is.na(df))

set.seed(327)
index <- sample(nrow(df), nrow(df)/2, replace = FALSE )

#training samples
df_tr <- df[index, -1]
```

```
#testing samples
df_ts <- df[-index, -1]

#summary(df_tr)
#summary(df_ts)
```

Model Selection

```
summary(rf_fit)
```

```
##                  Length Class  Mode     
## call                4   -none- call     
## type                1   -none- character
## predicted        4000   factor numeric  
## err.rate         1500   -none- numeric  
## confusion           6   -none- numeric  
## votes            8000   matrix numeric  
## oob.times        4000   -none- numeric  
## classes             2   -none- character
## importance          9   -none- numeric  
## importanceSD        0   -none- NULL     
## localImportance     0   -none- NULL     
## proximity           0   -none- NULL     
## ntree               1   -none- numeric  
## mtry                1   -none- numeric  
## forest             14   -none- list     
## y                4000   factor numeric  
## test                0   -none- NULL     
## inbag               0   -none- NULL     
## terms               3   terms  call
```

Model Validation

```
## 
## Call:
##  randomForest(formula = as.factor(change_job) ~ city_id_n + city_dev_index +      gender + relevant_
##                Type of random forest: classification
##                      Number of trees: 500
## No. of variables tried at each split: 3
## 
##         OOB estimate of  error rate: 23%
## Confusion matrix:
##      0   1 class.error
## 0 2656 377   0.1242994
## 1  543 424   0.5615305
```

```
## [1] "The Random Forest correctly predict <U+200E>79.97% of people's choices."
```

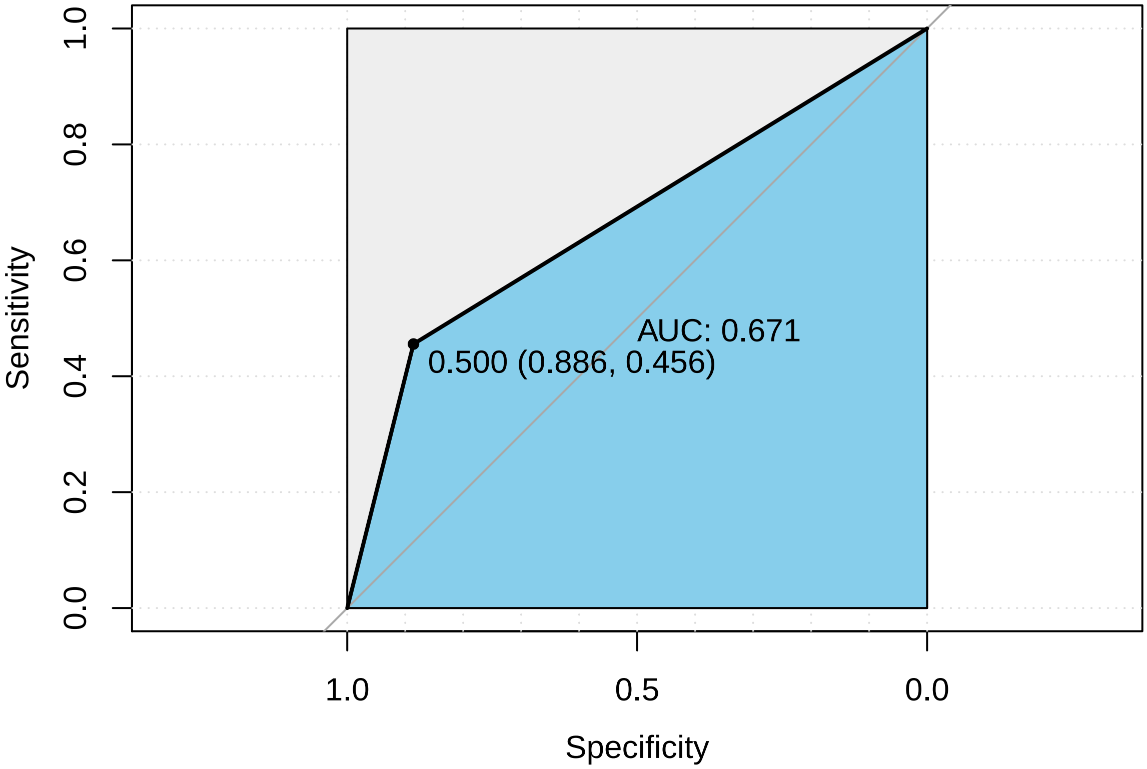
Model Evaluation

```
## Setting levels: control = 0, case = 1
```

```
## Setting direction: controls < cases
```

```
##
```

```
## Call:
## roc.default(response = as.ordered(df_ts$change_job), predictor = as.ordered(rf_pre))
##
## Data: as.ordered(rf_pre) in 3021 controls (as.ordered(df_ts$change_job) 0) < 979 cases (as.ordered(d
## Area under the curve: 0.6707
```



The area under curve here is 0.6707, which means that the model can accurately predict 67.1% of the sample. Generally, 0.7 to 0.8 is considered acceptable. The random forest we used here is suitable.

Discussion

The advantages of random forests are that random forest can process high-dimensional data without reducing features and if a large part of the features are missing, the accuracy can still be maintained. Meanwhile, the random forest can directly deal with qualitative variables without creating dummy variables. Therefore, we choose random forest here.

Limitations

There are too many missing values in the variable "Gender", which will affects our prediction. The data processing methods we use here are that replacing an entire range of values with a specific value and treating many numerical values as factors, which may affect the accuracy of our prediction.

Reference

1.[r package]"data.table","magrittr", "tidyr","formattable","MASS", "randomForest", "caret", "pROC"

2.[random forest] (https://easyai.tech/ai-definition/random-forest/#tests)

3.[pRoc] (https://cran.r-project.org/web/packages/pROC/pROC.pdf)

Appendix

Processing the test sample

```r
test <- data.frame(test)
test %<>% separate(city_id, c("city_x", "city_id_n"), sep = "_" )
test <- test[, -3]

test$gender[which(test$gender == "")] <- "unknown"

test$major_discipline[which(test$major_discipline == "")] <- "STEM"
test$enrolled_university[which(test$enrolled_university == "")] <- "no_enrollment"
test$education_level[which(test$education_level == "")] <- "Graduate"
test$company_size[which(test$company_size == "")] <- "unknown"
test$company_type[which(test$company_type == "")] <- "Pvt Ltd"
test$city_id_n <- as.numeric(test$city_id_n)

names_fac <- c(5:13)
test[,names_fac] <- lapply(test[,names_fac], as.factor)

ts_pred <- predict(rf_fit, test, type = "class")
submission$change_job <- ts_pred
#sum(is.na(test))

write.csv(submission, file = "submission.csv")
```