Final Project for Data Science II

Group 6

Contents

#Introduction ## Data Description This dataset is designed to understand the factors that lead a person to leave current job. Information contained in the dataset are demographics(city, gender, etc), education(education level, major displine, etc), experience(experience, company size, etc) of employees. The outcome is the variable "target"(binary), where "0" represents the employee is not looking for job change while "1" represents the employee is looking for a job change. Using this dataset, we can predict the probability of a employee to look for a new job based on their demographic, edcation and experience information. The more specific information of the data can be found at here.

```
job = read_csv("job/aug_train.csv") %>% select(-c(1,2))
```

Data Preproccessing

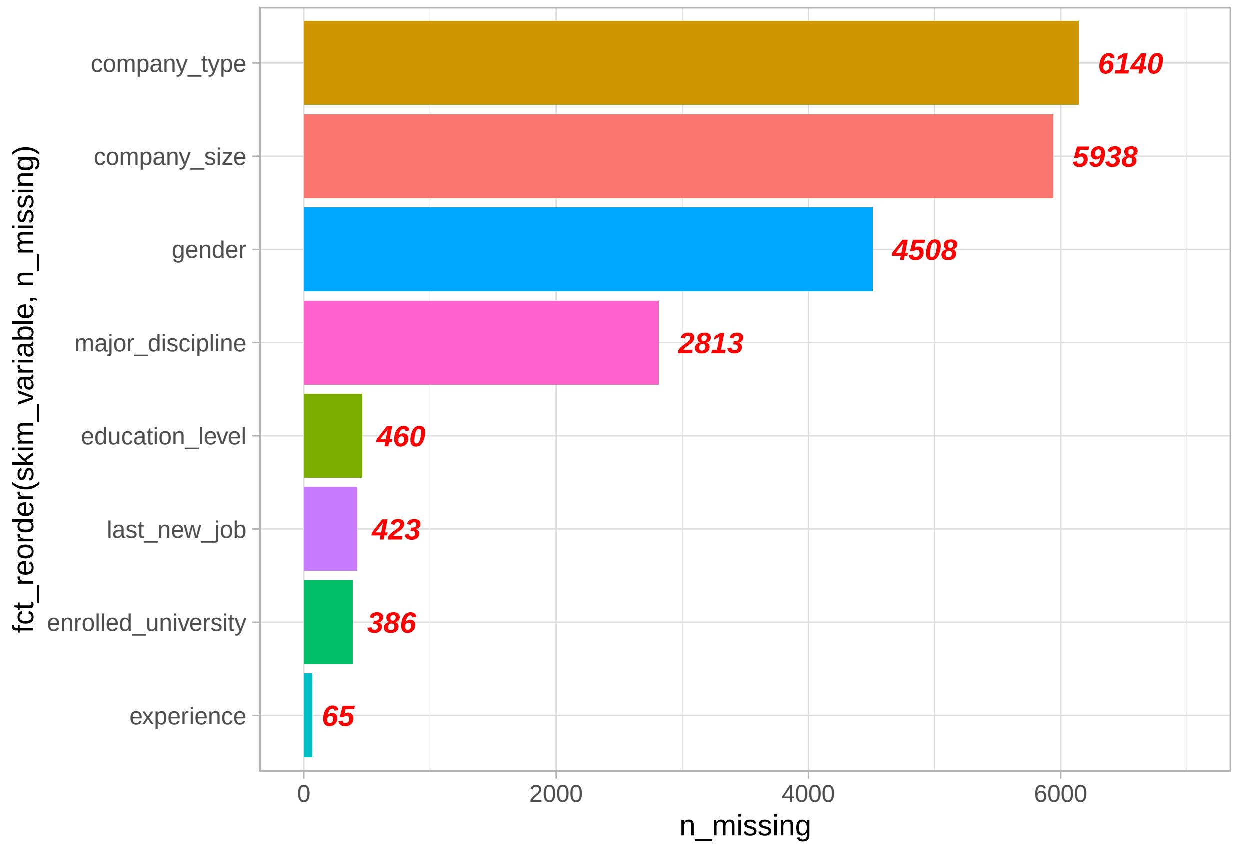
Predictors Selection

There are 13 features in this dataset, but enrollee_id is not a predictor. What's more, a more meaningful way to assess the influence of a city is through its extend development, so I excluded "city" since we have "city_development_index" feature.

Missing data

According to the misssingness figure[figure1], "company_type" and "company_zise", "gender", "major_displine" has relatively large propotion of missingness. There are also some missingness in education_leval, enrolled_university, last_new_job and experience, but those missingness only account for a small proportion. For the predictor that has small proportion of missingness, I simply droped the observations that has such missingess. For the four variable that has high proportion of missingness, I used missForest to do the imputation. Before doing the imputation, I tansfered all characters into factors.

```
Skimmed <- skimr::skim(job)

Skimmed %>% select(skim_variable, n_missing) %>%
  filter(n_missing != 0) %>%
  ggplot(aes(
    x = fct_reorder(skim_variable, n_missing),
    y = n_missing,
    label = n_missing,
    fill = skim_variable
  )) +
  geom_col() +
  geom_text(hjust = -0.3,
            color = "red",
            fontface = "bold.italic") +
  coord_flip() +
  scale_y_continuous(limits = c(0, 7000)) +
  theme_light() +
  theme(legend.position = "none")
```




```
job =job%>%
  drop_na(c(4, 5, 7, 10)) %>%
  rename(relevant_experience = relevent_experience) %>%
  mutate(target = recode_factor(target,
                              `1` = "change", `0` = "no_change" )) %>%
  mutate_if(is.character, as.factor) %>%
  as.data.frame()

summary(job)
```

```
##  city_development_index    gender                      relevant_experience
##  Min.   :0.4480          Female: 1206   Has relevent experience:13190
##  1st Qu.:0.7450          Male  :12772   No relevent experience : 4824
##  Median :0.9100          Other :  173
##  Mean   :0.8317          NA's  : 3863
##  3rd Qu.:0.9200
##  Max.   :0.9490
##
##        enrolled_university       education_level         major_discipline
##  Full time course: 3517     Graduate      :11188    Arts            :  248
##  no_enrollment   :13348     High School   : 1908    Business Degree:  322
##  Part time course: 1149     Masters       : 4228    Humanities      :  653
##                             Phd           :  399    No Major        :  212
##                             Primary School:  291    Other           :  364
##                                                     STEM            :13993
##                                                     NA's            : 2222
```

```
##    experience      company_size               company_type  last_new_job
##   >20     :3182   50-99    :2950   Early Stage Startup: 562   >4   :3210
##   5       :1337   100-500  :2483   Funded Startup     : 975   1    :7789
##   4       :1298   10000+   :1964   NGO                : 500   2    :2827
##   3       :1223   10/49    :1394   Other              : 114   3    : 991
##   6       :1143   1000-4999:1282   Public Sector      : 912   4    :1010
##   2       : 997   (Other)  :2631   Pvt Ltd            :9475   never:2187
##   (Other):8834   NA's     :5310   NA's               :5476
##   training_hours        target
##   Min.   :  1.00   change   : 4421
##   1st Qu.: 23.00   no_change:13593
##   Median : 47.00
##   Mean   : 65.35
##   3rd Qu.: 88.00
##   Max.   :336.00
##
```

```
set.seed(2021)

rowTrain = createDataPartition(y = job$target,
                               p = 0.8,
                               list = FALSE)
dat_tr = job[rowTrain,]
dat_te = job[-rowTrain,]
```

```
set.seed(2021)
imputed_tr <- missForest(dat_tr, maxiter = 2, ntree = 20)
```

```
##   missForest iteration 1 in progress...done!
##   missForest iteration 2 in progress...done!
```

```
imputed_te <- missForest(dat_te, maxiter = 2, ntree = 20)
```

```
##   missForest iteration 1 in progress...done!
##   missForest iteration 2 in progress...done!
```

```
job_tr = imputed_tr$ximp
job_te = imputed_te$ximp
```

Exploratory Analysis

```
tab <- tableby(target ~ relevant_experience +gender + enrolled_university+education_level+major_discipli
summary(tab, title = "Descriptive Statistics: Job Change",  text=T)
```

```
##
## Table: Descriptive Statistics: Job Change
##
## |                            | change (N=4421) | no_change (N=13593) | Total (N=18014) | p value|
## |:---------------------------|:---------------:|:-------------------:|:---------------:|-------:|
```

```
## |relevant_experience          |                    |                     |                    | < 0.001|
## |-  Has relevent experience   |  2772 (62.7%)      |    10418 (76.6%)    |  13190 (73.2%)     |        |
## |-  No relevent experience    |  1649 (37.3%)      |    3175 (23.4%)     |  4824 (26.8%)      |        |
## |gender                       |                    |                     |                    |   0.069|
## |-  Female                    |   356 (8.1%)       |     960 (7.1%)      |   1316 (7.3%)      |        |
## |-  Male                      |  4017 (90.9%)      |    12501 (92.0%)    |   16518 (91.7%)    |        |
## |-  Other                     |    48 (1.1%)       |     132 (1.0%)      |   180 (1.0%)       |        |
## |enrolled_university          |                    |                     |                    | < 0.001|
## |-  Full time course          |  1333 (30.2%)      |    2184 (16.1%)     |  3517 (19.5%)      |        |
## |-  no_enrollment             |  2806 (63.5%)      |    10542 (77.6%)    |  13348 (74.1%)     |        |
## |-  Part time course          |   282 (6.4%)       |    867 (6.4%)       |   1149 (6.4%)      |        |
## |education_level              |                    |                     |                    | < 0.001|
## |-  Graduate                  |  3073 (69.5%)      |    8115 (59.7%)     |  11188 (62.1%)     |        |
## |-  High School               |   372 (8.4%)       |    1536 (11.3%)     |  1908 (10.6%)      |        |
## |-  Masters                   |   884 (20.0%)      |    3344 (24.6%)     |  4228 (23.5%)      |        |
## |-  Phd                       |    55 (1.2%)       |    344 (2.5%)       |   399 (2.2%)       |        |
## |-  Primary School            |    37 (0.8%)       |    254 (1.9%)       |   291 (1.6%)       |        |
## |major_discipline             |                    |                     |                    |   0.103|
## |-  Arts                      |    52 (1.2%)       |     197 (1.4%)      |   249 (1.4%)       |        |
## |-  Business Degree           |    85 (1.9%)       |     240 (1.8%)      |   325 (1.8%)       |        |
## |-  Humanities                |   136 (3.1%)       |     533 (3.9%)      |   669 (3.7%)       |        |
## |-  No Major                  |    52 (1.2%)       |     162 (1.2%)      |   214 (1.2%)       |        |
## |-  Other                     |    94 (2.1%)       |     274 (2.0%)      |   368 (2.0%)       |        |
## |-  STEM                      |  4002 (90.5%)      |    12187 (89.7%)    |  16189 (89.9%)     |        |
## |experience                   |                    |                     |                    | < 0.001|
## |-  <1                        |   206 (4.7%)       |     245 (1.8%)      |   451 (2.5%)       |        |
## |-  >20                       |   486 (11.0%)      |    2696 (19.8%)     |  3182 (17.7%)      |        |
## |-  1                         |   200 (4.5%)       |     275 (2.0%)      |   475 (2.6%)       |        |
## |-  10                        |   202 (4.6%)       |     744 (5.5%)      |   946 (5.3%)       |        |
## |-  11                        |   148 (3.3%)       |     501 (3.7%)      |   649 (3.6%)       |        |
## |-  12                        |    85 (1.9%)       |     390 (2.9%)      |   475 (2.6%)       |        |
## |-  13                        |    73 (1.7%)       |     314 (2.3%)      |   387 (2.1%)       |        |
## |-  14                        |   101 (2.3%)       |     468 (3.4%)      |   569 (3.2%)       |        |
## |-  15                        |   111 (2.5%)       |     557 (4.1%)      |   668 (3.7%)       |        |
## |-  16                        |    65 (1.5%)       |     423 (3.1%)      |   488 (2.7%)       |        |
## |-  17                        |    56 (1.3%)       |     275 (2.0%)      |   331 (1.8%)       |        |
## |-  18                        |    39 (0.9%)       |     234 (1.7%)      |   273 (1.5%)       |        |
## |-  19                        |    48 (1.1%)       |     246 (1.8%)      |   294 (1.6%)       |        |
## |-  2                         |   336 (7.6%)       |     661 (4.9%)      |   997 (5.5%)       |        |
## |-  20                        |    33 (0.7%)       |     109 (0.8%)      |   142 (0.8%)       |        |
## |-  3                         |   427 (9.7%)       |     796 (5.9%)      |   1223 (6.8%)      |        |
## |-  4                         |   417 (9.4%)       |     881 (6.5%)      |   1298 (7.2%)      |        |
## |-  5                         |   389 (8.8%)       |     948 (7.0%)      |   1337 (7.4%)      |        |
## |-  6                         |   327 (7.4%)       |     816 (6.0%)      |   1143 (6.3%)      |        |
## |-  7                         |   290 (6.6%)       |     692 (5.1%)      |   982 (5.5%)       |        |
## |-  8                         |   184 (4.2%)       |     584 (4.3%)      |   768 (4.3%)       |        |
## |-  9                         |   198 (4.5%)       |     738 (5.4%)      |   936 (5.2%)       |        |
## |company_size                 |                    |                     |                    | < 0.001|
## |-  <10                       |   462 (10.5%)      |    1546 (11.4%)     |  2008 (11.1%)      |        |
## |-  10/49                     |   636 (14.4%)      |    1362 (10.0%)     |  1998 (11.1%)      |        |
## |-  100-500                   |   647 (14.6%)      |    2494 (18.3%)     |  3141 (17.4%)      |        |
## |-  1000-4999                 |   496 (11.2%)      |    1591 (11.7%)     |  2087 (11.6%)      |        |
## |-  10000+                    |   797 (18.0%)      |    2160 (15.9%)     |  2957 (16.4%)      |        |
## |-  50-99                     |   744 (16.8%)      |    2743 (20.2%)     |  3487 (19.4%)      |        |
```

```
## |-  500-999                  |  381 (8.6%)    |    1015 (7.5%)     |  1396 (7.7%)   |        |
## |-  5000-9999                |  258 (5.8%)    |    682 (5.0%)      |  940 (5.2%)    |        |
## |company_type                |                |                    |                | < 0.001|
## |-  Early Stage Startup      |  284 (6.4%)    |    735 (5.4%)      |  1019 (5.7%)   |        |
## |-  Funded Startup           |  228 (5.2%)    |    1046 (7.7%)     |  1274 (7.1%)   |        |
## |-  NGO                      |  166 (3.8%)    |    482 (3.5%)      |  648 (3.6%)    |        |
## |-  Other                    |   41 (0.9%)    |    101 (0.7%)      |  142 (0.8%)    |        |
## |-  Public Sector            |  539 (12.2%)   |    1025 (7.5%)     |  1564 (8.7%)   |        |
## |-  Pvt Ltd                  | 3163 (71.5%)   |   10204 (75.1%)    | 13367 (74.2%)  |        |
## |last_new_job                |                |                    |                | < 0.001|
## |-  >4                       |  578 (13.1%)   |   2632 (19.4%)     | 3210 (17.8%)   |        |
## |-  1                        | 2038 (46.1%)   |   5751 (42.3%)     | 7789 (43.2%)   |        |
## |-  2                        |  679 (15.4%)   |   2148 (15.8%)     | 2827 (15.7%)   |        |
## |-  3                        |  223 (5.0%)    |    768 (5.6%)      |  991 (5.5%)    |        |
## |-  4                        |  220 (5.0%)    |    790 (5.8%)      |  1010 (5.6%)   |        |
## |-  never                    |  683 (15.4%)   |   1504 (11.1%)     | 2187 (12.1%)   |        |
```

Visualazation of categorical variables

For categorical data, I make the table to show the percentage of each level accounted for the two classes. From the descriptive statistics of catogorical data,there is no very explicit structure of the data.But from the table, we can see that, for some of the predictors there are various levels that could result in too many dummy variables in the following model build process. After carefully look into different levels and their percentage,I decided to make some data collapse to reduce some of the levels. I believe it could save some computing effort without severly hurt the prediction.

```
job_tr = job_tr %>% mutate(enrolled_university = case_when(
                                            enrolled_university == "no_enrollment" ~ "noEnroll",
                                            enrolled_university %in% c("Full time course", "Part time c
              mutate(education_level = case_when(
                                        education_level %in% c("Masters", "Phd") ~ "aboveCollege",
                                        education_level %in% c("Primary School", "High School") ~ "noCol
                                        TRUE ~ "college")) %>%
             mutate(major_discipline = case_when(
                                         major_discipline == "STEM" ~ "STEM",
                                         TRUE ~ "non_STEM")) %>%
             mutate(experience = case_when(
                                  experience  == ">20" ~ "twenty",
                                  experience == "<1" ~ "one",
                                  TRUE ~ "oneTotwenty")) %>%
             mutate(company_size = case_when(
                                    company_size %in% c("<10","10/49","50-99","100-500") ~ "small",
                                    company_size %in% c("500-999","1000-4999", "5000-9999") ~ "medium",
                                    TRUE ~ "big"))%>%
             mutate(last_new_job = case_when(
                                    last_new_job == "1" ~ "one",
                                    last_new_job == "never" ~ "never",
                                    TRUE ~ "two")) %>%
            mutate(company_type = case_when(
                                  company_type %in% c("Early Stage Startup", "Funded Startup", "NGO", "
                                  company_type == "Pvt Ltd" ~ "pvtLtd")) %>%
  mutate(relevant_experience = case_when(
                      relevant_experience == "Has relevent experience" ~ "yes",
```

```
                      relevant_experience == "No relevent experience" ~ "no"
  )) %>%
  mutate_if(is.character, as.factor)

job_te = job_te %>% mutate(enrolled_university = case_when(
                                             enrolled_university == "no_enrollment" ~ "noEnroll",
                                             enrolled_university %in% c("Full time course", "Part time co
              mutate(education_level = case_when(
                                         education_level %in% c("Masters", "Phd") ~ "aboveCollege",
                                         education_level %in% c("Primary School", "High School") ~ "noCol
                                         TRUE ~ "college")) %>%
             mutate(major_discipline = case_when(
                                          major_discipline == "STEM" ~ "STEM",
                                          TRUE ~ "non_STEM")) %>%
             mutate(experience = case_when(
                                   experience  == ">20" ~ "twenty",
                                   experience == "<1" ~ "one",
                                   TRUE ~ "oneTotwenty")) %>%
             mutate(company_size = case_when(
                                     company_size %in% c("<10","10/49","50-99","100-500") ~ "small",
                                     company_size %in% c("500-999","1000-4999", "5000-9999") ~ "medium",
                                     TRUE ~ "big"))%>%
             mutate(last_new_job = case_when(
                                     last_new_job == "1" ~ "one",
                                     last_new_job == "never" ~ "never",
                                     TRUE ~ "two")) %>%
            mutate(company_type = case_when(
                                   company_type %in% c("Early Stage Startup", "Funded Startup", "NGO", "(
                                   company_type == "Pvt Ltd" ~ "pvtLtd")) %>%
  mutate(relevant_experience = case_when(
                      relevant_experience == "Has relevent experience" ~ "yes",
                      relevant_experience == "No relevent experience" ~ "no"
  )) %>%
  mutate_if(is.character, as.factor)


tab2 <- tableby(target ~ relevant_experience+enrolled_university+education_level+major_discipline+experi
summary(tab2, title = "Descriptive Statistics: Job Change",  text=T)
```

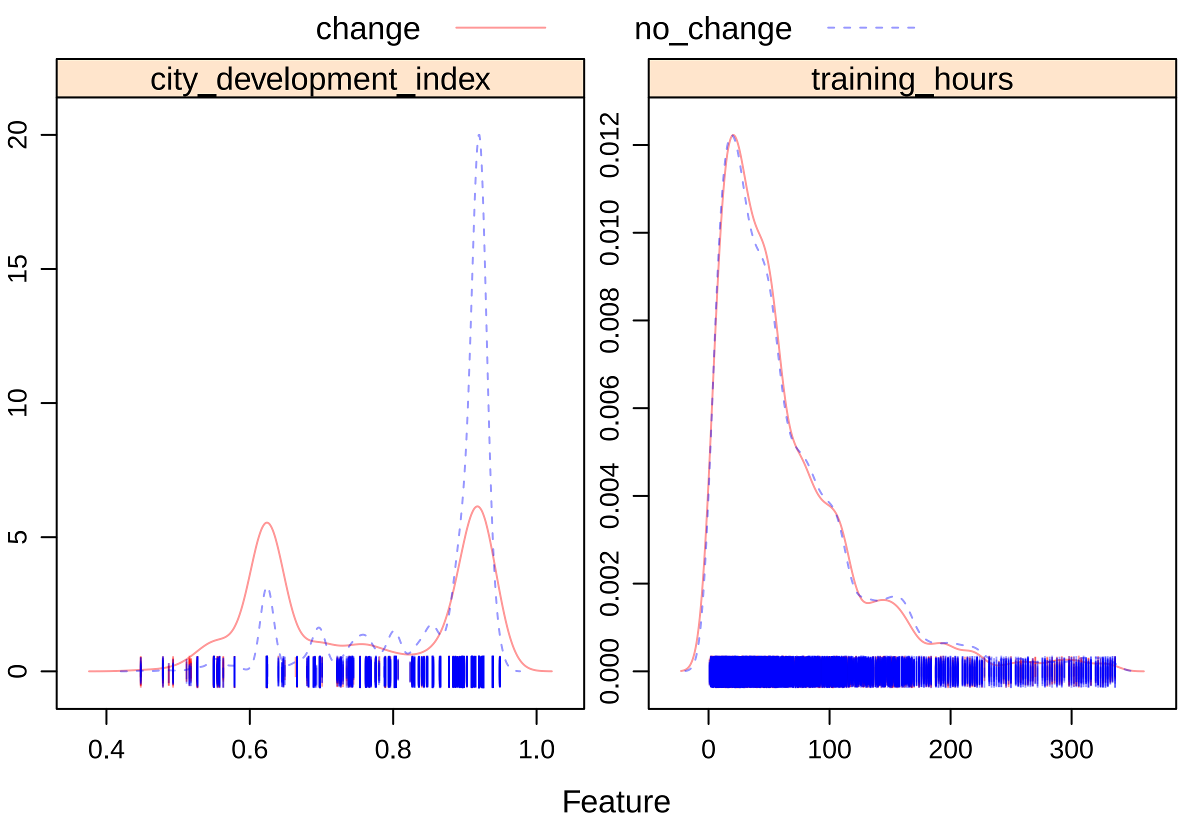
```
##
## Table: Descriptive Statistics: Job Change
##
## |                     | change (N=4421) | no_change (N=13593) | Total (N=18014) | p value|
## |:--------------------|:---------------:|:-------------------:|:---------------:|-------:|
## |relevant_experience  |                 |                     |                 | < 0.001|
## |-   no               |  1649 (37.3%)   |    3175 (23.4%)     |  4824 (26.8%)   |        |
## |-   yes              |  2772 (62.7%)   |    10418 (76.6%)    |  13190 (73.2%)  |        |
## |enrolled_university  |                 |                     |                 | < 0.001|
## |-   enrolled         |  1615 (36.5%)   |    3051 (22.4%)     |  4666 (25.9%)   |        |
## |-   noEnroll         |  2806 (63.5%)   |    10542 (77.6%)    |  13348 (74.1%)  |        |
## |education_level      |                 |                     |                 | < 0.001|
## |-   aboveCollege     |   939 (21.2%)   |    3688 (27.1%)     |  4627 (25.7%)   |        |
## |-   college          |  3073 (69.5%)   |    8115 (59.7%)     |  11188 (62.1%)  |        |
```

```
## |-  noCollege         |  409 (9.3%)     |    1790 (13.2%)     |  2199 (12.2%)    |        |
## |major_discipline     |                 |                     |                  |   0.097|
## |-  non_STEM          |  419 (9.5%)     |    1406 (10.3%)     |  1825 (10.1%)    |        |
## |-  STEM              | 4002 (90.5%)    |    12187 (89.7%)    |  16189 (89.9%)   |        |
## |experience           |                 |                     |                  | < 0.001|
## |-  one               |  206 (4.7%)     |     245 (1.8%)      |   451 (2.5%)     |        |
## |-  oneTotwenty       | 3729 (84.3%)    |    10652 (78.4%)    |  14381 (79.8%)   |        |
## |-  twenty            |  486 (11.0%)    |    2696 (19.8%)     |  3182 (17.7%)    |        |
## |company_size         |                 |                     |                  | < 0.001|
## |-  big               |  797 (18.0%)    |    2160 (15.9%)     |  2957 (16.4%)    |        |
## |-  medium            | 1135 (25.7%)    |    3288 (24.2%)     |  4423 (24.6%)    |        |
## |-  small             | 2489 (56.3%)    |    8145 (59.9%)     |  10634 (59.0%)   |        |
## |company_type         |                 |                     |                  | < 0.001|
## |-  other             | 1258 (28.5%)    |    3389 (24.9%)     |  4647 (25.8%)    |        |
## |-  pvtLtd            | 3163 (71.5%)    |    10204 (75.1%)    |  13367 (74.2%)   |        |
## |last_new_job         |                 |                     |                  | < 0.001|
## |-  never             |  683 (15.4%)    |    1504 (11.1%)     |  2187 (12.1%)    |        |
## |-  one               | 2038 (46.1%)    |    5751 (42.3%)     |  7789 (43.2%)    |        |
## |-  two               | 1700 (38.5%)    |    6338 (46.6%)     |  8038 (44.6%)    |        |
## |gender               |                 |                     |                  |   0.069|
## |-  Female            |  356 (8.1%)     |     960 (7.1%)      |   1316 (7.3%)    |        |
## |-  Male              | 4017 (90.9%)    |    12501 (92.0%)    |  16518 (91.7%)   |        |
## |-  Other             |   48 (1.1%)     |     132 (1.0%)      |   180 (1.0%)     |        |
```

Visualization of continuous variables

From the density curve of city_development_index and training hours for different outcomes[figure2], it can be seen that city_development_index might play an important role in predicting the outcome.

```r
theme1 <- transparentTheme(trans = .4)
trellis.par.set(theme1)

full_data = rbind(job_tr,job_te)

featurePlot(x = full_data[, c(1, 11)],
            y = full_data$target,
            scales = list(x = list(relation = "free"),
                          y = list(relation = "free")),
            plot = "density", pch = "|",
            auto.key = list(columns = 2))
```



Models

```
x_tr = model.matrix(target~., job_tr)[, -1]
y_tr = job_tr[,12]

x_te = model.matrix(target~., job_te)[, -1]
y_te = job_te[,12]
```

```
sample_1000 = sample_n(job_tr, 1000)

x_tr_1000 = model.matrix(target~., sample_1000)[, -1]
y_tr_1000 = sample_1000[, 12]
```

```
ctrl = trainControl(method = "cv",
                    summaryFunction = twoClassSummary,
                    classProbs = TRUE)
```

```
set.seed(2)
model.glm = train(x = x_tr_1000,
                  y = y_tr_1000,
                  method = 'glm',
                  metric = "ROC",
                 trControl = ctrl)
#We consider the simple classifier with a cut-off of 0.5 and evaluate its performance on the test data.
```

```r
test.pred.prob = predict(model.glm, newdata = x_te, type = "prob")[,1]
test.pred = rep("change", length(test.pred.prob))
test.pred[test.pred.prob<0.5] = "no_change"

confusionMatrix(data = as.factor(test.pred),
                reference = y_te,
                positive = "change")
```

```
## Confusion Matrix and Statistics
##
##            Reference
## Prediction  change no_change
##   change       197       152
##   no_change    687      2566
##
##                Accuracy : 0.7671
##                  95% CI : (0.7529, 0.7808)
##     No Information Rate : 0.7546
##     P-Value [Acc > NIR] : 0.04185
##
##                   Kappa : 0.2098
##
##  Mcnemar's Test P-Value : < 2e-16
##
##             Sensitivity : 0.22285
##             Specificity : 0.94408
##          Pos Pred Value : 0.56447
##          Neg Pred Value : 0.78881
##              Prevalence : 0.24542
##          Detection Rate : 0.05469
##    Detection Prevalence : 0.09689
##       Balanced Accuracy : 0.58346
##
##        'Positive' Class : change
##
```

```r
model.glm$bestTune
```

```
##   parameter
## 1      none
```

```r
glmnGrid <- expand.grid(.alpha = seq(0, 1, length = 6),
                        .lambda = exp(seq(-8, -2, length = 20)))
set.seed(2)
model.glmn <- train(x = x_tr_1000,
                    y = y_tr_1000,
                    method = "glmnet",
                    tuneGrid = glmnGrid,
                    metric = "ROC",
                    trControl = ctrl)
#We consider the simple classifier with a cut-off of 0.5 and evaluate its performance on the test data.
```

```r
test.pred.prob = predict(model.glmn, newdata = x_te, type = "prob")[,1]
test.pred = rep("change", length(test.pred.prob))
test.pred[test.pred.prob<0.5] = "no_change"

confusionMatrix(data = as.factor(test.pred),
                reference = y_te,
                positive = "change")
```

```
## Confusion Matrix and Statistics
##
##            Reference
## Prediction  change no_change
##   change       184       140
##   no_change    700      2578
##
##                Accuracy : 0.7668
##                  95% CI : (0.7526, 0.7805)
##     No Information Rate : 0.7546
##     P-Value [Acc > NIR] : 0.04547
##
##                   Kappa : 0.1992
##
##  Mcnemar's Test P-Value : < 2e-16
##
##             Sensitivity : 0.20814
##             Specificity : 0.94849
##          Pos Pred Value : 0.56790
##          Neg Pred Value : 0.78646
##              Prevalence : 0.24542
##          Detection Rate : 0.05108
##    Detection Prevalence : 0.08995
##       Balanced Accuracy : 0.57832
##
##        'Positive' Class : change
##
```

```r
model.glmn$bestTune
```

```
##     alpha      lambda
## 108     1 0.003059592
```

```r
set.seed(2)
model.gam <- train(x = x_tr_1000,
                   y = y_tr_1000,
                   method = "gam",
                   metric = "ROC",
                   trControl = ctrl)

#We consider the simple classifier with a cut-off of 0.5 and evaluate its performance on the test data.

test.pred.prob = predict(model.gam, newdata = x_te, type = "prob")[,1]
test.pred = rep("change", length(test.pred.prob))
```

```
test.pred[test.pred.prob<0.5] = "no_change"

confusionMatrix(data = as.factor(test.pred),
                reference = y_te,
                positive = "change")
```

```
## Confusion Matrix and Statistics
##
##            Reference
## Prediction  change no_change
##   change       270       208
##   no_change    614      2510
##
##                Accuracy : 0.7718
##                  95% CI : (0.7577, 0.7854)
##     No Information Rate : 0.7546
##     P-Value [Acc > NIR] : 0.008261
##
##                   Kappa : 0.2709
##
##  Mcnemar's Test P-Value : < 2.2e-16
##
##             Sensitivity : 0.30543
##             Specificity : 0.92347
##          Pos Pred Value : 0.56485
##          Neg Pred Value : 0.80346
##              Prevalence : 0.24542
##          Detection Rate : 0.07496
##    Detection Prevalence : 0.13270
##       Balanced Accuracy : 0.61445
##
##        'Positive' Class : change
##
```

```
model.gam$bestTune
```

```
##   select method
## 1  FALSE GCV.Cp
```

```
set.seed(2)
model.mars <- train(x = x_tr_1000,
                    y = y_tr_1000,
                    method = "earth",
                    tuneGrid = expand.grid(degree = 1:3,
                                           nprune = 2:15),
                    metric = "ROC",
                    trControl = ctrl)
```

```
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
```

```
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
```

```
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
```

```
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
```

```
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
```

```
test.pred.prob = predict(model.mars, newdata = x_te, type = "prob")[,1]
test.pred = rep("change", length(test.pred.prob))
test.pred[test.pred.prob<0.5] = "no_change"

confusionMatrix(data = as.factor(test.pred),
                reference = y_te,
                positive = "change")
```

```
## Confusion Matrix and Statistics
##
##            Reference
## Prediction  change no_change
##   change       359       257
##   no_change    525      2461
##
##                Accuracy : 0.7829
##                  95% CI : (0.7691, 0.7963)
##     No Information Rate : 0.7546
##     P-Value [Acc > NIR] : 3.438e-05
##
##                   Kappa : 0.3471
##
##  Mcnemar's Test P-Value : < 2.2e-16
##
##             Sensitivity : 0.40611
##             Specificity : 0.90545
##          Pos Pred Value : 0.58279
##          Neg Pred Value : 0.82418
##              Prevalence : 0.24542
##          Detection Rate : 0.09967
##    Detection Prevalence : 0.17102
##       Balanced Accuracy : 0.65578
##
##        'Positive' Class : change
##
```

```r
model.mars$bestTune
```

```
##    nprune degree
## 25     12      2
```

```r
set.seed(2)
model.lda = train(x = x_tr_1000,
                  y = y_tr_1000,
                  method = "lda",
                  metric = "ROC",
                  trControl = ctrl)

test.pred.prob = predict(model.lda, newdata = x_te, type = "prob")[,1]
test.pred = rep("change", length(test.pred.prob))
test.pred[test.pred.prob<0.5] = "no_change"

confusionMatrix(data = as.factor(test.pred),
                reference = y_te,
                positive = "change")
```

```
## Confusion Matrix and Statistics
##
##            Reference
## Prediction  change no_change
##   change       286       230
##   no_change    598      2488
##
##                Accuracy : 0.7701
##                  95% CI : (0.756, 0.7838)
##     No Information Rate : 0.7546
##     P-Value [Acc > NIR] : 0.01534
##
##                   Kappa : 0.2779
##
##  Mcnemar's Test P-Value : < 2e-16
##
##             Sensitivity : 0.3235
##             Specificity : 0.9154
##          Pos Pred Value : 0.5543
##          Neg Pred Value : 0.8062
##              Prevalence : 0.2454
##          Detection Rate : 0.0794
##    Detection Prevalence : 0.1433
##       Balanced Accuracy : 0.6195
##
##        'Positive' Class : change
##
```

```r
model.lda$bestTune
```

```
##   parameter
## 1      none
```

```r
set.seed(2)
model.qda = train(x = x_tr_1000,
                  y = y_tr_1000,
                  method = "qda",
                  metric = "ROC",
                  trControl = ctrl)
```

```
## Warning: model fit failed for Fold03: parameter=none Error in qda.default(x, grouping, ...) : rank d

## Warning in nominalTrainWorkflow(x = x, y = y, wts = weights, info = trainInfo, :
## There were missing values in resampled performance measures.
```

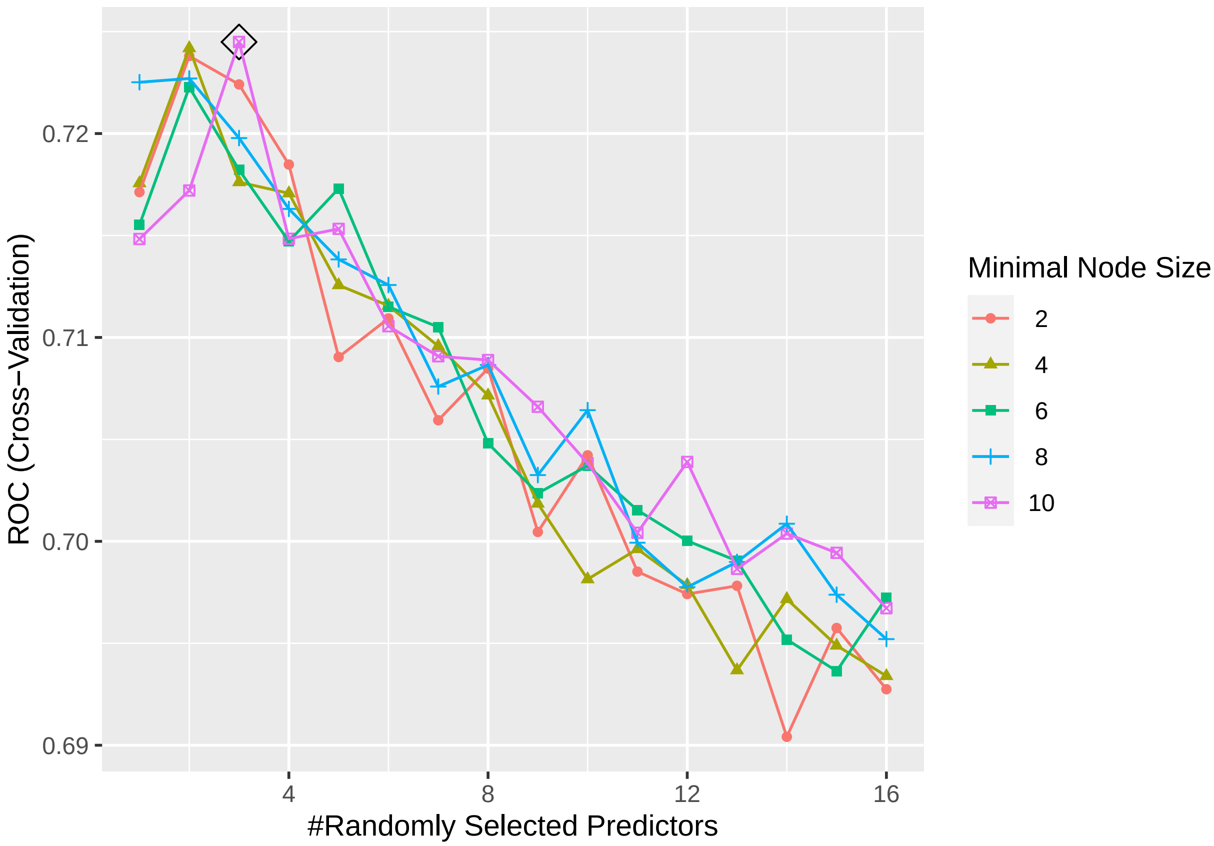
```r
test.pred.prob = predict(model.qda, newdata = x_te, type = "prob")[,1]
test.pred = rep("change", length(test.pred.prob))
test.pred[test.pred.prob<0.5] = "no_change"

confusionMatrix(data = as.factor(test.pred),
                reference = y_te,
                positive = "change")
```

```
## Confusion Matrix and Statistics
##
##            Reference
## Prediction  change no_change
##   change       404       437
##   no_change    480      2281
##
##                Accuracy : 0.7454
##                  95% CI : (0.7309, 0.7596)
##     No Information Rate : 0.7546
##     P-Value [Acc > NIR] : 0.9023
##
##                   Kappa : 0.3012
##
##  Mcnemar's Test P-Value : 0.1655
##
##             Sensitivity : 0.4570
##             Specificity : 0.8392
##          Pos Pred Value : 0.4804
##          Neg Pred Value : 0.8261
##              Prevalence : 0.2454
##          Detection Rate : 0.1122
##    Detection Prevalence : 0.2335
##       Balanced Accuracy : 0.6481
##
##        'Positive' Class : change
##
```

```r
model.qda$bestTune
```

```
##   parameter
## 1      none
```

```
rf.grid <- expand.grid(mtry = 1:16,
                       splitrule = "gini",
                       min.node.size = seq(from = 2, to = 10, by = 2))
set.seed(2)
model.rf <- train(x = x_tr_1000,
                  y = y_tr_1000,
                method = "ranger",
                tuneGrid = rf.grid,
                metric = "ROC",
                trControl = ctrl)

ggplot(model.rf, highlight = TRUE)
```




```
test.pred.prob = predict(model.rf, newdata = x_te, type = "prob")[,1]
test.pred = rep("change", length(test.pred.prob))
test.pred[test.pred.prob<0.5] = "no_change"

confusionMatrix(data = as.factor(test.pred),
                reference = y_te,
                positive = "change")
```

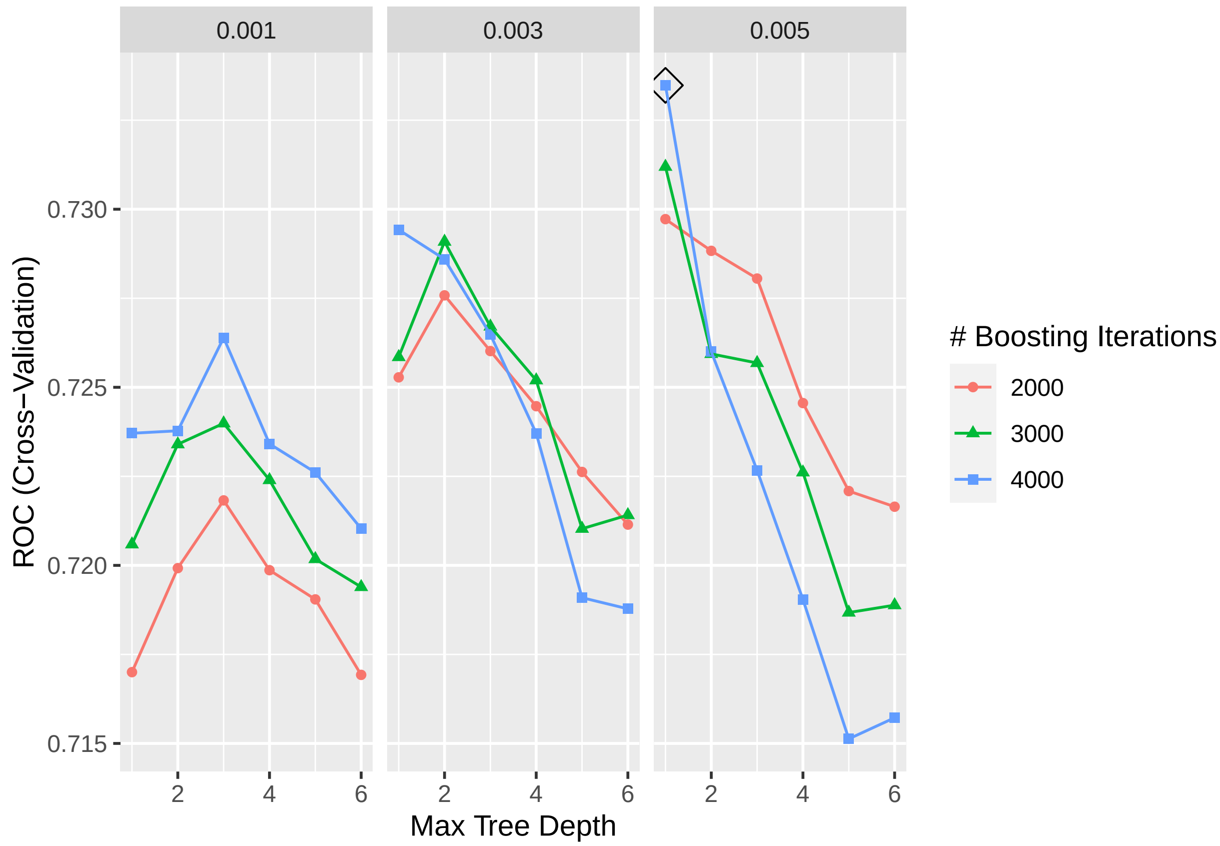
```
## Confusion Matrix and Statistics
##
##             Reference
## Prediction  change no_change
```

```
##   change       211       157
##   no_change    673      2561
##
##                Accuracy : 0.7696
##                  95% CI : (0.7555, 0.7832)
##     No Information Rate : 0.7546
##     P-Value [Acc > NIR] : 0.01864
##
##                   Kappa : 0.2253
##
##  Mcnemar's Test P-Value : < 2e-16
##
##             Sensitivity : 0.23869
##             Specificity : 0.94224
##          Pos Pred Value : 0.57337
##          Neg Pred Value : 0.79190
##              Prevalence : 0.24542
##          Detection Rate : 0.05858
##    Detection Prevalence : 0.10217
##       Balanced Accuracy : 0.59046
##
##        'Positive' Class : change
##
```

```r
model.rf$bestTune
```

```
##    mtry splitrule min.node.size
## 15    3      gini            10
```

```r
gbmA.grid <- expand.grid(n.trees = c(2000,3000,4000),
                         interaction.depth = 1:6,
                         shrinkage = c(0.001,0.003,0.005),
                         n.minobsinnode = 1)
set.seed(2)
model.gbma <- train(x = x_tr_1000,
                  y = y_tr_1000,
                  tuneGrid = gbmA.grid,
                  trControl = ctrl,
                  method = "gbm",
                  distribution = "adaboost",
                  metric = "ROC",
                  verbose = FALSE)

ggplot(model.gbma, highlight = TRUE)
```




```r
test.pred.prob = predict(model.gbma, newdata = x_te, type = "prob")[,1]
test.pred = rep("change", length(test.pred.prob))
test.pred[test.pred.prob<0.5] = "no_change"

confusionMatrix(data = as.factor(test.pred),
                reference = y_te,
                positive = "change")
```

```
## Confusion Matrix and Statistics
##
##            Reference
## Prediction  change no_change
##   change       274       202
##   no_change    610      2516
##
##                Accuracy : 0.7746
##                  95% CI : (0.7606, 0.7881)
##     No Information Rate : 0.7546
##     P-Value [Acc > NIR] : 0.002625
##
##                   Kappa : 0.2791
##
##  Mcnemar's Test P-Value : < 2.2e-16
##
##             Sensitivity : 0.30995
##             Specificity : 0.92568
```

```
##          Pos Pred Value : 0.57563
##          Neg Pred Value : 0.80486
##              Prevalence : 0.24542
##          Detection Rate : 0.07607
##    Detection Prevalence : 0.13215
##       Balanced Accuracy : 0.61782
##
##        'Positive' Class : change
##
```

```
model.gbma$bestTune
```

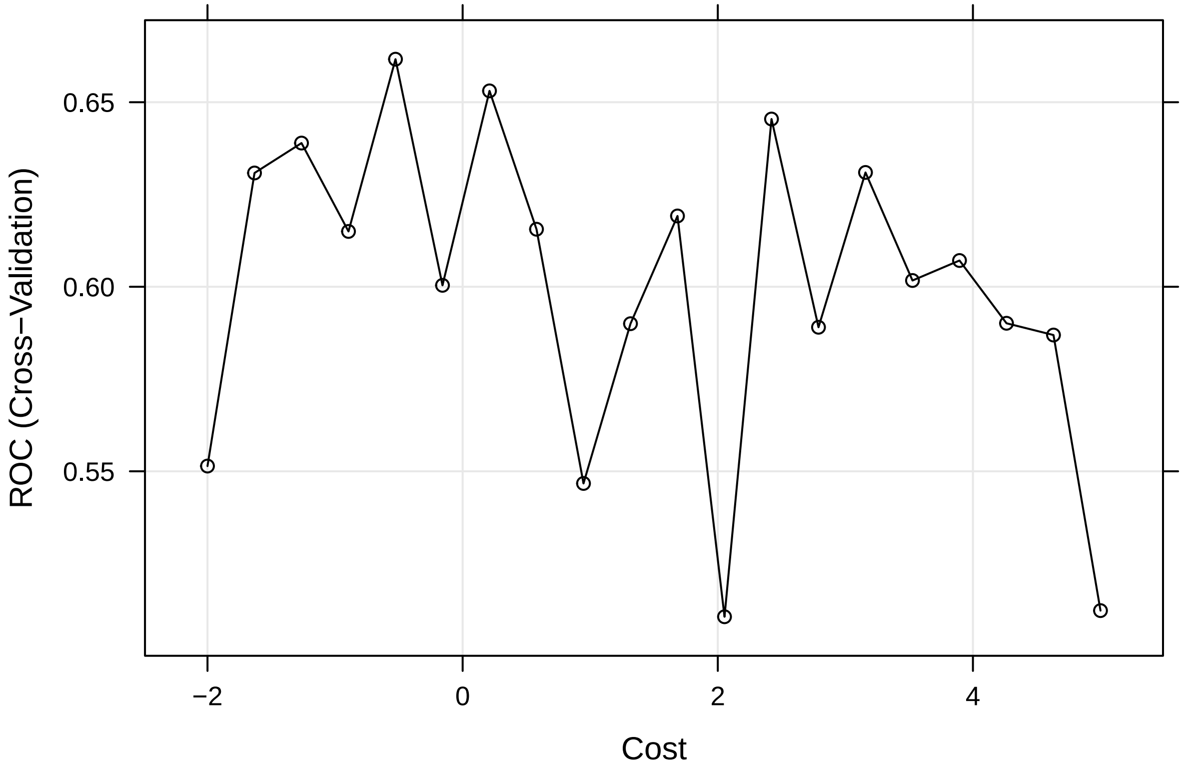
```
##    n.trees interaction.depth shrinkage n.minobsinnode
## 39    4000                 1     0.005              1
```

```
set.seed(2)
model.svml <- train(x = x_tr_1000,
                  y = y_tr_1000,
                  method = "svmLinear",
                  # preProcess = c("center", "scale"),
                  tuneGrid = data.frame(C = exp(seq(-2,5,len=20))),
                  trControl = ctrl)
```

```
## Warning in train.default(x = x_tr_1000, y = y_tr_1000, method = "svmLinear", :
## The metric "Accuracy" was not in the result set. ROC will be used instead.
```

```
## maximum number of iterations reached 0.0003719384 -0.0003684621maximum number of iterations reached (
```

```
plot(model.svml, highlight = TRUE, xTrans = log)
```



```
test.pred.prob = predict(model.svml, newdata = x_te, type = "prob")[,1]
test.pred = rep("change", length(test.pred.prob))
test.pred[test.pred.prob<0.5] = "no_change"

confusionMatrix(data = as.factor(test.pred),
                reference = y_te,
                positive = "change")
```

```
## Warning in confusionMatrix.default(data = as.factor(test.pred), reference =
## y_te, : Levels are not in the same order for reference and data. Refactoring
## data to match.
```

```
## Confusion Matrix and Statistics
##
##            Reference
## Prediction  change no_change
##   change         0         0
##   no_change    884      2718
##
##                Accuracy : 0.7546
##                  95% CI : (0.7402, 0.7686)
##     No Information Rate : 0.7546
##     P-Value [Acc > NIR] : 0.509
##
##                   Kappa : 0
##
##  Mcnemar's Test P-Value : <2e-16
##
##             Sensitivity : 0.0000
##             Specificity : 1.0000
##          Pos Pred Value :    NaN
##          Neg Pred Value : 0.7546
##              Prevalence : 0.2454
##          Detection Rate : 0.0000
##    Detection Prevalence : 0.0000
##       Balanced Accuracy : 0.5000
##
##        'Positive' Class : change
##
```

```
model.svml$bestTune
```

```
##           C
## 5 0.5907775
```

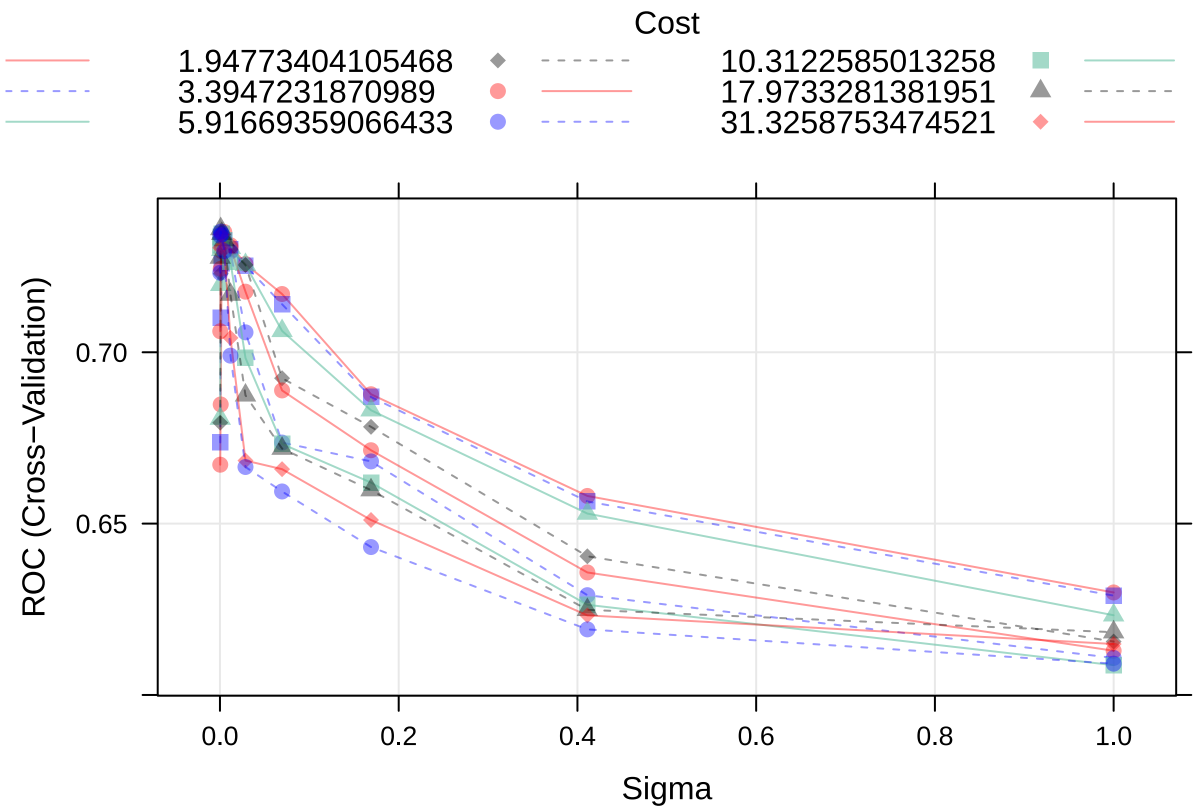
```
svmr.grid <- expand.grid(C = exp(seq(-1,4,len=10)),
                         sigma = exp(seq(-8,0,len=10)))

# tunes over both cost and sigma
set.seed(2)
model.svmr <- train(x = x_tr_1000,
                  y = y_tr_1000,
```

```
                method = "svmRadialSigma",
                preProcess = c("center", "scale"),
                tuneGrid = svmr.grid,
                trControl = ctrl)
```

```
## Warning in train.default(x = x_tr_1000, y = y_tr_1000, method =
## "svmRadialSigma", : The metric "Accuracy" was not in the result set. ROC will be
## used instead.
```

```
## maximum number of iterations reached 0.003271909 -0.003161147maximum number of iterations reached 0.0
```

```
plot(model.svmr, highlight = TRUE)
```




```
test.pred.prob = predict(model.svmr, newdata = x_te, type = "prob")[,1]
test.pred = rep("change", length(test.pred.prob))
test.pred[test.pred.prob<0.5] = "no_change"

confusionMatrix(data = as.factor(test.pred),
                reference = y_te,
                positive = "change")
```

```
## Confusion Matrix and Statistics
##
##             Reference
## Prediction   change no_change
##    change         2         1
```

```
##   no_change    882      2717
##
##                  Accuracy : 0.7549
##                    95% CI : (0.7405, 0.7688)
##     No Information Rate : 0.7546
##     P-Value [Acc > NIR] : 0.4936
##
##                     Kappa : 0.0029
##
##  Mcnemar's Test P-Value : <2e-16
##
##               Sensitivity : 0.0022624
##               Specificity : 0.9996321
##            Pos Pred Value : 0.6666667
##            Neg Pred Value : 0.7549319
##                Prevalence : 0.2454192
##            Detection Rate : 0.0005552
##      Detection Prevalence : 0.0008329
##         Balanced Accuracy : 0.5009473
##
##          'Positive' Class : change
##
```

```
model.svmr$bestTune
```

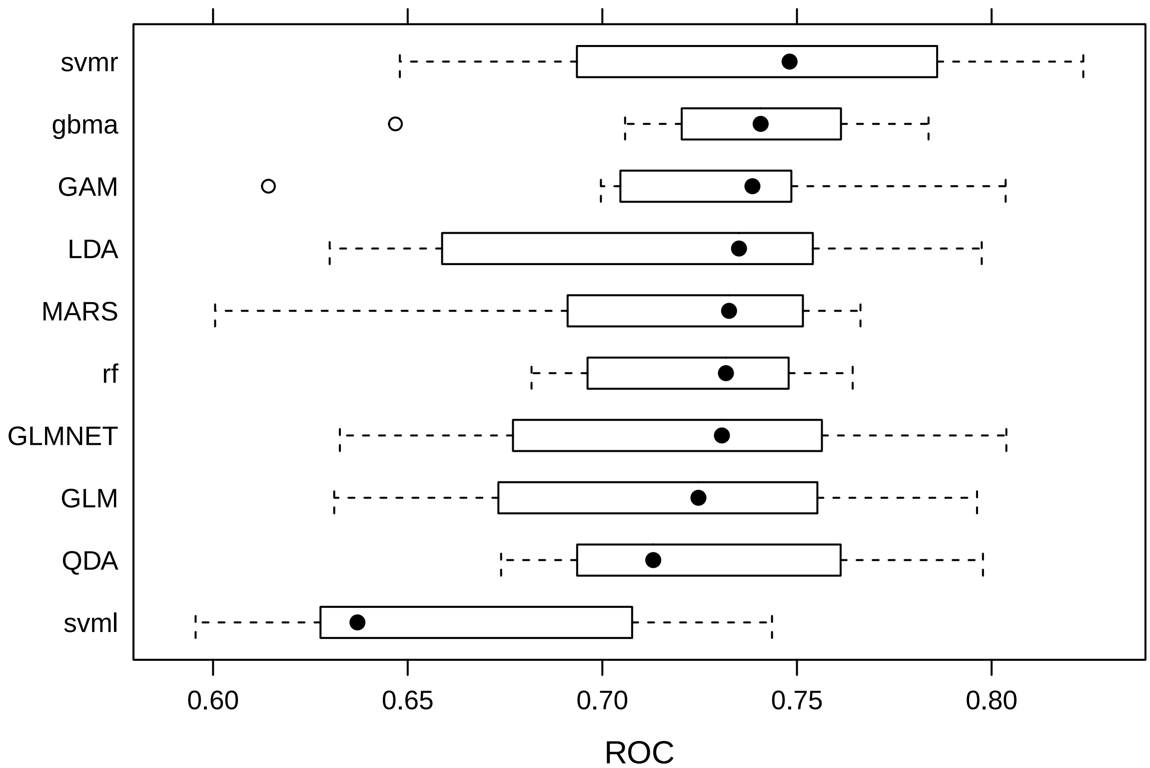
```
##           sigma        C
## 72 0.0008159878 17.97333
```

```
res <- resamples(list(GLM = model.glm,
                      GLMNET = model.glmn,
                      GAM = model.gam,
                      MARS = model.mars,
                      LDA = model.lda,
                      QDA= model.qda,
                      rf = model.rf,
                      gbma = model.gbma,
                      svmr = model.svmr,
                      svml = model.svml))
summary(res)
```

```
##
## Call:
## summary.resamples(object = res)
##
## Models: GLM, GLMNET, GAM, MARS, LDA, QDA, rf, gbma, svmr, svml
## Number of resamples: 10
##
## ROC
##             Min.   1st Qu.    Median      Mean   3rd Qu.      Max. NA's
## GLM    0.6311189 0.6804619 0.7246881 0.7204334 0.7520876 0.7962628    0
## GLMNET 0.6325758 0.6828414 0.7307250 0.7244253 0.7552448 0.8037975    0
## GAM    0.6142191 0.7103677 0.7385663 0.7280193 0.7464029 0.8036131    0
## MARS   0.6005245 0.6964228 0.7325607 0.7172285 0.7507851 0.7663170    0
```

```
## LDA    0.6299534 0.6668790 0.7351088 0.7195657 0.7540515 0.7974684    0
## QDA    0.6739927 0.6935287 0.7130802 0.7266508 0.7612198 0.7977855    1
## rf     0.6818182 0.6995109 0.7317487 0.7244915 0.7456294 0.7643159    0
## gbma   0.6468531 0.7218235 0.7406725 0.7334777 0.7568223 0.7837995    0
## svmr   0.6479807 0.6939449 0.7481116 0.7359801 0.7798373 0.8235653    0
## svml   0.5955121 0.6288156 0.6370886 0.6616542 0.7011960 0.7435897    0
##
## Sens
##             Min.   1st Qu.    Median       Mean    3rd Qu.      Max. NA's
## GLM    0.1818182 0.1996753 0.2326840 0.24675325 0.27272727 0.3809524    0
## GLMNET 0.1818182 0.1904762 0.2326840 0.23744589 0.26406926 0.3809524    0
## GAM    0.1818182 0.2759740 0.3257576 0.31731602 0.35606061 0.4285714    0
## MARS   0.1818182 0.2727273 0.3019481 0.31233766 0.35606061 0.4285714    0
## LDA    0.2272727 0.2938312 0.3484848 0.34480519 0.40205628 0.4285714    0
## QDA    0.3809524 0.5000000 0.5238095 0.53246753 0.59090909 0.6363636    1
## rf     0.1363636 0.1996753 0.2326840 0.24718615 0.27380952 0.4285714    0
## gbma   0.2272727 0.2759740 0.3409091 0.32597403 0.37662338 0.4285714    0
## svmr   0.0000000 0.0000000 0.0000000 0.04675325 0.08333333 0.1818182    0
## svml   0.0000000 0.0000000 0.0000000 0.00000000 0.00000000 0.0000000    0
##
## Spec
##             Min.   1st Qu.    Median      Mean   3rd Qu.      Max. NA's
## GLM    0.8974359 0.9361003 0.9490425 0.9476793 0.9619036 0.9873418    0
## GLMNET 0.9102564 0.9391026 0.9554528 0.9514930 0.9620253 0.9873418    0
## GAM    0.9102564 0.9168695 0.9491237 0.9451477 0.9712756 0.9746835    0
## MARS   0.9102564 0.9230769 0.9299740 0.9337228 0.9492048 0.9620253    0
## LDA    0.8974359 0.9137050 0.9427134 0.9336904 0.9493671 0.9620253    0
## QDA    0.7820513 0.8076923 0.8333333 0.8412132 0.8717949 0.9240506    1
## rf     0.9358974 0.9488802 0.9554528 0.9553716 0.9619036 0.9746835    0
## gbma   0.9113924 0.9230769 0.9363843 0.9388348 0.9493671 0.9746835    0
## svmr   0.9615385 0.9873418 1.0000000 0.9910906 1.0000000 1.0000000    0
## svml   1.0000000 1.0000000 1.0000000 1.0000000 1.0000000 1.0000000    0
```

```
bwplot(res, metric = "ROC")
```



Predictors

I included all the 11 predictors – 2 continuous variables (city_development_index and training_hours) and 8 catgorical variables(gender,relevant_experience,enrolled_university,education_level,major_discipline,experience,company_s to built models.
In the model building precess, I built 6 models: a logistic regression model, a penalized logistic regression model, a GAM model, a MARS model , a LDA model and a QDA model. I used caret to train all the six models and then made the comparison.

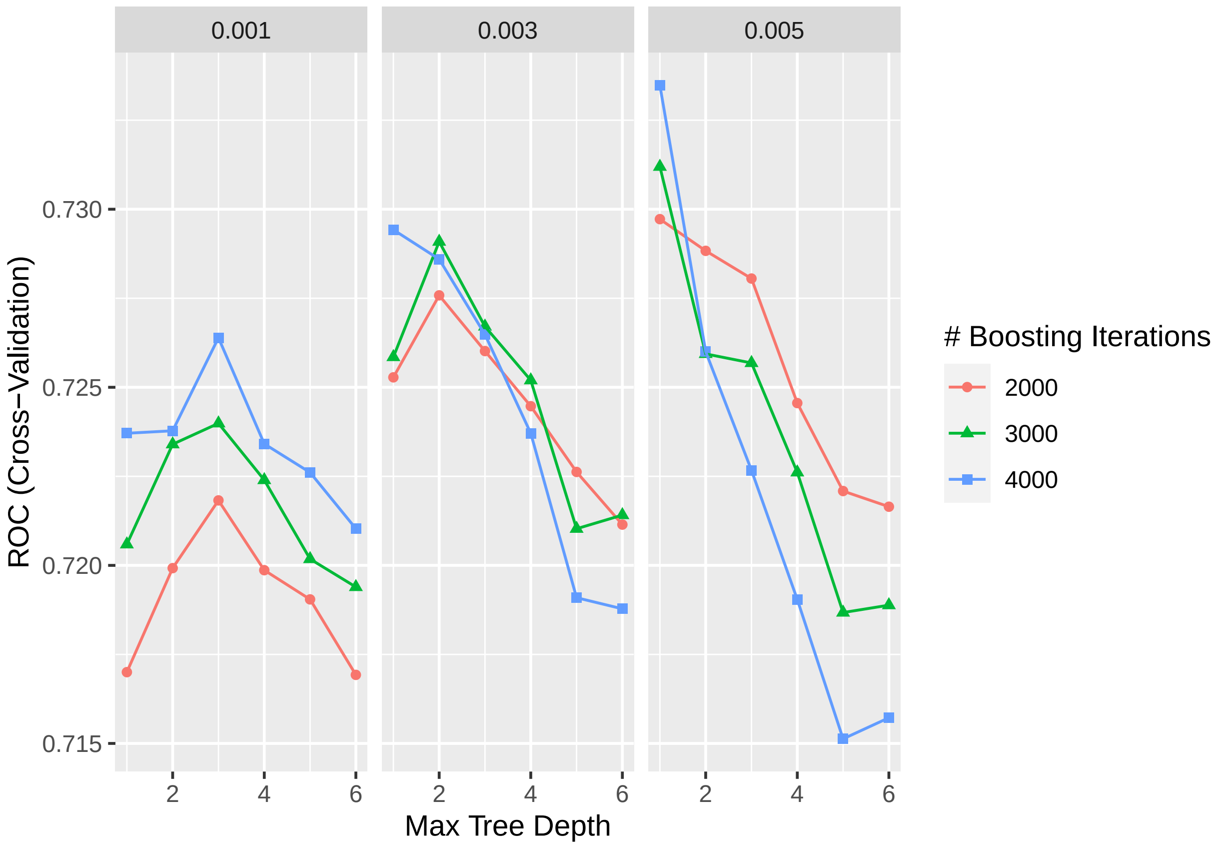
Technique

According to the result[figure3], the MARS model has the largest AUC, and thus became the final model I choose.

Tuning parameters

There are two tuning parameters associated with the MARS model: the degree of interactions and the number of retained terms. I performed a grid(degree = 1:3,nprune = 2:15) search to identify the optimal combination of these hyperparameters that minimize prediction error. According to the result of cross validation[figure4], the best combination of tuning parameter would be:
degree of interaction:1
number of retained terms: 10

```
ggplot(model.gbma)
```




```r
model.gbma$bestTune
```

```
##    n.trees interaction.depth shrinkage n.minobsinnode
## 39    4000                 1     0.005              1
```

```r
coef(model.gbma$finalModel)
```

```
## NULL
```

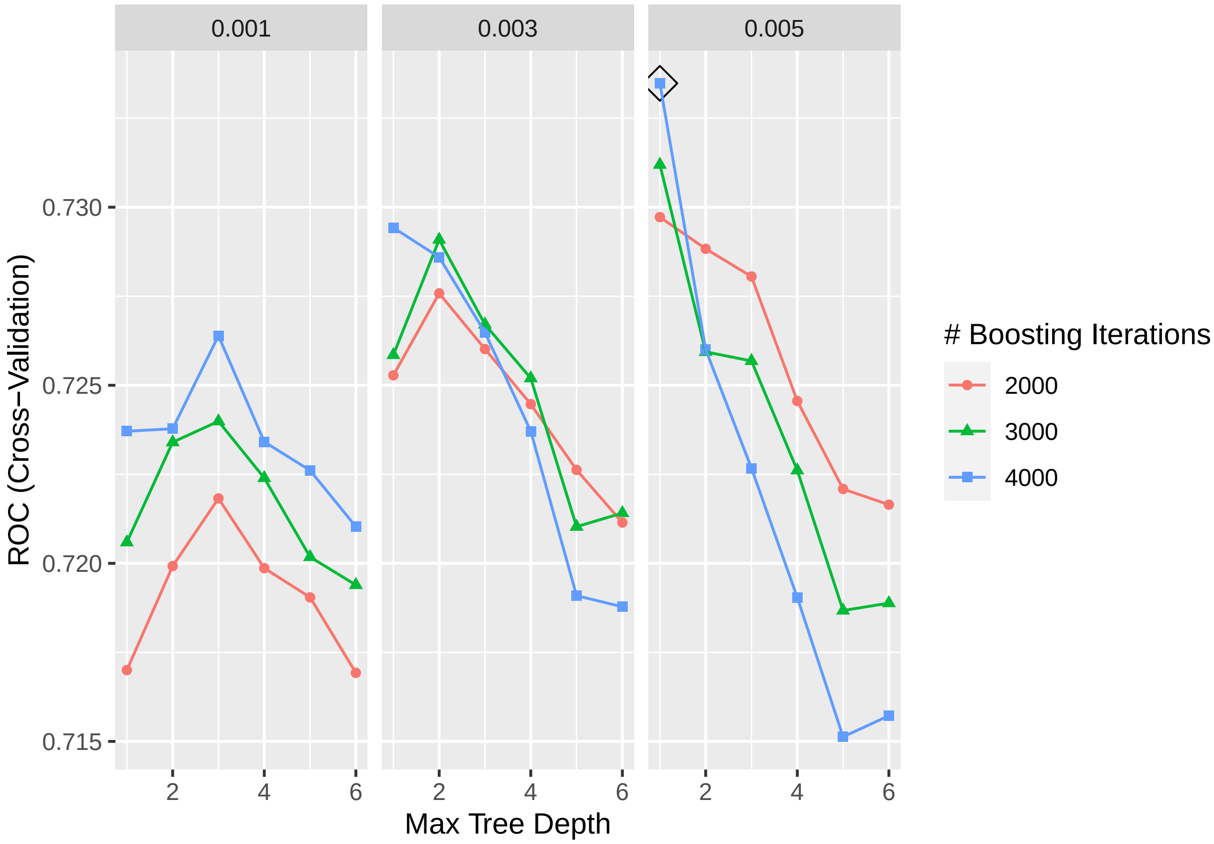
Trainig/Testing performamce

For test dataset, I used the MARS model to make prediction on test dataset, then made a confusion matrix based on the predicted value and the true value. According to the ROC curve[figure5], the AUC for test data is 0.7645, the overall accuracy is 78.04%, and the Kappa is 0.3393, which had a moderate performance. The sensitivity is 0.40045 and the specificity is 0.90397. The training dataset, according to the result of resampling, has mean AUC 0.7594879, mean sensitivity 0.3910189 and mean specificity 0.9101620.

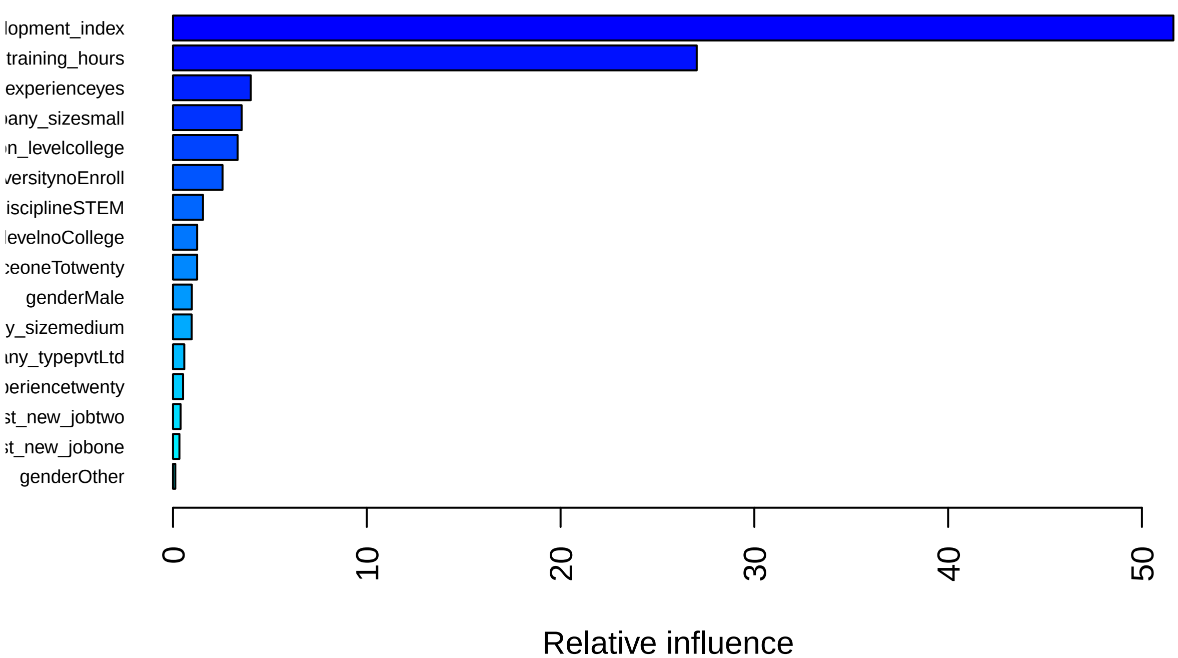
Important variables

I used vip function to find the important variables. According to the result, the most important variable is "city_development_index", which aligned with the previous finding in visualization. Followed city development index is the relevant experiences. Besides, the vip result also showed that education_level:College and enrolled_university:enrolled also play important roles in predicting the outcome.

```r
ggplot(model.gbma, highlight = TRUE)
```




```r
summary(model.gbma$finalModel, las = 2, cBars = 19, cex.names = 0.6)
```




```
##                                                         var     rel.inf
```

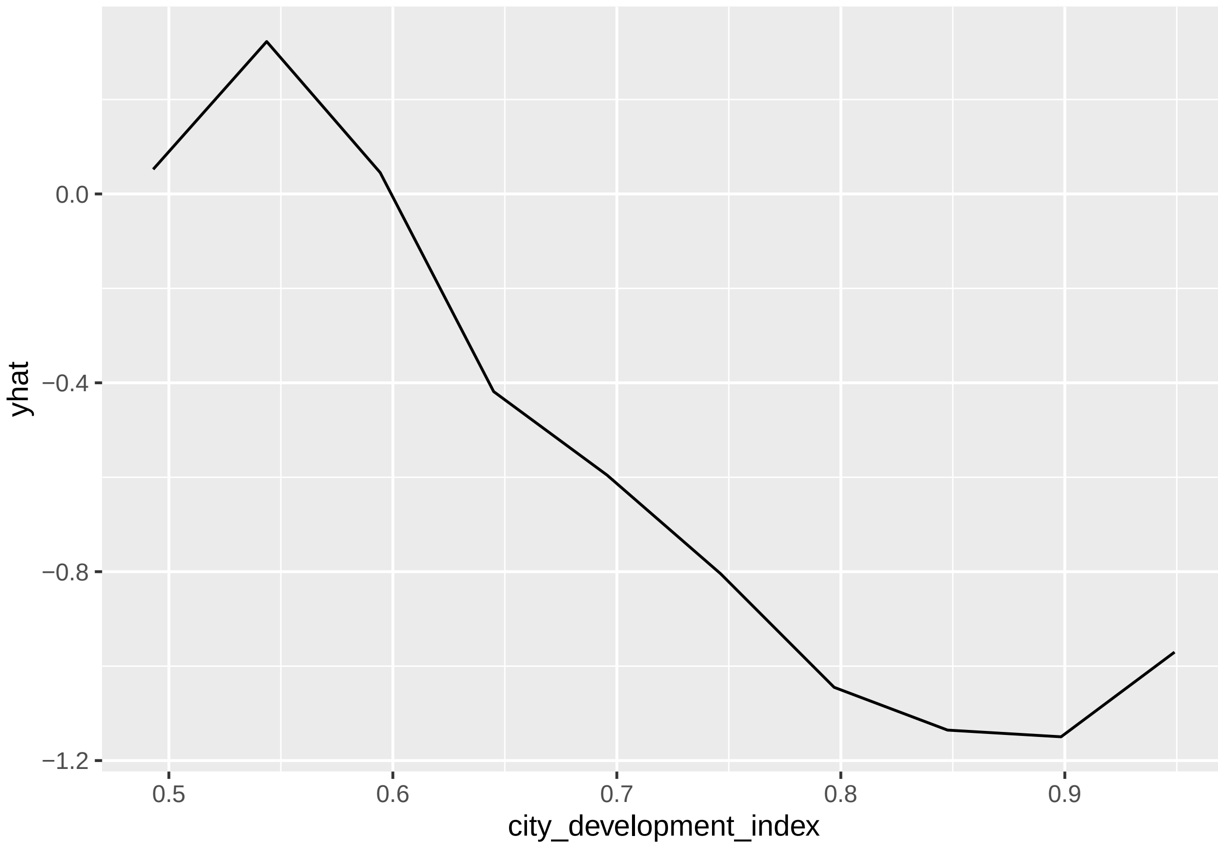
```
## city_development_index           city_development_index 51.6238898
## training_hours                           training_hours 27.0252057
## relevant_experienceyes           relevant_experienceyes  4.0098709
## company_sizesmall                     company_sizesmall  3.5424685
## education_levelcollege           education_levelcollege  3.3312991
## enrolled_universitynoEnroll enrolled_universitynoEnroll  2.5554644
## major_disciplineSTEM               major_disciplineSTEM  1.5470739
## education_levelnoCollege       education_levelnoCollege  1.2458086
## experienceoneTotwenty             experienceoneTotwenty  1.2439092
## genderMale                                   genderMale  0.9711265
## company_sizemedium                   company_sizemedium  0.9625830
## company_typepvtLtd                   company_typepvtLtd  0.5824331
## experiencetwenty                       experiencetwenty  0.5204806
## last_new_jobtwo                         last_new_jobtwo  0.3899711
## last_new_jobone                         last_new_jobone  0.3297607
## genderOther                                 genderOther  0.1186551
```

Partial Dependence Plot

```
p1 <- pdp::partial(model.gbma, pred.var = c("city_development_index"), grid.resolution = 10) %>% autopl
 p1
```

```
## Warning: Use of `object[[1L]]` is discouraged. Use `.data[[1L]]` instead.
```

```
## Warning: Use of `object[["yhat"]]` is discouraged. Use `.data[["yhat"]]`
## instead.
```



Individual conditional expectation curve

```
ice1.gbma <- model.gbma %>%
  partial(pred.var = "city_development_index",
          grid.resolution = 100,
          ice = TRUE) %>%
  autoplot(train = x_tr_1000, alpha = .1) +
  ggtitle("ICE, not centered")
```

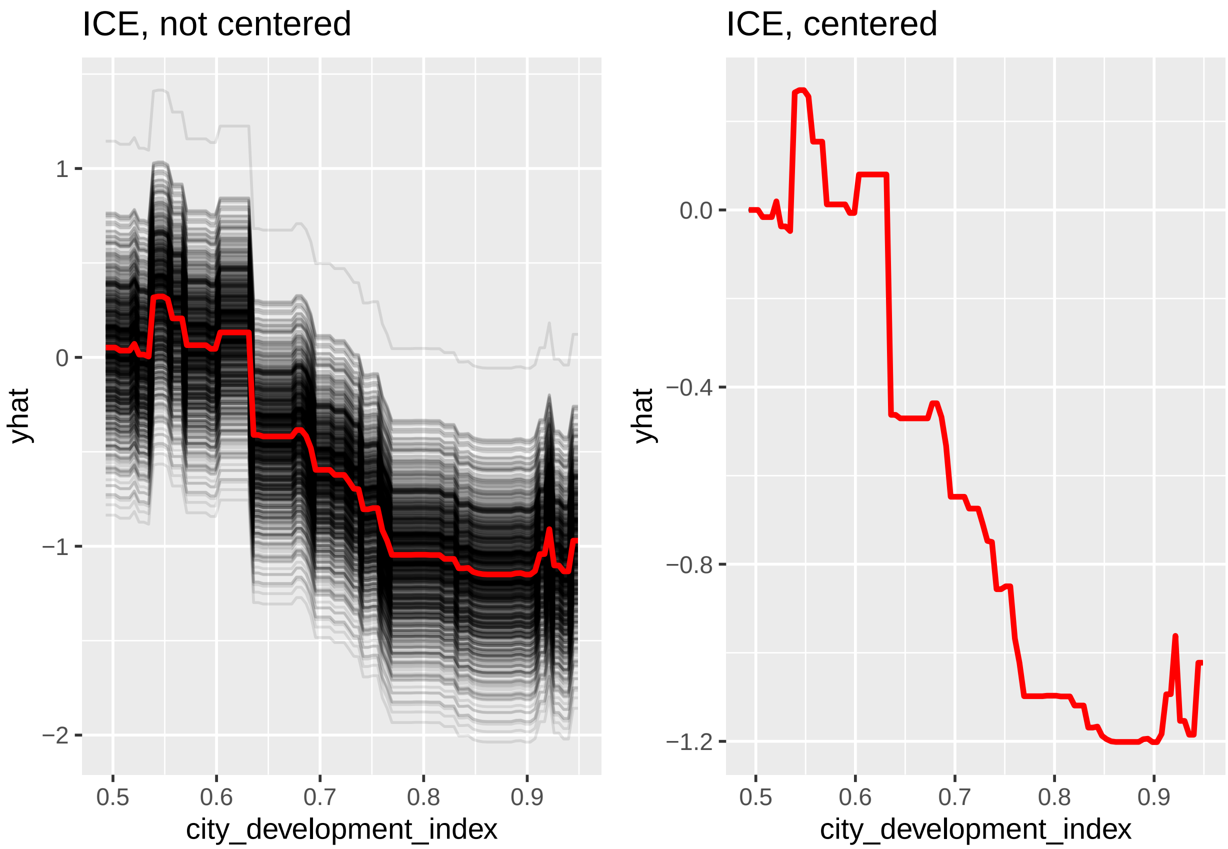
```
## Warning: `fun.y` is deprecated. Use `fun` instead.
```

```
ice2.gbma <- model.gbma %>%
  partial(pred.var = "city_development_index",
          grid.resolution = 100,
          ice = TRUE) %>%
  autoplot(train = x_tr_1000, alpha = .1,
           center = TRUE) +
  ggtitle("ICE, centered")
```

```
## Warning: `fun.y` is deprecated. Use `fun` instead.
```

```
grid.arrange(ice1.gbma, ice2.gbma, nrow = 1)
```

```
## Warning: Use of `object[["yhat.id"]]` is discouraged. Use `.data[["yhat.id"]]`
## instead.

## Warning: Use of `object[[1L]]` is discouraged. Use `.data[[1L]]` instead.

## Warning: Use of `object[["yhat"]]` is discouraged. Use `.data[["yhat"]]`
## instead.

## Warning: Use of `object[[1L]]` is discouraged. Use `.data[[1L]]` instead.

## Warning: Use of `object[["yhat"]]` is discouraged. Use `.data[["yhat"]]`
## instead.

## Warning: Use of `object[["yhat.id"]]` is discouraged. Use `.data[["yhat.id"]]`
## instead.

## Warning: Use of `object[[1L]]` is discouraged. Use `.data[[1L]]` instead.

## Warning: Use of `object[["yhat"]]` is discouraged. Use `.data[["yhat"]]`
## instead.

## Warning: Use of `object[[1L]]` is discouraged. Use `.data[[1L]]` instead.

## Warning: Use of `object[["yhat"]]` is discouraged. Use `.data[["yhat"]]`
## instead.
```



Limitation of the model

I think one of the problem of build a MARS model is the speed. During the model built process, the MARS model need the longest time to train.
Besides speed, there is also the problem of global optimization vs. local optimization. The fitting process for MARS regression is done in a stepwise greedy manner. That way, only the best basis function given the current model is added/removed. So the model could be inaccurate if the local linear relationships are incorrect.

Flexibility

I think the model is flexible enough to capture the underlying truth.

Conclusions

According to the model, people have relevant experience in data science field, who has less than college education are more likely to change job. According to the MARS model, city development index is the most important predictors for predicting whether the person want a new job or not(target).To better understand the relatioship between the features and the target, I created partial dependence plots for city_development_index. This is used to examine the marginal effects of predictors.

According to the plot, people live in the city with higher development index are more likely to change job, which make sense–high developed cities usually have more opportunities and challenges. People struggling in these kind of cities are mostly younger people who always seeking for better opportunities, and also they are more adaptable to changes.