Financial Econometric in R/Python
Assignment Two

Group 2
Hessa Alabbas 02513615
Xin Zhan 02299544
Alex Rached 01894052
Yan Cai 02381303
Kexin Liu 02362049

The Business School, Imperial College London

18-11-2023

Contents

List of Tables

Introduction

Load Packages

```
library(lubridate)
library(knitr)
library(dplyr)
library(lmtest)
library(readxl)
library(moments)
library(sandwich)
library(estimatr)
library(margins)
library(randomForest)
library(e1071)
library(MASS)
library(class)
library(lmtest)
library(tree)
library(knitr)
library(rpart)
library(ROSE)
```

Load Data

```
data <-  read_excel('employment_08_09.xlsx')
```

Question A

What fraction of workers in the sample were employed in April 2009? Use your answer to compute a 95% confidence interval for the probability that a worker was employed in April 2009, conditional on being employed in April 2008.

```
#sample size
n <- nrow(data)

#employed fraction
p <- sum(data$employed == 1) / n
cat("Fraction of workers employed in April 2009:", p, "\n")
```

```
## Fraction of workers employed in April 2009: 0.8754619
```

```
#margin of error
margin <- qnorm(0.975)*sqrt(p*(1-p)/n)

#lower and upper intervals
lowerinterval <- p - margin
```

```r
upperinterval <- p + margin

cat(lowerinterval,upperinterval)
```

```
## 0.8666648 0.884259
```

The fraction of workers in the sample is 0.8754619. The interpretation is that, based on the sample data and statistical analysis, we are 95% confident that the true probability of a worker being employed in April 2009, given they were employed in April 2008, lies between 0.8666648 and 0.884259.

Question B

Regress Employed on Age and Age**2 , using a linear probability model.

```r
binary_lm <- lm(employed ~ age + I(age ^ 2), data = data)
summary(binary_lm)
```

```
Call:
lm(formula = employed ~ age + I(age^2), data = data)

Residuals:
     Min       1Q   Median       3Q      Max 
-0.91925  0.08314  0.10020  0.13831  0.28944 

Coefficients:
              Estimate Std. Error t value Pr(>|t|)    
(Intercept)  3.075e-01  5.472e-02   5.619 2.02e-08 ***
age          2.827e-02  2.747e-03  10.293  < 2e-16 ***
I(age^2)    -3.266e-04  3.276e-05  -9.971  < 2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.327 on 5409 degrees of freedom
Multiple R-squared:  0.01966,	Adjusted R-squared:  0.01929 
F-statistic: 54.22 on 2 and 5409 DF,  p-value: < 2.2e-16
```

```r
coefest_df <- coeftest(binary_lm, vcov = vcovHC(binary_lm), type = "HC1")
```

i)

Based on this regression, was the age a statistically significant determinant of employment in April 2009.

The positive coefficient for the ‘Age’ variable suggests that, on average, the probability of being employed increases with age. The coefficient is statistically significant with a p-value < 0.001, so there is a statistically significant relationship between age and employment in April 2009. Although it is statistically significant, the overall fit of the model (Multiple R-squared and Adjusted R-squared) indicates that age explains a very small proportion of the variability in employment status. In this case, only about 1.966% of the variability in employment status is explained by age and its squared term. The low p-value (< 2.2e-16) indicates that at least one of the predictors (age or age^2) is related to the dependent variable.

ii)

Is there evidence of a nonlinear effect of age on probability of being employed?

Yes, there is evidence of a nonlinear effect of age on the probability of being employed based on the regression results. The negative coefficient for the squared term 'Age^2' is also statistically significant (p-value < 0.001). This suggests that as age increases, the positive effect of age on the probability of being employed diminishes, indicating a curvature or nonlinear pattern in the relationship.

iii)

Compute the predicted probability of employment for a 20-year-old worker, a 40year-old worker, and a 60-year-old worker.

```
predicted_probabilities <- predict(binary_lm,
                                   newdata = data.frame(age = c(20,40,60)),
                                   type = "response")

print(predicted_probabilities)
```

```
##         1         2         3 
## 0.7422841 0.9157685 0.8279458
```

The predicted probability of employment for a 20-year-old worker is approximately 74.23%. The predicted probability of employment for a 40-year-old worker is approximately 91.58%. The predicted probability of employment for a 60-year-old worker is approximately 82.79%.

Question C

Repeat (b) using a probit regression.

```
binary_probit <- glm(employed ~ age + I(age ^ 2),
                     family = binomial(link = "probit"),
                     data)
summary(binary_probit)
```

```
## 
## Call:
## glm(formula = employed ~ age + I(age^2), family = binomial(link = "probit"), 
##     data = data)
## 
## Coefficients:
##               Estimate Std. Error z value Pr(>|z|)    
## (Intercept) -1.2579285  0.2466845  -5.099 3.41e-07 ***
## age          0.1217230  0.0126633   9.612  < 2e-16 ***
## I(age^2)    -0.0014125  0.0001522  -9.279  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## 
## (Dispersion parameter for binomial family taken to be 1)
```

```
##
##     Null deviance: 4068.4  on 5411  degrees of freedom
## Residual deviance: 3973.8  on 5409  degrees of freedom
## AIC: 3979.8
##
## Number of Fisher Scoring iterations: 4
```

```
coeftest(binary_probit, vcov = vcovHC(binary_probit), type = "HC1")
```

```
##
## z test of coefficients:
##
##                Estimate  Std. Error z value  Pr(>|z|)
## (Intercept) -1.25792851  0.25246321 -4.9826 6.273e-07 ***
## age          0.12172302  0.01306499  9.3167 < 2.2e-16 ***
## I(age^2)    -0.00141246  0.00015776 -8.9530 < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

The positive coefficient for 'age' (0.1217230) indicates that, on average, the log-odds of being employed increase with age. The negative coefficient for the squared term 'I(age^2)' (-0.0014125) suggests a nonlinear effect. As 'age' increases, the positive effect on the log-odds of being employed diminishes. The z-tests show that all coefficients are statistically significant at the 0.05 significance level, indicating that age and its squared term are significantly related to the log-odds of being employed.

i)

Based on this regression, was the age a statistically significant determinant of employment in April 2009.

Yes, based on the results of the probit regression model, age appears to be a statistically significant determinant of employment in April 2009. The positive coefficient for the 'age' variable suggests that, on average, the log-odds of being employed increase with age. This coefficient is statistically significant with a very low p-value (< 0.001), indicating strong evidence that the effect of age on employment is different from zero.

The low p-values indicate that both 'age' and 'age^2' are highly unlikely to have coefficients equal to zero, suggesting that both linear and nonlinear components of age are important in determining employment status.

ii)

Is there evidence of a nonlinear effect of age on probability of being employed?

The negative coefficient for the squared term 'I(age^2)' (-0.0014) suggests a nonlinear effect. Specifically, as 'age' increases, the positive effect on the log-odds of being employed diminishes. This coefficient is also statistically significant with a very low p-value (< 0.001).

iii)

Compute the predicted probability of employment for a 20-year-old worker, a 40-year-old worker, and a 60-year-old worker.

```
predicted_probabilities_probit <- predict(binary_probit,
                                          newdata = data.frame(age = c(20,40,60)),
                                          type = "response")

print(predicted_probabilities_probit)
```

```
##         1         2         3 
## 0.7295817 0.9116616 0.8316237
```

The predicted probability of employment for a 20-year-old worker is approximately 72.96%. The predicted probability of employment for a 40-year-old worker is approximately 91.17%. The predicted probability of employment for a 60-year-old worker is approximately 83.16%.

Question D

Repeat (b) using a logit regression.

```
binary_logit <- glm(employed ~ age + I(age ^ 2),
                    family = binomial(link = "logit"),
                    data)
summary(binary_logit)
```

```
## 
## Call:
## glm(formula = employed ~ age + I(age^2), family = binomial(link = "logit"), 
##     data = data)
## 
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)    
## (Intercept) -2.4897541  0.4373964  -5.692 1.25e-08 ***
## age          0.2254662  0.0228093   9.885  < 2e-16 ***
## I(age^2)    -0.0026237  0.0002757  -9.518  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## 
## (Dispersion parameter for binomial family taken to be 1)
## 
##     Null deviance: 4068.4  on 5411  degrees of freedom
## Residual deviance: 3972.9  on 5409  degrees of freedom
## AIC: 3978.9
## 
## Number of Fisher Scoring iterations: 4
```

```
coeftest(binary_logit, vcov = vcovHC(binary_logit), type = "HC1")
```

```
## 
## z test of coefficients:
## 
##                Estimate  Std. Error z value  Pr(>|z|)    
## (Intercept) -2.48975412  0.44690359 -5.5711 2.531e-08 ***
## age           0.22546624  0.02348717  9.5995 < 2.2e-16 ***
## I(age^2)     -0.00262366  0.00028515 -9.2010 < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

i)

Based on this regression, was the age a statistically significant determinant of employment in April 2009.

Yes, age was a statistically significant determinant of employment in April 2009, as evidenced by the low p-values for both the ‘Age’ and ‘I(age^2)’ coefficients. The positive coefficient for ‘Age’ indicates a positive linear relationship. These findings highlight the importance of considering age as a factor influencing employment outcomes during the specified period. The residual deviance is 3972.9 on 5409 degrees of freedom, indicating a reasonable fit of the model to the data.

ii)

Is there evidence of a nonlinear effect of age on probability of being employed?

Yes, there is evidence of a nonlinear effect of age on the probability of being employed, as indicated by the coefficient for the quadratic term ‘I(age^2)’ in the logistic regression model.

The coefficient for ‘I(age^2)’ is estimated to be -0.0026, and its associated p-value is < 2e-16, which is highly statistically significant. This implies that the relationship between age and the log-odds of employment is not purely linear but involves a quadratic component. In other words, the impact of age on employment probability is not constant; it changes nonlinearly with age. This finding underscores the importance of considering not only the linear effect of age but also its quadratic effect when modeling employment outcomes.

iii)

Compute the predicted probability of employment for a 20-year-old worker, a 40-year-old worker, and a 60-year-old worker.

```r
predicted_probabilities_logit <- predict(binary_logit,
                                         newdata = data.frame(age = c(20,40,60)),
                                         type = "response")
print(predicted_probabilities_logit)
```

```
##         1         2         3
## 0.7251410 0.9114157 0.8310454
```

The predicted probability of employment for a 20-year-old worker is approximately 72.51%. The predicted probability of employment for a 40-year-old worker is approximately 91.14%. The predicted probability of employment for a 60-year-old worker is approximately 83.10%.

Question E

Are there important differences in your answers to (b)-(d)? Explain.

The estimated coefficients and standard errors of each independent variables (age and age^2) and associated intercepts are highest in logit regression and lowest in linear probability model. In addition, the estimated coefficient of intercept in linear probability model is positive, while negative for logit and probit model.

The predicted probability for employed for a 20-year-old and 40-year-old worker is highest in linear model, followed by logit and probit. The predicted probability for employed for a 60-year-old worker is highest in logit model, followed by linear and probit.

Coefficients and standard errors and predicted probabilities differ among models because of the different functional forms, including linear, logit and probit in our case.

Question F

```
# we_state and educ_adv are deleted for collinearity and
# na number in earnwke are dropped as well
binary_lpm_modified <- lm(employed ~ age + I(age^2)+as.factor(race)+earnwke+
                            married+ne_states+so_states+ce_states+we_states+
                            educ_lths+ educ_hs+educ_somecol+educ_aa+educ_bac+
                            educ_adv+female,
                        data,na.action = "na.omit")
summary(binary_lpm_modified)
```

```
##
## Call:
## lm(formula = employed ~ age + I(age^2) + as.factor(race) + earnwke +
##     married + ne_states + so_states + ce_states + we_states +
##     educ_lths + educ_hs + educ_somecol + educ_aa + educ_bac +
##     educ_adv + female, data = data, na.action = "na.omit")
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.99186  0.06504  0.10404  0.14595  0.38047
##
## Coefficients: (2 not defined because of singularities)
##                    Estimate Std. Error t value Pr(>|t|)
## (Intercept)       3.524e-01  6.116e-02   5.762 8.84e-09 ***
## age               2.487e-02  3.043e-03   8.170 3.91e-16 ***
## I(age^2)         -2.900e-04  3.613e-05  -8.026 1.26e-15 ***
## as.factor(race)2 -3.842e-02  1.700e-02  -2.260 0.023854 *
## as.factor(race)3 -4.750e-03  1.864e-02  -0.255 0.798879
## earnwke           3.406e-05  9.732e-06   3.499 0.000470 ***
## married          -2.610e-03  1.046e-02  -0.249 0.803012
## ne_states         1.676e-02  1.427e-02   1.174 0.240283
## so_states         2.395e-02  1.331e-02   1.799 0.072107 .
## ce_states         4.392e-02  1.368e-02   3.211 0.001334 **
## we_states                NA         NA      NA       NA
## educ_lths        -8.236e-02  2.399e-02  -3.433 0.000601 ***
## educ_hs          -2.082e-02  1.736e-02  -1.200 0.230302
## educ_somecol      2.405e-04  1.816e-02   0.013 0.989438
## educ_aa           7.347e-03  2.010e-02   0.366 0.714682
## educ_bac         -1.284e-02  1.702e-02  -0.755 0.450565
## educ_adv                 NA         NA      NA       NA
## female           -4.849e-03  9.971e-03  -0.486 0.626739
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.3257 on 4757 degrees of freedom
##   (639 observations deleted due to missingness)
## Multiple R-squared:  0.03231,    Adjusted R-squared:  0.02926
## F-statistic: 10.59 on 15 and 4757 DF,  p-value: < 2.2e-16
```

```
coeftest(binary_lpm_modified, vcov = vcovHC(binary_lpm_modified), type = "HC1")
```

```
##
## t test of coefficients:
##
##                      Estimate  Std. Error t value  Pr(>|t|)
## (Intercept)        3.5240e-01  7.2419e-02  4.8661 1.175e-06 ***
## age                2.4866e-02  3.5748e-03  6.9560 3.977e-12 ***
## I(age^2)          -2.8996e-04  4.2178e-05 -6.8746 7.021e-12 ***
## as.factor(race)2 -3.8416e-02  1.8713e-02 -2.0529 0.0401368 *
## as.factor(race)3 -4.7501e-03  1.9209e-02 -0.2473 0.8046987
## earnwke            3.4057e-05  9.5945e-06  3.5496 0.0003895 ***
## married           -2.6098e-03  1.0509e-02 -0.2483 0.8038795
## ne_states          1.6760e-02  1.4487e-02  1.1570 0.2473497
## so_states          2.3948e-02  1.3615e-02  1.7590 0.0786501 .
## ce_states          4.3921e-02  1.3551e-02  3.2412 0.0011987 **
## educ_lths         -8.2361e-02  2.7005e-02 -3.0499 0.0023018 **
## educ_hs           -2.0822e-02  1.6573e-02 -1.2564 0.2090417
## educ_somecol       2.4045e-04  1.6980e-02  0.0142 0.9887021
## educ_aa            7.3469e-03  1.8034e-02  0.4074 0.6837357
## educ_bac          -1.2841e-02  1.5348e-02 -0.8367 0.4028313
## female            -4.8493e-03  1.0061e-02 -0.4820 0.6298405
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

i)

By adding those covariates to the linear probability model regression of point (b), investigate whether the conclusions on the effect of Age on employment from (b) are affected by omitted variable bias.

The impact of age on employment is susceptible to omitted variable bias, a phenomenon characterized by two essential conditions: the independent variables must be correlated with the omitted variable, and the omitted variable must be a determinant of the dependent variable.

In our analysis, we try to address omitted variable bias by incorporating variables representing workers' educational attainment, gender, race, marital status, region of residence, and weekly earnings. From the regression, we find the coefficients for ‘earnwke,’ ‘ce_states,’ ‘educ_lths,’ and ’ as.factor(race)2,’ are all found to be statistically significant at the 1% level.

However, to discuss potential omitted variables, we focus on following ones.

The variable ‘earnwke,’ which captures average weekly earnings, satisfies both conditions for omitted variable bias. Firstly, average weekly earnings exhibit correlation with age, given that the elderly population typically earns less than their younger counterparts, owing to factors such as technological changes and physical limitations. Secondly, ‘earnwke’ serves as a determinant of employment, as individuals with lower salaries are more prone to job displacement. Consequently, the coefficients of age may experience downward bias due to the positive correlation between employment and 'earnwke,' and the negative correlation between 'earnwke' and age.

Similarly, the variable 'educ_lths,' indicating whether a worker's highest level of education is less than a high school graduate, satisfies both conditions. Firstly, education levels correlate with age, as the elderly tend to have lower educational exposure than younger groups. Secondly, education levels influence salary to some extent, reflecting increasing market demands for skilled workers. Consequently, the coefficients of age may suffer from downward bias due to the negative correlation between employment and 'educ_lths,' and the positive correlation between 'educ_lths' and age.

Moreover, the observed decrease in the magnitude (ignoring sign) of the coefficients for age and age^2 further validates the presence of downward bias resulting from omitted variables.

ii)

Use the regression results to discuss the characteristics of workers who were hurt the most by the 2008 financial crisis.

From the regression results, I can conclude that those workers who are aged, black, having lower weekly earnings, not living in the central state and highest level of education is less than a high school graduate were hurt most by the 2008 financial crisis.

Question G

Use the models in (b)-(d) to assess the in-sample accuracy of the classification. What is the proportion of correctly assigned classes?

```r
# Create a function to calucate the precision and recall
normalize_cm <- function(cm, type='precision') {
  if (type == 'precision') {
    col_sum <- colSums(cm)
    col_sum[col_sum == 0] <- 1
    precision_matrix <- sweep(cm, 2, col_sum, FUN="/")
    return(precision_matrix)
  } else if (type == 'recall') {
    row_sum <- rowSums(cm)
    row_sum[row_sum == 0] <- 1
    recall_matrix <- sweep(cm, 1, row_sum, FUN="/")
    return(recall_matrix)
  } else {
    stop("Type must be either 'precision' or 'recall'")
  }
}

# To get recall and precision from the matrix
extract_values <- function(matrix_recall,matrix_precision) {
  recall <- diag(matrix_recall)
  precision <- diag(matrix_precision)


  recall[is.na(recall)] <- 0
  precision[is.na(precision)] <- 0

  return(cbind(recall, precision))
}
```

```
# LPM predictions
LPM_predictions_raw <- predict(binary_lm, newdata = data, type = "response")
LPM_predictions<- ifelse(LPM_predictions_raw > 0.5, 1, 0)

# Logit predictions
logit_predictions_raw <- predict(binary_logit, newdata = data, type = "response")
logit_predictions<- ifelse(logit_predictions_raw > 0.5, 1, 0)

# Probit predictions
probit_predictions_raw <- predict(binary_probit, newdata = data, type = "response")
probit_predictions<-  ifelse(probit_predictions_raw > 0.5, 1, 0)

LPM_accuracy    <- sum(LPM_predictions== data$employed) / nrow(data)
logit_accuracy  <- sum( logit_predictions== data$employed) / nrow(data)
probit_accuracy <- sum( probit_predictions== data$employed) / nrow(data)

confusion_matrix_LPM    <- table(Actual = data$employed,
                                 Predicted = factor(LPM_predictions,
                                                    levels = c(0, 1)))
confusion_matrix_logit  <- table(Actual = data$employed,
                                 Predicted = factor(logit_predictions,
                                                    levels = c(0,1)))
confusion_matrix_probit <- table(Actual = data$employed,
                                 Predicted = factor(probit_predictions,
                                                    levels=c(0,1)))
```

Table 1: Accuracy for Three Models

	x
Logit	0.8754619
Probit	0.8754619
LPM	0.8754619

The accuracy of all three models is 87.546%.

```
confusion_matrix_LPM
```

```
##       Predicted
## Actual    0    1
##      0    0  674
##      1    0 4738
```

These three models have identical confusion matrices, and from the results, we can observe the following:

True Negatives (TN): 0 (the number of actual class 0 predicted as class 0). The models did not correctly predict any instances that were actually class 0.

False Positives (FP): 674 (the number of actual class 0 predicted as class 1). All instances that were actually class 0 were incorrectly predicted as class 1.

False Negatives (FN): 0 (the number of actual class 1 predicted as class 0). The models did not incorrectly predict any instances that were actually class 1 as class 0.

True Positives (TP): 4738 (the number of actual class 1 predicted as class 1). The models correctly predicted all instances that were actually class 1.

This performance of the models may indicate a significant issue with data imbalance, or a bias in feature recognition and learning within the models, leading them to recognize only one class.

Question H

Optional: Repeat point (g) using one or more (at your discretion) of the following classification algorithms: Naïve Bayes Classifier, Linear Discriminant Analysis, Quadratic Discriminant Analysis, Decision trees, Random forests, K-Nearest Neighbours.

Build Models

We start by constructing models using six distinct classification algorithms, and get all the output predictions.

1. Naïve Bayes

```
# Naïve Bayes
data$age_square <- data$age **2
nb_model <- naiveBayes(as.factor(employed) ~ age + age_square, data)
nb_predictions_raw <- predict(nb_model, newdata = data, type = "raw")
nb_predictions <- ifelse(nb_predictions_raw[,2] > 0.5, 1, 0)
```

2. Linear Discriminant Analysis

```
# Fit LDA Classifer
lda_model <- lda(as.factor(employed) ~ age + I(age ^ 2), data)
lda_predictions <- predict(lda_model, newdata = data)$class
```

3. Quadratic Discriminant Analysis

```
# Fit QDA Classifier
qda_model <- qda(as.factor(employed) ~ age + I(age ^ 2), data)
qda_predictions <- predict(qda_model, newdata = data)$class
```

4. Decision Tree

```r
# Decision Tree
set.seed(123)
decision_tree_model <- rpart(as.factor(employed) ~ age +  I(age ^ 2), data = data)
dt_predictions <- predict(decision_tree_model, newdata = data, type = "class")
```

5. Random Forests

```r
# Random forests
rf_model <- randomForest(as.factor(employed) ~ age +  I(age ^ 2),
                         data= data,
                         num.trees= 100)
rf_predictions <- predict(rf_model , data)
```

6. K-Nearest Neighbours

```r
# kNN Classifier
data$age_squared <- data$age ** 2
scaled_data <- scale(data[, c("age", "age_squared")])
# Define the number of neighbors
k <- 7
knn_predictions <- knn(train = scaled_data, test = scaled_data,
                       cl = as.factor(data$employed), k = k)

# Compute All Accuracies
nb_accuracy <- sum(nb_predictions == data$employed) / nrow(data)
lda_accuracy <- sum(lda_predictions == data$employed) / nrow(data)
qda_accuracy <- sum(qda_predictions == data$employed) / nrow(data)
dt_accuracy <- sum(dt_predictions == data$employed) / nrow(data)
rf_accuracy <- sum(rf_predictions == data$employed) / nrow(data)
knn_accuracy <- sum(knn_predictions == data$employed) / nrow(data)
```

Calculate Algorithm Accuracies

Next, we calculate the accuracy of each algorithm and put the results into a table for comparison.

Table 2: Accuracy for All Models

Model	Accuracy
naiveBayes	0.8754619
LDA Classifer	0.8754619
QDA Classifier	0.8610495
Decision Tree	0.8754619
Random Forest	0.8754619
KNN	0.8754619

Confusion Matrix for algorithms

We computed the confusion matrices for all algorithms and observed that, except for Quadratic Discriminant Analysis (QDA), the matrices for all algorithms (Linear Discriminant Analysis, Decision Trees, Random Forests, and K-Nearest Neighbours) were identical. To maintain brevity, we present the confusion matrix for Naive Bayes as representative for these four algorithms. Additionally, a separate confusion matrix for QDA is provided.

- Confusion Matrix for Naive Bayes (Representative for LDA, Decision Tree, Random Forest, and KNN)

```r
confusion_matrix_nb <- table(Actual = data$employed,
                             Predicted = factor(nb_predictions,levels = c(0, 1)))
confusion_matrix_nb_precision <- normalize_cm(confusion_matrix_nb)
confusion_matrix_nb_recall <- normalize_cm(confusion_matrix_nb,'recall')

print(confusion_matrix_nb)
```

```
##       Predicted
## Actual    0    1
##      0    0  674
##      1    0 4738
```

- Confusion Matrix for QDA

```r
confusion_matrix_qda <- table(Actual = data$employed,
                              Predicted = factor(qda_predictions,levels = c(0, 1)))
confusion_matrix_qda_precision <- normalize_cm(confusion_matrix_qda)
confusion_matrix_qda_recall <- normalize_cm(confusion_matrix_qda,'recall')

print(confusion_matrix_qda)
```

```
##       Predicted
## Actual    0    1
##      0   55  619
##      1  133 4605
```

By combining the accuracies and confusion matrices, we observe that the accuracies for all algorithms, except for the QDA classifier, are high and identical at approximately 87.55%. Generally, an accuracy of 87.55% indicates a good rate of correctly predicted data. However, upon examining the confusion matrices for the five algorithms excluding QDA, we notice that both values in the first column are 0. This suggests that the models fail to classify any data points as class 0, predicting all instances as class 1, regardless of their actual class. This pattern implies that our models have not learned effectively, except for the QDA algorithm, which correctly classifies 55 data points as class 0.

We recognize this problem as the problem of the original data set, as analyzed in Question a, 87.55% of workers are in class 1 (which just equals to the accuracy of the models), which is imbalance. Therefore, we chose to use the undersampling technique to preprocess the data by removing samples from class 1.

Improvment

Since the dataset is unbalanced, so we can used over sample to create some new rows and put the into Random Forest model to train it

```r
clean_data <- na.omit(data)


clean_data <- clean_data %>%
  mutate(employed=as.factor(employed)) %>%
  mutate(race=as.factor(race))

clean_data <- subset(clean_data, select = -unemployed)

set.seed(122)
sample_size <- floor(0.70 * nrow(clean_data))
train_index <- sample(seq_len(nrow(clean_data)), size = sample_size)
train_data <- clean_data[train_index, ]
test_data <- clean_data[-train_index, ]


balanced_data <- ovun.sample(employed~.,data = train_data, N=nrow(train_data), p=0.5,
                                seed=1, method="both")$data


rf_model <- randomForest(as.factor(employed) ~. , data= balanced_data, num.trees= 100)

rf_predictions <- predict(rf_model , balanced_data)
rf_predictions_test <- predict(rf_model , test_data)

# Accuracy
rf_accuracy <- sum(rf_predictions == balanced_data$employed)/ nrow(balanced_data)
rf_accuracy_test <- sum(rf_predictions_test == test_data$employed)/ nrow(test_data)



confusion_matrix_rf <- table(Actual = factor(balanced_data$employed,levels=c(0,1)),
                             Predicted = factor(rf_predictions,levels = c(0, 1)))


confusion_matrix_rf_test <- table(Actual = test_data$employed,
                             Predicted = factor(rf_predictions_test,levels = c(0, 1)))
```

Table 3: Employed Fraction of Dataset

	0	1
train_data	401	2940
balanced_data	1726	1615
test_data	195	1237

To address the imbalance in the dataset, we initially divided the data into a training set (70%) and a test set (30%). We then applied the oversampling method, specifically the ovun.sample method, to equalize the number of samples between class 0 and class 1. As indicated in the table, the balanced dataset now has an equal fraction of rows for both classes, which should enhance the model's analytical capabilities. We train a Random Forest model in the balanced dataset, and test the model in test dataset.

Table 4: Accuracy for train and Test

DataSet	Accuracy
Balanced Set accuracy	0.9916193
Test Set accuracy	0.7856145

From the table, it is evident that the accuracy on the training set is extremely high, reaching 99.16%. However, the accuracy on the test set is 78.56%, which is significantly lower than that of the training set and also lower than the initial model's 87.5%. To better evaluate the model's fitting, we will continue to examine the results of the confusion matrix.

```
confusion_matrix_rf
```

```
##       Predicted
## Actual    0    1
##      0 1608    7
##      1   21 1705
```

```
confusion_matrix_rf_test
```

```
##       Predicted
## Actual    0    1
##      0   50  145
##      1  162 1075
```

The confusion matrix reveals that the model excels on the training data but experiences a notable performance decline with the test data. However, compared to the initial model, it demonstrates improved fitting for class 0 data, aligning with our dataset's specific requirements. To enhance performance further, exploring methods to mitigate overfitting could be advantageous. This may include implementing more regularization, adjusting the model's complexity, or expanding the diversity of the training data to better generalize to new datasets.