Week 11 Mini Lecture/Demo/Discussion

Qian Jiang

2022-10-25

Using `LR_train.csv` and `LR_test.csv`, answer the following questions. Before answering, load the following libraries.

```
library(tidyverse)
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.2 --
## v ggplot2 3.3.6      v purrr   0.3.4
## v tibble  3.1.8      v dplyr   1.0.10
## v tidyr   1.2.1      v stringr 1.4.1
## v readr   2.1.2      v forcats 0.5.2
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```
library(PerformanceAnalytics)
```

```
## Loading required package: xts
## Loading required package: zoo
##
## Attaching package: 'zoo'
##
## The following objects are masked from 'package:base':
##
##     as.Date, as.Date.numeric
##
##
## Attaching package: 'xts'
##
## The following objects are masked from 'package:dplyr':
##
##     first, last
##
##
## Attaching package: 'PerformanceAnalytics'
##
## The following object is masked from 'package:graphics':
##
##     legend
```

```
library(pROC)
```

```
## Type 'citation("pROC")' for a citation.
##
## Attaching package: 'pROC'
```

```
##
## The following objects are masked from 'package:stats':
##
##     cov, smooth, var
```

```
source("helperfunctions.R")
```

A. Ask 1. Describe the business problem. Explain why the business problem is relevant to the company/department.

WorldTel is a telco company providing phone and internet services to customers. WorldTel is experiencing a high volume of churn in its customers, as they move to competitors who have entered the market recently. WorldTel needs to curtail the loss in revenue, by intervening early and retaining customers. It needs to do so with limited budget on customer retention programs.

2. State the business aim or question.

The business aim is to develop a regression model using customer data from the previous year, to predict the likelihood of customers to churn. The predictions should guide the customer retention programs to maximise the overall profit margin in the next 2 years.

B. Acquire 1. Load `LR_train.csv` into the variable `train` and `LR_test.csv` into the variable `test`. How large is the train and test data?

```
train <- read.csv("../data/LR_train.csv", stringsAsFactors=TRUE)
test <- read.csv("../data/LR_test.csv", stringsAsFactors=TRUE)
```

2. Inspect the structure of the dataset. State the number of rows and columns in the dataset. For each column, state whether the variable is continuous, categorical, count etc.

```
str(train)
```

```
## 'data.frame':    2404 obs. of  20 variables:
##  $ gender           : Factor w/ 2 levels "Female","Male": 1 2 2 2 1 1 2 1 1 2 ...
##  $ senior_citizen   : Factor w/ 2 levels "No","Yes": 1 1 1 1 1 1 1 1 1 1 ...
##  $ partner          : Factor w/ 2 levels "No","Yes": 2 1 1 1 1 1 1 1 2 1 ...
##  $ dependents       : Factor w/ 2 levels "No","Yes": 1 1 1 1 1 1 2 1 1 2 ...
##  $ tenure           : num  1.6 35 2.7 45.9 2.2 8.2 22.5 10.4 28.9 62.2 ...
##  $ phone_service    : Factor w/ 2 levels "No","Yes": 1 2 2 1 2 2 2 1 2 2 ...
##  $ multiple_lines   : Factor w/ 2 levels "No","Yes": 1 1 1 1 1 2 2 1 2 1 ...
##  $ internet_service : Factor w/ 3 levels "DSL","FiberOptic",..: 1 1 1 1 2 2 2 1 2 1 ...
##  $ online_security  : Factor w/ 2 levels "No","Yes": 1 2 2 2 1 1 1 2 1 2 ...
##  $ online_backup    : Factor w/ 2 levels "No","Yes": 2 1 2 1 1 1 2 1 1 2 ...
##  $ device_protection: Factor w/ 2 levels "No","Yes": 1 2 1 2 1 2 1 1 2 1 ...
##  $ tech_support     : Factor w/ 2 levels "No","Yes": 1 1 1 2 1 1 1 1 2 1 ...
##  $ streaming_tv     : Factor w/ 2 levels "No","Yes": 1 1 1 1 1 2 2 1 2 1 ...
##  $ streaming_movies : Factor w/ 2 levels "No","Yes": 1 1 1 1 1 2 1 1 2 1 ...
##  $ contract         : Factor w/ 3 levels "MonthToMonth",..: 1 2 1 2 1 1 1 1 1 2 ...
##  $ paperless_billing: Factor w/ 2 levels "No","Yes": 2 1 2 1 2 2 2 1 2 1 ...
##  $ payment_method   : Factor w/ 4 levels "BankTransferAutomatic",..: 3 4 4 1 3 3 2 4 3 1 ...
##  $ monthly_charges  : num  29.9 57 53.9 42.3 70.7 ...
##  $ total_charges    : num  29.9 1889.5 108.2 1840.8 151.7 ...
##  $ churn            : int  0 0 1 0 1 1 0 0 1 0 ...
```

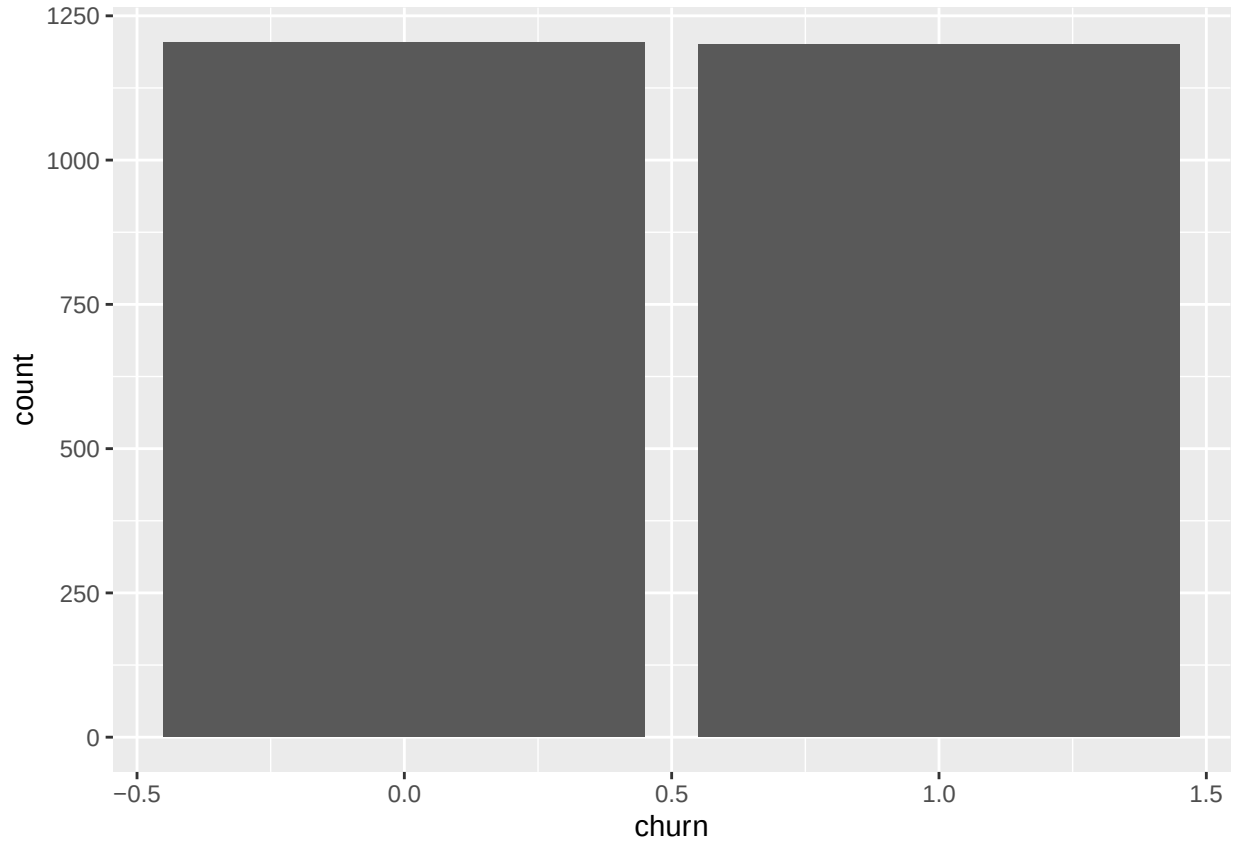
3. Change the default reference levels in `internet_service` as "No", `contract` as "Two year" and `payment_method` as "Mailed Check".

```
levels(train$internet_service) <- c("No", "DSL", "FiberOptic")
levels(test$internet_service) <- c("No", "DSL", "FiberOptic")
levels(train$contract) <- c("TwoYear", "MonthToMonth", "OneYear")
levels(test$contract) <- c("TwoYear", "MonthToMonth", "OneYear")
levels(train$payment_method) <- c("MailedCheck", "BankTransferAutomatic", "CreditCardAutomatic", "Electr
levels(test$payment_method) <- c("MailedCheck", "BankTransferAutomatic", "CreditCardAutomatic", "Electro
```

C. Analyse

Exploratory Data Analysis 1. Describe and visualise the dependent variable, `churn`. What proportion of customers churn in the train and test data? Is the data balanced?

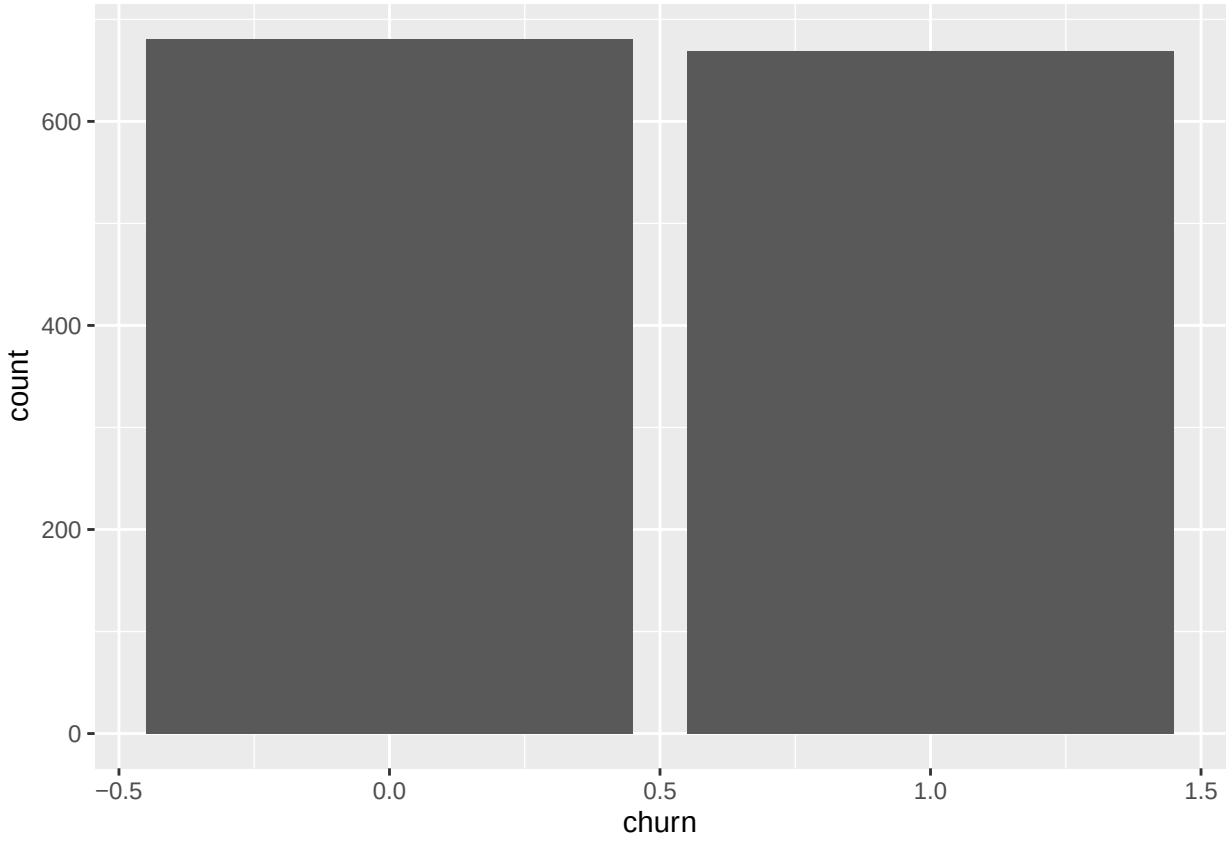
```
# Bar plot of churn (train)
ggplot(train) + geom_bar(aes(x=churn, fill=churn))
```




```
# Proportion of churn (train)
train %>%
  group_by(churn) %>%
  summarise(prop = n()/nrow(train))
```

```
## # A tibble: 2 x 2
##   churn  prop
##   <int> <dbl>
## 1     0 0.501
## 2     1 0.499
```

```
# Bar plot of churn (test)
ggplot(test) + geom_bar(aes(x=churn, fill=churn))
```




```
# Proportion of churn (test)
test %>%
  group_by(churn) %>%
  summarise(prop = n()/nrow(test))
```

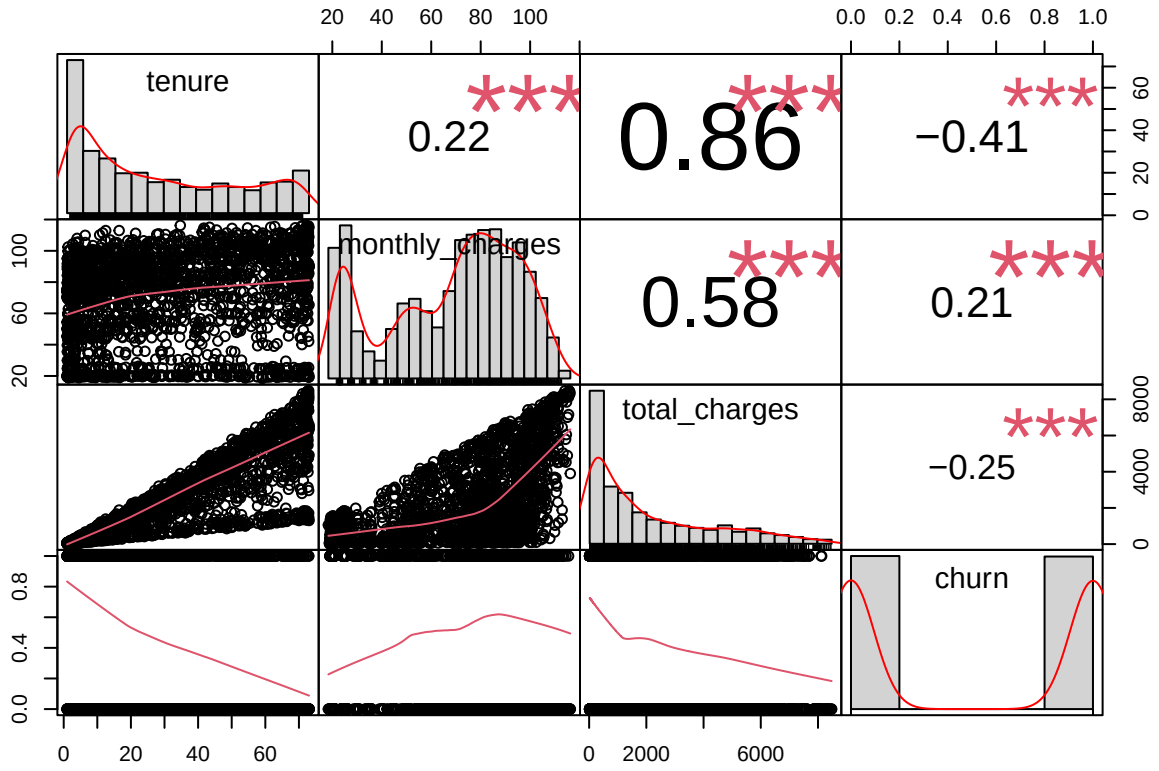
```
## # A tibble: 2 x 2
##   churn  prop
##   <int> <dbl>
## 1     0 0.504
## 2     1 0.496
```

2. Visualise the correlation between the independent continuous variables `tenure`, `monthly_charges`, `total_charges`. Which variable can be removed to reduce multicollinearity issues?

```
numvars <- unlist(lapply(train, is.numeric))
chart.Correlation(train[,numvars], histogram=TRUE)
```

```
## Warning in par(usr): argument 1 does not name a graphical parameter

## Warning in par(usr): argument 1 does not name a graphical parameter

## Warning in par(usr): argument 1 does not name a graphical parameter
```

```
## Warning in par(usr): argument 1 does not name a graphical parameter

## Warning in par(usr): argument 1 does not name a graphical parameter

## Warning in par(usr): argument 1 does not name a graphical parameter
```



Predictive Analytics 1. Fit a logistic regression model on `train`.

```
mod1 <- glm(formula=churn~.,
            family="binomial", data=train)
summary(mod1)
```

```
##
## Call:
## glm(formula = churn ~ ., family = "binomial", data = train)
##
## Deviance Residuals:
##      Min        1Q    Median        3Q       Max
## -2.33808  -0.75588  -0.07039   0.73502   3.14571
##
## Coefficients:
##                                  Estimate Std. Error z value Pr(>|z|)
## (Intercept)                     4.1819987  1.3041007   3.207 0.001342 **
## genderMale                      0.0160106  0.1040369   0.154 0.877693
## senior_citizenYes               0.1098850  0.1385243   0.793 0.427630
## partnerYes                     -0.0445829  0.1274075  -0.350 0.726396
## dependentsYes                  -0.2142383  0.1409217  -1.520 0.128445
```

```
## tenure                                -0.0581918  0.0093337  -6.235 4.53e-10 ***
## phone_serviceYes                       1.7415677  1.0387622   1.677 0.093625 .
## multiple_linesYes                      0.8955901  0.2852391   3.140 0.001691 **
## internet_serviceDSL                    3.6031002  1.2799405   2.815 0.004877 **
## internet_serviceFiberOptic            -3.9159361  1.2862500  -3.044 0.002331 **
## online_securityYes                     0.1315551  0.2823970   0.466 0.641322
## online_backupYes                       0.4471869  0.2824832   1.583 0.113409
## device_protectionYes                   0.6150023  0.2827645   2.175 0.029633 *
## tech_supportYes                       -0.0219260  0.2841163  -0.077 0.938486
## streaming_tvYes                        1.2572680  0.5188466   2.423 0.015385 *
## streaming_moviesYes                    1.4250642  0.5234713   2.722 0.006482 **
## contractMonthToMonth                  -0.7139642  0.1626432  -4.390 1.13e-05 ***
## contractOneYear                       -1.6012889  0.2642696  -6.059 1.37e-09 ***
## paperless_billingYes                   0.2838287  0.1178195   2.409 0.015996 *
## payment_methodBankTransferAutomatic -0.1185341  0.1730523  -0.685 0.493368
## payment_methodCreditCardAutomatic      0.4902533  0.1470449   3.334 0.000856 ***
## payment_methodElectronicCheck          0.0007764  0.1802147   0.004 0.996562
## monthly_charges                       -0.1190499  0.0507626  -2.345 0.019015 *
## total_charges                          0.0003447  0.0001072   3.214 0.001308 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 3332.6  on 2403  degrees of freedom
## Residual deviance: 2269.6  on 2380  degrees of freedom
## AIC: 2317.6
##
## Number of Fisher Scoring iterations: 6
```

2. Perform automatic backward selection to ensure that all coefficient estimates are significant at the 5% level of statistical significance.

```
trainm <- data.frame(model.matrix(mod1)[,-1]) %>%
  mutate(churn=train$churn)
testm <- data.frame(model.matrix(glm(
  formula=churn~.,
  family="binomial", data=test))[,-1]) %>%
  mutate(churn=test$churn)
```

```
mod2 <- sig_removal(glmobject=mod1, response="churn", sig=0.05, data=trainm)
```

```
## Removing:  payment_methodElectronicCheck
## Removing:  tech_supportYes
## Removing:  genderMale
## Removing:  partnerYes
## Removing:  payment_methodBankTransferAutomatic
## Removing:  senior_citizenYes
## Removing:  online_securityYes
```

```
summary(mod2)
```

```
##
## Call:
## glm(formula = formula, family = binomial, data = data)
##
```

```
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.2973  -0.7593  -0.0724   0.7333   3.1607
##
## Coefficients:
##                                    Estimate Std. Error z value Pr(>|z|)
## (Intercept)                         3.9557480  0.5318256   7.438 1.02e-13 ***
## dependentsYes                      -0.2499941  0.1233112  -2.027 0.042627 *
## tenure                             -0.0584651  0.0090949  -6.428 1.29e-10 ***
## phone_serviceYes                    1.5570128  0.4066944   3.828 0.000129 ***
## multiple_linesYes                   0.8487847  0.1527177   5.558 2.73e-08 ***
## internet_serviceDSL                 3.3741191  0.4244361   7.950 1.87e-15 ***
## internet_serviceFiberOptic         -3.6849757  0.5138968  -7.171 7.46e-13 ***
## online_backupYes                    0.4048770  0.1494990   2.708 0.006764 **
## device_protectionYes                0.5671206  0.1551850   3.654 0.000258 ***
## streaming_tvYes                     1.1601295  0.2141849   5.416 6.08e-08 ***
## streaming_moviesYes                 1.3283359  0.2147262   6.186 6.16e-10 ***
## contractMonthToMonth               -0.7263929  0.1618358  -4.488 7.17e-06 ***
## contractOneYear                    -1.6150688  0.2612929  -6.181 6.37e-10 ***
## paperless_billingYes                0.2808297  0.1175359   2.389 0.016880 *
## payment_methodCreditCardAutomatic   0.5329333  0.1102282   4.835 1.33e-06 ***
## monthly_charges                    -0.1099595  0.0169614  -6.483 9.00e-11 ***
## total_charges                       0.0003474  0.0001061   3.273 0.001063 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 3332.6  on 2403  degrees of freedom
## Residual deviance: 2271.7  on 2387  degrees of freedom
## AIC: 2305.7
##
## Number of Fisher Scoring iterations: 6
```

3. Using the logistic regression model with a threshold of 0.5, generate predictions for the first two observations in `train`. Classify each of these predictions as true positives, true negatives, false positives or false negatives. Explain.

```
# Observation 1
predict(mod2, newdata=trainm[1,], type="response")
```

```
##         1
## 0.8592316
```

```
trainm[1,"churn"]
```

```
## [1] 0
```

```
# Observation 2
predict(mod2, newdata=trainm[2,], type="response")
```

```
##          2
## 0.09124038
```

```
trainm[2,"churn"]
```

```
## [1] 0
```

4. Using the logistic regression model with a threshold of 0.5, generate the confusion matrix for `train`.

```
y_true <- trainm$churn
y_prob <- predict(mod2, newdat=trainm, type="response")
y_pred <- ifelse(y_prob>0.5, "Yes", "No")
confusion_matrix <- table(y_true, y_pred)

TN <- confusion_matrix[1,1]
TP <- confusion_matrix[2,2]
FP <- confusion_matrix[1,2]
FN <- confusion_matrix[2,1]
```

5. Calculate the accuracy of the logistic regression model.

```
accuracy <- (TP + TN)/(TP + TN + FP + FN)
accuracy
```

```
## [1] 0.7737105
```

6. Calculate the sensitivity of the logistic regression model.

```
all_positives <- TP + FN
sensitivity <- TP/all_positives
sensitivity
```

```
## [1] 0.8175
```

7. Calculate the specificity of the logistic regression model.

```
all_negatives <- TN + FP
specificity <- TN/all_negatives
specificity
```

```
## [1] 0.7300664
```

8. Calculate the precision of the logistic regression model.

```
all_pred_positives <- FP + TP
precision <- TP/all_pred_positives
precision
```

```
## [1] 0.7511485
```

9. Calculate the F1-Score of the logistic regression model.

```
recall <- sensitivity
f1_score <- (2 * precision * recall) /(precision + recall)
f1_score
```

```
## [1] 0.782921
```

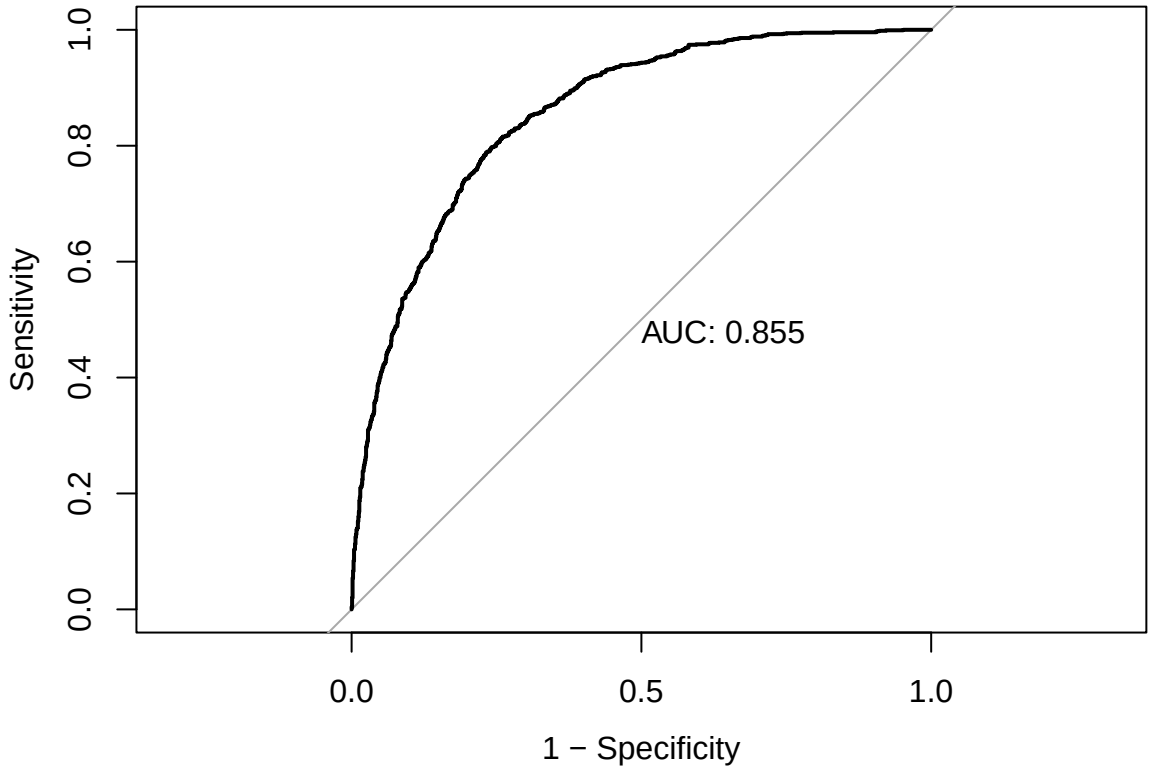
10. Plot the ROC Curve of the logistic regression model. Hence, or otherwise, calculate the Area Under Curve (AUC) of the model.

```
rocobj <- roc(y_true, y_prob)
```

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

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

```
plot(rocobj, legacy.axes = TRUE, print.auc = TRUE)
```




```
auc <- auc(rocobj)
auc
```

```
## Area under the curve: 0.8552
```

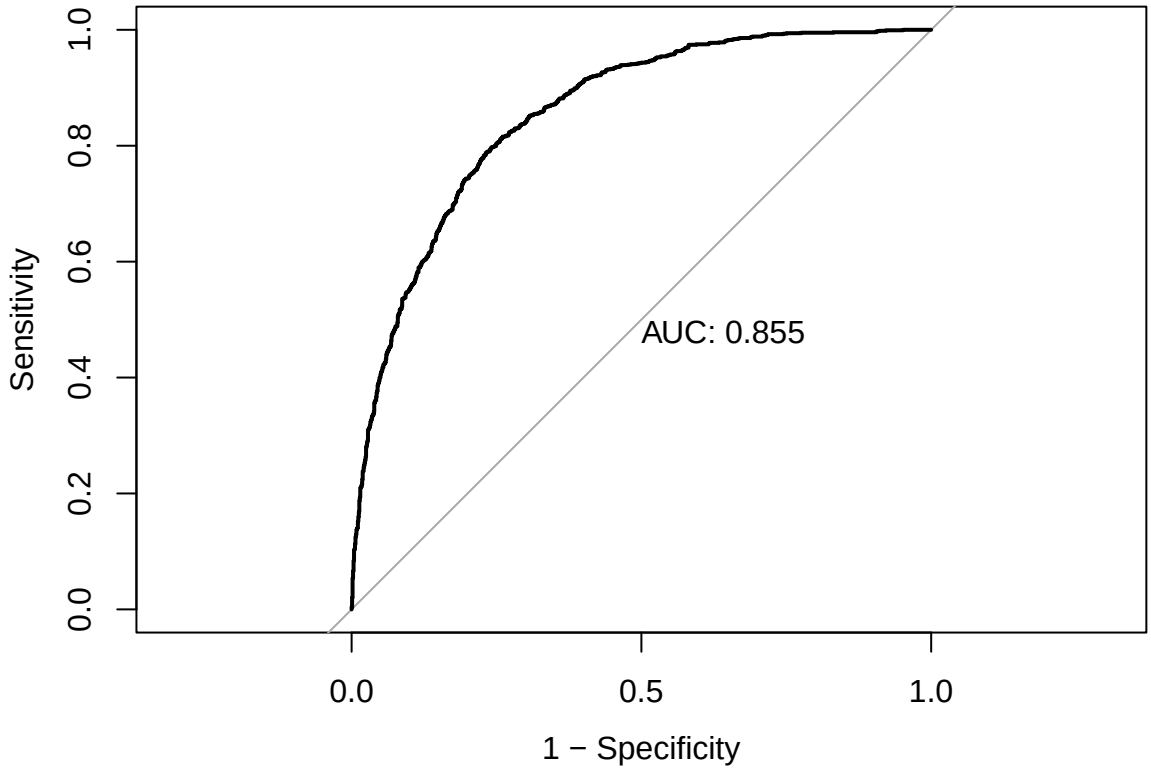
11 Use `eval_logistic_model()` to generate the evaluation metrics for a logistic regression model which uses a threshold of 0.5 and 0.6 respectively.

```
# Evaluation metrics for train/test on mod3 with threshold 0.5
eval_logistic_model(mod2, trainm, 0.5)
```

```
##       y_pred
## y_true  No Yes
##      0 879 325
##      1 219 981
```

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

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

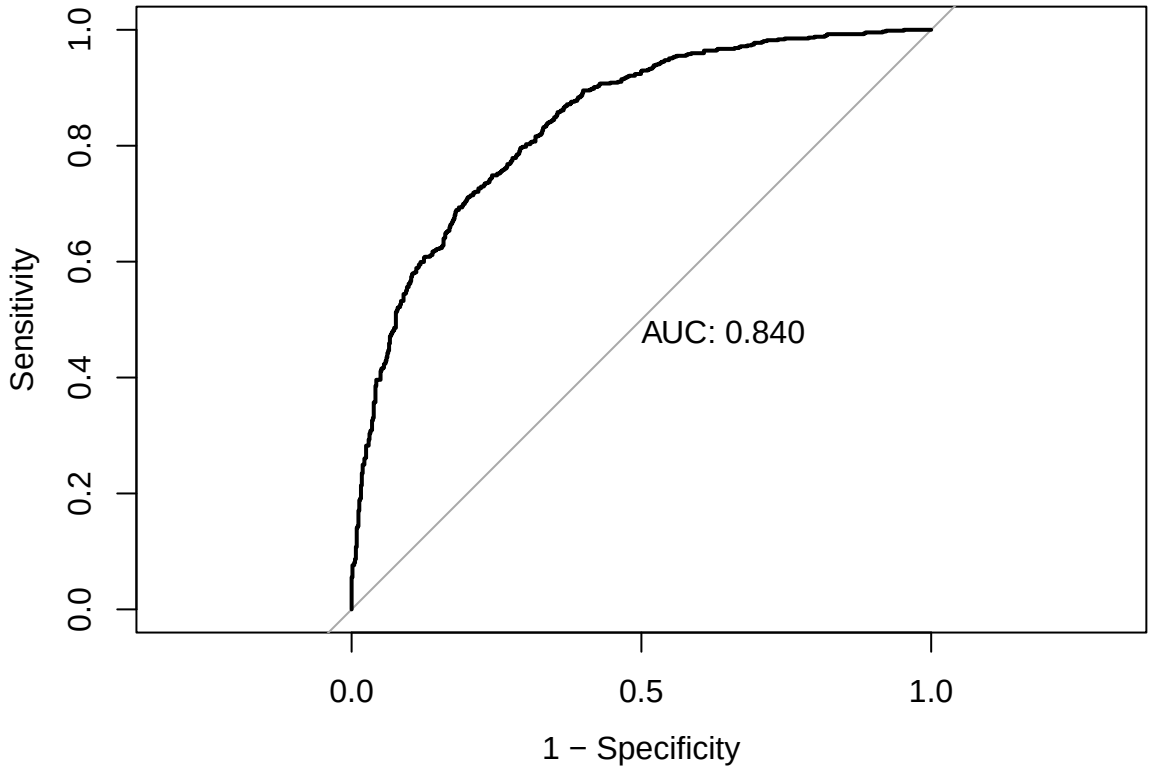



```
##       Metrics     Value
## 1    accuracy 0.7737105
## 2 sensitivity 0.8175000
## 3 specificity 0.7300664
## 4   precision 0.7511485
## 5    F1_score 0.7829210
## 6         AUC 0.8551800
```

```
eval_logistic_model(mod2, testm, 0.5)
```

```
##       y_pred
## y_true  No Yes
##      0 506 175
##      1 165 504
```

```
## Setting levels: control = 0, case = 1
## Setting direction: controls < cases
```



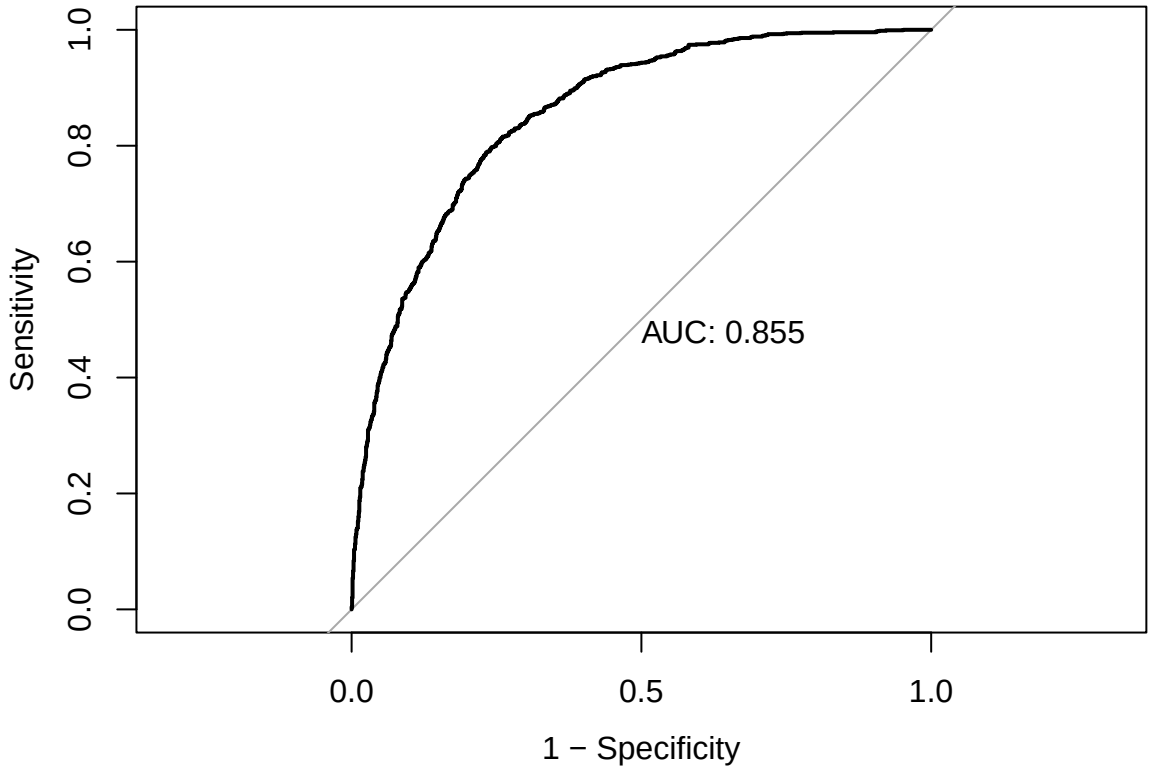

```
##       Metrics     Value
## 1    accuracy 0.7481481
## 2 sensitivity 0.7533632
## 3 specificity 0.7430250
## 4   precision 0.7422680
## 5    F1_score 0.7477745
## 6         AUC 0.8401959
```

```
# Evaluation metrics for train/test on mod3 with threshold 0.6
eval_logistic_model(mod2, trainm, 0.6)
```

```
##       y_pred
## y_true  No Yes
##      0 975 229
##      1 329 871
```

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

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

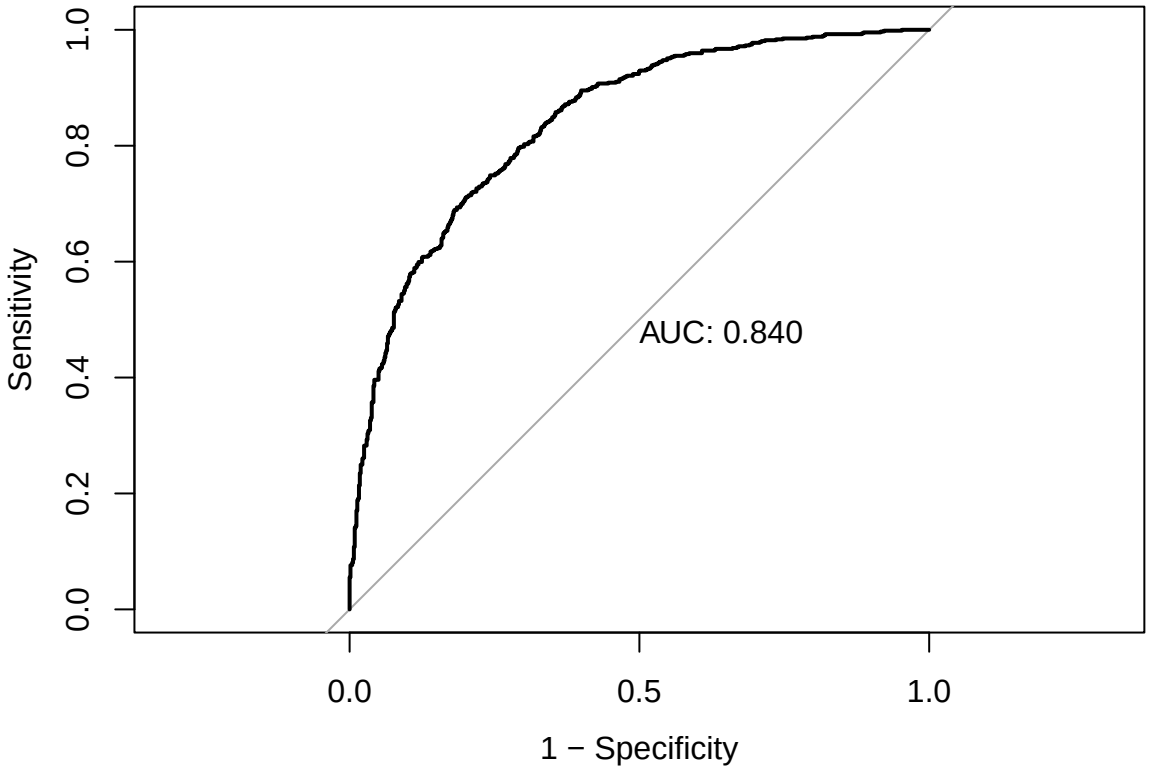



```
##       Metrics     Value
## 1    accuracy 0.7678869
## 2 sensitivity 0.7258333
## 3 specificity 0.8098007
## 4   precision 0.7918182
## 5    F1_score 0.7573913
## 6         AUC 0.8551800
```

```
eval_logistic_model(mod2, testm, 0.6)
```

```
##       y_pred
## y_true  No Yes
##      0 555 126
##      1 206 463
```

```
## Setting levels: control = 0, case = 1
## Setting direction: controls < cases
```




```
##       Metrics     Value
## 1    accuracy 0.7540741
## 2 sensitivity 0.6920777
## 3 specificity 0.8149780
## 4   precision 0.7860781
## 5    F1_score 0.7360890
## 6         AUC 0.8401959
```