Assignment 3

PART 1: Model Building

```
# Load the data
data <- read.csv('./Student_Dropout.csv')

# Preparing data for our analysis (We delete studentID from our data as it is
not necessary and causes ambiguity)
data = select(data, -c("StudentID"))
str(data)
```

```
## 'data.frame':    4424 obs. of  8 variables:
##  $ Gender          : chr  "Female" "Female" "Female" "Male" ...
##  $ BirthYear       : int  2004 2005 2005 2004 1979 1974 2006 2002 2003
2006 ...
##  $ Major           : chr  "APSC" "HSCI" "ARTS" "EDUC" ...
##  $ GPA             : num  2.64 3.46 2.64 2.64 2.17 ...
##  $ CreditsCompleted: int  81 56 88 73 104 79 73 73 99 84 ...
##  $ PartTime        : chr  "Yes" "Yes" "Yes" "Yes" ...
##  $ Scholarship     : chr  "No" "No" "No" "No" ...
##  $ Dropout         : int  1 0 1 0 0 0 0 1 0 1 ...
```

a)

```
# Total number of students
total_students <- nrow(data)

# Percentage of dropouts
dropouts_per <- mean(data$Dropout) * 100

cat("Total number of students:", total_students, "\n")
```

```
## Total number of students: 4424
```

```
cat("Percentage of students who dropped out:", dropouts_per, "%\n")
```

```
## Percentage of students who dropped out: 32.12025 %
```

b)

```r
# We need to get Age of students in order to build a suitable regression
model
data <- mutate(data, BirthYear = 2024 - BirthYear)
names(data)[names(data) == "BirthYear"] <- "Age"

# Splitting the data into training (75%) and testing (25%) sets
set.seed(1000)
split = sample.split(data$Dropout, SplitRatio = 0.75)

train <- subset(data, split == TRUE)
test <- subset(data, split == FALSE)

# Building a best-fit logistic regression model to predict the variable
Dropout
model_log <- glm(Dropout ~ ., data = train, family = "binomial")
best_model = step(model_log)

## Start:  AIC=3668.39
## Dropout ~ Gender + Age + Major + GPA + CreditsCompleted + PartTime +
##     Scholarship
##
##                     Df Deviance    AIC
## - CreditsCompleted  1   3641.2 3667.2
## <none>                  3640.4 3668.4
## - GPA               1   3642.7 3668.7
## - PartTime          1   3645.1 3671.1
## - Gender            1   3685.3 3711.3
## - Major             7   3735.0 3749.0
## - Age               1   3737.3 3763.3
## - Scholarship       1   3768.9 3794.9
##
## Step:  AIC=3667.17
## Dropout ~ Gender + Age + Major + GPA + PartTime + Scholarship
##
##               Df Deviance    AIC
## <none>             3641.2 3667.2
## - GPA          1   3643.4 3667.4
## - PartTime     1   3645.9 3669.9
## - Gender       1   3686.2 3710.2
## - Major        7   3735.5 3747.5
## - Age          1   3738.2 3762.2
## - Scholarship  1   3769.4 3793.4
```

```
summary(best_model)
```

```
##
## Call:
## glm(formula = Dropout ~ Gender + Age + Major + GPA + PartTime +
##     Scholarship, family = "binomial", data = train)
##
## Coefficients:
##                 Estimate Std. Error z value Pr(>|z|)
## (Intercept)    -1.179644   0.515355  -2.289   0.0221 *
## GenderMale     -0.575537   0.085608  -6.723 1.78e-11 ***
## Age             0.057720   0.006056   9.531  < 2e-16 ***
## MajorARTS      -0.676183   0.145499  -4.647 3.36e-06 ***
## MajorBUS       -0.388055   0.155864  -2.490   0.0128 *
## MajorEDUC       0.358198   0.153990   2.326   0.0200 *
## MajorENV        0.417096   0.197095   2.116   0.0343 *
## MajorFCAT       0.153144   0.156847   0.976   0.3289
## MajorHSCI       0.278941   0.174322   1.600   0.1096
## MajorSCI       -0.418089   0.180974  -2.310   0.0209 *
## GPA            -0.215871   0.144613  -1.493   0.1355
## PartTimeYes     0.413226   0.190352   2.171   0.0299 *
## ScholarshipYes -1.265294   0.121861 -10.383  < 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: 4166.3  on 3317  degrees of freedom
## Residual deviance: 3641.2  on 3305  degrees of freedom
## AIC: 3667.2
##
## Number of Fisher Scoring iterations: 4
```

c)

We can see that the variable CreditsCompleted has been eliminated from the model by using step(). We see intriguing findings in our model, such as the coefficients of Age (≈ 0.058), Scholarship (≈ −1.27), and PartTimeYes (≈ 0.413) which show that older students have a higher dropout rate, students who get scholarships have a lower dropout rate and part time students also tend to have a higher drop out rate. In our model, the three variables MajorFCAT, MajorHSCI, and GPA are the only ones that are not significant considering a 95% confidence interval. The remaining ones are all significant.

d)

The coefficient of the GPA variable in our model is -0.216, which means that for each one-unit increase in GPA, the log-odds of dropping out decreases by ≈ 0.216, holding all other variables constant. Furthermore, it can be expressed in terms of odds: e^(-0.216) ≈ 0.806. It may be observed that (0.806 ∗ 100) − 100 = −19.4. Thus, assuming all other variables remain constant, the probabilities of dropping out reduce by approximately 19.4% for every unit rise in GPA. Hence, higher GPAs equate into a lower chance of dropping out. But we should be cautious while dealing with the GPA variable as it is not a significant predictor in our model.

e)

```
val_pred <- data.frame(Gender = "Male", Age = 23, Major = "BUS", GPA = 3.1,
CreditsCompleted = 100, PartTime = "No", Scholarship = "Yes")

pred_value <- predict(best_model, val_pred, type = "response")
cat("Predicted value of Dropout:", pred_value, "\n")

## Predicted value of Dropout: 0.06007722
```

According to our model, this person has a 6% chance of dropping out.

PART 2: Model Performance on Training Dataset

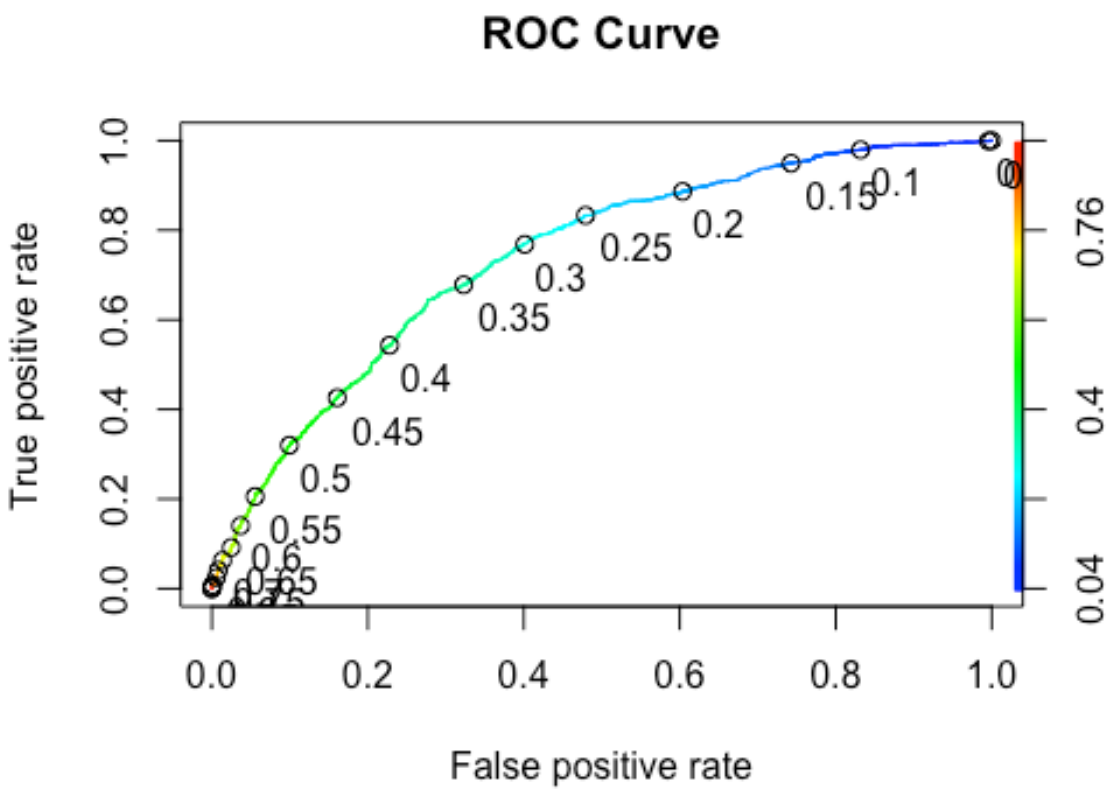
a)

```
# Calculating predicted probabilities for training set
predicted_train <- predict(best_model, type = "response")

# Combining the train and predict_train dataset
train <- cbind(train, predicted_train)

# Creating confusion matrix
confusion_matrix <- table(train$Dropout, train$predicted_train > 0.5)
print(confusion_matrix)

##
##     FALSE TRUE
##   0  2028  224
##   1   725  341

# Calculating specificity, sensitivity, and accuracy
specificity <- confusion_matrix[1,1] / (confusion_matrix[1,1] +
confusion_matrix[1,2])
sensitivity <- confusion_matrix[2,2] / (confusion_matrix[2,2] +
```

```
confusion_matrix[2,1])
accuracy <- sum(diag(confusion_matrix)) / sum(confusion_matrix)

cat("Specificity:", specificity, "\n")

## Specificity: 0.9005329

cat("Sensitivity:", sensitivity, "\n")

## Sensitivity: 0.3198874

cat("Accuracy:", accuracy, "\n")

## Accuracy: 0.7139843
```

b)

```
# Prediction function
ROC_curve_pred = prediction(predicted_train, train$Dropout)

# Performance function
ROC_curve_perf = performance(ROC_curve_pred, "tpr", "fpr")

# Plotting ROC with colors and adding threshold labels
plot(ROC_curve_perf, main = "ROC Curve", colorize = TRUE, print.cutoffs.at =
seq(0, 1, by = 0.05), text.adj = c(-0.2, 1.7))
```




```
# Training Set AUC
AUC <- as.numeric(performance(ROC_curve_pred, "auc")@y.values)

cat("AUC value:", AUC, "\n")

## AUC value: 0.7387303
```

It should be noted that the institution seeks to identify students who are at a high risk of leaving school early in the given situation. Finding more students who are likely to drop out is thus our top focus. Therefore, we may choose True Positive Rate above False Positive Rate. In light of that, choosing the threshold value of 0.35 makes sense.

c)

```
# creating confusion matrix 2
confusion_matrix2 <- table(train$Dropout, train$predicted_train > 0.35)
print(confusion_matrix2)

##
##     FALSE TRUE
##   0  1525  727
##   1   343  723
```

```r
# Calculating specificity, sensitivity, and accuracy
specificity <- confusion_matrix2[1,1] / (confusion_matrix2[1,1] +
confusion_matrix2[1,2])
sensitivity <- confusion_matrix2[2,2] / (confusion_matrix2[2,2] +
confusion_matrix2[2,1])
accuracy <- sum(diag(confusion_matrix2)) / sum(confusion_matrix2)

cat("Specificity:", specificity, "\n")
```

```
## Specificity: 0.6771758
```

```r
cat("Sensitivity:", sensitivity, "\n")
```

```
## Sensitivity: 0.6782364
```

```r
cat("Accuracy:", accuracy, "\n")
```

```
## Accuracy: 0.6775166
```

PART 3: Performance Evaluation on Testing Dataset

a)

```r
# Calculating predicted probabilities for testing set
predicted_test <- predict(best_model, newdata = test, type = "response")

# Combining the train and predictTrain dataset
test <- cbind(test, predicted_test)

# Creating confusion matrix 3
confusion_matrix3 <- table(test$Dropout, test$predicted_test > 0.35)
print(confusion_matrix3)
```

```
##
##     FALSE TRUE
##   0   532  219
##   1   117  238
```

```r
# Calculating specificity, sensitivity, and accuracy
specificity <- confusion_matrix3[1,1] / (confusion_matrix3[1,1] +
confusion_matrix3[1,2])
sensitivity <- confusion_matrix3[2,2] / (confusion_matrix3[2,2] +
confusion_matrix3[2,1])
accuracy <- sum(diag(confusion_matrix3)) / sum(confusion_matrix3)

cat("Specificity:", specificity, "\n")
```

```
## Specificity: 0.7083888
```

```r
cat("Sensitivity:", sensitivity, "\n")
```

```
## Sensitivity: 0.6704225

cat("Accuracy:", accuracy, "\n")

## Accuracy: 0.6962025
```

b)

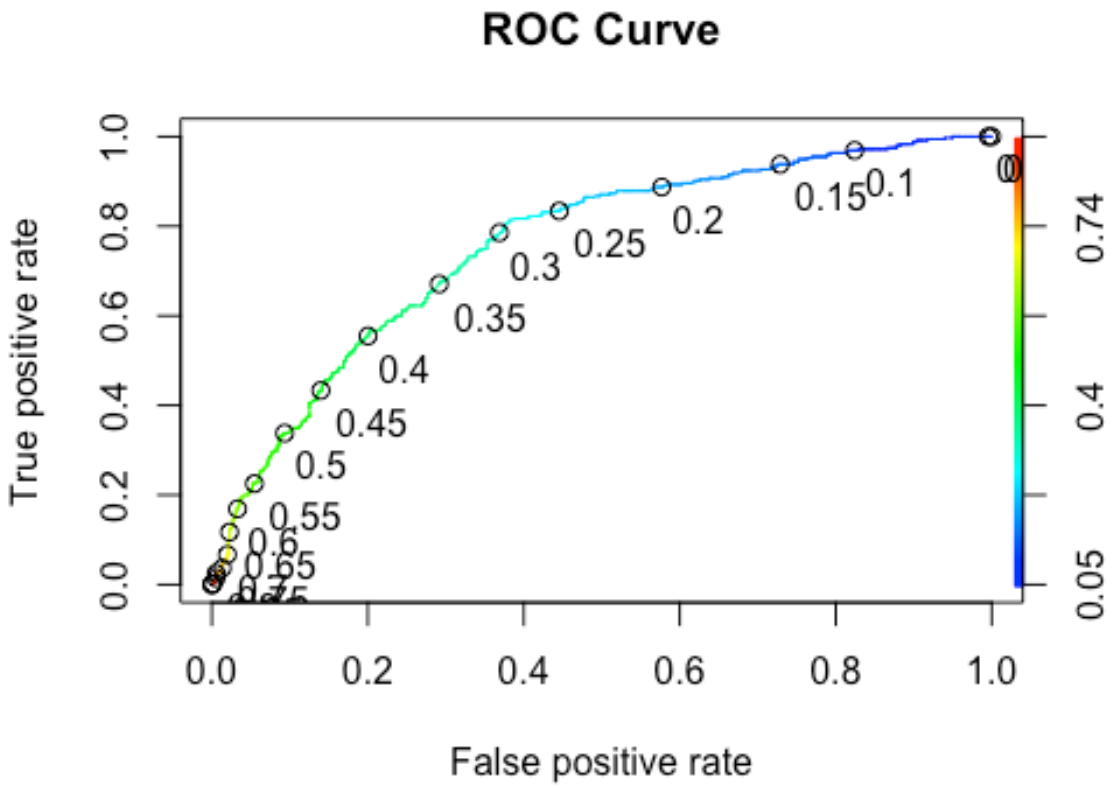
```
# Creating the baseline model
test %>% group_by(Dropout) %>% summarise(n = n())

## # A tibble: 2 × 2
##   Dropout     n
##     <int> <int>
## 1       0   751
## 2       1   355
```

Accuracy of the baseline model = (751+0) / 1106 ≈ 0.679.

This means that our baseline model accurately predicts for about 67.9% of the dataset. The accuracy for our model was ≈ 69.6%, hence our model improves over this simple model.

c)

```
# Prediction function
ROC_curve_pred = prediction(predicted_test, test$Dropout)

# Performance function
ROC_curve_perf = performance(ROC_curve_pred, "tpr", "fpr")

# Plotting ROC with colors and adding threshold labels
plot(ROC_curve_perf, main = "ROC Curve", colorize = TRUE, print.cutoffs.at =
seq(0, 1, by = 0.05), text.adj = c(-0.2, 1.7))
```




```
# Testing Set AUC
AUC <- as.numeric(performance(ROC_curve_pred, "auc")@y.values)

cat("AUC value:", AUC, "\n")

## AUC value: 0.7564956
```

This indicates that our model does significantly better than random guessing, the AUC value of 0.765 indicates that there is a 76.5% chance that the model will correctly differentiate between a randomly chosen student who drops out and a randomly chosen student who does not.

d)

Yes, the model is useful to SFU because it offers a more balanced approach by detecting a higher proportion of students who are considered to be at-risk while retaining a respectable degree of precision and specificity. Early intervention initiatives to support students who are at risk of dropping out can be aided by it for SFU. The following factors can be considered:

(i) AUC (Area Under the Curve): The model performs much better than random guessing, with an AUC of around 0.765, which is far higher than 0.5. This indicates that there is a 76.5% probability that the model will accurately distinguish between a randomly selected student who drops out and a randomly selected student who does not.

(ii) Accuracy: The model's accuracy of 69.6% is higher than that of the baseline model, which was 67.9%. This suggests that generally, our model predicts more accurately than the baseline model.

(iii) Sensitivity: Compared to the baseline model's 0% sensitivity, the model's 67% sensitivity is a significant increase. The algorithm is now considerably more adept at detecting students who are at risk of dropping out because to this improvement in sensitivity.

(iv) Specificity: Compared to the baseline's 100%, the model's specificity is 70.8%. The specificity is still very high, despite the decline, guaranteeing that the majority of students who were expected to continue in school do so.

The sensitivity greatly rises with only a minor drop in accuracy when the threshold is changed. This modification is essential because it enables the model to recognise more students who are at danger of dropping out, which is consistent with SFU's objective of proactively identifying and assisting these students.