Statistical Modeling Course

Logistic Regression Assignment

We will use the `Default` data set in the `ISLR` package for this assignment.

Problem 1

Fit a logistic regression model that uses `student`, `income` and `balance` to predict `default` . Interpret the coefficients.

```
fit1 <- glm(default~., data = Default, family = "binomial")
summary(fit1)
```

```
##
## Call:
## glm(formula = default ~ ., family = "binomial", data = Default)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.4691  -0.1418  -0.0557  -0.0203   3.7383
##
## Coefficients:
##               Estimate Std. Error z value Pr(>|z|)
## (Intercept) -1.087e+01  4.923e-01 -22.080  < 2e-16 ***
## studentYes  -6.468e-01  2.363e-01  -2.738  0.00619 **
## balance      5.737e-03  2.319e-04  24.738  < 2e-16 ***
## income       3.033e-06  8.203e-06   0.370  0.71152
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 2920.6  on 9999  degrees of freedom
## Residual deviance: 1571.5  on 9996  degrees of freedom
## AIC: 1579.5
##
## Number of Fisher Scoring iterations: 8
```

```
exp(coef(fit1))
```

```
##  (Intercept)   studentYes      balance       income
## 1.903854e-05 5.237317e-01 1.005753e+00 1.000003e+00
```

Answer: The odds of student defaulting is lower than non-students with the same balance and income by a factor of 0.52. Income does not have a significant effect on the odds of defaulting. For each dollar increase in balance odds of default increase by a factor of 1.0057.

Problem 2

Using the validation set approach, estimate the test error of this model. To do this, perform the following steps:

- Write a function that takes 2 arguments: a formula and a dateset
- The function should do the following:
  - Split the sample set intro a training set and a validation set.
  - Fit a multiple logistic regression model using only the training observations.
  - Obtain a prediction of default status for each individual in the validation set by computing the estimated probability of default for that individual, and classifying the individual to the `default` category if the estimated probability is greater than 0.5.
  - Compute the validation set error, which is the fraction of the observations in the validation set that are misclassified.
- The function should return the validation error.

```
set.seed(2019)
getTestError <- function(mod_formula, Default){
  trainIDs <- sample(1:nrow(Default), nrow(Default)*.8)
  train <- Default[trainIDs, ]
  test <- Default[-trainIDs, ]

  fit <- glm(mod_formula, data=train, family = binomial)
  probs <- predict(fit, newdata = test, type="response")
  test$pred <- ifelse(probs>0.5, "Yes", "No")
  sum(test$default != test$pred)/nrow(test)
}

getTestError(formula(default~income+balance), Default)
```

```
## [1] 0.023
```

Problem 3

Use your function from Problem 2 to repeat the process ten times, using ten different splits of the observations into a training set and a validation set, then get the average of these test errors. Comment on the results obtained.

```
set.seed(2020)
testerror <- c() # Vector to store test error rates
for (i in 1:10){ # Loop over splits
  testerror[i] <- getTestError(formula(default~income+balance), Default)
}
testerror
```

```
##  [1] 0.0290 0.0220 0.0230 0.0225 0.0300 0.0300 0.0250 0.0235 0.0245 0.0230
```

```
mean(testerror)
```

```
## [1] 0.02525
```

The test error rates are all between 2.2% to 3%.

Problem 4

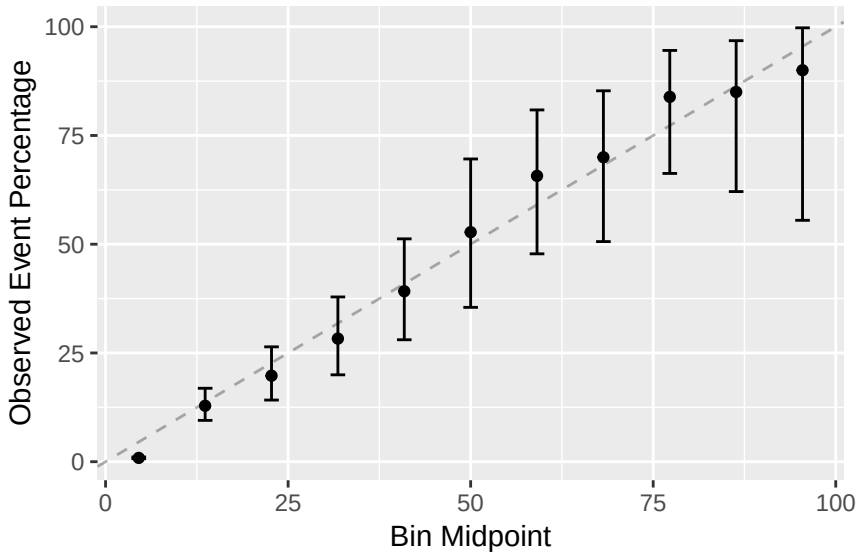
Using your choice of goodness of fit test, determine if the logistic regression model is reliable for inference. Compare the model's descriptive and explanatory power from its predictive accuracy in Problem 3.

```
logitgof(Default$default, fitted(fit1), g = 15)
```

```
## Warning in logitgof(Default$default, fitted(fit1), g = 15): At least one
## cell in the expected frequencies table is < 1. Chi-square approximation may
## be incorrect.
```

```
##
##  Hosmer and Lemeshow test (binary model)
##
## data:  Default$default, fitted(fit1)
## X-squared = 6.1854, df = 13, p-value = 0.9391
```

```
Default$preds <- predict(fit1, type="response")
cal <- calibration(factor(default, levels=c("Yes", "No"))~preds, data = Default)
ggplot(cal, bwidth=2, dwidth = 3)
```



Answer: Both the Hoslem-Lemeshow test and the calibration plot indicate that the model is a good fit, this in addition to producing accurate predictions.

Problem 5

Add your predicted probabilites to the Default data frame and call them `preds`. Use the following code to plot the ROC curve. Calculate the confusion matrix and AUC. Comment on the results results.

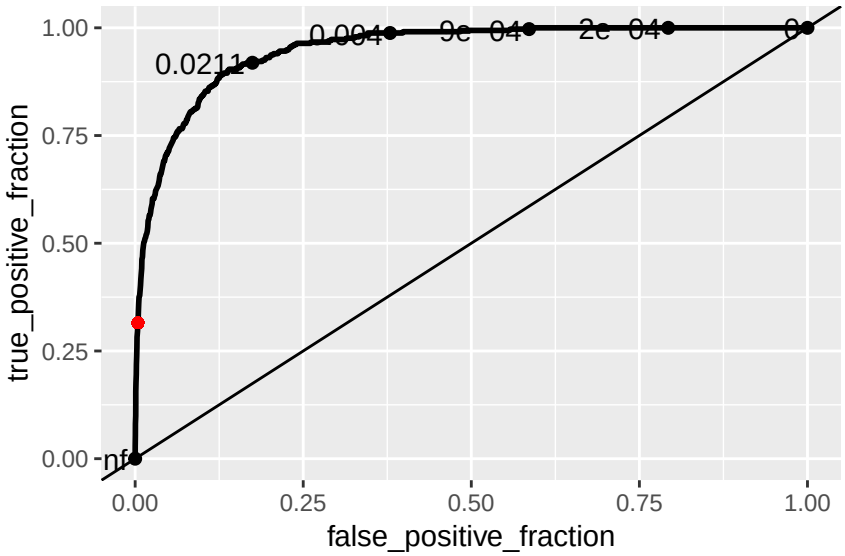
```
Default <-
  Default %>%
  mutate(default_num = as.numeric(default)-1)

# Check true positive and false positive rate when prob = 0.5
ModelMetrics::confusionMatrix(Default$default_num, Default$preds, cutoff= 0.5)
```

```
##      [,1] [,2]
## [1,] 9627  228
## [2,]   40  105
```

```
TPR <- 105/(105+228)
FPR <- 40/(40+9627)

ggplot(Default, aes(d=default_num, m=preds)) +
  geom_roc(n.cuts = 6, labelround = 4) +
  geom_abline(intercept = 0, slope = 1) +
  geom_point(aes(x=FPR, y=TPR), col="red")
```




```
#AUC
auc(Default$default_num, Default$preds)
```

```
## [1] 0.9495581
```

```
# Number of true positives and negatives
table(Default$default)
```

```
##
##   No  Yes
## 9667  333
```

```
# Two ways of getting confusion matrix
Default <-
  Default %>%
  mutate(default_hat = factor(ifelse(preds < 0.5, "No", "Yes"),
                              levels = c("No", "Yes")))
table(Default$default_hat, Default$default)
```

```
##
##         No  Yes
##   No  9627  228
##   Yes   40  105
```

```
# Second method
ModelMetrics::confusionMatrix(Default$default_num, Default$preds)
```

```
##      [,1] [,2]
## [1,] 9627  228
## [2,]   40  105
```

Answer: The ROC plot also indicates a good model, because there is a small true positive fraction when the cutoff is small and it increases much more rapidly than the true positive fraction. The AUC is close to 1, also indicating a good model. From the confusion matrix we can see that this data set is unbalanced. We have many more negatives than positives and the number of false positives is more than twice as high as the number of true positives our balanced accuracy is 0.85.