Predicting Retention of First-Year College Students

EDLD 654 Machine Learning (Fall 2022)

Amy N. Warnock

December 9, 2022

Research Problem

The purpose of this project is to predict the fourth-year retention of undergraduate students based on their responses to a survey of engagement administered during the spring of their first year and various demographic characteristics. The benefits of having a predictive tool for student retention would be the ability to identify students who may be at risk for not persisting and intervening to provide them with support, identify student-level predictors that are malleable and which can be intervened upon, identify institutional-level predictors that the UO can address (e.g., programming, services, quality of instruction, factors related to sense of belonging), and to hopefully improve fourth-year retention rates.

Description of the Data

The participants in the dataset are 531 first-year students who attended UO during the 2018-2019 school year and responded to the National Survey of Student Engagement (NSSE) in the spring of 2019. In addition to responses and some demographics collected during the administration of NSSE, additional demographic variables and retention to the fall of the students' fourth year were sourced from UO's Integrative Data and Reporting (IDR) data warehouse, which stores a vast amount of institutional student records. I selected 32 variables from the original NSSE dataset and IDR for this project. These variables include a fictitious student identification number, fourth-year retention outcome (i.e., whether or not the student went on to register for fall term of their fourth year), and 30 predictors representing a range of demographic variables, responses to single-items from NSSE, and scale scores of NSSE engagement indicators. The NSSE and variables are described in further detail below, along with output from the `ff_glimpse()` summarizing descriptives and frequences for all variables.

The National Survey of Student Engagement (NSSE)

The National Survey of Student Engagement (NSSE) is a survey administered to first-year and senior undergraduate students at hundreds of four-year institutions across the United States and Canada. It has been administered approximately every 2 years to students at UO since 2003. The purposes of collecting this data include national benchmarking purposes (i.e., how responses from UO students compare to other institutions), accountability (i.e., evidence of quality of education and programs), and for program review and assessment purposes (i.e., identifying areas for improvement).

In addition to collecting some demographic information, the NSSE ask students about their engagement and participation with campus programs, services, and activities, their learning, and their experiences. NSSE has developed 10 "Engagement Indicators" representing different areas of student engagement, experiences, and perceptions and provides scale scores for each indicator. The Engagement Indicators are Higher-Order

Learning (HO), Reflective & Integrative Learning (RI), Learning Strategies (LS), Quantitative Reasoning (QR), Collaborative Learning (CL), Discussions with Diverse Others (DD), Student-Faculty Interaction (SF), Effective Teaching Practices (ET), Quality of Interactions (QI), and Supportive Environment (SE). I included all 10 indicators as predictors in this analysis. All of the engagement indicators are numeric scale scores. In addition, I included responses to the following single items, which are not represented in the indicator scale scores: `returnexp` (whether the student expects to return), `evalexp` (how the student rates their educational experience at UO), `edaspire` (the highest level of education the student plans to complete), `tmworkhrs` (the number of hours working for pay each week), `tmcocurrhrs` (the number of hours participating in cocurricular activities each week), `tmservicehrs` (number of hours volunteering each week), `tmcommutehrs` (hours commuting per week), `tmcarehrs` (hours caring for a dependent each week), `tmreadinghrs` (hours spent reading for courses each week), and `tmrelaxhrs` (hours spent relaxing each week). The `returnexp`, `evalexp`, and `edaspire` variables are categorical. All weekly hour time variables are numeric.

Demographic Variables (Institution-Reported and NSSE)

The institution-reported demographic variables included in this analysis were race/ethnicity, Oregon residency status, first-generation status, and binary gender. Race/ethnicity was recoded into three categories in order to protect student privacy and confidentiality (i.e., cell sizes were < 10 for some original categories). The three categories were Traditionally Marginalized Domestic Students, White, and Multiracial. International students were not included in this analysis. Traditionally Marginalized Domestic Students are non-International students whose race/ethnicity is reported as American Indian/Alaska Native, Asian, Black, Native Hawaiian/Pacific Islander, or Hispanic or Latine/a/o/x. Oregon residency status is whether or not the student is considered a resident of Oregon by the UO. First-generation status is determined by the highest level of education obtained by parents/guardians. Students are considered first generation if neither parent/guardian attained a bachelor's degree. Continuing-generation students are those students with at least one parent/guardian who obtained a bachelor's degree or higher. Binary gender (male or female) was selected to protect student privacy and confidentiality (i.e., cell sizes for non-binary gender categories were < 10).

Demographic variables collected by NSSE were the student's living situation (e.g., campus housing, residence within walking distance, residence further than walking distance, etc.), student athlete status (student athlete or not), Greek life involvement (involved or not), disability (e.g., sensory impairment, mobility impairment, learning disability, mental health condition, etc.), and category of first major (e.g., arts and humanities, business, etc.).

All demographic variables are categorical.

Retention

The retention variable in this analysis was pulled from the UO IDR data warehouse. It corresponds to whether or not the student registered for and attended fall term of their fourth year at UO. The variable is coded as 1 = "did not attend fall term of their fourth year", 0 = "did register and attend fall term of their fourth year. The positive was assigned in this way because we would like to be able to predict the students who are at risk of not persisting to their fourth year.

Missing Data Analysis and Descriptive Statistics

I evaluated all variables for missingness using the `ff_glimpse()` function. There were not any variables with more than 75% of observations missing, so no variables were removed prior to analysis. During the data preparation phase, I used imputation for missing values. Mean imputation was used for numeric variables, while mode imputation was used for categorical variables.

```
ff_glimpse(data)
```

```
## $Continuous
##
## id
## ret_fall_term_yr4
## tmworkhrs         Estimated number of hrs working for pay recoded and summed by NSSE from tmworkonhrs
## tmcocurrhrs                                                                    Estimated hours: tmcocurr re
## tmservicehrs                                                                   Estimated hours: tmservice re
## tmcommutehrs                                                                   Estimated hours: tmcommute re
## tmcarehrs                                                                        Estimated hours: tmcare re
## tmreadinghrs
## tmrelaxhrs                                                                     Estimated hours: tmrelax re
## HO
## RI
## LS
## QR
## CL
## DD
## SF
## ET
## QI
## SE
##                    var_type   n missing_n missing_percent      mean    sd
## id                    <dbl> 531         0             0.0 2115004.0 153.4
## ret_fall_term_yr4     <dbl> 531         0             0.0       0.2   0.4
## tmworkhrs             <dbl> 445        86            16.2       5.3   9.5
## tmcocurrhrs           <dbl> 442        89            16.8       5.1   6.1
## tmservicehrs          <dbl> 446        85            16.0       2.3   4.2
## tmcommutehrs          <dbl> 448        83            15.6       3.4   4.9
## tmcarehrs             <dbl> 448        83            15.6       0.8   3.7
## tmreadinghrs          <dbl> 443        88            16.6       7.6   5.8
## tmrelaxhrs            <dbl> 447        84            15.8      13.5   8.2
## HO                    <dbl> 463        68            12.8      37.8  12.1
## RI                    <dbl> 491        40             7.5      36.9  11.6
## LS                    <dbl> 452        79            14.9      35.6  13.2
## QR                    <dbl> 452        79            14.9      26.9  14.6
## CL                    <dbl> 516        15             2.8      33.3  13.8
## DD                    <dbl> 449        82            15.4      39.0  13.9
## SF                    <dbl> 466        65            12.2      19.9  13.5
## ET                    <dbl> 462        69            13.0      37.1  11.1
## QI                    <dbl> 434        97            18.3      41.7  10.4
## SE                    <dbl> 449        82            15.4      35.7  11.7
##                         min quartile_25    median quartile_75       max
## id                2114739.0   2114871.5 2115004.0   2115136.5 2115269.0
## ret_fall_term_yr4       0.0         0.0       0.0         0.0       1.0
## tmworkhrs               0.0         0.0       0.0         8.0      66.0
## tmcocurrhrs             0.0         0.0       3.0         8.0      33.0
## tmservicehrs            0.0         0.0       0.0         3.0      33.0
## tmcommutehrs            0.0         0.0       3.0         3.0      33.0
## tmcarehrs               0.0         0.0       0.0         0.0      33.0
## tmreadinghrs            0.0         3.2       6.0         9.8      29.7
## tmrelaxhrs              0.0         8.0      13.0        18.0      33.0
```

```
## HO                     0.0        30.0      40.0        45.0      60.0
## RI                     0.0        28.6      37.1        42.9      60.0
## LS                     0.0        26.7      33.3        41.7      60.0
## QR                     0.0        20.0      26.7        40.0      60.0
## CL                     5.0        20.0      35.0        40.0      60.0
## DD                     0.0        30.0      40.0        50.0      60.0
## SF                     0.0        10.0      20.0        25.0      60.0
## ET                     4.0        32.0      36.0        44.0      60.0
## QI                     0.0        36.0      42.0        48.0      60.0
## SE                     5.0        27.5      35.0        42.5      60.0
##
## $Categorical
##                          label var_type   n missing_n missing_percent levels_n
## IRsex19                IRsex19    <fct> 531         0             0.0        2
## race_eth              race_eth    <fct> 507        24             4.5        3
## residency            residency    <fct> 530         1             0.2        2
## firstgen              firstgen    <fct> 506        25             4.7        2
## living18              living18    <fct> 442        89            16.8        6
## athlete                athlete    <fct> 444        87            16.4        2
## greek                    greek    <fct> 444        87            16.4        2
## disability_all disability_all    <fct> 443        88            16.6        8
## MAJfirstcol       MAJfirstcol    <fct> 446        85            16.0       11
## returnexp           returnexp    <fct> 447        84            15.8        3
## sameinst             sameinst    <fct> 444        87            16.4        4
## evalexp               evalexp    <fct> 443        88            16.6        4
## edaspire             edaspire    <fct> 443        88            16.6        4
##                                                    levels
## IRsex19                                           "0", "1"
## race_eth       "Multiracial", "TMDS", "White", "(Missing)"
## residency       "Non-Resident", "OR Resident", "(Missing)"
## firstgen        "Continuing Gen", "First Gen", "(Missing)"
## living18                                                -
## athlete                            "0", "1", "(Missing)"
## greek                              "0", "1", "(Missing)"
## disability_all                                          -
## MAJfirstcol                                             -
## returnexp                     "0", "1", "9", "(Missing)"
## sameinst                 "1", "2", "3", "4", "(Missing)"
## evalexp                  "1", "2", "3", "4", "(Missing)"
## edaspire                 "1", "2", "3", "4", "(Missing)"
##                         levels_count                    levels_percent
## IRsex19                    372, 159                          70, 30
## race_eth           37, 158, 312, 24            7.0, 29.8, 58.8,  4.5
## residency               196, 334, 1              36.91, 62.90,  0.19
## firstgen               338, 168, 25                 63.7, 31.6,  4.7
## living18                          -                               -
## athlete                428, 16, 87                       81,  3, 16
## greek                  384, 60, 87                       72, 11, 16
## disability_all                    -                               -
## MAJfirstcol                       -                               -
## returnexp           16, 418, 13, 84            3.0, 78.7,  2.4, 15.8
## sameinst       15, 52, 215, 162, 87      2.8,  9.8, 40.5, 30.5, 16.4
## evalexp         5, 56, 260, 122, 88  0.94, 10.55, 48.96, 22.98, 16.57
## edaspire      19, 148, 175, 101, 88      3.6, 27.9, 33.0, 19.0, 16.6
```

Description of the Models

Three models were selected to predict student retention outcomes: (a) logistic regression with no regularization, (b) logistic regression with ridge penalty, and (c) logistic regression with lasso penalty. Logistic regression was selected because the outcome, retention, is a binary variable. Each model is increasingly complex. The lambda hyperparameter was tuned for the logistic regression with ridge penalty. For the logistic regression with lasso penalty, I set alpha equal to 1 and tuned the lambda hyperparameter. For all three models, I used an 80-20 training/test data split and 10-fold cross validation. The classification cut point used for all three models was 0.50.

In addition to having a binary dependent variable, assumptions of linear regression include independence of observations and little to no multicollinearity between independent variables. To address any issues of multicollinearity, I include a step in the `recipes()` blueprint to remove any variables that were strongly correlated with each other (i.e., $r > .75$).

Data Preprocessing

Prior to training and fitting the three models, I prepared the data using `recipes()` to assign roles to each variable ("id", "outcome", "predictor") and process the variables. First, I created an indicator variable for missingness for all predictors. This enables the model to assess missingness as a predictor. I then removed all predictors with zero variance, as these would not contribute to the predictive ability of the models. I followed this with mean and mode imputation and normalizing all numeric predictors. Finally, I created dummy variables for all categorical predictors, removed any variables that were strongly correlated with each other (as described previously), and transformed the outcome variable into a factor. The `recipe()` used for the blueprint in this analysis is below.

```
# create objects for categorical and numeric predictors
# (although I did not end up needing numpreds in my recipe)

catpreds <- data %>%
  select(where(is.factor)) %>%
  colnames()

catpreds
```

```
##  [1] "IRsex19"        "race_eth"       "residency"      "firstgen"
##  [5] "living18"       "athlete"        "greek"          "disability_all"
##  [9] "MAJfirstcol"    "returnexp"      "sameinst"       "evalexp"
## [13] "edaspire"
```

```
numpreds <- data %>%
  select(where(is.numeric),
         -id,
         -ret_fall_term_yr4) %>%
  colnames()

numpreds
```

```
##  [1] "tmworkhrs"    "tmcocurrhrs"  "tmservicehrs" "tmcommutehrs" "tmcarehrs"
##  [6] "tmreadinghrs" "tmrelaxhrs"   "HO"           "RI"           "LS"
## [11] "QR"           "CL"           "DD"           "SF"           "ET"
## [16] "QI"           "SE"
```

```r
# Use recipes to create a blueprint

blueprint <- recipe(x = data,
                    vars = colnames(data),
                    roles = c('id', 'outcome', rep('predictor', 30))) %>%
  step_indicate_na(all_predictors()) %>% #indicator of missingness for all preds
  step_zv(all_predictors()) %>% #remove  preds with 0 variance
  step_impute_mean(all_numeric_predictors()) %>% #replace NA w/ mean for num
  step_impute_mode(all_of(catpreds)) %>% #replace NA w/ mode for cat
  step_normalize(all_numeric_predictors()) %>%  #normalize num preds
  step_dummy(all_of(catpreds), one_hot = TRUE) %>%  #dummy code
  step_corr(all_numeric_predictors(), threshold = 0.75) %>% #remove variables
  #which are highly correlated with each other to address assumptions of
  # logistic regression
  step_num2factor(ret_fall_term_yr4,
                  transform = function(x) x + 1,
                  levels = c('Negative', 'Positive'))

blueprint
```

```
## Recipe
##
## Inputs:
##
##       role #variables
##         id          1
##    outcome          1
##  predictor         30
##
## Operations:
##
## Creating missing data variable indicators for all_predictors()
## Zero variance filter on all_predictors()
## Mean imputation for all_numeric_predictors()
## Mode imputation for all_of(catpreds)
## Centering and scaling for all_numeric_predictors()
## Dummy variables from all_of(catpreds)
## Correlation filter on all_numeric_predictors()
## Factor variables from ret_fall_term_yr4
```

```r
# Create training and test datasets using an 80-20 split

set.seed(303949)

loc <- sample(1:nrow(data), round(nrow(data) * 0.8))

data_tr <- data[loc, ]

data_te <- data[-loc, ]
```

```r
# Examine dimensions

dim(data_tr)
```

```
## [1] 425  32
```

```
dim(data_te)
```

```
## [1] 106  32
```

```
# Randomly shuffle the training dataset

set.seed(34322) # for reproducibility

data_tr <- data_tr[sample(nrow(data_tr)), ]
```

```
# Create 10 folds with equal size

folds <- cut(seq(1, nrow(data_tr)), breaks = 10, labels = FALSE)

# Create the list for each fold

my.indices <- vector('list', 10)

for(i in 1:10){
  my.indices[[i]] <- which(folds != i)
}
```

```
cv <- trainControl(method = "cv",
                   index = my.indices,
                   classProbs = TRUE,
                   summaryFunction = mnLogLoss)
```

Evaluating Model Performance

To evaluate model performance, I calculated and compared logLoss (LL), Area Under the Curve (AUC), accuracy (ACC), True Positive Rate (TPR or sensitivity), True Negative Rate (TNR or specificity), and precision (PRE). As the purpose of this project is to predict students who do not persist to their fourth year (i.e., students who are categorized as "positive" for not retaining), I prioritized the AUC and TPR metrics in my evaluation.

Model Fit

Model 1: Logistic Regression With No Regularization

```
# Train the logistic regression model (no regularization) using
# 10-fold cross validation

caret_mod <- caret::train(blueprint,
                          data = data_tr,
                          method = "glm",
                          family = 'binomial',
                          metric = 'logLoss',
                          trControl = cv)
```

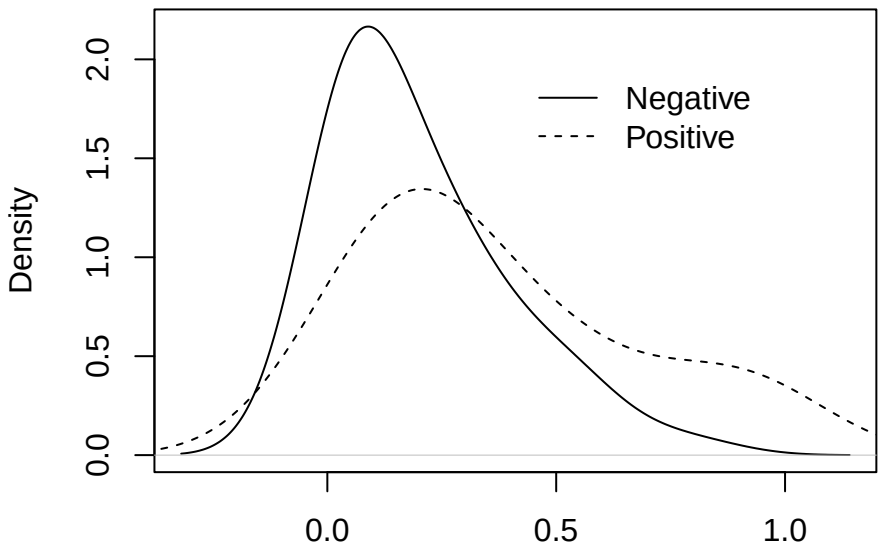
```r
# output - results not included in PDF due to length

caret_mod
```

```
## Generalized Linear Model
##
## 425 samples
##  31 predictor
##   2 classes: 'Negative', 'Positive'
##
## Recipe steps: indicate_na, zv, impute_mean, impute_mode, normalize,
##  dummy, corr, num2factor
## Resampling: Cross-Validated (10 fold)
## Summary of sample sizes: 382, 383, 382, 383, 382, 383, ...
## Resampling results:
##
##   logLoss
##   2.216441
```

```r
# save logLoss

logLoss_noreg <- 2.216441
```

```r
# apply the model to the test dataset

predicted_te <- predict(caret_mod, data_te, type = 'prob')
```

```r
# Plot separation of distributions

group0 <- which(data_te$ret_fall_term_yr4 == 0)

group1 <- which(data_te$ret_fall_term_yr4 == 1)

plot(density(predicted_te[group0, ]$Positive, adjust = 1.5),
     xlab = '', main = '')

points(density(predicted_te[group1, ]$Positive, adjust = 1.5),
       lty = 2, type = 'l')

legend(x = .4, y = 2, c('Negative', 'Positive'), lty = c(1, 2), bty = 'n')
```




```
# Calculate AUC

cut.obj <- cutpointr(x = predicted_te$Positive,
                     class = data_te$ret_fall_term_yr4)

auc_noreg <- auc(cut.obj)

auc_noreg
```

```
## [1] 0.6691358
```

```
# Confusion matrix with the threshold set at .5

pred_class <- ifelse(predicted_te$Positive > .5, 1, 0)

confusion <- table(pred_class, data_te$ret_fall_term_yr4)
```

```
# Accuracy

(confusion[2, 2] + confusion[1, 1]) /
  (confusion[1, 1] + confusion[1, 2] + confusion[2, 1] + confusion[2, 2])
```

```
## [1] 0.745283
```

```r
acc_noreg <- (confusion[2, 2] + confusion[1, 1]) /
  (confusion[1, 1] + confusion[1, 2] + confusion[2, 1] + confusion[2, 2])

# True Positive Rate (sensitivity)

confusion[2, 2] / (confusion[2, 2] + confusion[1, 2])
```

```
## [1] 0.24
```

```r
tpr_noreg <- confusion[2, 2] / (confusion[2, 2] + confusion[1, 2])

# True Negative Rate (specificity)

confusion[1, 1] / (confusion[1, 1] + confusion[2, 1])
```

```
## [1] 0.9012346
```

```r
tnr_noreg <- confusion[1, 1] / (confusion[1, 1] + confusion[2, 1])

# Precision

confusion[2, 2] / (confusion[2, 2] + confusion[2, 1])
```

```
## [1] 0.4285714
```

```r
pre_noreg <- confusion[2, 2] / (confusion[2, 2] + confusion[2, 1])
```

Model 2: Logistic Regression With Ridge Penalty

```r
# Hyperparameter tuning grid for ridge penalty (lambda), alpha = 0

grid <- data.frame(alpha = 0, lambda = c(seq(0, 1, .01)))
```

```r
# Train the logistic regression model

# Sys.time()

caret_mod_ridge <- caret::train(blueprint,
                                data      = data_tr,
                                method    = "glmnet",
                                family    = 'binomial',
                                metric    = 'logLoss',
                                trControl = cv,
                                tuneGrid  = grid)

# Sys.time()
```

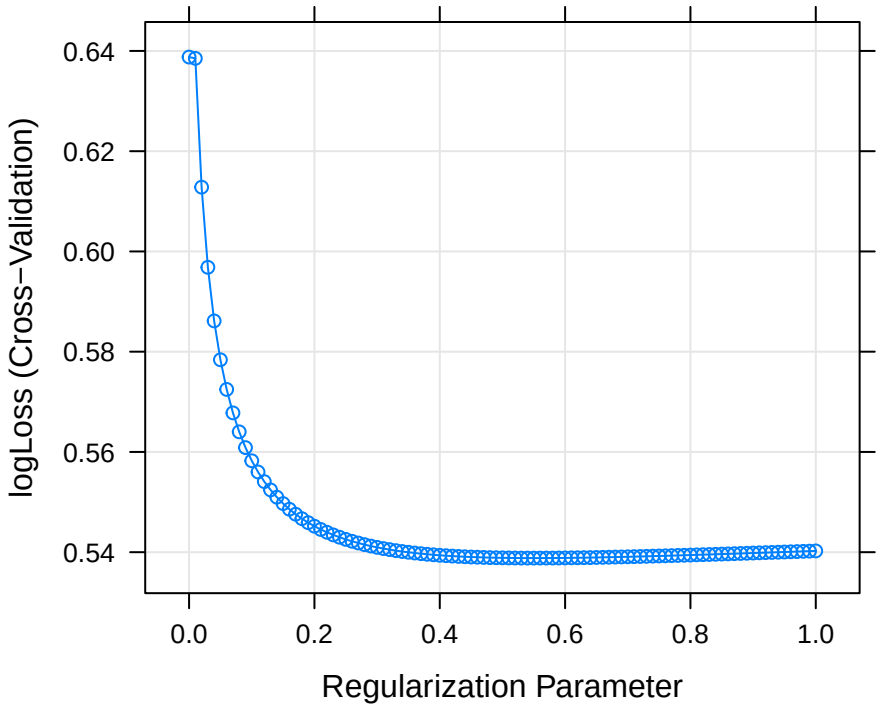
```
# output - results not included in PDF due to length

caret_mod_ridge
```

```
# determine optimal lambda

caret_mod_ridge$bestTune
```

```
##    alpha lambda
## 55     0   0.54
```

```
# save logLoss

logLoss_ridge <- 0.5388230
```

```
plot(caret_mod_ridge)
```




```
# Apply the model to the test dataset

predicted_ridge_te <- predict(caret_mod_ridge, data_te, type = 'prob')
```

```
# AUC

cut.obj.ridge <- cutpointr(x = predicted_ridge_te$Positive,
                           class = data_te$ret_fall_term_yr4)
```

```r
auc_ridge <- auc(cut.obj.ridge)

auc_ridge
```

```
## [1] 0.668642
```

```r
# Ridge confusion matrix with the threshold at 0.5

pred_class_ridge <- ifelse(predicted_ridge_te$Positive > .5, 1, 0)

confusion_ridge <- table(pred_class_ridge, data_te$ret_fall_term_yr4)
```

```r
# Accuracy

(confusion_ridge[2, 2] + confusion_ridge[1, 1]) /
  (confusion_ridge[1, 1] + confusion_ridge[1, 2] + confusion_ridge[2, 1] +
     confusion_ridge[2, 2])
```

```
## [1] 0.7924528
```

```r
acc_ridge <- (confusion_ridge[2, 2] + confusion_ridge[1, 1]) /
  (confusion_ridge[1, 1] + confusion_ridge[1, 2] + confusion_ridge[2, 1] +
     confusion_ridge[2, 2])

# True Positive Rate (sensitivity)

confusion_ridge[2, 2] / (confusion_ridge[2, 2] + confusion_ridge[1, 2])
```

```
## [1] 0.12
```

```r
tpr_ridge <- confusion_ridge[2, 2] /
  (confusion_ridge[2, 2] + confusion_ridge[1, 2])

# True Negative Rate (specificity)

confusion_ridge[1, 1] / (confusion_ridge[1, 1] + confusion_ridge[2, 1])
```

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

```r
tnr_ridge <- confusion_ridge[1, 1] /
  (confusion_ridge[1, 1] + confusion_ridge[2, 1])

# Precision

confusion_ridge[2, 2] / (confusion_ridge[2, 2] + confusion_ridge[2, 1])
```

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

```
pre_ridge <- confusion_ridge[2, 2] /
  (confusion_ridge[2, 2] + confusion_ridge[2, 1])
```

Model 3: Logistic Regression With Lasso Penalty

```
# Hyperparameter tuning grid for ridge penalty (lambda), alpha = 0

grid_lasso <- data.frame(alpha = 1, lambda = seq(0, 1, .01))
```

```
# Train the logistic regression model

# Sys.time()

caret_mod_lasso <- caret::train(blueprint,
                                data      = data_tr,
                                method    = "glmnet",
                                family    = 'binomial',
                                metric    = 'logLoss',
                                trControl = cv,
                                tuneGrid  = grid_lasso)

# Sys.time()
```

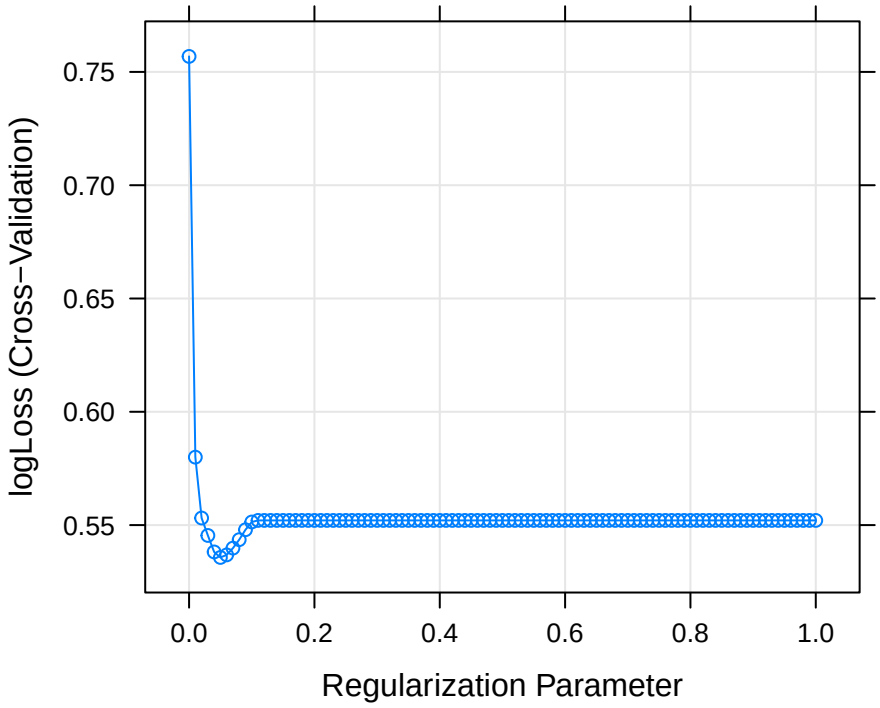
```
# output - results not included in PDF due to length

caret_mod_lasso
```

```
# determine optimal lambda

caret_mod_lasso$bestTune
```

```
##   alpha lambda
## 6     1   0.05
```

```
# save logLoss

logLoss_lasso <- 0.5357162
```

```
plot(caret_mod_lasso)
```




```
# Apply the model to the test dataset

predicted_lasso_te <- predict(caret_mod_lasso, data_te, type = 'prob')
```

```
# AUC

cut.obj.lasso <- cutpointr(x = predicted_lasso_te$Positive,
                           class = data_te$ret_fall_term_yr4,
                           pos_class = 1)

auc_lasso <- auc(cut.obj.lasso)

auc_lasso
```

```
## [1] 0.4051852
```

```
# Lasso confusion matrix with the threshold at 0.5

pred_class_lasso <- ifelse(predicted_lasso_te$Positive > .5, 1, 0)

confusion_lasso <- table(pred_class_lasso, data_te$ret_fall_term_yr4)
```

```
# Accuracy

(confusion_lasso[2, 2] + confusion_lasso[1, 1]) /
  (confusion_lasso[1, 1] + confusion_lasso[1, 2] + confusion_lasso[2, 1] +
     confusion_lasso[2, 2])
```

```
## [1] 0.8018868
```

```
acc_lasso <- (confusion_lasso[2, 2] + confusion_lasso[1, 1]) /
  (confusion_lasso[1, 1] + confusion_lasso[1, 2] + confusion_lasso[2, 1] +
     confusion_lasso[2, 2])

# True Positive Rate (sensitivity)

confusion_lasso[2, 2] / (confusion_lasso[2, 2] + confusion_lasso[1, 2])
```

```
## [1] 0.16
```

```
tpr_lasso <- confusion_lasso[2, 2] /
  (confusion_lasso[2, 2] + confusion_lasso[1, 2])

# True Negative Rate (specificity)

confusion_lasso[1, 1] / (confusion_lasso[1, 1] + confusion_lasso[2, 1])
```

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

```
tnr_lasso <- confusion_lasso[1, 1] /
  (confusion_lasso[1, 1] + confusion_lasso[2, 1])

# Precision

confusion_lasso[2, 2] / (confusion_lasso[2, 2] + confusion_lasso[2, 1])
```

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

```
pre_lasso <- confusion_lasso[2, 2] /
  (confusion_lasso[2, 2] + confusion_lasso[2, 1])
```

Model Performance and Comparison

To evaluate and compare performance of the three models, I calculated and tabled the LL, AUC, TPR, TNR, and PRE metrics for each model. I evaluated these in conjunction with the confusion matrix for each model as an accompanying visual aid. The logistic regression model with lasso penalty has the lowest AUC. The logistic regression model with no regularization has approximately the same AUC as the logistic regression model with ridge penalty. However, I would select the model with no penalty given its higher TPR. As the purpose of this predictive model would be to predict students who may not retain to their fourth year, this metric is of more importance than TNR or PRE. Over-predicting would not have negative consequences for students (assuming any actions taken by the institution would not be negative and were not overly intrusive). Alternatively, under-predicting would mean missing more students who may not persist to their fourth year. Actions taken by the institution for students predicted as being at risk would not be punitive. There would be no harm in contacting a student or connecting them with supports in the event they were falsely predicted as being at risk. There may be resource considerations, depending on what intervention or prevention measures were taken.

Table 1: Model Performance Metrics

Model	LL	AUC	ACC	TPR	TNR	PRE
Logistic Regression: No Regularization	2.216	0.669	0.745	0.24	0.901	0.429
Logistic Regression: Ridge Penalty	0.539	0.669	0.792	0.12	1.000	1.000
Logistic Regression: Lasso Penalty	0.536	0.405	0.802	0.16	1.000	1.000

```
# Confusion matrix for logistic regression with no regularization

confusion
```

```
##
## pred_class  0  1
##          0 73 19
##          1  8  6
```

```
# Confusion matrix for logistic regression with ridge penalty

confusion_ridge
```

```
##
## pred_class_ridge  0  1
##                0 81 22
##                1  0  3
```

```
# Confusion matrix for logistic regression with lasso penalty

confusion_lasso
```

```
##
## pred_class_lasso  0  1
##                0 81 21
##                1  0  4
```

Discussion and Conclusion

It was surprising to me that the logistic regression model with no regularization performed the best. After selecting the logistic regression model with no regularization, I plotted its top-ten predictors. Interestingly, the Quantitative Reasoning (QR) and Reflective & Integrative Learning (RI) were the top two predictors. Less surprisingly, the student anticipating that they would return to UO was the third-most important predictors. Other predictors of importance included responding "probably yes" that they would choose to go to UO if they could do their education over again, the missing indicator for RI, Learning Strategies (LS), the missing indicator for first-generation status, Student-Faculty Interaction (SF), being an Oregon resident, and Quality of Interactions (QI). I think these results may shed some insight on areas of engagement that are related to student retention. My next step will be to examine the individual items that are used to calculate the QR, RI, LS, SF, and QI indicator scores to see if there are patterns, insights, or potentially areas that could be improved by UO. I think it would also be worthwhile conducting a similar analysis with a greater sample size and with additional variables included. It is worth noting that not all first-year students complete the survey, and it is not administered every year. Regardless, results from these models may still be useful in identifying institutional-level areas of importance that are working well or need improvement.

```r
# plot of top 10 predictors

vip(caret_mod, num_features = 10, geom = "point") +
  theme_minimal() +
  theme(plot.title.position = "plot") +
  labs(title = "Top 10 Features of Log. Regression With No Regularization")
```