Adv Data mining group project 2

Libraries required

Loading the Data

```
kent_rt <- as.data.frame(read.csv("COB RT Data.csv", na = c("", "---")))
kent_ct <- as.data.frame(read.csv("COB CT Data.csv", na = c("", "---")))
```

Data Cleaning

Removing the variables that are not significant

```
variables_to_rem <- (c("StudNum","GENDER","TERM", "CAMPUS", "STATE_CODE","ETHNICITY_COD
E", "COUNTY_CODE","ZIP_CODE", "RESIDENCY_CODE", "LEGAL_COUNTRY", "LEGAL_COUNTRY_DESC",
"HIGH_SCHOOL_CODE", "HIGH_SCHOOL_DESC", "TRANS_HRS", "F1SEQ1_COLLEGE","F1SEQ1_DEGREE_1",
"F1SEQ1_MAJOR", "F1SEQ2_MAJOR", "F1SEQ1_CUMGPA","F1SEQ1_CUMERNHRS", "F1SEQ2_COLLEGE", "F
1SEQ2_CUMERNHRS", "F1SEQ2_MIDTERM_GPA","F1SEQ2_COLLEGE", "F1SEQ2_CUM_GPA", "S1SEQ1_CAMPU
S","S1SEQ1_COLLEGE",  "S1SEQ1_MAJOR", "S1SEQ2_MAJOR", "S1SEQ2_COLLEGE", "S1SEQ2_CUMERNHR
S", "S1SEQ2_MIDTERM_GPA", "S1SEQ2_CUM_GPA", "F2SEQ1_CAMPUS",  "F2SEQ1_COLLEGE", "F2SEQ1_
MAJOR","F2SEQ2_MAJOR", "F2SEQ2_COLLEGE", "F2SEQ2_CUMERNHRS", "F2SEQ2_MIDTERM_GPA", "F2SE
Q2_CUM_GPA", "S2SEQ1_CAMPUS", "S2SEQ1_COLLEGE", "S2SEQ1_MAJOR", "S2SEQ2_MAJOR", "S2SEQ2_
COLLEGE", "S2SEQ2_CUMERNHRS", "S2SEQ2_MIDTERM_GPA", "S2SEQ2_CUM_GPA", "F3SEQ1_CAMPUS",
"F3SEQ1_COLLEGE","F3SEQ1_MAJOR", "F3SEQ2_MAJOR","F3SEQ2_COLLEGE", "F3SEQ2_CUMERNHRS", "F
3SEQ2_MIDTERM_GPA","F3SEQ2_CUM_GPA", "S3SEQ1_CAMPUS","S3SEQ1_COLLEGE", "S3SEQ1_MAJOR","S
3SEQ2_MAJOR","S3SEQ2_COLLEGE","S3SEQ2_CUMERNHRS","S3SEQ2_MIDTERM_GPA","S3SEQ2_CUM_GPA",
"F4SEQ1_CAMPUS","F4SEQ1_COLLEGE","F4SEQ1_MAJOR","F4SEQ2_MAJOR","F4SEQ2_COLLEGE","F4SEQ2_
CUMERNHRS","F4SEQ2_MIDTERM_GPA","F4SEQ2_CUM_GPA","S4SEQ1_CAMPUS","S4SEQ1_COLLEGE","S4SEQ
1_MAJOR","S4SEQ2_MAJOR","S4SEQ2_COLLEGE","S4SEQ2_CUMERNHRS","S4SEQ2_CUM_GPA","F5SEQ1_CAM
PUS","F5SEQ1_COLLEGE","F5SEQ1_MAJOR","F5SEQ2_MAJOR","F5SEQ2_COLLEGE","F5SEQ2_CUMERNHRS",
"F5SEQ2_MIDTERM_GPA","F5SEQ2_CUM_GPA","S5SEQ1_CAMPUS","S5SEQ1_COLLEGE", "S5SEQ1_MAJOR",
"S5SEQ2_MAJOR", "S5SEQ2_COLLEGE", "S5SEQ2_CUMERNHRS", "S5SEQ2_MIDTERM_GPA","S5SEQ2_CUM_G
PA",  "F6SEQ1_CAMPUS","F6SEQ1_COLLEGE", "F6SEQ1_MAJOR","F6SEQ2_MAJOR","F6SEQ2_COLLEGE",
"S4SEQ2_MIDTERM_GPA","F6SEQ2_CUMERNHRS","F6SEQ2_MIDTERM_GPA","F6SEQ2_CUM_GPA","S6SEQ1_CA
MPUS","S6SEQ1_COLLEGE","S6SEQ1_MAJOR","S6SEQ2_MAJOR","S6SEQ2_COLLEGE","S6SEQ2_CUMERNHRS"
,"S6SEQ2_MIDTERM_GPA","S6SEQ2_CUM_GPA", "GRAD_TERM_BACHELOR", "GRAD_COLLEGE_BACHELOR",
"GRAD_MAJOR_BACHELOR"))
```

```
kent_na <- kent_rt %>% dplyr::select(-c(variables_to_rem))
```

```
head(kent_na)
```

```
##   URS_IND ONCAMPUS_IND FIRST_GEN_IND PELL_ELIG_IND AGE INTERNATIONAL_IND
## 1       N            N             N             N  18                 N
## 2       N            Y             N             Y  20                 N
## 3       N            Y             Y             N  18                 N
## 4       N            Y             N             N  18                 Y
## 5       N            Y             N             N  19                 N
## 6       N            Y             N             Y  18                 N
##   HIGH_SCHOOL_GPA ATHLETE_IND VETERAN_IND HONORS_REGISTERED_IND ACT_ENGL
## 1              NA           N        <NA>                  <NA>       22
## 2            2.66           N           N                     N       19
## 3            3.22           N        <NA>                  <NA>       18
## 4            3.64           N        <NA>                  <NA>       NA
## 5              NA           N        <NA>                  <NA>       18
## 6              NA           N        <NA>                  <NA>       31
##   ACT_MATH ACT_SOC ACT_NSCI ACT_WRITING ACT_COMP F1SEQ2_CURATTHRS
## 1       27      22       24          NA       24               16
## 2       17      16       21          NA       18               14
## 3       17      16       21          NA       18               15
## 4       NA      NA       NA          NA       NA               17
## 5       25      20       23          NA       21               15
## 6       26      35       27          NA       30               14
##   F1SEQ2_CURERNHRS F1SEQ2_TERM_GPA RET_S1 S1SEQ2_CURATTHRS
## 1               16            2.94      Y               15
## 2               14            1.81      Y               14
## 3               15            3.39      Y               16
## 4               14            2.24      Y               15
## 5               12            2.04      N                0
## 6               14            3.54      Y               16
##   S1SEQ2_CURERNHRS S1SEQ2_TERM_GPA RET_F2 F2SEQ2_CURATTHRS
## 1               15            1.98      Y               16
## 2               12            3.06      Y               17
## 3               13            2.98      Y               14
## 4               15            3.36      Y               19
## 5                0            0.00      Y               15
## 6               16            3.45      Y               12
##   F2SEQ2_CURERNHRS F2SEQ2_TERM_GPA RET_S2 S2SEQ2_CURATTHRS
## 1               16            2.88      Y               15
## 2               17            2.75      Y               18
## 3               14            2.78      Y               16
## 4               12            2.18      Y               16
## 5               15            2.78      Y               18
## 6               12            3.75      Y               15
##   S2SEQ2_CURERNHRS S2SEQ2_TERM_GPA RET_F3 F3SEQ2_CURATTHRS
## 1               12            2.10      Y               15
## 2               14            1.85      Y               18
## 3               14            3.01      Y               18
## 4               13            3.08      Y               18
## 5               18            2.07      Y               16
## 6               12            3.78      Y               15
##   F3SEQ2_CURERNHRS F3SEQ2_TERM_GPA RET_S3 S3SEQ2_CURATTHRS
## 1               15            2.46      Y               15
## 2               13            1.74      Y               15
## 3               12            1.68      Y               16
## 4               11            3.14      Y               18
## 5               16            2.68      Y                9
## 6               15            3.74      Y               15
##   S3SEQ2_CURERNHRS S3SEQ2_TERM_GPA RET_F4 F4SEQ2_CURATTHRS
## 1               12            2.93      Y               12
## 2               12            1.90      Y               15
## 3               13            1.80      Y               19
## 4               12            1.93      Y               19
## 5                9            2.67      Y               15
## 6               15            3.68      Y               15
##   F4SEQ2_CURERNHRS F4SEQ2_TERM_GPA RET_S4 S4SEQ2_CURATTHRS
## 1               12            2.60      Y               12
## 2                6            1.35      Y               15
## 3               16            1.98      Y               18
## 4               13            1.35      Y               18
## 5               12            3.58      Y               18
## 6               15            3.28      Y               12
##   S4SEQ2_CURERNHRS S4SEQ2_TERM_GPA RET_F5 F5SEQ2_CURATTHRS
## 1               12            2.58      Y                9
## 2                3            2.00      N                0
## 3               12            1.65      Y               15
## 4                9            1.75      Y               16
## 5               18            2.78      N                0
## 6               12            3.33      Y               15
##   F5SEQ2_CURERNHRS F5SEQ2_TERM_GPA RET_S5 S5SEQ2_CURATTHRS
## 1                9            2.57      Y                6
## 2                0            0.00      N                0
## 3               12            2.58      Y               18
## 4               15            2.94      Y                6
## 5                0            0.00      N                0
## 6               15            3.66      N                0
##   S5SEQ2_CURERNHRS S5SEQ2_TERM_GPA RET_F6 F6SEQ2_CURATTHRS
## 1                6            2.15      N                0
## 2                0            0.00      N                0
## 3               12            3.08      Y               15
## 4                6            2.35      N                0
## 5                0            0.00      N                0
## 6                0            0.00      N                0
##   F6SEQ2_CURERNHRS F6SEQ2_TERM_GPA RET_S6 S6SEQ2_CURATTHRS
## 1                0             0.0      N                0
## 2                0             0.0      N                0
## 3                9             3.1      Y                9
## 4                0             0.0      N                0
## 5                0             0.0      N                0
## 6                0             0.0      N                0
##   S6SEQ2_CURERNHRS S6SEQ2_TERM_GPA GRADUATEIND
## 1                0            0.00           Y
## 2                0            0.00           N
## 3                6            2.65           N
## 4                0            0.00           Y
## 5                0            0.00           Y
## 6                0            0.00           Y
```

```
graduated <- kent_na$GRADUATEIND
```

KNN Imputation

```
kent_knn <- as.data.frame((kent_na[,1:16]))
kent_knn <- knnImputation(kent_knn)
kent_zero <- kent_na[,17:63]
kent_zero <- kent_zero %>% mutate_all(funs(replace_na(.,0)))
```

```
kent <- cbind(kent_knn, kent_zero)
kent <- cbind(kent, graduated)
```

Spring semester 1

Spring Semester 1 (retain)

```
s1_r <- (kent[,1:20])
s1_r <- cbind(s1_r,graduated)
```

```
sample_train<- sample(seq_len(nrow(s1_r)), size = floor(0.80*nrow(s1_r)))
sample_test <- sample(seq_len(nrow(s1_r)), size = floor(0.20*nrow(s1_r)))

s1_r_train <- s1_r[sample_train, ]
s1_r_test  <- s1_r[sample_test, ]
```

```
retain_s1 <- glm(RET_S1 ~ ., family = binomial, data = s1_r_train)
predict_s1_r <- predict(retain_s1, s1_r_test, type = "response")
```

```
prob <- as.factor(ifelse(as.numeric(predict_s1_r > .25)==1, "Y", "N"))
confusionMatrix(data = prob, reference = s1_r_test$RET_S1, mode = "prec_recall")
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction    N    Y
##          N    4    0
##          Y   59 1126
##
##                Accuracy : 0.9504
##                  95% CI : (0.9365, 0.962)
##     No Information Rate : 0.947
##     P-Value [Acc > NIR] : 0.3308
##
##                   Kappa : 0.1138
##
##  Mcnemar's Test P-Value : 4.321e-14
##
##               Precision : 1.000000
##                  Recall : 0.063492
##                      F1 : 0.119403
##              Prevalence : 0.052986
##          Detection Rate : 0.003364
##    Detection Prevalence : 0.003364
##       Balanced Accuracy : 0.531746
##
##        'Positive' Class : N
##
```

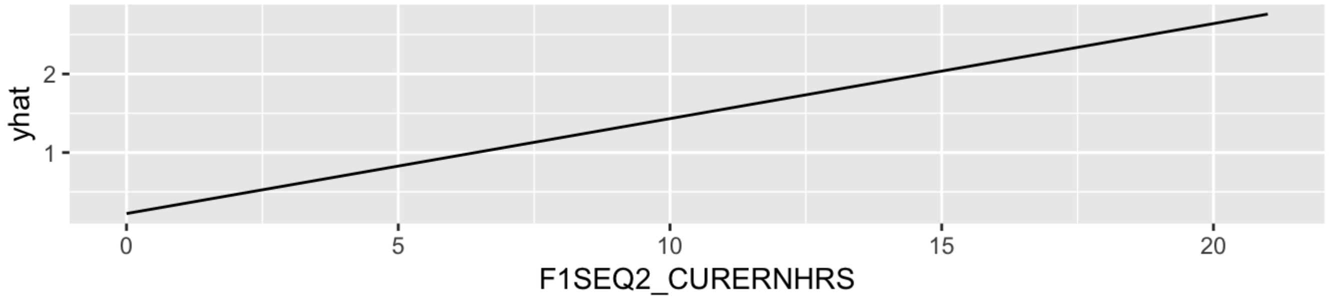
```
roc(s1_r_test$RET_S1, as.numeric(predict_s1_r))
```

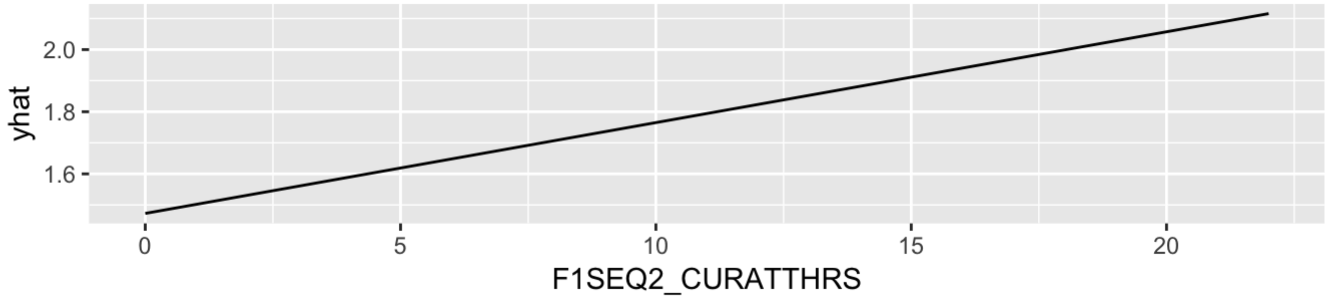
```
##
## Call:
## roc.default(response = s1_r_test$RET_S1, predictor = as.numeric(predict_s1_r))
##
## Data: as.numeric(predict_s1_r) in 63 controls (s1_r_test$RET_S1 N) < 1126 cases (s1_r
_test$RET_S1 Y).
## Area under the curve: 0.8311
```

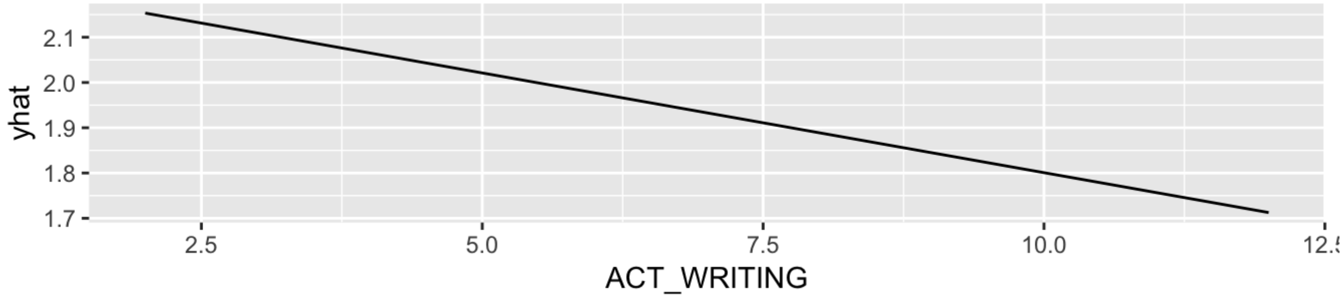
Partial dependency plot for Spring semester 1 (retain) - top 3 variables

```
s1_r_imp <- varImp(retain_s1, scale = FALSE)
```

```
par_s1_r_erhr <- partial(retain_s1, pred.var = c("F1SEQ2_CURERNHRS"), chull = TRUE)
plot_s1_r_erhr  <- autoplot(par_s1_r_erhr , contour = TRUE)
par_s1_r_athr  <- partial(retain_s1, pred.var = c("F1SEQ2_CURATTHRS"), chull = TRUE)
plot_s1_r_athr  <- autoplot(par_s1_r_athr, contour = TRUE)
par_s1_r_act  <- partial(retain_s1, pred.var = c("ACT_WRITING"), chull = TRUE)
plot_s1_r_act  <- autoplot(par_s1_r_act, contour = TRUE)
grid.arrange(plot_s1_r_erhr, plot_s1_r_athr, plot_s1_r_act)
```







Spring Semester 1 (graduate)

```
s1_g <- s1_r %>% filter(RET_S1 == "Y")
s1_g <- s1_g %>% dplyr::select(-(RET_S1))
```

```
sample_train<- sample(seq_len(nrow(s1_g)), size = floor(0.80*nrow(s1_g)))
sample_test <- sample(seq_len(nrow(s1_g)), size = floor(0.20*nrow(s1_g)))

s1_g_train <- s1_g[sample_train, ]
s1_g_test  <- s1_g[sample_test, ]
```

```
graduate_s1 <- glm(graduated ~ ., family = binomial, data = s1_g_train)
predict_s1_g <- predict(graduate_s1, s1_g_test, type = "response")
```

```
prob <- as.factor(ifelse(as.numeric(predict_s1_g > .25)==1, "Y", "N"))
confusionMatrix(data = prob, reference = s1_g_test$graduated, mode = "prec_recall")
```

```
## Confusion Matrix and Statistics
## 
##           Reference
## Prediction   N   Y
##          N  78  15
##          Y 515 515
##                                           
##                Accuracy : 0.528           
##                  95% CI : (0.4984, 0.5576)
##     No Information Rate : 0.528           
##     P-Value [Acc > NIR] : 0.5121          
##                                           
##                   Kappa : 0.0983          
##                                           
##  Mcnemar's Test P-Value : <2e-16          
##                                           
##               Precision : 0.83871         
##                  Recall : 0.13153         
##                      F1 : 0.22741         
##              Prevalence : 0.52805         
##          Detection Rate : 0.06946         
##    Detection Prevalence : 0.08281         
##       Balanced Accuracy : 0.55162         
##                                           
##        'Positive' Class : N               
## 
```

```
roc(s1_g_test$graduated, as.numeric(predict_s1_g))
```

```
## 
## Call:
## roc.default(response = s1_g_test$graduated, predictor = as.numeric(predict_s1_g))
## 
## Data: as.numeric(predict_s1_g) in 593 controls (s1_g_test$graduated N) < 530 cases (s
1_g_test$graduated Y).
## Area under the curve: 0.6361
```

Fall Semester 2

Fall Semester 2 (retain)

```
f2_r <- cbind(kent[,1:24], graduated) %>% filter(RET_S1 == "Y")
f2_r <- f2_r %>% dplyr::select(-RET_S1)
```

```
sample_train <- sample(seq_len(nrow(f2_r)), size = floor(0.80*nrow(f2_r)))
sample_test <- sample(seq_len(nrow(f2_r)), size = floor(0.20*nrow(f2_r)))

f2_r_train <- f2_r[sample_train, ]
f2_r_test  <- f2_r[sample_test, ]
```

```
retain_f2 <- glm(RET_F2 ~ ., family = binomial, data = f2_r_train)
predict_f2_r <- predict(retain_f2, f2_r_test, type = "response")
```

```
prob <- as.factor(ifelse(as.numeric(predict_f2_r > .25)==1, "Y", "N"))
confusionMatrix(data = prob, reference = f2_r_test$RET_F2, mode = "prec_recall")
```

```
## Confusion Matrix and Statistics
## 
##           Reference
## Prediction   N   Y
##          N  20   5
##          Y 103 995
## 
##                Accuracy : 0.9038
##                  95% CI : (0.8851, 0.9204)
##     No Information Rate : 0.8905
##     P-Value [Acc > NIR] : 0.08107
## 
##                   Kappa : 0.2422
## 
##  Mcnemar's Test P-Value : < 2e-16
## 
##               Precision : 0.80000
##                  Recall : 0.16260
##                      F1 : 0.27027
##              Prevalence : 0.10953
##          Detection Rate : 0.01781
##    Detection Prevalence : 0.02226
##       Balanced Accuracy : 0.57880
## 
##        'Positive' Class : N
## 
```

```
roc(f2_r_test$RET_F2, as.numeric(predict_f2_r))
```

```
## 
## Call:
## roc.default(response = f2_r_test$RET_F2, predictor = as.numeric(predict_f2_r))
## 
## Data: as.numeric(predict_f2_r) in 123 controls (f2_r_test$RET_F2 N) < 1000 cases (f2_
r_test$RET_F2 Y).
## Area under the curve: 0.88
```

Fall Semester 2 (graduate)

```
f2_g <- f2_r %>% filter(RET_F2 == "Y")
f2_g <- f2_g %>% dplyr::select(-RET_F2)
```

```
sample_train<- sample(seq_len(nrow(f2_g)), size = floor(0.80*nrow(f2_g)))
sample_test <- sample(seq_len(nrow(f2_g)), size = floor(0.20*nrow(f2_g)))

f2_g_train <- f2_g[sample_train, ]
f2_g_test  <- f2_g[sample_test, ]
```

```
graduate_f2 <- glm(graduated ~ ., family = binomial, data = f2_g_train)
predict_f2_g <- predict(graduate_f2, f2_g_test, type = "response")
```

```
prob <- as.factor(ifelse(as.numeric(predict_f2_g > .25)==1, "Y", "N"))
confusionMatrix(data = prob, reference = f2_g_test$graduated, mode = "prec_recall")
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction   N   Y
##          N  31   6
##          Y 466 489
##
##                Accuracy : 0.5242
##                  95% CI : (0.4926, 0.5557)
##     No Information Rate : 0.501
##     P-Value [Acc > NIR] : 0.07652
##
##                   Kappa : 0.0502
##
##  Mcnemar's Test P-Value : < 2e-16
##
##               Precision : 0.83784
##                  Recall : 0.06237
##                      F1 : 0.11610
##              Prevalence : 0.50101
##          Detection Rate : 0.03125
##    Detection Prevalence : 0.03730
##       Balanced Accuracy : 0.52513
##
##        'Positive' Class : N
##
```

```
roc(f2_g_test$graduated, as.numeric(predict_f2_g))
```

```
##
## Call:
## roc.default(response = f2_g_test$graduated, predictor = as.numeric(predict_f2_g))
##
## Data: as.numeric(predict_f2_g) in 497 controls (f2_g_test$graduated N) < 495 cases (f
2_g_test$graduated Y).
## Area under the curve: 0.6277
```

Spring Semester 2

Spring Semester 2(retain)

```
s2_r <- cbind(kent[,1:28], graduated) %>% filter(RET_F2 == "Y")
s2_r <- s2_r %>% dplyr::select(-RET_F2)
```

```
sample_train<- sample(seq_len(nrow(s2_r)), size = floor(0.80*nrow(s2_r)))
sample_test <- sample(seq_len(nrow(s2_r)), size = floor(0.20*nrow(s2_r)))

s2_r_train <- s2_r[sample_train, ]
s2_r_test  <- s2_r[sample_test, ]
```

```
retain_s2 <- glm(RET_S2 ~ ., family = binomial, data = s2_r_train)
predict_s2_r <- predict(retain_s2, s2_r_test, type = "response")
```

```
prob <- as.factor(ifelse(as.numeric(predict_s2_r > .25)==1, "Y", "N"))
confusionMatrix(data = prob, reference = s2_r_test$RET_S2, mode = "prec_recall")
```

```
## Confusion Matrix and Statistics
## 
##           Reference
## Prediction   N   Y
##          N  77   6
##          Y  38 879
##                                           
##                Accuracy : 0.956           
##                  95% CI : (0.9414, 0.9679)
##     No Information Rate : 0.885           
##     P-Value [Acc > NIR] : 1.966e-15       
##                                           
##                   Kappa : 0.7541          
##                                           
##  Mcnemar's Test P-Value : 2.962e-06       
##                                           
##               Precision : 0.9277          
##                  Recall : 0.6696          
##                      F1 : 0.7778          
##              Prevalence : 0.1150          
##          Detection Rate : 0.0770          
##    Detection Prevalence : 0.0830          
##       Balanced Accuracy : 0.8314          
##                                           
##        'Positive' Class : N               
## 
```

```
roc(s2_r_test$RET_S2, as.numeric(predict_s2_r))
```

```
## 
## Call:
## roc.default(response = s2_r_test$RET_S2, predictor = as.numeric(predict_s2_r))
## 
## Data: as.numeric(predict_s2_r) in 115 controls (s2_r_test$RET_S2 N) < 885 cases (s2_r
_test$RET_S2 Y).
## Area under the curve: 0.9238
```

Spring Semester 2 (graduate)

```
s2_g <- s2_r %>% filter(RET_S2 == "Y")
s2_g <- s2_g %>% dplyr::select(-RET_S2)
```

```
sample_train<- sample(seq_len(nrow(s2_g)), size = floor(0.80*nrow(s2_g)))
sample_test <- sample(seq_len(nrow(s2_g)), size = floor(0.20*nrow(s2_g)))

s2_g_train <- s2_g[sample_train, ]
s2_g_test  <- s2_g[sample_test, ]
```

```
graduate_s2 <- glm(graduated ~ ., family = binomial, data = s2_g_train)
predict_s2_g <- predict(graduate_s2, s2_g_test, type = "response")
```

```
prob <- as.factor(ifelse(as.numeric(predict_s2_g > .25)==1, "Y", "N"))
confusionMatrix(data = prob, reference = s2_g_test$graduated, mode = "prec_recall")
```

```
## Confusion Matrix and Statistics
## 
##           Reference
## Prediction   N   Y
##          N  27   5
##          Y 332 525
## 
##                Accuracy : 0.6209
##                  95% CI : (0.5881, 0.6529)
##     No Information Rate : 0.5962
##     P-Value [Acc > NIR] : 0.07046
## 
##                   Kappa : 0.0771
## 
##  Mcnemar's Test P-Value : < 2e-16
## 
##               Precision : 0.84375
##                  Recall : 0.07521
##                      F1 : 0.13811
##              Prevalence : 0.40382
##          Detection Rate : 0.03037
##    Detection Prevalence : 0.03600
##       Balanced Accuracy : 0.53289
## 
##        'Positive' Class : N
## 
```

```
roc(s2_g_test$graduated, as.numeric(predict_s2_g))
```

```
## 
## Call:
## roc.default(response = s2_g_test$graduated, predictor = as.numeric(predict_s2_g))
## 
## Data: as.numeric(predict_s2_g) in 359 controls (s2_g_test$graduated N) < 530 cases (s
2_g_test$graduated Y).
## Area under the curve: 0.6623
```

Fall semester 3

Fall Semester 3 (retain)

```
f3_r <- cbind(kent[,1:32], graduated) %>% filter(RET_S2 == "Y")
f3_r <- f3_r %>% dplyr::select(-RET_S2)
```

```
sample_train <- sample(seq_len(nrow(f3_r)), size = floor(0.80*nrow(f3_r)))
sample_test <- sample(seq_len(nrow(f3_r)), size = floor(0.20*nrow(f3_r)))

f3_r_train <- f3_r[sample_train, ]
f3_r_test  <- f3_r[sample_test, ]
```

```
retain_f3 <- glm(RET_F3 ~ ., family = binomial, data = f3_r_train)
predict_f3_r <- predict(retain_f3, f3_r_test, type = "response")
```

```
prob <- as.factor(ifelse(as.numeric(predict_f3_r > .25)==1, "Y", "N"))
confusionMatrix(data = prob, reference = f3_r_test$RET_F3, mode = "prec_recall")
```

```
## Confusion Matrix and Statistics
## 
##           Reference
## Prediction   N   Y
##          N   2   0
##          Y  60 837
##                                           
##                Accuracy : 0.9333          
##                  95% CI : (0.9149, 0.9487)
##     No Information Rate : 0.931           
##     P-Value [Acc > NIR] : 0.4289          
##                                           
##                   Kappa : 0.0584          
##                                           
##  Mcnemar's Test P-Value : 2.599e-14       
##                                           
##               Precision : 1.000000        
##                  Recall : 0.032258        
##                      F1 : 0.062500        
##              Prevalence : 0.068966        
##          Detection Rate : 0.002225        
##    Detection Prevalence : 0.002225        
##       Balanced Accuracy : 0.516129        
##                                           
##        'Positive' Class : N               
## 
```

```
roc(f3_r_test$RET_F3, as.numeric(predict_f3_r))
```

```
##
## Call:
## roc.default(response = f3_r_test$RET_F3, predictor = as.numeric(predict_f3_r))
##
## Data: as.numeric(predict_f3_r) in 62 controls (f3_r_test$RET_F3 N) < 837 cases (f3_r_
test$RET_F3 Y).
## Area under the curve: 0.8812
```

Fall Semester 3 (graduate)

```
f3_g <- f3_r %>% filter(RET_F3 == "Y")
f3_g <- f3_g %>% dplyr::select(-RET_F3)
```

```
sample_train<- sample(seq_len(nrow(f3_g)), size = floor(0.80*nrow(f3_g)))
sample_test <- sample(seq_len(nrow(f3_g)), size = floor(0.20*nrow(f3_g)))

f3_g_train <- f3_g[sample_train, ]
f3_g_test  <- f3_g[sample_test, ]
```

```
graduate_f3 <- glm(graduated ~ ., family = binomial, data = f3_g_train)
predict_f3_g <- predict(graduate_f3, f3_g_test, type = "response")
```

```
prob <- as.factor(ifelse(as.numeric(predict_f3_g > .25)==1, "Y", "N"))
confusionMatrix(data = prob, reference = f3_g_test$graduated, mode = "prec_recall")
```

```
## Confusion Matrix and Statistics
## 
##           Reference
## Prediction   N   Y
##          N  11   4
##          Y 320 502
##                                          
##                Accuracy : 0.6129         
##                  95% CI : (0.579, 0.6461)
##     No Information Rate : 0.6045         
##     P-Value [Acc > NIR] : 0.3236         
##                                          
##                   Kappa : 0.0303         
##                                          
##  Mcnemar's Test P-Value : <2e-16         
##                                          
##               Precision : 0.73333        
##                  Recall : 0.03323        
##                      F1 : 0.06358        
##              Prevalence : 0.39546        
##          Detection Rate : 0.01314        
##    Detection Prevalence : 0.01792        
##       Balanced Accuracy : 0.51266        
##                                          
##        'Positive' Class : N              
## 
```

```
roc(f3_g_test$graduated, as.numeric(predict_f3_g))
```

```
## 
## Call:
## roc.default(response = f3_g_test$graduated, predictor = as.numeric(predict_f3_g))
## 
## Data: as.numeric(predict_f3_g) in 331 controls (f3_g_test$graduated N) < 506 cases (f
3_g_test$graduated Y).
## Area under the curve: 0.6723
```

Spring semester 3

Spring Semester 3 (retain)

```
s3_r <- cbind(kent[,1:36], graduated) %>% filter(RET_F3 == "Y")
s3_r <- s3_r %>% dplyr::select(-RET_F3)
```

```
sample_train<- sample(seq_len(nrow(s3_r)), size = floor(0.80*nrow(s3_r)))
sample_test <- sample(seq_len(nrow(s3_r)), size = floor(0.20*nrow(s3_r)))

s3_r_train <- s3_r[sample_train, ]
s3_r_test  <- s3_r[sample_test, ]
```

```
retain_s3 <- glm(RET_S3 ~ ., family = binomial, data = s3_r_train)
predict_s3_r <- predict(retain_s3, s3_r_test, type = "response")
```

```
prob <- as.factor(ifelse(as.numeric(predict_s3_r > .25)==1, "Y", "N"))
confusionMatrix(data = prob, reference = s3_r_test$RET_S3, mode = "prec_recall")
```

```
## Confusion Matrix and Statistics
## 
##           Reference
## Prediction   N   Y
##          N 116   7
##          Y  33 698
##                                          
##                Accuracy : 0.9532         
##                  95% CI : (0.9368, 0.9663)
##     No Information Rate : 0.8255         
##     P-Value [Acc > NIR] : < 2.2e-16      
##                                          
##                   Kappa : 0.8254         
##                                          
##  Mcnemar's Test P-Value : 7.723e-05      
##                                          
##               Precision : 0.9431         
##                  Recall : 0.7785         
##                      F1 : 0.8529         
##              Prevalence : 0.1745         
##          Detection Rate : 0.1358         
##    Detection Prevalence : 0.1440         
##       Balanced Accuracy : 0.8843         
##                                          
##        'Positive' Class : N              
## 
```

```
roc(s3_r_test$RET_S3, as.numeric(predict_s3_r))
```

```
## 
## Call:
## roc.default(response = s3_r_test$RET_S3, predictor = as.numeric(predict_s3_r))
## 
## Data: as.numeric(predict_s3_r) in 149 controls (s3_r_test$RET_S3 N) < 705 cases (s3_r
_test$RET_S3 Y).
## Area under the curve: 0.9496
```

Spring Semester 3 (graduate)

```
s3_g <- s3_r %>% filter(RET_S3 == "Y")
s3_g <- s3_g %>% dplyr::select(-RET_S3)
```

```
sample_train<- sample(seq_len(nrow(s3_g)), size = floor(0.80*nrow(s3_g)))
sample_test <- sample(seq_len(nrow(s3_g)), size = floor(0.20*nrow(s3_g)))

s3_g_train <- s3_g[sample_train, ]
s3_g_test  <- s3_g[sample_test, ]
```

```
graduate_s3 <- glm(graduated ~ ., family = binomial, data = s3_g_train)
predict_s3_g <- predict(graduate_s3, s3_g_test, type = "response")
```

```
prob <- as.factor(ifelse(as.numeric(predict_s3_g > .25)==1, "Y", "N"))
confusionMatrix(data = prob, reference = s3_g_test$graduated, mode = "prec_recall")
```

```
## Confusion Matrix and Statistics
## 
##           Reference
## Prediction   N   Y
##          N  11   5
##          Y 212 500
##                                           
##                Accuracy : 0.7019          
##                  95% CI : (0.6672, 0.735) 
##     No Information Rate : 0.6937          
##     P-Value [Acc > NIR] : 0.3307          
##                                           
##                   Kappa : 0.0532          
##                                           
##  Mcnemar's Test P-Value : <2e-16          
##                                           
##               Precision : 0.68750         
##                  Recall : 0.04933         
##                      F1 : 0.09205         
##              Prevalence : 0.30632         
##          Detection Rate : 0.01511         
##    Detection Prevalence : 0.02198         
##       Balanced Accuracy : 0.51971         
##                                           
##        'Positive' Class : N               
## 
```

```
roc(s3_g_test$graduated, as.numeric(predict_s3_g))
```

```
##
## Call:
## roc.default(response = s3_g_test$graduated, predictor = as.numeric(predict_s3_g))
##
## Data: as.numeric(predict_s3_g) in 223 controls (s3_g_test$graduated N) < 505 cases (s
3_g_test$graduated Y).
## Area under the curve: 0.7189
```

Fall semester 4

Fall Semester 4 (retain)

```
f4_r <- cbind(kent[,1:40], graduated) %>% filter(RET_S3 == "Y")
f4_r <- f4_r %>% dplyr::select(-RET_S3)
```

```
sample_train <- sample(seq_len(nrow(f4_r)), size = floor(0.80*nrow(f4_r)))
sample_test <- sample(seq_len(nrow(f4_r)), size = floor(0.20*nrow(f4_r)))

f4_r_train <- f4_r[sample_train, ]
f4_r_test  <- f4_r[sample_test, ]
```

```
retain_f4 <- glm(RET_F4 ~ ., family = binomial, data = f4_r_train)
predict_f4_r <- predict(retain_f4, f4_r_test, type = "response")
```

```
prob <- as.factor(ifelse(as.numeric(predict_f4_r > .25)==1, "Y", "N"))
confusionMatrix(data = prob, reference = f4_r_test$RET_F4, mode = "prec_recall")
```

```
## Warning in confusionMatrix.default(data = prob, reference =
## f4_r_test$RET_F4, : Levels are not in the same order for reference and
## data. Refactoring data to match.
## Confusion Matrix and Statistics
## 
##           Reference
## Prediction   N   Y
##          N   0   0
##          Y  52 687
##                                          
##                Accuracy : 0.9296         
##                  95% CI : (0.9087, 0.947)
##     No Information Rate : 0.9296         
##     P-Value [Acc > NIR] : 0.5368         
##                                          
##                   Kappa : 0              
##                                          
##  Mcnemar's Test P-Value : 1.522e-12      
##                                          
##               Precision :      NA        
##                  Recall : 0.00000        
##                      F1 :      NA        
##              Prevalence : 0.07037        
##          Detection Rate : 0.00000        
##    Detection Prevalence : 0.00000        
##       Balanced Accuracy : 0.50000        
##                                          
##        'Positive' Class : N              
## 
```

```
roc(f4_r_test$RET_F4, as.numeric(predict_f4_r))
```

```
## 
## Call:
## roc.default(response = f4_r_test$RET_F4, predictor = as.numeric(predict_f4_r))
## 
## Data: as.numeric(predict_f4_r) in 52 controls (f4_r_test$RET_F4 N) < 687 cases (f4_r_test$RET_F4 Y).
## Area under the curve: 0.6967
```

Fall Semester 4 (graduate)

```
f4_g <- f4_r %>% filter(RET_F4 == "Y")
f4_g <- f4_g %>% dplyr::select(-RET_F4)
```

```
sample_train<- sample(seq_len(nrow(f4_g)), size = floor(0.80*nrow(f4_g)))
sample_test <- sample(seq_len(nrow(f4_g)), size = floor(0.20*nrow(f4_g)))

f4_g_train <- f4_g[sample_train, ]
f4_g_test  <- f4_g[sample_test, ]
```

```
graduate_f4 <- glm(graduated ~ ., family = binomial, data = f4_g_train)
predict_f4_g <- predict(graduate_f4, f4_g_test, type = "response")
```

```
prob <- as.factor(ifelse(as.numeric(predict_f4_g > .25)==1, "Y", "N"))
confusionMatrix(data = prob, reference = f4_g_test$graduated, mode = "prec_recall")
```

```
## Confusion Matrix and Statistics
## 
##           Reference
## Prediction   N   Y
##          N   8   0
##          Y 189 488
##                                          
##                Accuracy : 0.7241         
##                  95% CI : (0.689, 0.7573)
##     No Information Rate : 0.7124         
##     P-Value [Acc > NIR] : 0.2645         
##                                          
##                   Kappa : 0.0569         
##                                          
##  Mcnemar's Test P-Value : <2e-16         
##                                          
##               Precision : 1.00000        
##                  Recall : 0.04061        
##                      F1 : 0.07805        
##              Prevalence : 0.28759        
##          Detection Rate : 0.01168        
##    Detection Prevalence : 0.01168        
##       Balanced Accuracy : 0.52030        
##                                          
##        'Positive' Class : N              
## 
```

```
roc(f4_g_test$graduated, as.numeric(predict_f4_g))
```

```
## 
## Call:
## roc.default(response = f4_g_test$graduated, predictor = as.numeric(predict_f4_g))
## 
## Data: as.numeric(predict_f4_g) in 197 controls (f4_g_test$graduated N) < 488 cases (f
4_g_test$graduated Y).
## Area under the curve: 0.6697
```

Spring semester 4

Spring Semester 4(retain)

```
s4_r <- cbind(kent[,1:44], graduated) %>% filter(RET_F4 == "Y")
s4_r <- s4_r %>% dplyr::select(-RET_F4)
```

```
sample_train<- sample(seq_len(nrow(s4_r)), size = floor(0.80*nrow(s4_r)))
sample_test <- sample(seq_len(nrow(s4_r)), size = floor(0.20*nrow(s4_r)))

s4_r_train <- s4_r[sample_train, ]
s4_r_test  <- s4_r[sample_test, ]
```

```
retain_s4 <- glm(RET_S4 ~ ., family = binomial, data = s4_r_train)
predict_s4_r <- predict(retain_s4, s4_r_test, type = "response")
```

```
prob <- as.factor(ifelse(as.numeric(predict_s4_r > .25)==1, "Y", "N"))
confusionMatrix(data = prob, reference = s4_r_test$RET_S4, mode = "prec_recall")
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction   N   Y
##          N 124   2
##          Y  34 546
##
##                Accuracy : 0.949
##                  95% CI : (0.9301, 0.964)
##     No Information Rate : 0.7762
##     P-Value [Acc > NIR] : < 2.2e-16
##
##                   Kappa : 0.8418
##
##  Mcnemar's Test P-Value : 2.383e-07
##
##               Precision : 0.9841
##                  Recall : 0.7848
##                      F1 : 0.8732
##              Prevalence : 0.2238
##          Detection Rate : 0.1756
##    Detection Prevalence : 0.1785
##       Balanced Accuracy : 0.8906
##
##        'Positive' Class : N
##
```

```
roc(s4_r_test$RET_S4, as.numeric(predict_s4_r))
```

```
##
## Call:
## roc.default(response = s4_r_test$RET_S4, predictor = as.numeric(predict_s4_r))
##
## Data: as.numeric(predict_s4_r) in 158 controls (s4_r_test$RET_S4 N) < 548 cases (s4_r
_test$RET_S4 Y).
## Area under the curve: 0.9419
```

Spring Semester 4 (graduate)

```
s4_g <- s4_r %>% filter(RET_S4 == "Y")
s4_g <- s4_g %>% dplyr::select(-RET_S4)
```

```
sample_train<- sample(seq_len(nrow(s4_g)), size = floor(0.80*nrow(s4_g)))
sample_test <- sample(seq_len(nrow(s4_g)), size = floor(0.20*nrow(s4_g)))

s4_g_train <- s4_g[sample_train, ]
s4_g_test  <- s4_g[sample_test, ]
```

```
graduate_s4 <- glm(graduated ~ ., family = binomial, data = s4_g_train)
predict_s4_g <- predict(graduate_s4, s4_g_test, type = "response")
```

```
prob <- as.factor(ifelse(as.numeric(predict_s4_g > .25)==1, "Y", "N"))
confusionMatrix(data = prob, reference = s4_g_test$graduated, mode = "prec_recall")
```

```
## Confusion Matrix and Statistics
## 
##           Reference
## Prediction   N   Y
##          N  16   4
##          Y  82 458
## 
##                Accuracy : 0.8464
##                  95% CI : (0.8139, 0.8753)
##     No Information Rate : 0.825
##     P-Value [Acc > NIR] : 0.09899
## 
##                   Kappa : 0.2252
## 
##  Mcnemar's Test P-Value : < 2e-16
## 
##               Precision : 0.80000
##                  Recall : 0.16327
##                      F1 : 0.27119
##              Prevalence : 0.17500
##          Detection Rate : 0.02857
##    Detection Prevalence : 0.03571
##       Balanced Accuracy : 0.57730
## 
##        'Positive' Class : N
## 
```

```
roc(s4_g_test$graduated, as.numeric(predict_s4_g))
```

```
## 
## Call:
## roc.default(response = s4_g_test$graduated, predictor = as.numeric(predict_s4_g))
## 
## Data: as.numeric(predict_s4_g) in 98 controls (s4_g_test$graduated N) < 462 cases (s4
_g_test$graduated Y).
## Area under the curve: 0.8347
```

Fall semester 5

Fall Semester 5 (retain)

```
f5_r <- cbind(kent[,1:48], graduated) %>% filter(RET_S4 == "Y")
f5_r <- f5_r %>% dplyr::select(-RET_S4)
```

```
sample_train <- sample(seq_len(nrow(f5_r)), size = floor(0.80*nrow(f5_r)))
sample_test <- sample(seq_len(nrow(f5_r)), size = floor(0.20*nrow(f5_r)))

f5_r_train <- f5_r[sample_train, ]
f5_r_test  <- f5_r[sample_test, ]
```

```
retain_f5 <- glm(RET_F5 ~ ., family = binomial, data = f5_r_train)
predict_f5_r <- predict(retain_f5, f5_r_test, type = "response")
```

```
prob <- as.factor(ifelse(as.numeric(predict_f5_r > .25)==1, "Y", "N"))
confusionMatrix(data = prob, reference = f5_r_test$RET_F5, mode = "prec_recall")
```

```
## Confusion Matrix and Statistics
## 
##           Reference
## Prediction   N   Y
##          N  77  10
##          Y 209 269
##                                           
##                Accuracy : 0.6124          
##                  95% CI : (0.5708, 0.6528)
##     No Information Rate : 0.5062          
##     P-Value [Acc > NIR] : 2.464e-07       
##                                           
##                   Kappa : 0.2314          
##                                           
##  Mcnemar's Test P-Value : < 2.2e-16       
##                                           
##               Precision : 0.8851          
##                  Recall : 0.2692          
##                      F1 : 0.4129          
##              Prevalence : 0.5062          
##          Detection Rate : 0.1363          
##    Detection Prevalence : 0.1540          
##       Balanced Accuracy : 0.6167          
##                                           
##        'Positive' Class : N               
## 
```

```
roc(f5_r_test$RET_F5, as.numeric(predict_f5_r))
```

```
## 
## Call:
## roc.default(response = f5_r_test$RET_F5, predictor = as.numeric(predict_f5_r))
## 
## Data: as.numeric(predict_f5_r) in 286 controls (f5_r_test$RET_F5 N) < 279 cases (f5_r
_test$RET_F5 Y).
## Area under the curve: 0.8186
```

Fall Semester 5 (graduate)

```
f5_g <- f5_r %>% filter(RET_F5 == "Y")
f5_g <- f5_g %>% dplyr::select(-RET_F5)
```

```
sample_train<- sample(seq_len(nrow(f5_g)), size = floor(0.80*nrow(f5_g)))
sample_test <- sample(seq_len(nrow(f5_g)), size = floor(0.20*nrow(f5_g)))

f5_g_train <- f5_g[sample_train, ]
f5_g_test  <- f5_g[sample_test, ]
```

```
graduate_f5 <- glm(graduated ~ ., family = binomial, data = f5_g_train)
predict_f5_g <- predict(graduate_f5, f5_g_test, type = "response")
```

```
prob <- as.factor(ifelse(as.numeric(predict_f5_g > .25)==1, "Y", "N"))
confusionMatrix(data = prob, reference = f5_g_test$graduated, mode = "prec_recall")
```

```
## Confusion Matrix and Statistics
## 
##           Reference
## Prediction   N   Y
##          N  10   1
##          Y  62 216
##                                           
##                Accuracy : 0.782           
##                  95% CI : (0.7299, 0.8282)
##     No Information Rate : 0.7509          
##     P-Value [Acc > NIR] : 0.123           
##                                           
##                   Kappa : 0.1873          
##                                           
##  Mcnemar's Test P-Value : 4.053e-14       
##                                           
##               Precision : 0.90909         
##                  Recall : 0.13889         
##                      F1 : 0.24096         
##              Prevalence : 0.24913         
##          Detection Rate : 0.03460         
##    Detection Prevalence : 0.03806         
##       Balanced Accuracy : 0.56714         
##                                           
##        'Positive' Class : N               
## 
```

```
roc(f5_g_test$graduated, as.numeric(predict_f5_g))
```

```
##
## Call:
## roc.default(response = f5_g_test$graduated, predictor = as.numeric(predict_f5_g))
##
## Data: as.numeric(predict_f5_g) in 72 controls (f5_g_test$graduated N) < 217 cases (f5
_g_test$graduated Y).
## Area under the curve: 0.7982
```

Spring semester 5

Spring Semester 5 (retain)

```
s5_r <- cbind(kent[,1:52], graduated) %>% filter(RET_F5 == "Y")
s5_r <- s5_r %>% dplyr::select(-RET_F5)
```

```
sample_train<- sample(seq_len(nrow(s5_r)), size = floor(0.80*nrow(s5_r)))
sample_test <- sample(seq_len(nrow(s5_r)), size = floor(0.20*nrow(s5_r)))

s5_r_train <- s5_r[sample_train, ]
s5_r_test  <- s5_r[sample_test, ]
```

```
retain_s5 <- glm(RET_S5 ~ ., family = binomial, data = s5_r_train)
predict_s5_r <- predict(retain_s5, s5_r_test, type = "response")
```

```
prob <- as.factor(ifelse(as.numeric(predict_s5_r > .25)==1, "Y", "N"))
confusionMatrix(data = prob, reference = s5_r_test$RET_S5, mode = "prec_recall")
```

```
## Confusion Matrix and Statistics
## 
##           Reference
## Prediction   N   Y
##          N  40   0
##          Y  90 170
##                                           
##                Accuracy : 0.7             
##                  95% CI : (0.6447, 0.7513)
##     No Information Rate : 0.5667          
##     P-Value [Acc > NIR] : 1.401e-06       
##                                           
##                   Kappa : 0.335           
##                                           
##  Mcnemar's Test P-Value : < 2.2e-16       
##                                           
##               Precision : 1.0000          
##                  Recall : 0.3077          
##                      F1 : 0.4706          
##              Prevalence : 0.4333          
##          Detection Rate : 0.1333          
##    Detection Prevalence : 0.1333          
##       Balanced Accuracy : 0.6538          
##                                           
##        'Positive' Class : N               
## 
```

```
roc(s5_r_test$RET_S5, as.numeric(predict_s5_r))
```

```
## 
## Call:
## roc.default(response = s5_r_test$RET_S5, predictor = as.numeric(predict_s5_r))
## 
## Data: as.numeric(predict_s5_r) in 130 controls (s5_r_test$RET_S5 N) < 170 cases (s5_r_test$RET_S5 Y).
## Area under the curve: 0.7813
```

Spring Semester 5 (graduate)

```
s5_g <- s5_r %>% filter(RET_S5 == "Y")
s5_g <- s5_g %>% dplyr::select(-RET_S5)
```

```
sample_train<- sample(seq_len(nrow(s5_g)), size = floor(0.80*nrow(s5_g)))
sample_test <- sample(seq_len(nrow(s5_g)), size = floor(0.20*nrow(s5_g)))

s5_g_train <- s5_g[sample_train, ]
s5_g_test  <- s5_g[sample_test, ]
```

```
graduate_s5 <- glm(graduated ~ ., family = binomial, data = s5_g_train)
predict_s5_g <- predict(graduate_s5, s5_g_test, type = "response")
```

```
prob <- as.factor(ifelse(as.numeric(predict_s5_g > .25)==1, "Y", "N"))
confusionMatrix(data = prob, reference = s5_g_test$graduated, mode = "prec_recall")
```

```
## Confusion Matrix and Statistics
## 
##           Reference
## Prediction   N   Y
##          N   5   4
##          Y  20 141
## 
##                Accuracy : 0.8588
##                  95% CI : (0.7973, 0.9074)
##     No Information Rate : 0.8529
##     P-Value [Acc > NIR] : 0.4670
## 
##                   Kappa : 0.2345
## 
##  Mcnemar's Test P-Value : 0.0022
## 
##               Precision : 0.55556
##                  Recall : 0.20000
##                      F1 : 0.29412
##              Prevalence : 0.14706
##          Detection Rate : 0.02941
##    Detection Prevalence : 0.05294
##       Balanced Accuracy : 0.58621
## 
##        'Positive' Class : N
## 
```

```
roc(s5_g_test$graduated, as.numeric(predict_s5_g))
```

```
## 
## Call:
## roc.default(response = s5_g_test$graduated, predictor = as.numeric(predict_s5_g))
## 
## Data: as.numeric(predict_s5_g) in 25 controls (s5_g_test$graduated N) < 145 cases (s5
_g_test$graduated Y).
## Area under the curve: 0.8284
```

Fall semester 6

Fall Semester 6 (retain)

```
f6_r <- cbind(kent[,1:56], graduated) %>% filter(RET_S5 == "Y")
f6_r <- f6_r %>% dplyr::select(-RET_S5)
```

```
sample_train <- sample(seq_len(nrow(f6_r)), size = floor(0.80*nrow(f6_r)))
sample_test <- sample(seq_len(nrow(f6_r)), size = floor(0.20*nrow(f6_r)))

f6_r_train <- f6_r[sample_train, ]
f6_r_test  <- f6_r[sample_test, ]
```

```
retain_f6 <- glm(RET_F6 ~ ., family = binomial, data = f6_r_train)
predict_f6_r <- predict(retain_f6, f6_r_test, type = "response")
```

```
prob <- as.factor(ifelse(as.numeric(predict_f6_r > .25)==1, "Y", "N"))
confusionMatrix(data = prob, reference = f6_r_test$RET_F6, mode = "prec_recall")
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction  N  Y
##          N 55  9
##          Y 57 56
##
##                Accuracy : 0.6271
##                  95% CI : (0.5514, 0.6985)
##     No Information Rate : 0.6328
##     P-Value [Acc > NIR] : 0.595
##
##                   Kappa : 0.3053
##
##  Mcnemar's Test P-Value : 7.238e-09
##
##               Precision : 0.8594
##                  Recall : 0.4911
##                      F1 : 0.6250
##              Prevalence : 0.6328
##          Detection Rate : 0.3107
##    Detection Prevalence : 0.3616
##       Balanced Accuracy : 0.6763
##
##        'Positive' Class : N
##
```

```
roc(f6_r_test$RET_F6, as.numeric(predict_f6_r))
```

```
##
## Call:
## roc.default(response = f6_r_test$RET_F6, predictor = as.numeric(predict_f6_r))
##
## Data: as.numeric(predict_f6_r) in 112 controls (f6_r_test$RET_F6 N) < 65 cases (f6_r_
test$RET_F6 Y).
## Area under the curve: 0.8058
```

Fall Semester 6 (graduate)

```
f6_g <- f6_r %>% filter(RET_F6 == "Y")
f6_g <- f6_g %>% dplyr::select(-RET_F6)
```

```
sample_train<- sample(seq_len(nrow(f6_g)), size = floor(0.80*nrow(f6_g)))
sample_test <- sample(seq_len(nrow(f6_g)), size = floor(0.20*nrow(f6_g)))

f6_g_train <- f6_g[sample_train, ]
f6_g_test  <- f6_g[sample_test, ]
```

```
graduate_f6 <- glm(graduated ~ ., family = binomial, data = f6_g_train)
predict_f6_g <- predict(graduate_f6, f6_g_test, type = "response")
```

```
prob <- as.factor(ifelse(as.numeric(predict_f6_g > .25)==1, "Y", "N"))
confusionMatrix(data = prob, reference = f6_g_test$graduated, mode = "prec_recall")
```

```
## Confusion Matrix and Statistics
## 
##           Reference
## Prediction  N  Y
##          N  8  1
##          Y 14 45
##                                           
##                Accuracy : 0.7794          
##                  95% CI : (0.6624, 0.871) 
##     No Information Rate : 0.6765          
##     P-Value [Acc > NIR] : 0.042702        
##                                           
##                   Kappa : 0.4042          
##                                           
##  Mcnemar's Test P-Value : 0.001946        
##                                           
##               Precision : 0.8889          
##                  Recall : 0.3636          
##                      F1 : 0.5161          
##              Prevalence : 0.3235          
##          Detection Rate : 0.1176          
##    Detection Prevalence : 0.1324          
##       Balanced Accuracy : 0.6709          
##                                           
##        'Positive' Class : N               
## 
```

```
roc(f6_g_test$graduated, as.numeric(predict_f6_g))
```

```
## 
## Call:
## roc.default(response = f6_g_test$graduated, predictor = as.numeric(predict_f6_g))
## 
## Data: as.numeric(predict_f6_g) in 22 controls (f6_g_test$graduated N) < 46 cases (f6_
g_test$graduated Y).
## Area under the curve: 0.8251
```

Spring semester 6

Spring Semester 6 (retain)

```
s6_r <- cbind(kent[,1:60], graduated) %>% filter(RET_F6 == "Y")
s6_r <- s6_r %>% dplyr::select(-RET_F6)
```

```
sample_train<- sample(seq_len(nrow(s6_r)), size = floor(0.80*nrow(s6_r)))
sample_test <- sample(seq_len(nrow(s6_r)), size = floor(0.20*nrow(s6_r)))

s6_r_train <- s6_r[sample_train, ]
s6_r_test  <- s6_r[sample_test, ]
```

```
retain_s6 <- glm(RET_S6 ~ ., family = binomial, data = s6_r_train)
predict_s6_r <- predict(retain_s6, s6_r_test, type = "response")
```

```
prob <- as.factor(ifelse(as.numeric(predict_s6_r > .25)==1, "Y", "N"))
confusionMatrix(data = prob, reference = s6_r_test$RET_S6, mode = "prec_recall")
```

```
## Confusion Matrix and Statistics
## 
##           Reference
## Prediction  N  Y
##          N 23  3
##          Y 17 34
## 
##                Accuracy : 0.7403
##                  95% CI : (0.6277, 0.8336)
##     No Information Rate : 0.5195
##     P-Value [Acc > NIR] : 6.073e-05
## 
##                   Kappa : 0.487
## 
##  Mcnemar's Test P-Value : 0.00365
## 
##               Precision : 0.8846
##                  Recall : 0.5750
##                      F1 : 0.6970
##              Prevalence : 0.5195
##          Detection Rate : 0.2987
##    Detection Prevalence : 0.3377
##       Balanced Accuracy : 0.7470
## 
##        'Positive' Class : N
## 
```

```
roc(s6_r_test$RET_S6, as.numeric(predict_s6_r))
```

```
## 
## Call:
## roc.default(response = s6_r_test$RET_S6, predictor = as.numeric(predict_s6_r))
## 
## Data: as.numeric(predict_s6_r) in 40 controls (s6_r_test$RET_S6 N) < 37 cases (s6_r_t
est$RET_S6 Y).
## Area under the curve: 0.8736
```

Spring Semester 6 (graduate)

```
s6_add <- kent %>% filter(RET_S1 == "Y") %>% filter(RET_F2 == "Y")  %>% filter(RET_S2 ==
"Y") %>% filter(RET_F3 == "Y") %>% filter(RET_S3 == "Y") %>% filter(RET_F4 == "Y") %>% f
ilter(RET_S4 == "Y") %>% filter(RET_F5 == "Y") %>% filter(RET_S5 == "Y") %>% filter(RET_
F6 == "Y") %>% filter(RET_S6 == "Y") %>% dplyr::select(c(S6SEQ2_CURATTHRS, S6SEQ2_CURERN
HRS, S6SEQ2_TERM_GPA))
nrow(s6_add)
```

```
## [1] 119
```

```
s6_g <- s6_r %>% filter(RET_S6 == "Y")
empty <- matrix(c(rep.int(NA,length(s6_add))), nrow = 67, ncol = length(s6_add))
colnames(empty) <- colnames(s6_add)
s6_add <- rbind(s6_add, empty)
s6_g <- cbind(s6_g, s6_add)
s6_g <- s6_g %>% dplyr::select(-c(URS_IND,ONCAMPUS_IND,FIRST_GEN_IND, PELL_ELIG_IND, INT
ERNATIONAL_IND, ATHLETE_IND, VETERAN_IND, HONORS_REGISTERED_IND, RET_S1, RET_F2, RET_S2,
RET_F3, RET_S3, RET_F4, RET_S4, RET_F5, RET_S5, RET_S6))
```

```
sample_train<- sample(seq_len(nrow(s6_g)), size = floor(0.80*nrow(s6_g)))
sample_test <- sample(seq_len(nrow(s6_g)), size = floor(0.20*nrow(s6_g)))

s6_g_train <- s6_g[sample_train, ] %>% as.data.frame()
s6_g_test  <- s6_g[sample_test, ] %>% as.data.frame()
```

```
graduate_s6 <- glm(graduated ~ ., family = "binomial", data = s6_g_train)
```

```
## Warning: glm.fit: algorithm did not converge
```

```
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
```

```
predict_s6_g <- predict(graduate_s6, s6_g_test, type = "response")
```

```
prob <- as.factor(ifelse(as.numeric(predict_s6_g > .25)==1, "Y", "N"))
confusionMatrix(data = prob, reference = s6_g_test$graduated, mode = "prec_recall")
```

```
## Confusion Matrix and Statistics
## 
##           Reference
## Prediction  N  Y
##          N  4  2
##          Y  0 17
##                                           
##                Accuracy : 0.913           
##                  95% CI : (0.7196, 0.9893)
##     No Information Rate : 0.8261          
##     P-Value [Acc > NIR] : 0.2106          
##                                           
##                   Kappa : 0.7473          
##                                           
##  Mcnemar's Test P-Value : 0.4795          
##                                           
##               Precision : 0.6667          
##                  Recall : 1.0000          
##                      F1 : 0.8000          
##              Prevalence : 0.1739          
##          Detection Rate : 0.1739          
##    Detection Prevalence : 0.2609          
##       Balanced Accuracy : 0.9474          
##                                           
##        'Positive' Class : N               
## 
```

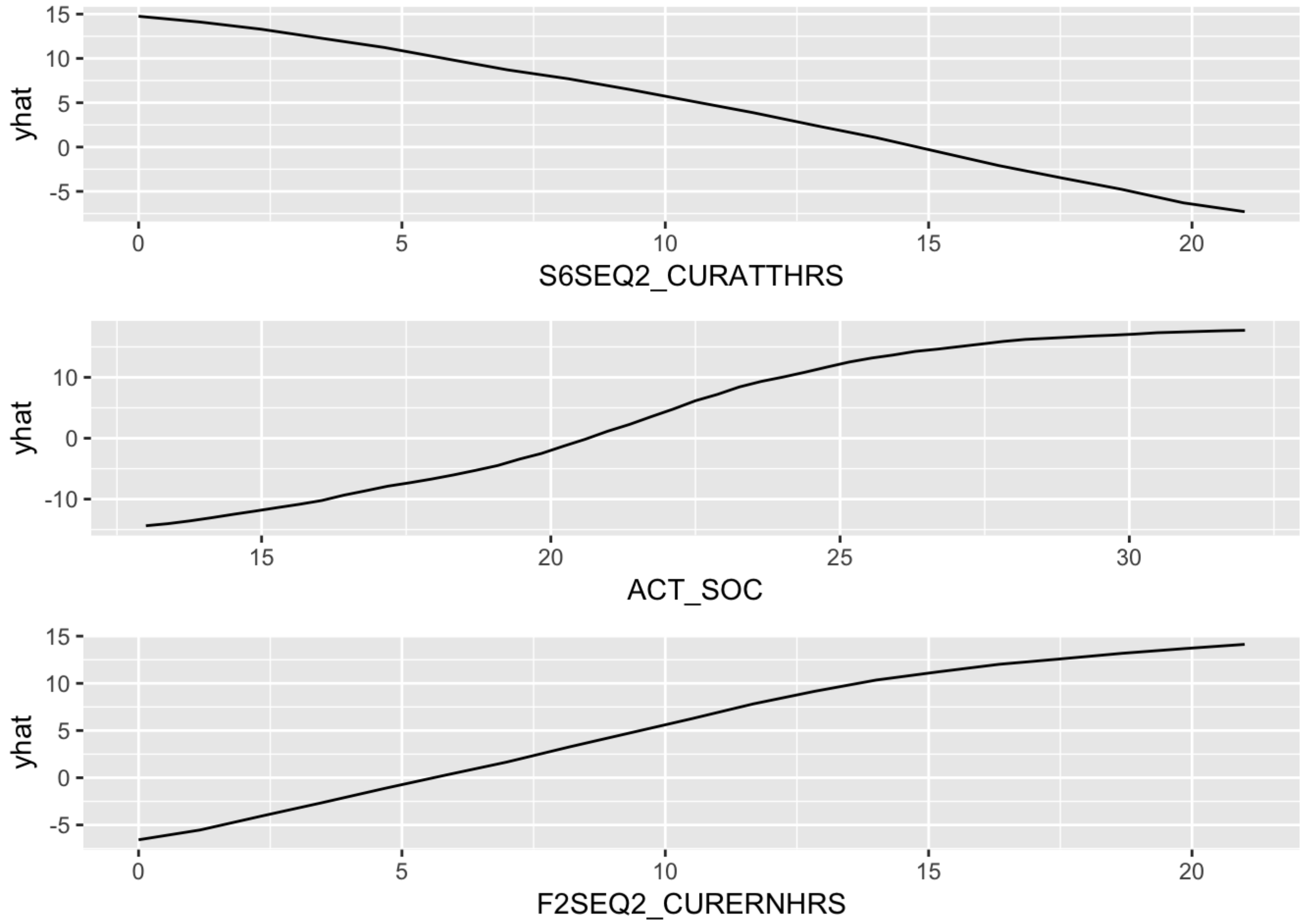
```
roc(s6_g_test$graduated, as.numeric(predict_s6_g))
```

```
## 
## Call:
## roc.default(response = s6_g_test$graduated, predictor = as.numeric(predict_s6_g))
## 
## Data: as.numeric(predict_s6_g) in 4 controls (s6_g_test$graduated N) < 19 cases (s6_g
_test$graduated Y).
## Area under the curve: 0.9737
```

Partial Dependency plot for Spring semester 6 (graduate) - top 3 variables

```
s6_g_imp <- varImp(graduate_s6, scale = FALSE)
```

```
par_s6_g_atthr <- partial(graduate_s6, pred.var = c("S6SEQ2_CURATTHRS"), chull = TRUE)
plot_s6_g_atthr  <- autoplot(par_s6_g_atthr , contour = TRUE)
par_s6_g_acts  <- partial(graduate_s6, pred.var = c("ACT_SOC"), chull = TRUE)
plot_s6_g_acts  <- autoplot(par_s6_g_acts, contour = TRUE)
par_s6_g_erhr  <- partial(graduate_s6, pred.var = c("F2SEQ2_CURERNHRS"), chull = TRUE)
plot_s6_g_erhr  <- autoplot(par_s6_g_erhr, contour = TRUE)
grid.arrange(plot_s6_g_atthr, plot_s6_g_acts, plot_s6_g_erhr)
```


yhat
S6SEQ2_CURATTHRS
yhat
ACT_SOC
yhat
F2SEQ2_CURERNHRS