title: “05-exercises” author: “Renaldo Williams” date: “2016-05-xx” output: html_document —

Reading:

- **APM** Chapter 8.1-8.5 “Regression Trees and Rule-Based Models” (25 pages)
- **APM** Chapter 14.1-14.5 “Classification Trees and Rule-Based”

Exercise 1: GermanCredit

Revisit the GermanCredit data. Use `caret` to build models of `Class` using the following techniques:

- glm
- rpart
- knn
- party::ctree
- randomForest
- A method of your choice from the Caret Model List (you will need to install any dependencies)

Save the caret objects with the names provided.

```
data("GermanCredit")

# Your work here.

# Using caret and train() for automatic parameter tuning. Train serves as standardized interface for ov
ctrl <- trainControl( method="cv", number=10, classProb=TRUE, savePrediction=TRUE)

fit.glm <- train(Class ~ ., data = GermanCredit, method="glm", family = "binomial", trControl=ctrl)

## KNN

#trainControl() is to set custom controls, specifically bootstrap
ctrl <- trainControl(method="cv", number=10, classProb=TRUE, savePrediction=TRUE)

fit.knn <- train(Class ~ ., data = GermanCredit,trControl=ctrl, method="knn",tuneGrid=data.frame(k=c(1,5

## RPART

#trainControl() is to set custom controls, specifically bootstrap
ctrl <- trainControl( method="cv", number=10, classProb=TRUE, savePrediction=TRUE )

fit.rpart <- train(Class ~ ., data = GermanCredit,trControl=ctrl, method="rpart", cp=0.02, tuneLength=20

## RF

#trainControl() is to set custom controls, specifically bootstrap
ctrl <- trainControl( method="cv", number=10, classProb=TRUE, savePrediction=TRUE )

fit.rf <- train(Class ~ ., data = GermanCredit,trControl=ctrl, method="rf")
```

- Compare the models using `caret::confusionMatrix`

- Comparing the models Using the `pROC` packages
- create ROC curves for the models

Show your work!

GLM Scores

```
## GLM
confusionMatrix(fit.glm$pred$pred,fit.glm$pred$obs, positive="Bad")
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction Bad Good
##       Bad  147  102
##       Good 153  598
##
##                Accuracy : 0.745
##                  95% CI : (0.7168, 0.7718)
##     No Information Rate : 0.7
##     P-Value [Acc > NIR] : 0.0009244
##
##                   Kappa : 0.3619
##  Mcnemar's Test P-Value : 0.0017414
##
##             Sensitivity : 0.4900
##             Specificity : 0.8543
##          Pos Pred Value : 0.5904
##          Neg Pred Value : 0.7963
##              Prevalence : 0.3000
##          Detection Rate : 0.1470
##    Detection Prevalence : 0.2490
##       Balanced Accuracy : 0.6721
##
##        'Positive' Class : Bad
##
```

```
summary(fit.glm)
```

```
##
## Call:
## NULL
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.6116  -0.7095   0.3752   0.6994   2.3410
##
## Coefficients: (13 not defined because of singularities)
##                                       Estimate Std. Error z value
## (Intercept)                          8.341e+00  1.409e+00   5.918
## Duration                            -2.786e-02  9.296e-03  -2.997
```

```
## Amount                                -1.283e-04  4.444e-05  -2.887
## InstallmentRatePercentage             -3.301e-01  8.828e-02  -3.739
## ResidenceDuration                     -4.776e-03  8.641e-02  -0.055
## Age                                    1.454e-02  9.222e-03   1.576
## NumberExistingCredits                 -2.721e-01  1.895e-01  -1.436
## NumberPeopleMaintenance               -2.647e-01  2.492e-01  -1.062
## Telephone                             -3.000e-01  2.013e-01  -1.491
## ForeignWorker                         -1.392e+00  6.258e-01  -2.225
## CheckingAccountStatus.lt.0            -1.712e+00  2.322e-01  -7.373
## CheckingAccountStatus.0.to.200        -1.337e+00  2.325e-01  -5.752
## CheckingAccountStatus.gt.200          -7.462e-01  3.831e-01  -1.948
## CheckingAccountStatus.none                    NA         NA      NA
## CreditHistory.NoCredit.AllPaid        -1.436e+00  4.399e-01  -3.264
## CreditHistory.ThisBank.AllPaid        -1.579e+00  4.381e-01  -3.605
## CreditHistory.PaidDuly                -8.497e-01  2.587e-01  -3.284
## CreditHistory.Delay                   -5.826e-01  3.345e-01  -1.742
## CreditHistory.Critical                        NA         NA      NA
## Purpose.NewCar                        -1.489e+00  7.764e-01  -1.918
## Purpose.UsedCar                        1.777e-01  8.081e-01   0.220
## Purpose.Furniture.Equipment           -6.972e-01  7.844e-01  -0.889
## Purpose.Radio.Television              -5.972e-01  7.841e-01  -0.762
## Purpose.DomesticAppliance             -9.660e-01  1.077e+00  -0.897
## Purpose.Repairs                       -1.272e+00  9.264e-01  -1.373
## Purpose.Education                     -1.525e+00  8.453e-01  -1.804
## Purpose.Vacation                              NA         NA      NA
## Purpose.Retraining                     5.706e-01  1.431e+00   0.399
## Purpose.Business                      -7.487e-01  7.998e-01  -0.936
## Purpose.Other                                 NA         NA      NA
## SavingsAccountBonds.lt.100            -9.467e-01  2.625e-01  -3.607
## SavingsAccountBonds.100.to.500        -5.889e-01  3.493e-01  -1.686
## SavingsAccountBonds.500.to.1000       -5.706e-01  4.492e-01  -1.270
## SavingsAccountBonds.gt.1000            3.925e-01  5.644e-01   0.695
## SavingsAccountBonds.Unknown                   NA         NA      NA
## EmploymentDuration.lt.1                6.691e-02  4.270e-01   0.157
## EmploymentDuration.1.to.4              1.828e-01  4.105e-01   0.445
## EmploymentDuration.4.to.7              8.310e-01  4.455e-01   1.866
## EmploymentDuration.gt.7                2.766e-01  4.134e-01   0.669
## EmploymentDuration.Unemployed                 NA         NA      NA
## Personal.Male.Divorced.Seperated      -3.671e-01  4.537e-01  -0.809
## Personal.Female.NotSingle             -9.162e-02  3.118e-01  -0.294
## Personal.Male.Single                   4.490e-01  3.152e-01   1.424
## Personal.Male.Married.Widowed                 NA         NA      NA
## Personal.Female.Single                        NA         NA      NA
## OtherDebtorsGuarantors.None           -9.786e-01  4.243e-01  -2.307
## OtherDebtorsGuarantors.CoApplicant    -1.415e+00  5.685e-01  -2.488
## OtherDebtorsGuarantors.Guarantor              NA         NA      NA
## Property.RealEstate                    7.304e-01  4.245e-01   1.721
## Property.Insurance                     4.490e-01  4.130e-01   1.087
## Property.CarOther                      5.359e-01  4.017e-01   1.334
## Property.Unknown                              NA         NA      NA
## OtherInstallmentPlans.Bank            -6.463e-01  2.391e-01  -2.703
## OtherInstallmentPlans.Stores          -5.231e-01  3.754e-01  -1.393
## OtherInstallmentPlans.None                    NA         NA      NA
## Housing.Rent                          -6.839e-01  4.770e-01  -1.434
```

```
## Housing.Own                              -2.402e-01  4.503e-01  -0.534
## Housing.ForFree                                 NA         NA      NA
## Job.UnemployedUnskilled                  4.795e-01  6.623e-01   0.724
## Job.UnskilledResident                   -5.666e-02  3.501e-01  -0.162
## Job.SkilledEmployee                     -7.524e-02  2.845e-01  -0.264
## Job.Management.SelfEmp.HighlyQualified          NA         NA      NA
##                                         Pr(>|z|)
## (Intercept)                             3.25e-09 ***
## Duration                                0.002724 **
## Amount                                  0.003894 **
## InstallmentRatePercentage               0.000185 ***
## ResidenceDuration                       0.955920
## Age                                     0.114982
## NumberExistingCredits                   0.151109
## NumberPeopleMaintenance                 0.288249
## Telephone                               0.136060
## ForeignWorker                           0.026095 *
## CheckingAccountStatus.lt.0              1.66e-13 ***
## CheckingAccountStatus.0.to.200          8.83e-09 ***
## CheckingAccountStatus.gt.200            0.051419 .
## CheckingAccountStatus.none                    NA
## CreditHistory.NoCredit.AllPaid          0.001099 **
## CreditHistory.ThisBank.AllPaid          0.000312 ***
## CreditHistory.PaidDuly                  0.001022 **
## CreditHistory.Delay                     0.081540 .
## CreditHistory.Critical                        NA
## Purpose.NewCar                          0.055163 .
## Purpose.UsedCar                         0.825966
## Purpose.Furniture.Equipment             0.374123
## Purpose.Radio.Television                0.446249
## Purpose.DomesticAppliance               0.369646
## Purpose.Repairs                         0.169598
## Purpose.Education                       0.071192 .
## Purpose.Vacation                              NA
## Purpose.Retraining                      0.690107
## Purpose.Business                        0.349202
## Purpose.Other                                 NA
## SavingsAccountBonds.lt.100              0.000310 ***
## SavingsAccountBonds.100.to.500          0.091805 .
## SavingsAccountBonds.500.to.1000         0.203940
## SavingsAccountBonds.gt.1000             0.486765
## SavingsAccountBonds.Unknown                   NA
## EmploymentDuration.lt.1                 0.875475
## EmploymentDuration.1.to.4               0.656049
## EmploymentDuration.4.to.7               0.062110 .
## EmploymentDuration.gt.7                 0.503410
## EmploymentDuration.Unemployed                 NA
## Personal.Male.Divorced.Seperated        0.418448
## Personal.Female.NotSingle               0.768908
## Personal.Male.Single                    0.154345
## Personal.Male.Married.Widowed                 NA
## Personal.Female.Single                        NA
## OtherDebtorsGuarantors.None             0.021072 *
## OtherDebtorsGuarantors.CoApplicant      0.012834 *
```

```
## OtherDebtorsGuarantors.Guarantor             NA
## Property.RealEstate                     0.085308 .
## Property.Insurance                      0.277005
## Property.CarOther                       0.182211
## Property.Unknown                             NA
## OtherInstallmentPlans.Bank              0.006871 **
## OtherInstallmentPlans.Stores            0.163501
## OtherInstallmentPlans.None                   NA
## Housing.Rent                            0.151657
## Housing.Own                             0.593687
## Housing.ForFree                              NA
## Job.UnemployedUnskilled                 0.469086
## Job.UnskilledResident                   0.871450
## Job.SkilledEmployee                     0.791419
## Job.Management.SelfEmp.HighlyQualified       NA
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 1221.73  on 999  degrees of freedom
## Residual deviance:  895.82  on 951  degrees of freedom
## AIC: 993.82
##
## Number of Fisher Scoring iterations: 5
```

```
fit.glm.sensitivity <- sensitivity(fit.glm$pred$pred, fit.glm$pred$obs, positive="Bad")

## Sensitivity at ~49%
fit.glm.sensitivity
```

```
## [1] 0.49
```

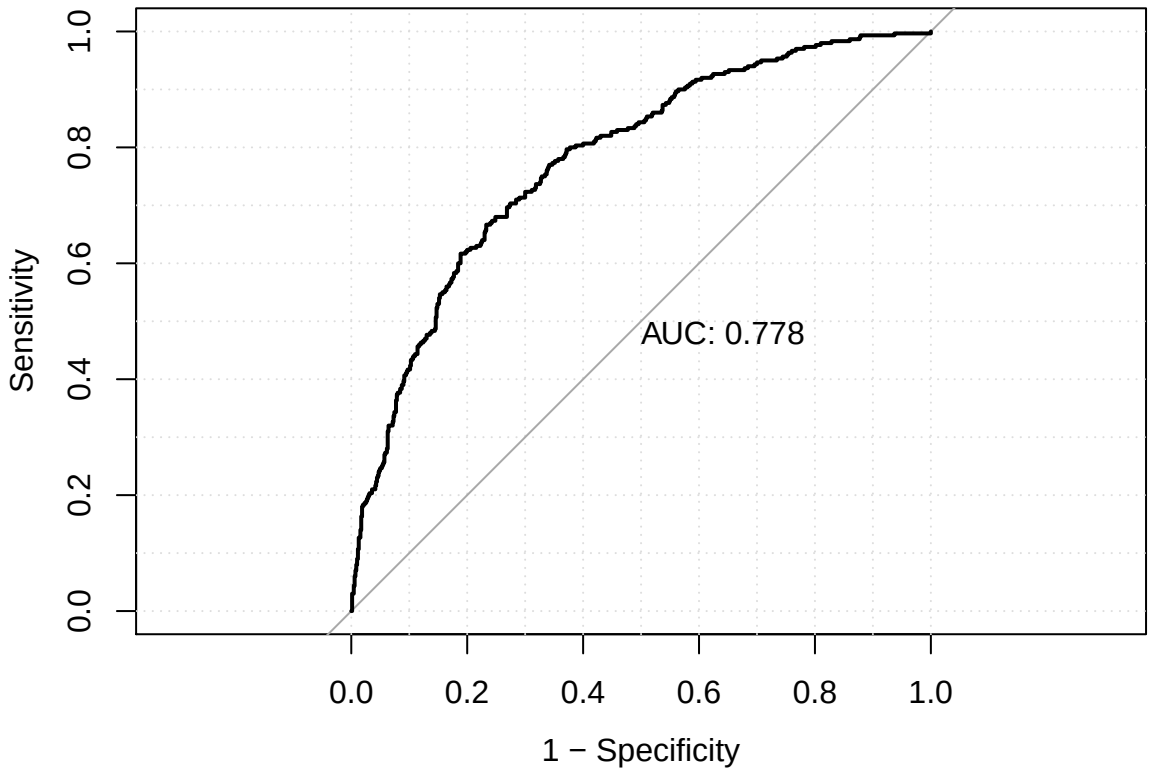
```
fit.glm.specificity <- specificity(fit.glm$pred$pred, fit.glm$pred$obs, positive="Bad")

## Specificity at ~89%
fit.glm.specificity
```

```
## [1] 0.8542857
```

```
## ROC Curve
## This function assumes that the second class is the event of interest, so we reverse the labels.
roc <- roc(fit.glm$pred$obs, fit.glm$pred$Bad, levels = rev(levels(fit.glm$pred$obs)), auc=TRUE )

## By default, the x-axis goes backwards, used the option legacy.axes = TRUE to get 1-spec on the x-axis
plot(roc, legacy.axes = TRUE,print.auc=TRUE,grid=TRUE)
```




```
##
## Call:
## roc.default(response = fit.glm$pred$obs, predictor = fit.glm$pred$Bad,     levels = rev(levels(fit.g
##
## Data: fit.glm$pred$Bad in 700 controls (fit.glm$pred$obs Good) < 300 cases (fit.glm$pred$obs Bad).
## Area under the curve: 0.7785
```

```
## AUC for glm is ~78%
auc(roc)
```

```
## Area under the curve: 0.7785
```

KNN Scores

```
## KNN
confusionMatrix(fit.knn$pred$pred,fit.knn$pred$obs, positive="Bad")
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction  Bad Good
##       Bad   306  445
##       Good 1794 4455
##
##                Accuracy : 0.6801
##                  95% CI : (0.6691, 0.6911)
```

```
##     No Information Rate : 0.7
##     P-Value [Acc > NIR] : 0.9999
##
##                   Kappa : 0.0672
##  Mcnemar's Test P-Value : <2e-16
##
##             Sensitivity : 0.14571
##             Specificity : 0.90918
##          Pos Pred Value : 0.40746
##          Neg Pred Value : 0.71291
##              Prevalence : 0.30000
##          Detection Rate : 0.04371
##    Detection Prevalence : 0.10729
##       Balanced Accuracy : 0.52745
##
##        'Positive' Class : Bad
##
```

```
summary(fit.knn)
```

```
##             Length Class      Mode
## learn        2     -none-     list
## k            1     -none-     numeric
## theDots      0     -none-     list
## xNames      61     -none-     character
## problemType  1     -none-     character
## tuneValue    1     data.frame list
## obsLevels    2     -none-     character
```

```
fit.knn.sensitivity <- sensitivity(fit.knn$pred$pred, fit.knn$pred$obs, positive="Bad")

## Sensitivity for KNN at ~14%s
fit.knn.sensitivity
```

```
## [1] 0.1457143
```

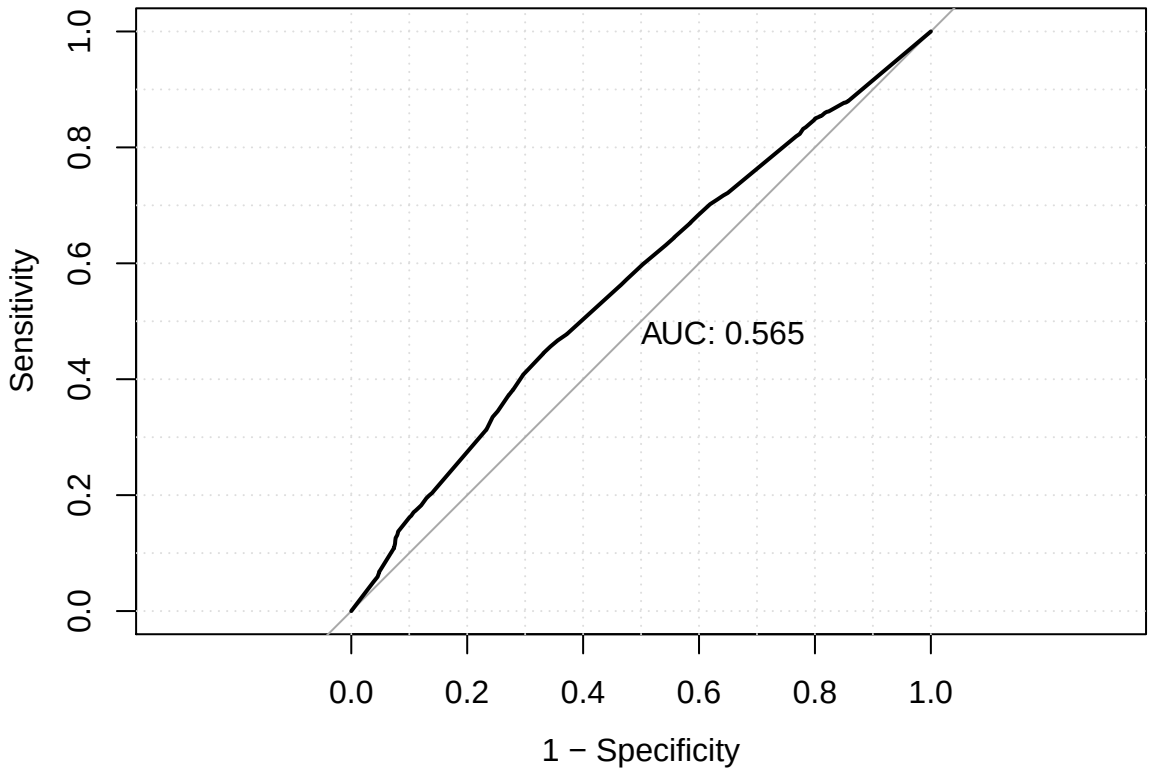
```
fit.knn.specificity <- specificity(fit.knn$pred$pred, fit.knn$pred$obs, positive="Bad")

## Specificity for KNN at ~90%
fit.knn.specificity
```

```
## [1] 0.9091837
```

```
## ROC Curve
## This function assumes that the second class is the event of interest, so we reverse the labels.
roc <- roc(fit.knn$pred$obs, fit.knn$pred$Bad, levels = rev(levels(fit.knn$pred$obs)), auc=TRUE )

## By default, the x-axis goes backwards, used the option legacy.axes = TRUE to get 1-spec on the x-axis
plot(roc, legacy.axes = TRUE,print.auc=TRUE,grid=TRUE)
```




```
##
## Call:
## roc.default(response = fit.knn$pred$obs, predictor = fit.knn$pred$Bad,     levels = rev(levels(fit.kn
##
## Data: fit.knn$pred$Bad in 4900 controls (fit.knn$pred$obs Good) < 2100 cases (fit.knn$pred$obs Bad).
## Area under the curve: 0.5646
```

```
## AUC for KNN is ~55%
auc(roc)
```

```
## Area under the curve: 0.5646
```

RPART Scores

```
## RPART
confusionMatrix(fit.rpart$pred$pred,fit.rpart$pred$obs, positive="Bad")
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction   Bad  Good
##       Bad   1874  1491
##       Good  4126 12509
##
##                Accuracy : 0.7192
##                  95% CI : (0.7129, 0.7254)
```

```
##     No Information Rate : 0.7
##     P-Value [Acc > NIR] : 1.44e-09
##
##                   Kappa : 0.2354
##  Mcnemar's Test P-Value : < 2.2e-16
##
##             Sensitivity : 0.3123
##             Specificity : 0.8935
##          Pos Pred Value : 0.5569
##          Neg Pred Value : 0.7520
##              Prevalence : 0.3000
##          Detection Rate : 0.0937
##    Detection Prevalence : 0.1683
##       Balanced Accuracy : 0.6029
##
##        'Positive' Class : Bad
##
```

```
#summary(fit.rpart)

fit.rpart.sensitivity <- sensitivity(fit.rpart$pred$pred, fit.rpart$pred$obs, positive="Bad")

## Sensitivity for RPART is 29%
fit.rpart.sensitivity
```

```
## [1] 0.3123333
```

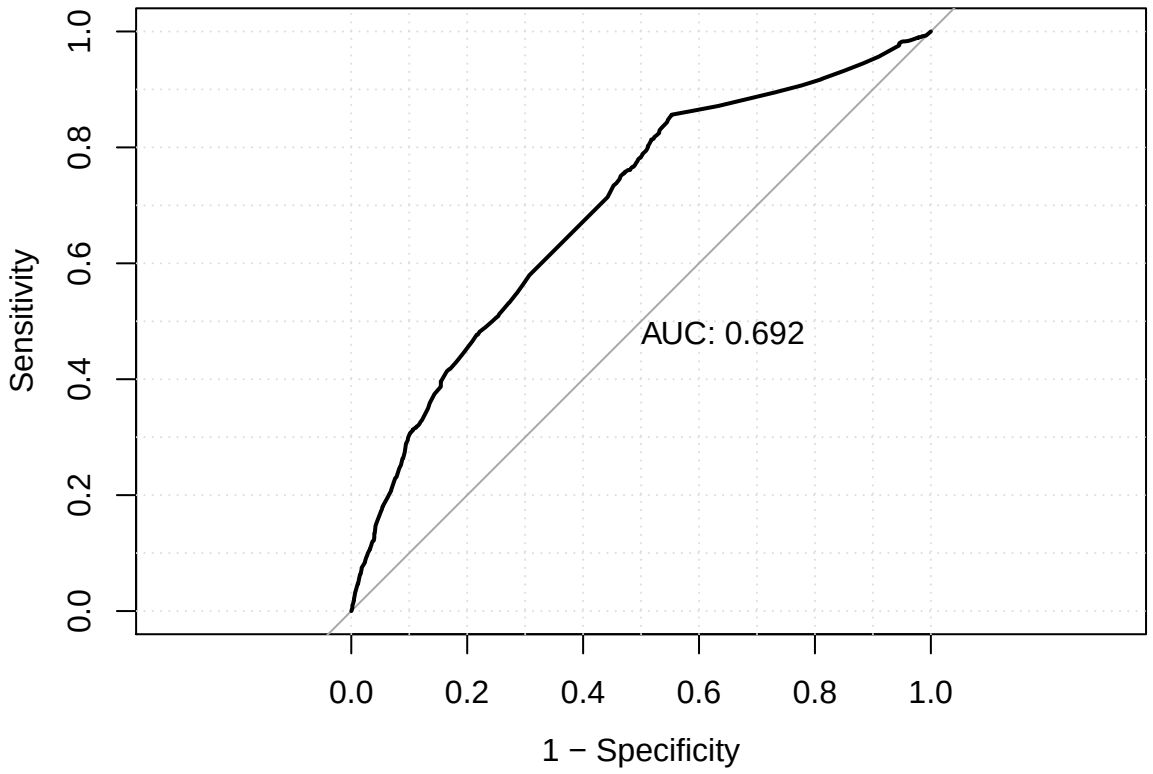
```
fit.rpart.specificity <- specificity(fit.rpart$pred$pred, fit.rpart$pred$obs, positive="Bad")

## Specificity for RPART at ~87%
fit.rpart.specificity
```

```
## [1] 0.8935
```

```
## ROC Curve
## This function assumes that the second class is the event of interest, so we reverse the labels.
roc <- roc(fit.rpart$pred$obs, fit.rpart$pred$Bad, levels = rev(levels(fit.rpart$pred$obs)), auc=TRUE )

## By default, the x-axis goes backwards, used the option legacy.axes = TRUE to get 1-spec on the x-axis
plot(roc, legacy.axes = TRUE,print.auc=TRUE,grid=TRUE)
```




```
##
## Call:
## roc.default(response = fit.rpart$pred$obs, predictor = fit.rpart$pred$Bad,     levels = rev(levels(fi
##
## Data: fit.rpart$pred$Bad in 14000 controls (fit.rpart$pred$obs Good) < 6000 cases (fit.rpart$pred$obs
## Area under the curve: 0.6924
```

```
## AUC for RPART is ~68%
auc(roc)
```

```
## Area under the curve: 0.6924
```

RF Scores

```
## RF
confusionMatrix(fit.rf$pred$pred,fit.rf$pred$obs, positive="Bad")
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction  Bad Good
##       Bad   298  155
##       Good  602 1945
##
##                Accuracy : 0.7477
##                  95% CI : (0.7317, 0.7631)
```

```
##     No Information Rate : 0.7
##     P-Value [Acc > NIR] : 3.985e-09
##
##                   Kappa : 0.2999
##  Mcnemar's Test P-Value : < 2.2e-16
##
##             Sensitivity : 0.33111
##             Specificity : 0.92619
##          Pos Pred Value : 0.65784
##          Neg Pred Value : 0.76364
##              Prevalence : 0.30000
##          Detection Rate : 0.09933
##    Detection Prevalence : 0.15100
##       Balanced Accuracy : 0.62865
##
##        'Positive' Class : Bad
##
```

```
summary(fit.rf)
```

```
##                 Length Class      Mode
## call               4   -none-     call
## type               1   -none-     character
## predicted       1000   factor     numeric
## err.rate        1500   -none-     numeric
## confusion          6   -none-     numeric
## votes           2000   matrix     numeric
## oob.times       1000   -none-     numeric
## classes            2   -none-     character
## importance        61   -none-     numeric
## importanceSD       0   -none-     NULL
## localImportance    0   -none-     NULL
## proximity          0   -none-     NULL
## ntree              1   -none-     numeric
## mtry               1   -none-     numeric
## forest            14   -none-     list
## y               1000   factor     numeric
## test               0   -none-     NULL
## inbag              0   -none-     NULL
## xNames            61   -none-     character
## problemType        1   -none-     character
## tuneValue          1   data.frame list
## obsLevels          2   -none-     character
```

```
fit.rf.sensitivity <- sensitivity(fit.rf$pred$pred, fit.rf$pred$obs, positive="Bad")

## Sensitivity for RF is 32%
fit.rf.sensitivity
```

```
## [1] 0.3311111
```

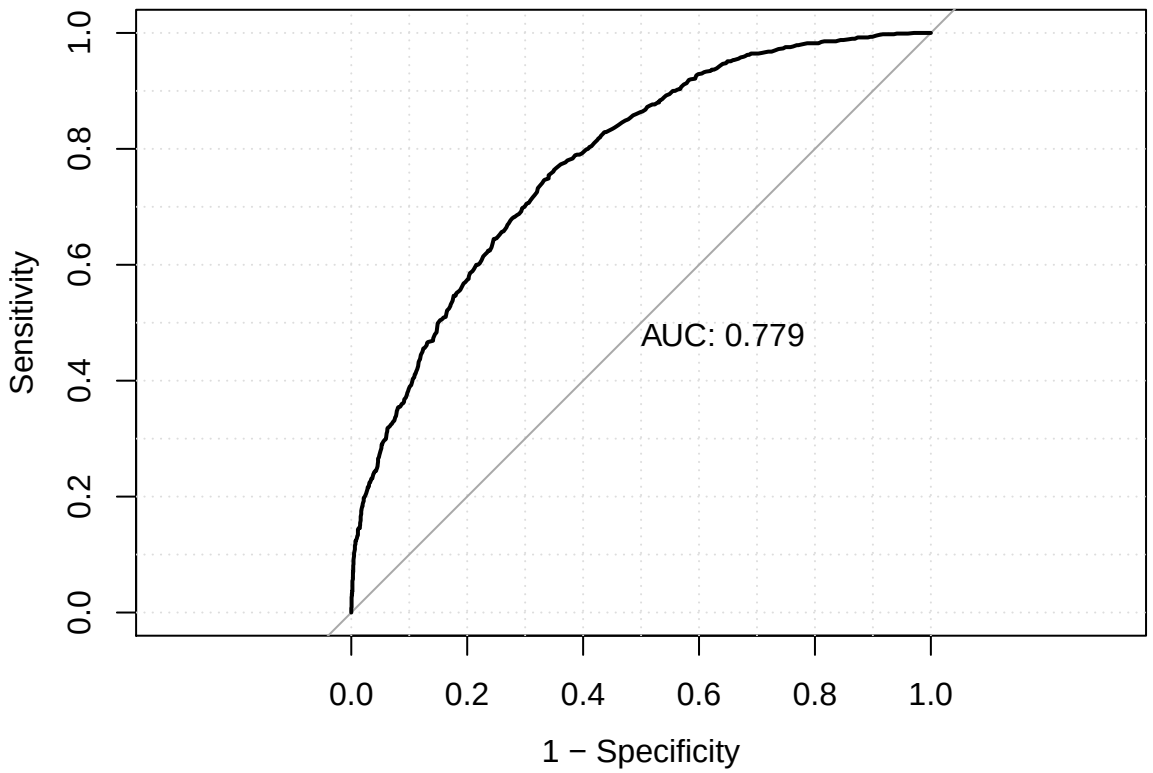
```
fit.rf.specificity <- specificity(fit.rf$pred$pred, fit.rf$pred$obs, positive="Bad")

## Specificity for RF is 91%
fit.rf.specificity
```

```
## [1] 0.9261905
```

```
## ROC Curve
## This function assumes that the second class is the event of interest, so we reverse the labels.
roc <- roc(fit.rf$pred$obs, fit.rf$pred$Bad, levels = rev(levels(fit.rf$pred$obs)), auc=TRUE )

## By default, the x-axis goes backwards, used the option legacy.axes = TRUE to get 1-spec on the x-axis
plot(roc, legacy.axes = TRUE,print.auc=TRUE,grid=TRUE)
```




```
##
## Call:
## roc.default(response = fit.rf$pred$obs, predictor = fit.rf$pred$Bad,     levels = rev(levels(fit.rf$p
##
## Data: fit.rf$pred$Bad in 2100 controls (fit.rf$pred$obs Good) < 900 cases (fit.rf$pred$obs Bad).
## Area under the curve: 0.7789
```

```
## AUC for RPART is ~77%
auc(roc)
```

```
## Area under the curve: 0.7789
```

Q: Which models would you select based on these tools?

```
.. # YOUR WORK HERE
"Q1: I would select the Random Forrest because the Sensivity(32%) and Specificity(91%) turned out to be
```

Q: If you assume that a `Class=="bad""` is 10 more costly than `Class=="good"`, determine your threshold for the model of your choice. Show your work.

```
    fit.rf$pred$allBad <- "Bad"
    fit.rf$pred$allGood <- "Good"

    confusionMatrix(fit.rf$pred$allBad,fit.rf$pred$obs, positive="Bad")
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction  Bad Good
##       Bad   900 2100
##       Good    0    0
##
##                Accuracy : 0.3
##                  95% CI : (0.2836, 0.3168)
##     No Information Rate : 0.7
##     P-Value [Acc > NIR] : 1
##
##                   Kappa : 0
##  Mcnemar's Test P-Value : <2e-16
##
##             Sensitivity : 1.0
##             Specificity : 0.0
##          Pos Pred Value : 0.3
##          Neg Pred Value : NaN
##              Prevalence : 0.3
##          Detection Rate : 0.3
##    Detection Prevalence : 1.0
##       Balanced Accuracy : 0.5
##
##        'Positive' Class : Bad
##
```

```
    confusionMatrix(fit.rf$pred$allGood,fit.rf$pred$obs, positive="Bad")
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction  Bad Good
##       Bad     0    0
##       Good  900 2100
##
##                Accuracy : 0.7
##                  95% CI : (0.6832, 0.7164)
##     No Information Rate : 0.7
```

```
##     P-Value [Acc > NIR] : 0.509
##
##                   Kappa : 0
##  Mcnemar's Test P-Value : <2e-16
##
##             Sensitivity : 0.0
##             Specificity : 1.0
##          Pos Pred Value : NaN
##          Neg Pred Value : 0.7
##              Prevalence : 0.3
##          Detection Rate : 0.0
##    Detection Prevalence : 0.0
##       Balanced Accuracy : 0.5
##
##        'Positive' Class : Bad
##
```