Homework 3

Matthew Perrotta

April 10, 2019

Load Libraries

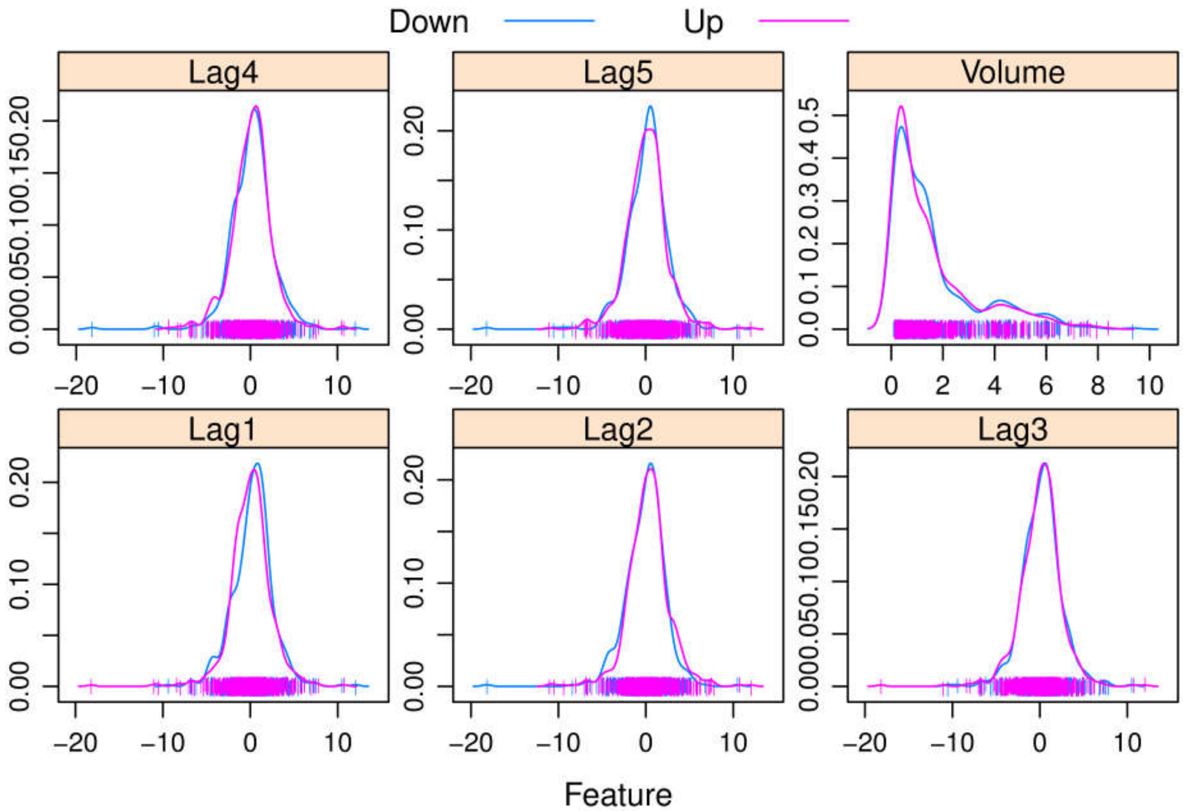
```
library(ISLR)
library(tidyverse)
library(caret)
library(corrplot)
library(pROC)
library(MASS)
```

```
data(Weekly)

x = model.matrix(Direction~., Weekly)[,3:8]

y = Weekly$Direction
```

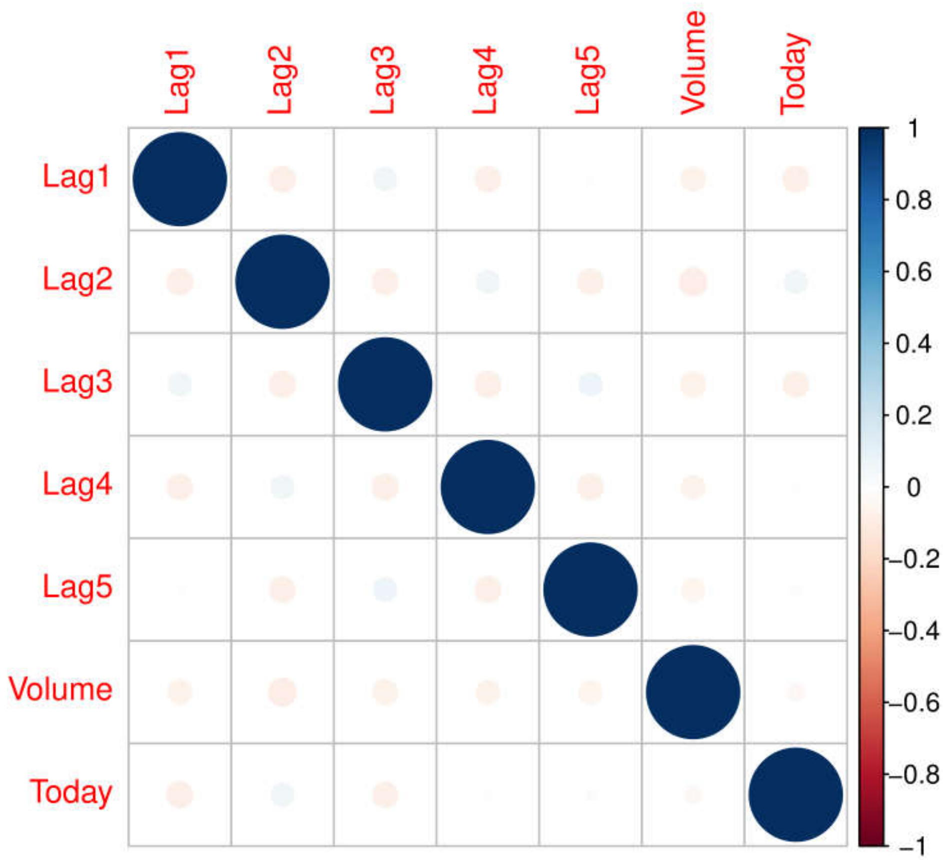
Problem (a)

EDA

```
featurePlot(x,
            y,
            scales = list(x=list(relation="free"),
                          y=list(relation="free")),
            plot = "density", pch = "|",
            auto.key = list(columns = 2))
```




```
corrplot::corrplot(cor(Weekly[2:8]))
```



All predictors are normally distributed except for `Volume`, which is right skewed. Also, according to the correlation plot, there is no collinearity between predictors.

Problem (b)

Logistic Regression

```
glm.fit <- glm(Direction ~ Lag1 + Lag2 + Lag3 + Lag4 + Lag5 + Volume,
               data = Weekly,
               family = binomial)

summary(glm.fit)
```

```
##
## Call:
## glm(formula = Direction ~ Lag1 + Lag2 + Lag3 + Lag4 + Lag5 +
##     Volume, family = binomial, data = Weekly)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.6949  -1.2565   0.9913   1.0849   1.4579
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
```

```
## (Intercept)  0.26686    0.08593   3.106   0.0019 **
## Lag1        -0.04127    0.02641  -1.563   0.1181
## Lag2         0.05844    0.02686   2.175   0.0296 *
## Lag3        -0.01606    0.02666  -0.602   0.5469
## Lag4        -0.02779    0.02646  -1.050   0.2937
## Lag5        -0.01447    0.02638  -0.549   0.5833
## Volume      -0.02274    0.03690  -0.616   0.5377
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 1496.2  on 1088  degrees of freedom
## Residual deviance: 1486.4  on 1082  degrees of freedom
## AIC: 1500.4
##
## Number of Fisher Scoring iterations: 4
```

The predictor `Lag2` is the only significant predictor with a p value of 0.0296.

Problem (c)

Confusion Matrix

```r
set.seed(1)
rowTrain <- createDataPartition(y,
                                p = 0.75,
                                list = FALSE)

test.pred.prob  <- predict(glm.fit, newdata = Weekly[-rowTrain,],
                           type = "response")
test.pred <- rep("Down", length(test.pred.prob))
test.pred[test.pred.prob > 0.5] <- "Up"

confusionMatrix(data = as.factor(test.pred),
                reference = Weekly$Direction[-rowTrain],
                positive = "Up")
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction Down  Up
##       Down    7  16
##       Up    114 135
##
##                Accuracy : 0.5221
##                  95% CI : (0.4609, 0.5827)
##     No Information Rate : 0.5551
##     P-Value [Acc > NIR] : 0.8767
##
##                   Kappa : -0.0523
##  Mcnemar's Test P-Value : <2e-16
```

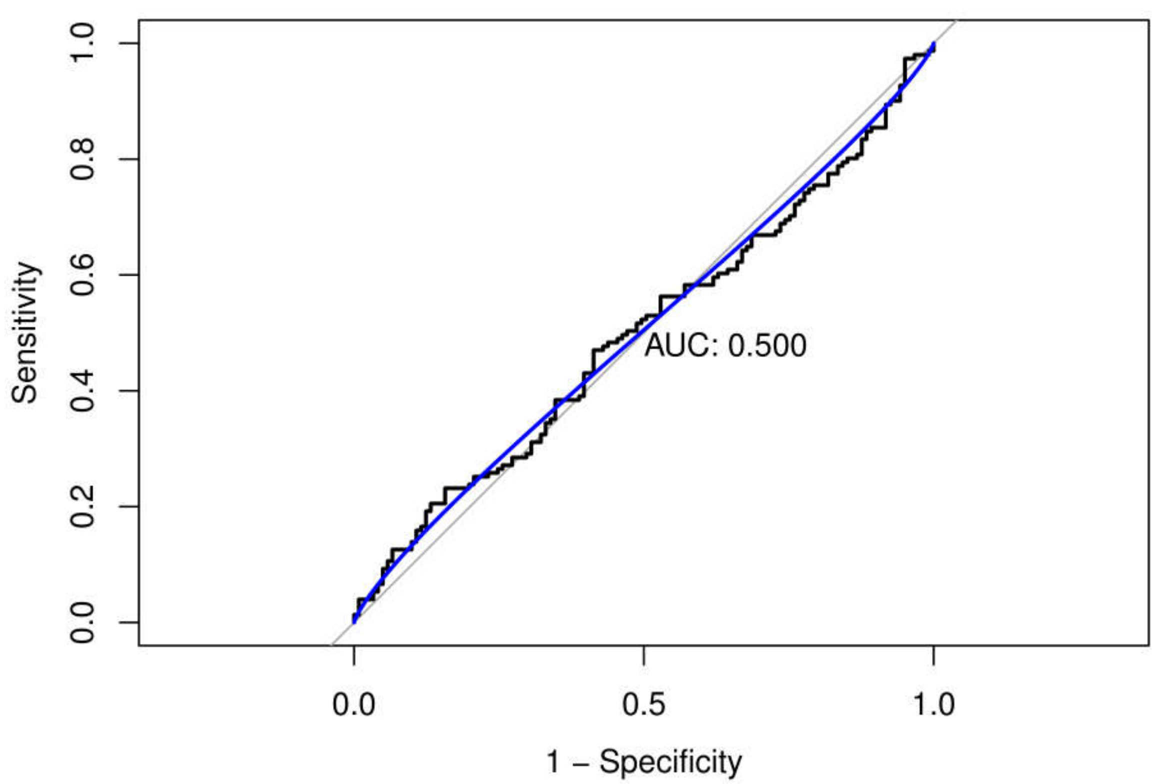
```
##
##             Sensitivity : 0.89404
##             Specificity : 0.05785
##          Pos Pred Value : 0.54217
##          Neg Pred Value : 0.30435
##              Prevalence : 0.55515
##          Detection Rate : 0.49632
##    Detection Prevalence : 0.91544
##       Balanced Accuracy : 0.47595
##
##        'Positive' Class : Up
##
```

The confusion matrix displays correct and incorrect predictions along the diagonals. The model `glm.fit` made 7 incorrect predictions and 135 correct predictions. The model has a total accuracy of 52.21%

Problem (d)

ROC Curve

```r
roc.glm <- roc(y[-rowTrain], test.pred.prob)

plot(roc.glm, legacy.axes = TRUE, print.auc = TRUE)
plot(smooth(roc.glm), col = 4, add = TRUE)
```



AUC = 0.500

Problem (e)

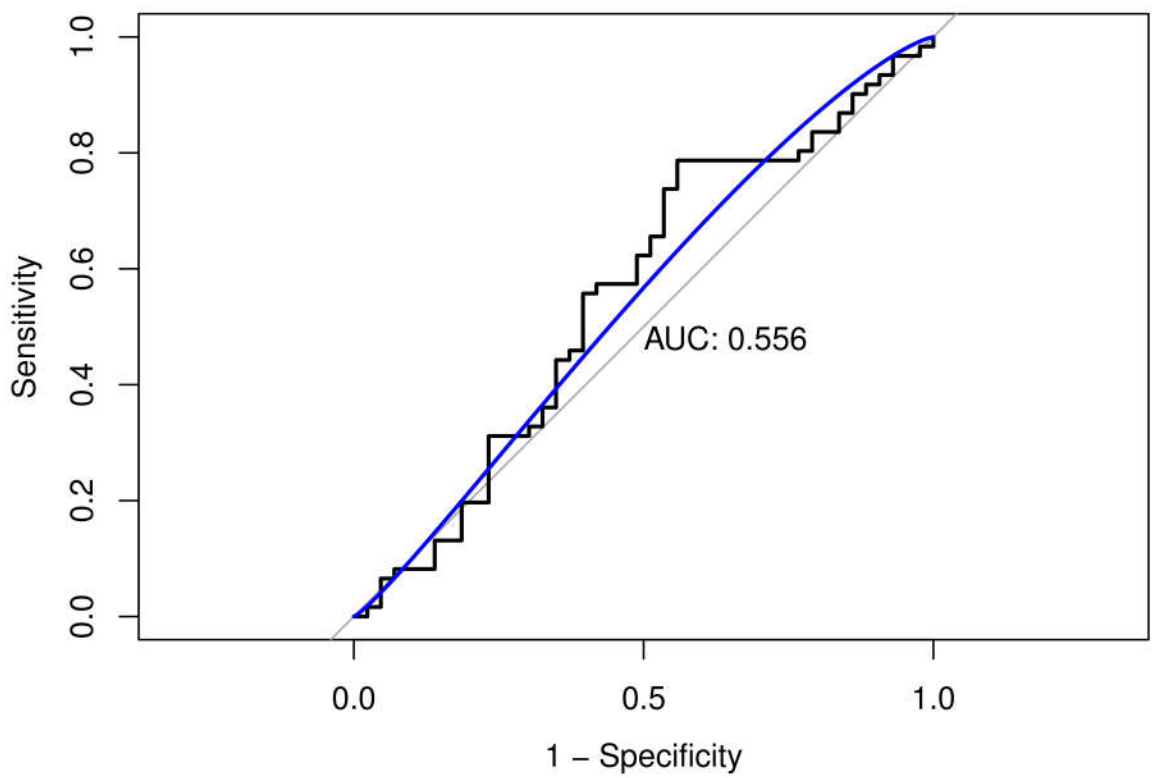
```r
train = subset(Weekly, Year >= 1990 & Year <= 2008)
test = subset(Weekly, Year > 2008)

trX = train[,2:3]

trY = train$Direction

teX = test[,2:3]

teY = test$Direction
```

```r
glm.fit2 <- glm(Direction ~ Lag1 + Lag2,
               data = train,
               family = binomial)

summary(glm.fit2)
```

```
##
## Call:
## glm(formula = Direction ~ Lag1 + Lag2, family = binomial, data = train)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.6149  -1.2565   0.9989   1.0875   1.5330
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)  0.21109    0.06456   3.269  0.00108 **
## Lag1        -0.05421    0.02886  -1.878  0.06034 .
## Lag2         0.05384    0.02905   1.854  0.06379 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 1354.7  on 984  degrees of freedom
## Residual deviance: 1347.0  on 982  degrees of freedom
## AIC: 1353
##
## Number of Fisher Scoring iterations: 4
```

```r
test.pred.prob2  <- predict(glm.fit2, teX,
                            type = "response")
test.pred2 <- rep("Down", length(test.pred.prob2))
test.pred2[test.pred.prob2 > 0.5] <- "Up"
```

```
roc.glm2 <- roc(teY, test.pred.prob2)

plot(roc.glm2, legacy.axes = TRUE, print.auc = TRUE)
plot(smooth(roc.glm2), col = 4, add = TRUE)
```



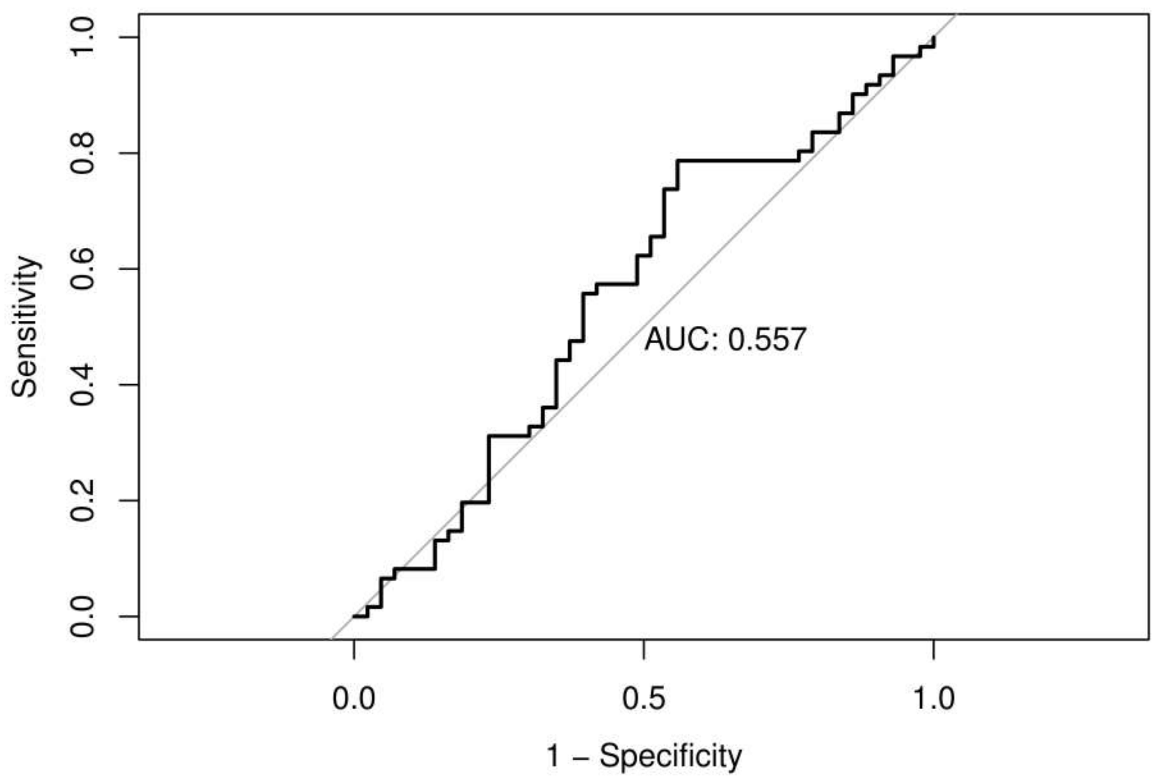
AUC = 0.556

Problem (f)

LDA

```
lda.fit <- lda(Direction ~ Lag1 + Lag2, data = train)
```

```
lda.pred <- predict(lda.fit, newdata = teX)

roc.lda <- roc(teY, lda.pred$posterior[,2],
               levels = c("Down", "Up"))

plot(roc.lda, legacy.axes = TRUE, print.auc = TRUE)
```

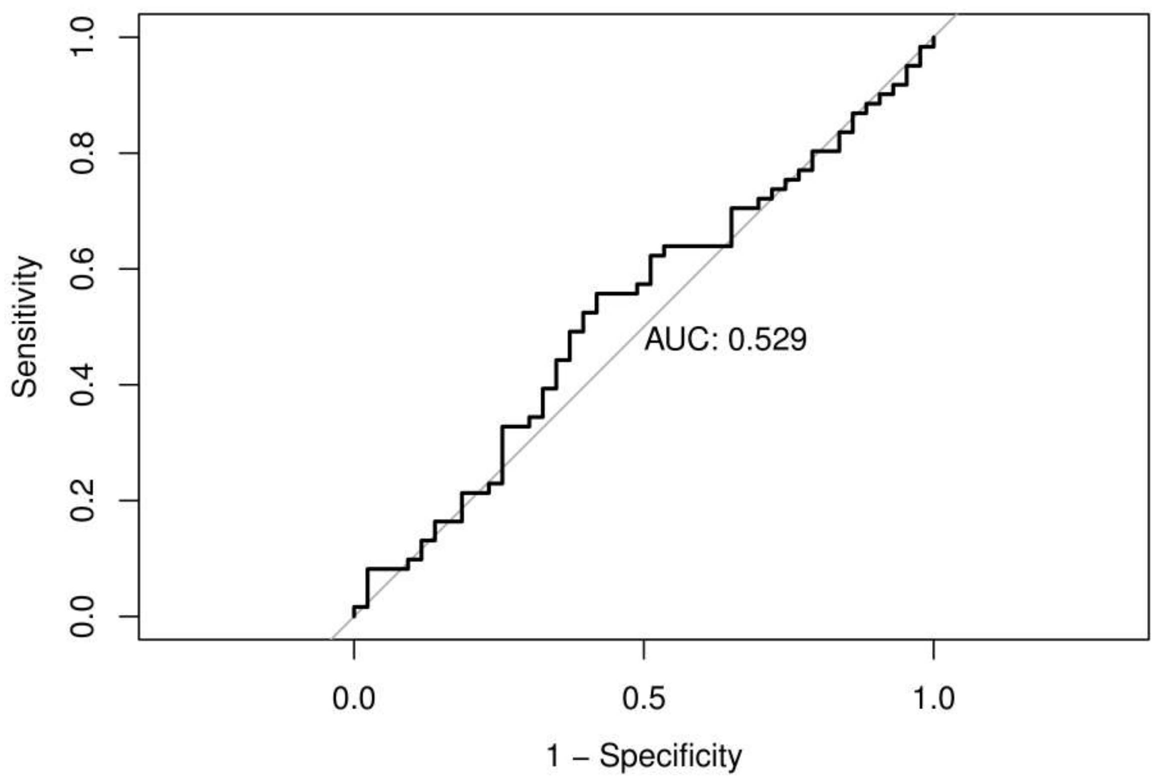


AUC = 0.557

QDA

```
qda.fit <- qda(Direction ~ Lag1 + Lag2, data = train)
```

```
qda.pred <- predict(qda.fit, newdata = teX)

roc.qda <- roc(teY, qda.pred$posterior[,2],
               levels = c("Down", "Up"))

plot(roc.qda, legacy.axes = TRUE, print.auc = TRUE)
```



AUC = 0.529

Problem (g)

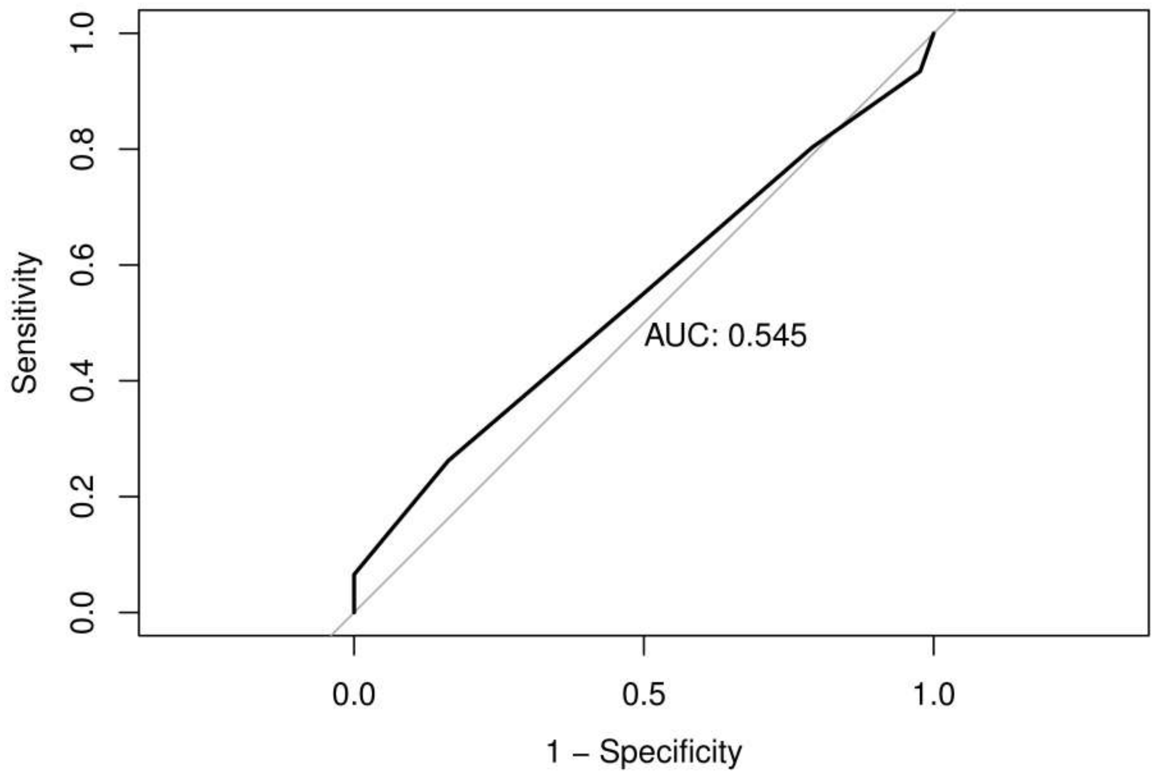
KNN

```
set.seed(1)

ctrl <- trainControl(method = "repeatedcv",
                     repeats = 5,
                     summaryFunction = twoClassSummary,
                     classProbs = TRUE)

model.knn <- train(trX,
                   trY,
                   method = "knn",
                   preProcess = c("center","scale"),
                   tuneGrid = data.frame(k = seq(1,20,by = 1)),
                   trControl = ctrl)
```

k = 7

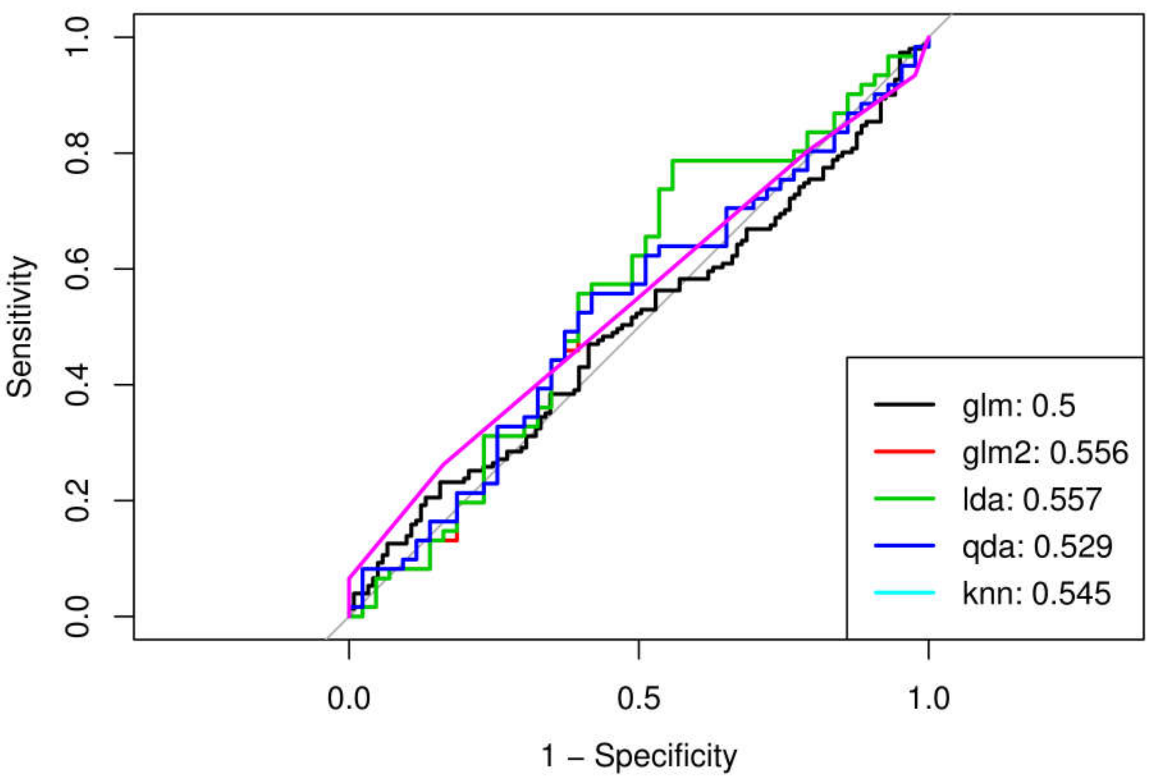
```
knn.pred <- predict(model.knn, newdata = teX, type = 'prob')[, 2]
```

```r
roc.knn <- roc(teY, knn.pred)

plot(roc.knn, legacy.axes = TRUE, print.auc = TRUE)
```



$\text{AUC} = 0.545$

```r
auc <- c(roc.glm$auc[1], roc.glm2$auc[1], roc.lda$auc[1],
         roc.qda$auc[1], roc.knn$auc[1])

plot(roc.glm, legacy.axes = TRUE)
plot(roc.glm2, col = 2, add = TRUE)
plot(roc.lda, col = 3, add = TRUE)
plot(roc.qda, col = 4, add = TRUE)
plot(roc.knn, col = 6, add = TRUE)
modelNames <- c("glm","glm2","lda","qda","knn")
legend("bottomright", legend = paste0(modelNames, ": ", round(auc,3)),
       col = 1:6, lwd = 2)
```



The greater the area under the ROC the better the model, with the best possible model having an AUC of 1. From the models produced above, LDA has the highest AUC and is therefore the better model of those tested. It should be noted that the two predictors `Lga1` and `Lga2` are not significantly associated with the response variable at an alpha of 0.05. While insignificant, they're p values are close enough to still consider these variables as useful predictors.