Homework 3

Na Yun Cho

```
library(tidyverse) # data manipulation
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.0 --

## v ggplot2 3.3.3     v purrr   0.3.4
## v tibble  3.0.6     v dplyr   1.0.4
## v tidyr   1.1.2     v stringr 1.4.0
## v readr   1.4.0     v forcats 0.5.1

## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```
library(dplyr)
library(ISLR) # for data
library(janitor) # clean names
```

```
##
## Attaching package: 'janitor'

## The following objects are masked from 'package:stats':
##
##     chisq.test, fisher.test
```

```
library(AppliedPredictiveModeling) # better plots
library(caret) # modeling
```

```
## Loading required package: lattice

##
## Attaching package: 'caret'

## The following object is masked from 'package:purrr':
##
##     lift
```

```
library(corrplot) # correlation plots
```

```
## corrplot 0.84 loaded
```

```
library(pROC) # ROC curve
```

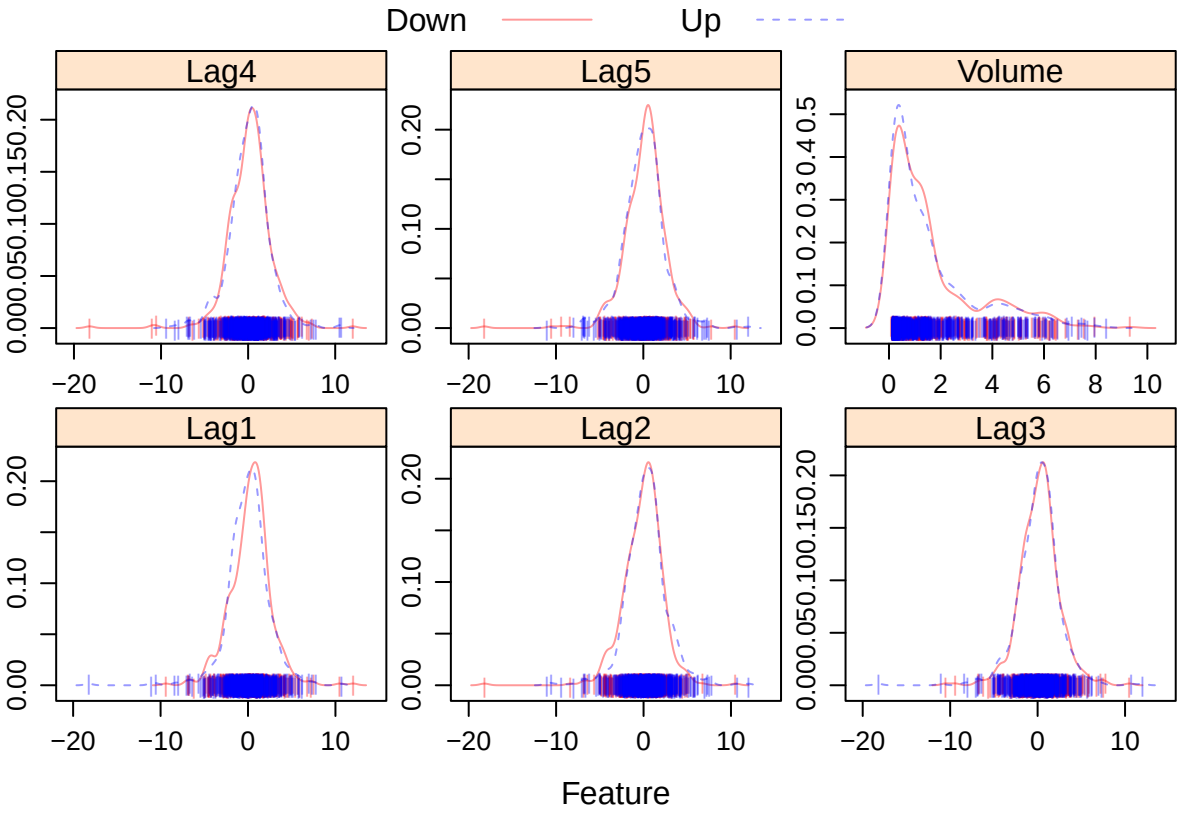
```
## Type 'citation("pROC")' for a citation.

##
## Attaching package: 'pROC'

## The following objects are masked from 'package:stats':
##
##     cov, smooth, var
```

```
library(MASS) # LDA
```

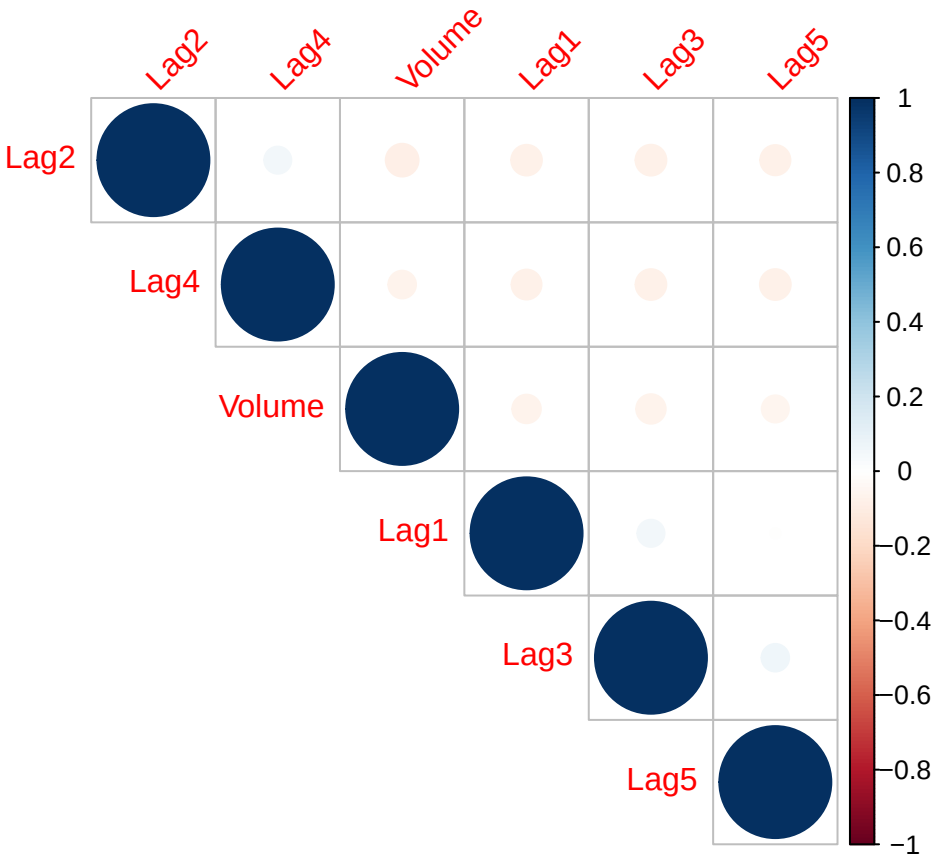
```
##
## Attaching package: 'MASS'

## The following object is masked from 'package:dplyr':
##
##     select
```

(a)

```
# Import data
data(Weekly)
weekly1 <-
  Weekly %>%
  dplyr::select(-Today, -Year)

# Feature plots

theme1 <- transparentTheme(trans = .4)
trellis.par.set(theme1)

featurePlot(x = weekly1[, 1:6],
            y = weekly1$Direction,
            scales = list(x = list(relation = "free"),
                          y = list(relation = "free")),
            plot = "density", pch = "|",
            auto.key = list(columns = 2))
```




```
# Correlation plot
corrplot(cor(weekly1[,-7]), tl.srt = 45, order = 'hclust', type = 'upper')
```



The feature plots show that the distribution of the response classes (Up or Down) highly overlap for each feature. The correlation plot shows that there is not much pairwise correlation between the features.

(b)

```
weekly <-
  Weekly %>%
  dplyr::select(-Today)

#training set
train_df =
  weekly %>%
  filter(Year < 2009) %>%
  dplyr::select(-Year)

#test set
test_df = anti_join(weekly, train_df)
```

```
## Joining, by = c("Lag1", "Lag2", "Lag3", "Lag4", "Lag5", "Volume", "Direction")
```

```
contrasts(weekly$Direction)
```

```
##      Up
## Down  0
## Up    1
```

```r
#perform logistic regression
glm_train <- glm(Direction ~ .,
                 data = train_df,
                 family = binomial(link = "logit"))
summary(glm_train)
```

```
##
## Call:
## glm(formula = Direction ~ ., family = binomial(link = "logit"),
##     data = train_df)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.7186  -1.2498   0.9823   1.0841   1.4911
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)  0.33258    0.09421   3.530 0.000415 ***
## Lag1        -0.06231    0.02935  -2.123 0.033762 *
## Lag2         0.04468    0.02982   1.499 0.134002
## Lag3        -0.01546    0.02948  -0.524 0.599933
## Lag4        -0.03111    0.02924  -1.064 0.287241
## Lag5        -0.03775    0.02924  -1.291 0.196774
## Volume      -0.08972    0.05410  -1.658 0.097240 .
## ---
## 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: 1342.3  on 978  degrees of freedom
## AIC: 1356.3
##
## Number of Fisher Scoring iterations: 4
```

It seems that the Lag1 predictor is statistically significant.

```r
#compute the confusion matrix and overall fraction of correct predictions using the test data
test.pred.prob <- predict(glm_train, newdata = test_df,
                          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 = test_df$Direction,
                positive = "Up")
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction Down Up
```

```
##       Down   31 44
##       Up     12 17
##
##                Accuracy : 0.4615
##                  95% CI : (0.3633, 0.562)
##     No Information Rate : 0.5865
##     P-Value [Acc > NIR] : 0.9962
##
##                   Kappa : -3e-04
##
##  Mcnemar's Test P-Value : 3.435e-05
##
##             Sensitivity : 0.2787
##             Specificity : 0.7209
##          Pos Pred Value : 0.5862
##          Neg Pred Value : 0.4133
##              Prevalence : 0.5865
##          Detection Rate : 0.1635
##    Detection Prevalence : 0.2788
##       Balanced Accuracy : 0.4998
##
##        'Positive' Class : Up
##
```

The confusion matrix indicates that the fraction of correct predictions is 0.4615. Because the No Information Rate value (0.5865) is higher than the Accuracy value, the classifier is not very meaningful. The large P-value also indicates that the accuracy is not significantly better than the No Information Rate. In addition, the Kappa Value that is close to 0 indicates that the probability of observed agreement is the same as the probability of agreement by chance. Thus, it indicates that the classification performance is not good. The sensitivity (0.2787) is low while the specificity (0.7209) is relatively high. Sensitivity measures the proportion of true positives that are correctly predicted and specificity measures the proportion of true negatives that are correctly predicted.

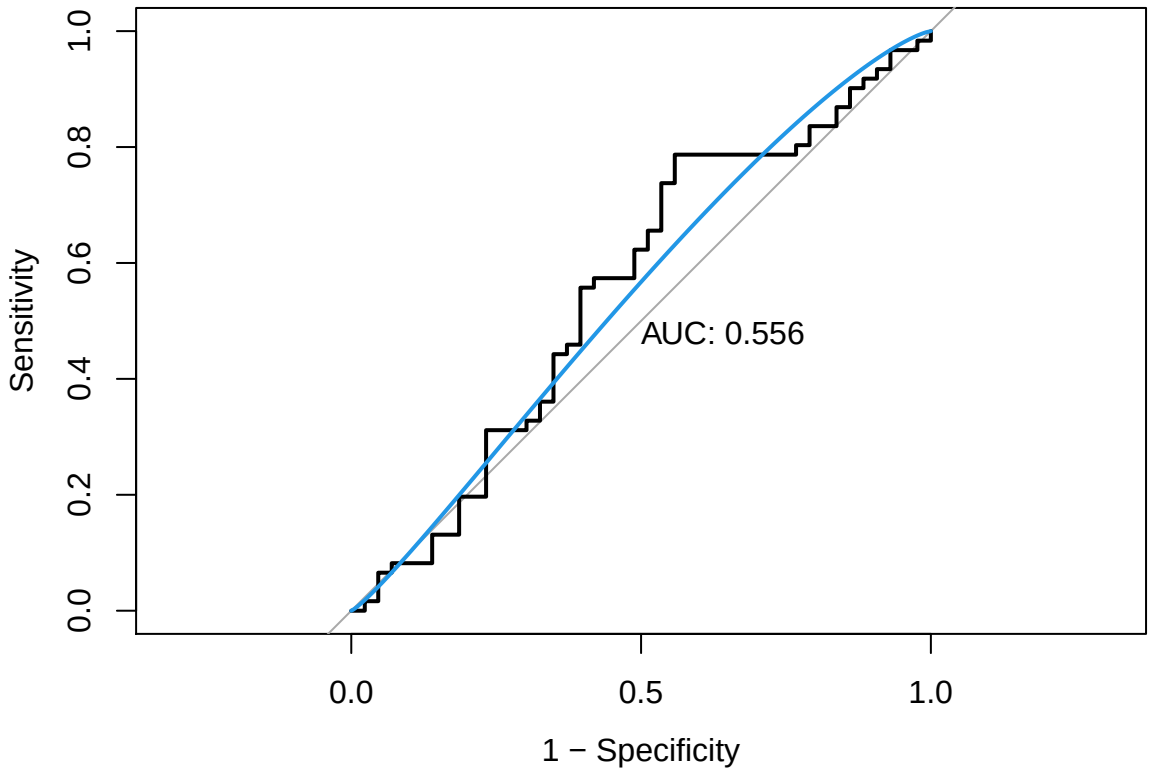
(c)

```
glm_train2 <- glm(Direction ~ Lag1 + Lag2,
                  data = train_df,
                  family = binomial(link = "logit"))


test.pred.prob <- predict(glm_train2, newdata = test_df,
                          type = "response")
test.pred <- rep("neg", length(test.pred.prob))
test.pred[test.pred.prob>0.5] <- "pos"

roc.glm <- roc(test_df$Direction, test.pred.prob)
```

```
## Setting levels: control = Down, case = Up
```

```
## Setting direction: controls < cases
```

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



The AUC is 0.5558.

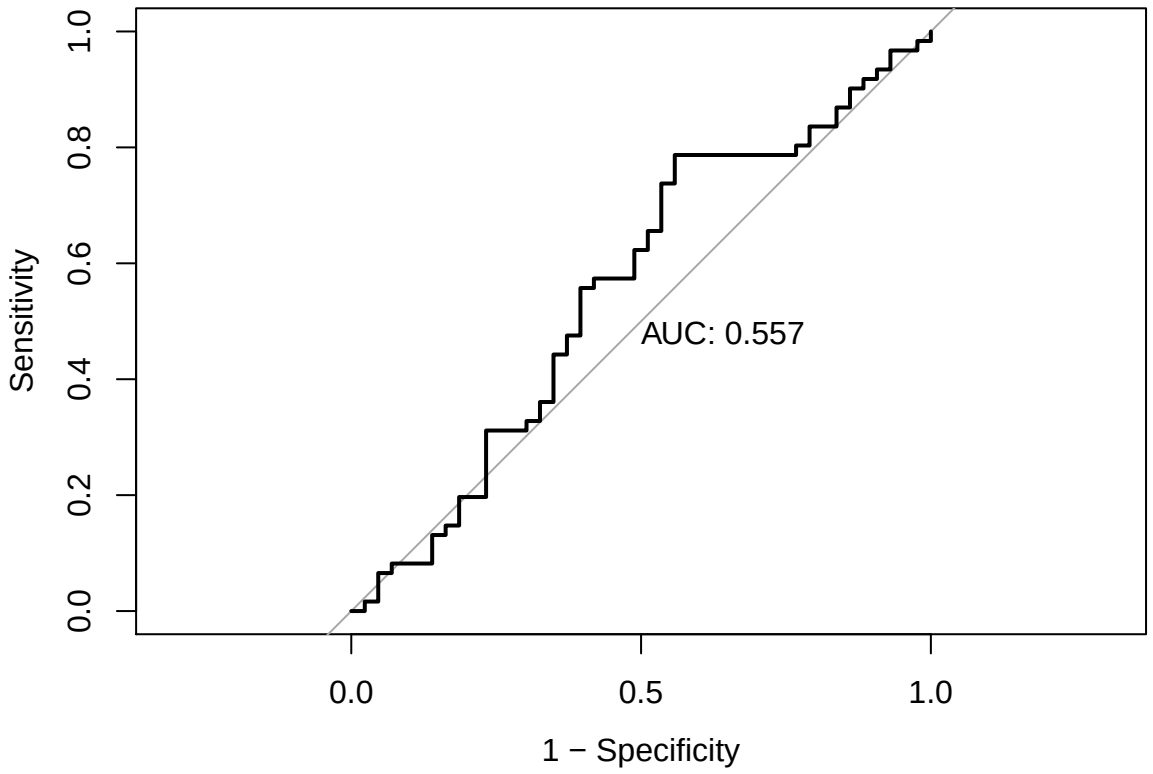
(d)

```
# LDA
lda.fit <- lda(Direction~ Lag1 + Lag2,
               data = train_df)
lda.pred <- predict(lda.fit, newdata = test_df)
roc.lda <- roc(test_df$Direction, lda.pred$posterior[,2])
```

```
## Setting levels: control = Down, case = Up
```

```
## Setting direction: controls < cases
```

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

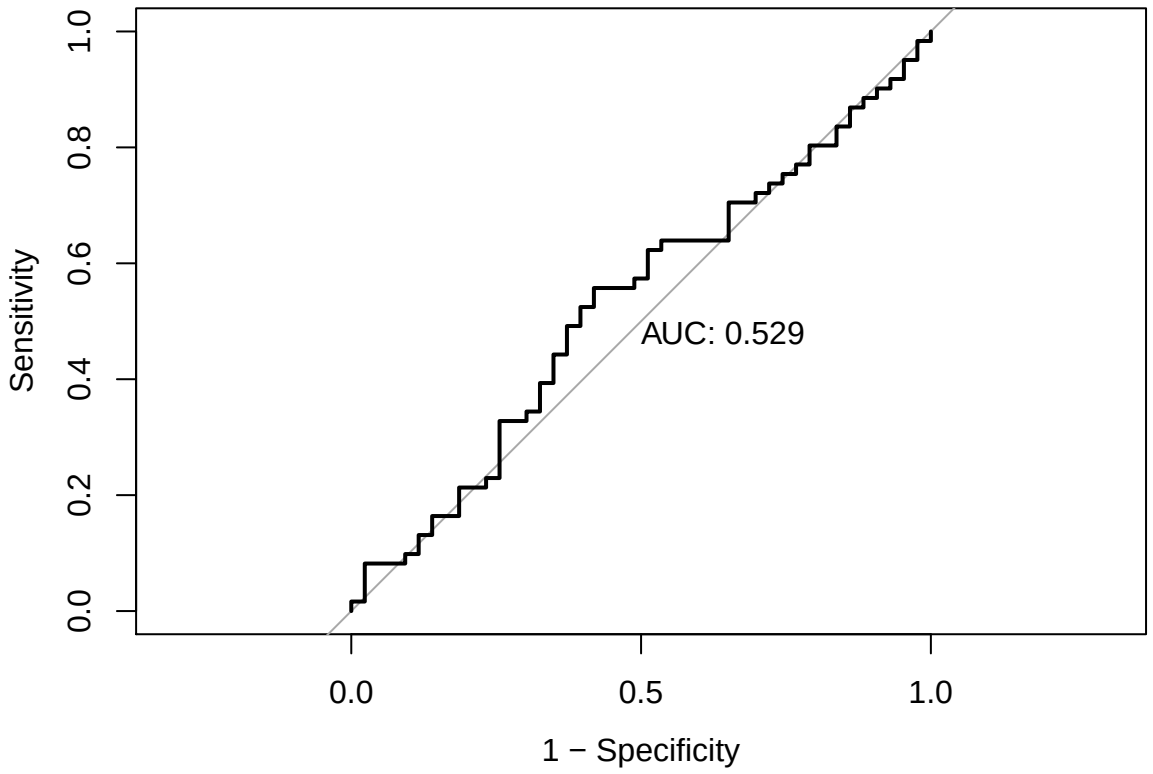



```
# QDA
qda.fit <- qda(Direction ~ Lag1 + Lag2,
               data = train_df)
qda.pred <- predict(qda.fit, newdata = test_df)
roc.qda <- roc(test_df$Direction, qda.pred$posterior[,2])
```

```
## Setting levels: control = Down, case = Up
```

```
## Setting direction: controls > cases
```

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




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

The AUC from LDA is 0.55661 and the AUC from QDA is 0.5288.

(e)

```
set.seed(10000)
ctrl <- trainControl(method = "repeatedcv",
                     repeats = 5,
                     summaryFunction = twoClassSummary,
                     classProbs = TRUE)

model.knn <- train(x = train_df[,1:2],
                   y = train_df$Direction,
                   method = "knn",
                   preProcess = c("center", "scale"),
                   tuneGrid = data.frame(k = seq(1, 499, by = 5)),
                   trControl = ctrl)
```

```
## Warning in train.default(x = train_df[, 1:2], y = train_df$Direction, method
## = "knn", : The metric "Accuracy" was not in the result set. ROC will be used
## instead.
```

```
model.knn$bestTune
```

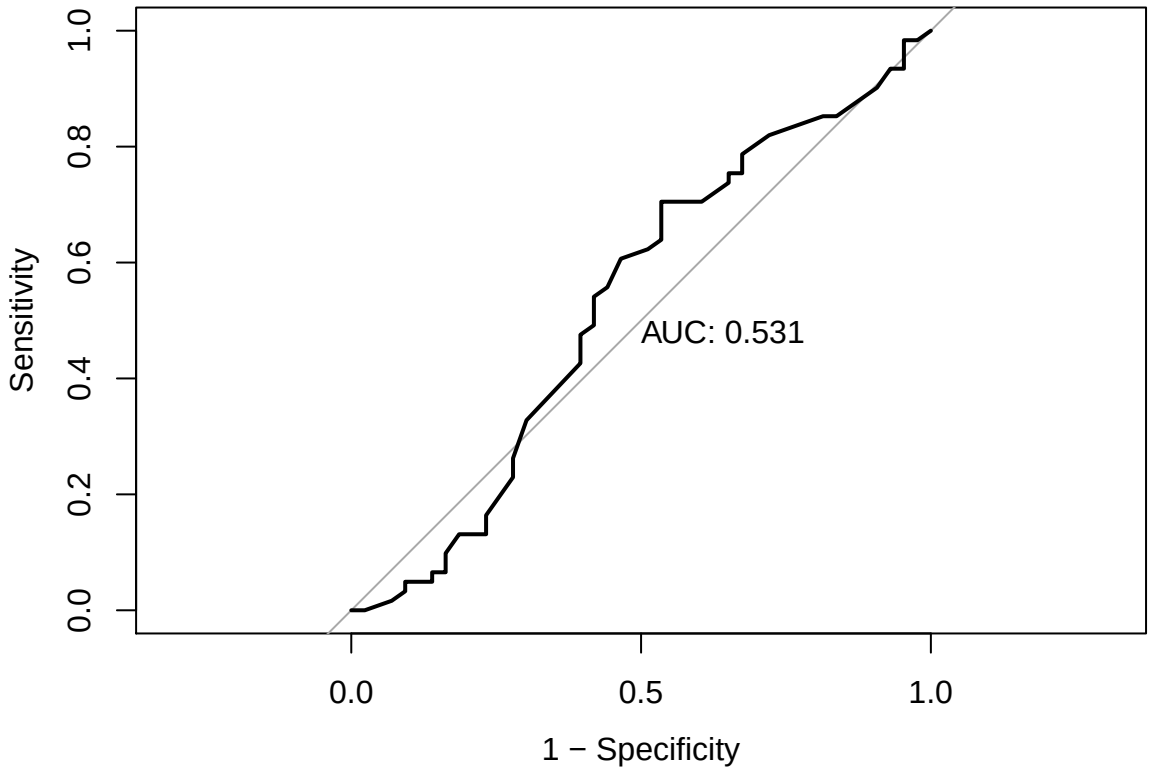
```
##      k
## 99 491
```

```
pred_knn = predict(model.knn, newdata = test_df, type = 'prob')
roc_knn <- roc(test_df$Direction, pred_knn[,2])
```

```
## Setting levels: control = Down, case = Up
```

```
## Setting direction: controls < cases
```

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



The AUC is 0.531.

Discussion of Results: I have fit different types of models using the training data from 1990 to 2008 and plotted the ROC curves using the test data from 2009 to 2010. From comparing all the AUC values, I can see that LDA generates the largest AUC among these models, which indicates that it shows the best classification performance. However, it is also apparent that the AUC's for these four models are similar and slightly above 0.5, which indicates that it is hard to correctly classify the response 'Direction' with the given predictors. In addition, the ROC for logistic regression, LDA, and QDA seem relatively stable since they do not involve tuning parameters. On the other hand, the ROC for the KNN method seems relatively less stable.