ds2_hw3

Minghe Wang

2025-04-01

```r
auto <- read_csv("./auto.csv")
```

```
## Rows: 392 Columns: 8
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr (1): mpg_cat
## dbl (7): cylinders, displacement, horsepower, weight, acceleration, year, or...
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```r
str(auto)
```

```
## spc_tbl_ [392 x 8] (S3: spec_tbl_df/tbl_df/tbl/data.frame)
##  $ cylinders   : num [1:392] 8 8 8 8 8 8 8 8 8 8 ...
##  $ displacement: num [1:392] 307 350 318 304 302 429 454 440 455 390 ...
##  $ horsepower  : num [1:392] 130 165 150 150 140 198 220 215 225 190 ...
##  $ weight      : num [1:392] 3504 3693 3436 3433 3449 ...
##  $ acceleration: num [1:392] 12 11.5 11 12 10.5 10 9 8.5 10 8.5 ...
##  $ year        : num [1:392] 70 70 70 70 70 70 70 70 70 70 ...
##  $ origin      : num [1:392] 1 1 1 1 1 1 1 1 1 1 ...
##  $ mpg_cat     : chr [1:392] "low" "low" "low" "low" ...
##  - attr(*, "spec")=
##   .. cols(
##   ..   cylinders = col_double(),
##   ..   displacement = col_double(),
##   ..   horsepower = col_double(),
##   ..   weight = col_double(),
##   ..   acceleration = col_double(),
##   ..   year = col_double(),
##   ..   origin = col_double(),
##   ..   mpg_cat = col_character()
##   .. )
##  - attr(*, "problems")=<externalptr>
```

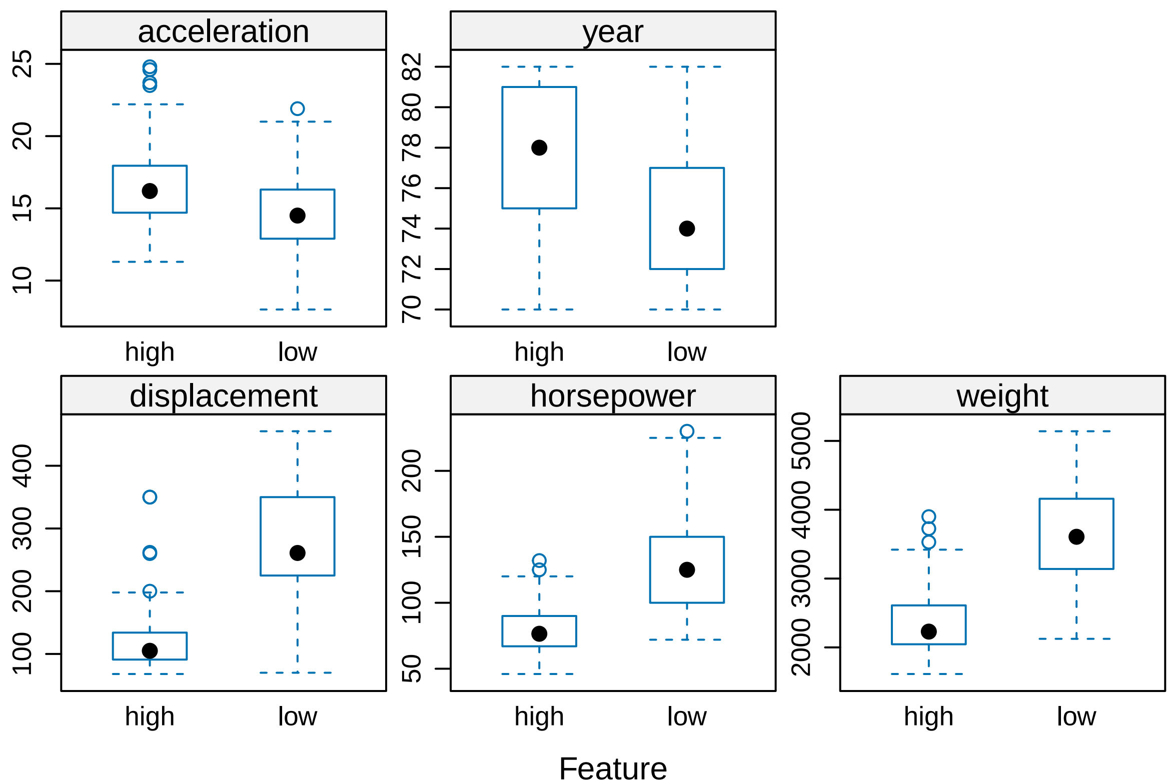
```r
auto$origin <- as.factor(auto$origin)
auto$cylinders <- as.factor(auto$cylinders)
auto$mpg_cat <- as.factor(auto$mpg_cat)
contrasts(auto$mpg_cat)
```

```
##      low
## high   0
## low    1
```

```r
auto <- na.omit(auto)

set.seed(1)
split_auto <- initial_split(auto, prop = 0.7)
train <- training(split_auto)
test <- testing(split_auto)

featurePlot(x = auto[, c("displacement", "horsepower", "weight", "acceleration", "year")],
            y = auto$mpg_cat,
            scales = list(x = list(relation = "free"),
                          y = list(relation = "free")),
            plot = "box")
```



Q1

```r
ctrl <- trainControl(method = "cv",
                     number = 10,
                     summaryFunction = twoClassSummary,
                     classProbs = TRUE)

glmnetGrid <- expand.grid(.alpha = seq(0, 1, length = 21), .lambda = exp(seq(-5, 5, length = 50)))

set.seed(1)
model.glmnet <- train(x = train[1:7],
                    y = train$mpg_cat,
                    method = "glmnet",
                    tuneGrid = glmnetGrid,
```

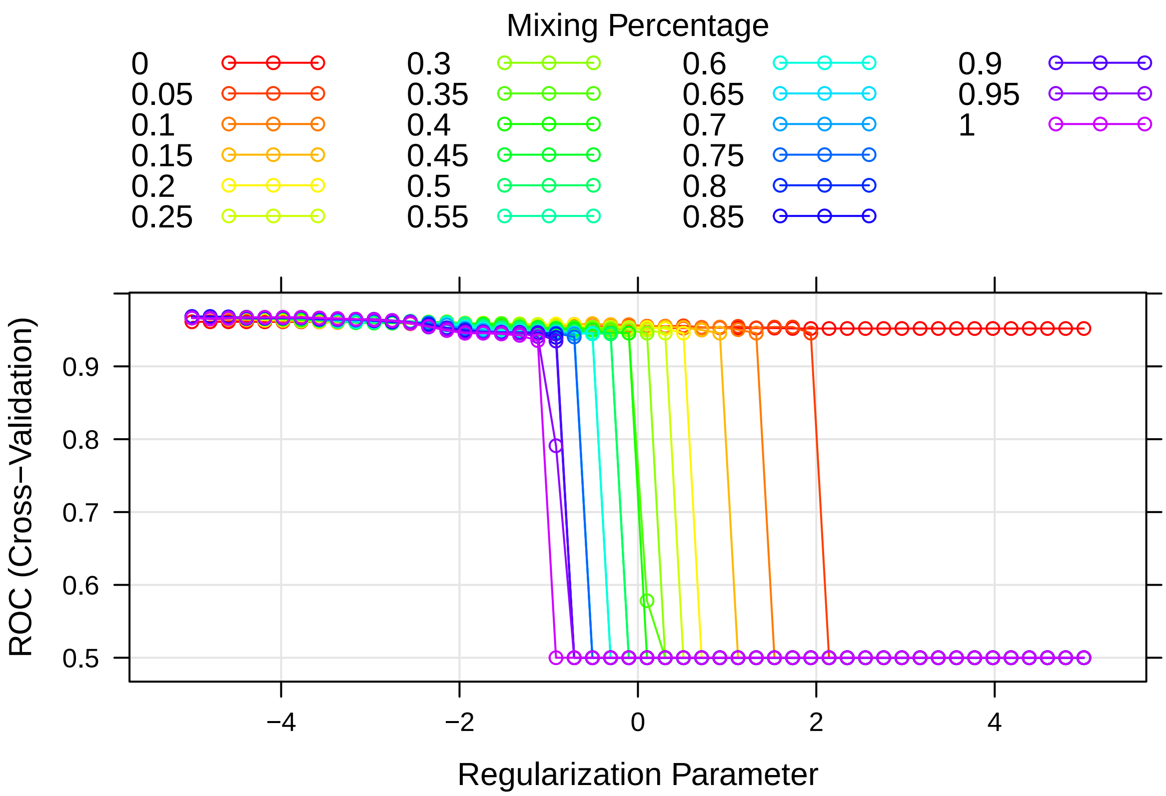
```r
                    metric = "ROC",
                    trControl = ctrl)

model.glmnet$bestTune
```

```
##     alpha      lambda
## 852  0.85 0.008263406
```

```r
best_model.glmnet <- model.glmnet$finalModel

myCol <- rainbow(25)
myPar <- list(superpose.symbol = list(col = myCol),
superpose.line = list(col = myCol))
plot(model.glmnet, par.settings = myPar, xTrans = function(x) log(x))
```




```r
as.matrix(coef(best_model.glmnet, s = model.glmnet$bestTune$lambda))
```

```
##                       s1
## (Intercept)  10.267331200
## cylinders     0.083132432
## displacement  0.007502264
## horsepower    0.020334125
## weight        0.002718235
## acceleration  0.000000000
## year         -0.284260667
## origin       -0.027856681
```

According to our best model, only the `acceleration` is redundant since it's coefficient is exactly 0. The coefficient table represents the effect of each predictor on the probability of a car having high or low gas mileage. Although weight only have weak effect, we still consider it somewhat important.

B

```r
set.seed(1)
model.mars <- train(x = train[1:7],
                    y = train$mpg_cat,
                    method = "earth",
                    tuneGrid = expand.grid(degree = 1:5,
                                           nprune = 2:25),
                    metric = "ROC",
                    trControl = ctrl)
```

```
## Loading required package: earth
## Loading required package: Formula
## Loading required package: plotmo
## Loading required package: plotrix
```

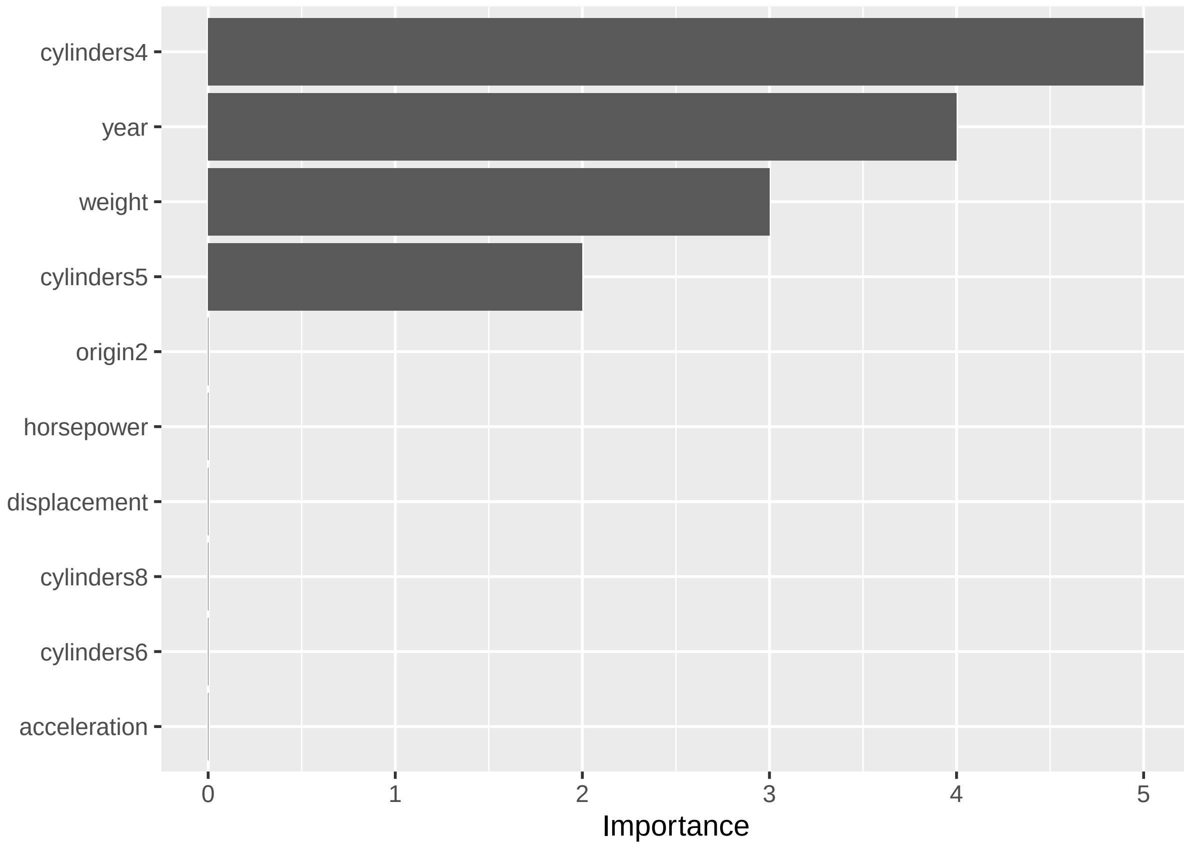
```r
model.mars$bestTune
```

```
##   nprune degree
## 5      6      1
```

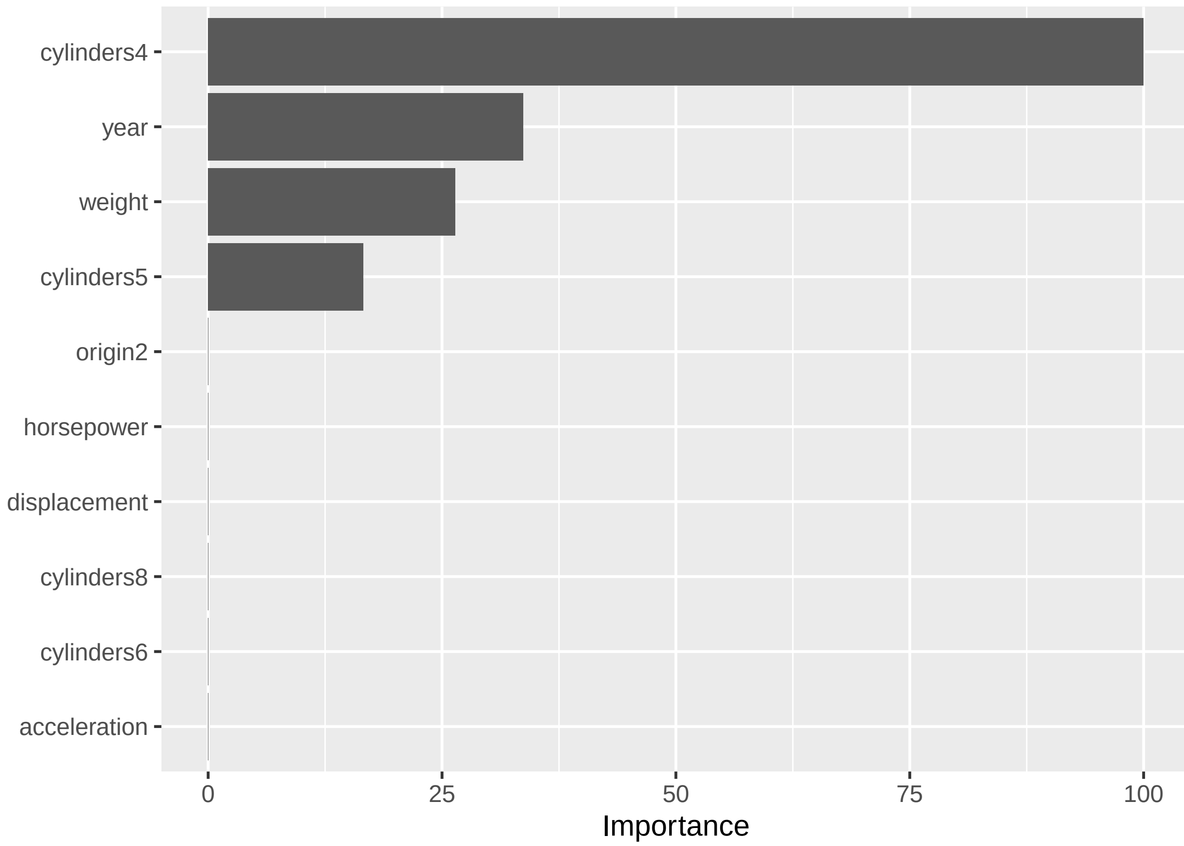
```r
summary(model.mars)
```

```
## Call: earth(x=tbl_df[274,7], y=factor.object, keepxy=TRUE,
##             glm=list(family=function.object, maxit=100), degree=1, nprune=6)
##
## GLM coefficients
##                         low
## (Intercept)       7.0613415
## cylinders4       -2.5619102
## cylinders5      -18.5646664
## h(3459-weight)   -0.0045272
## h(72-year)       -1.0213086
## h(year-72)       -0.6078387
##
## GLM (family binomial, link logit):
##  nulldev  df       dev  df   devratio     AIC iters converged
##  379.786 273   91.5397 268      0.759   103.5    15         1
##
## Earth selected 6 of 17 terms, and 4 of 11 predictors (nprune=6)
## Termination condition: Reached nk 23
## Importance: cylinders4, year, weight, cylinders5, cylinders6-unused, ...
## Number of terms at each degree of interaction: 1 5 (additive model)
## Earth GCV 0.06315864    RSS 15.94387    GRSq 0.7491526    RSq 0.7671932
```

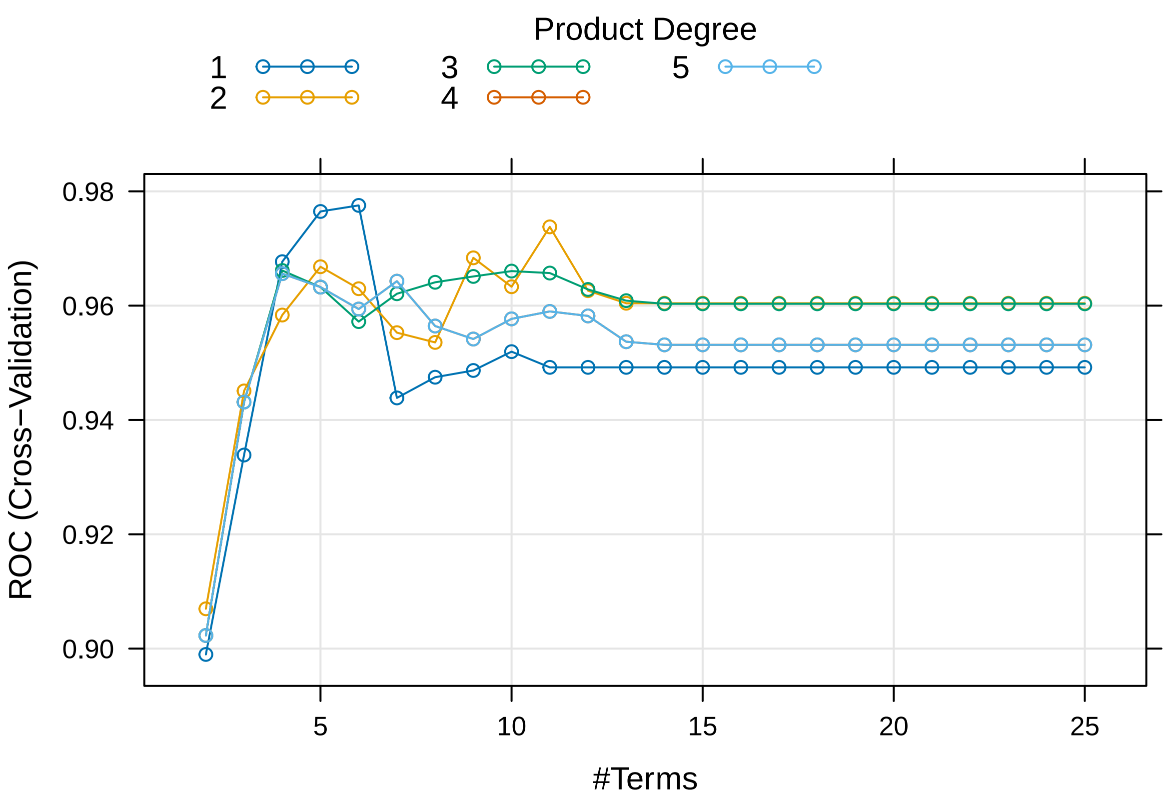
```r
vip(model.mars$finalModel, type = "nsubsets")
```




```r
vip(model.mars$finalModel, type = "rss")
```




```r
plot(model.mars)
```



```r
glmnet_pred <- predict(model.glmnet, newdata = test, type = "prob")[,2]
mars_pred <- predict(model.mars, newdata = test, type = "prob")[,2]
roc_glmnet <- roc(test$mpg_cat, glmnet_pred)
```

```
## Setting levels: control = high, case = low
```

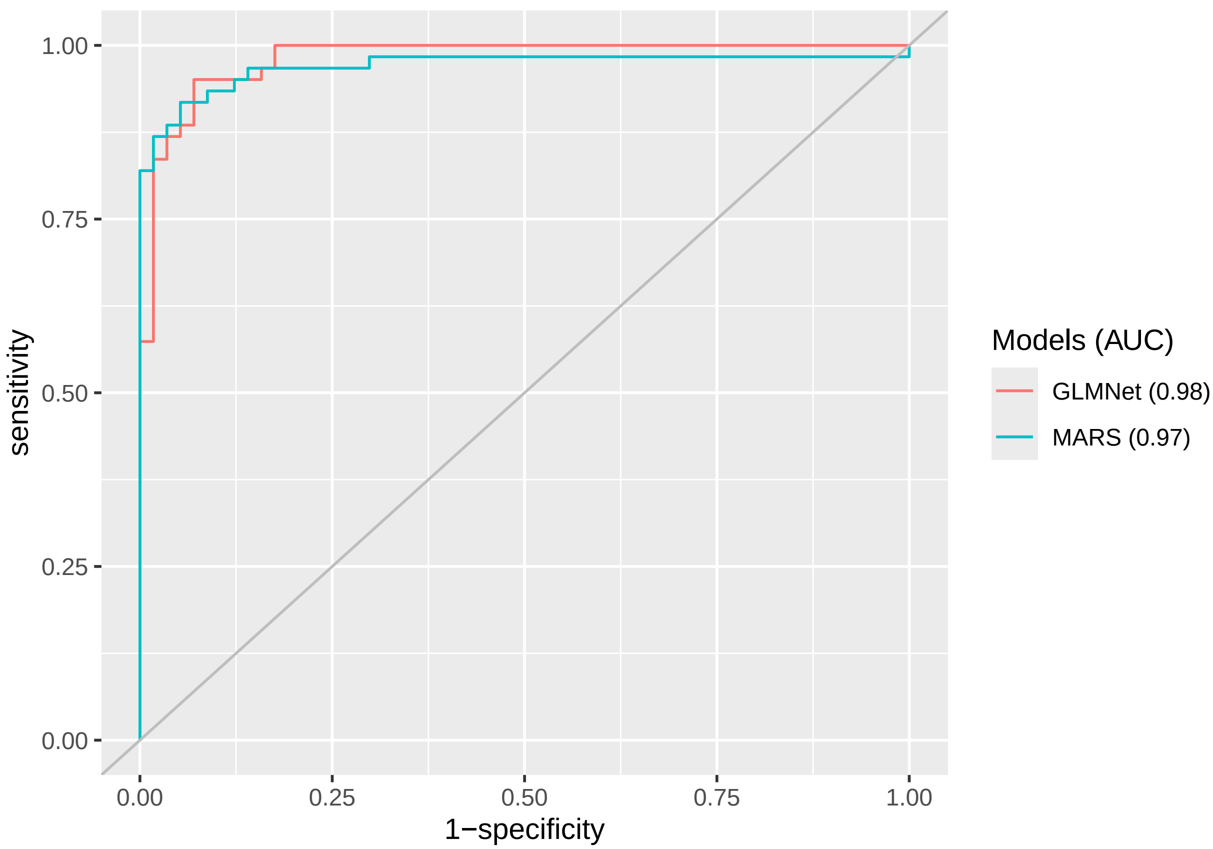
```
## Setting direction: controls < cases
```

```r
roc_mars <- roc(test$mpg_cat, mars_pred)
```

```
## Setting levels: control = high, case = low
## Setting direction: controls < cases
```

```r
# Compute AUC for both models
auc_glmnet <- auc(roc_glmnet)
auc_mars <- auc(roc_mars)

auc <- c(roc_glmnet$auc[1],
         roc_mars$auc[1])
modelNames <- c("GLMNet","MARS")
ggroc(list(roc_glmnet, roc_mars), legacy.axes = TRUE) +
  scale_color_discrete(labels = paste0(modelNames, " (", round(auc,3),")"),
                       name = "Models (AUC)") +
  geom_abline(intercept = 0, slope = 1, color = "grey")
```




```r
print(paste("AUC for glmnet model: ", auc_glmnet))
```

```
## [1] "AUC for glmnet model:  0.980442910555076"
```

```r
print(paste("AUC for MARS model: ", auc_mars))
```

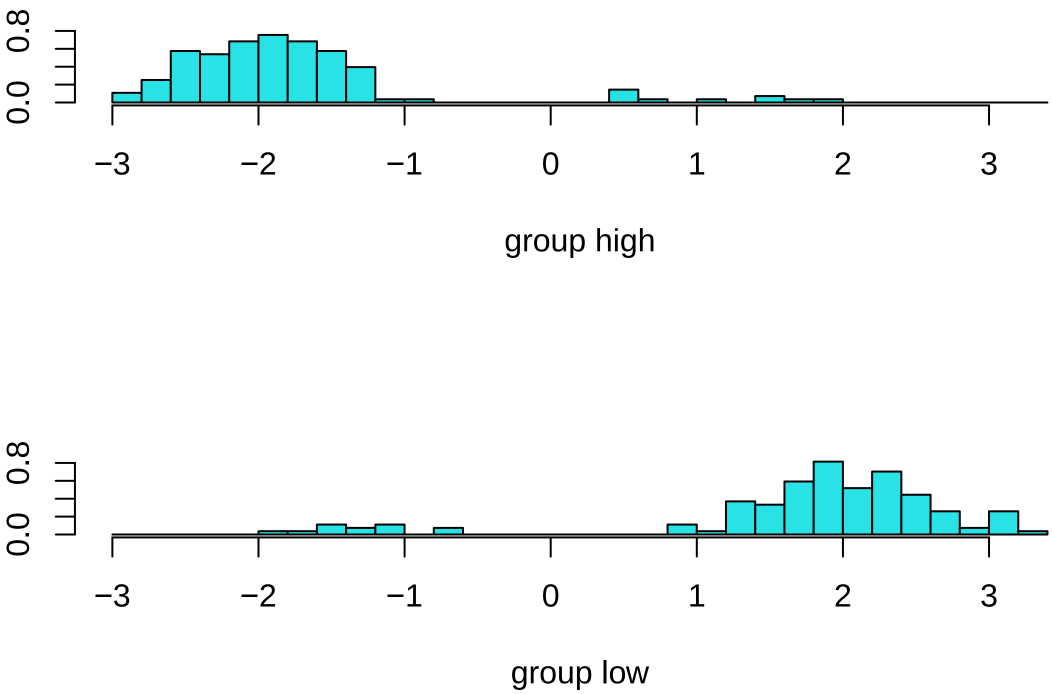
```
## [1] "AUC for MARS model:  0.969801553062986"
```

The best MARS model, has interaction terms order up to 1 and uses 6 basis function, does not contains complex relationship. `Origin` and some categories of `cylinders` are not used and `cylinders4` is the most influential(most frequent value group among the factor predictor) predictor in this model. The prediction performance by ROC AUC are higher for glmnet than MARS model, indicating no significant improvement on prediction when using MARS model.

C

```r
set.seed(1)
lda.fit <- lda(mpg_cat~., data = train)

plot(lda.fit)
```

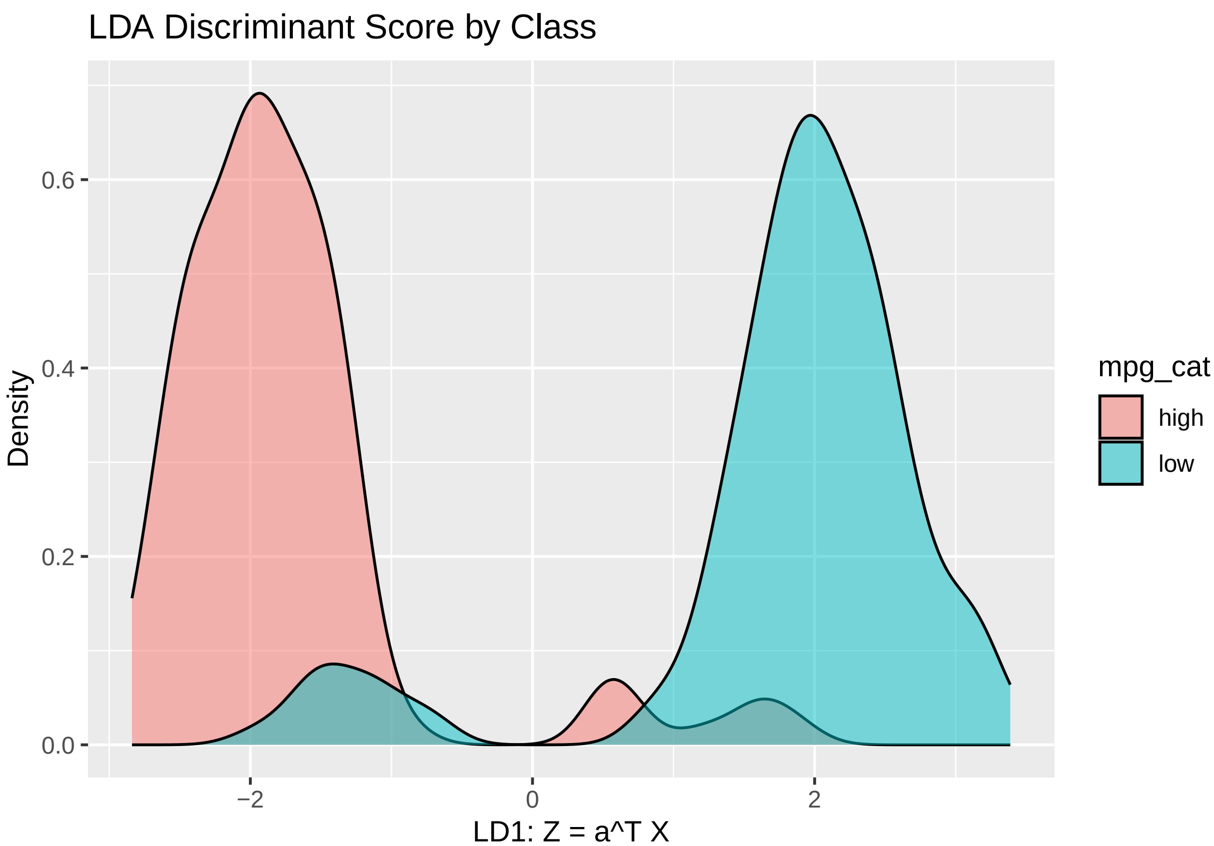



```r
lda.scaling <- lda.fit$scaling
lda.scaling
```

```
##                        LD1
## cylinders4   -4.2644777669
## cylinders5   -5.4495775938
## cylinders6   -1.7796780932
## cylinders8   -2.2810841546
## displacement  0.0033083314
## horsepower   -0.0009447846
## weight        0.0007064576
## acceleration  0.0482372868
## year         -0.0957892750
## origin2      -0.0111108562
## origin3      -0.1477883452
```

```r
train$Z <- predict(lda.fit, newdata = train[, 1:7])$x

ggplot(train, aes(x = Z, fill = mpg_cat)) +
  geom_density(alpha = 0.5) +
  labs(title = "LDA Discriminant Score by Class",
       x = "LD1: Z = a^T X",
       y = "Density")
```



The distribution of predicted response groups(high vs low mpg_cat) are overall symmetric centered at 0. There are some outliers within high mpg-cat group but they are neglegible.

D

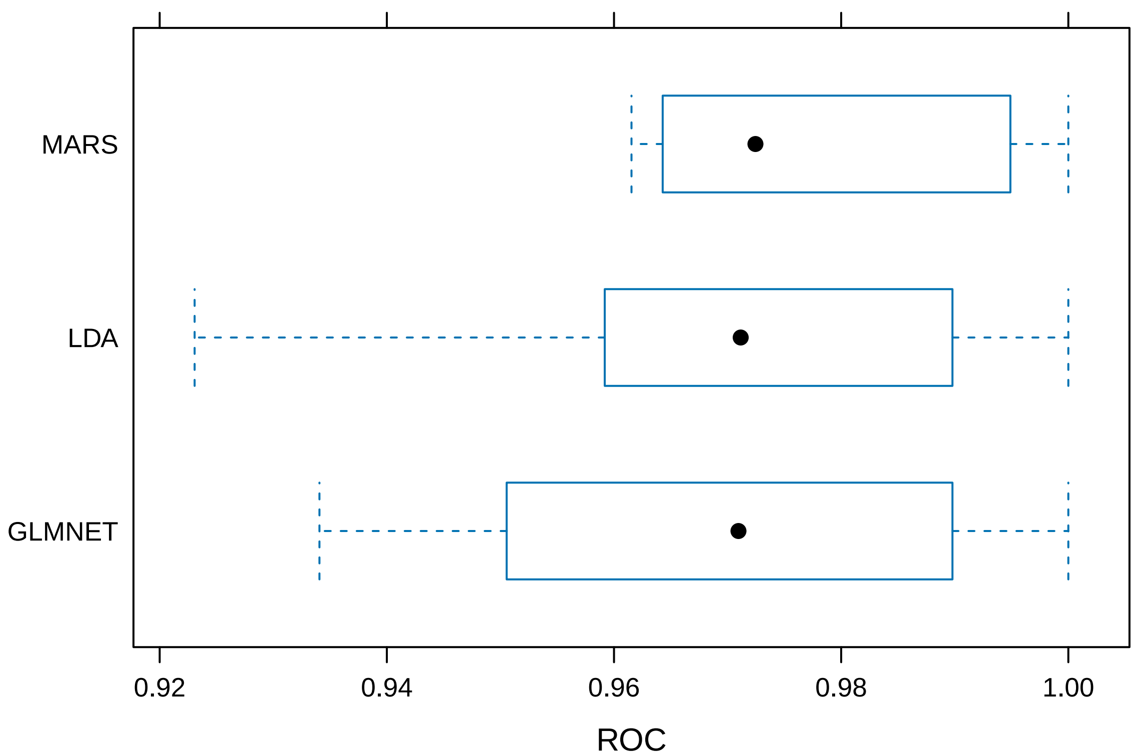
```r
set.seed(1)
model.lda <- train(mpg_cat ~ cylinders + displacement + horsepower + weight + acceleration + year + orig
                   data = train,
                   method = "lda",
                   metric = "ROC",
                   trControl = ctrl)

res <- resamples(list(GLMNET = model.glmnet,
                      MARS = model.mars,
                      LDA = model.lda))
summary(res)
```

```
##
## Call:
```

```
## summary.resamples(object = res)
##
## Models: GLMNET, MARS, LDA
## Number of resamples: 10
##
## ROC
##             Min.   1st Qu.    Median      Mean   3rd Qu. Max. NA's
## GLMNET 0.9340659 0.9527080 0.9709576 0.9684911 0.9872449    1    0
## MARS   0.9615385 0.9649725 0.9724520 0.9775359 0.9920526    1    0
## LDA    0.9230769 0.9597724 0.9711538 0.9710754 0.9897959    1    0
##
## Sens
##             Min.   1st Qu.    Median      Mean   3rd Qu. Max. NA's
## GLMNET 0.8461538 0.9285714 0.9285714 0.9346154 0.9821429    1    0
## MARS   0.9230769 0.9285714 0.9642857 0.9637363 1.0000000    1    0
## LDA    0.7857143 0.8750000 0.9285714 0.9274725 1.0000000    1    0
##
## Spec
##             Min.   1st Qu.    Median      Mean   3rd Qu. Max. NA's
## GLMNET 0.6923077 0.8461538 0.9285714 0.8873626 0.9285714    1    0
## MARS   0.8461538 0.9230769 0.9285714 0.9181319 0.9285714    1    0
## LDA    0.7857143 0.9230769 0.9230769 0.9109890 0.9285714    1    0
```

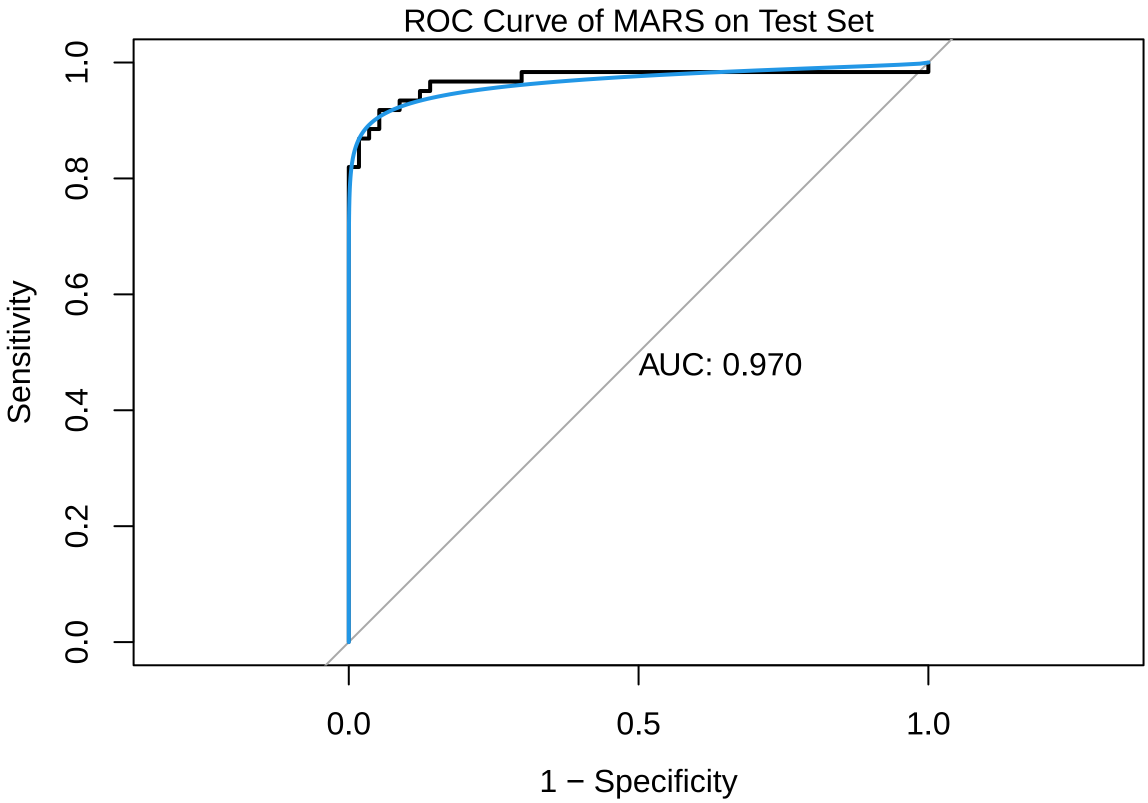
```r
bwplot(res, metric = "ROC")
```




```r
# modelNames <- c("GLMNet","MARS","LDA")
# ggroc(list(roc_glmnet, roc_mars,roc_lda), legacy.axes = TRUE) +
#   scale_color_discrete(labels = paste0(modelNames, " (", round(auc,3),")"),
#                        name = "Models (AUC)") +
#   geom_abline(intercept = 0, slope = 1, color = "grey")
```

According to resampling, MARS has the best overall performance (highest on avg ROC AUC, specificity) and well recognize the pattern of low mpg_cat response. Therefore, we select MARS model to compute the confusion matrix and further analysis.

```r
plot(roc_mars, legacy.axes = TRUE, print.auc = TRUE)
plot(smooth(roc_mars), col = 4, add = TRUE)
mtext("ROC Curve of MARS on Test Set", side = 3, line = 2,cex=1)
```




```r
glmn.class <- ifelse(mars_pred > 0.5, "low", "high")
glmn.class <- factor(glmn.class, levels = levels(test$mpg_cat))
confusionMatrix(glmn.class, test$mpg_cat)
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction high low
##       high   54   7
##       low     3  54
##
##                Accuracy : 0.9153
##                  95% CI : (0.8497, 0.9586)
##     No Information Rate : 0.5169
##     P-Value [Acc > NIR] : <2e-16
##
##                   Kappa : 0.8307
##
##  Mcnemar's Test P-Value : 0.3428
##
##             Sensitivity : 0.9474
```

```
##             Specificity : 0.8852
##          Pos Pred Value : 0.8852
##          Neg Pred Value : 0.9474
##              Prevalence : 0.4831
##          Detection Rate : 0.4576
##    Detection Prevalence : 0.5169
##       Balanced Accuracy : 0.9163
##
##        'Positive' Class : high
##
```

We select MARS model as the final model because of it has highest average ROC score based on the resampling results. It indicates that MARS was better at distinguishing between high and low mpg groups.

Then we plot MARS model's ROC curve and its AUC(=0.97) and confusion matrix metrics (we choose 0.5 as threshold according to observation from section C): MARS has relatively high accuracy in prediction(=91.53%) and reliable confidence interval; it also has 94.74% of actual high-mileage cars were correctly predicted as high and 88.52% of actual low-mileage cars were correctly predicted as low. Kappa Statistic(=0.8307) indicates strong agreement between the predicted and actual classifications. Overall, MARS is our best model.