ITEC 621 - Homework 4 - Non-Linear and Classification Models

Kogod School of Business

O’Malley, Conie

2025-03-05

Table of Contents

## Submission:

Download the **HW4\_YourLastName.Rmd** R Markdown file and save it with your own last name. Complete all your work in that template file, **Knit** the corresponding Word or PDF file. Your knitted file **MUST display your R commands**. In order to do this, please ensure that the knittr:: global option command above has echo=T.

Please prepare your R Markdown file with a **professional appearance**, as you would for top management or an important client. Please ensure that your text and R code are in the correct sections and use appropriate tags and formats.

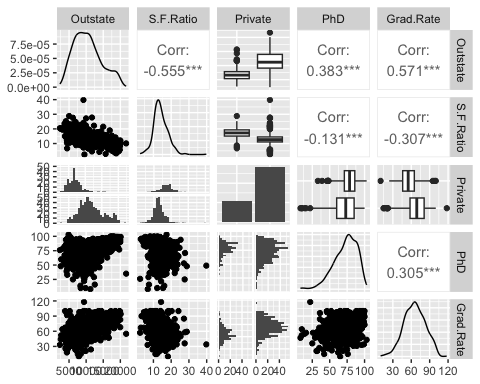
## Knitting

Improper or no knitting may have up to 10 pts. in deductions

## 1. (15 pts.) Inspecting for Non-Linearity

1.1 Load the **{ISLR}** and **{GGally}** libraries. Load and attach the **College{ISLR}** data set. Inspect the data with the **ggpairs(){GGally}** function, but do not run the ggpairs plots on all variables because it will take a **very long time**. Only include the following variables, as follows: College[,c("Outstate","S.F.Ratio","Private","PhD","Grad.Rate")].

library(ISLR)  
library(GGally)  
attach(ISLR::College)  
GGally::ggpairs(College[,c("Outstate","S.F.Ratio","Private","PhD","Grad.Rate")])



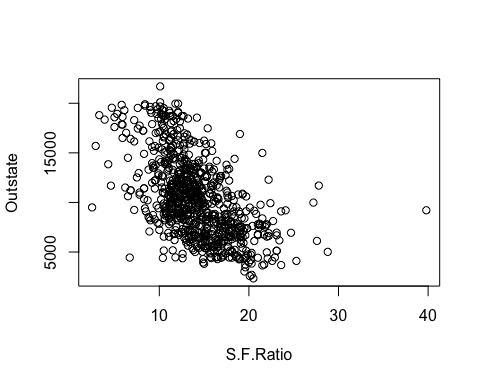
1.2 Briefly answer: if we are interested in predicting out of state tuition (**Outstate**), can you tell from the plots if any of the predictors have a curvilinear relationship with Outstate? Briefly explain.

#### Analysis

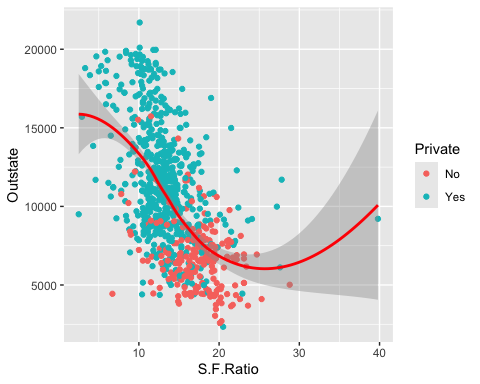
After looking at the correlation plots, the only variable that potentially has a curvilinear relationship with Outstate is PhD. Looking at the scatterplot relationship between the two variables, there appears to be a slightly curvilinear dispersion of the data points. However, this is not a strong indication of a curvilinear relationship and further analysis would be needed to confirm.

1.3 Regardless of your answer, let’s inspect **Outstate** closer by plotting it (Y axis) against **S.F.Ratio** (X axis) to examine if there is a linear or curvilinear association of these two variables. Do you now see a more curvilinear pattern in the relationship (no need to answer, just think)?

plot(S.F.Ratio, Outstate)



# Yes, this larger plot appears to show a non-linear relationship between these two variables.  
  
# Some students used ggplot(), which I thought was clever, there were several ways how the ggplot() was specified. This graph captures some of the best ideas I in student solutions:  
  
# Done for you  
  
library(ggplot2)  
ggplot(College, aes(y=Outstate, x=S.F.Ratio, color=Private)) + geom\_point() + geom\_smooth(color="red")



## 2. (15 pts.) Interaction Model

2.1 Let’s start with a **linear** model as a starting baseline. Fit a **linear model** to predict **Outstate** as a function of the other 4 predictors in your ggpairs plot, with the College data set. Store your model results in an object named **fit.linear**. Display a summary of your results.

fit.linear <- stats::lm(Outstate ~ S.F.Ratio + Private + PhD + Grad.Rate, data = College)  
  
summary(fit.linear)

##   
## Call:  
## stats::lm(formula = Outstate ~ S.F.Ratio + Private + PhD + Grad.Rate,   
## data = College)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -7920.9 -1627.9 -79.9 1572.4 13040.4   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 1240.515 736.104 1.685 0.0923 .   
## S.F.Ratio -244.339 25.963 -9.411 <2e-16 \*\*\*  
## PrivateYes 3653.040 243.210 15.020 <2e-16 \*\*\*  
## PhD 82.791 5.988 13.826 <2e-16 \*\*\*  
## Grad.Rate 60.658 5.896 10.288 <2e-16 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 2441 on 772 degrees of freedom  
## Multiple R-squared: 0.6337, Adjusted R-squared: 0.6318   
## F-statistic: 333.9 on 4 and 772 DF, p-value: < 2.2e-16

2.2 The model above has a good R-Squared and all predictors are significant, which is good. But suppose that your business knowledge about university enrollements makes you believe that **S.F.Ratio** and **Private** may have an interaction effect on **Outstate**. That is, your business knowledge tells you that higher student-to-faculty ratios lead to lower out of state tuition, but that this negative effect is stronger with private universites than public universities (i.e., you analytics goal is inference). To test this hypothesis, please add an **interaction** term for **S.F.Ratio** and **Private**. Store your model results in an object named **fit.inter**. Display a summary() of your results. Then do an anova() test to evaluate if **fit.inter** has more predictive power than **fit.linear**.

fit.inter <- stats::lm(Outstate ~ S.F.Ratio \* Private + PhD + Grad.Rate, data = College)  
  
summary(fit.inter)

##   
## Call:  
## stats::lm(formula = Outstate ~ S.F.Ratio \* Private + PhD + Grad.Rate,   
## data = College)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -8026.2 -1588.7 -72.2 1504.9 12879.2   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) -958.756 1007.011 -0.952 0.34135   
## S.F.Ratio -111.583 49.091 -2.273 0.02330 \*   
## PrivateYes 6565.770 947.547 6.929 8.94e-12 \*\*\*  
## PhD 82.521 5.954 13.861 < 2e-16 \*\*\*  
## Grad.Rate 59.671 5.870 10.165 < 2e-16 \*\*\*  
## S.F.Ratio:PrivateYes -181.125 56.972 -3.179 0.00154 \*\*   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 2427 on 771 degrees of freedom  
## Multiple R-squared: 0.6384, Adjusted R-squared: 0.6361   
## F-statistic: 272.3 on 5 and 771 DF, p-value: < 2.2e-16

stats::anova(fit.linear, fit.inter)

## Analysis of Variance Table  
##   
## Model 1: Outstate ~ S.F.Ratio + Private + PhD + Grad.Rate  
## Model 2: Outstate ~ S.F.Ratio \* Private + PhD + Grad.Rate  
## Res.Df RSS Df Sum of Sq F Pr(>F)   
## 1 772 4600356791   
## 2 771 4540828867 1 59527924 10.107 0.001536 \*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

2.3 Briefly interpret the **ANOVA results** and also the **coefficient** for the interaction term. Is the interaction effect hypothesis retained or rejected? Why.

#### Analysis

The fit.inter model is an improvement on the original fit.linear model as it passes the , indicating that the interaction term is statistically significant and should be retained in the model. The also decreases with the addition of the interaction term. The coefficient is negative, meaning the interaction effect is negative - which offsets PrivateYes, but reinforces S.F.Ratio, since their coefficients are positive and negative, respectively.

## 3. (15 pts.) Polynomial Model

3.1 Using the poly() function, fit a polynomial of degree 4 for **S.F.Ratio**. In order to obtain **raw** (rather than orthogonal) coefficients, please include the raw=T attribute inside of the poly() function. Also, include linear predictors for the other 3 variables **Private**, **PhD** and **Grad.Rate**. Store your model results in an object named **fit.poly**. Display the summary results. Then conduct an **anova()** test to evaluate if **fit.poly** has more predictive power than **fit.linear**.

fit.poly <- stats::lm(Outstate ~ stats::poly(S.F.Ratio, degree = 4, raw = TRUE) + Private + PhD + Grad.Rate, data = College)  
  
summary(fit.poly)

##   
## Call:  
## stats::lm(formula = Outstate ~ stats::poly(S.F.Ratio, degree = 4,   
## raw = TRUE) + Private + PhD + Grad.Rate, data = College)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -8271.7 -1508.4 -18.3 1555.2 12598.0   
##   
## Coefficients:  
## Estimate Std. Error t value  
## (Intercept) -1.293e+03 2.127e+03 -0.608  
## stats::poly(S.F.Ratio, degree = 4, raw = TRUE)1 1.052e+03 5.516e+02 1.907  
## stats::poly(S.F.Ratio, degree = 4, raw = TRUE)2 -1.507e+02 5.160e+01 -2.920  
## stats::poly(S.F.Ratio, degree = 4, raw = TRUE)3 6.125e+00 1.946e+00 3.148  
## stats::poly(S.F.Ratio, degree = 4, raw = TRUE)4 -7.517e-02 2.462e-02 -3.053  
## PrivateYes 3.438e+03 2.469e+02 13.923  
## PhD 8.181e+01 5.858e+00 13.966  
## Grad.Rate 6.017e+01 5.782e+00 10.406  
## Pr(>|t|)   
## (Intercept) 0.54334   
## stats::poly(S.F.Ratio, degree = 4, raw = TRUE)1 0.05687 .   
## stats::poly(S.F.Ratio, degree = 4, raw = TRUE)2 0.00361 \*\*   
## stats::poly(S.F.Ratio, degree = 4, raw = TRUE)3 0.00171 \*\*   
## stats::poly(S.F.Ratio, degree = 4, raw = TRUE)4 0.00234 \*\*   
## PrivateYes < 2e-16 \*\*\*  
## PhD < 2e-16 \*\*\*  
## Grad.Rate < 2e-16 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 2386 on 769 degrees of freedom  
## Multiple R-squared: 0.6513, Adjusted R-squared: 0.6481   
## F-statistic: 205.2 on 7 and 769 DF, p-value: < 2.2e-16

anova(fit.linear, fit.poly)

## Analysis of Variance Table  
##   
## Model 1: Outstate ~ S.F.Ratio + Private + PhD + Grad.Rate  
## Model 2: Outstate ~ stats::poly(S.F.Ratio, degree = 4, raw = TRUE) + Private +   
## PhD + Grad.Rate  
## Res.Df RSS Df Sum of Sq F Pr(>F)   
## 1 772 4600356791   
## 2 769 4379475120 3 220881671 12.928 3.022e-08 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

3.2 Briefly interpret your results (done for you in the R code). Comment on both, the summary() output of the model and the anova() test results.

#### Analysis

The orthogonalized terms for S.F.Ratio all are statistically significant to the model - . There is also an improvement in from .6361 to .6481. The fit.poly model also passes the , indicating that the orthoganalized terms are statistically significant and should be retained in the model.

## 4. (15 pts.) Piecewise Linear Model

4.1 Given that the above polynomial model has more predictive power than the linear model, it is worth exploring a **piecewise Linear Regression** model with **2 knots** at **S.F.Ratio=10** and **S.F.Ratio=20**, to predict **out of state tuition** as a function of **S.F.Ratio** (only) and save the model in an object named **fit.piece** (Note: your piecewise linear model will have 3 separate straight lines changing at each of these 2 points.) Display the summary results.

# Note: the complete model should also include Private, PhD and Grad.Rate as predictors, but we exclude them below for the visualization.  
  
fit.piece=lm(Outstate~S.F.Ratio+  
 I((S.F.Ratio-10)\*(S.F.Ratio>10))+  
 I((S.F.Ratio-20)\*(S.F.Ratio>20)),  
 data=College)  
  
summary(fit.piece)

##   
## Call:  
## lm(formula = Outstate ~ S.F.Ratio + I((S.F.Ratio - 10) \* (S.F.Ratio >   
## 10)) + I((S.F.Ratio - 20) \* (S.F.Ratio > 20)), data = College)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -10355.8 -2433.4 -134.7 2210.6 10136.0   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 17822.5 1323.8 13.464 < 2e-16 \*\*\*  
## S.F.Ratio -451.7 139.8 -3.231 0.00128 \*\*   
## I((S.F.Ratio - 10) \* (S.F.Ratio > 10)) -275.0 159.1 -1.729 0.08426 .   
## I((S.F.Ratio - 20) \* (S.F.Ratio > 20)) 942.6 138.1 6.828 1.74e-11 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 3257 on 773 degrees of freedom  
## Multiple R-squared: 0.3473, Adjusted R-squared: 0.3448   
## F-statistic: 137.1 on 3 and 773 DF, p-value: < 2.2e-16

4.2 Briefly interpret your results

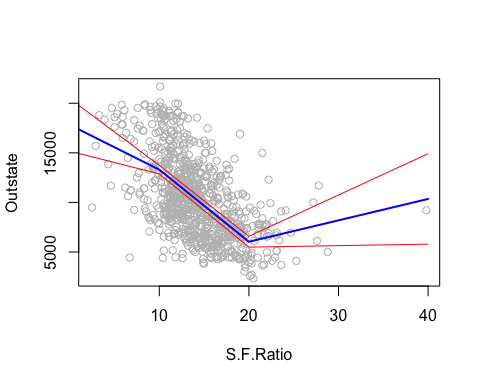
#### Analysis

The second line segment (I((S.F.Ratio - 10) \* (S.F.Ratio > 10))) is not statistically significant to the model in this case, and the model does not have very good predictive power - .

4.3 Now **Visualize**

**Plotting Results.** Create a vector called **S.F.Ratio.knots** with 4 values: **0, 10, 20 and 40**. Note: we will use these knots only to draw regression lines (the model is already fitted) – i.e., 2 of the knots correspond to the model knots; 1 is at 0 (the intercept) and the other one is at 40 (about the end of the data range). Then, use the **predict()** function to make predictions at these 4 knots.Store the predictions in an object named **preds**. Then do a **plot** of **S.F.Ratio** (X axis) vs. **Outstate** (Y axis). Use the **col=“gray”** attribute to get somewhat faded dots. Then use the **“lines()”** function to draw the regression lines connecting the 4 knots above (use thicker blue), along with the respective 95% confidence interval lines (use thinner red) (Tip: use +/- 2\*preds$se.fit).

# Done for you, but I encourage you to try to do this yourself  
  
S.F.Ratio.knots=c(0,10,20,40)  
  
preds=predict(fit.piece,list(S.F.Ratio=S.F.Ratio.knots), se=T)  
plot(S.F.Ratio, Outstate, col="gray")  
  
lines(S.F.Ratio.knots,preds$fit,col="blue",lwd=2)  
lines(S.F.Ratio.knots,preds$fit-2\*preds$se.fit,col="red",lwd=1)  
lines(S.F.Ratio.knots,preds$fit+2\*preds$se.fit,col="red",lwd=1)



## 5. (20 pts.) Binomial Classification Model

5.1 Data Preparation

We will use the dataset **Auto{ISLR}** to develop a binomial classification model to predict the likelihood of automobiles having high gas mileage. So, first load the **{ISLR}** library. Since we don’t have a dummy variable to classify high vs. low gas mileage vehicles, let’s use the quantitative value of miles per gallon **mpg** to create a binary variable called **mpg.hi** if a vehicle has higher **mpg** than the median. Let’s first calculate the **median mpg** value using the median() function and store the results in an object named **med.mpg**. Then create a column in the **Auto** dataset named **mpg.hi** with a binary value of 1 if mpg>med.mpg and 0 otherwise, using the ifelse() function.

For a quick visual inspection, display the **med.mpg** value and then a 2-column data frame using cbind() the first 20 values for **mpg** and **mpg.hi**. Please label the columns as shown below. Don’t answer this,but quickly verify that your **med.hi** variable was created correctly.

attach(ISLR::Auto)  
med.mpg <- stats::median(Auto$mpg)  
Auto$mpg.hi <- ifelse(mpg > med.mpg, 1, 0)  
cbind(med.mpg, mpg.hi = Auto$mpg.hi[1:20], mpg = Auto$mpg[1:20])

## med.mpg mpg.hi mpg  
## [1,] 22.75 0 18  
## [2,] 22.75 0 15  
## [3,] 22.75 0 18  
## [4,] 22.75 0 16  
## [5,] 22.75 0 17  
## [6,] 22.75 0 15  
## [7,] 22.75 0 14  
## [8,] 22.75 0 14  
## [9,] 22.75 0 14  
## [10,] 22.75 0 15  
## [11,] 22.75 0 15  
## [12,] 22.75 0 14  
## [13,] 22.75 0 15  
## [14,] 22.75 0 14  
## [15,] 22.75 1 24  
## [16,] 22.75 0 22  
## [17,] 22.75 0 18  
## [18,] 22.75 0 21  
## [19,] 22.75 1 27  
## [20,] 22.75 1 26

5.2 Train and Test Sub-Samples

Set the **seed** to 1 to get replicable results. Then draw a random sample of index numbers into a vector called **train** containing **80%** of the row numbers in **Auto**. This vector will contain a random sample of 80% of the nrow(Auto) row numbers, which you will use to index the training sub-sample. **Note:** if your resulst deviate **slightly** from mine it is probably due to random sampling and it is not a problem. Display your resulting **train** index vector so that you can compare your results with mine.

set.seed(1)  
train <- sample(nrow(Auto), size = .8\*nrow(Auto))  
print(train)

## [1] 324 167 129 299 270 187 307 85 277 362 330 263 329 79 213 37 105 217  
## [19] 366 165 290 383 89 289 340 326 382 42 111 20 44 343 70 121 40 172  
## [37] 25 248 198 39 298 280 160 14 130 45 22 206 230 193 104 367 255 341  
## [55] 342 103 331 13 296 375 176 279 110 84 29 141 252 221 108 304 33 347  
## [73] 149 287 102 145 118 323 107 64 224 337 51 325 372 138 390 389 282 143  
## [91] 285 170 48 204 295 24 181 214 225 163 43 1 328 78 284 116 233 61  
## [109] 86 374 49 242 246 247 239 219 135 364 363 310 53 348 65 376 124 77  
## [127] 218 98 194 19 31 174 237 75 16 358 9 50 92 122 152 386 207 244  
## [145] 229 350 355 391 223 373 309 140 126 349 344 319 258 15 271 388 195 201  
## [163] 318 17 212 127 133 41 384 392 159 117 72 36 315 294 157 378 313 306  
## [181] 272 106 185 88 281 228 238 368 80 30 93 234 220 240 369 164 168 243  
## [199] 200 184 260 100 113 359 73 27 333 235 38 62 134 132 35 125 99 267  
## [217] 269 71 153 262 377 28 183 148 308 227 365 60 171 354 173 12 202 305  
## [235] 371 265 26 322 334 208 288 297 357 249 210 278 82 97 264 250 56 216  
## [253] 101 336 259 2 192 131 275 169 292 291 317 83 90 23 68 222 190 91  
## [271] 57 311 345 52 179 34 139 109 180 74 360 232 353 266 231 87 301 155  
## [289] 114 11 335 205 182 128 178 257 261 199 142 356 361 81 209 188 321 370  
## [307] 251 346 3 273 381 332 119

5.3 Logistic Model

Fit a **Logistic** model on the **train** sub-sample, to predict the likelihood that a vehicle has **high gas mileage** (use **mpg.hi** as the response variable), using **cylinders, displacement, horsepower, weight, year** and **origin** as predictors. Store the results in an object named **fit.logit**. Display the summary results.

fit.logit <- stats::glm(mpg.hi ~ cylinders + displacement + horsepower + weight + year + origin, data = Auto[train, ], family = "binomial")  
  
summary(fit.logit)

##   
## Call:  
## stats::glm(formula = mpg.hi ~ cylinders + displacement + horsepower +   
## weight + year + origin, family = "binomial", data = Auto[train,   
## ])  
##   
## Coefficients:  
## Estimate Std. Error z value Pr(>|z|)   
## (Intercept) -16.194199 5.272918 -3.071 0.00213 \*\*   
## cylinders -0.142125 0.441053 -0.322 0.74727   
## displacement 0.006085 0.012687 0.480 0.63153   
## horsepower -0.038895 0.018424 -2.111 0.03477 \*   
## weight -0.004703 0.001125 -4.180 2.92e-05 \*\*\*  
## year 0.424170 0.083221 5.097 3.45e-07 \*\*\*  
## origin 0.411067 0.385415 1.067 0.28617   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## (Dispersion parameter for binomial family taken to be 1)  
##   
## Null deviance: 433.83 on 312 degrees of freedom  
## Residual deviance: 135.33 on 306 degrees of freedom  
## AIC: 149.33  
##   
## Number of Fisher Scoring iterations: 7

5.6 Briefly interpret the effect and significance of all coefficients with p<0.05. Use the units of the respective variables in your interpretation narrative (enter ?Auto from the console)

#### Analysis

* horsepower: - for every one unit increase in horsepower, there is a 0.038895 decrease in the log odds of mpg.hi being greater than the median (med.mpg)
* weight: - for every one unit increase in weight, there is a 0.004703 decrease in the log odds of mpg.hi being greater than the median (med.mpg)
* year: - for every one unit increase in year (meaning the newer the vehicle is), there is a 0.424170 increase in the log odds of mpg.hi being greater than the median (med.mpg)

5.7 As you know, the coefficients of this logit model represent the variable effects on the **log-odds** of having a high mileage car, so the interpretation of the log-odds is hard to interpret and explain to a manager. To make this easier to interpret, create 2 vectors: **log.odds** containing the coefficients from **fit.logit** and **odds** containing the equivalent odds of these coefficients. Before you print the results, enter the function options(scipen=4) to minimize the display of scientific notation. Then use the **cbind()** function to diaplay both vectors together.

options(scipen=4)  
log.odds <- stats::coef(fit.logit)  
odds <- exp(log.odds)  
cbind(log.odds, odds)

## log.odds odds  
## (Intercept) -16.194199158 0.00000009267203  
## cylinders -0.142124961 0.86751284422180  
## displacement 0.006084510 1.00610305797550  
## horsepower -0.038895131 0.96185157238634  
## weight -0.004703228 0.99530781475532  
## year 0.424170100 1.52832153884322  
## origin 0.411066915 1.50842628925795

## 6. (20 Pts.) Confusion Matrix and ROC Curve

6.1 Use the **predict()** function with the attribute **type=“response”** (which gives probabilities) to make predictions on the test data (tip: [-train,]) and save the results in an object named **pred.probs**. Then create a vector named **pred.hi.mpg** containing a 1 if the probability of the vehicle being high mileage is greater than 50% and 0 otherwise (tip: use this function **ifelse(pred.probs>0.5, 1, 0)**). Then create and display a **confusion matrix** with these test predictions using the **table()** function (tip: use table(pred.hi.mpg, Auto$mpg.hi[-train]). Name the confusion matrix **conf.mat**.

pred.probs <- stats::predict(fit.logit, Auto[-train,], type = "response")  
  
pred.hi.mpg <- ifelse(pred.probs > 0.5, 1, 0)  
  
conf.mat <- table(pred.hi.mpg, Auto$mpg.hi[-train])  
  
conf.mat

##   
## pred.hi.mpg 0 1  
## 0 38 2  
## 1 4 35

6.2 Cross-Validation Fit Statistics

Now use the data in the confusion matrix to compute the accuracy rate, error rate, sensitivity, specificity and false positives rate. Follow the formulas in the classification models script.

**Technical tip:** In the classification models script I provided for the course, I have formulas to compute sensitivity, specificity, etc. If you name the confusion matrix **conf.mat()** you can simply copy paste the code from the script (e.g., TruN, TruP, etc.) and it will work here. Once you are done with the calculations, use the c() function to display your results in a single row, with each value properly labeled as shown below.

TruN <- conf.mat[1,1] # true negatives  
TruP <- conf.mat[2,2] # true positives  
FalP <- conf.mat[2,1] # false positives  
FalN <- conf.mat[1,2] # false negatives  
TotN <- conf.mat[1,1] + conf.mat[2,1] # total negatives  
TotP <- conf.mat[1,2] + conf.mat[2,2] # total positives  
Tot <- TotN+TotP # matrix total  
  
Accuracy <- (TruN+TruP)/Tot  
Error <- (FalN+FalP)/Tot  
Sensitivity <- TruP/TotP # correct positives  
Specificity <- TruN/TotN # correct negatives  
FalPos <- 1 - Specificity  
  
cbind(Accuracy, Error, Sensitivity, Specificity, FalPos)

## Accuracy Error Sensitivity Specificity FalPos  
## [1,] 0.9240506 0.07594937 0.9459459 0.9047619 0.0952381

6.3 Interprtation

Provide a brief interpretation of each of the results displayed.

#### Analysis

* Accuracy Rate (0.9240): our model correctly predicted 92.4% of all predicted values (true positives and true negatives)
* Error Rate (0.0759): our model incorrectly predicted 7.59% of all predicted values (false positives and false negatives)
* Sensitivity (0.9459): our model identified 94.59% of the total actual positives
* Specificity (0.9047): our model identified 90.47% of the total actual negatives
* False Positive Rate (0.0952): our model incorrectly predicted positive 9.52% of the time

6.4 Classification Threshold

In the formulas above, we used a classification threshold of pred.probs>0.5 to predict a vehicle as high mileage (i.e., 1). But as you know, we can change this threshold and change the fit statistics to suit our needs. When you change this classification threshold, the sensitivy and specificity will also change but in oposite directions. Briefly describe how the sensitivity and specificity will change if we change the classification threshold to 0.3 and 0.7.

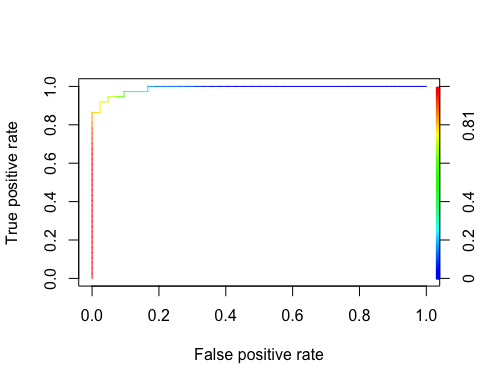
6.5 ROC and AUC

To better examine how the model performs as we change the classification threshold, let’s plot the ROC Curve and compute the corresponding area under the curve (AUC). First, load the **{ROCR}** library, which contains the necessary functions for ROC computation. You already predicted high mileage classifications with the test subsample above and stored it in **pred.probs**. But this time we need to use the **ROCR** prediction() function to create an ROC prediction object. Use pred.probs (i.e., predicted classifications in the test subsample) and Auto$mpg.hi[-train] (actual classifications in the test subsample) as arguments in the prediction() function and store the results in an object named **pred**.

Then feed this **pred** object into the perf() function, but also include the "tpr" and "fpr" arguments to obtain true positive and false positive rates, respectively (which is what goes on the ROC Curve). Then plot the **perf** object using colorize=T to get a colorful plot.

Then feed the **pred** object into the perf() function again, but this time use "auc" as the argument, to get the area under the ROC Curve, AUC. Then display the name and value of the AUC with c(auc@y.name[[1]], auc@y.values[[1]])

library(ROCR)  
pred <- ROCR::prediction(pred.probs, Auto$mpg.hi[-train])  
  
perf <- ROCR::performance(pred, "tpr", "fpr")  
plot(perf, colorize=T)



auc <- ROCR::performance(pred, "auc")  
c(auc@y.name[[1]], auc@y.values[[1]])

## [1] "Area under the ROC curve" "0.99034749034749"

6.6 Interpretation

Provide a brief interpretation of the ROC Curve and AUC. Is this a good model? Why or why not?

#### Analysis

* ROC plot: the elbow is very high towards the upper left corner of the plot, meaning our model is doing a very good job of capturing true positives in relationship to the false positives.
* : this confirms the findings of the ROC plot - the true positive : false positive rate is 99.03% meaning we are highly accurate at predicting true positives