STAT 423/504 - Homework 4

Due date: Wednesday, March 11th. Please submit your homework to the 423/504 mailbox at Padelford B-313 on Wednesday by 5:00 PM on the designated day. Please submit the output and plots, but not your R code unless the quesiton specifically asks for it. Total possible points: 28.

1. (7 points) **Logistic Regression for Binary Data** A car manufacturer instructed a market research company to analyze which families are going to buy a new car next year using a logistic regession model. The data stems from a random sample of 33 families from an agglomeration area. Assessed variables cover the yearly household income (in 1000 US \$) and the age of the oldest car in the family (in years). 12 months later, interviewers assessed which families had bought a new car in the meantime. The data is available in the file `car.RDS` on Canvas.
   (a) (1 point) Perform a logistic regression and report the fitted regression equation.
   (b) (2 points) Estimate $\exp(\hat{\beta}_{income})$ and $\exp(\hat{\beta}_{age})$ and give an interpretation of these estimates.
   (c) (1 point) How large is the estimated probability that a family with a yearly household income of 50 000 US \$ and whose oldest car is 3 years old will buy a new car?
   (d) (1 point) Check for the presence of points with a large Cook's distance.
   (e) (1 point) Is the predictor `age` significant at the 5% level?
   (f) (1 point) Is there a non-negligible interaction between `income` and `age`?

2. (7 points) **Logistic Regression for Binomial Data** In this task we analyze an example concerning hypertension. First, we need to enter the data. This is done as follows:

```
> no.yes <- c("No", "Yes")
> smoking <- gl(2,1,7, no.yes)
> obesity <- gl(2,2,7, no.yes)
> snoring <- gl(2,4,7, no.yes)
> n.total <- c(60, 17, 8, 187, 85, 51, 23)
> n.hyper <- c(5, 2, 1, 35, 13, 15, 8)
```

   Here, the function `gl` creates a factor variable with the given levels. The factors `smoking`, `obesity` and `snoring` have an obvious meaning, `n.total` is the number of observations and `n.hyper` is the number of people with hypertension in each group.
   (a) (1 point) In order to fit a binomial logistic regression model construct a response matrix with two columns containing the number of people with and without hypertension, respectively.
   (b) (1 point) Fit a binomial regression model to the data. Assess the goodness-of-fit via the chi-square test for the residual deviance.
   (c) (2 points) Which variables in the model are significant at the 5% level? Use the likelihood-ratio test to obtain the answer. Hint: `drop1` function in R.
   (d) (2 points) Find a suitable sub-model compared to the model above using likelihood-ratio tests and backward elimination based on p-values. What is the model that you would choose?

(e) (1 point) Compare the observed and fitted proportions for hypertension using the model you found in d). Additionally, compare the fitted and observed counts of hypertension in each group. Note that the fitted count is not always a whole number.

3. (4 points) In this exercise we are investigating the ozone dataset. This dataset ozone is available in numerous R-packages, e.g. in the package gss. You can load it with data(ozone, package = ”gss”). A short description of the variables is available at help(ozone, package = ”gss”).

   (a) (2 points) Load the data. Apply a log transformation on the response `upo3` and remove the outlier (observation number 92).

   The following code generate a design matrix for fitting a cubic penalized regression model that accounts for all 3-way interactions.

```
require(sfsmisc)
ff <- wrapFormula(logupo3~., data=d.ozone.e, wrapString="poly(*,degree=3)")
ff <- update(ff, logupo3 ~ .^3)
mm <- model.matrix(ff, data=d.ozone.e)
```

   Fit a cubic penalized regression model that accounts for all 3-way interactions to the data. Use ridge, lasso and elastic net regression for the regularization problem. Plot the ridge, lasso and elastic net traces. How do they differ?

   R-Hints:

   To perform penalized regression via ridge and lasso use the `glmnet` function in the package of the same name.

```
require(glmnet)
ridge <- glmnet(mm, ?, alpha=?)
lasso <- glmnet(mm, ?, alpha=?)
elnet <- glmnet(mm, ?, alpha=.5)

## For plotting the traces:
plot(?, xvar="lambda")
```

   (b) (2 points) Select an optimal tuning parameter λ with an elastic net penalty $\alpha = 0.5$ via 10-fold cross validation. Find an optimal λ according to the ”1-std error rule” from a plot that shows the mean squared error as a function of $\log(\lambda)$.

   R-Hints:

   To perform cross validation for the elastic net use the `cv.glmnet` function

```
set.seed(1)
cv.eln <- cv.glmnet(?,?,alpha=?, nfolds=?)
plot(cv.eln)
```

4. (10 points) The file `CustomerWinBack.rda` on Canvas provides a dataframe called `cwb`. It contains information about how long could a company hold customers that cancelled the contract at some point in the past and re-opened their contracts afterwards. There are 295 observations of the following variables:

`duration`	target variable, duration of the customer relationship in days
`offer`	value of the present offered at re-acquisition
`lapse`	time until the customer could be re-acquired
`price`	offered price change in comparison to the first contract
`gender`	gender. 0 = female and 1 = male
`age`	age of the customer

The goal is to find a good model for the duration of the new customer relationship.

   (a) (2 points) First fit an OLS with all variables and perform a residual analysis. Hint: Check whether all varialbes are encoded properly (see as.factor).

(b) (1 point) Choose a model using stepwise model selection (forward-backward) starting from the model given in part a) and the AIC criterion. What predictors are included in the optimal model according to the above selection?

(c) (1 point) Choose a model using stepwise model selection (forward-backward) starting from the model given in part a) and the BIC criterion. What predictors are included in the optimal model according to the above selection?

(d) (2 points) Use the following to generate a design matrix for fitting a penalized regression:

```
library(glmnet)
## Lasso does not work with factor variables
xx        <- model.matrix(duration~0+., cwb)[,-4]
yy        <- cwb$duration
```

Now, fit a ridge regression with optimized λ. You can use `cv.glmnet` to fit the regrerssion with optimized λ.

What is the optimal lambda (see lambda.1se in R)? What predictors are included in this model? What is the fitted ridge equation?

(e) (2 points) Fit a lasso regression with optimized λ.

What is the optimal lambda (see lambda.1se in R)? What predictors are included in this model? What is the fitted lasso equation?

(f) (2 points) Finally, use a 5-fold cross validation to compare the predictive performance of all of the models in this task. What are the best and worst performing models?

For the cross validation, we need the following code:

```
## cross validation preparation
pre.ols   <- c()
pre.aic   <- c()
pre.bic   <- c()
pre.rr <- c()
pre.las <- c()
folds      <- 5
sb         <- round(seq(0,nrow(cwb),length=(folds+1)))
```

First, we define an object for each method that will serve to save the respective predictions. To perform 5-fold cross validation, the object `sb` contains the split points for the samples.

You need to complete the following code - parts written as ??? - using what you have learned above.

```
## cross validation Loop
for (i in 1:folds)
{
  ## define training and test datasets
  test    <- (sb[((folds+1)-i)]+1):(sb[((folds+2)-i)])
  train   <- (1:nrow(cwb))[-test]

  ## fit models
  fit.ols <- lm(duration ~ ., data=cwb[train,])
  fit.aic <- lm(duration ~ ???, data=cwb[train,])
  fit.bic <- lm(duration ~ ???, data=cwb[train,])

  xx      <- model.matrix(duration~0+., cwb[train,])[,-4]
  yy      <- cwb$duration[train]

  fit.rr  <- glmnet(xx,yy, lambda = ???, alpha = ???)
  fit.las <- glmnet(xx,yy, lambda = ???, alpha = ???)

  ## create predictions
  pre.ols[test] <- predict(fit.ols, newdata=cwb[test,])
  pre.aic[test] <- predict(fit.aic, newdata=cwb[test,])
  pre.bic[test] <- predict(fit.bic, newdata=cwb[test,])
  pre.rr[test]  <- model.matrix(duration~., cwb[test,])%*%as.numeric(coef(fit.rr))
  pre.las[test] <- model.matrix(duration~., cwb[test,])%*%as.numeric(coef(fit.las))
}

## Finally, compute the mean squared prediction error:

mean((cwb$duration-pre.ols)^2)
mean((cwb$duration-pre.aic)^2)
mean((cwb$duration-pre.bic)^2)
mean((cwb$duration-pre.rr)^2)
mean((cwb$duration-pre.las)^2)
```