Homework 2

PSTAT 131/231, Fall 2021

Due on Wednesday October 27, 2021 at 23:59 pm

Following packages are needed below:

```
library(tidyverse)
library(ISLR)
library(ROCR)
```

Linear regression (12 pts)

In this problem, we will make use of the *Auto* data set, which is part of the *ISLR* package and can be directly accessed by the name **Auto** once the *ISLR* package is loaded. The dataset contains 9 variables of 392 observations of automobiles. The qualitative variable **origin** takes three values: 1, 2, and 3, where 1 stands for American car, 2 stands for European car, and 3 stands for Japanese car.

1. (2 pts) Fit a linear model to the data, in order to predict **mpg** using all of the other predictors except for **name**. Present the estimated coefficients. (2 pts) With a 0.01 threshold, comment on whether you can reject the null hypothesis that there is no linear association between **mpg** with **any** of the predictors.
2. (2 pts) Take the whole dataset as training set. What is the training mean squared error of this model?
3. (2 pts) What gas mileage do you predict for an European car with 4 cylinders, displacement 122, horsepower of 105, weight of 3100, acceleration of 32, built in the year 1991? (Be sure to check how **year** is coded in the dataset).
4. (1 pts) On average, holding all other covariates fixed, what is the difference between the mpg of a Japanese car and the mpg of an American car? (1 pts) What is the difference between the mpg of a European car and the mpg of an Ameircan car?
5. (2 pts) On average, holding all other predictor variables fixed, what is the change in mpg associated with a 10-unit increase in displacement?

Algae Classification using Logistic regression (15 pts)

Get the dataset `algaeBloom.txt` from the homework archive file, and read it with the following code:

```
algae <- read_table2("algaeBloom.txt", col_names=
                       c('season','size','speed','mxPH','mnO2','Cl','NO3','NH4',
                         'oPO4','PO4','Chla','a1','a2','a3','a4','a5','a6','a7'),
                     na="XXXXXXX")
```

In homework 1, we investigated basic exploratory data analysis for the `algaeBloom` dataset. One of the explaining variables is `a1`, which is a numerical attribute. Here, after standardization, we will transform `a1` into a categorical variable with 2 levels: high and low, and conduct its classification using those 11 variables (i.e. do not include a2, a3,. . . , a7).

We first improve the normality of the numerical attributes by taking the log of all chemical variables. *After* log transformation, we **impute** missing values using the median method. Finally, we transform the variable `a1` into a categorical variable with two levels: high if a1 is greater than 5, and low if a1 is smaller than or equal to 5.

```
algae.transformed <- algae %>% mutate_at(vars(4:11), funs(log(.)))
algae.transformed <- algae.transformed %>%
  mutate_at(vars(4:11),funs(ifelse(is.na(.),median(.,na.rm=TRUE),.)))
# a1 == 0 means low
algae.transformed <- algae.transformed %>% mutate(a1 = factor(as.integer(a1 > 5), levels = c(0, 1)))
```

Classification Task: We will build classification models to classify `a1` into `high` vs. `low` using the dataset `algae.transformed` as above, and evaluate its training error rates and test error rates. We define a new function, named `calc_error_rate()`, that will calculate misclassification error rate.

```
calc_error_rate <- function(predicted.value, true.value){
  return(mean(true.value!=predicted.value))
}
```

Training/test sets: Split randomly the data set in a train and a test set:

```
set.seed(1)
test.indices = sample(1:nrow(algae.transformed), 50)
algae.train=algae.transformed[-test.indices,]
algae.test=algae.transformed[test.indices,]
```

In a binary classification problem, let p represent the probability of class label "1", which implies that $1-p$ represents probability of class label "0". The *logistic function* (also called the "inverse logit") is the cumulative distribution function of logistic distribution, which maps a real number z to the open interval $(0, 1)$:

$$p(z) = \frac{e^z}{1+e^z}. \tag{1}$$

1. (2 pts) Show that indeed the inverse of a logistic function is the *logit* function:

$$z(p) = \ln\left(\frac{p}{1-p}\right). \tag{2}$$

2. Assume that $z = \beta_0 + \beta_1 x_1$, and $p = \text{logistic}(z)$. (2 pts) How does the odds of the outcome change if you increase x_1 by two? (1 pts) Assume β_1 is negative: what value does p approach as $x_1 \to \infty$? (1 pts) What value does p approach as $x_1 \to -\infty$?

3. Use logistic regression to perform classification in the data application above. Logistic regression specifically estimates the probability that an observation as a particular class label. We can define a probability threshold for assigning class labels based on the probabilities returned by the `glm` fit.

   In this problem, we will simply use the "majority rule". If the probability is larger than 50% class as label "1". (2 pts) Fit a logistic regression to predict `a1` given all other features in the dataset using the `glm` function. (2 pts) Estimate the class labels using the majority rule and (2 pts) calculate the training and test errors using the **calc_error_rate** defined earlier.

   For logistic regression one needs to predict type `response`

   ```
   predict(glm.obj, test.data, type="response")
   ```

4. We will construct ROC curve based on the predictions of the *test* data from the model we obtained from the logistic regression above. (3 pts) Plot the ROC for the test data for the logistic regression fit. Compute the area under the curve(AUC).

   Hints: In order to construct the ROC curves one needs to use the vector of predicted probabilities for the test data. The usage of the function `predict()` may be different from model to model. For logistic regression one needs to predict type `response`, see Lab 3.

```
predict(glm.obj, test.data, type="response")
```

Fundamentals of the bootstrap (10 pts)

In the first part of this problem we will explore to understand the fact that approximately 1/3 of the observations in a bootstrap sample are *out-of-bag*.

1. (4 pts) Given a sample of size n, what is the probability that any observation j is *not* in a bootstrap sample? Express your answer as a function of n.
2. (2 pts) Compute the above probability for $n = 1000$.
3. (4 pts) Verify that your calculation is reasonable by resampling the numbers 1 to 1000 with replacement and printing the ratio of missing observations. Hint: use the `unique` and `length` functions to identify how many unique observations are in the sample. Note that the answer does not have to be exactly the same as what you get in (2) due to randomness in sampling.

Cross-validation estimate of test error (12 pts)

In this problem, we will apply cross-validation to estimate test error rate of logistic regression on the `Smarket` dataset available in `ISLR` package. The dataset contains daily percentage returns for the S&P 500 stock index between 2001 and 2005. In particular, the data contains 1250 observations on the following 9 variables:

- Year: The year that the observation was recorded
- Lag1: Percentage return for previous day
- Lag2: Percentage return for 2 days previous
- Lag3: Percentage return for 3 days previous
- Lag4: Percentage return for 4 days previous
- Lag5: Percentage return for 5 days previous
- Volume: Volume of shares traded (number of daily shares traded in billions)
- Today: Percentage return for today
- Direction: A factor with levels Down and Up indicating whether the market had a positive or negative return on a given day

We are interested in building a classifier in order to predict `Direction` using all variables except for `Year` and `Today` as predictors. We do the following transformation to convert the factor response into binary values: 0 for `Down` and 1 for `Up`.

```
dat = subset(Smarket, select = -c(Year,Today))
dat$Direction = ifelse(dat$Direction == "Up", 1, 0)
```

In this problem, we will again simply use the "majority rule". If the predicted probability is larger than 50%, classify the observation as `1`.

1. (2 pts) Splite `dat` into a training set of 700 observations, and a test set of the remaining observations. (2 pts) Fit a logistic regression model, on the training data, to predict the `Direction` using all other variables except for `Year` and `Today` as predictors. (2 pts) Calculate the error rate of this model on the test data. Use `set.seed(123)` in the begining of your answer.
2. (4 pts) Use a 10-fold cross-validation approach on the whole `dat` to estimate the test error rate. (2 pts) Report the estimated test error rate you obtain. Use `set.seed(123)` in the begining of your answer.

Just as what we did in Lab4, you can use the following key function to carry out k-fold cross-validation.

```
do.chunk <- function(chunkid, folddef, dat, ...){
  # Get training index
  train = (folddef!=chunkid)
  # Get training set and validation set
```

```
  dat.train = dat[train, ]
  dat.val = dat[-train, ]
  # Train logistic regression model on training data
  fit.train = glm(Direction ~ ., family = binomial, data = dat.train)
  # get predicted value on the validation set
  pred.val = predict(fit.train, newdata = dat.val, type = "response")
  pred.val = ifelse(pred.val > .5, 1,0)

  data.frame(fold = chunkid,
             val.error = mean(pred.val != dat.val$Direction))
}
```

Problems below for 231 students only (28 pts)

Discrinant functions (8 pts)

A multivariate normal distribution has density

$$f(x) = \frac{1}{(2\pi)^{p/2}|\Sigma|^{1/2}} exp\left(-\frac{1}{2}(x-\mu)^T\Sigma^{-1}(x-\mu)\right)$$

In quadratic discriminant analysis with two groups we use Bayes rule to calculate the probability that Y has class label "1":

$$Pr(Y = 1 \mid X = x) = \frac{f_1(x)\pi_1}{\pi_1 f_1(x) + \pi_2 f_2(x)}$$

where $\pi_2 = 1 - \pi_1$ is the prior probability of being in group 2. Suppose we classify $\hat{Y} = k$ whenever $Pr(Y = k \mid X = x) > \tau$ for some probability threshold τ and that f_k is a multivariate normal density with covariance Σ_k and mean μ_k. Note that for a vector x of length p and a $p \times p$ symmetric matrix A, x^TAx is the *vector quadratic form* (the multivariate analog of x^2). Show that the decision boundary is indeed quadratic by showing that $\hat{Y} = 1$ if

$$\delta_1(x) - \delta_2(x) > M(\tau)$$

where

$$\delta_k(x) = -\frac{1}{2}(x-\mu_k)^T\Sigma_k^{-1}(x-\mu_k) - \frac{1}{2}\log|\Sigma_k| + \log\pi_k$$

and $M(\tau)$ is some function of the probability threshold τ. What is the decision threshold, M(1/2), corresponding to a probability threshold of 1/2?

Linear and Quadratic Discriminant Analysis (12 pts)

Use the `algae.transformed` dataset we had earlier.

1. (4 pts) In LDA we assume that $\Sigma_1 = \Sigma_2$. Use LDA to predict whether `a1` is high or low using the `MASS::lda()` function. The `CV` argument in the `MASS::lda` function uses Leave-one-out cross validation LOOCV) when estimating the fitted values to avoid overfitting. We will talk about cross-validation in detail later in the class. For now, set the `CV` argument to true, so the program will automatically do cross-validation. Plot an ROC curve for the fitted values.

2. Quadratic discriminant analysis is strictly more flexible than LDA because it is not required that $\Sigma_1 = \Sigma_2$. In this sense, LDA can be considered a special case of QDA with the covariances constrained to be the same. (2 pts) Use a quadratic discriminant model to predict the `a1` using the function `MASS::qda`. Again setting `CV=TRUE` and plot the ROC on the same plot as the LDA ROC. (2 pts) Compute the area under the ROC (AUC) for each model. To get the predicted class probabilities look at the value of `posterior` in the `lda` and `qda` objects. (2 pts) Which model has better performance? (2 pts) Briefly explain, in terms of the bias-variance tradeoff, why you believe the better model outperforms the worse model?

Leave-one-out cross-validation (8 pts)

Consider the following intercept-only model:

$$Y = \beta + \varepsilon \qquad \varepsilon \sim N(0, \sigma^2),$$

where β is the paremeter we want to estimate. Suppose that we have n observations of response, i.e., $y_1, \ldots, y_n$, with uncorrelated errors.

1. (4 pts) Derive the least-squares estimate of β.
2. (4 pts) Suppose that we perform leave-one-out cross-validation (LOOCV). Recall that in LOOCV, we divide the data into n folds. What is the covariance between $\hat{\beta}^{(1)}$, the least squares estimator of β that we obtain from taking the 1st fold as validation set, and $\hat{\beta}^{(2)}$, the least squares estimator of β that we obtain from taking the 2nd fold as validation set?