Homework 1 R markdown

Abra Brisbin

2023-01-29

Load packages here.

library(ISLR)  
library(FNN)  
library(dplyr)  
library(ggformula)

#### Intellectual Property:

These problems are the intellectual property of the instructors and may not be reproduced outside of this course.

## Problem 1: Analyzing Gas Mileage

In this problem, you will use analyze gas mileage using the **Auto** data set from the ISLR library in R.

#### Question 1 (2 points)

Load the **ISLR** library into R and look at the first few rows of the **Auto** data set.

head(Auto)

## mpg cylinders displacement horsepower weight acceleration year origin  
## 1 18 8 307 130 3504 12.0 70 1  
## 2 15 8 350 165 3693 11.5 70 1  
## 3 18 8 318 150 3436 11.0 70 1  
## 4 16 8 304 150 3433 12.0 70 1  
## 5 17 8 302 140 3449 10.5 70 1  
## 6 15 8 429 198 4341 10.0 70 1  
## name  
## 1 chevrolet chevelle malibu  
## 2 buick skylark 320  
## 3 plymouth satellite  
## 4 amc rebel sst  
## 5 ford torino  
## 6 ford galaxie 500

What data mining strategy would you use to investigate the following questions? (Note that the orderings for the scenarios and answer choices on Canvas might differ from those shown below.)

* You are building an app for a used-car website that will take information about the year, engine displacement, and weight of cars, and determine whether they are most likely American (origin = 1), European (2), or Japanese (3).

**Multiple Choice Answer (AUTOGRADED on Canvas)**: one of *Regression*, *Classification*, or *Unsupervised learning*

* The manager of a used-car lot wants to arrange groups of similar cars on the lot. The manager wants to understand the relationships between the year, engine displacement, and weight of cars to identify informative groupings.

**Multiple Choice Answer (AUTOGRADED on Canvas)**: one of *Regression*, *Classification*, or *Unsupervised learning*

* You are building an app for a used-car website that will take information about the year, engine displacement, and weight of cars, and estimate their horsepower.

**Multiple Choice Answer (AUTOGRADED on Canvas)**: one of *Regression*, *Classification*, or *Unsupervised learning*

#### Question 2 (3 points)

We would like to use K-nearest neighbors to predict the gas mileage (mpg, miles per gallon) of cars based on their weight (in pounds) and their year of manufacture. Fill in the blanks: In this analysis, the main reason why standardizing the data is a (good/bad) idea is because the variables (mpg and weight, mpg and year, weight and year) have very (similar/different) (means/standard deviations).

**Fill-in-the-blank Answer**:

#### Question 3 (2 points)

Set R’s seed to 1 (for Homework 1) with: **set.seed(1)**

Create a groups vector of 256 copies of the number 1 (to represent observations that will be in the training set) and 136 copies of the number 2 (to represent observations that will be in the validation set. Then use **sample()** to randomize the order of the vector.

Make a vector that contains TRUE for each data point of Auto that will be in the training set, and FALSE for each data point that will be in the test (or validation) set.

Enter your R code below.

**Code Answer**:

set.seed(1)  
  
groups = c(rep(1, 256), rep(2, 136))  
random\_groups = sample(groups, 392)  
in\_train = (random\_groups == 1)  
  
x\_train <- Auto %>%  
 filter(in\_train)  
  
x\_test <- Auto %>%  
 filter(!in\_train)

#### Question 4 (2 points)

Standardize the weight and year columns of the training set. Then standardize the weight and year columns of the test set, *using the original mean and standard deviation of the training set*.

Enter your R code below.  
**Code Answer**:

#quant\_train\_std = scale(Default2[in\_train, 3:4])  
  
  
x\_train <- x\_train %>%  
 select(weight, year)  
  
x\_train = scale(x\_train)  
  
x\_test <- x\_test %>%  
 select(weight, year)  
  
x\_test = scale(x\_test, center = attr(x\_train, "scaled:center"),  
 scale = attr(x\_train, "scaled:scale"))

#### Question 5: **(3 points)**

Use 1-nearest neighbor regression (fit on the standardized training data) to predict the gas mileage of the cars in the standardized validation set. Compute the mean squared error.

Enter your R code below.

**Code Answer**:

prediction = knn.reg(train = x\_train,  
 test = x\_test,  
 y = Auto$mpg[in\_train],  
 k = 1)  
  
mean( (prediction$pred - Auto$mpg[!in\_train])^2 )

## [1] 15.24743

#### Question 6 (1 point)

What is the MSE for the validation set? (Round your answer to 2 decimal places.)

Your Answer:  
**Numeric Answer (AUTOGRADED on Canvas)**:

mean( (prediction$pred - Auto$mpg[!in\_train])^2 )

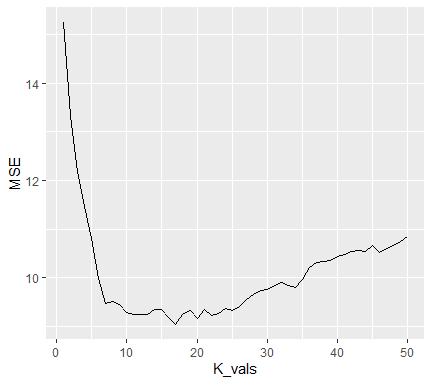
## [1] 15.24743

#### Question 7 (4 points)\*\*

Use a for() loop to apply K-nearest neighbors regression to the same training and validation sets, for values of k from 1 to 50. Make a plot of the MSE (calculated for the validation set predictions) as a function of k.

Enter your R code and plot on this question on Canvas. (Use **Insert** -> **Image** to insert the plot.)  
**Code and Graph Answer**:

K\_vals = seq(1, 50, by = 1)  
MSE = numeric(length = length(K\_vals))  
  
for(ii in 1:length(K\_vals)){  
 predictions = knn.reg(train = x\_train,  
 test = x\_test,  
 y = Auto$mpg[in\_train],  
 k = K\_vals[ii])  
  
   
 ### Not quite there  
 MSE[ii] = mean( (predictions$pred - Auto$mpg[!in\_train])^2 )  
}  
  
gf\_line(MSE ~ K\_vals)



#### Question 8: **(2 points)**

In your opinion, which value of k is the best choice? Why?

**Text Answer**:

## Problem 2:

In this problem, you will use K-nearest neighbors to classify people’s income as >$50,000 or $50,000.

You are about to start **Problem 2 of 2**, which analyzes personal income using the Census\_income.csv data file (available under Canvas Lesson 1 resources). You can find more information at [the data source](https://archive.ics.uci.edu/ml/datasets/census+income).

Data Source: Kohavi, R and B. Becker. (1996). [UCI Machine Learning Repository](http://archive.ics.uci.edu/ml). Irvine, CA: University of California, School of Information and Computer Science.

#### Question 9: (2 points)

Read the data into R. One-hot encode the variable Sex, using Male as the default value. (Unfortunately, because this data set is from the US Census, it did not allow allow an option of Intersex, or distinguish between sex and gender.)

Enter your R code below.  
**Code Answer**:

library(readr)  
Census\_income <- read\_csv("Census\_income.csv")  
  
Census\_income1 <- Census\_income %>% # preserve original df  
 mutate(Sex = ifelse(Sex == 'Male', 1, 0))

#### Question 10 (2 points)

Set R’s random seed to 1 again. Then randomly sample 20,000 individuals to be in the training set.

Enter your R code below.  
**Code Answer**:

set.seed(1)  
a <- dim(Census\_income1)[1]  
b <- 20000  
c <- a-b #because math is hard  
groups <- c(rep(1, b), rep(2, c))  
random\_groups = sample(groups, a)  
  
in\_train = (random\_groups == 1)  
  
x\_train <- Census\_income1 %>%  
 filter(in\_train)  
  
x\_test <- Census\_income1 %>%  
 filter(!in\_train)

#### Question 11 (2 points)

Standardize the EducYears and Age variables from the training data. Combine these with the unstandardized, one-hot encoded Sex variable from the training data to create a data frame x\_train.

Use the original means and standard deviations from the training data to standardize EducYears and Age from the validation data. Combine these with the unstandardized, one-hot encoded Sex variable to create a data frame x\_test.

Enter your R code below.  
**Code Answer**:

#x\_train\_std <- x\_train[, c("EducYears", "Age")]  
x\_train\_std <- scale(Census\_income1[in\_train, c("EducYears", "Age")])  
#x\_test\_std <- x\_test[, c("EducYears", "Age")]  
x\_test\_std <- scale(Census\_income1[in\_train, c("EducYears", "Age")],   
 center = attr(x\_train\_std, "scaled:center"),   
 scale = attr(x\_train\_std, "scaled:scale"))  
  
x\_train <- cbind(Census\_income1$Sex[in\_train],   
 x\_train\_std)  
x\_test <- cbind(Census\_income1$Sex[!in\_train],   
 x\_test\_std)

## Warning in cbind(Census\_income1$Sex[!in\_train], x\_test\_std): number of rows of  
## result is not a multiple of vector length (arg 1)

#### Question 12 (2 points)

Use 25-nearest neighbor classification (fit on the training set) to predict whether the income of each individual in the validation set is >50K or <=50K.

Find the confusion matrix. You should be able to produce a matrix table with two rows and two columns, similar to the one below.

Please enter the information as whole numbers. Note carefully the labels for the rows and columns, and be sure to orient your table accordingly.

prediction <- knn(train = x\_train,  
 test = x\_test,  
 cl = Census\_income1$Income[in\_train],  
 k = 25)

Please enter the information *exactly as it appears in R*.

| . | Actual income <= 50K | Actual Income > 50K |
| --- | --- | --- |
| Classified <= 50K | **[A]** | **[B]** |
| Classified > 50K | **[C]** | **[D]** |

**Numeric Answer (AUTOGRADED on Canvas)**:  
[A] = 8764 [B] = 1708 [C] = 832 [D] = 1257

#### Question 13 (1 point)

What is the overall error rate on the validation set? Enter your answer as a decimal between 0 and 1, rounded to 4 decimal places.

round((832+1708)/12561,4)

## [1] 0.2022

**Numeric Answer (AUTOGRADED on Canvas)**:

#### Question 14 (1 point)

What proportion of people making > $50,000 were misclassified? Enter your answer as a decimal between 0 and 1, rounded to 4 decimal places.

round(1257/(1708+1257),4)

## [1] 0.4239

**Numeric Answer(AUTOGRADED on Canvas)**:

#### Question 15 (2 points)

Make a grid of example points with values of education from 1 to 16, ages from 17 to 75, and sex from 0 to 1. Standardize education and age using the original mean and standard deviation of the training set (from question 11).

Create a data frame, x\_example, containing the standardized education and age and the unstandardized sex from the example points. The order of the columns should match the order of the columns in x\_train.

Enter your R code below.  
**Code Answer**:

educ\_vals = seq(1, 16, by = 1)  
ages\_vals = seq(17, 75, by = 1)  
sex\_vals = c(0, 1)  
  
example\_grid = expand.grid(educ\_vals, ages\_vals, sex\_vals)  
  
example\_std = scale(example\_grid[,1:2],  
 center = attr(x\_train\_std, "scaled:center"),  
 scale = attr(x\_train\_std, "scaled:scale"))  
  
x\_example = cbind(example\_grid$Var3, example\_std)

#### Question 16 (2 points)

Use 25-nearest neighbors to predict the income classifications of the example points, using the same training data as in question 12. Make graphs showing the relationship between education, age, sex, and predicted income.

Use **Insert** -> **Image** to upload your graphs to this question on Canvas.

**Graph Answer**:

#### Question 17 (3 points)

Write 3-6 sentences interpreting the graphs you made in the previous question. (For purposes of interpreting the results, note that the data are from the 1990s.)

**Text Answer**: