CS 422: Data Mining

Department of Computer Science
Illinois Institute of Technology
Vijay K. Gurbani, Ph.D.

Fall 2024: Homework 3 (10 points)

Due date: Wed, September 25, 2024, 11:59:59 PM Chicago Time

Please read all of the parts of the homework carefully before attempting any question. If you detect any ambiguities in the instructions, please let me know right away instead of waiting until after the homework has been graded.

1. Exercises (2 points)

1.1 [1 point] ISLR 2e (Gareth James, et al.)

Section 3.7 (Exercises), page 123: Exercise 6. (Hint: The least squares line is given by the equation below.)

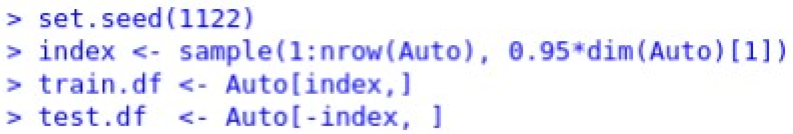
$$y_i = \hat{\beta}_0 + \hat{\beta}_1 x_i$$

1.2 [1 point] Section 3.7 (Exercises), page 120: Exercises 1, 3, 4-a.

2. Programming Problem (8 points divided evenly by each subsection)

Install package ISLR. This package contains a dataset called Auto.

In this problem, you will tackle multi-variate linear regression using the Auto dataset from Problem 2.2. You will set aside a portion of this dataset (5%) for testing, and use the remaining portion (95%) to train your OLS model. To divide the dataset into training and testing sets, issue the following commands **exactly as shown:**



```
> set.seed(1122)
> index <- sample(1:nrow(Auto), 0.95*dim(Auto)[1])
> train.df <- Auto[index,]
> test.df  <- Auto[-index, ]
```

The set.seed() command seeds the pseudo-random number generator for reproducibility. The sample() command picks 352 random numbers between 1 and the total number of rows in the Auto dataset, without replacement. These numbers correspond to the indices of the Auto dataset, and all observations at these indices constitute the training dataset (the third line of code above). The remaining numbers (note the use of -index in the fourth line of code above) correspond to indices not chosen through random sampling, and thus, they become the test set of the dataset. Once you have divided your dataset as shown above, you will train on the train.df dataframe and test your trained model on the test.df dataframe.

(a) You will create a regression model.

(i) From the training dataset, create a regression model using all the predictors except name to predict mpg. Return the model object.

(ii) Why is using name as a predictor not a reasonable thing to do? Please describe in 1-2 sentences and put these sentences in your PDF file for Exercise 1. Please label the answer in the PDF as “Q2(a)(i) ...”

(iii) How well does the model fits the data? Study the R^2 , adjusted R^2, RSE and RMSE values to answer this question.

Hint: You can extract the values of R^2 and adjusted R^2 from the object returned to you by the summary() method. You will need to write R code to get the values of RSE and RMSE. Recall that:

$$RSE = \sqrt{\frac{1}{n-p-1}RSS}$$

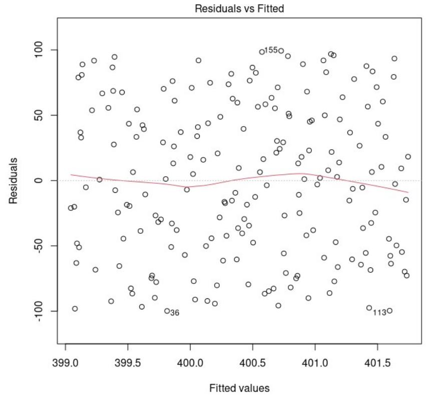
$$\text{RMSE} = \sqrt{\frac{\sum(y_i - \hat{y}_i)^2}{N}}$$

$$\text{RSS:} \quad \sum_{i=1}^{n}(y_i - \hat{y}_i)^2$$

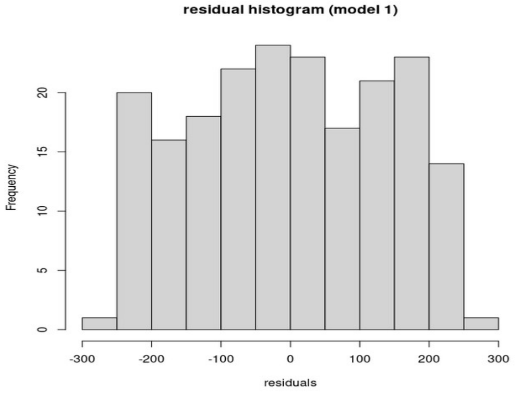
Return a string of the format shown below that contains the values of R^2, adjusted R^2 , RSE, and RMSE rounded to two decimal digits:

```
r-sq=X.XX,adjusted-r-sq=X.XX,rse=X.XX,rmse=X.XX
```

(iv) Plot the residuals of the model. Your graph should look like the following (this is NOT the answer, just a sample of what your graph should look like):



(v) Plot a histogram of the residuals of the model. The title of the plot should be “residual histogram (model 1)” (all lowercase); the x-label should be “residuals” (lowercase). Your graph should look like the following (this is NOT the answer):



(vi) Does the histogram follow a Gaussian distribution? What can you say about the distribution of the residuals? Please describe in 1-2 sentences and put these sentences in your PDF file. Please label the answer in the PDF as "Q2(a)(vi): ..."

(b) Starting with the regression model you have created in (a), your aim is to narrow down the features to the 3 attributes will act as the best predictors for regressing on **mpg**. To do so, go back to the model you created in (a) and determine which **three** predictors are the most statistically significant.

(i) Using these three statistically significant predictors create a **new regression model** using the same training data as in (a). Return your new regression model.

(ii) How well does the model fits the data? Study the R^2 , adjusted R^2, RSE and RMSE values to answer this question. Return a string of the format shown below that contains the values of R^2, adjusted R^2 , RSE, and RMSE rounded to two decimal digits:

```
r-sq=X.XX,adjusted-r-sq=X.XX,rse=X.XX,rmse=X.XX
```

(iii) Comparing the R^2 , adjusted R^2, RSE, and RMSE values of this model and the model you created in 2(a)(i), which model do you think is better? Return a string that has one of the following values:

"`model in 2(a)(i)`" or "`model in 2(b)(i)`".

Any other values will be unacceptable.

(iv) Please provide a justification in 1-2 sentences for your choice in the above question. Please put these sentences in your PDF file, and label the answer in the PDF as "Q2(b)(vi): …"

(c) Using the predict() method, fit the **test dataset** to the model you created in **2(a)(i) (Model 1)** to determine how many predictions from the held out **test** dataset fell within the range of the prediction interval.

To best assist you with the analysis, create a new data frame that has 5 columns; the column names must be "prediction", "response", "lower", "upper", and "matches" (see example below). The dataframe's "prediction", "upper", and "lower" columns will be extracted from the object return by the predict() method. The "response" column will be extracted from the **test** dataset, and the "matches" column will be calculated as described below.

Count how many of the fitted values matched the **mpg** in the **test** dataset at a 95% confidence level by creating **prediction** intervals. To be considered a match, the **response** value of each observation in in the test dataset should be in the prediction interval created by predict(). Note that we have the ground truth in our test dataset, so we can count the matches as one measure of accuracy.

To help facilitate this counting, add a column (called "matches") in the data frame you created above that contains 1 if the response value was in the prediction interval, and 0 otherwise. To find out the total observations correctly predicted in the **test** dataset, one can simply count the number of 1's in the "matches" column.

Your function should return this dataframe.

As an example, this dataframe will look like the following:

prediction	response	lower	upper	matches
56.23	56.10	50.54	62.20	1
30.10	34.19	31.25	34.50	0

…