CptS 475/575: Data Science, Fall 2023

Assignment 5: Regression and Classification

Release Date: Friday, Oct. 13, 2023 **Due Date:** Friday, Oct. 27, 2023 (11:59 pm)

General instruction: The first part of this assignment assesses your understanding of linear and logistic regression. The second part focuses on classification; it requires you to take a set of ecommerce product descriptions and classify them based on which category they belong to. The assignment is broken into four problems with points assigned to each.

Your solution will be submitted as a PDF/HTML file, which must **include your full, functional code and relevant results as stated in each part**. You are encouraged to use R Markdown to prepare your file if you work with R. You are free to use Python to solve the problems; you can use Jupyter notebook to prepare your file in that case.

1) (15 points) This question involves the use of multiple linear regression on the cars_graphics data set available on Canvas in the Datasets for Assignments Module. Ensure that values are represented in the appropriate types.
   a. (5 points) Perform a multiple linear regression with MPG as the response and all other variables except Car as the predictors. Show a printout of the result (including coefficient, error, and t values for each predictor). Comment on the output:
      i) Which predictors appear to have a statistically significant relationship to the response, and how do you determine this?
      ii) What does the coefficient for the Weight variable suggest, in simple terms?
   b. (5 points) Produce diagnostic plots of the linear regression fit. Comment on any problems you see with the fit. Do the residual plots suggest any unusually large outliers? Does the leverage plot identify any observations with unusually high leverage?
   c. (5 points) Fit linear regression models (at least 3) with interaction effects with Horsepower as the response. Do any interactions appear to be statistically significant?

2) (30 points) This problem involves the Boston data set, which can be attached from library MASS in R and is also made available in the Datasets for Assignments module on Canvas. We will now try to predict per capita crime rate (crim) using the other variables in this data set. In other words, per capita crime rate is the response, and the other variables are the predictors.
   a. (6 points) For each predictor, fit a simple linear regression model to predict the response. Include the code, but not the output for all models in your solution.
   b. (6 points) In which of the models is there a statistically significant association between the predictor and the response? Considering the meaning of each variable, discuss the relationship between crim and each of the predictors nox, chas, rm, dis and medv. How do these relationships differ?
   c. (6 points) Fit a multiple regression model to predict the response using all the predictors. Describe your results. For which predictors can we reject the null hypothesis $H_0 : \beta_j = 0$?
   d. (6 points) How do your results from (a) compare to your results from (c)? Create a plot displaying the univariate regression coefficients from (a) on the x-axis, and the multiple regression coefficients from (c) on the y-axis. That is, each predictor is displayed as a single

point in the plot. Its coefficient in a simple linear regression model is shown on the x-axis, and its coefficient estimate in the multiple linear regression model is shown on the y-axis. What does this plot tell you about the various predictors?

e. (6 points) Is there evidence of non-linear association between any of the predictors and the response? To answer this question, for each predictor X, fit a model of the form:

$$Y = \beta_0 + \beta_1 X + \beta_2 X^2 + \beta_3 X^3 + \varepsilon$$

Hint: use the poly() function in R. Again, include the code, but not the output for each model in your solution, and instead describe any non-linear trends you uncover.

3) (15 points) Suppose we collect data for a group of students in a statistics class with variables:

X_1 = hours studied,

X_2 = undergrad GPA,

X_3 = PSQI score (a sleep quality index), and

Y = receive an A.

We fit a logistic regression and produce estimated coefficient, $\beta_0 = -7$, $\beta_1 = 0.1$, $\beta_2 = 1$, $\beta_3 = -.04$.

a. (5 points) Estimate the probability that a student who studies for 35 h, has a PSQI score of 11 and has an undergrad GPA of 3.0 gets an A in the class. Show your work.

b. (5 points) How many hours would the student in part (a) need to study to have a 60 % chance of getting an A in the class? Show your work.

c. (5 points) How many hours would a student with a 3.0 GPA and a PSQI score of 4 need to study to have a 60 % chance of getting an A in the class? Show your work.

4) (40 points) For this question, you will use a naïve Bayes model to classify ecommerce product descriptions by their category. The product descriptions have been pre-processed and cleared of any major confounding factors such as HTML tags, but it is up to you to check for other problems and to prepare them for classification. The ecommerceData dataset can be found on the Modules page under Datasets for Assignments. The dataset consists of the ecommerce product descriptions (text) and the category (label) it belongs to. There are a total of 4 categories. Prepare the dataset for classification as suggested below.

a. (20 points) Tokenization
In order to use Naïve Bayes effectively, you will need to split your text into tokens. It is common practice when doing this to reduce your words to their stems so that conjugations produce less noise in your data. For example, the words "speak", "spoke", and "speaking" are all likely to denote a similar context, and so a stemmed tokenization will merge all of them into a single stem. R has several libraries for tokenization, stemming, and text mining. Examples of such libraries that you may want to use as a starting point are tokenizers, SnowballC, and tm, respectively. Alternatively, some of you may want to consider using quanteda, which will handle these functionalities along with others needed in building your

model in the next step. Similarly, Python has libraries such as sklearn and nltk for processing text.

You will need to produce a document-term matrix from your stemmed tokenized data. This will have a very wide feature set (to be reduced in the following step) where each word is a feature, and each article has a list of values representing the number of occurrences of each word in its context.

Before representing the feature set in a non-compact storage format (such as a plain matrix), you will want to remove any word which appears in too few documents. For this assignment, you will remove 1% of the words corresponding to the least frequent words in the document i.e., only 99% of the terms should be kept. To demonstrate your completion of this part, you can simply select and print the text of a random product description along with the non-zero entries of its feature vector.

b. (20 points) Classification

For the final portion of this assignment, you will build and test a Naïve Bayes classifier with your data. Since we have multiple classes in the given dataset, a Multinominal Naïve Bayes model would be more appropriate. First, you will need to use feature selection to reduce your feature set. A popular library in R for this is caret. It has many functionalities for reducing feature sets, including removing highly correlated features. You may wish to try several different methods to see which produces the best results for the following steps.

Next, you will split your data into a training set and a test set. Your training set should comprise approximately 85% of your articles, however, you may try several sizes to find which produces the best results. Whatever way you split your training and test sets, ensure that your four categories are equally represented in both sets.

Next, you will build your Naïve Bayes classifier from your training data. The e1071 package in R and sklearn library in Python are commonly used for this. Finally, you can use your model to predict the categories of your test data.

Once you have produced a model that generates the best predictions you can get, print a confusion matrix of the results to demonstrate your completion of this task. For each class, give scores for precision (TruePositives / TruePositives+FalsePositives) and recall (TruePositives / TruePositives+FalseNegatives).