CS 4232/5232 – Fall 2023 – Homework 4
Assigned: 11/17/2023
Due: 11/30/2023 at 11:59 p.m.
Maximum Grade: 100 pts.

Objectives: The objectives of this homework are the following:

- Learn how to fit decision trees and logistic regression in R.
- Learn how the algorithm for fitting unthresholded perceptrons works.
- Learn how to implement logistic regression.
- Learn how to construct and train neural networks.

Instructions:

- This homework is to be done individually.
- Submit **a single HTML file** named: yourlastname_hw4.html **and also the source R markdown file** (with a suffix of .Rmd) with the answers to your homework. To generate this HTML file, you **must use R markdown**. Check the video here for more information https://rmarkdown.rstudio.com/lesson-1.html.

Activity 1: (25 pts.) (Decision Trees in R) (Fitting decision trees with R) In this exercise, you will learn that CART trees can be used not only for classification but also for regression. To this end and using the Caret package/library of R, do the following:

a) Download the Student Performance dataset from UCI: https://archive.ics.uci.edu/ml/datasets/Student+Performance. In this activity, you will be using only the "student-mat.csv" file, which corresponds to the performance of students in the final mathematics evaluation. Read carefully the description of the dataset from the website.

b) (1 pts.) Read the dataset into R's workspace. You can use the read_delim function of the readr package, which will help you deal with the semicolons. Do not forget to cast categorical attributes to factors; you can do this using the col_names and col_types arguments of read_delim.

c) (5 pt.) What attributes of the dataset would you choose to predict the performance of the students in their final math evaluation G3? Explain why you chose those attributes in detail.

d) (3 pts.) Using the CART algorithm of Caret (rpart2) and the attributes that you selected in Task c, build a decision tree to predict the performance of the students in their final math evaluation G3. To accomplish this, you will first use grid-search CV (use train and trainControl) with k = 5, to find the "maximum depth" hyperparameter value in the range [1, 2, 3, 4] such that it leads to the highest accuracy. Then, build a decision tree using this best value for the maximum depth on the training set. Print out the table with the accuracies for all the different hyperparameter values and then generate the plot that R automatically creates with these values.

e) (1 pt.) "Pretty print" the tree obtained in Task d using fancyRPartPlot.

f) (3 pt.) Using k-fold Leave-one-out cross validation with k = 5, estimate the error rate/ accuracy of the decision tree that you build built in Task d.

g) (3 pts.) Repeat Task d, but instead use the C5.0 algorithm of the Caret package. This might require you to install the C50 package. Use minCases as the hyperparameter to tune.
h) (3 pts.) Using k-fold Leave-one-out k-fold cross validation with k = 5, estimate the error rate / accuracy of the decision tree that you found in Task g.
i) (6 pts.) What insights into the domain can you gather from the trees that you trained? Do they make sense according to your knowledge of used cars student performance in the classroom? Why? Explain in depth.

Activity 2: (25 pts.) (Logistic Regression in R) Using R, perform the following activities.

a) (10 pts.) Implement in R the algorithm we saw in class to learn logistic regression models. The gradient is shown in Slide 58 of Lecture Topic 5 - Classification Part IIa.
b) (5 pts.) Using your implementation of part a and the Adult training set (https://archive.ics.uci.edu/ml/machine-learning-databases/adult/), train a logistic regression model to predict if an "adult" earns less than 50K a year given the values of the other predictors in the dataset. You may need to take a sample of your dataset if it takes too long.
c) (2 pts.) Repeat part b but this time using R's built-in logistic regression model training tools.
d) (8 pts.) Comment on the differences between the execution times of your implementation and R's. Why is there a difference? Cite your sources.

Activity 3: (30 pts.) (Fitting Unthresholded Perceptrons) Perform the following activities related to unthresholded perceptrons:

a) (15 pts.) Run by hand, in pencil and paper, two epochs of the algorithm I showed in class to fit an unthresholded perceptron. Use a bias node, a learning rate of 0.1., random initial weights of 0.2, and the following dataset:

X1	X2	y
1	1	1.2
0	1	0.3

b) (15 pts.) Repeat part a) but this time using the stochastic gradient descent algorithm I presented in class.

Activity 4: (20 pts.) (Fitting Neural Networks with R or Python) In this activity, you need to use either R or Python to fit a neural network for solving the classification problem described below. Because TensorFlow/Keras can be very tricky to set up in some machines, this activity offers two options: Option 1 for those who want to install and use TensorFlow/Keras, and Option 2, for those that not willing to do that. In this activity, you can choose to do only one of the options and still get full credit.

Option 1: Using TensorFlow/Keras. For this option, do the following:

a) Download the Adult dataset from the UCI website https://archive.ics.uci.edu/ml/machine-learning-databases/adult/

and pre-process this dataset as you see fit.

b) (1 pt.) Standardize or Normalize the values of the input predictors.

c) (5 pts.) Using the Adult training set, train the following neural networks (NNs) to predict if an "adult" earns less than 50K a year given the values of the other predictors in the dataset:
   i. Three-layers, no dropout
   ii. Three-layers, with dropout
   iii. Four-layers, no dropout
   iv. Four-layers, with dropout
   v. Any other NN architecture(s) that you want

   In all these architectures, you are free to choose any number of units for each layer.

d) (5 pts.) Plot the training and the validation accuracy as functions of the number of epochs for each of these neural networks. You must make a single plot containing the performance of all neural networks. Your plot needs to have a title that identifies the architecture that generated each of the results.

e) (5 pts.) Write your conclusions after generating the plots of Activity c. Does the dropout layer improve the validation accuracy? Why? Why not? What is the impact of the number of layers? Why? Which one is the best architecture? Explain.

f) (4 pts.) From the five architectures of Activity c, select the one with the largest validation accuracy and evaluate it in the Adult Test set. Which one is better? Is the performance difference significant?

Option 2: Not using TensorFlow/Keras. For this option, do the following:

a) Download the Adult dataset from the UCI website
https://archive.ics.uci.edu/ml/machine-learning-databases/adult/
and pre-process this dataset as you see fit.

b) (1 pt.) Standardize or Normalize the values of the input predictors.

c) (5 pts.) Using the Adult training set, train the following neural networks (NN) to predict if an "adult" earns less than 50K a year given the values of the other predictors in the dataset:
   i. Three-layers
   ii. Four-layers
   iii. Any other NN architecture(s) that you want

   In all these architectures, you can choose any number of units for each layer; however, since the *neuralnet* and *scikit-learn* packages are not for deep learning, you should select layers with few neurons. If the training time is still too long, you can train on a sample of the entire set.

d) (5 pts.) Evaluate the three architectures of Activity c on the test set and make *a single figure* containing the bar plots with the test set accuracy of all three architectures. Your plot needs to have appropriate labels and a title identifying the results of each architecture. Remember that once you see the test set accuracy, you cannot go back and retrain your neural networks.

e) (5 pts.) Write your conclusions after generating the plots of Activity d. What is the impact of the number of layers? Why? Which one is the best architecture? Why? Explain.
f) (4 pts.) From the three architectures of Activity d, select the one with the largest test accuracy. Which one is better? Are the performance differences significant?

Important Notes:

- **You cannot have all your code in a single R markdown chunk**. You need to break down your markdown file into meaningful chunks. For example, one chunk can be a function definition, another a function call, another the definition of a set of related variables, etc.

Notes:

- If you are not using Tensorflow/Keras to fit your neural network, look at the *neuralnet* package if you are using R, and a look at the *MLPClassifier* class if you are using Python.
- As I mentioned in the first lab, **you cannot have all your code in a single R markdown chunk**. You need to break down your markdown file into meaningful chunks. For example, one chunk can be a function definition, another a function call, another the definition of a set of related variables, etc.