Assignment 2

Due date: 2nd October 5:59pm

Learning Objectives:

1. Using Caret for tuning.
2. Compare the performance of different cross-validation approach.
3. Know how to calculate AUC by predicted probabilities for classification.
4. Using neural network for classification.
5. Using neural network for regression.

Q1: Model Selection Using Caret (4 Marks)

Please use Caret to train several models with C5.0 algorithm on the credit.csv dataset on IVLE for Q1 of A2. Please complete "A2_Q1_template.R" by following the instructions in that file.

This question is testing your understanding in coding cross-validation and grid search by Caret. Performance of your tuning won't be graded.

Additional Remarks: Caret does not support stratified CV. But you can use the code from this posting to achieve stratified CV.
https://stackoverflow.com/questions/35907477/caret-package-stratified-cross-validation-in-train-function

Q2: Neural Network Classification (3 Marks)

In this question, you don't need to use Caret. You can directly use "neuralnet" package for classification on the same dataset of Q1. To build a classification model using "neuralnet", you need to use logistic activation function and set the output layer to use the same activation function as hidden layers. These two options are (act.fct = "logistic", linear.output = FALSE). Now you can take a look at your output vector and it should be between 0 to 1. This output can be interpreted as predicted probability. Last, calculate AUC by ROCR package's performance command performance(pred,"auc")@y.values. Performance does not matter for this question and you don't need to tune neuralnet. You will practice that in the next question. Remember to set.seed(42) before building your neuralnet model, which will allow the grader to verify your auc value against the model answer.

Additional Remarks: In the future, you can use Caret with "nnet" packages for classification.

Q3: Neural Network Regression (5 Marks)

Suppose you are going to buy a diamond and you don't know whether the price listed on the Internet is fair or enough. So you build a machine learning model by 9 features to predict the diamond price. For this question, you will use the data from diamonds_train.csv file to build a model and diamonds_test.csv for testing. The data contains 3000 rows with following columns:

1. Carat. Weight of the diamond
2. Cut. Quality of the cut. (Fair, Good, Very Good, Premium, Ideal)
3. Color. Diamond colour from J (worst) to D (best)
4. Clarity. A measurement of how clear the diamond is (I1 (worst), SI2, SI1, VS2, VS1, VVS2, VVS1, IF (best))
5. X. Size in x dimension.
6. Y. Size in x dimension.
7. Z. Size in x dimension.
8. Depth. Depth percentage = 2 * z / (x + y)
9. Table. Width of top of diamond relative to widest point.
10. Price.

Requirement:

1. You are required to use neuralnet package in R.
2. You are required to tune by Caret in R.
3. Train and tune a two-level neural network for this prediction task by tanh activation function and linear output. Report the optimal number of nodes in each layer.
4. The performance metric is RMSE.
5. Aside from your R code, also submit your predicted prices on the test dataset in a csv file, which contains a single column "price". There should be 739 data rows in this csv. Remember to scale back if you normalize your "price" column for training.
6. Remember to `set.seed` before training your model, such that your result is reproducible. If your predictions.csv cannot be reproduced by running your code, you will receive 0.
7. Top 10% submissions will receive bonus +1 mark, bottom 10% submissions will receive penalty -1 mark, subject to a maximum mark of 12 total, and a minimum mark of 0 total for this assignment.

Hints:

1. "neuralnet" function requires all the features name to be explicitly specified in the formula (i.e. it won't accept something like "price ~ .",

we'd need to expand . into column names). You can use the following two lines of codes

   a. n <- names(train_data_processed)
   b. formula <- as.formula(paste("price~", paste(n[!(n %in% "price")], collapse = " + ")))

2. It is recommended that you perform data pre-processing carefully. NN requires numerical features.
   a. So you must use onehot to convert categorical variables to numerical variables.
   b. Use max-min to scale your numerical features. This package does not do this for you. Also, you may need to carefully rescale your predicted Y so you can compute the correct RMSE.
3. If the speed is too slow for neuralnet package, you can try to change the following 3 parameters (threshold, stepmax, rep). But all of these just force your program to stop earlier, especially stepmax and rep. So although your program stops, the prediction output may not be optimal.

Submissions and Grading

- You need to submit three files:
  1. a *.R (or *.Rmd) file
  2. a *.PDF (or *.html generated by your rmarkdown) file of you results and answers.
  3. a *.csv for Q3
- **Name all files by your student number (e.g., A0123456X.R, A0123456X.html, A0123456X.csv) and upload to IVLE workbin submission folder "A2". <u>Do not zip your submissions.</u>**
- In your R script you can assume that dataset files are in the same directory as the R script, i.e. train_data <- read.csv("loan_train.csv", stringsAsFactors = TRUE)
- You may use any packages, but make sure your R file is runnable and has all the dependency packages imported e.g. library(C50).
- The page limit of the pdf file is maximum 2 pages including everything. The formatting is A4, default margin, 12 font size, single-spacing. There is no need to try to fill 2 pages. Correct answers are much more important than the length of your answers for grading.
- You may revise and submit as many times before deadline. Make sure to remove any old version that you don't wish to be graded.
- If you have questions about A1, feel free to email TA and cc me. Later, if you have questions about grading of A1, then you can email TA and cc me

because TA (not me) will grade your assignment by following my grading rules listed below.

Grading Rules

- Zero tolerance for Plagiarism
- Every day of late submission will result in 3 marks deducted, i.e. 4 days late = 0 mark.
- For Q1, you should use the provided R code template. The completed R code should be correct.
- For Q2, you should have correct R code that can produce a neuralnet classification model and the resulting auc value should match that of the model answer (which assumes the RNG seed of 42).
- For Q3, correct R code that satisfies all the question requirements as stated above.
- Submissions without a runnable R/Rmd file will receive a failing grade. Make sure all the dependency packages are imported e.g. library(C50)
- TA can judge the quality of your code and deduct up to 2 marks. For example, if you included quite a number of unnecessary codes (which shows you do not really know which line of R command is the real one that helps you conduct analysis). You can provide comments into your codes to show your understanding.
- TA can give you up to +2 bonus if you did an excellent job. The max of A1 is still 12 marks.