ENGSCI / STATS 255 | Assignment 4 | Due: 1:00pm, October 10 2019

Instructions for handing in. Please...

1. Attempt **ALL** questions.
2. Assignments may be typed in a word processor of your choice, or handwritten **neatly**.
3. Set answers out in order of the questions. Do **NOT** jump between questions.
4. There is no need to copy the questions out in your submission.
5. If answering a question using R, all commands that you enter should be provided as part of the answer, along with all relevant output. Include your ID number on output, where possible (e.g. as part of the title of a graph).

Submission: Assignments must be submitted using Canvas as a **single PDF** file, before the due date and time. Handwritten assignments will need to be scanned. **Prepare your assignments well in advance of the deadline in case of technical issues, as no extensions will be provided in this case.**

Notes:

- Summarising, analysing and communicating information is an important part of Operations Research. For this reason you will be expected to write answers which clearly communicate your thoughts. The mark you receive will be based on **your written English** as well as your technical work – review the relevant section in the Course Outline.
- **We encourage working together.** Discussing assignments and methods of solution with other students or getting help in understanding from staff and students is acceptable and encouraged. You must **write up your final assignment individually, in your own words.**
- By submitting this assignment, you confirm that you understand the University's policies on cheating, plagiarism and group work; that your submission is entirely your own work and you have not allowed access to any part of the assignment to any other person. **See the appropriate sections in the Course Outline for more details.**
- This assignment consists of **TWO** questions, and is marked out of **50 marks**.
  This assignment makes up **6%** of the final assessment for this course.

1. Breast Cancer Diagnosis (25 marks)

For this question we will be using the `BreastCancer` data set from the `mlbench` library. To load the data into R, simply load the library, and then type: `data("BreastCancer", package = "mlbench")`. Once the data is loaded, we recommend removing the `Id` attribute, since this has no value for classification: `BreastCancer$Id = NULL`. Information about the data set can be found by typing `?BreastCancer` into the console.

Note: completing the extra tutorial question from week 8 (available on Canvas) will be helpful for this question.

(a) Set the seed of the random number generator to 100 (`set.seed(100)`), and then generate a training data set of 400 data points using the `sample` function (the remaining 299 data points will be the test set).

(b) Create `pairs` plots for both the training and test data sets, colouring each point based on the `Class` of the data point. Comment on any observations.

(c) Using the `rpart.control(...)` arguments for `rpart`, set the termination criteria for generating the classification tree to be the *max depth* of the tree. (Disable any other termination criteria; i.e. set `minsplit=1` and `cp=0`.)

i. Generate three classification trees to classify the `Class` as each `benign` or `malignant`, using all the other attributes as predictor variables, varying the `maxdepth` termination criterion from 3 to 9 in steps of 3.

ii. Visualise the tree with a maximum depth of 3.

iii. For each tree, give the in-sample and out-of-sample confusion matrices.

iv. Create a table specifying the accuracy of each model, both in-sample and out-of-sample.

v. Comment on and explain what you notice about the in-sample vs. out-of-sample accuracy.

(d) Set the termination criteria to be a max depth of 3 for the following question (i.e. set `maxdepth=3`, `minsplit=1` and `cp=0`.)

i. By modifying the *loss matrix*, generate four classification trees (using all of the independent attributes), which range from having no *false positives* to no *false negatives* in the training data.

ii. For each tree, give the in-sample and out-of-sample confusion matrices.

iii. On a 2D scatterplot show the sensitivity vs. specificity of each classification model, include both the in-sample and out-of-sample values in different colours. (This plot can be generated in R, Matlab or Excel.)

iv. Comment on and explain any observations about the in-sample vs out-of-sample performance seen in the plot.

v. Suppose that we are much more concerned about false negatives than false positives. Explain what this means, and then recommend which of the classification models (from above) that we should choose.

2. Character Recognition (25 marks)

Download the file `letters.csv` from Canvas and read it into R; this file contains a data set, in which each of the 20,000 data points corresponds to a digitised capital letter that is known (this is given by the `lettr` attribute in column 1). There are also 17 independent attributes (columns 2–18), that have been computed from the digital image of each letter (e.g. `onpix`, is a count of the number of pixels that are black in the image). If you are interested, further details can be found here: `https://archive.ics.uci.edu/ml/datasets/Letter+Recognition`.

(a) Set the seed of the random number generator to 50, and then generate a random training data set of 18000 data points using the `sample` function (the remaining 2000 data points will be the test set).

(b) Filter the data set to create a training set called `letters.train` and a test set called `letters.test`

(c) The question uses `geom_jitter()` for `ggplot()`; this plot is just like a scatter plot, but 'jitters' the data so discrete values don't overlap. Plot two jitter plots for the `letters.test` data set:

- `xbox` vs. `onpix` with the points coloured by `lettr`; and
- `x2bar` vs. `y2bar` with the points coloured by `lettr`.

Which of these pairs of attributes would be better for classifying the data? Explain why.

(d) Using the `letters.train` data set create three random forests, with `ntree` set to 10, 100 and 1000, to predict the `lettr` attribute, given the other 16 attributes. Note the following:

- set the random seed to 100 immediately before each tree is constructed, and
- report the OOB (out-of-bag / out-of-sample) estimate of error rate, for each tree.

(e) For the random forest with 1000 trees, apply the `predict` function to the test set.

- Create a 26×26 confusion matrix based on these predictions.
- Which letter is the letter `P` occasionally misclassified as?
- What is the accuracy of this classifier on this test set?

We now wish to see if the Naïve Bayes classifier can be also be used to predict the correct letters for the same data set.

(f) Apply the Naïve Bayes method to the training data set to determine the class `lettr` using all the other attributes.

(g) Using the `predict()` function determine the in-sample and out-of-sample accuracy for this method. (R will report some warnings, but you can ignore them.)

(h) Show the confusion matrix for the out-of-sample predictions, above, and discuss this in comparison to the corresponding confusion matrix for the random forest.