COMP10200: Assignment 1, part b

Get Some Machine Learning Data

Go to the UCI Machine Learning Repository at https://archive.ics.uci.edu/ml/ and click the "View All Data Sets" link in the header.

In the selectors at right, choose "Classification", "Numerical", either "10 to 100" or "greater than 100" attributes (columns), and either "100 to 1000 "or "Greater than 1000" instances (rows).

Aside: Don't Use These Data Sets

The following data sets are not ok to use. If you hand in a solution using one of these, I will send it back and ask you to change it.

- Any dataset that appears in the Most Popular Data Sets list
- Wine datasets
- Cervical Cancer
- Leaf / Leaves
- Parkinsons

Read over the description of the data set and look at the data files. Make sure that the data is the right kind of data. What you're looking for is all numerical attributes except for the class label which can be a string or a set of discrete values (0, 1, 2, etc.). Make sure the data set makes sense for a classification task – if you're not sure, ask the instructor.

Some data sets have a single file, others separate training and test data, still others separate labels into a separate file from the data. For this exercise, we want a single CSV file with data and labels combined and with a header row that contains the name of each feature. You can stitch it together in Excel if you need to. In some cases, you will have to enter the feature names yourself. Here's an example of the type of file you want to end up with. Of course, yours will have more features and more examples.

```
ID, Age, Height, Weight, Diagnosis
1, 22, 180, 50, Sick
2, 45, 175, 55, Healthy
3, 99, 130, 35, Healthy
```

Read & Split the Data

Write a Python script that reads the CSV file and splits the data set randomly into testing and training data. You'll need 5 Numpy arrays for this: feature names, training data, training labels, testing data, and testing labels. Make sure the training set contains about 75% of the data and the testing set has the other 25%. Each time the script is run, it should produce a different split. There are a number of ways to do this with tools built in to **numpy**.

Aside: Just Use Numpy

Please do not use **sklearn**, **pandas** or any other library to read and split the data. There lots of tools out there, but I would like you to do this from scratch in **numpy** just this once.

Summarize It

Print summary information for the training and testing data: The name of each feature, followed by it's minimum, maximum, mean, and median in the training set and in the testing set. Make proper use of the NumPy array type for this (i.e. no loops!)

Graph It

Explore your training data by creating 2D scatter plots of pairs of features. Make sure the different class labels are represented with different colors. Find 3 pairs that seem promising for separating the classes and add code to your Python script to produce those 3 scatter plots. (Hint: Call `plt.figure(1)` and draw the first plot, then call `plt.figure(2)` and draw the second plot, etc.). Make sure each graph has a useful title and has the axes labelled.

Then produce a bar chart showing the overall frequency of each label. You can find information on how to do bar charts this in Chapter 3 of *Data Science from Scratch* (on eLearn).

Make proper use of the NumPy array type for this (i.e. no loops!)

Commenting

Follow the documentation conventions for the course. You'll find these in a python file in the Student Resources section of eLearn.

Within the code, put a headline on each section (Read It, Summarize It, Graph It, etc.) using a ## comment. Use # comments on lines where you think it might not be obvious what you're doing.

Handing In

Zip together your Python scripts plus the original or modified file that you got from the UCI Machine Learning Repository (i.e. the one that your script reads), and hand it in to the drop box.

Evaluation

Your code will be evaluated for Performance, Structure, and Documentation using the Rubric attached to the drop box.