COMP10200: Assignment 2 – Part 1

Overview

For this assignment, you will obtain some machine learning data for classification, test several versions of the K-Nearest Neighbour algorithm and a Decision Tree learner using that data, and then write a short report or record a short video presentation on your findings.

This is Part 1, in which you code and test your own version of K-Nearest Neighbour.

The Data

You can use the same data set as Assignment 1b, and you can recycle the code you wrote to read and split the data into training and testing sets. You can also feel free to get a new data set from the UCI Machine Learning Repository (see Assignment 1b for instructions on how to get data) or from somewhere else. If you are unsure whether your data is appropriate for this assignment, check with your instructor.

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

The Code

The code you use for this part of the assignment should be written in Python, should be created entirely by you, and should take maximum advantage of the array operations supported by Numpy. You can use other libraries and modules, but the k-NN code must be written by you from scratch.

The Task

Your task is to test at least 8 different versions of the k-NN algorithm to see how they perform on your data set. The different versions should be formed by picking 3 parameters (k value, distance metric, normalization, etc.) and then identifying at least 2 different values for each parameter. Then try all permutations of the two values for the 3 parameters (2^3 = 8 different versions). This is a minimum – feel free to try more versions than the basic 8, or to start with the basic 8 and then explore further.

When the program runs, you should perform many runs of each of the 8 versions and report the average result. Each run should use a different training/testing split, and you should perform as many runs of each version as is feasible (if you have a very large, slow data set, try to do at least 5 runs, but if the code runs faster, try for 50 or even more).

The reason for using results averaged over multiple runs is to get a true estimate of how the learner will perform in the wild. The accuracy for individual runs can vary quite a lot depending on what data ended up in the training or testing set. By averaging over 5, 50, or even 1000 runs, you will get much more stable results that should not vary by much each time you run the program.

The Report

Imagine that your boss asked you to test the k-NN algorithm to find a good configuration for this data set. You should write a short report, using word processing software, that contains the following sections:

1. **Data Description** (data set name, source, description of features, description of classification task, plus some statistics – number of features, number of items in each class, range of each feature, etc.)
2. **Description of the Tests** (describe the 8 or more versions of k-NN you experimented with, describe how you approached generating a different training/testing for each run, etc.)
3. **Results** (accuracy for at least 5 runs for all 8 versions, plus average accuracy for each version, and at least one graph summarizing your results)
4. **Discussion and Recommendation** (are there clear winners or losers? Which of the 3 variations seemed to make the most difference on your data set. Give some solid ideas for why some versions might be better or not as good than others, with reference to the characteristics of your data set. End with a recommendation for your boss – which configuration would be best if we were to use k-NN?)
5. **Future Work** (If you had more time, what other variations of k-NN would you like to explore?)

Throughout your report, make sure you are using standard grammar and spelling, and make sure you make proper use of correct machine learning technology wherever appropriate.

If you quote or reference any information about k-NN or issues in Machine Learning that were not explicitly covered in class, you should cite the source for that information using correct APA format.

See Canvas for an example report template that you can use.

Report Option: A Video Presentation

If you would prefer, you could also record a video presentation of your results and submit that instead, along with a handout or slides show the results. Your video should include all the information that would be in the report (see items 1 through 5 above). The handout or slides should show all the information from part 3 above (you can use that part of the report template) and any references for quotes or info that you used that wasn't covered in class (in APA format). When you get to part 3 of the report, you can just refer the viewer to the handout or slides, you don't have to read them out.

The content of your presentation will be judged using the same rubric as the report would be, just replace the phrase "well written" with "well presented". Make sure you're using correct machine learning technology wherever appropriate.

Note: If you are taking the video presentation option, you should probably wait until both parts of the assignment are complete before recording, then combine the two presentations into one.

Part 2 Preview

In part 2, you will use SKLearn to create and test a decision tree learner using the same data set. You will be asked to expand your report to include results from the decision tree learner as well.

Handing In

Wait until you have completed both parts of the assignment before handing in. Then zip up your report, handout, or slides along with your data file(s), and the code for all the versions of kNN you used. It should be possible for me to unzip your folder and run your code easily (i.e. without having to move files around or make changes to the code). If necessary, include instructions at the top of each code file with the correct command to run the code. If you made a video, you could host it as a private video on youtube and post the link in the assignment folder.

See the drop box for the exact due date.

Evaluation

This assignment will be evaluated based on: 1. the quality of the report/video you produce; 2. how well you met the requirements of the assignment; and 3. the quality of the code you handed in (including quality of documentation and referencing within the code).

See the rubric in the drop box for more information.