Computer Science 3202/6915
Assignment 2 – Cross-validation and performance curves

Goal
Getting familiar with different scikit-learn functions to perform cross-validation (CV) and being able to plot and interpret ROC and PR curves.

Due date
Sunday February 14th by 11:30pm.

Specifications
Your program should be called A2_CV.py and it should run in Linux. You program should take one command-line argument: A filename specifying a tab-delimited plain-text file containing the data.

For example, we should be able to execute your program as follows:

```
$python3 A2_CV.py data.txt
```

where the $ indicates the terminal prompt.

The data you will use is provided in Brigthspace (filename is A2_training_dataset.tsv). This is a tab-delimited text file containing 3817 observations. The file has neither a header nor instance IDs. The last column is the class. There are 71 attributes, and two classes. Attributes 5 and 7 are categorical and all other attributes are quantitative. The classification task is the one described at the end of Lecture 3.

Functionality
Your program should do the following:

1. Read the command-line arguments: name of the input file.
2. Read the input data. You can assume the input file is in the working directory.
3. Perform grid-search CV using the scikit-learn function GridSearchCV to find the optimal hyper-parameter setting for KNN. Use the scikit-learn function KNeighborsClassifier. You need to optimize at least two KNN hyper-parameters. You can choose from number of neighbors (n_neighbors), weights, and distance metric (metric). You also need to select the performance metric to optimize from the ones listed here: https://scikit-learn.org/stable/modules/model_evaluation.html#scoring-parameter
4. From grid-search CV obtain the best hyper-parameter setting and print a table with the grid scores.
5. Using the best hyper-parameter setting, perform K-fold CV and plot ROC and PR curves for the K splits and the average CV curve using the scikit-learn functions plot_roc_curve and plot_precision_recall_curve. The curves must include the AUROC or the average precision score, and the curve for the random classifier. See: https://scikit-learn.org/stable/auto_examples/model_selection/plot_roc_crossval.html for a sample ROC curve.

Submission
Submit through Brightspace the following (one submission per team):

a) Your python code in a single file called A2_CV.py

b) A PDF file containing:
   1. a brief description and justification of the grid and performance metric you used for grid-search CV,
   2. the best hyper-parameter setting found,
   3. the CV grid scores,
   4. the ROC and PR curves (make sure that your figures have axis labels and captions),
   5. brief answers to the questions given in the next section,
   6. an acknowledgement section listing your collaborations and online sources, and
   7. a program specification section listing the Python version and libraries you used.

Questions to answer in your submission:

1. Did you normalize some of the attributes? Why? If yes, which attributes?
2. Would it make sense to expand your grid (i.e, explore other values for the hyper-parameters)? Why? If yes, which values would you include?
3. Did you use stratified cross-validation? Why?
4. Looking at the ROC and PR curves, where would you recommend to have the threshold for predicting positives (you can indicate the point(s) using coordinates referring to your plots)? Why?
5. Which graphical representation of performance (ROC or PR curve) is more suitable for this task?

Common pitfalls to avoid (i.e., DO NOT do the following or points will be deducted):

1. Submit files in a compressed file.
2. Hardcode filename of the input or characteristics of the input (e.g., number of observations).
3. Forget to test your program in linux (i.e., your program fails to run in linux).
4. Fail to include some of the sections in the PDF file.
5. Fail to have axis labels and captions in your figures.
6. Forget to acknowledge a collaboration or source.
7. Fail to follow specifications.
8. Modify the input data (i.e., your program fails to run with the original data).

Online examples/tutorials:

- https://scikit-learn.org/stable/getting_started.html
- https://scikit-learn.org/stable/auto_examples/model_selection/plot_grid_search_digits.html#sphx-glr-auto-examples-model-selection-plot-grid-search-digits-py
- https://scikit-learn.org/stable/modules/cross_validation.html#cross-validation-iterators
- https://machinelearningmastery.com/k-fold-cross-validation/
- https://towardsdatascience.com/complete-guide-to-pythons-cross-validation-with-examples-a9676b5cac12