Computer Science 6915 - Winter 2020
Assignment 2

Task 1.
Modify your program from assignment 1 so that it performs grid-search k-fold cross-validation (CV). Your program needs to find the number of neighbours (K) and the distance function that maximizes the average Spearman correlation of the KNN classifier. Use the same dataset as the one provided for assignment 1.

You need to implement grid-search k-fold CV from scratch, and you need to select at least among three values of K and two distance functions.

Your program should be called TF_KNN_CV.py and it should run in Linux. You program should take two command-line arguments (given in the following order):

1. A filename specifying a tab-delimited plain-text file containing the attributes (X) of the training data (i.e., a string per TF representing the protein sequence of each TF). The file has two columns: the name of the TFs and the sequence of the TFs.
2. A filename specifying a tab-delimited plain-text file containing the output (Y) of the training data. Each column corresponds to the output vector of one TF. Each row corresponds to a specific string of eight characters representing a unique DNA sequence. This file contains a header row indicating the name of the TF.

We might execute your program as follows:
$python3 TF_KNN_CV.py X_train.txt Y_train.txt

where the $ indicates the terminal prompt. Your program should create a tab-delimited text plain-text file called model_selection_table.txt containing the average Spearman correlation plus/minus standard deviation for each combination of K and distance function. The last line of the output file should indicate the model chosen. For example, a sample output may look like this:

```
     D1         D2
K=3  .6±.1      .5±.2
K=5  .7±.15     .65±.12
K=7  .75±.1     .7±.21
Model chosen: K=7, D1
```

For this task, submit through D2L the following (one submission per team):

a) Your python code in a single file called TF_KNN_CV.py
b) A two-page description of your grid-search k-fold CV implementation including pseudo-code, description of distance functions assessed, and graphical representations of performance of the KNN model chosen in comparison with alternative models considered (e.g., models using different number of neighbours). This description has to be submitted as a PDF file.

This task will be graded based on the correctness of the grid-search k-fold CV implementation, performance of the model selected, and clarity of the description.

Computer Science 6915 - Winter 2020
Assignment 2

Task 2.
Implement from scratch a Python function to calculate precision-recall pairs for different confidence thresholds to be used to plot a precision-recall (PR) curve (the functionality of this function is very similar to that of the sklearn.metrics.precision_recall_curve function). Use this function in a Python script that takes a tab-delimited text file with confidence values in the first column and the actual class (Y) on the second column (a sample input file A2_T2_input.txt is available in D2L), use your function to calculate the precision-recall pairs and then plot the PR curve. Your program should create a tab-delimited text file called PR_table.txt with the precision-recall pairs and corresponding threshold, and a PNG or JPG file called PRC.png with the corresponding PR curve.

We might execute your program as follows:
$python3 PRC_calculator.py A2_T2_input.txt

For this task, submit through D2L the following (one submission per team):
a) Your python code in a single file called PRC_calculator.py

This task will be graded based on the correctness of the performance metrics calculation and plotting of the PR curve.

For a bonus point in the assignment: Calculate the AUPRC.