Computer Science 6915 - Winter 2019
Assignment 2

Task 1.
Implement a Python3 function that calculates for a multi-class classifier (number of classes > 2) the following classification performance metrics: the overall accuracy and, for each class: precision, recall, specificity and FDR. You should also write a Python program that 1) reads a multi-class confusion matrix given as a tab-delimited text file, and 2) outputs in the standard output (i.e., the terminal), the overall accuracy and then for each class, the class label and its performance metrics (one class per line, the class label and performance metrics should be separated by tabs). Your program should run in Linux and take one command-line argument: the name of the file containing the confusion matrix. You can assume that the True Class is given in the columns and the Predicted Class is given in the rows of the confusion matrix.

For example, a sample input may look like this:

```
      C1    C2    C3
C1    50    20    15
C2    25    45    10
C3    12    20    60
```

The corresponding output looks like this:

```
Ac    0.60
      P     R     Sp    FDR
C1    0.59  0.57  0.79  0.41
C2    0.56  0.53  0.80  0.44
C3    0.65  0.71  0.81  0.35
```

For this task, you need to submit through D2L your team's Python program called A2_t1.py
This task will be graded based on the correctness of the performance metrics calculation.

Task 2.
Implement k-fold cross-validation (k-fold CV) from scratch in Python3. Use your implementation of k-fold CV to choose the number of neighbours and features to obtain the best KNN model for the dataset provided in D2L (A2_t2_dataset.tsv). For KNN, you can use the KNN classifier you wrote for Assignment 1 or you can use the KNN classifier provided by scikit learn (https://scikit-learn.org/stable/modules/neighbors.html). The complete program should take one command-line argument: the name of the file with the data. The program should output the details (number of neighbours and features) of the model chosen and its performance metrics. The data is given in a tab-delimited plain-text format with the last column indicating the class of the instance (1 = positive, 0 = negative).

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

a) Your python code (including your implementation of KNN if applicable) in a single file called A2_t2.py
b) A two-page description of your k-fold CV implementation including pseudo-code, loss function used to choose the model, and graphical representations of performance of the KNN model chosen in comparison with alternative models considered (e.g., models using the same features but a 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 k-fold CV implementation, performance of the model selected, and clarity of the description.