**Project 1 CSCE 5380 Data Mining**

The purpose of this project is to examine the performance of two rule-based classifiers, namely the Decision Tree and PART in terms of their *stability*. Stability refers to how well a classifier copes with changes in data distribution. A classifier whose accuracy remains within a narrow range under data distributional changes is said to be more stable than one whose accuracy fluctuates within a wider band of values.

The project may be undertaken on an individual basis or in group of size 2.

The choice of classifiers was influenced by the fact that they both produce actionable rules that could be used for decision making. They are two good examples of data mining algorithms as their models produce knowledge that could be interpreted and used in a straightforward manner without the need for any further processing.

The project will require you to perform data pre-processing that is needed for each dataset that will be used in this project. Thereafter, you will be required to build models for each of the two classifiers and optimize them for accuracy. Finally, you will experiment with different data distributions and then compare the two classifiers in terms of stability of their model accuracy. The following tasks detail the requirements of this project.

Task 1 (15 marks)

Use the Breast Cancer ([UCI Machine Learning Repository: Breast Cancer Data Set](https://archive.ics.uci.edu/ml/datasets/Breast+Cancer)) and German Credit Card ([UCI Machine Learning Repository: Statlog (German Credit Data) Data Set](https://archive.ics.uci.edu/ml/datasets/Statlog+%28German+Credit+Data%29)) datasets for experimentation. For each of these datasets:

1. Describe (without using any form of code), what pre-processing operations you performed.
2. Develop the necessary code in R. In your Google Colab R program, be sure to include this code under a heading of “Task 1 code”.

Task 2 (20 marks)

1. Describe (without using any form of code), your choice of parameter values for each of the two classifiers.
2. Develop the code needed to *generate and optimize* models for the Decision Tree and PART classifiers.
3. For the decision tree classifier, visualize the tree obtained.
4. Generate the rules obtained for both types of classifiers.

Task 3 (20 marks)

In this task we will focus on testing the stability of the two classifiers. This will be done by partitioning the Breast cancer dataset into training and testing segments. Use 70% of data for training and the rest for testing. Fit a model to the training segment and then deploy the model on 100 different test segments and then record accuracy across each of the different test segments. This process will then be repeated for the other dataset. You will now have 100 pairs of accuracy values.

1. Develop the code required to implement the above requirements.
2. For each dataset, use the F test to compare the variances across the two groups (group 1 is the decision tree accuracy values across the 100 runs, similarly group 2 is the vector of accuracy values for PART). Develop the code for the F test.

Task 4 (30 marks)

In this task you will introduce changes in the data using your own change management mechanism rather than relying on random chance For each dataset and for each classifier, identify the numeric feature that is most highly correlated with the class feature, *and which is used by the classifier to build the model*. Now introduce 3 levels of change to this feature as follows. For each level l (10%, 20% and 30%) modify the feature value for each sample as follows: fv(new)=fv(old)+l\*fv(old) if fv(old)<=mean(fv), otherwise fv(new)=fv(old)-l\*fv(old).

1. Develop the code required to implement the above requirements.
2. For each dataset, use the F test as in Task 3 to compare the variances of the two classifiers.

Task 5 (15 marks)

Based on the results of your experimentation in this project, would you judge one classifier to be better than the other? Justify your answer.

End of project specification

**Note:**

All code must meet good programming practices such as naming variables, modularity (using functions for repetitive tasks), and adequate comments at key points in the code.

The deadline for submission of this project is 4 July 2022 at 11.59 pm in Canvas. The following documents need to be submitted:

Your code in a Google Colab sheet

A copy of your code in pdf form. This document must contain the name(s) of the persons who have undertaken the assignment. In addition, it must have a brief description of how the workload was distributed amongst the project partners if work was done in group mode.

A separate pdf document that provides written answers (not code) to the questions asked in this project specification.