# ELG 5255 Applied Machine Learning Summer 2021

# Assignment 1 (Multiclass Classification)

Start Date: *May* 14th 2021

Due Date: *May* 21st 2021 23:59 **Eastern Time (US and Canada)**

## Submission

You must submit your assignment on-line with Bright Space. This is the only method by which we accept assignment submissions. We do not accept assignments sent via email, and we are not able to enter a mark if the assignment is not submitted on Bright Space! The deadline date is firm since you cannot submit a assignment passed the deadline. It is a student's responsibility to ensure that the assignment has been submitted properly. A mark of 0 will be assigned to any missing assignment.

## Goal

This assignment aims to implement One-versus-Rest (OvR) strategy transforming multiclass classification problems to multiple binary classification problems.

## Dataset

During this assignment, the well-known Iris flower dataset is used to test OvR strategy.

In python:

```python
from sklearn.datasets import load_iris
iris = load_iris()
```

In matlab:

```matlab
load fisheriris;
```

Iris dataset contains 150 samples, 4 features (i.e., sepal length, sepal width, petal length, petal width), and 3 classes (i.e., Iris-Setosa, Iris-Versicolour, Iris-Virginica). In order to plot and have a better understanding about the dataset, the first 2 features would be dropped (i.e., sepal length and sepal width).

Before building OvR, the performance of binary classification models (Logistic Regression (LR) and Support Vector Machine (SVM)) should be compared; hence, the first class (Iris-Setosa) should be dropped to form a binary class dataset.

## One-versus-Rest

OvR involves training a binary class classifier for each class. During testing process, each classifier will predict the confidence of each class and OvR will select the one with highest confidence.

Training:

```pseudocode
Inputs: X, y, estimator
    yBin = binarize(y)
    build a list of estimators for each class
Output: a list of estimators
```

Prediction:

```pseudocode
Inputs: X, a list of estimators
    argmax of each estimator's confidence score on X
```

## What You Need to Do and Implement

Please note the difference among Iris dataset (with 4 features and 3 classes), 2D Iris dataset (with 2 features and 3 classes). Please do NOT separate the dataset to training and testing set. Don't use testing or validation set. Please only use training set.

Please submit your code and report (including screen shot of code and the relevant figures).

1. Load the Iris dataset

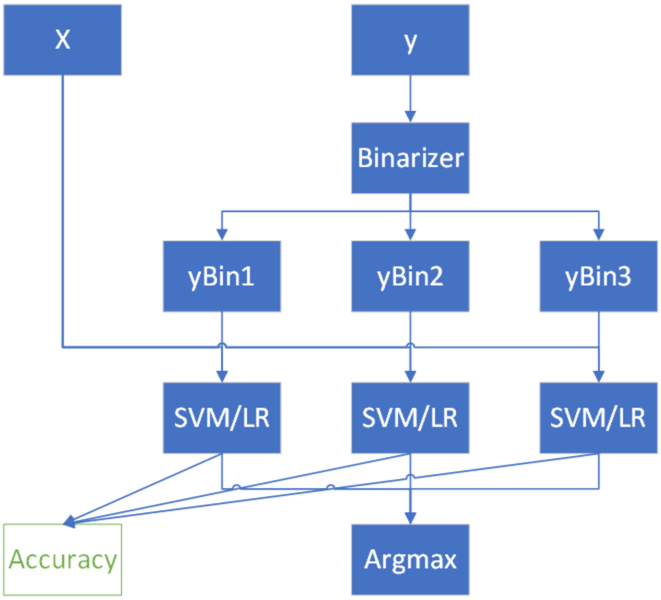
2. Drop the sepal length and sepal width features to form a 2D Iris dataset (**5 marks**)

3. Build OvR-LR and OvR-SVM and test on 2D Iris dataset (which contains 3 classes). For each class: (**20 marks for each class; 60 marks in total**)

    1. Obtain the binarized label (1 for positive class, -1 for negative class) (**4 marks**)
    2. Obtain the LR's confusion matrix and accuracy (**4 marks**)
    3. Obtain the SVM's confusion matrix and accuracy (**4 marks**)
    4. Plot LR's decision boundary (**4 marks**)
    5. Plot SVM's decision boundary (**4 marks**)

4. Use argmax to aggregate confidence scores and obtain the final label and obtain the performance (i.e., confusion matrix, accuracy, plotting correct and wrong prediction points) of OvR-LR (**10 marks**) and OvR-SVM (**10 marks**)

5. Implement alternative aggregation strategy instead of existing argmax function based OvR, using the third step's results (**10 marks**), obtain the performance (i.e., confusion matrix, accuracy, plotting correct and wrong prediction points) of your own strategy, and explain why your strategy is better/worse than existing OvR (**10 marks (5 marks as bonus)**)


X
y
Binarizer
yBin1
yBin2
yBin3
SVM/LR
SVM/LR
SVM/LR
Accuracy
Argmax