**Assignment 2 KNN implementation Machine learning**

Yuhao Liao 201474244

SeyedJavad KhataeiPour 201793290

## Pseudo-code of the whole program:

Let F total number of selected features

Let K total number of folds

Let N total number of neighbors for KNN implementation

1. Prepare the data
   1. Read the data and variables
   2. Shuffle rows
   3. Split the data to features and target datasets
   4. Sort the features based on importance, i.e. relative standard deviation
   5. Eliminate features based on their correlations, i.e. Unsupervised feature selection
   6. Scale features
2. For f from 1 to F
   1. For n from 1 to N
      1. Split the data pseudo randomly into K folds
      2. For k from 1 to K
         1. Set the k’th fold as test dataset and merge all other folds and set to train dataset
         2. Implement KNN with n neighbor
         3. Store the KNN accuracy in a 2D matrix
3. Calculated the average score by dividing the matrix by K
4. Find the maximum score and return its indices as best number of features and neighbors

Pseudo code for Cross Validation:

Split Part:

1. Create an empty 3-D array to store folds
2. Divide the data based on the class in to 2 classes, A and B
   1. For s from 1 to size of class A:
      1. For k from 1 to number of folds
         1. Send Class A instances, s, to k’th fold
   2. For s from 1 to size of class B
      * 1. Send Class B instances, s, to k’th fold

Merge part:

1. Merge all the folds but the selected one
2. Drop all NaN values
3. Return the selected fold as test set and the merged folds as training set

Loss function:

To compare performances between different selection of features and KNN parameters, we have used the accuracy given by KNN. We implement Scikit-Learn function and used its score. The scores for all folds are added and then the average is calculated and considered for comparison. All the accuracies are stored in a matrix and can be used to draw 2-D and 3-D plots.