# Introduction to Machine Learning (CS5710)
# Assignment-1 (80 points)
# K-NN
# Due by 17th September (Thursday) 11:59pm

*You are allowed to discuss the problem and solution design with others, but the code you submit must be your own. Your solution must include the certification of authenticity "I certify that the codes/answers of this assignment are entirely my own work."*

## Datasets

The training and test files will follow the same format as the text files in the UCI datasets. Datasets and description of the datasets (pendigits, satellite and yeast) are uploaded with this assignment onto Blackboard. For each dataset, a training file and a test file are provided. The name of each file indicates what dataset the file belongs to, and whether the file contains training or test data. Your code should also work with all the training and test files in the UCI datasets.

## Question: K-NN

In this task you will implement k-nearest neighbor classification. Your program will be invoked as follows:

*knn_classify* with following two command line arguments: *<training_file> <test_file>*

- *<training_file>:* The first argument, <training_file> is the path name of the training file, where the training data is stored. The path name can specify any file stored on the local computer.
- *<test_file>:* The second argument, <test_file> is the path name of the test file, where the test data is stored. The path name can specify any file stored on the local computer.

## Value of K

Run the program for K=1, 3, 5, 7, 9 for K-NN

## K-NN Distance Calculation

Use the L1 distance (the absolute distance) for computing the nearest neighbors.

## K-NN Classification (Testing) Stage

For each K and each test object you should print a line containing the following info:

- object ID. This is the line number where that object occurs in the test file. Start with 0 in numbering the objects, not with 1.
- predicted class (the result of the classification). If your classification result is a tie among two or more classes, choose one of them randomly.
- true class (from the last column of the test file).
- accuracy. This is defined as follows:

- If there were no ties in your classification result, and the predicted class is correct, the accuracy is 1.
- If there were no ties in your classification result, and the predicted class is incorrect, the accuracy is 0.
- If there were ties in your classification result, and the correct class was one of the classes that tied for best, the accuracy is 1 divided by the number of classes that tied for best.
- If there were ties in your classification result, and the correct class was NOT one of the classes that tied for best, the accuracy is 0.

After you have printed the results for all test objects, you should print the *overall classification accuracy*, which is defined as the average of the classification accuracies you printed out for each test object.

**Optimal K**

Find the optimal K from the list of K-numbers for K-NN

## Submission Guidelines and Requirements

- You need to use Python to write your code
- You can use either IDLE python or jupyter notebook to write your code
- Include your name, UCM ID and Certification statement in your solution:
  //Your name
  // Your UCM ID
  //Certificate of Authenticity: "I certify that the codes/answers of this assignment are entirely my own work."
- Add comments (about the function/variable/class) to your code as much as possible
- Zip your project including source file (knn_classify.py or knn_classify.ipynb) and input text data files (if any)
- Upload the zipped project file onto Blackboard
- **Grading standards:**
  - Compiles: 20%
  - Print the correct answers in the correct output format: 50%
  - Your code work with all the training and test files (with different number of features) in the UCI datasets: 20%
  - Your program takes three command line arguments: 5%
  - Is well commended and well designed: 5%
  - There will be automatic 10 points penalty if your program is missing certificate of authenticity
  - You will get 0 if your code matches with another student in your class or matches with any previous semester's submissions