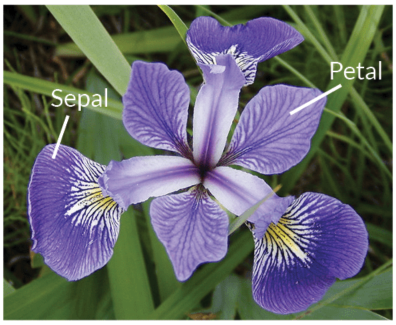
k Nearest Neighbors

In this assignment, you will implement a K-Nearest Neighbor classifier from scratch and use it on two popular datasets: *iris dataset* and *MNIST dataset*. You are **not** allowed to use sklearn KNN learner.

1. **Iris Dataset** (65 pts)
   First download the *iris dataset* from here. This is Iris flower species dataset and predict the flower species based on flower measurements.
   There are 150 observations with 4 attributes and a class label. Here's the description of data file:

First column:	sepal length in cm
Second column:	sepal width in cm
Third column:	petal length in cm
Fourth column:	petal width in cm
Fourth column:	Class label



Iris Versicolor

Iris Setosa

Iris Virginica

Before using the data you need to perform two simple tasks:

- Encode the class labels
- Split the dataset into train and test sets. Keep 20% of data for testing and the rest will be training data. Keep that in mind that in order to get good results, you need to make sure labels are distributed evenly in train and test data (*stratified sampling*). You can use methods provided by*sklearn* package, such as train-test-split from *sklearncross-validation*

Proceed to implement the k-nearest neighbors algorithm. Recall that this simple algorithm only requires the following steps:

- **Step1:** Calculate the distance from test data (Euclidean distance)
- **Step2:** Find the set I of k observations with smallest distances
- **Step3:** Assign a label by taking a majority vote on I

(a) Compare all four features distribution in each iris class using boxplots.

(b) Start with $k = 1$, plot the decision boundary using the first two features (Sepal length and width)

(c) Perform the prediction using $k = 2, 4, 6, 10, 15$ and plot the decision boundaries. How does the decision boundary change by increasing the number of neighbors?

(d) Use a new distance measure L_3 (Minkowski formula for p = 3), and redo the previous step. How does changing the distance function affect the classification?

(e) For all cases, report accuracy.

2. **MNIST Dataset** (35 pts)
   MNIST consists of handwritten digit images of all numbers from zero to nine. In MNIST, each image contains a single grayscale digit drawn by hand. And each image is a 784 dimensional vector (28 pixels for both height and width) of floating-point numbers where each value represents a pixel's brightness. The training set has 60000 examples and the test set has 10000 examples:

   Download the csv format of training data and test data.

   (a) Perform the 2-nearest neighbors on MNIST dataset using 500, 1000, 2500, 5000, 10000, 30000, and 60000 training examples. How does the classification error change with number of training example? plot it.
   (You can use 1000 test examples)

   (b) What is the confusion matrix of the best test?

   (c) Change the majority based voting with a method of your choosing. How does it affect the error rate using half of the training examples?

Rules

1. This is an **individual assignment**. It is not a group activity.
2. Please include some proper explanations for your results. Do not submit a notebook with code cells only. You need to properly describe your methods and discuss/analyze your observations.
3. We will look at the quality of your work for grading. You submission should be coherent and well documented.
4. There are many online discussions and demos on both of these datasets. It is okay if you look them up, but you must write your own code and analyze the data by yourself.
5. We will run your code though MOSS software to detect copying and plagiarism.

Submission

Submit everything through Gradescope and Blackboard. You will need to upload:

1. The Jupyter notebook all your work is in (.ipynb file) on Blackboard
2. PDF version of your Jupyter notebook on Gradescope

In order to make grading easier for your TA, please use the following format for naming your files:

- netid-hw3-418 { .pynb, .pdf }