# Assignment-1

In this assignment you will practice putting together a simple image classification pipeline, based on the k-Nearest Neighbor or the SVM/Softmax classifier. The goals of this assignment are as follows:

- understand the basic **Image Classification pipeline** and the data-driven approach (train/predict stages)
- understand the train/val/test **splits** and the use of validation data for **hyperparameter tuning**.
- develop proficiency in writing efficient **vectorized** code with numpy
- implement and apply a k-Nearest Neighbor (**kNN**) classifier
- implement and applay a binary class Support Vector Machine (**SVM**) classifier
- implement and apply a Multiclass Support Vector Machine (**Multi-Class SVM**) classifier
- implement and apply a **Softmax** classifier
- understand the differences and tradeoffs between these classifiers

**Installing Dependencies and Required Softwares:** To complete the assignment you will need a working ipython with required dependencies on your machine. To make the process easy you can simply download the Anaconda for your desired OS.

**Start IPython:** You should start the IPython notebook server from the `assignment1` directory.

```
ipython notebook
```

If you are unfamiliar with IPython, you should read the refersher on the IPython tutorial.

## Submitting your work:

Once you are done working produce a file called `assignment1.zip` and upload this file on the slate page for the course.

## Q1: k-Nearest Neighbor classifier (30 points)

The IPython Notebook **knn.ipynb** will walk you through implementing the kNN classifier.

## Q2: Training a Binary Support Vector Machine (10 points)

The IPython Notebook **Binary-SVM-Regularized-Unsolved.ipynb.ipynb** will walk you through implementing the binary-SVM classifier with gradient descent.

## Q3: Training a Multi-Class Support Vector Machine (30 points)

The IPython Notebook **svm.ipynb** will walk you through implementing the SVM classifier.

## Q4: Implement a Softmax classifier (30 points)

The IPython Notebook **softmax.ipynb** will walk you through implementing the Softmax classifier.