CS-482/682 Machine Learning

Introduction to Machine Learning and K-NN Classifier

In the first assignment, you will perform the tasks in two parts, Part 1 and Part 2. In Part 1, you will run the given tutorials and turn in the output. This is due Week 2 Friday at 11:59PM. The answers to Part 1 will be graded as Pass/Fail and you need a passing grade to get any points in Part 2.

Part 1 of the Assignment (DUE WEEK 2 FRIDAY BY 11:59)

a) Anaconda and Spyder (Read and execute each step and sign at the end stating that everything worked as described.)

b) Numpy Tutorial : Attach TWO files, one a .py file with code and another a text file or pdf containing the code and output of code. (the tutorial is attached)

c) Matplotlib: Attach TWO files, one a .py file with code and another a pdf file (created from word document) containing the code and output of code. (the tutorial is attached)

Once the parts a) b) and c) are done as stated above, then you will implement the first machine learning algorithm, i.e KNN and cross validate the results as follows.

Part 2 of the Assignment (DUE WEEK 3 FRIDAY AT 11:59PM)

You will turn in a report (pdf file) as described later in this document *in addition to* turning in the code (.py file) for the following tasks. Review chapters 1 and 2 from the class before starting this task.

1. **Dataset Selection:**
   Choose the following dataset. The data set must be in .csv format with headers containing feature names. If the data is not this format, you must edit the csv file to contain headers. If the data is in .data extension, make sure you open it using Microsoft Excel and save it as .csv file. Add headers to the columns in .csv file if they are missing.

   Breast Cancer Dataset with 2 classes
   https://archive.ics.uci.edu/ml/datasets/Breast+Cancer+Wisconsin+%28Diagnostic%29

Any columns that indicates UNIQUE IDENTIFIER of the data (a string or a int) should be deleted from the data as it will have no effect on target.

2. **Loading the data**: Write a function to load the data that is in the .csv format with headers return the following:

   data-numpy array of shape (n,p+1)[n samples and p predictors and 1 target]
   target – numpy array of shape (n,) [just the target column]
   target name: string
   feature_names : list of strings [The order of features must match with order of columns in the data

3. **Model Development and Training:** Split the data into 80% training and 20% as test values using train_test_split with *shuffle on and random state set to 42* for reproducibility. Perform k-NN algorithm with various number of neighbors varying from 1 to sqrt(n)+ 3, where n is the number of samples in your dataset. You must skip even number of neighbors in your check such due to possible tie if your problem is binary classification. If it is not binary classification, you can have even numbers as well.

   From the step above determine the "best" k for kNN using code. The 'best' is one for which the difference between training and test performance is smallest.

4. **Cross Validation:** Now using the best value of k found in the above step, use 5-fold cross validation with StratifiedKFold. This must also use 20% of the data as test set. Present a table of training and test values for each fold and present the mean accuracy as shown in the sample below. Is the training and test accuracy in Step 4 validated using cross validation? Why or why not?

Once the tasks above are complete you will then **write a report (pdf file created from word document)** using the results of the above task with the following sections.

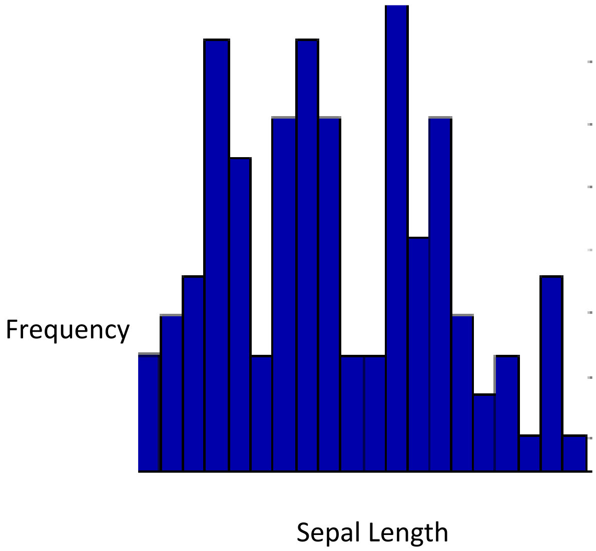
1. **Meet the Data Section:** Provide the following information about the data.

   a) Number of features (comes from the load function above)
   b) Names of the features (comes from load function above)
   c) Name of target (comes from load function above)
   d) Number of samples (comes from the load function above)
   e) First five rows of the data

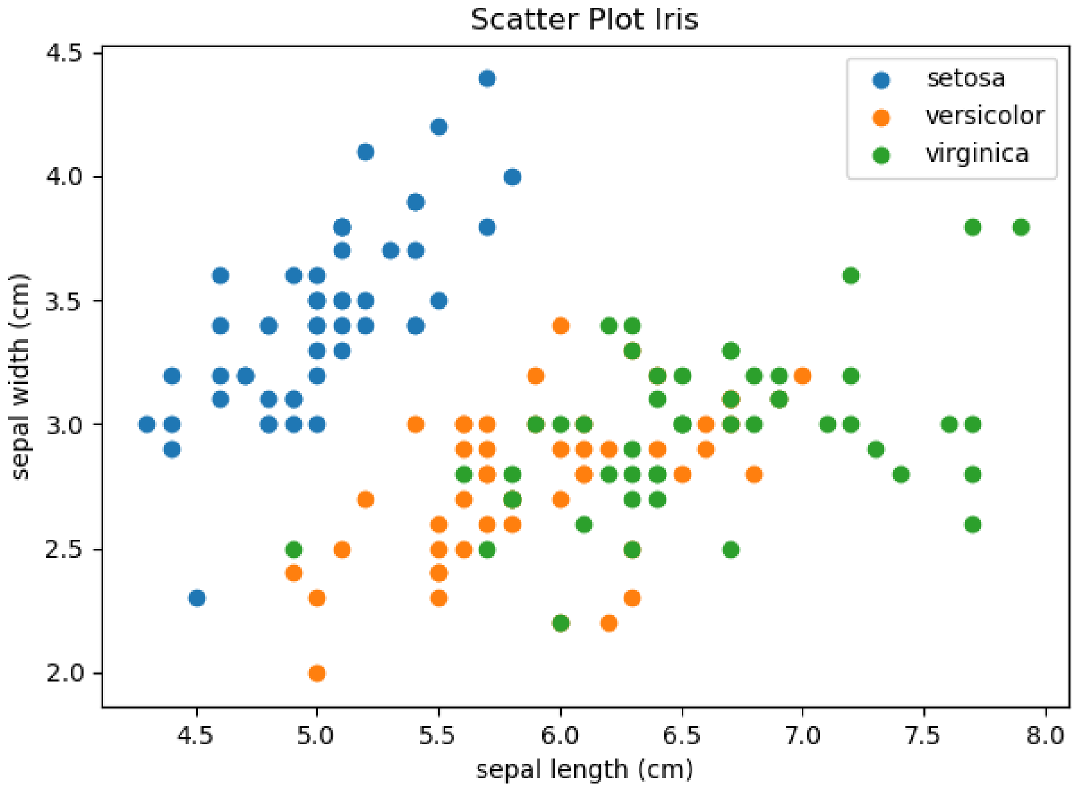
2. Learning from training data

Split the data into 80% training and 20% as test values using train_test_split with *shuffle on and random state set to 42* for reproducibility.

f) Histograms: Select first two features of the *training data* and display using matplotlib two histograms indicating how many samples contain a particular value of each the feature. Label your plots. Use blue color for your histogram. One histogram is shown below.



g) Pair Plot Target: Select 2 influential features from the training data and display scatter plot of the target values against these two features. Use different colors for each class of the target with a legend describing what each color represents.



3. Results of KNN performance

Show the plot similar to the one below and state the value of k that gives the best test accuracy. Do not use cross validation at this time

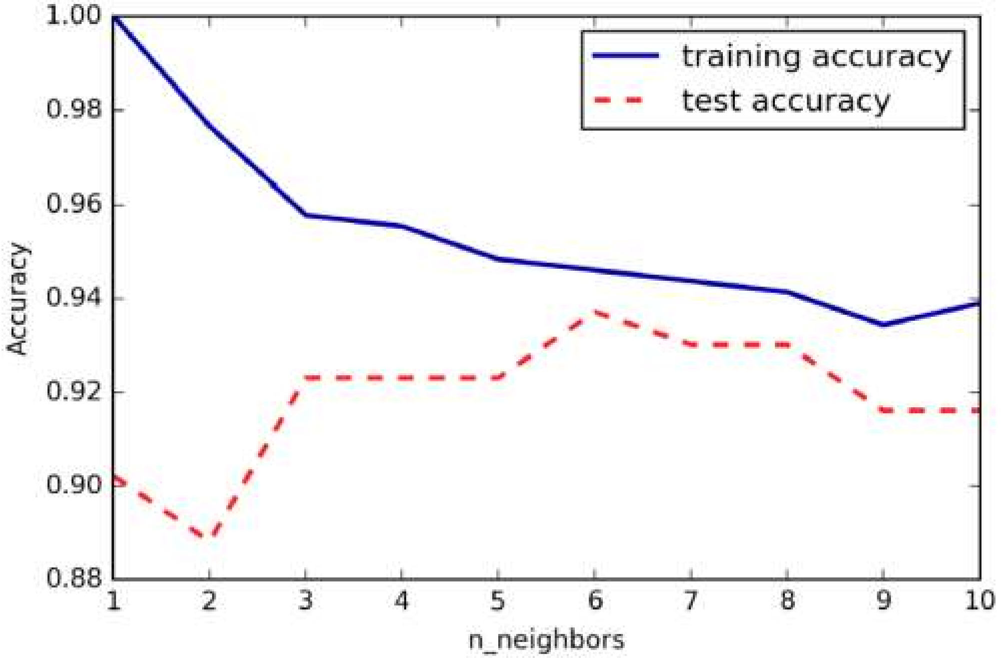


Figure 2-7. Comparison of training and test accuracy as a function of n_neighbors

Compute the best k value so the difference between test and training accuracy is the least.

4. **Results of cross validation**

Using the best of k for KNN, present the results of performance of 5 way cross validation.

	Fold-1	Fold-2	Fold-3	Fold-4	Fold-5	Mean
Training Accuracy						
Test Accuracy						