ECON 426 Data Mining

**Homework Assignment 1**

This homework includes two parts, for practicing python programming and mastering various visualization techniques (not including PCA), respectively.

**Coding Requirement** You will need to write your codes to accomplish the following question sets, one by one. Please implement all objectives in a single script file, or use a script file as the single entry – you can implement each objective in a separate function file. You will need submit your codes and make it self-contained.

**Part I: Python coding**

Please read carefully the provided tutorials for python programming and complete the following coding sets for various tasks. In your writeup, include the lines of codes and any results the code might generate (e.g., figures, tables, strings).

**Question 1.1** The following program is used to sort the list according to the second element in the sublist. Please complete the missing lines.

a=[['A',34],['B',21],['C',26]]

for i in range(0,len(a)):

for j in range(0,len(a)-i-1):

if(a[j][1]>a[j+1][1]):

temp=a[j]

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

print(a)

**Question 1.2** The following program is used to clean up the string and remove all the punctuation marks. Please complete the missing lines.

# define punctuation

punctuations = '''!()-[]{};:'"\,<>./?@#$%^&\*\_~'''

my\_str = "Hello!!!, he said ---and went."

# To take input from the user

# my\_str = input("Enter a string: ")

# remove punctuation from the string

no\_punct = ""

for char in my\_str:

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**# display the unpunctuated string**

**print(no\_punct)**

Question 1.3. Complete the following program to multiply two matrices using nested loops.

# Program to multiply two matrices using nested loops

# 3x3 matrix

X = [[12,7,3],

[4 ,5,6],

[7 ,8,9]]

# 3x4 matrix

Y = [[5,8,1,2],

[6,7,3,0],

[4,5,9,1]]

# result is 3x4

result = [[0,0,0,0],

[0,0,0,0],

[0,0,0,0]]

# iterate through rows of X

for i in range(len(X)):

# iterate through columns of Y

for j in range(len(Y[0])):

# iterate through rows of Y

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

for r in result:

print(r)

**Part II: Data Exploration**

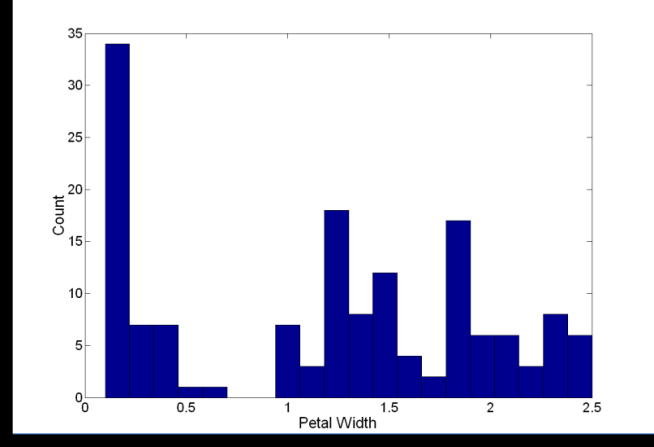
**Dataset**

In this project, you will work on the Iris Dataset that includes 100+ flower samples from three categories: setosa, virginica and versicolour. For each sample, there are four attributes, sepal width and length, petal width and length. The complete data and descriptions can be found online: <http://www.ics.uci.edu/~mlearn/MLRepository.html>.

This homework includes the iris.csv file.

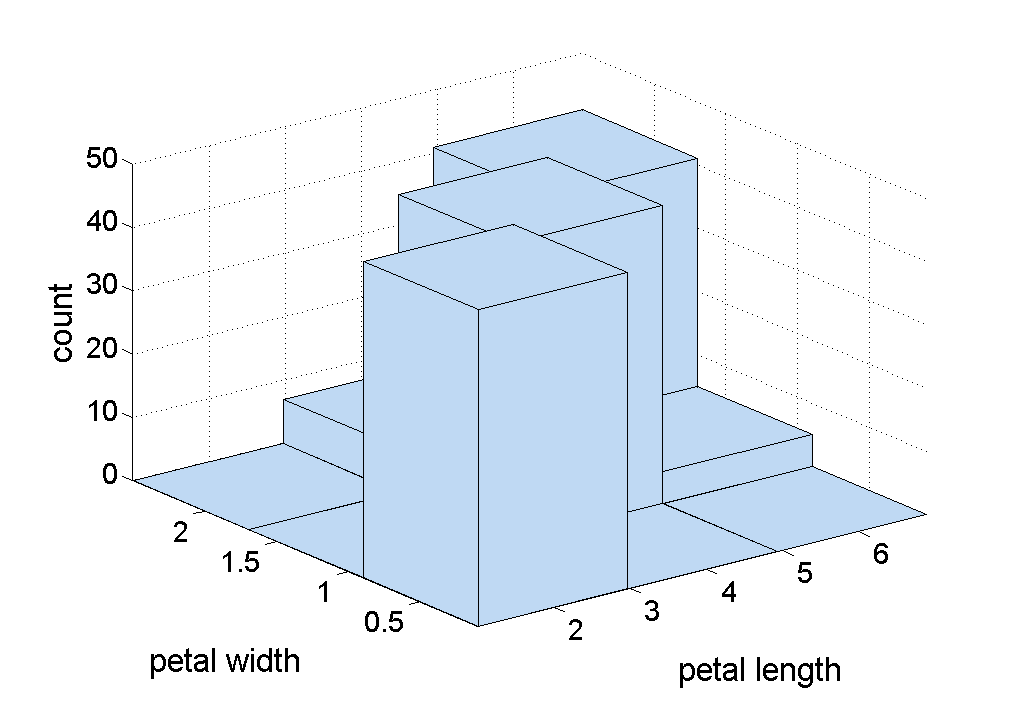
**Question Set**

2.1 For each of the three flower categories, please quantize the petal width into 10 bins, and account the frequency of each bin, i.e., generating a 1-D histogram. Compare the three histograms side by side and explain your observations. You may use the function hist() or write your codes to implement the above objective.



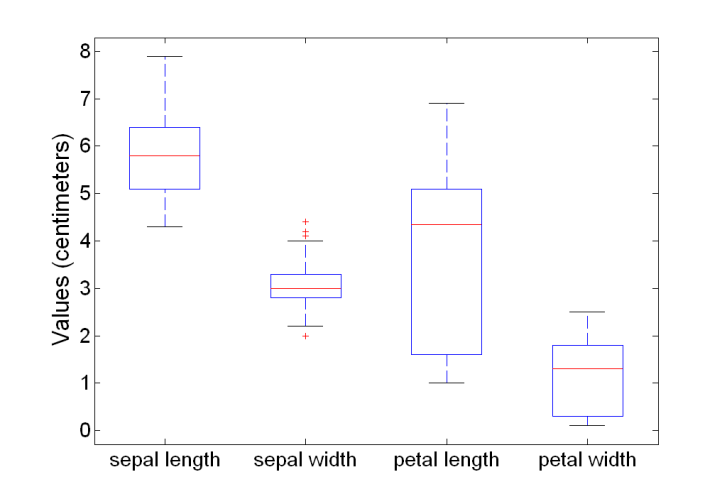
Extra-credit will be applied if you write your own function for extracting histograms, instead of using the built-in hist().

2.2 For each of the three categories, generate a 2D histogram as the following:



Compare all three 2D histograms side by side and explain your observations if any.

2.3 For each category, generate the BOX PLOTs for all four features, as the following.



You will visualize the three BOX-PLOT side by side.

2.4 Visualization of similarity matrix. Represent every sample with a four-dimension vector (sepal length, sepal width, petal length, petal width). For every two samples, compute their pair-wise similarity. You may do so using the Euclidean distance or other metrics. This leads to a similarity matrix where the element (i,j) stores the similarity between samples i and j. Please sort all samples so that samples from the same category appear together. Visualize the matrix using the function matplotlib.pyplot.imshow or other functions.

Extra-credit will be applied if you tried different types of similarity metrics.

**Submission instructions: what to hand in**

* Prepare a single PDF to include all required figures, analysis and results
* Submit your source codes (self-contained) and PDF file through the course site
* No hardcopy is needed.