**Exercise 4- First Step**

1. To use the functions we need, the following directories must be imported:

**import** math  
**import** csv

1. Define point tagged vector and 2 tagged vectors data1, data2 as follows:

point = [1, 0, 0, **'?'**] (an unknown tag)  
data1 = [1, 1, 1, **'M'**]

data2 = [1, 2, 0, **'F'**]

1. The two vectors data1, data2 must be printed separately from their tags so that the output looks like this:

The vector [1, 1, 1] has tag M

The vector [1, 2, 0] has tag F

1. Given a function that calculates the Euclidean distance between two vectors. It receives two vectors, and a length that determines where to read (so you can make sure the tag doesn't read). You must calculate and print the distance between data1 and data2.

**def** euclideanDistance(instance1, instance2, length):  
 distance = 0  
 **for** x **in** range(length):  
 *#print ('x is ' , x)* num1=float(instance1[x])

num2=float(instance2[x])  
 distance += pow(num1-num2, 2)  
 **return** math.sqrt(distance)

1. Enter code to read from myFile.csv file into list (you may need to

update the path).

**with** open(**'myFile.csv'**, **'r'**) **as** myCsvfile:  
 lines = csv.reader(myCsvfile)  
 dataWithHeader = list(lines)  
  
*#put data in dataset without header line*dataset = dataWithHeader[1:]

* 1. You must print the first 2 vectors of the file
  2. The distance between the above vectors must be calculated and printed

1. Suppose the first vector is untagged. We'll call him a point. You must create a list called eucDistances, which contains the point-to-point distance for any other vector in the file, along with the tagging of the vector.

The structure (object) will be:

**class** distClass:  
 dist = -1 *#distance of current point from test point* tag = **'-'** *#tag of current point*

The way to set up eucDistances would be:

eucDistances = [] *# list of distances, will hold objects of type distClass*

Helping code:

temp=dataset[1]  
label=temp[-1]  
d=euclideanDistance(point,temp,3)  
**print**(**"The distances between "** + str(point) + **" and "** + str(temp) + **" is "** + str(d))  
**print**(**" and the label is "** + label)  
obj = distClass() *#one record's distance and tag*obj.dist=d  
obj.tag=label  
eucDistances.append(obj)

1. You must sort the list you created in the previous section.

Hint: To sort a list of distClass objects named distList, if you want to

sort the list by distances, use the following code:

distList.sort(key=**lambda** x: x.dist)

1. Print all their vectors and distances from the first point.
2. What is the tag (tag) / category of the first record (point) with k = 1?

9. What is the tag (category) of the first record (point) with k = 3?