In [3]:
```
import csv
with open('iris.data.txt')as csvfile:
    lines=csv.reader(csvfile)
    for row in lines:
        print(','.join(row))
```

```
5.1,3.5,1.4,0.2,Iris-setosa
4.9,3.0,1.4,0.2,Iris-setosa
4.7,3.2,1.3,0.2,Iris-setosa
4.6,3.1,1.5,0.2,Iris-setosa
5.0,3.6,1.4,0.2,Iris-setosa
5.4,3.9,1.7,0.4,Iris-setosa
4.6,3.4,1.4,0.3,Iris-setosa
5.0,3.4,1.5,0.2,Iris-setosa
4.4,2.9,1.4,0.2,Iris-setosa
4.9,3.1,1.5,0.1,Iris-setosa
5.4,3.7,1.5,0.2,Iris-setosa
4.8,3.4,1.6,0.2,Iris-setosa
4.8,3.0,1.4,0.1,Iris-setosa
4.3,3.0,1.1,0.1,Iris-setosa
5.8,4.0,1.2,0.2,Iris-setosa
5.7,4.4,1.5,0.4,Iris-setosa
5.4,3.9,1.3,0.4,Iris-setosa
5.1,3.5,1.4,0.3,Iris-setosa
5.7,3.8,1.7,0.3,Iris-setosa
```

In [4]:
```
import csv
import random
def loadDataset(filename,split,trainingSet=[],testSet=[]):
    with open(filename,'r') as csvfile:
        lines=csv.reader(csvfile)
        dataset=list(lines)
        for x in range(len(dataset)-1):
            for y in range(4):
                dataset[x][y]=float(dataset[x][y])
            if random.random()<split:
                trainingSet.append(dataset[x])
            else:
                testSet.append(dataset[x])
```

In [5]:
```
trainingSet=[]
testSet=[]
loadDataset('iris.data.txt',0.66,trainingSet,testSet)
print('Train: '+ repr(len(trainingSet)))
print('Test: '+ repr(len(testSet)))
```

```
Train: 95
Test: 54
```

In [6]:
```
import math
def euclideanDistance(instance1,instance2,length):
    distance=0
    for x in range(length):
        distance+=pow((instance1[x]-instance2[x]),2)
    return math.sqrt(distance)
```

In [7]:
```
data1=[2,2,2,'a']
data2=[4,4,4,'b']
distance=euclideanDistance(data1,data2,3)
print('Distance: '+repr(distance))
```

```
Distance: 3.4641016151377544
```

In [8]:
```
import operator
def getNeighbors(trainingSet,testInstance,k):
    distances=[]
    length=len(testInstance)-1
    for x in range(len(trainingSet)):
        dist=euclideanDistance(testInstance,trainingSet[x],length)
        distances.append((trainingSet[x],dist))
    distances.sort(key=operator.itemgetter(1))
    neighbors=[]
    for x in range(k):
        neighbors.append(distances[x][0])
    return neighbors
```

In [9]:
```
trainSet=[[2,2,2,'a'],[4,4,4,'b']]
testInstance=[5,5,5]
k=1
neighbors=getNeighbors(trainSet,testInstance,k)
print(neighbors)
```

```
[[4, 4, 4, 'b']]
```

In [14]:
```
import operator
def getResponse(neighbors):
    classVotes={}
    for x in range(len(neighbors)):
        response=neighbors[x][-1]
        if response in classVotes:
            classVotes[response]+=1
        else:
            classVotes[response]=1
    sortedVotes=sorted(classVotes.items(),key=operator.itemgetter(1),reverse=T
    return sortedVotes[0][0]
```

In [15]:
```
neighbors=[[1,1,1,'a'],[2,2,2,'a'],[3,3,3,'a']]
response=getResponse(neighbors)
print(response)
```

```
a
```

In [16]:
```
def getAccuracy(testSet,predictions):
    correct=0
    for x in range(len(testSet)):
        if testSet[x][-1] is predictions[x]:
            correct+=1
    return (correct/float(len(testSet)))*100.0
```

In [19]:
```
testSet=[[1,1,1,'a'],[2,2,2,'a'],[3,3,3,'b']]
predictions=['a','a','a','a']
accuracy=getAccuracy(testSet,predictions)
print(accuracy)
```

```
66.66666666666666
```

In [21]:
```
split=0.67
trainingSet=[]
testSet=[]
loadDataset('iris.data.txt',split,trainingSet,testSet)
print( 'Train set: ' + repr(len(trainingSet)))
print( 'Test set: '+ repr(len(testSet)))
predictions=[]
k=3
for x in range(len(testSet)):
    neighbors=getNeighbors(trainingSet,testSet[x],k)
    result=getResponse(neighbors)
    predictions.append(result)
    print('> predicted=' + repr(result)+', actual='+ repr(testSet[x][-1]))
accuracy=getAccuracy(testSet,predictions)
print('Accuracy: '+repr(accuracy)+'%')
```

```
Train set: 90
Test set: 59
> predicted='Iris-setosa', actual='Iris-setosa'
> predicted='Iris-setosa', actual='Iris-setosa'
> predicted='Iris-setosa', actual='Iris-setosa'
> predicted='Iris-setosa', actual='Iris-setosa'
> predicted='Iris-setosa', actual='Iris-setosa'
> predicted='Iris-setosa', actual='Iris-setosa'
> predicted='Iris-setosa', actual='Iris-setosa'
> predicted='Iris-setosa', actual='Iris-setosa'
> predicted='Iris-setosa', actual='Iris-setosa'
> predicted='Iris-setosa', actual='Iris-setosa'
> predicted='Iris-setosa', actual='Iris-setosa'
> predicted='Iris-setosa', actual='Iris-setosa'
> predicted='Iris-setosa', actual='Iris-setosa'
> predicted='Iris-setosa', actual='Iris-setosa'
> predicted='Iris-setosa', actual='Iris-setosa'
> predicted='Iris-setosa', actual='Iris-setosa'
> predicted='Iris-setosa', actual='Iris-setosa'
> predicted='Iris-setosa', actual='Iris-setosa'
> predicted='Iris-setosa', actual='Iris-setosa'
> predicted='Iris-versicolor', actual='Iris-versicolor'
> predicted='Iris-versicolor', actual='Iris-versicolor'
> predicted='Iris-versicolor', actual='Iris-versicolor'
> predicted='Iris-versicolor', actual='Iris-versicolor'
> predicted='Iris-versicolor', actual='Iris-versicolor'
> predicted='Iris-versicolor', actual='Iris-versicolor'
> predicted='Iris-versicolor', actual='Iris-versicolor'
> predicted='Iris-versicolor', actual='Iris-versicolor'
> predicted='Iris-versicolor', actual='Iris-versicolor'
> predicted='Iris-versicolor', actual='Iris-versicolor'
> predicted='Iris-versicolor', actual='Iris-versicolor'
> predicted='Iris-versicolor', actual='Iris-versicolor'
> predicted='Iris-versicolor', actual='Iris-versicolor'
> predicted='Iris-versicolor', actual='Iris-versicolor'
> predicted='Iris-versicolor', actual='Iris-versicolor'
> predicted='Iris-virginica', actual='Iris-versicolor'
> predicted='Iris-versicolor', actual='Iris-versicolor'
> predicted='Iris-versicolor', actual='Iris-versicolor'
> predicted='Iris-versicolor', actual='Iris-versicolor'
> predicted='Iris-versicolor', actual='Iris-versicolor'
> predicted='Iris-versicolor', actual='Iris-versicolor'
> predicted='Iris-versicolor', actual='Iris-versicolor'
> predicted='Iris-virginica', actual='Iris-virginica'
> predicted='Iris-virginica', actual='Iris-virginica'
> predicted='Iris-virginica', actual='Iris-virginica'
> predicted='Iris-virginica', actual='Iris-virginica'
> predicted='Iris-versicolor', actual='Iris-virginica'
> predicted='Iris-virginica', actual='Iris-virginica'
> predicted='Iris-virginica', actual='Iris-virginica'
> predicted='Iris-virginica', actual='Iris-virginica'
> predicted='Iris-virginica', actual='Iris-virginica'
> predicted='Iris-virginica', actual='Iris-virginica'
> predicted='Iris-virginica', actual='Iris-virginica'
> predicted='Iris-virginica', actual='Iris-virginica'
```

```
> predicted='Iris-virginica', actual='Iris-virginica'
> predicted='Iris-virginica', actual='Iris-virginica'
> predicted='Iris-virginica', actual='Iris-virginica'
> predicted='Iris-virginica', actual='Iris-virginica'
Accuracy: 0.0%
```

In [ ]: