1. Handle Data

```
import csv
csvfile=open('iris.data.txt','r')
lines = csv.reader(csvfile)
for row in lines :
  print (', '.join(row))
```

```
6.1, 3.0, 4.6, 1.4, Iris-versicolor
5.8, 2.6, 4.0, 1.2, Iris-versicolor
5.0, 2.3, 3.3, 1.0, Iris-versicolor
5.6, 2.7, 4.2, 1.3, Iris-versicolor
5.7, 3.0, 4.2, 1.2, Iris-versicolor
5.7, 2.9, 4.2, 1.3, Iris-versicolor
6.2, 2.9, 4.3, 1.3, Iris-versicolor
5.1, 2.5, 3.0, 1.1, Iris-versicolor
5.7, 2.8, 4.1, 1.3, Iris-versicolor
6.3, 3.3, 6.0, 2.5, Iris-virginica
5.8, 2.7, 5.1, 1.9, Iris-virginica
7.1, 3.0, 5.9, 2.1, Iris-virginica
6.3, 2.9, 5.6, 1.8, Iris-virginica
6.5, 3.0, 5.8, 2.2, Iris-virginica
7.6, 3.0, 6.6, 2.1, Iris-virginica
4.9, 2.5, 4.5, 1.7, Iris-virginica
7.3, 2.9, 6.3, 1.8, Iris-virginica
6.7, 2.5, 5.8, 1.8, Iris-virginica
7.2, 3.6, 6.1, 2.5, Iris-virginica
6.5, 3.2, 5.1, 2.0, Iris-virginica
6.4, 2.7, 5.3, 1.9, Iris-virginica
6.8, 3.0, 5.5, 2.1, Iris-virginica
5.7, 2.5, 5.0, 2.0, Iris-virginica
5.8, 2.8, 5.1, 2.4, Iris-virginica
6.4, 3.2, 5.3, 2.3, Iris-virginica
6.5, 3.0, 5.5, 1.8, Iris-virginica
7.7, 3.8, 6.7, 2.2, Iris-virginica
7.7, 2.6, 6.9, 2.3, Iris-virginica
6.0, 2.2, 5.0, 1.5, Iris-virginica
6.9, 3.2, 5.7, 2.3, Iris-virginica
5.6, 2.8, 4.9, 2.0, Iris-virginica
7.7, 2.8, 6.7, 2.0, Iris-virginica
6.3, 2.7, 4.9, 1.8, Iris-virginica
6.7, 3.3, 5.7, 2.1, Iris-virginica
7.2, 3.2, 6.0, 1.8, Iris-virginica
6.2, 2.8, 4.8, 1.8, Iris-virginica
6.1, 3.0, 4.9, 1.8, Iris-virginica
6.4, 2.8, 5.6, 2.1, Iris-virginica
7.2, 3.0, 5.8, 1.6, Iris-virginica
7.4, 2.8, 6.1, 1.9, Iris-virginica
7.9, 3.8, 6.4, 2.0, Iris-virginica
6.4, 2.8, 5.6, 2.2, Iris-virginica
6.3, 2.8, 5.1, 1.5, Iris-virginica
6.1, 2.6, 5.6, 1.4, Iris-virginica
7.7, 3.0, 6.1, 2.3, Iris-virginica
6.3, 3.4, 5.6, 2.4, Iris-virginica
6.4, 3.1, 5.5, 1.8, Iris-virginica
6.0, 3.0, 4.8, 1.8, Iris-virginica
6.9, 3.1, 5.4, 2.1, Iris-virginica
6.7, 3.1, 5.6, 2.4, Iris-virginica
```

```
    6.7, 3.1, 5.6, 2.4, Iris-virginica
    6.9, 3.1, 5.1, 2.3, Iris-virginica
    5.8, 2.7, 5.1, 1.9, Iris-virginica
    6.8, 3.2, 5.9, 2.3, Iris-virginica
    6.7, 3.3, 5.7, 2.5, Iris-virginica
    6.7, 3.0, 5.2, 2.3, Iris-virginica
    6.3, 2.5, 5.0, 1.9, Iris-virginica
    6.5, 3.0, 5.2, 2.0, Iris-virginica
    6.2, 3.4, 5.4, 2.3, Iris-virginica
    5.9, 3.0, 5.1, 1.8, Iris-virginica
```

```
import csv
import random
def loadDataset(filename, split, trainingSet=[] , testSet=[]):
  csvfile=open('iris.data.txt', 'r')
  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])

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

```
    Train: 96
    Test: 53
```

2. Similarity

```
import math
from math import sqrt
def euclideanDistance(instance1, instance2, length):
  d=0
  for x in range(length):
    d=d+((instance1[x]-instance2[x])**2)
    return sqrt(d)

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

```
    Distance: 2.0
```

3. Neighbors

```
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

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

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

4. Response

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

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

```
---------------------------------------------------------------------------
```

5. Accuracy

```
     11 neighbors = [[1,1,1,'a'], [2,2,2,'a'], [3,3,3,'b']]
```

```
def getAccuracy(testSet, predictions):
  for x in range(len(testSet)):


  return (correct/float(len(testSet))) * 100.0
```

```
----> 9     return sortedVotes[0][0]
```

6. Main

[illegible]

SEARCH STACK OVERFLOW

0 s terminée à 21:59