In [1]:

```
import numpy as np
import pprint as pr
import matplotlib as mpl
import matplotlib.pyplot as plt

def file2matrix(filename):
    '''Function that takes a filename string as input a
    nd returns a nx3 matrix and a n-label vector '''

    fr = open(filename)
    numberOfLines = len(fr.readlines())         #find the no. of lines
    returnMat = np.zeros((numberOfLines,3))
    classLabelVector = []

    fr = open(filename)
    index = 0                           #start from line 0 in the list
    for line in fr.readlines():
        line = line.strip()             #Remove \n from every line
        listFromLine = line.split('\t')
        returnMat[index,:] = listFromLine[0:3]          #first three elements are
        classLabelVector.append(int(listFromLine[-1])) #last element in a line is label
        index += 1
    return returnMat,classLabelVector

dataSet,labelVector=file2matrix('datingTestSet2.txt')
```

In [17]:

```
# Unnormalized dataset
dataSet[0:10]
```

Out[17]:

```
array([[4.0920000e+04, 8.3269760e+00, 9.5395200e-01],
       [1.4488000e+04, 7.1534690e+00, 1.6739040e+00],
       [2.6052000e+04, 1.4418710e+00, 8.0512400e-01],
       [7.5136000e+04, 1.3147394e+01, 4.2896400e-01],
       [3.8344000e+04, 1.6697880e+00, 1.3429600e-01],
       [7.2993000e+04, 1.0141740e+01, 1.0329550e+00],
       [3.5948000e+04, 6.8307920e+00, 1.2131920e+00],
       [4.2666000e+04, 1.3276369e+01, 5.4388000e-01],
       [6.7497000e+04, 8.6315770e+00, 7.4927800e-01],
       [3.5483000e+04, 1.2273169e+01, 1.5080530e+00]])
```

In [2]:

```
minVals = dataSet.min(0)
print(minVals)
```

```
[0.       0.       0.001156]
```

In [3]:

```
maxVals = dataSet.max(0)
print(maxVals)
```

```
[9.1273000e+04 2.0919349e+01 1.6955170e+00]
```

In [4]:

```
ranges = maxVals - minVals
print(ranges)
```

```
[9.1273000e+04 2.0919349e+01 1.6943610e+00]
```

In [15]:

```
normDataSet = np.zeros(np.shape(dataSet))
normDataSet[0:10]
```

Out[15]:

```
array([[0., 0., 0.],
       [0., 0., 0.],
       [0., 0., 0.],
       [0., 0., 0.],
       [0., 0., 0.],
       [0., 0., 0.],
       [0., 0., 0.],
       [0., 0., 0.],
       [0., 0., 0.],
       [0., 0., 0.]])
```

In [9]:

```
m = dataSet.shape[0]
print(m) #no of entries
```

```
1000
```

In [12]:

```
# newValue = (oldValue-min)/(max-min)
normDataSet = dataSet - np.tile(minVals, (m,1))         # m-rows
normDataSet = normDataSet/np.tile(ranges, (m,1))
```

In [14]:

```
normDataSet[0:10] #FINAL NORMALIZED DATA
```

Out[14]:

```
array([[0.44832535, 0.39805139, 0.56233353],
       [0.15873259, 0.34195467, 0.98724416],
       [0.28542943, 0.06892523, 0.47449629],
       [0.82320073, 0.62848007, 0.25248929],
       [0.42010233, 0.07982027, 0.0785783 ],
       [0.79972171, 0.48480189, 0.60896055],
       [0.39385141, 0.32652986, 0.71533516],
       [0.46745478, 0.63464542, 0.32031191],
       [0.73950675, 0.41261212, 0.44153637],
       [0.38875681, 0.58668982, 0.88936006]])
```