Machine Learning Project Report

PROJECT 1

Name: Manishkumarreddy Jarugu
Person No.: 50206843

Data Import:

The data was given in excel format. We used the xlrd package for traversing the excel cells. Please find the code below for importing data from the excel sheet. The four rows are copied to the corresponding arrays.

```
import xlrd as xl
workbook=xl.open_workbook("university data.xlsx")
sheet=workbook.sheet_by_index(0)

for n in range(1, 50):
    b = np.array([[sheet.cell_value(n,2)]])
    CS_array1 = np.concatenate((CS_array1,b), axis=1)
    c = np.array([[sheet.cell_value(n,3)]])
    Research_array1 = np.concatenate((Research_array1, c), axis=1)
    d = np.array([[sheet.cell_value(n,4)]])
    Admin_array1 = np.concatenate((Admin_array1, d), axis=1)
    e = np.array([[sheet.cell_value(n,5)]])
    Tuition_array1 = np.concatenate((Tuition_array1, e), axis=1)
```

Problem 1:

The problem 1 consists of finding the mean, variance and standard deviation of the given data set for which the values are obtained using the numpy function as shown below.

```
import numpy as np
```

For mean:

```
mu1=np.around(np.mean(CS_array),3)
mu2=np.around(np.mean(Research_array),3)
mu3=np.around(np.mean(Admin_array),3)
mu4=np.around(np.mean(Tuition_array),3)
```

For Variance:

```
var1=np.around(np.var(CS_array),3)
var2=np.around(np.var(Research_array),3)
var3=np.around(np.var(Admin_array),3)
var4=np.around(np.var(Tuition_array),3)
```

For Standard Deviation:

```
sigma1=np.around(np.std(CS_array),3)
sigma2=np.around(np.std(Research_array),3)
sigma3=np.around(np.std(Admin_array),3)
sigma4=np.around(np.std(Tuition_array),3)
```

Problem 2:

The covariance and correlation of all the four variables are computed and displayed using a function in numpy.

```
X = np.vstack((CS_array,Research_array,Admin_array,Tuition_array))
XName=('CS_array','Research_array','Admin_array','Tuition_array')
covarianceMat=np.cov(X)
correlationMat=np.corrcoef(X)
```

For plotting the data points:

```
graphsno = 1

for i in range(0,3):
    for j in range(i+1,4):
        mapy.figure(graphsno)
        mapy.scatter(X[i],X[j],s=50)
        mapy.xlabel(XName[i])
        mapy.ylabel(XName[j])
        mapy.title(XName[i] + " Vs " + XName[j])
        graphsno += 1
mapy.show()
```

For finding the most correlated and least correlated points:

```
Considering correlations ignoring the diagonal and the upper triangle,
```

```
formaxmin=np.array(correlationMat)
for x, y in np.ndindex(formaxmin.shape):
    if not x > y:
        formaxmin[x][y] = None
    else:
        formaxmin[x][y] = np.abs(formaxmin[x][y])

maxi=np.nanmax(formaxmin)
mini=np.nanmin(formaxmin)

for x, y in np.ndindex(formaxmin.shape):
    if formaxmin[x][y]==maxi:
        print "Maximum correlation between "+XName[x]+" and "+XName[y]
    elif formaxmin[x][y] == mini:
        print "Minimum correlation between " + XName[x] + " and " +
XName[y]
```

Problem 3:

For log-likelihood we implemented the formula

$$\mathbf{L}(\mathbf{x}_1, ..\mathbf{x}_N) = \sum_{i=1}^{N} log\ p(\mathbf{x}_i)$$

Calculating the p(x_i) for each of the four variables separately using the Gaussian distribution probability density function and added them finally to get the log-likelihood.

$$p(x) = \frac{1}{\sqrt{2\pi}\sigma} \exp\left[-\frac{1}{2}\left(\frac{x-\mu}{\sigma}\right)^2\right]$$

Programmatically,

```
def loghood(a1,mu,sigma):
    a1=((1/(abs(sigma)*np.sqrt(2*np.pi)))*(np.exp((-0.5)*((a1-
mu)/abs(sigma))**2)))
    a1 = np.log(a1)
    return a1


P=np.vstack((loghood(CS_array,mu1,sigma1),loghood(Research_array,mu2,sigm
a2),loghood(Admin_array,mu3,sigma3),loghood(Tuition_array,mu4,sigma4)))
logLikelihood=np.around(np.sum(P),3)
```

Problem 4 & 5:

Using the correlation values, we constructed a Bayesian Network graph, which is an acyclic graph as shown below:

```
[[ 0  0  0  0]
 [ 1  0  0  0]
 [ 1  1  0  0]
 [ 1  1  1  0]]
```

The log-likelihood of the Bayesian network has been calculated using a series of formulas as shown below

First we construct the A matrix as

$$A = \begin{pmatrix} \sum_{n=1}^{N} x_0[n]x_0[n] & \sum_{n=1}^{N} x_1[n]x_0[n] & \dots & \sum_{n=1}^{N} x_k[n]x_0[n] \\ \sum_{n=1}^{N} x_0[n]x_1[n] & \sum_{n=1}^{N} x_1[n]x_1[n] & \dots & \sum_{n=1}^{N} x_k[n]x_1[n] \\ \vdots & \vdots & & \vdots \\ \sum_{n=1}^{N} x_0[n]x_k[n] & \sum_{n=1}^{N} x_1[n]x_k[n] & \dots & \sum_{n=1}^{N} x_k[n]x_k[n] \end{pmatrix}$$

Then the value of y is calculated as

$$\boldsymbol{y} = \begin{pmatrix} \sum_{n=1}^{N} y[n]x_0[n] \\ \sum_{n=1}^{N} y[n]x_1[n] \\ \vdots \\ \sum_{n=1}^{N} y[n]x_k[n] \end{pmatrix}.$$

From solving this equation, the value of B is obtained

B=A^{-1}y

And the value of σ^2 is obtained using the formula

$$\sigma^2 = \frac{1}{N}\sum_{n=1}^{N}\left(\beta_0 x_0[n] + \beta_1 x_1[n] + \dots + \beta_k x_k[n] - y[n]\right)^2$$

From which the log likely hood in Linear Gaussian model is represented as

$$\begin{aligned} L(\boldsymbol{\theta}) &= \log\left(P\left(Y|X_1, \dots, X_k\right)\right) \\ &= \sum_{n=1}^{N}\left[-\frac{1}{2}\log\left(2\pi\sigma^2\right) - \frac{1}{2\sigma^2}\left(\beta_0 x_0[n] + \beta_1 x_1[n] + \dots + \beta_k x_k[n] - y[n]\right)^2\right] \end{aligned}$$

And this can be further simplified to

$$L(\boldsymbol{\theta}) = -\frac{N}{2}\left[\log\left(2\pi\sigma^2\right) + 1\right]$$

Programatically,

```
def linearloglikelyhood(one,two):
    N=len(one)
    two.insert(0,[1]*N)
    K = len(two)
    A=[]
    for i in range(K):
        row=[]
        for j in range(K):
            ele=0
            for ik in range(N):
                ele+=two[i][ik]*two[j][ik]
            row.append(ele)
        A.append(row)

    B=[]
    for j in range(K):
        ele=0
        for ik in range(N):
            ele+=one[ik]*two[j][ik]
        B.append([ele])

    Beta=np.linalg.solve(A,B)

    finalsum=0
    for i in range(N):
        sums=0
        for j in range(K):
            sums+=Beta[j]*two[j][i]
        finalsum+=(sums - one[i])**2
    sigs=finalsum.item(0)/N

    logfinal=(-0.5)*N*(np.log(2*np.pi*sigs)+1)
    return logfinal

BNlogLikelihood=np.around(linearloglikelyhood(CS_array,[Research_array,Ad
min_array,Tuition_array])+linearloglikelyhood(Research_array,[Admin_array
,Tuition_array])+linearloglikelyhood(Admin_array,[Tuition_array])+linearl
oglikelyhood(Tuition_array,[]),3)
print "BNlogLikelihood ="
print BNlogLikelihood
```

This can be further repeated for different BN Graphs like(`BNlogLikelihood1)`

```
[[ 0  0  0  0]

 [ 0  0  0  0]

 [ 0  0  0  0]

 [ 0  0  0  0]]
```

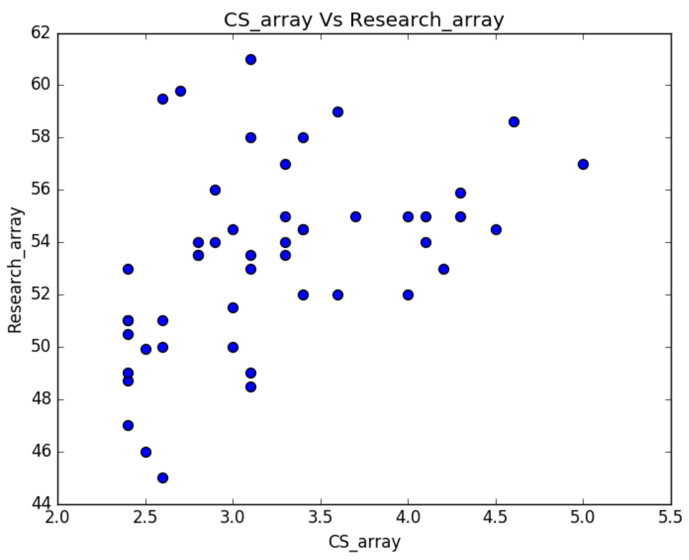
And considering only the top four correlations (`BNlogLikelihood2)`

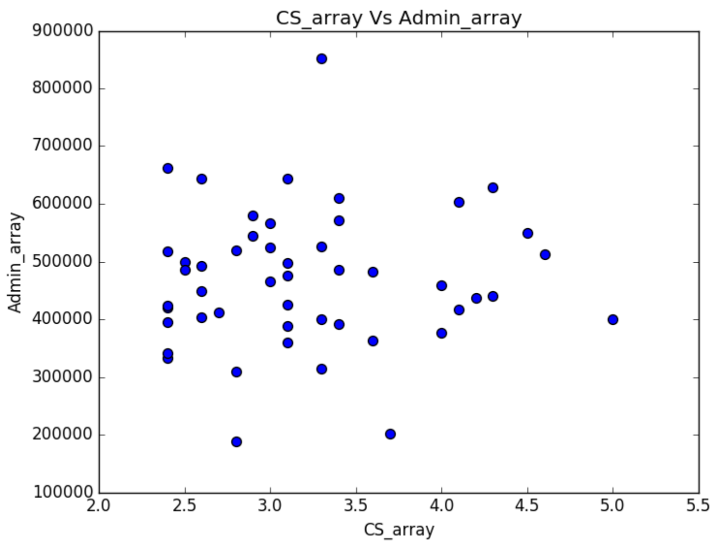
```
[[ 0  0  0  0]

 [ 1  0  0  0]

 [ 0  1  0  0]

 [ 1  0  1  0]]
```

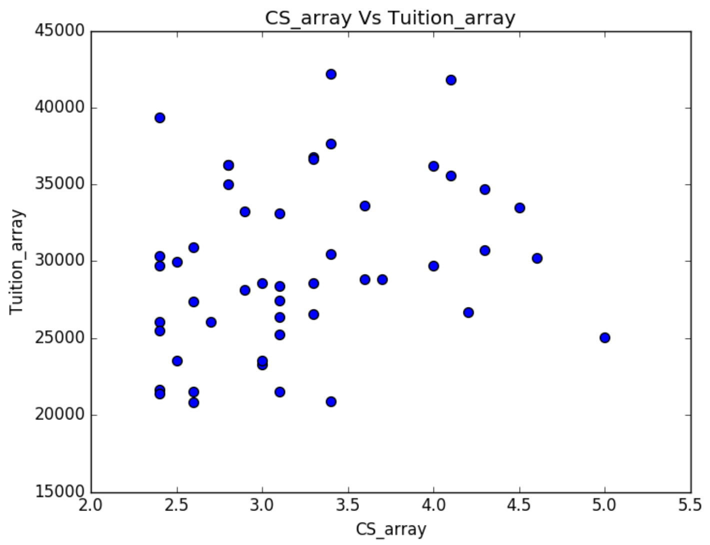
Outputs:

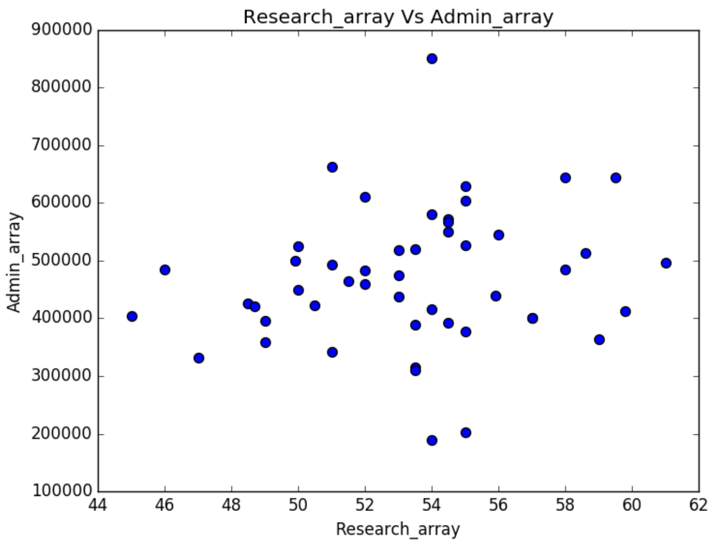
```
UBitName = mjarugu
personNumber = 50206843
mu1 = 3.214
mu2 = 53.386
mu3 = 469178.816
mu4 = 29711.959
var1 = 0.448
var2 = 12.588
var3 = 13900134681.7
var4 = 30727538.733
sigma1 = 0.669
sigma2 = 3.548
sigma3 = 117898.832
sigma4 = 5543.243
covarianceMat =
[[  4.57000000e-01   1.10600000e+00   3.87978200e+03   1.05848000e+03]
 [  1.10600000e+00   1.28500000e+01   7.02793760e+04   2.80578900e+03]
 [  3.87978200e+03   7.02793760e+04   1.41897208e+10  -1.63685641e+08]
 [  1.05848000e+03   2.80578900e+03  -1.63685641e+08   3.13676958e+07]]
correlationMat =
[[ 1.     0.456  0.048  0.279]
 [ 0.456  1.     0.165  0.14 ]
 [ 0.048  0.165  1.    -0.245]
 [ 0.279  0.14  -0.245  1.   ]]
Maximum correlation between Research_array and CS_array
Minimum correlation between Admin_array and CS_array
logLikelihood =
-1315.099
BNgraph =
[[0 0 0 0]
 [1 0 0 0]
 [1 1 0 0]
 [1 1 1 0]]
BNlogLikelihood =
-1304.758
The unconnected BN log likelyhood
-1315.099
The unconnected BN log likelyhood
-1305.681
```

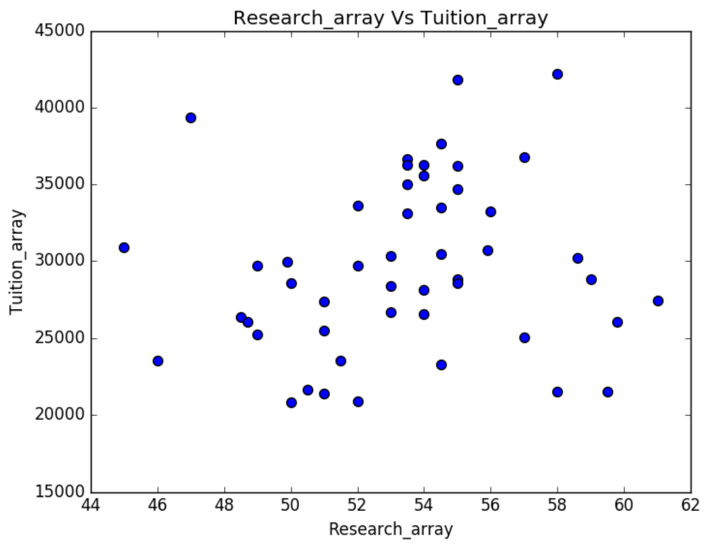
Graphs:

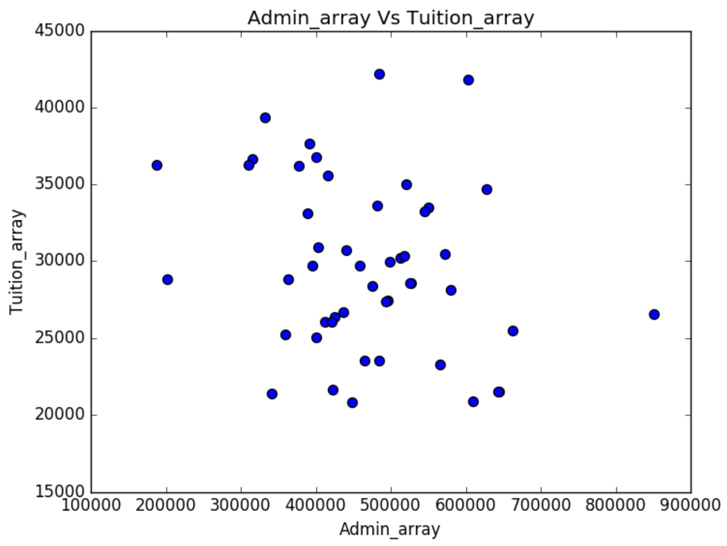

CS_array Vs Research_array
Research_array
CS_array
62
60
58
56
54
52
50
48
46
44
2.0
2.5
3.0
3.5
4.0
4.5
5.0
5.5


CS_array Vs Admin_array
Admin_array
CS_array
900000
800000
700000
600000
500000
400000
300000
200000
100000
2.0
2.5
3.0
3.5
4.0
4.5
5.0
5.5


CS_array Vs Tuition_array
45000
40000
35000
30000
25000
20000
15000
Tuition_array
2.0
2.5
3.0
3.5
4.0
4.5
5.0
5.5
CS_array


Research_array Vs Admin_array
900000
800000
700000
600000
500000
400000
300000
200000
100000
Admin_array
44
46
48
50
52
54
56
58
60
62
Research_array


Research_array Vs Tuition_array
Tuition_array
Research_array
45000
40000
35000
30000
25000
20000
15000
44
46
48
50
52
54
56
58
60
62


Admin_array Vs Tuition_array
Tuition_array
Admin_array
45000
40000
35000
30000
25000
20000
15000
100000
200000
300000
400000
500000
600000
700000
800000
900000