In [4]:

```
import pandas as pd
import numpy as np
from sklearn.model_selection import train_test_split
from sklearn import preprocessing
from sklearn.preprocessing import StandardScaler
from sklearn.model_selection import KFold
```

In [5]:

```

col_names=['x1','x2','result']
ds=pd.read_csv("/home/iignis/Downloads/studentt.csv",names=col_names)
```

In [6]:

```
print(ds)
x = ds.iloc[:,[0,1]].values
y = ds.iloc[:,2].values
```

```
           x1         x2  result
0   34.623660  78.024693       0
1   30.286711  43.894998       0
2   35.847409  72.902198       0
3   60.182599  86.308552       1
4   79.032736  75.344376       1
..        ...        ...     ...
95  83.489163  48.380286       1
96  42.261701  87.103851       1
97  99.315009  68.775409       1
98  55.340018  64.931938       1
99  74.775893  89.529813       1

[100 rows x 3 columns]
```

In [7]:

```
xp=preprocessing.scale(x)
kf=KFold(n_splits=5)
```

In [8]:

```
for train_index,test_index in kf.split(xp):
    xtrain,xtest,ytrain,ytest=train_test_split(xp,y,test_size=0.20,random_state=0
    x1=xtrain[:,0]
    x2=xtrain[:,1]
    b0=0.0
    b1=0.0
    b2=0.0
    epoch=100
    alpha=0.001
    while(epoch>0):
        for i in range(len(xtrain)):
            prediction=1/(1+np.exp(-(b0+b1*x1[i]+b2*x2[i])))
            b0=b0+alpha*(ytrain[i]-prediction)*prediction*(1-prediction)*1.0
            b1=b1+alpha*(ytrain[i]-prediction)*prediction*(1-prediction)*x1[i]
            b2=b2+alpha*(ytrain[i]-prediction)*prediction*(1-prediction)*x2[i]
        epoch=epoch-1
        print(b0)
        print(b1)
        print(b2)
```

```
0.40283747803219944
0.1389505439335714
0.39976783605883964
0.40604351640377373
0.13997369882457278
0.40289501313359166
0.40923220510884845
0.1409902209539282
0.40600527052913227
0.41240369822365686
0.14200017886087474
0.40909876441896814
0.41555814849022576
0.1430036404016529
0.41217564953288743
0.4186957073119175
0.14400067274983724
0.4152360791546616
0.4218165247499544
```

In [9]:

```
final_prediction=[]
x3=xtest[:,0]
x4=xtest[:,1]
print(ytest)
```

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

In [10]:

```


y_pred=[0]*len(xtest)
```

In [11]:

```
for i in range(len(xtest)):
    y_pred[i]=np.round(1/(1+np.exp(-(b0+b1*x3[i]+b2*x4[i]))))
```

In [12]:

```
final_prediction.append(np.ceil(y_pred[i]))
print(final_prediction)
```

```
[1.0]
```

In [13]:

```
from sklearn.metrics import accuracy_score
print("Accuracy",accuracy_score(ytest,y_pred))
```

```
Accuracy 0.8
```

In [ ]: