Simulating data:

```
#importing numpy and pandas
import random
import numpy as np
import pandas as pd
from scipy.stats import bernoulli
#opening file in write mode
file1 = open("myfile.csv","w")
#simulation of data
#here we
file1.write("no,gender,age,diabates,oxygen_level,covid_test_result\n")

for i in range(128):
    b=i+1
    a=random.randint(0,1)
    c=random.randint(12,80)
    d=np.random.normal(80,20)
    e=np.random.normal(95,2.25)
    f=bernoulli.rvs(p=0.3)
    file1.write(str(b)+","+str(a)+","+str(c)+","+str(d)+","+str(e)+","+str(f)+"\n")

text=open("myfile.csv","r")
print(text.read(1000))
print("success")
```

```
no,gender,age,diabates,oxygen_level,covid_test_result
1,1,64,107.37827291412023,92.80400861920597,0
2,1,15,85.30914453816627,100.30653125474808,1
3,1,49,102.52960614931592,95.98632162241087,0
4,1,80,79.99742342207229,93.91341744043042,0
5,1,21,106.04354550779513,95.0556068531797,1
6,0,47,116.17681278036278,95.74328790865383,0
7,1,25,109.39763079347216,93.2255830505931,0
8,1,26,107.91941529603767,96.67364171096546,0
9,1,75,89.22210464392325,96.01812885387163,0
10,0,70,81.62930364754043,92.37178636872265,0
11,0,60,65.1467981016061,92.04127632245735,0
12,1,17,59.45983400765493,97.99754385396382,0
13,0,55,98.827402017759,93.73565005917814,0
14,1,21,55.32516631158924,95.55529537104569,0
15,1,27,107.74568853800406,89.93901974786228,1
16,1,27,97.56540244343844,92.82020451339822,0
17,1,51,83.15769858770119,96.58277254576515,0
18,0,18,73.32795462759937,90.99269256161507,0
19,1,41,19.250354643719973,91.74234770206274,0
20,1,70,97.40423993058822,95.0169428196571,0
21,0,36,48.410807054762365,94.97
success
```

In [4]:

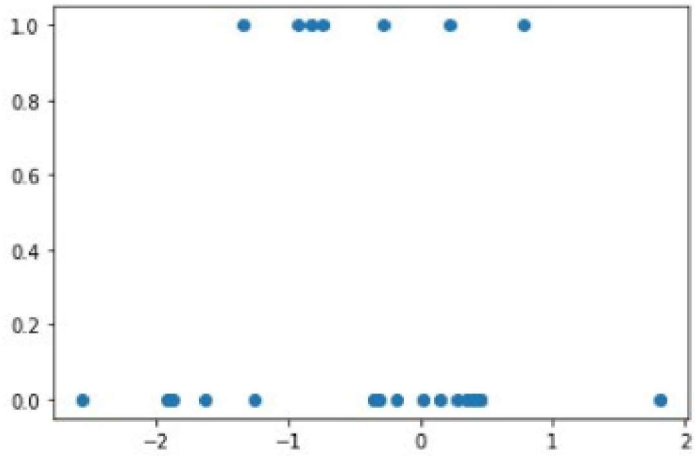
```
from sklearn.linear_model import LogisticRegression
from sklearn.model_selection import train_test_split
from sklearn.metrics import confusion_matrix
import matplotlib.pyplot as plt
from sklearn.discriminant_analysis import LinearDiscriminantAnalysis as LDA
import pandas as pd


#To read the dataset
df=pd.read_csv("myfile.csv")

#To know more about the dataset
print(df.describe())

#Define the independent and dependent variables
#dependent variable is Decision
y= df['covid_test_result']
x= df.drop(['covid_test_result'], axis=1)

# splitting the data
x_train, x_test, y_train, y_test = train_test_split(x,y, test_size= 0.2)

#Implementing Logistic Regression using sklearn
modelLogistic = LogisticRegression()
modelLogistic.fit(x_train,y_train)

#print the regression coefficients
print("The intercept b0= ", modelLogistic.intercept_)
print("The coefficient b1= ", modelLogistic.coef_)

#Make prediction for the test data
y_pred= modelLogistic.predict(x_test)

#Creating confusion matrix
ConfusionMatrix = confusion_matrix(y_test, y_pred)
print(ConfusionMatrix)

lda = LDA(n_components=1)
X_train = lda.fit_transform(x_train, y_train)
X_test = lda.transform(x_test)

plt.scatter(X_test,y_test)
plt.show()
```

```
                no      gender         age    diabates  oxygen_level  \
count  128.000000  128.000000  128.000000  128.000000    128.000000
mean    64.500000    0.531250   43.234375   82.048350     95.328219
std     37.094474    0.500983   21.297105   20.990092      2.349439
min      1.000000    0.000000   12.000000   35.472896     88.762674
25%     32.750000    0.000000   22.000000   68.053545     93.617270
50%     64.500000    1.000000   44.000000   83.375346     95.064723
75%     96.250000    1.000000   59.000000   95.443831     96.855583
max    128.000000    1.000000   80.000000  128.811930    101.302784
```

```
       covid_test_result
count         128.000000
mean            0.250000
std             0.434714
min             0.000000
```



```
25%             0.000000
50%             0.000000
75%             0.250000
max             1.000000
The intercept b0=  [-20.63978702]
The coefficient b1=  [[ 0.00591778 -0.51003829 -0.00993328  0.00293731  0.
20431669]]
[[19  0]
 [ 7  0]]
```



Now we simulate data for pool sample:

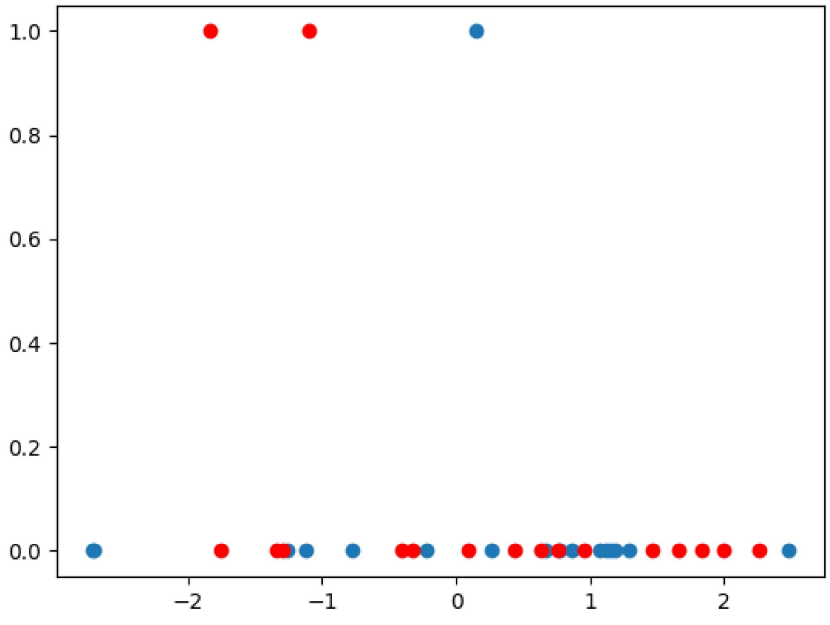
```
import random
import numpy as np
import pandas as pd
file1 = open("myfile.csv","w")


file1.write("no,gender,age,diabates,oxygen_level\n")

for i in range(126):
    b=i+1
    a=random.randint(0,1)
    c=random.randint(12,80)
    d=np.random.normal(80,20)
    e=np.random.normal(95,2.25)
    file1.write(str(b)+","+str(a)+","+str(c)+","+str(d)+","+str(e)+"\n")
print(success)
```

Here we've simulated data on the basis of the covariates used in previous one,but here we do not know if an individual is covid positive or not.

Then we divide the total data into three parts,(pools) and test each pools.

Testing the three pools we get two of them are positive,so we divide this two pools into another three or more pool and continue the testing.

Result:



which is similar to previous scatter plot. The two colors indicate data from two pools.

But in second one the cost is low.

Discussion:

Pools sample are useful if the disease prevalence is low and the dataset is too large. But in case of high rates of disease prevalence and low number of samples this method sometimes fails too meet it’s desired purpose of cost efficient.