Assignment no - 5

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
```

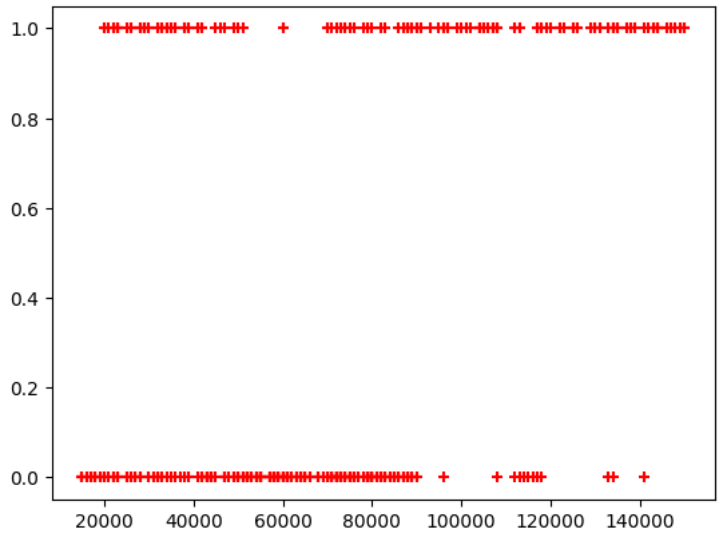
```
df = pd.read_csv("Social_Network_Ads.csv")
df
```

	User ID	Gender	Age	EstimatedSalary	Purchased
0	15624510	Male	19	19000	0
1	15810944	Male	35	20000	0
2	15668575	Female	26	43000	0
3	15603246	Female	27	57000	0
4	15804002	Male	19	76000	0
...	...	...	...	...	...
395	15691863	Female	46	41000	1
396	15706071	Male	51	23000	1
397	15654296	Female	50	20000	1
398	15755018	Male	36	33000	0
399	15594041	Female	49	36000	1

400 rows × 5 columns

```
plt.scatter(df.EstimatedSalary,df.Purchased,marker= '+',color='red')
```

```
<matplotlib.collections.PathCollection at 0x7efbf4164670>
```




```
# input
x = df[['Age','EstimatedSalary']]
y = df['Purchased']
```

```
from sklearn.model_selection import train_test_split
xtrain, xtest, ytrain, ytest = train_test_split(  x, y, test_size = 0.1,random_state = 0)
```

```
from sklearn.preprocessing import StandardScaler
sc_x = StandardScaler()
xtrain = sc_x.fit_transform(xtrain)
xtest = sc_x.transform(xtest)
print (xtrain[0:10, :])
```

```
[[-1.05714987  0.53420426]
 [ 0.2798728  -0.51764734]
 [-1.05714987  0.41733186]
 [-0.29313691 -1.45262654]
```

```
 [ 0.47087604  1.23543867]
 [-1.05714987 -0.34233874]
 [-0.10213368  0.30045946]
 [ 1.33039061  0.59264046]
 [-1.15265148 -1.16044554]
 [ 1.04388575  0.47576806]]
```

```
# check the lenth of taining
len(xtrain)
len(xtest)
```

```
40
```

```
# check the traing values of eacg element
xtrain
xtest
```

```
array([[-0.77064501,  0.50498616],
       [-0.00663206, -0.57608354],
       [-0.29313691,  0.15436896],
       [-0.77064501,  0.27124136],
       [-0.29313691, -0.57608354],
       [-1.05714987, -1.45262654],
       [-0.67514339, -1.59871705],
       [-0.19763529,  2.17041787],
       [-1.91666444, -0.05015774],
       [ 0.85288251, -0.78061024],
       [-0.77064501, -0.60530164],
       [-0.96164825, -0.42999304],
       [-0.10213368, -0.42999304],
       [ 0.08886956,  0.21280516],
       [-1.7256612 ,  0.47576806],
       [-0.57964177,  1.38152917],
       [-0.10213368,  0.21280516],
       [-1.82116282,  0.44654996],
       [ 1.61689547,  1.76136447],
       [-0.29313691, -1.39419034],
       [-0.29313691, -0.66373784],
       [ 0.85288251,  2.17041787],
       [ 0.2798728 , -0.54686544],
       [ 0.85288251,  1.03091197],
       [-1.43915634, -1.21888174],
       [ 1.04388575,  2.08276357],
       [-0.96164825,  0.50498616],
       [-0.86614663,  0.30045946],
       [-0.10213368, -0.22546634],
       [-0.57964177,  0.47576806],
       [-1.63015958,  0.53420426],
       [-0.10213368,  0.27124136],
       [ 1.8078987 , -0.28390254],
       [-0.10213368, -0.48842924],
       [-1.34365472, -0.34233874],
       [-1.91666444, -0.51764734],
       [-1.53465796,  0.32967756],
       [-0.38863853, -0.78061024],
       [-0.67514339, -1.04357314],
       [ 1.04388575, -0.98513694]])
```

```
# create a model of logistic regression
from sklearn.linear_model import LogisticRegression
model = LogisticRegression()
```

```
model.fit(xtrain,ytrain)
```

```
▾ LogisticRegression
LogisticRegression()
```

```
model.predict(xtest)
```

```
array([0, 0, 0, 0, 0, 0, 0, 1, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 1,
       0, 1, 0, 1, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 1])
```

```
model.score(xtest,ytest)
```

```
0.95
```