Data Analytics II

1. Implement logistic regression using Python/R to perform classification on Social_Network_Ads.csv dataset.
2. Compute Confusion matrix to find TP, FP, TN, FN, Accuracy, Error rate, Precision, Recall on the given dataset.

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

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

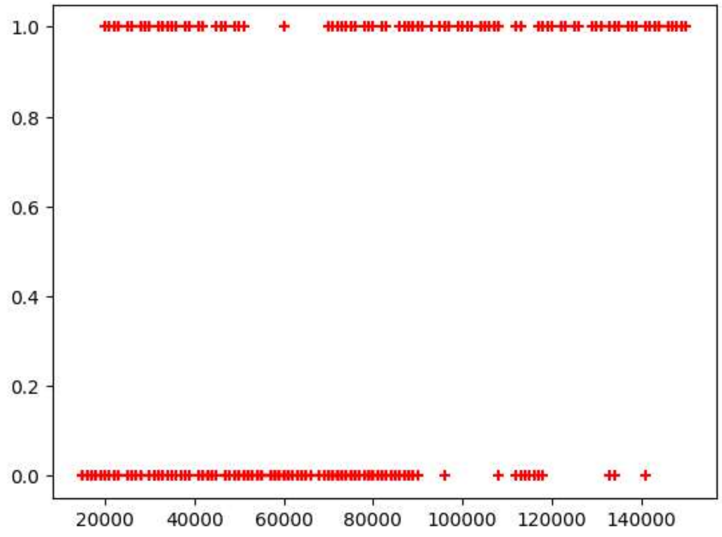
	User ID	Gender	Age	EstimatedSalary	Purchased
0	15624510	Male	19	19000	0
1	15810944	Male	35	20000	0
2	15668575	Female	26	43000	0
3	15603246	Female	27	57000	0
4	15804002	Male	19	76000	0
...	...	...	...	...	...
395	15691863	Female	46	41000	1
396	15706071	Male	51	23000	1
397	15654296	Female	50	20000	1
398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

400 rows × 5 columns

Now, to predict whether a user will purchase the product or not, one needs to find out the relationship between Age and Estimated Salary. Here User ID and Gender are not important factors for finding out this.

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

```
<matplotlib.collections.PathCollection at 0x7f768bb184f0>
```



```
x = df[['Age','EstimatedSalary']]
x
```

	Age	EstimatedSalary
0	19	19000
1	35	20000
2	26	43000
3	27	57000
4	19	76000
...	...	...
395	46	41000
396	51	23000
397	50	20000

```
y = df['Purchased']
y
```

```
0      0
1      0
2      0
3      0
4      0
      ..
395    1
396    1
397    1
398    0
399    1
Name: Purchased, Length: 400, dtype: int64
```

```
from sklearn.model_selection import train_test_split
xtrain, xtest, ytrain, ytest = train_test_split( x, y, test_size = 0.25,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, :])
```

```
[[ 0.58164944 -0.88670699]
 [-0.60673761  1.46173768]
 [-0.01254409 -0.5677824 ]
 [-0.60673761  1.89663484]
 [ 1.37390747 -1.40858358]
 [ 1.47293972  0.99784738]
 [ 0.08648817 -0.79972756]
 [-0.01254409 -0.24885782]
 [-0.21060859 -0.5677824 ]
 [-0.21060859 -0.19087153]]
```

Here once see that Age and Estimated salary features values are sacled and now there in the -1 to 1. Hence, each feature will contribute equally in decision making i.e. finalizing the hypothesis. Finally, we are training our Logistic Regression model.

```
from sklearn.linear_model import LogisticRegression
model = LogisticRegression()
```

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

```
▾ LogisticRegression
LogisticRegression()
```

```
y_pred = model.predict(xtest)
y_pred
```

```
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, 0, 0, 0, 0,
       1, 0, 0, 1, 0, 1, 1, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 1,
            0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 1, 0, 1, 1, 1, 1, 0, 0, 1, 1, 0, 1,
            0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 1, 1])
```

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

```
0.89
```

```
from sklearn.metrics import confusion_matrix
cm = confusion_matrix(ytest, y_pred)
```

```
print ("Confusion Matrix : \n", cm)
```

```
Confusion Matrix :
 [[65  3]
 [ 8 24]]
```

Out of 100 : TruePostive + TrueNegative = 65 + 24

FalsePositive + FalseNegative = 3 + 8

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

```
Accuracy :  0.89
```

To find the accuracy of a confusion matrix and all other metrics,