```
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.
```

In [41]:
```
import pandas as pd
import matplotlib.pyplot as plt
import numpy as np

from sklearn.metrics import confusion_matrix, ConfusionMatrixDisplay
df=pd.read_csv("/home/admin1/Social_Network_Ads1.csv")
df
```

Out[41]:

	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

In [11]:
```
df.shape
```

Out[11]:
```
(400, 5)
```

In [16]:
```
X = df.iloc[:, [2, 3]].values    #iloc to access all rows of column 2 and 3  (slice operation)
y = df.iloc[:, 4].values         #iloc to access all rows of column 4
print (X)
print(y)
```

In [25]:
```
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size = 0.25, random_state = 0)
print(X_train.shape)
print (X_test.shape)
print(y_train.shape)
print(y_test.shape)
```

```
(300, 2)
(100, 2)
(300,)
(100,)
```

In [36]:
```
from sklearn.preprocessing import StandardScaler
sc = StandardScaler()
X_train = sc.fit_transform(X_train)  #to standardize features by removing mean and scaling to unit
X_test = sc.transform(X_test)
```

In [37]:
```
from sklearn.svm import SVC
classifier = SVC(kernel='linear',random_state=0)
classifier.fit(X_train, y_train)
```

Out[37]:
```
SVC(kernel='linear', random_state=0)
```

In [46]:
```
y_pred = classifier.predict(X_test)
```

In [47]:
```
from sklearn.metrics import confusion_matrix
cm = confusion_matrix(y_test, y_pred)
# Display the Confusion matrix
cm
```

Out[47]:
```
array([[66,  2],
       [ 8, 24]])
```

In [50]:
```
import sklearn.metrics
acc= sklearn.metrics.accuracy_score(y_test, y_pred)
print("Accuracy : ", acc)
prec=sklearn.metrics.precision_score(y_test, y_pred)
print("Precision : ", prec)
rec=sklearn.metrics.recall_score(y_test, y_pred)
print("Recall : ", rec)
```

```
Accuracy :  0.9
Precision :  0.9230769230769231
Recall :  0.75
```