In [1]:

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

In [4]:

```
dataset = pd.read_csv(r"D:\College\TE\SEM-2\Practical\DSBDA\5\Social_Network_Ads.csv")
```

In [5]:

```
dataset.head()
```

Out[5]:

	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

In [6]:

```
dataset.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 400 entries, 0 to 399
Data columns (total 5 columns):
User ID            400 non-null int64
Gender             400 non-null object
Age                400 non-null int64
EstimatedSalary    400 non-null int64
Purchased          400 non-null int64
dtypes: int64(4), object(1)
memory usage: 15.8+ KB
```

In [7]:

```
dataset.isnull().sum()
```

Out[7]:

```
User ID            0
Gender             0
Age                0
EstimatedSalary    0
Purchased          0
dtype: int64
```

In [8]:

```
dataset.shape
```

Out[8]:

```
(400, 5)
```

In [10]:

```
x = dataset.iloc[:, [2, 3]].values
y = dataset.iloc[:, 4].values
```

```
In [11]: print(x)
```

```
[[    19  19000]
 [    35  20000]
 [    26  43000]
 [    27  57000]
 [    19  76000]
 [    27  58000]
 [    27  84000]
 [    32 150000]
 [    25  33000]
 [    35  65000]
 [    26  80000]
 [    26  52000]
 [    20  86000]
 [    32  18000]
 [    18  82000]
 [    29  80000]
 [    47  25000]
 [    45  26000]
 [    46  28000]
```

```
In [12]: from sklearn.model_selection import train_test_split
         x_train, x_test, y_train, y_test = train_test_split(x, y, test_size = 0.30, random_stat
```

```
In [13]: from sklearn.preprocessing import StandardScaler
         sc = StandardScaler()
         x_train = sc.fit_transform(x_train)
         x_test = sc.transform(x_test)
```

```
In [14]: from sklearn.linear_model import LogisticRegression
         classifier = LogisticRegression(random_state = 0)
         classifier.fit(x_train, y_train)
```

```
C:\Users\HP\Anaconda3\lib\site-packages\sklearn\linear_model\logistic.py:432: FutureWa
rning: Default solver will be changed to 'lbfgs' in 0.22. Specify a solver to silence
this warning.
  FutureWarning)
```

```
Out[14]: LogisticRegression(C=1.0, class_weight=None, dual=False, fit_intercept=True,
                            intercept_scaling=1, l1_ratio=None, max_iter=100,
                            multi_class='warn', n_jobs=None, penalty='l2',
                            random_state=0, solver='warn', tol=0.0001, verbose=0,
                            warm_start=False)
```

```
In [15]: y_pred = classifier.predict(x_test)
```

```
In [16]: print(y_pred)
```

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

In [17]:
```
print(y_test)
```

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

In [18]:
```
from sklearn.metrics import confusion_matrix
cm = confusion_matrix(y_test, y_pred)
print(cm)
```

```
[[74  5]
 [10 31]]
```

In [19]:
```
Accuracy=(74+31)/120
Accuracy
```

Out[19]:
```
0.875
```

In [20]:
```
Error_rate=(5+10)/120
Error_rate
```

Out[20]:
```
0.125
```

In [21]:
```
from sklearn.metrics import precision_score, recall_score
precision_score(y_test, y_pred)
```

Out[21]:
```
0.8611111111111112
```

In [22]:
```
recall_score(y_test, y_pred)
```

Out[22]:
```
0.7560975609756098
```

In [23]:
```
from sklearn.metrics import f1_score
f1_score(y_test, y_pred)
```

Out[23]:
```
0.8051948051948052
```

In [ ]:
```

```