```
from google.colab import files
uploaded = files.upload()
```

Choose Files   Social_Network_Ads.csv
- **Social_Network_Ads.csv**(text/csv) - 4903 bytes, last modified: 5/29/2023 - 100% done
  Saving Social_Network_Ads.csv to Social_Network_Ads.csv

```
import numpy as np
import pandas as pd
```

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

```
df
```

	Age	EstimatedSalary	Purchased
0	19	19000	0
1	35	20000	0
2	26	43000	0
3	27	57000	0
4	19	76000	0
...	...	...	...
395	46	41000	1
396	51	23000	1
397	50	20000	1
398	36	33000	0
399	49	36000	1

400 rows × 3 columns

```
df.isnull().sum()
```

```
Age                0
EstimatedSalary    0
Purchased          0
dtype: int64
```

```
x = df.iloc[:,0:2]
```

```
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
398	36	33000
399	49	36000

400 rows × 2 columns

```
y = df.iloc[:,-1]
```

```
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
```

```
x_train,x_test,y_train,y_test = train_test_split(x,y,test_size=0.2)
```

x_train

	Age	EstimatedSalary
319	36	60000
253	37	146000
152	31	76000
309	38	50000
286	37	62000
...	...	...
294	35	57000
41	33	51000
73	33	113000
201	49	74000
162	37	33000

320 rows × 2 columns

x_test

	Age	EstimatedSalary
364	42	104000
183	33	43000
128	30	17000
32	21	16000
163	35	38000
...	...	...
74	32	18000
119	41	59000
225	37	53000
271	59	76000
103	33	149000

80 rows × 2 columns

y_train

```
319     0
253     1
152     0
309     0
286     0
       ..
294     0
41      0
73      0
201     0
162     0
Name: Purchased, Length: 320, dtype: int64
```

y_test

```
    364    1
    183    0
    128    0
    32     0
    163    0
           ..
    74     0
    119    0
    225    0
    271    1
    103    1
    Name: Purchased, Length: 80, dtype: int64
```

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

```
model = lr.fit(x_train,y_train)
```

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

```
y_test
```

```
    364    1
    183    0
    128    0
    32     0
    163    0
           ..
    74     0
    119    0
    225    0
    271    1
    103    1
    Name: Purchased, Length: 80, dtype: int64
```

```
y_pred
```

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

```
from sklearn.metrics import mean_squared_error
mse = mean_squared_error(y_test,y_pred)
```

```
mse
```

```
    0.275
```

```
from sklearn.metrics import accuracy_score
accuracy = accuracy_score(y_test,y_pred)
```

```
accuracy
```

```
    0.725
```

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

```
cm
```

```
    array([[58,  0],
           [22,  0]])
```

```
import seaborn as sns
```

```
sns.heatmap(cm,annot=True)
```

<Axes: >


```python
from sklearn.metrics import precision_score
ps = precision_score(y_test,y_pred,average="micro")
```

```python
ps
```

    0.725

```python
from sklearn.metrics import recall_score
rs = recall_score(y_test,y_pred,average="micro")
```

```python
rs
```

    0.725