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

```
df=pd.read_csv('/content/Social_Network_Ads - Social_Network_Ads (1).csv')
```

```
df.head(4)
```

	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

Next steps: Generate code with df New interactive sheet

```
from sklearn.preprocessing import LabelEncoder
lb=LabelEncoder()
```

```
df['Gender']=lb.fit_transform(df['Gender'])
```

```
x=df.drop(columns=["Purchased"])
y=df['Purchased']
```

```
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,random_state=42)
```

```
from sklearn.preprocessing import StandardScaler
sc=StandardScaler()
```

```
x_train_new=sc.fit_transform(x_train)
```

```
x_test_new=sc.transform(x_test)
```

```
from sklearn.naive_bayes import GaussianNB
model=GaussianNB()
```

```
model.fit(x_train_new,y_train)
```

```
▾ GaussianNB
GaussianNB()
```

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

```
y_pred
```

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

```
from sklearn.metrics import accuracy_score,confusion_matrix,classification_report
```

```
cn=confusion_matrix(y_pred,y_test)
```

```
cn
```

```
array([[49,  4],
       [ 3, 24]])
```

```
from sklearn.metrics import precision_score
```

```
precision = precision_score(y_test, y_pred)
```

```
precision
```

```
0.8888888888888888
```

```
from sklearn.metrics import recall_score
```

```
recall = recall_score(y_test, y_pred)
```

```
recall
```

```
0.8571428571428571
```

```
from sklearn.metrics import f1_score
```

```
f1=f1_score(y_test, y_pred)
```

```
f1
```

```
0.8727272727272727
```

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

```
df=pd.read_csv("/content/credit_scoring - credit_scoring (1).....csv")
```

```
df.head(2)
```

	User ID	Gender	Age	EstimatedSalary	Purchased
0	15624510	1	19	19000	0
1	15810944	1	35	20000	0

Next steps: Generate code with df New interactive sheet

```
x=df.drop(columns=["User ID"])
y=df['User ID']
```

```
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,random_state=42)
```

```
from sklearn.preprocessing import StandardScaler
sc=StandardScaler()
```

```
x_train_new=sc.fit_transform(x_train)
```

```
x_test_new=sc.transform(x_test)
```

```
from sklearn.naive_bayes import GaussianNB
model=GaussianNB()
```

```
model.fit(x_train_new,y_train)
```

```
▾ GaussianNB
GaussianNB()
```

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

```
y_pred
```

```
array([15810075, 15569641, 15579781, 15780572, 15673367, 15595324,
       15733973, 15744919, 15582066, 15769902, 15725794, 15759684,
       15762228, 15662067, 15783029, 15570932, 15759684, 15617877,
       15668385, 15814553, 15614420, 15595917, 15733883, 15569641,
       15741094, 15707634, 15649668, 15723373, 15654230, 15662067,
       15748589, 15767871, 15694879, 15602373, 15589449, 15776844,
       15723373, 15631070, 15745232, 15638963, 15807837, 15762228,
       15578006, 15705113, 15694946, 15815236, 15675185, 15595324,
       15807837, 15709441, 15745083, 15780572, 15750335, 15745083,
       15753874, 15706185, 15774744, 15636428, 15602373, 15595917,
       15800215, 15575002, 15780572, 15631912, 15675185, 15598840,
       15721007, 15761950, 15697020, 15789109, 15772073, 15746139,
       15718071, 15679760, 15807837, 15619087, 15708791, 15663249,
       15709441, 15594762])
```

```
from sklearn.metrics import accuracy_score,confusion_matrix,classification_report
```

```
cn=confusion_matrix(y_pred,y_test)
```

```
cn
```

```
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]])
```

```
from sklearn.metrics import precision_score
```

```
precision = precision_score(y_test, y_pred,average='micro')
```

```
from sklearn.metrics import recall_score
```

```
recall = recall_score(y_test, y_pred,average='micro')
```

```
f1=f1_score(y_test, y_pred,average='micro')
```

```
f1
```

```
0.0
```