ASSIGNMENT 5

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 [69]:

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LogisticRegression
from sklearn.preprocessing import StandardScaler
from sklearn.metrics import confusion_matrix,ConfusionMatrixDisplay,classification_
```

In [70]:

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

In [71]:

```
df.head()
```

Out[71]:

	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 [72]:

```
df.info()
```

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

In [73]:

```
df.shape
```

Out[73]:

```
(400, 5)
```

In [74]:

```
df.describe()
```

Out[74]:

	User ID	Age	EstimatedSalary	Purchased
count	4.000000e+02	400.000000	400.000000	400.000000
mean	1.569154e+07	37.655000	69742.500000	0.357500
std	7.165832e+04	10.482877	34096.960282	0.479864
min	1.556669e+07	18.000000	15000.000000	0.000000
25%	1.562676e+07	29.750000	43000.000000	0.000000
50%	1.569434e+07	37.000000	70000.000000	0.000000
75%	1.575036e+07	46.000000	88000.000000	1.000000
max	1.581524e+07	60.000000	150000.000000	1.000000

In [75]:

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

Out[75]:

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

DATA PREPARATION

In [76]:

```
x = df[["Age", "EstimatedSalary"]]
y = df["Purchased"]
```

In [77]:

```
scaler = StandardScaler()
x = scaler.fit_transform(x)
```

In [78]:

```
x_train, x_test, y_train, y_test = train_test_split(x, y, test_size=0.3, random_sta
```

In [79]:

```
x_train.shape, x_test.shape, y_train.shape, y_test.shape
```

Out[79]:

```
((280, 2), (120, 2), (280,), (120,))
```

Model building

In [80]:

```
model = LogisticRegression(n_jobs=-1)
```

In [81]:

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

Out[81]:

```
▾ LogisticRegression
LogisticRegression(n_jobs=-1)
```

In [82]:

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

Confusion Matrix

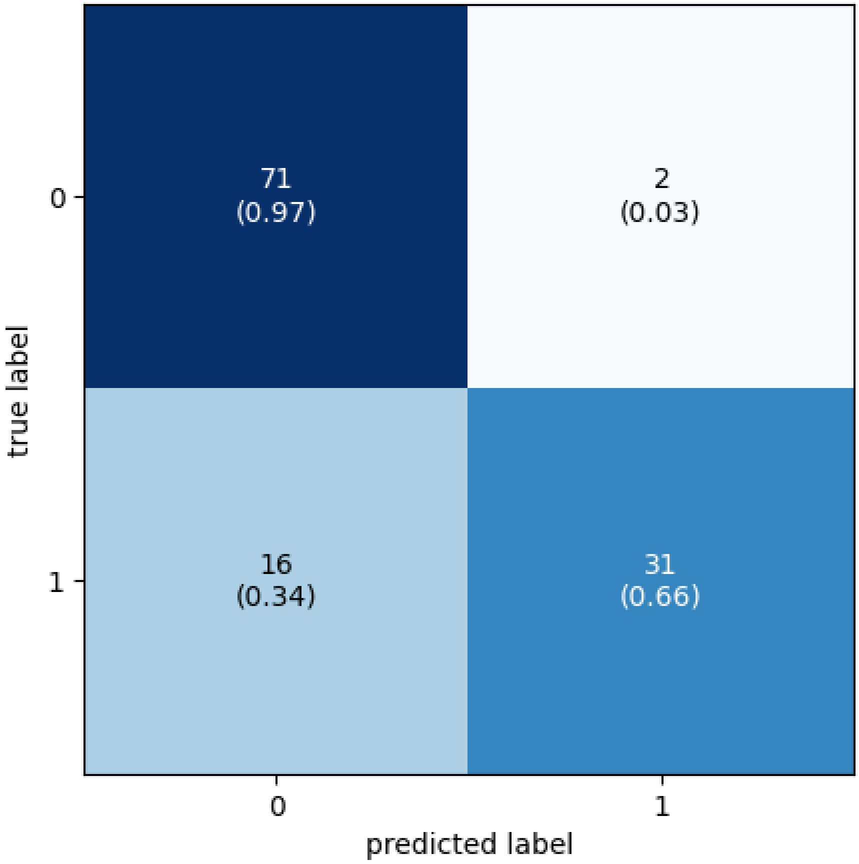
In [83]:

```
cm = confusion_matrix(y_test, y_pred)
print(cm)
```

```
[[71  2]
 [16 31]]
```

In [84]:

```
plot_confusion_matrix(conf_mat=cm, figsize=(5,5), show_normed=True)
plt.show()
```



In [85]:

```
print(f"TN value is {cm[0][0]}")
print(f"FP value is {cm[0][1]}")
print(f"FN value is {cm[1][0]}")
print(f"TP value is {cm[1][1]}")
```

```
TN value is 71
FP value is 2
FN value is 16
TP value is 31
```

In [86]:

```
tn=71
fp=2
fn=16
tp=31
```

In [87]:

```
print('Accuracy score is :',(tn+tp)/(tn+fp+fn+tp))
```

```
Accuracy score is : 0.85
```

In [88]:

```
print('Error Rate: ',(fp+fn)/(tp+tn+fn+fp))
```

```
Error Rate:  0.15
```

In [89]:

```
print('Precision :',tp/(tp+fp))
```

```
Precision : 0.9393939393939394
```

In [90]:

```
print('Recall :',tp/(tp+fn))
```

```
Recall : 0.6595744680851063
```