In [1]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.preprocessing import StandardScaler
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LogisticRegression
from mlxtend.plotting import plot_confusion_matrix
from sklearn.metrics import confusion_matrix, classification_report, accurac
import warnings
warnings.filterwarnings('ignore')
%matplotlib inline
```

In [3]:
```
df = pd.read_csv("C:\\Users\\harsh\\Downloads\\Social_Network_Ads.csv")
```

In [4]:
```
df.head()
```

Out[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
4	15804002	Male	19	76000	0

In [5]:
```
x = df[["Age", "EstimatedSalary"]]
y = df["Purchased"]
```

In [6]:
```
scaler = StandardScaler()
x = scaler.fit_transform(x)
```

In [7]:
```
x_train, x_test, y_train, y_test = train_test_split(x, y, test_size=0.2, rar
```

In [8]:
```
x_train.shape, x_test.shape, y_train.shape, y_test.shape
```

Out[8]:
```
((320, 2), (80, 2), (320,), (80,))
```

In [10]:
```
model = LogisticRegression()
```

In [11]:
```
model.fit(x_train, y_train)
```

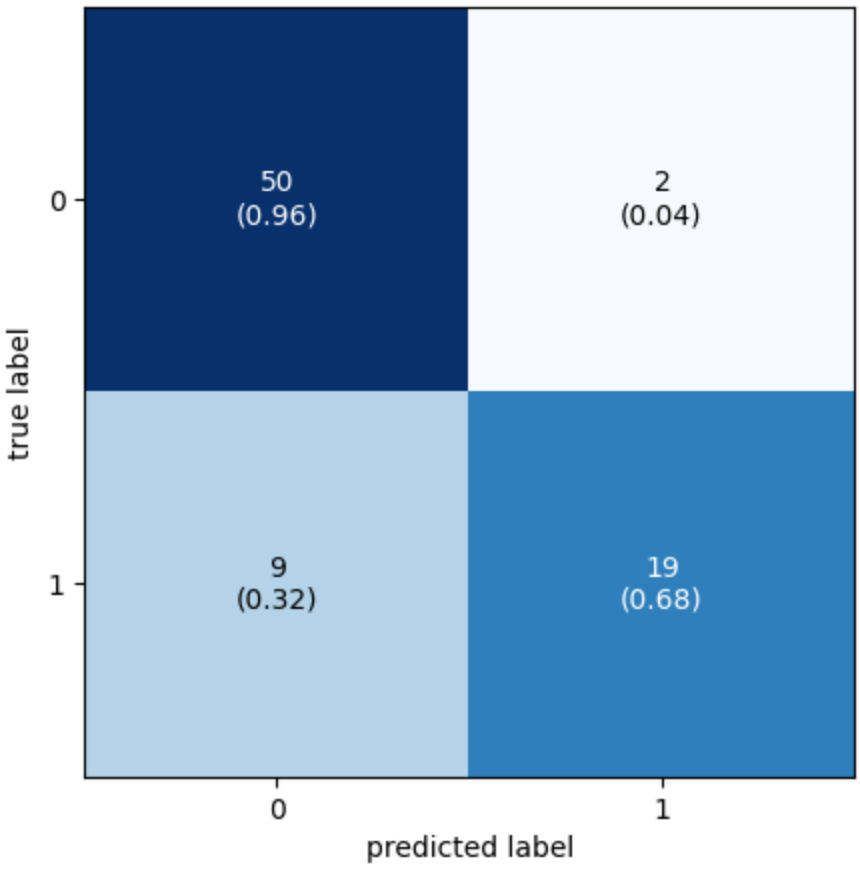
Out[11]:
```
LogisticRegression()
```

In [12]:
```
y_pred = model.predict(x_test)
```

In [13]:
```
cm = confusion_matrix(y_test, y_pred)
print(cm)
```

```
[[50  2]
 [ 9 19]]
```

In [14]:
```
plot_confusion_matrix(conf_mat=cm, figsize=(5,5), show_normed=True)
plt.show()
```



In [15]:
```
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 50
FP value is 2
FN value is 9
TP value is 19
```

In [16]:
```
print(f"Accuracy score is {accuracy_score(y_test, y_pred)}")
```

```
Accuracy score is 0.8625
```

In [17]:
```
print(f"Error rate is {1-accuracy_score(y_test, y_pred)}")
```

```
Error rate is 0.13749999999999996
```

In [18]:
```
print(f"Precision score is {precision_score(y_test, y_pred)}")
```

```
Precision score is 0.9047619047619048
```

In [19]:
```
print(f"Recall score is {recall_score(y_test, y_pred)}")
```

```
Recall score is 0.6785714285714286
```

In [20]:
```
print(classification_report(y_test, y_pred))
```

```
              precision    recall  f1-score   support

           0       0.85      0.96      0.90        52
           1       0.90      0.68      0.78        28

    accuracy                           0.86        80
   macro avg       0.88      0.82      0.84        80
weighted avg       0.87      0.86      0.86        80
```

In [21]:
```
from matplotlib.colors import ListedColormap

x_set, y_set = x_train, y_train
x1, x2 = np.meshgrid(np.arange(start=x_set[:, 0].min() - 1, stop=x_set[:, 0]
                     np.arange(start=x_set[:, 1].min() - 1, stop=x_set[:, 1]

plt.contourf(x1, x2, model.predict(np.array([x1.ravel(), x2.ravel()]).T).res
             alpha=0.75, cmap=ListedColormap(('red', 'green')))

plt.scatter(x_set[:, 0], x_set[:, 1], c=y_set, cmap=ListedColormap(('red', '
plt.title('Logistic Regression Decision Boundary')
plt.xlabel('Feature 1')
plt.ylabel('Feature 2')
plt.show()
```

In [ ]: