Name: Harsh Patel

Roll NO: A42

Experiment No:5

Aim: Write a python program to evaluate a Applying Logistic Regression on iris Dataset

Code:

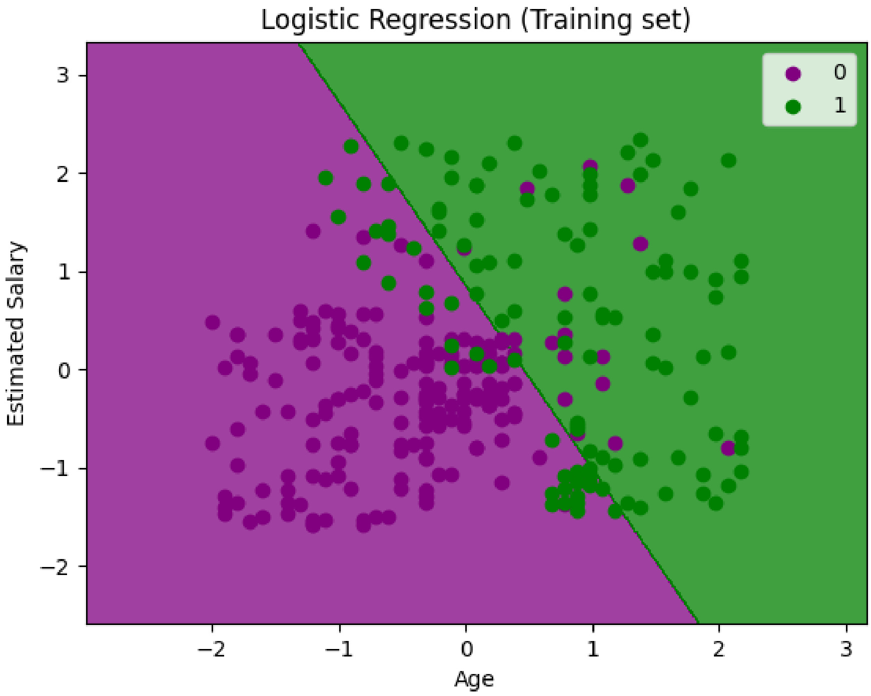
```
import numpy as np

import matplotlib.pyplot as plt

import pandas as pd

from sklearn.model_selection import train_test_split

from sklearn.preprocessing import StandardScaler

from sklearn.linear_model import LogisticRegression

from sklearn.metrics import confusion_matrix

from matplotlib.colors import ListedColormap


# Importing dataset

data_set = pd.read_csv('user_data.csv')


# Extracting Independent and dependent Variable

x = data_set.iloc[:, [2, 3]].values

y = data_set.iloc[:, 4].values


# Splitting the dataset into training and test set

x_train, x_test, y_train, y_test = train_test_split(x, y, test_size=0.25, random_state=0)


# Feature Scaling

sc_x = StandardScaler()

x_train = sc_x.fit_transform(x_train)

x_test = sc_x.transform(x_test)
```

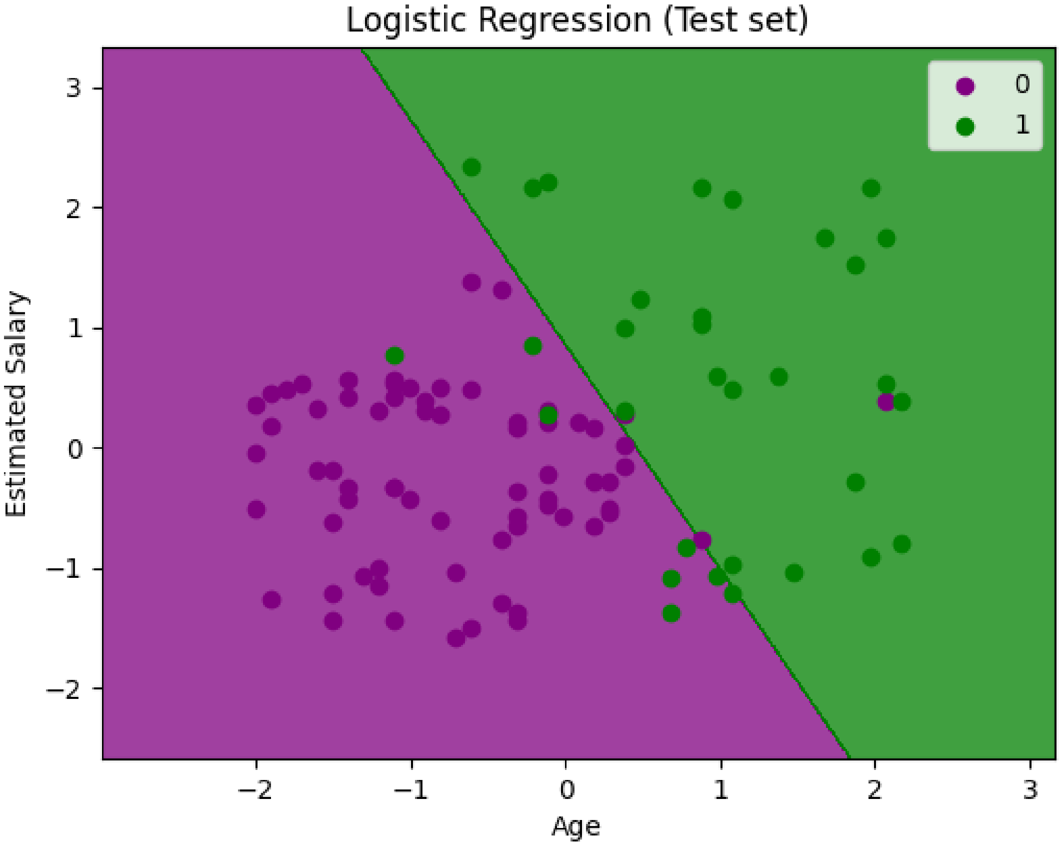
```
# Fitting Logistic Regression to the training set
classifier = LogisticRegression(random_state=0)
classifier.fit(x_train, y_train)

# Predicting the test set result
y_pred = classifier.predict(x_test)

# Creating the Confusion matrix
cm = confusion_matrix(y_test, y_pred)
print("Confusion Matrix:")
print(cm)

# Visualizing the training set result
x_set, y_set = x_train, y_train
x1, x2 = np.meshgrid(np.arange(start=x_set[:, 0].min() - 1, stop=x_set[:, 0].max() + 1, step=0.01),
                     np.arange(start=x_set[:, 1].min() - 1, stop=x_set[:, 1].max() + 1, step=0.01))
plt.contourf(x1, x2, classifier.predict(np.array([x1.ravel(), x2.ravel()]).T).reshape(x1.shape),
             alpha=0.75, cmap=ListedColormap(('purple', 'green')))
plt.xlim(x1.min(), x1.max())
plt.ylim(x2.min(), x2.max())
for i, j in enumerate(np.unique(y_set)):
    plt.scatter(x_set[y_set == j, 0], x_set[y_set == j, 1],
                c=ListedColormap(('purple', 'green'))(i), label=j)
plt.title('Logistic Regression (Training set)')
plt.xlabel('Age')
plt.ylabel('Estimated Salary')
plt.legend()
plt.show()
```

```
# Visualizing the test set result
x_set, y_set = x_test, y_test
x1, x2 = np.meshgrid(np.arange(start=x_set[:, 0].min() - 1, stop=x_set[:, 0].max() + 1, step=0.01),
                     np.arange(start=x_set[:, 1].min() - 1, stop=x_set[:, 1].max() + 1, step=0.01))
plt.contourf(x1, x2, classifier.predict(np.array([x1.ravel(), x2.ravel()]).T).reshape(x1.shape),
             alpha=0.75, cmap=ListedColormap(('purple', 'green')))
plt.xlim(x1.min(), x1.max())
plt.ylim(x2.min(), x2.max())
for i, j in enumerate(np.unique(y_set)):
    plt.scatter(x_set[y_set == j, 0], x_set[y_set == j, 1],
                c=ListedColormap(('purple', 'green'))(i), label=j)
plt.title('Logistic Regression (Test set)')
plt.xlabel('Age')
plt.ylabel('Estimated Salary')
plt.legend()
plt.show()
```

<u>OUTPUT:</u>

```
Confusion Matrix:
[[65  3]
 [ 8 24]]
```


Logistic Regression (Training set)
0
1
3
2
1
0
−1
−2
Estimated Salary
−2
−1
0
1
2
3
Age


Logistic Regression (Test set)
0
1
3
2
1
0
−1
−2
Estimated Salary
−2
−1
0
1
2
3
Age