In [ ]:
```
import numpy as nm
import matplotlib.pyplot as mtp
import pandas as pd
```

In [ ]:
```
data_set = pd.read_csv("datasets/Mall_Customers.csv")
```

In [ ]:
```
x= data_set.iloc[:,[2,3]].values
y = data_set.iloc[:,4].values
```

In [ ]:
```
from sklearn.model_selection import train_test_split
```

In [ ]:
```
x_train,x_test,y_train,y_test = train_test_split(x,y,test_size =0.25,random_state =
```

In [ ]:
```
x_train
```

Out[ ]:
```
array([[ 27, 255],
       [ 32,  97],
       [ 59, 255],
       [ 18,  33],
       [ 44, 155],
       [ 31,  70],
       [ 52, 240],
       [ 47, 278],
       [ 56, 190],
       [ 40,  87],
       [ 35,  21],
       [ 48,  77],
       [ 66,  63],
       [ 30, 292],
       [ 31,  17],
       [ 32, 222],
       [ 19,  46],
       [ 23,  70],
       [ 31,  43],
       [ 43,  48],
       [ 45,  28],
       [ 66,  63],
       [ 27, 288],
       [ 36, 234],
       [ 65,  38],
       [ 24,  20],
       [ 23,  16],
       [ 34, 296],
       [ 54,  28],
       [ 60,  30],
       [ 36, 238],
       [ 70,  46],
       [ 51,  44],
       [ 32,  73],
       [ 38,  71],
       [ 35,  23],
       [ 28, 101],
       [ 20,  73],
       [ 50,  43],
       [ 43, 140],
       [ 57,  54],
       [ 33,  42],
       [ 20,  16],
       [ 31, 325],
       [ 49,  62],
       [ 41,  99],
       [ 55,  57],
       [ 67,  19],
       [ 36,  87],
       [ 59,  93],
       [ 29, 267],
       [ 24,  38],
       [ 37,  20],
       [ 35,  28],
       [ 49,  42],
       [ 58, 250],
```

```
       [ 35,  24],
       [ 37, 167],
       [ 50,  40],
       [ 28,  87],
       [ 30, 297],
       [ 39,  69],
       [ 28,  77],
       [ 36, 201],
       [ 67,  47],
       [ 19,  74],
       [ 34, 103],
       [ 38,  78],
       [ 29,  73],
       [ 31,  39],
       [ 45, 126],
       [ 25,  77],
       [ 30,  78],
       [ 38,  64],
       [ 46,  98],
       [ 48,  61],
       [ 34,  78],
       [ 40,  60],
       [ 34,  78],
       [ 38, 275],
       [ 19,  64],
       [ 19,  15],
       [ 32,  60],
       [ 24,  60],
       [ 32,  48],
       [ 31,  40],
       [ 29,  40],
       [ 48,  54],
       [ 29, 313],
       [ 32,  75],
       [ 31,  25],
       [ 40, 230],
       [ 32,  76],
       [ 23,  54],
       [ 23,  62],
       [ 39,  71],
       [ 39, 141],
       [ 19, 316],
       [ 35,  18],
       [ 19,  48],
       [ 21,  54],
       [ 38,  67],
       [ 41, 267],
       [ 36, 103],
       [ 19, 194],
       [ 30, 177],
       [ 32, 103],
       [ 18,  59],
       [ 31, 194],
       [ 35,  19],
       [ 50,  67],
       [ 67,  62],
```

```
[ 21,  33],
[ 69,  44],
[ 32,  87],
[ 18,  48],
[ 21,  15],
[ 27,  67],
[ 30, 137],
[ 48,  39],
[ 21,  62],
[ 25,  72],
[ 43, 273],
[ 20,  21],
[ 36,  37],
[ 31,  72],
[ 59,  43],
[ 50,  85],
[ 50, 200],
[ 59,  71],
[ 49,  33],
[ 40,  29],
[ 29,  98],
[ 18,  65],
[ 31,  81],
[ 39,  78],
[ 21,  30],
[ 27, 160],
[ 40,  71],
[ 30,  99],
[ 28, 233],
[ 46, 266],
[ 53,  33],
[ 33,  86],
[ 28,  76],
[ 30, 189],
[ 32,  77],
[ 23,  29],
[ 27,  88],
[ 20,  61],
[ 67,  54],
[ 56, 311],
[ 30,  88],
[ 49,  54],
[ 32, 126],
[ 34, 188],
[ 19,  65],
[ 34,  78],
[ 37, 288],
[ 60,  50],
[ 40,  54],
[ 29,  28],
[ 36,  85],
[ 38,  54],
[ 30, 268],
[ 52,  88],
[ 44, 275],
[ 20,  37],
```

```
       [ 37, 258],
       [ 38, 113],
       [ 27,  46],
       [ 57,  75],
       [ 34,  58],
       [ 42, 208],
       [ 70,  49],
       [ 22,  57],
       [ 42,  34],
       [ 25,  24],
       [ 29, 192],
       [ 30,  19],
       [ 26,  62],
       [ 35, 120],
       [ 68,  48],
       [ 33, 113],
       [ 49,  65],
       [ 27,  40],
       [ 36,  87]], dtype=int64)
```

In [ ]:
```
from sklearn.preprocessing import StandardScaler
```

In [ ]:
```
st_x = StandardScaler()
x_train = st_x.fit_transform(x_train)
x_test = st_x.transform(x_test)
```

In [ ]:
```
x_train
```

Out[ ]:
```
array([[-0.76895498,  1.90031564],
       [-0.3828256 , -0.03311156],
       [ 1.70227306,  1.90031564],
       [-1.46398787, -0.81627195],
       [ 0.54388492,  0.67662754],
       [-0.46005148, -0.36350735],
       [ 1.16169193,  1.71676243],
       [ 0.77556254,  2.1817639 ],
       [ 1.47059543,  1.10491837],
       [ 0.23498141, -0.15548037],
       [-0.15114797, -0.96311452],
       [ 0.85278842, -0.27784918],
       [ 2.2428542 , -0.44916552],
       [-0.53727735,  2.35308024],
       [-0.46005148, -1.01206204],
       [-0.3828256 ,  1.49649857],
       [-1.38676199, -0.65719249],
       [-1.07785849, -0.36350735],
       [-0.46005148, -0.69390314],
       [ 0.46665904, -0.63271873],
       [ 0.62111079, -0.87745635],
       [ 2.2428542 , -0.44916552],
       [-0.76895498,  2.30413271],
       [-0.0739221 ,  1.64334114],
       [ 2.16562832, -0.75508754],
       [-1.00063261, -0.9753514 ],
       [-1.07785849, -1.02429892],
       [-0.22837385,  2.40202776],
       [ 1.31614368, -0.87745635],
       [ 1.77949894, -0.85298259],
       [-0.0739221 ,  1.69228866],
       [ 2.5517577 , -0.65719249],
       [ 1.08446605, -0.68166625],
       [-0.3828256 , -0.3267967 ],
       [ 0.08052966, -0.35127047],
       [-0.15114797, -0.93864076],
       [-0.69172911,  0.01583596],
       [-1.30953612, -0.3267967 ],
       [ 1.00724017, -0.69390314],
       [ 0.46665904,  0.49307432],
       [ 1.54782131, -0.55929744],
       [-0.30559972, -0.70614002],
       [-1.30953612, -1.02429892],
       [-0.46005148,  2.75689731],
       [ 0.9300143 , -0.4614024 ],
       [ 0.31220729, -0.0086378 ],
       [ 1.39336956, -0.5225868 ],
       [ 2.32008007, -0.98758828],
       [-0.0739221 , -0.15548037],
       [ 1.70227306, -0.08205908],
       [-0.61450323,  2.04715821],
       [-1.00063261, -0.75508754],
       [ 0.00330378, -0.9753514 ],
       [-0.15114797, -0.87745635],
       [ 0.9300143 , -0.70614002],
       [ 1.62504718,  1.83913124],
```

```
[-0.15114797, -0.92640387],
[ 0.00330378,  0.82347011],
[ 1.00724017, -0.73061378],
[-0.69172911, -0.15548037],
[-0.53727735,  2.41426464],
[ 0.15775553, -0.37574423],
[-0.69172911, -0.27784918],
[-0.0739221 ,  1.23952407],
[ 2.32008007, -0.64495561],
[-1.38676199, -0.31455982],
[-0.22837385,  0.04030973],
[ 0.08052966, -0.2656123 ],
[-0.61450323, -0.3267967 ],
[-0.46005148, -0.74285066],
[ 0.62111079,  0.32175799],
[-0.92340674, -0.27784918],
[-0.53727735, -0.2656123 ],
[ 0.08052966, -0.43692863],
[ 0.69833667, -0.02087468],
[ 0.85278842, -0.47363928],
[-0.22837385, -0.2656123 ],
[ 0.23498141, -0.48587616],
[-0.22837385, -0.2656123 ],
[ 0.08052966,  2.14505326],
[-1.38676199, -0.43692863],
[-1.38676199, -1.0365358 ],
[-0.3828256 , -0.48587616],
[-1.00063261, -0.48587616],
[-0.3828256 , -0.63271873],
[-0.46005148, -0.73061378],
[-0.61450323, -0.73061378],
[ 0.85278842, -0.55929744],
[-0.61450323,  2.61005474],
[-0.3828256 , -0.30232294],
[-0.46005148, -0.91416699],
[ 0.23498141,  1.59439362],
[-0.3828256 , -0.29008606],
[-1.07785849, -0.55929744],
[-1.07785849, -0.4614024 ],
[ 0.15775553, -0.35127047],
[ 0.15775553,  0.5053112 ],
[-1.38676199,  2.64676538],
[-0.15114797, -0.99982516],
[-1.38676199, -0.63271873],
[-1.23231024, -0.55929744],
[ 0.08052966, -0.40021799],
[ 0.31220729,  2.04715821],
[-0.0739221 ,  0.04030973],
[-1.38676199,  1.1538659 ],
[-0.53727735,  0.94583892],
[-0.3828256 ,  0.04030973],
[-1.46398787, -0.49811304],
[-0.46005148,  1.1538659 ],
[-0.15114797, -0.98758828],
[ 1.00724017, -0.40021799],
[ 2.32008007, -0.4614024 ],
```

```
[-1.23231024, -0.81627195],
[ 2.47453183, -0.68166625],
[-0.3828256 , -0.15548037],
[-1.46398787, -0.63271873],
[-1.23231024, -1.0365358 ],
[-0.76895498, -0.40021799],
[-0.53727735,  0.45636368],
[ 0.85278842, -0.74285066],
[-1.23231024, -0.4614024 ],
[-0.92340674, -0.33903359],
[ 0.46665904,  2.1205795 ],
[-1.30953612, -0.96311452],
[-0.0739221 , -0.76732442],
[-0.46005148, -0.33903359],
[ 1.70227306, -0.69390314],
[ 1.00724017, -0.17995413],
[ 1.00724017,  1.22728718],
[ 1.70227306, -0.35127047],
[ 0.9300143 , -0.81627195],
[ 0.23498141, -0.86521947],
[-0.61450323, -0.02087468],
[-1.46398787, -0.42469175],
[-0.46005148, -0.22890166],
[ 0.15775553, -0.2656123 ],
[-1.23231024, -0.85298259],
[-0.76895498,  0.73781194],
[ 0.23498141, -0.35127047],
[-0.53727735, -0.0086378 ],
[-0.69172911,  1.63110426],
[ 0.69833667,  2.03492133],
[ 1.2389178 , -0.81627195],
[-0.30559972, -0.16771725],
[-0.69172911, -0.29008606],
[-0.53727735,  1.09268149],
[-0.3828256 , -0.27784918],
[-1.07785849, -0.86521947],
[-0.76895498, -0.14324349],
[-1.30953612, -0.47363928],
[ 2.32008007, -0.55929744],
[ 1.47059543,  2.58558098],
[-0.53727735, -0.14324349],
[ 0.9300143 , -0.55929744],
[-0.3828256 ,  0.32175799],
[-0.22837385,  1.08044461],
[-1.38676199, -0.42469175],
[-0.22837385, -0.2656123 ],
[ 0.00330378,  2.30413271],
[ 1.77949894, -0.60824497],
[ 0.23498141, -0.55929744],
[-0.61450323, -0.87745635],
[-0.0739221 , -0.17995413],
[ 0.08052966, -0.55929744],
[-0.53727735,  2.05939509],
[ 1.16169193, -0.14324349],
[ 0.54388492,  2.14505326],
[-1.30953612, -0.76732442],
```

```
       [ 0.00330378,  1.93702628],
       [ 0.08052966,  0.16267854],
       [-0.76895498, -0.65719249],
       [ 1.54782131, -0.30232294],
       [-0.22837385, -0.51034992],
       [ 0.38943316,  1.32518223],
       [ 2.5517577 , -0.62048185],
       [-1.15508436, -0.5225868 ],
       [ 0.38943316, -0.80403506],
       [-0.92340674, -0.92640387],
       [-0.61450323,  1.12939214],
       [-0.53727735, -0.98758828],
       [-0.84618086, -0.4614024 ],
       [-0.15114797,  0.2483367 ],
       [ 2.39730595, -0.63271873],
       [-0.30559972,  0.16267854],
       [ 0.9300143 , -0.42469175],
       [-0.76895498, -0.73061378],
       [-0.0739221 , -0.15548037]])
```

In [ ]:
```
from sklearn.linear_model import LogisticRegression
```

In [ ]:
```
classifier = LogisticRegression(random_state=0)
classifier.fit(x_train,y_train)
```

Out[ ]:
```
LogisticRegression(random_state=0)
```

In [ ]:
```
LogisticRegression(C=1.0,class_weight=None, dual=False,fit_intercept=True,intercept
                   l1_ratio=None,max_iter=100,multi_class='warn',n_jobs=None,penalt
                   solver='warn',tol=0.0001,verbose=0,warm_start=False)
```

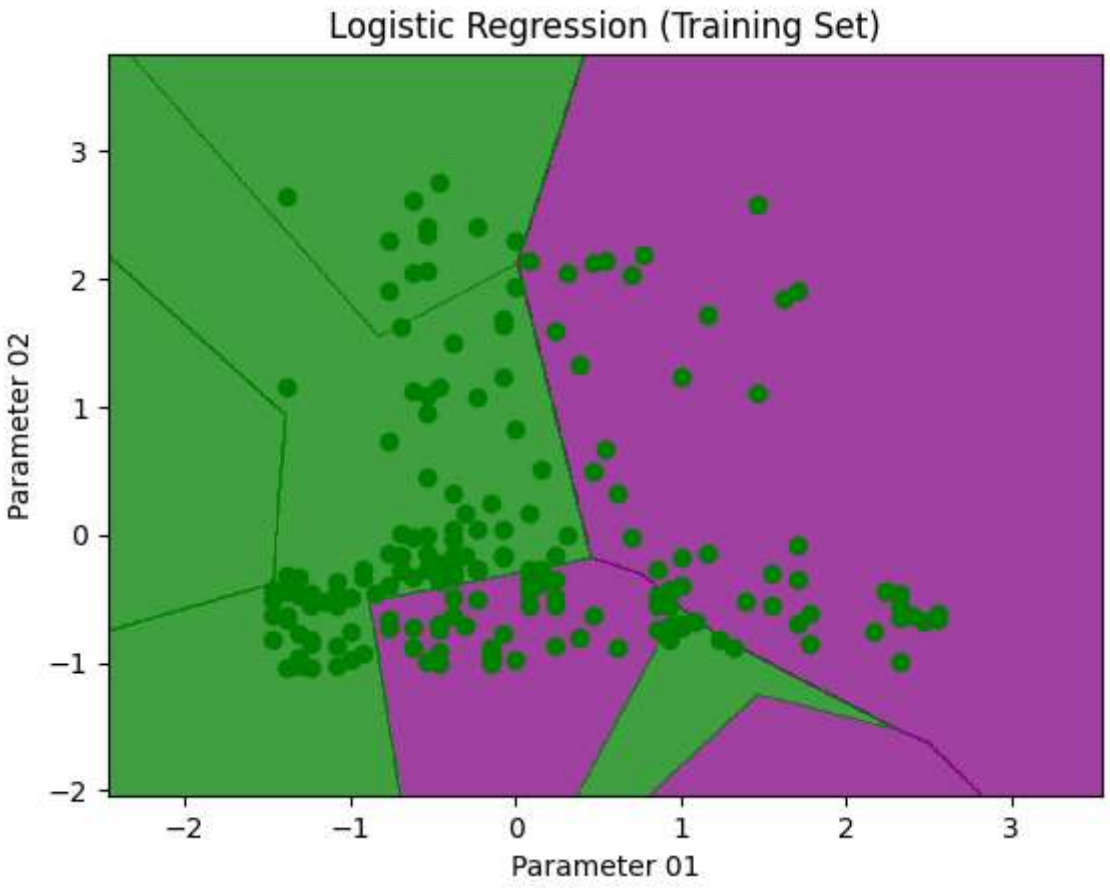
Out[ ]:
```
LogisticRegression(multi_class='warn', random_state=0, solver='warn')
```

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

In [ ]:
```
from sklearn.metrics import confusion_matrix
cm = confusion_matrix(y_test,y_pred)
```

In [ ]:
```
from matplotlib.colors import ListedColormap
x_set,y_set = x_train,y_train

x1,x2 = nm.meshgrid(nm.arange(start = x_set[:,0].min() - 1, stop = x_set[:,0].max()
nm.arange(start = x_set[:,1].min() - 1,stop = x_set[:,1].max() + 1, step = 0.01))

mtp.contourf(x1,x2,classifier.predict(nm.array([x1.ravel(),x2.ravel()]).T).reshape(
             alpha = 0.75,cmap = ListedColormap(('purple','green')))

mtp.xlim(x1.min(),x1.max())
mtp.ylim(x2.min(),x2.max())
```

```
for i,j in enumerate(nm.unique(y_set)):
    mtp.scatter(x_set[y_set == j,0], x_set[y_set == j,1],
                color = ListedColormap(('purple','green'))(i),label = j)


mtp.title("Logistic Regression (Training Set)")
mtp.xlabel("Parameter 01")
mtp.ylabel("Parameter 02")
mtp.show()
```



In [ ]:
```
x_set,y_set = x_test,y_test

x1,x2 = nm.meshgrid(nm.arange(start = x_set[:,0].min() - 1,stop = x_set[:,0].max()
                    nm.arange(start = x_set[:,1].min() - 1, stop = x_set[:,1].max()

mtp.contourf(x1,x2,classifier.predict(nm.array([x1.ravel(),x2.ravel()]).T).reshape(
            alpha = 0.75,cmap = ListedColormap(('red','green')))

mtp.xlim(x1.min(),x1.max())
mtp.ylim(x2.min(),x2.max())

for i, j in enumerate(nm.unique(y_set)):
    mtp.scatter(x_set[y_set == j,0],x_set[y_set == j,1],
                color = ListedColormap(('red','green'))(i),label = j)

mtp.title("Logistic Regression (Test Set)")
mtp.xlabel("Parameter 01")
```

```
mtp.ylabel("Parameter 02")
mtp.show()
```