In [2]:
```
import numpy as np
import matplotlib.pyplot as plt
import pandas as pd
```

In [7]:
```
df = pd.read_csv(r"C:\Users\System21\Desktop\Social_Network_Ads (1).csv")
```

In [9]:
```
df
```

Out[9]:

	User ID	Gender	Age	EstimatedSalary	Purchased
0	15624510	Male	19.0	19000.0	0
1	15810944	Male	35.0	20000.0	0
2	15668575	Female	26.0	43000.0	0
3	15603246	Female	27.0	57000.0	0
4	15804002	Male	19.0	76000.0	0
...	...	...	...	...	...
395	15691863	Female	46.0	41000.0	1
396	15706071	Male	51.0	23000.0	1
397	15654296	Female	50.0	20000.0	1
398	15755018	Male	36.0	33000.0	0
399	15594041	Female	49.0	36000.0	1

400 rows × 5 columns

In [11]:
```
X = dataset.iloc[:, [2, 3]].values
y = dataset.iloc[:, 4].values
```

In [13]:
```
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size = 0.25, rar
```

In [15]:
```
from sklearn.preprocessing import StandardScaler
sc = StandardScaler()
X_train = sc.fit_transform(X_train)
X_test = sc.transform(X_test)
```

In [17]:
```
from sklearn.linear_model import LogisticRegression
log_reg = LogisticRegression(random_state = 0)
log_reg.fit(X_train, y_train)
```

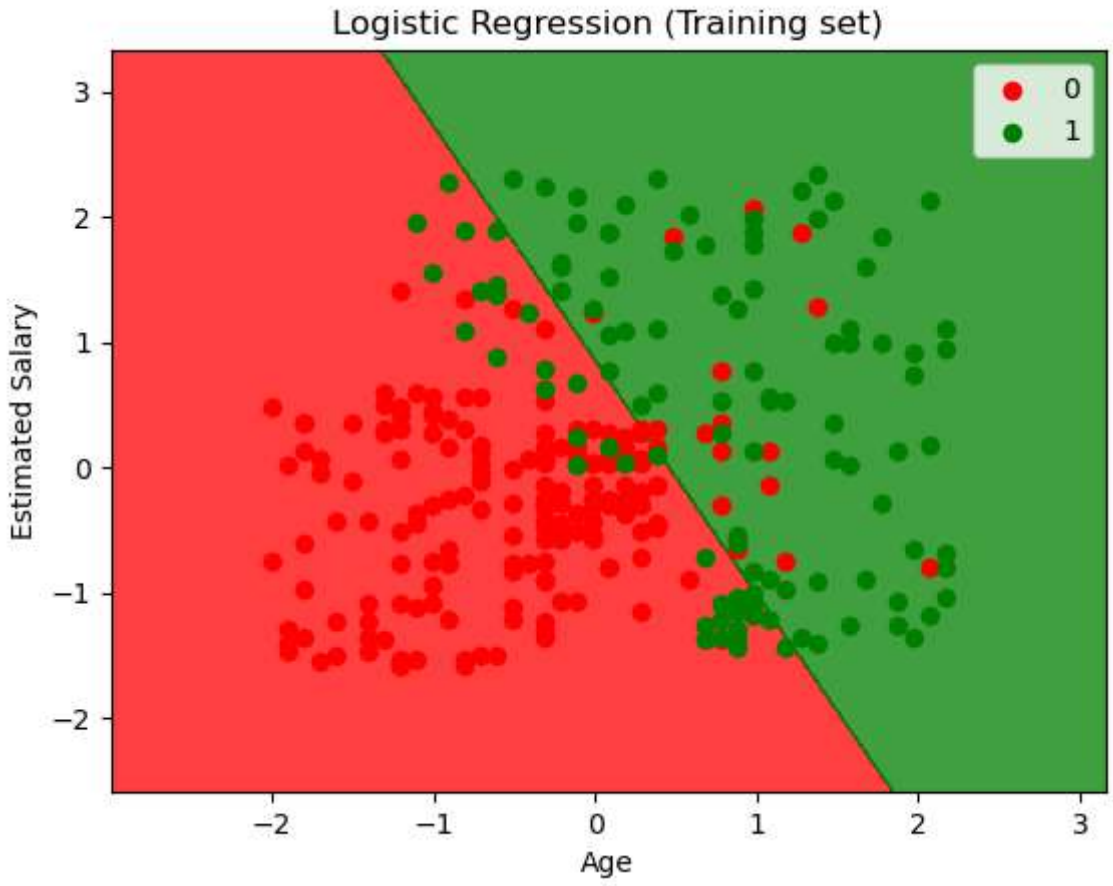
Out[17]:
```
LogisticRegression
LogisticRegression(random_state=0)
```

In [19]:
```
y_pred = log_reg.predict(X_test)
```

In [21]:
```
from sklearn.metrics import confusion_matrix
cm = confusion_matrix(y_test, y_pred)
```

In [23]:
```
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, log_reg.predict(np.array([X1.ravel(), X2.ravel()]).T).resh
             alpha = 0.75, cmap = ListedColormap(('red', '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(('red', 'green'))(i), label = j)
plt.title('Logistic Regression (Training set)')
plt.xlabel('Age')
plt.ylabel('Estimated Salary')
plt.legend()
plt.show()
```

```
C:\Users\System21\AppData\Local\Temp\ipykernel_8564\1052239858.py:10: UserWar
ning: *c* argument looks like a single numeric RGB or RGBA sequence, which sh
ould be avoided as value-mapping will have precedence in case its length matc
hes with *x* & *y*.  Please use the *color* keyword-argument or provide a 2D
array with a single row if you intend to specify the same RGB or RGBA value f
or all points.
  plt.scatter(X_set[y_set == j, 0], X_set[y_set == j, 1],
```



In [24]:
```
from matplotlib.colors import ListedColormap
X_set, y_set = X_test, y_test
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, log_reg.predict(np.array([X1.ravel(), X2.ravel()]).T).resh
             alpha = 0.75, cmap = ListedColormap(('red', '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(('red', 'green'))(i), label = j)
plt.title('Logistic Regression (Test set)')
plt.xlabel('Age')
plt.ylabel('Estimated Salary')
plt.legend()
plt.show()
```

```
C:\Users\System21\AppData\Local\Temp\ipykernel_8564\28934714.py:10: UserWarni
ng: *c* argument looks like a single numeric RGB or RGBA sequence, which shou
ld be avoided as value-mapping will have precedence in case its length matche
s with *x* & *y*.  Please use the *color* keyword-argument or provide a 2D ar
ray with a single row if you intend to specify the same RGB or RGBA value for
all points.
  plt.scatter(X_set[y_set == j, 0], X_set[y_set == j, 1],
```