

Program:

```
[1] import pandas as pd
import numpy as np
import matplotlib.pyplot as plt

dataset = pd.read_csv("User_Data.csv")

# input
x = dataset.iloc[:, [2, 3]].values

# output
y = dataset.iloc[:, 4].values

from sklearn.model_selection import train_test_split

xtrain, xtest, ytrain, ytest = train_test_split(
    x, y, test_size=0.25, random_state=0)

from sklearn.preprocessing import StandardScaler

sc_x = StandardScaler()
xtrain = sc_x.fit_transform(xtrain)
xtest = sc_x.transform(xtest)

print (xtrain[0:10, :])

[[ 2.149452 -1.02601437]
 [-0.28717375  0.70708966]
 [-1.26182405  0.4720925 ]
 [-0.40900504 -0.49727077]
 [-0.28717375 -0.0566511 ]
 [ 0.32198269 -1.23163688]
 [ 0.68747655  0.14897141]
 [ 0.32198269  2.6458162 ]
 [ 1.90578942 -0.99663973]
 [-0.40900504 -0.23289897]]
```

```
from sklearn.linear_model import LogisticRegression

classifier = LogisticRegression(random_state = 0)
classifier.fit(xtrain, ytrain)

y_pred = classifier.predict(xtest)

from sklearn.metrics import confusion_matrix

cm = confusion_matrix(ytest, y_pred)
print ("Confusion Matrix : \n", cm)
```

```
Confusion Matrix :
[[4 0]
 [0 1]]
```

```
from sklearn.metrics import accuracy_score
print ("Accuracy : ", accuracy_score(ytest, y_pred))
```

```
Accuracy :  1.0
```

```

▶ from matplotlib.colors import ListedColormap

X_set, y_set = xtest, ytest
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(('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('Classifier (Test set)')
plt.xlabel('Age')
plt.ylabel('Estimated Salary')
plt.legend()
plt.show()

```

<ipython-input-8-4d6cd9cec348>:17: UserWarning: *c* argument looks like a single numeric RGB or RGBA sequence
 plt.scatter(X_set[y_set == j, 0], X_set[y_set == j, 1],

