LOGISTIC REGRESSION

ANAKHA R MENON

CH.EN.U4CSE20103

CSE-B

In [ ]:

```
import seaborn as sns
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
```

In [ ]:

```
df=pd.read_csv("car_data.csv")
df.head()
```

Out[ ]:

	User ID	Gender	Age	AnnualSalary	Purchased
0	385	Male	35	20000	0
1	681	Male	40	43500	0
2	353	Male	49	74000	0
3	895	Male	40	107500	1
4	661	Male	25	79000	0

In [ ]:

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

In [ ]:

```
from sklearn.model_selection import train_test_split
x_train,x_test,y_train,y_test = train_test_split(x,y,test_size=0.2)
from sklearn.preprocessing import StandardScaler
sc_x = StandardScaler()
x_train = sc_x.fit_transform(x_train)
x_test = sc_x.transform(x_test)
```

In [ ]:

```
from sklearn.linear_model import LogisticRegression
model = LogisticRegression()
model.fit(x_train,y_train)
```

Out[ ]:

```
LogisticRegression()
```

In [ ]:

```
y_prep = model.predict(x_test)
```

In [ ]:

```
from sklearn.metrics import confusion_matrix
print("confusion matrix\n",confusion_matrix(y_prep,y_test))
```

```
confusion matrix
 [[104  27]
 [ 15  54]]
```

In [ ]:

```
from sklearn.metrics import accuracy_score
print("accuracy\n",accuracy_score(y_test,y_prep))
```

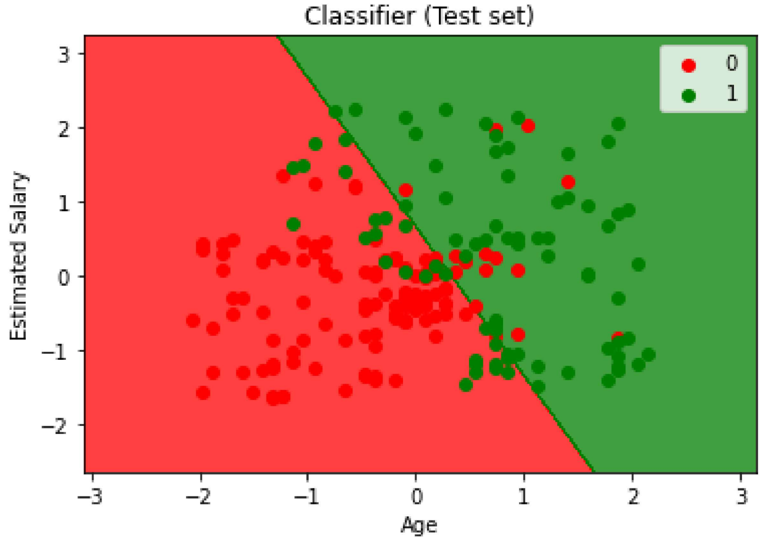
```
accuracy
 0.79
```

In [ ]:

```
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].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, model.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()
```

```
WARNING:matplotlib.axes._axes:*c* argument looks like a single numeric RGB or RGBA
sequence, which should be avoided as value-mapping will have precedence in case it
s length matches with *x* & *y*.  Please use the *color* keyword-argument or provi
de a 2-D array with a single row if you intend to specify the same RGB or RGBA val
ue for all points.
WARNING:matplotlib.axes._axes:*c* argument looks like a single numeric RGB or RGBA
sequence, which should be avoided as value-mapping will have precedence in case it
s length matches with *x* & *y*.  Please use the *color* keyword-argument or provi
de a 2-D array with a single row if you intend to specify the same RGB or RGBA val
ue for all points.
```



RESULT : Thus we have successfully implemented the concepts of linear and logistic regression in python programming.