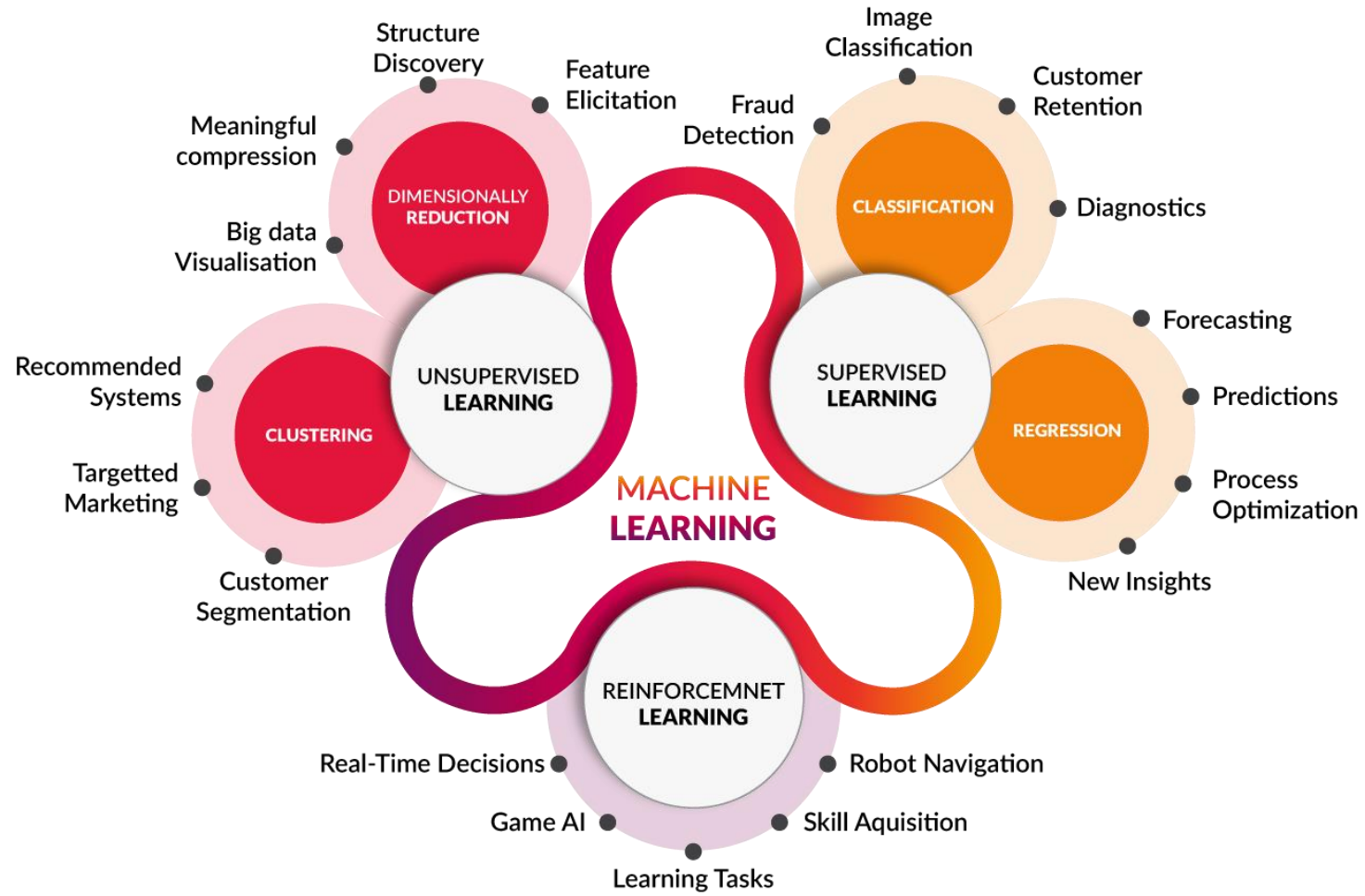


Logistic Regression

Machine Learning

Dr. Adnan Abid



Logistic Regression

Linear Regression:

- **Simple:**

$$y = b_0 + b_1 * x$$

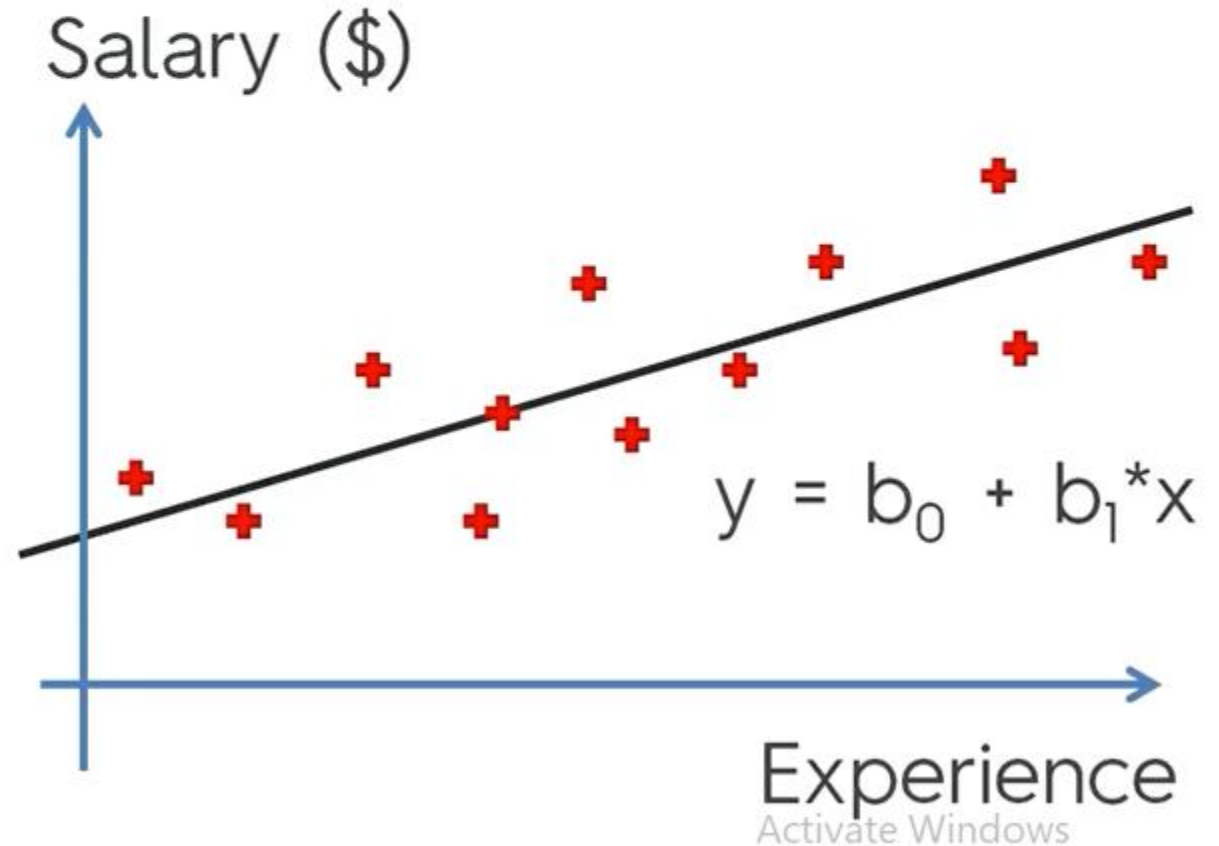
- **Multiple:**

$$y = b_0 + b_1 * x_1 + ... + b_n * x_n$$

Logistic Regression

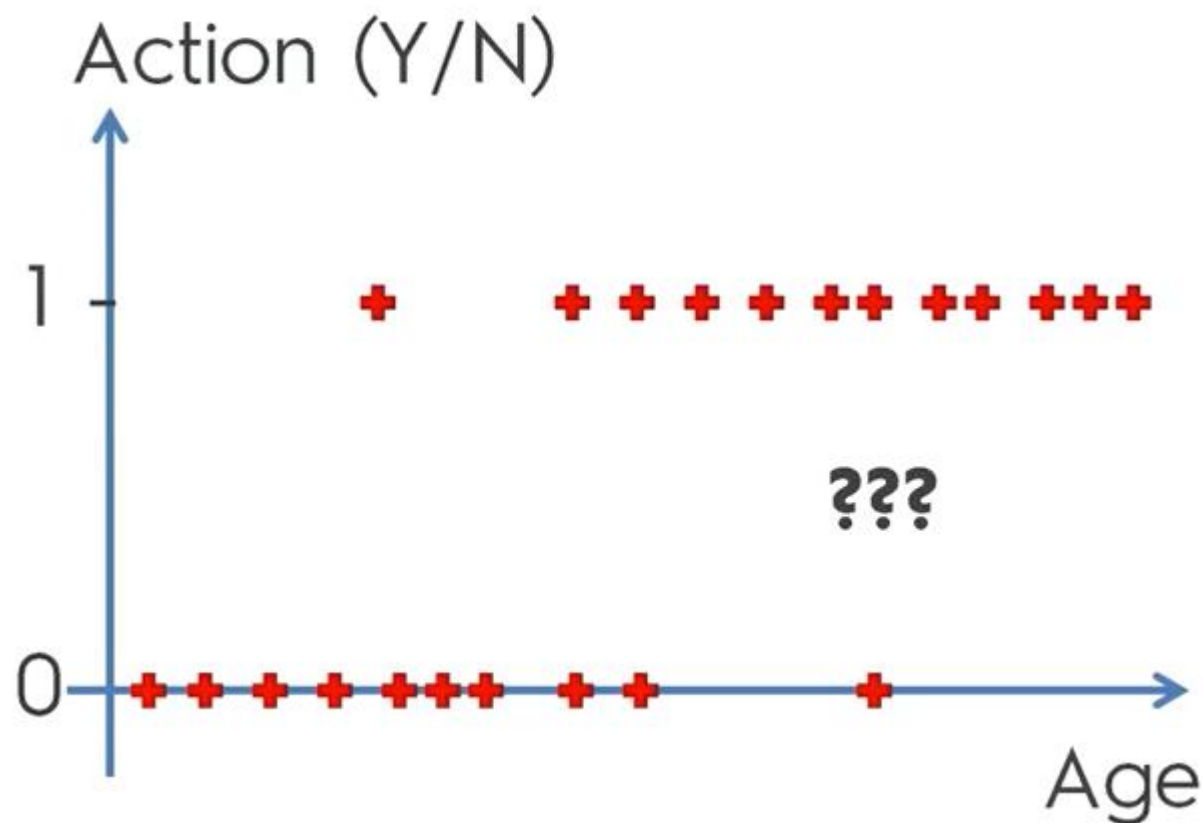
This is new:

We know this:

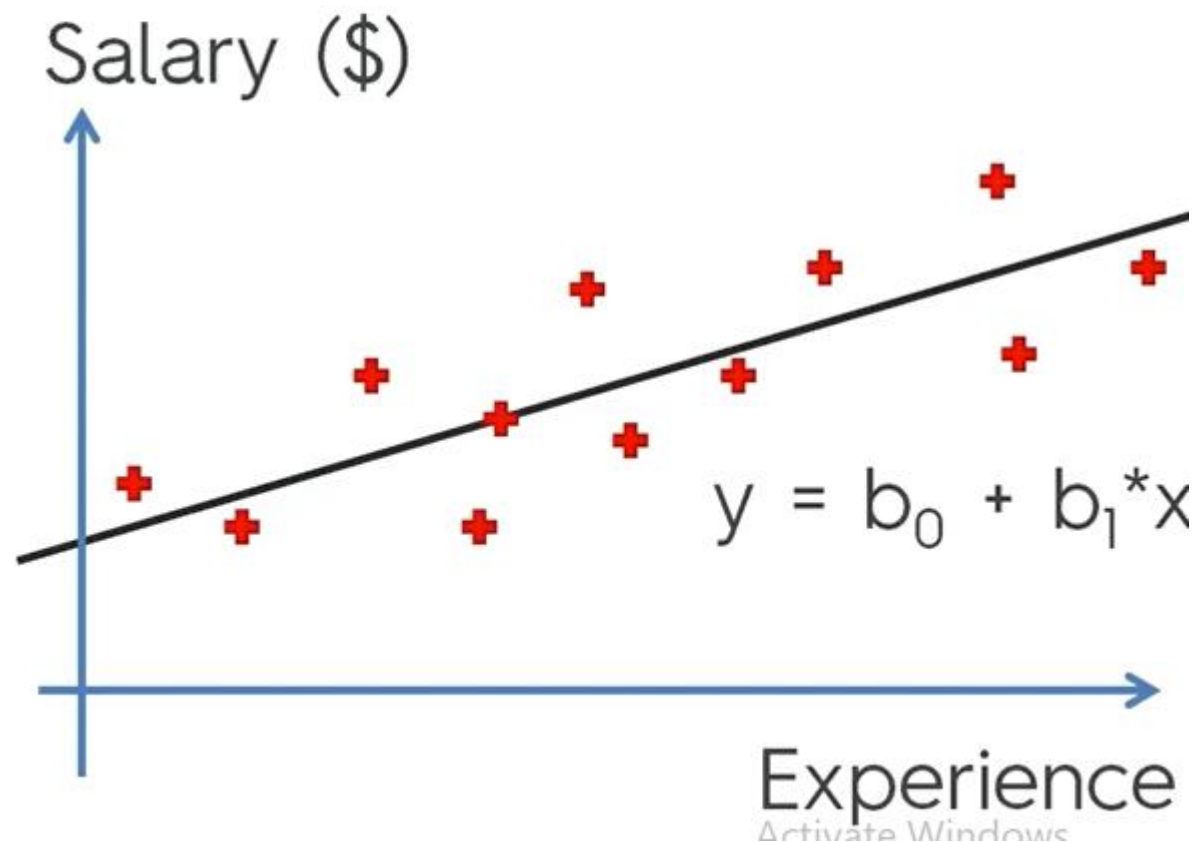


Logistic Regression

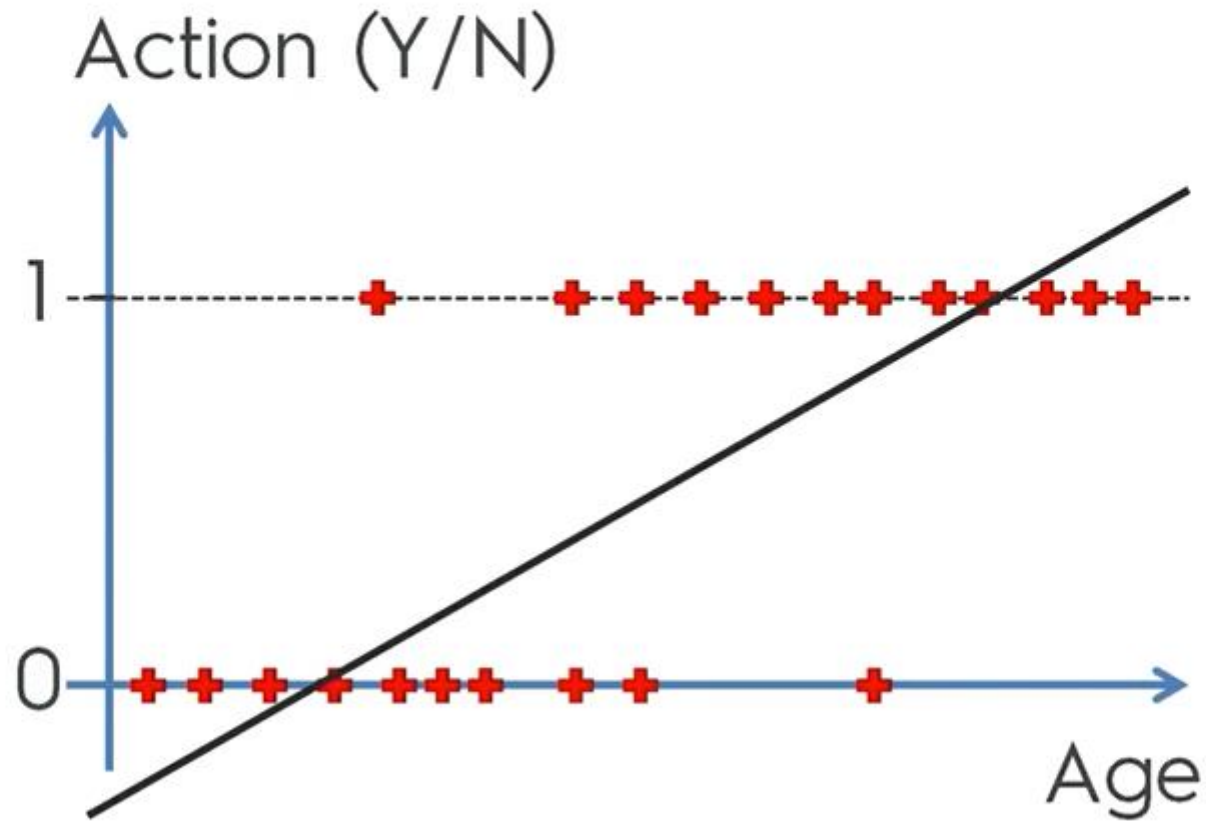
This is new:



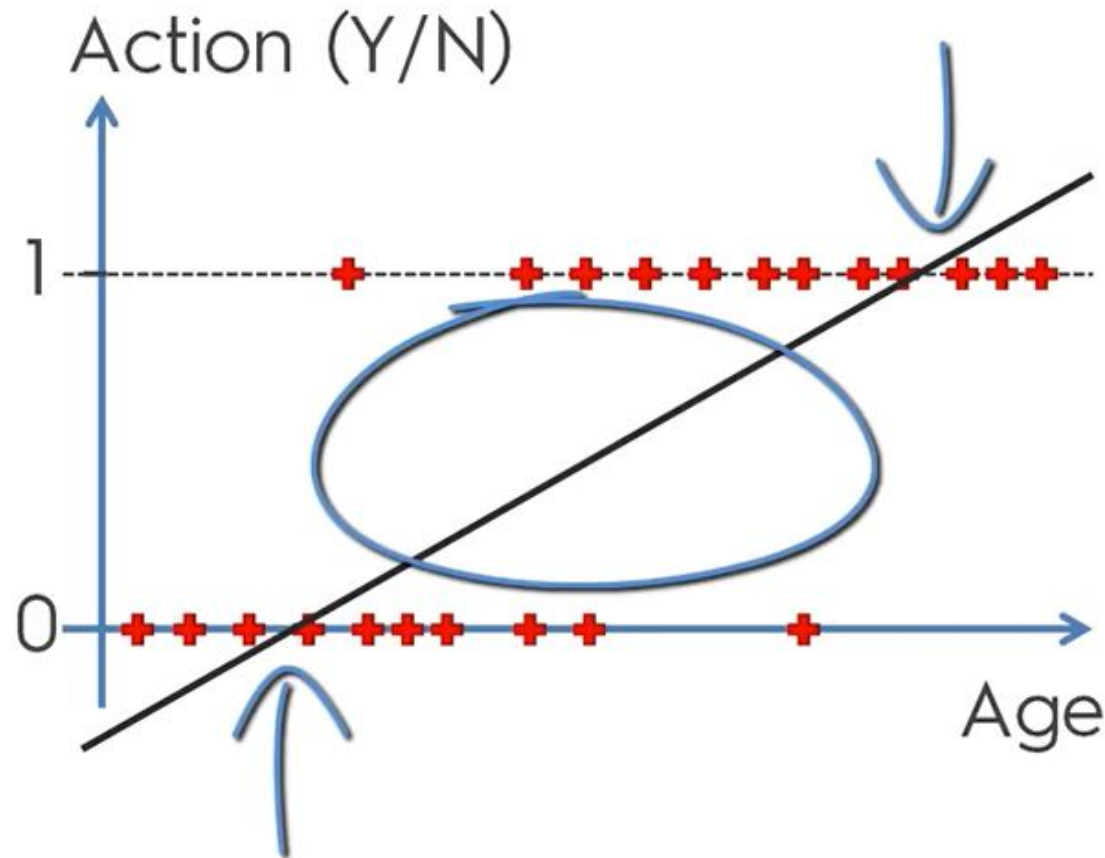
We know this:



Logistic Regression

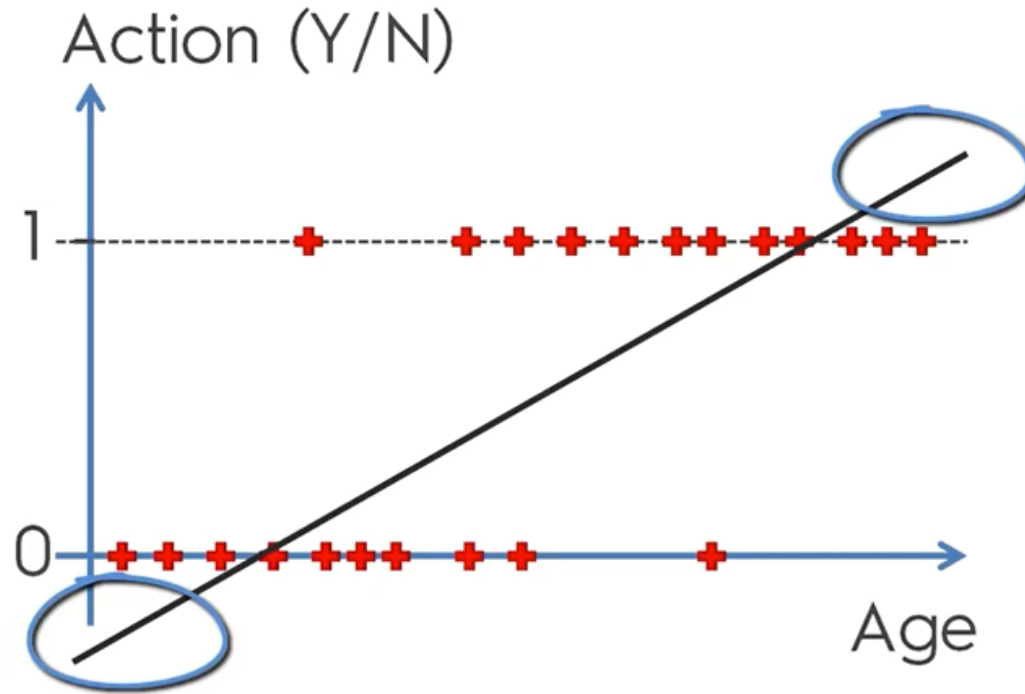


Logistic Regression



The graph shows the probability for the people between the age of 30 and 50 taking up this offer. While as the age increases there is a greater probability for them taking the offer.

Logistic Regression

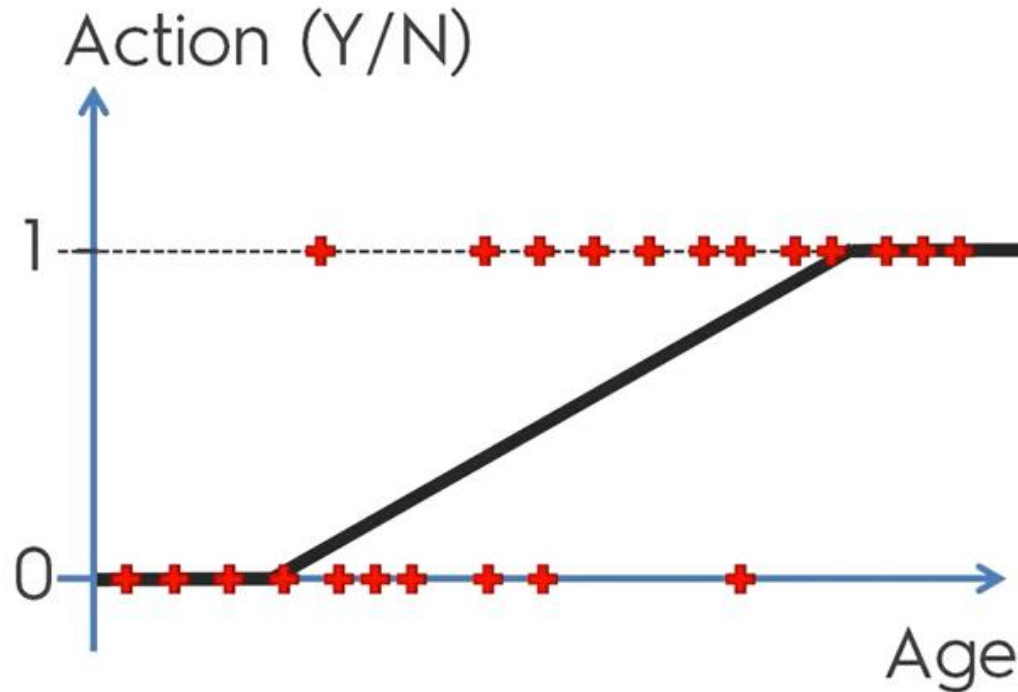


The graph shows the probability for the people between the age of 30 and 50 taking up this offer. While as the age increases there is a greater probability for them taking the offer.

However, for people below 30 the probability seems to be $-ve$, and for the ones above 50 the probability seems to be greater than 1. Both conditions are not possible.

Active In simple words the people below 30 are not going to take it, and the ones above 50 will for sure take it.

Logistic Regression



The graph shows the probability for the people between the age of 30 and 50 taking up this offer. While as the age increases there is a greater probability for them taking the offer.

However, for people below 30 the probability seems to be $-ve$, and for the ones above 50 the probability seems to be greater than 1. Both conditions are not possible.

In simple words the people below 30 are not going to take it, and the ones above 50 will for sure take it.

So we cut those bits off and linear regression looks like this.

Logistic Regression

Scientific Explanation for Logistic Regression

Blue box is the equation for simple linear regression.

If we apply Sigmoid Function to this equation which is shown in purple box, can calculate y from it.

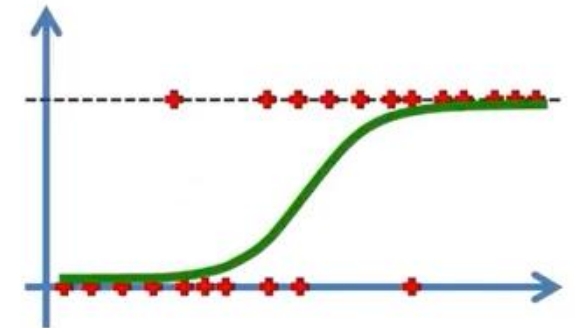
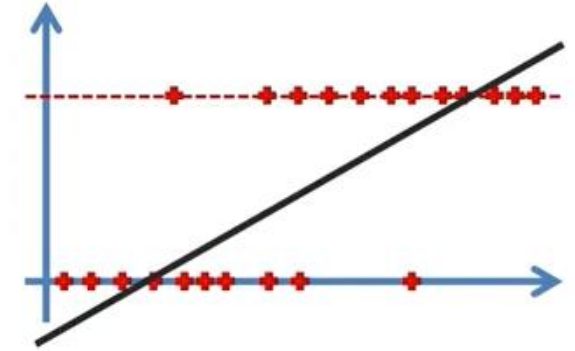
Then we place this y in the blue box and solve it, then we come up with the expression in green box. So, the equation for linear regression would look like this. Which is logistic regression function.

$$y = b_0 + b_1 * x$$

Sigmoid Function

$$p = \frac{1}{1 + e^{-y}}$$

$$\ln \left(\frac{p}{1 - p} \right) = b_0 + b_1 * x$$



Activate Windows

$$p = 1/(1+e^{-y})$$

$$p * (1+e^{-y}) = 1$$

$$1 + e^{-y} = 1/p$$

$$e^{-y} = 1/p - 1$$

$$e^{-y} = (1-p)/p$$

$$e^y = p/(1-p)$$

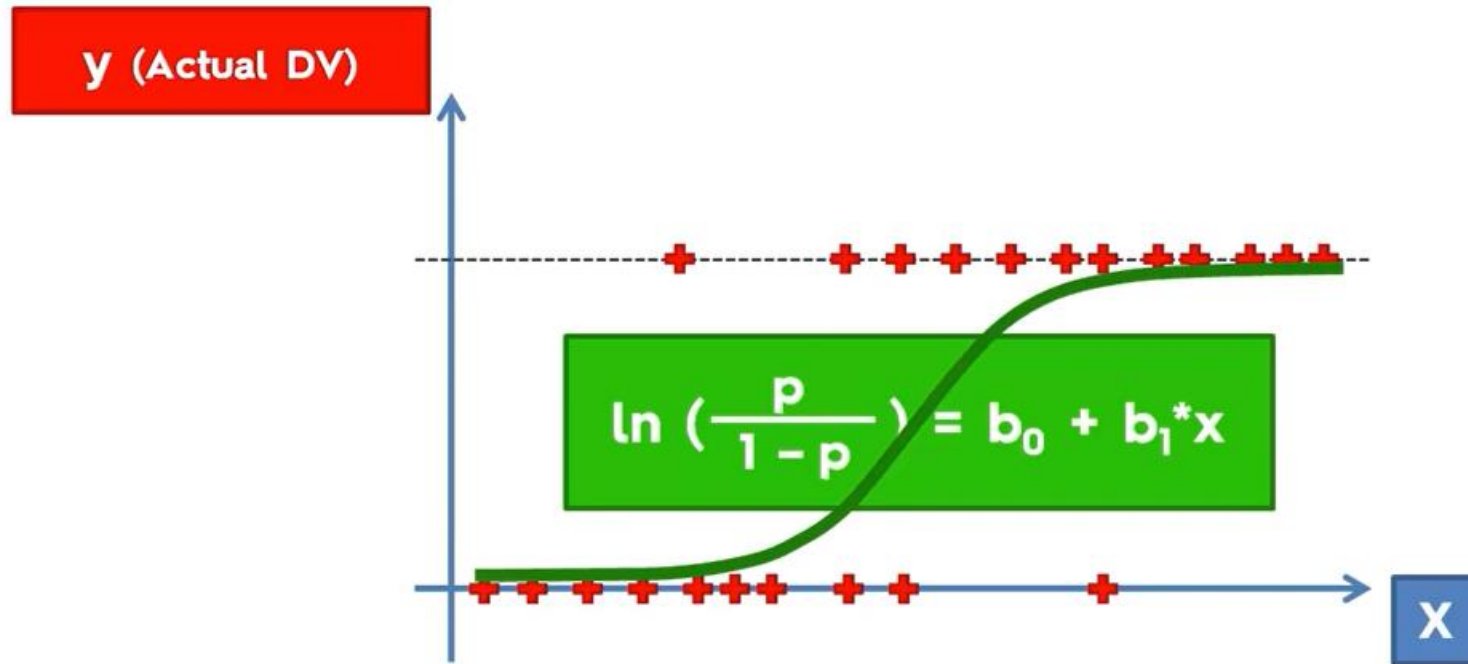
$$\ln(e^y) = \ln(p/(1-p))$$

$$y \cdot \ln(e) = \ln(p/(1-p))$$

$$y \cdot 1 = \ln(p/(1-p))$$

$$y = \ln(p/(1-p))$$

Logistic Regression

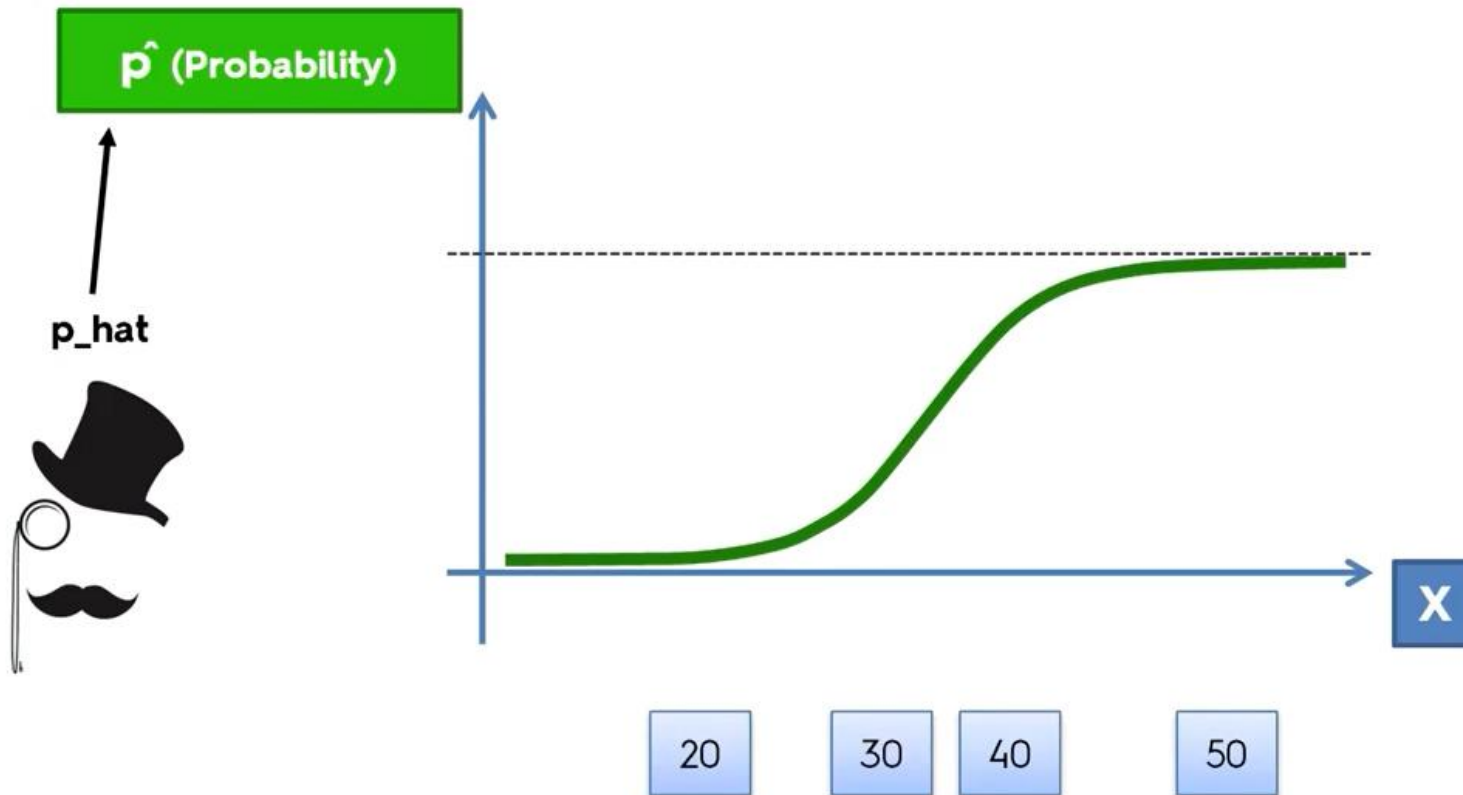


Scientific Explanation for Logistic Regression

The graph shows 'x' the independent variable, 'y' the dependent variable, and a trend-line using the logistic regression formula and the data points.

Again, like simple linear regression we can draw many different lines for the given data and find the best fit line.

Logistic Regression



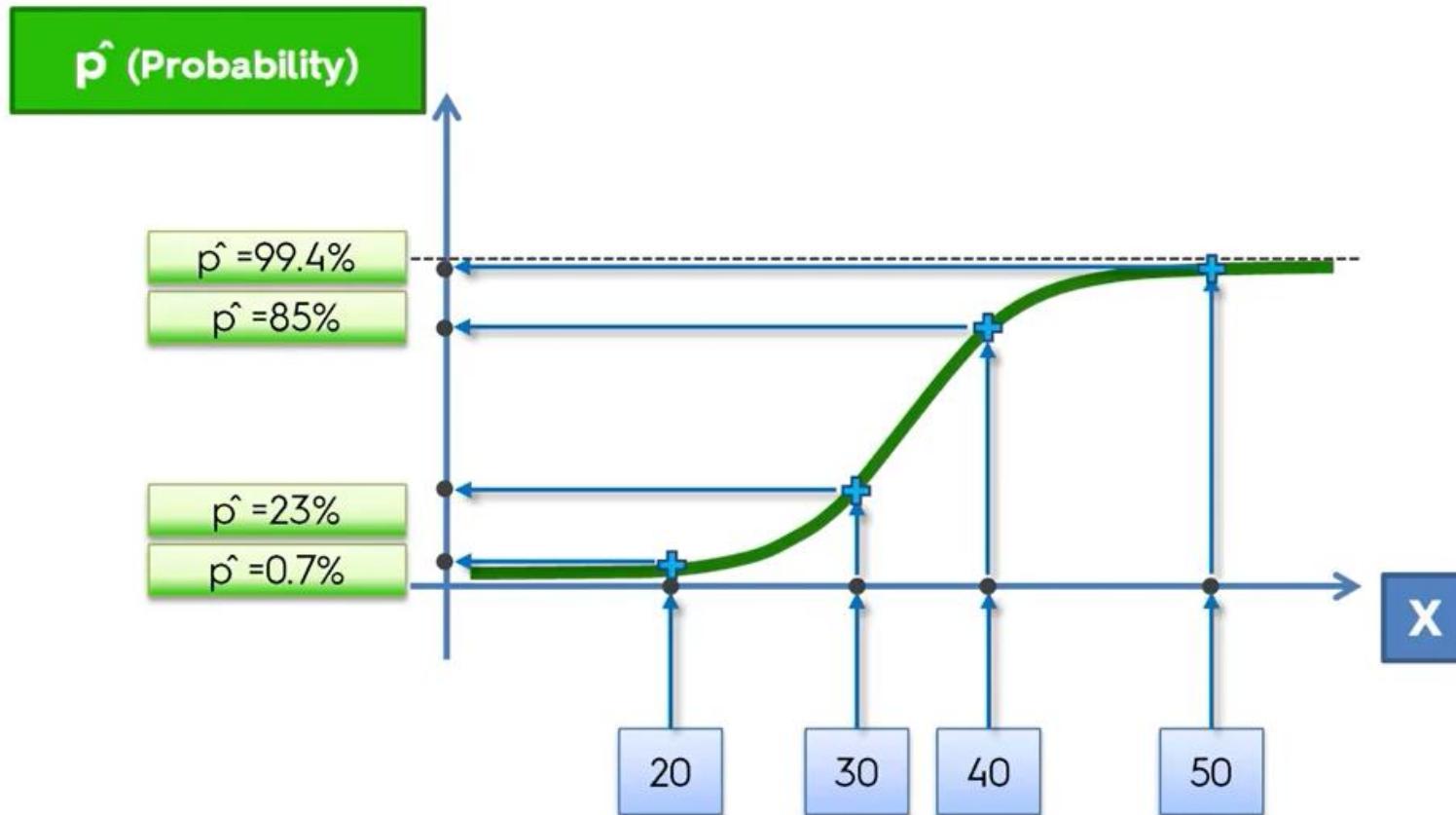
Scientific Explanation for Logistic Regression

The graph shows 'x' the independent variable, 'y' the dependent variable, and a trend-line using the logistic regression formula and the data points.

Again, like simple linear regression we can draw many different lines for the given data and find the best fit line.

We actually calculate probability (\hat{p}) for the y value for different possible x values; i.e. we don't compute y-actual but we calculate y-hat i.e. predicted y variable.

Logistic Regression

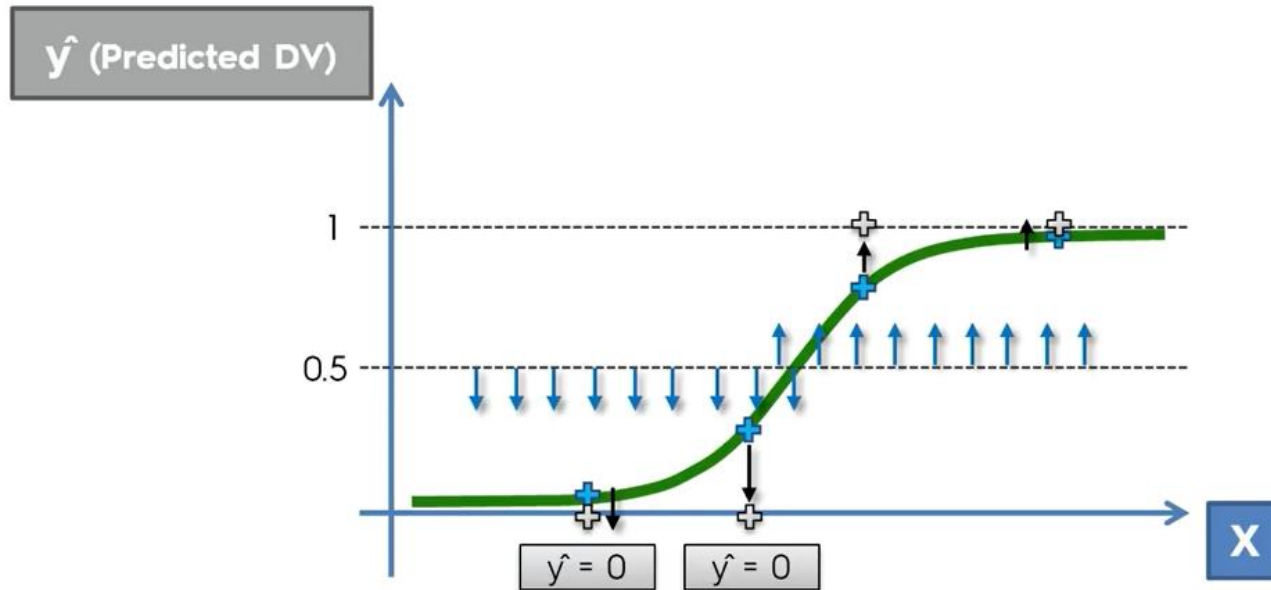


Scientific Explanation for Logistic Regression

As, we actually calculate probability (\hat{p}) for the y value for different possible x values; i.e. we don't compute y -actual but we calculate y -hat i.e. predicted y variable.

Here we can see some \hat{p} values which show the probability of taking up to the offer for different sample age groups.

Logistic Regression



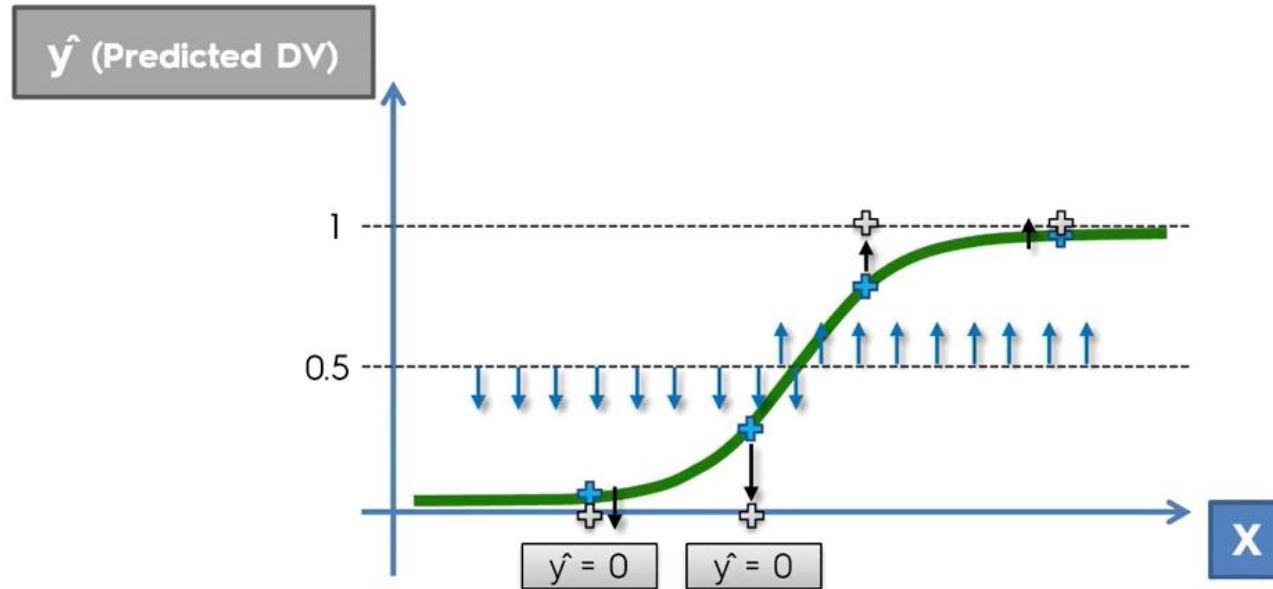
Scientific Explanation for Logistic Regression

As, we actually calculate probability (\hat{p}) for the y value for different possible x values; i.e. we don't compute y -actual but we calculate \hat{y} i.e. predicted y variable.

Here we can see some \hat{p} values which show the probability of taking up to the offer for different sample age groups.

As a matter of fact, we draw this line of probability = 0.5 (in most of the cases), and map all points below to this value to $\hat{y} = 0$; and for all values above it to $\hat{y} = 1$ i.e. they will take this offer.

Logistic Regression



Scientific Explanation for Logistic Regression

It is pertinent to understand that it works like linear regression, i.e. we agree on a line and try to fit a best line for our input data, and try to draw inferences from this line.

We can calculate the probabilities for different events. We can also get the predicted value for the dependent variable based on where we select this arbitrary line i.e. 0.5 in this case.

We can place this line at different positions depending upon the nature of the problem and our domain knowledge to get the best predictions.

Sample Example and Implementation

- Purchase SUV or Not
- Features
 - Age
 - Estimated Salary
- Disregard feature
 - User id
 - Gender

	A	B	C	D	E	F
1	User ID	Gender	Age	Estimated	Purchased	
2	15624510	Male	19	19000	0	
3	15810944	Male	35	20000	0	
4	15668575	Female	26	43000	0	
5	15603246	Female	27	57000	0	
6	15804002	Male	19	76000	0	
7	15728773	Male	27	58000	0	
8	15598044	Female	27	84000	0	
9	15694829	Female	32	150000	1	
10	15600575	Male	25	33000	0	
11	15727311	Female	35	65000	0	
12	15570769	Female	26	80000	0	
13	15606274	Female	26	52000	0	
14	15746139	Male	20	86000	0	
15	15704987	Male	32	18000	0	
16	15628972	Male	18	82000	0	
17	15697686	Male	29	80000	0	
18	15733883	Male	47	25000	1	
19	15617482	Male	45	26000	1	
20	15704583	Male	46	28000	1	

```
1 # Logistic Regression
2
3 # Importing the libraries
4 import numpy as np
5 import matplotlib.pyplot as plt
6 import pandas as pd
7
8 # Importing the dataset
9 dataset = pd.read_csv('Social_Network_Ads.csv')
10 #we intend to make classification decision based on age and salary parameters only
11 X = dataset.iloc[:, [2, 3]].values
12 y = dataset.iloc[:, -1].values
13
14 # Splitting the dataset into the Training set and Test set
15 from sklearn.model_selection import train_test_split
16 X_train, X_test, y_train, y_test = train_test_split(X, y, test_size = 0.25, random_state = 0)
17
18 # Feature Scaling
19 #in this example we need feature scaling because the value ranges differ significantly
20 from sklearn.preprocessing import StandardScaler
21 sc = StandardScaler()
22 X_train = sc.fit_transform(X_train)
23 X_test = sc.transform(X_test)
24
25 # Training the Logistic Regression model on the Training set
26 from sklearn.linear_model import LogisticRegression
27 classifier = LogisticRegression(random_state = 0)
28 classifier.fit(X_train, y_train)
29
```

```

30 # Predicting the Test set results
31 y_pred = classifier.predict(X_test)
32
33 # Making the Confusion Matrix
34 #confusion matrix presents correctly classified data and incorrectly classified data for different classes
35 from sklearn.metrics import confusion_matrix
36 cm = confusion_matrix(y_test, y_pred)
37 print(cm)
38
39 # Visualising the Training set results
40 from matplotlib.colors import ListedColormap
41 X_set, y_set = X_train, y_train
42 X1, X2 = np.meshgrid(np.arange(start = X_set[:, 0].min() - 1, stop = X_set[:, 0].max() + 1, step = 0.01),
43                      np.arange(start = X_set[:, 1].min() - 1, stop = X_set[:, 1].max() + 1, step = 0.01))
44 plt.contourf(X1, X2, classifier.predict(np.array([X1.ravel(), X2.ravel()]).T).reshape(X1.shape),
45             alpha = 0.75, cmap = ListedColormap(('red', 'green')))
46 plt.xlim(X1.min(), X1.max())
47 plt.ylim(X2.min(), X2.max())
48 for i, j in enumerate(np.unique(y_set)):
49     plt.scatter(X_set[y_set == j, 0], X_set[y_set == j, 1],
50               c = ListedColormap(('red', 'green'))(i), label = j)
51 plt.title('Logistic Regression (Training set)')
52 plt.xlabel('Age')
53 plt.ylabel('Estimated Salary')
54 plt.legend()
55 plt.show()
56

```

```

...: print(cm)
[[65  3]
 [ 8 24]]

```

```
56
57 # Visualising the Test set results
58 from matplotlib.colors import ListedColormap
59 X_set, y_set = X_test, y_test
60 X1, X2 = np.meshgrid(np.arange(start = X_set[:, 0].min() - 1, stop = X_set[:, 0].max() + 1, step = 0.01),
61                      np.arange(start = X_set[:, 1].min() - 1, stop = X_set[:, 1].max() + 1, step = 0.01))
62 plt.contourf(X1, X2, classifier.predict(np.array([X1.ravel(), X2.ravel()]).T).reshape(X1.shape),
63             alpha = 0.75, cmap = ListedColormap(('red', 'green')))
64 plt.xlim(X1.min(), X1.max())
65 plt.ylim(X2.min(), X2.max())
66 for i, j in enumerate(np.unique(y_set)):
67     plt.scatter(X_set[y_set == j, 0], X_set[y_set == j, 1],
68               c = ListedColormap(('red', 'green'))(i), label = j)
69 plt.title('Logistic Regression (Test set)')
70 plt.xlabel('Age')
71 plt.ylabel('Estimated Salary')
72 plt.legend()
73 plt.show()
```

```

1 # Logistic Regression
2
3 # Importing the libraries
4 import numpy as np
5 import matplotlib.pyplot as plt
6 import pandas as pd
7
8 # Importing the dataset
9 dataset = pd.read_csv('Social_Network_Ads.csv')
10 #we intend to make classification decision based on age and salary parameters only
11 X = dataset.iloc[:, [2, 3]].values
12 y = dataset.iloc[:, -1].values
13
14 # Splitting the dataset into the Training set and Test set
15 from sklearn.model_selection import train_test_split
16 X_train, X_test, y_train, y_test = train_test_split(X, y, test_size = 0.25, random_state = 0)
17
18 # Feature Scaling
19 #in this example we need feature scaling because the value ranges differ significantly
20 from sklearn.preprocessing import StandardScaler
21 sc = StandardScaler()
22 X_train = sc.fit_transform(X_train)
23 X_test = sc.transform(X_test)
24
25 # Training the Logistic Regression model on the Training set
26 from sklearn.linear_model import LogisticRegression
27 classifier = LogisticRegression(random_state = 0)
28 classifier.fit(X_train, y_train)
29
30 # Predicting the Test set results
31 y_pred = classifier.predict(X_test)
32
33 # Making the Confusion Matrix
34 #confusion matrix presents correctly classified data and incorrectly classified data for different classes
35 from sklearn.metrics import confusion_matrix
36 cm = confusion_matrix(y_test, y_pred)
37 print(cm)
38

```

Name	Type
X	int64
X_test	float64
X_train	float64
cm	int64
dataset	DataFrame
y	int64
y_pred	int64
y_test	int64

Variable explorer File

IPython console

Console 1/A

verbose=0,

In [6]: y_pred = cl

In [7]: from sklear
...: cm = confus
...: print(cm)
[[65 3]
[8 24]]

In [8]: