LOGISTIC REGRESSION

MOHAN M J

INTRODUCTION

- Logistic regression helps us estimate a probability of falling into a certain level of the categorical response given a set of predictors
- For example, we could use logistic regression to model the relationship between various measurements of a manufactured specimen (such as dimensions and chemical composition) to predict if a crack greater than 10 mils will occur (a binary variable: either yes or no)

BINARY LOGISTIC REGRESSION

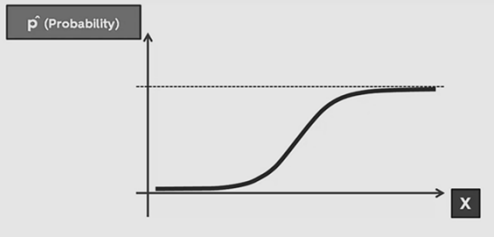
- Used to develop models when the output or response variable y is binary
- The output variable will be binary, coded as either success or failure
- Models' probability of success p which lies between 0 and 1
- If estimate of p>=0.5 then classified as success, otherwise failure

BINARY LOGISTIC REGRESSION

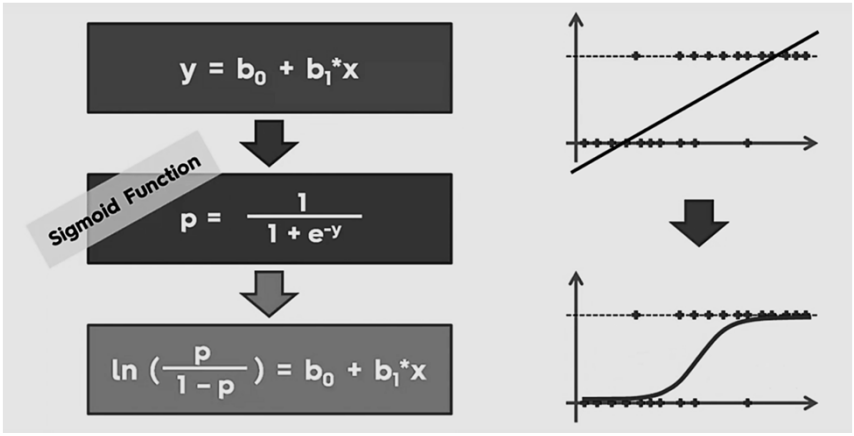
The multiple binary logistic regression model

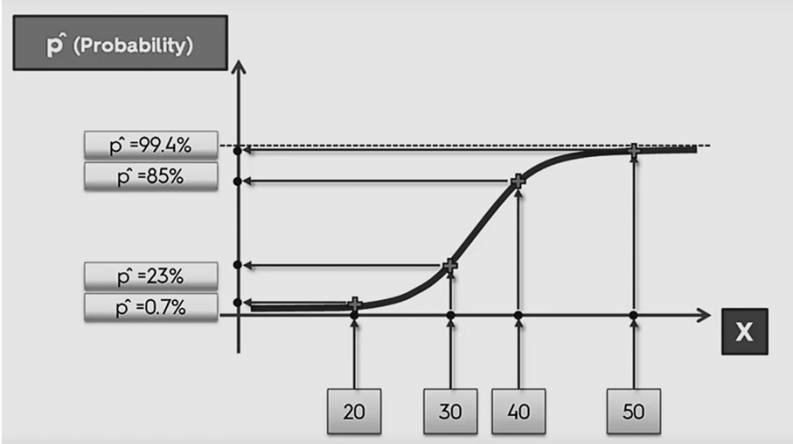
$$\pi = \frac{\exp(\beta_0 + \beta_1 X_1 + \ldots + \beta_{p-1} X_{p-1})}{1 + \exp(\beta_0 + \beta_1 X_1 + \ldots + \beta_{p-1} X_{p-1})} = \frac{\exp(\mathbf{X}\beta)}{1 + \exp(\mathbf{X}\beta)} = \frac{1}{1 + \exp(-\mathbf{X}\beta)}$$

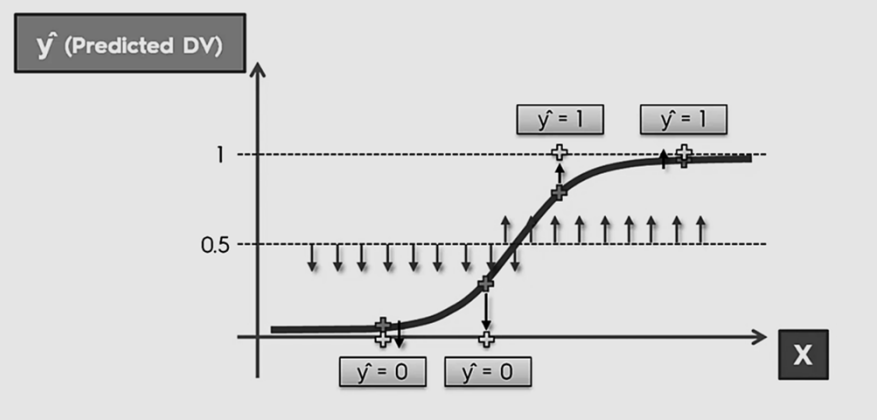
where here π denotes probability



- probability of an event happening as a function of X variables
- estimates of π from equations like the one above will always be between 0 and 1


$y = b_0 + b_1 * x$
Sigmoid Function
$p = \frac{1}{1 + e^{-y}}$
$\ln\left(\frac{p}{1-p}\right) = b_0 + b_1 * x$


p̂ (Probability)
p̂ =99.4%
p̂ =85%
p̂ =23%
p̂ =0.7%
X
20
30
40
50



EXERCISE:

- Develop a model to predict the non payment of overdrafts by customers of a multinational banking institution. The data collected is given in Logistic_Reg.csv

Factors and response considered are given below

1 – Individual expected level of activity score

2 – Transaction speed score

3 – Peer comparison score in terms of transaction volume

Outcome - 0 : Not Paid

1 : Paid

PYTHON CODE

```
import numpy as np
import matplotlib.pyplot as plt
import pandas as pd
# Importing the dataset
myData = pd.read_csv('Logistic_Reg.csv')
myData
X = myData.iloc[:, :-1].values
y = myData.iloc[:, 3].values
```

PYTHON CODE

```
# Splitting the dataset into the Training set and Test set
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, random_state =
0)
# Feature Scaling
from sklearn.preprocessing import StandardScaler
sc = StandardScaler()
X_train = sc.fit_transform(X_train)
X_test = sc.transform(X_test)
```

PYTHON CODE

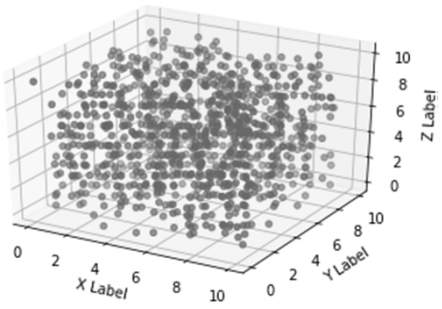
```
# Fitting Logistic Regression to the Training set
from sklearn.linear_model import LogisticRegression
classifier = LogisticRegression(random_state = 0)
classifier.fit(X_train, y_train)
# Predicting the Test set results
y_pred = classifier.predict(X_test)
```

PYTHON CODE

```
#Confusion Matrix
from sklearn.metrics import confusion_matrix
cm = confusion_matrix(y_test, y_pred)
cm
confusion_matrix(y, classifier.predict(sc.fit_transform(X)))
```

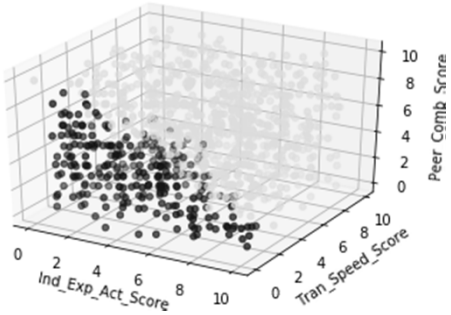
```
array([[ 69,  6],
       [ 0, 170]],
       dtype=int64)

array([[252, 19],
       [ 10, 699]],
       dtype=int64)
```

PYTHON CODE




```
from mpl_toolkits.mplot3d import Axes3D
fig = plt.figure()
ax = fig.add_subplot(111, projection='3d')
ax.scatter(myData.iloc[:,0],myData.iloc[:,1],myData.iloc[:,2])
ax.set_xlabel('X Label')
ax.set_ylabel('Y Label')
ax.set_zlabel('Z Label')
plt.show()
```

PYTHON CODE




```
fig = plt.figure()
ax = fig.add_subplot(111, projection='3d')
ax.scatter(myData.iloc[:,0],myData.iloc[:,1],myData.iloc[:,2],c=y, marker='o')
ax.set_xlabel('Ind_Exp_Act_Score')
ax.set_ylabel('Tran_Speed_Score')
ax.set_zlabel('Peer_Comb_Score')
plt.show()
```

THANK YOU