Chapter 3

Logistic Regression

Linear vs Logistic Regression

- ***Linear regression*** is well suited…
  - *when the response variable y is* CONTINUOS ______________

Revenue from Advertising.

Price of stock

Linear vs Logistic Regression

- ***Linear regression*** is well suited…
  - *when the response variable y is* ***continuous***

Linear vs Logistic Regression

ABSENT PRESENT

NO YES.

- What about when y is **discrete**?
  - *Example: y is **binary**, i.e., $y \in \{0,1\}$*
  - *This is precisely handled by*

LOGISTIC REGRESSION

Linear vs Logistic Regression

- What about when y is **discrete**?
  - *Example: y is **binary**, i.e., $y \in \{0,1\}$*
  - *This is precisely handled by **Logistic Regression***

Linear vs Logistic Regression

- Example
  - ***Image/ video***: PERSON PRESENT/ABSENT
  - ***Medical imaging***: DISEASE PRESENT/ABSENT

Linear vs Logistic Regression

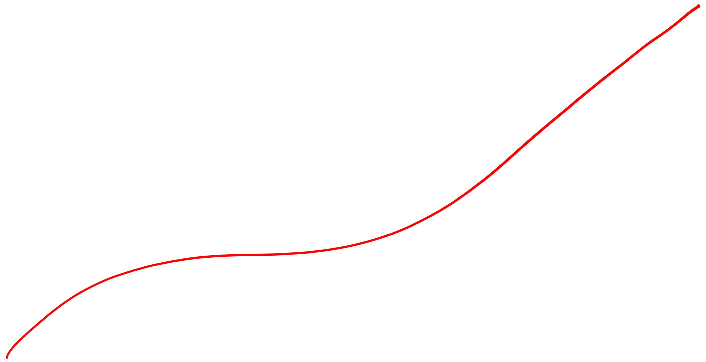
- Example
  - ***Image/ video***: *Person present/ absent*
  - ***Medical imaging***: *Disease present/ absent*

Logistic function

- The **logistic function** is given below

Smooth
Continuos
Differentiable

$$f(z) = \frac{1}{1+e^{-z}}$$



- Also termed the Sigmoid Function

Logistic function

- The **logistic function** is given below

$$f(z) = \frac{1}{1 + e^{-z}}$$

- Also termed the **sigmoid function**

Logistic function

- Observe

$$f(z) = \frac{1}{1+e^{-z}}$$

$$\lim_{z \to \infty} f(z) = 1$$

$$\lim_{z \to -\infty} f(z) = 0$$

$$z \to \infty \Rightarrow e^{-z} \longrightarrow e^{-\infty} = 0$$

$$\Rightarrow \frac{1}{1+e^{-z}} \longrightarrow \frac{1}{1+0} = 1$$

$$z \to -\infty \Rightarrow e^{-z} \to e^{\infty} \to \infty$$

$$\Rightarrow \frac{1}{1+e^{-z}} \longrightarrow \frac{1}{1+\infty} = 0$$

Logistic function

Model Probability

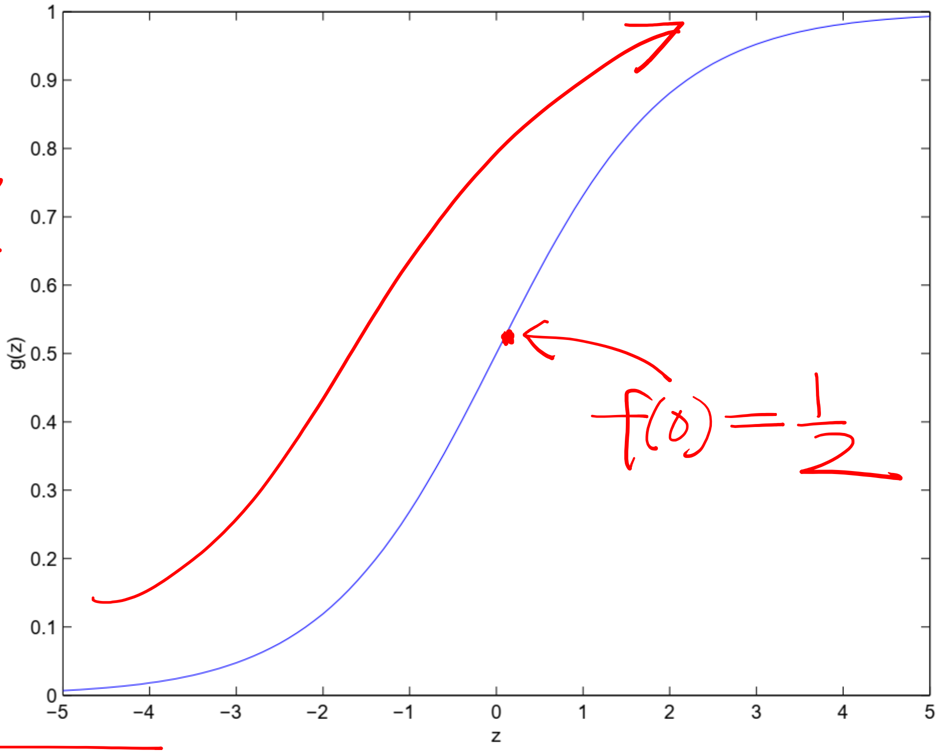
- Observe

$$f(z) = \frac{1}{1+e^{-z}} \to 0 \text{ as } z \to -\infty$$

$$\frac{1}{1+e^{-z}} \to 1 \text{ as } z \to \infty$$

$$f(0) = \frac{1}{1+e^{-0}} = \frac{1}{1+1} = \frac{1}{2}$$

$f(z) \to 1$



$f(0) = \frac{1}{2}$

$0 \leftarrow f(z).$

Probability

- In ***Logistic Regression***, the probabilities are given as

$$P(y = 1|\bar{\mathbf{x}}) = \frac{1}{1 + e^{-\bar{x}^T \bar{h}}}$$

Regression vector

$$\begin{bmatrix} 1 \\ x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix}$$

Regressors

$$\bar{x}^T \bar{h} = h_0 + h_1 x_1 + h_2 x_2 + \cdots + h_n x_n$$

$$\bar{h} = \text{Vector of } \underline{\text{Regression coefficients}} \quad \begin{bmatrix} h_0 \\ h_1 \\ \vdots \\ h_n \end{bmatrix}$$

Probability

- In ***Logistic Regression***, the probabilities are given as

$$P(y = 1|\bar{\mathbf{x}}) = \frac{1}{1 + e^{-\bar{\mathbf{x}}^T \bar{\mathbf{h}}}} = g(\bar{\mathbf{x}})$$

Probability

$$P(y=0|\bar{\mathbf{x}}) = 1 - p(y=1|\bar{x})$$

$$= 1 - \frac{1}{1+e^{-\bar{x}^T \bar{h}}} = \frac{e^{-\bar{x}^T \bar{h}}}{1+e^{-\bar{x}^T \bar{h}}}$$

Probability

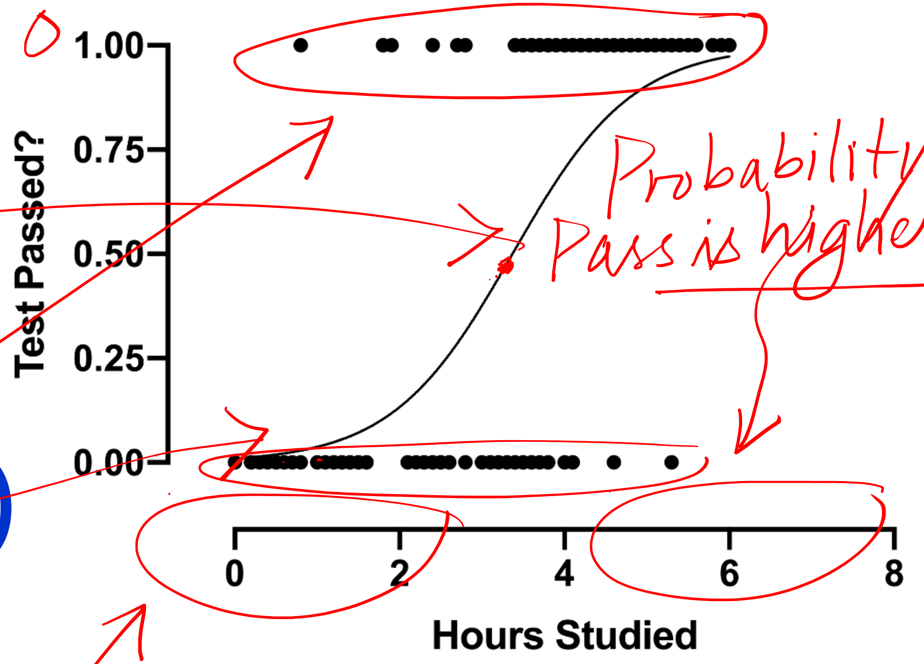
$$P(y=0|\bar{\mathbf{x}}) = \frac{e^{-\bar{\mathbf{x}}^T\bar{\mathbf{h}}}}{1+e^{-\bar{\mathbf{x}}^T\bar{\mathbf{h}}}} = 1 - g(\bar{\mathbf{x}})$$

Probability

Test Pass/Fail
$y=1$ $y=0$

- **Example:**

$$P(\text{Pass}|\text{Hours studied})$$



Probability Pass is higher

Actual Responses.

Probability of Fail is higher

Logistic Regression fit for Probability of Pass/Fail.

Likelihood

Regression parameter vector

- How to determine the **regression parameter** $\bar{\mathbf{h}}$ in this case?
  - We use *the* Maximum Likelihood (ML).

Training set:

$$\left\{\begin{array}{l} \bar{x}(1),\ y(1) \\ \bar{x}(2),\ y(2) \\ \vdots \\ \bar{x}(M),\ y(M) \end{array}\right.$$

Likelihood

- How to determine the **regression parameter** $\bar{\mathbf{h}}$ in this case?
  - *We use the* ***Maximum Likelihood*** *technique*

Maximum Likelihood

- The **update rule** reduces to

iteratively

Step size

Learning rate

$$\bar{h}(k+1) = \bar{h}(k) + \eta\left(y(k+1) - \underline{g(\bar{x}(k+1))\Big|_{\bar{h}=\bar{h}(k)}}\right)\underline{\bar{x}(k+1)}$$

Regression coefficient Vector iteration k

True Response of k+1 training point

Predicted Response

Regression vector k+1

iteration k+1

$$g(\bar{x}(k+1))\Big|_{\bar{h}=\bar{h}(k)} = \underbrace{\frac{1}{1+e^{-\bar{x}^T(k+1)\bar{h}(k)}}}_{\text{Predicted Response for } \bar{x}(k+1)}$$

Predicted Response for $\bar{x}(k+1)$.

$$y(k+1) - g(\bar{x}(k+1))$$

Prediction error $e(k+1)$.

Using current estimate of Regression coefficient vector

Maximum Likelihood

- The **update rule** reduces to

current estimate

current ML model.

Prediction error

$e(k+1)$.

Predicted response by current model

$$\bar{\mathbf{h}}(k+1) = \bar{\mathbf{h}}(k) + \eta\left(y(k+1) - g\left(\bar{\mathbf{x}}(k+1)\right)\Big|_{\bar{\mathbf{h}}=\bar{\mathbf{h}}(k)}\right)\bar{\mathbf{x}}(k+1)$$

True Response

Update:

$$g\left(\bar{\mathbf{x}}(k+1)\right) = \frac{1}{1+e^{-\bar{\mathbf{x}}(k+1)^T\bar{\mathbf{h}}(k)}}$$

$f(\bar{x}^T(k+1)\bar{h}(k))$.

LMS Rule: Least Mean Squares.

- Proof in Appendix

Online Learning Algorithm

```python

import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.metrics import confusion_matrix
```

PANDAS
Datahanding
Library

Scikit-learn
ML Algorithms

Confusion matrix
Metric to characterize
Performance of Logistic
Regression

TO Split data
into test & Training
components

```python
from sklearn.linear_model import LogisticRegression
from sklearn.preprocessing import StandardScaler
import matplotlib.pyplot as plt
import numpy as np
```

LR
Logistic Regression algorithm

Preprocessing Library

For plotting

Scale data
For improved performance
Preprocessing Step

Numerical manipulation

```python
def sigmoid(x):
  return 1/(1 + np.exp(-x))
```

Numpy

$$\underbrace{\frac{1}{1+e^{-x}} = f(x)}_{\text{Sigmoid Logistic Function}}$$

Purchase Dataset

```
purchaseData = pd.read_csv('Purchase_Logistic.csv')
```

Purchase pattern of car

Regressors

Binary. Response

Purchase — 1

Not Purchased — 0

- This dataset contains
- 0 User ID
- 1 Gender
- 2 Age
- 3 Estimated Salary
- Purchased column - Has data as 0 and 1
- where 1 denotes that car is purchased.

```
X = purchaseData.iloc[:, [2, 3]].values
Y = purchaseData.iloc[:, 4].values
```

Age Salary.

Response
Purchase/Not.

```
scaler = StandardScaler();
X = scaler.fit_transform(X)
```

ScaleData X

- Standardize features by removing the mean and scaling to unit variance.

Centering Data.

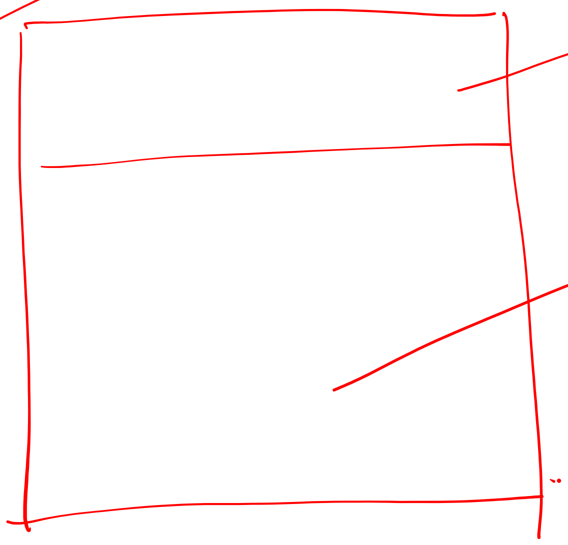
Variance $\sigma^2 = 1$

- To Bring Age and Salary on same scale

Same Scaler

0-100 >100,000

```
Xtrain, Xtest, Ytrain, Ytest \
= train_test_split(X, Y, test_size = 0.25, random_state = 0)
```



Test 25%

Training 75%

Fixes Rows chosen as Test & Training set.

```python
logr = LogisticRegression(random_state = 0)
logr.fit(Xtrain, Ytrain)
Ypred = logr.predict(Xtest)
```

logr contains LR parameters.

initializes iterative Algorithm

Logistic Regression Algorithm

Fit LR for Training Data

Use LR for Prediction of Response for Training data?

```
cmat = confusion_matrix(Ytest, Ypred)
```

```
In [12]: cmat
Out[12]:
array([[65,  3],
       [ 8, 24]], dtype=int64)

In [13]:
```

misclassified

misclassified

confusion matrix } to characterize performance.

Accurately classified.

```python
plt.figure(1);
plt.scatter(X[:, 0], X[:, 1], c = Y)
plt.suptitle('Purchase Data')
plt.xlabel('Scaled Age')
plt.ylabel('Scaled Income')
plt.grid(1,which='both')
plt.axis('tight')
plt.show()
```

Label x axis.

Label y axis.

Grid

Figure 1

Title: Purchase Data

Age

Salary Income

Limits axis Range To data range

c = Response
c = 0, 1

True Data
True Response

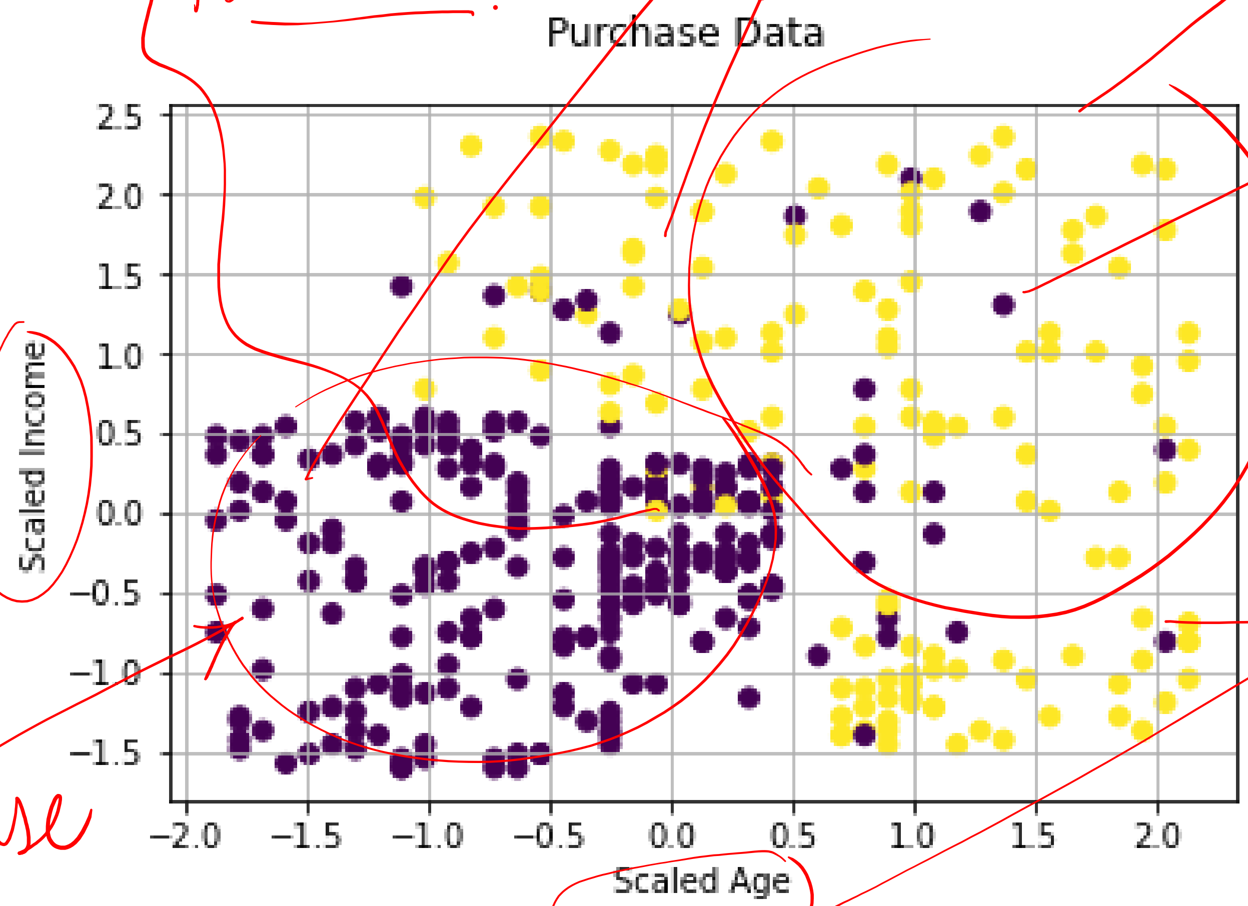
Low Salary
+ Low age
Purchase

0 or 1.
only 2 colors.

Purchase

-1.5 to 2.5

High Salary
+ high age
Not Purchase



Scaled Age
-2 to 2

True
Response

No Purchase

```
col = sigmoid(np.dot(X, np.transpose(logr.coef_)) + logr.intercept_)
cf = logr.coef_;
x1 = np.arange(-1.0,1.2,0.01);
x2 = -(cf[0,0]*x1 + logr.intercept_)/cf[0,1]
```

intercept
Bias

$\bar{x}^T\bar{h}$

$x1$: Scaled age from -1 to 1.2
steps of 0.01.

$P(y=1|\bar{x})$

Coefficients of
Logistic Regression

$$x_2 = \frac{-(h_1 x_1 + h_0)}{h_2} \Rightarrow h_2 x_2 + h_1 x_1 + h_0 = 0$$

$$\bar{x}^T\bar{h} = 0$$

$$\frac{1}{1+e^{-\bar{x}^T\bar{h}}} = \frac{1}{1+e^0} = \frac{1}{2}$$

```
col = sigmoid(np.dot(X, np.transpose(logr.coef_)) + logr.intercept_)
cf = logr.coef_;
x1 = np.arange(-1.0,1.2,0.01);
x2 = -(cf[0,0]*x1 + logr.intercept_)/cf[0,1]
```

Predicted Probability of Purchase

Normalized Salary for each Normalized Age such that $P(\text{purchase}) = \frac{1}{2}$

```python
plt.figure(2);
plt.scatter(X[:, 0], X[:, 1], c = col)
plt.plot(x1,x2,'g')
plt.suptitle('Logistic Regression Purchase Data')
plt.xlabel('Scaled Age')
plt.ylabel('Scaled Income')
plt.grid(1,which='both')
plt.axis('tight')
plt.show()
```

ylabel

Grid

Tight Axes.

Title: Logistic Regression Purchase Data

Scaled Age

Scaled Salary

$P = \frac{1}{2}$ line

Scaled: xlabel Age

Predicted Response using LR

Gradient

Probability ≈ 1

$P = \frac{1}{2}$ line

Prob purchase = Prob Not purchase $= \frac{1}{2}$

Probability ≈ 0

Instructors may use this white area (14.5 cm / 25.4 cm) for the text.
Three options provided below for the font size.

Font: Avenir (Book), Size: 32, Colour: Dark Grey

Font: Avenir (Book), Size: 28, Colour: Dark Grey

Font: Avenir (Book), Size: 24, Colour: Dark Grey

Do not use the space below.

Appendix

Likelihood

- The **likelihood** of $\left(y(k), \bar{\mathbf{x}}(k)\right)$ can be written as

Likelihood

- The **likelihood** of $\left(y(k), \bar{\mathbf{x}}(k)\right)$ can be written as

$$\left(g\left(\bar{\mathbf{x}}(k)\right)\right)^{y(k)}\left(1-g\left(\bar{\mathbf{x}}(k)\right)\right)^{1-y(k)}$$

Likelihood

- The **joint likelihood** of all outputs/ responses is given as

$$L(\bar{\mathbf{h}}) =$$

Likelihood

- The **joint likelihood** of all outputs/ responses is given as

$$L(\bar{\mathbf{h}}) = \prod_{k=1}^{M} \left(g\left(\bar{\mathbf{x}}(k)\right)\right)^{y(k)} \left(1 - g\left(\bar{\mathbf{x}}(k)\right)\right)^{1-y(k)}$$

Likelihood

- The **log-likelihood** is given as

$$\ln L(\bar{\mathbf{h}}) = l(\bar{\mathbf{h}})$$

$$=$$

Likelihood

- The **log-likelihood** is given as

$$\ln L(\bar{\mathbf{h}}) = l(\bar{\mathbf{h}})$$

$$= \sum_{k=1}^{M} y(k) \ln g(\bar{\mathbf{x}}(k)) + (1 - y(k)) \ln\left(1 - g(\bar{\mathbf{x}}(k))\right)$$

Maximum Likelihood

- To maximize the **log-likelihood**, one can employ the ***gradient ascent*** technique

Maximum Likelihood

- The **update rule** reduces to

Maximum Likelihood

- The **update rule** reduces to

$$\bar{\mathbf{h}}(k+1) = \bar{\mathbf{h}}(k) + \eta \left(y(k+1) - g\big(\bar{\mathbf{x}}(k+1)\big)\Big|_{\bar{\mathbf{h}}=\bar{\mathbf{h}}(k)} \right) \bar{\mathbf{x}}(k+1)$$

$$g\big(\bar{\mathbf{x}}(k+1)\big) = \frac{1}{1 + e^{-\bar{\mathbf{x}}(k+1)^T \bar{\mathbf{h}}(k)}}$$