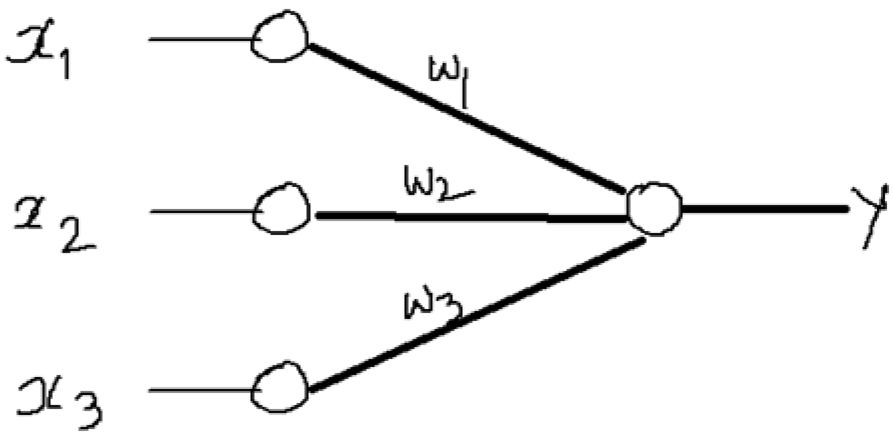
Logistic Regression

Logistic Unit

- If the combination of inputs
  - $\{x_1, x_2, x_3, \ldots, x_n\}$
- Result in a response that is a "categorical variable"
  - With two possible states: 0 and 1
- Then, we have a unit that is known as the
  - Logistic Unit
- And we need a function that will
  - Trigger 0 or 1 as an output, based on the inputs
  - Such a function is known as an **Activation Function**

The process of deciding the predicted value of a categorical variable is known as Classification

Logistic Unit and Logistic Regression



has two possible states $< \begin{matrix} 0 \\ 1 \end{matrix}$

Let $a = w_1x_1 + w_2x_2 + w_3x_3 + b$ (linear combination of x_i)

We need a function that will convert 'a' into either 0 or 1

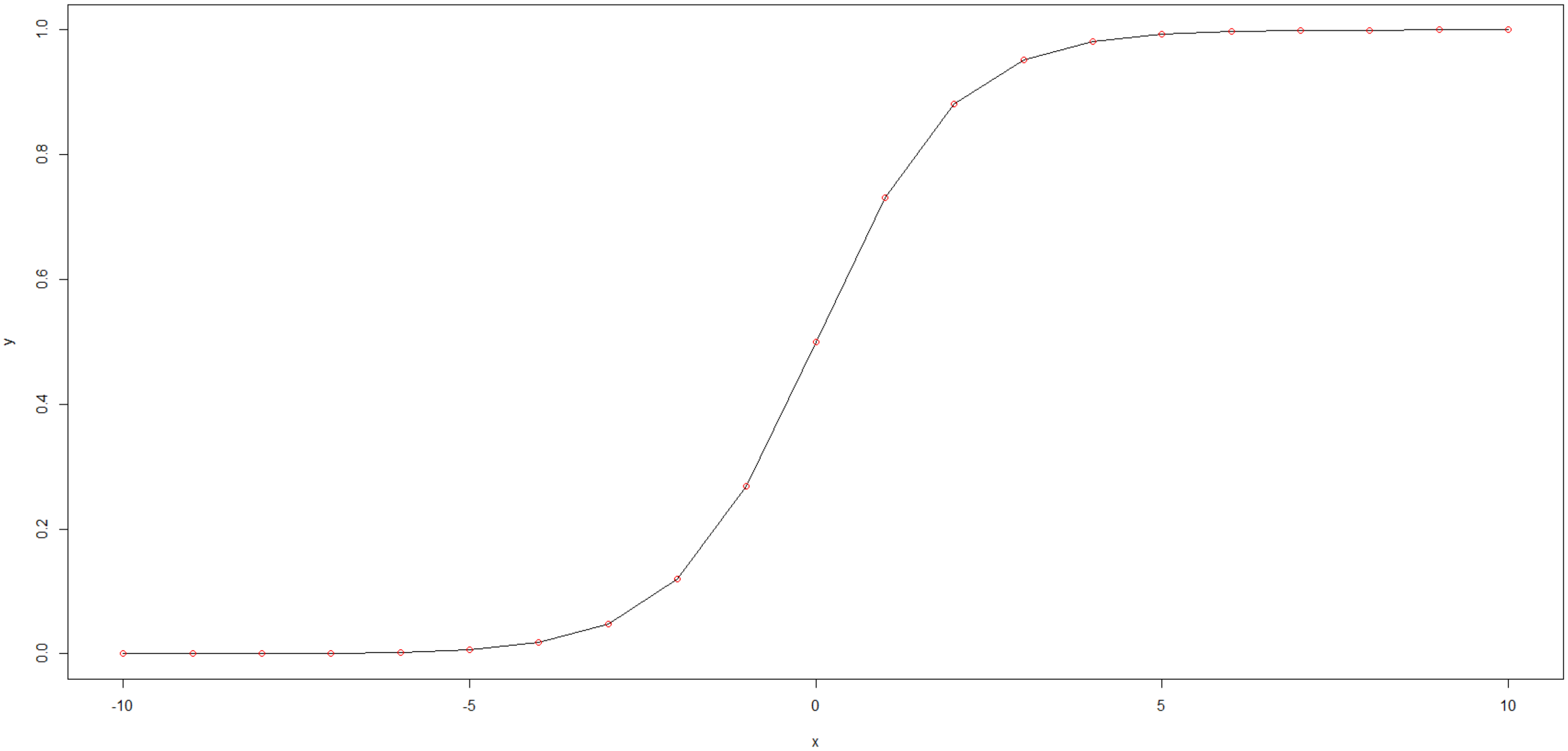
The sigmoid function $\sigma(a) = \dfrac{1}{1+e^{-a}}$ has such a property

Its shape is

We can express $p(y|x) = \sigma(a) = \dfrac{1}{1+\frac{1}{e^a}}$

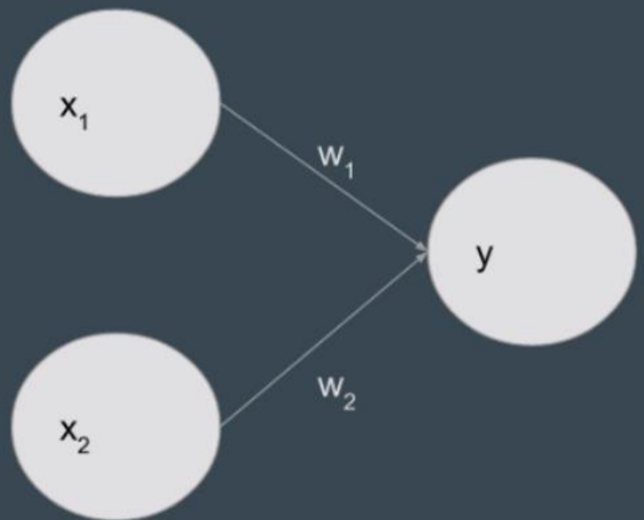
SIGMOID

- $S(a) = 1/(1 + e^{-a})$



Logistic Regression simplified

Logistic Regression

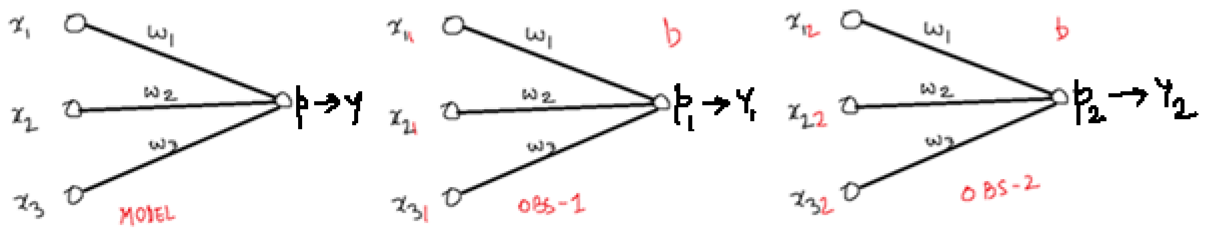


$$a = x_1 w_1 + x_2 w_2 + b$$

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

$$\begin{aligned} \text{prediction} &= \text{round}(p(y|x)) \\ &= 1 \text{ if } p(y|x) > 0.5, \text{ else } 0 \end{aligned}$$

Logistic Regression: Notations



In this model there are 3 features (x_1, x_2, x_3) which result in the output (0, 1). We are interested in finding out $p(y=1|x)$

x_{ij} represent the j^{th} observation (data point) of the i^{th} feature

∴ In the above case we have two input data points or observations

pt1 → (x_{11}, x_{21}, x_{31}) &

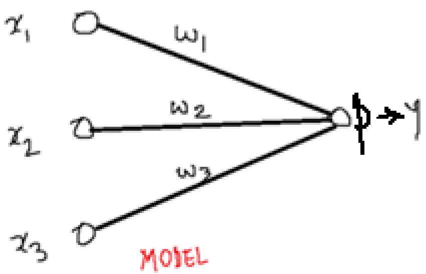
pt2 → (x_{12}, x_{22}, x_{32})

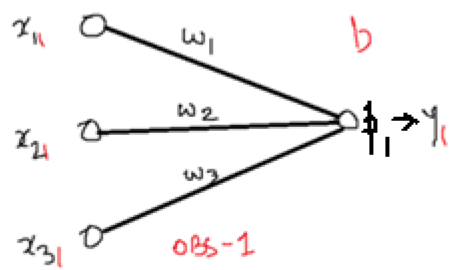
Y_1 & Y_2 are the known outcomes associated with pt_1 & pt_2

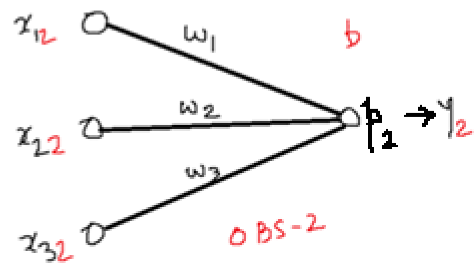
Note:

Known outcomes are also referred to as 't_i'
ie: $t_1 = Y_1$ & $t_2 = Y_2$ in the above case

Matrix Notations: Logistic Regression







$$p_1 = \sigma(w_1 \cdot x_{11} + w_2 \cdot x_{21} + w_3 x_{31} + b)$$

$$\sigma\left\{[w_1 \ w_2 \ w_3] \cdot \begin{bmatrix} x_{11} \\ x_{21} \\ x_{31} \end{bmatrix} + b\right\}$$

$$p_2 = \sigma(w_1 \cdot x_{12} + w_2 \cdot x_{22} + w_3 x_{32} + b)$$

$$= \sigma\left\{[w_1 \ w_2 \ w_3] \cdot \begin{bmatrix} x_{12} \\ x_{22} \\ x_{32} \end{bmatrix} + b\right\}$$

$$p_1 = \sigma\left\{\begin{bmatrix} w_1 \\ w_2 \\ w_3 \end{bmatrix}^T \cdot \begin{bmatrix} x_{11} \\ x_{21} \\ x_{31} \end{bmatrix} + b\right\}$$

$$p_2 = \sigma\left\{\begin{bmatrix} w_1 \\ w_2 \\ w_3 \end{bmatrix}^T \cdot \begin{bmatrix} x_{12} \\ x_{22} \\ x_{32} \end{bmatrix} + b\right\}$$

$$\therefore \begin{bmatrix} p_1 \\ p_2 \end{bmatrix} = \sigma\left\{\begin{bmatrix} w_1 \\ w_2 \\ w_3 \end{bmatrix}^T \cdot \begin{bmatrix} x_{11} & x_{12} \\ x_{21} & x_{22} \\ x_{31} & x_{32} \end{bmatrix} + b\right\}$$

$$p(Y|X) = \sigma\{W^T X + b\}$$

--- Since SIGMOID ranges between $0 \to 1$, RHS can be viewed as the probability of getting a specific Y given X

Matrix Operations

D features $\quad X = D \text{ features} \times N \text{ observations}$

N observations

$p(Y|x) = \text{SCALAR } (1 \times 1)$

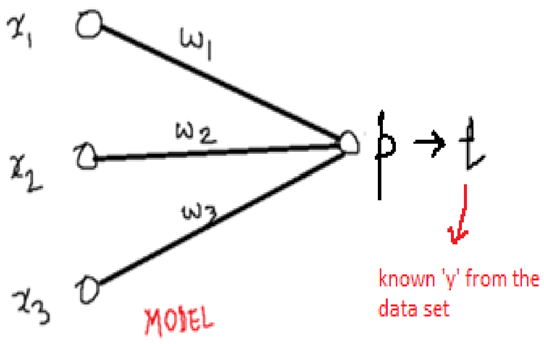
$\text{WEIGHTS} = D \text{ features} \times 1$

$p(Y|x) = \sigma(w^T x + b)$

$= 1 \times D \cdot D \times N$

$= \underline{\underline{1 \times N}}$

Logistic Regression: Calculating the weights w_i



- Given
  - N observations (data points: X)
  - Corresponding targets (t) … see Note below
- Goal
  - Calculate weights w_i
  - Such that the **likelihood** of getting the desired targets is **maximized** given the observations X

- Note:
  - In the **training phase**, input data points x_j and the output y_j are both known. In this case, y_j is known as the target and denoted by t_j
    - The known x and t are used to find w
  - While **predicting**, the input data point x is known and the w's are known, and the corresponding output p and thereby, y = ROUND(p), is found out
    - Recall that p is calculated using the SIGMOID function and ranges from 0-1. It can be therefore viewed as a 'probability' p(y=1|x). If this probability is more than 0.5, the output is set to 1, else 0.

Theory: How to calculate the weights?

Let's consider a logistic regression model with 2 possible output states

Based on the input values of **X** and weights **W** the output can be expressed as

$$p(y|x) = \sigma(w^T x + b)$$

This is the predicted probability

In the data set, every 'y' has two possible values, **0** or **1**

If the probability of y being 1 is denoted as 'p'
then the probability of y being 0 will be '(1-p)'

Therefore the value returned by $\sigma(w^T x)$

should be close to '1' if t = 1

should be close to '0' if t = 0

Theory: How to calculate the weights?

In essence, we want to maximize the likelihood of our predicted outcomes being close to the targets

Now, t = observed outcome $(0, 1)$

p = probability of $t = 1$

$(1-p)$ = probability of $t = 0$

If $t = 1$, we would be interested in maximizing 'p'

If $t = 0$, we should be interested in maximizing $(1-p)$ — A

In order to find 'w', these should be the goals across all the observed data (ie. training set = N observations)

$\therefore$ we define Likelihood $= L = \prod_{n=1}^{N} p_n^{t_n} \cdot (1-p_n)^{(1-t_n)}$

Maximizing 'L' gets us the desired weights (w_i)

observe: Expression for 'L' satisfies requirement A

Theory: How to calculate the weights?

$$L = \prod_{n=1}^{N} p_n^{t_n} (1-p_n)^{(1-t_n)}$$

Now $$\log(L) = \sum_{n=1}^{N} t_n \cdot \log p_n + (1-t_n)\log(1-p_n)$$

$\log(L)$ is known as the "LOG LIKELIHOOD"

Since L and $\log(L)$ are both related monotonically, maximizing the log likelihood $\Rightarrow$ maximizing the likelihood.

We define $$J = -\log(L)$$

$$J = -\sum_{n=1}^{N} t_n \cdot \log p_n + (1-t_n)\log(1-p_n)$$

$\therefore$ minimizing $J \Rightarrow$ maximizing $\log(L) \Rightarrow$ maximizing L

-Log(L) = also known as the "error function"

Maximizing Log Likelihood

Maximizing Likelihood = Minimizing the error function:

$$J = \sum_{n=1}^{N} \left\{ t_n \log p_n + (1-t_n) \log(1-p_n) \right\}$$

Goal:

- Minimize this error function
- Since we want to find out the weights to minimize the error function with respect to weights w. That is:

$$\frac{\partial J}{\partial w}$$

Minimizing the error function (Logistic)

$$J = -\sum_{n=1}^{N} \{ t_n \log p_n + (1-t_n)\log(1-p_n) \}$$

$$\frac{\partial J}{\partial w_i} = -\sum_{n=1}^{N} \frac{\partial J}{\partial p_n} \cdot \frac{\partial p_n}{\partial a_n} \cdot \frac{\partial a_n}{\partial w_i} \quad \therefore a_n = w^T x_n$$

$$\frac{\partial J}{\partial p_n} = t_n \frac{1}{p_n} + (1-t_n)\frac{1}{(1-p_n)} \cdot (-1)$$

$$\boxed{\frac{\partial J}{\partial p_n} = \frac{t_n}{p_n} - \frac{1-t_n}{1-p_n}}$$

$$p_n = \sigma(a_n) = \frac{1}{1+e^{-a_n}} \quad \text{and} \quad \boxed{(1-p_n) = 1 - \frac{1}{1+e^{-a_n}} = \frac{e^{-a_n}}{1+e^{-a_n}}}$$

$$\frac{\partial p_n}{\partial a_n} = \frac{-1}{(1+e^{-a_n})^2}(e^{-a_n})(-1)$$

$$\frac{\partial p_n}{\partial a_n} = \frac{e^{-a_n}}{(1+e^{-a_n})^2} = \frac{1}{(1+e^{-a_n})} \cdot \frac{e^{-a_n}}{(1+e^{-a_n})}$$

$$\boxed{\frac{\partial p_n}{\partial a_n} = p_n(1-p_n)}$$

$$a_n = w^T x_n = w_0 + w_1 x_{n1} + w_2 x_{n2} + \cdots$$

$$\boxed{\frac{\partial a_n}{\partial w_i} = x_{ni}}$$

$$\therefore \frac{\partial J}{\partial w_i} = -\sum \left\{ \left(\frac{t_n}{p_n} - \frac{1-t_n}{1-p_n} \right) \times p_n(1-p_n) \times x_{ni} \right\}$$

$$= -\sum \{ (t_n(1-p_n) - p_n(1-t_n)) \cdot x_{ni} \}$$

$$= -\sum (t_n - t_n p_n - p_n + p_n t_n) \cdot x_{ni}$$

$$= -\sum (t_n - p_n) \cdot x_{ni}$$

$$\frac{\partial J}{\partial w} = \sum (p_n - t_n) \cdot x_{i}$$

$$\boxed{\frac{\partial J}{\partial w} = X^T(P - T)} \quad \text{OR} \quad \boxed{\frac{\partial J}{\partial w} = (P-T)^T X}$$

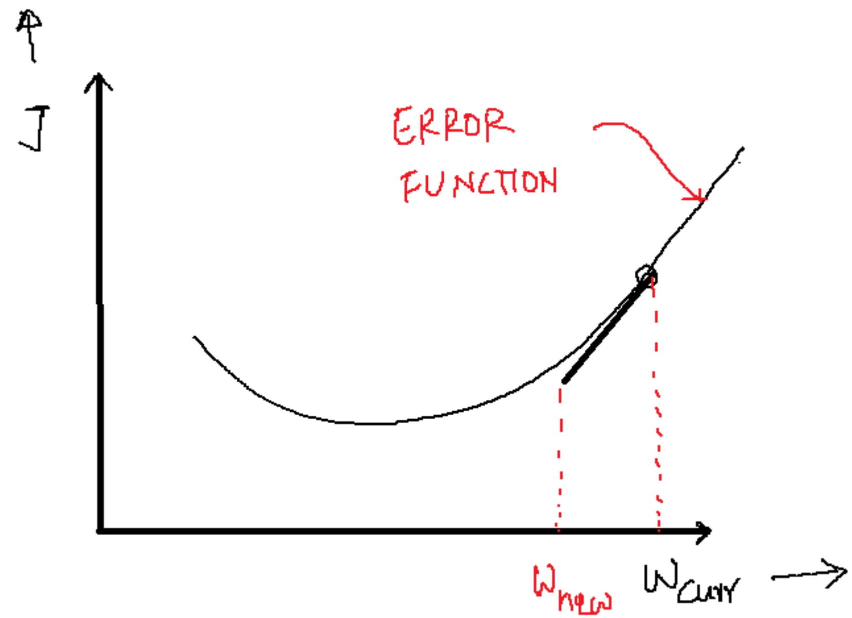
Calculating the weights w_i

- Now that we have an expression for minimizing the error function with respect to w_i
- We can now begin the process of calculating the weights themselves
- This is an iterative procedure known as the "Gradient Descent Method" and it works as follows: (also refer next slide)
  1. Initialize **w** randomly
  2. Find out the predicted **p**
  3. Find out gradient $\frac{\partial J}{\partial w} = x^T (p - t)$
  4. Descend along the gradient to get new weights
  5. Repeat until termination criteria is reached

Basis of the Gradient Descent method

- The error function is given by:

$$J = \sum_{n=1}^{N} \{ t_n \log p_n + (1 - t_n) \log(1 - p_n) \}$$



- The gradient of this error function is:

$$\frac{\partial J}{\partial w} = X^T (P - T)$$

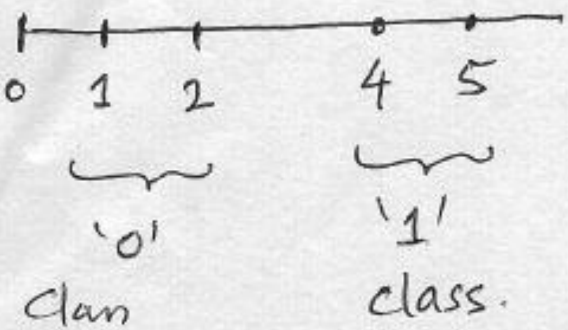
- In gradient descent, the weights are updated as:

$$W_{new} = W_{curr} - \eta \cdot \frac{dJ}{dW}$$

$$w \leftarrow w - \eta \nabla J$$

$$\boxed{\eta \longrightarrow \text{LEARNING RATE}}$$

Logistic Regression: Example



- One dimensional problem
  - only one feature: x.
- The observations

observation (x)		class (t)
1	$\rightarrow$	0
2	$\rightarrow$	0
4	$\rightarrow$	1
5	$\rightarrow$	1

Problem: Find out a function such that

$Y = f(x)$ results in the above classification

Logistic Regression: Example

In the context of our example

$$\begin{array}{ll} \text{obs} & [1 \quad 2 \quad 4 \quad 5] \\ & \Downarrow \\ t & [0 \quad 0 \quad 1 \quad 1] \end{array}$$

using this data we want to teach our model.

⇓ Create a model, and

$$\begin{array}{ll} Y & \begin{bmatrix} Y_1 & Y_2 & Y_3 & Y_4 \end{bmatrix} \\ & \;\;(0|1) \;\; (0|1) \;\; (0|1) \;\; (0|1) \end{array}$$

Predict the outcome using the model

$$P(Y|x) = \sigma(w^T x)$$

Solution steps:

1. Start by assuming some values for W
2. Find out $\frac{\partial J}{\partial w}$
3. $w_{new} = w_{old} - \eta \cdot \nabla J$
4. Iterate until not much difference between w_{new} & w_{old}

or determined ~~set~~ number of steps are executed.

Logistic Regression: Example

In our example,

$$a = \begin{bmatrix} w_0 \\ w_1 \\ w_2 \\ \vdots \end{bmatrix}^T \begin{bmatrix} 1 & 1 & \cdots & 1 \\ x_{11} & & & \vdots \\ x_{21} & & & \vdots \\ \vdots & & & \vdots \\ x_{k1} & & & x_{kn} \end{bmatrix}$$

Since we have only one variable, this takes the form

$$a = \begin{bmatrix} w_0 \\ w_1 \end{bmatrix}^T \begin{bmatrix} 1 & 1 & 1 & 1 \\ x_{11} & x_{12} & x_{13} & x_{14} \end{bmatrix}$$

$$= \begin{bmatrix} 1 \\ 1 \end{bmatrix}^T \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 2 & 4 & 5 \end{bmatrix} = \begin{bmatrix} 1 & 1 \end{bmatrix} \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 2 & 4 & 5 \end{bmatrix}$$

$$\therefore a = \begin{bmatrix} 2 & 3 & 5 & 6 \end{bmatrix}$$

Logistic Regression: Example

$$p(y|x) = \sigma(a) = \sigma\left(\begin{bmatrix} 2 & 3 & 5 & 6 \end{bmatrix}\right)$$

$$p = \begin{bmatrix} 0.88 & 0.95 & 0.993 & 0.9975 \end{bmatrix}$$

$$T = \begin{bmatrix} 0 & 0 & 1 & 1 \end{bmatrix}$$

$$(p - T) = \begin{bmatrix} 0.88 & 0.95 & -0.007 & -0.0025 \end{bmatrix}$$

$$\frac{\partial J}{\partial w} = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 2 & 4 & 5 \end{bmatrix} \begin{bmatrix} 0.88 & 0.95 & -0.007 & -0.0025 \end{bmatrix}^T$$

$$= \begin{bmatrix} 1.8242 \\ 2.7468 \end{bmatrix}$$

$$w_{new} = \begin{bmatrix} 1 \\ 1 \end{bmatrix} - 0.1 \times \begin{bmatrix} 1.8242 \\ 2.7468 \end{bmatrix}$$

$$w_{new} = \begin{bmatrix} 0.817 \\ 0.7253 \end{bmatrix} \quad \text{---- Iteration 1.}$$

Logistic Regression: Example

Iteration 2

$$a = w_{new}^{T} \cdot X = \begin{bmatrix} 0.817 \\ 0.7253 \end{bmatrix}^{T} \cdot \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 2 & 4 & 5 \end{bmatrix}$$

$$a = \begin{bmatrix} 1.543 & 2.268 & 3.7188 & 4.444 \end{bmatrix}$$

$$p(Y|x) = \sigma(a) = \begin{bmatrix} 0.8239 & 0.9062 & 0.9763 & 0.988 \end{bmatrix}$$

$$(P - T) = \begin{bmatrix} 0.8239 & 0.9062 & -0.0237 & -0.0116 \end{bmatrix}$$

$$\frac{\partial J}{\partial w} = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 2 & 4 & 5 \end{bmatrix} \cdot \begin{bmatrix} 0.8239 & 0.9062 & -0.0237 & -0.0116 \end{bmatrix}^{T}$$

$$\frac{\partial J}{\partial w} = \begin{bmatrix} 1.69 \\ 2.48 \end{bmatrix}$$

$$w_{new} = \begin{bmatrix} 0.817 \\ 0.7253 \end{bmatrix} - 0.1 \times \begin{bmatrix} 1.69 \\ 2.48 \end{bmatrix}$$

$$w_{new} = \begin{bmatrix} 0.648 \\ 0.476 \end{bmatrix}$$

Logistic Regression: Example

Iteration 3.

$$a = \begin{bmatrix} 1.125 & 1.602 & 2.556 & 3.033 \end{bmatrix}$$

$$p = \begin{bmatrix} 0.7549 & 0.8323 & 0.9279 & 0.9540 \end{bmatrix}$$

$$(p - T) = \begin{bmatrix} 0.7549 & 0.8323 & -0.072 & -0.0459 \end{bmatrix}$$

$$\partial J / \partial w = \begin{bmatrix} 1.469 \\ 1.901 \end{bmatrix}$$

$$w_{new} = \begin{bmatrix} 0.501 \\ 0.286 \end{bmatrix}$$

Iteration 25 $\quad w_{new} = \begin{bmatrix} -1.040 \\ 0.570 \end{bmatrix}$

$$p = \begin{bmatrix} 0.384 & 0.525 & 0.776 & 0.859 \end{bmatrix}$$

$$y = \begin{bmatrix} 0 & 1 & 1 & 1 \end{bmatrix}$$

Logistic Regression: Example

Iteration 50 $\quad w_{new} = \begin{bmatrix} -2.165 \\ 0.9105 \end{bmatrix}$

$$p = [0.22 \quad 0.414 \quad 0.814 \quad 0.915]$$

$$y = [0 \quad 0 \quad 1 \quad 1]$$

Iteration 100 $\quad w_{new} = \begin{bmatrix} -3.582 \\ 1.353 \end{bmatrix}$

$$p = [0.09 \quad 0.294 \quad 0.861 \quad 0.960]$$

$$y = [0 \quad 0 \quad 1 \quad 1]$$

Iteration

$$\boxed{a = -3.582 + 1.353 \cdot x}$$

Logistic Regression: Example

Let take different Values of x and find out the p's and y's.

x	a	$p = \sigma(a)$	y	t
-5	-10.347	3.2×10^{-5}	0	
0	-3.582	2.7×10^{-2}	0	---
1	-2.229	9.717×10^{-2}	0	--- 0
2	-0.876	$2.94 \times 10^{-1} = 0.294$	0	--- 0
3	0.477	$6.17 \times 10^{-1} = 0.617$	1	
4	1.830	0.861	1	--- 1
5	3.183	0.960	1	--- 1
6	4.536	0.989	1	
10	9.948	0.999	1	

Logistic Regression: Quality Metrics

	Predicted: NO	Predicted: YES
Actual: NO	TN	FP
Actual: YES	FN	TP

CONFUSION MATRIX

- **Accuracy:** Overall, how often is the classifier correct?
  - (TP+TN)/total
- **Misclassification Rate:** Overall, how often is it wrong?
  - (FP+FN)/total
  - equivalent to 1 minus Accuracy
  - also known as "Error Rate"
- **True Positive Rate:** When it's actually yes, how often does it predict yes?
  - TP/actual yes
  - also known as "Sensitivity" or "Recall"
- **False Positive Rate:** When it's actually no, how often does it predict yes?
  - FP/actual no
- **True Negative Rate:** When it's actually no, how often does it predict no?
  - TN/actual no
  - equivalent to 1 minus False Positive Rate
  - also known as "Specificity"
- **Precision:** When it predicts yes, how often is it correct?
  - TP/predicted yes
- **Prevalence:** How often does the yes condition actually occur in our sample?
  - actual yes/total

ROC Curves

Confusion Matrix can be created for various threshold values of classification probability

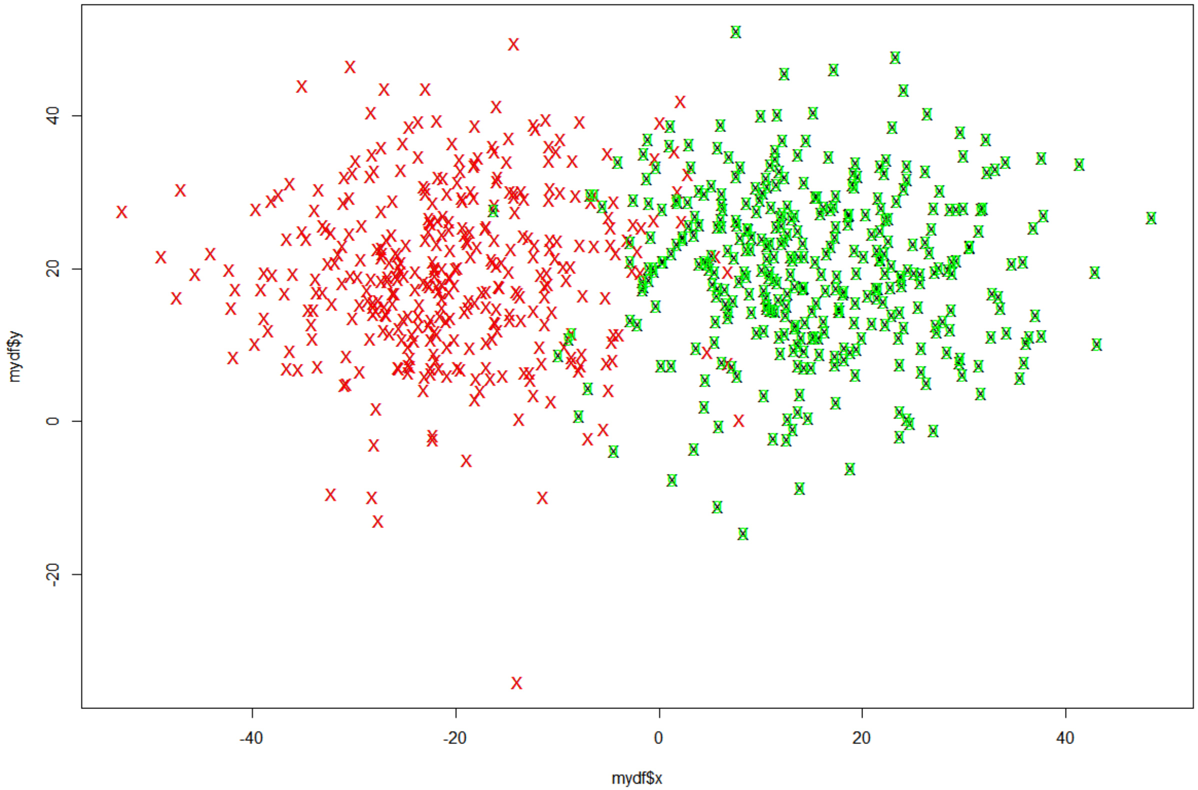
Default threshold value of probability is '0.5'

if $p < 0.5 \Rightarrow$ classification is '0'

if $p \geq 0.5 \Rightarrow$ classification is '1'

For every value of 'p', the corresponding TPR & FPR can be calculated & plotted to get the ROC curve

Logistic Regression Example



This figure shows visual depiction of a data set (x1, x2, y). Here, x1 and x2 are independent variables and y is the dependent variable - and it takes only two values: 0 or 1.

As y is a discrete variable, this problem is one of **classification** and a classification model can be created using the method of **Logistic Regression**

red points $\Rightarrow y=0$
green points $\Rightarrow y=1$
each point has coordinates (x_{1n}, x_{2n})

Confusion Matrix v/s Threshold Values

```
          Reference
Prediction   0   1
         0   0   0
         1 360 360
```

```
          Reference
Prediction   0   1
         0 341  10
         1  19 350
```

```
          Reference
Prediction   0   1
         0 341  12
         1  19 348
```

```
          Reference
Prediction   0   1
         0 343  14
         1  17 346
```

```
          Reference
Prediction   0   1
         0 343  15
         1  17 345
```

```
          Reference
Prediction   0   1
         0 345  16
         1  15 344
```

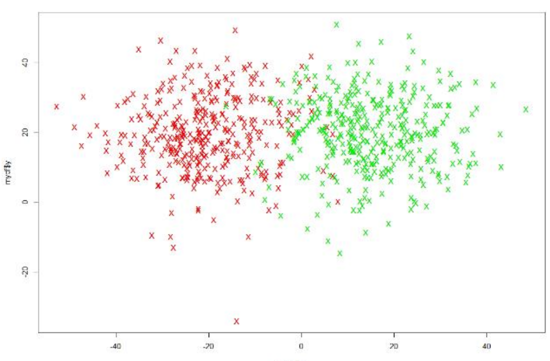
```
          Reference
Prediction   0   1
         0 346  19
         1  14 341
```

```
          Reference
Prediction   0   1
         0 347  19
         1  13 341
```

```
          Reference
Prediction   0   1
         0 347  21
         1  13 339
```

```
          Reference
Prediction   0   1
         0 348  31
         1  12 329
```

```
          Reference
Prediction   0   1
         0 360 318
         1   0  42
```



Recollect that y_pred = ROUND(p) ie. y_pred is obtained by rounding of p to the nearest integer (0 or 1) depending on a threshold value of 'p'.

If p <= 0.5, y = 0
if > 0.5, y =1

This default threshold results in a default confusion matrix.

If we vary the threshold value, the confusion matrix will change.

The ones alongside correspond to threshold values ranging from 0 to 1 in increments of 0.1

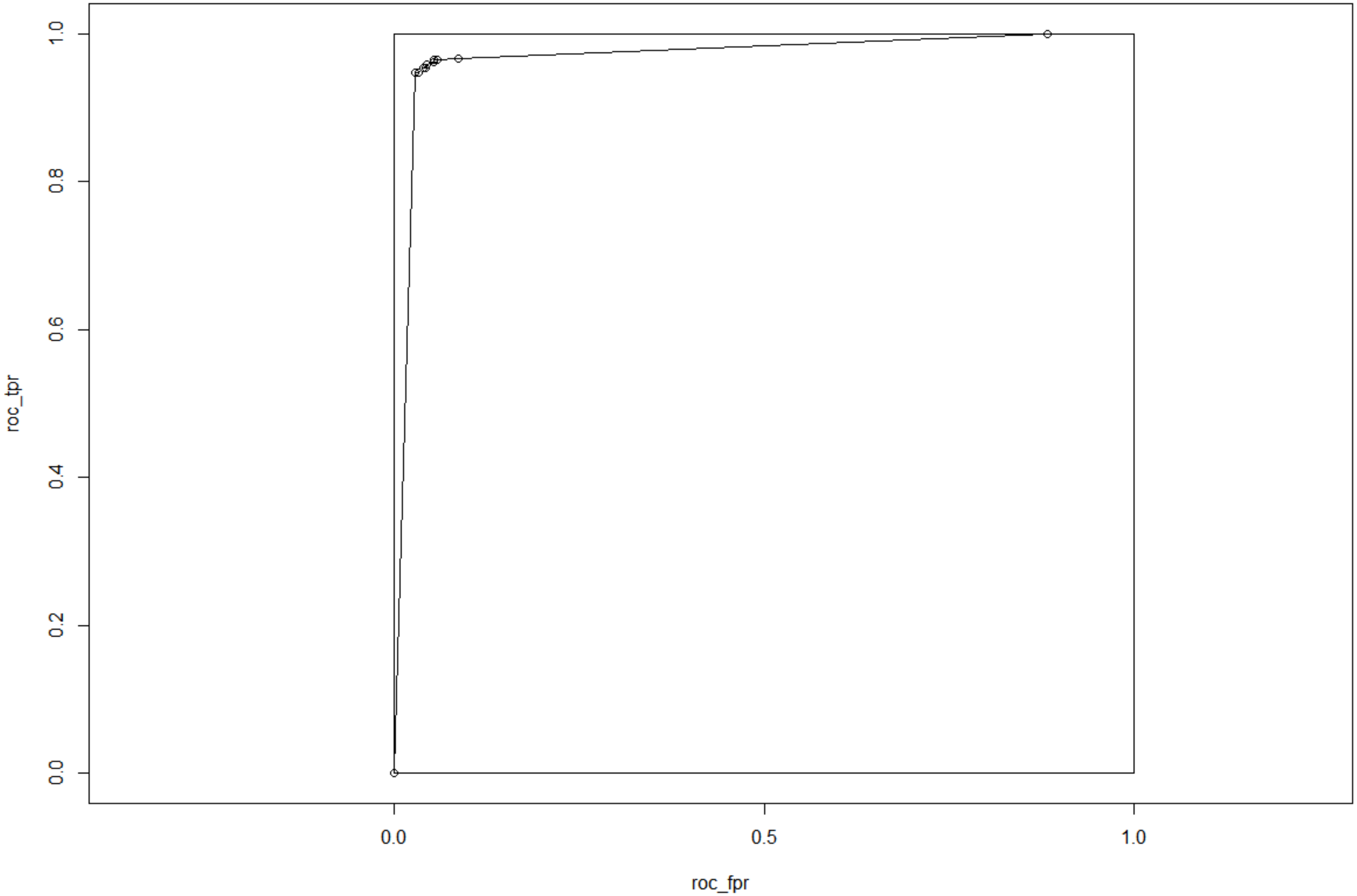
Logistic Regression Metrics v/s Threshold

Threshold	Accuracy	Sensitivity (TPR)	Specificity (TNR)	(FPR)
0.0	0.5	0.0	1.0	0.0
0.1	0.9597	0.9472	0.9722	0.0278
0.2	0.9569	0.9472	0.9667	0.0333
0.3	0.9569	0.9528	0.9611	0.0389
0.4	0.9556	0.9528	0.9583	0.0417
0.5	0.9569	0.9583	0.9556	0.0444
0.6	0.9542	0.9611	0.9472	0.0528
0.7	0.9556	0.9639	0.9472	0.0528
0.8	0.9528	0.9639	0.9417	0.0583
0.9	0.9403	0.9667	0.9139	0.0861
1.0	0.5583	1.000	0.1167	0.8833

The ROC Plot

An ROC (Receiver Operating Characteristic) plot is created by varying the threshold value of 'p' and calculating the corresponding values of TPR and FPR.

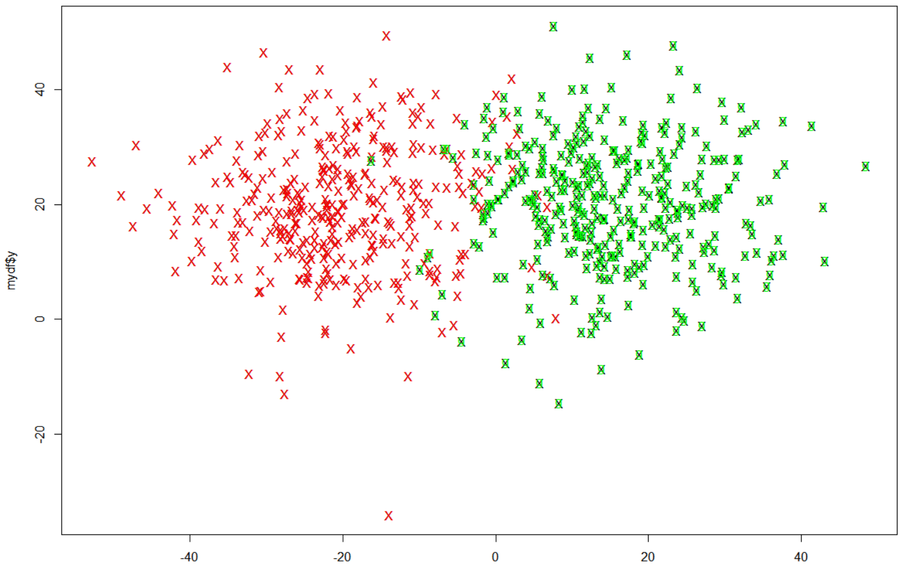
Every point on the ROC ie (FPR, TPR) corresponds to a unique probability threshold value

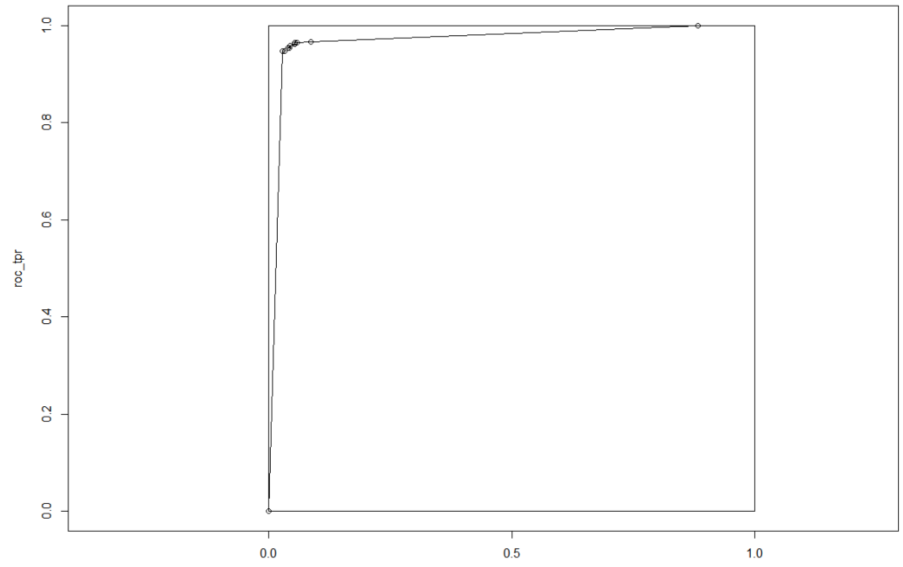


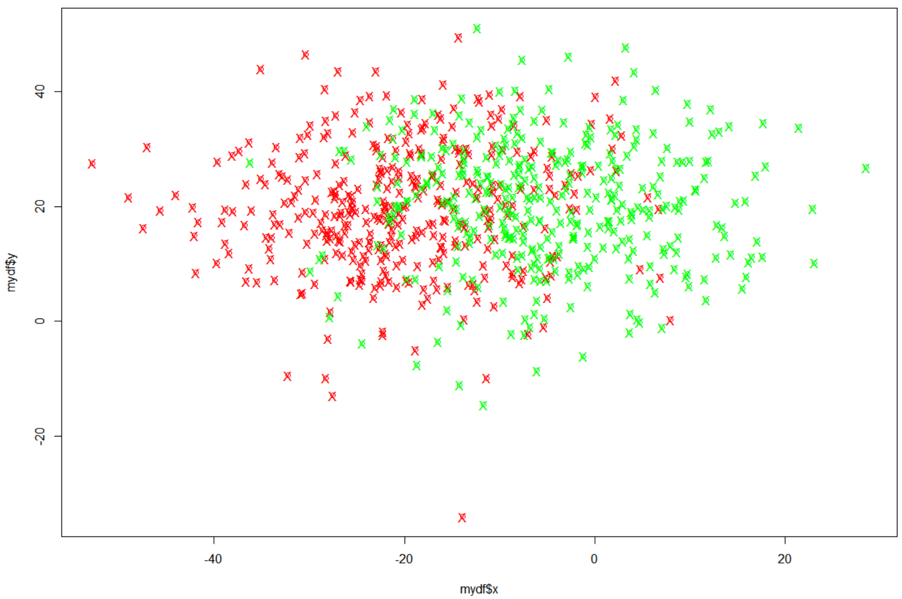
The ROC Plot

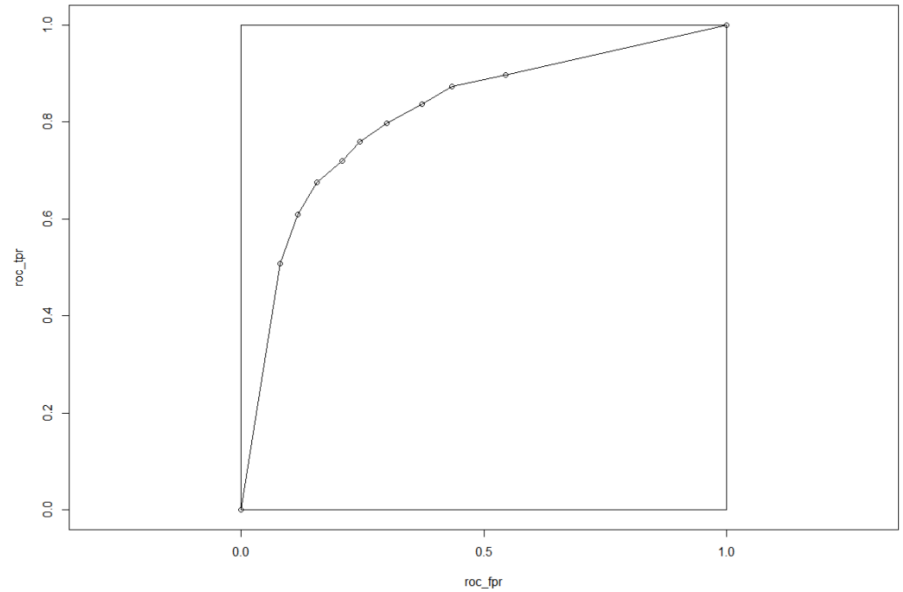
- The ROC plot indicates the quality of the classifier.
- It reflects the fact that a good classification system (classifier) will exhaust all correct classifications (ie. True Positives) before it starts making wrong classifications (ie. False Positives)
- Therefore, in a good classifier, the TPR values should reach close to 1 before the FPR values start increasing.
- In other words, the ROC curve of a good classifier will sharply rise vertically, before it moves horizontally.
- Therefore, in a good classifier, the AUC (Area Under the Curve) of an ROC plot will be close to 1.

Data Sets v/s ROC Plots

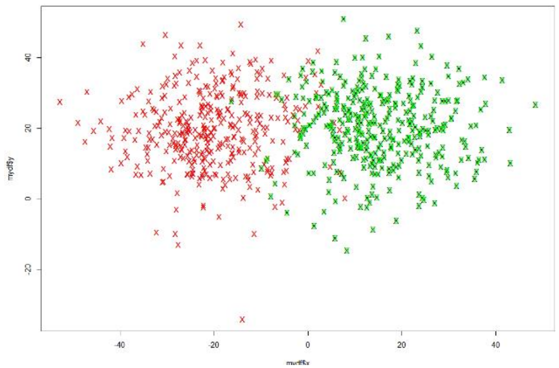


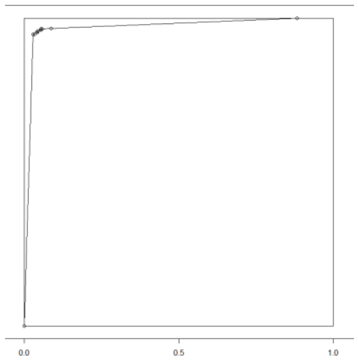


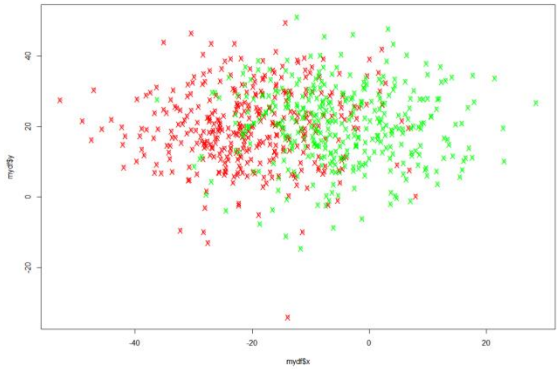


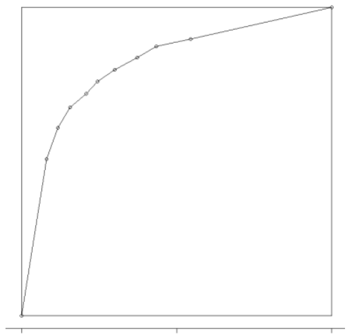


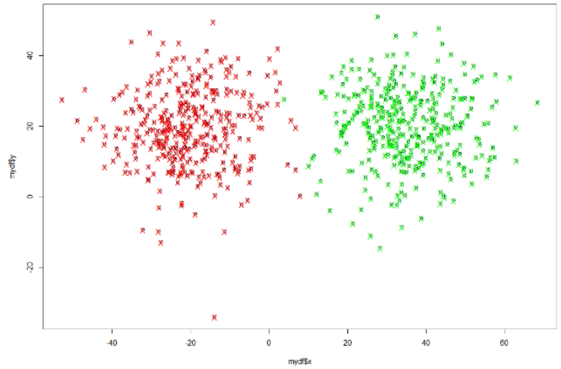
Data Sets v/s ROC Plots

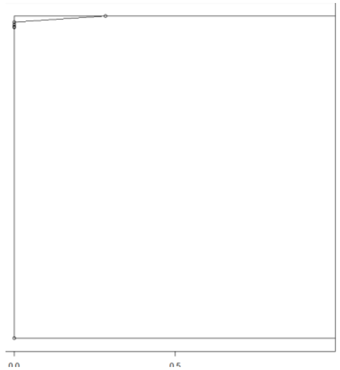




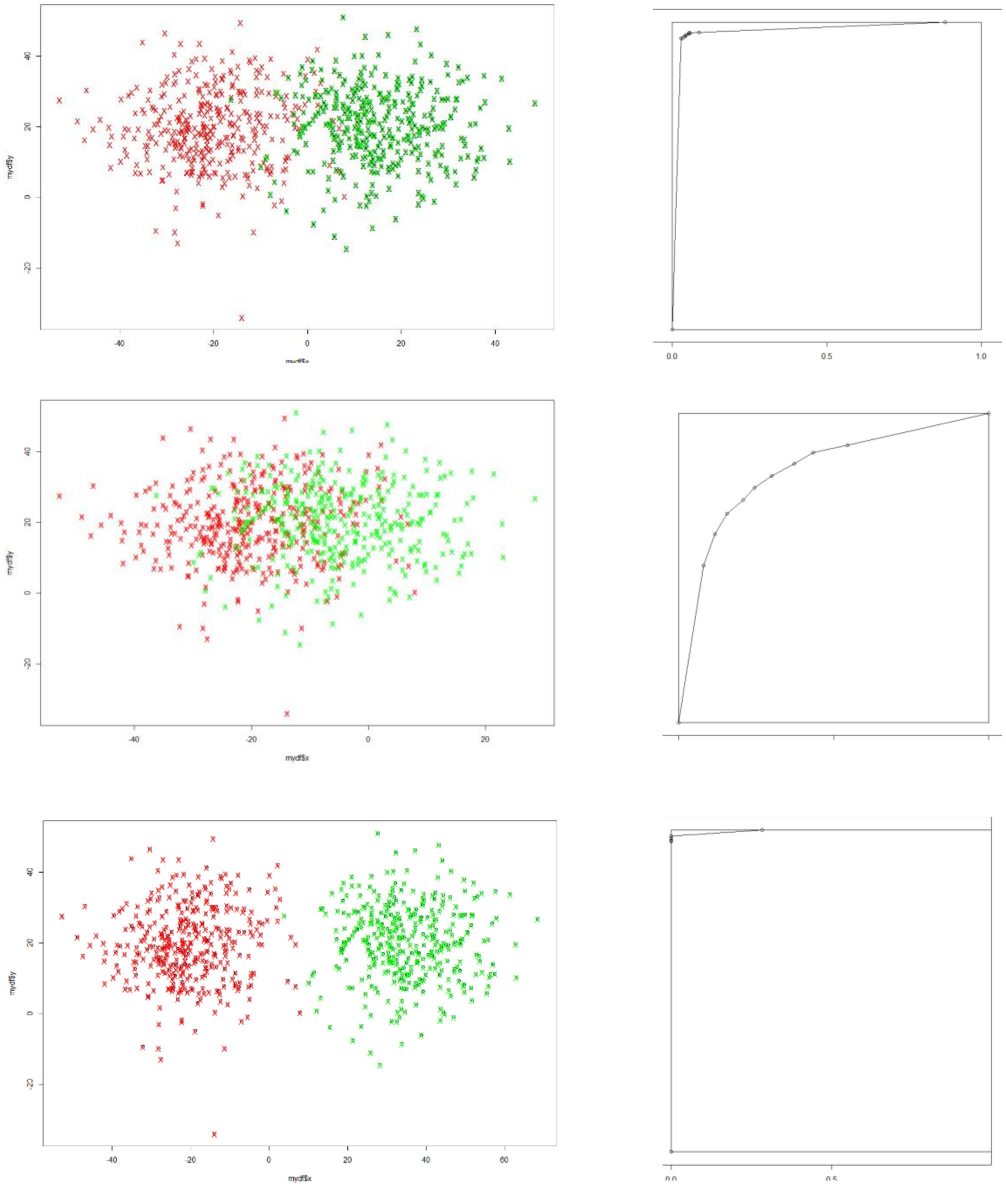








Data Sets v/s ROC Plots



Quality of the classifier is indicated by the area under the ROC curve (known as AUC).

AUC should be as close to '1' as possible