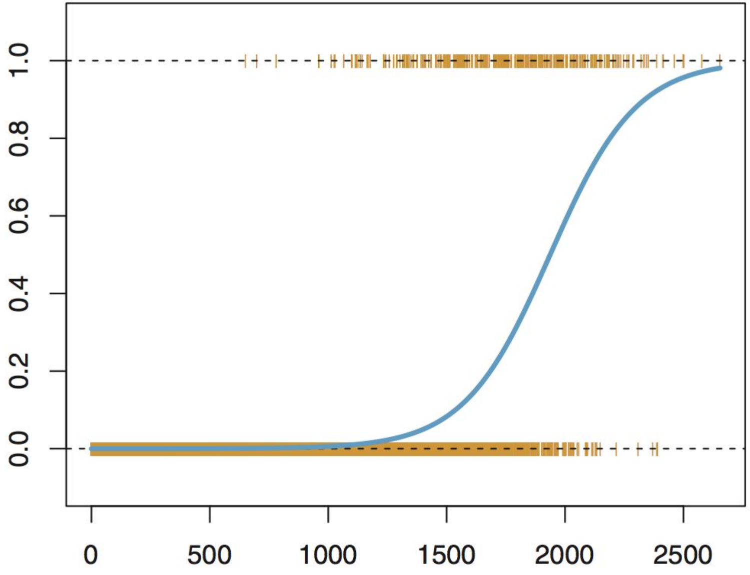
Logistic Regression

Classification, metrics and ROC curves

DSI, jf.omhover



Logistic Regression

Classification, metrics and ROC curves

DSI, jf.omhover

OBJECTIVES (morning)

- **Relate** Regression to Classification in the context of supervised learning
- **Compare** Logistic Regression to Linear Regression
- **Define** and **compute** metrics for evaluating classifiers
- **Describe** the process for computing parameter values in LogReg
- **Use** the parameters of a LogReg model to **compute** the class of an obverstion

Supervised Learning

Learning / Estimating FUNCTIONS based on examples

Reality VS Model : assumptions and learning

REALITY

OBJECTIVE:
descriptive
predictive
normative
...

MODEL

	type	income	education	prestige
accountant	prof	62	86	82
pilot	prof	72	76	83
architect	prof	75	92	90
author	prof	55	90	76
chemist	prof	64	86	90
minister	prof	21	84	87
professor	prof	64	93	93
dentist	prof	80	100	90
reporter	wc	67	87	52
engineer	prof	72	86	88
undertaker	prof	42	74	57
lawyer	prof	76	98	89

data

$$(x_1, y_1)$$

...

$$(x_n, y_n)$$

x y

$$y = f(x) + \epsilon$$

take a function as an assumption

$$\sum (y_i - \hat{f}(x_i))^2$$

COST FUNCTION

$$\hat{y} = \hat{f}(x)$$

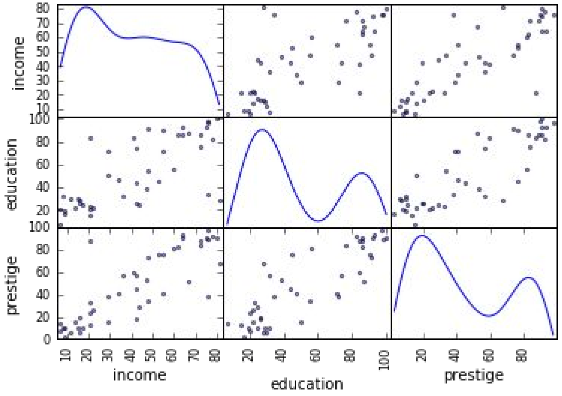
Estimator of the function

Linear Regression - General Process

REALITY

MODEL

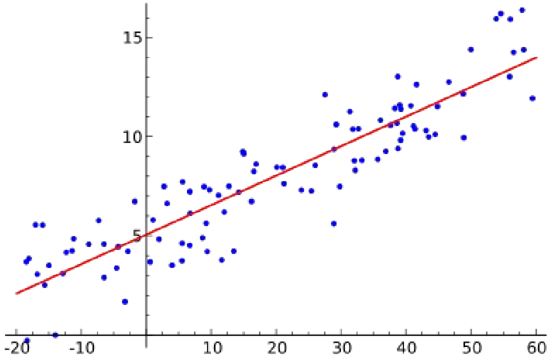
1) Having a data sample
Observing an underlying behavior



2) Make an assumption
on the <u>model</u> underlying the data

$$y \approx \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \ldots + \beta_p x_p$$

linear relation
(+ assumptions)



3) **<u>Find</u>** the instance of the model
that **<u>fits</u>** with data sample

Multi-Linear Regression

COST FUNCTION (Residual Sum of Squares)

$$RSS(\beta) = (y - X\beta)^T(y - X\beta)$$

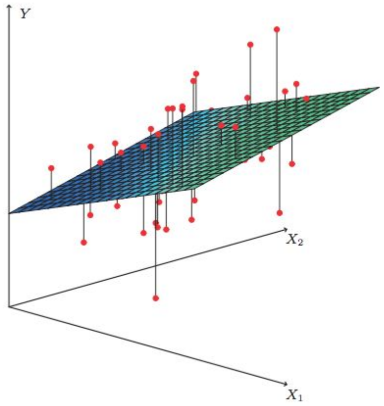
O.L.S.

REALITY

MODEL

DATA

$$X = \begin{bmatrix} 1 & x_{1,1} & \cdots & x_{1,p-1} \\ 1 & x_{2,1} & \cdots & x_{2,p-1} \\ \vdots & \vdots & & \vdots \\ 1 & x_{n,1} & \cdots & x_{n,p-1} \end{bmatrix} \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{bmatrix} = y$$



model class

$$y \approx X\beta$$

PROBLEM

$$\hat{y} = X\hat{\beta}$$

$$\beta = \begin{bmatrix} \beta_0 \\ \beta_1 \\ \beta_2 \\ \vdots \\ \beta_{p-1} \end{bmatrix}$$

model instance estimator parameters

SOLUTION

$$\boxed{\hat{\beta} = (X^TX)^{-1}X^Ty}$$

Figure from [ISL]

Classification

Learning / Estimating “models of classes” based on examples

Reality vs Model : assumptions and learning

REALITY

OBJECTIVE:
descriptive
predictive
normative
...

MODEL

$$y = f(x) + \epsilon$$

take a function as
an assumption

$$X = \begin{bmatrix} 1 & x_{1,1} & \cdots & x_{1,p-1} \\ 1 & x_{2,1} & \cdots & x_{2,p-1} \\ \vdots & \vdots & & \vdots \\ 1 & x_{n,1} & \cdots & x_{n,p-1} \end{bmatrix} \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{bmatrix} = y$$

$$\Sigma(y_i - \hat{f}(x_i))^2$$

COST FUNCTION

$$\hat{y} = \hat{f}(x)$$

Estimator
of the function

Mapping // Classification algorithms

Logistic Regression

k-NN

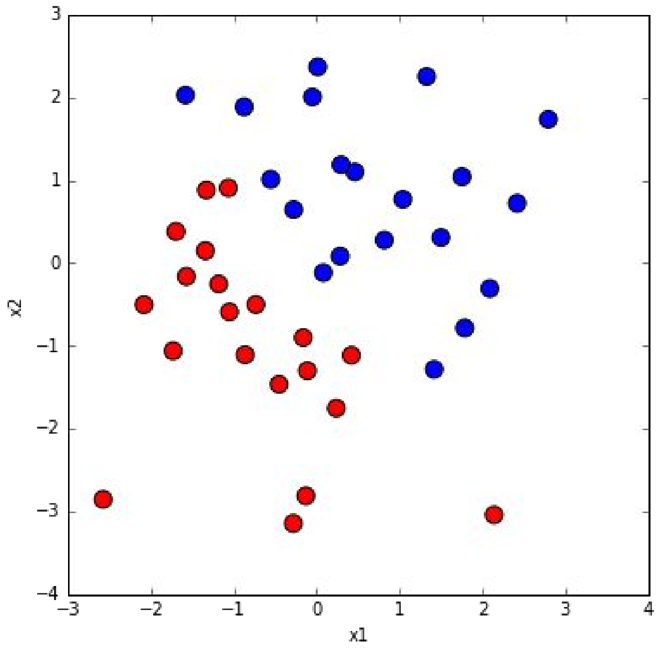
Decision Trees

Random Forest, Boosting

Support Vector Machines (SVM)

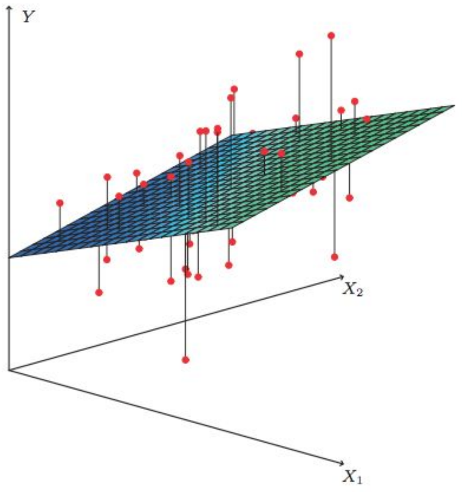
Neural Networks

...

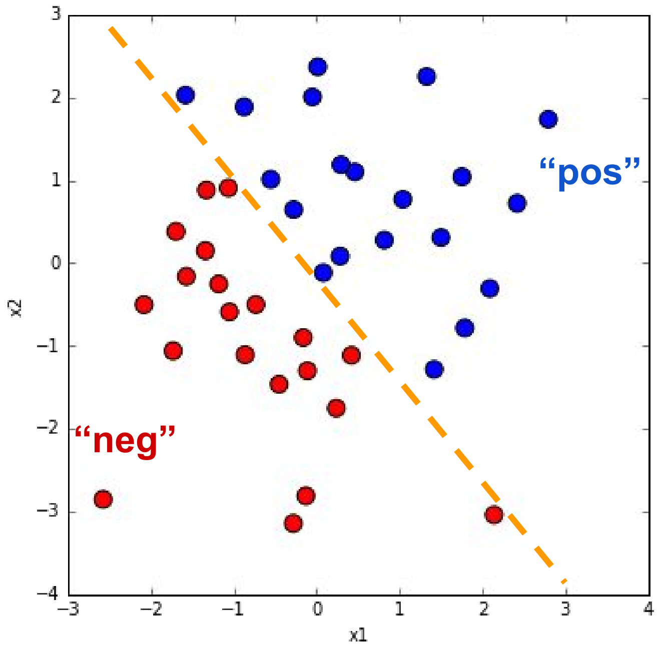


Regression vs Classification

Quantitative response y in R



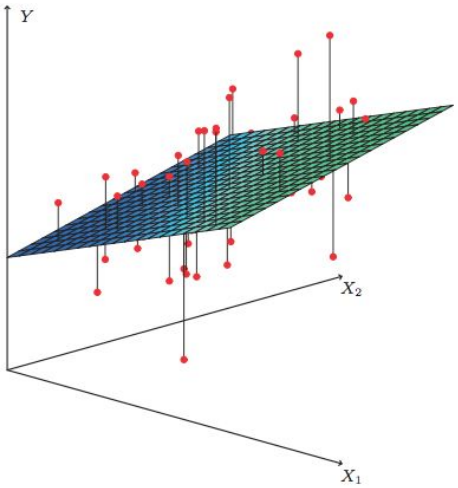
Categorial response y



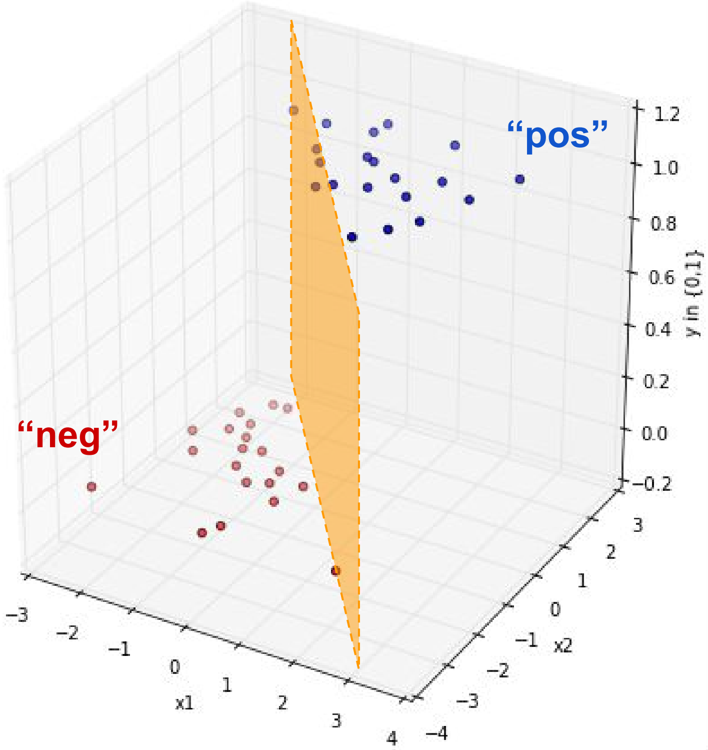
Assigning y = 0 to neg, y = 1 to pos

Regression vs Classification

Quantitative response y in R

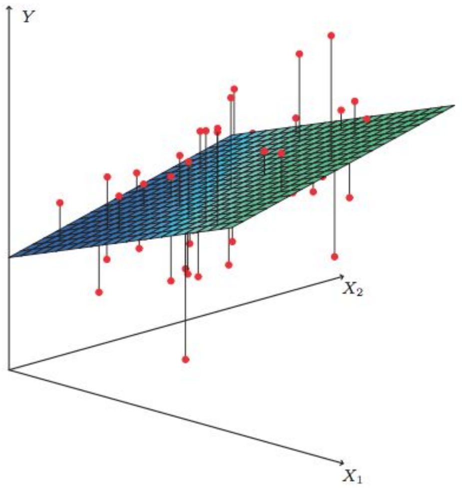


Categorial response y in {0,1}

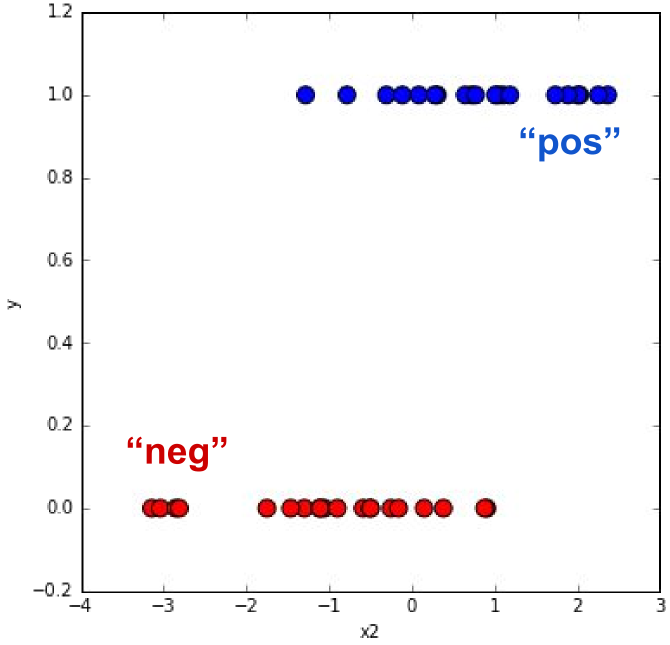


Regression vs Classification

Quantitative response y in R

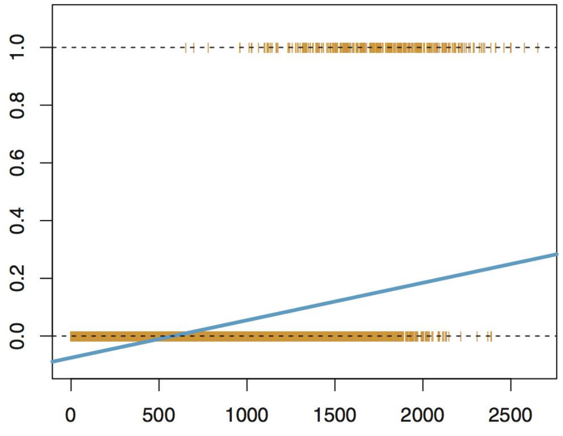


Categorial response y in {0,1}



Trying to apply LinReg to y

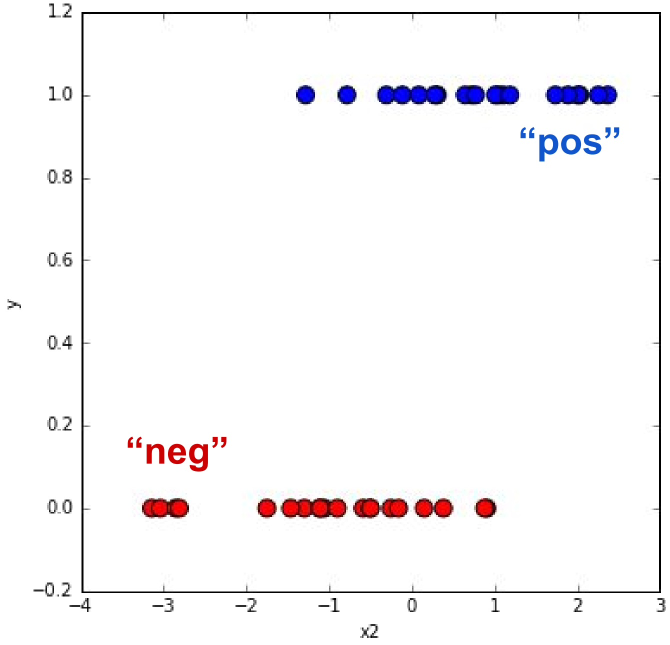
Quantitative response y in R



$$y \approx \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \ldots + \beta_p x_p$$

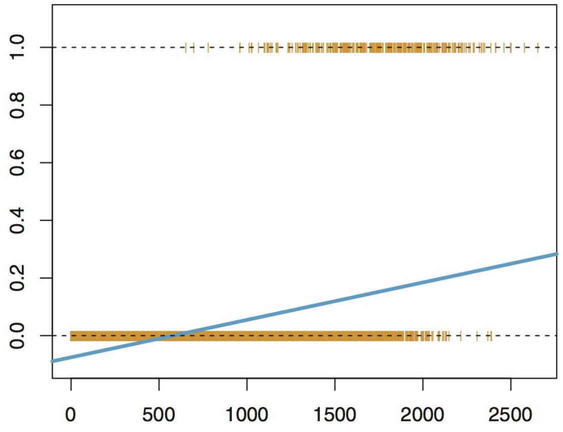
Negative probabilities ?
How to cut-off ?

Categorial response y in {0,1}



LogReg as model of probability

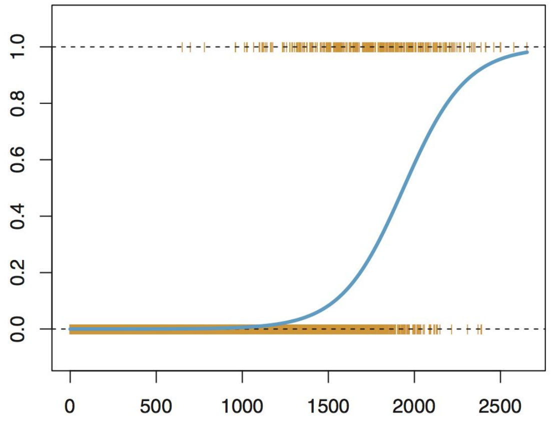
Quantitative response y in R



$$y \approx \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \ldots + \beta_p x_p$$

Negative probabilities ?
How to cut-off ?

Categorial response y in {0,1}



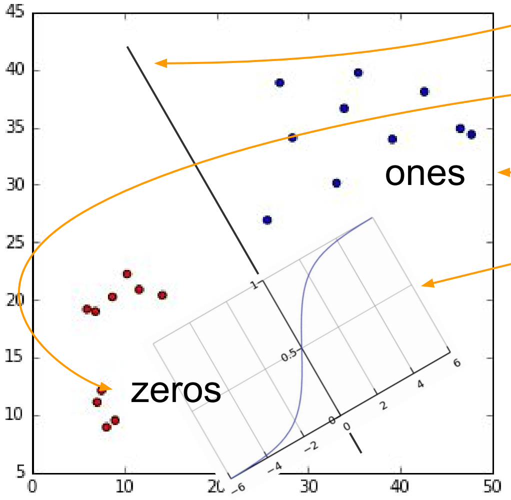
$$p(X) = h(\beta_0 + \beta_1 . x_1 + \cdots + +\beta_p . x_p)$$

Idea : model probability of being positive
as a function of a linear model

LogReg in a nutshell

REALITY

MODEL

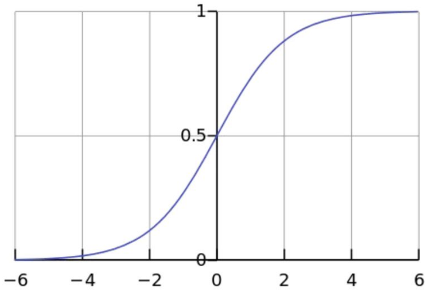


$$p(X) = h(\beta_0 + \beta_1 . x_1 + \cdots + +\beta_p . x_p)$$

It (badly) translates as :
computes the probability of being in one of the two classes
depending on of the side and distance of the plan

$$h : \mathbb{R} \to [0, 1]$$

$$h(t) = \frac{1}{1+e^{-t}}$$



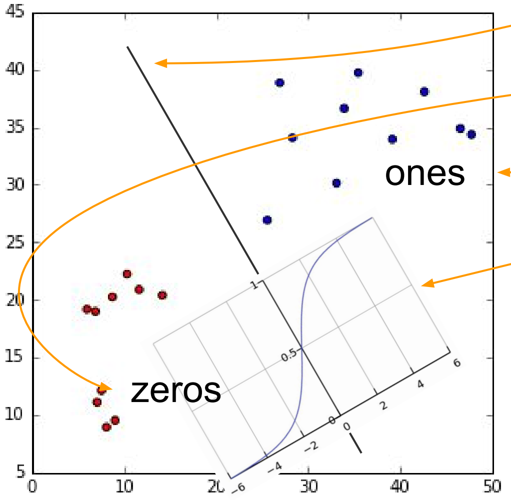
Using LogReg to predict

Let’s suppose we have a LogReg model already...

LogReg in a nutshell

REALITY

MODEL

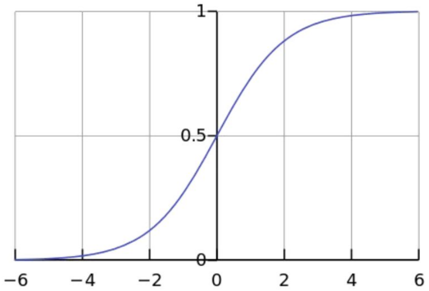


$$p(X) = h(\beta_0 + \beta_1.x_1 + \cdots + +\beta_p.x_p)$$

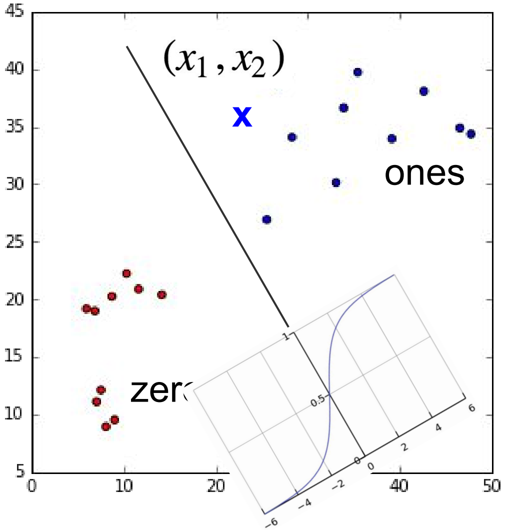
It (badly) translates as :
computes the probability of being in one of the two classes
depending on of the side and distance of the plan

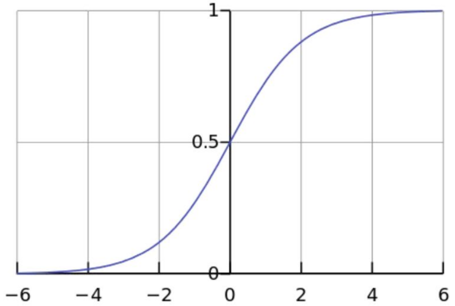
$$h : \mathbb{R} \to [0, 1]$$

$$h(t) = \frac{1}{1+e^{-t}}$$



Propagation of a change in attributes



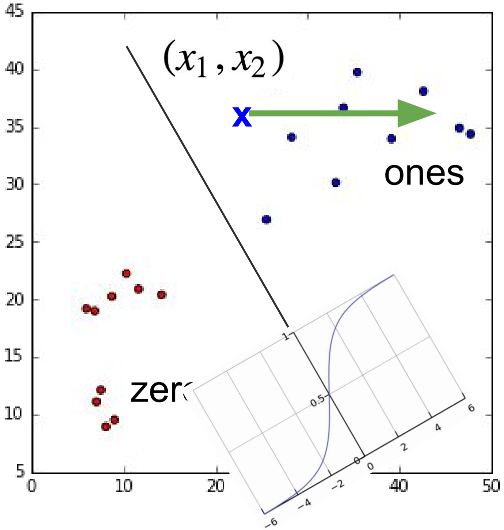


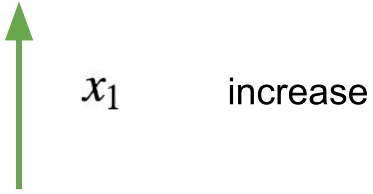
$$h : \mathbb{R} \to [0, 1]$$

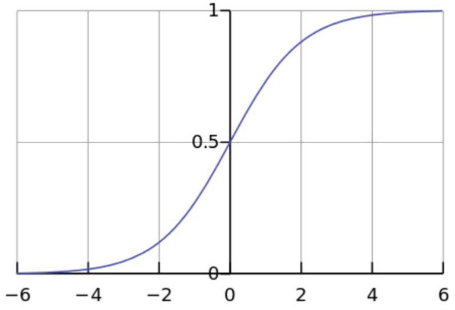
$$h(t) = \frac{1}{1+e^{-t}}$$

$$p(X) = h(\beta_0 + \beta_1 . x_1 + \cdots + +\beta_p . x_p)$$

Propagation of a change in attributes





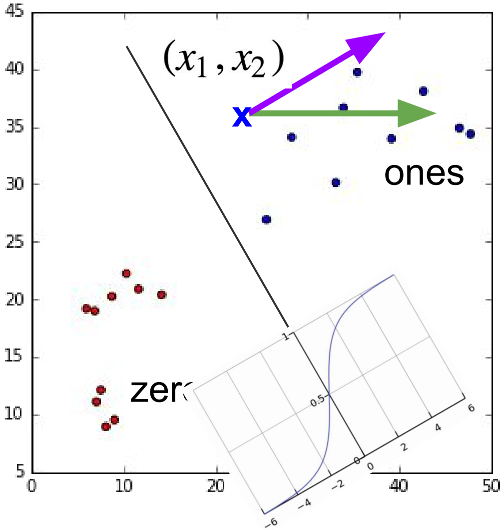


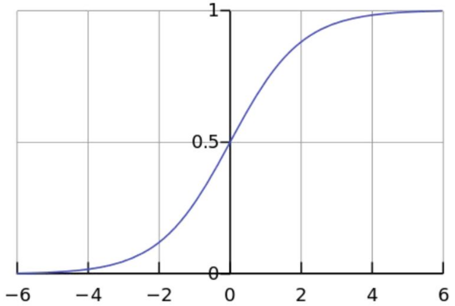
$h : \mathbb{R} \to [0, 1]$

$h(t) = \frac{1}{1+e^{-t}}$

$$p(X) = h(\beta_0 + \beta_1 . x_1 + \cdots + +\beta_p . x_p)$$

Propagation of a change in attributes





x_1 increase

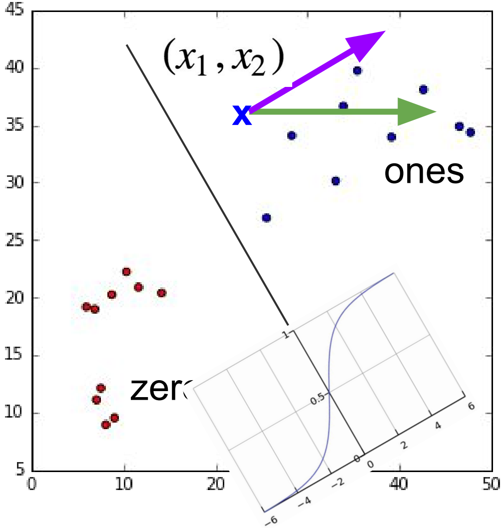
$\beta_1 . x_1$

$h : \mathbb{R} \rightarrow [0, 1]$

$h(t) = \frac{1}{1+e^{-t}}$

$$p(X) = h(\beta_0 + \beta_1 . x_1 + \cdots + +\beta_p . x_p)$$

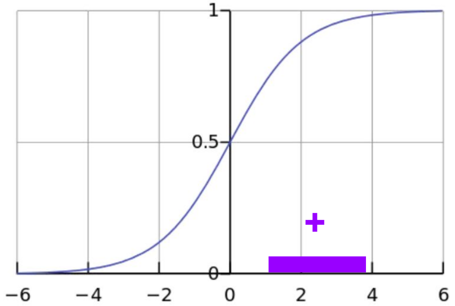
Propagation of a change in attributes



x_1 increase

$\beta_1 . x_1$

+ $(\beta_0 + \beta_1 . x_1 + \cdots + +\beta_p . x_p)$

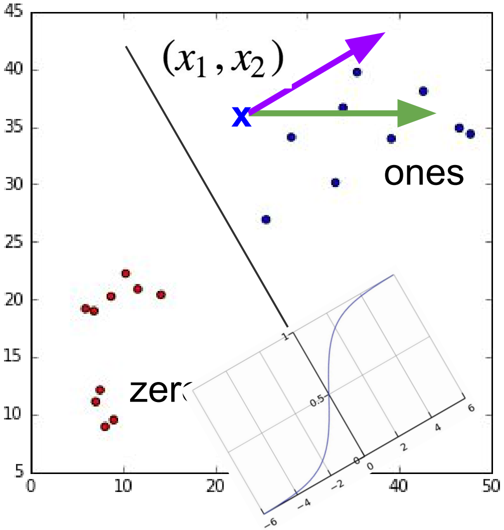


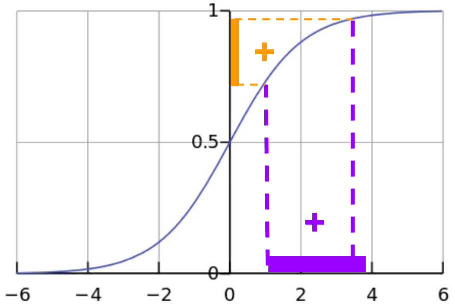
$h : \mathbb{R} \to [0, 1]$

$h(t) = \frac{1}{1+e^{-t}}$

$$p(X) = h(\beta_0 + \beta_1 . x_1 + \cdots + +\beta_p . x_p)$$

Propagation of a change in attributes





$h : \mathbb{R} \to [0, 1]$

$h(t) = \frac{1}{1+e^{-t}}$

x_1 increase

$\beta_1 . x_1$

$+$ $(\beta_0 + \beta_1 . x_1 + \cdots + +\beta_p . x_p)$

$+$ $h(\beta_0 + \beta_1 . x_1 + \cdots + +\beta_p . x_p)$

$$p(X) = h(\beta_0 + \beta_1 . x_1 + \cdots + +\beta_p . x_p)$$

Interpreting coefficients

Making sense of the logistic function

Probs, odds, log-odds, odds-ratio

Probabilities range between 0 and 1.

[examples link]

$$p(x)$$

Suppose that seven out of 10 males are admitted to an engineering school while three of 10 females are admitted.

For males: **p = 7/10 = .7** **1 - p = 1 - .7 = .3**
For females: **p = 3/10 = .3** **1 - p = 1 - .3 = .7**

Odds are defined as the ratio of the probability of success and the probability of failure.

$$\frac{p(X)}{1-p(X)}$$

odds(male) = .7/.3 = 2.33333
odds(female) = .3/.7 = .42857

Log-odds are the log of odds

$$log(\frac{p(X)}{1-p(X)})$$

Odds-ratio is comparing two properties in terms of odds.

$$\frac{odds(A)}{odds(B)}$$

OR = 2.3333/.42857 = 5.44

Thus, for a male, the odds of being admitted are 5.44 times larger than the odds for a female being admitted.

Probs, odds, log-odds, odds-ratio in LogReg

Probabilities range between 0 and 1.

$$p(x) = \frac{e^{\beta^T.x}}{1+e^{\beta^T.x}}$$

[examples link]

Odds are defined as the ratio of the probability of success and the probability of failure.

$$\frac{p(X)}{1-p(X)} = e^{\beta^T.x}$$

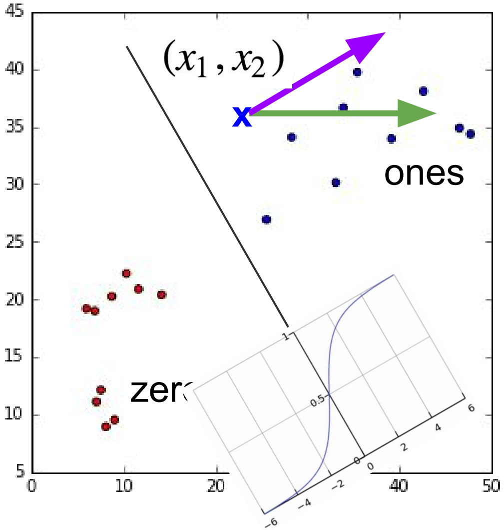
Log-odds are the log of odds

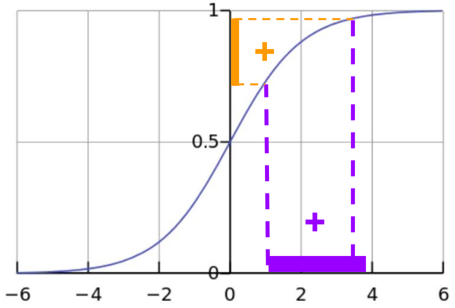
$$log(\frac{p(X)}{1-p(X)}) = \beta^T.x = \beta_0 + \beta_1.x_1 + \ldots + \beta_n.x_n$$

Odds-ratio is comparing two properties in terms of odds.

$$\frac{odds(A)}{odds(B)} \qquad OR = e^{\beta_i}$$

Propagation of a change in attributes





$h : \mathbb{R} \to [0, 1]$

$h(t) = \frac{1}{1+e^{-t}}$

x_1 increase

$\beta_1 . x_1$

+ $(\beta_0 + \beta_1 . x_1 + \cdots + +\beta_p . x_p)$

+ $h(\beta_0 + \beta_1 . x_1 + \cdots + +\beta_p . x_p)$

$$p(X) = h(\beta_0 + \beta_1 . x_1 + \cdots + +\beta_p . x_p)$$

Estimating a LogReg Model

NOW, machine, it's your turn to learn...

Notations

$$X = \begin{bmatrix} 1 & x_{1,1} & \cdots & x_{1,p-1} \\ 1 & x_{2,1} & \cdots & x_{2,p-1} \\ \vdots & \vdots & & \vdots \\ 1 & x_{n,1} & \cdots & x_{n,p-1} \end{bmatrix} \quad \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{bmatrix} = y$$

x_i	y_i
...	1
...	0
...	0
...	1
...	1
...	1
...	0
...	1

	$p(x_i)$
1	0.95
0	0.21
1	0.55
0	0.43
1	0.77
0	0.44
0	0.15
1	0.81

$$\beta = \begin{bmatrix} \beta_0 \\ \beta_1 \\ \beta_2 \\ \vdots \\ \beta_{p-1} \end{bmatrix}$$

$$p(x_i) = \frac{e^{\beta^T x_i}}{1+e^{\beta^T x_i}}$$

Likelihood of the LogReg model

x_i	y_i
...	1
...	0
...	0
...	1
...	1
...	1
...	0
...	1

	$p(x_i)$
1	0.95
0	0.21
1	0.55
0	0.43
1	0.77
0	0.44
0	0.15
1	0.81

Let's suppose we have $\beta = \begin{bmatrix} \beta_0 \\ \beta_1 \\ \beta_2 \\ \vdots \\ \beta_{p-1} \end{bmatrix}$ which gives us $p(x_i) = \dfrac{e^{\beta^T x_i}}{1+e^{\beta^T x_i}}$

Likelihood of the LogReg model

x_i, y_i	
...	1
...	0
...	0
...	1
...	1
...	1
...	0
...	1

$p(x_i)$	
1	0.95
0	0.21
1	0.55
0	0.43
1	0.77
0	0.44
0	0.15
1	0.81

Let's suppose we have $\beta = \begin{bmatrix} \beta_0 \\ \beta_1 \\ \beta_2 \\ \vdots \\ \beta_{p-1} \end{bmatrix}$ which gives us $p(x_i) = \frac{e^{\beta^T x_i}}{1+e^{\beta^T x_i}}$

What is the "likelihood" of our dataset to have be drawn out of that probability ?

Likelihood of the LogReg model

x_i	y_i
...	1
...	0
...	0
...	1
...	1
...	1
...	0
...	1

	$p(x_i)$
1	0.95
0	0.21
1	0.55
0	0.43
1	0.77
0	0.44
0	0.15
1	0.81

Let's suppose we have $\beta = \begin{bmatrix} \beta_0 \\ \beta_1 \\ \beta_2 \\ \vdots \\ \beta_{p-1} \end{bmatrix}$ which gives us $p(x_i) = \dfrac{e^{\beta^T x_i}}{1+e^{\beta^T x_i}}$

What is the "likelihood" of our dataset to have be drawn out of that probability ?

Let's first do that for each observation

$$y_i = 1 \implies p(x_i)$$

$$y_i = 0 \implies 1 - p(x_i)$$

Likelihood of the LogReg model

x_i, y_i	
...	1
...	0
...	0
...	1
...	1
...	1
...	0
...	1

$p(x_i)$	
1	0.95
0	0.21
1	0.55
0	0.43
1	0.77
0	0.44
0	0.15
1	0.81

Let's suppose we have $\beta = \begin{bmatrix} \beta_0 \\ \beta_1 \\ \beta_2 \\ \vdots \\ \beta_{p-1} \end{bmatrix}$ which gives us $p(x_i) = \dfrac{e^{\beta^T x_i}}{1+e^{\beta^T x_i}}$

What is the "likelihood" of our dataset to have be drawn out of that probability ?

Let's first do that for each observation

$$y_i = 1 \implies p(x_i)$$

$$y_i = 0 \implies 1 - p(x_i)$$

Let's do that for the whole dataset $L(\beta) = \prod_{i:y_i=1} p(x_i) * \prod_{i:y_i=0} (1 - p(x_i))$

Optimization of the likelihood

x_i, y_i	
...	1
...	0
...	0
...	1
...	1
...	1
...	0
...	1

$p(x_i)$	
1	0.95
0	0.21
1	0.55
0	0.43
1	0.77
0	0.44
0	0.15
1	0.81

Let's suppose we have $\beta = \begin{bmatrix} \beta_0 \\ \beta_1 \\ \beta_2 \\ \vdots \\ \beta_{p-1} \end{bmatrix}$ which gives us $p(x_i) = \dfrac{e^{\beta^T x_i}}{1+e^{\beta^T x_i}}$

$$L(\beta) = \prod_{i:y_i=1} p(x_i) * \prod_{i:y_i=0} (1 - p(x_i))$$

Can we find the maximum of that likelihood ?

Optimization of the likelihood

x_i	y_i
...	1
...	0
...	0
...	1
...	1
...	1
...	0
...	1

$p(x_i)$	
1	0.95
0	0.21
1	0.55
0	0.43
1	0.77
0	0.44
0	0.15
1	0.81

Let's suppose we have $\beta = \begin{bmatrix} \beta_0 \\ \beta_1 \\ \beta_2 \\ \vdots \\ \beta_{p-1} \end{bmatrix}$ which gives us $p(x_i) = \dfrac{e^{\beta^T x_i}}{1+e^{\beta^T x_i}}$

$$L(\beta) = \prod_{i:y_i=1} p(x_i) * \prod_{i:y_i=0} (1 - p(x_i))$$

Can we find the maximum of that likelihood ?

$$LogL(\beta) = \sum_{i:y_i=1} log(p(x_i)) + \sum_{i:y_i=0} log(1 - p(x_i))$$

$$LogL(\beta) = \sum_i y_i . log(p(x_i)) + (1 - y_i). log(1 - p(x_i))$$

$$LogL(\beta) = \sum_i y_i . \beta^T x_i - log(1 + e^{\beta^T x_i})$$

Optimization of the likelihood

x_i, y_i	
...	1
...	0
...	0
...	1
...	1
...	1
...	0
...	1

$p(x_i)$	
1	0.95
0	0.21
1	0.55
0	0.43
1	0.77
0	0.44
0	0.15
1	0.81

Let's suppose we have $\beta = \begin{bmatrix} \beta_0 \\ \beta_1 \\ \beta_2 \\ \vdots \\ \beta_{p-1} \end{bmatrix}$ which gives us $p(x_i) = \frac{e^{\beta^T x_i}}{1+e^{\beta^T x_i}}$

$$L(\beta) = \prod_{i:y_i=1} p(x_i) * \prod_{i:y_i=0} (1 - p(x_i))$$

Can we find the maximum of that likelihood ?

$$LogL(\beta) = \sum_{i:y_i=1} log(p(x_i)) + \sum_{i:y_i=0} log(1 - p(x_i))$$

$$LogL(\beta) = \sum_i y_i . log(p(x_i)) + (1 - y_i). log(1 - p(x_i))$$

$$LogL(\beta) = \sum_i y_i . \beta^T x_i - log(1 + e^{\beta^T x_i})$$

that we can differentiate...

$$\frac{\partial LL(\beta)}{\partial \beta} = \sum_i x_i . (y_i - p(x_i; \beta))$$

Optimization of the likelihood

x_i	y_i
...	1
...	0
...	0
...	1
...	1
...	1
...	0
...	1

	$p(x_i)$
1	0.95
0	0.21
1	0.55
0	0.43
1	0.77
0	0.44
0	0.15
1	0.81

Let's suppose we have $\beta = \begin{bmatrix} \beta_0 \\ \beta_1 \\ \beta_2 \\ \vdots \\ \beta_{p-1} \end{bmatrix}$ which gives us $p(x_i) = \dfrac{e^{\beta^T x_i}}{1+e^{\beta^T x_i}}$

$$L(\beta) = \prod_{i:y_i=1} p(x_i) * \prod_{i:y_i=0} (1 - p(x_i))$$

Can we find the maximum of that likelihood ?

$$LogL(\beta) = \sum_{i:y_i=1} log(p(x_i)) + \sum_{i:y_i=0} log(1 - p(x_i))$$

$$LogL(\beta) = \sum_i y_i . log(p(x_i)) + (1 - y_i). log(1 - p(x_i))$$

$$LogL(\beta) = \sum_i y_i . \beta^T x_i - log(1 + e^{\beta^T x_i})$$

that ... ntiate...

spoiler alert

$$\frac{\partial LL(\beta)}{\partial \beta} = \sum_i x_i . (y_i - p(x_i; \beta))$$

How to evaluate a classifier ?

Conforming a classifier to the actual response

x_1	x_2	x_3	x_4	y
...	...	...	...	1
...	...	...	...	0
...	...	...	...	0
...	...	...	...	1
...	...	...	...	1
...	...	...	...	1
…	...	...	...	0
...	...	...	...	1

$\hat{y} = \hat{f}(x)$	
1	0.95
0	0.21
1	0.55
0	0.43
1	0.77
0	0.44
0	0.15
1	0.81

$$\hat{y} = \hat{f}(x)$$

y		Pred P	Pred N	
	Actual P	3	2	P = 5
	Actual N	1	2	N = 3
		P*	N*	

Confusion Matrix

$$\hat{y} = \hat{f}(x)$$

y

	Pred P	Pred N	
Actual P	True Positive	False Negative	P = 5
Actual N	False Positive	True Negative	N = 3
	P*	N*	

Confusion Matrix - Metrics

The proportion of <u>observations</u> that are <u>correctly</u> classified ?

Accuracy :

The proportion of <u>positives</u> that are <u>correctly</u> identified as such ?

True Pos Rate :

(aka recall, sensitivity)

The proportion of <u>negatives</u> that are <u>correctly</u> identified as such

True Neg Rate :

(aka specificity)

$$\hat{y} = \hat{f}(x)$$

y

	Pred P	Pred N	
Actual P	True Positive	False Negative	P = 5
Actual N	False Positive	True Negative	N = 3
	P*	N*	

Confusion Matrix - Metrics

The proportion of <u>observations</u> that are <u>correctly</u> classified ?

Accuracy : (TN + TP) / (N + P)

The proportion of <u>positives</u> that are <u>correctly</u> identified as such ?

True Pos Rate : TP / P

(aka recall, sensitivity)

The proportion of <u>negatives</u> that are <u>correctly</u> identified as such

True Neg Rate : TN / N

(aka specificity)

$$\hat{y} = \hat{f}(x)$$

y		Pred P	Pred N	
	Actual P	True Positive	False Negative	P = 5
	Actual N	False Positive	True Negative	N = 3
		P*	N*	

Confusion Matrix - Metrics

The proportion of observations that are NOT correctly classified ?

Error rate :

The proportion of positives that are NOT correctly identified as such ?

False Neg Rate :

(aka fall-out)

The proportion of negatives that are NOT correctly identified as such

False Pos Rate :

(aka 1-specificity)

$$\hat{y} = \hat{f}(x)$$

y

	Pred P	Pred N	
Actual P	True Positive	False Negative	P = 5
Actual N	False Positive	True Negative	N = 3
	P*	N*	

Confusion Matrix - Metrics

The proportion of <u>observations</u> that are <u>NOT correctly</u> classified ?

Error rate : (FN + FP) / (N + P)

The proportion of <u>positives</u> that are <u>NOT correctly</u> identified as such ?

False Neg Rate : FN / P

(aka fall-out)

The proportion of <u>negatives</u> that are <u>NOT correctly</u> identified as such

False Pos Rate : FP / N

(aka 1-specificity)

$$\hat{y} = \hat{f}(x)$$

y	Pred P	Pred N	
Actual P	True Positive	False Negative	P = 5
Actual N	False Positive	True Negative	N = 3
	P*	N*	

Confusion Matrix - Metrics

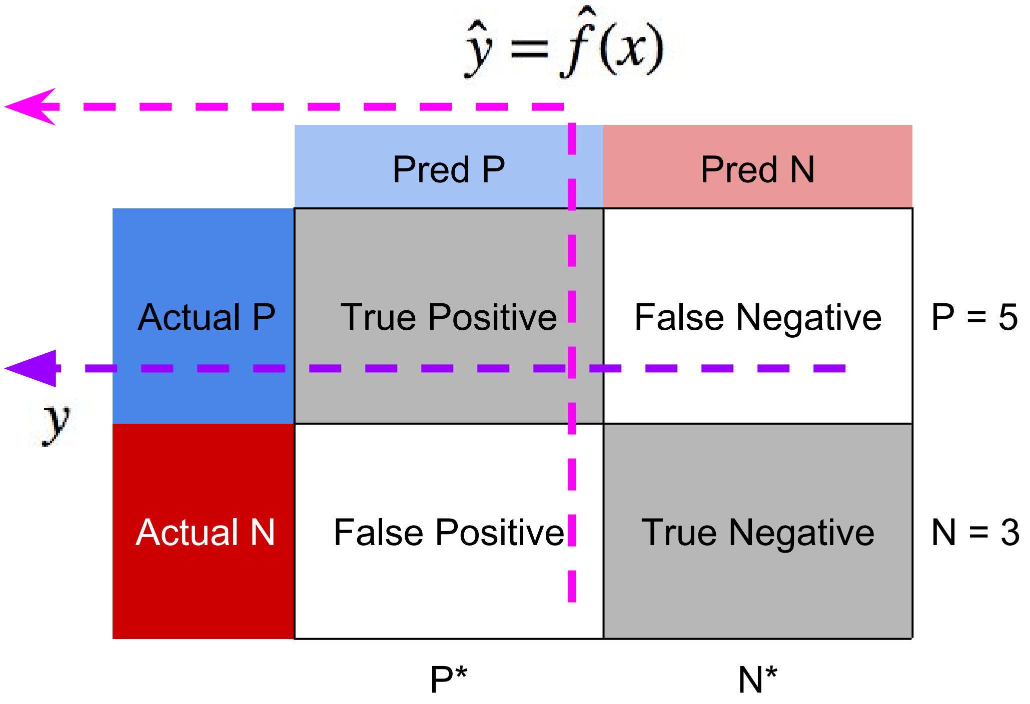
The proportion of <u>actual positives</u>
<u>in those identified</u> as such ?

Precision : TP / (FP + TP)

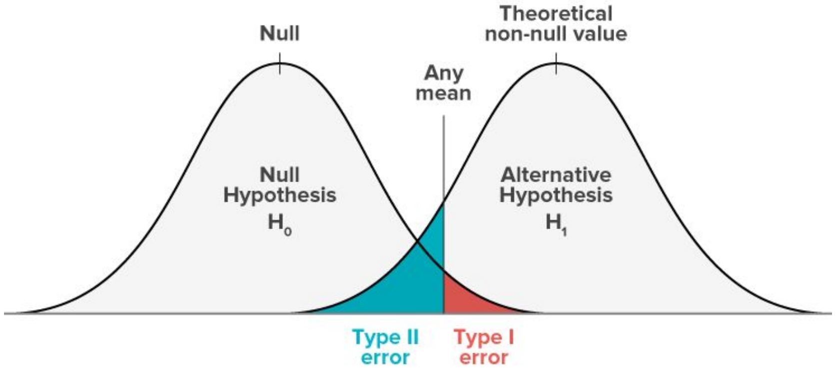
The proportion of <u>positives</u> that are
<u>correctly</u> identified as such ?

Recall : TP / P

(aka TPR, sensitivity)



Confusion Matrix - type I and type II error



$$\hat{y} = \hat{f}(x)$$

y

	Pred P	Pred N	
Actual P	good	Type II error	P = 5
Actual N	Type I error	good	N = 3
	P*	N*	

Using response probabilities

x_1	x_2	x_3	x_4	y
...	...	...	...	1
...	...	...	...	0
...	...	...	...	0
...	...	...	...	1
...	...	...	...	1
...	...	...	...	1
…	...	...	...	0
...	...	...	...	1

P > 0.5

1	0.95
0	0.21
1	0.55
0	0.43
1	0.77
0	0.44
0	0.15
1	0.81

$$\hat{y} = \hat{f}(x)$$

y		Pred P	Pred N	
	Actual P	True Positive	False Negative	P = 5
	Actual N	False Positive	True Negative	N = 3
		P*	N*	

Cut-offs on probabilities

x_1	x_2	x_3	x_4	y	P > 0.5		P > 0.6		P > 0.7		P > 0.8		P > 0.9	
...	...	...	...	**1**	**1**	0.95	**1**	0.95	**1**	0.95	**1**	0.95	**1**	0.95
...	...	...	...	**0**	**0**	0.21	**0**	0.21	**0**	0.21	**0**	0.21	**0**	0.21
...	...	...	...	**0**	**1**	0.55	**0**	**0.55**	**0**	0.55	**0**	0.55	**0**	0.55
...	...	...	...	**1**	**0**	0.43	**0**	0.43	**0**	0.43	**0**	0.43	**0**	0.43
...	...	...	...	**1**	**1**	0.77	**1**	0.77	**1**	0.77	**0**	**0.77**	**0**	0.77
...	...	...	...	**1**	**0**	0.44	**0**	0.44	**0**	0.44	**0**	0.44	**0**	0.44
…	...	...	...	**0**	**0**	0.15	**0**	0.15	**0**	0.15	**0**	0.15	**0**	0.15
...	...	...	...	**1**	**1**	0.81	**1**	0.81	**1**	0.81	**1**	0.81	**0**	**0.81**

Those are sure ones ! => low FPR ! (high precision)
But we miss so many ones ! => low TPR ! (low recall)

Cut-offs on probabilities

x_1	x_2	x_3	x_4	y
...	...	...	...	1
...	...	...	...	0
...	...	...	...	0
...	...	...	...	1
...	...	...	...	1
...	...	...	...	1
...	...	...	...	0
...	...	...	...	1

P > 0.5

1	0.95
0	0.21
1	0.55
0	0.43
1	0.77
0	0.44
0	0.15
1	0.81

P > 0.4

1	0.95
0	0.21
1	0.55
1	**0.43**
1	0.77
1	**0.44**
0	0.15
1	0.81

P > 0.3

1	0.95
0	0.21
1	0.55
1	0.43
1	0.77
1	0.44
0	0.15
1	0.81

P > 0.2

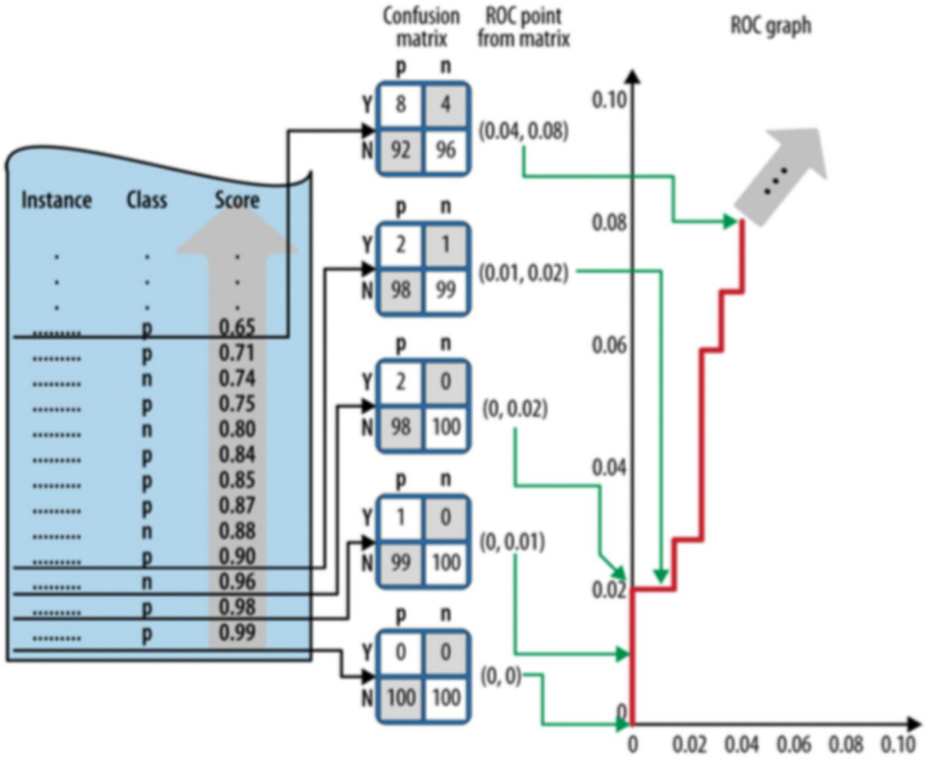
1	0.95
1	**0.21**
1	0.55
1	0.43
1	0.77
1	0.44
0	0.15
1	0.81

P > 0.1

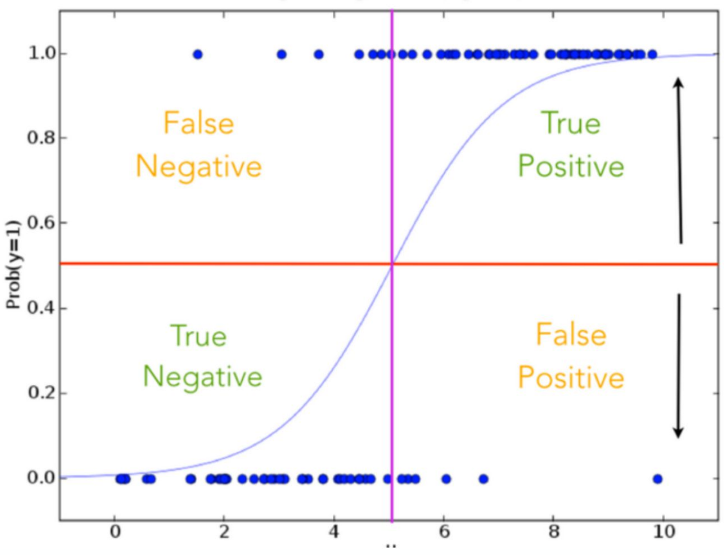
1	0.95
1	0.21
1	0.55
1	0.43
1	0.77
1	0.44
1	0.15
1	0.81

We have so many FP ! => high FPR ! (low precision)
But we capture all the ones ! => high TPR ! (high recall)

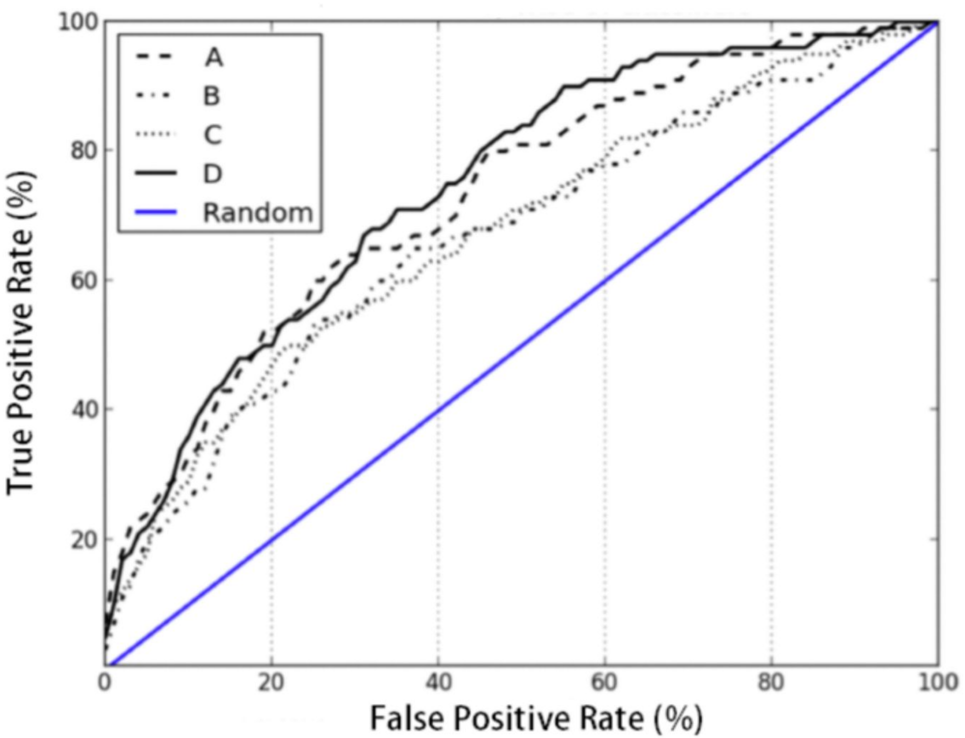
ROC curve (receiver operating characteristic)



For LogReg, think of it as sliding
the purple/red line along the sigmoid function

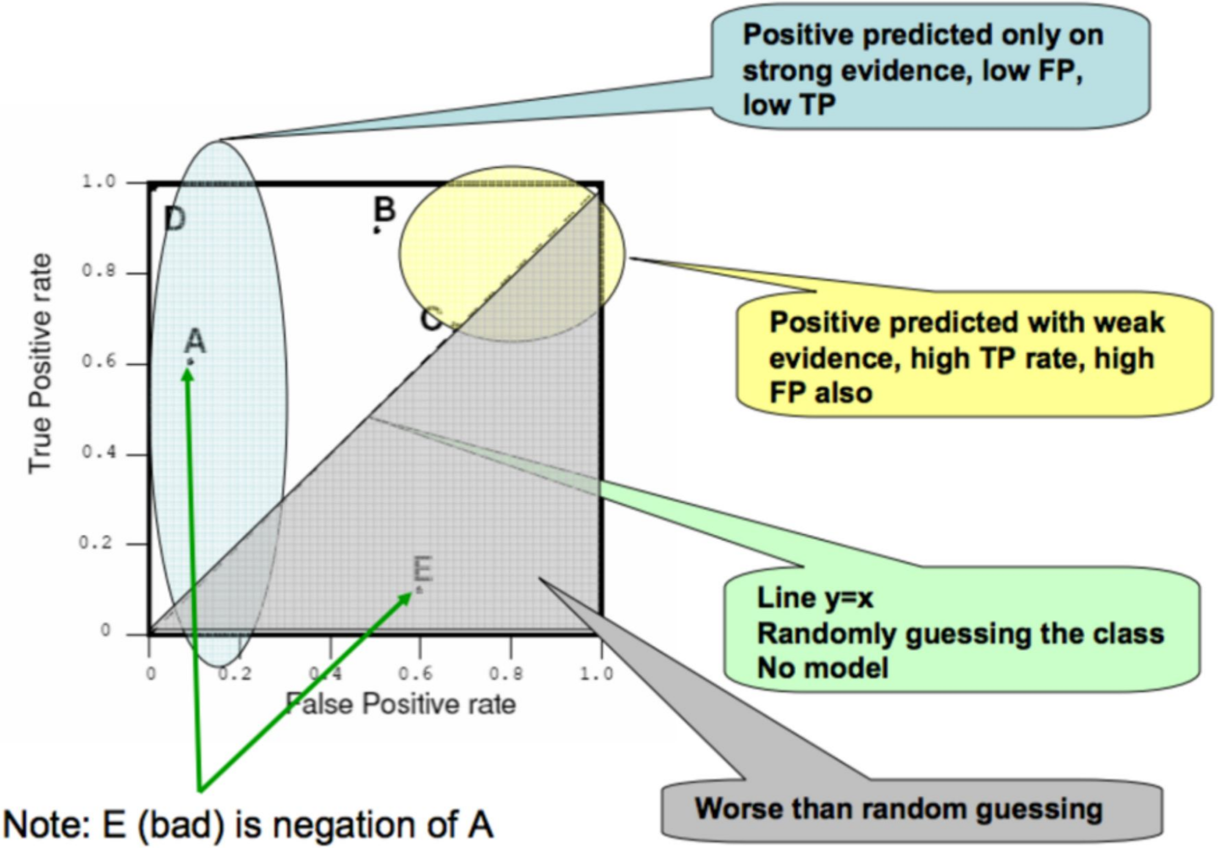


Comparing classifiers based on their ROC curve



Possible metric : AUC
Area-under-curve

What is the “ideal” / “worst” classifier ?



Note: E (bad) is negation of A

Logistic Regression

Classification, metrics and ROC curves

DSI, jf.omhover

OBJECTIVES (morning)

- **Relate** Regression to Classification in the context of supervised learning
- **Compare** Logistic Regression to Linear Regression
- **Define** and **compute** metrics for evaluating classifiers
- **Describe** the process for computing parameter values in LogReg
- **Use** the parameters of a LogReg model to **compute** the class of an obverstion