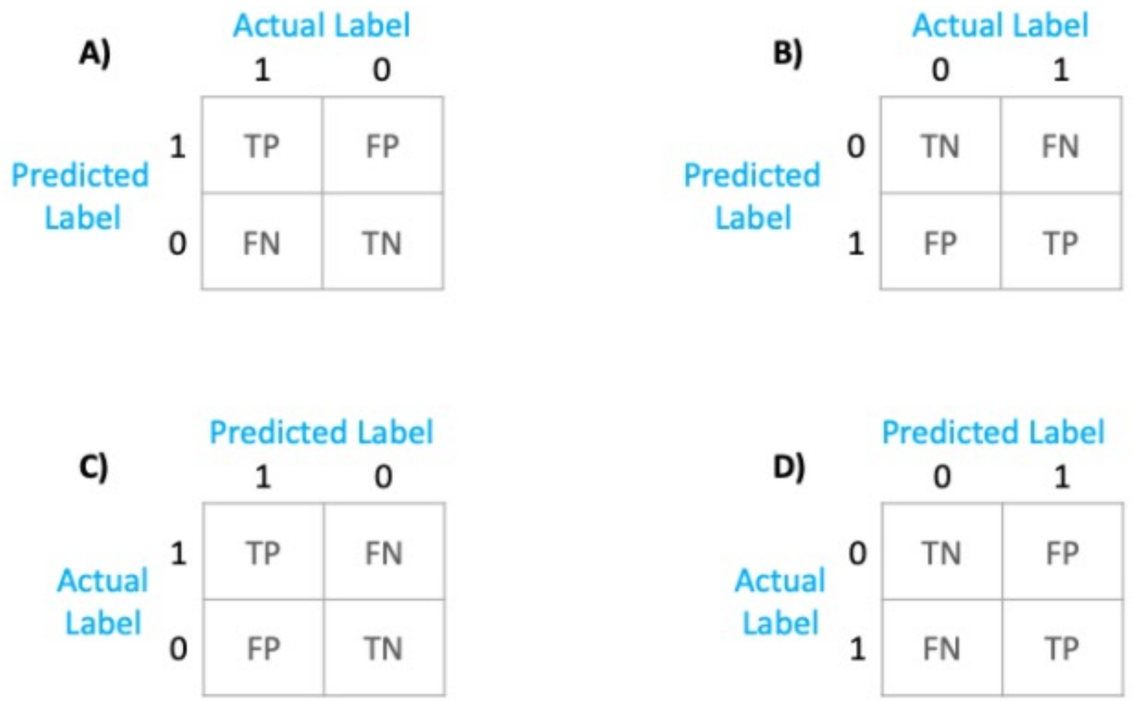
Confusion Matrix

Sklearn Representation

Scikit learn documentation says — Wikipedia and other references may use a different convention for axes.



https://towardsdatascience.com/understanding-the-confusion-matrix-from-scikit-learn-c51d88929c79

F_β-Score

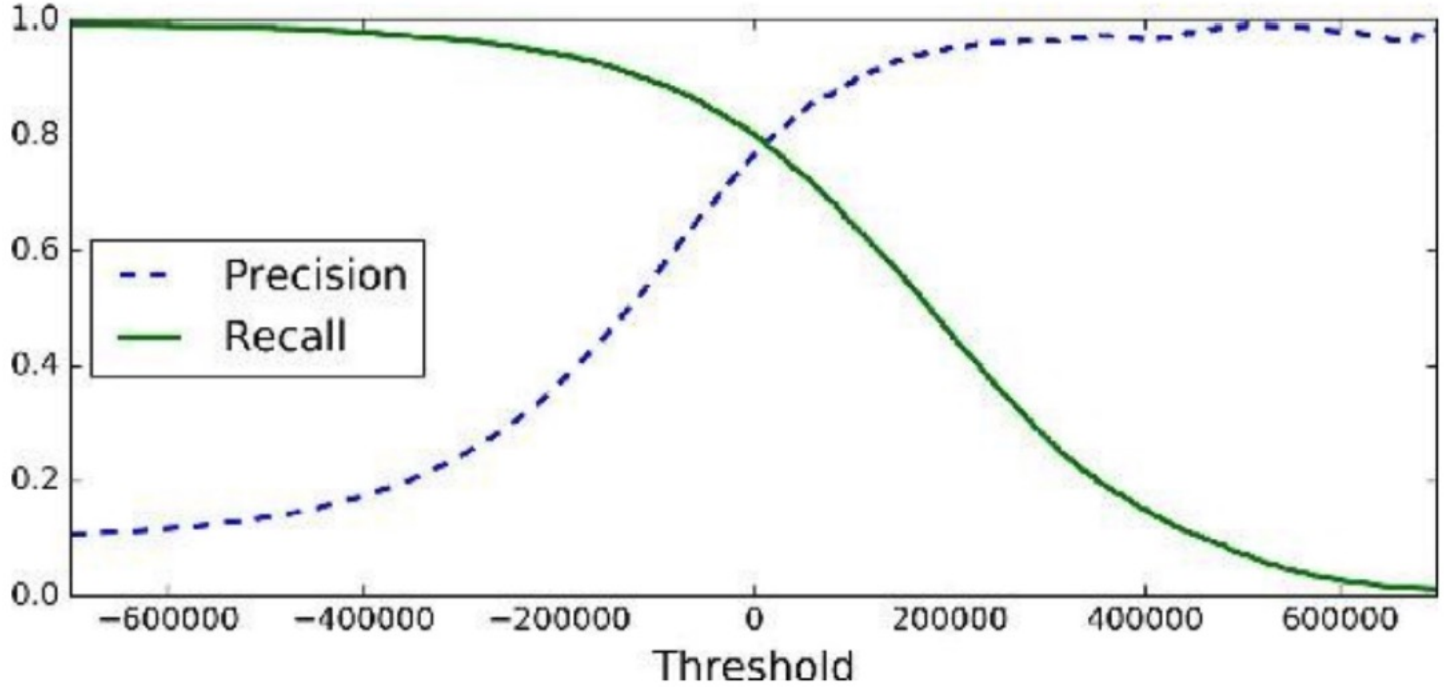
$$F_\beta = (1 + \beta^2) \cdot \frac{\mathrm{precision} \cdot \mathrm{recall}}{(\beta^2 \cdot \mathrm{precision}) + \mathrm{recall}}$$

$\beta = 1$

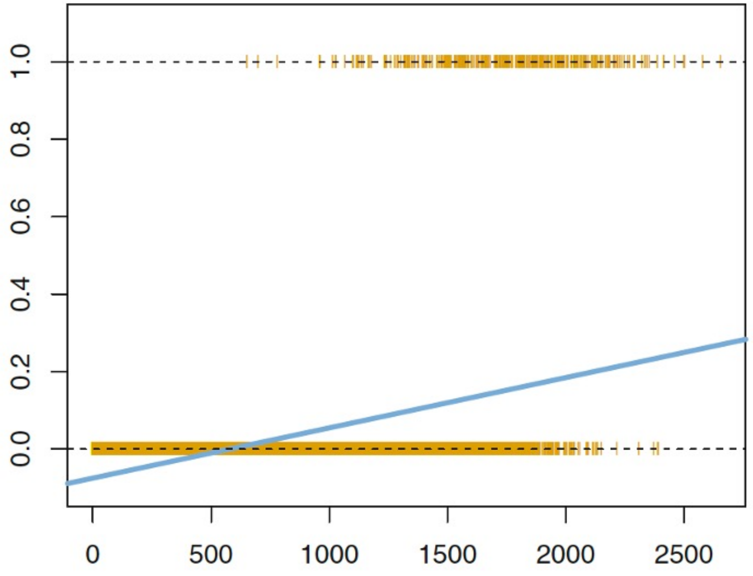
$$F_1 = \frac{2}{\mathrm{recall}^{-1} + \mathrm{precision}^{-1}} = 2\frac{\mathrm{precision} \cdot \mathrm{recall}}{\mathrm{precision} + \mathrm{recall}} = \frac{\mathrm{tp}}{\mathrm{tp} + \frac{1}{2}(\mathrm{fp} + \mathrm{fn})}$$

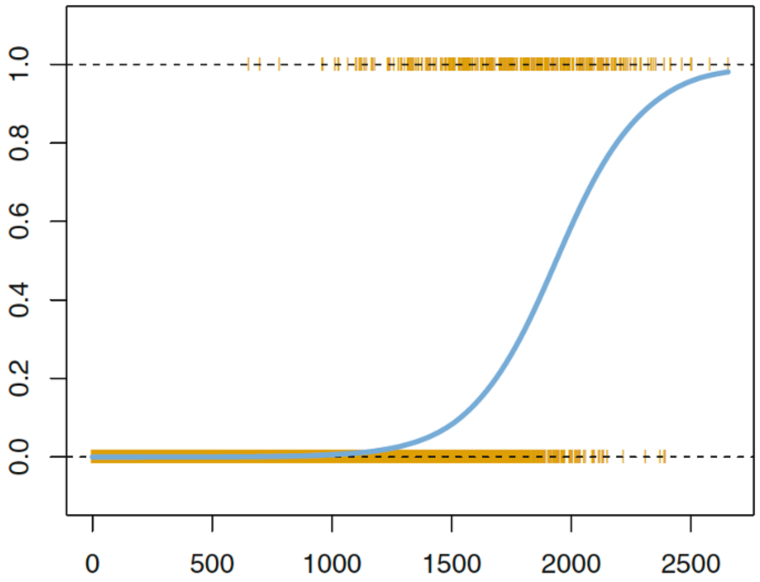
harmonic mean

Precission – Racall Trade off



Logistic Regression





$$p(X) = \beta_0 + \beta_1 X$$

$$p(X) = \frac{e^{\beta_0+\beta_1 X}}{1+e^{\beta_0+\beta_1 X}}$$

Logistic Regression

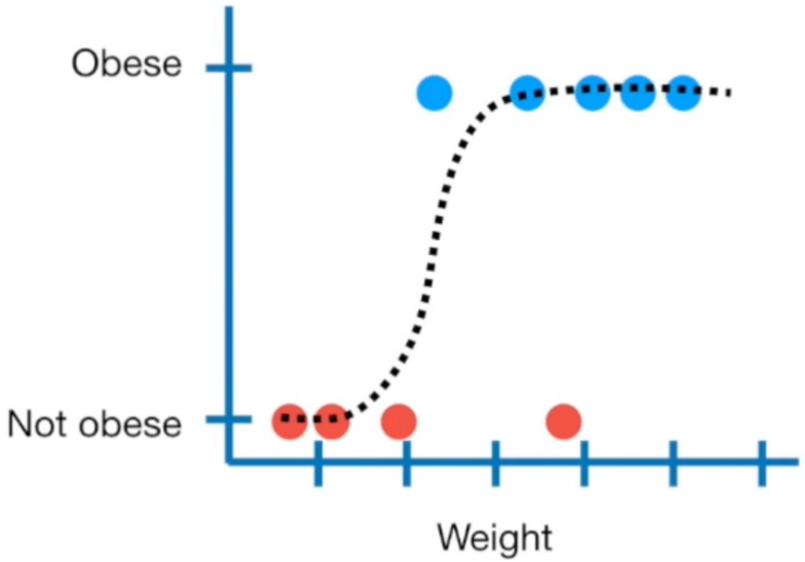
$$p(X) = \frac{e^{\beta_0+\beta_1 X}}{1 + e^{\beta_0+\beta_1 X}}$$

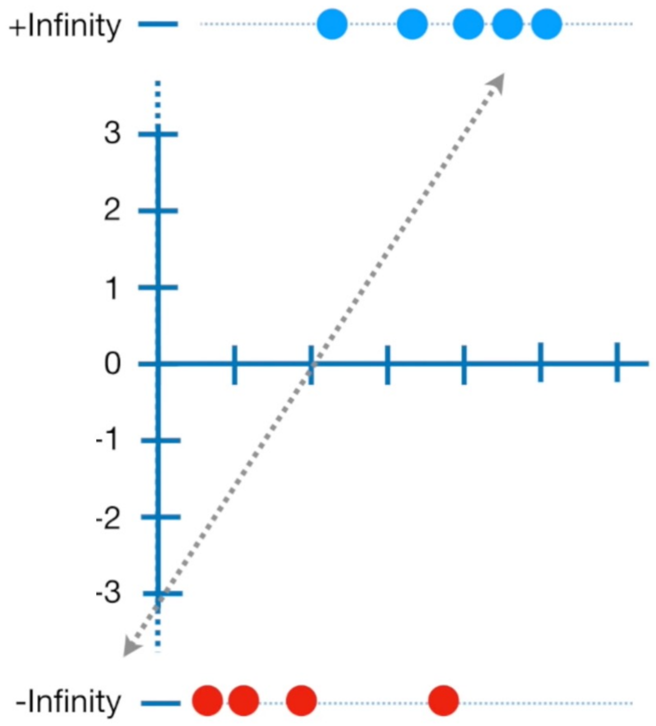
$$\frac{p(X)}{1 - p(X)} = e^{\beta_0+\beta_1 X}$$

$$\log\left(\frac{p(X)}{1 - p(X)}\right) = \beta_0 + \beta_1 X$$

(log odds)

Logistic Regression

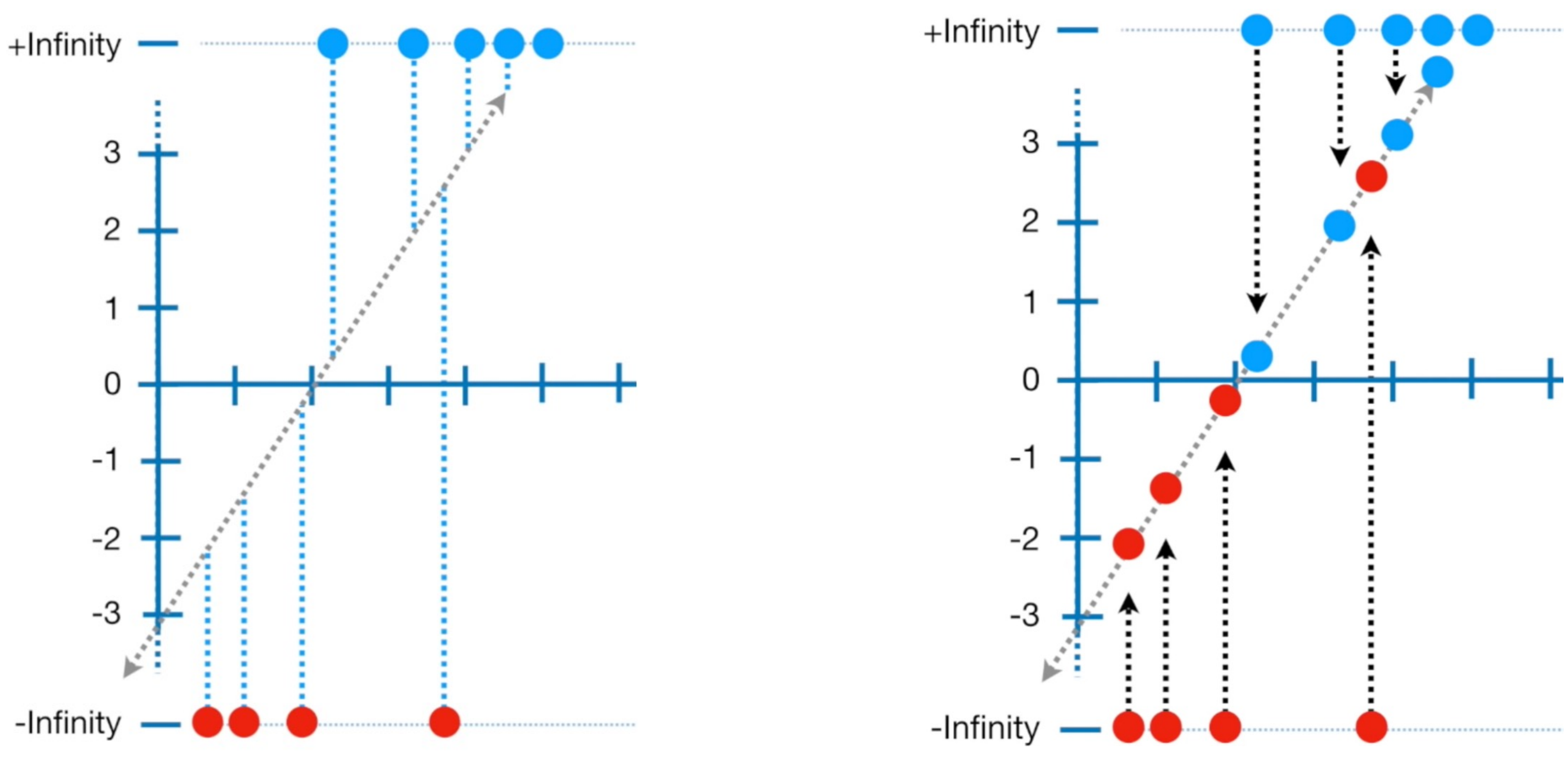




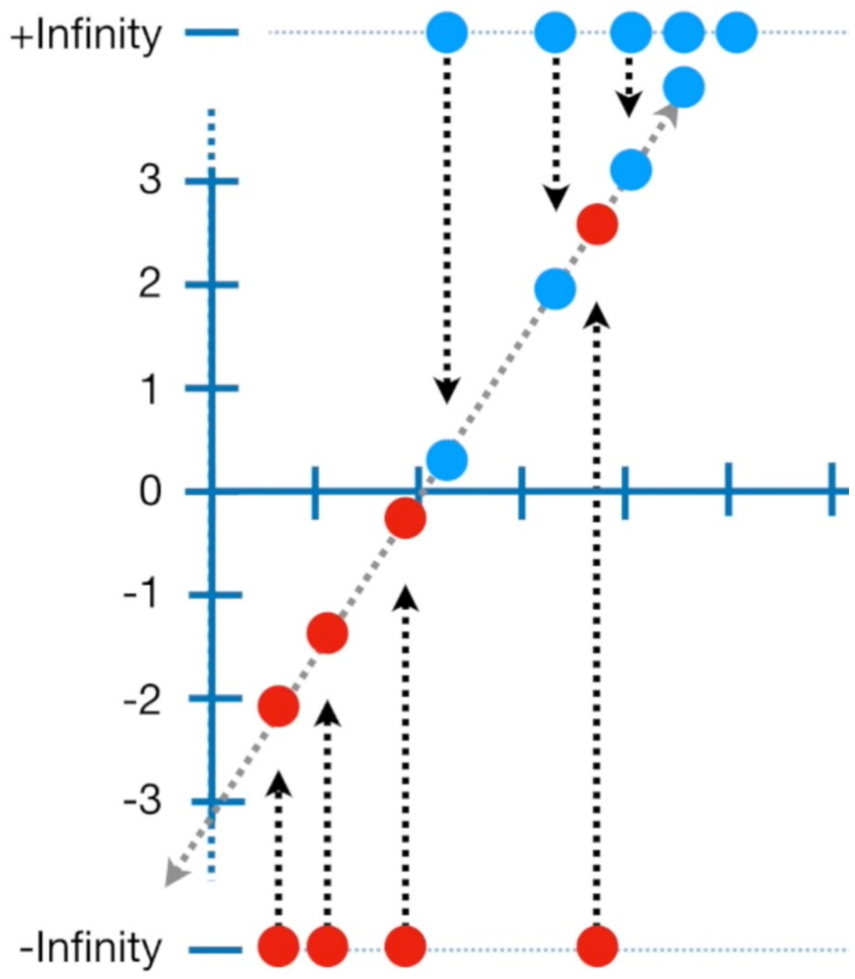
$$p(X) = \frac{e^{\beta_0+\beta_1 X}}{1+e^{\beta_0+\beta_1 X}}$$

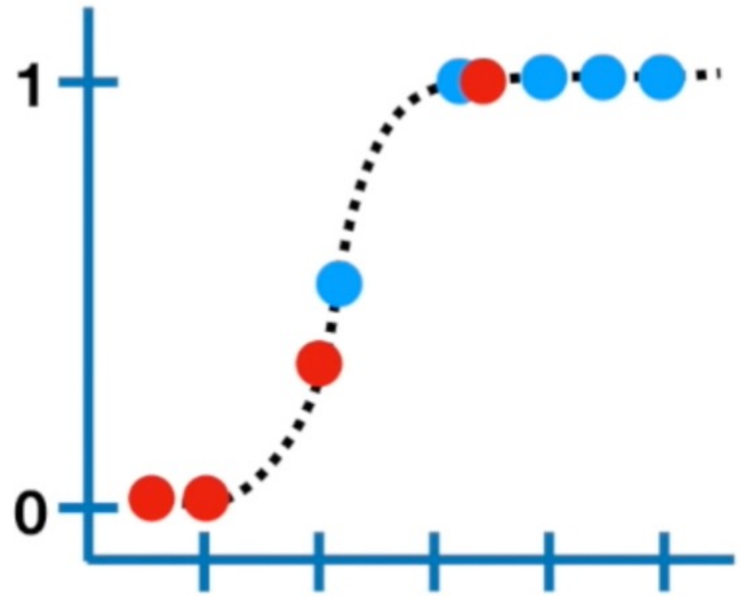
$$\log\left(\frac{p(X)}{1-p(X)}\right) = \beta_0 + \beta_1 X$$

Logistic Regression

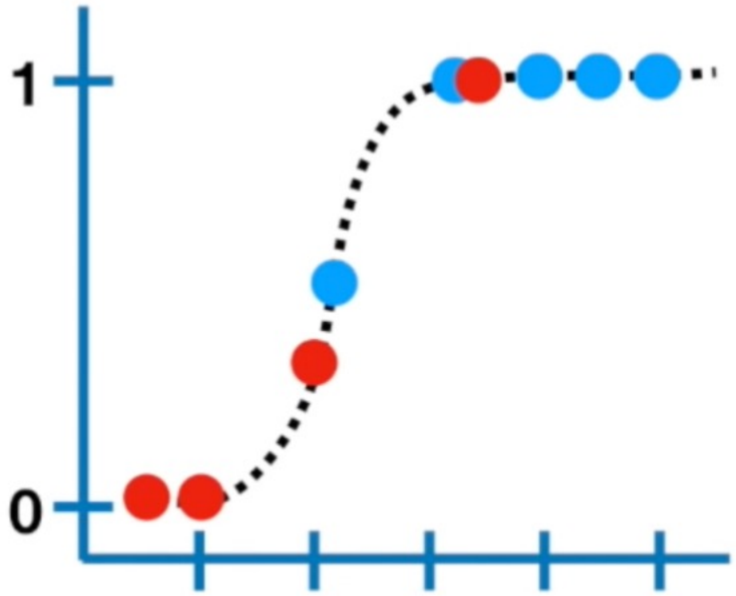


Logistic Regression





Logistic Regression



Likelihood:

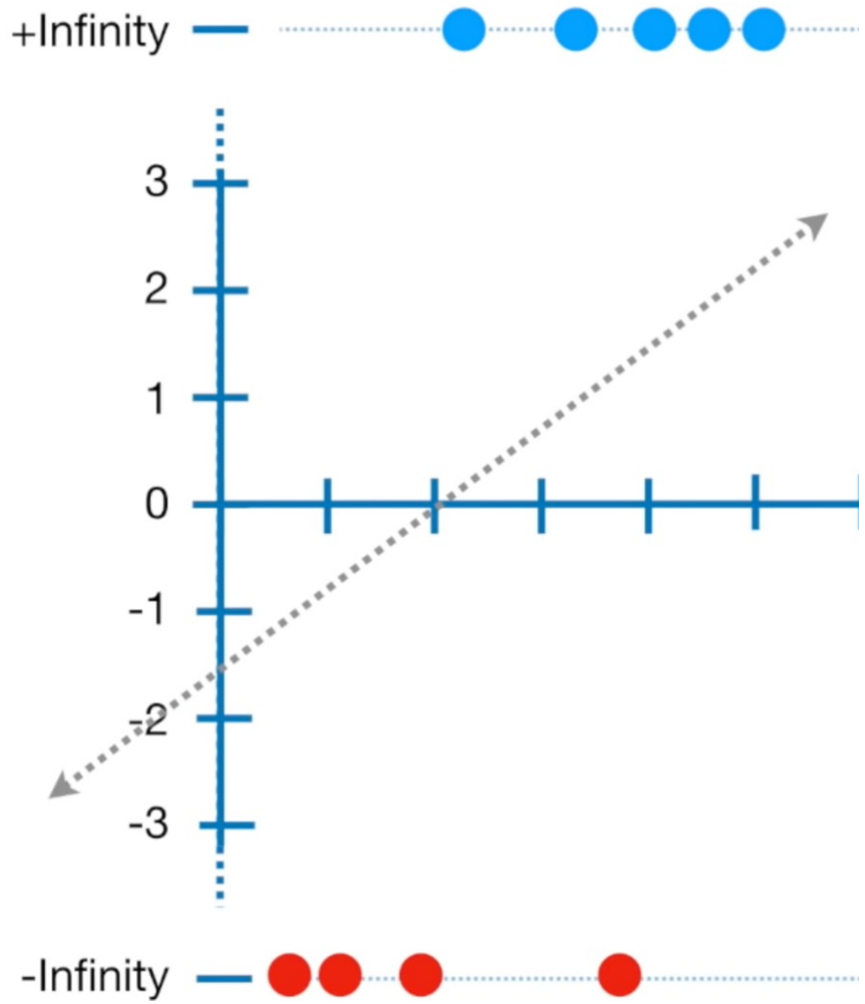
$$\ell(\beta_0, \beta_1) = \prod_{i:y_i=1} p(x_i) \prod_{i^{i}:y_{i'}=0} (1 - p(x_{i'}))$$

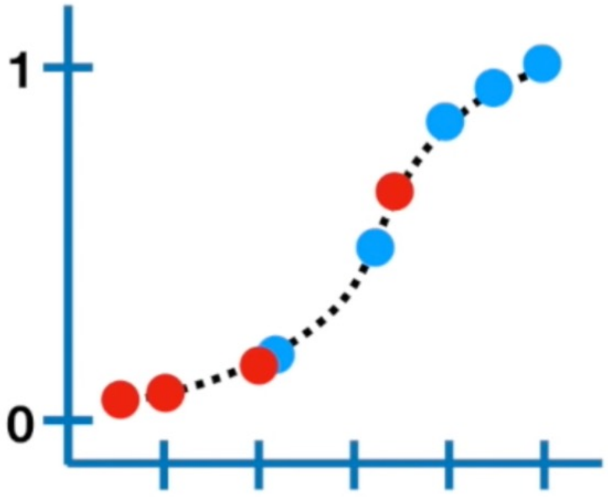
Log-likelihood:

$$\log \ell(\beta_0, \beta_1) = \sum_{i:y_i=1} \log p(x_i) + \sum_{i^{i}:y_{i'}=0} \log(1 - p(x_{i'}))$$

= log(0.49) + log(0.9) + log(0.91) + log(0.91) +
log(0.92) + log(1 - 0.9) + log(1 - 0.3) +
log(1 - 0.01) + log(1 - 0.01)

Logistic Regression





$$= \log(0.22) + \log(0.4) + \log(0.8) + \log(0.89) + \log(0.92) + \log(1 - 0.6) + \log(1 - 0.2) + \log(1 - 0.1) + \log(1 - 0.05)$$

Logistic Regression – Loss function

Linear Regression:

$$J(\theta) = \frac{1}{m} \sum_{i=1}^{m} \frac{1}{2} \left(h_\theta(x^{(i)}) - y^{(i)} \right)^2$$

$$\mathrm{Cost}(h_\theta(x^{(i)}), y^{(i)}) = \frac{1}{2} \left(h_\theta(x^{(i)}) - y^{(i)} \right)^2$$

non – convex

Logistic Regression:

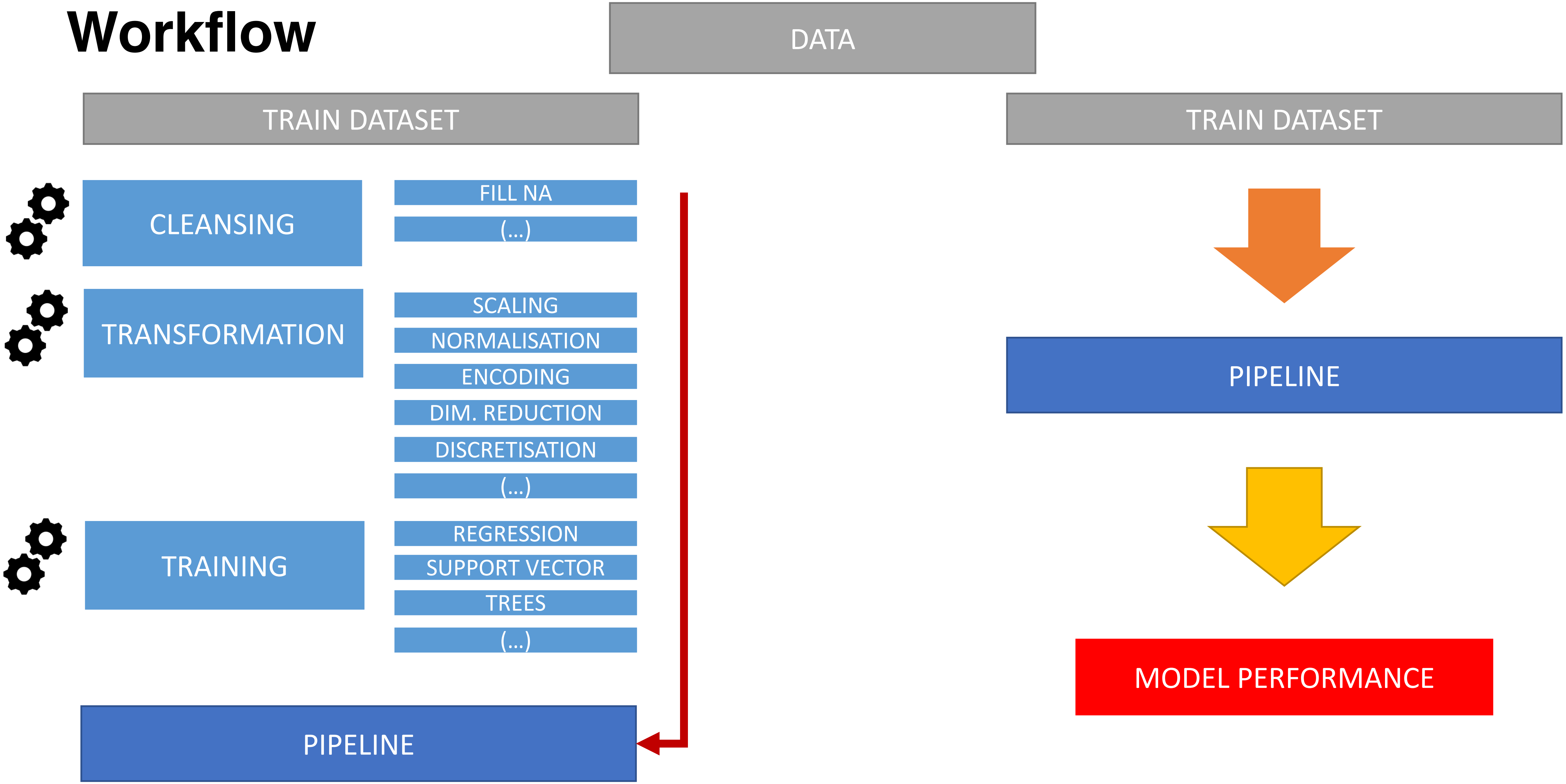
$$\mathrm{Cost}(h_\theta(x), y) = \begin{cases} -\log(h_\theta(x)) & \text{if } y = 1 \\ -\log(1 - h_\theta(x)) & \text{if } y = 0 \end{cases}$$

$$J(\theta) = \frac{1}{m} \sum_{i=1}^{m} \mathrm{Cost}(h_\theta(x^{(i)}), y^{(i)})$$

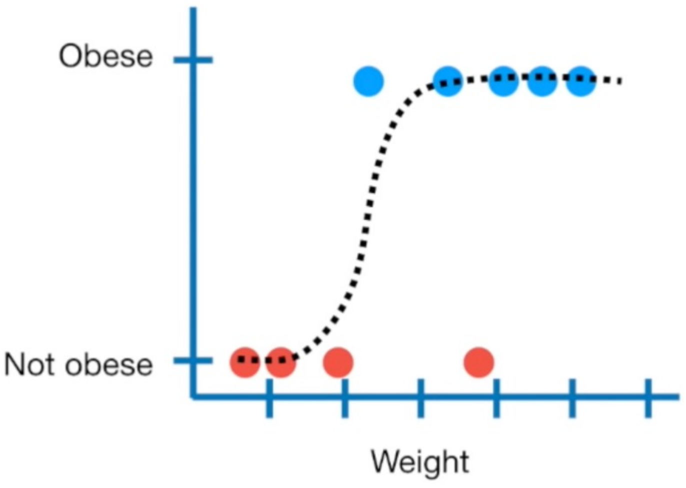
$$= -\frac{1}{m} [\sum_{i=1}^{m} y^{(i)} \log h_\theta(x^{(i)}) + (1 - y^{(i)}) \log (1 - h_\theta(x^{(i)}))]$$

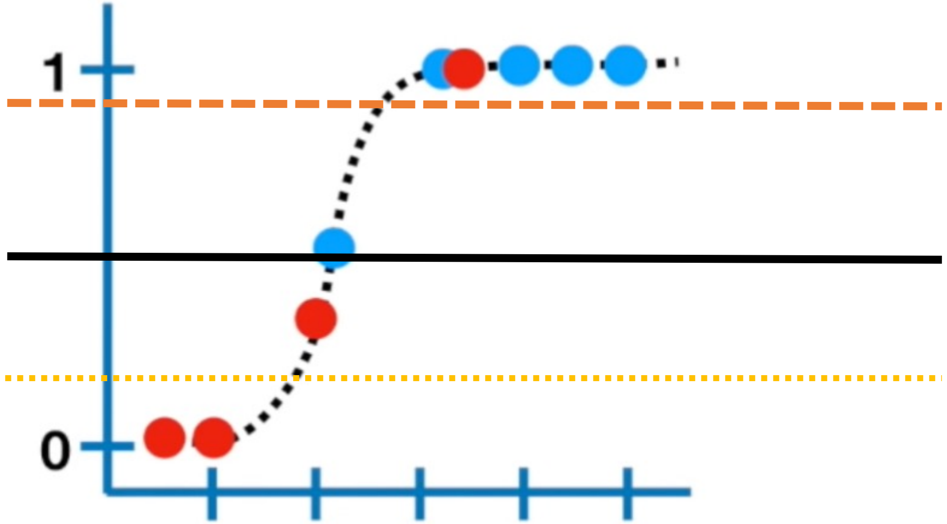
cross-entropy

Workflow


DATA
TRAIN DATASET
CLEANSING
FILL NA
(…)
TRANSFORMATION
SCALING
NORMALISATION
ENCODING
DIM. REDUCTION
DISCRETISATION
(…)
TRAINING
REGRESSION
SUPPORT VECTOR
TREES
(…)
PIPELINE
TRAIN DATASET
PIPELINE
MODEL PERFORMANCE

ROC-AUC





ACTUALS

PREDICTIONS	1	0
1		
0		

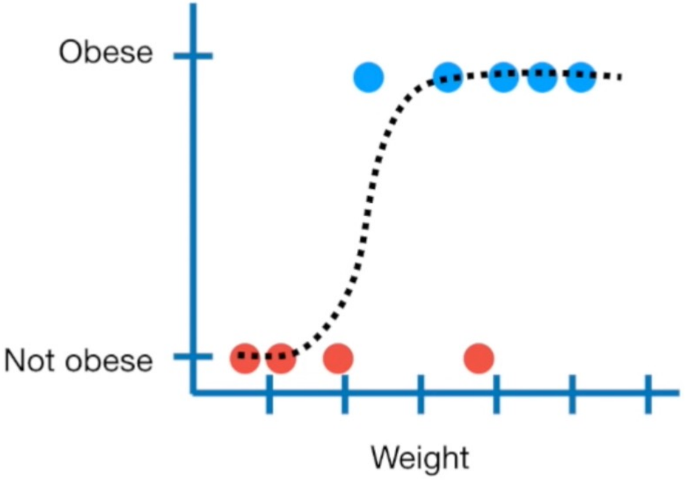
ACTUALS

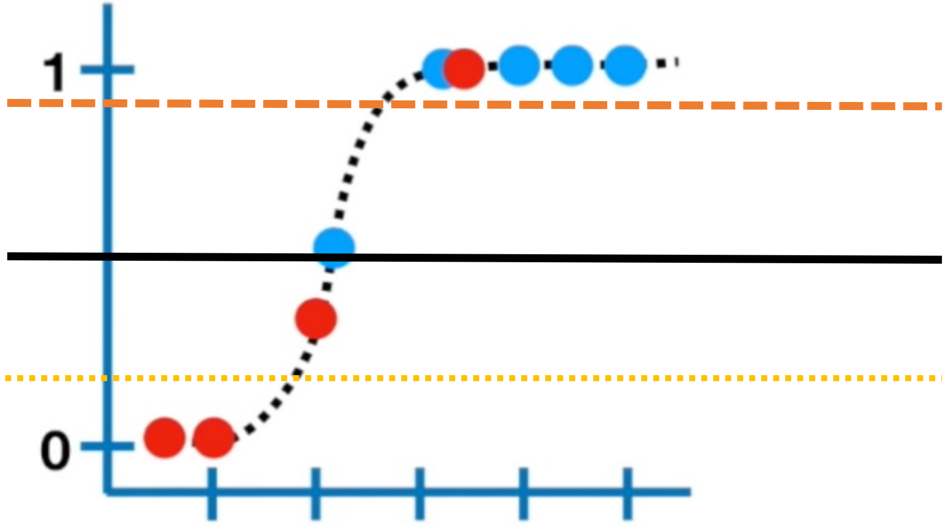
PREDICTIONS	1	0
1		
0		

ACTUALS

PREDICTIONS	1	0
1		
0		

ROC-AUC



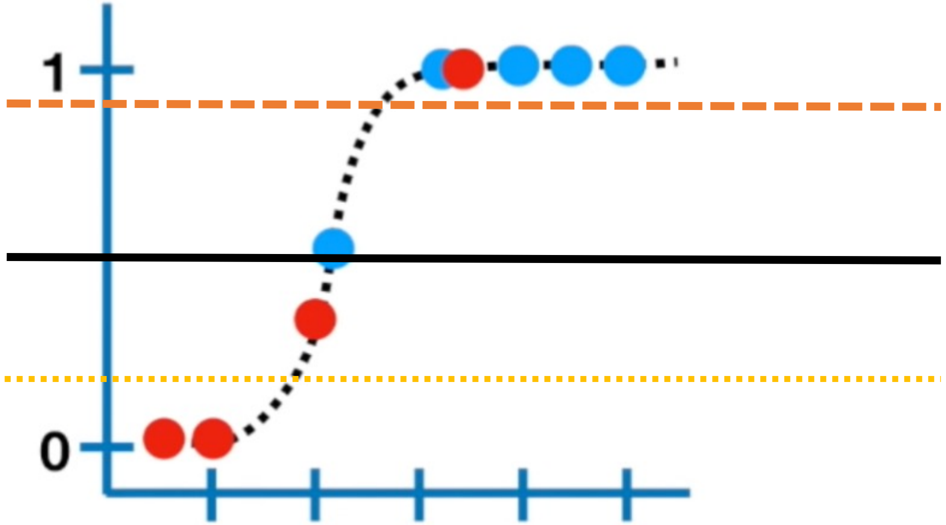


		ACTUALS	
		1	0
PREDICTIONS	1	4	1
	0	1	3

		ACTUALS	
		1	0
PREDICTIONS	1	5	0
	0	1	3

		ACTUALS	
		1	0
PREDICTIONS	1	5	0
	0	2	2

ROC-AUC



		ACTUALS 1	ACTUALS 0
PREDICTIONS	1	4	1
	0	1	3

$$TPR = \frac{TP}{TP + FN} = \frac{4}{4+1}$$

$$FPR = 1 - TPR = \frac{FP}{FP + TN} = \frac{1}{1+3}$$

		ACTUALS 1	ACTUALS 0
PREDICTIONS	1	5	0
	0	1	3

$$TPR = \frac{5}{5+1}$$

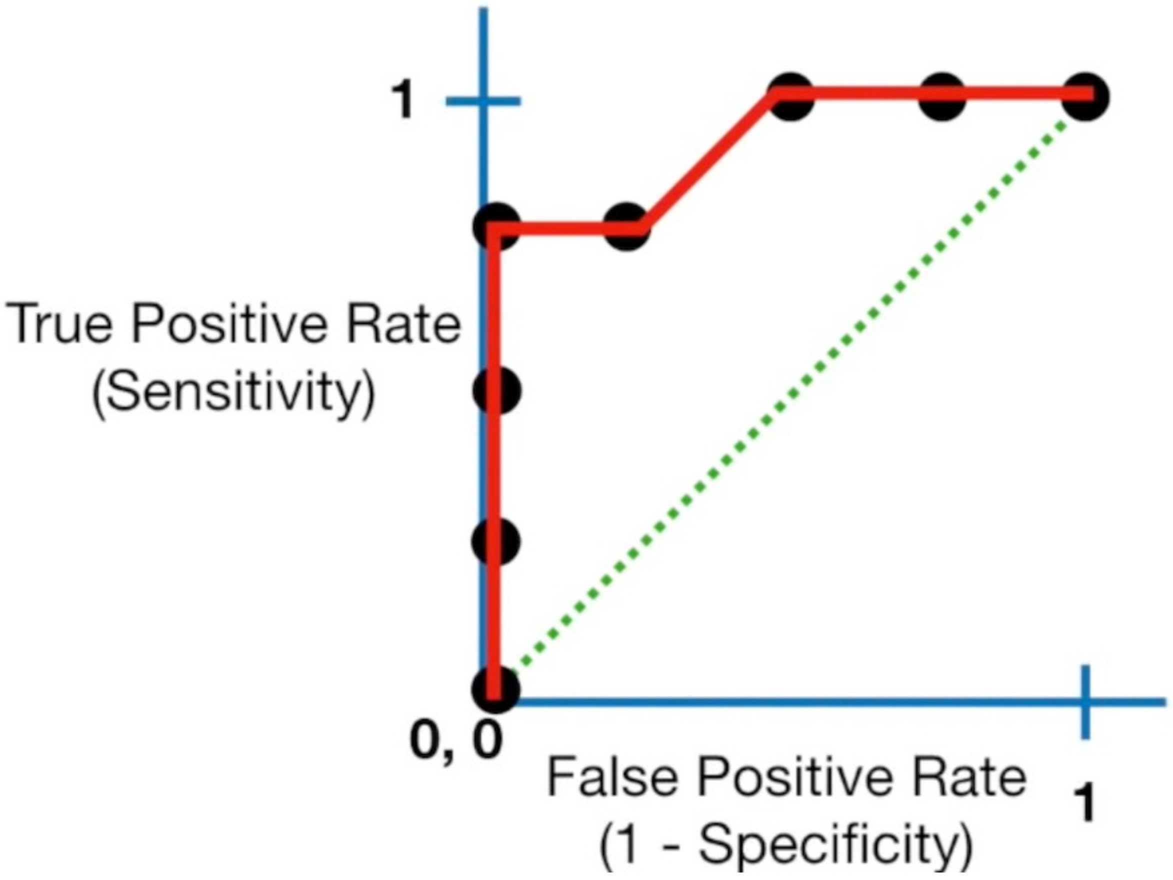
$$FPR = \frac{0}{0+3}$$

		ACTUALS 1	ACTUALS 0
PREDICTIONS	1	5	0
	0	2	2

$$TPR = \frac{5}{5+2}$$

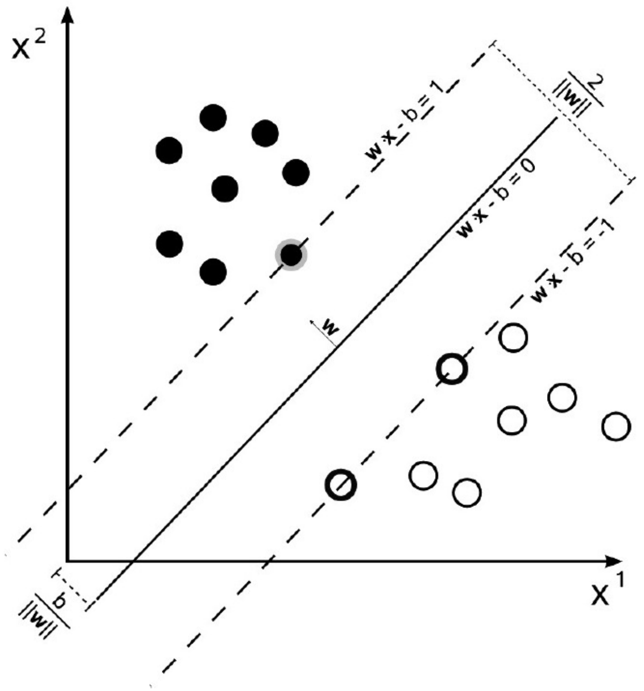
$$FPR = \frac{0}{0+2}$$

ROC-AUC



1
True Positive Rate
(Sensitivity)
0, 0
False Positive Rate
(1 - Specificity)
1

SVM – Hard Margin



$$f(x) = w^T x - b$$

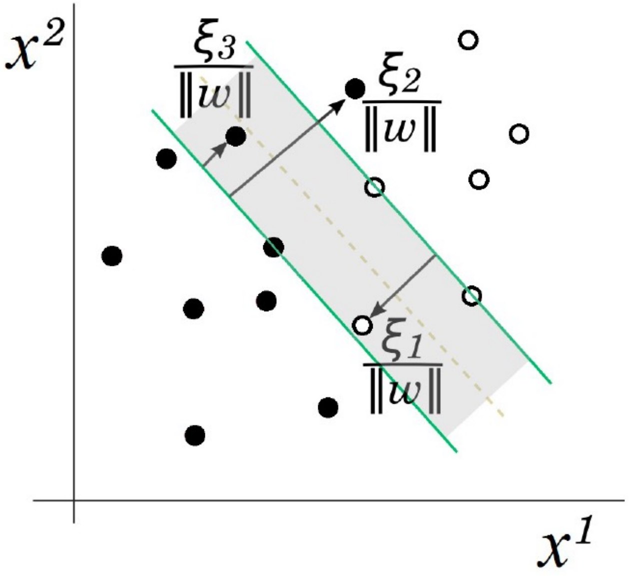
$$(w, b) = \arg\min_{w,b} ||w||^2$$

Constraints:

$$w^T x_i - b \geq \quad 1, \qquad x_i \in Class$$
$$w^T x_i - b \leq -1, \qquad x_i \notin Class$$

$$\Rightarrow \quad (w^T x_i - b) y_i \geq \quad 1$$

SVM – Soft Margin



$$\xi_i = \max\{1 - f(x_i)\, y_i, 0\}$$ Hinge loss

$$f(x) = w^T x - b$$

$$(w, b) = \underset{w,b}{\arg\min} \sum_i \xi_i + \lambda \|w\|^2$$

$$\lambda > 0$$

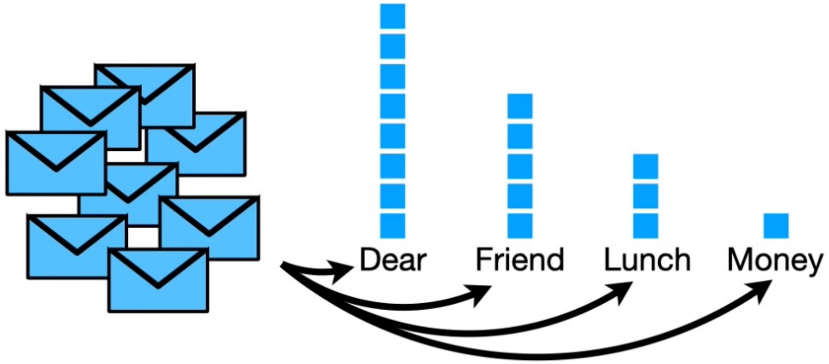
Constraints, for each i:

$$\xi_i \geq 0$$
$$(w^T x_i - b)\, y_i \geq 1 - \xi_i$$

Naïve Bayes

Likelihoods

$$p(word|N) = \frac{\#word}{\#total\ words\ in\ N}$$



p(**Dear** | **N**) = 0.47

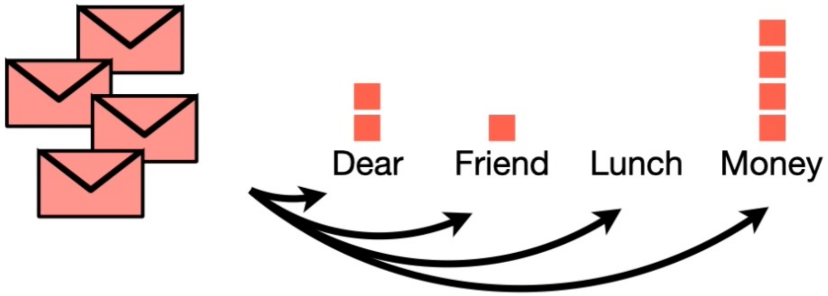
p(**Friend** | **N**) = 0.29

p(**Lunch** | **N**) = 0.18

p(**Money** | **N**) = 0.06

Prior *probability* $p(N) = \frac{\#N}{\#total\ emails}$

$$p(word|S) = \frac{\#word}{\#total\ words\ in\ S}$$



p(**Dear** | **S**) = 0.29

p(**Friend** | **S**) = 0.14

p(**Lunch** | **S**) = 0.00

p(**Money** | **S**) = 0.57

Prior *probability* $p(S) = \frac{\#S}{\#total\ emails}$

Dear Friend 

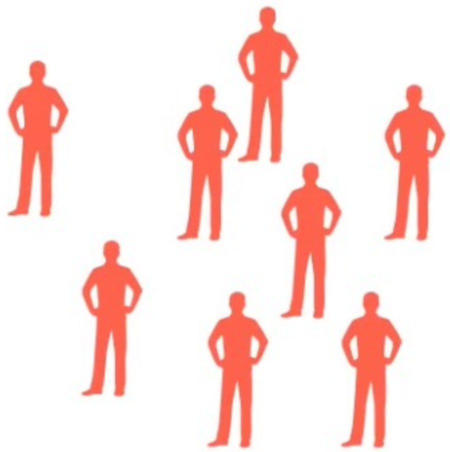
p(**N**) x p(**Dear** | **N**) x p(**Friend** | **N**)

p(**S**) x p(**Dear** | **S**) x p(**Friend** | **S**)

Naïve Bayes - Gaussian

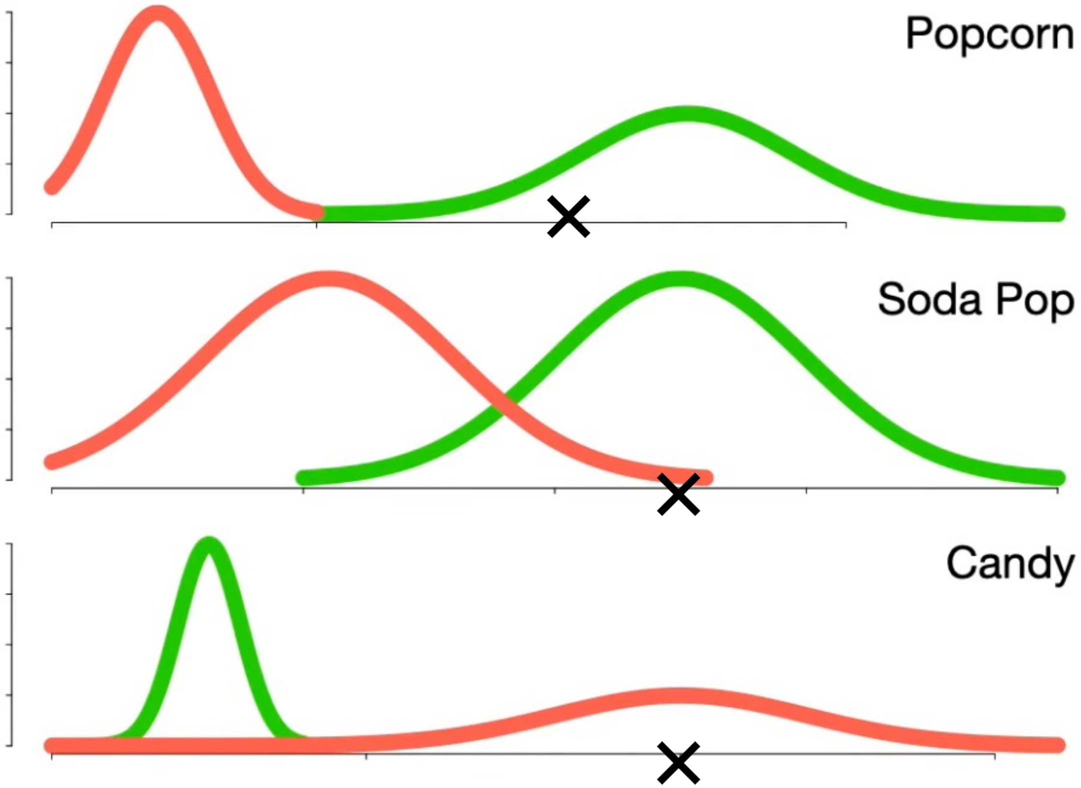
$$p(G) = \frac{\#G}{\#total}$$

Popcorn (grams)	Soda Pop (ml)	Candy (grams)
24.3	750.7	0.2
28.2	533.2	50.5
etc.	etc.	etc.



$$p(R) = \frac{\#R}{\#total}$$

Popcorn (grams)	Soda Pop (ml)	Candy (grams)
2.1	120.5	90.7
4.8	110.9	102.3
etc.	etc.	etc.



$$p(G) \times L(popcorn = 20|G) \times L(soda\ pop|G) \times L(candy = 25|G)$$

$$p(R) \times L(popcorn = 20|R) \times L(soda\ pop|R) \times L(candy = 25|R)$$

Decission Tree - CART

Gini:

$$H(Q_m) = \sum_k p_{mk}(1 - p_{mk})$$

Log Loss or Entropy:

$$H(Q_m) = - \sum_k p_{mk} \log(p_{mk})$$

Random Forrest

Step 1: Bootstraping

Original Dataset

Chest Pain	Good Blood Circ.	Blocked Arteries	Weight	Heart Disease
No	No	No	125	No
Yes	Yes	Yes	180	Yes
Yes	Yes	No	210	No
Yes	No	Yes	167	Yes

Bootstrapped Dataset

Chest Pain	Good Blood Circ.	Blocked Arteries	Weight	Heart Disease
Yes	Yes	Yes	180	Yes
No	No	No	125	No
Yes	No	Yes	167	Yes
Yes	No	Yes	167	Yes

Random Forrest

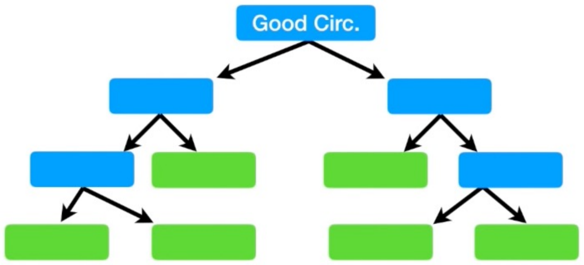
Step 2: Create Decission Tree .

Randomly select subset of features.

Find best split.

Randomly select subset of features to node split.

Create Tree considering subset of features.

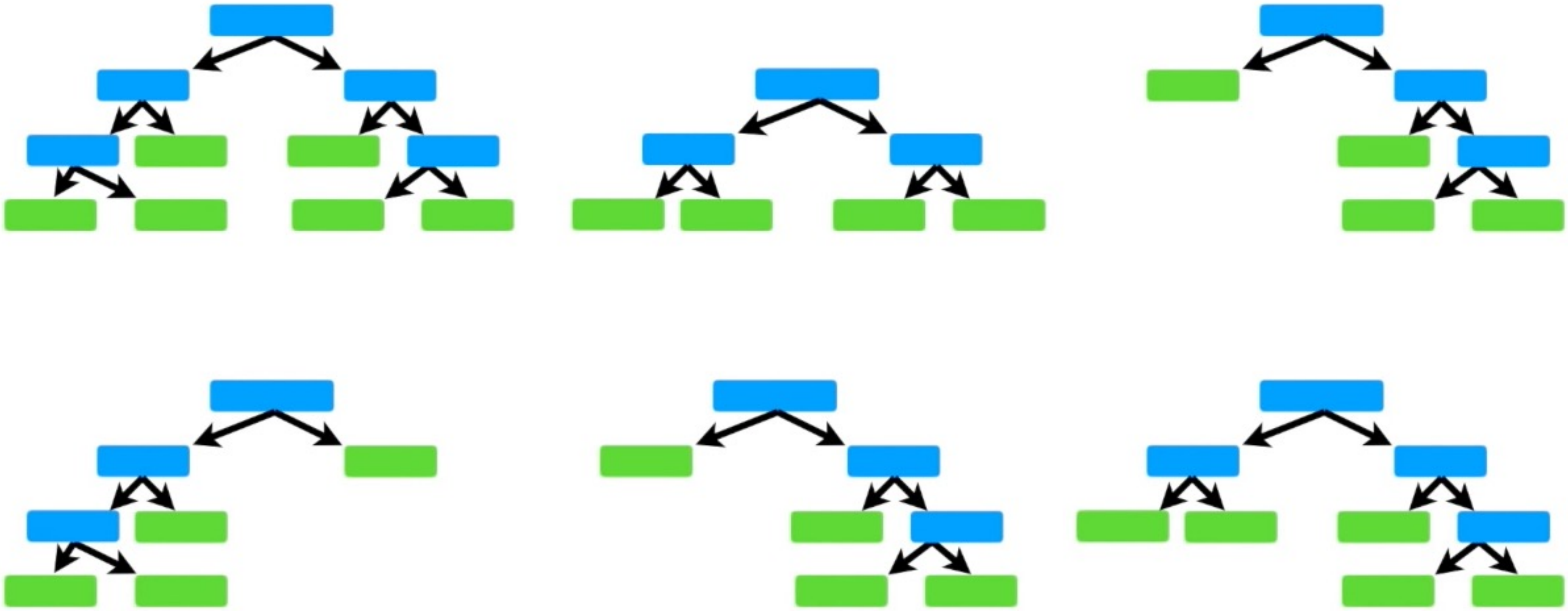


Bootstrapped Dataset

Chest Pain	Good Blood Circ.	Blocked Arteries	Weight	Heart Disease
Yes	Yes	Yes	180	Yes
No	No	No	125	No
Yes	No	Yes	167	Yes
Yes	No	Yes	167	Yes

Random Forrest

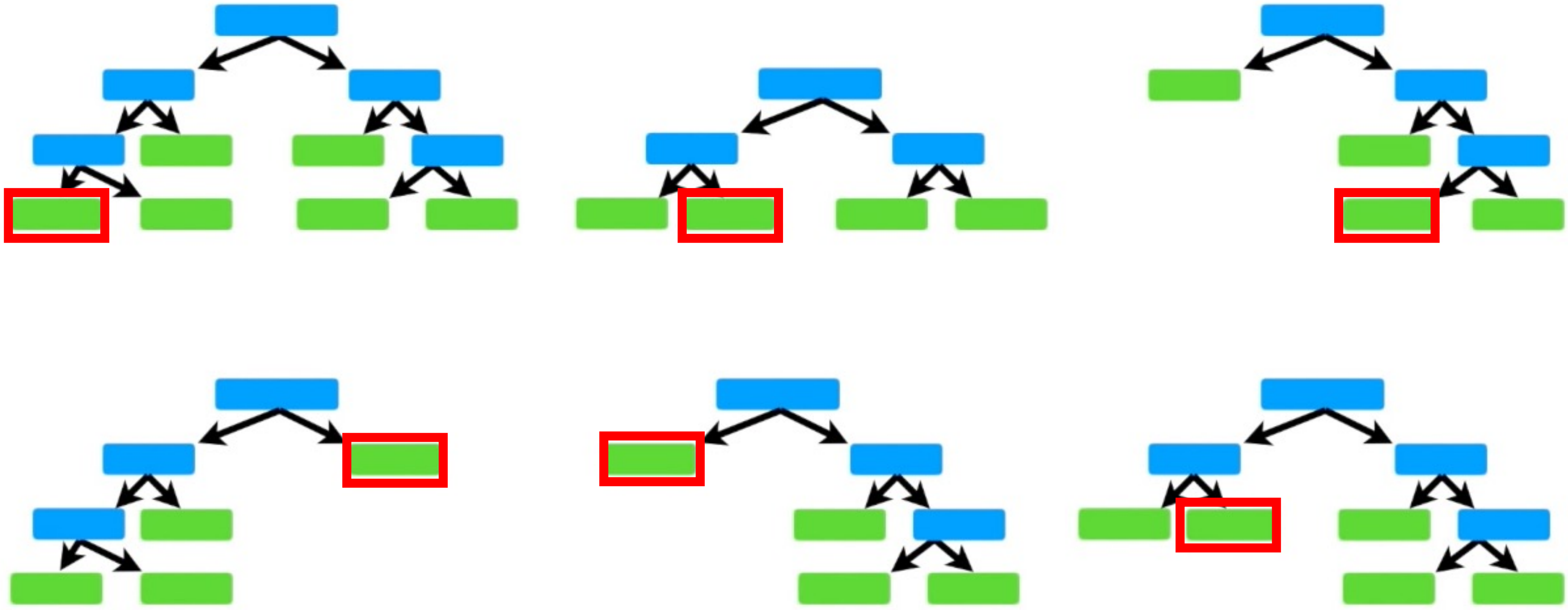
Repeat Step 1 & Step 2 creating next trees.



Random Forrest

Predictions.

Chest Pain	Good Blood Circ.	Blocked Arteries	Weight	Heart Disease
Yes	No	No	168	



Regresyjne Lasy Losowe

Out-Of-Bag Dataset

Bootstrapped Dataset

Chest Pain	Good Blood Circ.	Blocked Arteries	Weight	Heart Disease
Yes	Yes	Yes	180	Yes
No	No	No	125	No
Yes	No	Yes	167	Yes
Yes	No	Yes	167	Yes

Original Dataset

Chest Pain	Good Blood Circ.	Blocked Arteries	Weight	Heart Disease
No	No	No	125	No
Yes	Yes	Yes	180	Yes
Yes	Yes	No	210	No
Yes	No	Yes	167	Yes

Chest Pain	Good Blood Circ.	Blocked Arteries	Weight	Sugar
Yes	Yes	No	210	96.50

We can make predictions for *oob* subset and calculate metrics.

In sklearn module sklearn.ensemble. RandomForestClassifier has ***oob_score_*** *attrbiiute returning accuracy.*

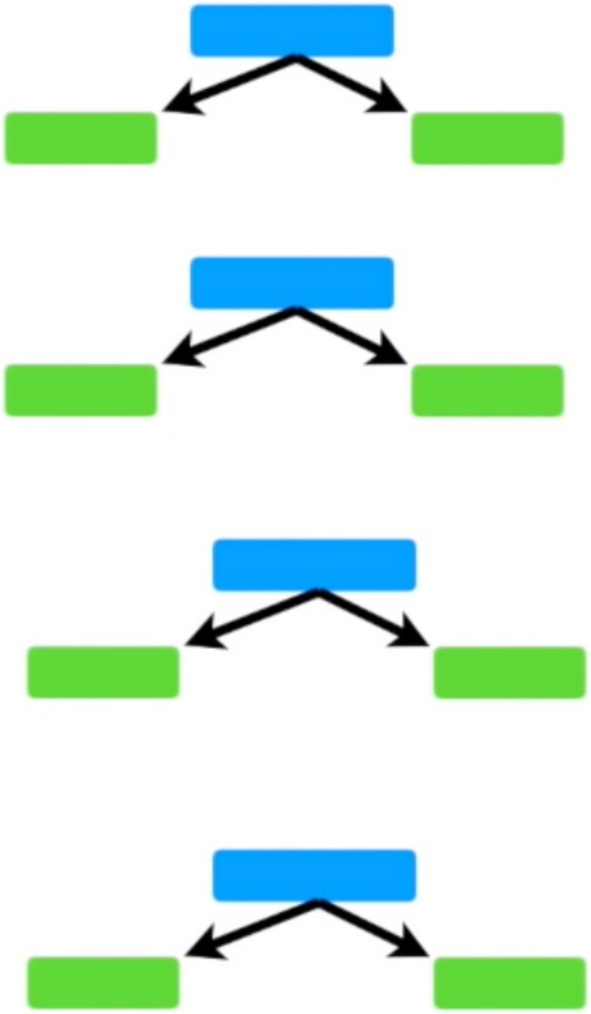
oob_score_ : *float*

Score of the training dataset obtained using an out-of-bag estimate. This attribute exists only when `oob_score` is True.

AdaBoost

Original Dataset

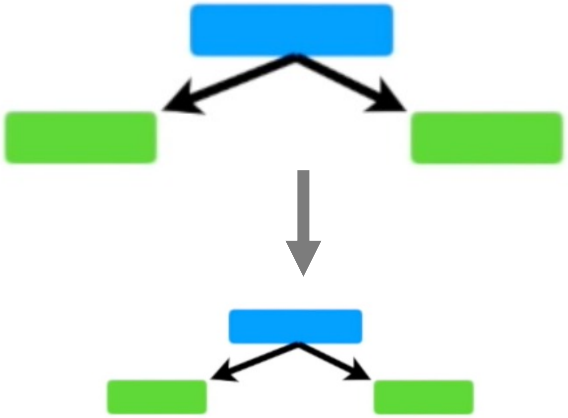
Chest Pain	Good Blood Circ.	Blocked Arteries	Weight	Heart Disease
No	No	No	125	No
Yes	Yes	Yes	180	Yes
Yes	Yes	No	210	No
Yes	No	Yes	167	Yes



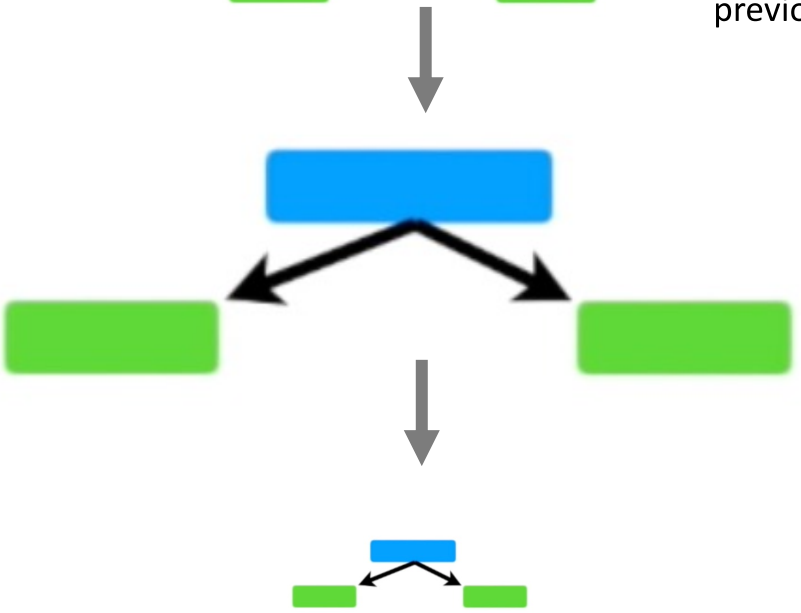
stump

AdaBoost

Create stumps

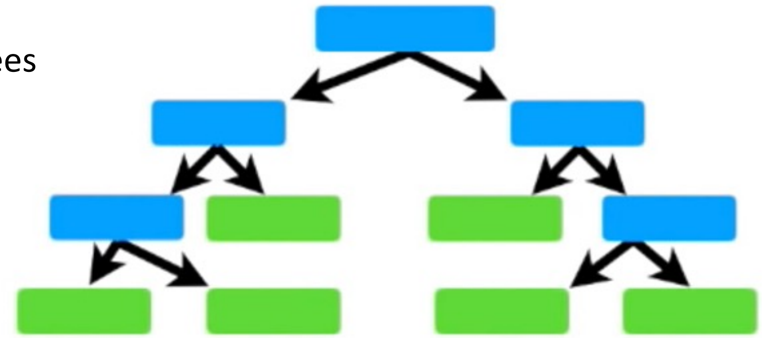


Stumps create based on previous stump.

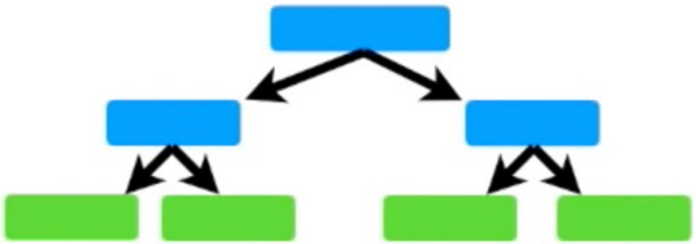


Stumps have no equal weights.

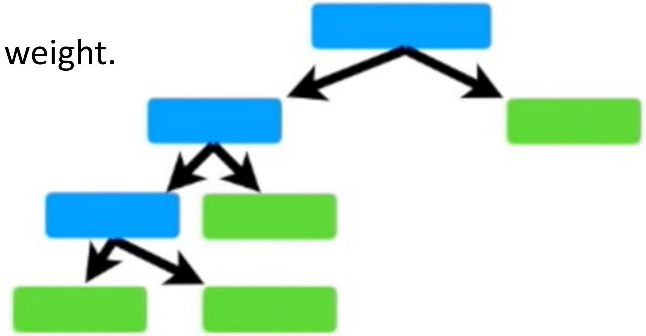
Create full trees



Independent trees



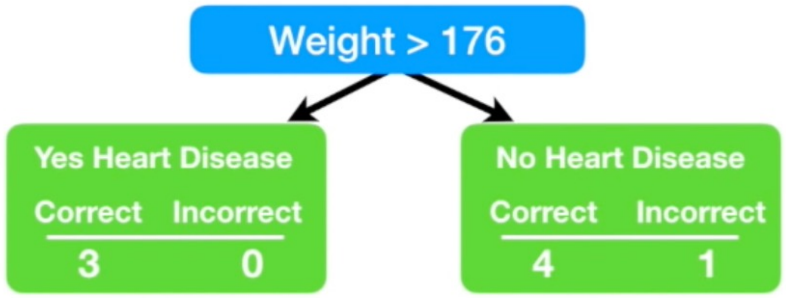
Each tree has equal weight.



AdaBoost

Step 1. Find best split minimizing Gini

Chest Pain	Blocked Arteries	Patient Weight	Heart Disease	Sample Weight
Yes	Yes	205	Yes	1/8
No	Yes	180	Yes	1/8
Yes	No	210	Yes	1/8
Yes	Yes	167	Yes	1/8
No	Yes	156	No	1/8
No	Yes	125	No	1/8
Yes	No	168	No	1/8
Yes	Yes	172	No	1/8



AdaBoost

Step 1. Find best split minimizing Gini

Step 2. Update sample weights.

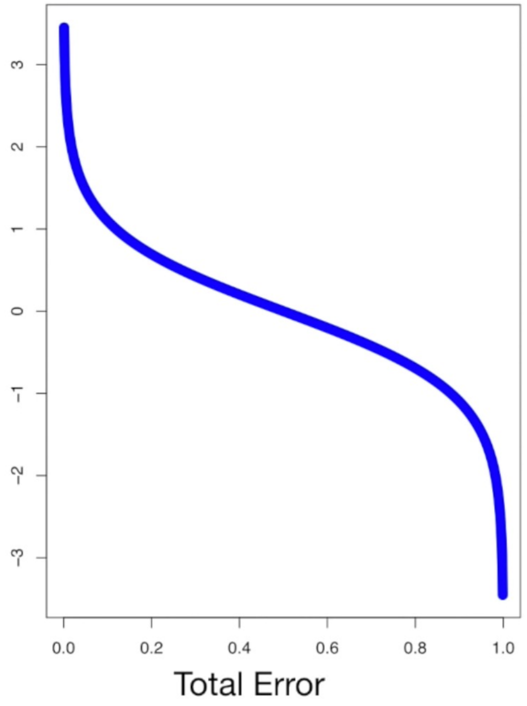
Chest Pain	Blocked Arteries	Patient Weight	Heart Disease	New Weight
Yes	Yes	205	Yes	0.05
No	Yes	180	Yes	0.05
Yes	No	210	Yes	0.05
Yes	Yes	167	Yes	0.33
No	Yes	156	No	0.05
No	Yes	125	No	0.05
Yes	No	168	No	0.05
Yes	Yes	172	No	0.05

$$I = \frac{1}{2}\log\left(\frac{1 - total\_error}{total\_error}\right)$$

$total\_error$ - sum of incorrect classified samples weights

$New\ weigth = weight \times e^{I}$ - True

$New\ weigth = weight \times e^{-I}$ - False



AdaBoost

Step 1. Find best split minimizing Gini

Step 2. Update sample weights.

Step 3. Normalize sample weights.

Chest Pain	Blocked Arteries	Patient Weight	Heart Disease	Sample Weight
Yes	Yes	205	Yes	0.07
No	Yes	180	Yes	0.07
Yes	No	210	Yes	0.07
Yes	Yes	167	Yes	0.49
No	Yes	156	No	0.07
No	Yes	125	No	0.07
Yes	No	168	No	0.07
Yes	Yes	172	No	0.07

AdaBoost

Step 1. Find best split minimizing Gini

Step 2. Update sample weights.

Step 3. Normalize sample weights.

Step 3. Bootstrap dataset using new sample weights.

Chest Pain	Blocked Arteries	Patient Weight	Heart Disease	Sample Weight
Yes	Yes	205	Yes	0.07
No	Yes	180	Yes	0.07
Yes	No	210	Yes	0.07
Yes	Yes	167	Yes	0.49
No	Yes	156	No	0.07
No	Yes	125	No	0.07
Yes	No	168	No	0.07
Yes	Yes	172	No	0.07

Chest Pain	Blocked Arteries	Patient Weight	Heart Disease	Sample Weight
No	Yes	156	No	1/8
Yes	Yes	167	Yes	1/8
No	Yes	125	No	1/8
Yes	Yes	167	Yes	1/8
Yes	Yes	167	Yes	1/8
Yes	Yes	172	No	1/8
Yes	Yes	205	Yes	1/8
Yes	Yes	167	Yes	1/8

AdaBoost

Chest Pain	Blocked Arteries	Patient Weight	Heart Disease	Sample Weight
No	Yes	156	No	1/8
Yes	Yes	167	Yes	1/8
No	Yes	125	No	1/8
Yes	Yes	167	Yes	1/8
Yes	Yes	167	Yes	1/8
Yes	Yes	172	No	1/8
Yes	Yes	205	Yes	1/8
Yes	Yes	167	Yes	1/8

Step 1. Find best split minimizing Gini

Step 2. Update sample weights.

Step 3. Normalize sample weights.

Step 4. Bootstrap dataset using new sample weights.

Step 5. Repeat using bootstrap dataset.

AdaBoost

$$I = \frac{1}{2}\log(\frac{1 - total\_error}{total\_error})$$

Chest Pain	Blocked Arteries	Patient Weight	Heart Disease
No	Yes	156	No

Calculate I

0.97

0.32

0.78

0.63

$\boxed{\text{Total=2.7}}$

Calculate I

0.41

0.82

Total=1.23

https://machinelearningmastery.com

https://towardsdatascience.com

https://www.youtube.com/c/joshstarmer

https://www.youtube.com/c/3blue1brown

https://www.youtube.com/c/TechWorldwithNana

https://www.youtube.com/c/TechWithTim

https://www.youtube.com/channel/UCMEQFEKrsRFBXnUIreTACxg

https://www.youtube.com/c/TensorFlow