



Machine Learning en **Spark** con **MLlib**

Rafael Caballero Roldán

+ Spark ML lib

Aprendizaje supervisado

El conjunto de entrenamiento
contiene valores junto con su
resultado. El programa
entrenado debe ser capaz de
predecir nuevos valores

Modelos lineales: regresión lineal y logística
Métricas

Naïve Bayes

Árboles de decisión



+ Spark ML lib

Aprendizaje supervisado

El conjunto de entrenamiento
contiene valores junto con su
resultado. El programa
entrenado debe ser capaz de
predecir nuevos valores

Modelos lineales: regresión lineal y logística

Métricas

Naïve Bayes

Árboles de decisión



Modelos lineales, planteamiento general

Partimos de:

- ✓ Un conjunto de n datos de entrenamiento $x_i \in \mathbb{R}^d$, $1 \leq i \leq n$
- ✓ Los valores asociados (aprendizaje supervisado) $y_i \in \mathbb{R}$, $1 \leq i \leq n$
- ✓ Un parámetro de regularización $\lambda \in \mathbb{R}$, $\lambda \geq 0$ (**regParam**) para evitar overfitting

Queremos:

- ✓ Para una cierta función lineal L y función de regularización R , se quiere encontrar un parámetro w que minimizan una función convexa f , $\min_{w \in \mathbb{R}^d} f(w)$

$$f(w) := \lambda R(w) + \frac{1}{n} \sum_{i=1}^n L(w; x_i, y_i)$$

Regresión Lineal

Predecir valores a partir de la correlación entre dos o más variables



Instancia de la fórmula general

$$L(w; x, y) := \frac{1}{2}(w^T x - y)^2$$



Correlación

La correlación indica el grado de correlación, pero no sirve para predecir resultados



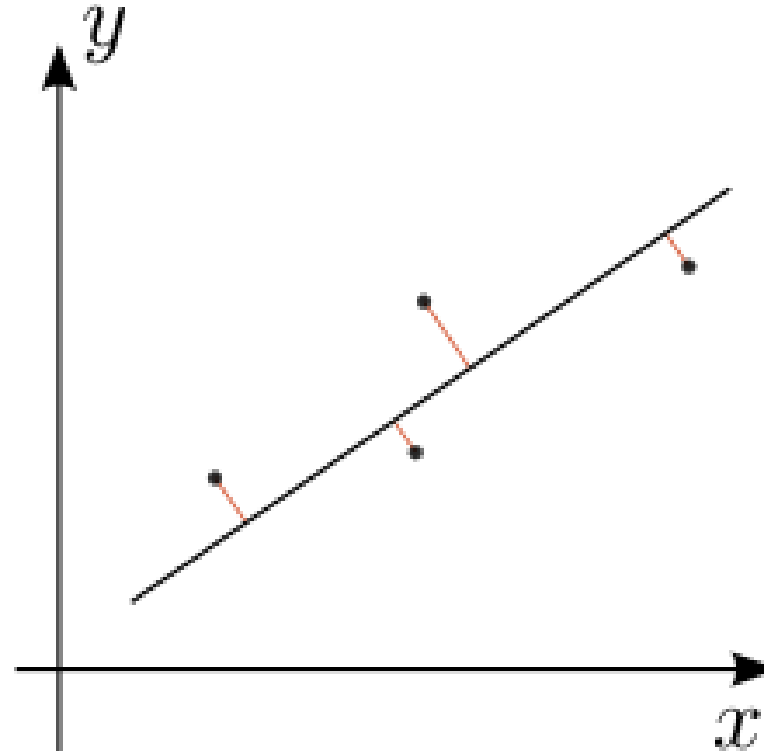
Parámetros

Se buscan los parámetros w que definan una recta que mejor “ajuste” al conjunto de entrenamiento



Error

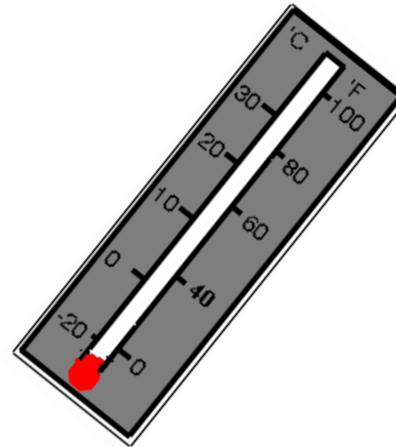
Se suele utilizar el error cuadrático medio para estimar la bondad del método





Ejemplo

Obtener una estimación
para las **ventas de helados**
a partir de la **temperatura
media** prevista para un día



Predicción de ventas de helados (pr. RegresionLineal)

Fichero de datos

```
215.0 1:24.2
325.0 1:26.4
185.0 1:21.9
332.0 1:25.2
406.0 1:28.5
522.0 1:32.1
412.0 1:29.4
614.0 1:35.1
544.0 1:33.4
421.0 1:28.1
445.0 1:32.6
408.0 1:27.2
```

Coeficientes: [30.088]

Intercept: -460.35

+-----+

| residuals|

+-----+

|-52.773485137420096|

|-8.966781084759248|

|-13.571403010656411|

| 34.138653068334804|

| 8.848709147326133|

| 16.532406688043864|

|-12.230366467494434|

| 18.26882130530862|

|-0.5818136444745505|

| 35.88385386502415|

|-75.51152420907874|

| 49.96292947984472|

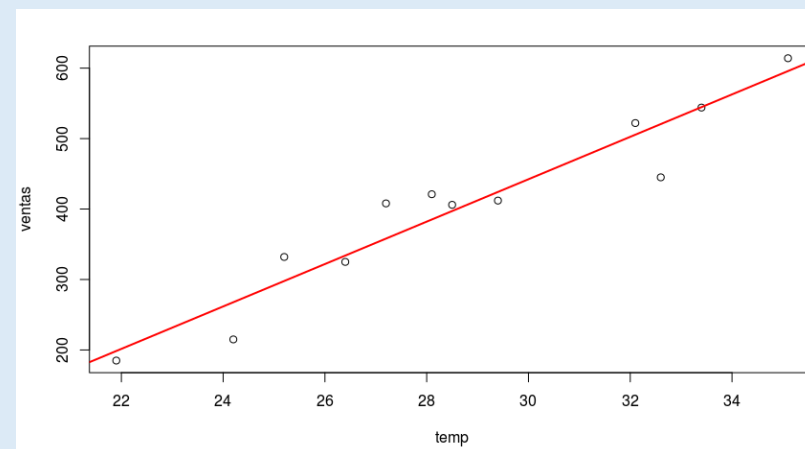
+-----+

RMSE: 34.80457395200398

r2: 0.9168189330919188

Temperatura: 30

Ventas estimadas: 442.2830835440415



Predicción de ventas de helados (pr. RegresionLineal)

215.0 1:24.2
325.0 1:26.4
185.0 1:21.9
332.0 1:25.2
406.0 1:28.5
522.0 1:32.1
412.0 1:29.4
614.0 1:35.1
544.0 1:33.4
421.0 1:28.1
445.0 1:32.6
408.0 1:27.2

Coeficientes: [30.088]

Intercept: -460.35

+-----+

| residuals|

+-----+

|-52.773485137420096|

|-8.966781084759248|

|-13.571403010656411|

| 34.138653068334804|

| 8.848709147326133|

| 16.532406688043864|

|-12.230366467494434|

| 18.26882130530862|

|-0.5818136444745505|

| 35.88385386502415|

|-75.51152420907874| > 2 RSME

| 49.96292947984472|

+-----+

RMSE: 34.80457395200398

r2: 0.9168189330919188

Temperatura: 30

Ventas estimadas: 442.2830835440415

+ Spark ML lib

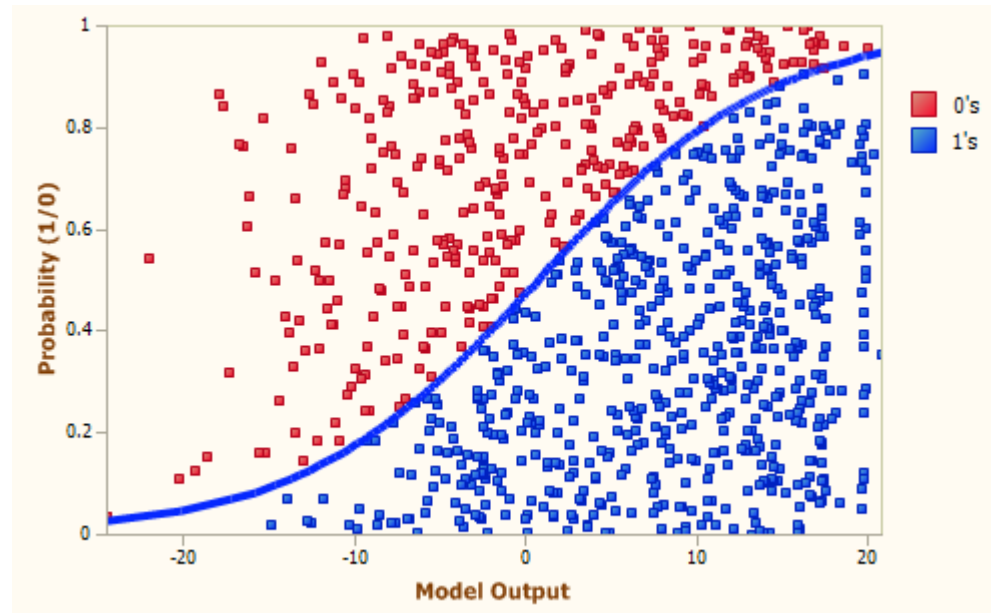
Aprendizaje supervisado

El conjunto de entrenamiento contiene valores junto con su resultado. El programa entrenado debe ser capaz de predecir nuevos valores

Modelos lineales: regresión logística

Naïve Bayes

Árboles de decisión

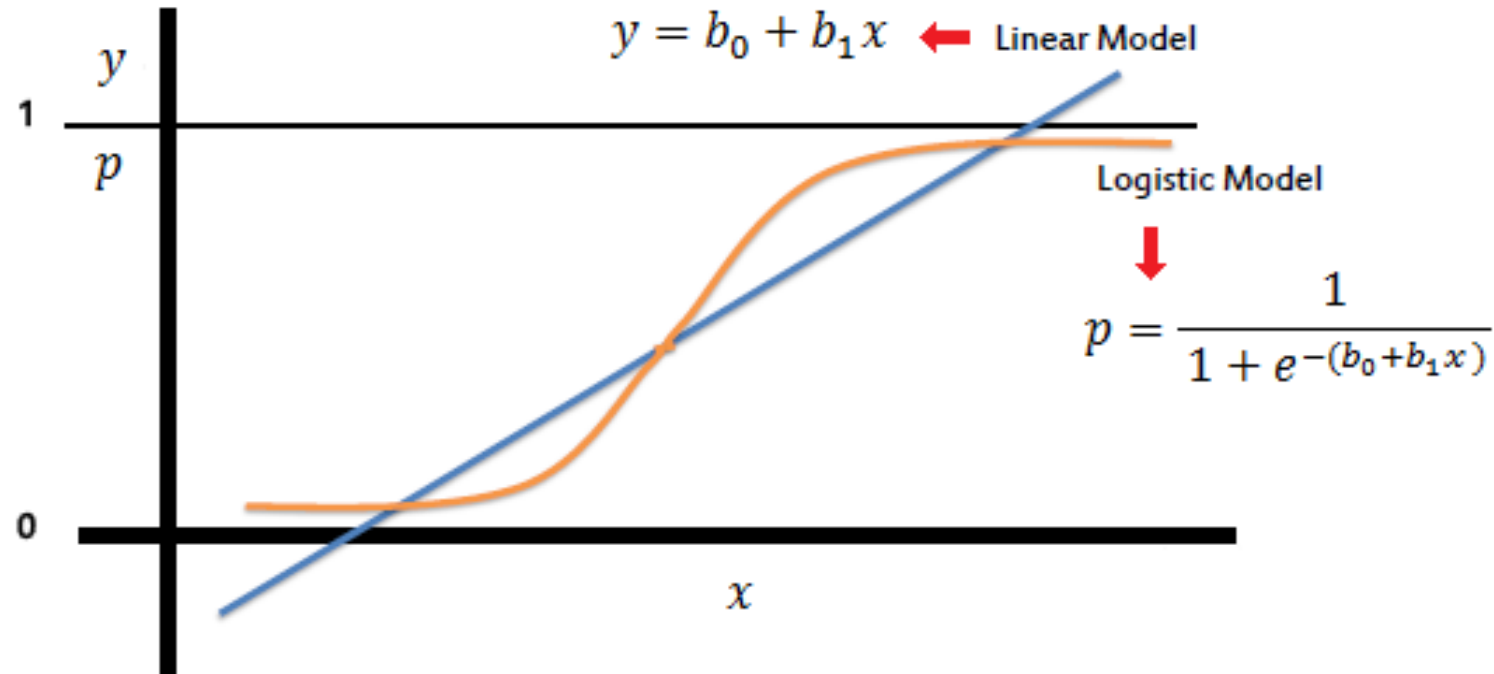


Regresión Logística

- ✓ Empleada principalmente para etiquetas binarias
- ✓ No requiere independencia entre las características
- ✓ Mejor que Bayes para poblaciones muy grandes

Regresión Logística

Nuestro primer ejemplo de clasificación



Idea

Se utiliza una función que extrema los resultados, intentado aproximar a dos valores: 0 y 1



Ejemplo

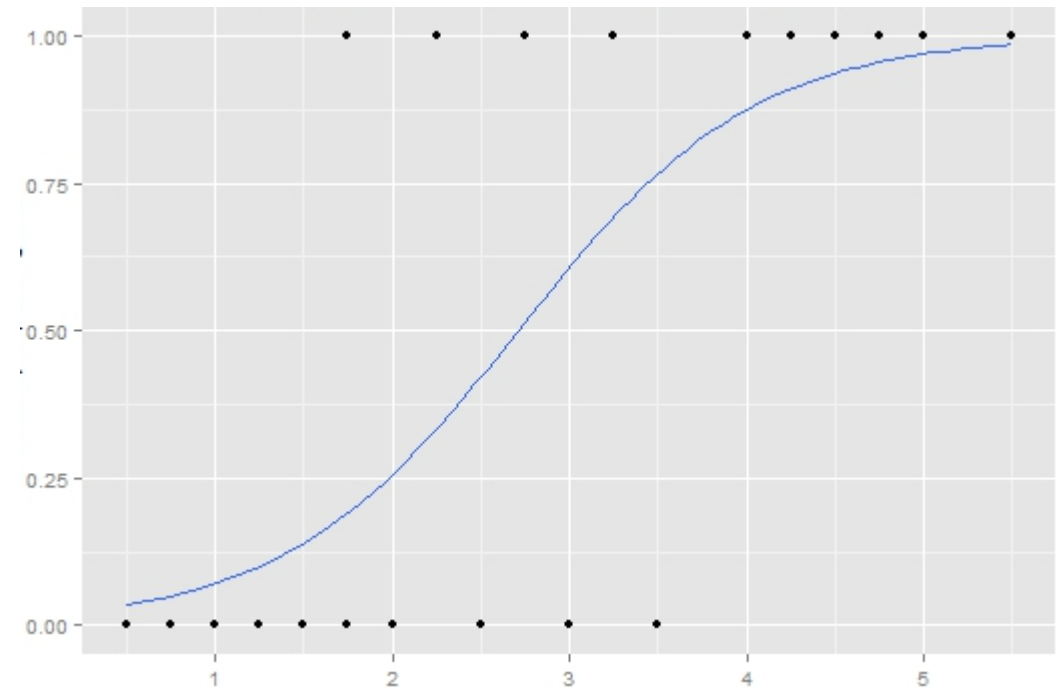
Tenemos una colección de datos que representa para cierto grupo de persona si han sido capaces de pasar un test matemático tras ingerir una cierta cantidad de una droga.

Queremos ser capaces de de obtener predicciones fiables.

Ejemplo: pasar el test

Si calculamos las probabilidades para poblaciones de este tipo a menudo obtenemos la llamada **función logística**.

El método usa la forma de esta función para obtener la función de aproximación



+ Spark ML lib

Métricas

Las métricas nos permiten definir el rendimiento de nuestro sistema con respecto a unos estándares, así como entender las respuestas proporcionadas por varios métodos

La exportación a PPML permite conectarse con otras herramientas

Modelos lineales: regresión

Métricas

Naïve Bayes

Excursión 2: etiquetas no numéricas y pipelines

Árboles de decisión



Predicciones: terminología

	<u>Predicted 1</u>	<u>Predicted 0</u>
<u>True 1</u>	true positive	false negative
<u>True 0</u>	false positive	true negative

	<u>Predicted 1</u>	<u>Predicted 0</u>
<u>True 1</u>	TP	FN
<u>True 0</u>	FP	TN

	<u>Predicted 1</u>	<u>Predicted 0</u>
<u>True 1</u>	hits	misses
<u>True 0</u>	false alarms	correct rejections

	<u>Predicted 1</u>	<u>Predicted 0</u>
<u>True 1</u>	$P(\text{pr1} \text{tr1})$	$P(\text{pr0} \text{tr1})$
<u>True 0</u>	$P(\text{pr1} \text{tr0})$	$P(\text{pr0} \text{tr0})$

Precisión: $a / (a+c)$

Proporción de los valores predichos que resulta acertada

Ejemplo: un clúster debe tener 25 miembros. El método le asigna solo 15, pero todos correctos

Precisión = $15 / (15+0) = 1$

	<u>Predicho 1</u>	<u>Predicho 0</u>
<u>True 1</u>	a	b
<u>True 0</u>	c	d

Exhaustividad (recall): $a / (a+b)$

Proporción de los valores correctos que ha sido predicha

Ejemplo: un clúster debe tener 25 miembros. El método le asigna solo 15, 10 correctos, 5 incorrectos

Precisión = $10 / (10 + 5) = 0,66$

Recall = $10 / 25 = 0,4$

	<u>Predicho 1</u>	<u>Predicho 0</u>
<u>True 1</u>	a	b
<u>True 0</u>	c	d

Accuracy: $(a+d) / (a+b+c+d)$

Proporción entre la suma de verdaderos positivos más verdaderos negativos con respecto a la población total

	<u>Predicho 1</u>	<u>Predicho 0</u>
<u>True 1</u>	a	b
<u>True 0</u>	c	d

Más ejemplos

En una clasificación binaria sobre una muestra se debe tener 90 sí y 10 no

Un sistema “**optimista**” que siempre clasifica **todos** como sí tendrá
.9 de precisión , 1 de recall

Un sistema “**pesimista**” que prediga 1 sí y se equivoque
0 de precisión (0/1), y 0 de recall (0 de 90)

Un sistema “**humilde**” que prevea 10 síes, todos acertados
1 de precisión, 10/90 de recall (0.11)

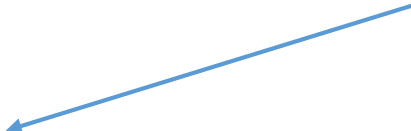
Combinando precisión y recall

Para combinar ambas se introducir el factor F, media armónica de precisión y recall

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

- F1=1 si precisión=1 y recall=1
- F1=0 sí alguno de los es cero (y el otro no)
- F1 indefinido si los dos son 0
- Precisión = 0.8 y recall=0.1 \rightarrow F1 = 0.08
- Precisión = 0.4 y recall=0.4 \rightarrow F1 = 0.2

F mide el
“equilibrio”



Generalización de F

En ocasiones se puede encontrar una generalización de F de la forma:

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

F_2 le da más importancia a **recall**

$F_{0.5}$ le da más importancia a la **precisión**

Precisión = 0.1 y recall=0.8 $\rightarrow F_1 = 0.08$, pero $F_2 = 0.4 / 1.2 = 0.3$

ROC

Receiver Operating Characteristic (ROC) es una curva útil para comprobar la eficiencia de un clasificador Binario

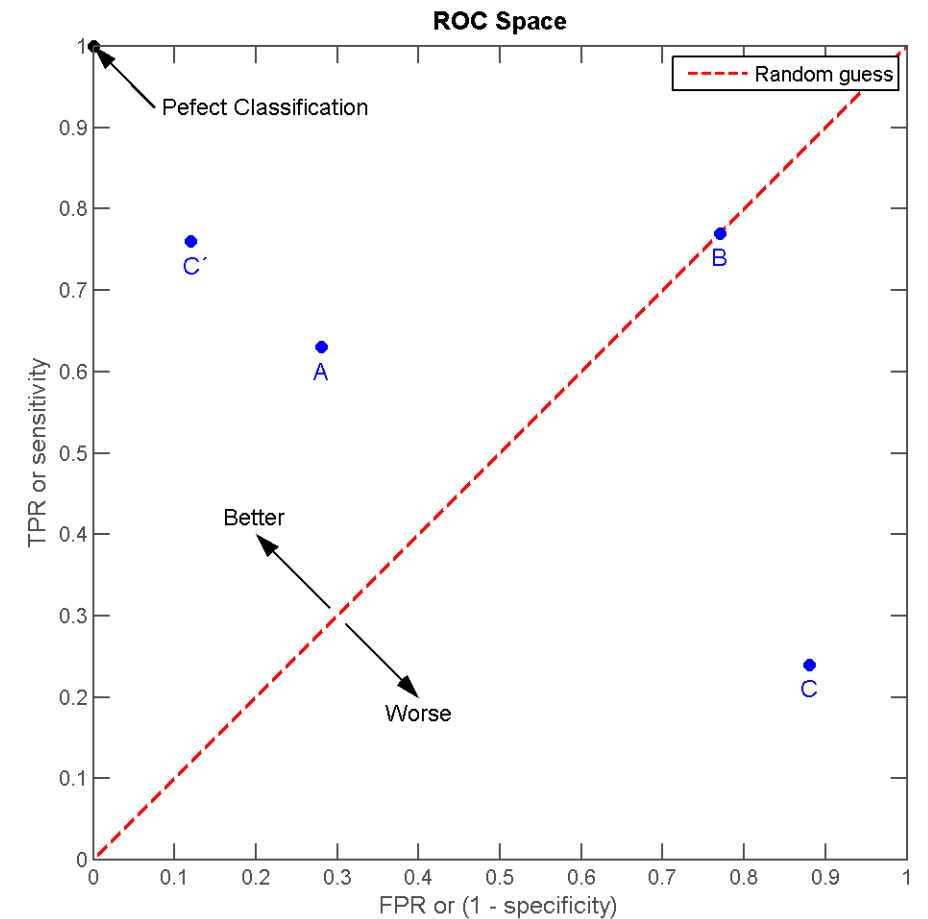
Representa el número de falsos positivos frente a falsos negativos para distintos umbrales (thresholds)

Útil para comparar modelos y quedarse con el más adecuado

También para cambiar los umbrales y hacer un “ajuste” del modelo orientándolo hacia falsos positivos o falsos negativos

ROC: ejemplo

A			B			C			C'		
TP=63	FP=28	91	TP=77	FP=77	154	TP=24	FP=88	112	TP=76	FP=12	88
FN=37	TN=72	109	FN=23	TN=23	46	FN=76	TN=12	88	FN=24	TN=88	112
100	100	200	100	100	200	100	100	200	100	100	200
TPR = 0.63			TPR = 0.77			TPR = 0.24			TPR = 0.76		
FPR = 0.28			FPR = 0.77			FPR = 0.88			FPR = 0.12		
PPV = 0.69			PPV = 0.50			PPV = 0.21			PPV = 0.86		
F1 = 0.66			F1 = 0.61			F1 = 0.22			F1 = 0.81		
ACC = 0.68			ACC = 0.50			ACC = 0.18			ACC = 0.82		



Kai walz (talk) - ROC_space.png, CC BY-SA 3.0,
<https://commons.wikimedia.org/w/index.php?curid=8326140>

ROC area

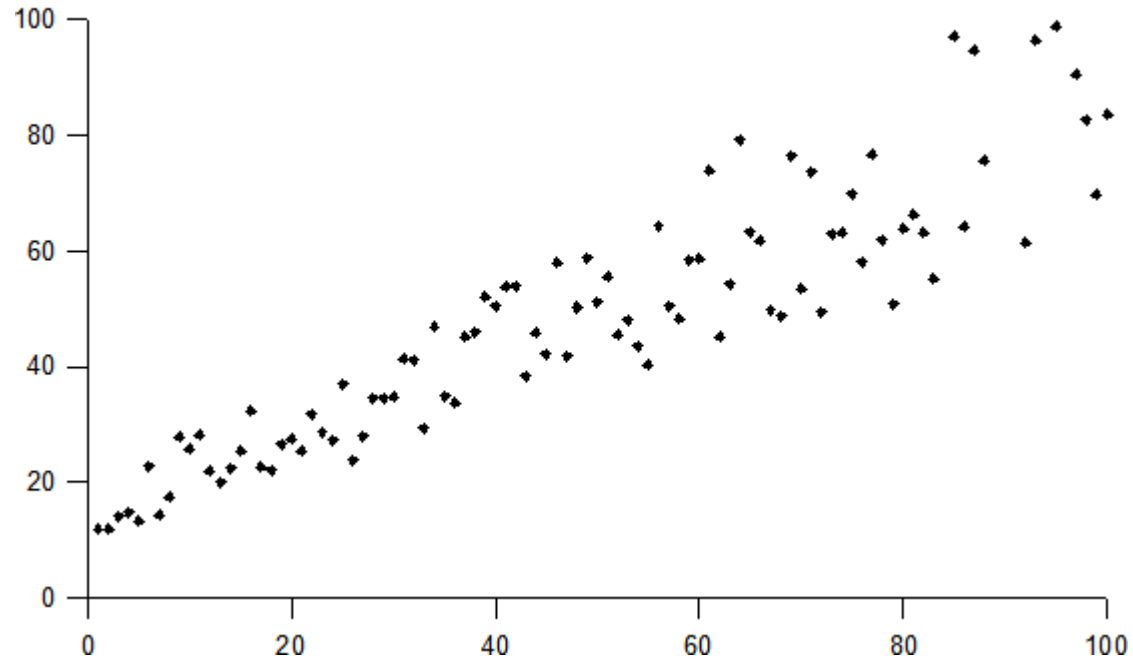
El área que queda bajo la curva ROC es un excelente indicador de Su eficiencia:

- ✓ Si el área es 1 la clasificación es perfecta
- ✓ Área 0.9: excelente predicción
- ✓ Área 0.7: Predicción mediocre
- ✓ Área 0.5: predicción que se obtiene al tirar un dado
- ✓ Área < 0.5 ¡¡¡ estamos haciendo algo mal!!!

Precauciones

Los modelos lineales piden una cuantas propiedades a los datos para ser aplicables

Heteroscedasticity



Distribución Normal

De los datos de entrada

Homocedasticidad

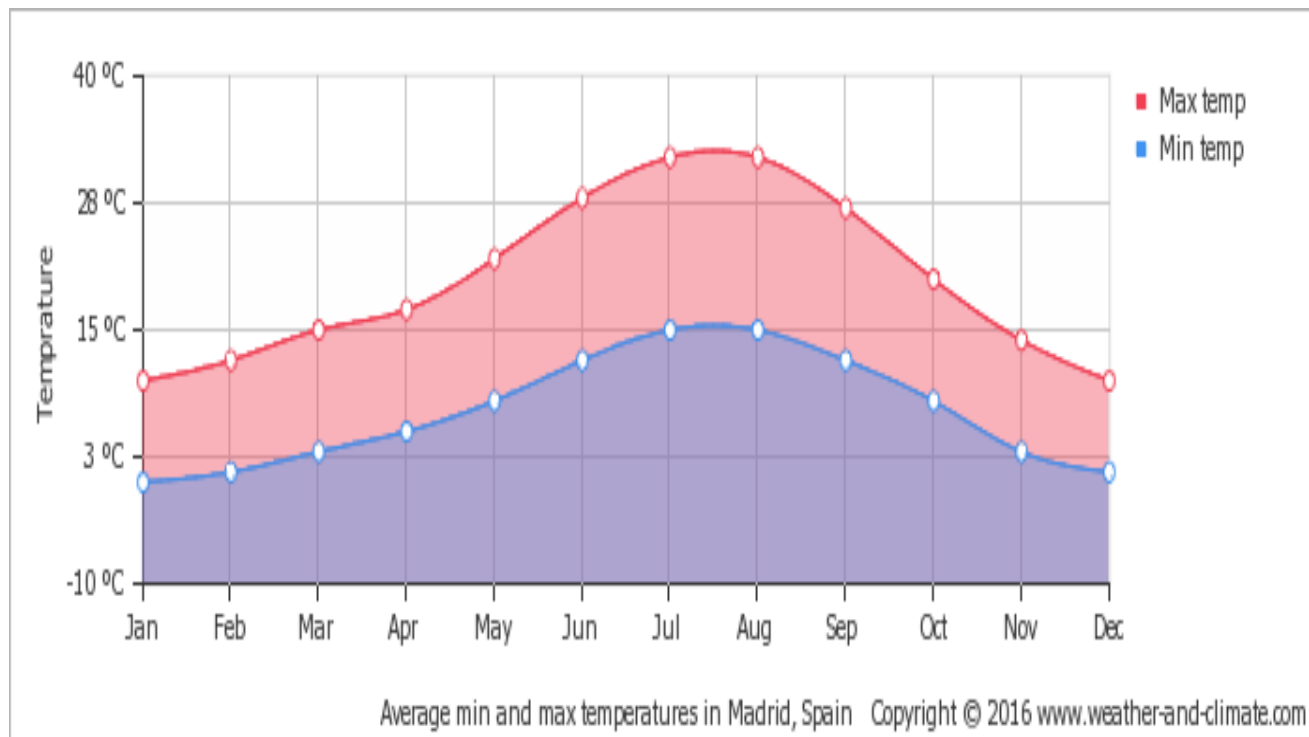
La varianza de y es homogénea sobre todos los valores de x

Independencia

Los valores de y para distintas x son independientes

¿Temperatura media es normal?

Es difícil determinar la normalidad de fenómenos físicos locales



Depende del lugar

En el ecuador no lo es, en otros lugares son dos normales (verano e invierno)

Madrid

Aspecto “razonable” de normal

+ Spark ML lib

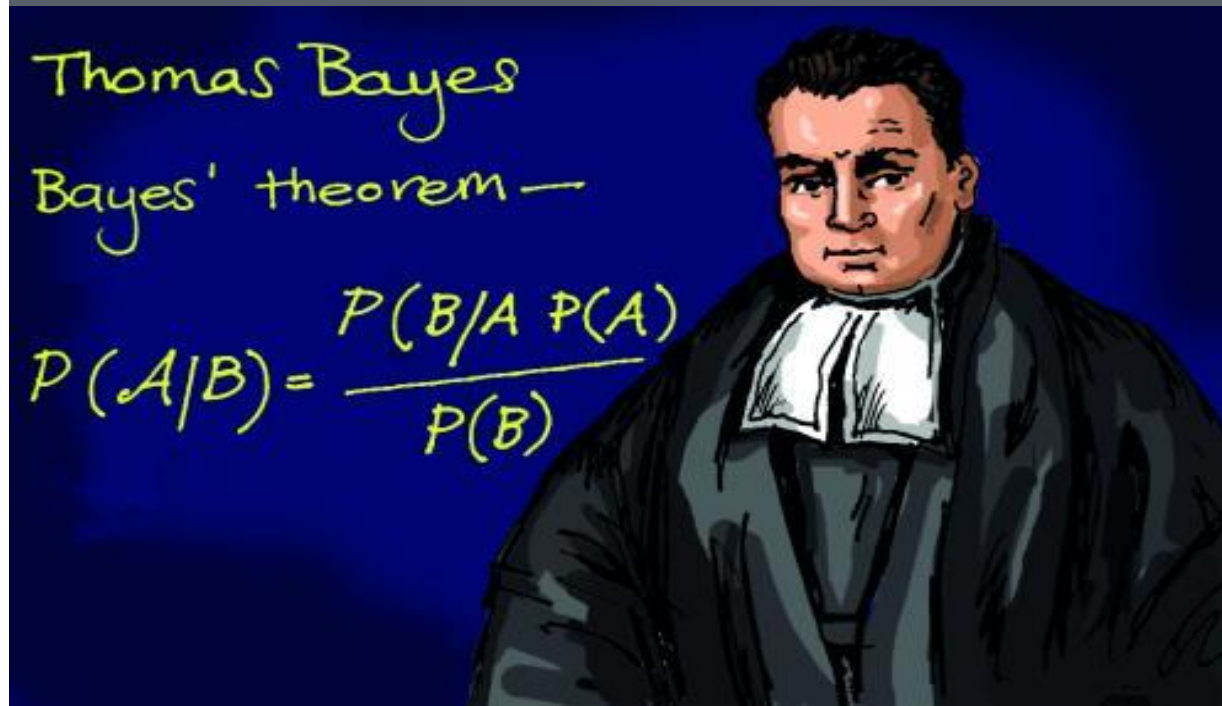
Aprendizaje supervisado

El conjunto de entrenamiento contiene valores junto con su resultado. El programa entrenado debe ser capaz de predecir nuevos valores

Modelos lineales: regresión
Métricas

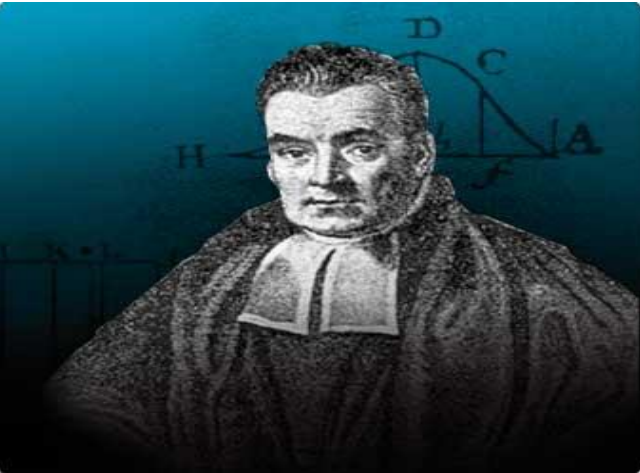
Naïve Bayes

Árboles de decisión



Naïve Bayes

La potencia de lo sencillo



Simplicidad

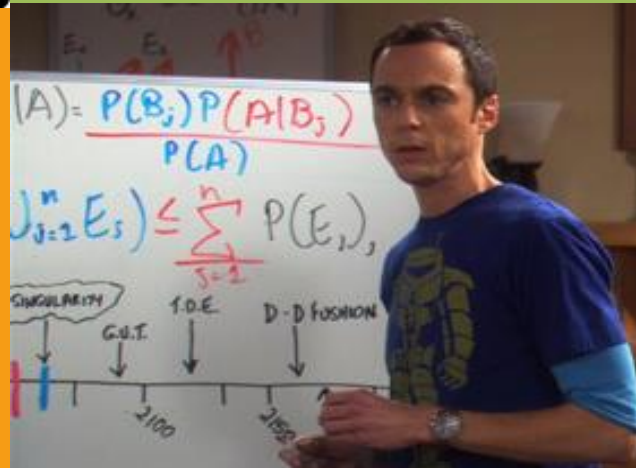
Fácil y rápido, muy
utilizado en
clasificación



Eficiente

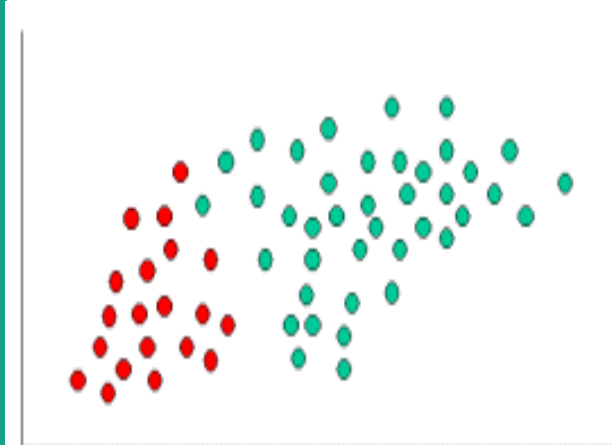
A menudo más
eficiente que otros
métodos más
complejos

Probabilidades
Método basado
directamente en el
cálculo de
probabilidades



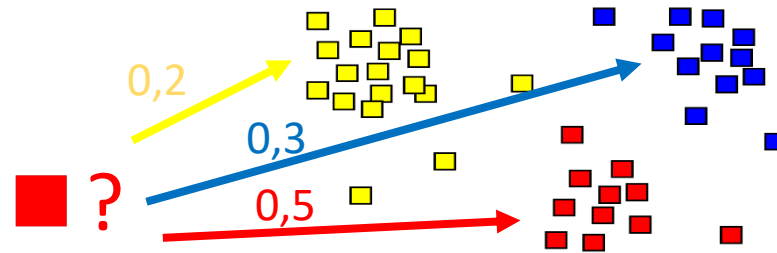
Discrección

Útil para categorizar a
partir de características
discretas



Bayes Naïve: idea

- Partimos de elementos con sus características divididos en grupos



- Tenemos un nuevo elemento con sus características y queremos averiguar a qué grupo pertenece
- Para esto calculamos la probabilidad de pertenencia a cada grupo



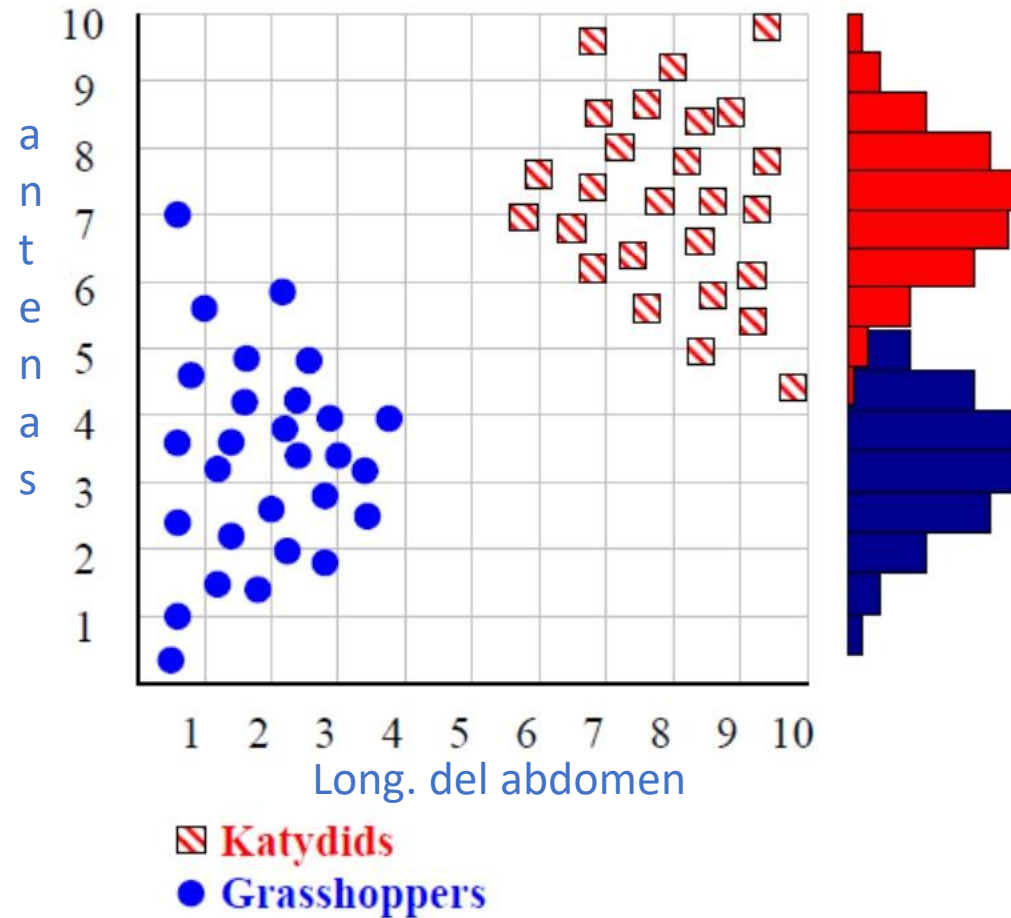
Ejemplo

Queremos distinguir entre
dos insectos similares:
saltamontes y **espezanzas**

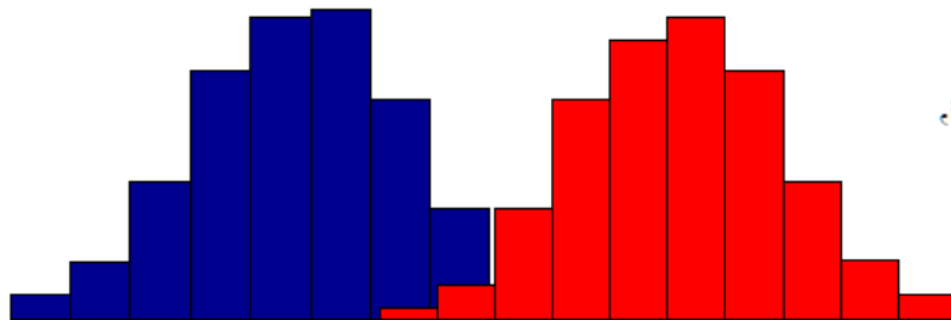
usamos la medida de sus
antenas y de su abdomen



Saltamontes y esperanzas



Longitud de las antenas: histograma

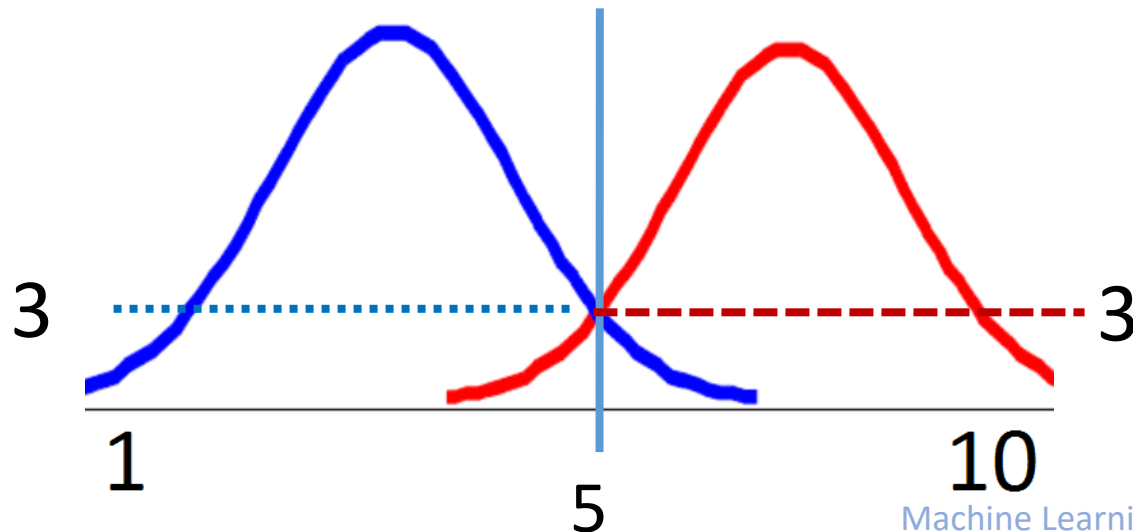


Queremos clasificar un insecto con longitud de antenas 5.
Tenemos 6 de cada tipo, 3 con antenas de longitud 5

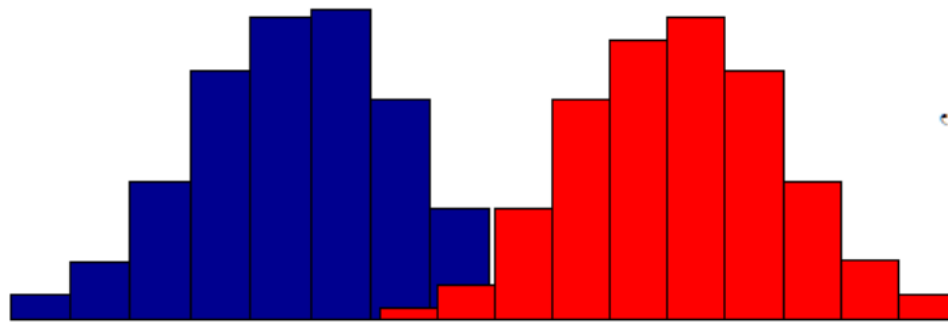
Probabilidad condicionada

$$P(\text{saltamontes} | 5) = 3 / (3+3) = 0,5$$

$$P(\text{esperanza} | 5) = 3 / (3+3) = 0,5$$



Longitud de las antenas: histograma

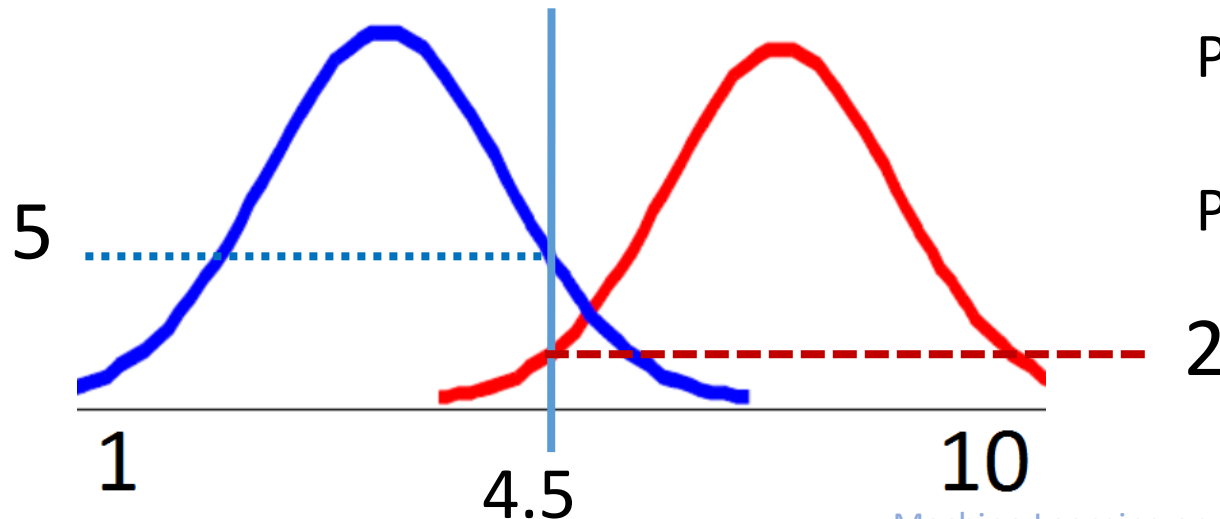


Queremos clasificar un insecto con longitud de antenas 4.5.
7 animalitos con antenas 4.5
5 saltamontes, 2 esperanzas

Probabilidad condicionada

$$P(\text{saltamontes} | 4.5) = 5 / (5+2) = 0,71$$

$$P(\text{esperanza} | 4.5) = 2 / (5+2) = 0,29$$



Longitud de antenas: ignorancia




Parece fácil esto de $P(\text{saltamontes} | 4.5) = 5 / (5+2) = 0,71$. Pero:

Busco una forma de distinguir saltamontes y esperanzas a partir de la longitud de las antenas **¡porque no sé distinguirlos!**

¿Cómo voy a saber que de los 7 insectos de mi muestra de 100 que tienen longitud de antenas 4.5, 2 son esperanzas y 5 saltamontes?



Longitud de antenas: ignorancia pero parcial

-  Bertoldo, un gran entomólogo tiene publicada una tabla en la que veo que entre 100 saltamontes es esperable encontrar 10 con antenas de 4.5 cm
 $P(4.5 \mid \text{saltamontes}) = 10/100 = 0.1$
-  En mi área en una muestra de 100 es esperable que 50 sean esperanzas y 50 saltamontes (fuente: CBS – Catálogo de Bichos Saltarines)
 $P(\text{saltamontes}) = 50/100 = 0.5$
-  Por mi parte tengo comprobado que de 100 bichejos 7 suelen tener antenas de 4.5
 $P(4.5) = 7/100 = 0.07$

$$\frac{P(4.5 \mid \text{saltamontes}) * P(\text{saltamontes})}{P(4.5)} = \frac{10/100 * 50/100}{7/100} = \frac{5}{7}$$

Bayes: teorema

$$P(\text{saltamontes} | 4.5) = \frac{P(4.5 | \text{saltamontes}) * P(\text{saltamontes})}{P(4.5)}$$

$$p(C_k | d) = \frac{p(C_k) \cdot p(d | C_k)}{p(d)}$$

$p(C_k | d)$ = probabilidad de que un objeto con las características d esté en el grupo $C_k \rightarrow$ lo que queremos averiguar

Bayes: teorema

$$P(\text{saltamontes} | 4.5) = \frac{P(4.5 | \text{saltamontes}) * P(\text{saltamontes})}{P(4.5)}$$

$$p(C_k | d) = \frac{p(C_k) \cdot p(d | C_k)}{p(d)}$$

$p(C_k | d)$ = probabilidad de que un objeto con las características d esté en el grupo $C_k \rightarrow$ lo que queremos averiguar

$p(C_k)$ = probabilidad de que un elemento esté en el grupo C_k

Bayes: teorema

$$P(\text{saltamontes} | 4.5) = \frac{P(4.5 | \text{saltamontes}) * P(\text{saltamontes})}{P(4.5)}$$

$$p(C_k | d) = \frac{p(C_k) \cdot p(d | C_k)}{p(d)}$$

$p(C_k | d)$ = probabilidad de que un objeto con las características d esté en el grupo $C_k \rightarrow$ lo que queremos averiguar

$p(C_k)$ = probabilidad de que un elemento esté en el grupo C_k

$p(d | C_k)$ = probabilidad de que un objeto del grupo C_k tenga las características d

Bayes: teorema

$$P(\text{saltamontes} | 4.5) = \frac{P(4.5 | \text{saltamontes}) * P(\text{saltamontes})}{P(4.5)}$$

$$p(C_k | d) = \frac{p(C_k) \cdot p(d | C_k)}{p(d)}$$

$p(C_k | d)$ = probabilidad de que un objeto con las características d esté en el grupo $C_k \rightarrow$ lo que queremos averiguar

$p(C_k)$ = probabilidad de que un elemento esté en el grupo C_k

$p(d | C_k)$ = probabilidad de que un objeto del grupo C_k tenga las características d

$p(d)$ = probabilidad de que un elemento esté en el grupo C_k

Bayes: teorema

$$P(\text{saltamontes} | 4.5) = \frac{P(4.5 | \text{saltamontes}) * P(\text{saltamontes})}{P(4.5)}$$

$$p(C_k | d) = \frac{p(C_k) \cdot p(d | C_k)}{p(d)}$$

$p(C_k | d)$ = probabilidad de que un objeto con las características d esté en el grupo $C_k \rightarrow$ lo que queremos averiguar

$p(C_k)$ = probabilidad de que un elemento esté en el grupo C_k

$p(d | C_k)$ = probabilidad de que un objeto del grupo C_k tenga las características d

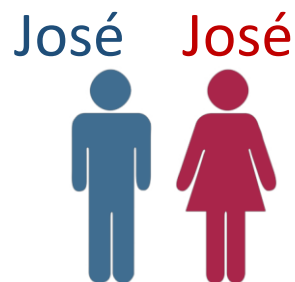
$p(d)$ = probabilidad de encontrar las características d en la muestra



Ejemplo

Tenemos una base de datos que incluye características de personas, pero no género, y queremos añadirla a partir del nombre.

Tenemos un conjunto de entrenamiento y hemos visto que hay nombres que corresponden a hombres y a mujeres.



Ejemplo: ¿Qué género debemos asignarle a “José”?

$$p(C_k|d) = \frac{p(C_k) \cdot p(d|C_k)}{p(d)}$$

Nombre	Género
José	Mujer
Claudia	Mujer
José	Hombre
José	Hombre
Alberto	Hombre
Natalia	Mujer
Nina	Mujer

Ejemplo: ¿Qué género debemos asignarle a “José”?

$$p(C_k|d) = \frac{p(C_k) \cdot p(d|C_k)}{p(d)}$$

$P(\text{Mujer}) = 4/7 = 0,57$ $p(\text{Hombre}) = 3/7 = 0,43$

$P(d) = P(\text{José}) = 3/7 = 0,43$

$P(\text{José} | \text{Hombre}) = 2/3 = 0,67$ $P(\text{José} | \text{Mujer}) = 0,25$

$P(\text{Hombre} | \text{José}) = 0,43 \times 0,67 / 0,43 = 0,67$

$P(\text{Mujer} | \text{José}) = 0,57 \times 0,25 / 0,43 = 0,33$

Nombre	Género
José	Mujer
Claudia	Mujer
José	Hombre
José	Hombre
Alberto	Hombre
Natalia	Mujer
Nina	Mujer

C1 = Hombre

C2 = Mujer

Bayes: Muchas características

A menudo tenemos más de una característica en d (d_1, d_2, d_3)
 d_1 =nombre, d_2 =altura, d_3 =color de ojos...la fórmula es la misma

$$p(C_k|d) = \frac{p(C_k) \cdot p(d|C_k)}{p(d)}$$

$$p(d|C_k) = p(d_1|C_k) \times \dots \times p(d_n|C_k)$$

¡Esto supone que los factores son independientes!



Ejemplo

Tenemos una serie de comentarios (sobre películas) clasificados como **positivos** o **negativos**.

Queremos usar las palabras de estos mensajes como conjuntos de entrenamiento para predecir el “sentimiento” de otros mensajes

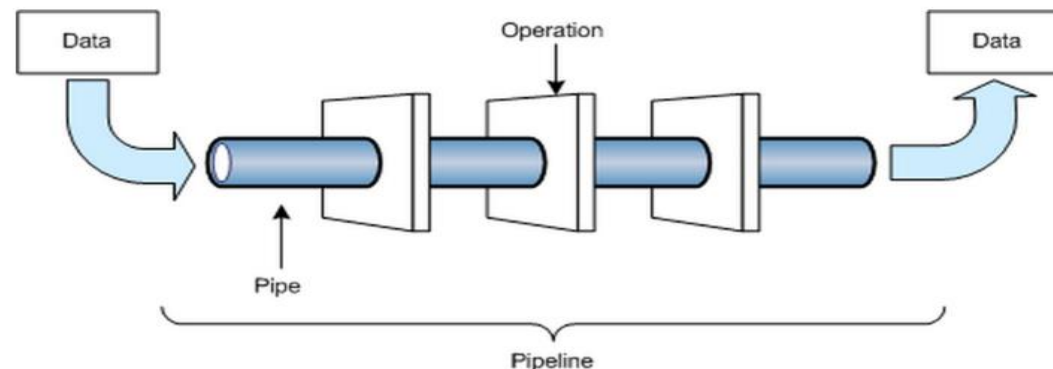
Ejemplo: análisis de sentimiento en MLlib

Idea:

Valoración	Awful	Good	Great	Bad	Best	Boring	...
0	X					X	
1		X					
0	X			X	X		
...

Tuberías o Pipelines

- ✓ En ML es normal que haya que encadenar varias operaciones
- ✓ Esto se puede hacer programando sin más
- ✓ Sin embargo vemos que hay un esquema común: la salida de un paso se obtiene con `transform()`, se lleva al paso siguiente, donde se aplica `transform()`....
- ✓ En Spark ML han creado el objeto `Pipeline` con este propósito
- ✓ Desde Spark 2.0 los pipelines se pueden grabar en disco y leer



Ejemplo: clientes: el fichero de datos

Formato

activo 1:3 2:4 3:5
constante 1:3 2:3 3:5
durmiente 1:1 2:1 3:1
activo 1:4 2:4 3:3
constante 1:3 2:3 3:2
....

Comenzamos por hacer una clase que represente cada línea del fichero

+ Spark ML lib

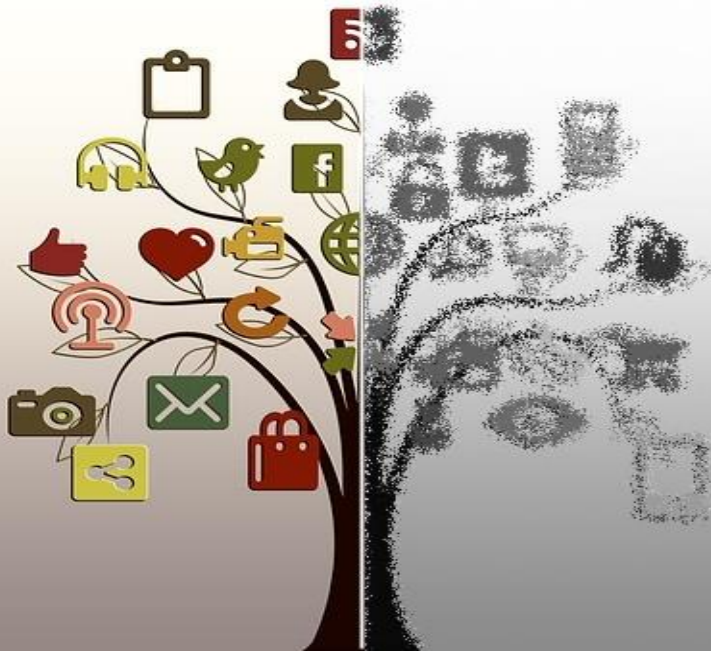
Aprendizaje supervisado

El conjunto de entrenamiento
contiene valores junto con su
resultado. El programa
entrenado debe ser capaz de
predecir nuevos valores

Modelos lineales: regresión

Naïve Bayes

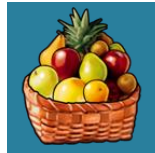
Árboles de decisión



Árboles de decisión

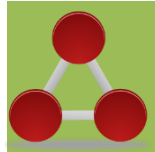
Discretos

Útiles para clasificar a partir
De características discretas



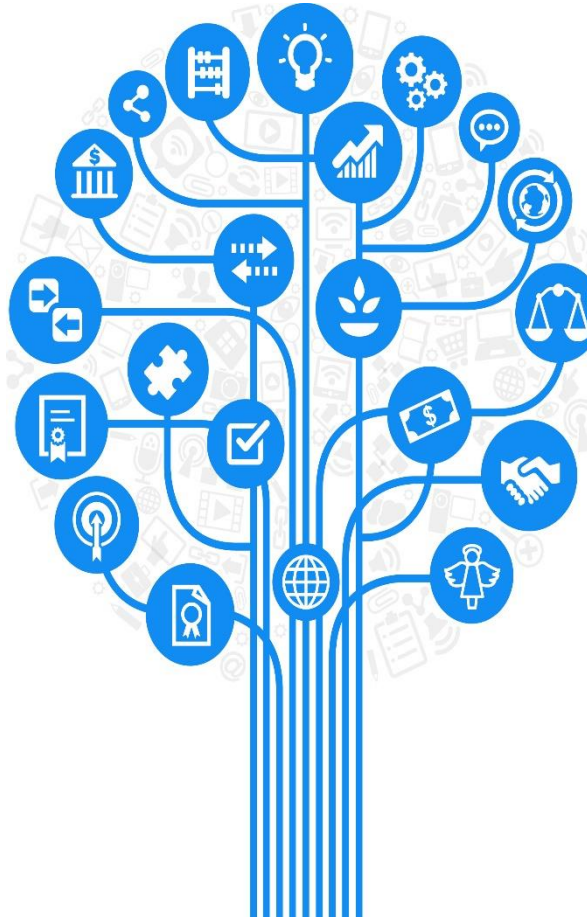
Nodos

Nodo → elección de un
atributo



Ramas

Selección de un valor para el
atributo elegido



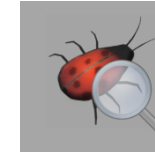
Hojas

Categoría a la que se asigna
el valor elegido



Errores

Errores ocasionales en el
conjunto de datos de entrada no
afectan al resultado



Valores desconocidos

Se admiten vectores de
entrada incompletos



¿Cómo construirlos?

Numerosos algoritmos. Casi todos variantes de ID3

- 1) Se elige para la raíz el atributo cuyos valores dividan la muestra lo mejor posible
 - 2) Se crea una rama para cada posible valor del atributo, y se reparte la muestra inicial según el valor asociado
 - 3) Se repite el proceso en cada hijo, teniendo en cuenta los atributos aún no empleados y el subconjunto de la muestra asociado
- El paso más delicado es el primero:

¿Cómo elegir el “mejor” atributo en cada nodo?

Entropía (2 características)

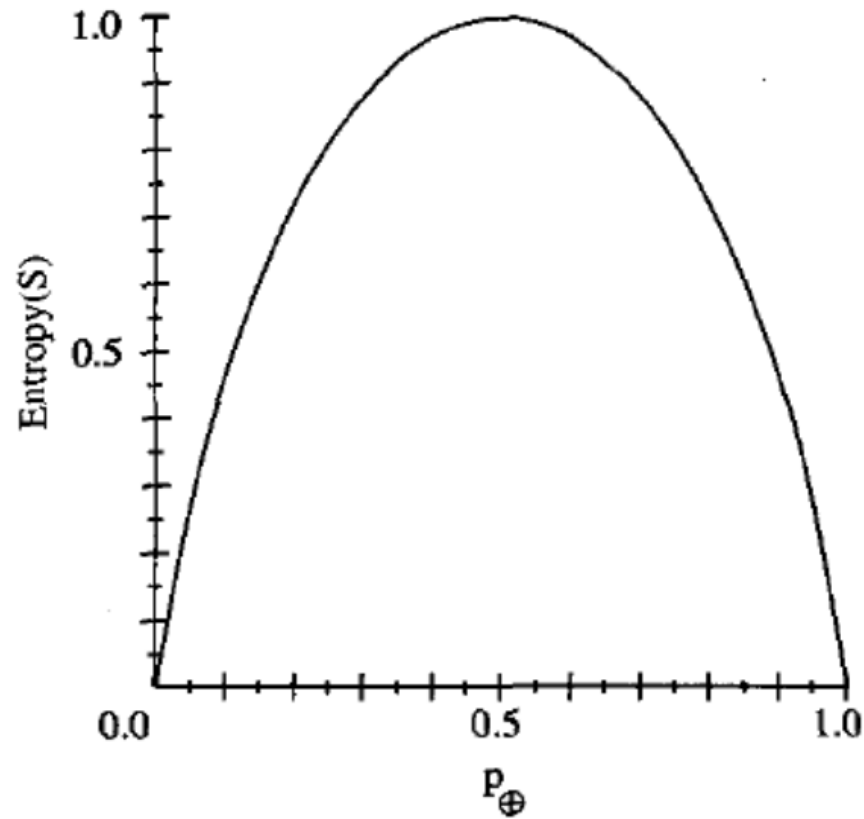
Indica el “orden” en el conjunto de entrada con respecto a la clasificación

- 1) La entropía es 0 si todos los elementos pertenecen a la misma clase
- 2) La entropía es 1 si están repartidos por igual entre todas las clases
- 3) En el caso de una clasificación binaria (Sí/No, 1/0) la entropía del conjunto de entrada S se define

$$\text{Entropía}(S) = -p_{\oplus} \log_2(p_{\oplus}) - p_{\ominus} \log_2(p_{\ominus})$$

- ✓ P_{\oplus} es la proporción de casos positivos en la muestra
- ✓ P_{\ominus} es la proporción de casos negativos

Entropía



Cuanto más equilibradas las probabilidades mayor es la entropía



Más trabajo tendremos que hacer para separar los dos grupos



Ejemplo

Tenemos dos muestras de 100 clientes

Muestra A: 8 morosos y 92 no

Muestra B: 14 morosos y 86 no

¿Qué muestra tiene mayor entropía?

$$E(A) = -0.08 \times \log_2(0.08) - 0.92 \times \log_2(0.92) = 0,40$$

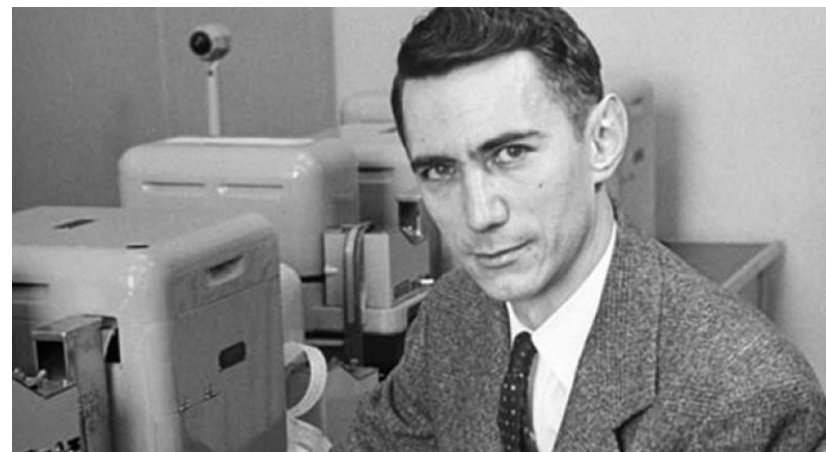
$$E(B) = -0.14 \times \log_2(0.14) - 0.86 \times \log_2(0.86) = 0,58$$



Entropía (n características)

La fórmula general para n características con probabilidades $p_1, p_2 \dots p_n$

$$E(S) = \sum_{i=1}^{i=n} -p_i \cdot \log_2 p_i$$



Claude E. Shannon en su artículo [“A Mathematical Theory of Communication”](#) (1948)

Ganancia de información

- 1) Buscamos un atributo/característica que divida la muestra en grupos de forma que la disminución de entropía sea máxima
- 2) En lugar de disminución de entropía hablamos de “ganancia de información”
- 3) Para un atributo A se define la ganancia de información sobre la muestra S:

$$G(S, A) = E(S) - \sum_{v \in \text{valores}(A)} \frac{|S_v|}{|S|} E(S_v)$$



Ejemplo

Tenemos una muestra **S** de 100 clientes

Muestra **S**: **8** morosos y **92** no

Hemos realizado un cuestionario tipo si/no.
Nos fijamos en 2 preguntas, P1 y P2

P1 Sí 50, incluyendo **7** morosos, No 50 (**1**)

P2 Sí 10 (**5** morosos), No 90 (**2**)

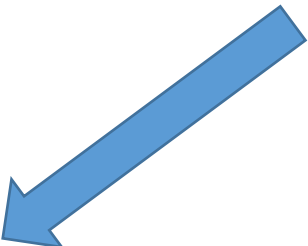
¿Cuál de las dos preguntas
seleccionaríamos primero para detectar
morosidad?

Ejemplo: detección rápida de morosos

$$G(S, A) = E(S) - \sum_{v \in \text{valores}(A)} \frac{|S_v|}{|S|} E(S_v)$$

$$E(S) = -0.08 \times \log_2(0.08) - 0.92 \times \log_2(0.92) = 0,40, |S| = 100$$

$$P1: |S_v| / |S| = 50/100 = 0,5 \quad E(S_{sí})$$



	P1	P2
$ S_{sí} / S $	0,50	0,10
$E(S_{sí})$	0,58	1
$ S_{no} / S $	0,50	0,90
$E(S_{no})$	0,14	0,14
$G(S,A)$	$0,40 - 0,36 = 0,04$	$0,40 - 0,23 = 0,17$

Árboles de decisión

- 1) Se selecciona el atributo A que proporciona mayor ganancia $G(S,A)$
- 2) Se reparte la población por las ramas (una por cada valor del atributo)
- 3) Se repite el proceso para cada nueva población que no tenga entropía 0

$$G(S, A) = E(S) - \sum_{v \in \text{valores}(A)} \frac{|S_v|}{|S|} E(S_v)$$

Árboles de decision en Spark ML



Discreto a continuo

Se permiten categorías continuas



Eficiencia

Se construyen rápidamente árboles a partir de conjuntos de hasta miles de millones de filas



No linealidad, dependencia

Al contrario que Bayes, no se pide independencia entre características



Agrupaciones

Spark ML incluye bosques aleatorios y árboles con incremento de gradiente





Ejemplo

En una empresa se hace un cuestionario a los clientes al año de empezar a serlo. También se hace una clasificación de clientes al tercer año:

Activo: Tiene contratos de mantenimiento, sigue interesándose por nuevos productos, participa en los actos de la empresa, etc.

Constante: No participa, pero tiene contratos de mantenimiento de varios productos

Durmiente: No se tiene apenas contacto, ni contratos de mantenimiento

¿Se podría usar el cuestionario para predecir el comportamiento de los clientes en el futuro?

Ejemplo: clientes: el fichero de datos

Formato

activo 1:3 2:4 3:5

constante 1:3 2:3 3:5

durmiente 1:1 2:1 3:1

activo 1:4 2:4 3:3

constante 1:3 2:3 3:2

....

Árboles de decisión: problemas

- ✓ **Complejos**: difíciles de construir y en casos complejos lentos
- ✓ **Cortos de vista**: Solo se examina una característica cada vez (problema “cajas rectangulares”)
- ✓ **No regresión**; solo clasificación de valores discretos
- ✓ **Overfitting**: conduce a complejidad innecesaria y a peores clasificaciones.
Soluciones
 - ✓ Poner límite a la poda, o bien por niveles o bien por tamaño de las muestras
 - ✓ Utilizar Random Forest

+ Spark ML lib

Aprendizaje supervisado

El conjunto de entrenamiento
contiene valores junto con su
resultado. El programa
entrenado debe ser capaz de
predecir nuevos valores

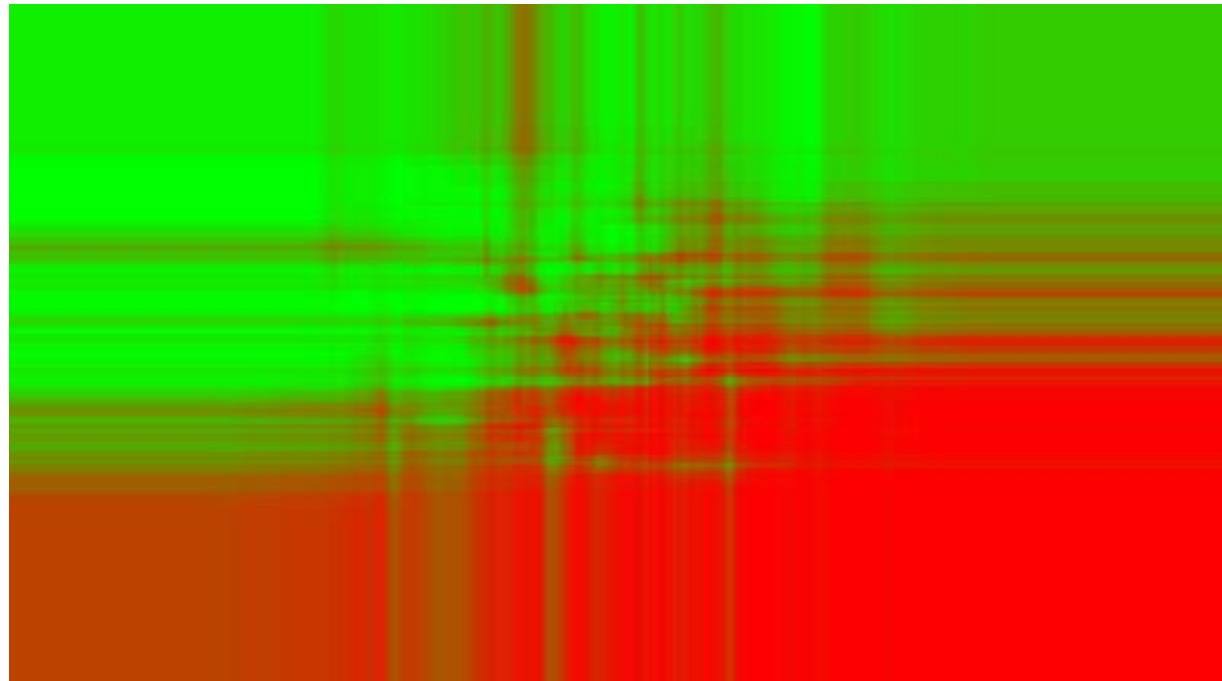
Modelos lineales: regresión

Excursión 1: lenguaje R

Naïve Bayes

Excursión 2: etiquetas no numéricas y pipelines

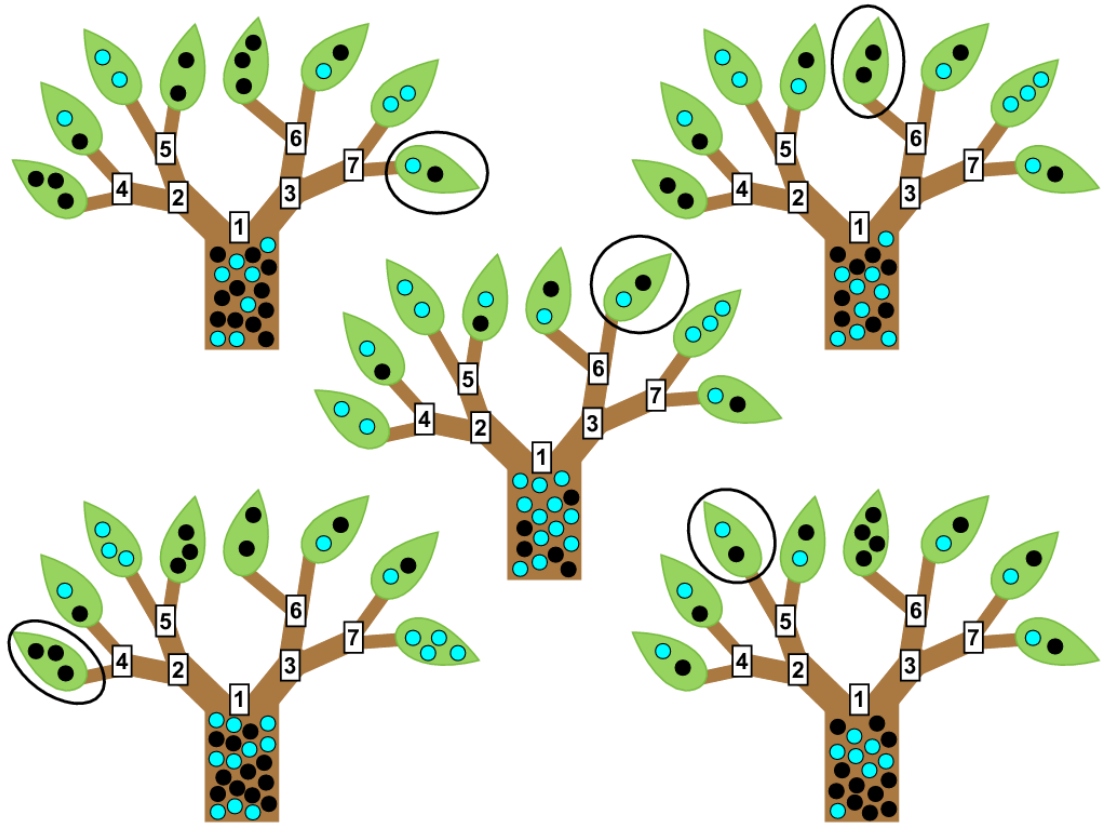
Árboles de decisión: random forest



Random Forest: ideas

1. Se parte del mismo conjunto de características m
 2. Se hace una preselección aleatoria de una cantidad $p < m$ de características
 3. Entre estas p etiquetas se elige la que mejor selecciona a la población
- ✓ La salida es un conjunto de árboles
 - ✓ A la hora de clasificar se prueba con todos
 - ✓ Se decide utilizando la “regla de la mayoría”: el valor más repetido en todas las hojas alcanzadas

Random Forest: ejemplo



¿Probabilidad de que se trate de un punto negro?