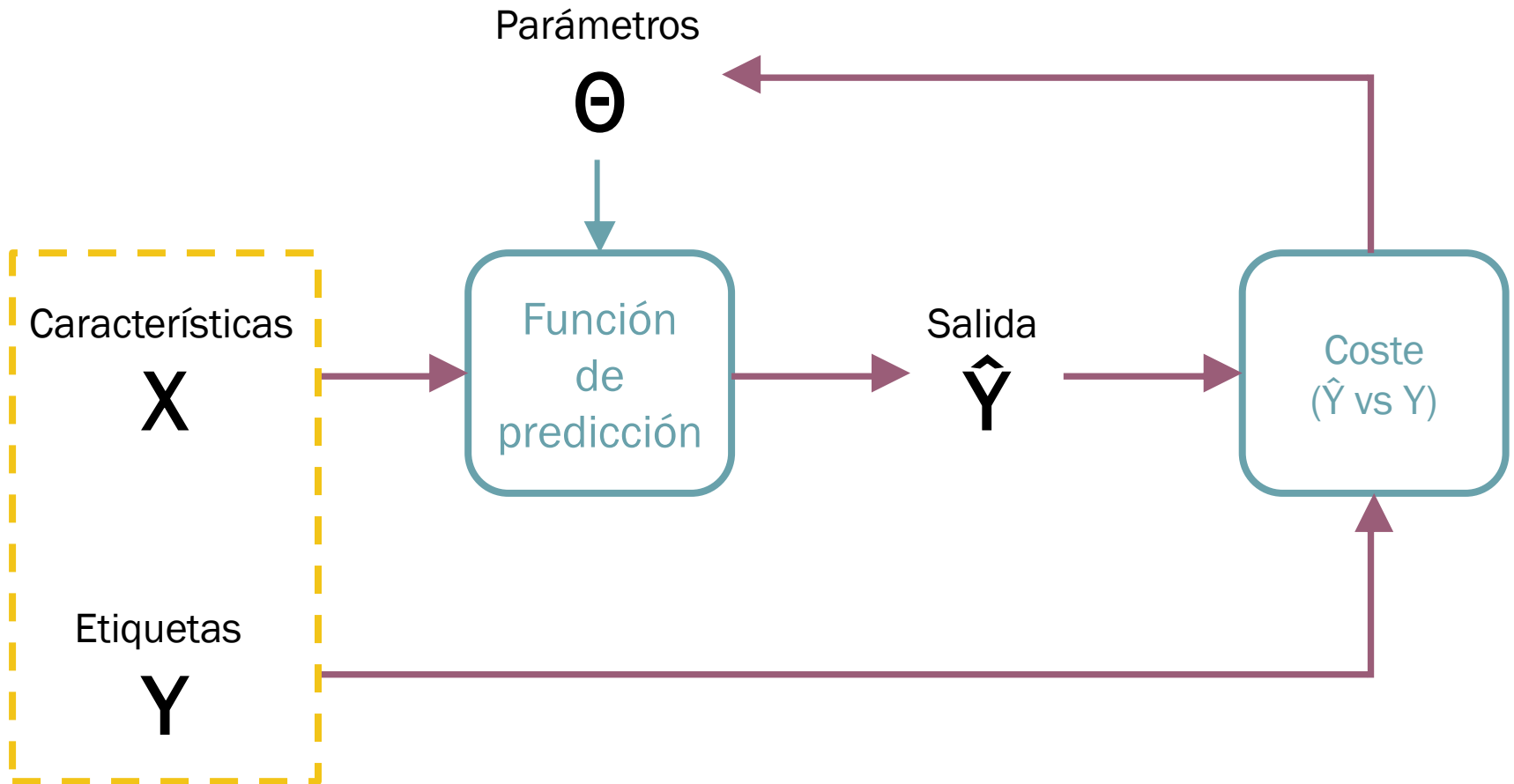


# ANÁLISIS DE SENTIMIENTOS UTILIZANDO APRENDIZAJE SUPERVISADO

Sistemas inteligentes. Aplicaciones

Aránzazu Jurío Munárriz

# Aprendizaje supervisado



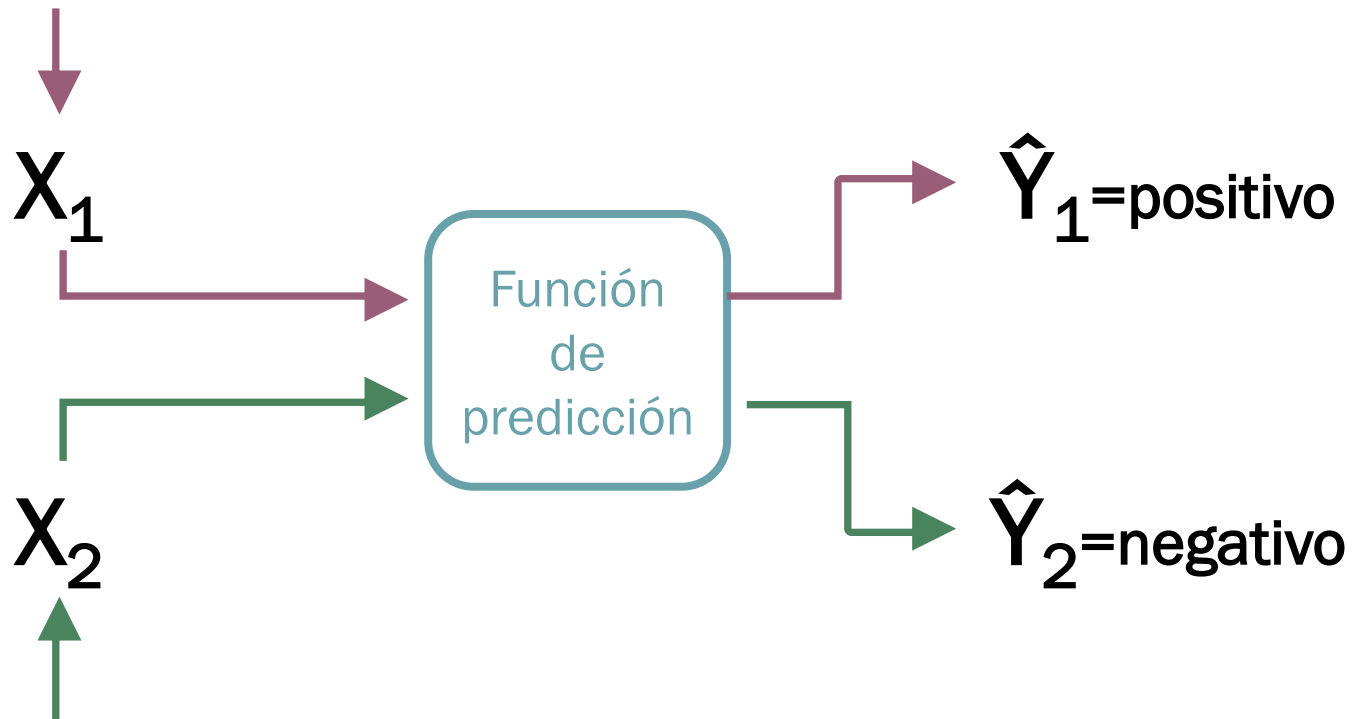
# Análisis de sentimientos



Mundos Mágicos @MagicMundi · 19h

...

Cada día es una lección, ¡siempre hay algo para aprender! ❤️



MARC @marcuantonioag1 · 31 ene.

...

LAS EXCUSAS EN EL CONQUIS ME DAN UNA PEREZA....QUE SI LOS  
GUANTES QUE SI LAS CHANCLETAS ...PUTA VIDA [#conquis3](#)

# Codificación del ejemplo

- One-hot encoding
  - *Para convertir variables categóricas en numéricas*

- Atributo: edad

- Posibles valores:

- *Infancia*
- *Juventud*
- *Madurez*
- *Senectud*

	Infancia	Juventud	Madurez	Senectud
Infancia	1	0	0	0
Juventud	0	1	0	0
Madurez	0	0	1	0
Senectud	0	0	0	1

# Codificación del ejemplo

- One-hot encoding para texto
  - *Tantas columnas como palabras existen en el diccionario*
    - Más de 93 000 palabras en el diccionario de la RAE
    - Matriz muy dispersa
    - Lento en aprendizaje y predicción

hoy feliz feliz →	a	aba	...	feliz	...	hoy	...	zuzón
	0	0	0	1	0	1	0	0
hoy feliz feliz →	a	aba	...	feliz	...	hoy	...	zuzón
	0	0	0	2	0	1	0	0

# Codificación del ejemplo

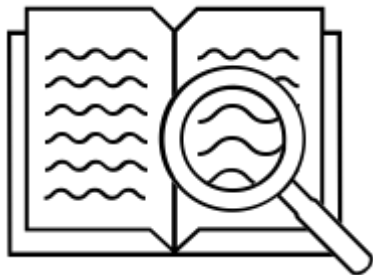
- One-hot encoding para texto
  - *Tantas columnas como palabras existen en el diccionario*



DB World ES | KameHameHa!!  
@DB\_WorldES

...

[#BuenasNoches](#) y [#FelizFinDeSemana](#) a todxs los fans de Dragon Ball



#BuenasNoches ✖  
y  
#FelizFinDeSemana ✖  
a  
todxs ✖

los  
fans ✖  
de  
Dragon ✖  
Ball ✖



y a los de

# Codificación del ejemplo

- Realizamos una codificación específica para el problema que estamos tratando
  - *A partir del corpus, medimos la frecuencia de cada palabra del vocabulario existente, tanto en la clase positiva como en la negativa*
- Corpus
  - *estoy feliz de aprender → POSITIVO*
  - *verte me hace feliz → POSITIVO*
  - *aburrida de tanto odio → NEGATIVO*
  - *me tienes aburrida → NEGATIVO*

# Codificación del ejemplo

- Realizamos una codificación específica para el problema que estamos tratando

- **Corpus**

- *estoy feliz de aprender → POSITIVO*
- *verte me hace feliz → POSITIVO*
- *aburrida de tanto odio → NEGATIVO*
- *me tienes aburrida → NEGATIVO*

	Frec. positiva	Frec. negativa
estoy	1	0
feliz	2	0
de	1	1
aprender	1	0
verte	1	0
me	1	1
hace	1	0
aburrida	0	2
tanto	0	1
odio	0	1
tienes	0	1



# Codificación del ejemplo

- Realizamos una codificación específica para el problema que estamos tratando
  - *Para codificar un ejemplo utilizamos dos características:*
    - *Suma de frecuencias positivas de sus palabras*
    - *Suma de frecuencias negativas de sus palabras*

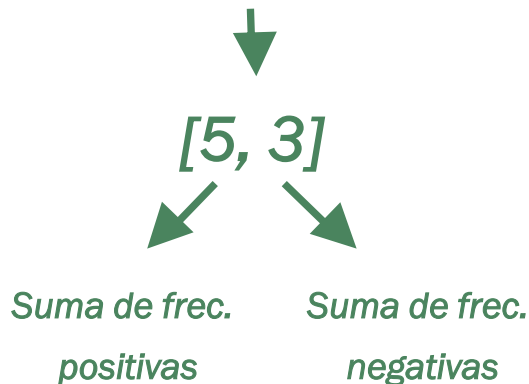
$$X_m = \left[ \sum_w \text{frec}(w, \text{positiva}), \quad \sum_w \text{frec}(w, \text{negativa}) \right]$$

# Codificación del ejemplo

- Realizamos una codificación específica para el problema que estamos tratando

- Nuevo ejemplo:

– *Me tienes feliz de verte*



	Frec. positiva	Frec. negativa
estoy	1	0
feliz	2	0
de	1	1
aprender	1	0
verte	1	0
me	1	1
hace	1	0
aburrida	0	2
tanto	0	1
odio	0	1
tienes	0	1

# Codificación del ejemplo

- Realizamos una codificación específica para el problema que estamos tratando
  - *Ya tenemos un dataset en formato tabular con tantas filas como ejemplos y 2 columnas*

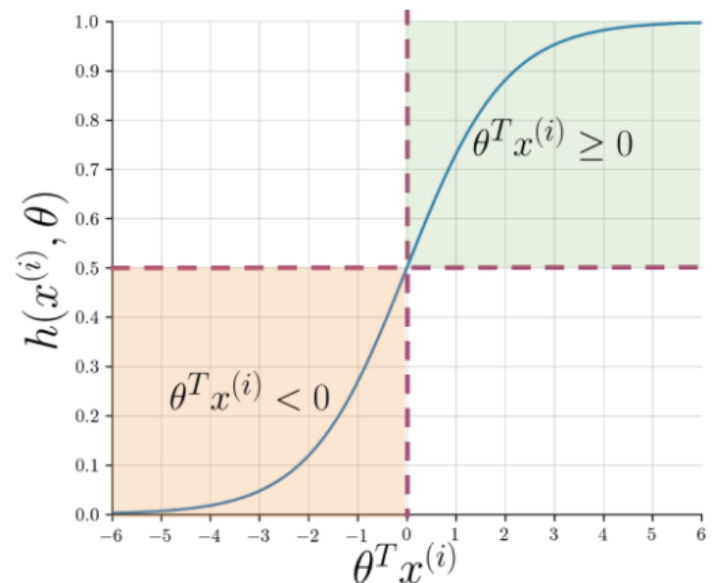
$$X = \begin{bmatrix} x_1^{(1)} & x_2^{(1)} \\ x_1^{(2)} & x_2^{(2)} \\ \dots & \dots \\ x_1^{(m)} & x_2^{(m)} \end{bmatrix}$$

# Regresión logística

- Sigmoide de la multiplicación de los pesos por los valores del ejemplo
  - Pesos:  $\theta$
  - Ejemplo:  $x^{(i)}$

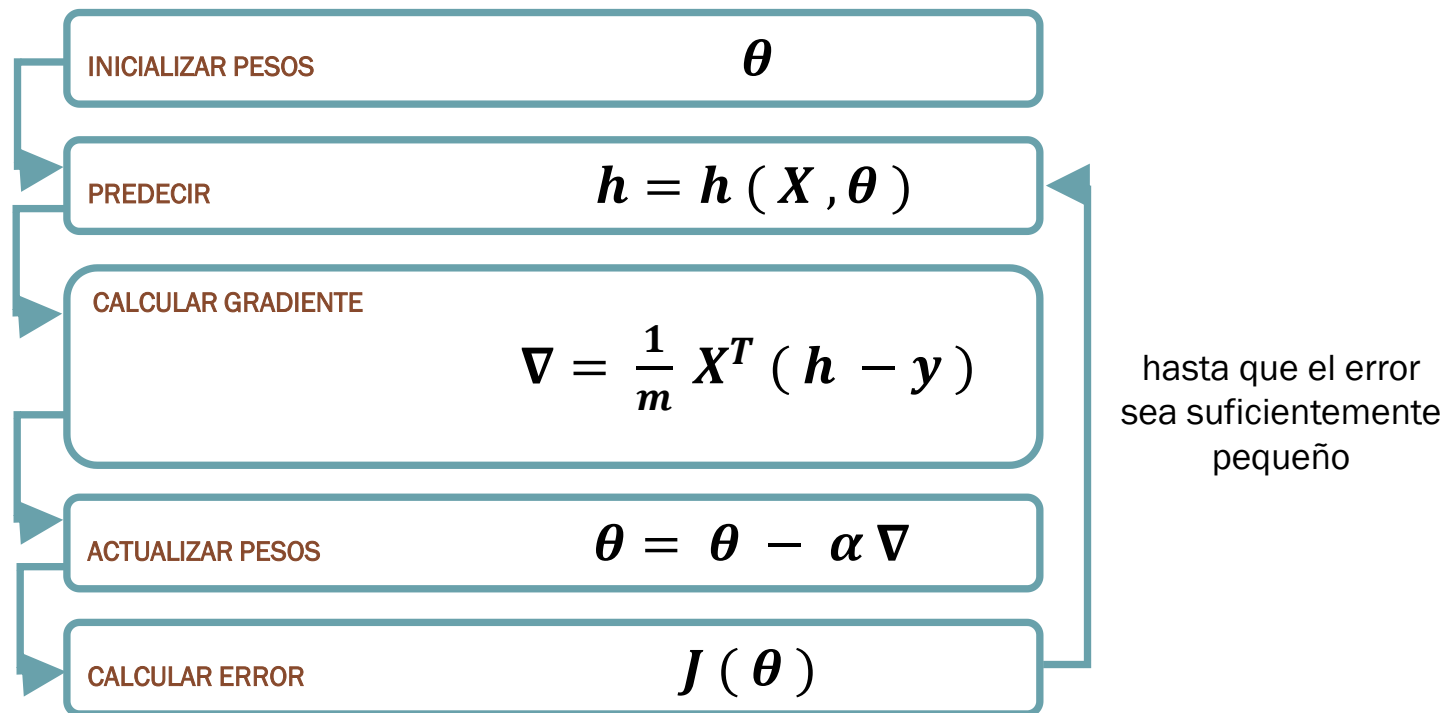
$$h(x^{(i)}, \theta) = \frac{1}{1 + e^{-\theta^T x^{(i)}}}$$

*\*Añadir columna de 1s a los ejemplos para el bias*



# Regresión logística

- ¿Cómo aprender los pesos de la regresión?
  - *Mediante descenso por gradiente*



# Regresión logística

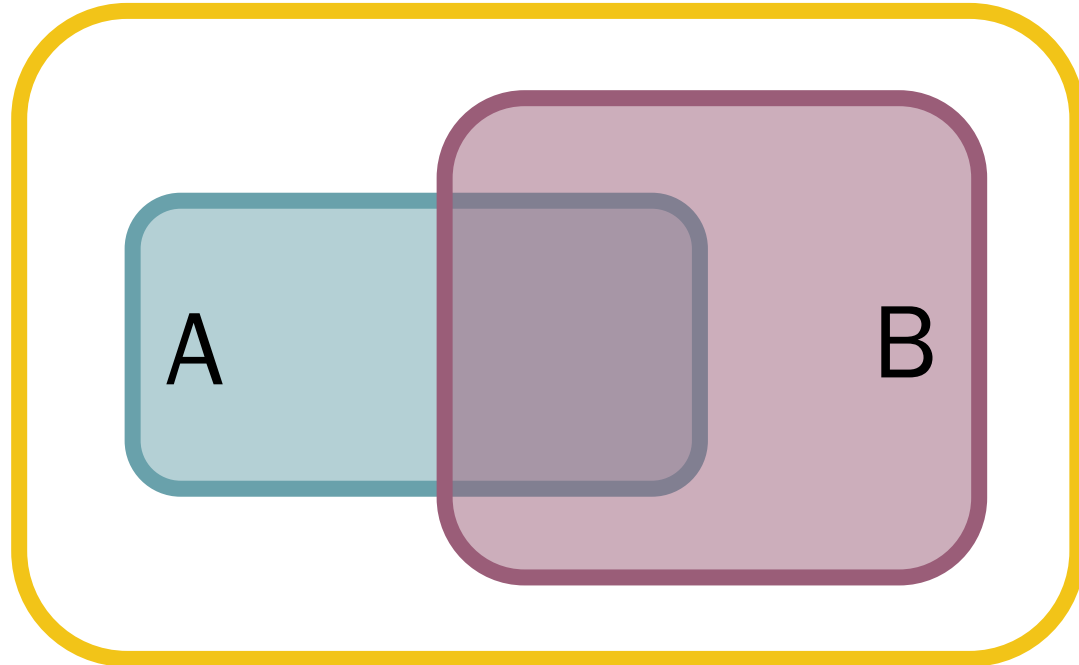
- ¿Cómo evaluar nuevos ejemplos?
  - *Aplicar la sigmoide a la multiplicación de los pesos y las características del ejemplo*
    - Si menor que 0,5 → clase negativa
    - Si mayor o igual que 0,5 → clase positiva
  - *Accuracy: ejemplos bien clasificados por uno*

$$acc = \sum_{i=1}^m \frac{(pred^{(i)} == y_{real}^{(i)})}{m}$$

# Naive Bayes

- Regla de Bayes

- ¿Es lo mismo calcular la probabilidad de A dado B que la probabilidad de B dado A?



# Naive Bayes

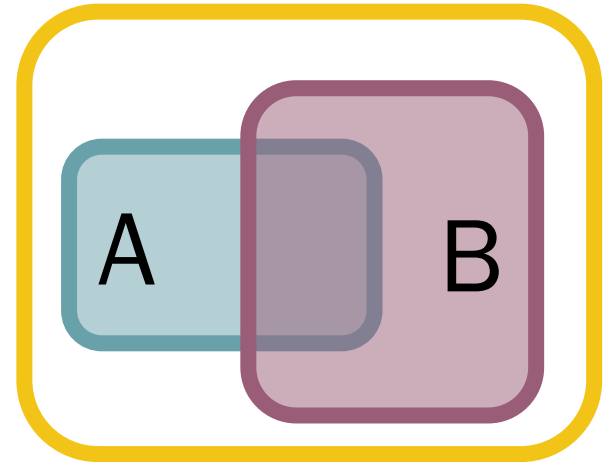
## ■ Regla de Bayes

- $P(A|B) = \frac{P(A \cap B)}{P(B)}$

- $P(B|A) = \frac{P(A \cap B)}{P(A)}$



- $P(A|B) = \frac{P(B|A) * P(A)}{P(B)}$





# Naive Bayes

- Clasificar un tweet

- *Analizar la probabilidad de que ese tweet sea positivo (o negativo)*

- Probabilidad de aparición en el corpus de cada una de las palabras del tweet

- Probabilidad de que si el tweet es positivo, aparezca una palabra concreta en él

- Probabilidad de que un tweet del corpus sea positivo

- $$P(pos|palabra) = \frac{P(palabra|pos)*P(pos)}{P(palabra)}$$

# Naive Bayes

- ¿Por qué se llama Naive – naif?
  - *Independencia de las características*
    - Todas las palabras son independientes
    - En el desierto del \_\_\_\_\_ hace calor
      - a) melón      b) bicicleta      c) Sahara
- Igualdad de aparición de todas las palabras en el corpus

# Naive Bayes

- Probabilidad de que si el tweet es positivo, aparezca una palabra concreta en él

- Corpus

- *estoy feliz de aprender* → POSITIVO
- *verte me hace feliz* → POSITIVO
- *aburrida de tanto odio* → NEGATIVO
- *me tienes aburrida* → NEGATIVO

	Frec. positiva	Frec. negativa
estoy	1	0
feliz	2	0
de	1	1
aprender	1	0
verte	1	0
me	1	1
hace	1	0
aburrida	0	2
tanto	0	1
odio	0	1
tienes	0	1

# Naive Bayes

- Probabilidad de que si el tweet es positivo, aparezca una palabra concreta en él
- Total palabras positivas: 8
- Total palabras negativas: 7


$$\begin{aligned} - P(\textit{palabra}|\textit{posit}) &= \frac{\textit{frec. positiva}}{\textit{total palabras posit.}} \\ - P(\textit{palabra}|\textit{negat}) &= \frac{\textit{frec. negativa}}{\textit{total palabras negat.}} \end{aligned}$$

	Frec. Positiva / P(palabra   posit)	Frec. Negativa / P(palabra   negat)
estoy	1 / 0,125	0 / 0
feliz	2 / 0,25	0 / 0
de	1 / 0,125	1 / 0,1429
aprender	1 / 0,125	0 / 0
verte	1 / 0,125	0 / 0
me	1 / 0,125	1 / 0,1429
hace	1 / 0,125	0 / 0
aburrida	0 / 0	2 / 0,2857
tanto	0 / 0	1 / 0,1429
odio	0 / 0	1 / 0,1429
tienes	0 / 0	1 / 0,1429

# Naive Bayes

- Probabilidad de que si el tweet es positivo, aparezca una palabra concreta en él

	P(palabra   posit)	P(palabra   negat)
estoy	0,125	0
feliz	0,25	0
de	0,125	0,1429
aprender	0,125	0
verte	0,125	0
me	0,125	0,1429
hace	0,125	0
aburrida	0	0,2857
tanto	0	0,1429
odio	0	0,1429
ienes	0	0,1429


$$\frac{0,125}{0,1429}$$

$$\frac{P(pal|posit)}{P(pal|negat)} = \begin{cases} < 1 & pal \text{ negativa} \\ \sim 1 & pal \text{ neutra} \\ > 1 & pal \text{ positiva} \end{cases}$$

# Naive Bayes

## ■ Inferencia

– *Clasificar: Me tienes feliz de verte*

$$- \frac{P(Me|pos)}{P(Me|neg)} * \frac{P(tienes|pos)}{P(tienes|neg)} * \frac{P(feliz|pos)}{P(feliz|neg)} * \frac{P(de|pos)}{P(de|neg)} * \frac{P(verte|pos)}{P(verte|neg)}$$

$$- \prod_{i=1}^m \frac{P(palabra_i | positiva)}{P(palabra_i | negativa)}$$

# Naive Bayes

## ■ Inferencia

– *Clasificar: Me tienes feliz de verte*

$$- \frac{P(Me|pos)}{P(Me|neg)} * \frac{P(tienes|pos)}{P(tienes|neg)} * \frac{P(feliz|pos)}{P(feliz|neg)} * \frac{P(de|pos)}{P(de|neg)} * \frac{P(verte|pos)}{P(verte|neg)}$$

	P(palabra   posit)	P(palabra   negat)
estoy	0,125	0
feliz	0,25	0
de	0,125	0,1429
aprender	0,125	0
verte	0,125	0
me	0,125	0,1429
hace	0,125	0
aburrida	0	0,2857
tanto	0	0,1429
odio	0	0,1429
tienes	0	0,1429

→ Inf (pos)

→ 0,87 (neutro)

→ Inf (pos)

→ 0,87 (neutro)

→ 0 (neg)

$$\text{inf} * 0,87 * \text{inf} * 0,87 * 0$$

# Naive Bayes

- Probabilidad de que si el tweet es positivo, aparezca una palabra concreta en él
  - *Suavizado laplaciano*: sumar 1 a cada frecuencia

	Frec. Positiva / P(palabra   posit)	Frec. Negativa / P(palabra   negat)
estoy	1 + 1	0 + 1
feliz	2 + 1	0 + 1
de	1 + 1	1 + 1
aprender	1 + 1	0 + 1
verte	1 + 1	0 + 1
me	1 + 1	1 + 1
hace	1 + 1	0 + 1
aburrida	0 + 1	2 + 1
tanto	0 + 1	1 + 1
odio	0 + 1	1 + 1
tienes	0 + 1	1 + 1

- Total palabras positivas: 19 (8 + 11)

- Total palabras negativas: 18 (7+11)

- |Vocabulario| = 11

- $$P(\text{palabra}|\text{posit}) = \frac{\text{frec. positiva} + 1}{\text{total palabras posit.} + |\text{Vocab}|}$$

- $$P(\text{palabra}|\text{negat}) = \frac{\text{frec. negativa} + 1}{\text{total palabras negat.} + |\text{Vocab}|}$$



# Naive Bayes

- Probabilidad de que si el tweet es positivo, aparezca una palabra concreta en él
  - *Sumar 1 a cada frecuencia*

	Frec. Positiva / P(palabra   posit)	Frec. Negativa / P(palabra   negat)
estoy	1 / 0,125	0 / 0
feliz	2 / 0,25	0 / 0
de	1 / 0,125	1 / 0,1429
aprender	1 / 0,125	0 / 0
verte	1 / 0,125	0 / 0
me	1 / 0,125	1 / 0,1429
hace	1 / 0,125	0 / 0
aburrida	0 / 0	2 / 0,2857
tanto	0 / 0	1 / 0,1429
odio	0 / 0	1 / 0,1429
tienes	0 / 0	1 / 0,1429

	Frec. Positiva / P(palabra   posit)	Frec. Negativa / P(palabra   negat)
estoy	2 / 0,1053	1 / 0,0556
feliz	3 / 0,1579	1 / 0,0556
de	2 / 0,1053	2 / 0,1111
aprender	2 / 0,1053	1 / 0,0556
verte	2 / 0,1053	1 / 0,0556
me	2 / 0,1053	2 / 0,1111
hace	2 / 0,1053	1 / 0,0556
aburrida	1 / 0,0526	3 / 0,1667
tanto	1 / 0,0526	2 / 0,1111
odio	1 / 0,0526	2 / 0,1111
tienes	1 / 0,0526	2 / 0,1111

# Naive Bayes

## ■ Inferencia

– *Clasificar: Me tienes feliz de verte*

$$- \frac{P(Me|pos)}{P(Me|neg)} * \frac{P(tienes|pos)}{P(tienes|neg)} * \frac{P(feliz|pos)}{P(feliz|neg)} * \frac{P(de|pos)}{P(de|neg)} * \frac{P(verte|pos)}{P(verte|neg)}$$

	P(palabra   posit)	P(palabra   negat)
estoy	0,1053	0,0556
feliz	0,1579	0,0556
de	0,1053	0,1111
aprender	0,1053	0,0556
verte	0,1053	0,0556
me	0,1053	0,1111
hace	0,1053	0,0556
aburrida	0,0526	0,1667
tanto	0,0526	0,1111
odio	0,0526	0,1111
tienes	0,0526	0,1111

→ 2,8399

→ 0,9478

→ 1,8939

→ 0,9478

$$\prod = 2,2873 \rightarrow \textit{positivo}$$

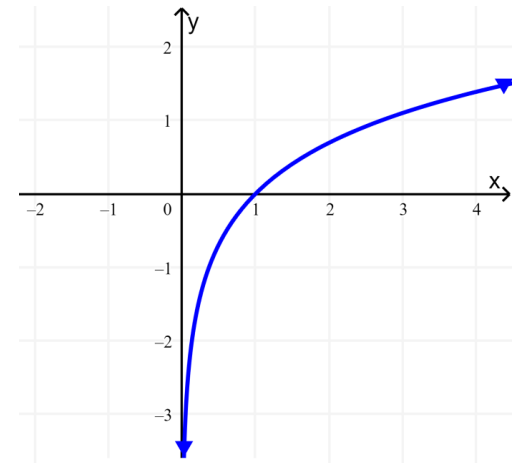
→ 0,4734

# Naive Bayes

- La inferencia de Naive Bayes consta de dos términos:
  - *likelihood*
    - multiplicación del ratio de las palabras a ser positivas
    - $\prod_{i=1}^m \frac{P(\text{palabra}_i | \text{positiva})}{P(\text{palabra}_i | \text{negativa})}$
  - *prior ratio*
    - Ratio de tweets positivos frente a negativos en el corpus
    - $\frac{P(\text{positivo})}{P(\text{negativo})} = \frac{\#doc\_pos / \#total\_doc}{\#doc\_neg / \#total\_doc} = \frac{\#dos\_pos}{\#doc\_neg}$
  - *Utilizamos la multiplicación de ambos términos*
    - *prior ratio* \* *likelihood*

# Naive Bayes

- Al trabajar con muchas multiplicaciones de números pequeños, podemos tener el problema del underflow
  - *Lo solucionamos cambiando a la escala logarítmica*
    - $\log(a * b) = \log(a) + \log(b)$
    - $\log(a/b) = \log(a) - \log(b)$



$$f(x) = \ln(x)$$

# Naive Bayes

- Log prior

- $prior = \frac{P(positivo)}{P(negativo)} = \frac{\#doc\_pos}{\#dos\_neg}$
- $logprior = \log(\#doc\_pos) - \log(\#doc\_neg)$

- Log likelihood

- $likelihood = \prod_{i=1}^m \frac{P(palabra_i | positiva)}{P(palabra_i | negativa)}$
- $loglikelihood = \sum_{i=1}^m \log \frac{P(palabra_i | positiva)}{P(palabra_i | negativa)}$

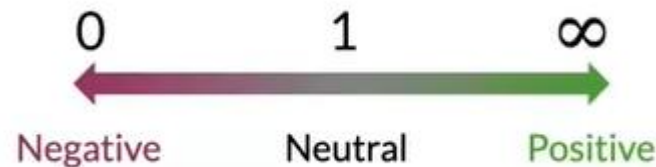
- Clasificación

- $logprior + loglikelihood$

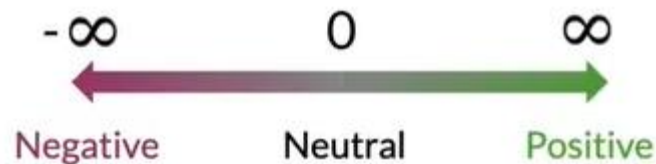
# Naive Bayes

- Log likelihood

- $likelihood = \prod_{i=1}^m \frac{P(palabra_i | positiva)}{P(palabra_i | negativa)}$



- $loglikelihood = \sum_{i=1}^m \log \frac{P(palabra_i | positiva)}{P(palabra_i | negativa)}$



# Naive Bayes

## ■ Inferencia

- $\lambda$  es el logaritmo del ratio de probabilidades (loglikelihood)
- *Clasificar: Me tienes feliz de verte*

	P(pal   pos)	P(pal   neg)	$\lambda$
estoy	0,1053	0,0556	0,6386
feliz	0,1579	0,0556	1,0438
de	0,1053	0,1111	-0,0536
aprender	0,1053	0,0556	0,6386
verte	0,1053	0,0556	0,6386
me	0,1053	0,1111	-0,0536
hace	0,1053	0,0556	0,6386
aburrida	0,0526	0,1667	-1,1535
tanto	0,0526	0,1111	-0,7477
odio	0,0526	0,1111	-0,7477
tienes	0,0526	0,1111	-0,7477

- $\#doc\_pos = 2$
- $\#doc\_neg = 2$
- $logprior = \log(2) - \log(2) = 0$
- $loglikelihood = -0,0536 - 0,7477 + 1,0438 - 0,0536 + 0,6386 = 0,8275$
- $total = 0 + 0,8275 = 0,8275$

# Naive Bayes

## ENTRENAMIENTO

CALCULAR FRECUENCIAS POR CLASE

CALCULAR PROBABILIDAD DE PALABRA EN CLASE

$$P(\text{pal}|\text{clase}) = \frac{\text{frec}(\text{pal}, \text{clase}) + 1}{\text{total\_pal\_clase} + |V|}$$

CALCULAR  $\lambda$  (LOGLIKELIHOOD DE CADA PALABRA)

$$\lambda(\text{pal}) = \log \frac{P(\text{pal}|\text{pos})}{P(\text{pal}|\text{neg})}$$

CALCULAR LOGPRIOR

$$\text{logprior} = \log \frac{\#doc\_pos}{\#doc\_neg}$$

## TEST

SUMAR EL LOGLIKELIHOOD DE LAS PALABRAS DEL TWEET QUE EXISTEN EN EL VOACUARIO

$$\text{loglikelihood} = \sum_{i=1}^m \lambda_i$$

SUMAR LOGPRIOR

$$\text{total} = \text{logprior} + \text{loglikelihood}$$

CLASIFICAR

si  $\text{total} > 0$     **positivo**  
si  $\text{total} < 0$     **negativo**