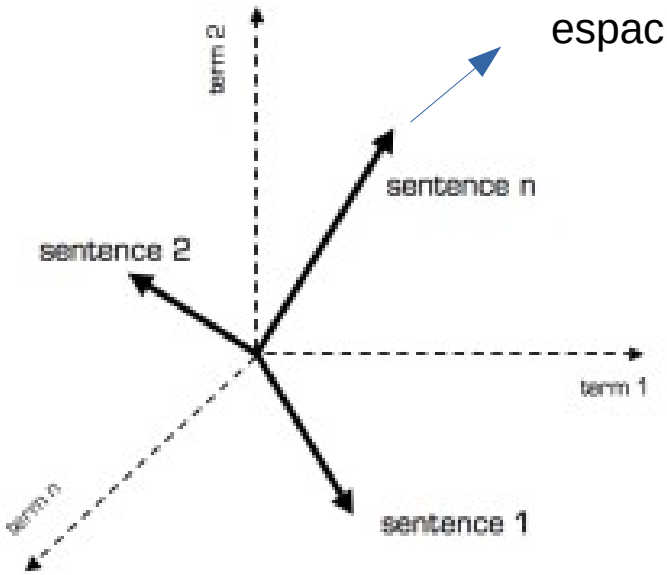
IIC3670 Procesamiento de Lenguaje Natural

https://github.com/marcelomendoza/IIC3670

- CLASIFICACIÓN DE DOCUMENTOS -

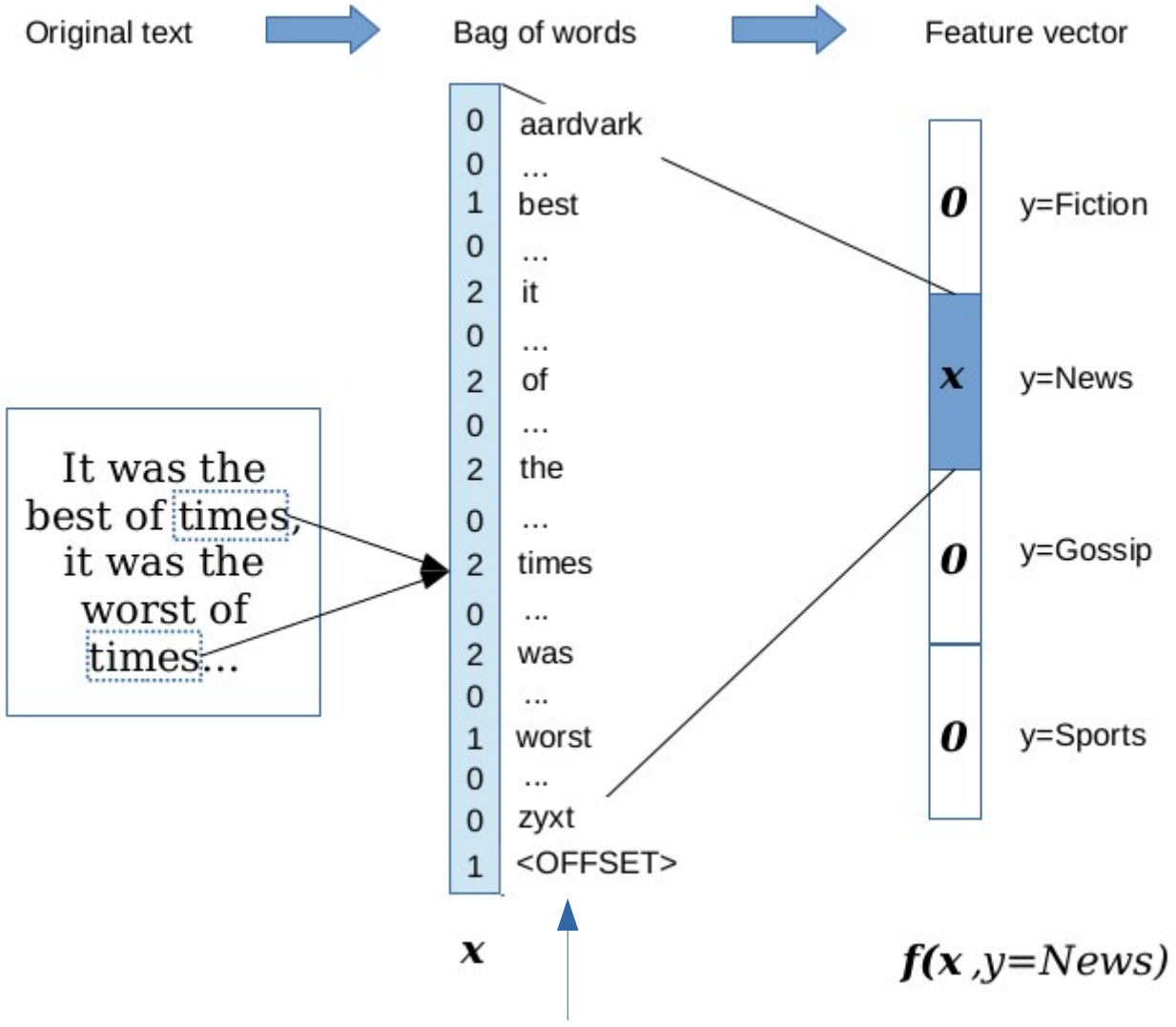
Vector-space model



Cada documento es un vector en el espacio vectorial de palabras

Cada palabra es una dimensión en el espacio vectorial de representación

BOW


Original text
Bag of words
Feature vector
It was the best of times, it was the worst of times...
0 aardvark
0 ...
1 best
0 ...
2 it
0 ...
2 of
0 ...
2 the
0 ...
2 times
0 ...
2 was
0 ...
1 worst
0 ...
0 zyxt
1 <OFFSET>
0 y=Fiction
x y=News
0 y=Gossip
0 y=Sports
x
f(x ,y=News)
vocabulario

$f_{i,j}$: # occs. de ti en dj

$\max f_{l,j}$

N : # docs

n_i : # docs donde ti ocurre

Term scoring functions:

- Tf: $$Tf_{i,j} = \frac{f_{i,j}}{\max f_{l,j}}$$

- Tf corregido: $$w_{i,j} = \begin{cases} 1 + \log_{10} f_{i,j} & \text{if } f_{i,j} > 0 \\ 0 & \text{e.t.o.c.} \end{cases}$$

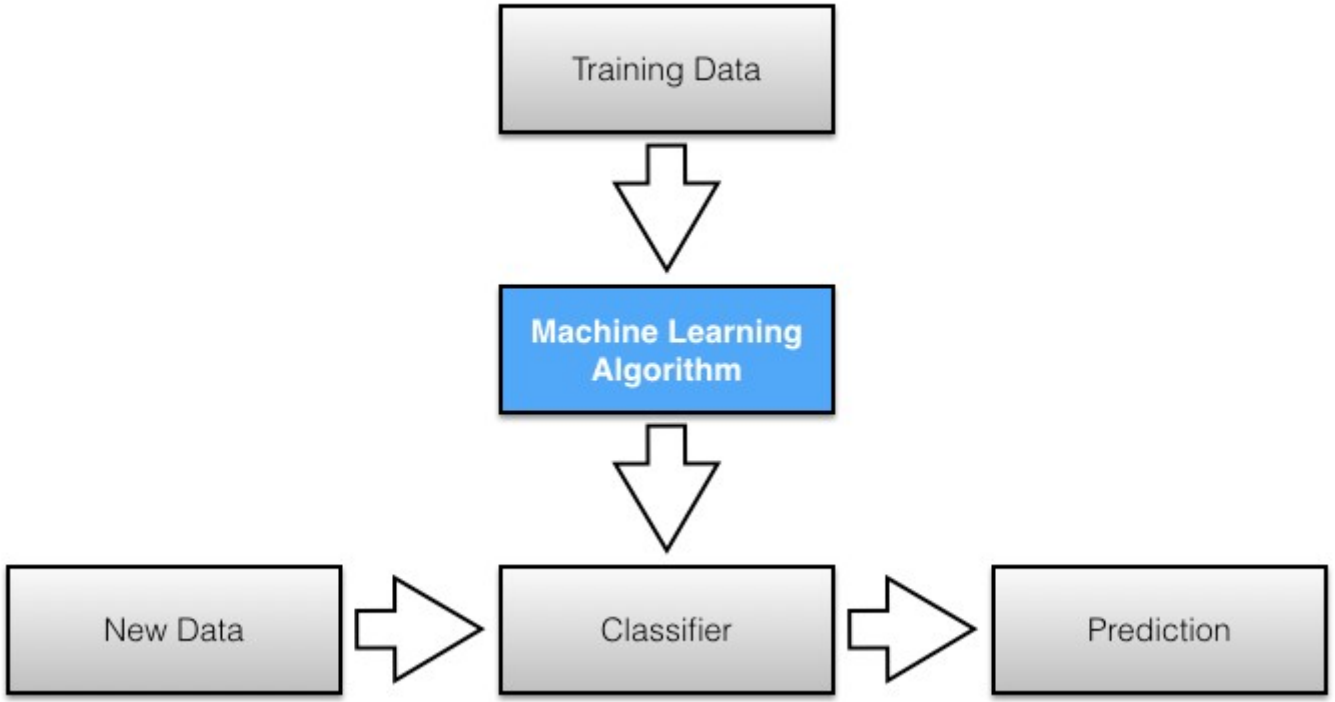
- Idf: $$\text{idf}_{ti} = \log_{10} \frac{N}{n_i}$$

- Tf-Idf (Salton): $$w_{i,j} = (1 + \log f_{l,j}) \cdot \log \frac{N}{n_i}$$

- Tf-Idf: $$w_{i,j} = \frac{f_{i,j}}{\max f_{l,j}} \cdot \log \frac{N}{n_i}$$

Clasificación de documentos

Síntesis. El enfoque de NLP (clásico)



Naive Bayes

Training (MLE)

$$\begin{aligned}\hat{\boldsymbol{\theta}} &= \operatorname*{argmax}_{\boldsymbol{\theta}} \mathrm{p}(\boldsymbol{x}^{(1:N)}, y^{(1:N)}; \boldsymbol{\theta}) \\ &= \operatorname*{argmax}_{\boldsymbol{\theta}} \prod_{i=1}^{N} \mathrm{p}(\boldsymbol{x}^{(i)}, y^{(i)}; \boldsymbol{\theta}) \\ &= \operatorname*{argmax}_{\boldsymbol{\theta}} \sum_{i=1}^{N} \log \mathrm{p}(\boldsymbol{x}^{(i)}, y^{(i)}; \boldsymbol{\theta}).\end{aligned}$$

Naive Bayes

Training (MLE)

$$\begin{aligned}\hat{\boldsymbol{\theta}} &= \operatorname*{argmax}_{\boldsymbol{\theta}} \mathrm{p}(\boldsymbol{x}^{(1:N)}, y^{(1:N)}; \boldsymbol{\theta}) \\ &= \operatorname*{argmax}_{\boldsymbol{\theta}} \prod_{i=1}^{N} \mathrm{p}(\boldsymbol{x}^{(i)}, y^{(i)}; \boldsymbol{\theta}) \\ &= \operatorname*{argmax}_{\boldsymbol{\theta}} \sum_{i=1}^{N} \log \mathrm{p}(\boldsymbol{x}^{(i)}, y^{(i)}; \boldsymbol{\theta}).\end{aligned}$$

Generative process:

Algorithm 1 Generative process for the Naïve Bayes classification model

for Instance $i \in \{1, 2, \ldots, N\}$ **do**:
  Draw the label $y^{(i)} \sim \text{Categorical}(\boldsymbol{\mu})$;
  Draw the word counts $\boldsymbol{x}^{(i)} \mid y^{(i)} \sim \text{Multinomial}(\boldsymbol{\phi}_{y^{(i)}})$.

Naive Bayes

Condicionado a y

$$\mathrm{p}_{\mathrm{mult}}(\boldsymbol{x};\boldsymbol{\phi}) = B(\boldsymbol{x}) \prod_{j=1}^{V} \phi_j^{x_j}$$

$$B(\boldsymbol{x}) = \frac{\left(\sum_{j=1}^{V} x_j\right)!}{\prod_{j=1}^{V}(x_j!)}.$$

¿Por qué es naive? Porque asume independencia entre los atributos.

Naive Bayes

Condicionado a y

$$\mathrm{p}_{\text{mult}}(\boldsymbol{x}; \boldsymbol{\phi}) = B(\boldsymbol{x}) \prod_{j=1}^{V} \phi_j^{x_j}$$

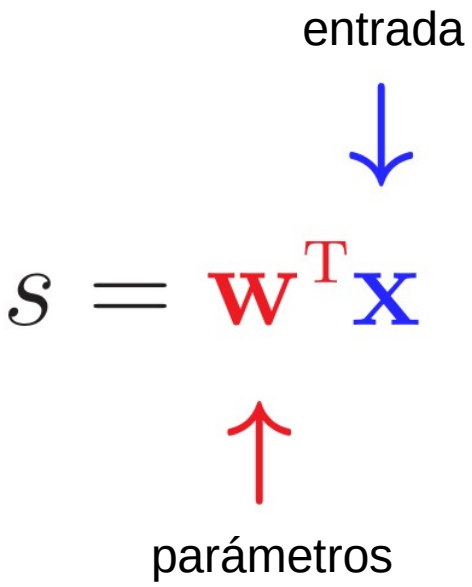
$$B(\boldsymbol{x}) = \frac{\left(\sum_{j=1}^{V} x_j\right)!}{\prod_{j=1}^{V} (x_j!)}.$$

¿Por qué es naive? Porque asume independencia entre los atributos.

Predicción:

$$\begin{aligned}\hat{y} &= \operatorname*{argmax}_y \log \mathrm{p}(\boldsymbol{x}, y; \boldsymbol{\mu}, \boldsymbol{\phi}) \\ &= \operatorname*{argmax}_y \log \mathrm{p}(\boldsymbol{x} \mid y; \boldsymbol{\phi}) + \log \mathrm{p}(y; \boldsymbol{\mu})\end{aligned}$$

Modelos lineales



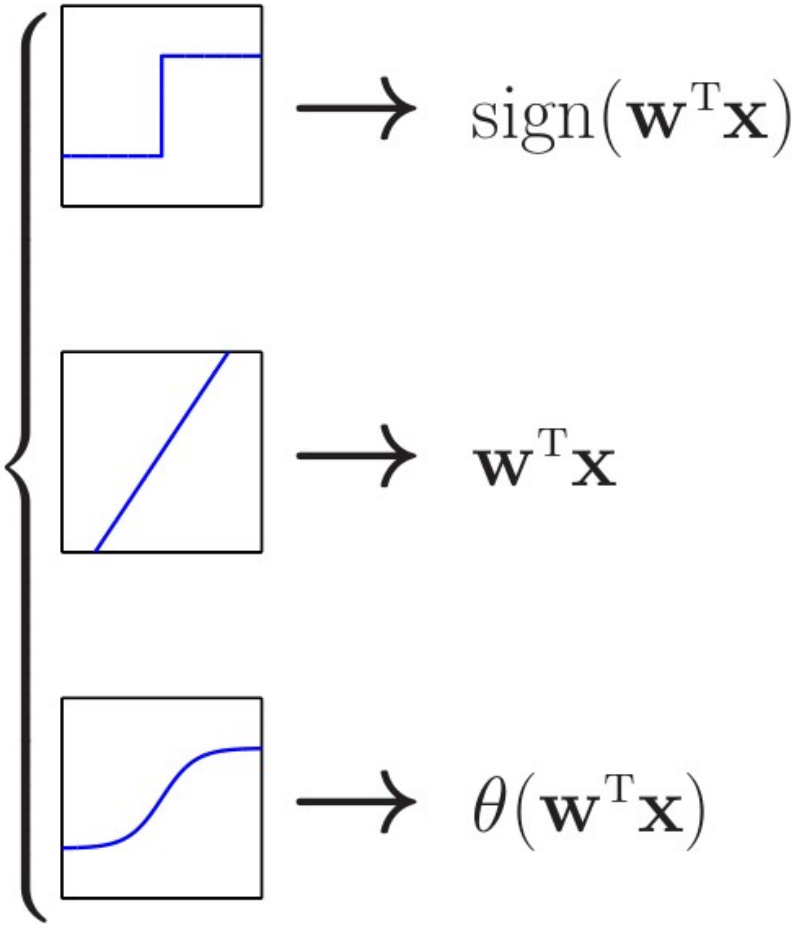
Modelos lineales

entrada

$s = \mathbf{w}^{\mathrm{T}}\mathbf{x}$

parámetros

$$y = \theta(s)$$

$s = \mathbf{w}^{\mathrm{T}}\mathbf{x}$ 

$\rightarrow \ \mathrm{sign}(\mathbf{w}^{\mathrm{T}}\mathbf{x}) \qquad \{-1, +1\}$

$\rightarrow \ \mathbf{w}^{\mathrm{T}}\mathbf{x} \qquad \mathbb{R}$

$\rightarrow \ \theta(\mathbf{w}^{\mathrm{T}}\mathbf{x}) \qquad [0, 1]$

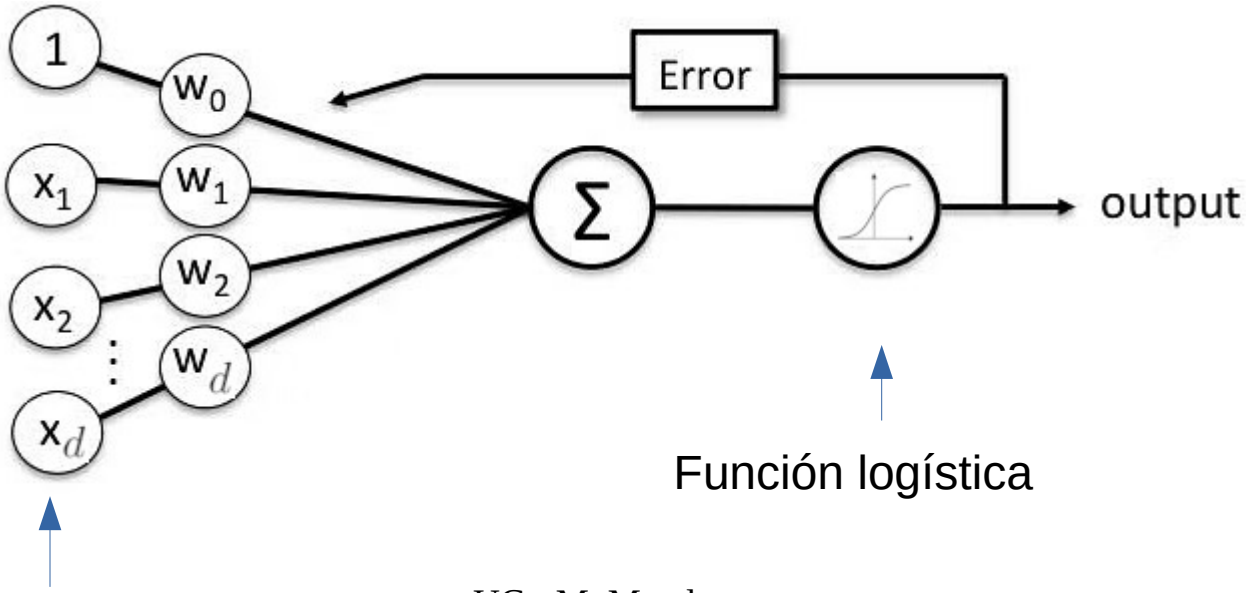
Regresión logística

Objetivo:

$$f(\mathbf{x}) = \mathbb{P}[y = +1 \mid \mathbf{x}].$$

Modelo:

$$h(\mathbf{x}) = \theta\left(\sum_{i=0}^{d} w_i x_i\right) = \theta(\mathbf{w}^{\mathrm{T}}\mathbf{x})$$



Función logística

Vector de características

Regresión logística

Objetivo:

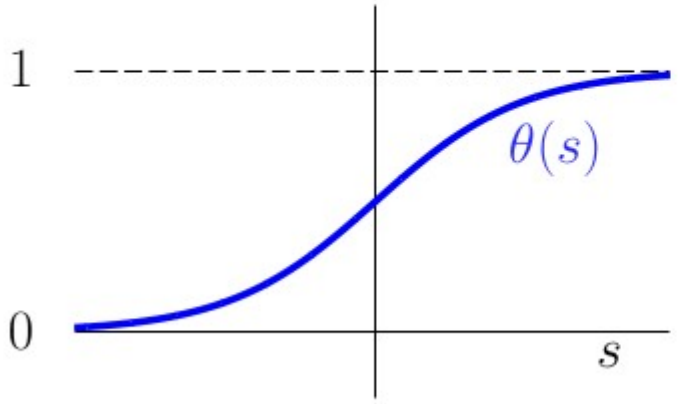
$$f(\mathbf{x}) = \mathbb{P}[y = +1 \mid \mathbf{x}].$$

Modelo:

$$h(\mathbf{x}) = \theta\left(\sum_{i=0}^{d} w_i x_i\right) = \theta(\mathbf{w}^{\mathrm{T}}\mathbf{x})$$

Función logística:

$$y \in [0, 1]$$



$$\theta(s) = \frac{e^s}{1+e^s} = \frac{1}{1+e^{-s}}.$$

$$\theta(-s) = \frac{e^{-s}}{1+e^{-s}} = \frac{1}{1+e^{s}} = 1 - \theta(s).$$

Regresión logística

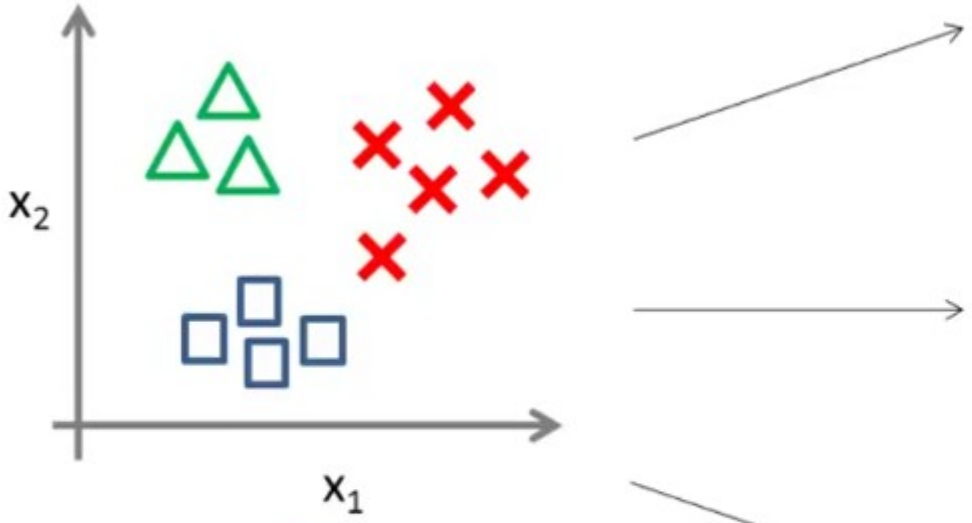
Notar que: $\mathcal{D} = (\mathbf{x}_1, y_1 = \pm 1), \cdots, (\mathbf{x}_N, y_N = \pm 1)$

Un buen modelo logra lo siguiente:

$$\begin{cases} h(\mathbf{x}_n) \approx 1 & \text{si } \; y_n = +1; \\ h(\mathbf{x}_n) \approx 0 & \text{si } \; y_n = -1. \end{cases}$$

Clasificación multiclase

One-vs-all (one-vs-rest):

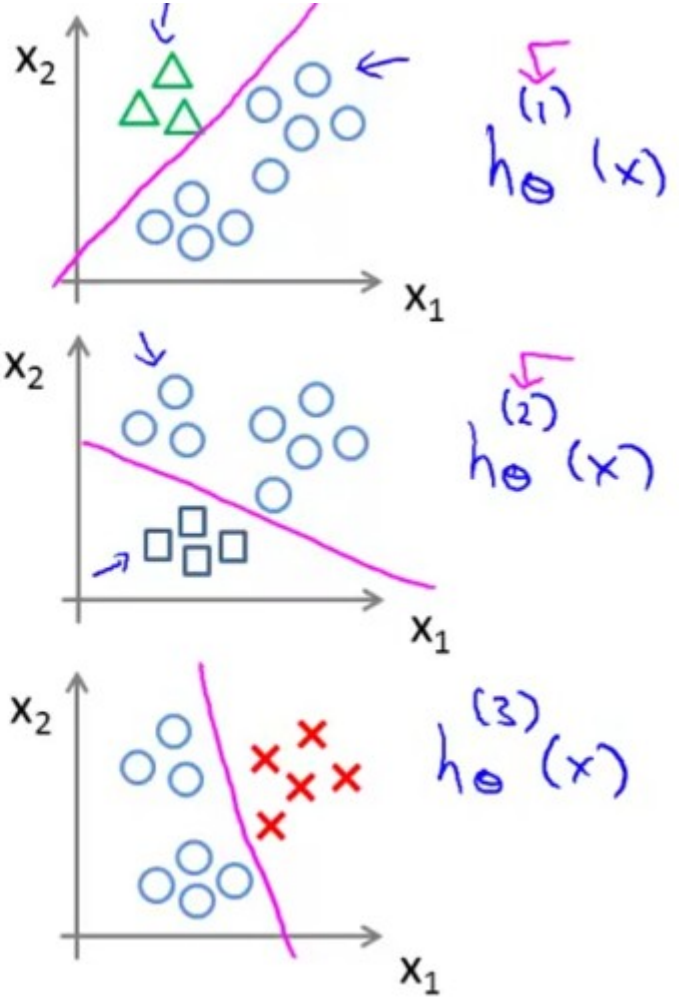


Class 1: △ ←

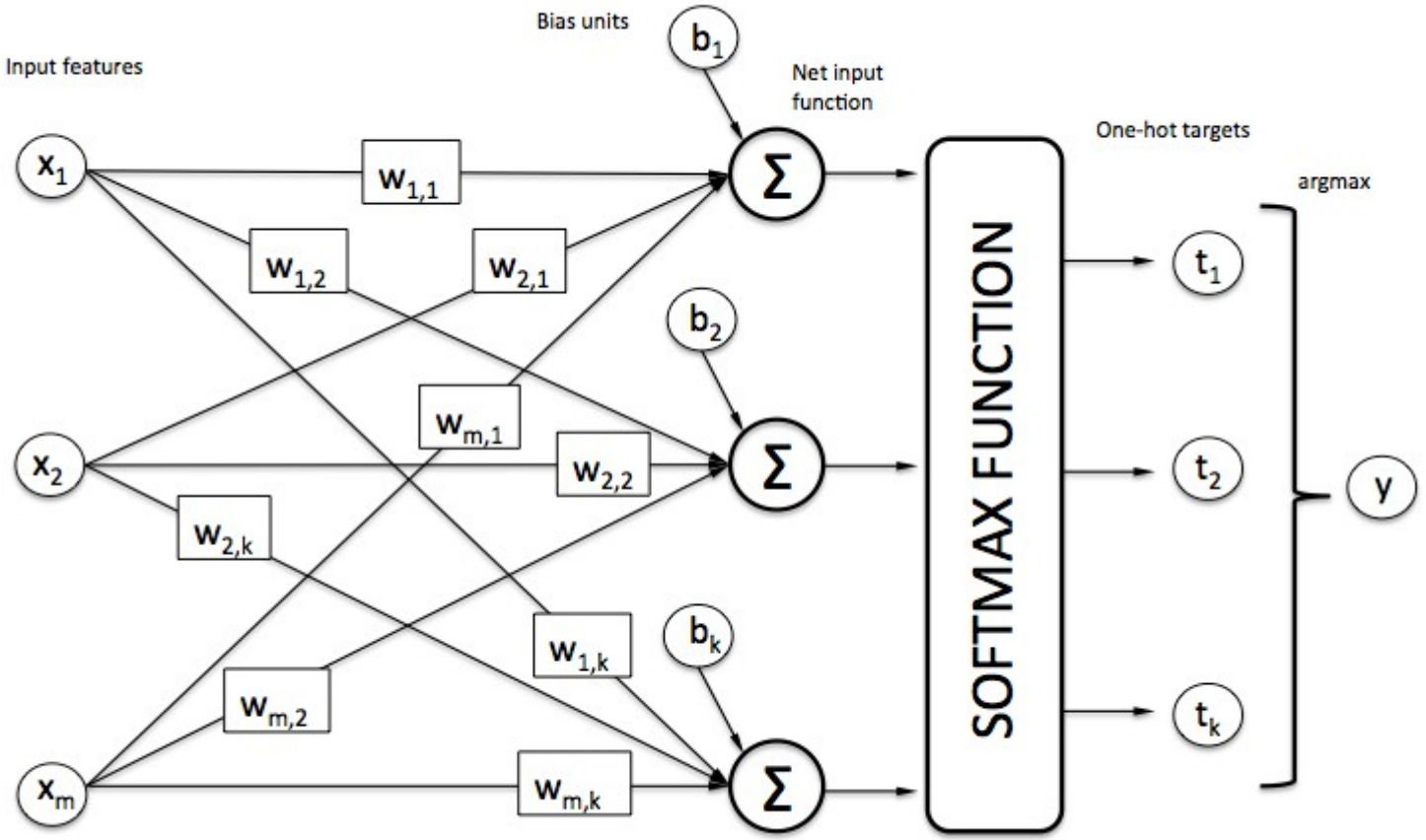
Class 2: □ ←

Class 3: ✕ ←

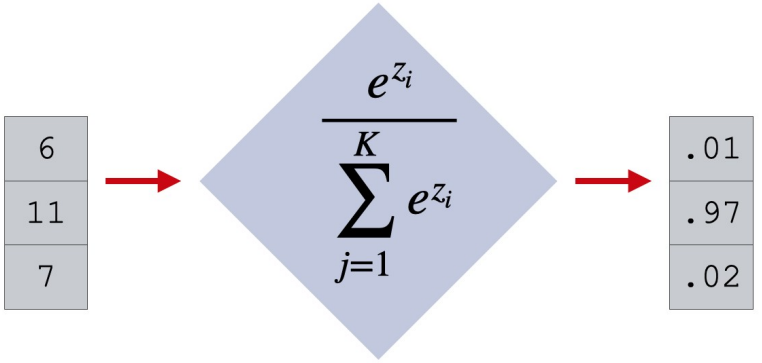
$h_\theta^{(i)}(x) = P(y = i|x;\theta) \qquad (i = 1, 2, 3)$



Clasificación multiclase

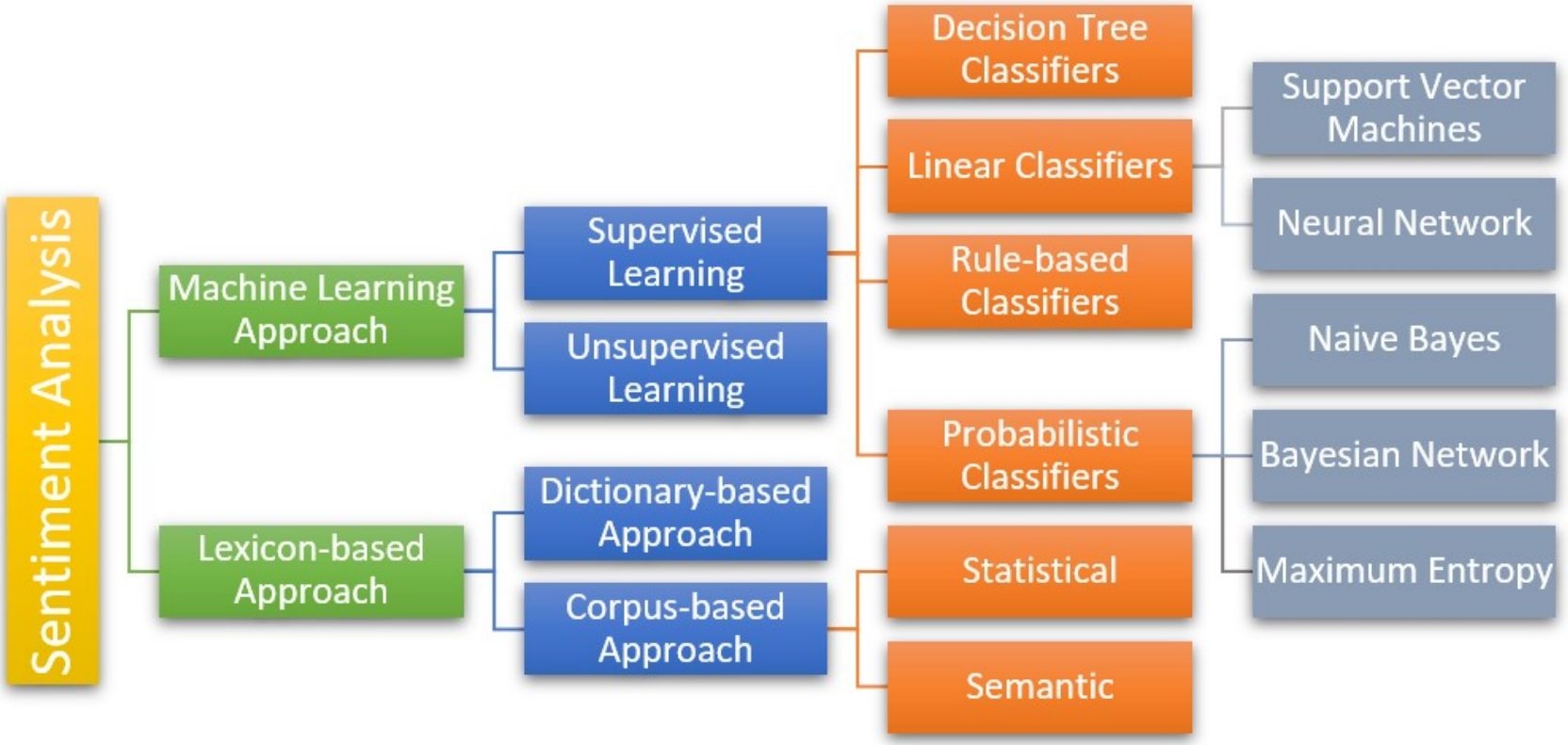


$$P(y = j \mid z^{(i)}) = \phi_{softmax}(z^{(i)})$$



- SENTIMENT ANALYSIS -

Sentiment analysis



Sentiment analysis (lexicon-based approach)

- Affective Norms for English Words (ANEW)

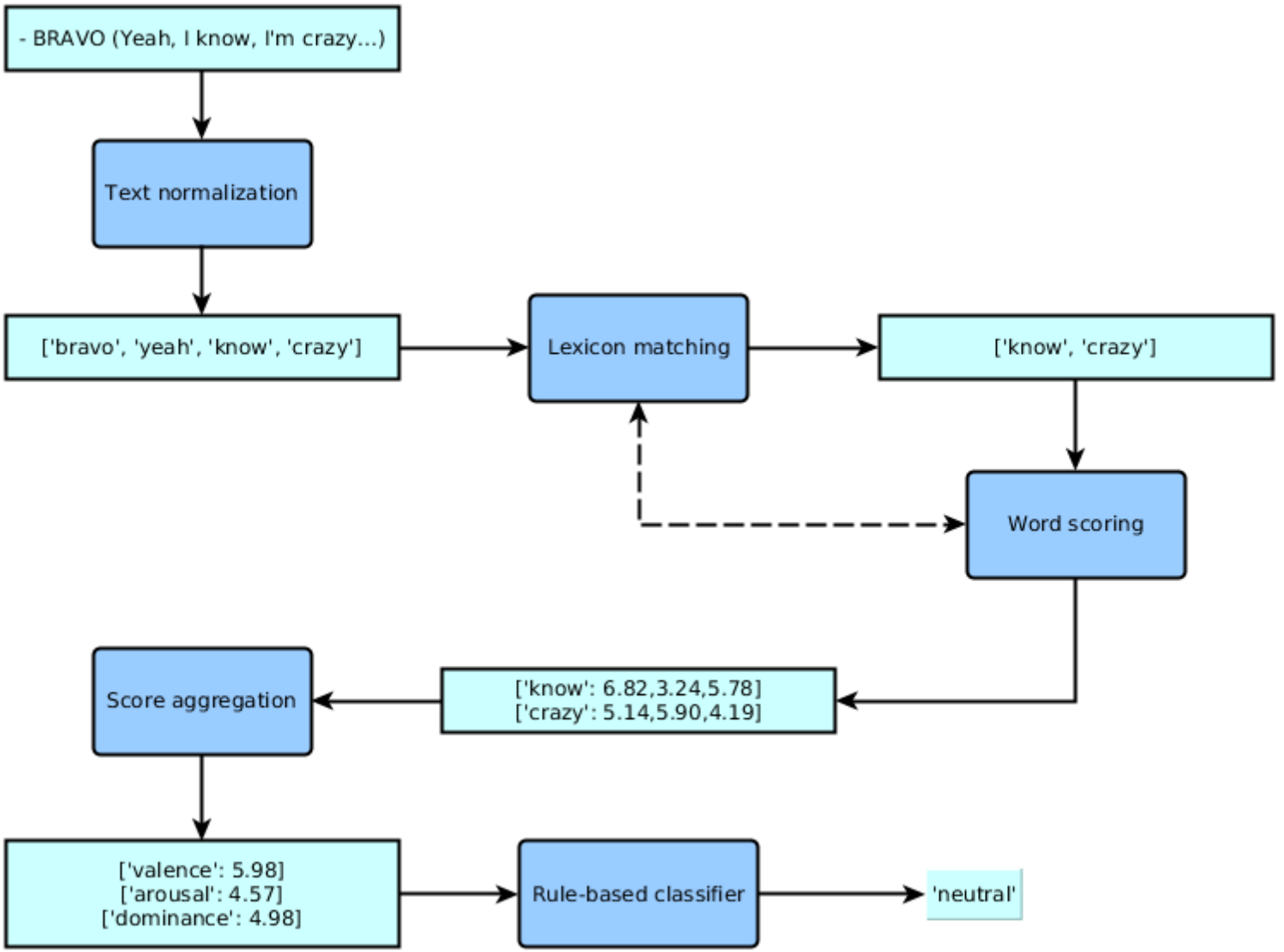
```
- valence (the pleasantness of a stimulus)
- arousal (the intensity of emotion provoked by a stimulus)
- dominance (the degree of control exerted by a stimulus)
```

Word,valence,arousal,dominance

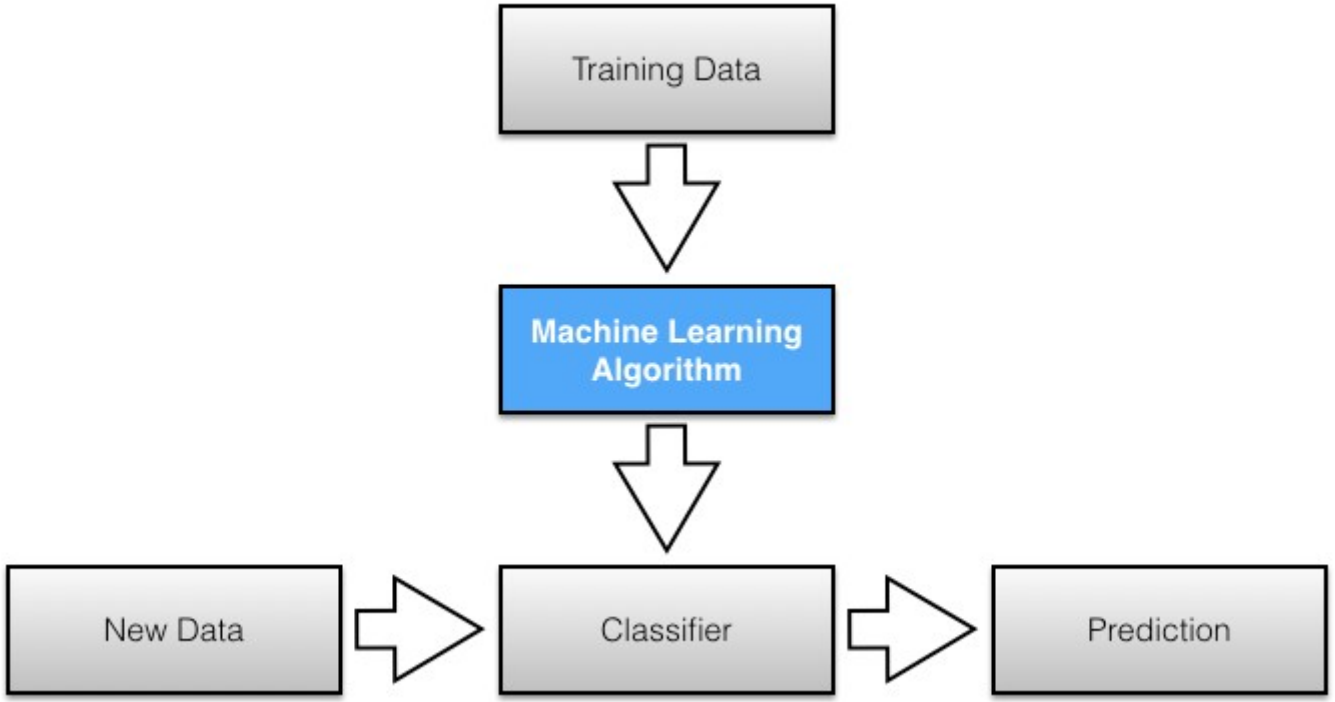
aardvark,6.26,2.41,4.27
abalone,5.3,2.65,4.95
abandon,2.84,3.73,3.32
abandonment,2.63,4.95,2.64
abbey,5.85,2.2,5
abdomen,5.43,3.68,5.15
abdominal,4.48,3.5,5.32
abduct,2.42,5.9,2.75
abduction,2.05,5.33,3.02
abide,5.52,3.26,5.33
abiding,5.57,3.59,6.6
ability,7,4.85,6.55
abject,4,3.94,4.35
ablaze,5.15,6.75,4.58
able,6.64,3.38,6.17
abnormal,3.53,4.48,4.7
abnormality,3.05,5,3.96
abode,5.28,2.9,5.05
...

VADER es el método
basado en ANEW en Python

Sentiment analysis (lexicon-based approach)



Sentiment analysis (machine learning approach)



- CARACTERÍSTICAS LINGÜÍSTICAS -

Características lingüísticas

Existen descriptores lingüísticos a nivel de oraciones y/o documentos. Los podemos usar para clasificar o para analizar textos.

(1) **Style**—This feature group captures the style and structure of the article. It includes POS (part of speech) tags and simple linguistic features such as number of quotes, punctuation, and all capitalized words. In total this group contains 55 features.

(2) **Complexity**—This feature group captures how complex the writing in the article is. It includes lexical diversity (type-token ratio), reading difficulty, length of words, and length of sentences. In total this group contains 6 features.

(3) **Bias**—This feature group captures the overall bias and subjectivity in the writing. This feature group is strongly based on Recasens et al. work [46] on detecting bias language. It includes number of hedges, factives, assertives, implicatives, and opinion words. It also include number of biased words according to the biased word lexicon in Reference [46] and how subjective the text is according to the subjectivity classifiers used in Reference [28]. In total this group contains 13 features.

(4) **Affect**—This feature group captures sentiment and emotion used in the text. It includes LIWC emotion features such as anger, anxiety, affect, and swear words [52]. It also includes positive and negative sentiment measures using VADER sentiment [32]. In total this group contains 12 features.

(5) **Moral**—This feature group is based on Moral Foundation Theory [22] and lexicons used in [38]. While this feature group has been used in previous studies, it has not been shown to perform well in the news setting or capture much meaningful signal. We include this group for completeness in our feature group analysis. In total this group contains 11 features.

https://dl.acm.org/doi/pdf/10.1145/3363818