citic

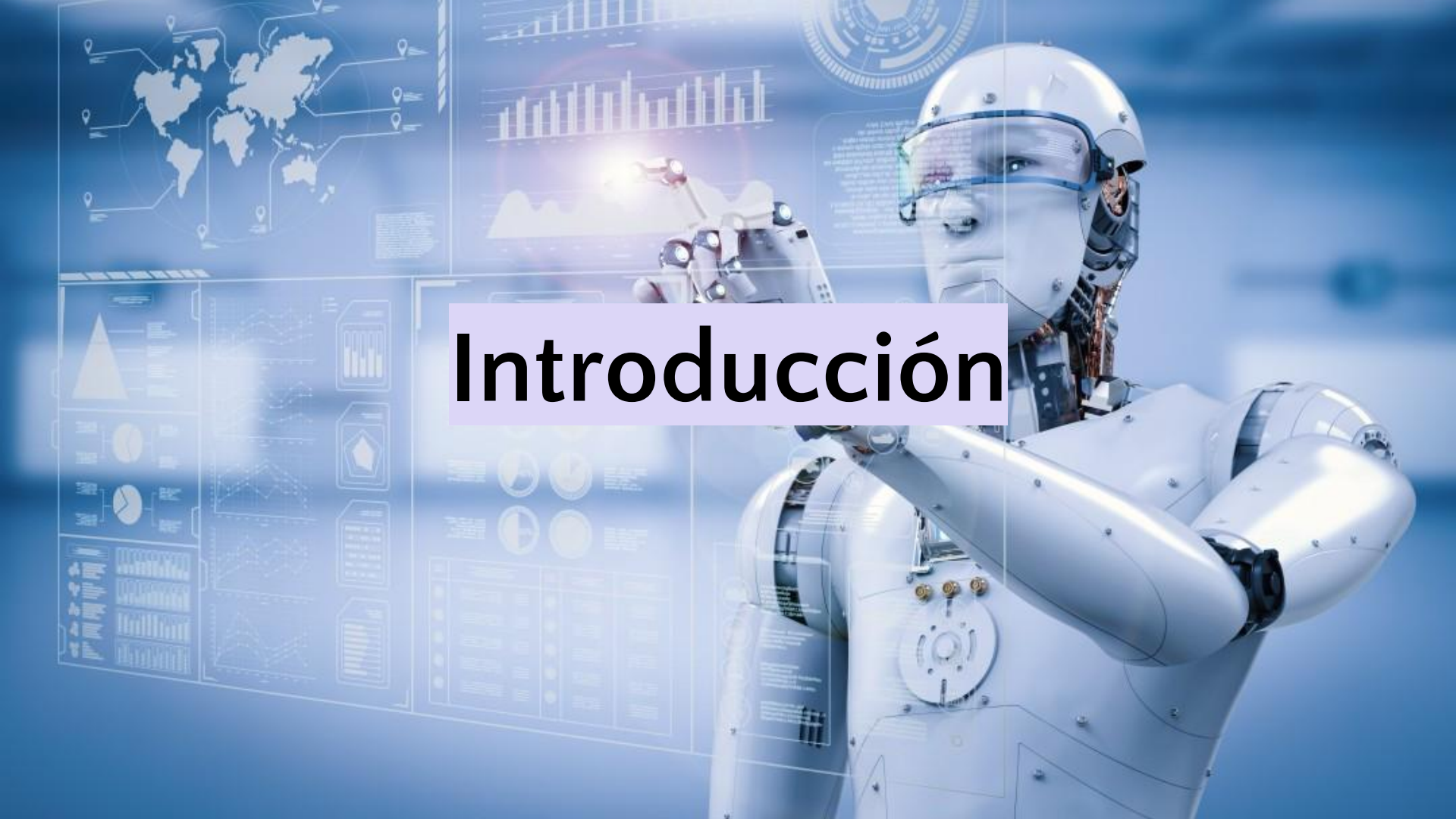
UNIVERSIDADE DA CORUÑA

Calidad vs. cantidad: La importancia de la selección de características en el análisis de datos

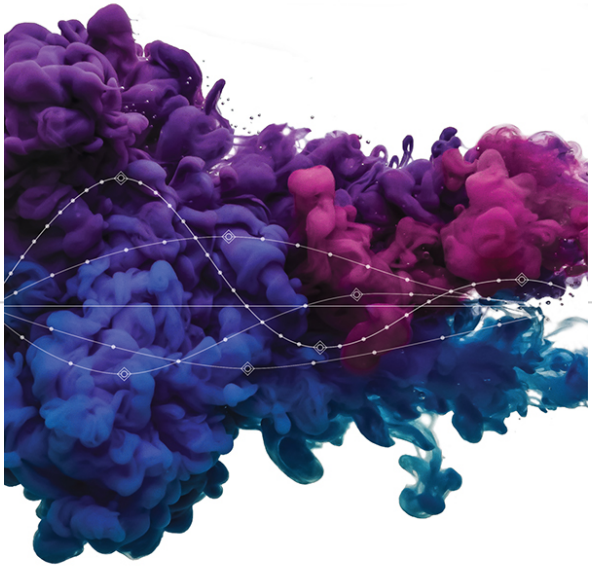
Verónica Bolón Canedo

Escuela de Verano de Inteligencia Artificial
[Sevilla, 15 Junio 2023]

AEPIA
ASOCIACIÓN ESPAÑOLA PARA LA INTELIGENCIA ARTIFICIAL



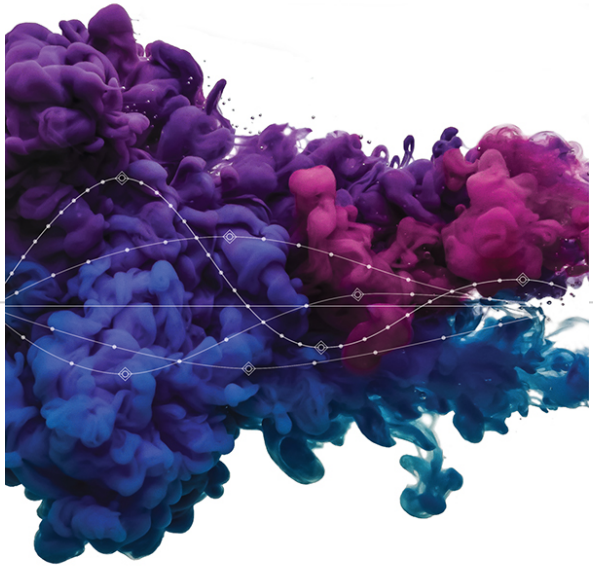
Introducción



Big data

“Los datos son el nuevo petróleo”

Clive Humby

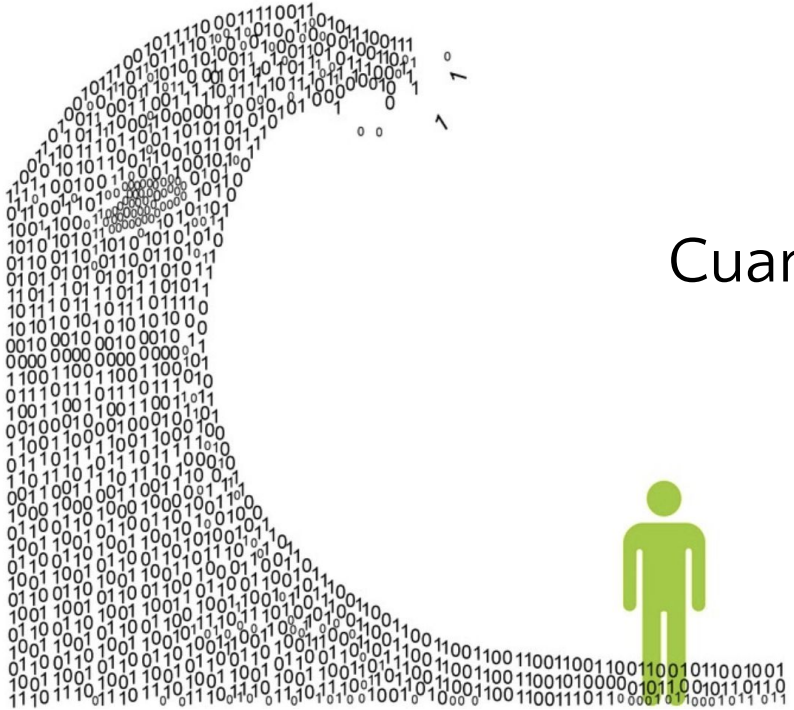


Big data

"Los datos son valiosos pero, si no están refinados, en realidad no se pueden usar"

Michael Palmer

Introducción

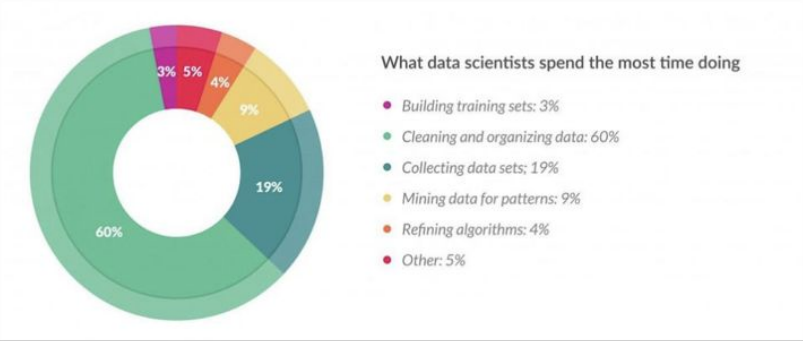


Cuantos más datos, ¿mejor?

Introducción

Introducción

Data preparation *accounts for about 80% of the work of data scientists*

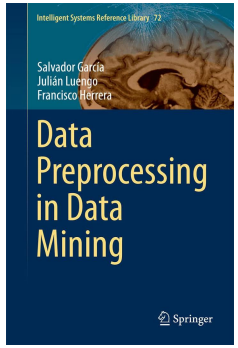


57% of data scientists regard cleaning and organizing data as the least enjoyable part of their work and 19% say this about collecting data sets.

¿Por qué se hace entonces el preprocesado de datos?

Porque los datos del mundo real son imperfectos

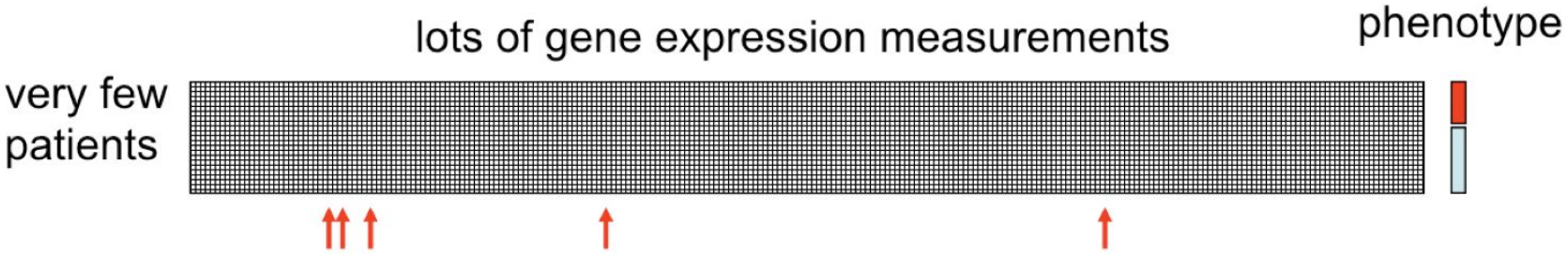
- Los datos del mundo real pueden estar contaminados y llevar a la extracción de patrones/reglas poco útiles.
- El preprocesado de datos puede general datasets más pequeños que el original, lo que nos permite mejorar la eficiencia del posterior aprendizaje
- Sin datos de calidad, no tendremos modelos de calidad



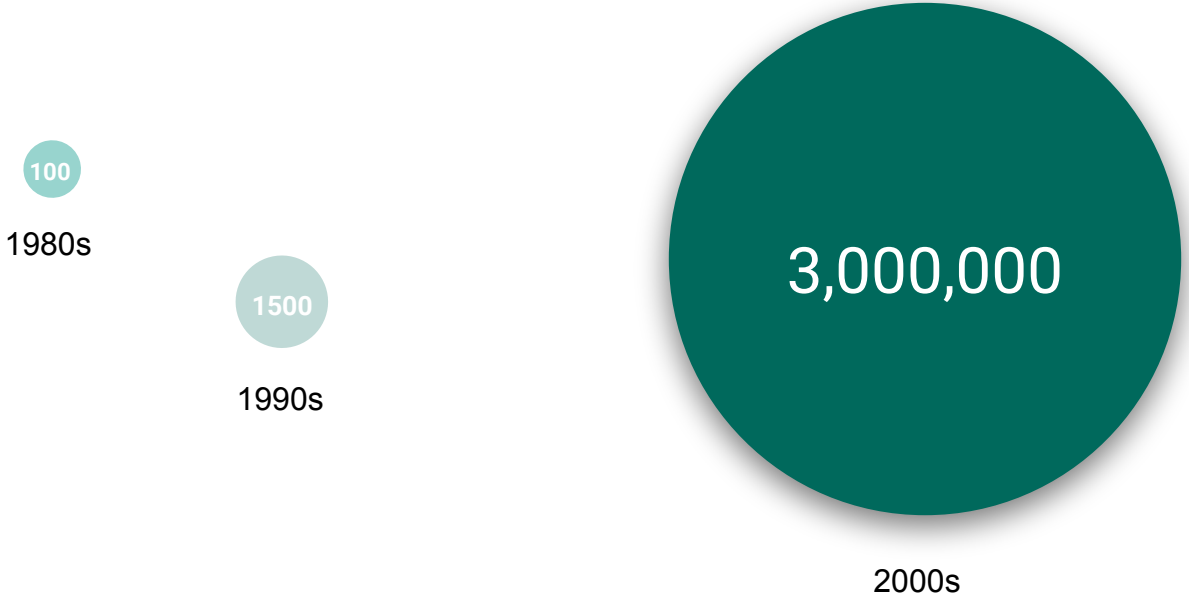
Reducción de la dimensionalidad

Big data -> Big dimension

Big Dimension: enorme número sin precedentes de características con las que hay que tratar actualmente



Big data -> Big dimension



Cuantas más características, ¿mejor?

La maldición de la dimensionalidad

Reducción de la dimensionalidad

Las técnicas de reducción de la dimensionalidad se aplican para reducir el número de características de los datos originales y, al mismo tiempo, mejorar el aprendizaje

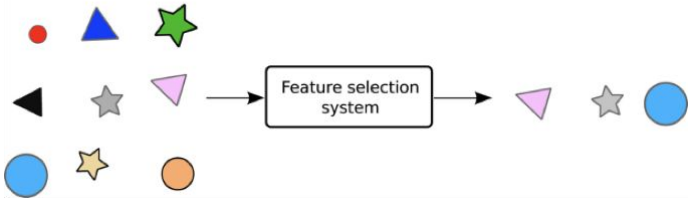
Técnicas de reducción de la dimensionalidad

Extracción de características

- Transforma las características originales en un subconjunto de nuevas características

Selección de características

- Elimina las características irrelevantes y/o redundantes



Extracción de características

Ventajas

- Nuevo conjunto de características normalmente menor que con selección de características -> Mayor poder discriminativo
- Muy útil para visualización

Inconvenientes

- La combinación de características puede no tener significado -> se pierde interpretabilidad y transparencia

Selección de características

La selección de características se puede definir como el proceso de seleccionar las características relevantes y eliminar aquellas que son irrelevantes o redundantes

¿Qué es una característica relevante?

Imaginemos que estamos tratando de adivinar el precio de un coche…

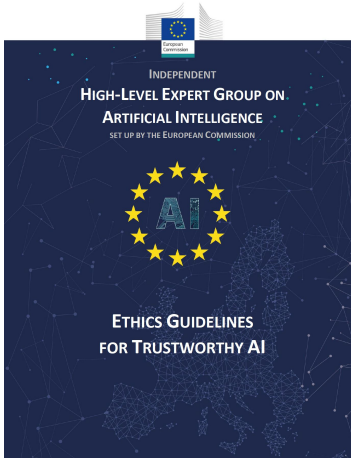
- Relevantes: motor, antigüedad, kilometraje, año de compra, presencia de óxido, …
- Irrelevantes: color de los limpiaparabrisas, presencia de pegatinas, …
- Redundantes: antigüedad / año de compra

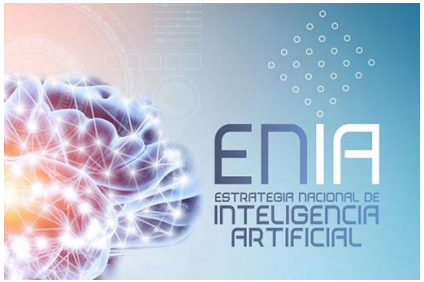
¿Por qué aplicar selección de características?

Reducción general de datos	Para limitar requisitos de almacenaje e incrementar velocidad de algoritmos
Reducción del conjunto de características	Para ahorrar recursos en la próxima ronda de recolección de datos
Mejora del rendimiento	Para mejorar la precisión de clasificación
Comprensión de los datos	Para adquirir conocimiento sobre el proceso que genera los datos o para visualización

Transparencia y explicabilidad

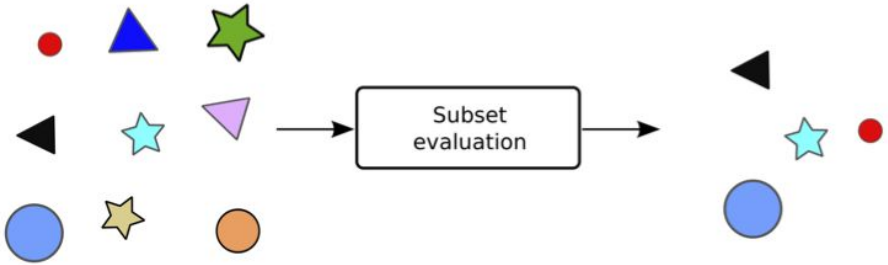
Necesitamos entender lo que hacen los algoritmos, cómo toman sus decisiones, en base a qué variables.

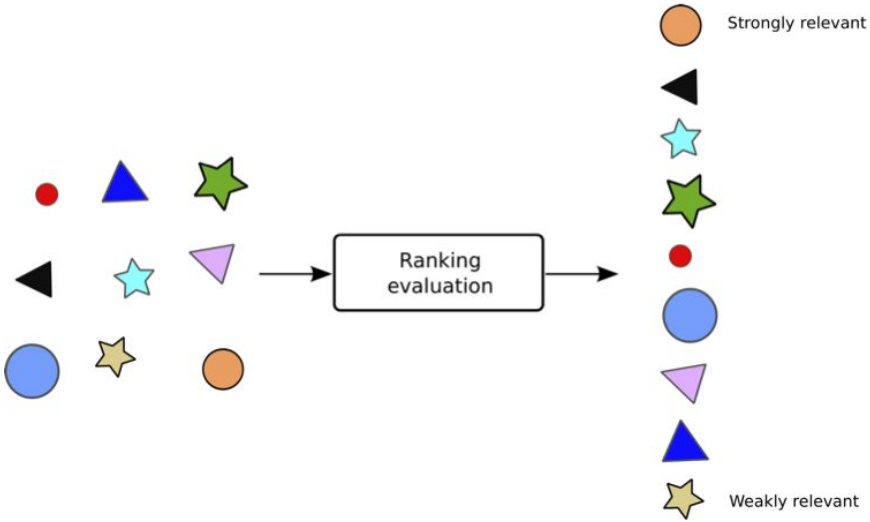




Selección de características

Tipos de métodos: Subconjunto vs ranker





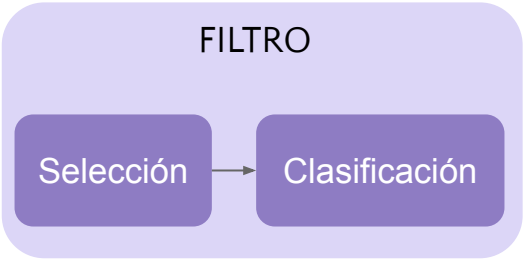
Componentes de un método de subconjunto

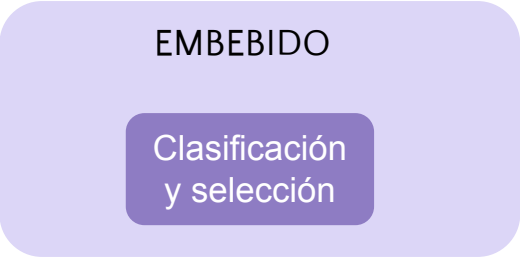
1. Estado inicial: En caso de que la búsqueda sea hacia atrás (*backward*), se empezará con el conjunto completo de características y se van eliminando paso a paso. Si la búsqueda es hacia delante (*forward*), se parte del conjunto vacío y se añaden características progresivamente
2. Espacio de estados: Formado por todos los posibles subconjuntos de características. Para un conjunto de n características, habría 2^n subconjuntos posibles (impracticable en la mayoría de los casos, por eso se usa una estrategia de búsqueda).
3. Estrategia de búsqueda: La búsqueda exhaustiva suele ser impracticable, por lo que se usan algoritmos como “ascensión a colinas” (*Hill climbing*) o “primero el mejor” (*best-first*).

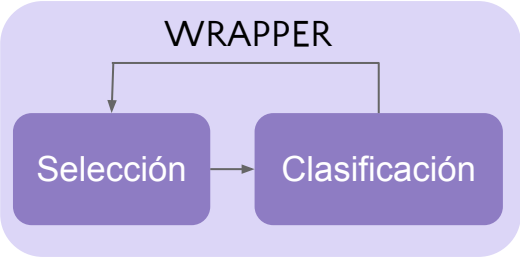
Componentes de un método de subconjunto

4. Función de evaluación: Determina la calidad de cada subconjunto de características. Esta función puede ser de varios tipos e incluir medidas de información, de distancia, de dependencia, de consistencia, de exactitud...
5. Criterio de parada: Establece el punto final para que el algoritmo deje de iterar. Por ejemplo, el proceso se acaba si la función de evaluación no mejora después de agregarse o eliminarse una nueva característica.

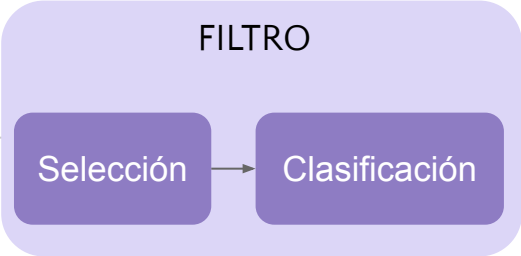
Tipos de métodos: Relación con clasificador







Tipos de métodos: Filtros



Ventajas

Independencia del clasificador
Bajo coste computacional
Rápido
Buena generalización

Inconvenientes

No interacción con el clasificador

Ejemplos

CFS
Información mutua
Fisher
ReliefF
Pearson

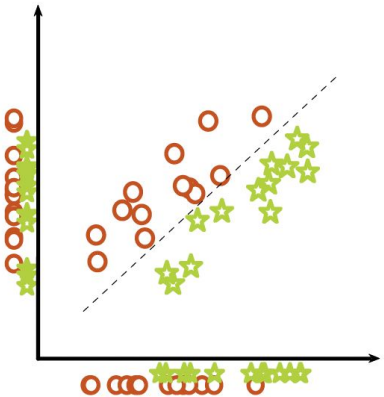
Tipos de métodos: Filtros

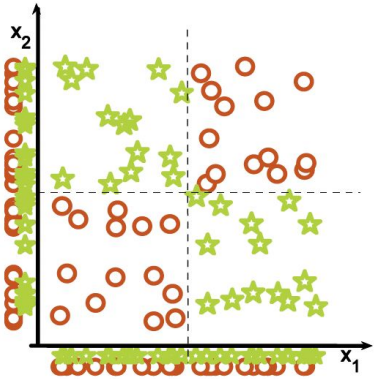
- Filtros univariados: Evalúan cada característica de forma independiente
- Filtros multivariados: Evalúan las características en el contexto de otras

Tipos de métodos: Filtros

- Filtros univariados: Evalúan cada característica de forma independiente

- Filtros multivariados: Evalúan las características en el contexto





Guyon & Elisseeff

Tipos de métodos: Filtros

Entrada: $D(X_1, X_2, \ldots, X_n)$ -> Conjunto de entrenamiento con *n* características

Entrada: S_0 -> subconjunto desde el que se empieza la búsqueda (estado inicial)

Entrada: δ -> criterio de parada

Salida: S_{opt} -> subconjunto óptimo de características

1. Se da un valor inicial $S_{opt} = S_0$
2. $\gamma_{mejor} = evaluar(S_{opt}, D, M)$ -> se evalúa S_{opt} mediante una función de evaluación *M*
3. Se repite*:*
    1. $S = generar(D)$ -> se genera un subconjunto para su evaluación
    2. $\gamma = evaluar(S, D, M)$ -> se evalúa el subconjunto actual mediante M
    3. Si γ es mejor que γ_{mejor} entonces:
        1. $\gamma_{mejor} = \gamma$
        2. $S_{opt} = S$
    4. Fin si
4. Hasta que (se alcanza δ)
5. Devuelve S_{opt}

Tipos de métodos: Embebidos

EMBEBIDO

Clasificación y selección

Ventajas

Interacción con el clasificador
Coste computacional más bajo que wrappers

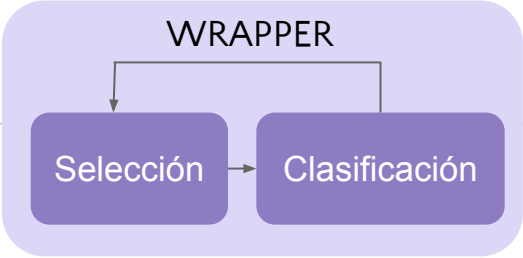
Inconvenientes

Dependiente del clasificador

Ejemplos

SVM-RFE
Lasso

Tipos de métodos: Wrappers



Ventajas

Interacción con el clasificador
Considera dependencia entre características

Inconvenientes

Alto coste computacional
Riesgo de sobreajuste
Dependiente del clasificador

Ejemplos

Sequential Forward Search
Sequential Backward Search

Tipos de métodos: Wrappers

Entrada: $D(X_1, X_2, \ldots, X_n)$ -> Conjunto de entrenamiento con *n* características

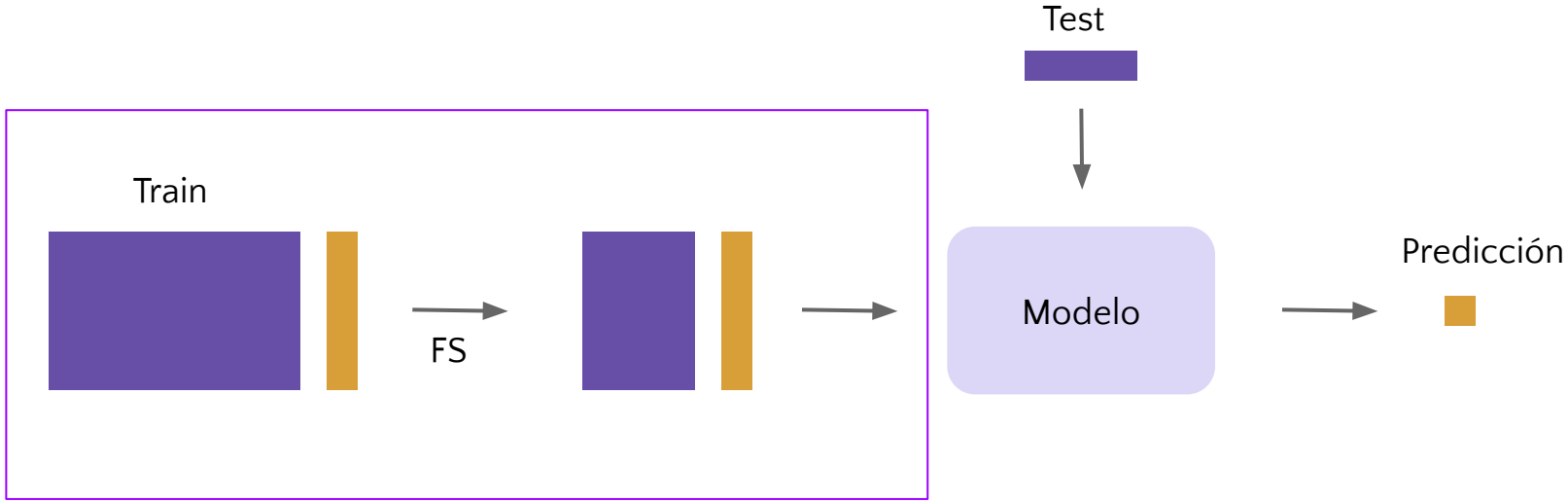
Entrada: S_0 -> subconjunto desde el que se empieza la búsqueda (estado inicial)

Entrada: δ -> criterio de parada

Salida: S_{opt} -> subconjunto óptimo de características

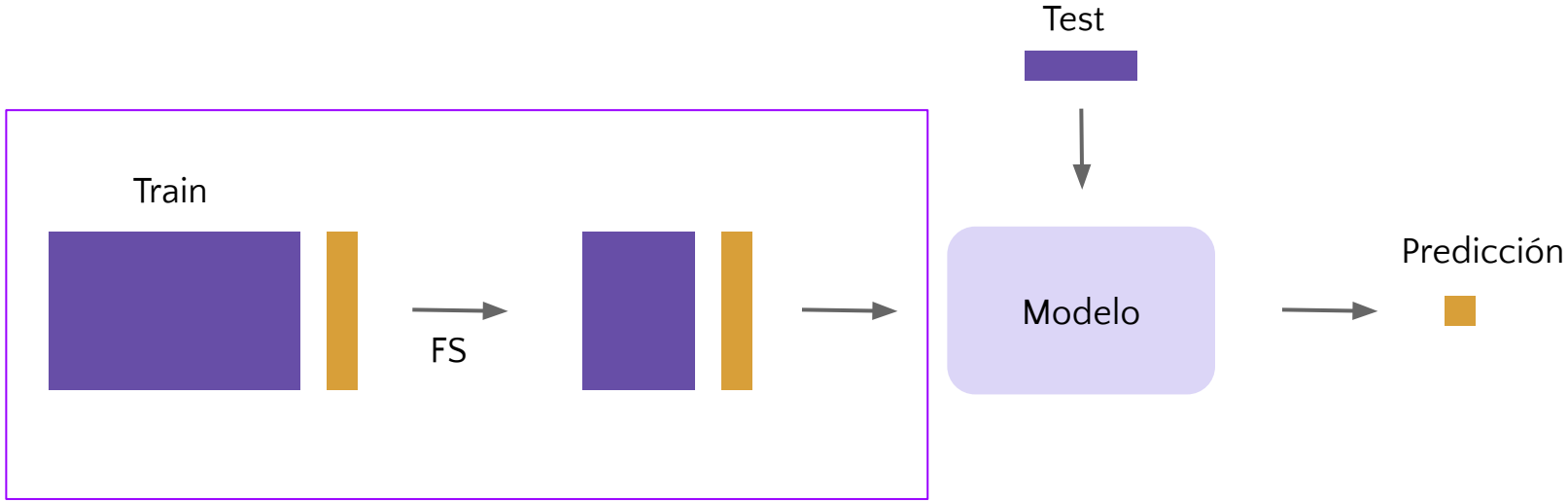
1. Se da un valor inicial $S_{opt} = S_0$
2. $\gamma_{mejor} = evaluar(S_{opt}, D, I)$ -> se evalúa S_{opt} mediante una función de evaluación *I*
3. Se repite*:*
    1. $S = generar(D)$ -> se genera un subconjunto para su evaluación
    2. $\gamma = evaluar(S, D, I)$ -> se evalúa el subconjunto actual mediante I
    3. Si γ es mejor que γ_{mejor} entonces:
        1. $\gamma_{mejor} = \gamma$
        2. $S_{opt} = S$
    4. Fin si
4. Hasta que (se alcanza δ)
5. Devuelve S_{opt}

Metodología general



Metodología general

La selección de características es parte del **entrenamiento**, NUNCA se debe hacer sobre datos de test



Métodos de selección de características

Subconjunto vs. ranker

Filtros vs. embebidos vs. wrappers

Univariados vs multivariados

¡El mejor método no existe!

Métodos de selección de características

	Uni/Multivariado	Ranking/Subconjunto	Autor y Año	Complejidad[3]
Chi-Squared	Univariado	Ranker	Liu & Setiono (1995)	nm
F-score (Fisher score)	Univariado	Ranker	Duda et al. (1999)	nm
Information Gain	Univariado	Ranker	Quinlan (1986)	nm
ReliefF	Multivariado	Ranker	Kononenko (1994)	n^2m
mRMR	Multivariado	Ranker	Peng et al. (2005)	nm^2
SVM-RFE	Multivariado	Ranker	Guyon et al. (2002)	$\max(n, m)m^2$
CFS	Multivariado	Subconjunto	Hall (1999)	nm^2
FCBF	Multivariado	Subconjunto	Yu & Liu (2003)	$nm \log m$
INTERACT	Multivariado	Subconjunto	Zhao & Liu (2007)	nm^2
Consistency	Multivariado	Subconjunto	Dash & Liu (2003)	nm^2

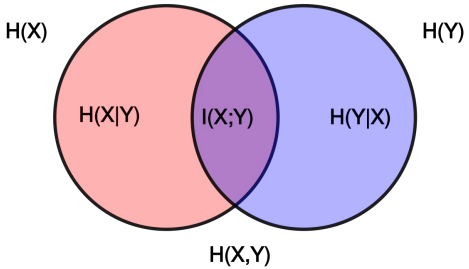
n es el número de ejemplos y *m* el de características

Información mutua

$$I(A;B) = \sum_{b \in B} \sum_{a \in A} p(a,b) \log \left(\frac{p(a,b)}{p(a)p(b)} \right)$$

Mide la dependencia entre la variable X y la clase Y

Cero cuando son independientes
Máxima cuando son idénticas



Información mutua

X	I(X;Y)
35	0,848
42	0,811
10	0,810
654	0,611
22	0,443
59	0,388
...	...
212	0,09
39	0,05

Información mutua

k=5

X	I(X;Y)
35	0,848
42	0,811
10	0,810
654	0,611
22	0,443
59	0,388
...	...
212	0,09
39	0,05

Información mutua

k=2

X	I(X;Y)
35	0,848
42	0,811
10	0,810
654	0,611
22	0,443
59	0,388
...	...
212	0,09
39	0,05

Información mutua

$I(X;Y) > 0,5$

X	I(X;Y)
35	0,848
42	0,811
10	0,810
654	0,611
22	0,443
59	0,388
...	...
212	0,09
39	0,05

mRMR

Algorithm 1 mRMR: original algorithm

```
INPUT: candidates, numFeaturesWanted
// candidates is the set of initial features
// numFeaturesWanted is the number of selected features
OUTPUT: selectedFeatures // The set of selected features.
for  feature fi in candidates do
   relevance = mutualInfo(fi, class);
   redundancy = 0;
   for  feature fj in candidates do
      redundancy += mutualInfo(fi, fj);
   end for
   mrmrValues[fi] = relevance - redundancy;
end for
selectedFeatures = sort(mrmrValues).take(numFeaturesWanted);
```

$$\max_{x_j \in X - S_{m-1}} \left[I\left(x_j; c\right) - \frac{1}{m-1} \sum_{x_i \in S_{m-1}} I\left(x_j; x_i\right) \right]$$

ReliefF

Algorithm 1: Pseudo-code of ReliefF algorithm.

Data: training set D, iterations m, attributes a
Result: the vector W of estimations of the qualities of attributes

```
set all weights W[A] := 0
for i ← 1 to m do
    randomly select an instance R_i
    find k nearest hits H_j
    for each class C ≠ class(R_i) do
        from class C find k nearest misses M_j(C)
    end
end
for f ← 1 to a do
    W[f] := W[f] − Σ_{j=1}^{k} diff(f,R_i,H_j) / (m·k) +
    Σ_{C≠class(R_i)} [ P(C)/(1−P(class(R_i))) Σ_{j=1}^{k} diff(f,R_i,M_j(C)) ] / (m·k)
end
```

$$diff(A,I_1,I_2) = \begin{cases} 0; & value(A,I_1) = value(A,I_2) \\ 1; & otherwise \end{cases}$$

Si R_i y H_j tienen valores diferentes para A, es un atributo no deseable

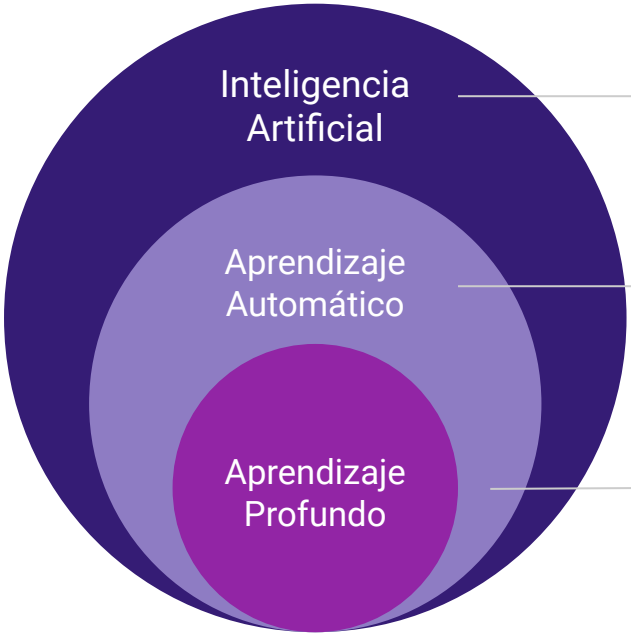
Si R_i y M_j tienen valores diferentes para A, es un atributo deseable

SVM-RFE

```
The pseudo code of SVM-RFE
Input: training dataset
Output: the ranked features dataset
Begin
    F={all input features}
    While (F is not empty) do
        Train F by a linear SVM
        Compute the weight vector w_i
        Rank the features in F by w_i^2
        Find the bottom ranked feature
        F = F — {the bottom ranked feature}
    Return the ranked features dataset
End
```

¿Por qué selección de características?

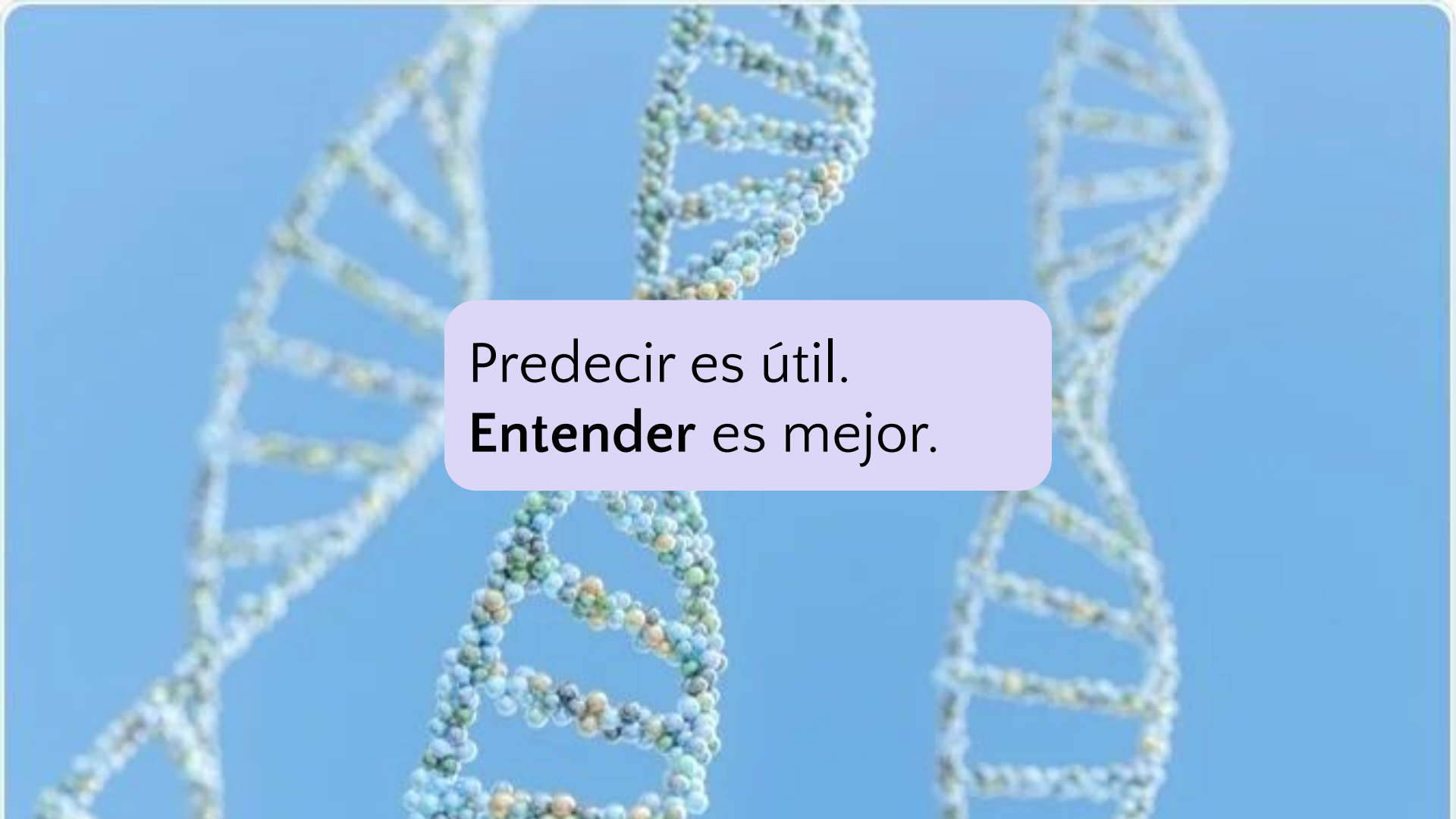
IA, Aprendizaje Automático y Aprendizaje Profundo



Capacidad de una máquina para imitar el comportamiento humano inteligente

Área de la IA que permite que un sistema aprenda y mejore automáticamente a partir de la experiencia

Área del aprendizaje automático que utiliza algoritmos complejos y redes neuronales profundas para entrenar un modelo


Predecir es útil.
Entender es mejor.

Necesidad de selección de características

Bioinformática: identificar biomoléculas clave para explicar un fenotipo

JOURNAL ARTICLE

Block HSIC Lasso: model-free biomarker detection for ultra-high dimensional data

Héctor Climente-González, Chloé-Agathe Azencott, Samuel Kaski, Makoto Yamada

Bioinformatics, Volume 35, Issue 14, July 2019, Pages i427–i435,
https://doi.org/10.1093/bioinformatics/btz333
Published: 05 July 2019

Necesidad de selección de características

Fairness: identificar las características usadas en un proceso de decisión, en lugar de centrarse en la justicia de los resultados de la decisión

Proceedings of the AAAI Conference on Artificial Intelligence

Current Archives About

Home / Archives / Vol. 32 No. 1 (2018): Thirty-Second AAAI Conference on Artificial Intelligence / AAAI Technical Track: Applications

Beyond Distributive Fairness in Algorithmic Decision Making: Feature Selection for Procedurally Fair Learning

Necesidad de selección de características

Nanotecnología: determinar las condiciones experimentales más relevantes y características psicoquímicas que deben tenerse en cuenta al realizar una evaluación de riesgos de nanotoxicología

Review Article

Nanotoxicology data for *in silico* tools: a literature review

Irini Furxhi, Finbarr Murphy, Martin Mullins, Athanasios Arvanitis & Craig A. Poland

Pages 612-637 | Received 05 Dec 2019, Accepted 09 Feb 2020, Published online: 26 Feb 2020

Necesidad de selección de características

Un aspecto común de estas aplicaciones es que no son tareas puras de clasificación. De hecho, comprender qué características son relevantes es tan importante como una clasificación precisa, ya que estas características pueden proporcionarnos nuevos conocimientos sobre el sistema subyacente. Esto es particularmente importante para lograr la transparencia de los sistemas de IA e implementar el derecho a la explicación.

Futuro de la selección de características

Retos

- Desarrollo de métodos eficientes y escalables
- Aprendizaje profundo y representaciones automáticas: a veces las aproximaciones aprendidas pueden tener dimensionalidad excesiva
- Dados desequilibrados y sesgados: puede haber características subrepresentadas
- La selección de características debe abordar el desafío de proteger la privacidad y la seguridad de datos sensibles

¡Gracias
por vuestra atención!

***Preguntas*?**

Contacto:

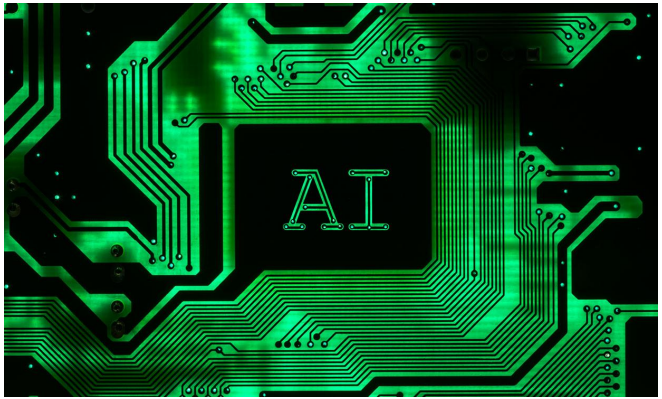
veronica.bolon@udc.es

@veronicabolon

IA sostenible

El término IA verde se refiere a investigación en IA que produce resultados novedosos pero que tengan en cuenta el coste computacional, promoviendo una reducción en los recursos consumidos.

El término IA roja se refiere a investigación en IA que busca mejorar la precisión de los modelos a través del uso de recursos computacionales masivos sin tener en cuenta el coste.



Selección de características verde

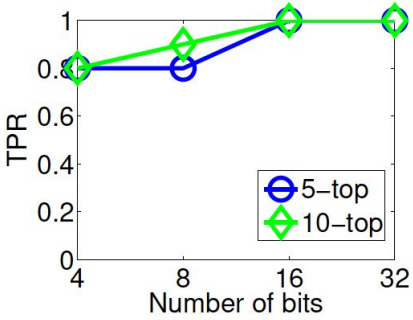
- Trabajamos con métodos basados en información mutua
- Parámetros en precisión reducida:
  - Cálculo del número de ocurrencias de cada evento
  - Uso de tabla look-up para calcular los logaritmos

$$p(a) \approx \frac{n_a}{n}$$

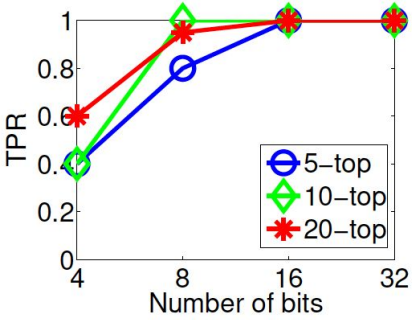
$$I(A;B) = \sum_{b \in B} \sum_{a \in A} p(a,b) \log\left(\frac{p(a,b)}{p(a)p(b)}\right)$$

$$p(a,b) \approx \frac{n_{a,b}}{n}$$

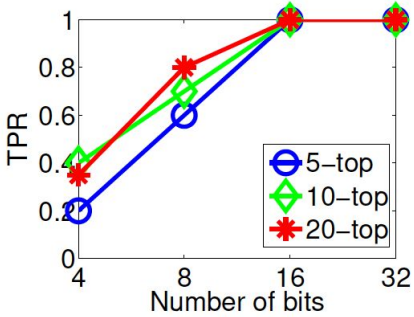
Selección de características verde



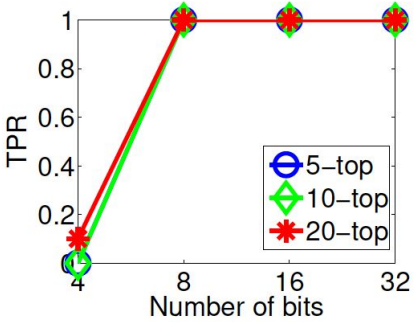
Congress



Waveform



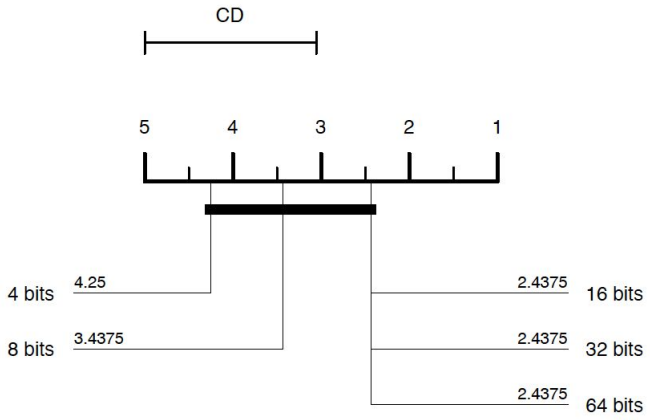
Gisette



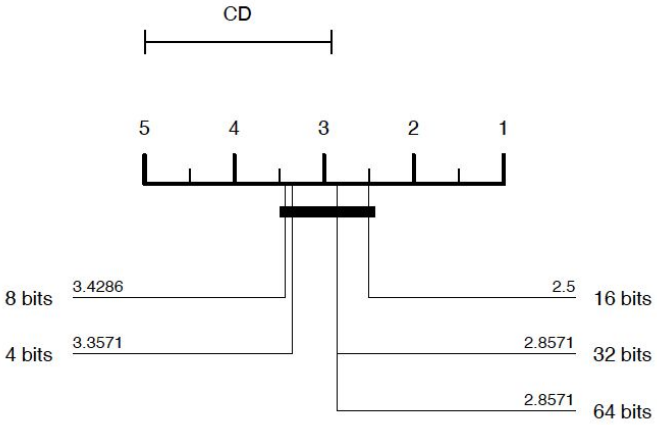
Arcene

Resultados para JMI en calidad de la selección: 16 bits son suficientes para seleccionar las mismas características que la versión de 64 bits

Selección de características verde



5-top features



20-top features

Resultados para JMI en clasificación: no hay diferencias significativas al usar menos bits