



ESCUELA TÉCNICA SUPERIOR DE INGENIEROS  
INFORMÁTICOS

UNIVERSIDAD POLITÉCNICA DE MADRID

---

# Aprendizaje automático para flujos de datos

---

TRABAJO FIN DE MÁSTER  
MÁSTER UNIVERSITARIO EN INTELIGENCIA ARTIFICIAL

AUTOR: Javier Ramos Fernández  
TUTOR/ES: María Concepción Bielza Lozoya y  
Pedro María Larrañaga Múgica



# Índice general

<b>1. Introducción</b>	<b>1</b>
1.1. Motivación . . . . .	1
1.2. Objetivos . . . . .	2
1.3. Metodología . . . . .	3
1.4. Organización de la memoria . . . . .	4
<b>2. Aprendizaje automático</b>	<b>5</b>
2.1. Notación . . . . .	6
2.2. Algoritmos de aprendizaje supervisado . . . . .	6
2.2.1. Clasificadores bayesianos . . . . .	7
2.2.2. Inducción de reglas . . . . .	7
2.2.3. Árboles de decisión . . . . .	7
2.2.4. Redes neuronales . . . . .	10
2.2.5. $k$ -Vecinos más cercanos . . . . .	12
2.2.6. Máquinas de vector soporte . . . . .	14
2.2.7. Regresión logística . . . . .	15
2.2.8. Métodos combinados de aprendizaje . . . . .	15
2.3. Algoritmos de aprendizaje no supervisado . . . . .	15
2.3.1. Agrupamiento . . . . .	16
2.4. Series temporales . . . . .	16
2.5. Redes bayesianas para el descubrimiento de conocimiento . . . . .	17
<b>3. Estado del arte: Aprendizaje automático para flujos de datos</b>	<b>19</b>
3.1. Introducción . . . . .	19
3.1.1. Conceptos . . . . .	19
3.2. Algoritmos de aprendizaje supervisado . . . . .	20
3.2.1. Clasificadores bayesianos . . . . .	20
3.2.2. Inducción de reglas . . . . .	20
3.2.3. Árboles de decisión . . . . .	20
3.2.4. Redes neuronales . . . . .	21
3.2.5. $k$ -Vecinos más cercanos . . . . .	21
3.2.6. Máquinas de vector soporte . . . . .	21
3.2.7. Regresión logística . . . . .	22
3.2.8. Métodos combinados de aprendizaje . . . . .	22
3.3. Algoritmos de aprendizaje no supervisado . . . . .	23

3.3.1. Agrupamiento . . . . .	23
3.4. Series temporales . . . . .	24
3.5. Redes bayesianas para el descubrimiento de conocimiento . . . . .	24
<b>4. Conclusiones y líneas futuras de trabajo</b>	<b>27</b>
<b>5. Sobre las referencias</b>	<b>29</b>
<b>A. Anexos</b>	<b>33</b>
<b>Bibliografía</b>	<b>35</b>

# Índice de figuras

2.1. Estructura de una red neuronal <a href="#">phuongphuong 4112716 and 405k4213231585</a>	10
2.2. Pseudocódigo del algoritmo KNN. Fuente: <a href="#">Larranaga et al.</a>	13
2.3. SVM. Fuente: <a href="#">Unagar and Unagar [2017]</a>	14



# Índice de tablas

2.1. Notación utilizada en el trabajo . . . . .	6
2.2. Problema de clasificación supervisada . . . . .	7





# Índice de algoritmos



# Capítulo 1

## Introducción

### 1.1. Motivación

Actualmente vivimos en la era de la información, una época en la que se está generando una cantidad ingente de información. Esta abundancia de datos se debe principalmente a la aparición de ordenadores y de otros dispositivos que son capaces de recoger información de lo que nos rodea, procesarla y transmitirla, como son los teléfonos móviles. Además, su capacidad de conexión a Internet hace que haya una generación de tal cantidad de información que es imposible tener un control absoluto de todo lo que circula por esta red informática de nivel mundial.

Este aumento exponencial del volumen y variedad de información ha generado la necesidad de llevar a cabo un almacenamiento masivo de los datos, y con ello el interés por analizar, interpretar y extraer información útil de los mismos con el objetivo de obtener conocimiento. Para manejar toda esta información, los sistemas tradicionales de almacenamiento de datos no son convenientes puesto que no tienen la capacidad necesaria para su correcto procesamiento. El volumen, la variedad y la velocidad de los grandes datos causan inconvenientes de rendimiento cuando se utilizan técnicas tradicionales de procesamiento de datos [Mohanty et al. \[2015\]](#). Por ello surge lo que se denomina **Big Data**, un concepto relativamente nuevo que se refiere a conjuntos de datos cuyo tamaño va más allá de la capacidad de las herramientas típicas de software de bases de datos para almacenar, gestionar y analizar datos [Oguntimilehin and Ademola \[2014\]](#).

Los datos son la materia prima para conseguir información provechosa, que se puede utilizar para llevar a cabo una toma de decisiones y la realización de conclusiones, por lo que se han desarrollado nuevas herramientas que sobrepasan las herramientas disponibles anteriormente para tratar este tipo de volúmenes de datos. De esta manera, surge el concepto de **minería de datos**, que se define como el proceso de extraer conocimiento útil y comprensible, previamente desconocido, desde grandes cantidades de datos almacenados en distintos formatos. Es decir, la tarea primordial de la minería de datos es encontrar modelos inteligibles a partir de los datos que posibiliten el hallazgo de aspectos previamente desconocidos de los mismos.

Para ejecutar el proceso de extracción del conocimiento, una de las posibilidades más populares es la aplicación de una rama de la inteligencia artificial denominada **aprendizaje automático**, la ciencia (y el arte) de programar computadoras para que puedan aprender de los datos [Gron \[2017\]](#). En este caso, el objetivo es que los ordenadores aprendan automáticamente sin intervención humana. Este proceso de aprendizaje se realiza proporcionándoles a los algoritmos pertinentes una serie de datos sobre los que se entrenan con el objetivo de buscar patrones en los mismos y llevar a cabo mejores decisiones en el futuro.

En general, los algoritmos de aprendizaje automático asumen que los datos están disponibles a la hora de llevar a cabo el entrenamiento y que se generan a partir de una distribución estática. No obstante, la información presente hoy en día circula en un entorno que cambia de manera *continua y rápida*, de manera que la distribución que genera los datos puede sufrir transformaciones. Todo esto ha propiciado la aparición de lo que se denominan **flujos de datos** (*data streams*), que son secuencias continuas y ordenadas de datos en tiempo real.

A la hora de tratar con flujos de datos, los algoritmos de aprendizaje automático tradicionales no son capaces de funcionar correctamente puesto que el volumen de los mismos puede llegar a ser *infinito*. Para que puedan realizar el entrenamiento, el conjunto de datos debe estar almacenado en memoria, y solo pueden llevar a cabo tareas de predicción cuando la fase de entrenamiento haya finalizado. Sin embargo, el almacenamiento de datos generados continuamente se hace inviable. Además, para que los algoritmos puedan manejar flujos de datos cuya distribución que los subyace puede cambiar es necesario un *procesamiento en tiempo real*, característica que los algoritmos tradicionales no poseen.

En este trabajo pretendemos realizar una revisión de literatura sobre métodos propuestos para aplicar aprendizaje automático en flujos de datos con el objetivo de tener una visión global de las diferentes posibilidades existentes para resolver el problema planteado anteriormente. Para ello, vamos a tener en cuenta diferentes algoritmos de aprendizaje automático existentes en el mercado actual.

## 1.2. Objetivos

El objetivo principal de este trabajo es realizar una **comparativa amplia de las diferentes aproximaciones propuestas para resolver el problema de aprendizaje automático para flujos de datos, principalmente tanto supervisado como no supervisado**. Para abordar esta meta, nos centraremos en los algoritmos de aprendizaje automático más populares. En el caso de aprendizaje supervisado, nos enfocaremos en propuestas relacionadas con *árboles de decisión*, *redes bayesianas*, *redes neuronales*, *inducción de reglas*, *vecinos más cercanos*, *máquinas de soporte vectorial* y *métodos combinados de aprendizaje* (ensemble). Con respecto al aprendizaje no supervisado, consideraremos primordialmente los algoritmos de *agrupamiento* (clustering), uno de los principales de este tipo de aprendizaje.

También se hará hincapié en algunas propuestas relacionadas con aprendizaje

### semisupervisado

Para alcanzar este objetivo, se han contemplado una serie de metas específicas dentro del proyecto:

- *Búsqueda de artículos relacionados con cada uno de los algoritmos de aprendizaje automático contemplados.* Con el objetivo de llevar a cabo una comparativa de los diferentes métodos propuestos de aprendizaje automático para flujos de datos, es de vital importancia realizar una búsqueda de diferentes artículos propuestos para cada uno de los algoritmos de aprendizaje automático mencionados anteriormente. Para ello hemos utilizado diferentes revistas que aborden esta rama de la inteligencia artificial, así como otros recursos como *Google Académico*.
- *Anotación de aspectos claves de los diferentes artículos encontrados.* Con ello se pretende tener disponible información resumida de las diferentes propuestas que se encuentran a nuestra disposición y utilizarla para compararlas con otros artículos con la finalidad de dar una perspectiva general de la utilidad de las diferentes aproximaciones existentes.
- *Estructuración de la revisión de la literatura de algoritmos de aprendizaje automático para flujos de datos en función de los algoritmos en los que se centren las diferentes propuestas encontradas.* Se persigue comparar los diferentes artículos hallados según el algoritmo de aprendizaje automático que aborden para que funcionen correctamente con flujos de datos. Para ello hemos realizado una división de los mismos en diferentes apartados.
- *Exploración de las diferentes revisiones halladas sobre algoritmos de aprendizaje automático para flujos de datos.* El objetivo de esto es conocer qué propuestas de las que hemos encontrado se encuentran referenciadas en esas revisiones para tener en cuenta qué artículos son novedosos con respecto a dichas revisiones. Además, se lleva a cabo esta exploración para tener conocimiento de que algoritmos de aprendizaje automático tratan con el fin de aportar nuevos algoritmos. Todo esto se realiza con la finalidad de establecer qué características nos diferencian de las revisiones encontradas.

## 1.3. Metodología

Para desarrollar este proyecto se ha seguido una metodología que permitiera sobrellevar las dificultades del mismo de la mejor forma posible. A continuación se enumeran los pasos ejecutados durante este proceso:

- En primer lugar, hemos procedido a realizar una búsqueda **exhaustiva de artículos relacionados con aprendizaje automático para flujos de datos**. Antes de realizar una comparación entre las diferentes propuestas, es necesario recabar la mayor cantidad de aproximaciones desarrolladas con el fin de

ampliar nuestra visión genérica del estado del arte del tema abordado. Para lograr esto, hemos indagado en numerosas revistas que incluyen dentro de su temática el aprendizaje automático para flujos de datos. Algunas que se han consultado y que son relevantes en el mundo académico son *Journal of Machine Learning Research*, *IEEE Transactions on Knowledge and Data Engineering* y *Machine Learning Journal*. También se han explorado diferentes editoriales como *Elsevier* y *Springer* en las que, aparte de artículos publicados en revistas, también aparecen propuestas de conferencias. De forma complementaria, hemos buscado artículos en *Google Académico*, una herramienta de búsqueda de Google que permite hallar literatura del mundo científico de diferentes recursos (bibliotecas, editoriales, etcétera).

- Tras aglomerar una cantidad aceptable de artículos, hemos iniciado la **lectura de los mismos**. Durante este proceso, hemos ido apuntando características relevantes de las propuestas con el fin de poder tener la información fundamental para realizar la comparativa entre diferentes artículos. Para guardar dicha información, hemos creado un documento en el que, por cada propuesta, se apunta el *título*, la *fecha de publicación* y *contenido de interés* de las mismas, así como *comparaciones con otras aproximaciones* que se encontraba dentro de los artículos para utilizarlas en el estado del arte. Asimismo, hemos añadido información adicional con respecto a la *presencia o no de los artículos de las diferentes revisiones* encontradas que abordan el tema de aprendizaje automático para flujos de datos con el objetivo de tener conocimiento sobre qué artículos de los que hemos encontrado aportan nueva información con respecto a dichas revisiones para establecer características que nos diferencian de las revisiones halladas. Estas propuestas se han ordenado por *fecha de publicación* para contemplar la evolución cronológica de las mismas.
- De forma paralela a la lectura de artículos, hemos procedido a **buscar más propuestas**, enfocándonos en encontrar aquellas que son más recientes para aportar más información que nos diferencie de las revisiones encontradas. También hemos aprovechado las referencias a otros papers que hemos hallado en los artículos revisados para añadirlos al estado del arte.
- Por otra parte, se ha creado un documento en el que se ha ido **almacenando información útil durante la lectura de los artículos para relacionar diferentes propuestas de la mejor manera posible**. Por ejemplo, hemos anotado diferentes categorías en las que se podrían clasificar las distintas propuestas.

## 1.4. Organización de la memoria

Pendiente

## Capítulo 2

# Aprendizaje automático

Arthur Samuel, uno de los pioneros del aprendizaje automático, estableció en 1959 una definición general de esta rama de la inteligencia artificial:

*El aprendizaje automático es el campo de estudio que da a las computadoras la capacidad de aprender sin estar programadas explícitamente.*

Tom Mitchell, otro investigador de aprendizaje automático reputado, propuso en 1997 una definición más precisa y más orientado a la ingeniería:

*Se dice que un programa de ordenador aprende de la experiencia  $E$  con respecto a alguna tarea  $T$  y alguna medida de rendimiento  $P$ , si su rendimiento en  $T$ , medido por  $P$ , mejora con la experiencia  $E$ .*

Por lo tanto, el aprendizaje automático se centra en aplicar sistemáticamente algoritmos para sintetizar las relaciones subyacentes de forma automática en un conjunto de datos proporcionados en forma de ejemplos a través de una fase de entrenamiento, de tal forma que en el futuro se utilice esta información para la ejecución de predicciones de eventos desconocidos y una mejor toma de decisiones. Los campos de aplicación que existen de esta rama de la inteligencia artificial son muy variados. Algunos de ellos son la predicción bursátil, predicción meteorológica, detección de correos spam, construcción de sistemas de recomendación y detección de fraude en el uso de tarjetas de crédito.

Según el propósito que persigan los algoritmos de aprendizaje automático, éstos se clasifican en dos categorías principales: **aprendizaje supervisado** y **aprendizaje no supervisado**. En el aprendizaje supervisado se engloban aquellos algoritmos que buscan aprender una *función de mapeo* entre una serie de características de entrada (variables predictoras) y una variable de salida (variable clase) mediante la ejecución de una fase de entrenamiento en la que se utilizan *datos de entrenamiento etiquetados* (valor de la variable clase conocida), de tal forma que se utiliza esta función para predecir el valor de la variable de salida a partir de los valores de las variables predictoras. En cambio, el aprendizaje no supervisado la finalidad es aprender una *función que modele la estructura o distribución subyacente en los datos*

a partir de datos de entrenamiento no etiquetados, de manera que se lleva a cabo una exploración de los datos sólo conociendo los valores de las variables predictoras.

Existe además otra categoría en la que se pueden incluir las técnicas de aprendizaje automático denominada **aprendizaje semisupervisado**. Los algoritmos que llevan a cabo este tipo de aprendizaje entrenan sobre un conjunto de datos parcialmente etiquetados, generalmente muchos datos sin etiquetar y una pequeña parte de datos etiquetados. Debido a que para la fase de entrenamiento se utilizan tanto datos etiquetados como no etiquetados, se considera que esta categoría se sitúa entre el aprendizaje supervisado y el no supervisado. En este trabajo se van a abordar algunas propuestas relacionadas con el aprendizaje semisupervisado para flujos de datos; no obstante, la contribución predominante de algoritmos de aprendizaje automático para flujos de datos que se va a realizar en este proyecto proviene de las categorías de aprendizaje supervisado y no supervisado.

## 2.1. Notación

Tabla 2.1: Notación utilizada en el trabajo

Símbolo	Explicación
$X_i$	Variable predictora $i$
$\mathbf{X}$	Conjunto de variables predictoras
$x_i$	Valor de la variable predictora $i$
$x_{ij}$	Valor $j$ de la variable predictora discreta $i$
$p(x_{ij})$	Probabilidad del valor $j$ de una variable $i$
$\mathbf{x}$	Instancia de las variables predictoras
$x_i^{(j)}$	Valor de la variable predictora $i$ de la instancia $j$
$\mathbf{x}^{(j)}$	Valores de las variables predictoras de la instancia $j$
$C$	Variable clase
$c_j$	Posible valor $i$ de la variable clase
$c^{(j)}$	Valor de la variable clase en la instancia $j$
$(\mathbf{x}^{(j)}, c^{(j)})$	Estructura de una instancia $j$
$N$	Número de instancias de un conjunto de datos
$n$	Número de variables predictoras
$m$	Número de valores que puede tomar una variable predictora
$D$	Fichero de casos

## 2.2. Algoritmos de aprendizaje supervisado

Improving Hoeffding Trees (2007)



Tabla 2.2: Problema de clasificación supervisada

	$X_1$	$\dots$	$X_n$	C
$(\mathbf{x}^{(1)}, c^{(1)})$	$x_1^{(1)}$	$\dots$	$x_n^{(1)}$	$c^{(1)}$
$(\mathbf{x}^{(2)}, c^{(2)})$	$x_1^{(2)}$	$\dots$	$x_n^{(2)}$	$c^{(2)}$
$\dots$		$\dots$		$\dots$
$(\mathbf{x}^{(N)}, c^{(N)})$	$x_1^{(N)}$	$\dots$	$x_n^{(N)}$	$c^{(N)}$
$\mathbf{x}^{(N+1)}$	$x_1^{(N+1)}$	$\dots$	$x_n^{(N+1)}$	???

### 2.2.1. Clasificadores bayesianos

Aprendizaje discriminativo de clasificadores Bayesianos Discrete Bayesian Network Classifiers: A Survey

### 2.2.2. Inducción de reglas

Quisque ullamcorper placerat ipsum. Cras nibh. Morbi vel justo vitae lacus tincidunt ultrices. Lorem ipsum dolor sit amet, consectetur adipiscing elit. In hac habitasse platea dictumst. Integer tempus convallis augue. Etiam facilisis. Nunc elementum fermentum wisi. Aenean placerat. Ut imperdiet, enim sed gravida sollicitudin, felis odio placerat quam, ac pulvinar elit purus eget enim. Nunc vitae tortor. Proin tempus nibh sit amet nisl. Vivamus quis tortor vitae risus porta vehicula.

Fusce mauris. Vestibulum luctus nibh at lectus. Sed bibendum, nulla a faucibus semper, leo velit ultricies tellus, ac venenatis arcu wisi vel nisl. Vestibulum diam. Aliquam pellentesque, augue quis sagittis posuere, turpis lacus congue quam, in hendrerit risus eros eget felis. Maecenas eget erat in sapien mattis porttitor. Vestibulum porttitor. Nulla facilisi. Sed a turpis eu lacus commodo facilisis. Morbi fringilla, wisi in dignissim interdum, justo lectus sagittis dui, et vehicula libero dui cursus dui. Mauris tempor ligula sed lacus. Duis cursus enim ut augue. Cras ac magna. Cras nulla. Nulla egestas. Curabitur a leo. Quisque egestas wisi eget nunc. Nam feugiat lacus vel est. Curabitur consectetur.

### 2.2.3. Árboles de decisión

El aprendizaje mediante **árboles de decisión** es un método de aproximación de una función objetivo en el cual la función objetivo es representada mediante un *árbol de decisión*, un clasificador expresado como una partición recursiva del espacio de instancias que consta de *nodos*, *ramas* y *hojas*.

- Los *nodos* representan atributos utilizados para particionar un conjunto de datos en subconjuntos de los mismos de acuerdo con una determinada función discreta de los valores de los atributos de entrada. Es decir, representa un test del valor de un atributo.

- Las *ramas* son los distintos valores de los atributos (nodos), que dan lugar a los diferentes hijos de los nodos (diferentes particiones). Las particiones que se realizan en los distintos nodos del árbol de decisión varían en función de si los atributos son *discretos* o *numéricos*. En el caso de los atributos *discretos*, se realizan las particiones en función de cada uno de los posibles valores de ese atributo. En el caso de los atributos *numéricos*, se particiona teniendo en cuenta diferentes rangos de valores.
- Las *hojas* representan las posibles etiquetas de clase. Los árboles de decisión también se conocen como *árboles de clasificación*, de tal forma que este paradigma clasifica una instancia a una determinada clase. No obstante, también existen los *árboles de regresión*, donde la variable de destino puede tomar valores continuos.

De esta manera, a la hora de clasificar una instancia, se comienza desde el nodo *raíz* y, en función de los valores de las variables predictoras de esa instancia, el ejemplo va recorriendo las ramas pertinentes asociadas a esos valores hasta que llega a una hoja, que tiene asignada una clase que se va a utilizar para clasificar la instancia. Por lo tanto, las *ramas* representan las conjunciones de características que conducen a esas etiquetas de clase.

A la hora de llevar a cabo la construcción de ese paradigma clasificatorio, es necesario que las particiones se realizan de tal forma que los subconjuntos resultantes sean lo más *puros* posible, es decir, que en cada subconjunto de instancias de las hojas todos los ejemplos pertenezcan a la misma clase, lo que conlleva a una correcta partición de los datos proporcionados. La construcción del árbol se basa principalmente escoger, en cada caso, el atributo que mejor particiona en cada nodo, y para tomar esta decisión se utiliza una *función de medida* del grado de impureza de la partición realizada por cada uno de los atributos considerados. Algunas de las funciones más populares son la **información mutua** (o *ganancia de información*) y el **índice Gini**. La *información mutua* se define como la cantidad que mide una relación entre dos variables aleatorias; concretamente mide cuánto se reduce la incertidumbre (*entropía*, medida de impureza) de una variable al conocer el valor de la otra variable. En el caso de la elección del mejor atributo en los árboles de decisión, interesa la cantidad de información mutua entre una variable predictora y la variable clase, de tal forma que se elige aquel atributo que más reduzca la incertidumbre. La fórmula matemática es la siguiente:

$$I(X_i, C) = H(C) - H(C|X_i) \quad (2.1)$$

donde  $H(C)$  representa la entropía de una variable, en este caso de la variable clase, que se define de la siguiente manera:

$$H(C) = - \sum_{j=1}^m p(c_j) \log_2 p(c_j) \quad (2.2)$$

y  $H(C_i)$  representa la entropía de una variable sabiendo el valor de otra variable, en este caso la entropía de la variable clase sabiendo el valor de una variable predictora,

que se define de la siguiente manera:

$$H(C|X_i) = - \sum_{j=1}^{m_C} \sum_{l=1}^{m_{X_i}} p(c_j, x_{il}) \log_2 p(c_j|x_{il}) \quad (2.3)$$

Por otra parte, el *índice Gini* es un criterio basado en impurezas que mide las divergencias entre la distribución de probabilidad de los valores de los atributos objetivo (Gulati et al. [2016]). Es una alternativa a la *ganancia de información*. Se define con la siguiente fórmula:

$$Gini = 1 - \sum_{j=1}^m p(c_j)^2 \quad (2.4)$$

Existen dos maneras llevar a cabo el proceso de construcción del árbol de decisión, desde **arriba hacia abajo** (*top-down*) y desde **abajo hasta arriba** (*bottom-up*). No obstante, la aproximación *top-down* es la que goza de mayor popularidad. Algunos de los algoritmos más conocidos que llevan a la práctica esta aproximación son el **ID3** (Quinlan [1986]), el **C4.5** (Quinlan [1993]) y **CART** (Breiman et al. [1984]). Estos algoritmos pertenecen a la familia de los *Top Down Induction of Decision Trees* (TDIDT), que inducen el modelo del árbol de decisión a partir de datos preclasificados. El algoritmo de construcción en el que se basan los árboles de esta familia es el *método de Hunt*, que es el siguiente:

(Meter algoritmo del documento "Tema 8. Árboles de decisión")

El algoritmo *ID3* construye el árbol de decisión mediante la aproximación *top-down* sin realizar *backtracking*, es decir, lleva a cabo una estrategia de búsqueda voraz a través del espacio de todos los árboles de clasificación posibles. Para tomar la decisión de elegir la variable que aporta mayor información a la hora de realizar las diferentes particiones el algoritmo utiliza la **ganancia de información**. El *ID3* tiene algunos inconvenientes: se sobreajusta a los datos de entrenamiento, no es capaz de manejar atributos numéricos ni valores faltantes y no soporta un podado del árbol.

El algoritmo *C4.5* se desarrolló como una mejora del algoritmo *ID3*. En primer lugar, en lugar de utilizar la *ganancia de información* para elegir la variable predictora más informativa en cada momento, emplea lo que se denomina la **proporción de ganancia** (*gain ratio*), que se calcula dividiendo la información mutua entre una variable predictora y la variable clase por la entropía de la variable predictora ( $I(X_i, C)/H(X_i)$ ), que evita a aquellas variables predictoras que tengan un mayor rango de valores no tengan tanta probabilidad de ser elegidas, como ocurría en el *ID3*. Además, permite trabajar con **atributos continuos** definiendo un umbral, de tal forma que particiona las instancias en función de si el valor del atributo es mayor o menor que el umbral. Por otra parte, con respecto al manejo de **datos faltantes**, se estiman los mismos imputación, y también **asigna pesos** diferentes a los atributos en función del coste asociado a los mismos. Asimismo, trata de lidiar el sobreajuste del modelo realizando una **poda**, que puede hacerse parando la construcción del árbol antes de que clasifique perfectamente los datos (*pre-pruning*)

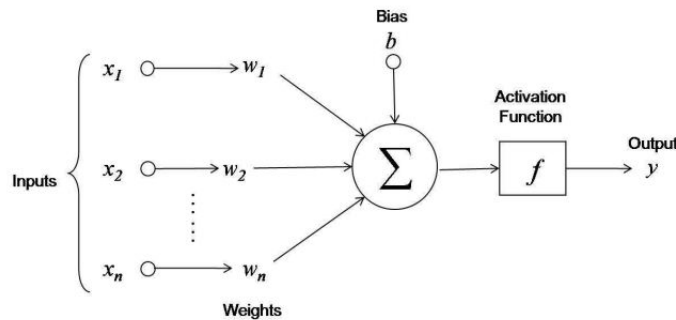
o permitiendo que el modelo se sobreajuste y luego sustituir subárboles por hojas (*post-prunning*) y es capaz de manejar el ruido. La limitación principal del *C4.5* es que produce modelos que no dan buenos resultados cuando los atributos tienen un amplio rango de valores.

Con respecto a *CART* (*Classification and Regression Tree*, Árboles de Clasificación y Regresión), produce árboles binarios. Una característica importante es que es capaz de generar *árboles de regresión*; en este caso, *CART* busca aquellas particiones que minimicen el *error cuadrático medio* y la predicción en cada hoja se realiza mediante una *media ponderada*. Por otra parte, para elegir el atributo que va a particionar en cada nodo del árbol *CART* utiliza el *índice Gini* y, al igual que el *C4.5*, es capaz de manejar atributos categóricos y numéricos, el ruido. También tiene la habilidad de tratar con valores atípicos. No obstante, este algoritmo puede producir árboles inestables y realiza particiones mediante una sola variable.

#### 2.2.4. Redes neuronales

Las **redes neuronales** son un modelo computacional de interconexión de neuronas en una red que colabora para producir un estímulo de salida. El bloque de construcción de este sistema son las *neuronas artificiales*, que son unidades computacionales simples que ponderan las señales de entrada y producen una señal de salida usando una función de activación:

Figura 2.1: Estructura de una red neuronal [phuongphuong 4112716 and 405k4213231585](#)



En primer lugar en el perceptrón multicapa se recibe una serie de **entradas**, que pueden ser características de un conjunto de entrenamiento o salidas de otras neuronas. A continuación, se aplican unos **pesos** a las entradas, que se suelen inicializar a valores aleatorios pequeños, como valores en el rango de 0 a 0.3. Después, las entradas ponderadas se suman junto con un *sesgo* o *bias* que tiene la neurona (se interpreta como una entrada que permite desplazar la función de activación a la izquierda o a la derecha, que siempre tiene el valor 1.0 y que también debe ser ponderada) que, a su vez, pasan a través de una **función de activación**, obteniendo así las **salidas**. Esta función de activación es un simple mapeo de la entrada ponderada sumada a la salida de la neurona; es decir, se utiliza para determinar la salida de

la red neuronal como si o no: mapea los valores resultantes de 0 a 1 o de -1 a 1, etc. (dependiendo de la función). Las distintas funciones de activación se engloban en dos tipos, *funciones de activación lineales* y *funciones de activación no lineales* y existen una gran variedad de ellas: *función sigmoide*, *Tanh*, *ReLU*, etc.

La estructura de una red neuronal se compone de las siguientes partes:

- **Capa de entrada:** La capa que toma la entrada del conjunto de datos se denomina *capa de entrada*, puesto que es la parte que se expone en la red de la red. Éstas no son neuronas como se describió anteriormente, sino que simplemente pasan los valores de las entradas a la siguiente capa.
- **Capas ocultas:** Las capas posteriores a la capa de entrada se denominan *capas ocultas* porque no se exponen directamente a la entrada y están formadas por aquellas neuronas cuyas entradas provienen de capas anteriores y cuyas salidas pasan a neuronas de capas posteriores.
- **Capa de salida:** La última capa de la red neuronal se denomina *capa de salida* y es responsable de producir un valor o un vector de valores que dependerán del problema a resolver. La elección de la función de activación en la capa de salida está fuertemente limitada por el tipo problema que se está modelando.

### Entrenamiento de una red neuronal

El primer paso para entrenar una red neuronal es *preparar los datos*, de tal forma que éstos deben ser numéricos y tienen que estar escalados de manera consistente. Con la *normalización* (reescalar al rango entre 0 y 1) y la *estandarización* (para que la distribución de cada columna tenga la media cero y la desviación estándar de 1, de modo que todas las entradas se expresan en rangos similares), el proceso de entrenamiento de la red neuronal se realiza con mucha mayor velocidad.

Uno de los algoritmos de entrenamiento para redes neuronales más populares se denomina **descenso por gradiente** (*gradient descent*). En este algoritmo la red procesa la entrada hacia delante activando las neuronas a medida que se va avanzando a través de las capas ocultas hasta que finalmente se obtiene un valor de salida; esto se denomina *propagación hacia delante* en la red neuronal. La salida del grafo se compara con la salida esperada y se calcula el error. Este error es entonces propagado de nuevo hacia atrás a través de la red neuronal, una capa a la vez, y los pesos son actualizados de acuerdo a su grado de contribución al error calculado (*algoritmo de retropropagación*).

A la hora de llevar a cabo el entrenamiento de la red neuronal, existen diferentes posibilidades de realizarlo según el número de instancias que se procesen antes de actualizar los pesos de la red neuronal, que está definido por el hiperparámetro denominado **tamaño del lote** (*batch size*). Una de ellas es procesar todo el conjunto de datos (el tamaño del lote es el número de instancias del conjunto de datos), guardar los errores de todos los ejemplos de entrenamiento y actualizar los parámetros de la red (*batch gradient descent*). Otra posibilidad es definir el tamaño del lote a una instancia, de tal manera que los pesos en la red neuronal se pueden actualizar al

procesar un solo ejemplo de entrenamiento (*stochastic gradient descent*). También existe la opción de definir un tamaño de lote que se encuentre entre una sola instancia y todo el conjunto de entrenamiento, del tal forma que se actualizan los pesos tras un número de instancias establecido (*mini-batch gradient descent*).

Por otra parte, existe un hiperparámetro denominado **época** (*epoch*), que establece el número de iteraciones que va a realizar el algoritmo de aprendizaje a través de todo el conjunto de datos de entrenamiento. Una época comprende uno o más lotes. En el caso del *batch gradient descent*, una época tiene un solo lote puesto que se actualizan los pesos tras procesar todo el conjunto de datos, y en el caso del *stochastic gradient descent* una época contiene tantos lotes como ejemplos haya en el conjunto de datos puesto que actualiza los parámetros de la red al procesar una instancia.

Por otra parte, el grado en el que se actualizan los pesos es controlado por un parámetro de configuración denominado **velocidad de aprendizaje** (*learning rate*). Este parámetro controla el cambio realizado en el peso de la red neuronal para un error determinado. A menudo se utilizan tamaños de peso pequeños tales como 0.1, 0.01 o más pequeños.

Una vez que una red neuronal ha sido entrenada puede ser usada para realizar predicciones. La topología de la red neuronal y el conjunto final de pesos es todo lo que se necesita para implantar el modelo. Las predicciones se realizan proporcionando la entrada a la red y ejecutando una propagación hacia delante que genera una salida que se utiliza como predicción.

### 2.2.5. $k$ -Vecinos más cercanos

El paradigma clasificatorio denominado  **$k$ -Vecinos más Cercanos** (*K-Nearest Neighbours*, K-NN) se fundamenta en la idea de identificar el grupo de  $k$  objetos en el conjunto de datos de entrenamiento que más cerca está de un nuevo objeto a clasificar, de manera que a este nuevo caso se le asigna la etiqueta de clase más frecuente en ese grupo de  $k$  objetos. Este método de clasificación supervisada se basa en tres componentes principales: un *conjunto de datos de entrenamiento etiquetados*, una *métrica de distancia* para calcular las distancias entre distintos objetos y el número  $k$  de vecinos más cercanos. De esta forma, utilizando la métrica de distancia correspondiente, se calcula la distancia entre el nuevo caso a clasificar y los casos etiquetados para averiguar cuáles son las  $k$  instancias etiquetas que más próximas están al nuevo objeto y, una vez ejecutado este paso, se calcula la clase más repetida en el grupo de  $k$  casos más cercanos y se le asigna al nuevo objeto.

En la siguiente figura se presenta un pseudocódigo para el clasificador KNN básico:

---

Figura 2.2: Pseudocódigo del algoritmo KNN. Fuente: [Larranaga et al.](#)

COMIENZO

Entrada:  $D = \{(\mathbf{x}^{(1)}, c^{(1)}), \dots, (\mathbf{x}^{(N)}, c^{(N)})\}$  $\mathbf{x} = (x_1, \dots, x_n)$  nuevo caso a clasificarPARA todo objeto ya clasificado  $(\mathbf{x}^{(i)}, c^{(i)})$ calcular  $d^{(i)} = d(\mathbf{x}^{(i)}, \mathbf{x})$ Ordenar  $d^{(i)} (i = 1, \dots, N)$  en orden ascendenteQuedarnos con los  $K$  casos  $D_{\mathbf{x}}^K$  ya clasificados más cercanos a  $\mathbf{x}$ Asignar a  $\mathbf{x}$  la clase más frecuente en  $D_{\mathbf{x}}^K$ 

FIN

donde:

- $D_{\mathbf{x}}^K$  es el conjunto de casos de tamaño  $K$  más cercano a la instancia  $\mathbf{x}$
- $d^{(i)}$  es la distancia entre el caso clasificado  $i$  y la nueva instancia a clasificar  $\mathbf{x}$

En este algoritmo puede ocurrir que, a la hora de averiguar la clase a la que se va a clasificar una nueva instancia, dos o más clases obtengan el mismo número de votos, por lo que hay que establecer *reglas de desempate*. Algunas de ellas pueden ser elegir aquella clase que tenga menor distancia media, aquella a la que pertenezca el vecino más cercano, etcétera ([Mohanty et al. \[2015\]](#)). Además, todos los datos deben estar *normalizados* para evitar que las características en el conjunto de entrada con valores más altos dominen el cálculo de la distancia.

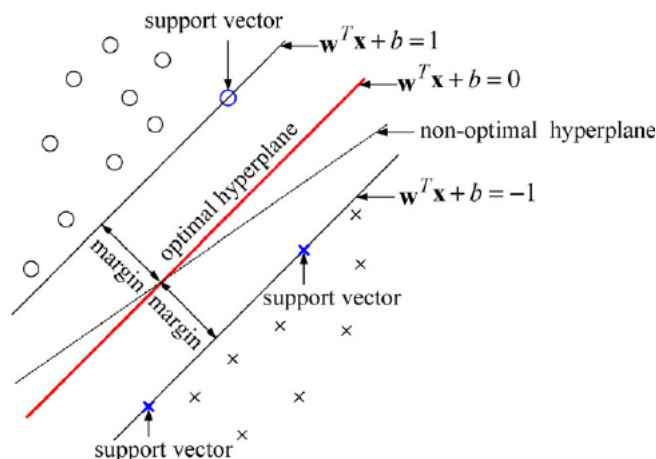
El algoritmo KNN lleva a cabo un **aprendizaje vago** (*lazy learning*) puesto que no aprende un modelo discriminativo de los datos de entrenamiento (*eager learning*), sino que memoriza el conjunto de datos de entrenamiento y la labor de predicción se realiza cuando llega un nuevo caso a clasificar. Por lo tanto, en contraste con otros paradigmas clasificatorios, donde hay un proceso de construcción del modelo de predicción y posteriormente su aplicación sobre nuevas instancias, el algoritmo KNN engloba estos dos pasos en uno. Además, es un algoritmo **no paramétrico** puesto que no hace ninguna suposición sobre la distribución de datos subyacente en el conjunto de datos de entrenamiento. Un aspecto importante de este paradigma clasificatorio es el número  $K$  de vecinos que se va a utilizar para decidir la clase a la que pertenece una nueva instancia. Una de las opciones para establecer este número es hacerlo de forma *fija*, y se ha constatado empíricamente que la proporción de casos clasificados correctamente es no monótono con respecto a  $K$ , de manera que el rendimiento del clasificador no aumenta siempre al incrementar  $K$ ; un valor adecuado sería entre 3 y 7 vecinos ([Larranaga et al.](#)). No obstante, no es práctico asignar un valor fijo a este número para todos los nuevos casos a clasificar, sino modificarlo en función de las características de cada uno de ellos ([Wang et al. \[2006\]](#), [Cheng et al. \[2014\]](#), [Zhang et al. \[2018\]](#)).

Con respecto al algoritmo básico del KNN, se han propuesto diferentes variantes con el objetivo de mejorar su rendimiento. Uno de ellos es el **KNN con rechazo**, en el que se tienen que cumplir una serie de condiciones que garanticen la clasificación del nuevo caso, como puede ser una mayoría absoluta de una determinada clase, o que supere un determinado umbral de votos. Otras variantes del KNN básico son **KNN con pesado de casos seleccionados**, en el que se le da más importancia a unas instancias que a otras a la hora de realizar la clasificación según la cercanía que tengan con el nuevo caso a predecir; **KNN con pesados de variables**, en el que se le da más relevancia a ciertas variables predictoras que a otras a la hora de calcular las distancias entre los casos clasificados y la nueva instancia a clasificar; etcétera.

### 2.2.6. Máquinas de vector soporte

Las **máquinas de vector soporte** (*Support Vector Machines*, SVMs son un algoritmo de aprendizaje supervisado que puede utilizarse tanto para desafíos de clasificación como de regresión; no obstante, se utiliza principalmente en problemas de clasificación. En este algoritmo trazamos cada elemento de datos (cada muestra de entrenamiento) como un punto en el espacio  $n$ -dimensional (donde  $n$  es el número de características del problema a resolver), con el valor de cada característica mapeándose al valor de una determinada coordenada. A continuación, dado este conjunto de ejemplos de entrenamiento, cada uno perteneciente a una clase, entrenamos una SVM para construir un modelo que prediga la clase de una nueva muestra mediante la construcción de un **hiperplano** que separe las clases de los datos de entrenamiento y maximice el margen entre esas clases (maximice la distancia entre los puntos más cercanos de cada clase al hiperplano de separación óptimo, llamados vectores soporte) en el espacio  $n$ -dimensional:

Figura 2.3: SVM. Fuente: [Unagar and Unagar \[2017\]](#)





### 2.2.7. Regresión logística

### 2.2.8. Métodos combinados de aprendizaje

Quisque ullamcorper placerat ipsum. Cras nibh. Morbi vel justo vitae lacus tincidunt ultrices. Lorem ipsum dolor sit amet, consectetur adipiscing elit. In hac habitasse platea dictumst. Integer tempus convallis augue. Etiam facilisis. Nunc elementum fermentum wisi. Aenean placerat. Ut imperdiet, enim sed gravida sollicitudin, felis odio placerat quam, ac pulvinar elit purus eget enim. Nunc vitae tortor. Proin tempus nibh sit amet nisl. Vivamus quis tortor vitae risus porta vehicula.

Fusce mauris. Vestibulum luctus nibh at lectus. Sed bibendum, nulla a faucibus semper, leo velit ultricies tellus, ac venenatis arcu wisi vel nisl. Vestibulum diam. Aliquam pellentesque, augue quis sagittis posuere, turpis lacus congue quam, in hendrerit risus eros eget felis. Maecenas eget erat in sapien mattis porttitor. Vestibulum porttitor. Nulla facilisi. Sed a turpis eu lacus commodo facilisis. Morbi fringilla, wisi in dignissim interdum, justo lectus sagittis dui, et vehicula libero dui cursus dui. Mauris tempor ligula sed lacus. Duis cursus enim ut augue. Cras ac magna. Cras nulla. Nulla egestas. Curabitur a leo. Quisque egestas wisi eget nunc. Nam feugiat lacus vel est. Curabitur consectetur.

## 2.3. Algoritmos de aprendizaje no supervisado

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetur id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

Nam dui ligula, fringilla a, euismod sodales, sollicitudin vel, wisi. Morbi auctor lorem non justo. Nam lacus libero, pretium at, lobortis vitae, ultricies et, tellus. Donec aliquet, tortor sed accumsan bibendum, erat ligula aliquet magna, vitae ornare odio metus a mi. Morbi ac orci et nisl hendrerit mollis. Suspendisse ut massa. Cras nec ante. Pellentesque a nulla. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Aliquam tincidunt urna. Nulla ullamcorper vestibulum turpis. Pellentesque cursus luctus mauris.

Nulla malesuada porttitor diam. Donec felis erat, congue non, volutpat at, tincidunt tristique, libero. Vivamus viverra fermentum felis. Donec nonummy pellentesque ante. Phasellus adipiscing semper elit. Proin fermentum massa ac quam. Sed diam turpis, molestie vitae, placerat a, molestie nec, leo. Maecenas lacinia. Nam ip-

sum ligula, eleifend at, accumsan nec, suscipit a, ipsum. Morbi blandit ligula feugiat magna. Nunc eleifend consequat lorem. Sed lacinia nulla vitae enim. Pellentesque tincidunt purus vel magna. Integer non enim. Praesent euismod nunc eu purus. Donec bibendum quam in tellus. Nullam cursus pulvinar lectus. Donec et mi. Nam vulputate metus eu enim. Vestibulum pellentesque felis eu massa.

### 2.3.1. Agrupamiento

Quisque ullamcorper placerat ipsum. Cras nibh. Morbi vel justo vitae lacus tincidunt ultrices. Lorem ipsum dolor sit amet, consectetur adipiscing elit. In hac habitasse platea dictumst. Integer tempus convallis augue. Etiam facilisis. Nunc elementum fermentum wisi. Aenean placerat. Ut imperdiet, enim sed gravida sollicitudin, felis odio placerat quam, ac pulvinar elit purus eget enim. Nunc vitae tortor. Proin tempus nibh sit amet nisl. Vivamus quis tortor vitae risus porta vehicula.

Fusce mauris. Vestibulum luctus nibh at lectus. Sed bibendum, nulla a faucibus semper, leo velit ultricies tellus, ac venenatis arcu wisi vel nisl. Vestibulum diam. Aliquam pellentesque, augue quis sagittis posuere, turpis lacus congue quam, in hendrerit risus eros eget felis. Maecenas eget erat in sapien mattis porttitor. Vestibulum porttitor. Nulla facilisi. Sed a turpis eu lacus commodo facilisis. Morbi fringilla, wisi in dignissim interdum, justo lectus sagittis dui, et vehicula libero dui cursus dui. Mauris tempor ligula sed lacus. Duis cursus enim ut augue. Cras ac magna. Cras nulla. Nulla egestas. Curabitur a leo. Quisque egestas wisi eget nunc. Nam feugiat lacus vel est. Curabitur consectetur.

## 2.4. Series temporales

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetur id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

Nam dui ligula, fringilla a, euismod sodales, sollicitudin vel, wisi. Morbi auctor lorem non justo. Nam lacus libero, pretium at, lobortis vitae, ultricies et, tellus. Donec aliquet, tortor sed accumsan bibendum, erat ligula aliquet magna, vitae ornare odio metus a mi. Morbi ac orci et nisl hendrerit mollis. Suspendisse ut massa. Cras nec ante. Pellentesque a nulla. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Aliquam tincidunt urna. Nulla ullamcorper vestibulum turpis. Pellentesque cursus luctus mauris.

Nulla malesuada porttitor diam. Donec felis erat, congue non, volutpat at, tincidunt tristique, libero. Vivamus viverra fermentum felis. Donec nonummy pellentesque ante. Phasellus adipiscing semper elit. Proin fermentum massa ac quam. Sed diam turpis, molestie vitae, placerat a, molestie nec, leo. Maecenas lacinia. Nam ipsum ligula, eleifend at, accumsan nec, suscipit a, ipsum. Morbi blandit ligula feugiat magna. Nunc eleifend consequat lorem. Sed lacinia nulla vitae enim. Pellentesque tincidunt purus vel magna. Integer non enim. Praesent euismod nunc eu purus. Donec bibendum quam in tellus. Nullam cursus pulvinar lectus. Donec et mi. Nam vulputate metus eu enim. Vestibulum pellentesque felis eu massa.

## 2.5. Redes bayesianas para el descubrimiento de conocimiento

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetur id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

Nam dui ligula, fringilla a, euismod sodales, sollicitudin vel, wisi. Morbi auctor lorem non justo. Nam lacus libero, pretium at, lobortis vitae, ultricies et, tellus. Donec aliquet, tortor sed accumsan bibendum, erat ligula aliquet magna, vitae ornare odio metus a mi. Morbi ac orci et nisl hendrerit mollis. Suspendisse ut massa. Cras nec ante. Pellentesque a nulla. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Aliquam tincidunt urna. Nulla ullamcorper vestibulum turpis. Pellentesque cursus luctus mauris.

Nulla malesuada porttitor diam. Donec felis erat, congue non, volutpat at, tincidunt tristique, libero. Vivamus viverra fermentum felis. Donec nonummy pellentesque ante. Phasellus adipiscing semper elit. Proin fermentum massa ac quam. Sed diam turpis, molestie vitae, placerat a, molestie nec, leo. Maecenas lacinia. Nam ipsum ligula, eleifend at, accumsan nec, suscipit a, ipsum. Morbi blandit ligula feugiat magna. Nunc eleifend consequat lorem. Sed lacinia nulla vitae enim. Pellentesque tincidunt purus vel magna. Integer non enim. Praesent euismod nunc eu purus. Donec bibendum quam in tellus. Nullam cursus pulvinar lectus. Donec et mi. Nam vulputate metus eu enim. Vestibulum pellentesque felis eu massa.

Durante la elaboración de la memoria del Trabajo Fin de Máster, el alumno deberá incluir toda la información que considere necesaria y útil para la descripción y justificación del trabajo desarrollado y de los resultados obtenidos. Además y con

objeto de asegurar que ha cubierto todas las competencias transversales asociadas al mismo, es obligatorio que en los diferentes apartados de la memoria se desarrollen cuidadosamente los aspectos indicados a continuación.

*Introducción y Objetivos:* Justificar la necesidad de desarrollar la tesis en lugar de adquirir o aplicar directamente lo existente a lo largo de la gama de categorías de procesos, productos y servicios de la empresa o institución usuaria de la tesis.

*Estado del arte:* Demostrar que se ha comprendido como es el ámbito de negocio donde se enmarca la tesis, sus hábitos y necesidades de productos o servicios tecnológicos.

*Evaluación de Riesgos:* Demostrar que se han considerado diferentes soluciones tanto clásicas como novedosas o innovadoras al problema, y se han evaluado los riesgos y ventajas de cada una de ellas.

*Resultados:* Justificar que la tecnología resultante de la tesis satisface los deseos o necesidades del cliente (real, potencial o ficticio).

*Conclusiones:* Establecer las conclusiones del trabajo apoyándose fundamentalmente en los datos y observaciones obtenidas durante su desarrollo. Discutir que medios, cauces, etapas y tecnologías harían falta (si procede) para llevar a cabo una implantación real de los resultados.

*Líneas futuras:* Discutir los límites de las tecnologías actuales aplicadas al problema, planteando líneas de I+D+i realistas y capaces de superarlos.

## Capítulo 3

# Estado del arte: Aprendizaje automático para flujos de datos

### 3.1. Introducción

On-Board Mining of Data Streams in Sensor Networks <http://i2pc.es/coss/Docencia/SignalP>  
Cost-Sensitive Perceptron Decision Trees for Imbalanced Drifting Data Streams Incremental Decision Tree based on order statistics Learning Decision Trees from Data Streams with Concept Drift Fast Feature Selection for Naive Bayes Classification in Data Stream Mining Making Data Stream Classification Tree-based Ensembles Lighter (2018)

#### 3.1.1. Conceptos

Quisque ullamcorper placerat ipsum. Cras nibh. Morbi vel justo vitae lacus tincidunt ultrices. Lorem ipsum dolor sit amet, consectetur adipiscing elit. In hac habitasse platea dictumst. Integer tempus convallis augue. Etiam facilisis. Nunc elementum fermentum wisi. Aenean placerat. Ut imperdiet, enim sed gravida sollicitudin, felis odio placerat quam, ac pulvinar elit purus eget enim. Nunc vitae tortor. Proin tempus nibh sit amet nisl. Vivamus quis tortor vitae risus porta vehicula.

Fusce mauris. Vestibulum luctus nibh at lectus. Sed bibendum, nulla a faucibus semper, leo velit ultricies tellus, ac venenatis arcu wisi vel nisl. Vestibulum diam. Aliquam pellentesque, augue quis sagittis posuere, turpis lacus congue quam, in hendrerit risus eros eget felis. Maecenas eget erat in sapien mattis porttitor. Vestibulum porttitor. Nulla facilisi. Sed a turpis eu lacus commodo facilisis. Morbi fringilla, wisi in dignissim interdum, justo lectus sagittis dui, et vehicula libero dui cursus dui. Mauris tempor ligula sed lacus. Duis cursus enim ut augue. Cras ac magna. Cras nulla. Nulla egestas. Curabitur a leo. Quisque egestas wisi eget nunc. Nam feugiat lacus vel est. Curabitur consectetur.

## 3.2. Algoritmos de aprendizaje supervisado

A review of supervised machine learning algorithms

### 3.2.1. Clasificadores bayesianos

Quisque ullamcorper placerat ipsum. Cras nibh. Morbi vel justo vitae lacus tincidunt ultrices. Lorem ipsum dolor sit amet, consectetur adipiscing elit. In hac habitasse platea dictumst. Integer tempus convallis augue. Etiam facilisis. Nunc elementum fermentum wisi. Aenean placerat. Ut imperdiet, enim sed gravida sollicitudin, felis odio placerat quam, ac pulvinar elit purus eget enim. Nunc vitae tortor. Proin tempus nibh sit amet nisl. Vivamus quis tortor vitae risus porta vehicula.

Fusce mauris. Vestibulum luctus nibh at lectus. Sed bibendum, nulla a faucibus semper, leo velit ultricies tellus, ac venenatis arcu wisi vel nisl. Vestibulum diam. Aliquam pellentesque, augue quis sagittis posuere, turpis lacus congue quam, in hendrerit risus eros eget felis. Maecenas eget erat in sapien mattis porttitor. Vestibulum porttitor. Nulla facilisi. Sed a turpis eu lacus commodo facilisis. Morbi fringilla, wisi in dignissim interdum, justo lectus sagittis dui, et vehicula libero dui cursus dui. Mauris tempor ligula sed lacus. Duis cursus enim ut augue. Cras ac magna. Cras nulla. Nulla egestas. Curabitur a leo. Quisque egestas wisi eget nunc. Nam feugiat lacus vel est. Curabitur consectetur.

### 3.2.2. Inducción de reglas

Quisque ullamcorper placerat ipsum. Cras nibh. Morbi vel justo vitae lacus tincidunt ultrices. Lorem ipsum dolor sit amet, consectetur adipiscing elit. In hac habitasse platea dictumst. Integer tempus convallis augue. Etiam facilisis. Nunc elementum fermentum wisi. Aenean placerat. Ut imperdiet, enim sed gravida sollicitudin, felis odio placerat quam, ac pulvinar elit purus eget enim. Nunc vitae tortor. Proin tempus nibh sit amet nisl. Vivamus quis tortor vitae risus porta vehicula.

Fusce mauris. Vestibulum luctus nibh at lectus. Sed bibendum, nulla a faucibus semper, leo velit ultricies tellus, ac venenatis arcu wisi vel nisl. Vestibulum diam. Aliquam pellentesque, augue quis sagittis posuere, turpis lacus congue quam, in hendrerit risus eros eget felis. Maecenas eget erat in sapien mattis porttitor. Vestibulum porttitor. Nulla facilisi. Sed a turpis eu lacus commodo facilisis. Morbi fringilla, wisi in dignissim interdum, justo lectus sagittis dui, et vehicula libero dui cursus dui. Mauris tempor ligula sed lacus. Duis cursus enim ut augue. Cras ac magna. Cras nulla. Nulla egestas. Curabitur a leo. Quisque egestas wisi eget nunc. Nam feugiat lacus vel est. Curabitur consectetur.

### 3.2.3. Árboles de decisión

A New Method for Data Stream Mining Based on the Misclassification Error (2014)

### 3.2.4. Redes neuronales

Quisque ullamcorper placerat ipsum. Cras nibh. Morbi vel justo vitae lacus tincidunt ultrices. Lorem ipsum dolor sit amet, consectetur adipiscing elit. In hac habitasse platea dictumst. Integer tempus convallis augue. Etiam facilisis. Nunc elementum fermentum wisi. Aenean placerat. Ut imperdiet, enim sed gravida sollicitudin, felis odio placerat quam, ac pulvinar elit purus eget enim. Nunc vitae tortor. Proin tempus nibh sit amet nisl. Vivamus quis tortor vitae risus porta vehicula.

Fusce mauris. Vestibulum luctus nibh at lectus. Sed bibendum, nulla a faucibus semper, leo velit ultricies tellus, ac venenatis arcu wisi vel nisl. Vestibulum diam. Aliquam pellentesque, augue quis sagittis posuere, turpis lacus congue quam, in hendrerit risus eros eget felis. Maecenas eget erat in sapien mattis porttitor. Vestibulum porttitor. Nulla facilisi. Sed a turpis eu lacus commodo facilisis. Morbi fringilla, wisi in dignissim interdum, justo lectus sagittis dui, et vehicula libero dui cursus dui. Mauris tempor ligula sed lacus. Duis cursus enim ut augue. Cras ac magna. Cras nulla. Nulla egestas. Curabitur a leo. Quisque egestas wisi eget nunc. Nam feugiat lacus vel est. Curabitur consectetur.

### 3.2.5. $k$ -Vecinos más cercanos

Quisque ullamcorper placerat ipsum. Cras nibh. Morbi vel justo vitae lacus tincidunt ultrices. Lorem ipsum dolor sit amet, consectetur adipiscing elit. In hac habitasse platea dictumst. Integer tempus convallis augue. Etiam facilisis. Nunc elementum fermentum wisi. Aenean placerat. Ut imperdiet, enim sed gravida sollicitudin, felis odio placerat quam, ac pulvinar elit purus eget enim. Nunc vitae tortor. Proin tempus nibh sit amet nisl. Vivamus quis tortor vitae risus porta vehicula.

Fusce mauris. Vestibulum luctus nibh at lectus. Sed bibendum, nulla a faucibus semper, leo velit ultricies tellus, ac venenatis arcu wisi vel nisl. Vestibulum diam. Aliquam pellentesque, augue quis sagittis posuere, turpis lacus congue quam, in hendrerit risus eros eget felis. Maecenas eget erat in sapien mattis porttitor. Vestibulum porttitor. Nulla facilisi. Sed a turpis eu lacus commodo facilisis. Morbi fringilla, wisi in dignissim interdum, justo lectus sagittis dui, et vehicula libero dui cursus dui. Mauris tempor ligula sed lacus. Duis cursus enim ut augue. Cras ac magna. Cras nulla. Nulla egestas. Curabitur a leo. Quisque egestas wisi eget nunc. Nam feugiat lacus vel est. Curabitur consectetur.

### 3.2.6. Máquinas de vector soporte

Quisque ullamcorper placerat ipsum. Cras nibh. Morbi vel justo vitae lacus tincidunt ultrices. Lorem ipsum dolor sit amet, consectetur adipiscing elit. In hac habitasse platea dictumst. Integer tempus convallis augue. Etiam facilisis. Nunc elementum fermentum wisi. Aenean placerat. Ut imperdiet, enim sed gravida sollicitudin, felis odio placerat quam, ac pulvinar elit purus eget enim. Nunc vitae tortor. Proin tempus nibh sit amet nisl. Vivamus quis tortor vitae risus porta vehicula.

Fusce mauris. Vestibulum luctus nibh at lectus. Sed bibendum, nulla a faucibus semper, leo velit ultricies tellus, ac venenatis arcu wisi vel nisl. Vestibulum diam. Aliquam pellentesque, augue quis sagittis posuere, turpis lacus congue quam, in hendrerit risus eros eget felis. Maecenas eget erat in sapien mattis porttitor. Vestibulum porttitor. Nulla facilisi. Sed a turpis eu lacus commodo facilisis. Morbi fringilla, wisi in dignissim interdum, justo lectus sagittis dui, et vehicula libero dui cursus dui. Mauris tempor ligula sed lacus. Duis cursus enim ut augue. Cras ac magna. Cras nulla. Nulla egestas. Curabitur a leo. Quisque egestas wisi eget nunc. Nam feugiat lacus vel est. Curabitur consectetur.

### 3.2.7. Regresión logística

Quisque ullamcorper placerat ipsum. Cras nibh. Morbi vel justo vitae lacus tincidunt ultrices. Lorem ipsum dolor sit amet, consectetur adipiscing elit. In hac habitasse platea dictumst. Integer tempus convallis augue. Etiam facilisis. Nunc elementum fermentum wisi. Aenean placerat. Ut imperdiet, enim sed gravida sollicitudin, felis odio placerat quam, ac pulvinar elit purus eget enim. Nunc vitae tortor. Proin tempus nibh sit amet nisl. Vivamus quis tortor vitae risus porta vehicula.

Fusce mauris. Vestibulum luctus nibh at lectus. Sed bibendum, nulla a faucibus semper, leo velit ultricies tellus, ac venenatis arcu wisi vel nisl. Vestibulum diam. Aliquam pellentesque, augue quis sagittis posuere, turpis lacus congue quam, in hendrerit risus eros eget felis. Maecenas eget erat in sapien mattis porttitor. Vestibulum porttitor. Nulla facilisi. Sed a turpis eu lacus commodo facilisis. Morbi fringilla, wisi in dignissim interdum, justo lectus sagittis dui, et vehicula libero dui cursus dui. Mauris tempor ligula sed lacus. Duis cursus enim ut augue. Cras ac magna. Cras nulla. Nulla egestas. Curabitur a leo. Quisque egestas wisi eget nunc. Nam feugiat lacus vel est. Curabitur consectetur.

### 3.2.8. Métodos combinados de aprendizaje

Quisque ullamcorper placerat ipsum. Cras nibh. Morbi vel justo vitae lacus tincidunt ultrices. Lorem ipsum dolor sit amet, consectetur adipiscing elit. In hac habitasse platea dictumst. Integer tempus convallis augue. Etiam facilisis. Nunc elementum fermentum wisi. Aenean placerat. Ut imperdiet, enim sed gravida sollicitudin, felis odio placerat quam, ac pulvinar elit purus eget enim. Nunc vitae tortor. Proin tempus nibh sit amet nisl. Vivamus quis tortor vitae risus porta vehicula.

Fusce mauris. Vestibulum luctus nibh at lectus. Sed bibendum, nulla a faucibus semper, leo velit ultricies tellus, ac venenatis arcu wisi vel nisl. Vestibulum diam. Aliquam pellentesque, augue quis sagittis posuere, turpis lacus congue quam, in hendrerit risus eros eget felis. Maecenas eget erat in sapien mattis porttitor. Vestibulum porttitor. Nulla facilisi. Sed a turpis eu lacus commodo facilisis. Morbi fringilla, wisi in dignissim interdum, justo lectus sagittis dui, et vehicula libero dui cursus dui. Mauris tempor ligula sed lacus. Duis cursus enim ut augue. Cras ac magna. Cras nulla. Nulla egestas. Curabitur a leo. Quisque egestas wisi eget nunc. Nam feugiat



lacus vel est. Curabitur consectetur.

### 3.3. Algoritmos de aprendizaje no supervisado

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetur id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

Nam dui ligula, fringilla a, euismod sodales, sollicitudin vel, wisi. Morbi auctor lorem non justo. Nam lacus libero, pretium at, lobortis vitae, ultricies et, tellus. Donec aliquet, tortor sed accumsan bibendum, erat ligula aliquet magna, vitae ornare odio metus a mi. Morbi ac orci et nisl hendrerit mollis. Suspendisse ut massa. Cras nec ante. Pellentesque a nulla. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Aliquam tincidunt urna. Nulla ullamcorper vestibulum turpis. Pellentesque cursus luctus mauris.

Nulla malesuada porttitor diam. Donec felis erat, congue non, volutpat at, tincidunt tristique, libero. Vivamus viverra fermentum felis. Donec nonummy pellentesque ante. Phasellus adipiscing semper elit. Proin fermentum massa ac quam. Sed diam turpis, molestie vitae, placerat a, molestie nec, leo. Maecenas lacinia. Nam ipsum ligula, eleifend at, accumsan nec, suscipit a, ipsum. Morbi blandit ligula feugiat magna. Nunc eleifend consequat lorem. Sed lacinia nulla vitae enim. Pellentesque tincidunt purus vel magna. Integer non enim. Praesent euismod nunc eu purus. Donec bibendum quam in tellus. Nullam cursus pulvinar lectus. Donec et mi. Nam vulputate metus eu enim. Vestibulum pellentesque felis eu massa.

#### 3.3.1. Agrupamiento

Quisque ullamcorper placerat ipsum. Cras nibh. Morbi vel justo vitae lacus tincidunt ultrices. Lorem ipsum dolor sit amet, consectetur adipiscing elit. In hac habitasse platea dictumst. Integer tempus convallis augue. Etiam facilisis. Nunc elementum fermentum wisi. Aenean placerat. Ut imperdiet, enim sed gravida sollicitudin, felis odio placerat quam, ac pulvinar elit purus eget enim. Nunc vitae tortor. Proin tempus nibh sit amet nisl. Vivamus quis tortor vitae risus porta vehicula.

Fusce mauris. Vestibulum luctus nibh at lectus. Sed bibendum, nulla a faucibus semper, leo velit ultricies tellus, ac venenatis arcu wisi vel nisl. Vestibulum diam. Aliquam pellentesque, augue quis sagittis posuere, turpis lacus congue quam, in hendrerit risus eros eget felis. Maecenas eget erat in sapien mattis porttitor. Vestibulum

porttitor. Nulla facilisi. Sed a turpis eu lacus commodo facilisis. Morbi fringilla, wisi in dignissim interdum, justo lectus sagittis dui, et vehicula libero dui cursus dui. Mauris tempor ligula sed lacus. Duis cursus enim ut augue. Cras ac magna. Cras nulla. Nulla egestas. Curabitur a leo. Quisque egestas wisi eget nunc. Nam feugiat lacus vel est. Curabitur consectetur.

### 3.4. Series temporales

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetur id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

Nam dui ligula, fringilla a, euismod sodales, sollicitudin vel, wisi. Morbi auctor lorem non justo. Nam lacus libero, pretium at, lobortis vitae, ultricies et, tellus. Donec aliquet, tortor sed accumsan bibendum, erat ligula aliquet magna, vitae ornare odio metus a mi. Morbi ac orci et nisl hendrerit mollis. Suspendisse ut massa. Cras nec ante. Pellentesque a nulla. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Aliquam tincidunt urna. Nulla ullamcorper vestibulum turpis. Pellentesque cursus luctus mauris.

Nulla malesuada porttitor diam. Donec felis erat, congue non, volutpat at, tincidunt tristique, libero. Vivamus viverra fermentum felis. Donec nonummy pellentesque ante. Phasellus adipiscing semper elit. Proin fermentum massa ac quam. Sed diam turpis, molestie vitae, placerat a, molestie nec, leo. Maecenas lacinia. Nam ipsum ligula, eleifend at, accumsan nec, suscipit a, ipsum. Morbi blandit ligula feugiat magna. Nunc eleifend consequat lorem. Sed lacinia nulla vitae enim. Pellentesque tincidunt purus vel magna. Integer non enim. Praesent euismod nunc eu purus. Donec bibendum quam in tellus. Nullam cursus pulvinar lectus. Donec et mi. Nam vulputate metus eu enim. Vestibulum pellentesque felis eu massa.

### 3.5. Redes bayesianas para el descubrimiento de conocimiento

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetur id, vulputate a, magna. Donec vehicula au-

gue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

Nam dui ligula, fringilla a, euismod sodales, sollicitudin vel, wisi. Morbi auctor lorem non justo. Nam lacus libero, pretium at, lobortis vitae, ultricies et, tellus. Donec aliquet, tortor sed accumsan bibendum, erat ligula aliquet magna, vitae ornare odio metus a mi. Morbi ac orci et nisl hendrerit mollis. Suspendisse ut massa. Cras nec ante. Pellentesque a nulla. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Aliquam tincidunt urna. Nulla ullamcorper vestibulum turpis. Pellentesque cursus luctus mauris.

Nulla malesuada porttitor diam. Donec felis erat, congue non, volutpat at, tincidunt tristique, libero. Vivamus viverra fermentum felis. Donec nonummy pellentesque ante. Phasellus adipiscing semper elit. Proin fermentum massa ac quam. Sed diam turpis, molestie vitae, placerat a, molestie nec, leo. Maecenas lacinia. Nam ipsum ligula, eleifend at, accumsan nec, suscipit a, ipsum. Morbi blandit ligula feugiat magna. Nunc eleifend consequat lorem. Sed lacinia nulla vitae enim. Pellentesque tincidunt purus vel magna. Integer non enim. Praesent euismod nunc eu purus. Donec bibendum quam in tellus. Nullam cursus pulvinar lectus. Donec et mi. Nam vulputate metus eu enim. Vestibulum pellentesque felis eu massa.

*Introducción y Objetivos:* Justificar la necesidad de desarrollar la tesis en lugar de adquirir o aplicar directamente lo existente a lo largo de la gama de categorías de procesos, productos y servicios de la empresa o institución usuaria de la tesis.

*Estado del arte:* Demostrar que se ha comprendido como es el ámbito de negocio donde se enmarca la tesis, sus hábitos y necesidades de productos o servicios tecnológicos.

*Evaluación de Riesgos:* Demostrar que se han considerado diferentes soluciones tanto clásicas como novedosas o innovadoras al problema, y se han evaluado los riesgos y ventajas de cada una de ellas.

*Resultados:* Justificar que la tecnología resultante de la tesis satisface los deseos o necesidades del cliente (real, potencial o ficticio).

*Conclusiones:* Establecer las conclusiones del trabajo apoyándose fundamentalmente en los datos y observaciones obtenidas durante su desarrollo. Discutir que medios, cauces, etapas y tecnologías harían falta (si procede) para llevar a cabo una implantación real de los resultados.

*Líneas futuras:* Discutir los límites de las tecnologías actuales aplicadas al problema, planteando líneas de I+D+i realistas y capaces de superarlos.

## Capítulo 4

# Conclusiones y líneas futuras de trabajo

Establecer las conclusiones del trabajo apoyándose fundamentalmente en los datos y observaciones obtenidas durante su desarrollo. Discutir que medios, cauces, etapas y tecnologías harían falta (si procede) para llevar a cabo una implantación real de los resultados. Discutir los límites de las tecnologías actuales aplicadas al problema, planteando líneas de I+D+i realistas y capaces de superarlos.



# Capítulo 5

## Sobre las referencias

La bibliografía o referencias deben aparecer siempre al final de la tesis, incluso en aquellos casos donde se hayan utilizado notas finales. La bibliografía debe incluir los materiales utilizados, incluida la edición, para que la cita pueda ser fácilmente verificada.

*Citar dentro del texto:* Las fuentes consultadas se describen brevemente dentro del texto y estas citas cortas se amplían en una lista de referencias final, en la que se ofrece la información bibliográfica completa.

La cita dentro del texto es una referencia corta que permite identificar la publicación de dónde se ha extraído una frase o parafraseado una idea, e indica la localización precisa dentro de la publicación fuente. Esta cita informa del apellido del autor, la fecha de publicación y la página (o páginas) y se redacta de la forma que puede verse a través de los siguientes ejemplos:

Cuando se citan las palabras exactas del autor deben presentarse entre comillas e indicarse, tras el apellido del autor y, entre paréntesis, la fecha de publicación de la obra citada, seguida de la/s página/s.

Si lo que se reproduce es la idea de un autor (no sus palabras exactas) no se pondrán comillas y se indicará, entre paréntesis, el apellido del autor seguido de la fecha de publicación de la obra a la que se refiere.

No se puede eliminar una parte del texto citado sin señalarse; debe indicarse siempre con puntos suspensivos entre corchetes [...]

Ejemplos de como citar una referencia en el texto son los siguientes ? o ???.  
[Oguntimilehin and Ademola \[2014\]](#)

*Cómo ordenar las referencias:*

1. Las referencias bibliográficas deben presentarse ordenadas alfabéticamente por el apellido del autor, o del primer autor en caso de que sean varios.
2. Si un autor tiene varias obras se ordenarán por orden de aparición.
3. Si de un mismo autor existen varias referencias de un mismo año se especificarán los años seguidos de una letra minúscula y se ordenarán alfabéticamente.

4. Si son trabajos de un autor en colaboración con otros autores, el orden vendrá indicado por el apellido del segundo autor, independientemente del año de publicación. Las publicaciones individuales se colocan antes de las obras en colaboración.

*Cómo citar un artículo de revista:* Un artículo de revista, siguiendo las normas de la APA, se cita de acuerdo con el siguiente esquema general: Apellido(s), Iniciales del nombre o nombres. (Año de publicación). Título del artículo. Título de la revista en cursiva, volumen de la revista (número del fascículo entre paréntesis), primera página- última página del artículo.

*Cómo citar una monografía/libro:* Las monografías, siguiendo las normas de la APA, se citan de acuerdo con el siguiente esquema general: Apellido(s), Iniciales del nombre. (Año de publicación). Título del libro en cursiva. Lugar de publicación: Editorial. Opcionalmente podremos poner la mención de edición, que irá entre paréntesis a continuación del título; y, si fuera el caso el volumen que irá en cursiva.

*Cómo citar un capítulo de un libro:* Los capítulos de los libros se citan de acuerdo con el siguiente esquema general: Apellido(s), Iniciales del nombre o nombres. (Año). Título del capítulo. En A. A. Apellido(s) Editor A, B. B. Apellido(s) Editor B, y C. Apellido(s) Editor C (Eds. o Comps. etc.), Título del libro en cursiva (pp. xxx-xxx). Lugar de publicación: Editorial.

*Cómo citar un acta de un congreso:* Apellido(s), Iniciales del nombre o nombres. (Año). Título del trabajo. En A. A. Apellido(s) Editor A, B. B. Apellido(s) Editor B, y C. Apellido(s) Editor C (Eds. o Comps. etc.), Nombre de los proceedings en cursiva (pp. xxx-xxx). Lugar de publicación: Editorial.

*Cómo citar tesis doctorales, trabajos fin de máster o proyectos fin de carrera:* Apellido(s), Nombre. (Año). Título de la obra en cursiva. (Tesis doctoral). Institución académica en la que se presenta. Lugar.

*Cómo citar un recurso de Internet:* Los recursos disponibles en Internet pueden presentar una tipología muy variada: revistas, monografías, portales, bases de datos... Por ello, es muy difícil dar una pauta general que sirva para cualquier tipo de recurso. Como mínimo una referencia de Internet debe tener los siguientes datos:

1. Título y autores del documento.
2. Fecha en que se consultó el documento.
3. Dirección (URL “uniform resource locator”)

Veamos, a través de distintos ejemplos, cómo se citan específicamente algunos tipos de recursos electrónicos.



*Monografías:* Se emplea la misma forma de cita que para las monografías en versión impresa. Debe agregar la URL y la fecha en que se consultó el documento

*Artículos de revistas:* Se emplea la misma forma de cita que para los artículos de revista en versión impresa. Debe agregar la URL y la fecha en que se consultó el documento.

*Artículos de revistas electrónicas que se encuentran en una base de datos:* Se emplea la misma forma de cita que para los artículos de revista en versión impresa, pero debe añadirse el nombre de la base datos, la fecha en que se consultó el documento.



Apéndice A

Anexos



# Bibliografía

- L. Breiman, J. H. Friedman, R. A. Olshen, and C. J. Stone. Classification and regression trees. 1984.
- D. Cheng, S. Zhang, Z. Deng, Y. Zhu, and M. Zong. knn algorithm with data-driven k value. pages 499–512, 12 2014. doi: 10.1007/978-3-319-14717-8\_39.
- A. Gron. *Hands-On Machine Learning with Scikit-Learn and TensorFlow: Concepts, Tools, and Techniques to Build Intelligent Systems*. O’Reilly Media, Inc., 1st edition, 2017. ISBN 1491962291, 9781491962299.
- P. Gulati, A. Sharma, and M. Gupta. Theoretical study of decision tree algorithms to identify pivotal factors for performance improvement: A review. *International Journal of Computer Applications*, 141:19–25, 05 2016. doi: 10.5120/ijca2016909926.
- P. Larranaga, I. Inza, and A. Moujahid. Tema 5. clasificadores k-nn.
- S. Mohanty, K. Nathrout, S. Barik, S. Das, and A. Prof. A study on evolution of data in traditional rdbms to big data analytics. *International Journal of Advanced Research in Computer and Communication Engineering*, 4:230–232, 11 2015. doi: 10.17148/IJARCCE.2015.41049.
- A. Oguntimilehin and O. Ademola. A review of big data management, benefits and challenges. *Journal of Emerging Trends in Computing and Information Sciences*, 5:433–438, 07 2014.
- phuongphuong 4112716 and G. M. M. 405k4213231585. Diagram of an artificial neural network. URL <https://tex.stackexchange.com/questions/132444/diagram-of-an-artificial-neural-network>.
- J. R. Quinlan. Induction of decision trees. *Machine Learning*, 1(1):81–106, Mar 1986. ISSN 1573-0565. doi: 10.1007/BF00116251. URL <https://doi.org/10.1007/BF00116251>.
- J. R. Quinlan. *C4.5: Programs for Machine Learning*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1993. ISBN 1-55860-238-0.

- A. Unagar and A. Unagar. Support vector machines. unwinded., Mar 2017. URL <https://medium.com/data-science-group-iitr/support-vector-machinessvm-unraveled-e0e7e3ccd49b>.
- J. Wang, P. Neskovic, and L. N. Cooper. Neighborhood size selection in the k-nearest-neighbor rule using statistical confidence. *Pattern Recognition*, 39(3):417 – 423, 2006. ISSN 0031-3203. doi: <https://doi.org/10.1016/j.patcog.2005.08.009>. URL <http://www.sciencedirect.com/science/article/pii/S0031320305003365>.
- S. Zhang, X. Li, M. Zong, X. Zhu, and R. Wang. Efficient knn classification with different numbers of nearest neighbors. *IEEE Transactions on Neural Networks and Learning Systems*, 29(5):1774–1785, May 2018. ISSN 2162-237X. doi: 10.1109/TNNLS.2017.2673241.