



**INTELIGENCIA ARTIFICIAL**

**APRENDIZAJE AUTOMÁTICO I**



**Structuralia**



Este documento es de uso único e intransferible para el alumno matriculado en el curso. Cualquier reproducción física o digital del documento sin permiso de los autores vulnera los derechos de propiedad intelectual de los mismos.

## INDICE

<b>INDICE.....</b>	<b>2</b>
<b>1. INTRODUCCIÓN.....</b>	<b>4</b>
<b>2. FUNDAMENTOS BÁSICOS DEL APRENDIZAJE AUTOMÁTICO.....</b>	<b>7</b>
2.1 El proceso de Aprendizaje Automático.....	7
2.2 Conceptos básicos.....	8
2.3 El modelo de aprendizaje.....	11
2.4 Técnicas de Aprendizaje Automático.....	14
<b>3. APRENDIZAJE SUPERVISADO.....</b>	<b>16</b>
3.1 Árboles de Decisión.....	17
3.2 Bosques aleatorios (Random Forest).....	23
3.3 Regresión lineal.....	24
3.4 Redes de Neuronas Artificiales.....	26
3.5 K-Nearest Neighbour.....	34
<b>4. CONCLUSIONES.....</b>	<b>36</b>
<b>5. REFERENCIAS.....</b>	<b>38</b>

## 1. INTRODUCCIÓN

La capacidad de aprender es una de las funciones mentales más importantes de los seres vivos ya que les permite adaptarse a las necesidades del entorno con el objetivo de aumentar sus posibilidades de supervivencia. Mediante el proceso de aprendizaje los seres vivos son capaces de modificar, adaptar y/o adquirir habilidades, destrezas, conocimientos y/o comportamientos mediante la interacción con el entorno que les rodea. Este proceso puede ser incluso mucho más complejo desde la perspectiva de los seres humanos donde el proceso de aprendizaje no se limita sólo al entorno y a sus interacciones con él, sino que también influyen ciertos factores sociales relacionados con valores y principios morales que influyen de manera significativa sobre las diferentes habilidades, destrezas, conocimientos y conductas que somos capaces de aprender. Este tipo de diferencias (sesgo) se pueden apreciar significativamente cuando comparamos dos individuos con las mismas capacidades cognitivas que han crecido en diferentes zonas geográficas del planeta donde existen significativas diferencias en la forma de vida. Probablemente ambos individuos se comportarán de forma diferente ante estímulos similares ya que han creado modelos de comportamientos diferentes en base a la información que han recibido.

Esta poderosa capacidad para construir modelos de comportamiento en base a la extracción y manipulación la información del entorno hicieron que los investigadores comenzaran a imitar el proceso de aprendizaje humano con el fin de aumentar las capacidades de los diferentes algoritmos de Inteligencia Artificial los cuales estaban limitados por el conocimiento experto que se les introducía para poder razonar y por los comportamientos (acciones, operaciones, etc.) que eran definidos por los diseñadores de los sistemas. Es decir, los diferentes algoritmos de Inteligencia Artificial no eran capaces de adaptarse a nuevas situaciones si están no habían sido previstas en los modelos que eran utilizados para producir el proceso de razonamiento. Este nuevo área conocimiento perteneciente a la Inteligencia Artificial comenzó a denominarse “Aprendizaje Automático” (Machine Learning, ML, en sus siglas en inglés) debido a que su objetivo era construir modelos de comportamiento mediante el aprendizaje en base a la información disponible del entorno. Dependiendo de cómo se produce el proceso de construcción de los modelos de aprendizaje podemos diferenciar entre diferentes modos de aprendizaje:

- Aprendizaje inductivo: Este tipo de modo de aprendizaje consiste en construir modelos a partir de un proceso de generalización mediante ejemplos simples. Es decir, permite identificar un conjunto de patrones comunes que permitan definir ciertas características de los diferentes ejemplos utilizados en el proceso de aprendizaje. Este proceso se basa

en el razonamiento inductivo [11] mediante el cual los humanos extraemos conclusiones generales de forma independiente a través de un proceso de análisis de la información disponible con el fin de resolver un determinado problema. El conocimiento utilizado siempre se considera nuevo y puede modificar o invalidar el modelo previamente aprendido.

- Aprendizaje analítico o deductivo: Este tipo de modo de aprendizaje consiste en construir modelos a partir de un proceso deductivo mediante la identificación de una descripción general a partir de un conjunto de ejemplos los cuales son explicados de forma específica y completa. Este proceso se basa en el razonamiento deductivo [12] mediante el cual los humanos aprendemos comunmente en el colegio donde se presentan una serie de reglas o leyes las cuales son demostradas en base a una serie de ejemplos que explican de forma el funcionamiento de las diferentes leyes. El nuevo conocimiento no invalida el conocimiento previo sólo lo completa o refuerza.
- Aprendizaje analógico: Este tipo de modo de aprendizaje consiste en construir modelos que permiten generar soluciones a problemas nuevos mediante la búsqueda de similitudes con problemas previamente resueltos de forma que la solución de los problemas similares es adaptada a las particularidades del nuevo problema con el fin de aprender una solución [13]. Este tipo de concepto de aprendizaje es el aplicado para diseñar los sistemas de razonamiento basados en caso descritos en el tema anterior. Algunos investigadores no lo consideran como un proceso total de aprendizaje ya que depende de una fase de adaptación de puede producir muchísimas variaciones en la solución final.
- Aprendizaje conexionista: Este tipo de modo de aprendizaje consiste en construir modelos basados en las conexiones de entidades sencillas con el fin de aprender ciertos conceptos en base a como se configuran estas conexiones. Este tipo de procesos de aprendizaje se han intentado aplicar en el proceso de construcción de redes de neuronas con el fin de crear sistemas dinámicos que adaptan sus conexiones.
- Aprendizaje conductista: Este tipo de modo de aprendizaje, basado en la psicología conductista, consiste en construir modelos en base a la observación y análisis de las reacciones (conductas) aplicadas en el entorno como respuesta a un determinado estímulo que puede suponer una determinada recompensa.

La aparición del Aprendizaje Automático[10] supuso un cambio radical en la forma en la que se estaba abordando el proceso de creación de sistemas basados en Inteligencia Artificial, ya que

a partir de este momento no se intentaban construir algoritmos que fuera capaces de resolver problemas de forma automática, sino que se intentan construir algoritmos que fueran capaces de aprender modelos en base a la información que tienen disponible. Esto supuso la aparición de diferentes tipos de familias de métodos que abordaban diferentes facetas del proceso de aprendizaje comenzando con las primeras versiones del perceptrón [2] [3] hasta los complejos modelos contruidos mediante la utilización del Aprendizaje Profundo (Deep Learning) en base a millones de datos y complejas estructuras basadas en redes de neuronas [4]. En los últimos años la aparición de diferentes Frameworks [14] [15], que facilitan el procesamiento y generación de modelos de aprendizaje, ha supuesto que las empresas hayan centrado sus esfuerzos en utilizar este tipo de técnicas para ofrecer nuevos servicios y funcionalidades en un amplio grupo de áreas como, por ejemplo:

- Movilidad: Identificación de rutas por carretera más eficientes que minimicen el número de embotellamientos en las principales arterias de la ciudad disminuyendo el tiempo, consumo y/o contaminación.
- Sistemas de producción: Predicción de errores de funcionamiento en sistema de producción supervisado o semisupervisado.
- Finanzas: Prevención de fraudes bancarías, lavado de dinero y personalizar la oferta de productos basados en datos del usuario y del mercado con el fin de conocer qué tipo de productos se adaptan mejor a un determinado usuario.
- Agricultura: Identificación de las mejores áreas de cultivo para cada tipo de producto, identificación de fechas de recogida de los productos en el momento óptimo de maduración, así como la identificación de los posibles problemas en el proceso de crecimiento, identificación de enfermedades (visión artificial).
- Energía: Predicción de errores e identificación de zonas en la cual es necesario realizar algún tipo de mantenimiento preventivo. Identificación de los cambios en el coste de la energía con el fin de recomendar a los clientes cuando aumentar o disminuir su consumo con el fin de minimizar los costes.
- Salud: Diagnostico y detección de enfermedades mediante el análisis de la información del usuario. Así como el uso de sistema de visión artificial para la detección de tumores.
- Farmacéutico: Optimizar de los diferentes estudios clínicos realizados mediante la selección automática de pacientes en base a sus características.
- Medios de comunicación: Personalización de publicidad y recomendaciones mediante la utilización de datos multimodales referentes a los usuarios.

- Logística: Optimización en tiempo real de precios de los productos, horarios de apertura, cantidad de producto almacenado en base a los comportamientos previos de los clientes en situaciones o fechas similares.
- Sector público y social: Optimización del proceso de asignación de recursos para el desarrollo urbano con el fin de mejorar la calidad de vida de los usuarios con el fin de minimizar la delincuencia, adaptar los servicios de limpieza a la cantidad de personas en un área. Por ejemplo, aumentar los efectivos de limpieza en ciertos días del año debido a la existencia de eventos o de una mayor afluencia de turísticas.
- Viajes y hotelería: Identificación rutas más eficientes con el fin optimizar itinerario de vuelos, vehículos de transporte de productos perecederos o para la realización de entregas mediante la utilización de transporte local por carretera.

## 2. FUNDAMENTOS BÁSICOS DEL APRENDIZAJE AUTOMÁTICO

En esta sección se presentan los fundamentos básicos del Aprendizaje Automático. Para ello se presentará una descripción teórica del concepto del Aprendizaje Automático basada en el concepto de Aprendizaje en humanos, se describirán los diferentes elementos básicos necesarios para entender cómo funcionan el proceso de generación de un modelo de aprendizaje, a continuación, se describirán aquellos aspectos básicos referentes a los modelos que podemos aprender y por último se describirán las diferentes técnicas agrupadas por tipo o familia.

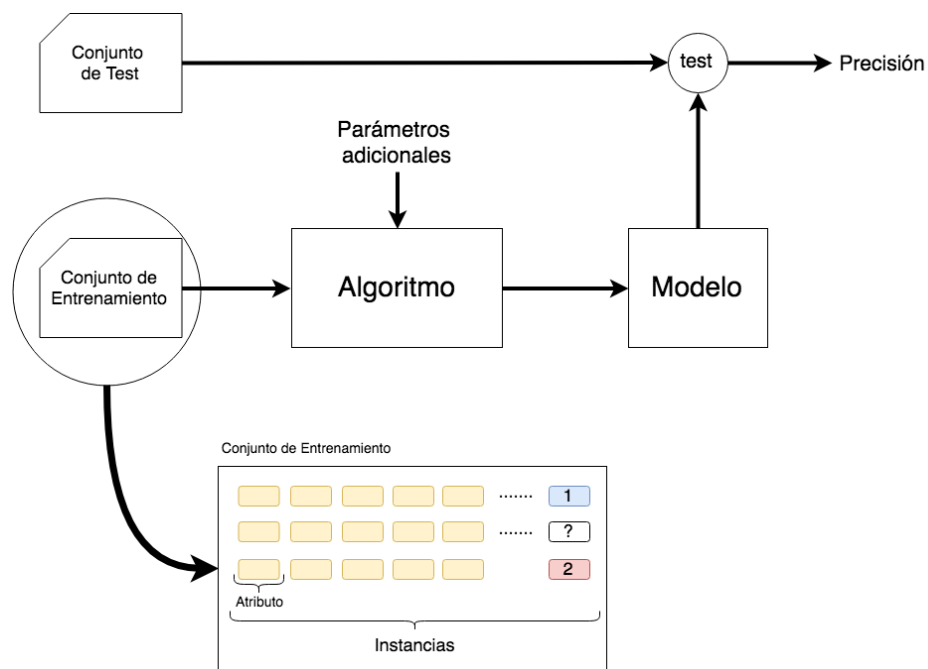
### 2.1 El proceso de Aprendizaje Automático

Desde la perspectiva de una máquina el aprendizaje automático puede definirse como el proceso de adquisición de conocimiento de manera automática mediante la utilización de ejemplos de entrenamiento que define las características del concepto que se desea aprender. Este proceso de adquisición de conocimiento (aprendizaje) puede ser visto como un proceso de generación de cambios en el sistema que aprende (estudiante) los cuales son definidos por la información obtenida del entorno (ejemplos de entrada). Esta información puede ser definida mediante otro sistema que nos enseña (profesor) que realiza una identificación de la información (datos etiquetados) o bien mediante la extracción de información en bruto que no está identificada (datos no etiquetados). Estos implican que los sistemas de aprendizaje deben ser capaces de trabajar con un rango muy amplio de información de entrada, que pueden incluir datos incompletos, inciertos, ruido, inconsistencias, etc., con el fin de construir modelos imperfectos los cuales

pueden ser analizados con el fin de conocer su grado de éxito y adecuación al problema. Dependiendo de cómo se utilice la información de entrada y salida durante el proceso de aprendizaje se pueden diferenciar entre cuatro grandes grupos o familias de técnicas: aprendizaje supervisado, aprendizaje no supervisado, aprendizaje semisupervisado y el aprendizaje por refuerzo. Cada una de estas familias serán descritas de forma detallada a lo largo de este tema y el siguiente.

## 2.2 Conceptos básicos

Para poder describir de forma correcta cómo funcionan las diferentes técnicas de Aprendizaje Automático es necesario introducir algunos conceptos básicos. En la **Figura 1** se presenta el proceso de aprendizaje de un algoritmo cualesquiera de tipo supervisado o semisupervisado. Cada uno de los diferentes elementos de la figura son descritos a continuación con el fin de describir en detalle cada uno de los básicos utilizados en proceso de aprendizaje.



**Figura 1:** Descripción básica del proceso de aprendizaje de un algoritmo de tipo supervisado o semisupervisado.

- **Atributo:** Los atributos, variables de entrada o componentes son las unidades básicas e indivisibles de información utilizada para representar algún tipo de conocimiento. Este tipo de información representa una característica básica que intenta describir algún tipo de



propiedad sobre los datos (color, tamaño, distancia, etc). Los atributos suelen clasificarse en dos categorías en base a los valores que pueden tomar:

- Continuos: Son aquellos atributos que toman un valor fijo dentro de un intervalo no finito perfectamente acotado donde dados dos valores observables siempre existen un tercer valor intermedio que podría tomar el atributo continuo. Por ejemplo, la temperatura de una determinada habitación medida en grados centígrados puede tomar valores de tipo real en un rango comprendido entre  $-50^{\circ}$  y  $50^{\circ}$  centígrados.
- Discretos: Son aquellos atributos que toman su valor entre los elementos que forman un conjunto finito. Por ejemplo, los posibles colores de pintura utilizados para pintar un vehículo.
- Instancia: Una instancia es la estructura de información básica utilizada para representar cada uno de los ejemplos que forma parte de los conjuntos de datos que serán utilizados en el proceso de aprendizaje. Cada instancia, a su vez, está compuesta por un conjunto finito de atributos que la describen. La instancia de un mismo conjunto tiene que estar formadas por el mismo tipo de atributos. La Tabla 1 muestra un conjunto de instancia referentes a la climatología del entorno con el fin de saber si se puede o no jugar un partido de tenis. Cada una de las instancias está compuesta por atributos de tipo continuo (Humedad) y discreto (Cielo, Temperatura y Viento).

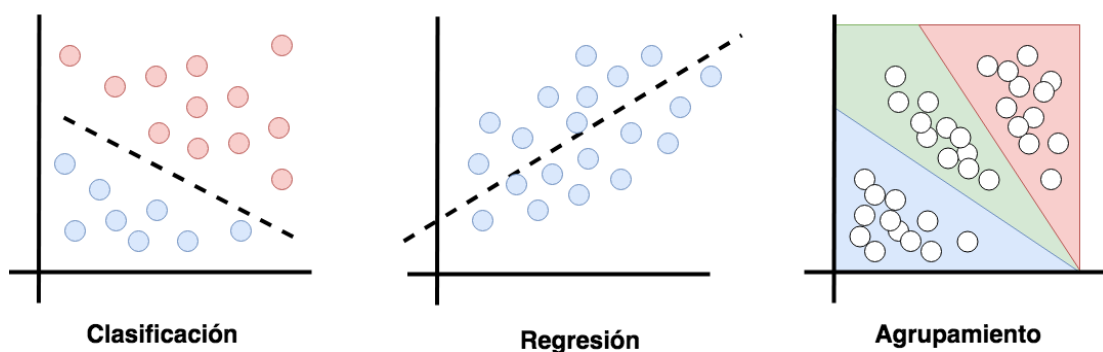
ID	Cielo	Temperatura	Humedad	Viento	Jugar (Etiqueta)
1	Soleado	Alta	65,28	Leve	Sí
2	Nublado	Alta	60,45	Leve	Sí
3	Lluvia	Normal	68,12	Intensa	No

Tabla 1 - Ejemplo de un conjunto de instancia con diferente tipo de atributos

- Objetivo (clase): Es un atributo especial utilizado para realizar la predicción, es decir es el objetivo de la predicción. Por ejemplo, la probabilidad de que un paciente tenga una determinada enfermedad o el precio al que se venderá una vivienda.
- Conjunto de datos: Es el conjunto de instancias que son utilizadas para el proceso de aprendizaje. La manera en la que este conjunto de información es utilizado depende del

tipo de técnica que estemos utilizando. Es decir, la información puede haber sido recogida previamente y almacenada en ficheros, u otro tipo de formato de almacenamiento, que serán utilizados para construir en modelos o puede ser recogida en tiempo real durante el proceso de entrenamiento.

- **Modelo:** Es el resultado del proceso de aprendizaje. De manera formal se puede definir como el conjunto de reglas o patrones inferidos a partir del conjunto de entrenamiento utilizado para construirlo.
- **Algoritmo:** Es el mecanismo mediante el cual se produce el proceso de aprendizaje. Este algoritmo funciona de manera diferentes dependiendo del tipo de salida y de entrada que queremos utilizar para construir el modelo. Cada uno de los diferentes algoritmos de aprendizaje supervisado, no supervisado y semisupervisado puede definirse como uno de los tres tipos de técnicas.



**Figura 2:** Ejemplo de la distribución espacial de un conjunto de instancias en base al funcionamiento de tres tipos de algoritmos

- **Clasificación:** Los algoritmos de clasificación son aquellos que tratan de predecir una salida de tipo discreto que consiste en un atributo de tipo discreto (clases) a partir de un conjunto de datos que pueden estar o no etiquetados (Imagen izquierda **Figura 2**). Las diferentes instancias de entrada (entrenamiento y test) se encuentran ordenadas en categorías o clases. Cada una de estas clases son las etiquetas que han sido asignadas a los datos. Se suelen diferenciar dos tipos de algoritmos: clasificación binaria donde sólo es necesario predecir dos clases objetivo (Si o No) y clasificación multiclase donde se debe predecir más de 2 clases objetivo.
- **Regresión:** Los algoritmos de regresión o estimación son aquellos que tratan de predecir una salida de tipo continuo que consiste en un valor numérico (número real)

a partir de un conjunto de datos etiquetados (Imagen central **Figura 2**). Cada una de las diferentes instancias de entrada (entrenamiento y test) tiene un atributo de “salida” de tipo continuo.

- Agrupamiento: Los algoritmos de agrupamiento (clustering) son aquellos que tratan de predecir una salida de tipo discreto a partir de un conjunto de datos que pueden estar o no etiquetados (Imagen derecha **Figura 2**). Es decir, estos algoritmos son capaces de agrupar los elementos en base a las similitudes entre sus atributos en un conjunto determinado de grupos, cuyo número debe ser incluido como una entrada más del algoritmo.

### 2.3 El modelo de aprendizaje

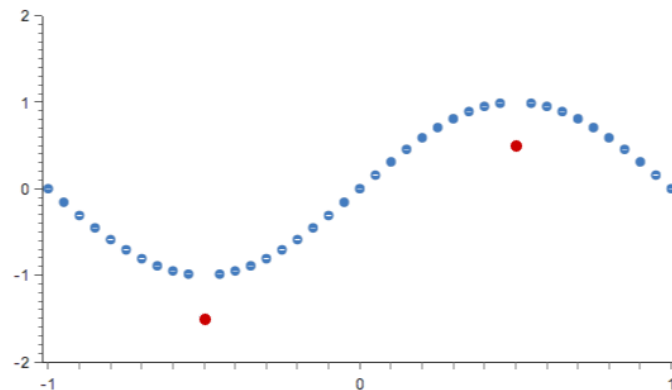
El proceso descrito en la sección anterior nos permite conocer cual son los componentes básicos para poder generar un modelo de aprendizaje, pero no es tan sencillo. En primer lugar, este proceso difiere dependiendo de cada una de las diferentes familias de métodos. Los algoritmos de aprendizaje no supervisado no suelen incluir una fase de test ya no la información utilizada para el proceso de aprendizaje no está etiquetada y los algoritmos de aprendizaje por refuerzo modelan la información mediante la utilización de estados, acciones y recompensas que de un punto de vista muy general pueden ser similares cambian por completo el proceso de aprendizaje. Los pares estado acción pueden ser visto como instancias y el refuerzo como algo parecido a la clase. Todo esto será descrito de forma detallada en el siguiente tema de este curso.

Con respecto al proceso de aprendizaje en general es necesario introducir algunos conceptos básicos referentes al modelo y al tratamiento de la información utilizada para construirlo.

- Discretización: Es el proceso mediante el cual una variable de tipo continuo es transformada en una variable de tipo discreto. Este proceso normalmente se realiza mediante la utilización de los valores disponibles en el conjunto de entrenamiento, en base a sus valores el conjunto es dividido en rangos donde cada uno se corresponderá con un valor discreto de la nueva variable. Esta división se realiza comúnmente de manera manual lo cual hace que se introduzcan sesgos en el proceso de aprendizaje. Aunque existen técnicas que realizan divisiones automáticas del conjunto en rangos de tamaño similar.

- Normalización: Es el proceso de ajuste de los diferentes valores de un atributo con el fin de disminuir importantes diferencias entre los valores o representarlos en otro tipo de escala que permita analizar de forma más efectiva su efecto sobre el resto de los atributos o su significado. Lo más común es realizar una normalización a valores entre 0 y 1.
- Valor atípico: Los valores atípicos o outliers son valores que pueden tomar los atributos muy distantes con respecto al resto de valores que poseen las otras instancias de los conjuntos (entrenamiento y test) empleados en el proceso de aprendizaje. Estos cambios tan importantes en los valores de un atributo pueden significar que se ha producido un error en medición o que se ha detectado un caso inusual que no es común que se reproduzca. Normalmente este tipo de valores son eliminados de los conjuntos de entrenamiento y/o test debido a que pueden afectar de manera significativa al proceso de generación del modelo. Aunque este proceso de eliminación puede ser bastante complicado si los conjuntos utilizados en el proceso de aprendizaje son muy grandes, por lo que algunas técnicas incluyen procesos para detección de valores atípicos con el fin de eliminarlos y/o tratarlos de manera diferente durante el proceso de aprendizaje. Algunos de los métodos más utilizados para la detección de estos valores son: (1) Métodos monovariantes que buscan valores atípicos de forma individual en cada atributo con el fin de eliminarlos; (2) los Métodos multivariantes que buscan valores atípicos de instancias mediante la combinación de múltiples atributos; y (3) el Método de error de Minkowski que a diferencia de los otros dos no intenta eliminar los valores atípicos sino minimizar su impacto en el modelo con el fin de no manipular el conjunto de entrenamiento como realizan los otros métodos. Un ejemplo de valores atípicos sería los presentando en la **Figura 3**, donde el valor de los atributos de los valores atípicos (rojo) difieren del valor que les debería corresponder siguiendo la trayectoria de la sigmoidea.





**Figura 3:** Ejemplo de un conjunto de datos con valores atípicos

- **Sobreajuste:** El sobreajuste, sobre aprendizaje u overfitting es el efecto que se produce al sobreentrenar un modelo mediante un algoritmo de aprendizaje supervisado. Uno de los problemas. El proceso de aprendizaje debe generar un modelo que permita predecir el resultado de otras instancias generalizando a partir de lo aprendido mediante la utilización de las instancias de entrenamiento. Sin embargo, cuando el conjunto de entrenamiento se encuentra formado por instancias que no representan de forma global el problema o está formado por casos muy minoritarios o poco comunes se puede producir un efecto de sobre entrenamiento. Este efecto hace que el modelo generado por el algoritmo no sea capaz de generalizar y quede ajustado a unas características muy específicas de los datos de entrenamiento que no tiene relación causal con la función objetivo.
- **Ruido:** El ruido es el conjunto de atributos o valores de un atributo específico que no aportan nada al proceso de aprendizaje y que pueden haber sido introducidos mediante un proceso de medición incorrecto, un error en la inserción de los datos y simplemente son valores atípicos que puede afectar al proceso de aprendizaje debido a su rareza. Es decir, es imposible utilizar este conjunto de valores y/o atributos con el fin de generar un modelo que sea capaz de generalizar en base a las instancias utilizadas para construirlo. El ruido normalmente aporta muchísimas desventajas a la hora de construir un modelo, ya que tiende a aumentar la complejidad de los datos complicando la ejecución de los algoritmos los cuales terminan generando modelos que no generalizan correctamente. Antes de construir un modelo es muy importante analizar las instancias de entrenamiento con el fin de eliminar el posible ruido que existe. Este proceso puede ser muy complicado dependiendo del tamaño del conjunto de entrenamiento y test, por lo que existen diferentes aplicaciones que intentan eliminar el ruido de manera automática, aunque este

tipo de herramientas no suele asegurar la eliminación total del ruido o incluso pueden eliminar de manera errónea información que es relevante para el proceso de aprendizaje.

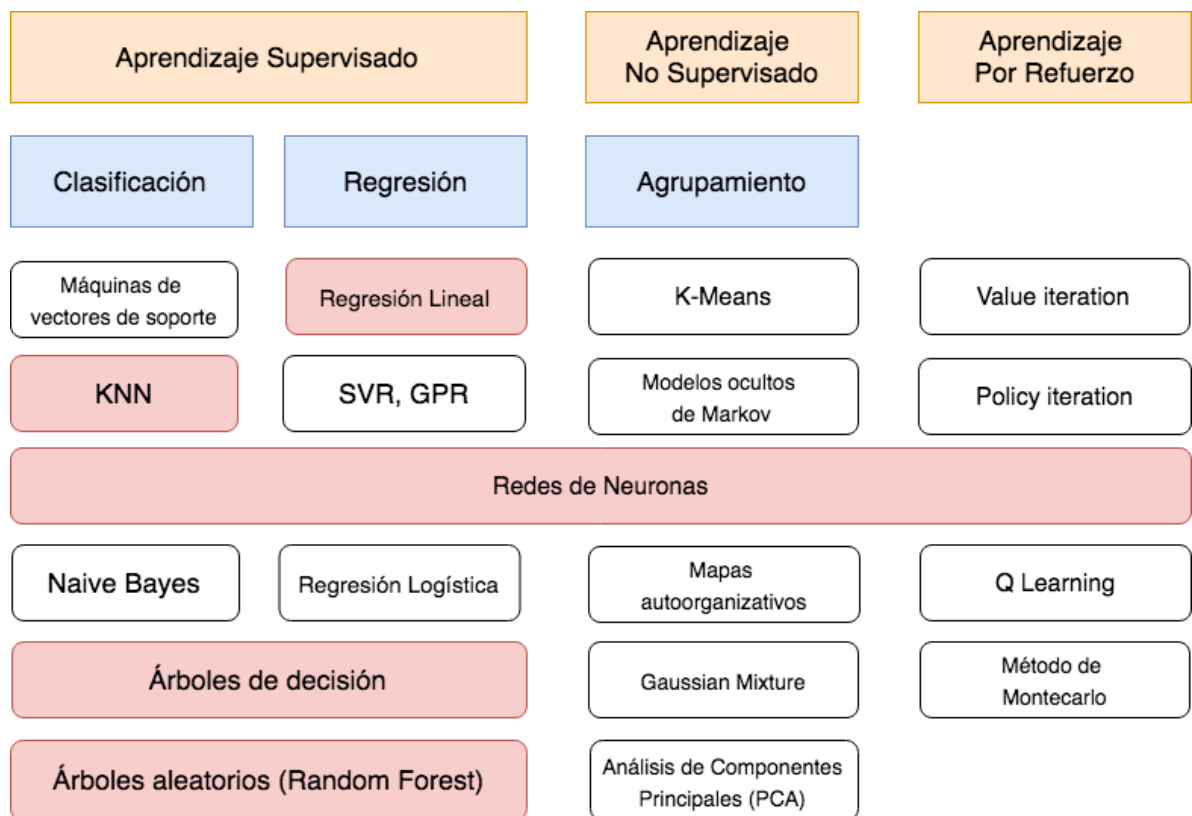
- **Sesgo:** El sesgo es probablemente uno de los conceptos más importante del Aprendizaje automático debido a su aparición puede producir que nuestros modelos sean inservibles. El sesgo puede considerarse como un error del modelo generado el cual realiza predicciones “sesgadas” debido a que los datos utilizados para construirlo están distorsionados de manera directa o indirecta. Un ejemplo típico de sesgo que ha aparecido en muchos de los primeros sistemas basados en visión artificial utilizados en los vehículos autónomos donde el proceso de entrenamiento había sido sólo realizado con adultos de estatura media. Esto producía que el modelo no fuera capaz de identificar a los niños como humanos. En el tema 9 de este curso se hará una descripción más detallada del concepto de sesgo y cuál es su influencia en las diferentes técnicas de Inteligencia Artificial.

## 2.4 Técnicas de Aprendizaje Automático

Actualmente el Aprendizaje Automático está formado por un amplio número de técnicas, las cuales pueden ser clasificados en cuatro familias de técnicas en base a cómo se encuentra modelada la información que usan para definir las entradas y la salida.

- **Supervisado:** Este tipo de técnicas son denominadas como técnicas de aprendizaje mediante profesor debido a que la información que utilizan durante el proceso de aprendizaje es completa. Es decir, las instancias de entrenamiento y test están formadas por atributos que definen las entradas y la salida esperada para cada instancia (datos etiquetados). Su funcionamiento consiste en la identificación de una función (modelo) que sea capaz de realizar un mapeo entre las entradas y las salidas conocidas. Las técnicas de este grupo están basadas en clasificación y regresión.
- **No supervisado:** Este tipo de técnicas son más complejas que las anteriores debido a que la información que utilizan durante el proceso de aprendizaje es incompleta. Es decir, las instancias de entrenamiento y test están formadas únicamente por atributos de entrada. No se conoce la salida esperada para ninguna de las instancias (datos no etiquetados). Su funcionamiento consiste en la identificación de una función (modelo) que sea capaz de agrupar las diferentes instancias en base a ciertas características comunes. Las técnicas de este grupo están basadas en clasificación y agrupamiento.

- Semisupervisado:** En algunas ocasiones este tipo de técnicas son incluidos en uno de los grupos anteriores ya que la mayor parte de los algoritmos de este grupo puede incluirse en alguno de los anteriores debido a que aplican algún tipo de transformación en los datos de entrada con el fin de poder aplicar alguno del algoritmo de los grupos anteriores o combinan algoritmos de ambos grupos. En mi opinión, debe ser clasificados en un grupo independiente debido a que utilizan información parcialmente incompleta. Es decir, las instancias de entrenamiento o test no tienen una estructura totalmente similar debido a que algunas instancias están etiquetadas y otras no.



**Figura 4:** Descripción de diferentes técnicas de Aprendizaje Automático agrupadas por familia.

- Por refuerzo:** Este es un tipo especial de técnicas de aprendizaje automático no supervisado debido a que utilizan una forma diferente de representación de los ejemplos de entrenamiento basada en la utilización de estados y acciones, donde las aplicaciones de las acciones producen una realimentación en el entorno mediante recompensas. En el aprendizaje por refuerzo el objetivo consiste en construir una política para resolver un

determinado problema. Esta política suele ser una secuencia de acciones que permiten resolver el problema de manera óptima.

En la **Figura 4** se presentan diferentes tipos de algoritmos de Aprendizaje Automático distribuidos por familia (Supervisado, No supervisado y Por refuerzo) y por técnica (Regresión, Clasificación y Agrupamiento). En esta figura sólo se presentan algunos de las técnicas más utilizados, pero existen muchas más e incluso diferentes versiones de cada una de ellas. En rojo se marcan las diferentes técnicas que serán descritas de forma detallada en este tema.

### 3. APRENDIZAJE SUPERVISADO

El aprendizaje supervisado es una de las familias de métodos de aprendizaje basada en el aprendizaje inductivo [4] debido a que el proceso de aprendizaje se produce mediante instancias (ejemplos) etiquetadas. Este tipo de modelo de aprendizaje consiste en definir la información del entorno en pares de la forma (entrada – salida) de manera que ante nuevas entradas cuya salida es desconocida el sistema sea capaz de predecir la salida esperada en base a lo que ha aprendido durante el proceso de entrenamiento. La entrada se corresponde con las diferentes características que definen las instancias (ejemplos) y la salida se corresponde con el objetivo (clase o valor) al que pertenece en ejemplo. El proceso para identificar o definir esta función está dividido en dos fases:

- Fase de entrenamiento: Es la fase principal del proceso de aprendizaje y consiste en seleccionar aquellas características más relevantes de cada uno de los ejemplos de entrada comparándolas con otros ejemplos analizados previamente, mediante algún proceso de cotejamiento (identificación de patrones), de manera que cuando se detectan diferencias significativas se produce una adaptación del modelo que está siendo construido en base al aprendizaje. Este proceso puede ser visto como un sistema de identificación de patrones mediante la generalización de las características similares entre los diferentes ejemplos con el fin de construir un modelo que sea capaz de identificar los diferentes grupos de ejemplos con características similares. Es decir, el resultado del proceso de entrenamiento es un sistema formado por un conjunto de patrones que han sido identificados en base a las características de los ejemplos utilizados en el proceso de entrenamiento.
- Fase de test: Esta es la fase secundaria y opcional del proceso de aprendizaje que nos permite validar el resultado de la fase entrenamiento mediante el uso de ejemplos que

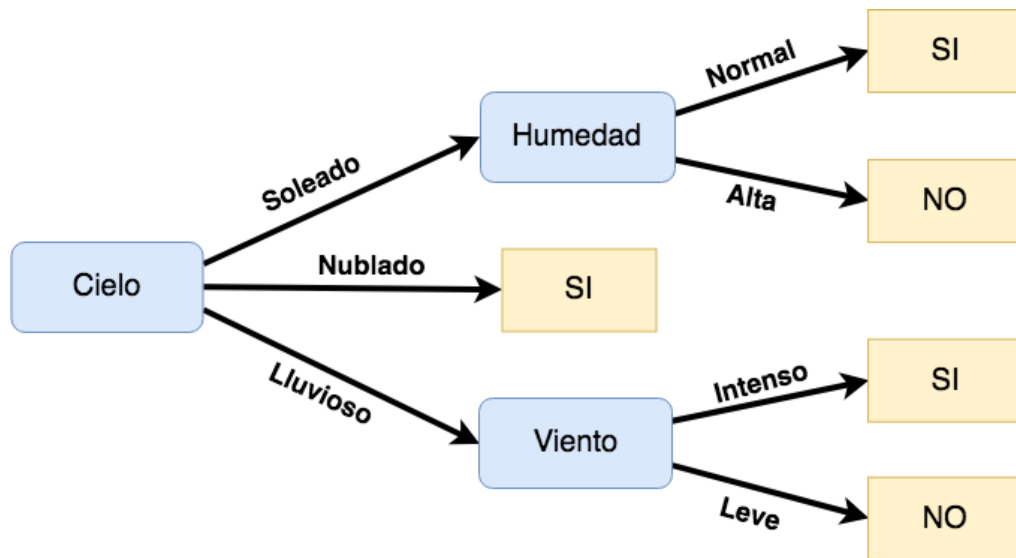


no han sido utilizados en la fase de entrenamiento con el fin de comprobar. Se suele considerar cómo una fase opcional debido a que no puede aplicarse a todos los tipos de algoritmos de Aprendizaje Automático.

### 3.1 Árboles de Decisión

Un árbol de decisión (clasificación o regresión) es una estructura de datos que permite representar la información en forma de un árbol. Esta estructura de datos está por formada por un único nodo, denominado raíz, el cual se encuentra conectado a una serie de nodos sucesores denominados nodos intermedios. Los nodos intermedios tienen un conjunto finito de sucesores que se encuentran conectados de forma directa con él mediante un arco o rama. Aquellos nodos que no tienen ningún sucesor son denominados nodos terminales u hojas. La forma en la que se va ramificando a nivel espacial esta estructura de datos recuerda a un árbol. La construcción de modelos aprendizaje mediante árboles de decisión es una de las técnicas de Aprendizaje automático basada en el aprendizaje inductivo más utilizada debido principalmente a que son capaces de construir un predictor mediante un árbol n-ario que representa y categoriza una serie de condiciones en base a las cuales se puede describir las diferentes instancias utilizadas para construir el árbol. Además, son capaces de representar visualmente el árbol resultante podremos obtener la predicción de cualquier nueva instancia sólo siguiendo un camino desde la raíz hasta uno de los nodos hoja. Este fenómeno permite fácilmente comprobar la calidad del modelo, su homogeneidad, etc. Los árboles de decisión y regresión están formados por tres elementos:

- Los nodos intermedios, incluida la raíz, que se corresponde con un determinado atributo de la estructura de las instancias.
- Los nodos hoja que se corresponden con las clases objetivos que dependiendo de si el árbol es de clasificación o regresión serán atributos de tipo discreto o continuo respectivamente.
- Los arcos (alternativas de la decisión) que unen los nodos intermedios con sus sucesores (nodos intermedios y hojas) que representan los posibles valores de los atributos que han sido seleccionado para producir el modelo.



**Figura 5:** Ejemplo de árbol de decisión

En la **Figura 5** se presenta un ejemplo de un árbol de clasificación que permite decidir si jugar o no un partido de tenis. Es un árbol de clasificación binario debido a todos los nodos hoja (clases) son de tipo discreto donde los posibles valores que puede tomar el atributo objetivo son dos (SI o NO). Las diferentes instancias utilizadas para construirlo están formadas por tres tipos de atributos (cielo, humedad y viento) cuyos valores son de tipo discreto.

### 3.1.1. Algoritmo ID3

El algoritmo ID3 (Iterative Dichotomiser 3) [5] es un método de aprendizaje considerado como el precursor de los algoritmos de la construcción de árboles de decisión ya que muchos de los diferentes algoritmos de este tipo están basados en él. Este algoritmo permite crear clasificadores estadísticos mediante la construcción de árboles de decisión a partir de instancias con atributos discretos mediante la utilización del concepto de entropía de la teoría matemática de la información [6], la cual permite medir la incertidumbre de un conjunto de información. Es decir, nos indica el grado de desordenación de los datos de un conjunto, cuyo valor oscila entre 0 y 1. La entropía para la construcción de un árbol de clasificación binaria para un conjunto de instancias  $S$  puede definirse como:

$$H(S) = -p(2) \log_2 p(2) - p(1) \log_2 p(1)$$

donde  $p(2)$  es la fracción de ejemplos de la clase 2 en  $S$  y  $p_1$  es la fracción de ejemplos de clase 1 en  $S$ . Si todos los ejemplos del conjunto  $S$  pertenecen sólo a una de las clases el valor de la entropía es 0. En cambio, si los valores de  $p_0$  y  $p_1$  están perfectamente balanceados es decir tiene un valor 0,5 el valor de la entropía sería 1, lo que significa que el conjunto  $S$  está perfectamente balanceado ya que incluye el mismo número de ejemplos de cada clase. Este concepto de la entropía se puede generalizar para árboles de clasificación multiclase mediante la siguiente formula, siendo  $c$  el número de clases:

$$H(S) = \sum_{i=1}^c -p(i) \log_2 p(i)$$

El valor de la entropía es utilizado para calcular la ganancia o efectividad de un atributo para dividir el conjunto de instancias en  $n$  subconjuntos (uno por cada posible valor del atributo seleccionado). La ganancia es el valor esperado de la entropía tras producir una partición del conjunto de instancia, que se calcula en base a la siguiente formula:

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

Donde  $A$  es el atributo seleccionado y  $S_v$  es el conjunto de instancia donde el atributo  $A$  tiene el valor  $v$ . El algoritmo ID3 es un proceso recursivo de divide y vencerás que intenta minimizar el valor de la entropía con el fin de conseguir árboles de menor tamaño. El funcionamiento del algoritmo consiste en un proceso de búsqueda avariciosa o voraz sobre diferentes subconjuntos de conjunto  $S$ , el cual está formado por las todas las instancias del problema, utilizando con medida de estimación la ganancia (heurística). El algoritmo comienza calculando el valor de la entropía de cada uno de los atributos no seleccionados de los elementos del conjunto  $S$  (todos en el paso inicial), seleccionándose aquel atributo que tiene el menor valor de entropía (mayor ganancia de información). Se crea un nodo del árbol para el atributo seleccionado y a continuación el conjunto  $S$  es dividido o particionado en  $n$  subconjuntos en base a los posibles valores del atributo. Para cada uno de los subconjuntos generado se aplica el mismo proceso calculándose el valor de la ganancia para todos los atributos y se seleccionado aquel atributo que minimiza la entropía hasta que:

- ✓ Todos los elementos del subconjunto pertenecen a la misma clase, por lo que se crea un nodo de tipo hoja y se le etiqueta con la clase.

- ✓ No existe ningún otro atributo que seleccionar, por lo que se crea un nodo de tipo hoja y se les etiqueta la clase mayoritaria a todos los elementos del subconjunto.

El proceso continuo hasta que todos los elementos del conjunto  $S$  hayan sido asignados a un nodo hoja. El funcionamiento de este algoritmo fue mejorado a través de múltiples versiones (ID4[7] e ID5r[4]) con el fin de mejorar el funcionamiento y el rendimiento del algoritmo.

### 3.1.2. Algoritmo CART

El algoritmo CART (Classification and Regression Trees) [8] es un método de aprendizaje que permite crear clasificadores y regresores estadísticos mediante la construcción de árboles de decisión (clasificación y regresión) a partir de instancias con atributos discretos y continuos mediante la utilización del concepto de impureza que define la homogeneidad de cada nodo. Es decir, el concepto de impureza define como similares son los elementos de un conjunto en base a un determinado criterio que en este caso es una de las posibles clases. El valor de impureza se calcula de manera diferentes dependiendo del tipo de árbol que se esté creando:

- Clasificación: La impureza para el proceso de generación de un árbol de clasificación puede ser calculada en base a la entropía o en base al índice de diversidad de Gini, el cual puede ser calculado en base a la siguiente fórmula para clasificación binaria:

$$i(S) = p(1) p(2)$$

donde  $p(1)$  es la fracción de ejemplos de la clase 1 en  $S$  y  $p(2)$  es la fracción de ejemplos de clase 2 en  $S$ . El cálculo de la impureza puede ser extendido para clasificación multiclase utilizando la siguiente fórmula, siendo  $c$  el número de clases:

$$i(S) = \sum_{i \neq j}^c p(i) p(j)$$

- Regresión: La impureza para el proceso de generación de un árbol de regresión se calcula como la agregación de las varianzas de todos los nodos terminales.



El algoritmo CART es un proceso formado por dos fases que generan un árbol, denominado saturado, que posteriormente es optimizado mediante la aplicación de diferentes técnicas de poda.

- Generación de un árbol saturado mediante un algoritmo recursivo de divide y vencerás que intenta minimizar el valor de la impureza de los nodos del árbol con el fin de conseguir una mayor homogeneidad en los ejemplos que pertenecen a cada uno de los nodos del árbol. Este proceso tiende a generar árboles de gran profundidad debido a que se han intentado crear nodos que representan instancias con la mayor homogeneidad posible.
- Optimización del árbol mediante un proceso de poda. La poda consiste en encontrar el subárbol del árbol saturado con la mejor calidad en base a la ratio de predicción de los resultados y menor vulnerabilidad al ruido de las instancias de entrada (entrenamiento). Para ello se realiza un análisis de calidad de los nodos terminales de los posibles subárboles del árbol saturado. Para un árbol  $T$  se define la calidad en base a la siguiente formula:

$$c(T) = \sum_{t \in \tilde{T}} p[t]r(t)$$

donde  $\tilde{T}$  es el conjunto de nodos terminales del árbol  $T$  y  $r(t)$  es una medida de calidad del nodo  $t$  la cual es similar a la suma de los cuadrados de los residuales en regresión lineal. Es posible utilizar la impureza del nodo  $t$  como medida de la calidad.

Este proceso de generación de árboles puede ser utilizado para la generación tanto de árboles de regresión como de clasificación variando la forma en la que se calcula la impureza y la calidad de los nodos terminales para el proceso de poda.

### 3.1.3. Algoritmo C4.5

El algoritmo C4.5[9], denominado J48 en su versión open source utilizada en muchos frameworks de aprendizaje, es un método de aprendizaje que mejora las diferentes versiones del algoritmo ID3 que permite crear clasificadores estadísticos mediante la construcción de árboles de decisión a partir de instancias con atributos discretos y continuos. Este método incluye importantes mejoras que resuelven algunos de los problemas que tenían sus antecesores siendo el más importante de ellos la posibilidad de utilizar atributos continuos. Para ello, el algoritmo realiza un

análisis de los atributos de las instancias de entrada con el fin de discretizar los atributos mediante su separación en rangos en base a los valores de las instancias de entrada. Además incluye dos importante mejoras con el fin de optimizar el proceso de generación y evitar el sobreaprendizaje: (1) Permite la utilización de procesos de poda (simplificación del tamaño del árbol) durante (se detiene el proceso de subdivisión den base a algún criterio) y al finalizar (se realiza una serie de modificaciones sobre el árbol de forma que ciertas ramas se fusionan o se transforman en nodos hoja) el proceso de generación; y (2) Permite ignorar los datos atributos perdidos durante el proceso de generación, es decir no se tienen en cuenta aquellas instancias que no tienen valor para el atributo que se está analizando.

Al igual que su antecesor, el algoritmo C4.5 es un proceso recursivo de divide y vencerás que intenta minimizar el valor de la entropía, pero introduciendo una nueva medida de la ganancia, denominado ratio de ganancia, que permite calcular de forma más eficiente la ganancia de elegir un atributo.

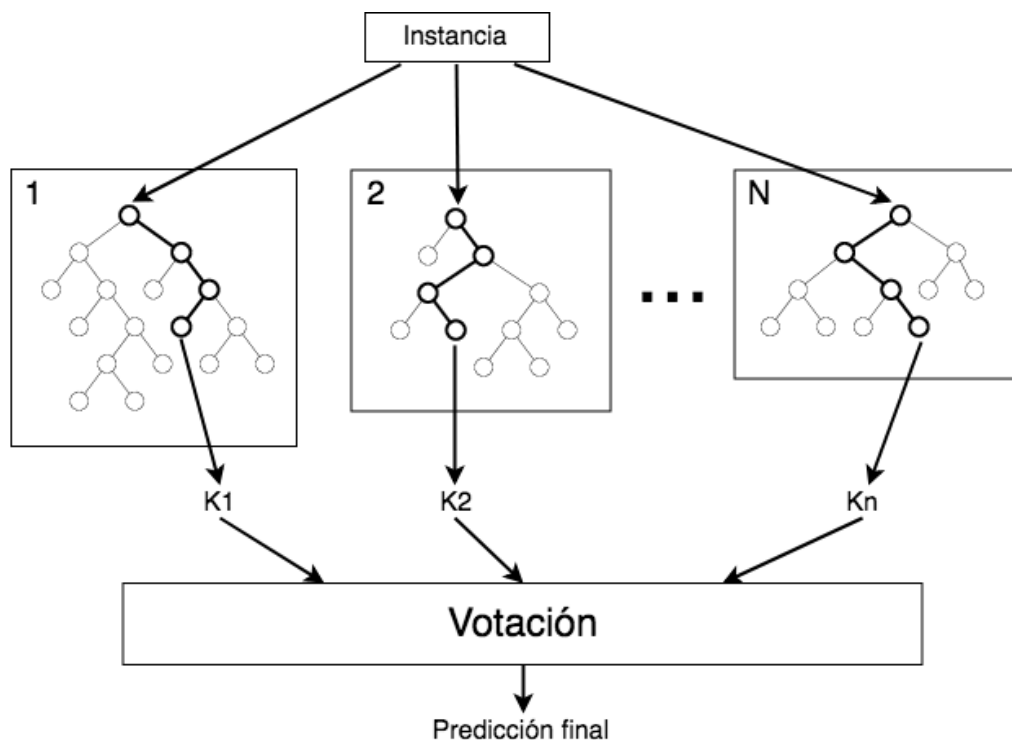
$$ratio(S, A) = \frac{G(S, A)}{H_{parcial}(S, A)}$$

$$H_{parcial}(S, A) = - \sum_{v \in valores(A)} \frac{|S_v|}{|S|} \log_2 \frac{|S_v|}{|S|}$$

El funcionamiento del algoritmo es similar al de ID3 salvo en la forma de calcular la ganancia de cada atributo donde se utilizan las dos formular presentadas previamente y en la fase de generación de los diferentes subconjuntos donde C4.5 genera nodos intermedios u hojas se introducen nuevos casos base con el fin de mejorar el tamaño del árbol como por ejemplo en situaciones donde no se obtiene ninguna ganancia de información para ninguno de los subconjuntos generados en base a los valores del atributo. En estos casos, se crea un nodo intermedio más arriba del árbol utilizando el valor esperado de la clase. El algoritmo C4.5 ha sido mejorado dando lugar a diferentes versiones (C5.0 y See5) con el fin de mejorar su funcionamiento (velocidad y uso de memoria) y añadir más funcionales que permiten construir árboles de menor tamaño o que eliminan atributos que no son utilizados en el proceso de clasificación.

### 3.2 Bosques aleatorios (Random Forest)

Los bosques aleatorios o los bosques de decisión [17] aleatorios son un método de aprendizaje para la generación de clasificadores o regresores basados en árboles de decisión que fueron definidos para evitar el proceso de sobreajuste de los modelos producidos por las diferentes técnicas de generación de árboles de decisión. Este método funciona mediante la generación de una población aleatoria de árboles de decisión que son utilizados para decidir en común el valor la variable de salida. Dependiendo del tipo de árboles de población el proceso decisión será diferente. Para una población formada por árboles de clasificación se realizará un proceso de votación siendo la clase mayoritariamente predicha la salida del método. En cambio, para una población formada por árboles de regresión se calcula la media de las predicciones de cada uno de los árboles. En la **Figura 6** se muestra un ejemplo del funcionamiento de clasificador o regresor construido mediante bosque de árboles aleatorios.



**Figura 6:** Ejemplo de funcionamiento de un bosque de decisión aleatorio

El proceso de generación de un bosque de decisión aleatorio consiste en manipular el conjunto de entrada (entrenamiento) con el fin de generar un conjunto de entrada aleatorios que serán utilizados para entrenar cada uno de los árboles del bosque. Para el proceso de generación de

los conjuntos de entrenamiento se utiliza un algoritmo de Bootstrap aggregating, también denominado algoritmo de embolsamiento (bagging en inglés). Dado un conjunto de entrenamiento  $E$  con un número  $n$  de elementos el algoritmo de embolsamiento generará  $m$  conjuntos de entrenamiento de tamaño  $n$  mediante la selección uniforme de instancias del conjunto  $E$ . Este sistema de generación basado en un muestreo con remplazo implica que algunas que una parte de las instancias que forman los diferentes conjuntos de entrenamiento estarán repetidas. Cada uno de estos conjuntos será utilizado para construir un árbol de decisión, los cuales pueden ser generados por medio de técnicas diferentes. Una vez definidos todos los árboles (modelos) son combinados con el fin de generar una predicción común calculada promediando la salida (para regresión) o votando (para clasificación).

### 3.3 Regresión lineal

La regresión lineal es un método matemático utilizado para estudiar la relación entre un conjunto de variables independientes (variables predictoras) y una variable dependiente (criterio o variable predecida) [18]. Dependiendo del número de variables independientes utilizadas en el proceso de predicción se pueden diferenciar entre regresión simple cuando sólo utilizamos una variable dependiente y regresión múltiple cuando utilizamos más de una variable independiente. Esta técnica permite desarrollar una ecuación lineal que puede ser utilizada para construir un modelo de predicción a partir de instancias con atributos discretos de entrada y salida.

#### 3.3.1. Regresión lineal simple

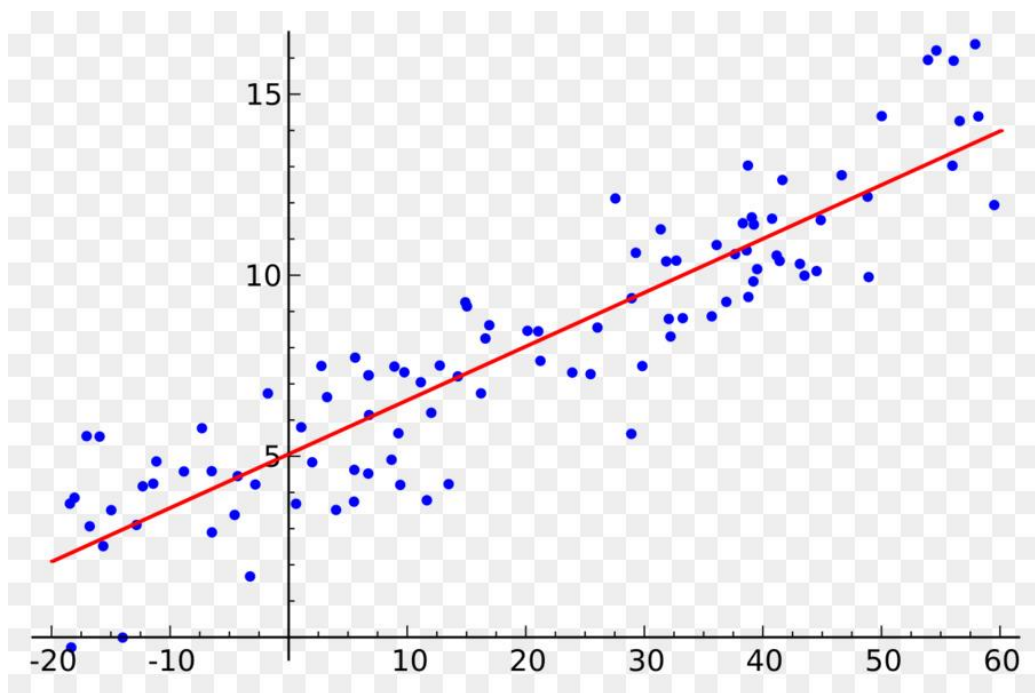
La regresión lineal simple consiste en calcular una ecuación lineal que relaciona una variable independiente  $x$  y una variable dependiente  $y$ . La ecuación lineal que debe ser calculada tiene la siguiente estructura:

$$y = \beta_0 + \beta_1 x + \varepsilon$$

donde  $y$  es la variable dependiente que queremos predecir en base a todas las demás,  $\beta_0$  y  $\beta_1$  son parámetros desconocidos que deben ser calculados,  $x$  es la variable independiente y  $\varepsilon$  es el error cometido en la predicción. El proceso de aprendizaje consistiría en utilizar instancias de entrada (entrenamiento), compuestas cada una por dos valores uno para la variable independiente  $x$  y otro para la variable dependiente, para calcular los valores  $\beta_0$  y  $\beta_1$  que darían lugar a la ecuación de una recta donde  $\beta_0$  sería la ordenada en el origen, es decir es la altura a



la que la recta corte el eje Y, y  $\beta_1$  es la pendiente de la recta, es decir es el incremento o decremento que se produce en la variable y cuando la variable x aumenta o disminuye una unidad. La Figura presenta el resultado de un proceso de regresión simple que consiste en aproximar una recta de regresión que se ajuste mejor la nube de puntos producidas por los valores utilizados para calcularla.



**Figura 7:** Ejemplo de la recta resultante de un proceso de regresión lineal simple

Uno de los grandes problemas de esta técnica es que dado un conjunto de instancias de entrada (entrenamiento) existen infinitas rectas que pueden pasar por la nube de puntos por lo que es necesario elegir la recta que pasa lo más cerca posible de la mayor parte de los puntos. Por lo que el objetivo de aprendizaje consiste en encontrar la recta que represente mejor el conjunto de puntos. Existen diferentes técnicas matemáticas que permiten ajustar una función simple, siendo la más utilizada la técnica de mínimos cuadrados que permite identificar la recta que hace mínima la suma de los cuadrados de las distancias verticales entre cada punto y la recta.

### 3.3.2. Regresión lineal múltiple

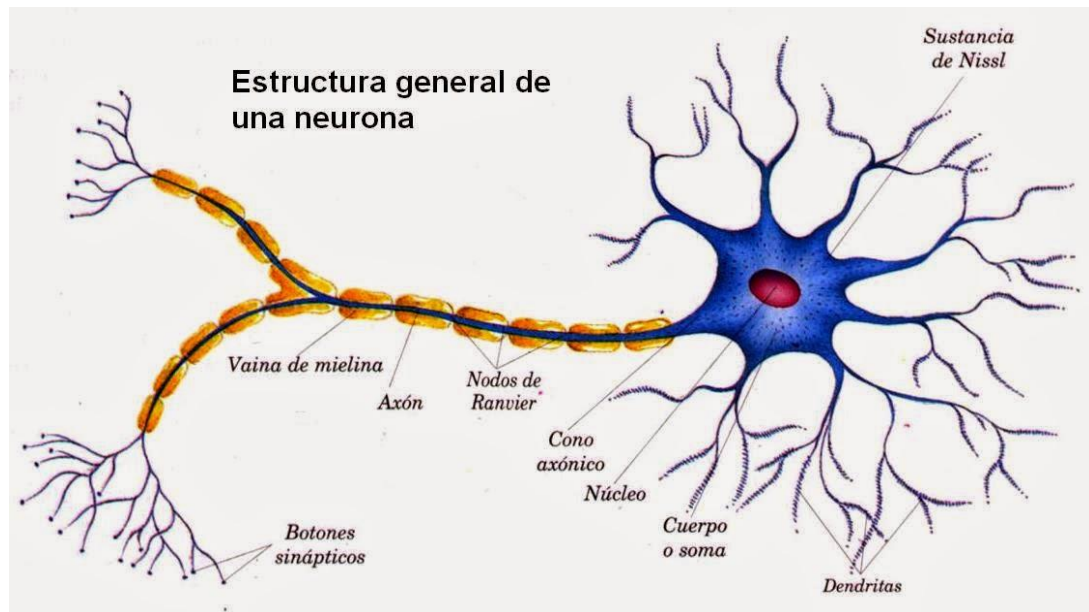
La regresión lineal simple es un caso particular de la regresión lineal múltiple que consiste en calcular una ecuación que relaciona un conjunto de variables independientes  $x_1, x_2, \dots, x_n$  y una variable dependiente y. La ecuación para este caso de regresión tiene la siguiente estructura:

$$y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \dots + \beta_n x_n + \varepsilon$$

El funcionamiento de la regresión lineal múltiple es exactamente similar a la regresión lineal simple salvo con la diferencia de que tenemos múltiples variables independiente y de que la ecuación no representa una recta, sino que representará un plano cuando tengamos dos variables independientes, un espacio tridimensional cuando tengamos tres variables independientes y un hiperespacio cuando tengamos más de tres variables.

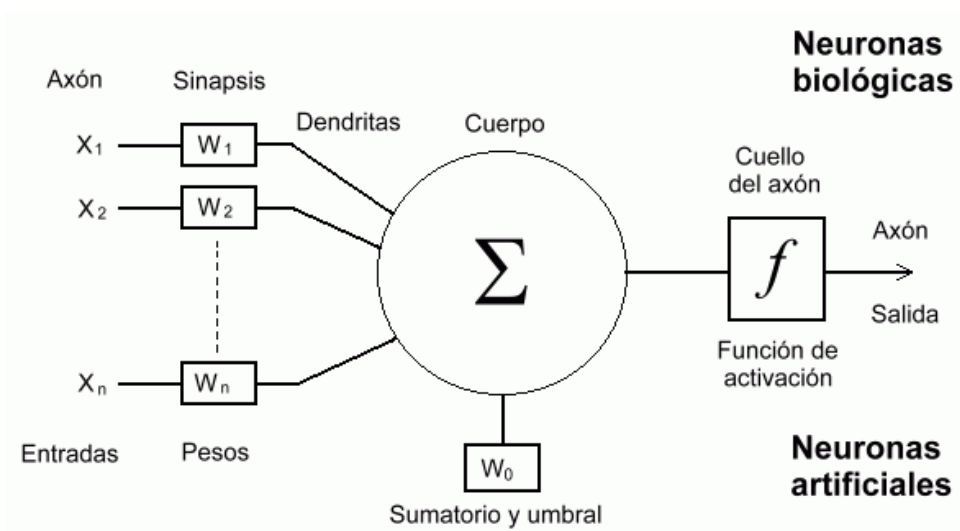
### 3.4 Redes de Neuronas Artificiales

Una Red de Neuronas Artificial (RNA) es una red representada de manera espacial mediante un grafo dirigido donde cada uno de los nodos representan neuronas y los arcos representan las sinapsis o conexiones entre las neuronas [19]. Esta estructura intenta imitar la distribución de las neuronas en el cerebro humano con el fin de construir un tipo de algoritmo que sea capaz de simular la capacidad de procesamiento de nuestros cerebros. Para ello es necesario realizar una breve introducción de la estructura de una neurona humana con el fin de entender cuales sus similitudes. Las neuronas fueron descubiertas o identificadas por el científico español Ramón y Cajal en 1888, donde definió el cerebro como una red de aproximadamente  $100 * 10^6$  células individuales, denominadas neuronas, ampliamente interconectadas entre sí. Las neuronas humanas están formadas por tres elementos: (1) el **soma** o cuerpo celular es la parte principal de la neuronas (10 a 80 micras de longitud); (2) las múltiples prolongaciones que salen de distintas partes del soma, denominadas **dendritas**, cuya función consiste en recibir impulsos de otras neuronas y enviarlos hasta el soma; y (3) el **axón**, que es una prolongación del soma que se extiende en dirección opuesta a las dendritas y tiene la función de conducir un impulso nervioso desde el soma hacia otra neurona, músculo o glándula del cuerpo humano. La **Figura 8** presenta la estructura detallada de una neurona humana y sus diferentes partes.



**Figura 8:** Estructura biológica de una neurona

En base a esta estructura biológica, los investigadores Pitts y McCulloch definieron el primer modelo de red neuronal en 1943[20]. Este modelo consistía en una red con dos capas de neuronas conectadas entre sí, donde la primera capa estaba formada por conjunto de nodos o neuronas formales (neuronas presinápticas) que representaban la entrada de la red y la segunda capa estaba formada por un único nodo o neurona formal (neurona postsináptica) que representaba la salida de la red. Una neurona formal era una puerta lógica con dos posibles estados internos (encendido o apagado) representados por una variable. Esta red funcionaba como un discriminador del estado de la puerta lógica, de forma que las neuronas de la primera capa recibían las entradas, que eran enviadas a la segunda capa donde se aplicaba una operación matemática obteniéndose un valor sobre el cual se aplicaba una función que activación, basada en un umbral, que definía el estado de la salida. Es decir, si el valor obtenido de la operación sobre las entradas era mayor que el umbral la salida era 1 y sino la salida era 0. A partir de este y otros modelos que se desarrollaron en los siguientes años, Rumelhart y McClelland introdujeron en 1986 lo que se conoce actualmente como el modelo estándar de neurona artificial, el cual se encuentra representado en la Figura , donde se presentan los componentes de una neurona artificial, así como la correspondencia entre los elementos de una neurona biológica y una neurona artificial. Como se puede observar en la imagen la neurona artificial intenta imitar de forma completa la estructura de la neurona biológica.



**Figura 9:** Elementos básicos de una neurona artificial

- Un conjunto de entradas  $x_j$  y un conjunto de pesos  $w_{ij}$ , denominados sinápticos, donde  $j$  es un valor entre 1 y  $n$ .
- Una función de propagación  $h_i$  definida a partir del conjunto de entradas y los pesos sinápticos. Es decir:

$$h_i(x_1, \dots, x_n, w_{i1}, \dots, w_{in})$$

La función de propagación define la operación matemática mediante la cual se combinan las diferentes entradas que recibe la neurona. La función de propagación más comúnmente utilizada consiste en combinar linealmente las entradas y los pesos sinápticos, mediante la siguiente operación matemática:

$$h_i(x_1, \dots, x_n, w_{i1}, \dots, w_{in}) = \sum_{j=1}^n w_{ij}x_j$$

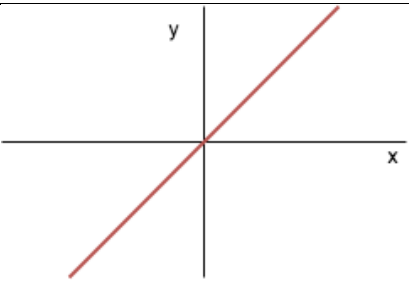
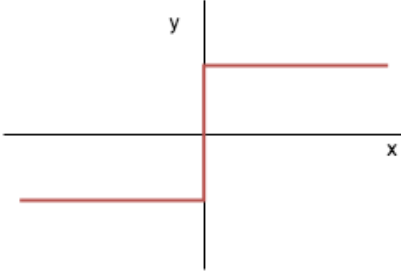
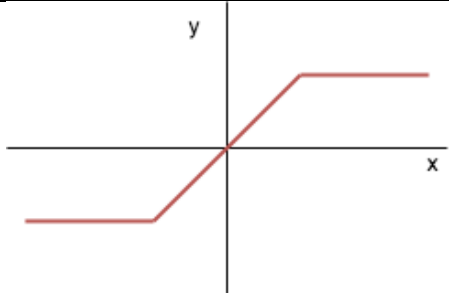
Además, es muy común añadir a la función de propagación un parámetro adicional  $\theta_i$ , que comúnmente denomina umbral, el cual se suele restar al resultado de la operación

$$h_i(x_1, \dots, x_n, w_{i1}, \dots, w_{in}) = \sum_{j=1}^n w_{ij}x_j - \theta_i$$

- Una función de activación, que representa simultáneamente la salida de la neurona y su estado de activación. De forma matemática la función de activación se puede definir:

$$y_i = f_i(x_i) = f_i\left(\sum_{j=1}^n w_{ij}x_j\right)$$

En la Tabla 2 se presentan algunas de las funciones de activación más comunes utilizadas por la red de neuronas. Normalmente se usan diferentes tipos de funciones de activación para cada una de las capas de la red.

Nombre	Función	Rango	Gráfica
Identidad	$y = x$	$[-\infty, +\infty]$	
Escalón	$y = \text{sign}(x)$	$\{-1, +1\}$	
	$y = H(x)$	$\{0, +1\}$	
Lineal	$y = \begin{cases} -1, & \text{si } x < -l \\ x, & \text{si } -l \leq x \leq +l \\ +1, & \text{si } x > +l \end{cases}$	$[-1, +1]$	
Sigmoidea	$y = \frac{1}{1 + e^{-x}}$	$[0, +1]$	

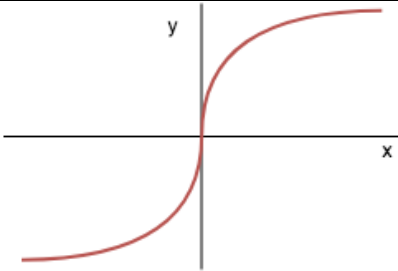
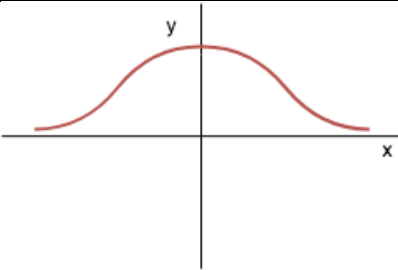
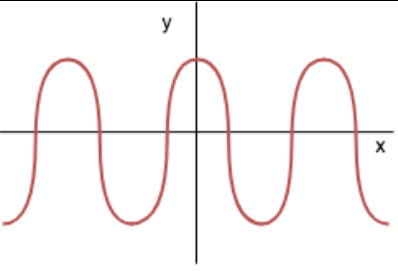
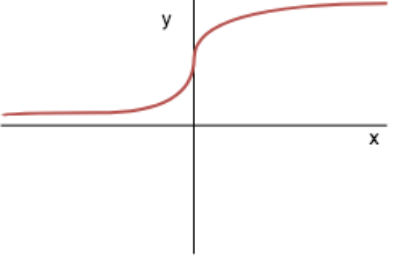
	$y = \tanh(x)$	$[-1, +1]$	
Gausiana	$y = Ae^{-Bx^2}$	$[0, +1]$	
Sinusoidal	$y = A \sin(\omega x + \varphi)$	$[-1, +1]$	
Softmax	$y = \frac{e^{x_i}}{\sum_{j=1}^i e^{x_j}}$	$[0, +1]$	

Tabla 2 - Funciones de activación más comunes

### 3.4.1. Arquitecturas de las redes de neuronas

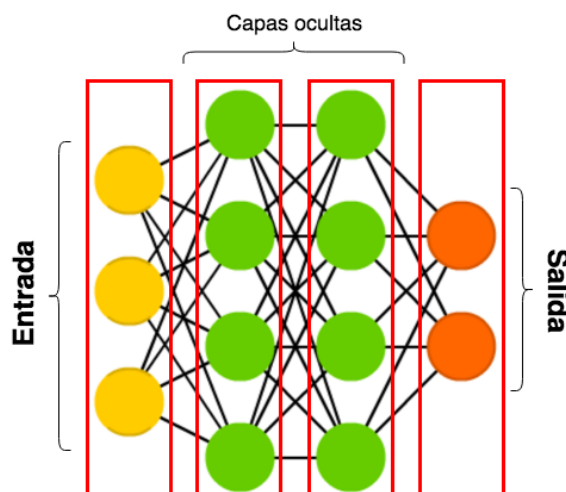
Una vez definidos los componentes básicos, es decir las neuronas, es necesario definir como se encuentran distribuidos y conectadas en la red. La distribución de las diferentes conexiones de la red determina en gran medida el funcionamiento de la red de neuronas. Los distintos nodos (neuronas) que componen la red se encuentran conectados entre sí mediante las sinapsis. Estas conexiones son unidireccionales, es decir, la información sólo se propaga en un único sentido desde la neurona presináptica a la neurona postináptica. En general, las neuronas se agrupan



entre sí en base a la función de propagación que utilizan en entidades más grandes denominadas capas. La combinación secuencial de diferentes capas constituye una red de neuronas artificial. En cualquier red de neuronas artificial se pueden distinguir tres tipos de capas:

- Capa de entrada o sensorial: Es la capa inicial de la red de neuronas y está compuesta por neuronas que reciben información (datos o señales) del entorno.
- Capa de salida: Es la capa final de la red de neuronas y genera la salidad o respuesta de la red.
- Capa oculta: Es la capa o capas intermedias de la red de neuronas donde se realizan las diferentes operaciones matemáticas que permiten a la red procesar y manipular la información con el fin de generar la salida de la red de neuronas. El caso más sencillo es que sólo exista una capa, pero pueden incluirse todas las capas que sean necesarias. La inclusión de múltiples capas ocultas ha dado lugar a las redes de neuronas profundas que se han popularizado en los últimos años mediante el Aprendizaje profundo (Deep Learning). Aunque este tipo de arquitecturas son muy eficientes para ciertos procesos aumenta en gran medida la complejidad de la red de neuronas.

En la **Figura 10** se presenta un ejemplo de una red de neuronas unidireccional compuesta por 4 capas (red multicapa), donde se presenta una capa con tres neuronas de entrada, dos capas ocultas con 4 neuronas completamente conectadas y una capa de salida con 2 neuronas.



**Figura 10:** Ejemplo de una red multicapa formada por una capa de entrada, 2 capas ocultas y una capa de salida

Dependiendo del número de capas ocultas que incluyamos en nuestra red de neuronas artificial podemos hablar de redes monocapa que sólo están compuesta por una única capa oculta o redes multicapa donde existen múltiples capas ocultas. Además, dependiendo de las conexiones entre las diferentes capas podemos distinguir entre tres tipos de redes de neuronas artificiales:

- ✓ **Redes Unidireccionales (feedforward):** Las redes de neuronas unidireccionales o prealimentadas (debido a que las neuronas obtienen información de la capa anterior o del entorno) son la primera y más sencilla estructura de red neuronal artificial construida por el ser humano. En esta estructura de red, la información se mueve en una única dirección. Es decir, de la capa de entrada, a través de las diferentes capas ocultas (si existen) hacia las capas de salida. En este tipo de redes no existe ningún tipo de bucle o retroalimentación hacia atrás. Este tipo de redes, a pesar de su sencillez, ofrecen muy buenos resultados en procesos en los cuales no se requiera el uso de información de eventos futuros dentro de la red.
- ✓ **Redes Elman (simple feedback):** Las redes de neuronas Elman son denominadas redes recurrentes simples ya que incluyen retroalimentación entre las capas contiguas. Es decir, en este tipo de red poseen una memoria de los eventos inmediatamente anteriores. Esta información es utilizada en el proceso de actualización de los pesos durante el proceso de aprendizaje.
- ✓ **Redes recurrentes (full feedback):** Las redes de neuronas recurrentes completamente conectadas, a diferencia de las redes de neuronas Elman, tienen conexiones de retroalimentación a todos los elementos que forman la red. Cada neurona de la red está conectada a todos los elementos que la rodean desde un punto de vista espacial. Es decir, una neurona está conectada a las neuronas de las capas posteriores y anteriores y a sí misma.

#### 3.4.2. Proceso de entrenamiento

Una vez que se ha definido la estructura de la red de neuronas es necesario computar el valor de los diferentes pesos de cada una de las capas de la red, para ello se utiliza un proceso de entrenamiento compuesto por dos fases:

- **Fase de entrenamiento:** En esta fase se calcula el valor de los pesos de la red, para ello se utiliza un conjunto de datos o patrones (ejemplos de entrenamiento) que se utilizan para calcular los pesos (parámetros) que definen el modelo de la red de neuronas. Los

diferentes pesos son calculados de manera iterativa, de acuerdo con los valores de los ejemplos entrenamiento, con el objeto de minimizar el error cometido entre la salida obtenida por la red neuronal y la salida deseada.

- Fase de test: En esta se realiza un análisis de la calidad del proceso de entrenamiento, es decir se analiza si el modelo construido en la fase anterior es capaz de generalizar a casos nuevos o se ha ajustado demasiado a las particularidades presentes en los ejemplos de entrenamiento (sobreajuste). Para evitar el problema del sobreajuste, es aconsejable utilizar un segundo grupo de datos diferentes a los de entrenamiento, el grupo de validación, que permita controlar el proceso de cálculo de los pesos.

### 3.4.3. Tipos de redes neuronas

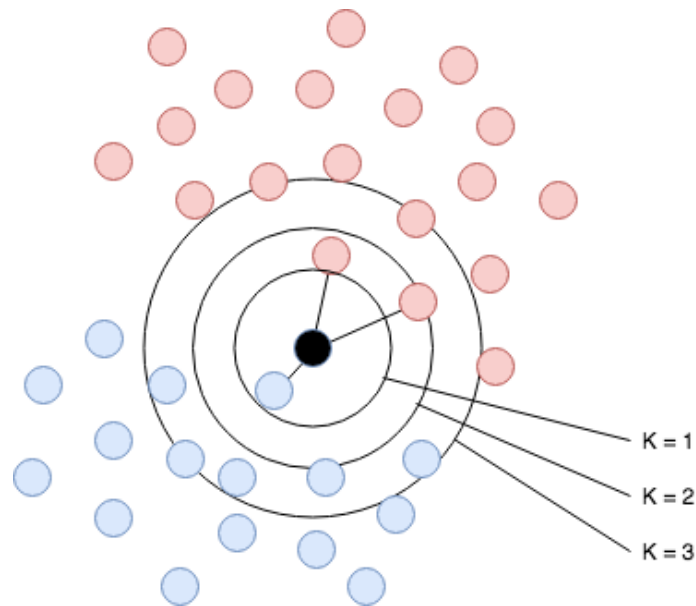
Las redes de neuronas se pueden construir de diferentes maneras dependiendo del número de capas, el tipo de neuronas utilizadas y la forma en la que están conectadas las diferentes capas. Pero existen una serie de configuraciones básicas que identifican ciertos tipos de redes de neuronas, algunas de las cuales son descritas a continuación:

- Perceptrón simple: El perceptrón (Perceptron, P en sus siglas en inglés) es la red de neuronas más sencilla formada por sólo dos capas (entrada y salida) que permite construir un discriminador simple.
- Perceptrón multicapa: El perceptrón multicapa (Multilayer Perceptron MP, en sus siglas en inglés) es una red de neuronas basada en el perceptrón simple que introduce el concepto de capa oculta. La estructura básica de esta red está formada por al menos tres capas (entrada, oculta y salida) cuya interacción permite construir un aproximador de tipo universal. Es decir, permite aproximar cualquier tipo de función continua definida en un espacio  $R^n$ .
- Redes recurrentes: Las redes de neuronas recurrentes (Recurrent Neural Networks, RNN en sus siglas en inglés) son una extensión del perceptrón multicapa donde se producen conexiones entre las neuronas en ambos sentidos (hacia delante y hacia atrás). Existe dos tipos de redes recurrentes: simples que sólo tienen conexión con las neuronas de las capas continuas; y completas que pueden estar conectadas con todas las neuronas de las diferentes capas incluidas ellas mismas.
- Redes largas de memoria a corto plazo: Las redes de neuronas largas con memoria a corto plazo (Long Short-Term Memory, LSTM en sus siglas en inglés) son un tipo de red de neuronas que incluye bloques de memoria que permite recordar cierto tipo de información.

- **Redes convolucionales:** Las redes de neuronas convolucionales (Convolutional Neural Networks, CNN en sus siglas en inglés) son un tipo de red de neuronas similar al perceptrón multicapa donde las neuronas de algunas de sus capas son de tipo convolucional, de reducción del muestreo y sencillas. Las neuronas convolucionales tratan de simular el funcionamiento de las neuronas de la corteza visual humana mediante la aplicación de filtros (operaciones matriciales) que permiten la extracción de características de los datos de entradas que se corresponden con los píxeles de una imagen.
- **Redes generativas antagónicas:** Las redes de neuronas generativas antagónicas (Generative Adversarial Network, GAN en sus siglas en inglés) son una combinación de dos redes de neuronas que compiten mutuamente en un juego de suma cero. El cual consiste una situación en la que la ganancia o pérdida de un participante se equilibra con exactitud con las pérdidas o ganancias del otro participante.

### 3.5 K-Nearest Neighbour

El algoritmo de K-vecinos más cercanos (K-Nearest Neighbour, KNN en sus siglas en inglés) es un método de aprendizaje no parametrizado para la generación de clasificadores o regresores basados en el concepto de cercanía [16]. Este algoritmo no produce ningún tipo de modelo tras la fase de aprendizaje, sino que almacena todas las instancias utilizadas para el proceso de aprendizaje en una estructura de datos que es consultada con el fin de predecir el valor de salida (clase o valor) de nuevas instancias.



**Figura 11:** Ejemplo del funcionamiento del proceso de clasificación del algoritmo KNN utilizando un diferente valor de K.

El funcionamiento del algoritmo es muy sencillo una vez que se ha producido la generación de la estructura de datos que almacena las instancias de entrenamiento. Dada una nueva instancia que debe ser clasificada se calcula la distancia a todas las instancias almacenadas en la estructura de datos, las cuales son ordenadas en base al valor de distancia obtenido de menor a mayor. A continuación, se seleccionan las  $k$  instancias más cercanas y se asigna a la nueva instancia la clase más frecuente de entre las instancias seleccionadas como más cercanas en el caso de que se esté realizando un proceso de clasificación; o se pondera el valor de salida en base a los valores de las instancias seleccionadas como más cercanas en el caso de que se esté realizando un proceso de regresión. En la **Figura 11** se presenta un ejemplo visual de un proceso de clasificación para una nueva instancia utilizando un algoritmo KNN de forma que se calcula la distancia al resto de instancia. En este caso se puede observar como el valor de la clase de la nueva instancia varía dependiendo del valor de  $k$  que seleccionemos.

- Para  $k = 1$  la nueva instancia (negro) sería clasificada como azul ya que la instancia más cerca es azul.
- Para  $k = 2$  la nueva instancia (negro) podría ser clasificada como roja o como azul ya que las dos instancias más cercanas pertenecen a dos clases diferentes. Sería necesario definir algún tipo de técnica que permitiera eliminar el empate.
- Para  $k = 3$  la nueva instancia (negro) sería clasificada como roja ya que de las tres instancias más cercanas dos son rojas y una es azul.

Teniendo en cuenta el funcionamiento observado en el ejemplo descrito en la **Figura 11** es necesario configurar el algoritmo KNN en base a dos elementos:

- El valor  $k$  que suele ser seleccionado manual tras el proceso de aprendizaje, pero en algunos casos es posible calcular el valor mediante algún tipo de algoritmo en base a la distribución de las instancias del conjunto de entrenamiento.
- La función de cálculo de la distancia que permitirá definir la similitud entre dos instancias. La función más común es la distancia euclídea presentada en la siguiente ecuación

$$d(x_i, x_j) = \sqrt{\sum_{r=1}^p (x_{ri} - x_{rj})^2}$$

KNN es uno de los algoritmos más utilizados para clasificación debido principalmente a su sencillez. El tiempo de clasificación de nuevas instancias puede ser muy elevado si el conjunto de instancias de entrenamiento es muy grande, debido a que se debe realizar el cálculo de la distancia con todos los elementos almacenados en la estructura de datos durante el proceso de aprendizaje. Existen diferentes versiones del algoritmo KNN que intentan explotar los dos elementos que pueden ser configurados, algunas de estas versiones son: (1) KNN con rechazo; (2) KNN con distancia media; y (3) KNN con distancia mínima.

## 4. CONCLUSIONES

El Aprendizaje Automático es una de las técnicas más prometedoras de la inteligencia artificial ya que nos permite construir modelos de razonamiento en base a la información disponible del entorno de forma similar a como creemos que funcionan los procesos de aprendizaje humanos. Aunque como muchas otras técnicas de Inteligencia Artificial es extremadamente sensible y dependiente a la forma en la que la información del entorno es modelada. En el caso del aprendizaje automático la definición de los datos, tanto a nivel de estructura como a nivel de suficiente número de ejemplos representativos del concepto que se quiere aprender, es muy importante ya que una mala definición de los atributos de la información utilizada durante el proceso de aprendizaje puede producir la extracción de conclusiones erróneas o producir modelos que no generalizando correctamente bien debido a la falta de información o a la utilización de información sesgada.

Algunos ejemplos de este tipo de problemas pueden observarse en la propia definición de los diferentes métodos Aprendizaje Supervisado descritos en este tema. Este tipo de técnicas



necesitan información precisa (etiquetas) acerca de los diferentes conceptos que representan la información que se les suministra para conseguir construir un modelo. Es decir, todos los ejemplos utilizados para el proceso de aprendizaje deben estar etiquetados con el fin de identificar sus características y relacionarlas con las etiquetas y poder predecir la etiqueta de futuros ejemplos no etiquetados. Este tipo de proceso presenta dos importantes problemas:

- El proceso de etiquetado de información del mundo real de manera lo suficientemente precisa con el fin de construir modelos muy específicos es muy complicada y conlleva un gran esfuerzo a nivel de recursos físicos. Por ejemplo, si quisiéramos crear un clasificador para todos los tipos de árboles que existen en el planeta tierra deberíamos tomar información específica de todos los posibles atributos de cada árbol, etiquetarlos y construir un conjunto de entrenamiento lo suficientemente representativo para crear un modelo que nos permita clasificar cualquier tipo de árbol y además deberíamos también crear un conjunto de datos de test con el fin de analizar la eficacia de nuestro modelo.
- La definición de los atributos que describen las instancias. Cuantos atributos y cuales son necesarios para describir de forma correcta un concepto, como es posible identificar toda la información que es necesario para identificar correctamente cada uno de los distintos tipos de árboles que hay en el planeta tierra. Este es otro de los grandes problemas del aprendizaje supervisado el cual sigue dependiendo demasiado de la forma en que los humanos describamos la información.

Pero tras conocer cómo funciona el aprendizaje supervisado nos debemos preguntar que si es posible etiquetar de forma precisa cualquier concepto o si somos capaces de definir de forma manual todos los atributos necesarios para describir estos conceptos. En el próximo tema describiremos cómo funcionan los diferentes métodos de Aprendizaje No Supervisado que son capaces de construir modelos mediante el análisis de las similitudes entre los atributos de los conjuntos de instancias sin la necesidad de conocer la etiqueta de cada una de estas instancias. Además, describiremos que es el Aprendizaje Profundo y cómo es capaz de extraer información y crear nuevas características de forma automática con el fin de mejorar el proceso de aprendizaje. Para finalizar aprenderemos que es el Aprendizaje por Refuerzo que intenta simular la forma en la que los seres vivos aprendemos mediante la utilización de recompensas que nos permiten aprender como interactuar de forma física en el entorno.

## 5. REFERENCIAS

- [1] M. Minsky. (1967). Computation. Finite and infinite machines. Prentice Hall.
- [2] M. Minsky. (1969). Perceptrons: an introduction to computational geometry. MIT Press, Cambridge, Massachusetts.
- [3] Schmidhuber, J. (2015). Deep Learning in Neural Networks: An Overview. Neural Networks. Volumen 61 pp 85-117.
- [4] Tom M. Mitchell. (1997) Machine Learning. McGraw-Hill, New York.
- [5] Quinlan, J.R. (1986) Induction of decision trees. Machine Learning, Volumen 1(1), pp 81-106.
- [6] Shannon, C.E. (1948) A mathematical theory of communication. Bell System Technical Journal, volumen 27, pp 379–423 y 623–656.
- [7] Utgoff Paul E. (1989) Incremental Induction of Decision, Trees, Machine Learning, volumen 4, pp 161-186.
- [8] Izenman, A.J. (2008) Modern Multivariate Statistical Techniques. Springer.
- [9] Duda, R. O., Hart, P. E., and Stork, D. G. (2001). Pattern Classification (2nd edición). John Wiley.
- [10] Arthur S. (1959) Some Studies in Machine Learning Using the Game of Checkers. IBM Journal of Research and Development. Volumen 3(3). pp 210–229.
- [11] Copi, I. M.; Cohen, C.; Flage, D. E. (2006). Essentials of Logic (Second edition). Upper Saddle River, NJ: Pearson Education. ISBN 978-0-13-238034-8.
- [12] Sternberg, R. J. (2009). Cognitive Psychology. Belmont, CA: Wadsworth. ISBN 978-0-495-50629-4.
- [13] Watson I. and Marir F. (1994). Case-based reasoning: A review, The Knowledge Engineering Review, volumen 9(4), pp 327-354.
- [14] Dean J. and Ghemawat S. (2004). MapReduce: simplified data processing on large clusters, Proceedings of the 6th conference on Symposium on operating systems design and implementation.
- [15] Zaharia M., Chowdhury M., Franklin M. J., Shenker S. and Stoica I. (2010). Spark: Cluster Computing with Working Sets. HotCloud 2010.

- [16] Altman, N. S. (1992). An introduction to kernel and nearest-neighbor nonparametric regression. *The American Statistician*. Volumen 46(3).
- [17] Ho, Tin Kam (1995). Random Decision Forests. *Proceedings of the 3rd International Conference on Document Analysis and Recognition*, Montreal, QC, pp.278–282.
- [18] Freedman D. A. (2009). *Statistical Models: Theory and Practice*. Cambridge University Press.
- [19] Haykin S. (1999). *Neural Networks: A Comprehensive Foundation*. Prentice Hall.
- [20] McCulloch, W. and Pitts W. (1943). A Logical Calculus of Ideas Immanent in Nervous Activity. *Bulletin of Mathematical Biophysics*. Volumen 5(4). pp 115-133.