

# **Métodos de aprendizaje automático. Aplicación al estudio de la despoblación en Aragón**



**Paula Arán Godés**  
Trabajo de fin de grado en Matemáticas  
Universidad de Zaragoza

Director del trabajo: Ricardo López Ruiz  
25 de marzo de 21



# Resumen

Este trabajo se centra en el aprendizaje automático, concretamente en las redes neuronales y su aplicación al estudio de la despoblación en Aragón.

La inteligencia artificial es la parte de la informática que estudia las técnicas que sirven para conseguir que el comportamiento de una máquina se asemeje al de un humano. En particular, el aprendizaje automático es la rama de la inteligencia artificial que estudia los algoritmos que permiten que las máquinas realicen un aprendizaje.

En el primer capítulo de este trabajo, se introducen los dos tipos de aprendizaje (supervisado y no supervisado) y se explican brevemente algunos de los algoritmos más utilizados en cada uno de ellos. En el segundo capítulo, nos centraremos en uno de los algoritmos de aprendizaje automático más utilizados en la actualidad: las redes neuronales. Para comenzar, veremos la relación entre las neuronas biológicas y las neuronas artificiales, elementos esenciales de las redes neuronales.

En 1943, McCulloch y Pitts presentaron el primer modelo de neurona artificial, una unidad de cálculo que, dado un vector de entrada, mediante la combinación lineal con otro vector denominado "de pesos sinápticos", y la aplicación de una función denominada "de activación", emite un vector de salida. Posteriormente, en 1958, Frank Rosenblatt desarrolló un nuevo modelo de neurona artificial basado en la neurona de McCulloch y Pitts y el aprendizaje de Hebb y lo denominó perceptrón simple. Esta neurona, sirvió como unidad esencial para la construcción de redes neuronales artificiales.

Así, las redes neuronales artificiales (ANN) son sistemas formados por neuronas artificiales conectadas entre sí. Estos sistemas están formados por tres tipos de capas: la capa de entrada, a la que se le proporciona la información inicial, la capa o capas ocultas, en las que se procesa la información, y la capa de salida, de la cual extraemos el resultado al final del proceso.

La primera red neuronal que se desarrolló fue el denominado perceptrón multicapa, compuesto por varias capas de perceptrones simples. Esta red se entrena con el algoritmo de propagación hacia atrás, que utiliza la técnica de descenso de gradiente. Posteriormente, tenemos las redes neuronales recurrentes, cuya principal característica es la introducción de la variable temporal, las redes neuronales convolucionales, que se caracterizan por realizar operaciones con la función convolución, y las redes neuronales generativas adversarias, que combinan dos redes neuronales, una para generar muestras y otra para analizar su calidad.

Finalmente, el *deep learning* es una rama del aprendizaje automático que utiliza redes neuronales profundas, es decir, redes que constan de un gran número de capas ocultas.

Para terminar, en el tercer capítulo, se introduce la herramienta *BigML*, que sirve para aplicar algoritmos de aprendizaje automático a un determinado conjunto de datos. Además, se presenta el problema de la despoblación en Aragón y se introducen tablas que presentan distintas variables demográficas y socioeconómicas que afectan a dicho problema. Finalmente, se utiliza el algoritmo *deeppnet* para averiguar cuáles de estas variables afectan más a las variaciones poblacionales y se extraen conclusiones en base a los resultados obtenidos.



# Abstract

This project is focused on machine learning, in particular on neural networks and their applications to the study of depopulation in Aragón.

Artificial intelligence is the branch of computer science that studies the techniques that allow us to make machine behaviour similar to human behaviour. Specifically, machine learning is the branch of artificial intelligence that studies the algorithms used to train machines.

In the first chapter, we introduce both types of learning, supervised and unsupervised, and we briefly explain the algorithms used in each type of learning. In the second chapter, we focus on one of the most widely used machine learning algorithms: artificial neural networks. Firstly, we will show the relationship between biological neurons and artificial neurons, which are essential elements of neural networks.

In 1943, McCulloch and Pitts introduced the first artificial neuron model, a calculus unit that given an input vector, by means of the linear combination with another vector called "synaptic weights" and the application of an "activation map", results in an output vector. Later, in 1958, Frank Rosenblatt developed a new model of the artificial neuron based on McCulloch and Pitts's neuron and Hebb's learning, which he called simple perceptron. This neuron became the essential unit for the construction of artificial neural networks.

Artificial neural networks (ANN) are artificial neural systems connected to each other. The systems consist of three types of layers: the input layer, to which the initial dataset is provided, the hidden layer or layers, in which the dataset is processed, and the output layer, from which we obtain the final result at the end of the process.

The first neural network developed was the multilayer perceptron, composed of several layers of simple perceptrons. The network is trained with the backpropagation algorithm, which uses the descent gradient technique. Then came recurrent neural networks, whose main characteristic is the introduction of the temporal variable, convolutional neural networks, characterized by carrying out operations with the convolutional map, and generative adversarial networks, which combine two neural networks, one to generate samples and the other to analyze their quality.

Finally, deep learning is the branch of machine learning that uses deep neural networks, that is, networks that consist of a large number of hidden layers.

In the third chapter, we introduce the BigML tool, which is used to apply algorithms of machine learning to a particular dataset. Furthermore, we introduce the issue of depopulation in Aragón and display some tables with different demographic and socioeconomic variables affecting problem. Lastly, we use the deepnet algorithm to discover which of these variables have a bigger impact on population variation and to draw conclusions from the results obtained.



# Índice general

<b>Resumen</b>	<b>III</b>
<b>Abstract</b>	<b>v</b>
<b>1. Aprendizaje automático</b>	<b>1</b>
1.1. Introducción . . . . .	1
1.2. Aprendizaje supervisado . . . . .	2
1.2.1. Tipos de problemas de aprendizaje supervisado . . . . .	2
1.2.2. Algoritmos de aprendizaje supervisado . . . . .	2
1.3. Aprendizaje no supervisado . . . . .	4
1.3.1. Tipos de técnicas de aprendizaje no supervisado . . . . .	4
1.3.2. Algoritmos de aprendizaje no supervisado . . . . .	4
<b>2. Redes neuronales y <i>deep learning</i></b>	<b>6</b>
2.1. La neurona artificial . . . . .	6
2.1.1. Introducción . . . . .	6
2.1.2. Modelo de McCulloch y Pitts . . . . .	6
2.1.3. Perceptrón simple . . . . .	7
2.2. Redes neuronales artificiales (ANN) . . . . .	9
2.2.1. Perceptrón multicapa . . . . .	9
2.2.2. Redes neuronales recurrentes (RNN) . . . . .	12
2.2.3. Redes neuronales convolucionales (CNN) . . . . .	13
2.2.4. Redes neuronales generativas adversarias . . . . .	14
2.3. <i>Deep learning</i> . . . . .	15
<b>3. Aplicación a la despoblación en Aragón.</b>	<b>17</b>
3.1. BigML . . . . .	17
3.2. Planteamiento del problema . . . . .	19
3.3. Búsqueda y tratamiento de datos . . . . .	19
3.4. Resultados y conclusiones . . . . .	22
3.4.1. Conclusiones . . . . .	23
3.4.2. Propuestas de mejora . . . . .	24
<b>A. Anexo I. Fórmulas utilizadas en la construcción de variables indirectas</b>	<b>26</b>
A.1. Variables demográficas . . . . .	26
A.2. Variables socioeconómicas . . . . .	27
<b>B. Anexo II. Tabla completa de datos</b>	<b>28</b>



# Capítulo 1

## Aprendizaje automático

### 1.1. Introducción

La **inteligencia artificial** (IA) es una rama de la informática en la que se estudian las técnicas para conseguir que una máquina se comporte de forma similar a un ser humano.

La inteligencia artificial surge en la década de 1950 gracias al trabajo del matemático británico Alan Turing. Éste publicó un artículo llamado *Computing machinery and intelligence* (pon referencia) en el que se planteaba si una máquina puede pensar y proponía una técnica para determinarlo. Dicha técnica se denominó Test de Turing y en la actualidad siguen publicándose trabajos de investigación sobre la misma.

Las aplicaciones de la inteligencia artificial son múltiples y abarcan diversos campos científicos. Por ejemplo, en medicina puede servir como sistema de apoyo para tomar decisiones en diagnósticos clínicos, en tecnologías de la información y la comunicación para desarrollar aplicaciones de reconocimiento de voz o gestos, en economía para ayudar en la organización de distintas operaciones bancarias y, en general, en cualquier actividad que la inteligencia humana pueda desarrollar.

Así, vamos a centrarnos en una de las ramas de la inteligencia artificial conocida como **aprendizaje automático** o *machine learning*. El objetivo de este área es el desarrollo de técnicas que sirvan para que las máquinas aprendan. Consideramos que una máquina realiza un aprendizaje si es capaz de mejorar el desarrollo de una tarea cuando va adquiriendo experiencia en la misma. Por ejemplo, si pensamos en una partida de ajedrez en la que uno de los oponentes es un ser humano y el otro un ordenador, consideramos que el ordenador aprende cuando, tras varias partidas, es capaz de reaccionar de forma distinta al mismo escenario, mejorando así el resultado que obtiene.

El esquema conceptual de un proceso sencillo de aprendizaje automático es el siguiente

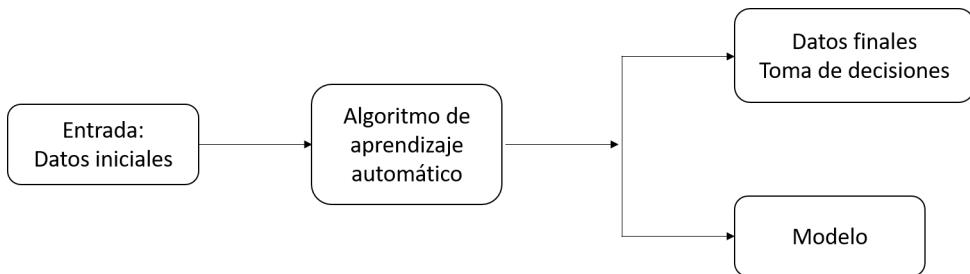


Figura 1.1: Esquema de un proceso de aprendizaje automático.

El aprendizaje automático puede lograrse a través de distintos modelos y algoritmos. Una clasificación general de los mismos distingue entre dos tipos de aprendizaje: el aprendizaje no supervisado y el aprendizaje supervisado. En las siguientes secciones desarrollaremos cada uno de ellos.

## 1.2. Aprendizaje supervisado

El aprendizaje supervisado se basa en entrenar a un algoritmo usando dos conjuntos de datos, los datos de entrada y los de salida. Dicho entrenamiento usa la información de ambos conjuntos con el objeto de que el algoritmo aprenda las relaciones existentes entre ellos. Una vez entrenado, el algoritmo aplica a un nuevo conjunto de datos de entrada con el objeto de predecir unos resultados de salida mejorados.

### 1.2.1. Tipos de problemas de aprendizaje supervisado

Los problemas que se resuelven con este tipo de aprendizaje se pueden clasificar en dos tipos: regresión y clasificación.

Los problemas de **regresión** son aquellos en los que los datos de salida son una variable continua. En este tipo de problemas buscamos que el resultado que obtenemos sea un valor numérico. Un ejemplo de un problema de este tipo es la predicción del precio que debe tener un determinado producto.

Los problemas de **clasificación** son los que tienen como resultado un valor perteneciente a una variable discreta. Por ejemplo, se nos puede presentar el problema de diagnóstico de la gravedad de un tumor. En este caso el conjunto de posibles resultados una vez se aplique el algoritmo correspondiente consta de dos elementos: maligno o benigno.

### 1.2.2. Algoritmos de aprendizaje supervisado

A continuación se explican de forma breve los distintos algoritmos utilizados en el aprendizaje supervisado.

- Regresión lineal

En este algoritmo se busca predecir la variable dependiente  $Y = (y_1, \dots, y_n)$  en función de los valores de la variable independientes  $X = (x_1, \dots, x_n)$ . Para realizar dicha predicción se busca una recta que se aproxime de la mejor forma posible a los puntos dados. Uno de los métodos para la construcción de dicha recta es el de mínimos cuadrados, que consiste en minimizar cierta función error  $E(X)$  que mida la distancia global de los puntos a la recta genérica  $y = ax + b$  que se busca como mejor ajuste. La función que más habitualmente se usa como función error es la raíz cuadrada de la suma de los cuadrados de las distancias verticales de los puntos a la recta

$$E(X) = \sqrt{(y_1 - ax_1 - b)^2 + \dots + (y_n - ax_n - b)^2}.$$

El problema se denomina de mínimos cuadrados porque en realidad se minimiza el cuadrado de dicha función, lo cual equivale matemáticamente a minimizar la función misma.

- Regresión logística

Este algoritmo es utilizado en problemas de clasificación y consiste en, dada una o más variables explicativas, estimar la probabilidad de una respuesta binaria. Este algoritmo es muy utilizado en problemas de aprendizaje automático debido a su eficiencia y su bajo coste computacional. Como veremos posteriormente, se trata de una red neuronal que consta de una única neurona.

- Árbol de decisión

Este tipo de algoritmo se utiliza principalmente en problemas de clasificación, aunque también se puede utilizar en los de regresión. Se puede representar mediante un diagrama en el que se parte de un nodo que da lugar a otros mediante ramificaciones, cada una de las cuales representa la toma de una decisión. La toma de decisiones se realiza utilizando distintos criterios como pueden ser la ganancia de información o el cálculo de la impureza de Gini.

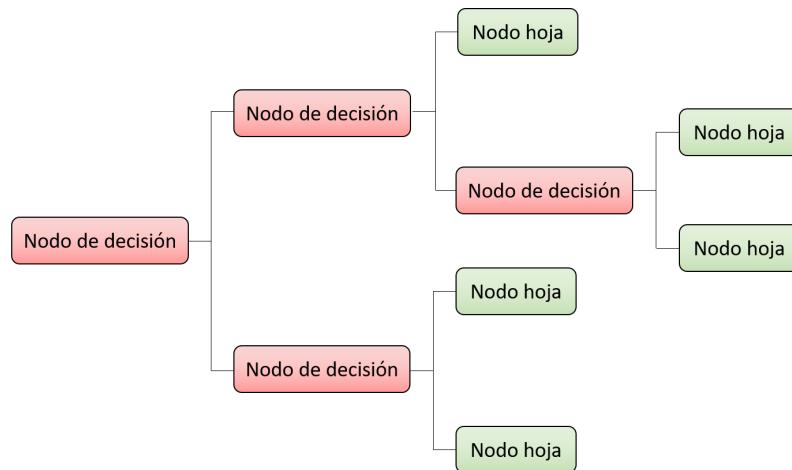


Figura 1.2: Diagrama de un árbol de decisión.

#### ■ Bosque aleatorio

Está formado por un conjunto de árboles aleatorios que se combinan utilizando *bagging*. Esta técnica consiste en que cada uno de los árboles de decisión que pertenece al algoritmo se entrena con un conjunto distinto de datos, que vienen dados por un vector aleatorio. Finalmente combinamos el resultado de estos datos y llegamos a una mejor aproximación ya que los posibles errores se compensan. Para la combinación se puede utilizar la media aritmética o bien el método *soft-voting* en el que se da más importancia a árboles que estén más seguros.

#### ■ Máquinas de vectores de soporte

Dados dos conjuntos de datos en  $\mathbb{R}^n$ , este algoritmo busca generar un hiperplano  $n - 1$ -dimensional de modo que dichos conjuntos queden separados. Con esta premisa podemos tener varias soluciones, por tanto, lo que buscamos es la solución óptima, es decir, el hiperplano que esté situado a la mayor distancia posible de cada uno de los datos.

#### ■ Clasificación de Naïve Bayes

Es un algoritmo de clasificación basado en el teorema de Bayes que establece que, dado un conjunto de sucesos independientes y exhaustivos  $\{A_1, A_2, \dots, A_n\}$  tales que  $P(A_i) \neq 0, \forall i = 1, 2, \dots, n$  y siendo  $B$  un suceso tal que conocemos la probabilidad condicionada  $P(B|A_i)$ , entonces

$$P(A_i|B) = \frac{P(B|A_i)P(A_i)}{P(B)}.$$

Así, dadas las probabilidades necesarias, se calcula la probabilidad condicionada dada por la fórmula anterior para cada uno de los datos y el resultado es la más alta de estas probabilidades.

#### ■ Redes neuronales y *deep learning*

Se denomina red neuronal al algoritmo que se construye superponiendo varias neuronas artificiales. Cada una de las neuronas de la red realiza el siguiente proceso: a partir de un conjunto de datos de entrada, y considerando un vector de parámetros denominados pesos sinápticos, da como resultado la combinación lineal de los dos conjuntos de valores que finalmente al pasar por una función de activación producen la salida o respuesta de la neurona.

El *deep learning* es un caso particular de red neuronal que consta de múltiples capas de neuronas conectadas entre sí. Estas capas se organizan mediante un sistema jerárquico, es decir, las últimas capas son mucho más específicas y complejas que las primeras.

En el segundo capítulo de este trabajo desarrollaremos con mayor profundidad los conceptos de red neuronal y de *deep learning*.

## 1.3. Aprendizaje no supervisado

El aprendizaje no supervisado es una técnica de aprendizaje automático en la que se entrena naranjos en base a un conjunto de datos de entrada. La principal diferencia con el aprendizaje supervisado es que en este caso no tenemos ninguna información acerca del conjunto de datos de salida, por lo que no podemos utilizarlos para el entrenamiento. De este modo, no podemos aplicar un algoritmo de este tipo a los problemas de regresión y clasificación vistos anteriormente.

### 1.3.1. Tipos de técnicas de aprendizaje no supervisado

Este tipo de aprendizaje automático se utiliza para resolver tres tipos de problemas: agrupamiento o *clustering*, asociación y reducción de dimensiones.

El **agrupamiento** o *clustering* es una técnica que consiste en, dado un conjunto de datos, realizar agrupaciones de los mismos de modo que cada grupo tenga propiedades o características similares. Este método se utiliza en problemas de análisis de datos estadísticos.

La técnica de **asociación** se utiliza cuando queremos observar la relación entre características. Así, sean dos características  $X$  y  $Y$ , la aplicación de esta técnica busca conocer cuánto afecta la presencia de  $X$  a la presencia de  $Y$ .

El método de **reducción de dimensiones** busca cuantificar la cantidad de información que proporciona cada uno de los datos de entrada. Así, se puede encontrar información que esté correlacionada con otra de modo que podemos representar los datos de una forma más simplificada.

### 1.3.2. Algoritmos de aprendizaje no supervisado

Del mismo modo que con el aprendizaje supervisado, vamos a explicar brevemente algunos de los algoritmos más utilizados en el aprendizaje no supervisado.

- *K-means*

Este algoritmo es el más utilizado en problemas de agrupamiento. Sea un conjunto de vectores  $d$ -dimensionales  $\{x_1, \dots, x_n\}$  se busca construir una partición de los vectores en  $k$  grupos ( $k \leq n$ ) de modo que la suma de los cuadrados dentro de cada grupo sea mínima. Es decir, sea  $\mu_i$  el centroide del grupo  $S_i$   $\forall i = 1, \dots, k$  buscamos minimizar la función

$$\sum_{i=1}^k \sum_{x_j \in S_i} \|x_j - \mu_i\|^2.$$

Este algoritmo es rápido y eficiente pero depende del valor que se asigne a  $k$ , es decir, del número de grupos que se construyen. Así, un valor grande de  $k$  hace que el error sea menor, pero a cambio la información que proporciona cada uno de los grupos disminuye.

- DBSCAN

Estas siglas significan agrupamiento espacial basado en densidad de aplicaciones con ruido y es un algoritmo utilizado en problemas de agrupamiento. Para aplicarlo a un conjunto de puntos necesitamos establecer dos parámetros:  $\epsilon$  (el radio de los entornos de los puntos de un grupo que serán observados) y minPts (el número mínimo de puntos requeridos para que una región sea considerada densa).

Se procede de la siguiente forma: comenzamos desde un punto  $P$  y visitamos todos los puntos que pertenezcan a un entorno que esté a distancia  $\epsilon$  de  $P$ . Supongamos que en este entorno existen  $n$  puntos. Si  $n < \text{minPts}$  el punto  $P$  se etiqueta como ruido, en cambio si  $n \geq \text{minPts}$  se crea un grupo sobre  $P$  que contiene a todos los puntos pertenecientes a su entorno. A continuación, se visitan los entornos de los  $n$  puntos del entorno de  $P$  y todos los puntos que se encuentren en dichos entornos se añaden al grupo.

Así, se continúa hasta que no se encuentren más puntos en los entornos, entonces se visita un nuevo punto y se procede del mismo modo. Al terminar tendremos todos los puntos del conjunto clasificados o bien como ruido, o bien como parte de un grupo.

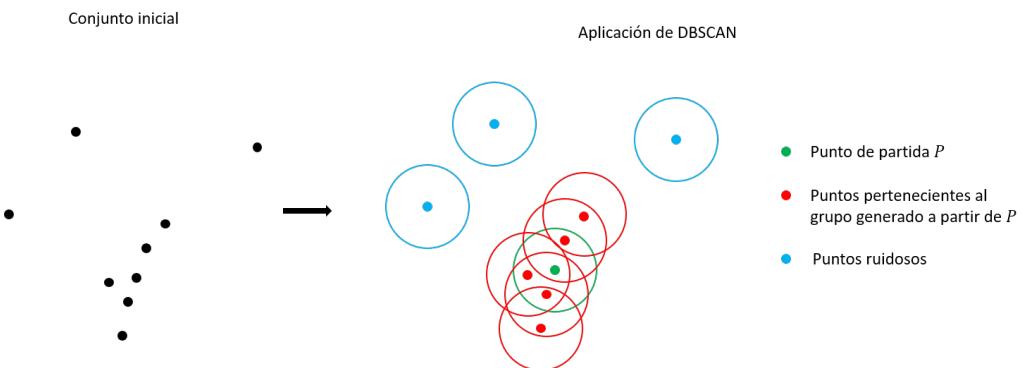


Figura 1.3: Ejemplo visual del algoritmo DBSCAN.

#### ■ Apriori

El algoritmo apriori se utiliza para resolver problemas de asociación. Para poder explicar en qué consiste, hemos de introducir los conceptos de soporte, confianza y *lift*.

Consideramos un problema que consta de  $n$  sucesos en el que el suceso  $A$  aparece  $n_1$  veces, el suceso  $B$  aparece  $n_2$  veces y los dos juntos aparecen  $n_3$  veces. En este caso el soporte de  $A$  es el número de veces que ocurre dividido por el número total de sucesos, es decir  $\frac{n_1}{n}$ . La confianza de  $B$  respecto de  $A$  es el número de veces que aparecen ambos sucesos a la vez dividido entre las veces que aparece  $A$ , es decir  $\frac{n_3}{n_1}$ . Finalmente el *lift* indica la probabilidad de que aparezca  $B$  cuando aparece  $A$  y es la confianza de  $B$  respecto a  $A$  dividido por el soporte de  $A$

$$\frac{\frac{n_3}{n_1}}{\frac{n_1}{n}} = \frac{n_3 n}{n_1^2}.$$

Para aplicar el algoritmo, primero se establece un valor mínimo para el soporte y la confianza y se toman los datos cuyo valor del soporte es mayor al establecido. Ahora extraemos los subconjuntos cuyo valor de confianza sea mayor al mínimo fijado y seleccionamos sus reglas, es decir, supongamos que en nuestro ejemplo la confianza de  $B$  respecto de  $A$  está por encima del umbral que hemos fijado, por tanto la regla seleccionada es  $A \Rightarrow B$ . Finalmente una vez hemos seleccionado todas las reglas, las ordenamos por orden descendente de *lift*.

#### ■ Análisis de componentes principales (PCA)

Este algoritmo se utiliza en problemas de reducción de dimensiones. Para comenzar tenemos un conjunto de  $n$  muestras, cada una de las cuales viene representada por  $m$  variables y queremos modificarlo de modo que vengan representadas por  $k$  variables con  $k < m$ .

El primer paso es normalizar estas muestras, calcular su matriz de covarianza y obtener los valores y vectores propios de esta matriz. En segundo lugar, se ordenan los valores propios, se eligen los  $k$  que tengan mayor valor y se construye la matriz de proyección con los  $k$  vectores propios asociados. Finalmente transformamos las muestras iniciales utilizando la matriz de proyección y obtenemos el nuevo conjunto de muestras.

# Capítulo 2

## Redes neuronales y *deep learning*

### 2.1. La neurona artificial

#### 2.1.1. Introducción

Biológicamente, las neuronas son células pertenecientes al sistema nervioso que constan de un cuerpo celular, del cual sale una rama principal denominada axón y unas prolongaciones cortas denominadas dendritas. La principal característica de las neuronas es la capacidad de recibir y emitir información.

Este proceso se denomina neurotransmisión y se realiza de la siguiente forma: una neurona recibe un impulso eléctrico a través de las dendritas y que llega al cuerpo celular. Allí, se procesa el impulso y se genera una nueva señal. Finalmente, la nueva señal eléctrica viaja por el axón y se transmite a otra neurona a través de las terminaciones nerviosas del mismo.

#### 2.1.2. Modelo de McCulloch y Pitts

En 1943 nace el modelo neuronal de McCulloch y Pitts, una unidad de cálculo que intenta emular el comportamiento de una neurona biológica. La señal de entrada para este modelo será un vector  $X = (x_1, x_2, \dots, x_n)$ . La información de entrada, representada por los datos del vector  $X$ , se transformará por un vector  $w$  denominado "de pesos sinápticos", cuyos valores amplificarán o atenuarán los valores de entrada para crear una emisión de salida.

Así, mediante la multiplicación tenemos

$$\sum_{i=1}^n x_i w_i.$$

A este resultado se le resta una constante  $\theta$  llamada umbral, que sirve para limitar la amplitud de la salida. De esta forma se tiene la suma ponderada

$$\sum_{i=1}^n x_i w_i - \theta.$$

A la expresión anterior o red le aplicamos una función  $s$  denominada función de activación. Las funciones que se pueden utilizar según el caso pueden ser

- Paso unitario:  $s(u) = \begin{cases} 1, & u \geq 0, \\ -1, & u < 0. \end{cases}$

- Lineal:  $s(u) = u.$

- Logística:  $s(u) = \frac{1}{1+e^{-u}}.$

- Tangente hiperbólica  $s(u) = \frac{e^{2u}-1}{e^{2u}+1}.$

En el caso del modelo de McCulloch y Pitts, esta función es

$$s(u) = \chi_{[0,\infty)}(u) = \begin{cases} 1, & u \geq 0, \\ 0, & u < 0. \end{cases}$$

Finalmente obtenemos el vector salida (*output*) que viene dado por la expresión

$$s\left(\sum_{i=1}^n x_i w_i - \theta\right).$$

El resultado puede ser o bien la variable respuesta o bien la entrada a otra neurona, componiendo así una red neuronal.

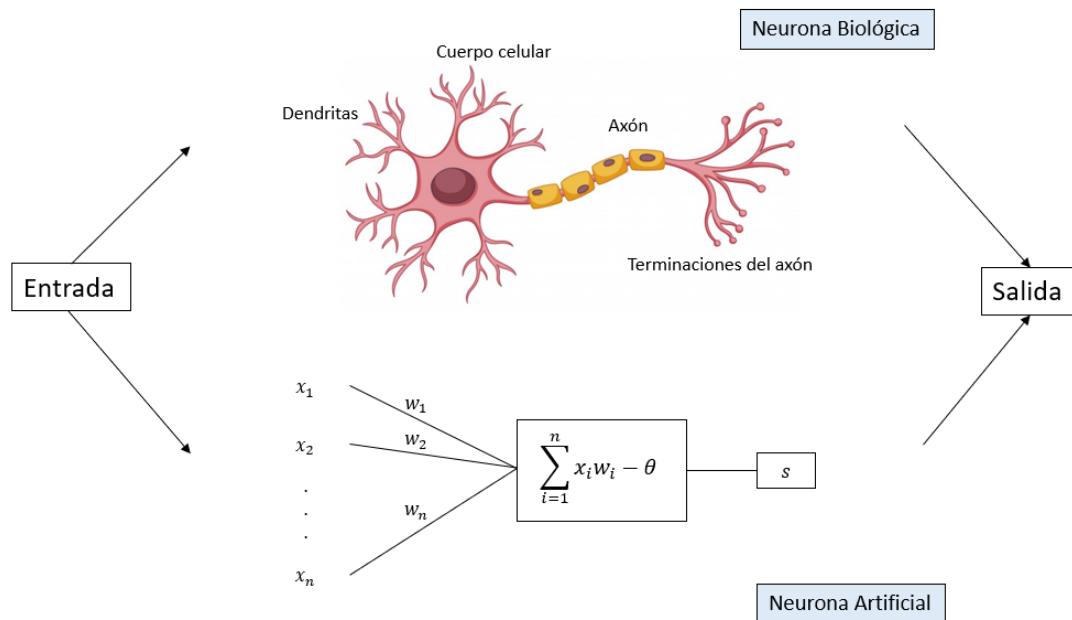


Figura 2.1: Comparación entre neurona biológica y artificial.

### 2.1.3. Perceptrón simple

En 1949, Donald Hebb inspiró el concepto de aprendizaje, clave en la representación neuronal. Hebb observó que cuando una neurona tiende a conectarse con otra, el peso de dicha conexión aumenta. Dicha propiedad se puede resumir de la siguiente forma: "neuronas que se activan juntas, permanecen juntas". La propuesta de Hebb fue fundamental en los algoritmos de aprendizaje que se desarrollaron posteriormente.

Posteriormente, en 1958, Frank Rosenblatt desarrolló un nuevo modelo de neurona artificial uniendo el enfoque de McCulloch y Pitts y las teorías de Hebb. En concreto, Rosenblatt implementó el aprendizaje de Hebb en el modelo de McCulloch y Pitts, de modo que los pesos de las neuronas que van estableciendo conexiones cambian para adaptarse a las relaciones entre ellas.

En el siguiente diagrama se representa un perceptrón simple. El vector de entrada  $X$  contiene  $k$  datos y la salida es un número real  $y$ . El vector de pesos  $W$  contiene los valores asociados a cada una de las conexiones entre las neuronas del vector de entrada con la salida.

Veamos a continuación como puede modelizarse matemáticamente el concepto de aprendizaje. Para ello, utilizaremos la siguiente notación

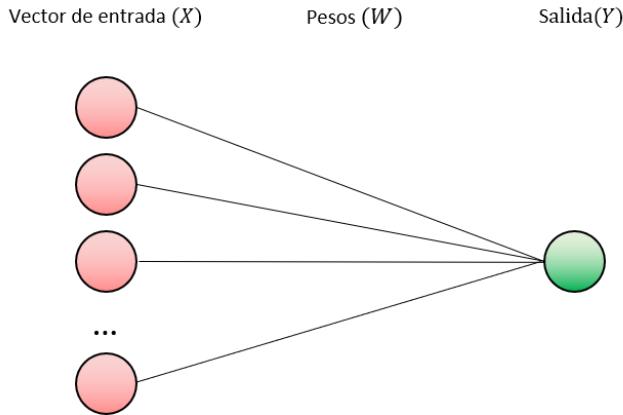


Figura 2.2: Diagrama del perceptrón simple.

- $X = (x_i)$  con  $i = 1, \dots, k$ : vector de entrada
- $y$ : valor de salida
- $W = (w_i)$ : peso asociado a la conexión entre  $x_i$  e  $y$
- $d$ : valor de salida esperado
- $\eta^{(n)}$ : tasa de aprendizaje en la iteración  $n$  que cumple  $0 < \eta \leq 1$
- $X^{(n)}$ : vector de entrada en la iteración  $n$
- $W^{(n)}$ : vector de pesos en la iteración  $n$
- $y^{(n)}$ : valor de salida en la iteración  $n$

La salida se obtienen mediante la fórmula

$$y^{(n)} = f\left(\sum_{i=1}^k w_i^{(n)} x_i^{(n)} - \theta^{(n)}\right).$$

donde para cada iteración  $X^{(n)}$  representa un vector aleatorio. El conjunto de vectores  $\{X^{(1)}, \dots, X^{(k)}\}$  se denomina patrón de entrenamiento.

Así, el aprendizaje se realiza modificando los pesos sinápticos de forma que se minimice cierta función error. En este caso se considera

$$E^{(n)} = \frac{1}{2}(d - y^{(n)})^2.$$

Para ello se utiliza el método del descenso de gradiente, en el cual al valor del peso en la iteración  $n$  se le resta un término de actualización para movernos en sentido contrario al del vector gradiente para que el error disminuya. El factor  $\eta$  o tasa de aprendizaje permite ajustar la magnitud de los pasos en cada iteración. Si  $\eta^{(n)}$  es demasiado grande se puede producir un efecto rebote que dificulta la localización del mínimo. Asimismo  $\eta^{(n)}$  se puede disminuir cuando estamos cerca de dicho mínimo para afinar el resultado. Por el contrario si al principio  $\eta^{(n)}$  es muy pequeño, el aprendizaje es más lento pero más estable.

$$w_i^{(n+1)} = w_i^{(n)} - \eta^{(n)} \frac{\partial E^{(n)}}{\partial w_i^{(n)}}.$$

Utilizando la regla de la cadena

$$\frac{\partial E^{(n)}}{\partial w_i^{(n)}} = -(d - y^{(n)}) \frac{\partial y^{(n)}}{\partial w_i^{(n)}} = -(d - y^{(n)}) x_i^{(n)} f'(\sum_{i=1}^k w_i^{(n)} x_i^{(n)} - \theta^{(n)}).$$

Y así finalmente si por ejemplo consideramos  $f(t) = t$  tenemos  $f'(t) = 1$  y

$$w_i^{(n+1)} = w_i^{(n)} + \eta^{(n)} (d - y^{(n)}) x_i^{(n)}.$$

Notar que si  $y = d$  las actualizaciones no se darían. En tal caso, tendríamos que el valor de salida es igual al valor de salida esperado, por lo que el perceptrón ya habría realizado el aprendizaje.

En el año 1962, Novikoff probó que este algoritmo es convergente tras un número de iteraciones finitas si los datos son linealmente separables y el número de errores está acotado por  $(\frac{R}{\gamma})^2$  con  $\|x\| < R$  siendo  $\gamma$  el margen de separación.

Así, en esta sección hemos visto cómo surgieron las neuronas artificiales y los primeros modelos que se plantearon, pero para emular la interacción entre ellas, necesitamos conocer cómo son las estructuras en las que se organizan. Estas estructuras se conocen como redes neuronales.

## 2.2. Redes neuronales artificiales (ANN)

Las redes neuronales artificiales son sistemas en los que se procesa una determinada información. Están formadas por elementos más simples denominados neuronas artificiales que están conectados entre sí. A cada una de estas conexiones se le asocia un valor al que se denomina peso sináptico. Este valor nos da información acerca de la cantidad de información que aporta dicho elemento en la conexión con el siguiente.

Estos elementos están organizados en tres tipos de capas:

- Capa de entrada: está compuesta por las neuronas a las que se les introduce la información inicial para comenzar el proceso.
- Capa oculta: en una red neuronal puede haber una o varias de ellas. Las neuronas de esta capa reciben y transmiten información, pero no se obtiene ningún resultado directo procedente de ellas.
- Capa de salida: en esta capa se encuentran las neuronas de las cuales extraemos la información al final del proceso.

### 2.2.1. Perceptrón multicapa

En 1969, Minsky y Papert observaron que el modelo del perceptrón simple no permitía resolver problemas no lineales. Para resolverlos, era necesaria la combinación de varios de ellos, pero hasta el momento no se había desarrollado un mecanismo que permitiese la adaptación de los pesos sinápticos de modo que los errores se propagasen hacia las capas ocultas. Así, desarrollaron una red neuronal conocida como perceptrón multicapa.

Está formado por una capa de entrada, una capa de salida y una o varias capas ocultas que pueden estar total o localmente conectadas de la forma que se puede ver en la figura. Como podemos observar, cada una de las salidas de las neuronas de la capa  $i$  es la entrada de las neuronas de la capa  $i + 1$ , es decir, las conexiones son siempre hacia delante.

En el entrenamiento de este tipo de redes, el algoritmo más utilizado es la propagación hacia atrás o *back-propagation*. Para explicar el funcionamiento del algoritmo vamos a considerar que tenemos una única capa oculta, pero si tuviésemos más se seguiría de forma análoga. Podemos considerar que este algoritmo se aplica en dos fases explicadas a continuación.

Como podemos ver en el siguiente diagrama, la capa de entrada contiene  $p$  neuronas a las que se les introducirán los datos de un vector de entrada  $X$  y la capa de salida contiene  $r$  neuronas mediante las cuales obtendremos el vector de salida  $Y$ . Las matrices de pesos  $W, V, \dots$  contienen los valores asociados a cada una de las conexiones entre las neuronas de las capas sucesivas.

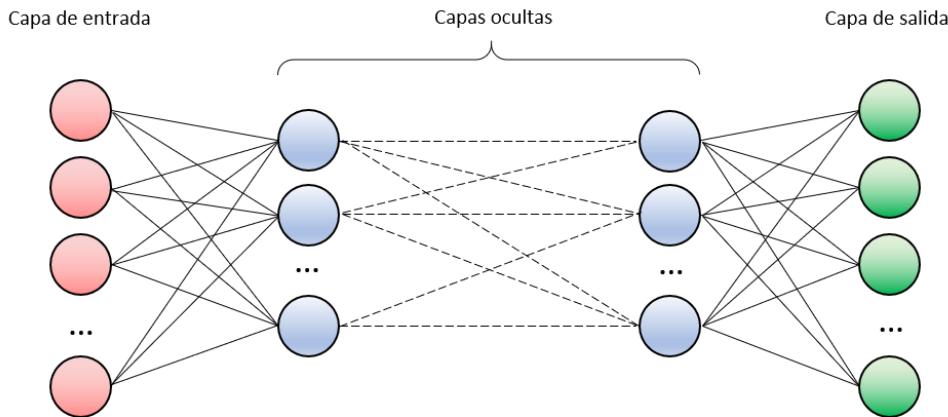


Figura 2.3: Diagrama del perceptrón multicapa con dos capas ocultas.

- Funcionamiento:** en esta primera fase, tomamos un conjunto de patrones iniciales y los procesamos a través de la red. El conjunto de entradas o grupo de entrenamiento  $\{X^{(1)}, \dots, X^{(N)}\}$  consta de  $N$  vectores con  $p$  componentes cada uno y mediante el uso de la red, obtendremos un conjunto de salida  $\{Y^{(1)}, \dots, Y^{(N)}\}$  de  $N$  vectores con  $r$  componentes. Introduciremos una sola capa oculta y las funciones de activación correspondientes a la interacción entre la capa de entrada y la capa oculta, y la capa oculta y la capa de salida serán  $f_1$  y  $f_2$  respectivamente. Estas funciones son diferenciables y no decrecientes. Usaremos la siguiente notación para cada iteración  $n$  con  $1 \leq n \leq N$

- $X^{(n)} = (x_i^{(n)})_{i=1,\dots,p}$ : vector de entrada
- $A^{(n)} = (a_j^{(n)})_{j=1,\dots,q}$ : vector con los datos de la capa oculta
- $Y^{(n)} = (y_k^{(n)})_{k=1,\dots,r}$ : vector de salida
- $W^{(n)} = (w_{ji}^{(n)})$ : matriz de pesos que almacena los valores correspondientes a las conexiones establecidas entre la capa de entrada y la capa oculta
- $V^{(n)} = (v_{kj}^{(n)})$ : matriz de pesos que almacena los valores correspondientes a las conexiones establecidas entre la capa oculta y la capa de salida
- $\theta_{1j}^{(n)}$ : umbral correspondiente a la función de activación  $f_1$
- $\theta_{2k}^{(n)}$ : umbral correspondiente a la función de activación  $f_2$

Así, en la iteración  $n$ , el valor de entrada para la neurona  $j$  de la capa oculta es la salida obtenida a partir de los datos de la capa de entrada, es decir

$$a_j^{(n)} = f_1\left(\sum_{i=1}^p x_i^{(n)} w_{ji}^{(n)} - \theta_{1j}^{(n)}\right).$$

Del mismo modo, en la iteración  $n$ , la salida en la neurona  $k$  se obtiene a partir de los datos de la capa oculta, es decir

$$y_k^{(n)} = f_2\left(\sum_{j=1}^q a_j^{(n)} v_{kj}^{(n)} - \theta_{2k}^{(n)}\right).$$

- Aprendizaje:** dado un conjunto de patrones al que llamaremos grupo de entrenamiento, en esta etapa se van modificando los pesos de las conexiones de modo que la diferencia entre la salida obtenida y la salida que buscamos sea mínima.

Sea  $D^{(n)} = (d_k^{(n)})$  el vector que contiene los datos de salida que buscamos, la función a minimizar es

$$E = \sum_{n=1}^N E^{(n)},$$

donde

$$E^{(n)} = \frac{1}{2} \sum_{k=1}^r (d_k^{(n)} - y_k^{(n)})^2.$$

Definamos  $b_k^{(n)}$  y  $c_k^{(n)}$  de la siguiente forma para simplificar los cálculos posteriores

$$b_k^{(n)} = \sum_{j=1}^q a_j^{(n)} v_{kj}^{(n)} - \theta_{2k}^{(n)}, \quad c_k^{(n)} = \sum_{i=1}^p w_{ji}^{(n)} x_i^{(n)} - \theta_{1j}^{(n)},$$

de forma que

$$y_k^{(n)} = f_2(b_k^{(n)}), \quad a_j^{(n)} = f_1(c_k^{(n)}).$$

De este modo, siendo  $\eta^{(n)}$  la tasa de aprendizaje en la iteración  $n$ , la fórmula de actualización de los pesos de la matriz  $V^{(n)}$  es

$$v_{kj}^{(n+1)} = v_{kj}^{(n)} - \eta^{(n)} \frac{\partial E^{(n)}}{\partial v_{kj}^{(n)}},$$

y aplicando la regla de la cadena

$$v_{kj}^{(n+1)} = v_{kj}^{(n)} + \eta^{(n)} \sum_{k=1}^r (d_k^{(n)} - y_k^{(n)}) f'_2(b_k^{(n)}) a_j^{(n)}.$$

Análogamente, la fórmula de actualización de los pesos de la matriz  $W^{(n)}$  es

$$w_{ji}^{(n+1)} = w_{ji}^{(n)} - \eta^{(n)} \frac{\partial E^{(n)}}{\partial w_{ji}^{(n)}},$$

y aplicando la regla de la cadena

$$\frac{\partial E^{(n)}}{\partial w_{ji}^{(n)}} = \sum_{k=1}^r (d_k^{(n)} - y_k^{(n)}) \left( -\frac{\partial y_k^{(n)}}{\partial w_{ji}^{(n)}} \right)$$

Como

$$\frac{\partial y_k^{(n)}}{\partial w_{ji}^{(n)}} = \frac{\partial y_k^{(n)}}{\partial a_j^{(n)}} \frac{\partial a_j^{(n)}}{\partial w_{ji}^{(n)}} = f'_2(b_k^{(n)}) v_{kj}^{(n)} f'_1(c_k^{(n)}) x_i^{(n)}$$

tenemos que

$$\frac{\partial E^{(n)}}{\partial w_{ji}^{(n)}} = - \sum_{k=1}^r (d_k^{(n)} - y_k^{(n)}) f'_2(b_k^{(n)}) v_{kj}^{(n)} f'_1(c_k^{(n)}) x_i^{(n)},$$

y finalmente

$$w_{ji}^{(n+1)} = w_{ji}^{(n)} + \eta^{(n)} \sum_{k=1}^r (d_k^{(n)} - y_k^{(n)}) f'_2(b_k^{(n)}) v_{kj}^{(n)} f'_1(c_k^{(n)}) x_i^{(n)}.$$

Este algoritmo se denomina de propagación hacia atrás porque se comienza modificando los pesos de las conexiones de las neuronas de la capa de salida, y, en función de estos, se van modificando los pesos de las capas anteriores.

La actualización de pesos se puede realizar o bien tras haber utilizado todos los patrones de entrenamiento (aprendizaje por lotes) o bien tras utilizar cada uno de los patrones (aprendizaje en serie). En este caso la hemos hecho utilizando el método de aprendizaje en serie.

Este tipo de red neuronal es uno de los más utilizados en la actualidad. Algunas de sus aplicaciones son la compresión de datos, el reconocimiento de caras o la segmentación de imágenes.

### 2.2.2. Redes neuronales recurrentes (RNN)

Una característica de los seres humanos que no hemos tenido en cuenta en los modelos explicados anteriormente es la memoria. Así una variable a tener en cuenta en el proceso del tratamiento de la información es el tiempo. Con esa premisa nacen en la década de 1980 las redes neuronales recurrentes. Estas redes nos permiten analizar secuencias que ocurren en un periodo de tiempo y predecir cual será el siguiente valor. Pese a que este modelo fue creado en la década de 1980, no se popularizó su uso hasta que los avances tecnológicos permitieron entrenarnas sin que el coste computacional fuese demasiado elevado.

Para introducir la variable temporal en los modelos de redes neuronales utilizamos una conexión entre la salida de una neurona y la entrada de la misma. Veamos qué sucede a lo largo del tiempo en el siguiente diagrama.

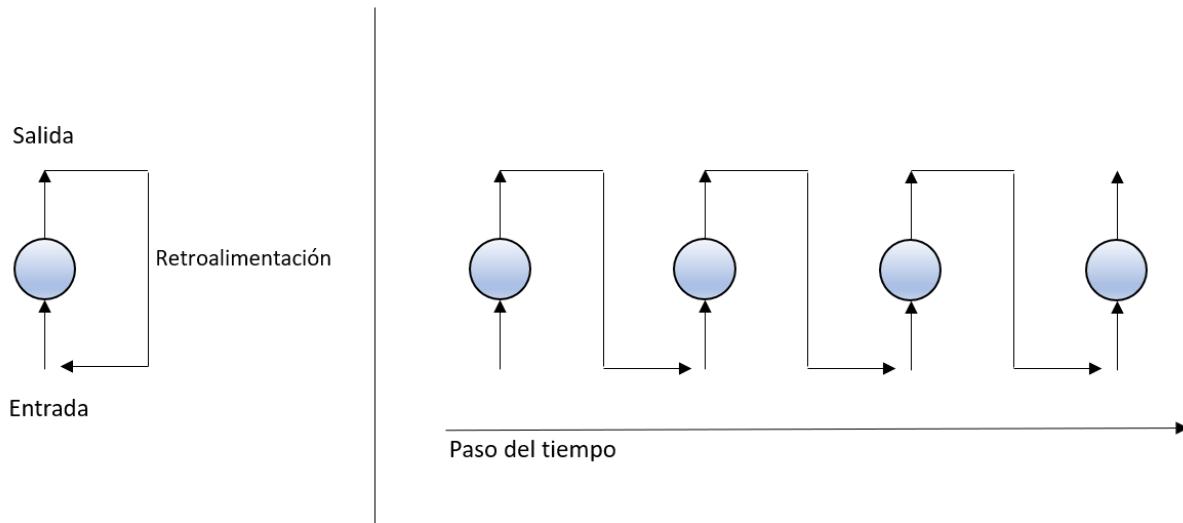


Figura 2.4: Diagrama de una neurona recurrente, a la izquierda en estático y a la derecha desarrollada en el tiempo.

Se dice que las RNN tienen memoria ya que su entrada en un determinado instante de tiempo  $t_1$  es la salida de la misma neurona en un instante  $t_0$ , por tanto, pueden recordar información de las entradas en los instantes anteriores.

Este tipo de redes han sido muy utilizadas recientemente dado que, al introducir la variable temporal, se pueden utilizar para resolver problemas como el reconocimiento de voz, el modelado de lenguaje o la traducción.

Un problema de las redes neuronales recurrentes es la falta de memoria a largo plazo, es decir, la red no puede recordar información que fue introducida hace muchos instantes de tiempo. Debido a esto, nace uno de los tipos más importantes de redes neuronales recurrentes, las denominadas *Long Short-Term Memory* (LSTM).

Estas redes tienen una estructura que permite ampliar la memoria, es decir, recordar entradas durante un periodo largo de tiempo (aunque también sirven a corto plazo). Además, tienen la capacidad de añadir y eliminar información de su memoria, por lo que resultan muy eficientes debido a que solo almacenan la información más relevante. Este flujo de información se controla mediante tres puertas: de entrada, de olvido y de salida. Como veremos posteriormente, estas redes son muy útiles en *deep learning*.

Además de las LSTM, existe otro tipo de RNN denominado GRU (*gated recurrent units*) muy similar al anterior pero más simple, ya que en vez de utilizar tres puertas de control utiliza dos: de reset y de actualización. La ventaja de GRU respecto de LSTM es que tiene un mejor rendimiento cuando el conjunto de datos es más pequeño. Una de las aplicaciones de este sistema es la del modelado de música polifónica o modelado de señales de voz.

### 2.2.3. Redes neuronales convolucionales (CNN)

Las redes neuronales convolucionales son una variación del perceptrón multicapa explicado anteriormente, pero en este caso los datos de entrada son matrices bidimensionales. Las neuronas pertenecientes a estas redes son similares a las neuronas biológicas pertenecientes a la corteza visual, por lo que este algoritmo está diseñado para tareas de visión artificial, como la segmentación y clasificación de imágenes.

Su característica principal es que en lugar de utilizar la multiplicación común de matrices, utiliza, al menos en alguna de sus capas, la convolución. Así, estas redes están construidas por tres tipos de capas:

- Capa de convolución: esta capa recibe como entrada una matriz y aplica sobre ella uno o varios filtros o *kernels* para reducirla. Estos filtros son matrices de menor tamaño que la de entrada. Son simétricas y sus valores son reales.

La forma de reducir la matriz que se ha introducido es, recorrerla en su totalidad y, tomando submatrices del mismo tamaño que el filtro, convolucionarlas con él de modo que la matriz de salida tendrá tantos elementos como submatrices del tamaño del filtro tenga la matriz de entrada.

Como vemos en el siguiente diagrama, la convolución de matrices se realiza multiplicandolos los elementos de ambas componentes a componente y sumándolos.

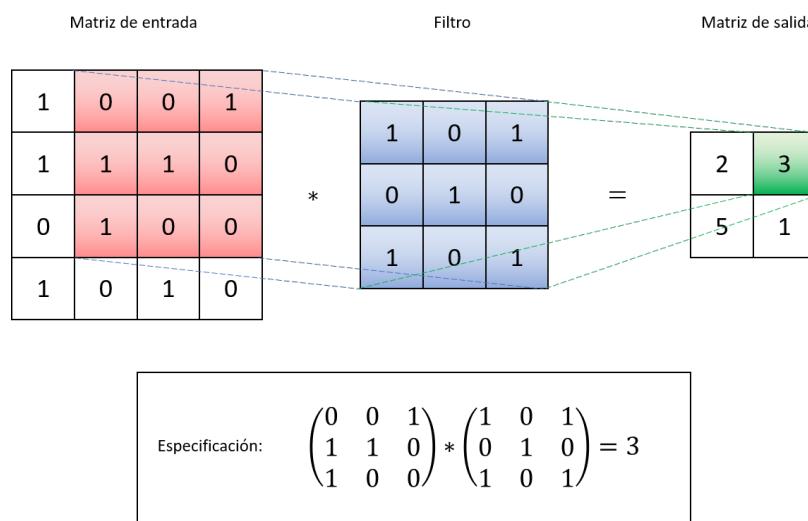


Figura 2.5: Funcionamiento de una capa de convolución.

Así, en el ejemplo de la figura, hemos reducido una matriz  $4 \times 4$  a una matriz  $2 \times 2$ .

- Capa de reducción o *pooling*: esta capa reduce la cantidad de parámetros de la matriz utilizando funciones como el promedio o el máximo en regiones de la misma.

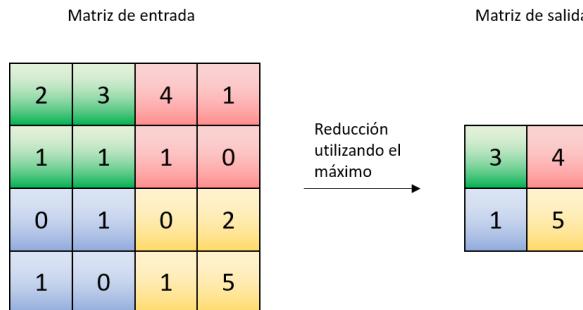


Figura 2.6: Funcionamiento de una capa de reducción.

- Capa clasificadora: esta capa está formada por neuronas y actúa del mismo modo que un perceptrón multicapa. Así, la penúltima de estas capas está totalmente conectada y su salida es un vector  $k$ -dimensional, donde  $k$  es el número de clasificaciones posibles, que contiene las probabilidades de que ocurra cada una de ellas.

Al final de esta capa se utiliza la denominada función *softmax*, una generalización de la función sigmoidea que normaliza el vector de salida.

Así, el proceso completo que realiza una red neuronal convolucional es el representado en el siguiente diagrama.

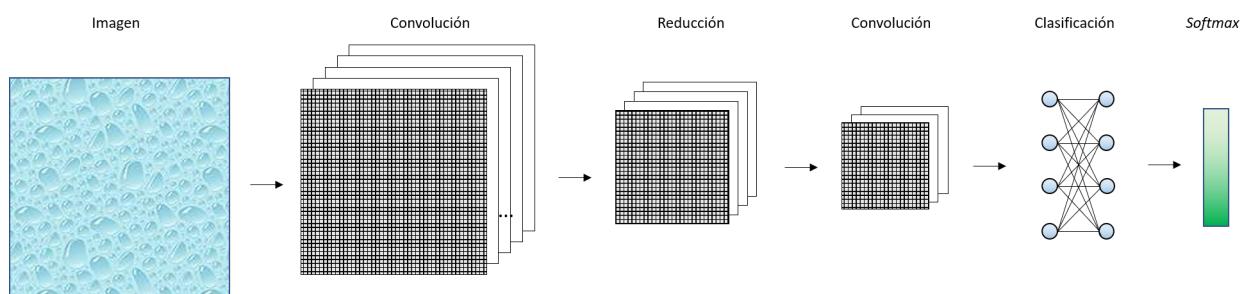


Figura 2.7: Diagrama de una red neuronal convolucional.

## 2.2.4. Redes neuronales generativas adversarias

Las redes neuronales generativas adversarias combinan dos redes neuronales, a las cuales denominaremos generador y discriminador. La red generadora creará muestras del elemento que busquemos (sonidos, textos, imágenes...) en base a un ruido aleatorio. La red discriminadora analiza las muestras de la red generadora y determinará la calidad de estas.

Dado un grupo de entrenamiento, consideramos que el sistema está bien entrenado cuando el discriminador no distingue si las salidas proceden de la red neuronal o del grupo de entrenamiento. El método que se utiliza para que cada vez las muestras se approximen más al resultado buscado es la propagación hacia atrás.

Este tipo de redes se han utilizado para resolver problemas tales como: producir muestras de elementos para escenarios de videojuegos, mejorar las imágenes con muy poca luz, construir modelos 3D a partir de imágenes 2D, mejorar imágenes astronómicas,...

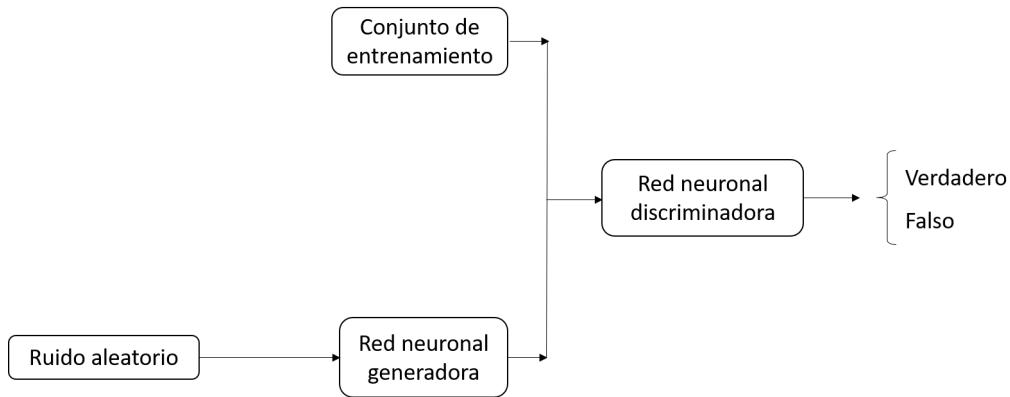


Figura 2.8: Diagrama de una red neuronal generativa adversaria.

### 2.3. Deep learning

Una vez introducidas las redes neuronales, veamos a qué se refiere el concepto de *deep learning*. Este aprendizaje es un tipo de aprendizaje automático, que a su vez es una parte de la inteligencia artificial.

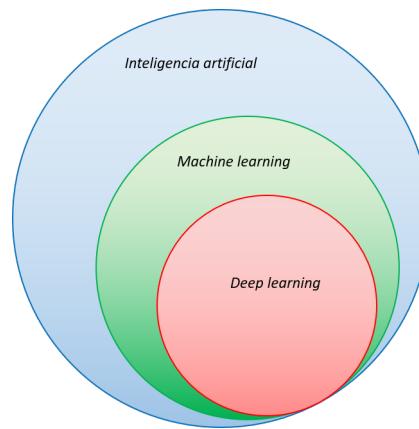


Figura 2.9: Diagrama de la inteligencia artificial.

Como se vio en el primer capítulo de este trabajo, la inteligencia artificial es la rama de la computación que estudia cómo programar una máquina para que su comportamiento se asemeje al de un humano. Asimismo, el aprendizaje automático es la parte de la inteligencia artificial que busca que las máquinas realicen un aprendizaje mediante algoritmos. De este modo pueden reconocer patrones en conjuntos de datos y hacer predicciones. Para ello se utilizan, entre otros algoritmos, las redes neuronales artificiales.

En los años 80, era muy difícil entrenar un algoritmo de machine learning por dos razones. Por una parte, el hardware existente requería de mucho tiempo de entrenamiento y, por otra, el volumen de datos ejemplo era muy escaso o difícilmente accesible. Por lo tanto, este campo fue prácticamente abandonado hasta que en 2010 surgió el hardware adecuado para agilizar la computación. Es entonces cuando el concepto de *deep learning* tomó más importancia ya que la nueva tecnología posibilitó añadir capas intermedias de neuronas, permitiendo realizar clasificaciones no lineales complejas.

Así, el *deep learning* puede definirse como un aprendizaje automático muy sofisticado que utiliza redes neuronales artificiales para intentar imitar el funcionamiento del cerebro humano. Se utilizan, en su mayoría, redes neuronales recurrentes, redes neuronales convolucionales y redes neuronales generativas adversarias. Estas redes trabajan con algoritmos de alto nivel y dos o más capas ocultas. En ese sentido puede considerarse que el aprendizaje que realizan es más profundo y de ahí su denominación.

Uno de los grandes potenciales del *deep learning*, es su capacidad de extraer atributos relevantes de los datos en bruto, a diferencia de los algoritmos “clásicos” de aprendizaje automático, en los que esa labor debe ser realizada manualmente, lo cual influye negativamente en el rendimiento del algoritmo.

Cabe resaltar que la tecnología de redes neuronales, para que sea efectiva, precisa de grandes volúmenes de información para ser entrenada (va muy ligada al concepto de *big data*). Además de estar orientada a resolver problemas complejos, tales como los relacionados con visión artificial, reconocimiento de voz, traducción automática de texto e imagen, diagnóstico de enfermedades, detección de fraudes financieros, automóviles con autonomía, etc.

No obstante, el *deep learning* no es la solución universal y, de hecho, muchas de las soluciones a problemas complejos son una combinación de ésta y algoritmos considerados propios del ámbito de *machine learning*.

# Capítulo 3

## Aplicación a la despoblación en Aragón.

### 3.1. BigML

BigML es una herramienta *on-line* que sirve para resolver problemas de aprendizaje automático. Esta plataforma permite introducir conjuntos de datos y aplicarles distintos modelos y algoritmos de modo que se puedan extraer conclusiones y hacer predicciones.

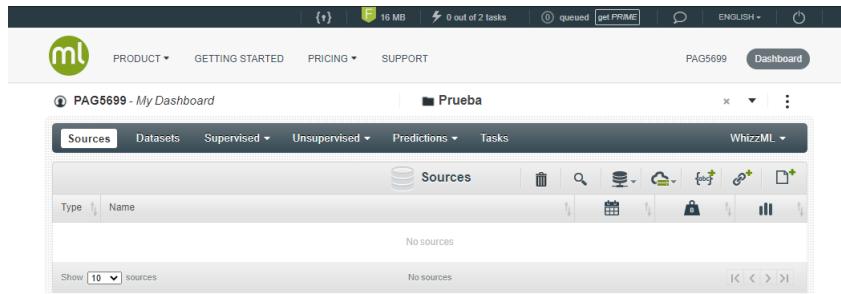


Figura 3.1: Vista inicial de la plataforma *BigML*.

A continuación vamos a explicar cómo funciona esta aplicación.

1. Creación de un proyecto: una vez tenemos preparada la tabla con los datos que queremos analizar, insertamos el documento en la aplicación arrastrándolo a la pantalla que aparece en la figura anterior.

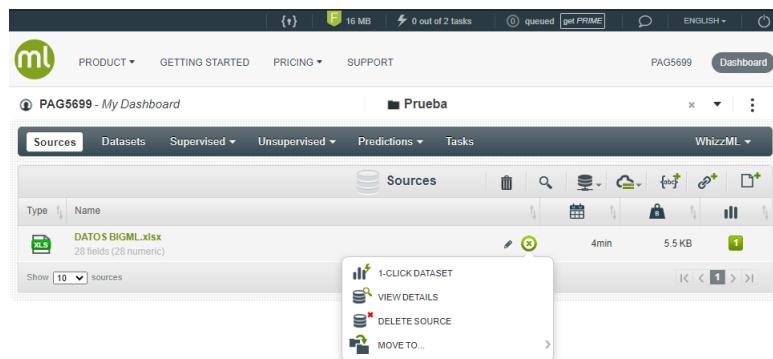


Figura 3.2: Representación del conjunto de datos.

Con el archivo ya insertado, nos aparecen los datos con las opciones que vemos en la figura. Para convertir los datos de la tabla en un grupo de entrenamiento, seleccionamos *1-CLICK DATASET*.

Otras opciones nos permiten ver los detalles de los datos, eliminarlos y moverlos a otro proyecto.

- Tratamiento de datos: el programa analiza los datos y los muestra en detalle según la configuración que se observa en la siguiente figura.

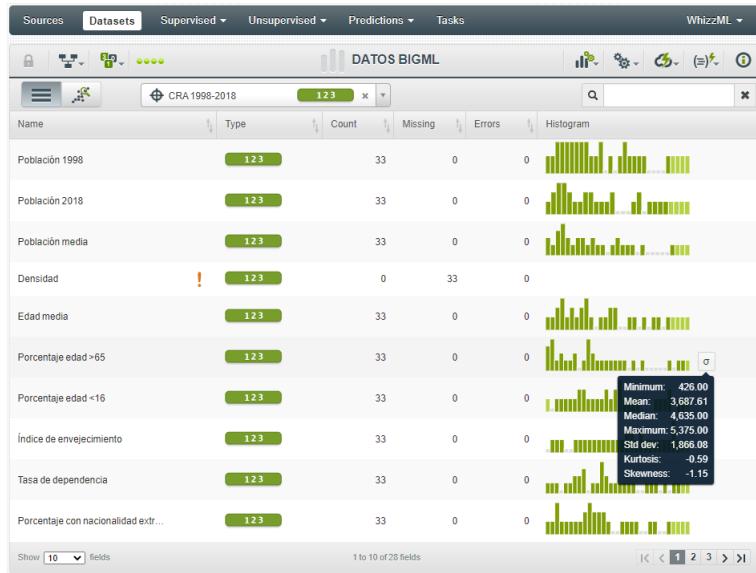


Figura 3.3: Opciones una vez hemos cargado los datos.

Para cada tipo de datos nos aparece la siguiente información: nombre, tipo, número de casillas, cantidad de datos que faltan, errores e histograma. Además, como podemos ver, el cursor está posicionado en el símbolo  $\sigma$  que nos aporta información acerca de las variables numéricas. Además, la variable que aparece encima con el símbolo  $\oplus$  se denomina variable objetivo, y es sobre la cual se hace el análisis.

- Elección del algoritmo y predicción.

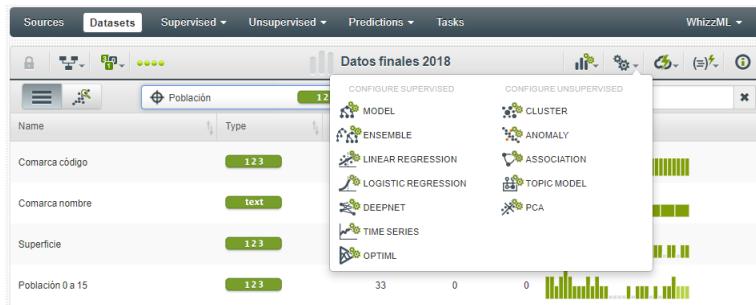


Figura 3.4: Opciones de algoritmia.

Así, tenemos un desplegable en el que podemos elegir entre algoritmos supervisados y no supervisados. En cada uno de ellos, también se pueden seleccionar distintos parámetros. En concreto en la siguiente sección el algoritmo que vamos a utilizar es *deepnet*. En él, podemos elegir qué porcentaje de los datos se utiliza como conjunto de entrenamiento, cuántas capas ocultas queremos que haya en el proceso, cual debe ser el tiempo máximo de entrenamiento, etc

Nuestro objetivo final es extraer conclusiones en base a los resultados obtenidos. Esta última parte del proceso se explicará con más detalle en las siguientes secciones, una vez hayamos introducido el contexto de nuestro caso particular.

### 3.2. Planteamiento del problema

La despoblación del medio rural en España, y, en particular en Aragón, es uno de los problemas demográficos que más afectan social y políticamente al país. En la siguiente tabla se muestra, por comarcas y provincias, la diferencia de población entre el año 2018 y el año 1998. Estos datos han sido extraídos de Aragón Open Data y el Instituto Aragonés de Estadística (IAEST) las páginas web que recogen las estadísticas elaboradas por el Gobierno de Aragón.

Comarca nombre	Provincia	Diferencia (D)	Tasa de variación (%)	Comarca nombre	Provincia	Diferencia (D)	Tasa de variación (%)
La Jacetania	Huesca	1032	6.22	Ribera Baja del Ebro	Zaragoza	-1200	-12.42
Alto Gállego	Huesca	1373	11.35	Bajo Aragón-Caspe / Baix Aragó-Casp	Zaragoza	1561	12.06
Sobrarbe	Huesca	642	9.53	Comunidad de Calatayud	Zaragoza	-2621	-6.58
La Ribagorza	Huesca	87	0.73	Campo de Cariñena	Zaragoza	-42	-0.42
Cinco Villas	Zaragoza	-1268	-4.00	Campo de Belchite	Zaragoza	-1111	-19.56
Hoya de Huesca / Plana de Uesca	Huesca	7974	13.41	Bajo Martín	Teruel	-1230	-16.16
Somontano de Barbastro	Huesca	1192	5.32	Campo de Daroca	Zaragoza	-1224	-18.05
Cinca Medio	Huesca	1943	8.95	Jiloca	Teruel	-1812	-12.87
La Litera / La Llitera	Huesca	-415	-2.23	Cuencas Mineras	Teruel	-1814	-18.36
Los Monegros	Huesca	-2934	-13.62	Andorra-Sierra de Arcos	Teruel	-1470	-12.71
Bajo Cinca / Baix Cinca	Huesca	2124	9.52	Bajo Aragón	Teruel	2625	10.11
Tarazona y el Moncayo	Zaragoza	-355	-2.51	Comunidad de Teruel	Teruel	3690	8.73
Campo de Borja	Zaragoza	-353	-2.49	Maestrazgo	Teruel	-521	-14.02
Aranda	Zaragoza	-1458	-17.94	Sierra de Albarracín	Teruel	-685	-13.41
Ribera Alta del Ebro	Zaragoza	5330	24.51	Gúdar-Javalambre	Teruel	-412	-5.30
Valdejalón	Zaragoza	7143	33.23	Matarraña / Matarranya	Teruel	-648	-7.31
Central	Zaragoza	106736	16.67				

Figura 3.5: Tabla de diferencia de población.

Como podemos ver, en las tres comarcas que contienen las capitales de provincia la población ha aumentado. No obstante, este crecimiento es asimétrico, puesto que el mayor aumento se ha producido en la comarca central (16,67 %), que es donde se encuentra la ciudad de Zaragoza. La Hoya de Huesca (13,41 %) y la Comunidad de Teruel (8,73 %) también presentan ciertas diferencias en la tasa de variación

$$T = \frac{D \cdot 100}{P_0}.$$

No obstante, el descenso de población sigue siendo preocupante sobretodo en algunas comarcas alejadas. Destacan los casos extremos de Campo de Belchite (-19,56 %), Campo de Daroca (-18,05 %) y Aranda (-17,94 %) en la provincia de Zaragoza, los de Cuencas Mineras (-18,36 %), Bajo Martín (-16,16 %), Maestrazgo (-14,02 %), Sierra de Albarracín (-13,41 %), Andorra-Sierra de Arcos (-12,71 %) y Jiloca (-12,87 %) en la provincia de Teruel, y Los Monegros (-13,62 %) en la provincia de Huesca.

Estamos interesados en analizar qué variables (demográficas, económicas, sociales, etc) influyen en la variación de población, así como cuales de ellas son más influyentes. De este modo, extraeremos algunas conclusiones acerca de por qué se puede producir esta despoblación.

### 3.3. Búsqueda y tratamiento de datos

Los datos aportados en este trabajo proceden del Instituto Aragonés de Estadística (IAEST), página perteneciente al gobierno de Aragón y de la página web del Instituto Nacional de Estadística (INE). La mayor parte de los datos se pueden exportar en diversos formatos. En nuestro caso, han sido descargados en .csv (*Comma Separated Values*) y posteriormente transformados a formato .xls (*Microsoft Excel*) con el objeto de poder prepararlos antes de introducirlos en la aplicación *BigML*.

Aunque la recopilación de datos puede hacerse de forma más exhaustiva, analizando la variación de población en cada municipio, esto excede la finalidad de este trabajo. Por este motivo, los datos se han recopilado por comarcas, pues aunque el estudio sea algo más grueso, también resulta interesante observar la despoblación en las diferentes demarcaciones políticas.

Obviamente en este estudio no se detectará la asimetría en el crecimiento o decrecimiento de la población, es decir, la diferencia entre la evolución de la población en los medios rural y urbano. Por ejemplo, en caso de las comarcas que contienen a las capitales de provincia, pese a que la diferencia de población sea positiva, esto no implica que no se produzca despoblación, dado que es posible que crezca en el medio urbano y descienda en el medio rural.

Así, se ha decidido considerar dos tipos de variable: las variables demográficas y las variables socioeconómicas. Las demográficas tienen finalidad analítica y están construidas con datos directamente relacionados con los movimientos de población, por lo que es claro que influyen sobre la variación de la misma. Por otra parte, las socioeconómicas modulan el comportamiento de las dinámicas poblacionales por lo que se introducen a modo explicativo seleccionando aquellas que pueden ayudar a definir e interpretar las causas del proceso de despoblación. En la siguiente tabla se muestra la lista completa de variables seleccionadas.

	Año	Tipo	Unidades
<b>Variables identificativas</b>			
Comarca código		Entero	
Comarca nombre		Categórico	
<b>VARIABLES DEMOGRÁFICAS</b>			
Población 1998	1998	Directa	Personas
Población 2018	2018	Directa	Personas
Población media	1998-2018	Indirecta	Personas
Densidad	1998-2018	Indirecta	Personas/km <sup>2</sup>
Edad media	1998-2018	Directa	Años
Porcentaje edad >65	2018	Directa	%
Porcentaje edad <16	2018	Directa	%
Índice de envejecimiento	2018	Directa	%
Tasa de dependencia	2018	Directa	%
Porcentaje con nacionalidad extranjera	2018	Indirecta	%
Tasa bruta de natalidad	2018	Indirecta	Personas por cada mil hbts
Tasa bruta de mortalidad	1998-2018	Indirecta	Personas por cada mil hbts
Tasa de crecimiento natural	1998-2018	Indirecta	Personas por cada mil hbts
<b>VARIABLES SOCIOECONÓMICAS</b>			
Renta per cápita	2018	Directa	Euros
Tasa atención sanitaria	2018	Indirecta	Personas por cada mil hbts
Farmacias	2018	Indirecta	Personas por cada mil hbts
Porcentaje de parados	2018	Indirecta	%
Tasa de actividad	2018	Indirecta	%
Viviendas nuevas	2018	Indirecta	Personas por cada mil hbts
Centros escolares	2018	Indirecta	Personas por cada mil hbts
Tasa Agricultura, ganadería y pesca	2018	Indirecta	%
Tasa Construcción	2018	Indirecta	%
Tasa Industria y energía	2018	Indirecta	%
Tasa Servicios	2018	Indirecta	%
Bibliotecas	2018	Indirecta	Personas por cada mil hbts
Residencias	2018	Indirecta	Personas por cada mil hbts
<b>Variable objetivo</b>			
CRA 1998-2018	1998-2018	Indirecta	%

Figura 3.6: Tabla de variables.

Como podemos ver, en la tabla se especifica para cada variable el año o rango de años al que corresponde el dato, el tipo y las unidades en que viene expresada.

Algunas de las variables se recogen en un periodo de tiempo y otras en el último año debido a la dificultad de conseguir datos suficientes en un intervalo homogéneo. Para las variables poblacionales en las que ha sido posible, se han recogido datos relativos al periodo de 20 años comprendido entre 1998 y 2018. Los datos de otras variables aparecen en el padrón municipal de los años citados, pero el hecho de haber agrupado el territorio por comarcas ha dificultado su extracción, por lo que se ha decidido considerar los datos relativos al último año. No obstante, algunos de ellos no varían sustancialmente de año en año, por lo que este hecho no modificará el resultado de forma significativa.

Las variables son de tipo directo e indirecto. Las de tipo directo han sido recogidas directamente de las fuentes antes citadas, mientras que las de tipo indirecto han sido construidas a partir de las anteriores porque son más adecuadas para nuestro interés. Fundamentalmente, se ha trabajado con tasas de variación, especialmente en las variables de tipo socioeconómico.

La variable escogida como campo objetivo es el Crecimiento Real Anual (CRA)

$$CRA = \left( \sqrt[t]{\frac{P_f}{P_i}} - 1 \right) \cdot 100$$

donde  $P_f$  es la población final (población en 2018),  $P_i$  la población inicial (población en 1998) y  $t$  el periodo de tiempo considerado (20 años).

Esta expresión se tiene considerando la evolución de la población final como  $P_f = \delta^t P_i$ , donde  $\delta$  es la tasa de crecimiento anual definida como  $\delta = 1 + \frac{CRA}{100}$ . Despejando CRA se obtiene la expresión.

Este indicador es el recomendado por Naciones Unidas (ONU) para estudiar la evolución de la población en un determinado periodo de tiempo, debido a que en él se recogen las dinámicas poblacionales naturales (los nacimientos y las defunciones), y las espaciales (las emigraciones y las inmigraciones).

A continuación se muestra el gráfico del CRA de las 33 comarcas agrupadas por provincias.

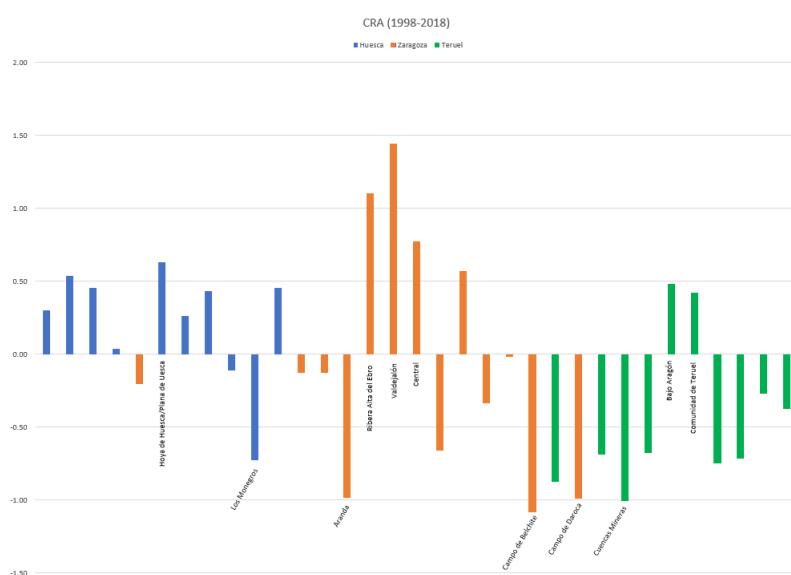


Figura 3.7: Gráfico del CRA agrupado por provincias.

Para interpretar los datos del gráfico anterior, se tienen en cuenta los siguientes indicadores:

- $CRA < -2\%$ : decrecimiento fuerte
- $-2\% < CRA < -1\%$ : decrecimiento moderado
- $-1\% < CRA < -0,1\%$ : decrecimiento débil

- $-0,1\% < CRA < 0,1\%$ : estancamiento
- $0,1\% < CRA < 1\%$ : crecimiento débil
- $1\% < CRA < 2\%$ : crecimiento moderado
- $2\% < CRA$ : crecimiento fuerte

Como podemos ver en el gráfico anterior, el CRA no alcanza valores extremos en ninguna de las comarcas. Esto puede deberse a la amplitud del periodo de tiempo considerado y la división territorial escogida, ya que en una misma comarca, municipios de alto crecimiento contrarrestan los valores negativos de los municipios más afectados por la despoblación. Aun así, puede observarse claramente la situación poblacional en Aragón.

Destaca el caso de la provincia de Teruel, en la cual todos los valores de CRA son negativos excepto dos de ellos (Bajo Aragón y Comunidad de Teruel). En la provincia de Zaragoza, llaman la atención los casos de Campo de Daroca, Aranda y Campo de Belchite, alcanzando este último el valor más bajo de toda la comunidad. Por otra parte, se observa claramente el crecimiento de la comarca Central y aledañas, debido a la influencia de la capital. El crecimiento en la provincia de Huesca ha sido estable en todas las comarcas, salvo en la de Los Monegros.

Para analizar la influencia de las variables escogidas en el CRA, se ha utilizado el programa *BigML*, y, en particular, el algoritmo de redes neuronales profundas. El programa escoge de forma automática los parámetros (tiempo de entrenamiento, porcentaje del conjunto de datos utilizado para el entrenamiento, etc) más óptimos según el conjunto de datos introducido. Aunque las redes neuronales suelen usarse en problemas con un mayor número de datos, hemos elegido este algoritmo porque tiene en cuenta las posibles dependencias entre variables, de forma que este hecho no afecte a los resultados.

Además, se ha utilizado el programa *Microsoft Excel* para calcular el coeficiente de correlación lineal entre la variable objetivo y cada una de las variables a estudiar. Esto no es estrictamente necesario, pero nos ha permitido hacer una aproximación previa orientativa.

### 3.4. Resultados y conclusiones

En las figuras 3.8 y 3.9 se recogen los resultados obtenidos, ordenados de mayor a menor, así como el histograma correspondiente a los pesos de las variables que más influyen en el campo objetivo.

	Coefficiente de Correlación	Red neuronal profunda
Porcentaje edad <16	0.87	15.89
Tasa de crecimiento natural	0.71	13.95
Tasa bruta de natalidad	0.67	10.06
Renta per cápita	0.59	8.07
Tasa de actividad	0.57	6.81
Porcentaje con nacionalidad extranjera	0.41	6.52
Densidad	0.36	5.57
Viviendas nuevas	0.35	5.43
Porcentaje de parados	0.35	3.78
Población 2018	0.29	3.24
Población media	0.29	3.21
Población 1998	0.28	3.14
Residencias	-0.17	2.17
Tasa Servicios	-0.21	1.67
Bibliotecas	-0.31	1.56
Tasa Construcción	-0.32	1.52
Tasa Agricultura, ganadería y pesca	-0.42	1.30
Tasa bruta de mortalidad	-0.56	0.96
Tasa atención sanitaria	-0.58	0.91
Centros escolares	-0.60	0.91
Farmacias por 1000	-0.61	0.86
Tasa de dependencia	-0.66	0.84
Tasa Industria y energía	-0.75	0.81
Índice de envejecimiento	-0.80	0.36
Porcentaje edad >65	-0.85	0.34
Edad media	-0.88	0.11

Figura 3.8: Tabla de resultados.

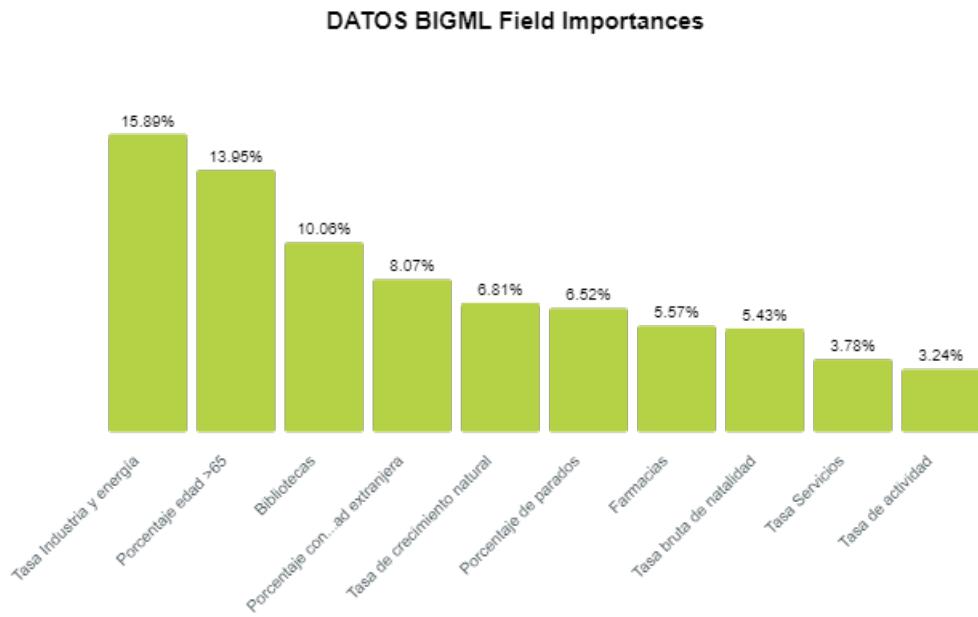


Figura 3.9: Histograma generado por *BigML*.

### 3.4.1. Conclusiones

Existen algunas variables que tienen una clara correlación con el CRA. Además de algunas de las demográficas, que explican el crecimiento natural, cabe destacar el porcentaje de personas menores de 16 años que es la que más influencia positiva tiene en nuestra variable de estudio. Por otro lado, y con influencia negativa, sobre la misma se hallan el índice de envejecimiento, la tasa de industria y energía -que mide el porcentaje de personas que trabajan en ese sector- y el porcentaje de mayores de 65 años lo cual refleja de forma clara el envejecimiento de la población aragonesa y su relación con la despoblación del territorio.

La tasa de industria y energía, aunque sea la variable que más influye en el CRA, tampoco es determinante en el mismo. Concluimos por tanto que, si bien algunos factores pesan más que otros, es una combinación adecuada de todos ellos la que podría funcionar para controlar la despoblación.

Estos resultados también se reflejan en el análisis realizado por el programa, pues las dos últimas variables son dos de las que más pesan en el CRA con porcentajes del 13,95% y el 10,06% respectivamente. Los porcentajes de bibliotecas, farmacias, tasa de atención sanitaria, residencias y centros escolares, si bien tienen una influencia moderada o baja por separado, podrían ser determinantes si los agrupáramos en una sola variable pues la suma de los pesos correspondientes es del 17.51%.

Otra función de la herramienta que hemos utilizado es la de generar predicciones. Así, se ha realizado una prueba aumentando las variables mencionadas anteriormente, para comprobar su efecto en la modificación del CRA. Como podemos ver, en la figura 3.10, cuando aumentamos estas variables, aumenta su valor. Aunque a priori pueda parecer que variables como el número de farmacias o de bibliotecas no afectan directamente a los movimientos de población, a la vista de los resultados obtenidos, parece lógico pensar que su importancia también se debe a que la población está muy envejecida.

Dado que la densidad en el medio rural aragonés es muy baja, y la edad media de la población muy alta (respecto a la media española), este tipo de servicios contribuye positivamente al asentamiento de la población. Este hecho, también puede influir en el peso tan negativo que se asocia a la tasa de industria y energía, ya que un aumento en este porcentaje, implica de forma indirecta una disminución en los porcentajes del resto de ocupaciones, que pueden ser más necesarias en base a las características de la población.

Finalmente, cabe mencionar otras variables que también tienen influencia en la variable objetivo. Estas son, el porcentaje de individuos con nacionalidad extranjera, la tasa de crecimiento natural y el porcentaje de parados. Todas estas variables son coherentes con los resultados que podemos esperar pensando de forma intuitiva.

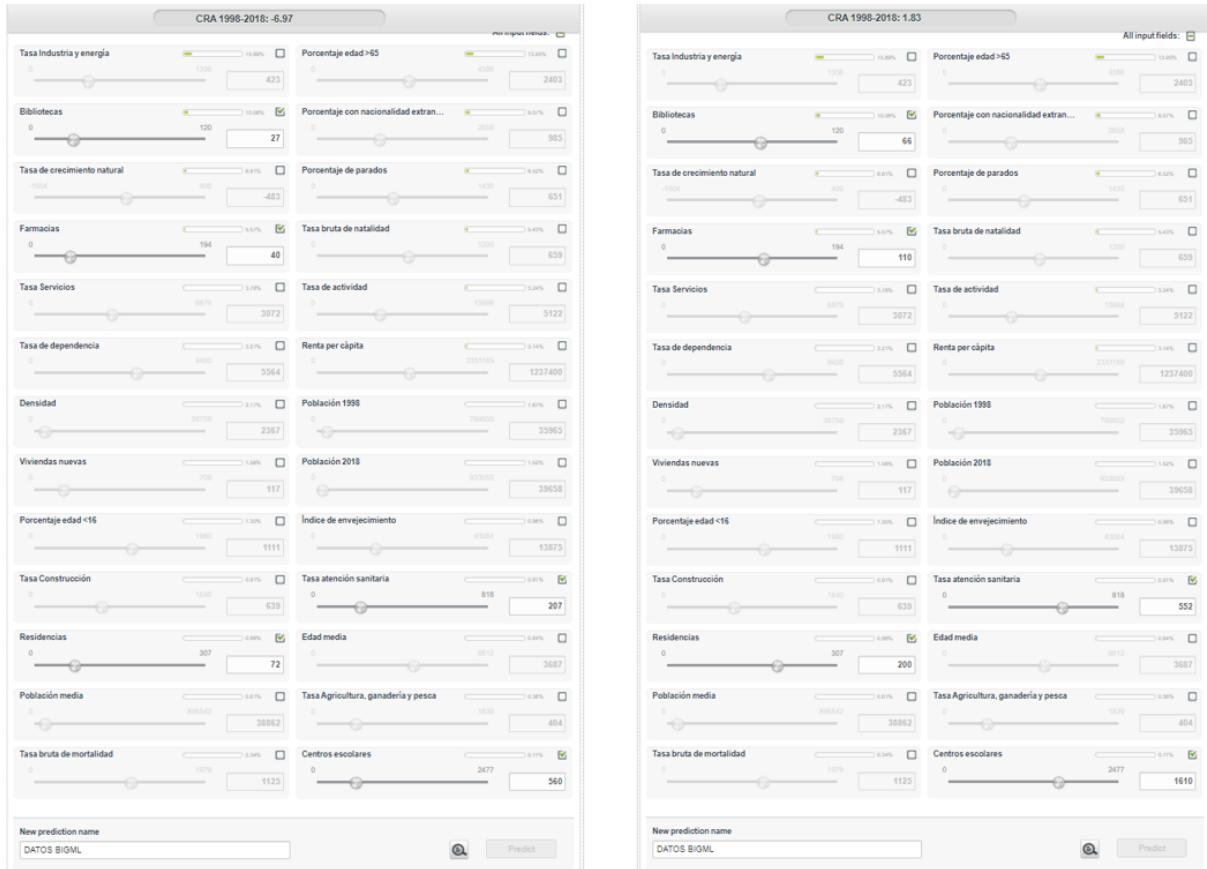


Figura 3.10: Predicciones realizadas con *BigML*.

### 3.4.2. Propuestas de mejora

No obstante, estos resultados no son determinantes ni un estudio exhaustivo, dadas las características y longitud de este trabajo, más bien podrían suponer un punto de partida para seleccionar variables más adecuadas.

Una de las posibles mejoras podría ser afinar el estudio bajando a nivel municipal. Esto nos permite observar de un mejor modo las diferencias entre los medios urbano y rural. Además, la recogida de datos es más sencilla debido a que se puede acudir directamente a los padrones municipales. Esto aumenta considerablemente el volumen de datos que se manejarían.

Otra forma de tratar el tema, sería utilizando más series temporales. Eso nos permitiría observar el comportamiento de la población en períodos más pequeños, ya que, al tomar un intervalo de 20 años, puede ser que se produzcan crecimientos y decrecimientos más puntuales, que en un rango mayor pueden compensarse entre sí.

Por otra parte, en nuestro estudio se han escogido 26 variables contando directas e indirectas. Este número podría ampliarse para realizar un análisis más profundo de la situación. Algunas propuestas de variables que se pueden introducir se recogen en la siguiente tabla.

Vehículos	Distancia a núcleos urbanos
Conexiones de internet	Distancia a capitales de provincia
Cobertura	Superficie agraria útil
Transporte público	Oficinas Bancarias
Redes ferroviarias	Valor catastral
Carreteras	Transacciones inmobiliarias
Nivel de estudios	Gasto

Figura 3.11: Predicciones realizadas con *BigML*.

El programa utilizado sería muy adecuado para manejar tal volumen de datos, dado que las en él se incluye el algoritmo *deepnet* que trabaja con redes profundas. Como se ha visto en el capítulo anterior, dichas redes cuentan con un gran número de capas ocultas y son óptimas para este tipo de estudio.

En cualquier caso, el uso de la inteligencia artificial, y en particular de los algoritmos que usan redes neuronales y *deep learning*, se manifiestan como una muy útil herramienta para analizar problemas de diferente índole mejorando incluso algunas de las técnicas estadísticas que, tradicionalmente, vienen usándose para el mismo fin.

## Apéndice A

# Anexo I. Fórmulas utilizadas en la construcción de variables indirectas

### A.1. Variables demográficas

- $P_t$ : Población en el instante  $t$ .
- $N(t, t+k)$ : Número de nacimientos entre los tiempos  $t$  y  $t+k$ .
- $D(t, t+k)$ : Número de defunciones entre los tiempos  $t$  y  $t+k$ .
- Población media:

$$\bar{P}_{[t, t+k]} = \frac{\sum_{i=t}^{t+k} P_i}{k}.$$

- Densidad:

$$d = \frac{\bar{P}_{[t, t+k]}}{\text{Superficie}}.$$

- Porcentaje con nacionalidad extranjera:

$$T_{extranjeros} = 100 \cdot \frac{\text{Nº de personas extranjeras en el año } t}{P_t}.$$

- Tasa bruta de natalidad por 1000 habitantes:

$$T_{natalidad} = 1000 \cdot \frac{N(t, t+k)}{k \cdot \bar{P}_{[t, t+k]}}.$$

- Tasa bruta de mortalidad por 1000 habitantes:

$$T_{mortalidad} = 1000 \cdot \frac{D(t, t+k)}{k \cdot \bar{P}_{[t, t+k]}}.$$

- Crecimiento natural:

$$CN(t, t+k) = N(t, t+k) - D(t, t+k).$$

## A.2. Variables socioeconómicas

- $P$ : Población total.
- $P_{15}$ : Población de 15 años o menos.
- $P_T$ : Población en edad de trabajar.
- $P_{65}$ : Población de 65 años o más.
- $A$ : Población activa.
- Tasa de atención sanitaria por 1000 habitantes:

$$T_{\text{Salud}} = 1000 \cdot \frac{\text{Nº de centros sanitarios}}{P}.$$

- Tasa de farmacias por 1000 habitantes:

$$T_{\text{Farmacias}} = 1000 \cdot \frac{\text{Nº farmacias}}{P}.$$

- Porcentaje de parados:

$$P_{\text{Paro}} = 100 \cdot \frac{\text{Nº de personas en el paro}}{P_T}.$$

- Tasa de actividad:

$$T_{\text{Actividad}} = 100 \cdot \frac{A}{P_T}.$$

- Tasa de viviendas nuevas por 1000 habitantes:

$$T_{\text{Viviendas}} = 1000 \cdot \frac{\text{Nº de viviendas nuevas}}{P_T + P_{65}}.$$

- Porcentaje de la población activa que trabaja en un determinado área:

$$P_{\text{Área}} = 100 \cdot \frac{\text{Nº de personas que trabajan en un área}}{A}.$$

- Tasa de centros escolares por 1000 habitantes:

$$T_{\text{Educación}} = 1000 \cdot \frac{\text{Nº de bibliotecas}}{P_{15}}.$$

- Tasa de bibliotecas por 1000 habitantes:

$$T_{\text{Bibliotecas}} = 1000 \cdot \frac{\text{Nº de bibliotecas}}{P}.$$

- Tasa de residencias por 1000 habitantes:

$$T_{\text{Residencias}} = 1000 \cdot \frac{\text{Nº de residencias}}{P_{65}}.$$

## Apéndice B

### Anexo II. Tabla completa de datos

Comarca código	1	2	3	4	5	6	7	8	9
Comarca nombre	La Jacetania	Alto Gállego	Sobrarbe	La Ribagorza	Cinco Villas	Hoya de Huesca / Plana de Uesca	Somontano de Barbastro	Cinca Medio	La Litera / La Litera
Población 1998	16531	12101	6740	11904	31738	55472	22339	21713	18631
Población 2018	17623	13474	7382	11931	30470	67446	23531	23656	18216
Población media 1998-2018	17810	13485	7312	12554	32251	64833	23509	23226	18744
Densidad 1998-2018	9.59	9.92	3.32	5.10	10.53	25.69	20.09	40.53	25.54
Edad media 2018	45.46	45.52	46.72	47.08	46.52	44.60	46.10	44.24	46.47
Porcentaje edad >65	21.19	21.07	22.83	23.78	23.32	21.52	23.72	21.06	25.27
Porcentaje edad <16	12.98	12.74	12.14	11.63	11.96	14.64	13.00	14.85	13.34
Índice de envejecimiento	121.67	121.12	142.07	150.93	141.35	111.07	134.13	107.53	140.43
Tasa de dependencia	51.91	51.07	53.76	54.96	54.52	56.65	58.03	56.04	62.90
Porcentaje con nacionalidad extranjera	5.27	9.57	11.38	11.53	11.64	7.86	8.54	13.20	11.90
Tasa bruta de natalidad por 1000 hbts	8.03	7.86	7.25	6.45	7.07	9.20	7.35	9.21	7.52
Tasa bruta de mortalidad por 1000 hbts	10.35	9.57	10.67	12.74	12.06	11.11	12.29	11.15	12.64
Tasa de crecimiento natural	-2.32	-1.71	-3.42	-6.23	-4.93	-1.91	-4.34	-1.94	-5.12
Renta per cápita	17725.03	17429.14	13960.05	13647.47	13086.80	17972.65	15204.24	14777.83	14251.39
Tasa atención sanitaria por 1000	2.44	1.04	3.39	3.75	1.61	1.41	1.87	0.63	0.93
Farmacias por 1000	0.74	0.59	0.81	0.58	0.56	0.40	0.68	0.51	0.71
Porcentaje de parados	5.26	6.08	4.54	3.83	7.11	6.51	6.81	7.04	6.05
Tasa de actividad	52.22	60.94	53.91	69.36	60.41	80.54	66.53	59.63	70.66
Viviendas nuevas por 1000	5.22	2.30	1.08	5.67	0.45	1.56	1.22	2.09	2.47
Centros escolares por 1000	6.56	8.16	14.51	9.27	7.14	4.76	5.22	5.41	5.76
Tasa Agricultura, ganadería y pesca	2.74	1.23	7.92	7.70	5.74	1.87	4.10	5.27	11.78
Tasa Construcción	9.87	8.52	13.14	9.61	5.83	4.28	4.65	5.69	6.18
Tasa Industria y energía	2.95	3.31	3.98	3.93	5.42	1.96	3.65	3.79	3.72
Tasa Servicios	55.66	47.87	70.40	45.76	28.66	28.88	33.66	33.16	33.02
Bibliotecas por 1000	0.34	0.45	0.95	0.50	0.36	0.25	0.59	0.34	0.49
Residencias por 1000	1.07	1.06	0.53	1.40	1.27	1.45	0.89	0.40	1.30
CRA (1998-2018)	0.30	0.54	0.46	0.04	-0.20	0.63	0.26	0.43	-0.11

Figura B.1: Tabla completa. Comarcas 1-9.

Comarca código	10	11	12	13	14	15	16	17
Comarca nombre	Los Monegros	Bajo Cinca / Baix Cinca	Tarazona y el Moncayo	Campo de Borja	Aranda	Ribera Alta del Ebro	Valdejalón	Central
Población 1998	21549	22301	14151	14177	8127	21746	21496	640346
Población 2018	18615	24425	13796	13824	6669	27076	28639	747082
Población media 1998-2018	20597	23750	14375	14602	7588	25386	26862	709953
Densidad 1998-2018	7.45	16.73	31.79	21.14	13.52	61.02	28.61	310.18
Edad media 2018	49.70	43.96	46.43	46.60	49.23	43.80	42.60	43.79
Porcentaje edad >65	29.07	20.61	24.86	24.69	26.77	19.65	18.64	20.29
Porcentaje edad <16	9.92	14.16	13.09	12.48	9.15	14.80	15.71	14.87
Índice de envejecimiento	216.27	110.15	140.86	143.89	193.60	102.80	91.64	104.02
Tasa de dependencia	63.91	53.31	61.17	59.15	56.04	52.54	52.32	54.23
Porcentaje con nacionalidad extranjera	9.31	20.04	5.70	10.45	7.68	9.47	20.18	9.79
Tasa bruta de natalidad por 1000 hbts	5.49	8.67	13.70	7.46	6.33	8.59	9.08	9.73
Tasa bruta de mortalidad por 1000 hbts	12.87	10.65	12.87	13.70	10.81	12.80	10.76	9.07
Tasa de crecimiento natural	-7.38	-1.98	0.83	-6.23	-4.48	-4.21	-1.68	0.66
Renta per cápita	11846.26	13373.22	12513.26	11952.98	9199.56	13582.31	11975.38	18861.29
Tasa atención sanitaria por 1000	2.58	0.57	1.23	1.30	1.80	0.63	0.59	0.11
Farmacias por 1000	1.02	0.61	0.72	0.72	0.75	0.70	0.63	0.47
Porcentaje de parados	4.64	6.11	8.46	6.97	9.31	7.67	9.72	8.55
Tasa de actividad	54.92	70.31	50.83	51.39	41.44	112.12	72.40	68.55
Viviendas nuevas por 1000	0.42	1.53	0.17	0.41	0.50	0.48	1.70	2.64
Centros escolares por 1000	15.16	7.52	4.43	8.70	9.84	6.49	5.11	3.53
Tasa Agricultura, ganadería y pesca	13.58	4.10	1.95	4.50	2.09	0.90	1.57	0.25
Tasa Construcción	6.54	4.19	6.89	6.47	7.00	2.83	4.51	3.82
Tasa Industria y energía	5.55	2.87	3.49	4.08	10.73	1.96	3.64	1.80
Tasa Servicios	32.28	26.53	35.51	35.80	43.53	13.35	22.33	29.05
Bibliotecas por 1000	0.75	0.33	0.14	0.51	0.30	0.41	0.42	0.11
Residencias por 1000	0.92	0.60	1.46	1.76	0.56	2.26	1.31	0.79
CRA (1998-2018)	-0.73	0.46	-0.13	-0.13	-0.98	1.10	1.44	0.77

Figura B.2: Tabla completa. Comarcas 10-17.

Comarca código	18	19	20	21	22	23	24	25
Comarca nombre	Ribera Baja del Ebro	Bajo Aragón-Caspe / Baix Arago-Casp	Comunidad de Calatayud	Campo de Cariñena	Campo de Bechite	Bajo Martín	Campo de Daroca	Jiloca
Población 1998	9664	12947	39851	9895	5680	7613	6781	14079
Población 2018	8464	14508	37230	9853	4569	6383	5557	12267
Población media 1998-2018	9225	13913	40134	10541	5221	7108	6327	13690
Densidad 1998-2018	9.32	13.95	15.97	13.65	5.00	8.94	5.66	7.08
Edad media 2018	47.85	44.58	47.83	47.36	53.75	49.36	52.72	49.30
Porcentaje edad >65	26.13	22.57	26.53	24.69	35.65	29.48	34.46	29.51
Porcentaje edad <16	11.85	14.22	11.18	11.10	7.66	11.17	8.12	11.31
Índice de envejecimiento	163.01	117.64	171.05	162.85	322.57	197.27	310.88	192.66
Tasa de dependencia	61.25	58.21	60.54	55.75	76.41	68.51	74.15	68.99
Porcentaje con nacionalidad extranjera	6.71	21.38	14.36	18.57	6.78	9.74	10.60	11.51
Tasa bruta de natalidad por 1000 hbts	6.40	8.34	6.45	6.55	4.21	5.91	4.27	6.14
Tasa bruta de mortalidad por 1000 hbts	12.14	12.51	13.63	13.09	16.09	15.34	15.02	14.10
Tasa de crecimiento natural	-5.75	-4.17	-7.18	-6.55	-11.87	-9.43	-10.75	-7.96
Renta per cápita	12973.33	10596.70	11785.02	10960.06	8442.19	11017.41	11009.72	11388.87
Tasa atención sanitaria por 1000	1.18	0.41	1.93	1.42	3.28	1.41	5.76	4.16
Farmacias por 1000	0.71	0.55	0.91	1.01	1.09	0.78	0.90	0.90
Porcentaje de parados	7.79	11.45	8.62	5.66	5.14	6.52	4.79	5.00
Tasa de actividad	67.77	55.08	44.80	76.76	45.91	77.30	59.26	52.38
Viviendas nuevas por 1000	1.07	0.72	0.18	0.11	0.24	0.88	0.00	0.55
Centros escolares por 1000	10.97	5.82	6.97	13.71	14.29	11.22	19.96	7.20
Tasa Agricultura, ganadería y pesca	5.54	4.40	1.77	2.08	9.84	5.81	5.29	7.13
Tasa Construcción	5.09	5.86	7.32	5.11	7.06	4.95	6.19	8.89
Tasa Industria y energía	5.99	3.50	3.60	3.73	6.14	5.02	4.49	5.18
Tasa Servicios	24.57	32.13	41.93	21.58	47.27	21.82	37.92	35.48
Bibliotecas por 1000	0.95	0.28	0.38	0.71	0.44	0.78	0.36	0.41
Residencias por 1000	1.36	0.61	1.11	1.23	2.46	1.59	1.57	1.38
CRA (1998-2018)	-0.66	0.57	-0.34	-0.02	-1.08	-0.88	-0.99	-0.69

Figura B.3: Tabla completa. Comarcas 18-25.

Comarca código	26	27	28	29	30	31	32	33
Comarca nombre	Cuencas Mineras	Andorra-Sierra de Arcos	Bajo Aragón	Comunidad de Teruel	Maestrazgo	Sierra de Albarracín	Gúdar-Javalambre	Matarraña / Matarranya
Población 1998	9881	11568	25962	42272	3717	5108	7779	8870
Población 2018	8067	10098	28587	45962	3196	4423	7367	8222
Población media 1998-2018	9178	11142	28616	45251	3596	4871	8091	8664
Densidad 1998-2018	6.52	16.50	21.94	16.21	2.99	3.44	3.44	9.29
Edad media 2018	47.60	46.35	44.46	45.06	49.29	50.89	46.45	48.51
Porcentaje edad >65	25.15	22.49	21.61	21.93	28.10	29.44	24.01	28.42
Porcentaje edad <16	11.22	11.92	14.82	13.91	10.29	8.14	12.46	12.08
Índice de envejecimiento	165.90	137.05	109.25	117.32	202.71	260.40	146.93	178.26
Tasa de dependencia	57.16	52.47	57.31	55.86	62.32	60.20	57.41	68.07
Porcentaje con nacionalidad extranjera	9.05	8.87	12.16	8.39	9.95	5.63	11.62	13.48
Tasa bruta de natalidad por 1000 hbts	6.54	7.09	9.23	8.71	6.12	4.93	7.04	6.58
Tasa bruta de mortalidad por 1000 hbts	11.44	10.05	11.64	11.18	13.63	13.55	12.73	15.12
Tasa de crecimiento natural	-4.90	-2.96	-2.41	-2.48	-7.51	-8.62	-5.69	-8.54
Renta per cápita	12511.69	15332.75	13630.07	16842.58	10343.53	9582.82	11732.45	10149.28
Tasa atención sanitaria por 1000	4.09	0.89	0.84	1.44	6.57	6.10	4.34	2.19
Farmacias por 1000	1.12	0.89	0.52	0.57	1.56	1.13	1.49	1.22
Porcentaje de parados	7.34	7.16	7.94	6.62	4.77	4.78	6.18	5.36
Tasa de actividad	45.12	37.79	55.90	81.71	48.96	43.90	66.00	65.78
Viviendas nuevas por 1000	0.56	0.22	0.74	1.97	1.05	0.98	2.95	0.83
Centros escolares por 1000	11.05	4.98	6.85	5.94	12.16	16.67	10.89	16.11
Tasa Agricultura, ganadería y pesca	2.16	1.96	3.14	1.01	14.73	2.56	3.04	8.83
Tasa Construcción	6.78	7.95	7.58	4.18	10.89	11.80	11.17	7.58
Tasa Industria y energía	6.39	4.79	4.32	2.12	7.68	6.19	5.54	6.18
Tasa Servicios	31.30	40.71	41.16	24.84	50.10	40.51	36.69	37.45
Bibliotecas por 1000	0.62	0.40	0.31	0.39	0.31	0.90	0.81	0.97
Residencias por 1000	1.48	0.88	0.81	0.50	0.00	1.54	1.70	0.86
CRA (1998-2018)	-1.01	-0.68	0.48	0.42	-0.75	-0.72	-0.27	-0.38

Figura B.4: Tabla completa. Comarcas 26-33.

# Bibliografía

- [1] RAMÓN LÓPEZ DE MÁNTARAS BADIA y PEDRO MESEGUEZ GONZÁLEZ, *¿Qué sabemos de? Inteligencia artificial.*, 2017.
- [2] FERNANDO SANCHO CAPARRINI, *Aprendizaje Supervisado y No supervisado*, <http://www.cs.us.es/~fsancho/?e=77>, Diciembre 2020.
- [3] MARGARET A. BODEN, traducción por INMACULADA PÉREZ PARRA, *Inteligencia artificial*, 2017.
- [4] *Regresión lineal*, [https://es.wikipedia.org/wiki/Regresión\\_lineal](https://es.wikipedia.org/wiki/Regresión_lineal).
- [5] JOSÉ MARTÍNEZ HERAS, *Regresión Logística para Clasificación*, <https://www.iartificial.net/regresion-logistica-para-clasificacion/>, Septiembre 2020.
- [6] ROSANA FERRERO y JUAN L. LÓPEZ, *Qué son los árboles de decisión y para qué sirven*, <https://www.maximaformacion.es/blog-dat/que-son-los-arboles-de-decision-y-para-que-sirven/>.
- [7] JOSÉ MARTÍNEZ HERAS, *Random Forest(Bosque Aleatorio): combinando árboles*, <https://www.iartificial.net/random-forest-bosque-aleatorio/>, Septiembre 2020.
- [8] JOSÉ MARTÍNEZ HERAS, *Máquinas de Vectores de Soporte (SVM)*, <https://www.iartificial.net/maquinas-de-vectores-de-soporte-svm/>, Mayo 2019.
- [9] CONSTANTINO MALAGÓN LUQUE, *Clasificadores bayesianos. El algoritmo de Naïve Bayes*, Universidad Nebrija, Mayo 2003.
- [10] *El algoritmo k-means aplicado a clasificación y procesamiento de imágenes*, [https://www.unioviedo.es/compnum/laboratorios\\_py/kmeans/kmeans.html](https://www.unioviedo.es/compnum/laboratorios_py/kmeans/kmeans.html), Universidad de Oviedo.
- [11] *DBSCAN*, <https://es.wikipedia.org/wiki/DBSCAN>.
- [12] FERNANDO BERZAL *Clustering basado en densidad*, Universidad de Granada.
- [13] *Algoritmo Apriori*, <https://aprendeia.com/algoritmo-apriori/>.
- [14] *Comprende Principal Component Analysis*, <https://www.aprendemachinelearning.com/comprende-principal-component-analysis/>.
- [15] *Neurona de McCulloch-Pitts*, [https://es.wikipedia.org/wiki/Neurona\\_de\\_McCulloch-Pitts](https://es.wikipedia.org/wiki/Neurona_de_McCulloch-Pitts).
- [16] DR. JUAN CARLOS CUEVAS TELLO, *Apuntes de Redes Neuronales Artificiales*, Universidad Autónoma de San Luis Potosí, Agosto 2017.
- [17] *Aprendizaje de Hebb*, <https://interactivechaos.com/es/manual/tutorial-de-deep-learning/aprendizaje-de-hebb>.

- [18] *Perceptrón*, <https://es.wikipedia.org/wiki/Perceptrón>.
- [19] *Perceptrón multicapa*, [https://es.wikipedia.org/wiki/Perceptrón\\_multicapa](https://es.wikipedia.org/wiki/Perceptrón_multicapa).
- [20] ALFONSO PALMER, JUAN JOSÉ MONTAÑO y RAFAEL JIMÉNEZ, *Tutorial sobre Redes Neuronales Artificiales: El perceptrón Multicapa*, Universitat de les Illes Balears, Revista electrónica de psicología, Vol.5, No.2, Julio 2001.
- [21] ANTONIO J. SERRANO, EMILIO SORIA y JOSÉ D. MARTÍN, *Redes neuronales artificiales*, Universitat de València, programa de doctorado, 2009.
- [22] JORDI TORRES, *Redes Neuronales Recurrentes*, <https://torres-ai.medium.com/redes-neuronales-recurrentes-13fb20dd8a6f>.
- [23] DIEGO CALVO, *Red Neuronal Convolutacional CNN*, <https://www.diegocalvo.es/red-neuronal-convolucional/>.
- [24] ÁLVARO ARTOLA MORENO, *Clasificación de imágenes usando redes neuronales convolucionales*, Trabajo de Fin de Grado, Universidad de Sevilla, 2019.
- [25] JUAN JULIÁN CEA MORÁN, *Redes neuronales recurrentes para la generación automática de música*, Trabajo de Fin de Máster, Universidad Politécnica de Madrid, 2020.
- [26] JOSÉ MARTÍNEZ HERAS, *Redes Neuronales Generativas (GANs)*, <https://www.iartificial.net/redes-neuronales-generativas-adversarias-gans/>, Octubre 2020.
- [27] JAVIER AGUADO, *Deep Learning: Un recorrido histórico*, <https://myclouddoor.com/deep-learning-un-recorrido-historico/>, Octubre 2017.
- [28] RODRIGO ALONSO, *IA, Machine Learning y Deep Learning, ¿cuál es la diferencia?*, <https://hardzone.es/tutoriales/rendimiento/diferencias-ia-deep-machine-learning/>, Abril 2020.
- [29] *Machine Learning made beautifully simple for everyone*, <https://bigml.com/>.
- [30] FRANCISCO RUIZ GONZÁLEZ y ÁNGEL RAÚL RUIZ PULPÓN, *Estudio sobre el proceso de despoblación en la provincia de Ciudad Real*, Universidad de Castilla-La Mancha, patrocinado por la Diputación Provincial de Ciudad Real, Mayo 2019.
- [31] *Instituto nacional de estadística(INE)*, <https://www.ine.es/>.
- [32] *Aragón Open Data*, <https://opendata.aragon.es/>.
- [33] *Instituto Aragonés de Estadística (IAEST)* <http://aplicaciones.aragon.es/mtiae/menu?idp=2&action=menu&tipo=3&idt=0>.
- [34] *Dibujo de la neurona natural*, <https://www.pinterest.com.mx/pin/858780222687300689/>