Minería de datos de los medios sociales, técnicas para el análisis de datos masivos

PID_00278309

Jordi Morales i Gras

Tiempo mínimo de dedicación recomendado: 3 horas

Universitat Oberta de Catalunya

Jordi Morales i Gras

Doctor en Sociología por la Universidad del País Vasco. Profesor de Análisis de redes, *machine learning* y *big data*, y socio director de Network Outsight, empresa especializada en el análisis sociológico de datos masivos.

El encargo y la creación de este recurso de aprendizaje UOC han sido coordinados por la profesora: Andrea Rosales Climent

Primera edición: octubre 2020

Av. Tibidabo, 39-43, 08035 Barcelona
Autoría: Jordi Morales i Gras
Producción: FUOC

Índice

Introducción

A estas alturas ya conocemos muy bien el paradigma de los datos masivos, la minería de datos y su cadena de valor. Ya sabemos que lo más habitual será que los datos habrán sido recogidos y registrados por alguien diferente al analista, y además con un propósito diferente. Esto implica que cedemos autonomía en el diseño del instrumento de captura de datos y que, a cambio, obtenemos un gran volumen de datos. Ahora bien, también sabemos que lo más importante no es el volumen de datos, sino la capacidad de interpretarlos para generar conocimiento e inteligencia.

He aquí la importancia de todas las estrategias que nos permiten exprimir los datos y añadirles valor. Una de estas técnicas es el análisis de las redes sociales, que permite generar conocimiento de acuerdo con la estructura relacional de las interacciones virtuales (interacciones entre usuarios e interacciones entre usuarios y otros objetos o plataformas). En la asignatura *Analítica avanzada en redes sociales* se estudia esta técnica en profundidad.

En este módulo nos centraremos en una estrategia clave para la minería de datos y la generación de valor en entornos de datos masivos: los algoritmos que generan conocimiento de acuerdo con los contenidos de los medios sociales. Para hacerlo, entraremos en el mundo de la inteligencia artificial, el aprendizaje automático y el procesamiento del lenguaje natural. Veremos la diferencia entre los algoritmos de aprendizaje supervisado y no supervisado, y también conoceremos los algoritmos híbridos y de conjunto, que mezclan las lógicas de los dos tipos anteriores. Posteriormente, nos centraremos en algunas de las técnicas específicas para el análisis de textos, uno de los contenidos más abundantes en los medios sociales, y veremos qué tipo de conocimiento es posible elaborar a partir de los datos. Para acabar, ubicaremos los algoritmos de datos masivos en la dimensión normativa del tratamiento justo de los datos, que comprende tanto la identificación de sesgos y discriminaciones en los sistemas de predicción y clasificación automatizada como el diseño consciente y mitigador de estas desigualdades.

1. Inteligencia artificial y *machine learning*

Por **inteligencia artificial** (IA) entendemos cualquier técnica que permite a un ordenador llevar a cabo una o varias acciones que aparenten o emulen alguna de las dimensiones de la inteligencia humana.

No hace falta que una IA se presente en forma de robot «humanoide» y que se haga pasar por una persona con sentimientos e ilusiones para considerar que una máquina es inteligente. En realidad, estamos aún muy lejos de este tipo de IA, pese a que la literatura y el cine la hayan abordado ya hace muchos años. De hecho, el ejemplo más habitual que podemos tener de IA en nuestro entorno son los brazos robóticos que se utilizan en la industria desde finales de los años setenta. Esta IA se denomina *IA débil*, y solo es capaz de llevar a cabo unas cuántas tareas «inteligentes». Contrastan con la *IA fuerte*, capaz de aprender cualquier tarea humana, aunque, por ahora, solo existe en la ficción.

Muchas IA débiles tienen la capacidad de aprender y hacer cada vez mejor aquella tarea en la que son expertas.

El **aprendizaje automático** o *machine learning* es la subdisciplina de la IA que persigue la mejora del propio sistema mediante el entrenamiento y la experiencia (Rebala y otros, 2019).

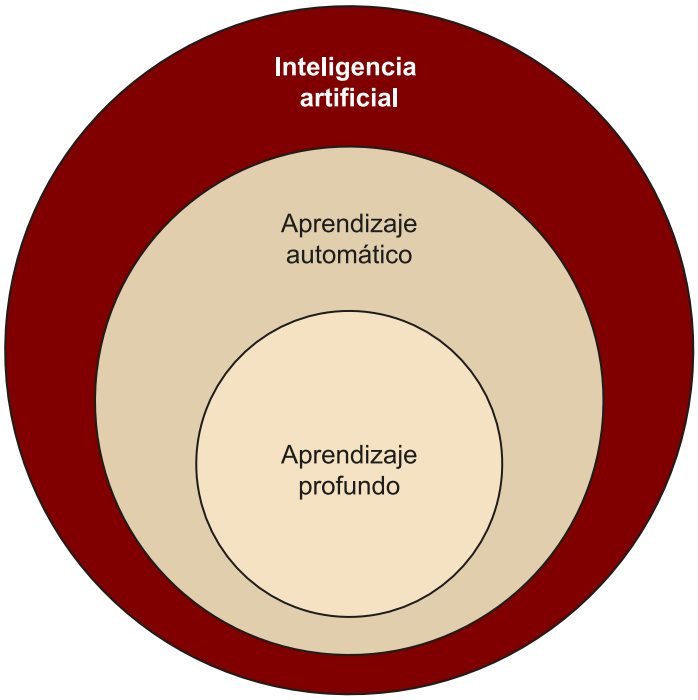
Esto quiere decir que, mediante el aprendizaje automático, un ordenador no solo es capaz de emular un comportamiento que parezca inteligente, sino que es capaz de hacerlo cada vez con más destreza y precisión. La mayoría de algoritmos que trabajan con datos masivos forman parte de esta familia de algoritmos.

Por ejemplo, los algoritmos que nos proponen contenidos en las redes sociales en función de nuestras amistades y de nuestras búsquedas, o los algoritmos que permiten llevar a cabo operaciones de «remarketing» en plataformas como Google o Amazon, o incluso los filtros de contenido basura más avanzados, que son capaces de adaptarse a las particularidades de cada usuario, en función de lo que este considere correo basura.

Tradicionalmente, ha habido dos tipologías de algoritmos de aprendizaje automático: el **aprendizaje supervisado y el no supervisado**. Se trata de técnicas preparadas para resolver problemas, como, por ejemplo, la clasificación automática de casos, la predicción numérica (la regresión) o la clusterización basándose en las propiedades intrínsecas de los datos.

Hoy, también existen una serie de **modelos híbridos** (por ejemplo, el aprendizaje semisupervisado o el aprendizaje reforzado) que combinan elementos de las dos tipologías y que profundizan en el concepto de autoaprendizaje por parte de la misma máquina. En esta última categoría, encontramos las técnicas de redes neuronales y de aprendizaje profundo o *deep learning* (figura 1), muy populares durante los últimos años y que actualmente están en fase de gran expansión y desarrollo, aunque también son técnicas discutidas por el tipo de resultados que proporcionan y por su opacidad.

Figura 1. Inteligencia artificial, aprendizaje automático y aprendizaje profundo



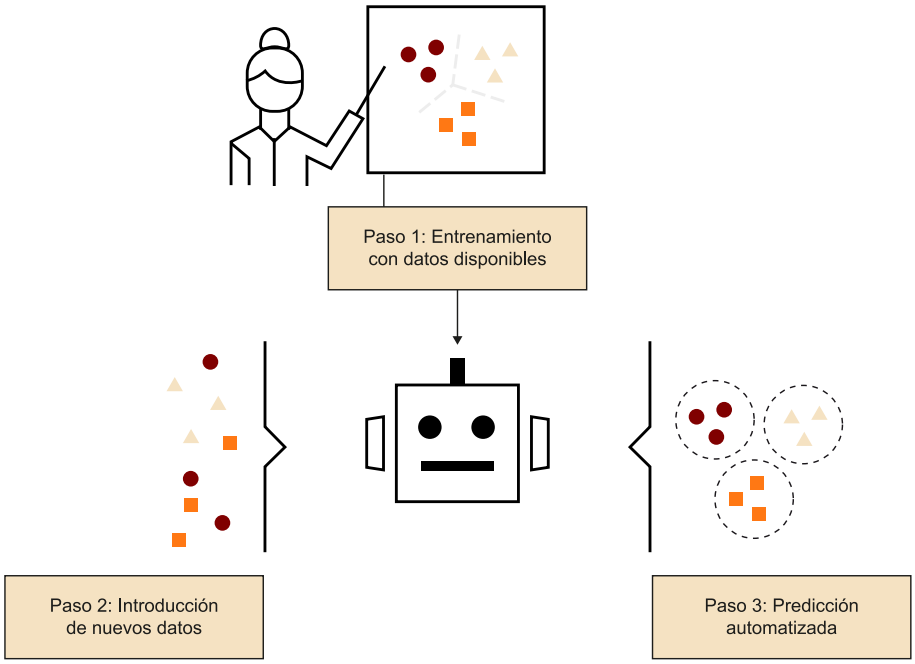
Fuente: elaboración propia.

1.1. Algoritmos de aprendizaje supervisado

Los **algoritmos de aprendizaje supervisado** persiguen la predicción de resultados futuros en función de series de datos conocidos.

El entrenamiento de un algoritmo mediante un proceso de aprendizaje supervisado se basa en que el analista proporcione al ordenador tanto los datos (los conjuntos de datos para entrenar y testar) como los resultados (el *output* esperado), y deje que sea el ordenador el que elabore un modelo basándose en alguna técnica específica debidamente parametrizada (el algoritmo). Es decir, el analista tendrá que alimentar el sistema con una serie de casos conocidos y previamente resueltos, y la máquina tendrá que aprender a resolver los nuevos casos con la ayuda de una secuencia de operaciones predefinida (figura 2).

Figura 2. Aprendizaje supervisado



Fuente: elaboración propia.

Los elementos que intervienen en una tarea de aprendizaje automático supervisado son tres, aunque se pueden presentar en diferentes formatos, que requerirán modos de articulación diferentes:

1) En primer lugar, están los **datos de introducción**: los datos a partir de los cuales el algoritmo tendrá que establecer la predicción. Estos datos toman diferentes nombres en la literatura científica, como, por ejemplo, «predictores», «características» (*features*), «variables de entrada» o «variables independientes».

2) En segundo lugar, está el conjunto de **datos que queremos predecir**. Estos datos se denominan «objetivo» (*target*), «variable de salida» o «variable dependiente», y se corresponderán con la «variable categoría» (*label*) con la que se habrá entrenado el modelo.

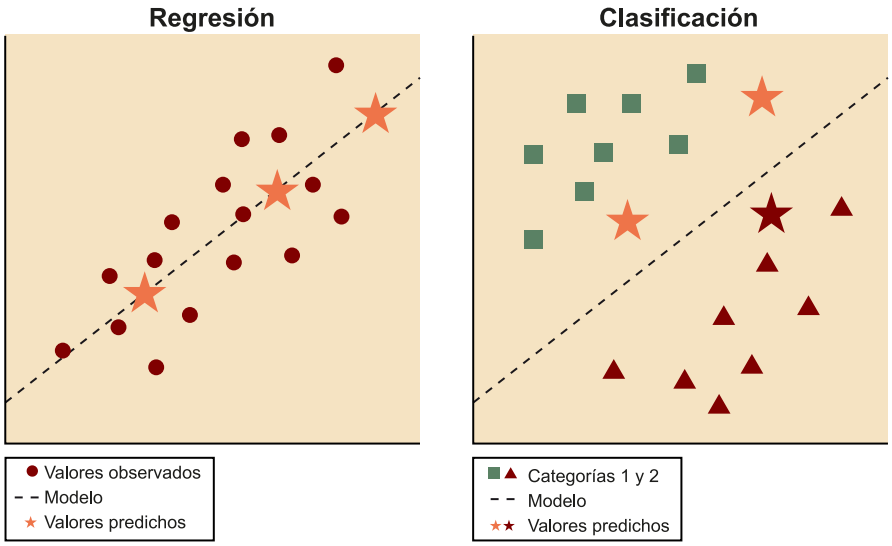
3) El tercer elemento es el **algoritmo de aprendizaje automático** que utilizaremos para elaborar el modelo. Consiste en un conjunto de instrucciones ordenadas y con un tipo de resultado previsible. En el contexto de una tarea de aprendizaje automático, el algoritmo a veces se denomina «aprendiz» (*learner*) o «inductor».

> Una de las tareas más habituales que tendrá que llevar a cabo un algoritmo es la clasificación de casos, como, por ejemplo, la separación de los correos deseados del correo basura. Para entrenar a un ordenador para llevar a cabo la tarea clasificatoria, habrá que contar con una base de datos con correos con toda la información posible que pueda ayudar al algoritmo en su aprendizaje: texto del correo, remitente, hora de envío, etcétera (datos de introducción). Adicionalmente, tendremos que proporcionar la respuesta esperada para unos cuantos casos –cuanto más, mejor; los algoritmos entrenados con miles o millones de casos ofrecerán resultados más precisos– divididos entre correos deseados o correos basura (datos objetivo). Finalmente, tendremos que decidir qué tipo de algorit-

mo de aprendizaje usamos para llevar a cabo el entrenamiento y, posteriormente, para establecer las predicciones (algoritmo inductor).

Los **algoritmos** son conjuntos de tareas que nos ayudan a resolver problemas matemáticos. En consecuencia, saber escoger y entrenar el mejor algoritmo posible para llevar a cabo una tarea determinada requiere entender muy bien el tipo de problema matemático que hay que resolver. Un primer elemento que condiciona todo el proceso es el tipo de variable dependiente. No es lo mismo predecir un número de una serie de una variable numérica o cuantitativa (por ejemplo, estimar el número de «Me gusta» que tendrá una publicación en Facebook) que predecir una categoría de una variable categórica o cualitativa (por ejemplo, clasificar un mensaje según el tema que trata). Por regla general, los problemas de predicción numérica los resolveremos con **modelos de regresión**, mientras que los problemas de predicción de categorías (por ejemplo, el tema de un mensaje) los resolveremos con **algoritmos clasificatorios** (figura 3).

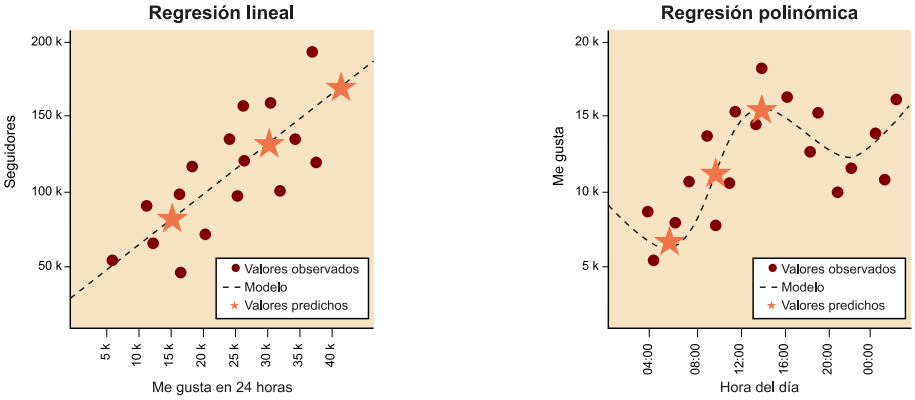
Figura 3. Regresión y clasificación



Fuente: elaboración propia.

Un segundo factor clave para escoger el algoritmo apropiado será la propia distribución de los datos: el patrón según el cual los datos toman sus valores por medio de las variables de la base de datos. Cuando nos encontramos con un problema de predicción numérica o cuantitativa, tendremos que preguntarnos cuál es el tipo de función que dibujan las variables en cuestión. No es lo mismo estimar un valor para una función lineal o para una función no lineal (figura 4). En el primer caso, más sencillo, optaremos por una **regresión lineal** (por ejemplo, predecir los «Me gusta» en función de los seguidores). En el segundo caso, tendremos que recurrir a métodos más complejos, como puede ser una **regresión polinómica** (por ejemplo, predecir los «Me gusta» según la hora del día).

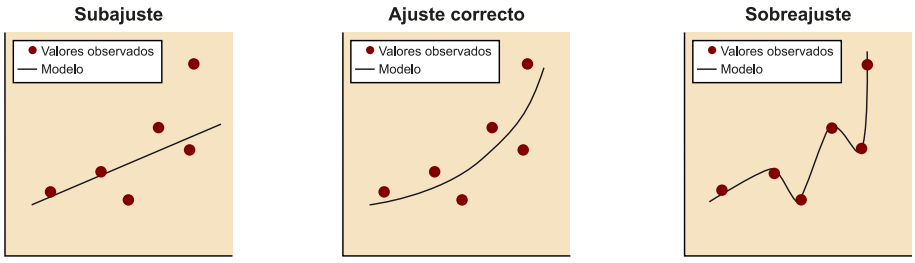
Figura 4. Regresión lineal y polinómica



Fuente: elaboración propia.

Las regresiones polinómicas son más complejas de implementar, pero esto no implica necesariamente que sean mejores ni que tengan una mayor capacidad predictiva que las lineales. Un peligro potencial de este método es el sobreajuste (*overfitting*), que consiste en el sobreentrenamiento de un algoritmo basándose en unos datos particulares, que lo incapacitan para hacer predicciones cuando los datos son diferentes. Por el contrario, el peligro potencial de las regresiones lineales es el subajuste (*underfitting*), que consiste en la simplificación excesiva del modelo, comprometiendo así su capacidad predictiva. Estos dos problemas (figura 5) son probablemente las causas principales de los errores en el aprendizaje automático y ponen de manifiesto hasta qué punto es de importante la interpretación de los datos y sus relaciones.

Figura 5. Problemas de subajuste y sobreajuste

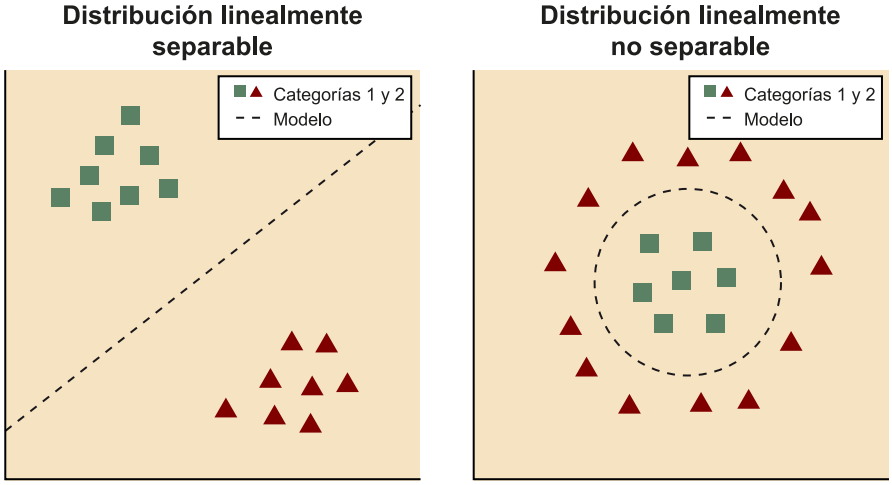


Fuente: elaboración propia.

En los problemas de predicción categórica, cuando queremos clasificar un caso nuevo en una u otra categoría de una variable cualitativa, también tendremos que distinguir los casos que son linealmente separables de los que no lo son (figura 6). Cuando nos encontramos con el caso de una distribución linealmente separable (por ejemplo, si tenemos que categorizar los mensajes en función de su idioma), podemos recurrir a algoritmos relativamente simples, como, por ejemplo, la regresión logística o el clasificador de vecino más cercano (k-NN). En cambio, cuando nos encontramos con distribuciones linealmente no separables, lo más habitual con datos provenientes de los medios sociales (por ejemplo, si tenemos que discriminar el sentimiento de un texto o decidir si un correo es correo basura o no), tendremos que recurrir a algoritmos más complejos, como, por ejemplo, el árbol de decisión (*decision tree*), o a algoritmos

híbridos –que incorporan elementos no supervisados– como, por ejemplo, el bosque aleatorio (*random forest*), el algoritmo de impulso adaptativo (*adaptive boosting* o AdaBoost) o las redes neuronales.

Figura 6. Categorías linealmente separables y no separables



Fuente: elaboración propia.

Para evaluar el **rendimiento de un algoritmo** desde un punto de vista matemático se utilizan técnicas estadísticas, como, por ejemplo, la retención (*holdout*) o la validación cruzada (*cross validation*).

Estas técnicas dividen los datos en conjuntos diferenciados que se utilizan como datos de entrenamiento o como datos de prueba, de forma que los datos de entrenamiento sirven para generar el modelo y los de prueba para validarlo. En realidad, la validación cruzada es una evolución perfeccionada de la retención. Mientras que el modelo de retención divide los datos solo en dos bloques, en el modelo de validación cruzada los datos se dividen en más grupos y las pruebas son más robustas: se obtiene más fiabilidad a cambio de un mayor coste computacional.

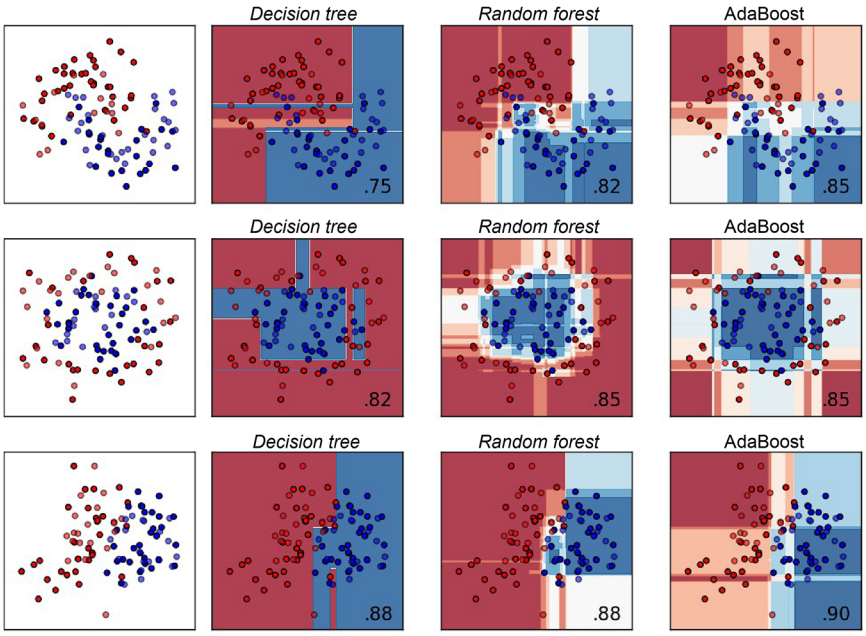
La principal métrica que nos permite cuantificar el **rendimiento de un modelo de regresión** es el coeficiente de determinación (R^2), que proporciona un valor entre –1 y 1 (valores cercanos a 1 indican un buen ajuste de la predicción, valores cercanos a 0 indican ausencia de capacidad predictiva del modelo y valores cercanos a –1 indican que el modelo tiene una capacidad de predicción peor que una simple línea horizontal), y nos informa de la capacidad del modelo para replicar los resultados obtenidos. En cambio, para cuantificar el **rendimiento de un modelo de clasificación**, usaremos métricas como la precisión clasificatoria (figura 7), que es el número medio de predicciones correctas hechas sobre los datos de prueba, expresadas como una porción del total. La métrica obtiene valores entre 0 y 1 y nos informa sobre la precisión del modelo (valores cercanos a 1 indican una capacidad predictiva cercana al 100 % y al 0 % cuando tienden a 0).

Los algoritmos que hemos denominado hasta ahora solo son unos pocos de los muchos disponibles en el software de *machine learning*. Muchos de ellos pueden ofrecer soluciones a problemas similares, en función de las características de dicho problema (el tipo de variable que se quiere predecir y la distribución de los datos) y, por eso, una práctica recomendable es poner a varios a competir entre sí y evaluar cuál es capaz de ofrecer una mejor solución a un mismo problema. Pero el rendimiento no lo es todo. El analista hará bien en escoger aquella solución que proporcione unos buenos resultados y, a la vez, permita una buena interpretación. Por ejemplo, como en el caso de las predicciones numéricas, en las categóricas también nos podemos encontrar con problemas de sobreajuste o subajuste. Estos problemas adquieren mayor potencial cuanto más complejo y difícil sea interpretar un algoritmo. El problema reside, como veremos más adelante, en el hecho de que en el paradigma del aprendizaje automático, muchas veces, a medida que aumenta la capacidad predictiva, también aumenta la opacidad algorítmica: cuanto más complejos e ininterpretables son los algoritmos, más capacidad de predicción demuestran.

Ved también

Ved estas cuestiones en mayor profundidad en el apartado «El tratamiento justo de datos».

Figura 7. Precisión clasificatoria de tres algoritmos para tres problemas



Fuente: https://martin-thoma.com/comparing-classifiers/.

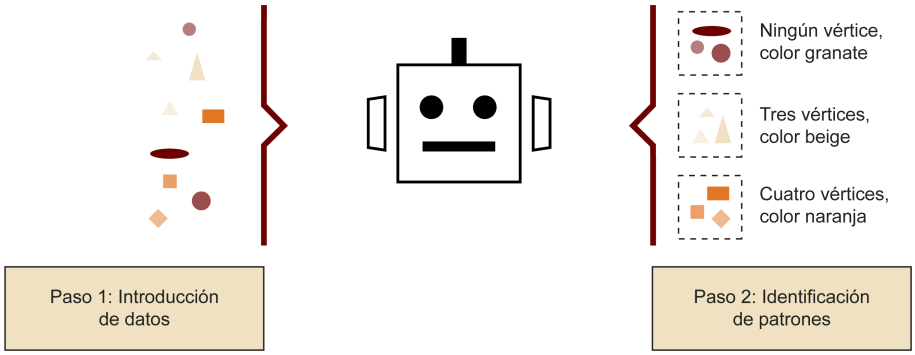
1.2. Algoritmos de aprendizaje no supervisado

Una de las posibilidades más interesantes del paradigma de los datos masivos es que, gracias al volumen y a la capacidad computacional de los sistemas actuales, podemos partir de modelos de investigación inductivos, donde la observación va antes que la teoría. Este es el espíritu que hay detrás de los algoritmos de aprendizaje no supervisado, cuyo objetivo es clasificar los datos en función de sus propiedades intrínsecas, sin partir de un modelo predictivo preentrenado.

Los **algoritmos de aprendizaje no supervisado** son capaces de identificar patrones en los datos sin recibir instrucciones específicas por parte del analista.

Aquí no se trata de que el investigador prepare el ordenador para llevar a cabo una clasificación o una predicción numérica, sino que sea la misma máquina la que descubra patrones en los datos por sí misma (figura 8) y que, de este modo, visibilice o haga emerger una serie de propiedades de los datos que el analista podría haber pasado por alto.

Figura 8. Aprendizaje no supervisado



Fuente: elaboración propia.

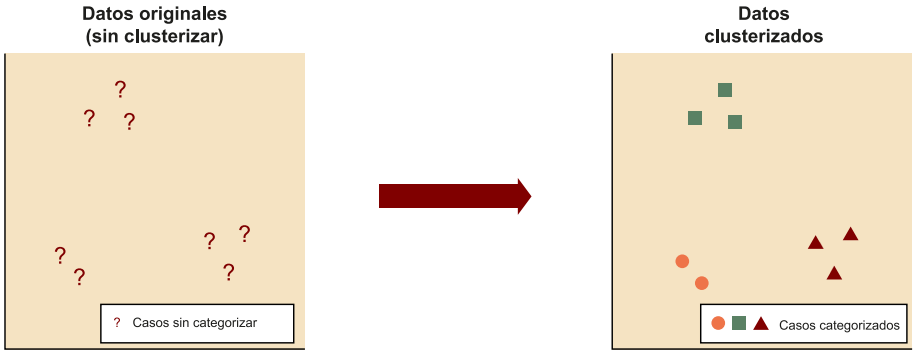
El aprendizaje no supervisado parte de un tipo de razonamiento que los seres humanos conocemos muy bien, el **método inductivo**, una estrategia que forma parte del método científico pero que nunca ha sido, hasta hoy, la estrategia más utilizada por los científicos. Tradicionalmente, estos han utilizado el método deductivo. La diferencia entre los dos tipos de razonamiento está en la relación entre la teoría y la observación. Cuando utilizamos el método inductivo, no partimos de una teoría o modelo de relación entre las variables que queremos validar, sino que partimos de la observación empírica e intentamos sintetizar una teoría o modelo. En la versión computacional de este tipo de razonamiento, hay un número más reducido de elementos que en los algoritmos de aprendizaje supervisado, puesto que no tendremos que diferenciar entre las variables dependientes e independientes, ni entre datos de entrenamiento y datos de prueba. Por un lado, tendremos los datos que queremos ordenar o simplificar y, por otro lado, tendremos el algoritmo que llevará a cabo la tarea en cuestión.

Para escoger el mejor algoritmo de aprendizaje no supervisado, como en el caso del aprendizaje supervisado, tendremos que entender muy bien cuál es el problema matemático que queremos resolver y qué tipo de datos intervienen. La tarea más sencilla que podemos llevar a cabo mediante un proceso no supervisado consiste en ordenar los datos y agrupar los casos similares. Esta tarea se denomina **clusterización** (figura 9) y podemos llevarla a cabo mediante técnicas como el algoritmo k-Means o el algoritmo Louvain multinivel, enormemente eficiente en datos reticulares o de redes. Otras tareas más complejas que

se pueden resolver con este tipo de algoritmos son la **reducción dimensional**, que querremos llevar a cabo cuando nos encontramos con muchas variables o dimensiones, o también la **resolución de problemas de reglas** y la **asociación entre variables**. Técnicas como el análisis de componentes principales (PCA) o el algoritmo Eclat pueden ayudarnos a llevar a cabo estas tareas.

Tan importante como el problema matemático o la tarea que se quiere resolver de manera no supervisada es el tipo de datos con que contamos, especialmente en el paradigma de los datos masivos, caracterizado por una enorme variabilidad de formatos. Hoy en día, hay algoritmos que nos ayudan a clasificar datos de cualquier tipo: textos, imágenes, vídeos, audios, etc. En el caso de los corpus documentales, son muy importantes los algoritmos de **modelado temático** (*topic modelling*), que identifican grupos de palabras utilizadas conjuntamente en textos (temas). En cuanto a las imágenes, vídeos y audios, lo más habitual es identificar los patrones de manera no supervisada después de haberlos transformado en vectores numéricos mediante algoritmos híbridos llamados ***encajes*** (*embeddings*).

Figura 9. Clusterización (no supervisada)



Font: elaboración propia.

El procedimiento de evaluación de los resultados obtenidos por un algoritmo de aprendizaje no supervisado es bastante diferente al de un algoritmo supervisado, porque no implica técnicas de validación cruzada.

> En las tareas de clusterización, es habitual comprobar hasta qué punto se han clasificado los casos correctamente, en particular y en conjunto, mediante métricas como el **valor «silhouette»** o el **estadístico de modularidad**.

Ambas métricas proporcionan una cifra entre –1 y 1 que nos informa hasta qué punto los casos están bien agrupados en las categorías identificadas (valores cercanos a 1 indican que los casos agrupados se asemejan entre sí y son muy diferentes del resto, valores cercanos a 0 indican que los casos agrupados se asemejan tanto entre sí como con el resto, y valores cercanos a –1 indican que los casos agrupados se asemejan más al resto que entre sí). No obstante, es importante entender que no se trata de métricas de validación puras, dado

que muchos algoritmos las incorporan como procedimientos de optimización de los mismos procesos. Esta diferencia, lejos de ser una debilidad del método inductivo, es una de sus principales virtudes, puesto que permiten que el analista se apoye en métricas para la construcción de las categorías analíticas.

Todas las virtudes del aprendizaje no supervisado hay que contrastarlas con sus limitaciones. Para empezar, es un error pensar que este tipo de modelos hayan convertido en obsoletas las teorías científicas, tal como han afirmado algunos observadores excesivamente optimistas (Anderson, 2008). El aprendizaje no supervisado actualmente disponible es un excelente asistente para la elaboración teórica, que acompaña y ayuda al analista, pero en ningún caso sustituye sus funciones. Muchas veces, tal como veremos más adelante, los algoritmos de aprendizaje no supervisado proporcionan resultados demasiado evidentes o redundantes. En el peor de los casos, los errores metodológicos cometidos en las fases de preparación de los datos o la falta de supervisión humana pueden conducir al hecho de que una máquina adquiera comportamientos discriminatorios e indeseables.

Un buen ejemplo es el algoritmo de selección de personal de Amazon, que discriminaba a las mujeres, y otro, el *chatbot* Tay de Microsoft, que en Twitter adquirió un lenguaje racista y una «ideología» neonazi en solo dos días de autoaprendizaje basado en conversaciones de jóvenes entre dieciocho y veinticuatro años (Metz, 2016). Por todo esto, es importante entender que se trata de procesos que pueden requerir un esfuerzo interpretativo elevado, que muchas veces serán difíciles de sistematizar.

Ved también

Ved el módulo «Datos masivos y minería de datos sociales, conceptos y herramientas básicas».

1.3. Algoritmos híbridos y de conjunto

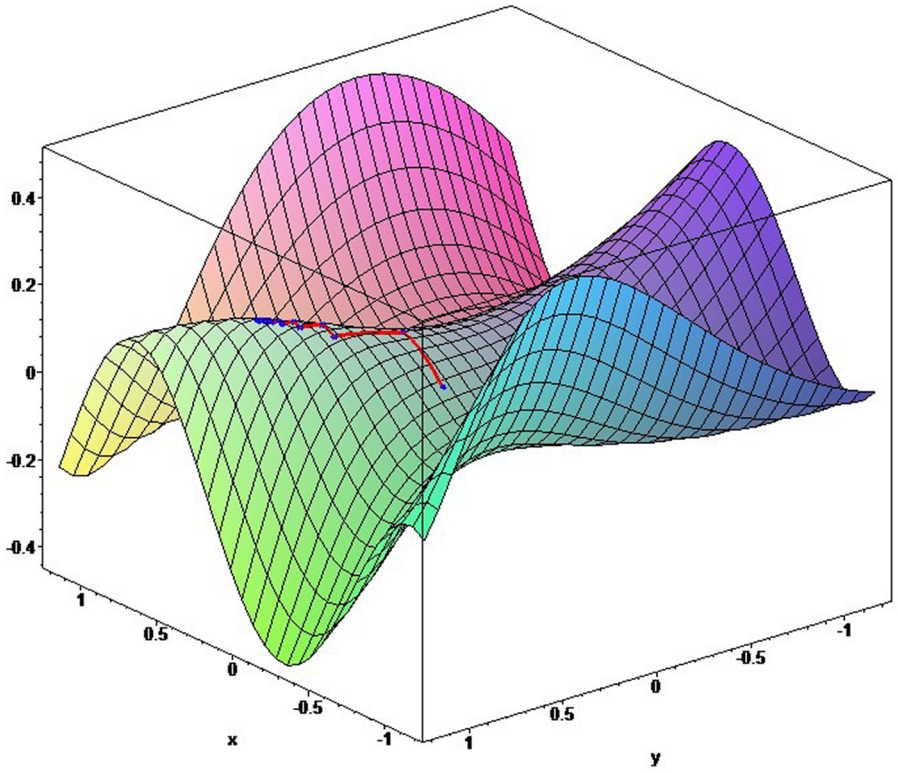
Los **algoritmos híbridos y de conjunto** combinan varios aspectos de los algoritmos que hemos visto anteriormente con el objetivo de mejorar el poder predictivo de un modelo.

Se trata de opciones muy interesantes cuando se quiere predecir un resultado en función de diferentes y múltiples causas.

Por ejemplo, para predecir los «Me gusta» de una publicación en función del tema, la longitud del texto, su contenido emocional, la hora del día, el tipo de imagen o vídeo, su tonalidad cromática o cualquier otro aspecto medible.

Algunas de estas técnicas son los algoritmos de **gradiente descendente** (*gradient descent*), de **impulso adaptativo** (*adaptive boosting* o AdaBoost) y de **bosques aleatorios** (*random forests*). También los algoritmos más populares de los últimos años, las **redes neuronales profundas**, forman parte de esta categoría. La principal virtud de estas técnicas es su capacidad para establecer correlaciones entre variables en múltiples dimensiones, lo que les permite lograr una gran capacidad predictiva, pero que dificulta enormemente la interpretación por parte del analista (figura 10).

Figura 10. Representación de una tarea multidimensional resuelta por un algoritmo de gradiente descendente



Fuente: adaptada de Wikicommons (imagen libre de derechos).

La función principal, y más explotada, de este tipo de algoritmos es la **resolución de problemas clásicos de categorización o de predicción numérica** mediante la incorporación de elementos de autoaprendizaje no supervisado. Es decir, el algoritmo parte de un entrenamiento implementado por el analista, pero es capaz de seguir aprendiendo de manera autónoma en función de sus propios éxitos y fracasos. Como ya hemos visto, este tipo de procedimientos corren el peligro de entrenarse basándose en datos sesgados y de producir resultados igualmente sesgados y de mala calidad, a veces incluso discriminatorios, por ejemplo, los casos ya mencionados del algoritmo de Amazon para la selección de personal o el *chatbot* racista de Microsoft.

Durante la última década, el campo de estudio más prometedor del aprendizaje automático han sido los **algoritmos de redes neuronales y de aprendizaje profundo**. Se trata de algoritmos de caja negra, totalmente orientados a la predicción, y que no pretenden, en ningún caso, ser utilizados por un analista que intente entender o comprender la relación entre las variables de un modelo. Es en este punto donde el aprendizaje automático toma mayor distancia respecto de la estadística inferencial, que tiene como objetivo principal la formalización y la interpretación de las relaciones entre las variables. Otra característica del aprendizaje profundo es el hecho de desentenderse totalmente del principio de parsimonia, que es reemplazado por la capacidad de computación.

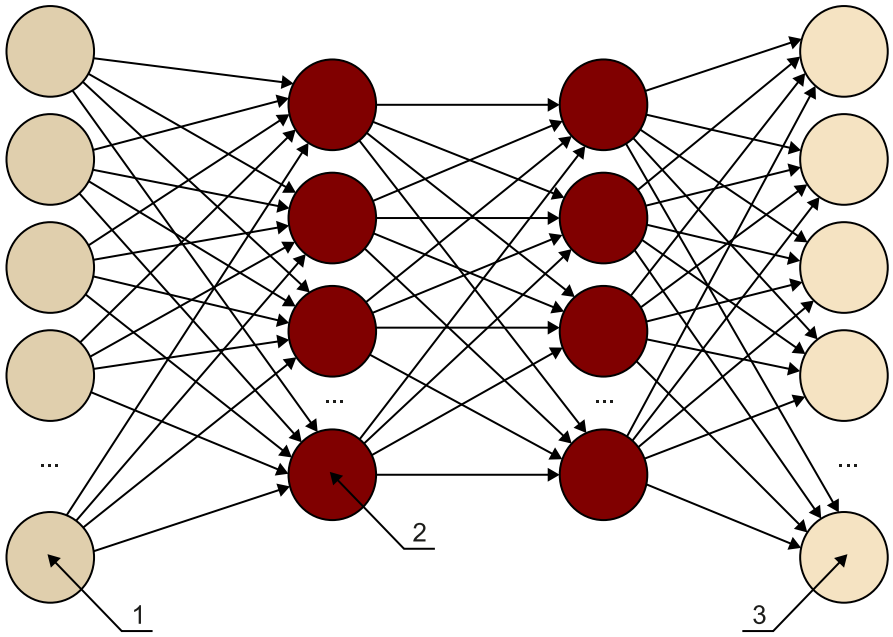
Principio de parsimonia y capacidad de computación

El **principio de parsimonia** se basa en que las explicaciones sencillas de pocas variables son superiores a las complejas si tienen capacidades explicativas parecidas.

En la **capacidad de computación**, carece de importancia el peso específico de una u otra variable mientras la máquina pueda procesar el problema en un tiempo asumible.

El estudio de las relaciones entre las variables y sus pesos específicos que anhela la estadística inferencial resulta del todo imposible cuando nos encontramos ante un modelo de caja negra. Como su nombre indica, los modelos de caja negra consisten en procesos internos que elaboran las máquinas y que no pueden ser interpretados por el analista. Las redes neuronales profundas (figura 11) establecen diferentes niveles de conocimiento de manera oculta, de forma que el analista solo tiene acceso al *input* y el *output* del sistema. Dado que son algoritmos que aprenden solos, son capaces de lograr una enorme capacidad predictiva con el tiempo, pero, por otro lado, son algoritmos bastante complicados de parametrizar, y todavía más difíciles de analizar, y que tienden a asumir e integrar sesgos no corregidos en los datos que utilizan para entrenarse.

Figura 11. Modelo de una red neuronal profunda



Fuente: Wikicommons (imagen libre de derechos).

Los algoritmos de caja negra plantean problemas a los analistas de datos, pero también presentan problemas importantes de opacidad y pueden, incluso, adquirir comportamientos discriminatorios, como el caso del algoritmo entrenado por Amazon para la selección de personal, que ya vimos, o como el *chatbot* Tay de Microsoft. Cada vez son más los expertos en inteligencia artificial que recomiendan no utilizarlos (Campolo y otros, 2017). La alternativa a los algoritmos de caja negra son los **algoritmos denominados de «caja blanca» o transparentes**: algoritmos fáciles de interpretar y que permiten explorar las relaciones establecidas entre las variables, como pretende la estadística. En realidad, más que de blancos y negros, se trata de una escalera de grises: mientras que las regresiones y los árboles de decisiones son más o menos sencillos de interpretar, la complejidad crece cuando se trata de interpretar algoritmos *random forest* o AdaBoost, especialmente adecuados para la clasificación de distribuciones no lineales.

Por lo tanto, decidir qué algoritmo utilizar depende de varios factores. Como ya hemos visto, es importante tener en cuenta criterios estrictamente matemáticos y vinculados al tipo de problema que queremos resolver: el tipo de datos, su distribución o sus formatos. También dependerá, en gran medida, del planteamiento metodológico de una investigación: el aprendizaje supervisado es útil para los planteamientos hipoteticodeductivos y el no supervisado para los inductivos. No obstante, cada vez son más las voces que reclaman que el rendimiento y la capacidad predictiva no tendrían que ser las únicas variables que se deben tener en cuenta. También es importante que los algoritmos sean transparentes e interpretables, puesto que el hecho de que no lo sean compromete totalmente el objetivo final de cualquier investigación, que no es otro que la creación de valor. Una vez más, es responsabilidad del analista mantener un equilibrio entre la transparencia (la facilidad de interpretación de los resultados) y la precisión (la capacidad predictiva) de los modelos.

2. Algoritmos para el procesamiento del lenguaje natural

Los contenidos de los medios sociales son, conjuntamente con las relaciones que se establecen, una enorme fuente de datos masivos. Entre los formatos de publicación, además de los textos, hay imágenes, vídeos y audios.

Nota

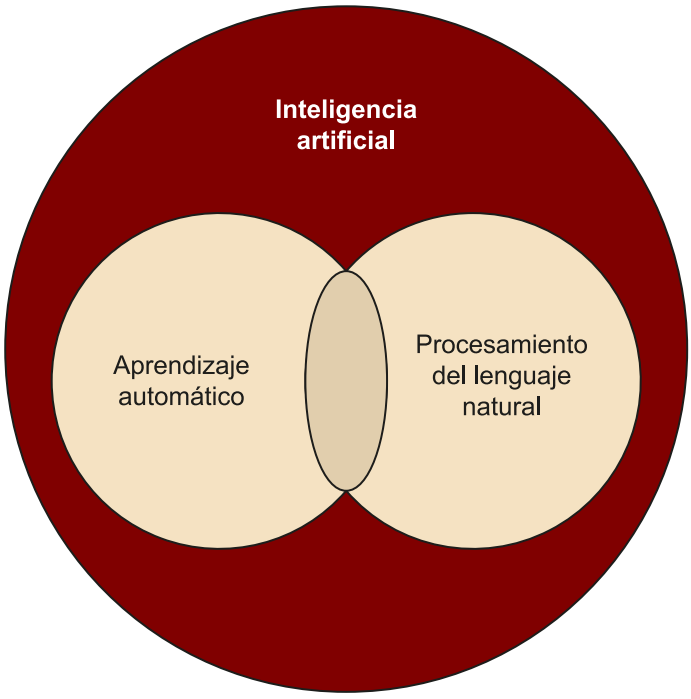
En este módulo no profundizaremos en las técnicas de análisis específicas de estos formatos –que generalmente se analizan después de ser convertidos en vectores numéricos llamados encajes o *embeds*, mediante técnicas parecidas a algunas de las que veremos aquí–, sino que nos centraremos en los contenidos escritos, que son, sin duda, los más abundantes y transversales, puesto que los encontramos en todas las plataformas.

La subdisciplina de la inteligencia artificial, la informática y la lingüística que se ocupa del análisis de textos y documentos es el **procesamiento del lenguaje natural** (PLN).

Se trata de un campo de estudio muy amplio que cubre tareas que van de desde la traducción automática o la predicción de textos hasta las vinculadas a la evaluación de discursos (Bird y otros, 2009). El mejor ejemplo de este tipo de técnicas son las funcionalidades de traducción y de texto predictivo de buscadores como Google o Bing, que los últimos años han mejorado considerablemente.

Se puede decir que el campo de análisis del PLN es tan amplio y multidimensional como lo es el mismo lenguaje humano y, por lo tanto, se nutre de elementos tan diferentes como la morfología, la gramática, el léxico, la semántica, la pragmática, la entonación, la fonética, etc. El PLN es un campo de la IA, pero no del aprendizaje automático (figura 12). Esto es así porque el PLN comprende procesos y algoritmos de diferente naturaleza y no todos implican modelos en los que la máquina aprende basándose en un entrenamiento o por sí misma. Es importante distinguir las aproximaciones basadas en **reglas heurísticas** (secuencias de operaciones que proporcionan soluciones «suficientemente buenas» para los objetivos del procedimiento y en un tiempo corto) de las **soluciones de aprendizaje automático**: supervisado, no supervisado e híbrido.

Figura 12. IA, aprendizaje automático y PLN



Fuente: elaboración propia.

2.1. Procesos de las reglas heurísticas

Las tareas más sencillas que se llevan a cabo en un PLN son de orden sintáctico y léxico, y suelen ser subprocesos o pasos preparatorios de algoritmos más complejos. Se trata de operaciones computacionalmente muy eficientes que se llevan a cabo basándose en reglas preestablecidas y que, por lo tanto, no implican un proceso de aprendizaje automático como tal. En estos casos, la máquina dispone de una guía o de un diccionario diseñado manualmente, identifica cada caso o situación y aplica la solución manualmente predefinida. Las siguientes son algunas de las **operaciones más habituales** de un PLN basadas en reglas heurísticas:

1) Tokenización. Es la segmentación del texto en partes más pequeñas llamadas *tokens* (normalmente palabras, pero también pueden ser frases, párrafos u otras partes de un texto), según unos espacios y signos de puntuación. En la mayoría de lenguas de nuestro entorno (románicas) esta tarea no implica ninguna dificultad. La dificultad aumenta en lenguas con características aglutinantes como, por ejemplo, el japonés, el finés o el euskera.

2) Filtraje. Conjuntamente con la tokenización, es habitual llevar a cabo un proceso de eliminación de palabras innecesarias o inadecuadas para el análisis. Las llamadas *palabras vacías* son las más comunes de una lengua, como, por ejemplo, las preposiciones y los artículos. Es conveniente que el analista conozca bien el listado de palabras que utiliza, puesto que no hay ningún cri-

terio unificado y puede ser que una palabra sea innecesaria en un contexto y necesaria en otro. El proceso de filtraje también se puede aplicar en positivo, dejando que solo un conjunto determinado de palabras pase el filtro.

3) Lematización. Consiste en asociar todas las palabras de un texto a su lema o forma canónica, previamente indexada (por ejemplo, «seremos», «erais», «eres» = «ser»). El proceso heurístico implica un análisis morfológico de cada palabra y un diccionario detallado del idioma o idiomas en que está escrito el texto.

4) Bolsa de palabras. Es un método que se puede aplicar después de la tokenización, el filtraje y la lematización, y que consiste en agrupar las palabras en «bolsas», de forma que cada texto quede asociado a las diferentes palabras que contiene, ignorando su orden original. Típicamente, las matrices de datos procesados con este método tienen tantas filas como piezas de texto y tantas columnas como palabras.

5) Valor TF-IDF. Es una técnica similar a la bolsa de palabras, pero que pondera cada palabra en función del número de veces que aparece en un documento en relación con la presencia de la palabra en el conjunto de documentos. De este modo, las palabras más idiosincrásicas de cada documento (las que lo distinguen del resto de documentos de un corpus) toman más protagonismo. Es una técnica muy útil si se quiere clasificar documentos maximizando sus diferencias.

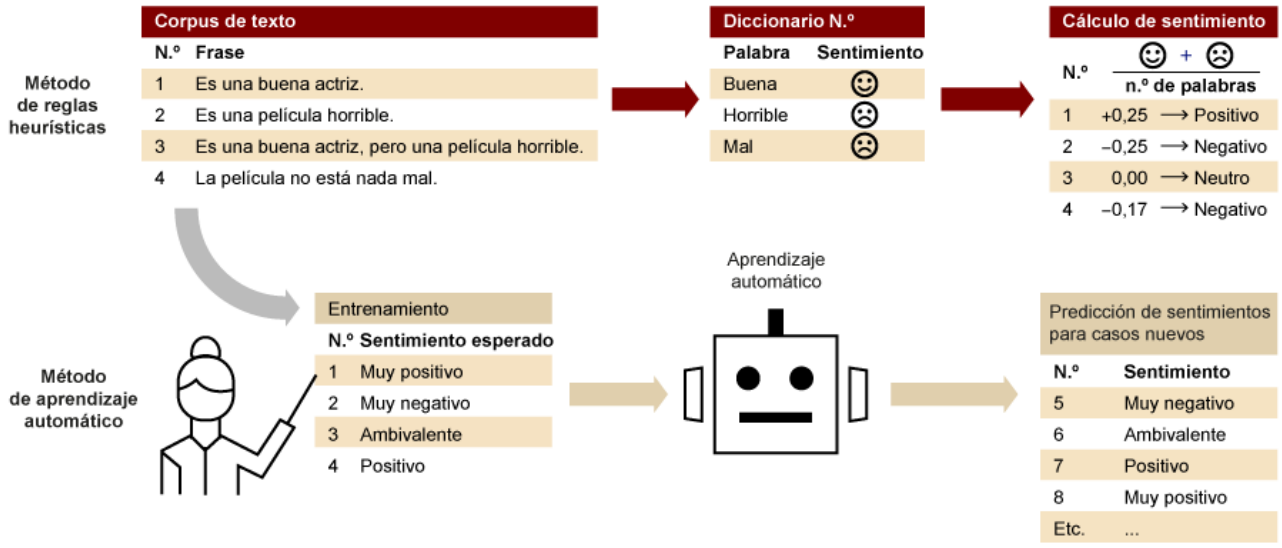
Las técnicas anteriores son algunas de las más habituales en cualquier análisis de textos, dado que constituyen precondiciones para multitud de análisis. No siempre será recomendable aplicarlas, ni hacerlo del mismo modo. Por ejemplo, en textos muy cortos (por ejemplo, tuits) muchas veces será preferible no lematizar las palabras, puesto que muchos algoritmos necesitan contar con una cierta diversidad morfológica para tener un buen rendimiento, especialmente los algoritmos no supervisados y los de aprendizaje profundo. Además, algunas de estas técnicas son mutuamente excluyentes, como, por ejemplo, la bolsa de palabras y la técnica TF-IDF.

Una de las técnicas más populares del PLN, el **análisis de sentimiento**, es otro ejemplo de procedimiento basado en reglas heurísticas. En realidad, su funcionamiento es muy sencillo: se parte de un listado de palabras llamadas *positivas* y de un listado de palabras llamadas *negativas*, y se procede a hacer un recuento sobre el corpus del texto en cuestión. El texto se considerará «positivo» o «negativo» en función del tipo de palabras más abundantes, y «neutro» cuando haya un empate. Este tipo de análisis puede ser muy útil en los medios sociales, puesto que nos permite automatizar la detección de crisis reputacionales o aproximarnos a la opinión del conjunto de una audiencia sobre una marca o producto. Ahora bien, es un procedimiento con limitaciones muy no-

tables, sobre todo vinculadas al contexto (por ejemplo, el adjetivo *fría* podrá ser adecuado para una bebida e inadecuado para una pizza) o a los elementos pragmáticos del lenguaje (por ejemplo, la ironía o el sarcasmo).

Durante los últimos años, se han desarrollado varias aproximaciones al análisis de sentimiento y a otros problemas clásicos del PLN desde paradigmas algorítmicos de aprendizaje supervisado e híbrido, especialmente desde las redes neuronales profundas (*deep learning*). Estos modelos han ofrecido mejoras sustanciales respecto a los modelos basados en reglas heurísticas (figura 13). También se han desarrollado recientemente modelos mixtos, que combinan las dos aproximaciones y ofrecen resultados bastantes buenos (Ray y Chakrabartib, 2019). De todas formas, en materia de análisis de sentimiento todavía estamos lejos de poder diseñar procedimientos y algoritmos que proporcionen resultados homologables a los que obtiene la cognición humana. Se trata de un área del PLN bastante subdesarrollada, sobre todo si la comparamos con los enormes adelantos de los últimos años en otros problemas clásicos, como, por ejemplo, las tareas de traducción de textos o de predicción de palabras.

Figura 13. Análisis de sentimiento con reglas heurísticas y con aprendizaje automático



Fuente: elaboración propia.

2.2. Aprendizaje supervisado aplicado a textos

La mejor manera de obtener buenos resultados con un algoritmo es entrenarlo específicamente para que lleve a cabo la tarea o las tareas que queremos que haga. De este modo, mediante modelos específicos de aprendizaje, es posible superar aspectos como la marcada dependencia contextual del lenguaje (por ejemplo, una bebida «fría» será probablemente adecuada, y una pizza «fría» será probablemente inadecuada). Muchas de las limitaciones de los sistemas basados en reglas heurísticas se pueden superar con modelos de aprendizaje supervisado, como en el caso del análisis de sentimiento. Al fin y al cabo, identificar el sentimiento de una frase, o identificar si contiene un tipo de lenguaje, o si pertenece a una lista predeterminada de temas, es un problema clásico

de categorización, que se puede resolver mediante un algoritmo del estilo de una regresión logística o de un árbol de decisión. La flexibilidad de los algoritmos de aprendizaje supervisado permiten que cualquier analista pueda crear su propio algoritmo para categorizar textos nuevos, si se dispone de suficientes textos previamente categorizados.

Los **elementos necesarios** para crear un algoritmo de clasificación de textos no son tan diferentes de los que podemos aplicar a cualquier problema que debemos resolver mediante el aprendizaje supervisado:

1) Los **datos de introducción** serán los textos a partir de los cuales se entrenará el algoritmo. Pueden ser textos largos (por ejemplo, novelas, noticias, webs, artículos de la Wikipedia, etc.) o cortos (por ejemplo, publicaciones de Facebook, tuits, etc.), y constituirán la materia prima para crear nuestras variables de entrada (el corpus). Generalmente, aplicaremos los procesos de las reglas heurísticas sobre el corpus para generar las variables independientes de nuestro modelo, como, por ejemplo, las palabras contenidas en cada texto (por ejemplo, modelo de bolsa de palabras o TF-IDF).

2) Los **datos que se van a predecir** podrán ser de dos tipos: cuantitativos o cualitativos. Por ejemplo, se puede intentar predecir el número de Compartidos o Me gusta de una publicación de Facebook (variable numérica) en función de las palabras que aparecen; o se puede predecir el autor de un libro o la temática (variable categórica) en función del mismo criterio. En cualquier caso, el analista tendrá que proporcionar las respuestas a una serie de casos para entrenar al ordenador, creando así un algoritmo único.

3) Finalmente, el **algoritmo inductor** utilizado tendrá que ajustarse a los estándares de la predicción numérica o categórica (por ejemplo, regresión lineal, logística, árbol de decisiones, *random forest*, etc.) y obtener un buen rendimiento en las pruebas de validación cruzada. Lo más recomendable es entrenar varios algoritmos y quedarse con el que proporcione mejores resultados, teniendo en cuenta también el delicado equilibrio entre la capacidad predictiva y la transparencia deseable en este tipo de algoritmos.

Debido a la gran complejidad del lenguaje humano, algoritmos como las redes neuronales profundas, que establecen relaciones entre variables (por ejemplo, palabras, posiciones en la frase, distancia respecto a otras palabras, etc.) a diferentes niveles de profundidad, han demostrado una capacidad predictiva muy superior a algoritmos más sencillos. El precio que hay que pagar es una parametrización complicada y una elevada opacidad en la interpretación de los resultados, lo que dificulta la fiscalización del rendimiento del algoritmo por parte del analista. Es importante que todos estos elementos estén sobre la mesa, conjuntamente con el caso de análisis particular, en el momento que el analista decida decantarse por una u otra técnica.

2.3. Aprendizaje no supervisado aplicado a textos

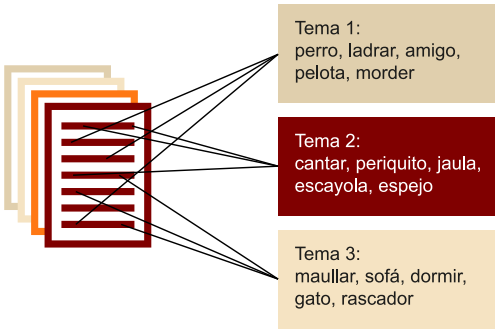
En cuanto a los modelos de aprendizaje no supervisado, hay que destacar, por un lado, que muchos modelos generales se pueden aplicar al análisis del corpus de los documentos. Por ejemplo, cada vez más analistas e investigadores aplican técnicas como el **análisis de redes** para identificar grupos de palabras en documentos (Gerlach y otros, 2018). Por otro lado, también hay algoritmos no supervisados propios del PLN, como, por ejemplo, las técnicas de **modelado temático** (*topic modelling*) basadas en la distribución probabilística de Dirichlet (Andrzejewski y otros, 2009).

Matriz de distancias

Una base de datos de doble entrada que proporciona una métrica sobre las relaciones entre sus elementos.

Para llevar a cabo este tipo de proceso, hay que procesar los textos mediante técnicas basadas en reglas heurísticas que permitan disponer de variables para aplicar los algoritmos no supervisados. Típicamente, lo que desearemos es generar una base de datos que se pueda interpretar como una matriz de distancias y, así, poder estudiar las relaciones entre las palabras y agruparlas en función de una serie de criterios. Los algoritmos de modelado temático identifican grupos de palabras que se utilizan de manera conjunta en documentos o en grupos de documentos. Por ejemplo, podemos utilizarlos para identificar los temas más importantes de una conversación o para descubrir grupos de mensajes parecidos emitidos por usuarios diferentes. Este tipo de algoritmos asumen que cada documento es una colección de temas, y que cada tema es una colección de palabras. De este modo, se identifican los temas en los conjuntos de documentos y se obtiene una cifra que cuantifica la vinculación de cada documento con cada tema (figura 14).

Figura 14. Modelado temático



	Tema		
Documento	1	2	3
1	0,2	1	0
2	0,4	0	0,2
3	1	0,2	0,9
4	0,7	0,5	0,8

Fuente: elaboración propia.

Actualmente, hay disponibles una gran variedad de algoritmos de modelado temático. Entre los más populares están la **asignación latente de Dirichlet** (LDA) o el **análisis semántico latente** (LSA o LSI). Se trata de algoritmos que proponen soluciones diferentes para problemas clasificatorios similares. Mientras que el modelo LDA solo agrupa las palabras utilizadas conjuntamente, el modelo LSA identifica aquellas que nunca se utilizan de manera conjunta. La elección de uno u otro algoritmo dependerá de una cantidad importante de factores que pueden alterar su resultado y que pueden ser complejos de controlar, como, por ejemplo, la longitud de los textos o su diversidad semántica y temática. Según las características del corpus documental, habrá que tomar

decisiones sobre el preprocesamiento (por ejemplo, lematizar sí o no, bolsa de palabras o TF-IDF) o incluso sobre el número óptimo de temas que se deben identificar.

La evaluación del **rendimiento de un modelado temático** se tiene que llevar a cabo considerando una serie de cuestiones diferentes, tanto de orden matemático como interpretativo y práctico. Métricas como la coherencia temática (una cifra entre el 0 y el 1 que cuantifica del grado de similitud semántica que hay entre las palabras más vinculadas a cada tema) o la probabilidad temática marginal (una cifra con una magnitud que varía según el procedimiento utilizado, que cuantifica la mayor o menor representación de un tema en un texto) pueden servir de guía para el analista y para ayudarlo a decidir qué procedimiento es más adecuado y qué resultado es mejor. Otros algoritmos, como, por ejemplo, el análisis de componentes principales o el algoritmo Louvain multinivel, también pueden ayudar a determinar el número óptimo de comunidades. Pero, en cualquier caso, hace falta que el analista interprete los resultados y los evalúe desde una perspectiva aplicada al caso de estudio, preguntándose si las comunidades obtenidas son fenomenológicamente relevantes: si aportan información importante e interesante sobre el fenómeno contenido en los documentos que se están analizando.

Finalmente, es importante entender que la definición de «tema» que elabora una máquina mediante un procedimiento no supervisado puede ser sustancialmente diferente al que una persona entienda como «tema». Para un ordenador, cualquier patrón es susceptible de ser un tema. Por ejemplo, un texto mal preprocesado a buen seguro que producirá resultados demasiado obvios o absurdos, como, por ejemplo, la agrupación de artículos y determinantes como palabras clave de un tema. Por esta razón, es importante entender muy bien los pasos que se llevan a cabo en este tipo de análisis, y también, el tipo de resultado que puede esperarse y cómo interpretarlos.

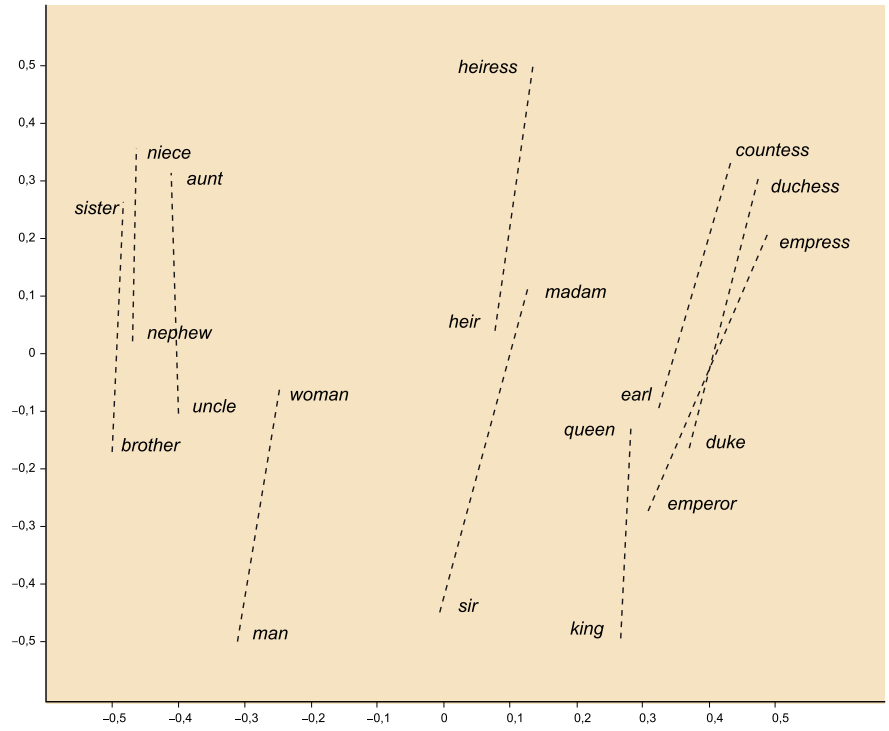
2.4. Algoritmos híbridos para el análisis de textos

Ya hemos podido ver que el análisis de textos mediante algoritmos de aprendizaje supervisado y no supervisado puede plantear problemas importantes. Por un lado, nos encontramos con el problema de que los algoritmos fáciles de interpretar y entender a menudo pueden resultar excesivamente sencillos para procesar la complejidad del lenguaje humano. Por otro lado, está la cuestión de que muchos algoritmos pueden ser complicados de ajustar, y aun así, es posible que ofrezcan resultados poco sorprendentes o interesantes desde la perspectiva de la creación de valor. Por todo esto, los algoritmos profundos y de conjunto (híbridos, que combinan una lógica supervisada y no supervisada) son enormemente populares en el PLN.

El tipo de técnicas más populares durante los últimos años han sido las **redes neuronales basadas en encajes de palabras** o *word embeddings* (Collobert y otros, 2008). Mediante estos procesos, podemos clasificar mensajes en función

de modelos preentrenados, capaces de identificar diferentes aspectos, como la temática o la emocionalidad. El proceso de encaje consiste en transformar una palabra en un número mediante un criterio predefinido (una red neuronal profunda previamente entrenada), que permite establecer relaciones matemáticas entre palabras del estilo «hermano-hermana = rey-reina» y, por lo tanto, establecer procesos de predicción mediante ecuaciones del estilo «hermano-hermana + rey = reina» (figura 15). La red neuronal Word2Vec (Mikolov y otros, 2013), desarrollada en el mismo corazón de Google, es el encaje de palabras más popular en la actualidad y es utilizado por una gran cantidad de software, incluido el mismo buscador de Google y su funcionalidad de texto predictivo.

Figura 15. Encajes de palabras vinculadas con el género



Fuente: adaptado de https://nlp.stanford.edu/projects/glove/.

Un analista que disponga de los conocimientos necesarios podrá entrenar una red neuronal profunda optimizando los encajes para la tarea particular de clasificación o predicción numérica que quiera llevar a cabo. Otra opción es utilizar encajes preentrenados, como, por ejemplo, GloVe o ELMo, que son redes neuronales profundas de código abierto entrenadas con textos de Twitter, de la Wikipedia o grandes compilaciones de webs con miles de millones de casos.

Las **redes neuronales preentrenadas** garantizan una precisión clasificatoria que, en la mayoría de los casos, mejoran sustancialmente los resultados de un algoritmo de aprendizaje supervisado, puesto que proporcionan una mayor contextualización y puntos de referencia para que los algoritmos establezcan predicciones.

El **proceso de predicción** consiste en incrustar cada palabra en los parámetros del modelo (por ejemplo, GloVe utiliza redes con hasta trescientos encajes) y establecer las correlaciones oportunas entre las palabras y los grupos de palabras en función de las puntuaciones obtenidas en cada uno de los encajes (tabla 1).

Tabla 1. Ejemplo de encajes de palabras con datos aleatorios

Palabra	Encaje				
	1	2	3	4	5
amigo	0,92	0,61	0,63	0,66	0,82
ladrar	0,74	0,38	0,33	0,20	0,90
cantar	0,35	0,74	0,27	0,57	0,57
dormir	0,02	0,63	0,24	0,31	0,35
alpiste	0,80	0,19	0,47	0,33	0,33

Fuente: elaboración propia.

Del mismo modo que hay redes neuronales profundas preentrenadas para facilitar encajes de palabras, también las hay que permiten el encaje de datos en otros formatos, como, por ejemplo, vídeos, audios e imágenes. Mediante esta técnica, cualquier formato de dato se puede transformar en una serie de vectores numéricos a partir de los cuales establecer criterios de similitud o algebraicos y, basándose en estos vectores, llevar a cabo operaciones de clusterización no supervisada, o de resolución de problemas de categorización o de predicción numérica típicos de los algoritmos supervisados.

La principal limitación de estas herramientas es, como ya se ha comentado, la imposibilidad de interpretar las relaciones entre las variables establecidas en una caja negra multidimensional que opera sobre distintas capas de conocimiento computacional autoadministrado y autogestionado.

3. El tratamiento justo de los datos

Operaciones y tareas cotidianas

Por ejemplo, cuando el móvil nos propone una ruta alternativa para llegar al trabajo para ahorrarnos la retención provocada por un accidente, o cuando YouTube nos propone contenidos que no hemos visto y que probablemente nos gustarán.

La inteligencia artificial y los algoritmos forman parte de nuestras vidas. Nos facilitan muchas de las operaciones y tareas que llevamos a cabo durante nuestro día a día. Estos algoritmos aprenden de nosotros, de nuestras preferencias y de nuestros patrones de consumo. También, como es lógico, aprenden de nuestros defectos, de nuestras fobias y de nuestros prejuicios. Todos estos elementos son absorbidos por los algoritmos y constituyen aquello que denominamos «**sesgo algorítmico**».

Un algoritmo puede incorporar este tipo de sesgos incluso cuando su creador no lo pretende y, por lo tanto, puede permanecer oculto e indetectable durante años, reproduciendo todos aquellos factores de desigualdad y todas las discriminaciones que ha aprendido y reforzado. Esto es lo que sucede cuando el algoritmo aprende de una serie de datos internamente correlacionados, sin que nadie haya evaluado estas correlaciones y las haya analizado desde un punto de vista normativo y prescriptivo.

Por ejemplo, si el algoritmo que está detrás de un portal de búsqueda de trabajo aprende de que la categoría «directivo» correlaciona con «hombre», y que la categoría «media jornada» correlaciona con «mujer», ofrecerá más trabajos de directivos a los hombres y más trabajos de media jornada a las mujeres. El algoritmo se convierte, de este modo, en un agente acrítico de reproducción social, que puede generar una gran cantidad de situaciones injustas y discriminatorias perfectamente evitables (por ejemplo, en procesos de selección de personal, decisiones financieras, evaluaciones de riesgo, adjudicaciones de subvenciones, etc.).

Los algoritmos no solamente pueden aprender de nuestros propios sesgos y de los sesgos contenidos en los datos, sino que son capaces de amplificarlos y magnificarlos, dando lugar a una serie de problemas de los que la comunidad de la minería de datos ya hace años que es consciente (DiFonzo, 2011).

Buenos ejemplos de ello son los problemas de clasificación sesgada y discriminatoria (por ejemplo, la discriminación hacia las mujeres, personas migrantes u otros colectivos en la publicación de ofertas de trabajo en los portales especializados de internet) o los problemas en la recomendación y prescripción sesgada de contenidos (por ejemplo, la propagación de noticias falsas en las redes sociales y la generación de estados globales de desinformación mediante la presentación de noticias que se ajustan a las creencias previas de los individuos). Desde la misma comunidad de la minería de datos se han propuesto diferentes tipos de soluciones para este tipo de problemas, tanto para la identificación o el descubrimiento de estos sesgos, como para la sinterización de algoritmos y procesos orientados a un tratamiento justo de los datos (Hajian y otros, 2016; Morales, 2020).

En cuanto a la identificación y el descubrimiento de sesgos algorítmicos, varios autores han propuesto **modelos de identificación de discriminaciones ocultas** en algoritmos basados en procesos heurísticos y en procedimientos de ingeniería inversa. No todos los algoritmos son igualmente accesibles y evaluables. Como ya hemos comentado antes, cuanto más opaco y poco transparente sea un algoritmo, más complicada será la interpretación de las relaciones establecidas entre las variables y también la detección de sus sesgos. Por eso, cada vez son más las voces que piden que se dejen de utilizar los algoritmos de caja negra, como, por ejemplo, las redes neuronales, en tareas de clasificación que pueden comportar elementos discriminatorios (Campolo y otros, 2017). En cuanto a la prevención de los sesgos y de sus efectos discriminatorios y socialmente indeseables, se han desarrollado tres tipos de estrategias diferentes, que actúan sobre los diferentes elementos comunes en varios algoritmos con finalidades clasificatorias o de recomendación de contenidos (Edizel y otros, 2020).

1) Intervenciones sobre los datos de entrenamiento. El primer tipo de medidas son aquellas que actúan sobre los datos a partir de los cuales se entrenará un algoritmo, a modo de operación de preprocesamiento de datos, como si se tratara de una operación de tokenización o de lematización.

2) Intervenciones sobre el algoritmo inductor. El segundo tipo de medidas son las que actúan sobre el procedimiento heurístico que se aplica sobre los datos introducidos, y son capaces de corregir, como mínimo, una parte de los sesgos durante la misma operación de procesamiento de los datos.

3) Intervenciones sobre los resultados del algoritmo. El último grupo de medidas son aquellas que actúan sobre el *output* del algoritmo: sobre la predicción, la clasificación o la recomendación. Se trata de operaciones de posprocesamiento de los datos que corrigen en ese momento los resultados discriminatorios de un algoritmo.

Para optar por uno u otro diseño preventivo deberemos hacerlo en función de una serie de cuestiones diferentes e interrelacionadas: la disponibilidad de datos históricos, el grado de conocimiento sobre las correlaciones que contienen, la transparencia del algoritmo inductor, etc.

El diseño de algoritmos conscientes de las discriminaciones estructurales que hay en nuestras sociedades y que contribuyan a mitigarlos es un campo de estudio en expansión y que avanza de la mano del mismo progreso social. Su desarrollo depende totalmente de la existencia de consensos sociales, de acuerdo con un código ético y normativo, como, por ejemplo, la igualdad entre hombres y mujeres, la protección de los derechos de los niños o la protección de las minorías y de los grupos sociales no dominantes. Los algoritmos pueden ser agentes de reproducción de las desigualdades presentes en una sociedad, pero también pueden servir para mitigar su efecto y sus consecuencias

indeseables. Precisamente por eso no se puede desligar el diseño de estos instrumentos de predicción y clasificación automatizada de un análisis social en clave normativa, sobre lo que es deseable e indeseable.

Bibliografía

Anderson, Chris (2008). «The end of theory: The data deluge makes the scientific method obsolete». *Wired Magazine* (vol. 16, núm. 7, págs. 16-07). Nueva York: Condé Nast Publications.

Andrzejewski, David; Zhu, Xiaojin; Craven, Mark (2009). «Incorporating domain knowledge into topic modeling via Dirichlet forest priors». En: A. Danyluk; L. Bottou; M. Littman. *Proceedings of the 26th annual international conference on machine learning* (págs. 25-32). Canadá: ICML.

Bird, Steven; Klein, Ewan; Loper, Eduard (2009). *Natural language processing with Python: analyzing text with the natural language toolkit*. Massachusetts: O'Reilly Media.

Campolo, Alex; Sanfilippo, Madelyn; Whittaker, Meredith; Crawford, Kate (2017). *AI now 2017 report*. Nueva York: AI Now Institute.

Collobert, Ronan; Weston, Jason; Bottou, Léon; Karlen, Michael; Kavukcuoglu, Koray; Kuksa, Pavel (2011). «Natural language processing (almost) from scratch». *Journal of Machine Learning Research* (vol. 12, págs. 2493-2537). New Jersey: AI Now Institute.

DiFonzo, Nicholas (2011). «The echo-chamber effect». *New York Times* (vol. 12, págs. 2493-2537). Nueva York: New York Times Company.

Edizel, Bora; Bonchi, Francesco; Hajian, Sara; Panisson, André; Tassa, Tamir (2020). «FaiRecSys: Mitigating algorithmic bias in recommender systems». *International Journal of Data Science and Analytics* (vol. 9, núm. 2, págs. 197-213). Nueva York: Springer.

Gerlach, Martin; Peixoto, Tiago P.; Altmann, Eduardo G. (2018). «A network approach to topic models». *Science Advances* (vol. 4, núm. 7). Pennsylvania: American Association for the Advancement of Science.

Hajian, Sara; Bonchi, Francesco; Castillo, Carlos (2016). «Algorithmic bias: From discrimination discovery to fairness-aware data mining». *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining* (págs. 2125-2126). Nueva York: Association for Computing Machinery.

Metz, Rachel (2016, agosto). «Why Microsoft Accidently Unleashed a Neo-Nazi Sexbot». *MIT Technology Review* [en línea]. Disponible en: <https://www.technologyreview.com/2016/03/24/161424/why-microsoft-accidentally-unleashed-a-neo-nazi-sexbot/>

Morales i Gras, Jordi (2020). «Cognitive Biases in Link Sharing Behavior and How to Get Rid of Them: Evidence from the 2019 Spanish General Election Twitter Conversation». *Social Media + Society* (vol. 6, núm. 2, págs. 1-4). Nueva York: SAGE Publications.

Mikolov, Tomas; Chen, Kai; Corrado, Greg; Dean, Jeffrey (2013). «Efficient estimation of word representations in vector space». *arXiv preprint arXiv* (págs. 1301-3781). Arizona: International Conference on Learning Representations.

Ray, Paramita; Chakrabarti, Amlan (2019). «A mixed approach of deep learning method and rule-based method to improve aspect level sentiment analysis». *Applied Computing and Informatics*. Amsterdam: Elsevier BV.

Rebala, Gopinath; Ravi, Ajay; Churiwala, Sanjay (2019). *An Introduction to Machine Learning*. Nueva York: Springer.