

Universidad Pontificia de Salamanca

Facultad de Informática

Grado en Ingeniería Informática

Trabajo Fin de Grado

**La Inteligencia Artificial aplicada**

**a la Inteligencia Emocional**

Jorge de Andrés González

Director:

Dr. D. Manuel Martín – Merino Acera

Salamanca, Junio de 2019

Resumen

El resumen del TFG debe contener entre 200 y 250 palabras.

Abstract

Traducción del resumen a inglés

Descriptores

Small Data, Data science, Inteligencia Artificial, Psicología, Psicopatología

Índice

[Introducción 1](#_Toc3201349)

[Presentación y Motivación del Trabajo 1](#_Toc3201350)

[Estado del Arte 2](#_Toc3201351)

[1. Capítulo 1: Fundamentos de Psicología 5](#_Toc3201352)

[1.1 La Psicología Conductista 6](#_Toc3201353)

[1.2 La Psicología Cognitiva 7](#_Toc3201354)

[1.3 La Psicología Cognitivo-Conductual 11](#_Toc3201355)

[1.4 Los Trastornos Psicológicos y la Psicopatología 13](#_Toc3201356)

[1.4.1 Los trastornos psicológicos 13](#_Toc3201357)

[1.4.1.1 Trastornos infantiles 14](#_Toc3201358)

[1.4.1.2 Trastornos de ansiedad 14](#_Toc3201359)

[1.4.1.2.1 Trastorno Obsesivo Compulsivo 15](#_Toc3201360)

[1.4.1.3 Trastornos del estado de ánimo 15](#_Toc3201361)

[1.4.1.3.1 Trastornos Depresivos 15](#_Toc3201362)

[1.4.1.4 Trastornos sexuales 15](#_Toc3201363)

[1.4.1.5 Trastornos de la conducta alimentaria 16](#_Toc3201364)

[1.4.1.6 Trastornos de la personalidad 16](#_Toc3201365)

[1.4.2 En este trabajo: Grupos de trastornos psicológicos 16](#_Toc3201366)

[1.4.3 Distorsiones de la percepción de la realidad: Las distorsiones cognitivas 17](#_Toc3201367)

[1.4.4 En este trabajo: Distorsiones cognitivas usadas 20](#_Toc3201368)

[1.4.4.1 En este trabajo: Otras variables usadas 21](#_Toc3201369)

[2. Capítulo 2: Data Science 23](#_Toc3201370)

[2.1 Mining Vs Machine Learning Vs Data Science Vs Big Data 23](#_Toc3201371)

[2.1.1 Data Mining 23](#_Toc3201372)

[2.1.2 Machine Learning 25](#_Toc3201373)

[2.1.3 Data Science 26](#_Toc3201374)

[2.1.4 Big Data 26](#_Toc3201375)

[2.2 Antes de hacer Data Mining 28](#_Toc3201376)

[2.2.1 En este trabajo: Obtención de los datos 29](#_Toc3201377)

[2.3 Data Mining 29](#_Toc3201378)

[2.3.1 Pasos previos y preparación de los datos 29](#_Toc3201379)

[2.3.2 En este trabajo: Preparación de los datos 35](#_Toc3201380)

[2.3.3 Análisis Exploratorio o Descriptivo 36](#_Toc3201381)

[2.3.3.1 Resumen de las estadísticas del Dataset 36](#_Toc3201382)

[2.3.3.2 OLAP y Análisis Multidimensional 39](#_Toc3201383)

[2.3.4 En este trabajo: Análisis Exploratorio 41](#_Toc3201384)

[2.3.5 Machine Learning 41](#_Toc3201385)

[2.3.5.1 ¿Cómo funciona un algoritmo de clasificación en machine learning? 41](#_Toc3201386)

[2.3.5.2 Problemas y soluciones con los clasificadores 42](#_Toc3201387)

[2.3.5.3 Algoritmos Supervisados 46](#_Toc3201388)

[2.3.5.3.1 K Nearest Neighbours (KNN) 46](#_Toc3201389)

[2.3.5.3.2 Árboles de Decisión 48](#_Toc3201390)

[2.3.5.3.3 Regresión Lineal 50](#_Toc3201391)

[2.3.5.3.4 Support Vector Machines (SVM) 50](#_Toc3201392)

[2.3.5.4 Algoritmos no Supervisados 52](#_Toc3201393)

[2.3.5.4.1 K Means 53](#_Toc3201394)

[2.3.5.4.2 Reglas de Asociación 57](#_Toc3201395)

[2.3.5.5 Algoritmos Semi-Supervisados 61](#_Toc3201396)

[2.3.5.6 Algoritmos de Aprendizaje por Refuerzo 61](#_Toc3201397)

[2.3.5.6.1 Q-Learning 62](#_Toc3201398)

[2.3.5.6.2 Diferencia Temporal (TD) 63](#_Toc3201399)

[2.3.5.7 Algoritmos de Redes Neuronales y Deep Learning 65](#_Toc3201400)

[2.3.5.7.1 Redes Neuronales 66](#_Toc3201401)

[2.3.5.7.2 Deep Learning 69](#_Toc3201402)

[2.3.6 Visualización 69](#_Toc3201403)

[3. Capítulo 3: Resultados Obtenidos y Conclusiones Finales 70](#_Toc3201404)

[3.1 Resultados Obtenidos 70](#_Toc3201405)

[3.2 Conclusiones 70](#_Toc3201406)

[3.3 Líneas Futuras, Ampliaciones y Entornos de Aplicación 70](#_Toc3201407)

[Anexo de Términos 72](#_Toc3201408)

[**Bibliografía 75**](#_Toc3201409)

[**Agradecimientos 77**](#_Toc3201410)

Índice de Figuras

[1‑1. Resumen Economía de Fichas 13](file:///D:\Escritorio\TFG\Proyecto\Memoria\Memoria%20TFG.docx#_Toc3202407)

[1‑2. Ejemplo Maximización 19](file:///D:\Escritorio\TFG\Proyecto\Memoria\Memoria%20TFG.docx#_Toc3202408)

[2‑1 Proceso KDD detallado 24](file:///D:\Escritorio\TFG\Proyecto\Memoria\Memoria%20TFG.docx#_Toc3202409)

[2‑2. Subdivisiones de la Inteligencia Artificial 25](file:///D:\Escritorio\TFG\Proyecto\Memoria\Memoria%20TFG.docx#_Toc3202410)

[2‑3. Las 3 V's del Big Data 28](file:///D:\Escritorio\TFG\Proyecto\Memoria\Memoria%20TFG.docx#_Toc3202411)

[2‑4. Reducción Dimensionalidad Dataset Iris 32](file:///D:\Escritorio\TFG\Proyecto\Memoria\Memoria%20TFG.docx#_Toc3202412)

[2‑5. Percentiles sobre una normal 37](file:///D:\Escritorio\TFG\Proyecto\Memoria\Memoria%20TFG.docx#_Toc3202413)

[2‑6. Ejemplo Data Cube 40](file:///D:\Escritorio\TFG\Proyecto\Memoria\Memoria%20TFG.docx#_Toc3202414)

[2‑7. Underfitting, Óptimo y Overfitting 43](file:///D:\Escritorio\TFG\Proyecto\Memoria\Memoria%20TFG.docx#_Toc3202415)

[2‑8. Ejemplo 5-fold Cross Validation 44](file:///D:\Escritorio\TFG\Proyecto\Memoria\Memoria%20TFG.docx#_Toc3202416)

[2‑9. KNN 47](file:///D:\Escritorio\TFG\Proyecto\Memoria\Memoria%20TFG.docx#_Toc3202417)

[2‑10. Estructura básica de un árbol de decisión 49](file:///D:\Escritorio\TFG\Proyecto\Memoria\Memoria%20TFG.docx#_Toc3202418)

[2‑11. SVM Linealmente Separable 50](file:///D:\Escritorio\TFG\Proyecto\Memoria\Memoria%20TFG.docx#_Toc3202419)

[2‑12. SVM No Lineal 52](file:///D:\Escritorio\TFG\Proyecto\Memoria\Memoria%20TFG.docx#_Toc3202420)

[2‑13. K-Means paso a paso 53](file:///D:\Escritorio\TFG\Proyecto\Memoria\Memoria%20TFG.docx#_Toc3202421)

[2‑14. Ejemplo en Inglés de tabla de reglas 59](file:///D:\Escritorio\TFG\Proyecto\Memoria\Memoria%20TFG.docx#_Toc3202422)

[2‑15. Ejemplo de Tabla Q 62](file:///D:\Escritorio\TFG\Proyecto\Memoria\Memoria%20TFG.docx#_Toc3202423)

[2‑16. Red Neuronal Multicapa Feed Forward 67](file:///D:\Escritorio\TFG\Proyecto\Memoria\Memoria%20TFG.docx#_Toc3202424)

Índice de Tablas

[Tabla 3‑1. Ejemplos Reglas Finales 57](#_Toc3202427)

# Introducción

## Presentación y Motivación del Trabajo

El presente trabajo versa sobre la predicción de trastornos psicológicos mediante el uso de data science, y más concretamente, técnicas de inteligencia artificial. Para conseguir este objetivo, se hará un análisis exhaustivo tanto de las diferentes distorsiones cognitivas que producen los trastornos psicológicos, como de los trastornos psicológicos en sí, y posteriormente otro análisis de las diferentes técnicas y pasos del proceso de data science que hay que llevar a cabo para conseguir unos resultados óptimos.

Es interesante este problema debido a la resolución de un problema real, puesto que todos los psicólogos al inicio de sus terapias primero deben de identificar las diferentes distorsiones cognitivas que posee el paciente para poder entonces valorar el trastorno psicológico que sufren. El tratamiento de la inteligencia emocional mediante técnicas de inteligencia artificial es extremadamente interesante, puesto que ambas inteligencias poseen numerosas características en común, como la capacidad de aprendizaje, y el uso de una para determinar los problemas de la otra resulta en un problema interdisciplinar muy atrayente.

Se podría decir que es un problema altamente complicado debido a la existencia de datos reales en muy pequeña cantidad, por lo que las cantidades de ruido y outliers principalmente se plantean como dos impedimentos de gran magnitud. Además, la pequeña cantidad de pacientes que se han podido recoger a mano plantea una dificultad extra hacia los modelos, puesto que prácticamente hay solo 3 veces más de pacientes que dimensiones tiene el problema.

También, es un problema motivante debido a la actualidad de ambos temas, puesto que la inteligencia artificial está presente en todo nuestro entorno, desde los móviles hasta internet, pasando por televisores, sistemas recomendadores o incluso la mayoría de procesos de ingeniería. También está en auge el uso de la misma para la transformación digital de los negocios, como las data driven enterprises. Por otra parte, la psicología es un tema que lleva en auge numerosos años, pero cada vez en mayor medida debido al aumento de los trastornos psicológicos en nuestra sociedad debido al ritmo de vida de los adultos y los cambios en educación hacia los niños.

En resumidas cuentas, debido a todo esto, creo que el problema que se plantea es un reto importante debido a las dificultades técnicas del mismo, pero a la vez muy interesante dados los temas, su actualidad y la conexión que se va a establecer entre ellos.

## Estado del Arte

El objetivo del trabajo es la construcción de un sistema automático de predicción de trastornos psicológicos en base a una serie de datos de pacientes reales, obtenidos a mano a partir de una consulta privada profesional de psicología. Este problema, tras una larga consulta en bibliografía, no ha sido encontrado, por lo que se puede afirmar que es el primero en abordar este problema de esta manera.

Junto con este primer objetivo de la construcción del sistema, también se hará un análisis exhaustivo de los datos para su correcta interpretación, además de la obtención de características interesantes de los mismos que puedan estar ocultas.

Posteriormente, se hará una visualización de los datos de tal forma que se puedan dejar claras ciertas características mediante la utilización de software profesional de visualización, como Tableau o similares.

La metodología que se seguirá será, en primera instancia, la obtención de los datos a mano, debido a la falta de repositorios con datos reales de los mismos.

Posteriormente, se hará un estudio exhaustivo de los diferentes métodos y técnicas que se pueden utilizar para el análisis, clasificación y predicción de los datos, dejando estos conocimientos plasmados en la memoria y posteriormente implementados en código.

Finalmente, se hará un estudio de las diversas técnicas de visualización de los datos, y se procederá a la visualización de características interesantes de los mismos mediante herramientas profesionales de visualización.

Como elemento global, para el control de los errores principalmente, usaré un sistema de control de tareas y errores muy similar a Kanban llamado Glo, ofrecido gratuitamente por GitKraken. Además, todo el código y la memoria estará continuamente siendo versionado mediante un VCS (GitHub en este caso) para el control del mismo y de los cambios, así como la sincronización entre los ordenadores en los que se pueda trabajar y como medida preventiva de “Backup”, puesto que durante el desarrollo este proyecto estará como repositorio privado.

Analizando los resultados esperados, es importante tener en cuenta de que este no es un “problema de laboratorio” con unos datos controlados, sino que es un problema real con datos reales que incluso algunos psicólogos profesionales sin una preparación óptima tienen problemas para resolver. De este modo, espero que los resultados no estén cerca del 100% de acierto, aunque bien es cierto de que, ante las variables escogidas cautelosamente, y el tratamiento de los datos, espero obtener porcentajes de acierto que estén alrededor del 75% - 80% en los algoritmos que más se adapten al problema.

Atendiendo a la organización de la memoria, esta comenzará con un primer capítulo introductorio de psicología. En él, se analizará el término de psicología en sí y las dos vertientes que han tenido que existir para llegar al enfoque actual y desde el cual se analizan los trastornos psicológicos, el cognitivo-conductual, el cual también será explicado.

Posteriormente, habrá un extenso capítulo sobre data science, en el cual se abordarán de una manera muy pormenorizada todos los pasos que se deben de tener en cuenta para un análisis de datos completo y fiable, haciendo especial hincapié en la zona de algoritmos de machine learning, que consistirá en el núcleo del trabajo. También habrá un apartado de visualización.

En el siguiente capítulo, se hará un análisis de todos los resultados obtenidos mediante las diferentes técnicas, de tal manera que se pueda comparar la eficacia de los algoritmos entre sí, e incluso de un mismo algoritmo con diferentes parámetros.

Finalmente, habrá un anexo de términos que puedan resultar desconocidos y que no se hayan explicado en profundidad, y una bibliografía del trabajo.

# Capítulo 1: Fundamentos de Psicología

Empezando desde el principio nos podemos preguntar: ¿Qué es la psicología? Y no es una respuesta fácil de dar, debido a que el campo que abarca la psicología es muy amplio y profundo. La mayoría de expertos darían una definición cercana a “Es la ciencia que investiga y trata la conducta y los procesos que se llevan a cabo en la mente”, pero esta definición no es suficientemente concreta, aunque puede ser válida.

La psicología trata numerosos ámbitos, desde explicar cómo percibimos información y como la procesamos, hasta como nos relacionamos con otras personas en las diferentes situaciones que se pueden dar en nuestra vida. También, la psicología da cabida y respuesta a todas aquellas distorsiones mentales que pueden darse y que derivan en distorsiones emocionales, que se reúnen bajo el nombre de psicopatologías.

La psicología tiene tres vías principales de investigación, que son las siguientes:

1. Psicología del desarrollo

La psicología del desarrollo trata de estudiar el desarrollo humano desde la niñez hasta la vejez, teniendo en cuenta todos los factores ambientales, culturales e individuales de cada persona.

1. Psicología fisiológica y la neurociencia

Esta rama de la psicología investiga las bases del comportamiento humano a partir de los efectos que producen elementos naturales en nuestro cerebro, actuando como repartidores de información. Estos mensajeros son principalmente hormonas.

También, esta rama estudia toda la farmacología que pueda estar relacionada con la mente humana, como pueden ser los medicamentos psicoactivos (como los recetados contra la depresión o los calmantes) o las llamadas drogas sociales (alcohol, tabaco o marihuana principalmente).

1. Psicología experimental

La psicología experimental se centra en responder una serie de cuestiones relacionadas con el aprendizaje humano, la memoria y las emociones entre otros elementos. Algunas de las preguntas que intentan responder son:

* ¿Qué es lo que nos hace olvidar cosas?
* ¿Las emociones son universales o son personalizadas?
* ¿La cultura influye en las emociones?

De esta manera, podemos ver que la psicología es una ciencia activa y con numerosas ramas, pero debido a la naturaleza del trabajo, a partir de ahora sólo me centraré en la primera de ellas, la rama de la psicología del desarrollo.

Dentro de la psicología del desarrollo, al igual que con numerosas ciencias a lo largo de la historia, ha habido diferentes convicciones sobre lo que es la verdad. A continuación, veremos las tres ramas más importantes de la historia de la psicología, lo que enunciaban y cuáles eran sus puntos fuertes y débiles.

## La Psicología Conductista

En psicología, la rama del conductismo es aquella que estima que el estudio que debe hacer la psicología debe ser sobre únicamente los comportamientos observables, y los efectos que estos puedan tener sobre estos comportamientos los estímulos que rodean a la persona.

El conductismo nació de la mano de John Broadus Watson (1878-1958). Watson, en la entrevista que se considera el inicio del conductismo (1913), afirma que la psicología debería de convertirse en una rama totalmente científica, y que para ello lo que debería de hacer es centrarse en el análisis de las conductas totalmente visibles de las personas, en vez de divagar entre estados mentales y la diferencia de conceptos como conciencia o mente.

Para Watson, así como para toda la vertiente conductivista, los seres humanos somos “cajas negras” cuyo interior nunca es observable, y cada estímulo que llega es procesado de una manera desconocida, obteniendo finalmente una respuesta por parte de la persona. Watson sostiene que, al ser este procesamiento inobservable, no debe de ser estudiado ni tenido en cuenta.

Esta es una posición muy radical dentro de la psicología, y como no podía ser de otra manera, otros psicólogos conductivistas fueron matizando posteriormente estas afirmaciones, aseverando que los procesos que tenían lugar dentro del cuerpo sí tenían una gran importancia, pero que la psicología no tenía que tenerlos en cuenta para poder tener explicaciones sobre la conducta humana.

Uno de los elementos más importantes del conductismo es su oposición al concepto de “enfermedad mental”. Es decir, según las raíces de esta vertiente, no pueden existir conductas patológicas, ya que estas conductas que tiene un ser humano siempre han de valorarse respecto a la adecuación de las mismas a un contexto. Así, los conductistas sostienen que las enfermedades deben de ser patologías bien aisladas y definidas.

Esto nos lleva a que los psicólogos conductistas se opongan frontalmente al uso de fármacos para poder tratar algunos problemas psicológicos como las fobias.

Algunos de los elementos básicos del conductismo son:

* Estímulo
* Respuesta
* Condicionamiento
* Refuerzo
* Castigo

Los explicaré brevemente a continuación:

El estímulo es cualquier señal, elemento o mensaje que produce una reacción, conocida como respuesta, en un organismo. En ese momento, al generar la respuesta, automáticamente tenemos un condicionamiento, que consiste en un aprendizaje que se deriva de la asociación entre estímulos y respuestas.

Una vez que la respuesta ha sido dada, acorde con nuestro condicionamiento, podemos entrenarnos para obtener diferentes respuestas las próximas veces. Esto lo haremos mediante refuerzos y castigos.

Los refuerzos son premios, actitudes, o cualquier elemento que nos invita a seguir manteniendo una cierta conducta al recibir un estímulo.

Los castigos son la oposición a los refuerzos. Como su propio nombre indica, consiste en cualquier elemento o acción que nos invita a no seguir manteniendo la respuesta obtenida ante un estímulo.

El conductismo empezó a entrar en declive a partir de los años 50, cuando surgió el cognitivismo y, como ya he comentado, se suavizaron las teorías.

El cognitivismo surgió como un modelo puramente teórico, y fue una reacción frontal al análisis de sólo las conductas observables del conductismo, dejando aparte la cognición de las situaciones. Este cambio es conocido como la “revolución cognitiva”. Esta revolución, entre otras cosas, surgió por un conjunto de anomalías empíricas en el conductismo que dieron lugar a una gran deceleración en diversas líneas de investigación y desarrollo.

## La Psicología Cognitiva

La psicología cognitiva es una rama de la psicología, encargada de estudiar, tal como su nombre dice, la cognición. Entendemos como cognición el conjunto de procesos mentales que están implicados en la obtención del conocimiento al ser humano y que, por lo tanto, no son observables. Podríamos decir que la psicología cognitiva es esa pieza de la mitad del puzle que le faltaba a la psicología conductual.

Como ya he expuesto anteriormente, entendemos el inicio de la psicología cognitiva en la década de 1950 en Estados Unidos aproximadamente, donde había una serie de teorías del aprendizaje y un sistema de psicología conductista que no acababa de cuadrar a la sociedad, y que se quedaba estancado en las experimentaciones.

Con la psicología cognitiva obtenemos el concepto de “representación mental”, pieza clave de la psicología cognitiva debido a su carácter central y a la posibilidad de poder operar con ellas. Estas representaciones mentales tienen que ser analizadas aparte, pero esto, para los cognitivistas, no es excusa para no tenerlas en cuenta a la hora de analizar el comportamiento humano. Además, todo esto coincide con una disminución de la importancia del contexto, sea afectos, cultura o historia, lo que centra aún más en el interior a esta vertiente psicológica. Es importante tener en cuenta que, al contrario que la conductual, la cognitiva no cierra la puerta de inicio a los factores externos, pero sí es verdad que los considera una parte secundaria de la psicología humana. Según esta vertiente, la investigación psicológica se facilita enormemente.

Para la investigación cognitiva hubo varios avances tecnológicos que facilitaron la supremacía de esta teoría a partir de los años 50:

1. Los avances en informática y cálculo:

Personas como Alan Turing (1912 - 1954) tuvieron mucho que ver indirectamente con los avances en psicología, porque las máquinas que crearon eran programables. Esto significa que estas máquinas pueden seguir una serie de pasos y finalmente tomar decisiones, tal como los seres humanos. Por ello, para el estudio del pensamiento humano estas máquinas tuvieron una importancia capital.

1. Los avances en cibernética

En cibernética, podemos destacar a Norbert Wiener (1894 – 1964), quien construyó servomecanismos. Estos elementos son aparatos que son capaces de mantener un cierto rumbo dependiendo sólo de factores externos. Hacían cálculos de variaciones del exterior y, mediante un sistema de retroalimentación, podían calcular internamente los cambios a hacer y ejecutarlos, funcionando de una manera similar a la teoría cognitiva.

1. Los avances en la teoría de la información

En este ámbito destacó mucho Claude E. Shannon (1916 – 2001), que hizo grandes aportes a este ámbito. Shannon afirmaba que la información no era más que una poda de las diferentes alternativas mediante elecciones, de una forma totalmente separada a los contenidos concretos que la forman.

Es interesante que la unidad básica para Shannon es el bit, ya que según su teoría la información se construye a partir de dos alternativas posibles.

De esta manera, y tal como he comentado anteriormente, los elementos sobre los que se apoya la psicología cognitiva tienen que ser elementos que estén principalmente en el interior de la persona, y que no sean tangibles ni observables. Por ello, y a la vista de los diferentes experimentos que ayudaron al cognitivismo a seguir adelante, podemos definir dos elementos como base de esta vertiente:

1. La representación de la información

Una definición simple pero acertada de representación es la que nos da Jean Matter Mandler (1929 - ) afirmando que “la representación es información almacenada por un sistema mental y dispuesta para ser utilizada por ese sistema”. No es una definición aceptada unánimemente, pero es simple y lo suficientemente precisa para usarla durante este trabajo.

Así, según esta autora, representación y conocimiento son dos conceptos realmente unidos entre sí, aunque enfatiza especialmente en que la representación es el formato en el cual se almacena el conocimiento. Es importante enfatizar que, al igual que en los ordenadores, para que haya una representación de la información primero hay que procesarla y hacer una serie de transformaciones. Pero, a partir de este punto, lo que puede pasar con la información puede seguir múltiples caminos:

Puede ser de diferentes tipos: Implícito o explícito, proposiciones o imágenes…

Puede ser de diferente nivel de abstracción: La información se puede representar de una manera muy simple y cercana a nuestra percepción visual (como ocurre a la hora de aprender las letras del abecedario), o de una manera muy compleja y elaborada, como ocurriría a la hora de razonar y memorizar las diferentes vías de resolución de un problema.

1. El procesamiento de la información

Este pilar de la psicología cognitiva quizás es el más importante, ya que las corrientes de estudio de esta rama llegaron a abordar casi por completo el estudio de la psicología cognitiva.

En este procesamiento de la información no hay una teoría unificada y aceptada por todos, ni siquiera por una mayoría, tal como puede suceder con otras teorías. En cambio, la teoría del procesamiento de la información está conformada por un conjunto de teorías muy diversas. Pero, a pesar de esto, todas estas teorías comparten una base común y unas características generales. Estos elementos comunes son:

1. Los fenómenos cognitivos en los seres humanos son bastante parecidos a los fenómenos que regulan el funcionamiento de los ordenadores:

Esto es lo que se ha llamado la “metáfora del ordenador”, e indica que la forma en la que las personas procesan la información es muy similar a la forma en la que un ordenador la procesa. Esto se puede ver en diferentes conceptos, como:

1. Ambos tienen que hacer conversiones a un lenguaje que entienden:

En el caso del ser humano, tenemos que configurar las representaciones mentales anteriormente descritas, mientras que en el ordenador se traduce a lenguaje máquina, es decir, lenguaje binario.

1. Ambos tienen que actuar sobre la información ya trasformada en el paso anterior:

En el caso del ordenador, el procesador ejecuta las acciones que le llegan en lenguaje binario, mientras que en el caso del ser humano la mente ejecuta acciones a partir de estructuras conceptuales en la mente.

1. Ambos dan respuestas hacia el exterior a través de elementos que están fabricados para tal uso:

En el caso del ordenador, emite una respuesta a través de los denominados periféricos de salida (pantalla, impresora…) mientras que en el ser humano podemos hacerlo mediante la voz, o el movimiento.

1. Una cantidad ciertamente pequeña de procesos básicos subyace a toda la cognición:

Podríamos decir que la actividad cognitiva que procesa la información entre la llegada del estímulo y la emisión de la respuesta se puede subdividir en elementos más básicos, que pueden ser subdivididos a su vez. Así, podemos simplificar el problema de la cognición a axiomas y componentes fundamentales cognitivos.

Esto, en los ordenadores, lo podemos ver en las arquitecturas RISC en los procesadores, donde un conjunto reducido de instrucciones es capaz de hacer todas las funciones de un procesador, ante una arquitectura CISC que tiene más instrucciones.

Por parte de la psicología, todavía no se ha llegado a una conclusión y aceptación de cuáles son estos elementos fundamentales de la cognición, pero sí se está de acuerdo en que esta subdivisión a elementos más simples es posible.

1. Los procesos que son individuales pueden cooperar y ejecutarse de manera organizada:

En el caso de la informática, el procesador puede hacer un número de instrucciones muy reducido a la vez (dependiendo principalmente del número de núcleos que tenga, o de tecnologías como el Hyper-Threading de Intel). Estas operaciones de poco sirven si no se juntan entre ellas para formar operaciones más complejas, y con ello finalmente formar rutinas y programas de muy alto nivel.

En psicología pasa igual. La comprensión de elementos fundamentales en la cognición, y la puesta en conjunto de ellos es lo que hace que hace que ejecutemos una determinada acción como humanos. En este caso, las relaciones y el orden en el que percibamos los estímulos es vital, ya que hay percepciones más importantes que otras, debido a que muchas veces se pueden clasificar de forma jerárquica.

1. El procesamiento tiene supuestamente limitaciones:

En el caso de la informática, como he comentado anteriormente, el número de operaciones que se pueden realizar al mismo tiempo es limitado. De esto, podemos inferir que hay un máximo en el número de tareas que podemos procesar en una unidad de tiempo. También tenemos que tener en cuenta de que, por cada golpe de reloj del procesador podemos procesar un bit por cada núcleo, lo cual no significa que completemos una cierta tarea, que puede llevar un cierto tiempo.

Y es más, hay tareas que necesitan procesarse de una forma secuencial (por ejemplo, para poder tener un dato en memoria que le da una tarea anterior), de tal manera que pueden tener que esperar a la finalización de otra para poder ejecutarse.

En psicología, respecto a la mente humana, hay diferentes tareas que podemos tener que llevar a cabo. Cada una de estas tareas demandan un “procesamiento" variable en nuestra mente, consumiendo una cantidad también variable de recursos. Así, el ser humano tiene la capacidad de ordenar las tareas en “automáticas” y “con esfuerzo”, dependiendo de la cantidad de recursos que consuman.

Además, la mente humana al igual que un procesador de ordenador puede procesar diversas tareas de forma simultánea si no necesitan de otra y consumen pocos recursos, mientras que si esto no es así puede hacerlo secuencialmente.

## La Psicología Cognitivo-Conductual

La psicología cognitivo-conductual es aquella que junta las bases de la teoría cognitiva de la psicología, y de su anterior vertiente conductual. Nace de 5 hechos primordiales:

1. El condicionamiento clásico

Investigado por el filósofo ruso Ivan Pavlov (1849 – 1936), se basa en que los individuos pueden relacionarse de una manera predictiva entre los diferentes estímulos que plantea el ambiente.

En el experimento de Pavlov, se sabía que los perros al darles comida generaban una respuesta en forma de salivar. Para conducir el experimento, Pavlov empezó a tocar una campana antes de dar comida al perro, de tal manera que el estímulo de la campana acabó haciendo salivar al perro sin llegar a ver la comida, es decir, un estímulo neutro que nada tenía que ver con la comida acabó produciendo la respuesta en el perro.

1. El condicionamiento operante

Investigado por Burrhus Frederick Skinner (1904 – 1990), el condicionamiento operante se basa en el hecho de que las conductas del ser humano se pueden adquirir, se pueden mantener y se pueden extinguir. Así, el ser humano asocia comportamientos con consecuencias.

Este condicionamiento operante tiene uno de los pilares en la teoría de la economía de fichas, que explicaré más adelante.

1. El aprendizaje social u observacional

Investigado por un grupo liderado por Albert Bandura (1925 - ), la teoría del aprendizaje social conjunta una serie de hipótesis mediante las cuales se afirma que el aprendizaje no solo viene de la experiencia de la propia persona, sino también de la información que puede recibir la persona mediante estímulos auditivos o visuales entre otros.

Se lleva a cabo a través de dos elementos:

* 1. Moldeamiento

Consiste en el proceso de observar e imitar un comportamiento en concreto que hemos visto en otra persona.

Un ejemplo de esto es el niño que ve a su padre ponerse la corbata, y quiere imitarle poniéndose una.

* 1. Neuronas Espejo

Las neuronas espejo son un conjunto de neuronas que, cuando se observa a una persona realizando una acción, emiten una serie de descargas eléctricas que impulsan a la persona a repetir la acción.

Un ejemplo de esto se da con los recién nacidos, con los que la acción de sacarles la lengua es imitada por ellos.

1. El trabajo de Beck y Ellis

Aaron Temkin Beck (1921 - ) y Albert Ellis (1913 – 2007) usaron los tres principios anteriormente explicados del condicionamiento clásico, condicionamiento operante y aprendizaje social para crear el enfoque cognitivo-conductual de hoy en día.

1. La visión incompleta de las dos teorías anteriores

Tal como hemos visto anteriormente, las dos ramas (cognitiva y conductual) son un puzle incompleto. Repasando, la conductual sería el principio y el final del puzle, obviando el centro, mientras que la cognitiva sería el núcleo del puzle, sin tener demasiado en cuenta el inicio y el fin. Por ello, la unión de las dos teorías da una visión mucho más completa del individuo y de su comportamiento.

Uno de los elementos más importantes que se heredan en la psicología cognitivo-conductual de la psicología cognitiva es el llamado “aprendizaje por economía de fichas”. En nuestra vida, toda acción conlleva una reacción. Por ejemplo, si alguien roba, se le multa para que obtenga un castigo y deje de hacerlo. Si alguien trabaja y es responsable en la empresa, seguirá cobrando y es posible que obtenga un ascenso para premiar su dinámica positiva. Esto es la base de la economía de fichas, los llamados “refuerzo” y “castigo”.

Así, una persona recibirá un refuerzo cuando tras hacer una acción es premiada por ello, lo que se denomina “refuerzo positivo”. Con esto, la persona tenderá a repetir más la conducta. También, un refuerzo consiste en la evitación de un castigo tras hacer una acción. En este caso, el hecho de no recibir algo desagradable es algo que nos impulsará a repetir esa acción en el futuro, y recibe el nombre de “refuerzo negativo”.

Por la otra parte, una persona recibirá un castigo si tras hacer una acción recibe algo desagradable. Con esto, la persona tenderá a extinguir la existencia de dicha conducta. También, se puede considerar castigo el hecho de que una persona se quede sin algo agradable tras una acción, lo cual también impulsará a no repetir la acción en situaciones futuras.

Lo podemos ver resumido en la siguiente figura:

1‑1. Resumen Economía de Fichas

Otro elemento muy importante de la psicología cognitivo-conductual, basado en el punto 5 anterior, es la sucesión siguiente, conocida como “registro cognitivo-conductual”:

1. Situación
2. Pensamiento
3. Emoción
4. Conducta

Este registro, si nos fijamos, en los puntos 1 y 4 tenemos la vertiente conductual, y en los puntos 2 y 3 la cognitiva.

Este esquema es fundamental, ya que todos seguimos esta secuencia a la hora de actuar. Primeramente, nos encontramos en una situación en la vida, pongamos que estamos en un restaurante abarrotado donde no hay aire acondicionado. Lo primero que hacemos es pensar, y un pensamiento posible ante esta situación sería algo como: “Me estoy agobiando, creo que me voy a desmayar”. En este momento, nuestro cuerpo sufre una serie de emociones, o de reacciones físicas, que en nuestro caso sería un aumento de la cadencia de respiración y un gran agobio. Finalmente, actuamos en consecuencia, lo que se ve en la conducta, como podría ser desarrollar un malestar cada vez que vemos un lugar con mucha gente y evitar entrar a toda costa.

El problema de que las personas no controlen esta secuencia es que desarrollarán una serie de pensamientos automáticos (no controlados e instantáneos) a partir de ciertos estímulos, que mayoritariamente irán con una carga emocional, y que es posible que la reacción conductual a estos estímulos sea irracional.

## Los Trastornos Psicológicos y la Psicopatología

### Los trastornos psicológicos

Los trastornos psicológicos son una enfermedad mental que puede tener muy distintas manifestaciones hacia el exterior. Normalmente, se caracterizan por una serie de distorsiones cognitivas (que veremos posteriormente), y unas alteraciones severas en la conducta y las relaciones con el resto de personas.

Para la cura de trastornos psicológicos, es muy importante el apoyo social que puedan tener estas personas, pero más importante si cabe es que acudan a un profesional cualificado para poder recibir un tratamiento especializado que necesiten. Este tratamiento puede dárselo el psicólogo, el psiquiatra o entre los dos, dependiendo del caso.

En los siguientes apartados, haré un análisis pormenorizado de los grupos más importantes de trastornos psicológicos que están aceptados en la actualidad. Existen también otros grupos, pero al ser más secundarios evitaré su explicación.

#### Trastornos infantiles

Los trastornos infantiles se definen como aquellos que se pueden diagnosticar por primera vez en la infancia o adolescencia de la persona, aunque no es una regla fija, sino más bien difusa. Es interesante destacar de estos trastornos que, aunque puede que se den en etapas tempranas de la vida de la persona, no sean diagnosticados hasta la edad adulta.

Es importante destacar que estos trastornos están encuadrados en esta sección por conveniencia, y no se debe desdeñar el hecho de que el paciente pueda estar también encuadrado en otro trastorno que pueda pertenecer a otro grupo.

Algunos de los subgrupos más comunes en el grupo de trastornos infantiles son:

* Retraso mental
* Trastornos del aprendizaje (lectura, cálculo…)
* Trastornos de las habilidades motoras (coordinación)
* Trastornos de la comunicación (expresión, fonología)
* Trastornos del desarrollo (autismo, Asperger…)
* Trastornos por déficit de atención (TDAH)
* Trastornos TICS (Tourette)
* Otros

#### Trastornos de ansiedad

Los trastornos de ansiedad son aquellos que se pueden dar bajo un contexto de angustia o agorafobia.

Un trastorno por crisis angustiosa es aquel en el que, repentinamente, la persona sufre un ataque de pánico, dándose algunos síntomas como palpitaciones, sudoración o ahogo.

Un trastorno por agorafobia es aquel en el que la persona sufre una aparición de angustia en un lugar donde la escapada es complicada, debido a que no dispone de ayuda o que se encuentra en un lugar público o embarazoso.

Es interesante que, en la combinación o falta de presencia de estos dos grupos podemos obtener diferentes trastornos y fobias, como la social.

Mención aparte merece el trastorno obsesivo-compulsivo (TOC), ya que es tan común que, como se verá en el apartado 2.4.2, lo usaremos como un grupo propio.

##### Trastorno Obsesivo Compulsivo

El trastorno obsesivo compulsivo consiste en un conjunto de pensamientos o impulsos que le aparecen al enfermo de una forma continua, y causan un malestar muy significativo en la persona. Estos pensamientos, llamados obsesiones, no tienen por qué ser ni siquiera de problemas de la vida real, ya que pueden ser irracionales.

En este caso, la persona sabe que estos pensamientos vienen de su mente, y los intenta ignorar de una forma incorrecta.

También, en este caso tenemos las compulsiones, definidas como comportamientos repetitivos que hace una persona para contrarrestar o contestar a una obsesión que posee, siguiendo unas estrictas reglas inventadas por la propia persona. Con esto, la persona busca reducir la obsesión y, por lo tanto, el malestar.

#### Trastornos del estado de ánimo

Los trastornos del estado de ánimo son aquellos que se dan tras un episodio afectivo, que sirve como fundamento al trastorno.

En este grupo, hay dos grupos de trastornos principalmente: Los trastornos bipolares y en el que nos vamos a centrar especialmente a continuación, debido a que es tan común que también lo usaremos como grupo general en sí, llamado trastornos depresivos.

##### Trastornos Depresivos

Los trastornos depresivos son aquellos en los que la persona ha sufrido uno o varios episodios afectivos depresivos anteriormente, y consisten en una alteración del estado de ánimo muy severa, similar a la tristeza pero en mayores dimensiones.

Algunas de los elementos más importantes que lo indican son:

* Tristeza muy severa y continua
* Desmotivación por los objetivos
* Gran irritabilidad
* Puede llegar incluso a ideas suicidas

Un caso muy concreto y a la vez vistoso de estos trastornos depresivos es el trastorno distímico, que simplificando consiste en un trastorno depresivo que solo se da ciertos días y a ciertas horas del día de forma crónica.

#### Trastornos sexuales

Los trastornos sexuales, como su propio nombre indica, son aquellos que psicológicamente conllevan alteraciones en cualquier ámbito relacionado con la sexualidad humana.

Se pueden dividir en diversos grupos, como son:

* Trastornos del deseo sexual (Deseo hipoactivo, trastornos en la excitación, dolores…)
* Parafilias (Fetichismo, pedofilia, masoquismo…)
* Trastornos de la identidad sexual (Transgénero, inadecuación con ningún sexo…)

#### Trastornos de la conducta alimentaria

Los trastornos de la conducta alimentaria son aquellos que conllevan, por un trastorno psicológico, un cambio en la alimentación de la persona, provocando reacciones también físicas en su cuerpo.

Existen diversos trastornos, pero hay dos que destacan por encima del resto:

* Anorexia nerviosa

La anorexia nerviosa es el rechazo frontal a mantener el peso corporal, intentando bajarlo muy por debajo del umbral que daría el IMC. Así, estas personas también tienen una alteración de la visión de su peso y figura, debido al miedo irracional que general a ganar peso.

Un trastorno muy grave, y a la vez muy actual dentro de la anorexia es la drunkorexia. Consiste en el rechazo a la alimentación para no engordar, pero sin dejar las bebidas alcohólicas.

* Bulimia nerviosa

La bulimia nerviosa consiste en un trastorno en el que la persona comete “atracones”. Un atracón se define como una ingesta de una gran cantidad de comida en un tiempo corto, donde la persona pierde totalmente el control sobre lo que come y cuanto come. Muchas veces estos atracones se dan debido a la búsqueda de refugio de malestar en la comida.

La bulimia conlleva otro problema, y este consiste en que la persona se da cuenta de que no quiere ganar peso, y por lo tanto se provoca el vómito o abusa de diuréticos o laxantes.

#### Trastornos de la personalidad

Los trastornos de la personalidad son aquellos en los que la conducta de la persona enferma es muy distinta a la que sería normal en el resto de la sociedad, siendo estas diferencias en la cognición, en la afectividad, en las relaciones interpersonales o en el control de los impulsos.

Así, estas personas suelen tener un gran deterioro social y/o laboral, especialmente porque se suele empezar a dar al final de la adolescencia o primeros años de la edad adulta.

Es interesante saber que no se conoce todavía de donde proceden estos trastornos de la personalidad, pero se sospecha que pueden tener una base genética.

### En este trabajo: Grupos de trastornos psicológicos

Para poder clasificar con más sencillez los trastornos psicológicos, me he visto obligado a resumir la gran cantidad de ellos que hay en grupos. Tras el estudio del DSM IV (manual de referencia) y la entrevista con la psicóloga profesional, se ha determinado que los pacientes se deben de clasificar en cuatro grandes grupos de trastornos psicológicos, que son los siguientes:

1. Trastornos TOC
2. Trastornos de la Ansiedad
3. Trastornos Depresivos
4. Trastornos de la Personalidad

Estos grupos de trastornos han sido escogidos debido a la altísima frecuencia con la que se presentan en la consulta privada, además de las diferencias entre ellos, por lo que hace la obtención de datos de pacientes sencilla.

La única similitud entre los grupos se da con los trastornos TOC y los trastornos de ansiedad, puesto que un TOC es un tipo muy determinado y específico de trastorno de ansiedad, pero debido a su altísima frecuencia como caso he valorado la separación del mismo como grupo aparte.

### Distorsiones de la percepción de la realidad: Las distorsiones cognitivas

Las distorsiones de la percepción de la realidad son pensamientos automáticos distorsionados que se deben a errores en el procesamiento de la información que le llega al individuo. Son la base de toda la psicopatología ya que la unión de ellas da lugar a cambios emocionales que pueden derivar en un trastorno psicológico.

Gracias a, entre otros, David D. Burns (1942 - ) hay una serie de distorsiones cognitivas aceptadas dentro de la terapia cognitivo-conductual, y son las siguientes:

1. Pensamiento Dicotómico:

El pensamiento dicotómico, también conocido como “Pensamiento Todo o Nada” o “Pensamiento Binario”, consiste en evaluar las cualidades de la propia persona en dos categorías extremas: blanco o negro. Esta distorsión cognitiva constituye la base de lo que denominamos perfeccionismo. Un ejemplo de esto sería la persona que sale de un examen de 50 preguntas, y tiene 3 mal. Si esa persona piensa que el examen ha sido un desastre, estará cayendo en un pensamiento dicotómico, pues el examen no ha sido ni perfecto ni horrible.

Esta visión es falsa, ya que aporta una visión de la vida que no es realista, porque muy escasas veces la vida acaba siendo blanca o negra. Así, una persona que intente situar sus experiencias y emociones en categorías absolutas lo único que va a conseguir es estar de una manera constante en depresión, debido a que las percepciones no se ajustarán a la realidad que esa persona anhela con una exactitud total.

1. Generalización Excesiva

La generalización excesiva consiste en llegar a la conclusión irracional de que algo que le ha ocurrido una vez, o de una manera escasa, volverá a sucederle de nuevo en el futuro. Normalmente, esta distorsión cognitiva se da en el ámbito negativo, de tal manera que las personas que la sufren se sienten constantemente abatidas debido a que se piensan que la situación desagradable que han vivido inevitablemente la volverán a vivir.

Un ejemplo de generalización excesiva es el aseverar que nunca se tendrá pareja e hijos debido al rechazo de una persona. Esto es un error por dos motivos:

1. Todas las personas no tienen el mismo gusto
2. Por el simple hecho de que una persona te haya rechazado, no tienen que rechazarte el resto
3. Filtro Mental

El filtro mental, también conocido como “abstracción selectiva”, consiste en estar o haber pasado una situación, y al analizarla centrarse en sólo un elemento de esa situación, haciendo caso omiso al resto. Normalmente, se desarrolla un filtro mental negativo, por lo que las personas que padecen esta distorsión ven toda la situación rodeada de negatividad.

Esto suele pasar cuando una persona está deprimida. La persona que lo sufre “se pone unas gafas” que le sirven de filtro para que nada sea positivo, y todo lo que “llega” a la mente son pensamientos negativos.

1. Descalificación de lo positivo

Esta distorsión es una maximización de la anterior, y consiste en transformar situaciones que pueden ser neutras, o incluso positivas, en situaciones negativas. Esto se hace ignorando la parte positiva de la situación, y dándole la vuelta con algún cierto argumento irracional para convertirlo en algo negativo.

Un ejemplo de esto sería cuando un compañero de trabajo te felicita por tus últimos logros, y tú sólo piensas en que lo que quieren es quedar bien, no te quieren felicitar de verdad. En ese momento, le has dado la vuelta a la situación y ya no te estás centrando en tus logros y trabajo bien hecho, sino que te estás centrando en tu creencia de que la gente que te rodea no se alegra por ti de una forma real.

Esta distorsión cognitiva es bastante común, y por desgracia es una de las distorsiones cognitivas más importantes para acabar con un cuadro depresivo.

1. Conclusiones Arbitrarias

Las conclusiones arbitrarias, también conocidas como “apresuradas”, son una distorsión cognitiva mediante la cual la persona toma decisiones y saca conclusiones de una determinada situación, normalmente negativas, de una manera no justificada.

Esta distorsión cognitiva se puede dividir en dos tipos de casos:

1. Lectura del pensamiento

La lectura del pensamiento consiste en estar convencido de una afirmación que es negativa para la propia persona, sin tener hechos fehacientes que lo demuestren, y no molestarse por comprobarlo, dándolo por hecho.

Un ejemplo de esto sería una chica que sale todos los fines de semana con sus amigas. Un fin de semana, las otras dos o tres chicas no pueden quedar por diversos motivos de fuerza mayor, y se lo hacen saber a la primera. Si esta chica empezara a pensar que no quieren quedar con ella “porque es una persona aburrida”, estaría cayendo en una lectura del pensamiento.

1. Error del Adivino

El error del adivino consiste en suponer, sin pruebas para ello, que va a ocurrir algo malo, y que siempre va a ser malo. Además, no solo lo supone, sino que lo toma como un hecho, algo asegurado.

Un ejemplo de esto se puede dar con las personas mayores que sufren alguna enfermedad. Siempre piensan que se van a morir con ello, y luego al tomarse alguna medicina se sienten mejor y ven que su pensamiento era erróneo. Otro ejemplo, muy común entre los estudiantes, es el pensamiento de los días antes del examen de “a ver si suspendo”, “a ver si me quedo en blanco”…

1. Maximización y Minimización

También conocido como magnificación y minimización, esta distorsión cognitiva se basa en el hecho de aumentar o disminuir las situaciones de una manera totalmente desproporcionada a la importancia que tienen, dando lugar a pensamientos catastróficos sobre pequeños errores y dando poca importancia a elementos positivos.

1‑2. Ejemplo Maximización

1. Razonamiento Emocional

El razonamiento emocional es otra distorsión cognitiva basada en la toma de las emociones propias como prueba irrefutable de verdad. Este razonamiento no es correcto, debido a que los sentimientos únicamente reflejan pensamientos, y estos son subjetivos.

Un ejemplo de razonamiento emocional sería: “Me siento como un fracasado, por lo tanto jamás aprobaré la carrera y nunca conseguiré un trabajo”.

Es importante destacar que, obviamente, el razonamiento emocional es una de las distorsiones cognitivas fundamentales para el diagnóstico de trastornos depresivos.

1. Los debería

“Los debería” es una distorsión cognitiva ciertamente pintoresca, pues parten del hecho de animarse a sí mismo y motivarse diciendo: “Debería hacer esto”. Pero esto es un arma de doble filo, ya que estas frases nos hacen una presión sobre nosotros mismos.

Además, los debería no solo se suelen dirigir a la persona propia, sino que muchas veces se suelen dirigir hacia otras personas, criticando elementos que suponemos que deberían de hacer. Pero la generación de críticas con los debería hacia los demás lo único que hacen es generar un cierto resentimiento en la persona propia.

De esta manera, los debería son “fácilmente” cambiables. Lo único que deberá de hacer la persona es cambiar sus expectativas de la realidad, antes irreales, hacia algunas más reales, porque de lo contrario la persona se convertirá en una amargada y una cínica.

1. Etiquetación

La etiquetación consiste en la creación de una imagen de sí mismo errónea y negativa, basada únicamente en los errores que se han cometido. Es el extremo del punto 2, la generalización excesiva.

Burns (1980) afirma: “La filosofía en la que se basa [la etiquetación] consiste en que la medida de un hombre la dan los errores que comete”. Por ejemplo, tras aprobar todos los exámenes de la carrera, uno suspende uno en el cuarto curso, ya cerca de sacarse el título. Una persona que no use la etiquetación podría decir: “He estudiado mal, es un problema, pero lo sacaré”. En cambio, una persona que padezca la distorsión cognitiva de la etiquetación pensará algo como: “Soy un perdedor, no soy capaz ni de sacarme la carrera”.

Así, a estas personas hay que hacerles ver que la vida no es sólo lo que hace uno, y que poner etiquetas consiste en detallar un hecho con palabras que conllevan una fortísima carga emocional y por lo tanto no son objetivas.

1. Personalización

La personalización es la distorsión máxima por la cual nos culpamos. Consiste en asumir la responsabilidad de algo malo que haya pasado, aún no teniendo relación ninguna con ese hecho.

El problema de la personalización, como he comentado antes, es que se siente en sí mismo una gran culpa, y se siente como si muchos elementos no positivos que ocurren a su alrededor solo dependieran de sí mismo.

Un ejemplo de esto es el profesor al que los alumnos no le hacen los deberes. Este profesor, tras razonar, puede pensar que la culpa es de los alumnos porque no cumplen con su deber, o puede pensar que la culpa es suya, y que por lo tanto es un malísimo profesor. En este segundo caso, el profesor estaría cayendo en una personalización.

### En este trabajo: Distorsiones cognitivas usadas

Debido a la naturaleza científica del trabajo en busca de predicciones y clasificaciones mediante técnicas de inteligencia artificial, debo de hacer una selección de qué distorsiones serán las que use para poder hacer predicciones de los grupos de trastornos.

Debido a lo aprendido, y tras consultar con una psicológica profesional con despacho privado, usaré todas las distorsiones cognitivas a excepción de la de la descalificación de lo positivo, debido a que prácticamente todas las descalificaciones de lo positivo se centran en un filtro mental previo, y habiendo este filtro no es necesario usar esta distorsión cognitiva en la recopilación de datos y en la predicción.

Esto es una ventaja, debido a que me ahorraré una variable tanto en la toma de datos como en el procesamiento, lo cual para mi dataset no será demasiada diferencia a la hora de computarlo, pero sí lo sería en un dataset con muchos más pacientes. En esos casos, la toma de estas decisiones resulta de un papel fundamental a la hora de hacer cálculos para ser más eficientes computacionalmente.

#### En este trabajo: Otras variables usadas

A continuación, expondré una lista del resto de variables que usaré en este trabajo. Estas variables han sido recopiladas de la entrevista que tuve con una psicológica clínica profesional con despacho privado y más de 25 años de experiencia, y junto con estas variables expondré su importancia.

1. Nombre

El nombre, dado sin apellidos ni otros datos personales debido a la LOPD, no se utilizará en predicciones ya que carece de relación con los trastornos. Únicamente se utilizará como identificativo de los pacientes y para el análisis exploratorio del dataset.

1. Edad

La edad es un factor que, aunque no es determinante en el análisis psicopatológico de la persona, puede tener que ver, ya que algunos trastornos como la anorexia nerviosa se dan con más frecuencia en un rango de edad determinado.

1. Sexo

El sexo es un factor importante, debido a que las personas de género femenino normalmente suelen ser más propensas a los trastornos psicológicos que las masculinas, además de que acuden más a consulta. Además, los cuadros no son iguales en un sexo o en otro. Si por ejemplo analizamos en los trastornos de la alimentación, la anorexia es un trastorno que lo tiene un ratio de mujeres muy superior al de los hombres, pero en cambio en la vigorexia ocurre todo lo contrario.

1. Relación con el Contexto

La relación con el contexto es una de las variables más importantes que recopilo. Esta se refiere a la relación que tiene el paciente con las personas más cercanas a su entorno, que normalmente suelen ser los padres o la pareja.

Debido a la complejidad de esta variable, y al número de opciones que se pueden dar, he tomado la decisión de dividir esta variable en tres, obteniendo lo siguiente:

1. Relación con el contexto mala:

En el caso de una relación mala con el contexto, la persona se lleva mal con sus personas cercanas debido a una discusión o un enfrentamiento similar.

1. Relación con el contexto de trauma:

En este caso, la persona no se lleva bien con el contexto debido a algún trauma que haya sufrido, como puede ser la pérdida de algún familiar cercano, abusos sexuales o elementos similares.

1. Relación con el contexto buena:

En este caso, la persona tiene una buena relación con sus personas cercanas.

1. Habilidades sociales

Con las habilidades sociales, me estoy refiriendo a la forma que tiene la persona de relacionarse con el exterior. Debido a esto, hay reconocidas tres grandes respuestas:

1. Inhibición

Cuando una persona es inhibida, no suele expresar lo que piensa o siente, y si lo hace suele hacerlo en momentos que no son los más adecuados para ello, o torpemente.

1. Asertividad

Cuando una persona es asertiva (también conocida como hábil), expresa lo que piensa de forma directa y adecuada, respetando tanto sus derechos como los de los demás.

1. Agresividad

Cuando una persona es agresiva, da a conocer lo que piensa y siente normalmente en segunda persona, y lo hace de una manera alterada y alienando la conversación con el resto de personas.

1. Impulsividad

Conocemos impulsividad como la tendencia que tiene una persona a realizar unos actos sin premeditación, y por ello sin tener en cuenta las consecuencias.

Las personas impulsivas son más propensas a la agresividad (lo veremos posteriormente en la práctica), y a veces pueden ser personas con problemas debido al consumo de sustancias como pueden ser el alcohol o las drogas, especialmente si se trata de un trastorno de la personalidad.

Las personas que tienen impulsividad suelen ser más propensas a la pérdida del autocontrol, y por ende, trastornos como la hiperactividad o cuadros de ansiedad son comunes en estas personas.

# 

# Capítulo 2: Data Science

## Mining Vs Machine Learning Vs Data Science Vs Big Data

Cuando se habla de temas como Big Data o data science, hay varios conceptos que se nos pueden venir a la cabeza, pero dos de ellos sin duda son el data mining (o, en español, minería de datos) y machine learning (aprendizaje automático).

Las diferencias entre estos términos pasan por alto a la mayoría de las personas, pero a continuación haré hincapié en ellas para tener los conceptos bien separados, y a partir de ahora saber exactamente a qué nos referimos:

### Data Mining

Los orígenes del data mining vienen de la dificultad de poder manejar diferentes tipos de datos con las herramientas existentes. De este trabajo se acabó derivando en ideas que se tomaron prestadas de otros campos, como la estimación o el muestreo tomados de la estadística, o los algoritmos y las técnicas de aprendizaje provenientes de la inteligencia artificial. También otras áreas tienen un papel esencial en todo lo que rodea al data mining, como es el área de visualización, de bases de datos o de computación.

Así, con data mining, o minería de datos, hacemos referencia al proceso de descubrir información que pueda ser útil, a través del análisis de grandes repositorios de datos. De este modo, con la minería de datos se intenta encontrar patrones y soluciones de preguntas que, de otro modo, estarían ocultas entre los datos.

Es importante distinguir entre data mining y la recogida de información. Mientras que data mining usa técnicas estadísticas y matemáticas para la obtención de información dentro de un dataset, la recogida de información consistiría en, por ejemplo, una búsqueda en una base de datos para un sujeto concreto. A pesar de centrarse los dos en los datos, son elementos y técnicas distintas y, por lo tanto, deberán de mantenerse por separado.

El descubrimiento de conocimiento es la última meta de la minería de datos. Conocido en la comunidad anglosajona como KDD (Knowledge Discovery in Databases), el descubrimiento de conocimiento podríamos decir que es el proceso total de convertir los datos puros de la base de datos en una información útil. Es decir, el descubrimiento de conocimiento es el concepto por el que, mediante el uso de data mining, obtenemos información útil de una gran cantidad de datos que, a priori, no nos da ninguna información a simple vista.



2‑1 Proceso KDD detallado

Este descubrimiento de información consiste en una serie de pasos, que van desde un preprocesamiento de los datos para su preparación, hasta un post-procesamiento para su posterior obtención de información. Observemos este proceso con más detenimiento:

1. Preprocesamiento de los datos:

El preprocesamiento de los datos es un paso esencial en data mining, debido a que los datos pueden estar guardados en una gran cantidad de formatos y formas, o incluso estar distribuidos en diferentes repositorios.

Una vez importado el dataset, o el conjunto de datasets con los que se va a trabajar, se debe de hacer este preprocesamiento de los datos para prepararlos de cara al data mining. De esta manera, acciones como la unión de tablas, la reducción de la cantidad de variables (también conocido como reducción de la dimensionalidad), o la obtención de subgrupos de datos, serán pasos muy importantes de cara a preparar los datos para los próximos pasos.

Normalmente, este preprocesamiento suele ser la parte que más tiempo consume en el proceso de la minería de datos, debido a que es muy manual y laboriosa.

1. Data Mining:

En este paso, usaremos las numerosas técnicas estadísticas y matemáticas que conforman el data mining, como pueden ser la unión por grupos, el estudio de la variabilidad, el estudio de las relaciones entre las observaciones o el estudio de la frecuencia entre muchas otras. Este conjunto de tareas recibe el nombre de “tareas descriptivas”, ya que el objetivo de las mismas es obtener patrones que resuman las relaciones que haya por debajo en los datos.

Además, junto con las tareas descriptivas podremos hacer un análisis predictivo para poder predecir ciertas características de futuras observaciones. Esto lo veremos más adelante.

1. Post-Procesamiento de los datos:

Con el post-procesamiento de los datos, nos referimos esencialmente a la quizás necesaria trasformación final de los datos de cara a la siempre necesaria visualización. Ya fuera de data mining, junto con esta visualización, siempre se debe hacer un análisis e interpretación de los datos obtenidos, de cara a la aclaración de los mismos y la finalización del proceso, obteniendo información útil.

También, otro post-procesamiento muy usado es la unión de los resultados obtenidos de data mining a otras herramientas, como pueden ser las de marketing, de tal manera que estos datos se puedan usar en otros ámbitos. Este proceso es conocido como “closing the loop”, que se puede traducir por “cierre del círculo”.

### Machine Learning

Si hablamos de machine learning, o como también se conoce en países de habla hispana, aprendizaje automático, nos estamos refiriendo a un subconjunto de la inteligencia artificial que, además de usar los principios de data mining, es capaz de hacer correlaciones de una manera automática, y también es capaz de aprender de los datos que se tienen y se tendrán, de tal manera que el modelo pueda seguir mejorando con el paso de mayor cantidad de datos.

El uso que se hace del machine learning por la población es diario, ya que se utiliza en numerosos ámbitos, tales como la publicidad (especialmente en internet), algoritmos de búsqueda, algoritmos de predicción del tiempo atmosférico…

Podríamos decir que, aunque son muy parecidas, data mining se centra un poco más en las relaciones de los datos que hay ahora mismo, y la obtención de información de ello, mientras que el machine learning usa estos principios de data mining para también obtener predicciones y clasificaciones, y aprender de los datos para mejorar estas técnicas. De este modo, el aprendizaje automático puede mirar patrones y aprender un comportamiento, mientras que el data mining es el recurso en el cual se basa el machine learning.

2‑2. Subdivisiones de la Inteligencia Artificial

Así, las tareas más relacionadas con machine learning se llaman tareas predictivas, ya que se basan en predecir un atributo particular a través de los valores del resto de atributos. A partir de ahora, nos referiremos a ellos como modelos predictivos.

Respecto a la posición relativa del machine learning respecto al data mining a la hora de hacer data science, hay varias versiones de donde se debe colocar. Es verdad que, mediante las técnicas de machine learning, estamos obteniendo información a partir de los datos, y además usamos muchos conceptos básicos de data mining en esta parte, por lo que podría incluirse como un subapartado de data mining, que se hace después del preprocesamiento y análisis exploratorio de los datos.

Otras teorías dicen que el machine learning deberá de ir como un apartado diferente de data mining y siempre después de este, debido a la naturaleza predictiva en vez de descriptiva.

Debido a estos dos enfoques, y a la vista de tomar una decisión, analizaré el machine learning como un apartado dentro de data mining, pero no sin antes volver a remarcar la gran conexión que tienen ambos conjuntos, con simplemente una naturaleza diferente.

### Data Science

De esta manera, si ya tenemos unos algoritmos que me permiten obtener información, y otros algoritmos que van mejorando con el paso de los datos y consiguen hasta predecirme información, ¿qué cabida tiene data science?

Con data science nos estamos refiriendo al elemento que cobija a data mining y a machine learning. Data science no es más que un término genérico que aúna un conjunto de técnicas o subdisciplinas, como data mining, machine learning y visualización de datos, entre otras, para obtener un conjunto de “insights” o conclusiones que sean útiles al usuario final, como puede ser una empresa (Business Intelligence) o cualquier otro usuario interesado.

De una manera más espectral, podríamos decir que data science consiste en la mezcla de una serie de procedimientos matemáticos que, junto con conocimientos del problema tratado y de tecnología especializada, consiguen obtener conclusiones efectivas y fácilmente entendibles para el usuario final.

### Big Data

Con Big Data, hacemos referencia a un término muy de moda en los últimos tiempos. Hemos visto que data science recoge todo el conjunto de técnicas desde la importación de la información hasta la obtención de los resultados finales con su información útil y entendible por cualquiera.

De este modo, “big data” hace referencia simplemente a la disciplina que trabaja con unas grandes cantidades de datos. Es una disciplina que, al igual que el “medium data” y el “small data”, están presentes en proyectos de data science.

1. Small Data

Con small data, nos referimos a proyectos en los cuales los datos están en un formato CSV pequeño, una base de datos pequeña o incluso en un Microsoft Excel. En estos proyectos, se puede trabajar con un ordenador estándar, y los datos se pueden cargar perfectamente en memoria RAM.

En mi caso, este proyecto usará un estilo de small data, debido a la pequeña cantidad de observaciones y variables que poseo.

1. Medium Data

Los proyectos que usan medium data son aquellos que usan una cantidad de datos más grande que los de small data, y, aunque los datos se pueden albergar normalmente en un ordenador, las técnicas de extracción de los mismos son distintas, ya que no se puede pedir toda la información de golpe a riesgo de bloquear el ordenador sobrecargando la memoria.

Para el tratamiento de los problemas de medium data se suele utilizar Apache Spark, puesto que este software permite evitar esta sobrecarga de memoria y de esta manera poder traer los datos según se vayan requiriendo.

1. Big Data

Finalmente, los proyectos de big data son aquellos en los que se necesitan varios ordenadores, o un servidor de grandes dimensiones, para poder tener toda la información. Estos proyectos son mucho más complejos, ya que suelen necesitar técnicas de sincronización entre ordenadores, cálculo en paralelo o en grid y técnicas similares.

En este tipo de problemas, soluciones combinadas como el uso de Apache Hadoop (junto con técnicas como MapReduce) y Apache Spark (para evitar los Overflow de RAM) son imprescindibles para obtener las conclusiones en un tiempo razonable de tiempo, además de hacer una gestión eficiente y segura de los datos.

De este modo, podemos afirmar que “Small Data”, “Medium Data” o “Big Data” son un contexto, un “framework” donde se mueven los proyectos de data science, y que dependiendo de cual sea necesario se necesitará una tecnología u otra para el tratamiento de los datos.

También, otro método más informal pero muy efectivo para saber bajo que paradigma vamos a trabajar es el de tener en cuenta las 3 V’s del Big Data: Velocity, Volume, Variety. Antes de entrar en detalle con ellas, es importante aclarar que hay más V’s, donde algunos expertos dan hasta 12. En este trabajo se repasarán las 3 centrales, las que más aceptadas están por todos.



2‑3. Las 3 V's del Big Data

* Velocity

Velocity se refiere a la velocidad de la creación de nuevos datos. Si se crea una gran cantidad de datos en breve lapso de tiempo, podremos decir que podríamos estar en un problema de Big Data y por lo tanto una base de datos no convencional (como una NoSQL) sería una opción interesante.

* Volume

Quizás el más obvio, el volumen de datos con los que vamos a trabajar influye enormemente en la manera de encarar el problema. No hay reglas escritas, pero un problema de Big Data se supone que no se puede resolver en un ordenador de casa, ni se pueden almacenar los datos en el mismo. De tal manera, los problemas que encuadraremos dentro de Big Data serán problemas de cientos de terabytes, petabytes o incluso mayores.

* Variety

La variedad de los datos, debido a la gran cantidad de estructuras de datos y fuentes, es otro de los indicadores principales de que nos encontramos ante un problema de Big Data. No es el requerimiento más importante a la hora de determinar bajo que paradigma resolveremos el problema (los dos anteriores se antojan vitales), pero sí es necesario tener en cuenta que en Big Data los datos de diferentes fuentes y diferentes formatos son muy comunes.

## Antes de hacer Data Mining

Un paso fundamental a la hora de afrontar un problema con datos es obtener estos datos. Los datos pueden venir de numerosas fuentes, y más cuando se puede afrontar un problema que sea bajo big data o medium data, que, como hemos visto, pueden estar distribuidos entre numerosos ordenadores, y las técnicas de petición de los datos son más complejas que una simple carga en memoria RAM.

Así, el conocimiento de tecnologías como Apache Spark, para el control del flujo de datos a memoria, se antoja esencial en aquellos proyectos que no pertenezcan al grupo de small data.

### En este trabajo: Obtención de los datos

En mi caso, podría considerar el problema de “small data”, y el dataset se ha obtenido a partir de la entrevista con una psicóloga profesional, donde me ha dado datos de pacientes reales de su consulta privada. Para mantener la confidencialidad, de dichas personas solo he recibido los datos clínicos, nombre, edad y sexo. Los apellidos, la ciudad e imágenes, además de irrelevantes para el trabajo, no han sido proporcionados por temas de privacidad.

Posteriormente, he confeccionado un dataset a mano, con formato CSV (Comma Separated Values), para poder importarlo posteriormente a RStudio. Esta entrevista y posterior confección del dataset conllevaron aproximadamente unas 5 horas de trabajo continuo. Preferí la confección a mano de un dataset real, y no la obtención en internet de uno, debido a que en internet no hay ningún dataset con esta información que necesito para estas predicciones, a fecha de enero de 2019.

Elegí el formato CSV debido a varios motivos, como son la facilidad de confección de dicho dataset, que este formato es muy común en problemas de Big Data e Inteligencia Artificial, y la facilidad de importación posterior para los programas que utilizaré.

También, considero importante comentar que la variable más importante se encuentra al final de cada fila del dataset, y corresponde al grupo al que pertenece el paciente. Estos grupos son una variable discreta que versa del 1 al 4, y se corresponde de la forma siguiente:

1. Trastorno Obsesivo – Compulsivo
2. Trastorno de Ansiedad
3. Trastorno de Depresión
4. Trastorno de Personalidad

## Data Mining

Como hemos visto, con data mining nos estamos refiriendo a todo el proceso de conseguir una información útil y entendible a partir de un conjunto de datos. Así, este apartado estará dividido en la explicación de los diferentes procesos que he llevado a cabo para conseguir hacer data mining, y la explicación de cómo han sido implementados en este proyecto.

### Pasos previos y preparación de los datos

Según IBM, empresa líder mundial en ventas de máquinas para negocios, a la hora de preparar los datos antes de un proceso de minería de los mismos es importante seguir tres pasos:

1. Entendimiento del negocio

Respecto a entendimiento del negocio, se entiende la comprensión de cuál es el objetivo que se busca, así como cuál es el indicador que nos dirá que hemos tenido éxito. Obtener una serie de información puede estar muy bien sin entender nada, pero con un background previo en el área de aplicación de esos datos podemos saber si vamos por el buen camino o no.

1. Entendimiento de los datos

Con el entendimiento de los datos, y parte más informática de las dos, se entiende la selección de los datos que se consideren más relevantes, y la completa compresión de estos datos. Ya que este paso se tendría que hacer “a ciegas”, por lo que el primer paso de comprensión del negocio se antoja vital. Después, mediante preprocesamiento de los datos podremos obtener matemáticamente cuales son las variables más importantes.

Es importante esta comprensión de los datos debido a que muchas veces los datos, especialmente en problemas de medium data y big data, vendrán desestructurados y en diferentes tablas, y la compresión de cada dimensión en profundidad para conocer lo que aporta hacia el problema, saber si los datasets tienen sentido juntos, y conocer si el cambio de algún dato o de alguna dimensión es necesario, puesto que si no el tratamiento de los datos se haría “a ciegas” y obviamente esto no es la forma óptima de realizarlo para obtener buenos resultados en la etapa de machine learning. Así, es necesaria una comprensión en profundidad del problema para la identificación de los datos más relevantes.

También es importante el entendimiento de los datos y del problema debido a que los datos nos pueden llegar de una manera desestructurada, como ocurre por ejemplo con los datos de las redes sociales. Estos datos tienen una estructura interna que no se puede ver a simple vista, puesto que no siguen un formato específico, lo cual implica un esfuerzo y un entendimiento extra para poder operar satisfactoriamente con los mismos.

1. Preparación de los datos en sí

Una vez que tenemos una serie de datos con los que vamos a trabajar, seleccionados y entendidos, y una serie de objetivos en mente, es hora de codificar. Para ello, lo primero que tenemos que hacer es una segunda preparación de los datos. Para ello, tendremos que comprobar los tipos de datos, y la calidad de los datos:

1. Tipos de datos:

La comprobación de los tipos de datos se antoja como una parte fundamental de la preparación de los datos, debido a que un dataset no es más que un conjunto de objetos de datos. A la hora de hacer clasificaciones y predicciones, la diferencia entre un carácter, una variable continua o una discreta puede marcar la diferencia de la calidad de la predicción, o incluso si el algoritmo predictor funciona o no.

1. Comprobación de que los datos no están comprometidos

La comprobación de la seguridad de los datos, especialmente si se trata de datos sensibles de personas, es fundamental tanto para la seguridad y el honor de las mismas como para la legalidad de la empresa. De esta manera, los servidores u ordenadores que posean los datos deberán de tener una serie de medidas de seguridad.

Por otra parte, si los algoritmos de machine learning producen cualquier resultado que pueda ser también confidencial (como, por ejemplo, la predicción de la ideología política de una persona), estos nuevos datos generados también se deberán de salvaguardar al igual que los anteriores.

1. Calidad de los datos:

En muchas ocasiones se hace minería de datos sobre datos que no han sido recogidos específicamente para ese momento o esa intención. Debido a esto, debemos de hacer una pequeña valoración de calidad al principio, en la obtención. La evitación de problemas de calidad de los datos es un elemento fundamental a la hora de hacer data mining. Así se llega hasta la primera parada técnica de un data scientist: La limpieza de datos. Con una correcta limpieza de los datos se obtiene un dataset que, aunque puede estar más incompleto en algunas ocasiones, posee todos sus datos de una manera igual, óptimamente usable y entendible por los diferentes procesos por los que pasarán después los datos.

Posteriormente, para continuar con la preparación de los datos, hay que valorar la opción de un preprocesamiento y transformación de los mismos, atendiendo a algunas técnicas como:

1. Agregación

La agregación consiste en la técnica de unir dos o más objetos en uno solo, y se basa en la filosofía de que a veces, “menos es más”.

Con la agregación, siempre y cuando tenga sentido hacerla, podemos reducir el número de filas o columnas, de tal manera que puede ser computacionalmente más eficiente y menos complejo, con lo que el tiempo de computación se reduce. Si tomamos la visión contraria, con un dataset más pequeño podemos usar algoritmos más complejos en un tiempo razonable de computación.

Uno de los problemas más obvios que tiene la agregación es la pérdida de información que conlleva una unión de datos, por lo que la valoración de si se hace una agregación o no es esencial de cara a los resultados que se pueden obtener posteriormente.

1. Muestreo

El muestreo es una técnica que se utiliza para coger sólo una cantidad fraccional de los datos totales que se tienen, y analizar ese nuevo conjunto. En ámbitos como la estadística que, como ya hemos visto, está muy relacionada con todo este mundo, esta técnica se lleva usando durante muchísimo tiempo para hacer un análisis preliminar de los datos.

El uso de muestras se suele usar cuando trabajamos bajo big data o medium data, debido a que el análisis de estas cantidades de datos es demasiado costoso y, si la muestra es fielmente representativa, obtendremos unos resultados casi tan buenos como con los del dataset completo.

El hecho de que una muestra sea representativa depende de si tiene un valor similar en una propiedad al valor de la misma propiedad en el conjunto de datos total. Es decir, si por ejemplo tomamos como medida de representatividad la media, si el valor de la media del conjunto de muestra es similar al valor de la media del conjunto total podremos decir que el conjunto es representativo. Pero obtener un conjunto representativo no es tan sencillo como coger unas cuantas muestras y usarlas. Hay diferentes aproximaciones que se pueden hacer, como son el muestreo aleatorio (cualquier elemento tiene la misma probabilidad de ser cogido que el resto) y el muestreo estratificado, que comienza con un conjunto de grupos ya preestablecidos.

1. Reducción de la dimensionalidad

La reducción de la dimensionalidad es una técnica consistente en la eliminación de columnas del dataset (y por lo tanto, dimensiones), de tal manera que mejore la eficacia de los algoritmos de data mining. En parte, esto se debe a que, en la reducción de dimensiones, algunas características innecesarias son eliminadas, además del ruido. Por otra parte, esta mejora viene dada por la llamada “maldición de la dimensionalidad”.

La maldición de la dimensionalidad consiste en el fenómeno basado en que la minería de datos aumenta en complejidad conforme el número de dimensiones aumenta. Esto, si lo vemos en el espacio, significa que cada vez el conjunto de los datos se vuelve más difuso, lo que complica la clasificación de los datos, lo que conlleva modelos más imprecisos y complicados de clasificar.

De este modo, la técnica más utilizada para el análisis de las dimensiones es conocida como PCA, acrónimo de Principal Components Analysis, que en castellano significa “Análisis de Componentes Principales”. PCA consiste en una técnica de álgebra lineal, aplicable a variables continuas, que encuentra nuevas variables a partir de la combinación lineal de variables anteriores. Haciendo esta técnica, si las variables son ortogonales, se obtendrá mucha información, mientras que si las variables tienen vectores muy similares en tamaño y dirección, se explicarán mucho la una a la otra y de este modo una de las dos será “innecesaria” en una gran medida.

En la siguiente figura podemos ver de forma gráfica una reducción de la dimensionalidad.

2‑4. Reducción Dimensionalidad Dataset Iris

PCA tiene numerosas características muy atractivas, como la posibilidad de encontrar eficientemente patrones en los datos, la posibilidad de reducir drásticamente la dimensión en una gran cantidad de problemas o la posibilidad de eliminar gran parte del ruido. Debido a todos estos beneficios, imprescindibles en data science, PCA posee una gran fuerza a la hora de hacer análisis de datos.

Es importante destacar que la diferencia entre las dimensiones las calcula mediante variabilidad, de tal manera que supone que el dataset entero tiene una variabilidad del 100%. Cuando dos vectores son muy similares, una de esas dos dimensiones no aporta prácticamente nada de variabilidad que pueda ayudar a explicar el problema, mientras que dos vectores muy distintos en tamaño y dirección aportarán una gran variabilidad, y por lo tanto explicación del problema de cara a los futuros algoritmos. De esta manera, con un cierto porcentaje de explicación los algoritmos de machine learning podrán trabajar de una manera muy aproximada reduciendo en un gran porcentaje la dimensionalidad, de tal manera que será computacionalmente más eficiente a costa de pequeños errores de clasificación o predicción.

Respecto al punto en el que se debe de parar de reducir, hay que saber asignarlo a cada problema, puesto que este tipo de técnicas variarán respecto del dataset, aunque por regla general, si el problema no necesita ser extremadamente preciso, con una explicación del 80% - 85% suele ser suficiente.

Para mirar esta explicación, es interesante observar unos valores llamados “eigenvalues”. Estos valores explican, para una cierta cantidad de dimensiones que tendría el problema, la explicación total que poseería, normalmente expresado en tanto por ciento o tanto por uno. También puede venir expresado de una manera distinta, donde se expresa como por cada dimensión que se añada la explicación que se añadiría al problema. Sea cual sea la representación que se obtenga, habrá siempre que mirar en qué momento la explicación baja de una manera más acusada con el aumento de las dimensiones, y ese punto, en la mayoría de los casos, será el óptimo para obtener el dataset con el que se trabajará posteriormente.

1. Creación de características

Como oposición a la reducción de la dimensionalidad, podemos poner en acción otra técnica llamada creación de características, o como se conoce en inglés, “Feature Creation”.

Esta técnica destaca por la creación de dimensiones a partir de las dimensiones ya existentes, de tal manera que se crea un nuevo dataset con unas dimensiones que capturan la información de una manera mucho más efectiva. Además, tiene como ventaja que se produce una reducción de la dimensionalidad, con todos los beneficios que hemos visto anteriormente. Por ello, en datasets con una muy alta dimensionalidad esta técnica es muy valiosa.

Existen tres métodos para la creación de características:

1. Extracción de características

Básicamente, la extracción de características consiste en la creación de un nuevo conjunto de características a partir de las anteriores. Desafortunadamente, esta técnica no puede ser usada demasiado a menudo, debido a que es muy específica de ciertos dominios, como el del análisis de píxeles de fotografías.

1. Mapeado de los datos a un nuevo espacio

La traslación de datos a un nuevo espacio para intentar ver patrones y características que antes pasaban desapercibidos debido a ruido u otros factores es una técnica sencilla y útil en muchos casos. Podríamos definirlo como “la búsqueda de un nuevo punto de vista”.

En caso de estar buscando patrones, una gran ayuda puede ser la aplicación de la transformada de Fourier, especialmente en el caso de las series temporales, ya que revelará información que, en este caso, tiene de forma explícita la frecuencia.

1. Construcción de características

Finalmente, tenemos la construcción de características como tercer método de la creación de las mismas. Este método se utiliza cuando en el dataset se tienen los datos correctos para obtener una información determinada, pero el algoritmo de data mining que se va a usar no acepta esta información. En este caso, la construcción de nuevas características construidas a partir de las originales puede dar lugar a unas características más útiles y aceptadas por el algoritmo determinado.

1. Discretización y transformación a binario

Muchas veces, cuando se tiene que ejecutar un algoritmo de clasificación, los datos deben de estar en forma categórica. También, algunos algoritmos para encontrar patrones necesitan que la información se encuentre de forma binaria. Así, para el primer caso hablamos de técnicas de discretización, mientras que para el segundo hablamos de técnicas de binarización. Vamos a verlos con más detalle:

1. Binarización

La binarización consiste en la técnica mediante la cual, para “m” valores categóricos, se le asigna un valor a cada uno que entre dentro del intervalo siguiente: . Una vez hecho esto, podemos convertir estos números en binario, de tal manera que obtendremos para cada observación un array con un valor binario correspondiente a la clase.

La binarización también se puede hacer de forma asimétrica, de tal manera que cada columna represente un estado, y por cada observación sólo puede haber un uno en este array, representando al estado al que pertenece, mientras que en el resto de estados esto es igual a cero. Esta técnica se suele dar, por ejemplo, cuando tenemos variables totalmente excluyentes entre ellas.

1. Discretización

La discretización de variables continuas depende del algoritmo a ser usado principalmente. Todas las discretizaciones conllevan dos subtareas imprescindibles, que son la elección del número de categorías que habrá, y la selección de intervalos de valores de pertenencia a cada variable.

1. Transformación de variables

La transformación de variables es una técnica que se aplica a todos los valores de una variable por diversos motivos. Hay dos tipos de transformación de variables, que explicaré a continuación:

1. Funciones Simples

Este proceso es tan sencillo como la aplicación de una función matemática a cada valor de la variable en cuestión. Las más usadas son la raíz cuadrada, el logaritmo y la inversa, para poder transformar un conjunto de datos que no siguen una curva gaussiana en otro conjunto que lo cumple.

Estas transformaciones deben de aplicarse con cautela, debido a que cambian la naturaleza de los datos, y no se trabaja por lo tanto con los originales. Por ejemplo, si hacemos la inversa, los valores superiores a 1 estarán siendo disminuidos, mientras que los menores estarán siendo aumentados. Por ello, nos debemos hacer unas preguntas previas, como pueden ser: ¿Es importante el valor exacto del dato? ¿Necesitamos tener un orden, una idea de qué observación tiene esa variable mayor que otra? ¿Cómo se aplica esa transformación a las variables extrañas, como al cero?

1. Estandarización

La estandarización de variables, también conocida como normalización (no confundir con una transformación gaussiana), es una técnica cuyo objetivo es que todo el dataset, o una parte de él, siga una determinada norma o propiedad.

Esta estandarización es muy necesaria en caso de que alguna de las variables destaque sobre el resto por algún motivo, especialmente si el valor es muy distinto al resto, debido a que cualquier método que use distancias euclídeas tendrá en cuenta la distancia y el peso de las variables, y si queremos que los modelos aprendan de una manera imparcial se deben de poner todas las variables bajo la misma norma.

### En este trabajo: Preparación de los datos

Siguiendo las bases teóricas expuestas en el punto anterior, en mi trabajo los datos han sido preparados de una manera premeditada para evitar cualquier problema. De este modo, primeramente, he conseguido un entendimiento del negocio y de los datos a través de la lectura de libros y entrevistas con una psicóloga profesional.

Para la confección del dataset, he tenido que tener en cuenta las necesidades posteriores de los algoritmos a usar. Para la simplificación del problema, he subdividido y posteriormente codificado muchas variables de forma binaria, de tal manera que la variable obtenga el valor de un uno cuando se dé el caso, y obtenga el valor de cero cuando no se dé, creando manualmente una binarización asimétrica. Esto se puede observar en:

* Relación con el contexto

Para la relación con el contexto, como comenté anteriormente, hay tres opciones: Relación mala, relación mala por trauma y relación buena. De este modo, estas tres variables formarán un array donde solo una de las tres puede ser posible, de tal manera que, por cada paciente, sólo una obtendrá el valor de un uno, y las otras dos obtendrán el valor de cero.

* Habilidades sociales

Para las habilidades sociales, también explicadas anteriormente, hemos visto que podemos tener tres valores: Inhibido, asertivo y agresivo. Como, al igual que en las variables anteriores, un paciente solo puede tener 1 tipo de habilidad social, por cada paciente una de ellas tendrá un uno, y las otras dos restantes un cero.

* Distorsiones cognitivas

Las distorsiones cognitivas, al contrario de las variables anteriores, sí pueden darse varias a la vez en un paciente, incluso con pacientes llegando a tener todas. Por ello, las distorsiones cognitivas formarán otro array donde cada distorsión representa a una variable. Si esta variable está presente en el paciente, se marcará con un uno, mientras que si no se presenta se marcará con un cero.

* Impulsividad

Respecto a la impulsividad, esta variable no forma array debido a que no considero los distintos tipos de impulsividad, y de este modo sólo será una variable binaria, donde una persona impulsiva tendrá un uno, y una persona no impulsiva tendrá un cero.

Para el resto de datos no hubo que hacer ninguna preparación previa, debido a que posteriormente, mediante código, cualquier cambio puede ser hecho, como la eliminación de columnas o el centrado y escalado (normalización) que posteriormente haré.

Respecto a la discretización, se efectúa cuando es necesaria y en las variables necesarias, siendo la variable correspondiente al grupo la más importante, debido a que es una variable puramente categórica.

Finalmente, si nos atenemos a calidad de los datos, como es un dataset obtenido a mano no hay datos incongruentes ni datos que falten, por lo que, excepcionalmente, en este problema no hay que realizar ningún movimiento en el ámbito de la calidad de los datos.

### Análisis Exploratorio o Descriptivo

El análisis exploratorio de datos consiste en un conjunto de técnicas estadísticas y de visualización para resumir y visualizar en primera instancia la realidad que se tiene, así como intentar encontrar patrones y relaciones entre los mismos, de tal manera que se pueda responder a alguna pregunta que previamente no se podría con una mirada simple hacia los datos. Además, la visualización de los datos históricos (especialmente en las series temporales) puede dar mucha idea de la posición en la que se encuentra ahora mismo el problema. Esta técnica fue inventada por el estadista John Turkey en la década de 1970.

Además de encontrar patrones y relaciones, el análisis exploratorio de los datos es un elemento que se antoja fundamental a aplicar antes del machine learning, debido a que en la mayor parte de los datasets hay datos outliers, que faltan o inconsistentes. Debido a esto, hacer un análisis exploratorio ciertamente profundo, ver si las relaciones que existen cuadran con la realidad (mediante un conocimiento previo del negocio), y la eliminación de variables outliers o que no aportan nueva información debido a su varianza cercana a cero se antoja fundamental para obtener unos modelos posteriores de machine learning que sean rápidos, eficaces y precisos.

Debido a esto, primero hay que hacer unas pequeñas comprobaciones para comprobar nuestros datos, que se dividen en la comprobación de los tipos de datos, y la comprobación de la calidad de los datos.

Una vez que hemos comprobado ambos, podemos empezar con la parte estadística del análisis exploratorio. Hay varios pasos que se pueden cubrir, que los iremos viendo en los siguientes subapartados.

#### Resumen de las estadísticas del Dataset

El resumen de las estadísticas del dataset no es más que un conjunto de números indicando varias características de un dataset. En estas estadísticas, hay varios apartados que tendremos en cuenta:

1. Frecuencias y la moda

La frecuencia de una variable consiste en un número, que puede estar entre cero y uno, que indica el tanto por uno de ocurrencias de dicho valor de la variable en una lista de m objetos. Por ello, sigue la siguiente fórmula:

Si hablamos de la moda, simplemente nos estaremos refiriendo al valor de cierta variable que tiene una frecuencia mayor que los otros.

1. Percentiles

Para datos ordenados, el percentil de un conjunto de valores nos da mucha información. Dado un número p entre 0 y 100, el p-esimo percentil es un valor de x donde el p% de los datos totales son inferiores a ese valor p-esimo. Así, podremos obtener qué valores destacan por encima del resto en un determinado porcentaje.

2‑5. Percentiles sobre una normal

1. Media y mediana

Para los datos que son continuos, dos de las estadísticas más básicas y a la vez más miradas son la media y la mediana.

Podríamos definir la mediana como el valor que está en el medio de un conjunto de datos. Si hubiera un número par de datos, sería la media de los dos valores del centro.

La media se puede definir mediante la siguiente función:

Pero la media suele tener problemas, debido a que puede tener valores outliers, o que sin ser outliers la distorsionen en cierto modo. Para ello se inventó el concepto de media recortada, que consiste en coger un porcentaje, que suele rondar entre el 1% y el 5%, y desechar ese porcentaje de datos tanto de la parte superior como de la inferior del dataset ordenado.

1. Rango y varianza

El rango y la varianza son las llamadas “medidas de dispersión”, ya que miden el dominio en el que se proyectan los datos, y la diferencia entre los valores.

La más simple de las dos es el rango, que se puede medir tanto con la resta del valor más alto menos el valor más bajo, como con un intervalo cerrado donde el primer valor del mismo sea el más pequeño, y el segundo el más grande.

La varianza, aunque más complicada, es el valor preferido a la hora de calcular la dispersión de los datos, y se suele representar como . La fórmula a la que atiende, siendo m el total de datos, es la siguiente:

La desviación típica entonces respondería a:

1. Resumen de estadísticas multivariable

Las estadísticas multivariable son aquella que poseen más de un atributo.

Para calcular las estadísticas en este caso, se debe de calcular la media y la mediana de manera separada a cada una de las variables.

En caso de la dispersión, podemos calcularlo también de manera separada para cada variable, aunque en este caso podemos comparar unas medidas con otras mediante la matriz de covarianza. Esta es una matriz que se representa bidimensionalmente (comparando variables dos a dos), y los valores que se muestran en la matriz son la covarianza de las que forman la fila y la columna.

Dadas las variables Xi y Xj, y la cantidad total de variables m, podemos calcular la covarianza atendiendo a la siguiente fórmula:

También es muy utilizada la llamada matriz de correlación. Esta es otra matriz bidimensional que representa la fuerza de relación lineal entre todas las variables de un dataset, y en cada posición se muestra la correlación entre las dos afectadas. Esta matriz es muy interesante, ya que es simétrica respecto a la diagonal, y además la diagonal consiste en las correlaciones de cada variable consigo misma, lo cual da un resultado siempre de 1.

La correlación sigue la siguiente fórmula:

#### OLAP y Análisis Multidimensional

La visión de la información en arrays multidimensionales conlleva una serie de técnicas determinadas, y unos sistemas de bases de datos que soporten este formato. Muchos sistemas gestores de bases de datos ya soportan este formato, especialmente los sistemas conocidos como OLAP (OnLine Analytical Processing). Debido a esto, el enfoque que daré a este apartado estará basado en estos sistemas OLAP.

1. Primer Paso: Representación de los datos

Al igual que muchas veces los datos podemos representarlos en forma de tabla, también en ciertas ocasiones podemos representarlos en arrays multidimensionales. Para la visualización, tenemos el problema de que no somos capaces de representar más de tres dimensiones, por lo que si queremos representar más dimensiones tenemos que hacer varias representaciones en un máximo de 3 a 3. También, para mayor claridad a la hora de visualizar los datos, las tablas y los arrays bidimensionales son las mejores opciones.

Así, en términos generales, el primer paso que se suele dar en la representación de los datos multidimensionales es la creación de una “fact table”, que no deja de ser una tabla donde se representan las combinaciones distintas que se pueden dar de los datos y la cantidad de observaciones que lo cumplen. Cada una de las observaciones de esta “fact table” es única, puesto que cada fila representa una combinación única.

Hacen falta dos pasos para la representación de los datos en un array multidimensional: La identificación de las dimensiones y la identificación de un atributo que sea el objetivo del análisis. Las dimensiones deberán de será atributos categóricos.

Cada combinación de esta “fact table” será una celda del array multidimensional, que contendrá como valor la cantidad de observaciones que lo cumplían. Así, podemos también decir que este valor es la cantidad de valores que estamos interesados en analizar.

1. Análisis de los datos multidimensionales

Para el análisis de los datos multidimensionales hay diferentes técnicas que se pueden usar. A continuación, analizaré cuatro técnicas muy utilizadas:

1. Data Cubes (Cubos de Datos)

La técnica de data cubes, o los cubos de datos en castellano, consiste en la aplicación de una operación aritmética de las dimensiones restantes de una determinada operación sobre unas dimensiones concretas.



2‑6. Ejemplo Data Cube

La representación multidimensional de los datos con todos los resultados de estas operaciones aritméticas se denomina data cube (o cubo de datos). A pesar de llevar el nombre de cubo, las dimensiones de esta figura no tienen por qué ser del mismo tamaño, ni tener tres dimensiones.

2. Reducción de la dimensionalidad y pivotaje

Las operaciones aritméticas que hemos visto en los data cubes hacen una reducción de la dimensionalidad, puesto que colapsan las celdas de una determinada columna en una única celda.

Si nos referimos al pivotaje, estamos hablando de reducir todas las dimensiones excepto dos de ellas mediante la agregación. Podríamos decir que nos estaríamos quedando con los totales de dos dimensiones.

3. “Slicing and Dicing”

Esta técnica consiste en dos operaciones muy simples. El “slicing” consiste en la selección de un grupo determinado de celdas de la matriz especificando un cierto valor, que se puede aplicar a una o a varias dimensiones. Si nos referidos a “dicing”, estaremos haciendo lo mismo pero, en vez de determinar un cierto valor, determinaremos un rango de valores.

4. “Roll-up and Drill-Down”

Estas técnicas se basan en el concepto de jerarquía, que se le puede aplicar a numerosos datos. Por ejemplo, una fecha se puede dividir en año, mes y día, y una localización se puede dividir en país, región, ciudad, calle.

Así, con este concepto en mente podremos hacer las operaciones de roll-up y de drill-down. La primera consiste en la agregación de todos los elementos que estén bajo un determinado nivel para conseguir un valor. Por ejemplo, la suma de todas las ventas diarias para conseguir las ventas mensuales. Por el contrario, la segunda consistiría en la subdivisión de todas las ventas mensuales por día, y esto solo podrá darse en el caso de que tengamos los datos de cada día.

Como se puede observar, con estas cuatro técnicas lo que conseguimos es la unión de datos similares bajo un determinado filtro que imponemos para juntarlos en una determinada celda, y finalmente conseguir reducciones de la dimensionalidad.

### En este trabajo: Análisis Exploratorio

### Machine Learning

A continuación, llego al núcleo de data science y de mi trabajo: El aprendizaje automático o machine learning. Este apartado es el centro debido a que las máquinas pueden aprender de los datos que se les pasa, y con ello predecir o clasificar otros datos de los cuales no aportamos la solución. Debido a ello, si los algoritmos de machine learning están bien entrenados se abre un gran abanico de posibilidades y respuestas ante las preguntas que se puedan plantear.

La clasificación se podría definir como la tarea de asignar un objeto o un conjunto de ellos a una categoría, normalmente predefinida anteriormente. Algunos ejemplos de uso de clasificación en machine learning son la clasificación de correos para la detección de spam o la clasificación de tumores a partir de imágenes, entre otras muchas.

Dentro de la clasificación podemos hacer una pequeña distinción con los algoritmos predictivos, puesto que estos algoritmos predicen a qué clase pertenecen unas observaciones desconocidas.

Las técnicas de clasificación funcionan bien con prácticamente cualquier conjunto de datos, pero hay que tener cuidado, puesto que con los datos ordinales y con las jerarquías los algoritmos de clasificación no funcionan óptimamente. Debido a esto, es importante conocer la naturaleza de los datos antes de empezar con este tipo de algoritmos, como se debería de tener hecho ya tras el análisis exploratorio de los datos.

Tras saber estos pequeños detalles, vamos a dar un paso más para irnos acercando a los algoritmos de clasificación. Es importante antes de codificar saber cómo funcionan estos algoritmos, puesto que no es banal los datos que se le deben de transferir para la creación del modelo y la posterior resolución.

#### ¿Cómo funciona un algoritmo de clasificación en machine learning?

Un algoritmo de clasificación en machine learning normalmente sigue unos pasos definidos, y que son ciertamente simples.

El primer elemento a tener en cuenta es que los datos deben de tener una dimensión (es decir, una columna) donde se indique la categoría real a la que pertenecen los mismos. Tras la obtención de esta columna, debemos de dividir el dataset en dos grupos, siendo el primer grupo para el entrenamiento del modelo y el segundo para el test del mismo. Es importante tener en cuenta que esta división no debe de ser igual, sino que el conjunto de entrenamiento debe de ser muy significativamente mayor al de test, con un ratio de aproximadamente 8 a 2, aunque esto depende de nuestros datos y se pueden conseguir resultados muy variados, como veremos posteriormente.

Ahora que tenemos el conjunto de entrenamiento y de test preparado, podemos aplicar este nuevo conjunto de datos de entrenamiento al algoritmo de clasificación que estemos usando (Random Forest, K-NN…). La unión del algoritmo y los datos es lo que conoceremos a partir de ahora como modelo, y en este caso será el modelo de entrenamiento.

Este modelo de entrenamiento se deberá poner a prueba ahora con el conjunto de test, no sin previamente haberle quitado la columna con la solución a la clasificación al conjunto de test. Tras obtener la matriz de confusión y calcular el número de aciertos en la diagonal de la misma respecto al número de fallos, podemos calcular la eficacia del modelo y, si es necesario, ajustarlo más para obtener mejores resultados en este test.

Así, la matriz de confusión es nuestra gran aliada en los métodos de clasificación y predicción para ver la eficacia del modelo. Podemos calcular el acierto y el error en tanto por uno de la siguiente manera:

#### Problemas y soluciones con los clasificadores

Los errores que se suelen cometer en un problema de clasificación se pueden dividir en dos grupos: Errores de generalización y errores de entrenamiento. El primero se puede definir como el error del modelo cuando se le aplican datos no vistos anteriormente, mientras que el segundo podríamos decir que es el número de elementos clasificados erróneamente en el entrenamiento. Es importante destacar que un error de entrenamiento alto no conlleva obligatoriamente un error de test alto, pues puede haber generalizado de una forma correcta y obtener resultados de test aceptables, mientras que un error de entrenamiento muy bajo puede ser debido a que se haya caído en overfitting o underfitting.

Se puede definir overfitting como un sobreajuste del modelo, de tal forma que no generaliza de una manera óptima, sino que está demasiado ajustado hacia el conjunto de entrenamiento, lo que conlleva una mayor tasa de fallos a la hora de clasificar otros datos no vistos previamente.

Podríamos definir underfitting como lo contrario al overfitting, que sería una generalización demasiado simple debido a que el algoritmo no ha conseguido aprender bien la estructura y relaciones de los datos.

De estos dos problemas, el más común es el overfitting, puesto que tendemos siempre a intentar mejorar el modelo y a veces esta mejora lo único que hace es empeorarlo. Debido a ello, voy a analizar a continuación algunas de las causas por las que puede haber overfitting en un modelo clasificatorio:

1. Debido a la presencia de ruido

La presencia de errores en el dataset hace que los resultados obtenidos de un árbol de clasificación se desvirtúen. Así, cuando unos datos mal clasificados se introduzcan para entrenar un modelo, ese modelo obviamente para esos casos y casos muy similares dará unos resultados erróneos cuando predigamos.

Esto dará como resultado un aumento del número de bifurcaciones del árbol, y con ello obtendremos el overfitting que se intenta evitar. Por ello, y más en datasets grandes, se suele tomar como éxito un error ciertamente pequeño, puesto que estos errores rara vez son evitables cuando se tienen grandes cantidades de datos.

1. Debido a la falta de muestras realmente representativas

Cuando se hacen clasificaciones con un número de muestras muy pequeño (como en el caso de este proyecto), la probabilidad de que aparezca overfitting es muy alta. La falta de elementos similares con características similares que pertenezcan al mismo grupo hace que el modelo generalice peor, con su correspondiente aumento del error de test.

2‑7. Underfitting, Óptimo y Overfitting

Debido a esto, debemos de buscar soluciones para evitar caer en underfitting y overfitting en cualquier clasificador. Para esto, hay 3 métodos ampliamente usados, que veremos a continuación:

1. Método Holdout

El método holdout empieza dividiendo los datos en dos conjuntos disjuntos, llamados “conjunto de entrenamiento” y “conjunto de test”. Una vez hecho esto, se crea el modelo con el conjunto de entrenamiento y se prueba su eficacia con el conjunto de test.

Este método no es en sí un método para mejorar los errores de generalización, pero sí hay que tener en cuenta la proporción en la que se dividan los datos, puesto que esto queda a juicio del experto. Si se cogen demasiados datos, podemos tener demasiado poco test y el acierto no ser del todo preciso, mientras que si cogemos pocos datos de entrenamiento podemos caer en el apartado b anterior: Falta de muestras representativas en el entrenamiento.

Por lo tanto, el método holdout es el primer paso en cualquier creación de modelo de inteligencia artificial y por ello hay que tenerlo siempre en cuenta.

1. Método Random Subsampling

El método de random subsampling (en castellano, submuestras aleatorias) simplemente consiste en la repetición del método holdout n veces para probar las mejoras que se pueden dar en el rendimiento del clasificador.

Pero random subsampling se encuentra problemas respecto al método holdout, puesto que este método no utiliza todos los datos disponibles para el entrenamiento, además de que no tiene ningún control de cuantas veces se utiliza una observación para el entrenamiento o el test, por lo que algunos se usarán para entrenar más veces que otros, y esto puede hacer que quede un modelo desigual y no entrenado óptimamente.

1. Método Cross-Validation

Ante los problemas de los métodos anteriores surge cross-validation (también conocido como X-Validation, y en castellano validación cruzada) y consiste en la utilización de cada uno de los registros el mismo número de veces para el entrenamiento, y solo una vez para el test, por lo que queda un modelo mucho más ajustado.

A continuación, debido a la importancia de este método, voy a explicar detenidamente paso por paso como este modelo funciona:

El primer paso es aplicar el método holdout; es decir, particionar los datos en un grupo de entrenamiento y un grupo de test. Se entrena el modelo y entonces cambiamos los roles de los grupos, siendo el de test el que hará el entrenamiento y el de entrenamiento el que hará de test. Esto es lo que se conoce como “2 fold cross validation”.

Si esto lo hacemos k veces, estaremos partiendo los datos en k grupos, y cambiando los roles de todos, de tal manera que todos los datos pasen una vez por ser del grupo de test, mientras que todos los datos también serán k-1 veces del grupo de test.

En la figura siguiente podemos ver un ejemplo de 5 fold cross validation con dos grupos de datos:

2‑8. Ejemplo 5-fold Cross Validation

Un caso especial de cross validation es en el que el número de grupos coincide numéricamente con la cantidad de datos que se tienen en el dataset, y este método es conocido como “leave one out”, que significa dejar uno fuera.

Este método tiene una gran ventaja en el hecho de que los grupos de test son mutuamente exclusivos, y por lo tanto cubren de una forma perfecta el dataset. Pero, por otra parte, computacionalmente este método es extremadamente costoso, especialmente en datasets ciertamente grandes, y por si eso no fuera suficiente, la varianza en las métricas de cada entrenamiento y test serán enormes. Debido a esto, este método deberá de ser usado como complementario a una validación cruzada con un k más pequeño, y sólo en datasets pequeños.

1. Bootstrap

Con los tres métodos anteriores hemos supuesto que los grupos de entrenamiento eran conjuntos de datos, aleatorios o no, sin reemplazamiento, y por ello no hay duplicidades en dicho set de datos, así como en el conjunto de test.

En el método Bootstrap esto cambia, haciendo muestras con reemplazamiento, por lo que tanto en el conjunto de entrenamiento como en el de test podemos tener muestras duplicadas.

Este método consiste en que los elementos que no se hayan elegido para el conjunto de entrenamiento se irán al conjunto de test. Viéndolo desde una perspectiva matemática, si cogemos el dataset original con N observaciones totales, la probabilidad de que una observación sea escogida para pertenecer al conjunto de entrenamiento sigue la siguiente fórmula:

Si N es suficientemente grande, llegaremos a una asíntota que nos indica que las posibilidades para cada elemento son del 63,2%. De este modo, con el método Bootstrap tendremos un conjunto de entrenamiento de alrededor del 63,2%, y un conjunto de test que sería de las observaciones restantes.

Una vez que tenemos esto, aplicamos estos datos al algoritmo para obtener el modelo, y probamos su eficacia. Podemos probar k veces la eficacia de Bootstrap haciendo la selección de los grupos y luego aplicando el modelo estas k veces.

Hay diferentes variaciones de cómo se calcula la precisión general del algoritmo. Una de las más importantes es la llamada “.632 bootstrap”, que combina las precisiones de cada conjunto de entrenamiento hecho por método Bootstrap (Ci) con la precisión de un set de test de los datos originales con todos los datos con el grupo al que pertenecen (Ct). La fórmula la expongo a continuación:

Ahora que hemos visto los principales problemas, es hora de hacer una clasificación en profundidad de los algoritmos de machine learning que existen. Hay diferentes métodos de clasificación, puesto que se puede atender al modo en el que aprenden, la estructura que siguen… pero mi clasificación será la siguiente debido a las grandes diferencias que existen entre los grupos:

* Algoritmos supervisados
* Algoritmos no supervisados
* Algoritmos de aprendizaje por refuerzo
* Algoritmos de redes neuronales y deep learning.

#### Algoritmos Supervisados

El aprendizaje supervisado comienza con un conjunto determinado de datos, y un entendimiento ciertamente profundo de la estructura de los datos. Este tipo de aprendizaje lo que busca es encontrar patrones en los datos, de tal manera que se puedan hacer procesos analíticos sobre unos datos ya etiquetados.

Estos algoritmos se entrenan usando ejemplos preprocesados, y su precisión se mide con un conjunto de test que es, como hemos visto anteriormente, excluyente respecto al conjunto de entrenamiento.

Este tipo de algoritmos se utilizan en numerosos ámbitos, como la detección de fraudes, análisis de riesgos, algoritmos de recomendación o incluso el reconocimiento de la voz.

Algunos de los algoritmos más importantes que están bajo este grupo son:

* KNN
* Árboles de Decisión
* Regresión Linear
* SVM

##### K Nearest Neighbours (KNN)

El algoritmo “k nearest neighbours”, traducido como “los k vecinos más cercanos”, es aquel que se basa en la búsqueda de atributos similares dentro del conjunto de los datos, y así poder predecir la clase a la que pertenece dicho atributo. Llevado hacia un razonamiento más humano, podríamos decir algo como: “Si se parece a un avión, es tan grande como un avión, vuela y van personas dentro, entonces es un avión”.

Este algoritmo destaca por tener la K delante, que viene a indicar el número de vecinos con los que se va a comparar la observación determinada para obtener cuál es su clase. Así, si K es igual a 1, el elemento que esté más cerca de la observación sobre la que se quiere saber la clase será la que determine la clase de la misma. En caso de que el número K sea un número mayor, la clase que posea más elementos cerca de la observación será la que determine el tipo de la observación. En caso de que haya dos o más clases en estado de empate, se resolverá de forma arbitraria.

Es importante que la búsqueda de estos K vecinos se realiza de una forma radial mediante distancias gaussianas, de tal manera que, cuanto mayor sea K, más grande será el círculo que se formará alrededor de la observación determinada para buscar los vecinos de la misma.

Como podemos ver en la figura 3-9, el elemento sobre el que estamos, representado con una estrella, ha buscado a los 3 elementos más cercanos y los ha introducido dentro de un círculo. Como se puede observar, es mayoría el número de elementos naranjas a los de azul dentro de este círculo, de tal manera que el elemento cuya clase estamos determinando tendrá inferida la clase B, correspondiente a los naranjas.

2‑9. KNN

El algoritmo funciona usando la distancia como elemento de similitud, puesto que dos elementos muy cercanos se supone que serán muy parecidos. Así, del elemento del que se quiere saber el grupo se obtiene una lista de elementos cercanos, y usando modernas técnicas de indexación las computaciones que se tienen que hacer para obtener esta lista son mucho menores.

Una vez que se tiene la lista, se clasifica en función del grupo que posea la mayoría, donde todas las observaciones de la lista tienen el mismo peso.

El algoritmo KNN tiene una serie de características propias que determinan cuando debe ser utilizado y los “peligros” que entraña, de tal forma que las expongo a continuación:

1. Este algoritmo no requiere de la construcción de un modelo

Como se explicó anteriormente, un modelo es el conjunto de un algoritmo con una serie de datos. El algoritmo KNN no crea modelo, de tal manera que no se pierde tiempo a la hora de la creación del mismo, pero a la hora de la computación del algoritmo es bastante costoso debido a la necesidad de calcular las distancias de todos los elementos a determinar con el resto de elementos.

1. Es un algoritmo que toma decisiones en zonas locales, no globalmente

Como se puede deducir, al mirar sólo a los elementos más cercanos se incurre en que se toma una decisión a nivel local, no como otros algoritmos como los árboles de decisión (que veremos posteriormente), que las toman a nivel global. De esta forma, habrá que tener cuidado con los valores de K, puesto que un valor pequeño es susceptible al ruido que pueda generar una zona y, por lo tanto, dar un valor erróneo.

Además, este modelo cae en underfitting y overfitting muy fácilmente si no se obtiene un valor de K preciso. En caso de obtener un valor demasiado pequeño, debido al ruido caeremos en overfitting, puesto que solo miraremos el elemento más cercano. Si el valor de K, por el contrario, es demasiado grande, caeremos en underfitting y el modelo se volverá demasiado simple. De este modo, como aproximación se suele aceptar que el valor de K sea la raíz cuadrada del número total de elementos a determinar.

1. KNN produce predicciones erróneas si no se hace un preprocesamiento correcto

KNN es un algoritmo muy delicado en términos de preprocesamiento, puesto que al trabajar con distancias es importante los cambios que se hagan a las medidas de los datos. Por ejemplo, si hay numerosas dimensiones cercanas a un número, y también hay otra dimensión con una variabilidad enorme, el algoritmo no funcionará bien puesto que esta última será la más influyente de todas.

De este modo, para trabajar con este algoritmo un preprocesamiento a base de centrado y escalado, y una eliminación de las columnas menos importantes escogidas con un PCA sería perfecto.

##### Árboles de Decisión

Una técnica muy utilizada en la clasificación que requiere una pequeña mención aparte es la utilización de los llamados árboles de decisión. Esta técnica es de una simpleza extrema, pero a la vez de una gran eficacia en una gran cantidad de problemas.

Los árboles de decisión recordemos que entran dentro de las técnicas de clasificación, y por lo tanto quieren hallar una respuesta a partir de unos datos previos. De este modo, un árbol de clasificación parte de un nodo raíz, que no tiene ninguna entrada, pero tiene salidas. Este nodo raíz se hace una pregunta, y según la respuesta que obtenga irá por un camino u otro. En cada uno de estos caminos seguirá haciendo preguntas, y seguirá bifurcándose por cada respuesta hasta que sea capaz de llegar a una decisión final en cada camino. Cada una de estas preguntas que se ha ido haciendo los llamaremos nodos intermedios, y las respuestas finales se denominarán como hojas, de las que por supuesto no saldrá ningún camino. Por cada nodo por el que pasan los datos, se va haciendo una criba, de tal manera que al final de cada rama a las hojas solo queda un pequeño grupo de datos que poseen numerosas cosas en común.

La construcción de los árboles de decisión, como se puede apreciar, no es demasiado sencilla a simple vista, debido a que hay que hacer las preguntas adecuadas en el momento adecuado, y en un dataset de alta dimensionalidad el número de preguntas que se pueden hacer hace que el número de árboles construibles tienda a infinito. Por ello, se han creado algunos algoritmos que construyen árboles de decisión dentro de un espacio óptimo en tiempos razonables, como el de Hunt.

Además, en la construcción se nos plantean otros interrogantes, como la elección de la pregunta adecuada o la condición de parada del algoritmo.

2‑10. Estructura básica de un árbol de decisión

Respecto a la primera, el algoritmo que se use deberá de tener un sistema implementado para la evaluación de la bonanza de cada pregunta hacia el propio algoritmo, de cara a aprender si la pregunta ha sido buena o no.

Respecto a la condición de parada del algoritmo, es obvio que es algo obligatorio ya que si no el algoritmo seguiría ejecutándose hasta que se acabaran las dimensiones sobre las que preguntar, y eso no siempre es algo positivo de cara al resultado final. Normalmente se usan criterios como que todos los elementos restantes tras las preguntas tengan el mismo valor, y ese valor será el que se utilizará como hoja final de esa rama, y como condición de parada al mismo tiempo.

A continuación, voy a hacer una explicación de cómo se puede controlar y solucionar el overfitting, puesto que es el problema más común, de tal manera que se puedan mejorar estos árboles de clasificación y obtener resultados más certeros.

1. Método de la Pre-poda

En el caso de usar este método, el algoritmo que hace crecer el árbol para antes de formar el árbol completo que encajaría perfectamente con todos los datos de entrenamiento.

Para hacer esto, se debe de poner una condición muy restrictiva para parar el algoritmo, como el aumento de una cierta impureza o esencialmente en el error de la generalización.

El problema de esta solución es que, si la restricción es demasiado restrictiva, el modelo quedará en underfitting y por lo tanto será poco certero, mientras que si la restricción es demasiado liviana el modelo caerá en overfitting y por lo tanto generalizará también mal.

2. Método de la Post-poda

Si nos decantamos por este método, se deja al algoritmo crecer hasta su máxima extensión, y tras la finalización del algoritmo empieza la poda. Esta se suele hacer cogiendo subárboles, y cambiando estos subárboles por una hoja final perteneciente al grupo que tiene a la mayoría de los individuos en ese subárbol.

Este método es el más utilizado debido a que da mejores resultados, fruto de una poda posterior donde las decisiones de donde recortar vienen dadas de un árbol completamente formado.

##### Regresión Lineal

La regresión es una técnica predictiva para estimar variables continuas. Algunos ejemplos de predicción a través del uso de regresión pueden incluir el precio de acciones en la bolsa, predicción de ventas en relación con gasto en publicidad…

Una definición más formal de regresión podría darse de la siguiente forma: Tarea de aprender una función objetivo de tal manera que al aplicar un valor x sobre la misma se obtenga correctamente un valor continuo y.

De esta manera, el objetivo que tienen los métodos de regresión es encontrar esta función objetivo que posea el mínimo error en la predicción de los datos. La función de error de la misma se suele dar de dos maneras:

Con estas ecuaciones vemos el sumatorio de la diferencia entre el valor real y el valor esperado aplicando x sobre la función objetivo. En el primer caso, ya que se pueden dar restas negativas debido a que el valor de la función objetivo sea mayor que el real, cada iteración del sumatorio se pone en valor absoluto, puesto que lo que nos interesa es la distancia de fallo.

En el caso de la segunda ecuación, al estar calculando el error cuadrático no es necesario aplicar el valor absoluto, puesto que cualquier valor elevado al cuadrado será positivo.

##### Support Vector Machines (SVM)

El algoritmo SVM es uno de los algoritmos más utilizados dentro del machine learning, puesto que tiene unas fuertes bases estadísticas y matemáticas y ha demostrado solvencia en muchísimas aplicaciones prácticas, como por ejemplo el reconocimiento de caracteres manuscritos. Para comprender cómo funciona SVM, es importante entender la idea de los hiperplanos, puesto que es el fundamento de esta técnica.

Los hiperplanos son planos infinitos en un determinado espacio. En el caso de SVM, estos hiperplanos se denominan “hiperplanos de margen máximo”. Estos hiperplanos se utilizan como soporte de las fronteras para separar dos o más muestras de datos por el grupo al que pertenecen, de tal manera que se pueda conseguir la mejor separación posible. Cada frontera, al ser infinita en una dirección, solo tendrá dos hiperplanos que la soporten, uno a cada lado.

2‑11. SVM Linealmente Separable

Si los datos son linealmente separables, habrá infinitas fronteras que los separen, puesto que por un punto pueden pasar infinitas rectas, y por lo tanto fronteras. Pero aquí se nos muestra un problema, y es que en el caso de que haya un nuevo dato, tendremos que saber cuál de todas esas infinitas fronteras será la que discierna mejor a qué grupo pertenecerá dicho punto. Aquí es donde entran en acción los hiperplanos. Como hemos visto, cada frontera tiene dos hiperplanos, que serán paralelos a la frontera, y su ubicación estará determinada por el elemento más cercano que se pueda encontrar desde la frontera hacia cada una de las clases de una forma ortogonal. De este modo, la frontera ideal será la que tenga una mayor distancia con sus hiperplanos de soporte, puesto que será la que tenga un mayor margen con cada una de las clases, y por lo tanto un menor índice de error, puesto que se hace una mejor generalización.

En caso de que la frontera deba de ser muy pequeña, normalmente se cae en un caso de overfitting, puesto que el margen de error para tomar las decisiones es pequeño y la frontera tendrá que ser más ajustada para poder tener unos hiperplanos de soporte lo más amplios posible.

Es importante a partir de ahora destacar que hay dos tipos de SVM, que son el lineal y el no lineal. En el caso del primero, se busca la frontera con el máximo margen con sus hiperplanos de apoyo, lo que hace que se le conozca también con el nombre de “clasificador de máximo margen". Si nos referimos a uno no linear, la técnica consiste en la transformación del espacio en el que se encuentran los datos, de tal forma que se pueda aplicar una frontera linear para separar las clases del problema. A continuación, expondré los diferentes casos que se pueden dar, de acuerdo con esta clasificación, y los detallaré en profundidad:

1. SVM Lineal: Caso Separable

En caso de tener un caso separable en un SVM lineal, como se ha comentado anteriormente se está en un caso de clasificador de márgenes máximos. Por ello, la frontera será una recta, y cumplirá la siguiente ecuación:

De este modo, el aprendizaje de este algoritmo será la determinación de los valores A y b de la ecuación anterior mediante los datos de entrenamiento.

1. SVM Lineal: Caso No Separable

El caso de tener unos datos no separables linealmente hace que haya que tener mucho más cuidado a la hora de elegir la frontera, puesto que a veces muchas fronteras que no incurren prácticamente en errores en el entrenamiento tienen unos márgenes muy pequeños, y generalizan francamente mal. Por ello, en estos casos hay que hacer una aproximación llamada “soft margin”, que consiste en la búsqueda de un equilibrio entre los márgenes de la frontera y el número de clasificaciones erróneas que se dan en el entrenamiento. Esto es reducible en cierto modo en la etapa previa al algoritmo, puesto que si se dividen los datos por su grupo real creando subgrupos, y se detectan los outliers, podemos simplificar este proceso de “soft margin”.

1. SVM No Lineal

Si nos encontramos en un caso de kernel no lineal, hay que hacer la transformación del espacio tal como se explicó anteriormente y como se muestra simplificadamente en la figura 2-12.

2‑12. SVM No Lineal

Así, tras la transformación del espacio, se habrá convertido un problema no lineal en un problema lineal, y se podrán aplicar las técnicas anteriormente descritas para resolver el problema.

Algunas de las características principales de SVM son:

1. Los problemas planteados con SVM son problemas de optimización, que por regla general llegan a la solución encontrando un mínimo global, mientras que otros algoritmos como las redes neuronales artificiales suelen caer en mínimos locales.
2. Al afrontar un problema con SVM, es importante decir el tipo de kernel que se usará (linear o radial), y controlar también la función de coste C.
3. SVM también funciona con datos categóricos, pero para ello hay que aplicar una binarización, que ya se ha visto anteriormente.

#### Algoritmos no Supervisados

El aprendizaje no supervisado se caracteriza por hacerse sobre un conjunto de datos sin etiquetas. Así, estos algoritmos tendrán que encontrar patrones en los datos y clasificarlos respecto a estos patrones sin ninguna intervención humana.

Estos algoritmos se utilizan en ámbitos muy diversos, como los sistemas anti-spam de los correos, reconocimiento de imágenes, obtención de información de redes sociales…

Algunos de los algoritmos más importantes que están en este grupo son:

* KMeans
* Reglas de Asociación

##### K Means

K-Means es una técnica de clustering de datos que se basa en la creación de un centroide, normalmente creado como la media de un grupo de objetos, que se aplica a objetos en un espacio n-dimensional.

El funcionamiento de K-Means es simple: Se empieza con la elección de K centroides, siendo K el número de grupos que queremos discernir. Entonces, a base de distancias gaussianas, se obtienen las distancias de cada punto con los centroides y se asignan a los diferentes grupos. Una vez hecho esto, se redefine el centroide de cada grupo en función de los objetos que estén formando en ese momento el cluster, y se vuelve a empezar. La condición de parada es la falta de elementos que cambien cualquier objeto de un cluster a otro, y por lo tanto que ninguno de los centroides tenga que cambiar su posición. Para evitar costes computacionales extras, en datasets extremadamente grandes se puede aplicar una parada cuando menos de un 1% de los puntos hagan cambios, de tal manera que no se tengan que recalcular ni las distancias ni los centroides y, por lo tanto, ahorrarnos pasadas en el bucle del algoritmo.

2‑13. K-Means paso a paso

Otra visión que se puede tener del algoritmo es la de que es un algoritmo de optimización, donde la función objetivo tiene que minimizar las distancias de los puntos con el centroide más cercano.

Es importante destacar que K-Means, como se ha comentado anteriormente, utiliza distancias euclídeas, pero que no son las únicas distancias que existen. Otra distancia que también sería compatible con este algoritmo sería la distancia de Manhattan, aunque con esta distancia en vez de usar la media para calcular los centroides se usaría la mediana. Por otra parte, la distancia de Jaccard es una distancia que se suele usar más en el análisis de documentos y la similitud entre los mismos y, por lo tanto, no es la más indicada para este algoritmo ni para este problema.

Una vez que tenemos claro que la distancia a usar en este algoritmo, y especialmente en este problema es la gaussiana, es interesante la explicación de la “suma del error cuadrado”. Esta suma consiste en la adición de las distancias de todos los puntos de un cluster con su centroide más cercano. Si esto se hace para varios clusters, el cluster más acertado será el que tenga una suma del error cuadrado menor. Al igual, si tenemos varios sets de clusters distintos, la mejor elección será la que posea la suma del error cuadrado más pequeña. La elección que se haga de los centroides al principio es vital para la suma del error cuadrado final.

La elección de los centroides iniciales es de gran importancia a la hora de iniciar el algoritmo de K-Means, puesto que las diferentes elecciones que se puedan hacer producen diferentes resultados y variar la suma del error cuadrado. Debido a esto, hay diferentes técnicas para la inicialización de estos centroides:

1. Inicialización de forma aleatoria

La inicialización de los centroides en un punto aleatorio del espacio hace que encontremos un mínimo local que pueda parecer óptimo, pero rara vez se consigue un mínimo global que sea la mejor solución del problema.

1. Sucesión de inicializaciones aleatorias

Una técnica que se suele utilizar es la inicialización del algoritmo N veces de forma aleatoria, llegando hasta el final y seleccionando los clusters con menor suma de error cuadrado. Esta técnica presenta numerosos problemas, puesto que por una parte es muy costosa computacionalmente, pero además de ello, la reinicialización del algoritmo sobre los mismos datos hace que muchos intentos sean fallidos. Por ejemplo, si pasamos como parámetro al algoritmo K = 4, y hay 4 grupos bien diferenciados, pero tres de los centroides comienzan en uno de los grupos, este grupo acabará siendo dividido y por lo tanto la formación de los clusters será errónea.

De este modo, debido a estos grandes problemas que a veces los algoritmos no son capaces de superar, dependiendo especialmente de los datos y de las necesidades, se han desarrollado otras técnicas que explicaré a continuación:

Un acercamiento que suele resultar bastante efectivo es la obtención de una muestra de puntos y hacer clustering de los mismos mediante clustering jerárquico. Tras estos primeros clusters, se pueden obtener los centroides de los mismos y aplicar K-Means desde ese punto. Esta aproximación es muy efectiva especialmente si la cantidad de elementos a hacer clustering es pequeña, y es extremadamente efectiva si además K es un valor también muy reducido respecto al número de elementos a agrupar. Esto es debido a que el clustering jerárquico es una técnica muy costosa computacionalmente, y un clustering de este tipo con una gran cantidad de datos y numerosos grupos tomaría demasiado tiempo computacional como para ser efectivo.

Otro acercamiento también sería la selección a dedo del centroide inicial, estando este situado en un punto determinado o siendo el centro de todos los puntos. Una vez hecho esto, los sucesivos centroides que se elijan tendrán que estar lo más separados posibles de este centroide primero. El problema de esta técnica es que se debe de hacer un análisis de los outliers perfecto en las etapas previas al machine learning, puesto que al elegir los elementos más separados se puede coger con gran facilidad un outlier y, por lo tanto, hacer un mal clustering. Otro problema que posee este método es que es bastante costoso computacionalmente el calcular el punto más alejado de un centroide. Debido a esto, esta técnica solo se utiliza normalmente en subconjuntos, y no en datasets enteros.

K-Means suele tener otros problemas, además de la elección del centroide inicial. En las siguientes líneas, analizaré estos problemas y las posibles soluciones que se pueden dar, o las recomendaciones a seguir a la hora de aplicar esta técnica:

1. Manejo de clusters vacíos

Uno de los problemas con los algoritmos de K-Means básicos es que se puede dar que ningún punto sea asignado a un centroide y, por lo tanto, se obtenga un cluster totalmente vacío en las etapas de asignación de puntos a centroides anteriormente vistas. Debido a esto, los algoritmos de K-Means deberán de tener una serie de políticas de reemplazamiento de centroides por otros en caso de que esto pase, porque en caso contrario la suma del error cuadrado será demasiado alta, debido a las grandes distancias que puede acabar habiendo en el resto de grupos por las malas clasificaciones.

Una aproximación que suelen hacer estos algoritmos es coger el punto más alejado, y que por tanto más suma al error cuadrado, y eliminarlo, de tal manera que ningún centroide pueda establecer allí su primera base y por lo tanto no haya posibilidad de obtener grupos vacíos. Si se mira con perspectiva, este método está haciendo una eliminación de un outlier.

1. Problemas con los outliers

Obviamente, si usamos el error cuadrado el hecho de que haya elementos outliers influirá de gran manera al resultado final. Esto se debe a que los centroides, al final del algoritmo, no estarán situados en el punto óptimo donde deberían, sino demasiado influenciados por los outliers, de tal manera que la suma del error cuadrado aumentará.

Por lo tanto, uno de los mayores problemas que tenemos a la hora de hacer clustering, y en especial con K-Means, es el problema de los outliers y su identificación. Hay muchas aproximaciones a identificar outliers, pero una de las más sencillas es la eliminación de puntos que presenten un error mucho más alto que los compañeros del cluster. También, si hay clusters especialmente pequeños, es interesante una valoración especial de si el cluster es válido, puesto que puede ser simplemente un grupo de outliers y no un grupo real válido.

Debido a esto dos problemas que nos encontramos, la búsqueda de soluciones para mejorar los algoritmos de clustering es obligatoria. Debido a esto, técnicas como el post procesado del clustering para reducir la suma del error cuadrado se plantean como técnicas interesantes a emplear.

Esta técnica, el post procesado del cluster, suele utilizarse para mejorar el clustering hecho por una primera iteración del algoritmo. Normalmente se puede mejorar el error obtenido aumentando la K, puesto que, al haber más centroides, si se inicializan de una manera correcta, los puntos estarán más cercanos a ellos y por lo tanto las distancias disminuirán. Pero normalmente no queremos subir el número de grupos que existen, por lo que debemos de empezar a pensar de una manera más global. Si no estamos conformes con el clustering que se ha hecho, es posible que K-Means haya caído en un mínimo local, lo que significa que hay una solución mejor pero que no ha sido capaz de llegar a ella. La repetición del algoritmo puede llevarnos hacia un mínimo global.

En caso de que estas técnicas anteriores no sean satisfactorias, se deberá de pensar en hacer un post procesado del clustering. Hay dos métodos:

1. Incremento del número de clusters

En caso de que no se haya encontrado un mínimo global y se quiera aumentar la precisión del algoritmo, el aumento del número de clusters, y por lo tanto de la K, es una solución. Así, se tienen dos alternativas:

1. División de clusters

En caso de la elección de la división de un cluster, se deberá de elegir obviamente el que tenga un mayor error. También, como opción alternativa, aquel que tenga una mayor varianza puede ser el que sea elegido. Una vez que tengamos elegido el cluster en cuestión, se procederá a la división del mismo en dos o más grupos, de tal manera que se obtengan grupos mucho más cohesionados y cercanos.

1. Introducción de un nuevo centroide

En caso de elegir la introducción de un nuevo centroide, la técnica que se suele escoger es la de la elección del punto más alejado de cualquier centroide de los clusters, y la introducción de un nuevo centroide en ese punto.

Debido a elementos explicados anteriormente, esta estrategia tiene dos problemas: El primero es el gran coste computacional que conlleva el cómputo del elemento más alejado de un dataset, y el segundo es la gran posibilidad de obtener un cluster muy reducido con el punto outlier que se elija, de tal manera que se debería de pensar en la eliminación de ese punto en caso de que esto pasara.

1. Decrementar el número de clusters

La reducción del número de clusters, mientras que se intenta minimizar el aumento en el error, puede seguir otras dos estrategias, que serán explicadas a continuación:

1. Dispersión de un cluster

La dispersión de un cluster consiste en la eliminación del centroide del cluster en cuestión, y la reasignación de los puntos de ese antiguo cluster a los restantes, siguiendo el mismo procedimiento que se sigue en el algoritmo mediante negociación por distancias euclídeas.

De una forma ideal, el cluster que debe de ser dispersado será aquél que aumenta el error total de una forma mínima.

1. Fusión de dos clusters

La unión de dos clusters es una buena opción en caso de tener dos clusters que sean muy unidos, puesto que aumentarán de una manera ligera el error total. También se puede hacer una computación que, aunque costosa, permite determinar qué dos clusters al unirse producirán un aumento del error total más pequeño, y actuar en consecuencia.

A continuación, explicaré algunas de las fortalezas y debilidades que tiene K-Means, puesto que al ser un algoritmo tan utilizado y conocido deberán de ser expuestas claramente:

K-means tiene grandes dificultades para obtener los clusters “naturales” que se pueden dar en la realidad, puesto que se centra principalmente, debido a sus distancias euclídeas, en clusters con un tamaño similar y una forma esférica. Además, suele tener problemas en clusters donde la densidad varía, puesto que suele poner un centroide donde haya una mayor densidad, a pesar de que ahí pueda haber dos o más grupos.

Por otra parte, K-means es un algoritmo que se puede utilizar para numerosos tipos de datos (pero no todos), además de que es bastante eficiente, especialmente cuando el número de datos a clasificar no es extremadamente alto. De esta manera, se pueden hacer varias ejecuciones para poder encontrar el mínimo global. Además, como hemos visto anteriormente, los errores por outliers son fácilmente identificables y solucionables.

##### Reglas de Asociación

Los clasificadores basados en reglas de asociación son aquellos que se basan en la clasificación de elementos usando reglas condicionales, de tal manera que siguen la estructura básica de programación if-then, que se puede ver en la forma siguiente:

If

…

Then

…

Las reglas, una vez hechas, se expresan mediante condiciones disjuntas, donde el operador ᴧ, equivalente a la conjunción “y”, delimita las reglas. Finalmente, las reglas acaban con una flecha con dirección hacia la derecha donde se muestra el resultado de la regla. En la tabla a continuación se puede ver gráficamente este formato de presentación de las reglas, donde presento 3 reglas distintas:

Tabla 2‑1. Ejemplos Reglas Finales

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Condición 1 | Operador | Condición 2 | Dirección | Consecuente |
| Es coche = T | ᴧ | Es eléctrico = F | 🡪 | Contaminante |
| Es coche = T | ᴧ | Es eléctrico = T | 🡪 | No Contaminante |
| Es coche = F |  |  | 🡪 | No es coche |

Como se puede ver en la tabla superior, si un elemento es un coche y es eléctrico las reglas lo meterán dentro de la categoría de no contaminante, mientras que, si es un coche y no es eléctrico, se introducirá dentro de la categoría de no contaminante. En cambio, si no es un coche directamente se introducirá dentro de una tercera categoría “no es coche”.

Es importante introducir que las reglas que están más a la izquierda son más determinantes a la hora de tomar las decisiones, puesto que son unos “antecedentes” o “precondiciones”, por lo que los elementos que pasen las reglas más a la izquierda serán más similares. Este sistema se parece en gran medida a los árboles de decisión (ver figura 3-10). También es importante decir que se pueden poner tantas condiciones como dimensiones tenga el problema, o incluso más, puesto que en los elementos numéricos pueden darse dos condiciones sobre una misma variable en caso de querer acotarla tanto superior como inferiormente.

Es importante definir dos términos a la hora de hablar de reglas, que son cobertura y precisión:

Cobertura se define como la fracción de los elementos en un dataset que activan una determinada regla.

Precisión, también conocido como factor de confianza, se define como la fracción de los elementos activados por una determinada regla cuyas clases sean iguales a las predichas.

En términos matemáticos, para un total D de elementos en un dataset, con una regla R, un número de elementos C que satisfacen las condiciones, y un grupo real G, se podrían definir en los siguientes términos:

Calculemos como ejemplo la cobertura y la precisión de una determinada regla sobre la ilustración 3-14. Pongamos como ejemplo la siguiente regla

en la cual estamos diciendo que si el tiempo atmosférico está lluvioso y la temperatura es fría no se jugará. Para la simplificación del ejemplo, no tendré en cuenta las condiciones de humedad ni de viento.

Ilustración ‑. Ejemplo en Inglés de tabla de reglas

En este caso, lo primero que se deberá de calcular a la cobertura, es decir, cuantas reglas respecto del total cumplen estas dos condiciones. Como se puede observar en las líneas 5 y 6, esas dos suposiciones cumplen la regla ejemplo respecto del total de 14, por lo que la cobertura será la siguiente:

A continuación, habrá que ver la precisión de la regla, es decir, de estas dos reglas qué porcentaje han cumplido la predicción:

Ya que la línea 5 afirma que se puede jugar, mientras que la línea 6 lo desmiente. Ya que mi regla de ejemplo lo desmentía, podemos solo tener un 50% de acierto.

Un clasificador en base a reglas clasifica en función de las reglas que haya cumplido un cierto elemento. De esta manera, el elemento central en estos clasificadores obviamente son las reglas, que pueden ser de diversos tipos:

* Reglas mutuamente excluyentes

En un conjunto de reglas, dos reglas serán mutuamente exclusivas si no hay dos reglas en todo el conjunto que puedan ser cumplidas por el mismo elemento del dataset.

* Reglas exhaustivas

Un conjunto de reglas es exhaustivo si hay una regla por cada combinación que se pueda dar de los atributos o dimensiones de los elementos del dataset. Eso asegura que absolutamente todos los elementos del dataset estarán cubiertos por al menos una regla.

Si se junta la propiedad de la mutua exclusividad y la exhaustividad, se obtiene un conjunto de reglas donde cada elemento del dataset está cubierto siempre y sólo por una regla. Desafortunadamente, prácticamente todas las clasificaciones por reglas no cumplen esta situación idílica.

Cuando el conjunto de reglas no es mutuamente excluyente, un elemento del dataset puede ser cubierto por varias reglas. De este modo, nos encontramos con un problema de decisión. Hay dos estrategias para resolver este conflicto:

1. Reglas ordenadas

En esta aproximación a la resolución, las reglas se ordenan de una forma decreciente atendiendo a la prioridad que tengan, que puede atender a parámetros tan diversos como la cobertura o la precisión. Una vez ordenadas, los elementos del dataset irán recorriéndolas en sentido descendente hasta encontrar la adecuada, y en ese momento el elemento quedará clasificado y no buscará más, por lo que estaremos excluyendo las reglas duplicadas menos prioritarias.

1. Reglas desordenadas

En esta segunda aproximación, un elemento del dataset puede clasificarse siguiendo varias reglas de clasificación, y el consecuente de cada regla es un voto hacia una categoría. Una vez recorridas todas las reglas, se hace el recuento de los votos hacia cada grupo, y el grupo mayoritario será el que decida la categoría del elemento.

En el caso de haber un empate, otros factores como la precisión de la regla serán vitales para atribuirle unos pesos a los votos y de esa forma desempatar la votación.

El algoritmo que más destaca en la generación de reglas es el RIPPER. Es un algoritmo que escala muy fácilmente con el aumento del dataset, y es especialmente efectivo cuando las clases son desiguales en términos de población, lo que ser bastante frecuente. También es bastante interesante a la hora de trabajar con datasets ruidosos, ya que usa un set de validación para prevenir el overfitting, por lo que se plantea como el algoritmo más preciso a la hora de trabajar con datos reales.

Para los problemas que tienen dos clases, el algoritmo RIPPER elige la clase que tiene la mayoría de elementos como clase base, y aprende desde ahí las reglas para conseguir detectar los elementos de la clase minoritaria. En el caso de que haya más de dos clases, el algoritmo RIPPER ordena las clases según la frecuencia en la que aparezcan de forma que la clase que tenga menos ocurrencias sea la primera, y la más ocurrente sea la última. Tras hacer esta primera ordenación, los elementos que pertenecen a la primera clase los etiqueta como muestras positivas, mientras que el resto de elementos pasarán a ser muestras negativas. Una vez hecho esto, el algoritmo RIPPER generará reglas para diferenciar entre los elementos positivos y los negativos. Una vez finalizada esta iteración, el algoritmo RIPPER pasará a la segunda clase, donde hará el etiquetado y la posterior generación de reglas para la diferenciación de los elementos de esa clase. El algoritmo hará sucesivas repeticiones hasta alcanzar la última clase, coincidente con la más frecuente, la cual designará como clase base.

Hay algunas características peculiares de los sistemas clasificadores basados en reglas que son importantes tener en cuenta antes de la ejecución de un modelo de este tipo:

1. Los clasificadores basados en reglas ofrecen modelos descriptivos mucho más fáciles de interpretar que otros métodos de machine learning, pero no tan buen rendimiento, quedándose en un lugar similar al árbol de decisión.
2. El coste computacional de hacer un conjunto de reglas para un dataset es muy comparable con el coste que sería el hacer un árbol de decisión del mismo dataset, puesto que un árbol de decisión, como se comentó anteriormente, está compuesto de reglas exhaustivas y exclusivas entre ellas. El problema que tienen los conjuntos de reglas es que raramente se obtienen reglas exhaustivas y exclusivas, por lo que se necesitan rehacer los parámetros de las reglas (y con ello cambiar las fronteras de decisión) para que el algoritmo funcione de una manera óptima.

#### Algoritmos Semi-Supervisados

Los algoritmos semi supervisados son un tipo de algoritmos que están cogiendo una gran fuerza hoy en día. Aunque se le puede hacer diversas aproximaciones al grupo, son ampliamente conocidos como unos algoritmos en los que solo una parte de los datos están etiquetados, mientras que otra parte, que suele ser la mayoría, no poseen etiqueta.

El procedimiento que hacen es aprender a partir de los algoritmos etiquetados, de tal manera que al pasarle los elementos no etiquetados el modelo pueda hacer predicciones.

Un ejemplo de un ámbito en el que se suele utilizar este algoritmo es un “call center” o centro de llamadas, donde se puede hacer análisis a la voz de la gente y obtener datos como su estado de ánimo o el género, e intentar inferir qué tipo de problema tiene y por lo tanto con qué interlocutor debe de ser redirigido de una manera automática.

Muchos algoritmos supervisados ya explicados pueden hacer la función como algoritmos semi-supervisados, entre los que se incluyen Supported Vector Machines (SVM) o Random Forest.

#### Algoritmos de Aprendizaje por Refuerzo

Los algoritmos de aprendizaje por refuerzo se basan en un cambio de comportamiento y filosofía respecto a los anteriores. El algoritmo va recibiendo un feedback desde la parte de analítica de datos, de tal forma que se le va guiando al usuario a la mejor solución. Como se puede observar, este algoritmo no está entrenado a la hora de que el usuario lo use, sino que va aprendiendo a base de prueba y error. Esto conlleva que una serie de errores harán al algoritmo aprender, mientras que una serie de aciertos le harán un refuerzo que le acercarán a la solución.

Como se puede observar, este tipo de algoritmos son muy parecidos a la anteriormente explicada economía de fichas, puesto que una serie de errores (castigos) harán que el algoritmo deje de seguir ese camino, mientras que una serie de aciertos (premios) harán que el algoritmo siga por ese camino, puesto que está llevando un buen camino de cara al futuro.

Este tipo de algoritmos se utilizan mucho en robótica y en los personajes de los videojuegos. Un ejemplo del segundo caso es en el que se lucha contra el personaje controlado por inteligencia artificial de cara a un objetivo, y el personaje aprende de nuestros movimientos que le perjudican para mejorar y poder conseguir el objetivo de una manera más óptima.

También se utiliza este algoritmo para los coches de conducción autónoma. En este caso, el uso del algoritmo es de una dificultad extrema, puesto que la cantidad de obstáculos que puede haber en la carretera, así como imprevistos, es altísimo. Si todos los coches fueran autónomos, mediante comunicación entre ellos sería más fácil, pero en la vida real con conducción humana el movimiento de los coches es impredecible.

Este tipo de algoritmos no son demasiado interesantes hacia mi trabajo, puesto que como he comentado, son mucho más usados en temas de robótica. Debido a esto, simplemente haré comentario teórico de los mismos a continuación.

Algunos de los algoritmos más importantes de aprendizaje por refuerzo son:

* Q-Learning
* Diferencia Temporal (TD)

##### Q-Learning

Q-Learning es considerado como uno de los algoritmos más importantes de aprendizaje por refuerzo. El algoritmo funciona de la siguiente manera:

El primer paso es la inicialización de la llamada “Tabla Q”, de la que se puede ver un ejemplo en la siguiente figura. Esta tabla consiste simplemente en una serie de celdas actualizables en cada iteración donde calculamos los beneficios futuros esperados por cada acción que se haga en la iteración. Así, esta tabla ayuda a guiar hacia la mejor acción en cada momento. Esta tabla Q se inicializará con una dimensión m x n, donde m sea el número de acciones disponibles y n sea el número de estados a los que se puede optar. Todas las celdas de la tabla empezarán inicializadas a cero.

2‑15. Ejemplo de Tabla Q

Los siguientes pasos consisten en la elección de la acción a realizar y la realización de la misma. Estos pasos se repetirán en bucle hasta que el entrenamiento del algoritmo acabe. En los primeros bucles, como toda la tabla tiene como puntuación cero, las acciones que se elijan serán totalmente aleatorias, pudiendo caer en elementos nocivos fácilmente, pero actualizando la tabla Q para no volver a caer en los mismos errores. Normalmente estos valores suelen ser negativos en caso de traer perjuicios, y positivos en caso de traer beneficios.

Una vez que el algoritmo está entrenado y la tabla Q correctamente cumplimentada, comienza la fase de evaluación, donde mediante funciones matemáticas se puede conocer la efectividad del modelo entrenado previamente.

##### Diferencia Temporal (TD)

El algoritmo de diferencia temporal es un algoritmo que, al igual que el de Q-Learning, aprende del entorno a base de sucesivas iteraciones sin tener un conocimiento previo del mismo. De esta manera, y como parece lógico, tendrá características en común con los algoritmos no supervisados.

Dentro del algoritmo de diferencia temporal, podemos distinguir 3 algoritmos:

* TD (0)
* TD (1)
* TD (*λ*)

Pero antes de explicar cada uno de estos algoritmos, es importante explicar cuatro parámetros que tienen que ser tenidos en cuenta, y son normalmente anotados con letras griegas:

* Parámetro Gamma (γ): Este parámetro se define como el ratio de descuento, y será un valor que fluctúe de 0 a 1. Cuanto más se acerque a uno, menor será el descuento que se hará en el algoritmo.
* Parámetro Lambda (λ): Lambda se define como la variable de asignación de crédito. Al igual que el parámetro anterior, fluctuará entre 0 y 1, y cuanto mayor sea el valor mayor crédito se puede asignar a acciones anteriores.
* Parámetro Alpha (α): El parámetro Alpha se puede denominar como el ratio de aprendizaje. Como los parámetros anteriores, este ratio se moverá entre 0 y 1, y un valor más cercano al 1 conllevará un ajuste muy agresivo del modelo, mientras que un valor más cercano a cero incurrirá en un modelo con un aprendizaje más conservador.
* Parámetro Delta (δ): Se define como delta cualquier cambio o diferencia de valor en el algoritmo.

Una vez que se conocen estos parámetros, se puede proceder a explicar los tres algoritmos que quedaron pendientes anteriormente: TD (0), TD (1) y TD (*λ*).

* TD (1)

El primer algoritmo que se debe de explicar es el TD (1). Este algoritmo se caracteriza por actualizar los valores de la misma manera que el método de Monte Carlo, es decir, al final de cada pasada.

Cuando se ejecuta una acción con este algoritmo, se hace una actualización a los estados previos, con un lambda correspondiente al número entre paréntesis. En este caso, al ser lambda con valor uno, el crédito que se le puede aplicar a las acciones anteriores tiene el valor extremo por encima. Esto es de vital importancia, debido a que este algoritmo funciona de una manera similar que el método de Monte Carlo; es decir, de una manera episódica que necesita un final establecido.

Este algoritmo se basa en la fórmula siguiente:

Si lo vamos interpretando parte a parte, podría leerse de la siguiente manera: Gt es la suma de los beneficios descontados. Cuando se va avanzando por el entorno, se van anotando los beneficios y los perjuicios, y todos los futuros se van multiplicando por un descuento (recordemos que estará entre 0 y 1, por lo que hará la función de descontar). Finalmente, en un futuro más lejano hay un descuento más acusado, puesto que elevamos el descuento por el número de iteraciones menos uno.

Una vez que tenemos solventada esta ecuación, hay que hacer una actualización al valor estimado, representado por V(s). Al igual que en Q-Learning, se empieza con una tabla toda inicializada a ceros o a valores aleatorios, de tal forma que los movimientos son totalmente descontrolados y erráticos al principio hasta que el algoritmo empieza a aprender.

Con esto, lo que se hará es la sustracción del Gt recién calculado menos el error estimado anterior, lo que dará un resultado conocido como “Error TD”. Esta sustracción será multiplicada por Alpha, de tal manera que se pueda ajustar el ajuste del error. Por lo tanto, esta fórmula quedaría de la siguiente manera:

Una vez efectuada esta cuenta, habremos completado la primera pasada del algoritmo o episodio.

* TD (0)

Una vez que se ha explicado TD (1), TD (0) es un algoritmo mucho más simple de entender, puesto que solo hay una diferencia entre ambos, aparte del valor de lambda.

Esta diferencia radica en la ecuación del cálculo de V(s), que es conveniente recordar que es el valor estimado. En este caso, en vez de usar Gt para el cálculo de la diferencia teniendo en cuenta todas las recompensas futuras, solo se mira la recompensa futura más inmediata, representada por Rt+1, más el descuento que hay que aplicar a esa misma iteración, representado por Vs+1. De este modo, las ecuaciones quedarán de la siguiente manera:

* TD (λ)

Este algoritmo se utiliza cuando se quieren poder actualizar valores antes del fin de ciclo (restricción del TD (1)) o cuando se quiere utilizar más de un valor futuro para la estimación (restricción del TD (0)).

Es importante destacar que hay dos implementaciones de TD (λ), una hacia delante y otra marcha atrás. Ambas implementaciones son equivalentes en efectividad, pero explicaré la implementación marcha atrás para no alargar demasiado el apartado.

En esta visión del algoritmo se actualizan los valores en cada paso. De este modo, tras cada paso se actualizan todos los valores de los pasos previos. En estos casos hay que hacer la asignación de crédito explicada anteriormente, pero no se ha determinado un valor lambda para esta función, por lo que se utiliza un valor calculado llamado “Rastreador de Elegibilidad”. Este valor guarda un registro de la cantidad de veces que el algoritmo recae en un estado determinado, y asigna crédito teniendo en cuenta el número de veces que se ha visitado ese estado y si ese estado tiene relación con llegar al estado final, con lo que se conseguirá una tabla finalmente muy precisa en la búsqueda del camino hacia este último estado.

Las ecuaciones que rigen este algoritmo son las siguientes:

#### Algoritmos de Redes Neuronales y Deep Learning

Deep Learning, o aprendizaje profundo, es una técnica de machine learning que recrea una red neuronal artificial de una serie de capas, de tal manera que el algoritmo pueda aprender de una manera iterativa.

Las redes neuronales artificiales, especialmente al principio, surgieron como un intento de copia de las redes neuronales biológicas para poder trabajar con abstracciones, al igual que la mente humana.

Una red neuronal artificial consiste en una red de nodos, llamados neuronas, que se distribuyen normalmente en un mínimo de tres capas:

1. Capa de entrada: En la que se reciben los datos.
2. Capa oculta: Capa con un número muy variable de neuronas donde los datos se modifican para el entrenamiento de la red. Esta capa es opcional.
3. Capa de salida: Capa en la que también se modifican los datos y finalmente se ofrece un resultado. En esta capa habrá tantos nodos como dimensiones tenga el resultado que se debe de obtener.

Con Deep learning nos referimos a una técnica dentro de machine learning en la que se usan redes neuronales de una forma jerárquica, donde cada red neuronal puede tener hasta millones de nodos densamente interconectados. Además, se diferencia también de las redes neuronales tradicionales en que suele tener más de una capa oculta.

Las redes neuronales se suelen utilizar en ámbitos como el reconocimiento de imágenes y la visión artificial, aunque también pueden actuar como algoritmos de regresión y clasificación.

##### Redes Neuronales

Como se ha comentado anteriormente, las redes neuronales artificiales surgieron con la intención de simular redes neuronales humanas. El cerebro humano consiste principalmente en neuronas que se intercomunican con otras neuronas mediante unos extremos llamados axones, que producen y reciben impulsos eléctricos. De una manera simplificada, podríamos decir que las neuronas humanas están conectadas a los axones de otras neuronas mediante las dendritas, que son extensiones del cuerpo de la neurona. La acción de enviar información por una dendrita hacia el axón de otra neurona es lo conocido como sinapsis.

De forma análoga, las redes neuronales artificiales tienen una serie de nodos interconectados entre sí que pasan una cierta información a otros nodos, de tal manera que se acaba llegando a una solución final.

La red neuronal más sencilla es el perceptrón. Debido a su sencillez, a continuación se explicará el funcionamiento del mismo, así el cómo los modelos pueden ser entrenados para resolver problemas de clasificación.

El perceptrón consiste en un tipo de red neuronal artificial que posee dos tipos de nodos, siendo unos de entrada y otro de salida. Los nodos de entrada sirven para representar la entrada de datos, y habrá un nodo por cada dimensión del problema. El nodo de salida obtendrá y sacará el resultado del problema. Estos nodos son los llamados neuronas.

En los perceptrones, los nodos están directamente conectados al nodo de salida mediante una especie de enlace con un peso determinado, que se podría comparar análogamente con la sinapsis. Este peso es el elemento más importante de la conexión, puesto que controla la fuerza de dicha conexión y por lo tanto su importancia en el resultado final, puesto que el nodo de salida hace la suma de cada dato de cada neurona, pero multiplicado por el peso de dicha conexión. Cuando un modelo de perceptrón se entrena, se calculan estos pesos hasta que el resultado final coincide con el que debería de obtenerse. De este modo, al introducir nuevos datos, las relaciones y sus pesos estarán ya establecidos y por lo tanto el resultado de este nuevo dato será, en condiciones óptimas, correcto. Los pesos, al iniciar el entrenamiento de la red neuronal normalmente son inicializados de manera aleatoria.

Es importante destacar que en el perceptrón no hay ninguna capa intermedia u oculta. Estas capas ocultas son otras capas que se pueden añadir a la red neuronal entre la capa de entrada y la capa de salida, y permiten hacer más operaciones de cara a la obtención del resultado final, complicando más la red y pudiendo obtener resultados óptimos de relaciones más complejas. Además, es también importante destacar que el número de funciones de activación de los nodos es muy reducido, lo cual también contribuye a que los resultados obtenidos por el perceptrón sólo sean buenos en problemas de bajísima complejidad.

En el proceso de aprendizaje o entrenamiento del perceptrón se tiene en cuenta un factor muy importante, llamado “ratio de aprendizaje”. Se podría definir como un hiperparámetro de las redes neuronales en el que se controla cuanto se están cambiando los pesos de una red neuronal en función del descenso del gradiente. En el caso de que este ratio sea pequeño, la red neuronal avanzará poco a poco, lo cual es un elemento a favor ya que se buscará con cautela un mínimo local donde converger, pero por otra parte cuanto más pequeño sea el ratio de aprendizaje más costoso computacionalmente será entrenar a la red. El ratio de aprendizaje también es ampliamente conocido con otro nombre: Decay.

Por lo tanto, en resumen, podemos decir que el cálculo de los nuevos pesos de la red neuronal atiende a la siguiente fórmula:

Continuando con la explicación, ahora el análisis se deberá de hacer sobre las redes neuronales artificiales multicapa. Estas redes son la evolución del perceptrón, debido a la falta de potencia del mismo ya que no es capaz de separar dos grupos de datos que no sean separables mediante un hiperplano.

Las redes neuronales multicapa tienen una serie de características propias:

1. La existencia de capas ocultas con nodos ocultos en ellas, al contrario del perceptrón.
2. El número de funciones de activación en este caso no se reduce sólo al signo, de tal manera que se pueden producir resultados no lineales.

2‑16. Red Neuronal Multicapa Feed Forward

Estos elementos hacen que estas redes neuronales más complejas permitan obtener resultados positivos sobre datos que poseen relaciones más complejas entre ellos, aunque computacionalmente sean más costosas. Es importante destacar que, cuantas más neuronas haya en la capa oculta, la red neuronal va a poder resolver problemas más complejos, pero a la vez hay mayores posibilidades de caer en overfitting. Debido a ello, el control del mismo con técnicas como el decay es algo que siempre se debe de tener en cuenta.

Además del hiperparámetro decay, es interesante el uso de otro hiperparámetro llamado softmax. Esta función de activación, de una manera simple, lo que hace es la transformación de los números que llegan a una cierta capa (normalmente la de salida) de la red neuronal en probabilidades, cuya suma total es igual a uno. Con ella, se consigue suavizar la posibilidad de que dos elementos puedan tener el mismo peso o pesos demasiado diferenciados, donde en este segundo en caso de equivocación el error tiende a infinito.

Suele usarse en clasificaciones, aunque uno de sus mayores problemas es el hecho de que no tiene en cuenta la posibilidad de que un elemento pueda pertenecer a dos categorías. En caso de que en el problema esto no sea así, siempre es bueno intentar resolver el problema con una red neuronal donde se tenga softmax activado para intentar mejorar los resultados en la capa de salida.

Dentro de las redes neuronales multicapa, es interesante distinguir entre dos tipos:

1. Redes Neuronales Feed Forward

Las redes neuronales feed forward, los nodos de una capa están conectados únicamente a los nodos de la siguiente capa. Esta red se puede ver de forma esquematizada en la figura 2-15.

1. Redes Neuronales Recurrentes

En las redes neuronales recurrentes, los nodos pueden estar conectados a cualquier nodo, incluyendo los de la siguiente capa, los de capas anteriores o incluso a nodos en la misma capa.

Para entender cómo funciona una red neuronal multicapa, es importante entender cómo funciona el perceptrón, puesto que el fundamento es el mismo: El cálculo de los pesos de las conexiones entre nodos para poder reducir el error al clasificar al máximo.

En la mayoría de ocasiones, el output que nos da una red neuronal multicapa no es una función lineal, y esto es debido a las funciones de activación que se hayan elegido. Esto lo que hace es que las soluciones que se puedan obtener no tengan por qué ser las globalmente óptimas, y algunas soluciones como el gradiente descendente han sido codificadas para intentar mejorar este problema de optimización.

El método del gradiente descendente puede ser usado para aprender los pesos de las capas intermedias y de salida de la red neuronal. Para los nodos que estén en la capa oculta, la computación a realizar no es para nada trivial, puesto que es difícil saber el error que generan esos nodos sin saber cual es el valor real que deberían de tener, debido a que no son la capa final. Ante este problema, se construyó una solución llamada backpropagation (en español, retro propagación), en la que se distinguen dos fases claramente diferenciadas:

1. Hacia delante

En esta primera fase, los pesos obtenidos en la iteración anterior son usados para computar el valor de salida de cada neurona en la red. Esta computación se da en orden, empezando por las neuronas de entrada y siguiendo un estricto orden hasta las neuronas de salida.

1. Hacia atrás:

En la segunda fase, y siguiendo un estricto orden desde las neuronas de salida hasta las de entrada, la fórmula de actualización de pesos se vuelve a ejecutar. Esta segunda pasada, elemento que aporta la retro propagación, ayuda a usar los errores de las neuronas en la capa n+1 para estimar los errores de las neuronas anteriores, de la capa n.

A continuación, voy a hacer un análisis de las características de las redes neuronales, así como sus ventajas y desventajas:

1. Las redes neuronales con una capa oculta al menos son unos aproximadores universales, lo que significa que se pueden utilizar para aproximar cualquier función a su óptimo. Esto las hace realmente útiles para hacer una primera aproximación a cualquier problema, pero por otra parte es frecuente caer en overfitting con ellas debido al intento de mejorar en exceso el problema.
2. Las redes neuronales artificiales son esencialmente buenas cuando se tenga la duda de si alguna variable no es demasiado interesante, puesto que ellas mismas reajustarán los pesos en función de la relación que tengan los datos, por lo que no hay que preocuparse de ello. Sin embargo, esto no indica que no se deba de hacer alguna técnica de reducción de la dimensionalidad antes, puesto que cuantas más dimensiones tenga el problema más neuronas en la capa de entrada tendrá que haber y mayor coste computacional habrá en el problema.
3. Las redes neuronales son muy sensibles hacia la presencia de ruido, especialmente en el set de datos de entrenamiento. Debido a esto, el planteamiento de soluciones es algo importante, donde se puede introducir un decay para reducir el overfitting o usar un set de test para comprobar el estado de la red neuronal tras el entrenamiento.
4. El método del gradiente descendente normalmente dirige el modelo hacia un mínimo local. Esto deberá ser tomado en cuenta siempre, y entrenar el algoritmo sucesivas veces comprobando el mejor resultado en el test siempre es una buena práctica en caso de ser computacionalmente posible.
5. En relación con el apartado anterior, el entrenamiento de una red neuronal artificial puede conllevar un gran tiempo de procesamiento, especialmente cuando la cantidad de datos y/o dimensiones es muy elevada. Esto también puede verse agravado por el número de nodos intermedios, puesto que se tendrán que hacer más cálculos por cada pasada en la red neuronal.

##### Deep Learning

### Visualización

# Capítulo 3: Resultados Obtenidos y Conclusiones Finales

## Resultados Obtenidos

## Conclusiones

## Líneas Futuras, Ampliaciones y Entornos de Aplicación

Ejemplo de inserción de referencias bibliográficas (1). Más referencias (2) (3).

Para insertar una referencia bibliográfica utilizar la opción “Referencias / Insertar Cita”. Como formato se ha utilizado el Estilo ISO 690- I, que identifica las referencias bibliográficas mediante números.

La sección de “Bibliografía” se actualiza con las nuevas citas con la opción “Actualizar citas y Bibliografía

# Anexo de Términos

Business Intelligence

Dataframe

Dataset

Discretización

Episodio Afectivo

Framework

Función de activación

Gradiente (Red Neuronal)

IMC

Intervalo

KDD

Monte-Carlo (Método)

Modelo (Machine Learning)

Outlier

Patrón

Sistema Gestor de Base de Datos

TDAH

Bibliografía

1. **Beriso Gómez-Escalonilla, Á., Plans Beriso, B., Sánchez-Guerra Roig, M. y Sánchez Peláez, D.** (2003). *Cuaderno de Terapia Cognitivo-Conductual (Una orientación pedagógica e integradora).* Madrid: EOS.

2. **Burns, D.** (1980). *Sentirse Bien*. Barcelona: Editorial Paidós.

3. **Morris, C. y Maisto, A.** (2005). *Introducción a la Psicología*. México: Prentice Hall.

4. **González Muñoz, M.M.** (2010). *Estrategias metodológicas para el desarrollo de las habilidades sociales en el ámbito educativo*. Salamanca: JetPrint.

5. **American Psychiatric Association.** (2002). *Manual Diagnóstico y Estadístico de los Trastornos Mentales DSM-IV-TR.* Barcelona: Masson.

6. **Tan, P., Steinbach, M. y Kumar, V.** (2006). *Introduction to Data Mining.* Boston: Pearson.

7. **Hurwitz, J. y Kirsch, D.** (2018). *Machine Learning.* Hoboken: John Wiley and Sons.

8. Monografías.com [En Línea] https://www.monografias.com/trabajos90/la-psicologia-cognitiva/la-psicologia-cognitiva.shtml

9. Universidad de Barcelona [En Línea] http://www.ub.edu/dppsed/fvillar/principal/pdf/proyecto/cap\_06\_proc\_info.pdf

10. Psicología y Mente [En Línea] https://psicologiaymente.com/psicologia/conductismo

11. Slideshare [En Línea] https://www.slideshare.net/Arlinzon/enfoque-cognitivo-conductual-historia-de-la-psicologia

12. Universidad de Alicante [En Línea] https://rua.ua.es/dspace/bitstream/10045/3834/29/TEMA%205\_PROCESOS%20PSICOL%C3%93GICOS%20BASICOS.pdf

13. Organización Mundial de la Salud [En Línea] https://www.who.int/mediacentre/factsheets/fs396/es/

14. Inside Big Data [En Línea] https://insidebigdata.com/2014/11/09/ask-data-scientist-importance-exploratory-data-analysis/

15. Aukera [En Línea] https://aukera.es/blog/data-science-que-es-y-que-no-es/

16. IBM [En Línea] https://www.ibm.com/support/knowledgecenter/en/SSEPGG\_10.1.0/com.ibm.datatools.datamining.doc/c\_dp\_datapreparationoverview.html

17. Medium.com [En Línea] https://medium.freecodecamp.org/an-introduction-to-q-learning-reinforcement-learning-14ac0b4493cc

18.Medium.com [En Línea] https://medium.com/@violante.andre/simple-reinforcement-learning-temporal-difference-learning-e883ea0d65b0

Agradecimientos