Introducción al aprendizaje automático y la minería de datos

Master en Business Intelligence y Big Data (Tenerife)

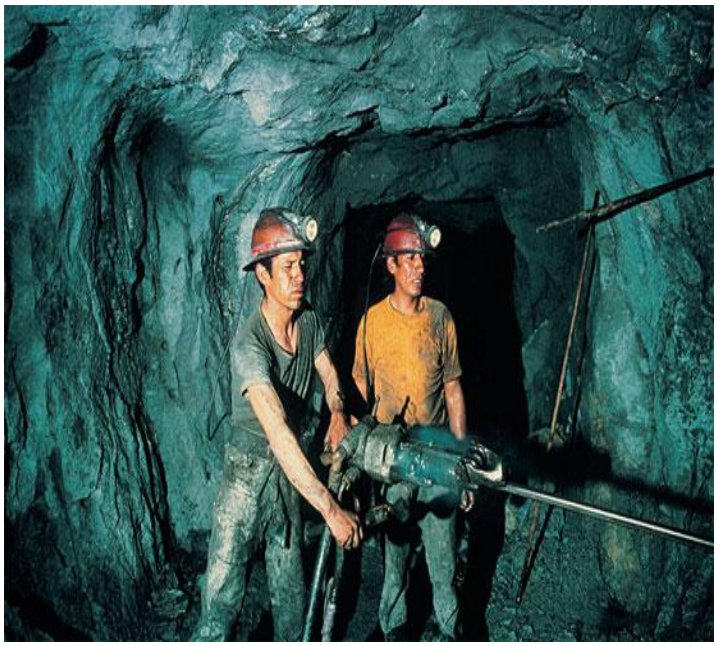
2018 - 2019

PROFESOR/A
Guillermo González Sánchez
Científico de datos en Strategy Big Data
Licenciado en Matematicas

1. Conceptos generales

*El **aprendizaje automático** es una rama de la inteligencia artificial, como es lógico, dado que el aprendizaje es uno de los procesos fundamentales para los seres inteligentes. Concretamente, es la disciplina se ocupa de desarrollar técnicas para que las máquinas "aprendan" a partir de conjuntos de datos. A estos datos se les llaman ejemplos o instancias. La idea general es la de extraer algún tipo de **modelo** a partir de los datos. Ese modelo es "lo aprendido" y nos sirve bien para clasificar o predecir, o bien para conocer alguna característica de los datos que antes no conocíamos.*

El campo del aprendizaje automático es muy amplio, y abarca desde la traducción automática (machine translation) hasta la visión artificial que se utiliza en robots industriales o, más recientemente, en drones. No obstante, aquí nos centramos solamente en problemas de aprendizaje automático que tengan un contexto de negocio o sirvan para resolver problemas u obtener nueva información en ese contexto.



*La **minería de datos** es un conjunto de actividades orientado a descubrir conocimiento (patrones, relaciones, hechos, etc.) en bases de datos típicamente de gran volumen. La idea de la "minería" es una metáfora para resaltar el hecho de que se "encuentran" cosas "valiosas" mediante un proceso de prospección, como en las minas de minerales.*

La analogía tiene implicaciones interesantes, entre otras:

- *La minería comienza con algún tipo de objetivo o hipótesis sobre la existencia de minerales valiosos en una zona. En la minería de datos, también hay algunas ideas de negocio o hipótesis iniciales.*
- *La minería la hacen los mineros, pero se sirven de diferentes herramientas, cada vez más sofisticadas. En la minería de datos, hay muchas de estas herramientas, **incluyendo las de aprendizaje automático**.*
- *La minería a veces tiene que excavar túneles alternativos. En la minería de datos, también hay un proceso de ensayo y error.*
- *En la minería es necesario extraer muestras de mineral y analizar su calidad. En la minería de datos, también es fundamental evaluar la calidad del conocimiento extraído.*

La minería de datos incluye por tanto muchas actividades previas al trabajo de extracción de patrones, incluyendo las de data warehousing, procesamiento de datos, obtención de datos de fuentes externas, etc. Aquí nos centramos en las tareas de aprendizaje automático como las herramientas del minero cuando está ya tratando con un conjunto de datos preparado y trata de extraer conocimiento valioso.
No obstante, es importante resaltar que en la minería de datos se necesitan dos tipos de competencias:

- ***Conocimiento del dominio**. Hay muchos hechos y teorías que son conocidas de manera general o por los expertos que sirven como guía para el proceso de minería. Por ejemplo, el que las ventas en un sector dependan de algunas variables macroeconómicas.*

- ***Conocimientos técnicos**. Son los conocimientos en sí de las tareas técnicas de la minería de datos, que incluyen el conocimiento de las técnicas de aprendizaje automático.*

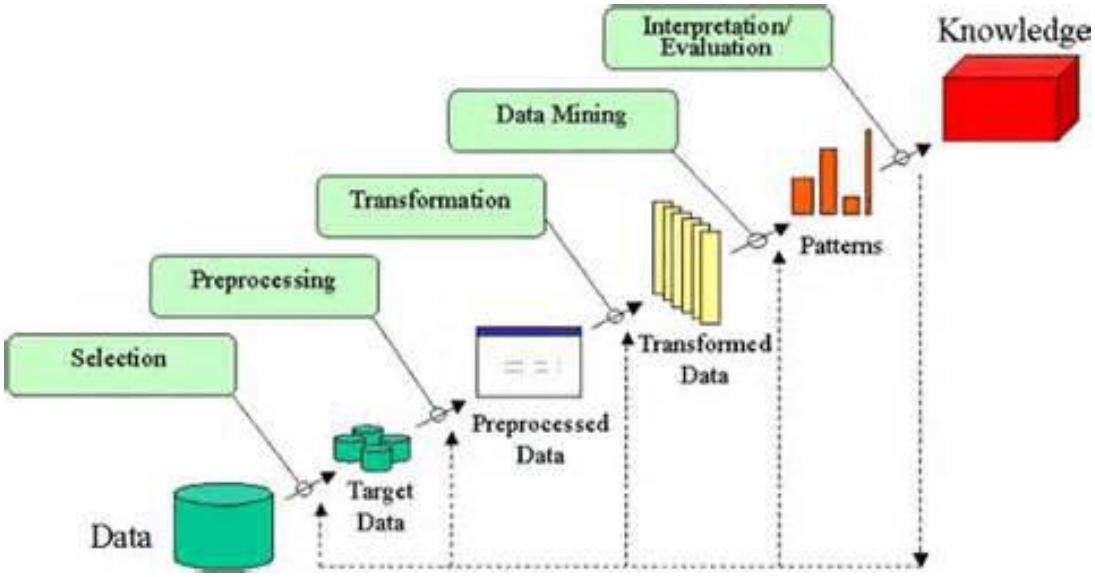
*Es importante también resaltar que la minería de datos en la mayoría de los casos es un **proceso iterativo**, que requiere una evaluación rigurosa antes de que se utilicen los modelos resultantes en la operación diaria. Por ejemplo, no se debe utilizar un modelo de predicción de fraude sin evaluar cuidadosamente su precisión, dado que si genera muchos falsos positivos, podría estar discriminando muchos clientes valiosos para el negocio.*

1.1. El proceso de minería de datos

*La disciplina del **"Descubrimiento de Conocimiento en Bases de Datos"** ("Knowledge Discovery in Databases" process, KDD) es el marco de actividades donde se encuadra la minería de datos y el aprendizaje automático.*
Un proceso de KDD genérico normalmente incluye las siguientes actividades:

1. *Determinar las fuentes de información que pueden ser útiles y dónde conseguirlas.*
2. *Diseñar el esquema de un almacén de datos (data warehouse) que consiga unificar de manera operativa toda la información recogida.*
3. *Implantación del almacén de datos que permita la "navegación" y visualización previa de sus datos.*
4. *Selección, limpieza y transformación de los datos.*
5. *Seleccionar y aplicar el método de minería de datos apropiado, que servirá para obtener patrones de los datos.*
6. *Evaluación, interpretación, transformación y representación de los patrones extraídos.*
7. *Comunicación y uso del nuevo conocimiento.*

La siguiente Figura esquematiza las actividades de KDD como un ciclo.



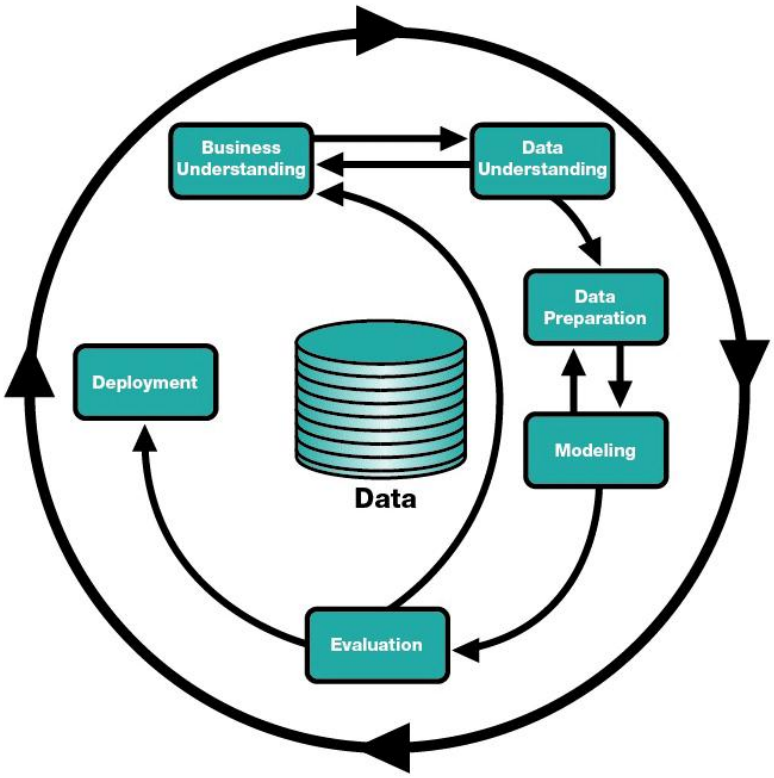
Como podemos ver, KDD es una disciplina que aplica actividades sistemáticas a descubrir conocimiento nuevo. KDD hace énfasis en la preparación de los datos en un almacén común, y el uso posterior de una fase de minería de datos.

1.1.1. Las actividades de la minería de datos

Si entendemos la minería de datos en sentido amplio, es decir, incluyendo el conjunto de actividades que normalmente se denominan KDD, tenemos un conjunto de actividades y tareas muy diversas, algunas relacionadas con comprender y hacer hipótesis sobre el negocio, y otras relacionadas con la selección, uso y evaluación técnica de tareas de aprendizaje automático. También incluyen tareas de limpieza, transformación y selección de datos.

*Para tener una visión más general de las actividades de minería, lo mejor es mirar a los estándares. Aunque no es un estándar de iure, el modelo **Cross Industry Standard Process for Data Mining (CRISP-DM)** es probablemente el más completo y amplio, así como el más utilizado.*

La siguiente Figura resume las fases de este modelo. Si hacemos un paralelo con las del proceso de KDD, será fácil encontrar analogías entre ellas.



1.1.1. Minería de datos y data science

*La minería de datos entendida de manera amplia como proceso de descubrimiento tiene similitudes con otros conceptos. Recientemente, con el énfasis en los datos que ha traído el Big Data, se ha popularizado el concepto de "científico de datos" (**data scientist**). En muchos aspectos, un data scientist dedicará parte de su tiempo a la minería de datos, aunque en otras ocasiones dedique su tiempo a tareas de análisis que no pueden clasificarse como minería de datos, sino que caen en la categoría de análisis estadístico más tradicional.*

Es difícil trazar una línea entre la minería de datos y el trabajo del data scientist. No obstante, el elemento que diferencia cuándo se hace minería está en la búsqueda de nuevos patrones (modelos) en los datos. Un análisis

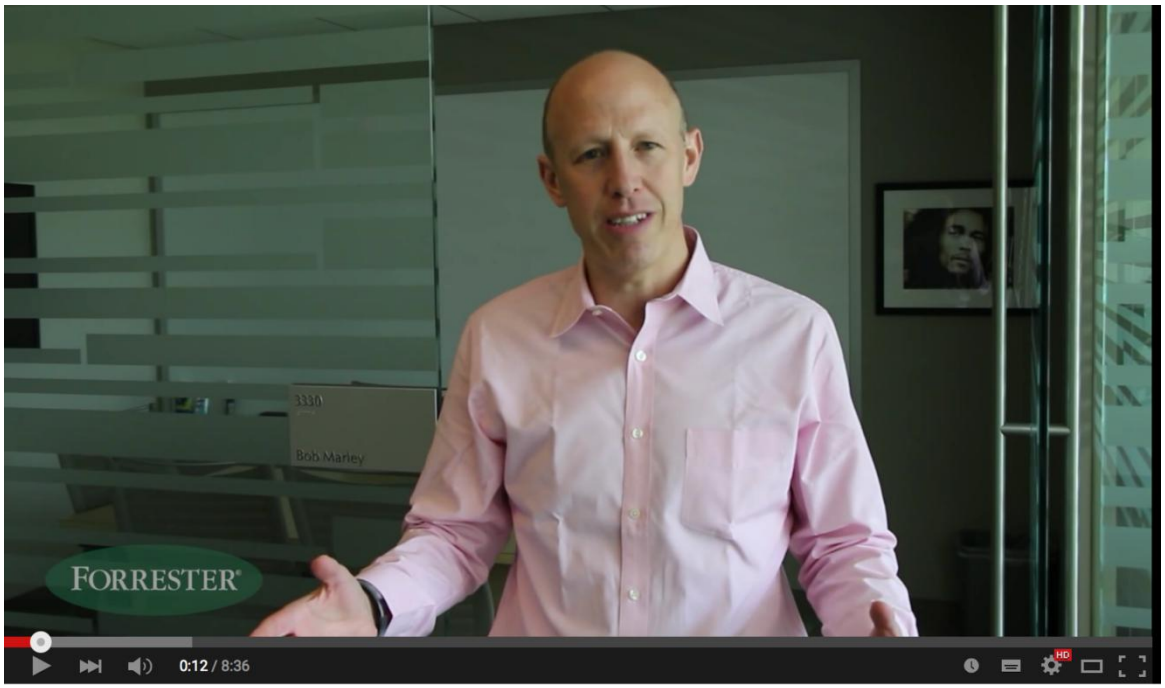
estadístico descriptivo o un contraste de hipótesis tradicional, por ejemplo, no encajarían en esa definición.

Data science es la práctica de la extracción de conocimiento generalizable a partir de los datos. Además de incorporar las técnicas y métodos del trabajo de la investigación científica, es intensiva en procesamiento estadístico, reconocimiento de patrones, visualización y modelización de la incertidumbre, entre otras técnicas.

*El "científico de datos" normalmente trabaja sobre algún tipo de entorno computacional como el entorno **R** o las bibliotecas científicas **ScyPy**, por mencionar dos ejemplos. Estos entornos proporcionan lenguajes de programación adaptados o extendidos para el trabajo estadístico, y un amplio abanico de algoritmos y técnicas de visualización para trabajar de forma interactiva sobre los datos.*

También en ocasiones el científico de datos trabaja sobre una infraestructura de procesamiento de "Big Data", o sobre un almacén de datos (data warehouse), pero no en todos los casos.
En el siguiente video, Mike Gualtieri explica el rol profesional del data scientist y otros roles profesionales relacionados pero diferentes.

https://www.youtube.com/watch?v=iQBat7e0MQs



1.2. Concepto de aprendizaje automático

El aprendizaje automático es una disciplina dentro de la Inteligencia Artificial. En este sentido, es un área donde investigadores e ingenieros trabajan en el desarrollo de modelos que permitan aprender de los datos.

Pero también es el resultado de esa labor, es decir, un conjunto de técnicas algorítmicas que están preparadas para que los analistas de datos o data scientists las utilicen, aún sin conocer en profundidad cómo funcionan internamente, es decir, como "cajas negras". Nosotros tratamos con el aprendizaje automático de esta segunda forma, dado que ***lo que nos interesa es aplicarlos dentro de un ciclo de minería de datos****, con objetivos de negocio claramente definidos.*

Hay muchas implementaciones de algoritmos de aprendizaje automático, en diferentes lenguajes de programación. Nosotros aquí vamos a utilizar como

ejemplo la implementación en Python. Concretamente, la biblioteca que los reúne se denomina scikit-learn:

http://scikit-learn.org/stable/

Aprenderemos a utilizarla de manera básica, para tener un contacto directo con los conceptos más importantes del uso de los algoritmos y modelos resultantes.

Es importante resaltar que a la hora de hacer aprendizaje automático, lo más complicado es encontrar el "algoritmo correcto". De hecho, es habitual probar con diferentes algoritmos para el mismo problema, y seleccionar el que de mejores resultados.

1.2.1. Un ejemplo gráfico

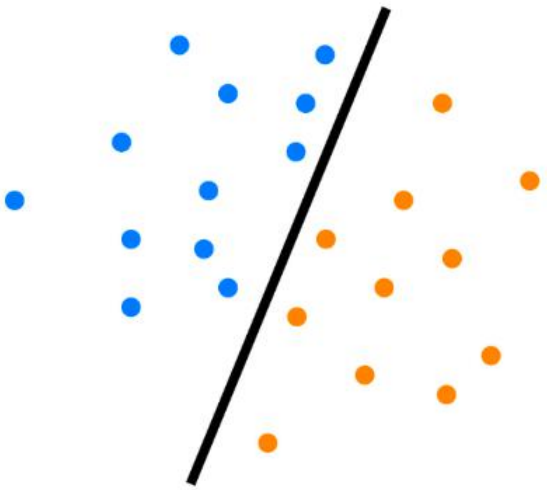
Muchos algoritmos de aprendizaje automático se utilizan para clasificar. Es decir, dada una población de instancias (clientes, cuentas, etc.) se quieren dividir en dos o más clases disjuntas.
Consideremos el siguiente ejemplo (es un ejemplo muy simplista y completamente ficticio):

- *Tenemos tres compañías donde quieren clasificar a sus clientes.*
- *Queremos clasificar a los clientes de acuerdo a los que son rentables y los que no, y tenemos una base de datos de 1000 clientes que ya están etiquetados, es decir, hemos determinado cuáles son rentables y cuáles no.*
- *Hay dos variables que los describen: tiempo como clientes y edad, y que hemos determinado que son interesantes.*

Podríamos tener varias situaciones al tratar de visualizar el conjunto de datos.

- ***Caso 1***

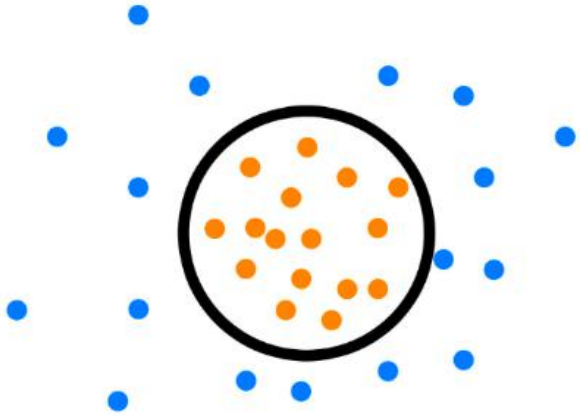
En el primer caso, si dibujamos los clientes como puntos, con el tiempo como clientes en la X y la edad en la Y, y pintamos de dos colores diferentes los clientes rentables y los que no lo son, nos encontramos con lo siguiente:



Aquí hemos trazado una línea recta que separa claramente las dos clases. Este es un caso en que un clasificador lineal fácilmente encontrará los parámetros de la función de esa línea recta, y tendrá además una separación prácticamente perfecta. Este es un caso claro en el que el aprendizaje automático se ajusta perfectamente a lo que queremos.

- ***Caso 2***

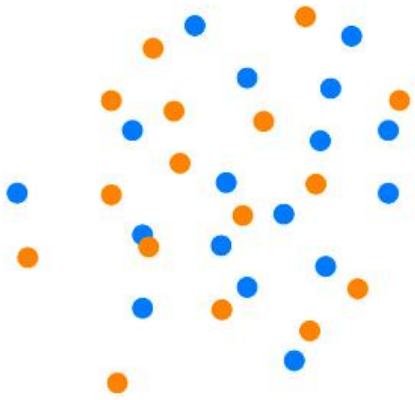
En el segundo caso, hacemos el mismo dibujo, pero los datos de la segunda compañía, al visualizarse, dan un panorama muy diferente.



Vemos que en este caso, no se puede separar bien a los puntos con una línea recta, sin embargo, sí se podría trazar otra forma, en este caso un círculo, que puede hacerlo. Este es un caso donde claramente las clases son separables con esas variables, pero un modelo lineal no será capaz de hacerlo.

- ***Caso 3***

En la tercera empresa, la visualización que obtenemos es la siguiente.



Este es un caso donde no parece que las clases puedan separarse claramente, con las variables consideradas. Esto nos indica que quizá esas variables no sean las relevantes para clasificar a los clientes en esta empresa. Aunque un

algoritmo de aprendizaje automático tratará de obtener un modelo para estos datos, su calidad será muy baja, y habrá que descartarlo.
En conclusión, los datos, sus distribuciones y la selección de variables son fundamentales para obtener buenos modelos con el aprendizaje automático. Además, aun cuando las variables sean relevantes, es importante seleccionar bien el algoritmo, como vimos en la diferencia de los casos 1 y 2.

1.2.2. Aprendizaje automático y Big Data

El aprendizaje automático está en el centro de las soluciones analíticas de Big Data. Se le ha ***denominado el "héroe anónimo" de estas soluciones****. Hay quien sostiene que el* ***aprendizaje automático es la clave del Big Data****. Para entender la relación entre ambos conceptos, es importante comprender el contexto de incremento de los datos disponibles, y cómo la tecnología se adapta a esos grandes volúmenes.*

- ***El incremento de la producción de datos en el mundo digital***

La tecnología e Internet ha hecho que cada vez las empresas y administraciones puedan tener más datos de clientes, usuarios y ciudadanos. Esta es la base de la hipótesis de que Big Data ofrece nuevas oportunidades a las técnicas ya clásicas del aprendizaje automático. No obstante, más datos no quiere decir necesariamente mejores resultados del aprendizaje, aunque sí aumenta significativamente las oportunidades para obtener buenos resultados.

El siguiente video muestra el impacto que la acumulación de datos sobre nuestro comportamiento puede tener, que es paralelo a las oportunidades que ofrece a las organizaciones para sus procesos de minería de datos.

https://www.youtube.com/watch?v=jlzH1_tid1U

- ***Aprendizaje automático y Big Data***

Nosotros podemos utilizar aprendizaje automático en nuestro portátil si los datos no son muchos, pero cuando se trata de Big Data, es necesario utilizar sistemas en la nube (o clusters privados en nuestra organización, si los tiene). Un ejemplo de estos entornos para machine learning en la nube es ***Microsoft Azure ML****. En estas soluciones, lo único que necesitamos es un navegador de Internet, y todo el procesamiento se hace en la nube. No obstante, los pasos, algoritmos y técnicas son exactamente los mismos que si lo hacemos en nuestro laptop.*

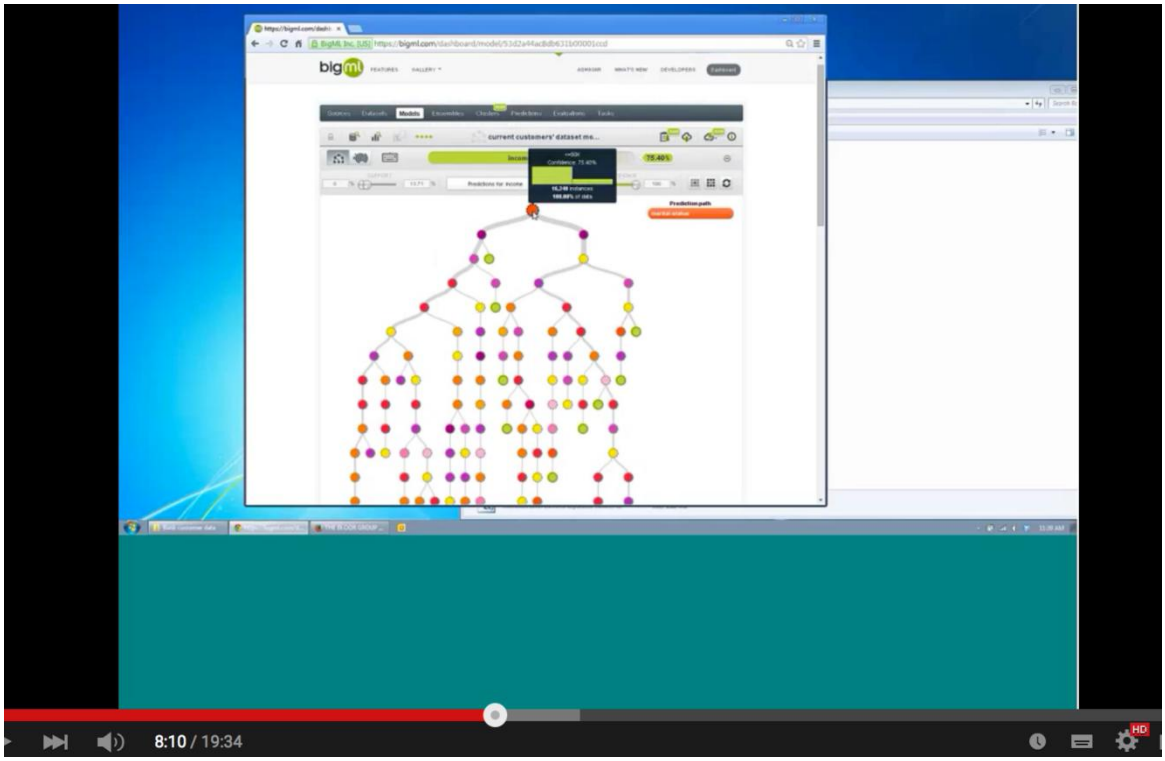
Existen también implementaciones escalables y paralelas de ciertos algoritmos de aprendizaje automático. Un ejemplo es la biblioteca ***MLib de Apache Spark****. En este caso, los algoritmos se han re-implementado para poder ejecutarse de manera paralela en múltiples máquinas.*

1.2.3. Un caso de aprendizaje automático en la nube

El siguiente video muestra una demostración de un servicio de aprendizaje automático en la nube, de la empresa BigML.

BigML es una aplicación en la nube, los usuarios se registran y suben conjuntos de datos a la nube (o utilizan los que ya están allí) y después desde su navegador aplican diferentes algoritmos a los datos, obteniendo directamente visualizaciones, sin necesidad por tanto de tener máquinas potentes por parte del usuario, BigML utiliza la nube de Amazon para eso.

Este es un caso que nos muestra las posibilidades de la nube y tecnologías de Big Data.
https://www.youtube.com/watch?v=SgUAyL5JAzE



Se aconseja ver este vídeo al principio y al final del módulo, dado que al final del módulo se entenderán mejor algunas cosas. También se recomienda registrarse en BigML y navegar sobre ejemplos que ya están hechos allí.

En ambos casos, se propone contestar a las siguientes preguntas:

- *¿Desde dónde puedo subir datos a BigML para analizarlos? ¿Puedo encontrar datos subidos por otros usuarios?*
- *Da ejemplos de tareas de transformación de datos que pueden hacerse con la interfaz de BigML.*
- *¿Qué tipos de datos estadísticos básicos nos da BigML sobre los datasets?*
- *¿Es capaz de sugerir que atributos deberíamos o no incluir en nuestro modelo?*
- *¿En qué formatos permite exportar los resultados del modelo predictivo? ¿son esos resultados directamente ejecutables?*
- *[avanzado] En la presentación se mencionan los "ensembles", investiga qué son y por qué puede mejorar la predicción.*

1.3. Un modelo predictivo. Regresión.

*Los **modelos predictivos** son aquellos que permiten determinar alguna característica (por ejemplo, la rentabilidad esperada) de un nuevo individuo (por ejemplo, de un cliente prospectivo) con un cierto nivel de seguridad. La metáfora de la "predicción" se utiliza porque se puede "adivinar" por ejemplo, si un cliente está a punto de abandonarme e irse a otra compañía.*

Un modelo predictivo tiene un valor muy grande para la empresa, porque le permite tomar decisiones y aporta datos nuevos valiosos.

Muchos de los algoritmos de aprendizaje automático obtienen modelos que (después de evaluarse rigurosamente pueden utilizarse como modelos predictivos.

1.3.1. Preguntas básicas sobre la analítica predictiva

*La **analítica predictiva** (predictive analytics) utiliza modelos predictivos para poder tomar decisiones para cada individuo (en este sentido se diferencia de predicciones más generales que caen en la categoría de forecasting).*

En el siguiente video, Eric Siegel contesta a una serie de preguntas clave sobre la analítica predictiva:

1. *What is predictive analytics?*
2. *Why is predictive analytics important?*
3. *Isn't prediction impossible?*
4. Is predictive analytics a big data thing?
5. Did Nate Silver use predictive analytics to forecast Obama's elections?
6. Does predictive analytics invade privacy?
7. What are the hottest trends in predictive analytics?
8. What is the coolest thing predictive analytics has done?

https://www.youtube.com/watch?v=m30LxzzbRik

Después de ver el vídeo, trata de contestar con tus propias palabras a las ocho preguntas anteriores. Particularmente, trata de definir "uplift modeling" y "ensemble model".

1.3.2. Una tipología por niveles de estudios

En el terreno de data science se suelen mencionar seis tipos de estudios, de los más simples a los más complejos, concretamente los siguientes:

- *Descriptivos*
- *Exploratorios*
- *Inferenciales*
- *Predictivos*

- *Causales*
- *Mecanísticos.*

La aplicación de aprendizaje automático normalmente comienza en el nivel predictivo, dado que los estudios solamente exploratorios o descriptivos se suelen considerar como estudios previos a la investigación en sí.

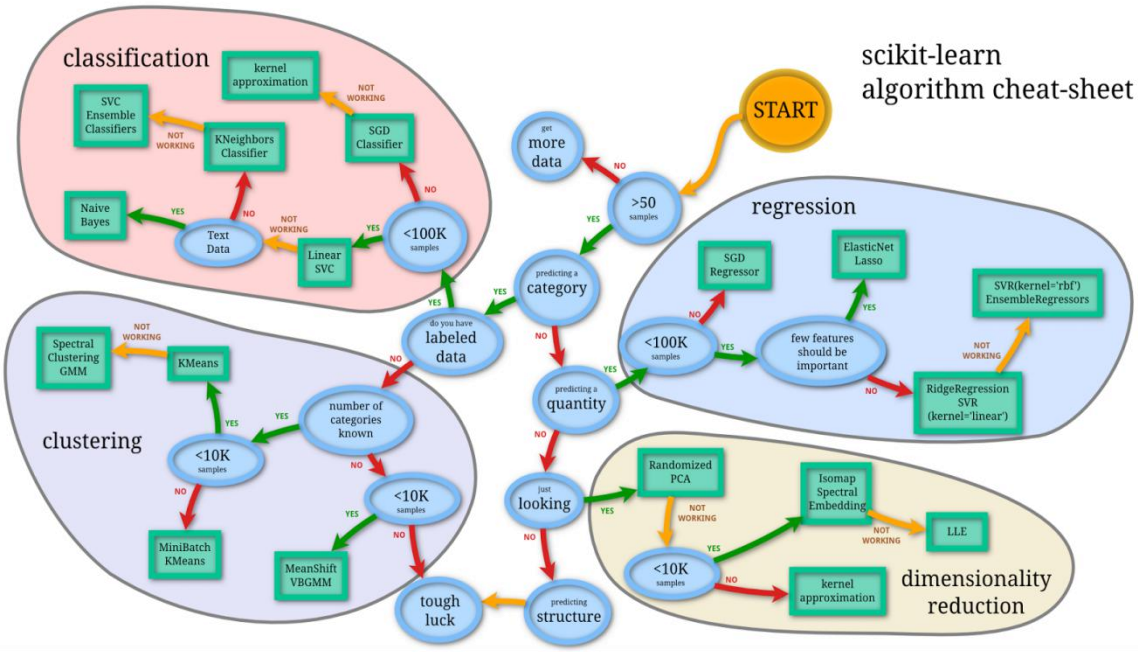
Es importante entender cada uno de estos niveles de análisis y en qué grado se pueden aplicar a diferentes contextos. Aquí hay una buena descripción de los mismos:

http://datascientistinsights.com/2013/01/29/six-types-of-analyses-every-data-scientist-should-know/

1.3.3. Tipología de técnicas de aprendizaje automático

Hay muchas formas de clasificar los algoritmos de minería de datos, más allá de las diferencias entre aprendizaje supervisado y no supervisado que son la categorización más clara.

Lo más interesante es encontrar tipologías que nos ayuden en la tarea de selección del algoritmo o conjunto de algoritmos para un problema o situación dado. En el framework scikit-learn que utilizamos aquí, se ha hecho muy popular la "chuleta" que se muestra en la siguiente imagen, que nos proporciona una guía (no muy precisa en todos los casos) para hacer la selección.



Por ejemplo, si miramos el nodo del centro (¿tienes datos etiquetados?) vemos que nos envía a las categorías de "classification" y "clustering", que son precisamente de aprendizaje supervisado y no supervisado. Como se ve más adelante, esa es la diferencia fundamental en esas dos categorías de algoritmos. También es interesante el nodo que indica ¿prediciendo una cantidad? que nos lleva a los modelos de regresión.

Si bien esta "chuleta" no cubre todas las técnicas y algoritmos y es en algunas partes algo imprecisa, nos puede servir como una primera guía para más adelante profundizar.

1.3.4. Un ejemplo de regresión

Los modelos de regresión se utilizan para predecir cantidades. El algoritmo de regresión que conoce cualquiera que haya atendido a un curso de estadística es el método de los mínimos cuadrados. Vamos a ver un caso en detalle de manera práctica, utilizando un Notebook de IPython, que explica paso a paso los elementos fundamentales.

En la carpeta de contenidos de este tema encontraréis una exportación del Notebook a PDF. Para entender el proceso y practicar con los Notebooks, podéis repetir el análisis en vuestra máquina.

1.4. Aprendizaje supervisado y no supervisado

*En el **aprendizaje supervisado**, nuestros datos están etiquetados, es decir, cada instancia de nuestro conjunto de datos tiene un atributo que es el que clasifica a esa instancia en dos o más clases. Por ejemplo, si tenemos un conjunto de datos de hipotecas, podríamos querer tener un modelo del imapago de las mismas. Entonces tendríamos en nuestro conjunto de datos:*

- *Atributos de entrada: podrían ser en el ejemplo el principal de la hipoteca, el tipo de interés, la edad del cliente, la profesión, etc.*
- *Un atributo que es la "salida", etiqueta u objetivo, que en este caso sería el campo que indicase si ese cliente impagó o no la hipoteca.*

Así, los algoritmos "aprenden" tratando de buscar diferencias entre las características de las instancias de clientes que impagaron y los que no. Lo fundamental de estas técnicas es que los datos tienen que venir previamente clasificados.

*En el caso del **aprendizaje no supervisado**, no se tiene (o no se considera) esa etiqueta que clasifica a priori a las instancias. Por ejemplo, podríamos querer simplemente buscar segmentos de clientes, es decir, buscar grupos de clientes homogéneos para crear productos personalizados, pero no sabemos a priori cuántos grupos habrá ni sus características. Este es un ejemplo de "agrupamiento" (clustering).*

Es importante entender que el uso de aprendizaje supervisado o no supervisado depende de los objetivos de nuestro estudio, y no solo del dataset. Por ejemplo, con el dataset de impago de hipotecas antes mencionado, podría

también utilizarse para un propósito de agrupamiento, excluyendo por ejemplo los clientes que impagaron de los datos. Aunque los datos utilizados vienen del mismo dataset, el propósito es completamente diferente.

1.4.1. Un ejemplo de análisis supervisado

- ***Tres tipos de aprendizaje supervisado***

Hay muchos tipos de aprendizaje supervisado, pero las siguientes son tres clases interesantes porque se aplican a categorías de problemas diferentes:

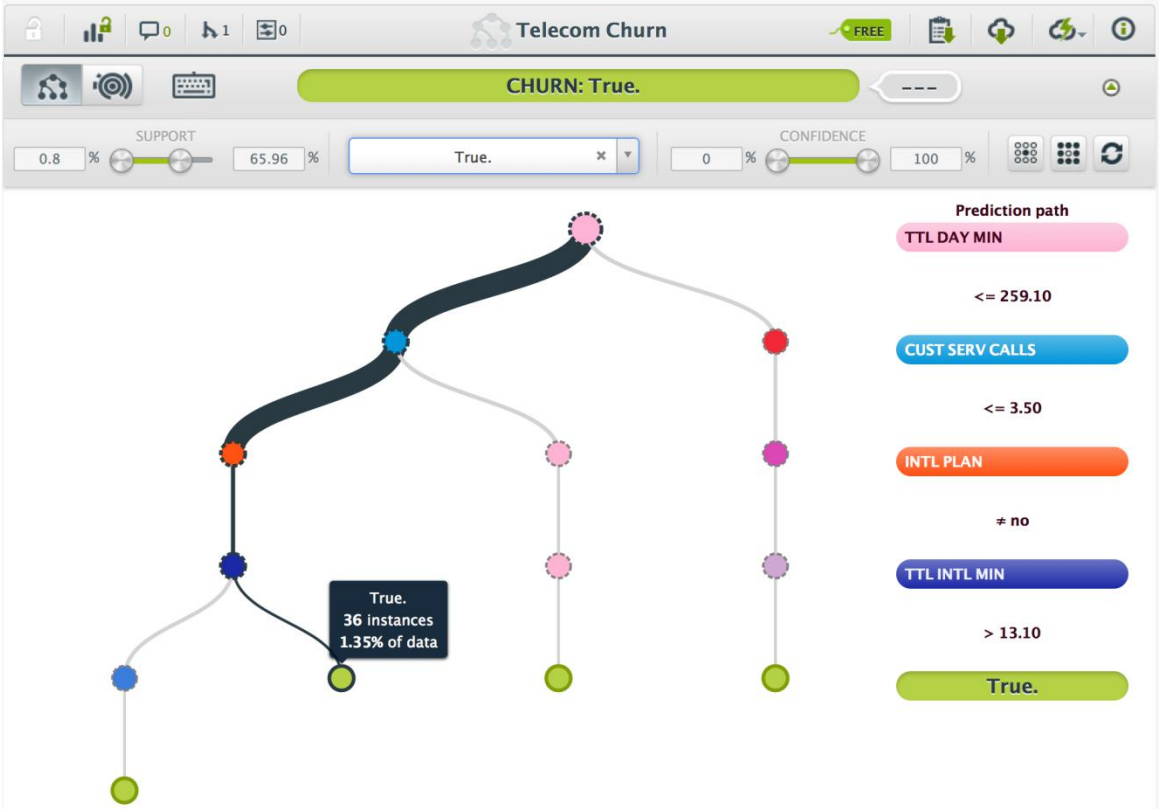
- *¿Comprará el cliente el servicio X si le doy el incentivo I? Este es un problema de **clasificación**, ya que el objetivo es binario (SI/NO).*
- *¿Qué paquete de servicio (X, Y, Z, W) comprará el cliente dado el incentivo I? También es una **clasificación**, pero el objetivo tiene **múltiples valores** o clases, en este caso cuatro.*
- *¿Cuánto usará el cliente el servicio X? En este caso el objetivo no es una etiqueta, sino un número. Esto se ajusta a los modelos de **regresión**.*

Por lo anterior vemos que los problemas difieren en la variable objetivo que guía el aprendizaje, que pueden ser etiquetas (valores nominales, clases) o valores numéricos que representan cantidades.

- ***Un ejemplo de clasificación: árboles de decisión***

Un árbol de decisión clasifica individuos en forma de reglas, que se suelen visualizar como un árbol.

La siguiente Figura muestra un ejemplo de un árbol de decisión sobre un dataset de abandono (churn) de una compañía de telecomunicaciones, obtenida de los modelos públicos de BigML.



En la imagen apreciamos que:

- *el objetivo de la clasificación es el atributo CHURN. En la Figura se muestra el árbol para churn=FALSE, es decir, para los usuarios que no abandonan.*
- *Cada nodo representa una condición sobre una de las variables o atributos.*
- *El grosor de las líneas representa la cantidad de instancias en cada rama.*
- *Cuando nos posicionamos sobre uno de los nodos "hoja", es decir, los de más abajo del árbol, nos indica el número de instancias clasificadas y el porcentaje del total que representan.*
- *También a la derecha nos indica las condiciones sobre los diferentes atributos para ese nodo.*

La representación del clasificador como reglas es la siguiente:

```
True.:
· 24.10%: TTL DAY MIN > 259.104545455 and VMAIL PLAN /= yes and TTL EVE
MIN > 184.9 and TTL NGHT MIN > 167.75
· 13.36%: TTL DAY MIN <= 259.104545455 and CUST SERV CALLS > 3.5 and TTL
DAY MIN <= 163.075 and TTL DAY MIN <= 133.75
· 12.38%: TTL DAY MIN <= 259.104545455 and CUST SERV CALLS <= 3.5 and
INTL PLAN /= no and TTL INTL MIN <= 13.1 and TTL INTL CALLS <= 2.5

...etc.
```

Vemos que hay un conjunto de reglas, que nos muestra cada uno de los caminos en el árbol. BigML nos lo muestra ordenado por el porcentaje de instancias cubiertas para el objetivo dado, de manera que podríamos descartar algunas reglas que tengan un soporte pequeño (no porque podamos afirmar que no sean acertadas, sino porque tenemos "poca evidencia" en nuestro conjunto de datos).

Por eso vemos que el modelo proporciona una manera de estimar si un nuevo cliente abandonará o no, siguiendo las secuencias de reglas. Esa secuencia de reglas es directamente trasladable a cualquier lenguaje de programación sin más esfuerzo.

1.4.2. Un ejemplo de análisis no supervisado

En ocasiones tenemos un conjunto de datos y hacemos preguntas sin un objetivo de predicción o clasificación, por ejemplo: ¿hay grupos diferentes en nuestra base de clientes?. Este es un caso de aprendizaje no supervisado.

Hay técnicas de aprendizaje que son específicas del aprendizaje no supervisado, concretamente:

- *El agrupamiento (**clustering**) intenta agrupar los individuos por su **similaridad**, sin ningún criterio especial a priori. Esta técnica es útil para observar los grupos naturales que existen, porque estos nos pueden sugerir objetivos para otras tareas de minería. También se utiliza para*

buscar estructura, por ejemplo ¿cómo se deberían agrupar nuestros equipos de ventas?

- *El **agrupamiento por co-ocurrencia** (o estudios de **asociación**) responde a preguntas como ¿qué productos se suelen comprar juntos? Mientras que el clustering se basa en los atributos de las instancias, en este caso, se mira a la co-ocurrencia de esas instancias en transacciones. Una **transacción** sería por ejemplo la compra de un cliente en un supermercado. Si encontramos que una cierta salsa se compra con mucha frecuencia junto a un cierto tipo de alimento, nos puede servir para diseñar estrategias de ofertas o ubicación de los productos. Dado que se usa mucho en el terreno de la compra, a estas técnicas se les llama de "market basket analysis".*

- *El **profiling** o descripción de comportamiento intenta **caracterizar el comportamiento típico de un individuo o grupo**. Una pregunta típica es: ¿cuál es el patrón de llamadas típico en este segmento de clientes? Sabiendo ese patrón (por ejemplo, que llama más en fin de semana) tenemos oportunidades para diseñar políticas de precios por ejemplo.*

Estas son tres grandes categorías, para cada una de ellas tenemos muchas posibilidades en cuanto a algoritmos. Veamos un ejemplo.

- ***¿Qué son los estudios de asociación?***

En ocasiones tenemos datos recogidos de manera sistemática pero no tenemos ni tan siquiera idea preliminares sobre qué relaciones puede haber entre ellos. También en ocasiones los datos tienen un número de variables muy alto, y hacer un estudio de relación entre variables considerando todas las posibles relaciones e interacciones puede requerir mucho tiempo y no ser práctico.

*En estos casos, se pueden buscar patrones entre los datos, sin ninguna hipótesis previa. A esto se le denomina **estudios de asociación**.*

*Los **estudios de asociación son por su naturaleza exploratorios**, dado que buscan "ciegamente" relaciones entre variables. Tradicionalmente, se les ha considerado como una tarea de minería de datos, dado que encajan en la definición habitual de búsqueda de conocimiento en grandes bases de datos.*

*Los estudios de asociación más habituales tratan de extraer **reglas de asociación**. Una regla de asociación representa una co-ocurrencia frecuente de valores en ciertos atributos.*
Por ejemplo, si tenemos un conjunto de datos con datos escolares en una ciudad con las siguientes variables:

- ***abandono** (SI/NO), que indica si el estudiante abandonó el programa.*
- ***distrito** (varios valores nominales), que indica el distrito en el que vive el estudiante.*
- ***repeticion** (numérico), que indica las veces que el estudiante ha repetido curso.*
- ***sexo (H/M):** indica el sexo del estudiante.*

Posibles reglas de asociación en este caso serían:

sexo = H -> abandono = SI (soporte 30%, confianza 77%)

distrito = "A" y repeticion="2" -> abandono = SI (soporte 15%, confianza 92%)

El primero de los casos ha descubierto una relación entre el sexo y el abandono. El soporte nos indica el porcentaje del total de los registros en nuestros datos que dan esa combinación de valores, y la confianza el porcentaje en que esos valores concretos se dan para ese par de variables. En este caso, se puede ver que hay una cierta mayor incidencia de varones que abandonan la Escuela.

Ese primer caso no obstante podría detectarse mejor con cualquier técnica de correlación. Por ello es más interesante el segundo. En el segundo, vemos que la combinación de distrito "A" y repetición dos años es un caso poco frecuente en general, pero en la mayor parte de los casos en que se da esa combinación, lleva al abandono. Esta regla sí nos ha dado un ***hallazgo o conocimiento nuevo*** *que es difícil de obtener por técnicas estadísticas convencionales de correlación y que podemos después interpretar.*

1.4.3. Limpieza y transformación de datos

Antes de aplicar cualquier algoritmo de aprendizaje automático, es necesario preparar los datos. Esta preparación en ocasiones es simplemente una labor de limpieza, que puede incluir eliminar instancias con datos incompletos o erróneos, o bien transformar los datos a otras unidades. En otras ocasiones, la tarea de preparación es más compleja, porque necesita analizar relaciones entre los datos, incluyendo la posibilidad de eliminar variables o atributos del conjunto de datos.
A continuación vemos a través de ejemplos prácticos casos de este tipo de tareas.

- ***Limpieza y transformaciones***

La limpieza de datos tiene que ver con los procesos y actividades relacionados con la calidad de los datos. Incluye una serie de actividades para conseguir un dataset o base de datos limpio.

Algunos ejemplos relacionados con criterios de calidad son los siguientes:

- *Exactitud: La exactitud suele concernir a la toma de datos, y es importante tenerla en cuenta.*
- *Integridad: Corregir datos que contienen anomalías, por ejemplo, datos negativos donde no puede haberlos.*

- *Uniformidad: En ocasiones se pueden detectar valores extremos ("outliers") que pueden ser indicativos de valores erróneos.*
- *Unicidad: Relacionado con datos duplicados que pueden aparecer y distorsionan los procesos.*
- *Completitud: Datos faltantes para algunos atributos. Hay que decidir qué hacer con ellos, si se sustituyen por valores por defectos, o se extrapolan de otros datos.*

La limpieza de datos requiere conocer la fuente de datos y cómo se han tomado.

La transformación de datos tiene que ver con cambios en los datos para poder procesarlos (por ejemplo, cambiar los tipos de datos, cambiar las unidades en las que se expresan) o bien para visualizarlos (por ejemplo, aplicar escalas logarítmicas). También es necesario cuando los datos vienen expresados utilizando diferentes categorías. Por ejemplo, datos que utilizan diferentes taxonomías de productos, donde un mismo producto se expresa con diferentes etiquetas. En ese caso, tenemos que "unificar el vocabulario".

- ***Correlaciones simples***

En muchos casos, simplemente dibujar en un nube de puntos la relación entre dos variables, nos puede servir para observar una posible correlación entre las mismas. Podemos utilizar también el coeficiente de correlación para obtener una medida de esa relación.

Si estamos comparando dos variables de entrada y están fuertemente relacionadas, es posible que incluir las dos en el modelo no aporte nada significativo, y solamente añada complejidad al mismo, por lo que deberíamos considerar eliminar una de ellas.

- ***Análisis de componentes principales***

Cuando tenemos un dataset con muchas variables, debemos considerar si queremos retenerlas todas en nuestro modelo, por varios motivos, que incluyen la complejidad (más variables hacen más difícil de comprender y utilizar el modelo) y también en caso de grandes volúmenes de datos, el tiempo que se tarda en el entrenamiento. A este proceso de seleccionar las variables "más importantes" para eliminar las demás pero teniendo aún un modelo "bueno" se le denomina ***reducción de dimensionalidad****, dado que se reduce el número de variables, atributos o dimensiones.*

Cuando tenemos muchas variables, no es práctico hacer este proceso probando modelos con todas las combinaciones posibles de variables. El ***análisis de componentes principales*** *(****ACP****, en inglés, principal component ananlysis,* ***PCA****) es una técnica utilizada para reducir la dimensionalidad de un conjunto de datos. La técnica sirve para hallar las causas de la variabilidad de un conjunto de datos y ordenarlas por importancia, de manera que se pueden descartar las menos importantes.*

El PCA es un paso previo a la aplicación de los algoritmos de aprendizaje automático, que permite simplificar los estudios desde el principio.

2. Caso de aprendizaje supervisado

En el aprendizaje supervisado, nuestros datos están etiquetados, es decir, cada instancia de nuestro conjunto de datos tiene un atributo que es el que clasifica a esa instancia en dos o más clases. Por ejemplo, si tenemos un conjunto de datos de hipotecas, podríamos querer tener un modelo del imapago de las mismas. Entonces tendríamos en nuestro conjunto de datos:

- *Atributos de entrada: podrían ser en el ejemplo el principal de la hipoteca, el tipo de interés, la edad del cliente, la profesión, etc.*
- *Un atributo que es la "salida", etiqueta u objetivo, que en este caso sería el campo que indicase si ese cliente impagó o no la hipoteca.*

Así, los algoritmos "aprenden" tratando de buscar diferencias entre las características de las instancias de clientes que impagaron y los que no.

2.1. Preliminares

En el Notebook 00_Preliminares tenemos una descripción general y recursos de scikit-learn.

Antes de comenzar a ver en detalle cómo se hace aprendizaje automático con scikit-learn, es importante entender qué es y qué dependencias tiene de otros paquetes.

Este primer Notebook es simplemente descriptivo, para conocer bien el contexto.

2.2. Preparando los datos

En el Notebook repasamos el uso de arrays y matrices, dado que los modelos de scikit-learn los utilizan como datos de entrada para el entrenamiento.

Vemos un primer caso de uso de un clasificador y los pasos de preparación, entrenamiento y predicción de manera básica.

También se introduce el módulo scikit.datasets que nos permite generar datasets ficticios para probar algoritmos, y el uso de los DataFrames de pandas para cargar datos de ficheros u otras fuentes externas.

2.3. Clasificación: conceptos básicos

El siguiente paso (Notebook) *es utilizar un clasificador y tratar de entender sus resultados y cómo es su modelo. En este caso vamos a utilizar un clasificador basado en vectores de soporte. Es una técnica de aprendizaje automático robusta y muy utilizada.*
Al final del Notebook se detallan los conceptos importantes a practicar. Es muy importante probar a cambiar los ejemplos del Notebook, y observar cómo las SVC se ajustan a diferentes situaciones.

2.4. Conceptos de evaluación de modelos

Una vez que sabemos crear y entrenar modelos, es importante entender cómo evaluarlos. Para eso utilizaremos .

*Evaluar un modelo con los mismos datos con los que se ha entrenado nos permite saber cuán bien se ha ajustado a esos datos. No obstante, eso no tiene necesariamente que determinar cómo se comportará el modelo cuando lo usemos para clasificar datos diferentes de los que se usaron para entrenar. De hecho, este es un problema que se da en muchos algoritmos y se denomina sobreajuste (**overfitting**), y es importante entenderlo y saber cómo identificarlo y en su caso atenuarlo.*

La técnica habitual para no acabar con modelos sobre-ajustados es la validación cruzada (cross-validation) que divide los datos disponibles en un conjunto de entrenamiento y un conjunto de prueba. Este último se utiliza para poner a prueba al modelo entrenado con el primero.

2.5. Conceptos de ingeniería de características

Una vez conocemos lo esencial de la evaluación de modelos, es importante profundizar en la selección de las variables o características de entrada, las que usamos para el entrenamiento. Lo haremos con el Notebook

.

Hay diferentes técnicas para seleccionar las más importantes y reducir la dimensionalidad del conjunto de datos, y también para transformar los datos. Aquí vemos algunas de ellas como ejemplo de este tipo de procesos.

3. Caso de aprendizaje no supervisado

En el aprendizaje no supervisado, nuestros datos no están etiquetados, y de lo que se trata es de buscar patrones, grupos o regularidades en los datos.

En el caso de los algoritmos de agrupamiento ("clustering") el resultado son grupos de individuos. Si esos grupos son considerandos como relevantes por el data scientist, pueden después utilizarse como clases para el aprendizaje supervisado. Por eso en muchas ocasiones el aprendizaje no supervisado se considera un paso previo al aprendizaje supervisado.

En este punto trabajaremos a través de Notebooks preparados paso a paso, que comentamos a continuación.

3.1. Clustering: conceptos básicos

En este primer Notebook) *se introducen los conceptos básicos del clustering, y se contrasta con la forma de entrenar en el aprendizaje supervisado.*

3.2. Clustering – conceptos básicos de evaluación

Cuando evaluamos el resultado de un algoritmo de clustering no siempre tenemos las etiquetas de las instancias que utilizamos para ajustar el modelo. En ese caso, tenemos que utilizar algún tipo de medida de la "homogeneidad" interna de los clusters. Estas medidas pueden variar con el número de clusters, por lo que es importante probar con diferentes configuraciones de los algoritmos. Lo tratamos en el Notebook .

3.3. Probando reglas de asociación

Finalmente utilizamos una implementación simple del algoritmo APriori para obtener reglas de asociación (Notebook). *Ejemplo para probar con casos sencillos de reglas de asociación, para entender los conceptos de k-itemset frecuentes, reglas, soporte y confianza.*