De la mano de uno de los mitos centrales de la imaginería vikinga, el del destino tejido, realizaremos un recorrido por un algoritmo de aprovechamiento de datos genérico. No entraremos en detalle, pero nos servirá para comprender algo mejor cómo funcionan y qué esperar de estos algoritmos.

[ILUSTRACIÓN: *THE THREE NORNS*, POR J.J. LUND]

La niebla gris cubre el sonido del mar, roto por el graznido de cuervos, espías del Padre de Todos. Entre las ruinas y cenizas del incendio quedan los cuerpos de los caídos en la incursión: la mayoría, caídos dentro de su armadura, cristianos, pero también unos pocos paganos sin ella.

Los siglos del VIII al XI vieron cómo los pueblos nórdicos asolaban con sus incursiones todo el continente europeo, gracias a una fiereza incomparable en combate, y a su desprecio por la muerte.

Las creencias de los daneses se centraban en una poderosísima noción de destino, de que lo que iba a pasar ya estaba decidido, con lo que solamente caían en batalla quienes iban a morir, lo que les daba valor para hacerlo, si fuera el caso, de forma gloriosa.

Según la mitología nórdica, todo lo que existe se encuentra en un gigantesco fresno —Yggdrasil— en el centro del cual está nuestro mundo, entre otros ocho. El árbol tiene también diversos habitantes, como una ardilla, llamada Ratatoskr, que lo recorre de arriba abajo constantemente, o, bajo sus raíces, tres mujeres. Mujeres más poderosas que ningún otro ser vivo, ante cuyos caprichos ni los dioses pueden nada.

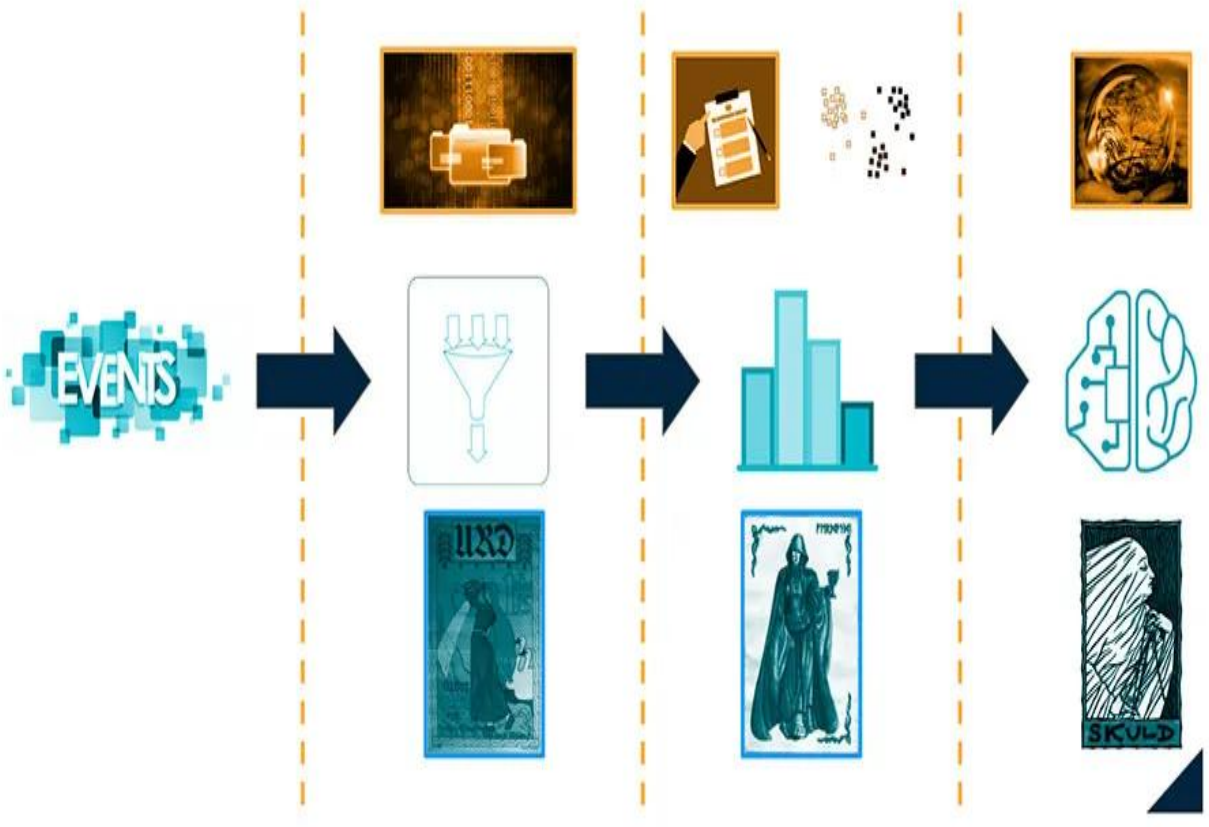
Cuando hablamos de machine learning y de big data, el sueño es ser capaces de comprender qué es lo que va a ocurrir, gracias a saber lo que ha ocurrido... No se aleja mucho del “destino” vikingo

Esas tres mujeres son las nornas, tres hermanas hilanderas que tejen en un telar todo lo que ha ocurrido, ocurre y ocurrirá en Yggdrasil. El destino de todo. La primera de las tres, Urd (lo que fue), vieja y fea, teje lo que ya ocurrió. En las imágenes que hay de ella, suele mirar hacia la izquierda, al pasado. La segunda, Verdandi (lo que es), joven, teje aquello que está ocurriendo. Representada normalmente mirando al frente, tiene cierta relación con la primavera según algunas lecturas de la mitología. Por último, Skuld (lo que será), la norna encargada de tejer lo que aún no ha ocurrido. Con la cara cubierta con un velo, imagen del misterio que encarna, es, por encargarse de lo que está por venir, la más mística de las hermanas.

En el mundo informático, cuando hablamos de *machine learning* y de *big data*, el sueño es ser capaces de comprender qué es lo que va a ocurrir, gracias a saber lo que ha ocurrido... No se aleja mucho del "destino" vikingo. De hecho, podríamos decir que lo que buscamos es adivinar el telar de las nornas gracias a los hilos de Urd.

Este paralelismo va más allá de lo anecdótico, y, de hecho, nos puede servir para comprender un poco mejor en qué consiste y cómo se realiza el aprovechamiento de los datos.

Vamos a intentar repasar un algoritmo *big data*, acompañados de esta metáfora, desde que nacen los datos hasta el final de su explotación.



En primer lugar, ocurren cosas en el mundo, en Yggdrasil, a las que llamamos eventos. Cualquier cosa que podamos registrar, es decir, que de lugar a datos, será un evento. Pero los eventos ocurren en Yggdrasil, no en el telar de las nornas, así que habrá que tejerlos, habrá que recolectarlos y, de eso, de esos acontecimientos pasados, se encargará la primera pieza de nuestro algoritmo: Urd. Al final de este proceso, lograremos tener una gran colección de datos crudos.

Tras esa recolección, tenemos que sacar utilidad a los datos que hemos recogido, hacerlos florecer, aprender de ellos y entender lo que está ocurriendo. Un trabajo para la segunda de nuestras piezas y de las hermanas, Verdandi, que nos dejará con un importante conocimiento humano de la información, en forma de indicadores, audiencias…

Por último, es la propia máquina la que debe aprender de los datos, ser capaz de predecir eventos futuros gracias a aquello que registró urd, lo que corresponde a la última de las tres piezas y de las tres nornas: Skuld. Gracias a este aprendizaje, conseguiremos obtener predicciones, esperamos, precisas.

Aproximémonos una a una a cada una de las piezas mencionadas.

Urd

La recolección de datos consiste en lo que en lenguaje informático llamamos ETLs —por *Extract, Transform & Load*—, secuencias en las cuales, cuando ocurre algo, recogemos lo que ha ocurrido, le damos un formato coherente con nuestra estrategia de guardado, y lo guardamos en nuestras bases de datos. Los eventos guardados los llamamos registros.

Como la norna que nombra a esta pieza, los resultados son feos, incomprensibles —un amalgama de datos—, pero necesarios para las fases posteriores.

Los retos asociados a la primera parte del algoritmo son sobre todo de índole técnica, asociados sobre todo a la grandísima cantidad de datos que se producen, y que deben ser procesados; así como la complejidad de hacerlo en tiempo real.

Verdandi

Los datos crudos, como los deja Urd, son poco más que inútiles por sí solos, han de florecer con la energía primaveral de Verdandi, para que podamos ver el presente en el telar.

En esta etapa, el propósito es utilizar los datos para obtener conocimiento, para lo que podemos utilizar indicadores, agregados —como totales de registros— y gráficas o visualizaciones.

Además, y quizá de forma más interesante, podemos buscar agrupaciones de registros parecidos. Pongamos por caso que nuestro modelo de negocio es de ventas o publicidad; entonces, esas agrupaciones serían lo que llamamos audiencias.

Si somos capaces de agrupar registros, podemos ver dónde y cómo se agrupan, y aprender de las características de los grupos que nos resulten más interesantes.

Pongamos por caso que buscamos maximizar el impacto de una campaña web, maximizando el número de personas que hacen click en nuestros anuncios. Podríamos generar audiencias según localización. Así, veríamos dónde hay más gente que reaccione como buscamos.

Igual que ese caso, cualquier otro parámetro por el cual generemos audiencias —el tipo de conexión que tenía el usuario de Internet, el anuncio que se ha presentado…— nos llevará a aprender de los datos que hemos recogido, a entender mejor cómo optimizar nuestra campaña publicitaria.

Como añadido, hay una forma más de agrupar, sin tener que decidir en función de qué parámetro hacerlo, sino dejando que el ordenador agrupe en función de varios parámetros al mismo tiempo. Esto es muy útil para poder descubrir nuevas relaciones entre registros y distintas audiencias, que nos permitan generar más conocimiento aún.

Las técnicas que utilizamos para esto último se llaman de clusterización, estrategias de machine *learning* que nos permiten generar esas agrupaciones. Para ello, necesitamos dos cosas: datos limpios y una función de parecido.

Datos limpios quiere decir datos coherentes, para que no engañen al ordenador. Por ejemplo, si estamos llevando a cabo una campaña publicitaria en España, siempre puede ocurrir que, por error, se haya mostrado algún anuncio fuera; se deben retirar estos registros, ya que, aunque sean pocos,pueden distorsionar mucho el conjunto.

En cuanto a la función de parecido, se refiere a una función que sea capaz de representar numéricamente cuánto se parecen dos registros concretos. Volviendo a nuestro ejemplo de la campaña publicitaria, digamos que tenemos un registro A correspondiente a que alguien ha visto en Barcelona, utilizando el móvil y conexión de datos uno de nuestros anuncios, un registro B que refleja que alguien lo ha visto en Madrid, utilizando también el móvil y conexión de datos, y un registro C asociado a que una tercera persona lo ha visto en Barcelona, con el ordenador y con conexión Wi-Fi. Una posible función de parecido diría que A y B se parecen

2, mientras que A y C se parecen 1, y B y C 0.

No existe una función de parecido “correcta”, y normalmente será necesario ir mejorando la que utilicemos.

Para realizar correctamente todo este proceso de aprendizaje continuado usamos un método una y otra vez, como la ardilla que comentamos al principio del artículo recorre una y otra vez Yggdrasil, y, como el sonido que haría esa ardilla, nuestro método es el que denominamos *EEK*.

La primera E corresponde a *Exploration*, porque lo primero que debemos hacer es precisamente explorar nuestros datos, para ser capaces de limpiarlos y de elegir una primera función de parecido.

La segunda E es por *Exploitation*, dado que explotaremos los datos aplicándoles nuestro algoritmo de *clustering*, para generar las agrupaciones correspondientes.

Por último, la K de *Knowledge*, puesto que llega el momento de estudiar las agrupaciones y aprender de ellas. Y, con el conocimiento que obtenemos, podemos volver a la fase de *Exploration*, consiguiendo una mejor limpieza de los datos, así como una mejor función de parecido, repitiendo el proceso y pudiendo conseguir cada vez un mayor y mejor conocimiento.

Tras todo esto, Verdandi habrá cumplido su objetivo, los datos recogidos por Urd habrán florecido en conocimiento humano de los mismos, podremos ver el tejido del presente en el telar, y podremos dar paso a la tercera fase, la tercera pieza de nuestro algoritmo.

Skuld

De la misma forma que no hay Verdandi sin Urd, no hay Skuld si falta alguna de sus dos hermanas. Para poder plantearnos el reto de levantar el velo de la más mística de las hermanas, es imprescindible haber aprendido todo lo posible de los datos antes pues, aunque lo que sigue siendo necesario es la limpieza de los mismos y una función (en este caso de adecuación), en esta etapa es aún más importante que la limpieza esté bien realizada, y la función bien escogida.

Cuando hablamos de limpieza en la etapa correspondiente a Skuld, nos referimos a lo mismo que ya hemos comentado en la etapa correspondiente a Verdandi.

En cuanto a la función de adecuación, nos sirve para saber hasta qué punto estamos siendo capaces de predecir correctamente, hasta qué punto estamos viendo como es la parte del telar que nos resta. Como la función de parecido expresa numéricamente lo similares que son dos registros entre sí, la de adecuación será capaz de expresar lo similar que es nuestra predicción y la realidad, utilizando parte de los datos que tenemos como conjunto de prueba – intentamos predecirlos y utilizamos nuestra función de adecuación entre nuestra predicción y los propios datos.

Con estos conceptos de limpieza y de adecuación, lo que hacemos en esta etapa es aplicar algoritmos de predicción, de los que obtenemos modelos de futuro.

Vamos a echar un pequeño vistazo a tres tipos de algoritmos de predicción, que podríamos decir que engloban la amplia mayoría de los mismos: geométricos, probabilísticos y de red neuronal. Los algoritmos geométricos agrupan los registros por parecido, aunque adaptan la función de parecido automáticamente utilizando la de adecuación, y en los modelos resultantes estiman que los eventos que generen registros dentro de una determinada agrupación, se comportarán como el resto de eventos que hayan generado registros en la misma agrupación.

Los probabilísticos, por su parte, intentan predecir los comportamientos más probables corrigiendo modelos probabilísticos gracias a la función de adecuación.

Por último, los algoritmos de red neuronal se comportan emulando una versión muy simplificada de las interconexiones neuronales en el cerebro humano, y utilizan la función de adecuación para reforzar determinadas “sinapsis”.

Con esto terminaría nuestro trayecto, desde que ocurre un evento, pasando por cómo lo guardamos como registro, cómo aprendemos de él y cómo, por último, la propia máquina aprende de él y lo utiliza para ver el futuro; o, dicho de otra manera, cómo Urd trenza lo que ya ha ocurrido, cómo Verdandi toma esos hilos y teje una imagen comprensible de lo que ocurre y cómo Skuld, por último, continúa esa imagen con una de lo que está por venir. Tres hermanas, cada una vital en el proceso. Tres piezas, cada una imprescindible en nuestro algoritmo.