

Key performance indicators
Abraham Zamudio

Una introducción a las Ideas principales

Que es o que significa predecir

Ejemplo : Es imprescindible que los laboratorios farmacéuticos cuenten con una gestión adecuada de la predicción de la demanda de sus productos, ya que así pueden asegurar el abastecimiento para cubrir las necesidades del mercado.

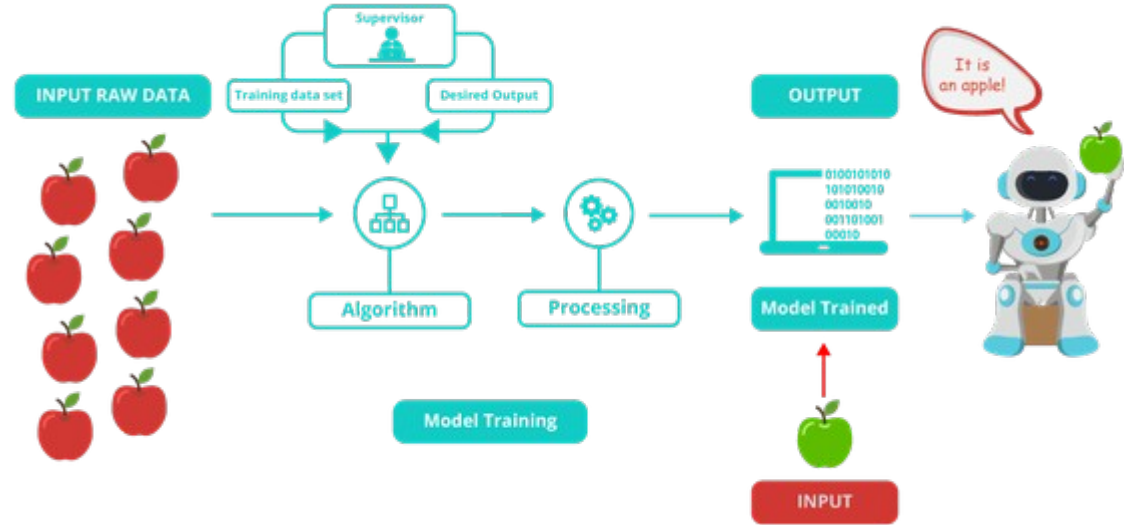
El concepto de previsión de la demanda se refiere a las actividades, estrategias y herramientas que se utilizan en la empresa para hacer estimaciones sobre la cantidad de ventas que puede tener la empresa en un futuro, y que se utilizan para poder ajustar lo máximo posible los procesos de decisión y planificación estratégica con el objetivo de ahorrar costes. De este modo se reduce la incertidumbre en los procesos de gestión corporativa de toma de decisiones basándolos en datos reales, aunque de carácter estimativo. Esto hace que ningún método de los que se pueda emplear sea perfecto ni seguro al 100%, sin embargo, ayudan a que los cálculos sean más fiables, y a medida que vaya disponiendo de más datos, estos se pueden ir incorporando al proceso decisorio con el fin de prever o ajustar posibles desviaciones presentes o futuras.



Técnicas de Aprendizaje Supervisado

El aprendizaje supervisado utiliza técnicas que permiten predecir los valores futuros, basados en datos históricos. Emplea problemas de clasificación y de regresión.

Los algoritmos supervisados o predictivos predicen el valor de un atributo (etiqueta) de un conjunto de datos, conocidos como otros atributos (atributos descriptivos). A partir de datos cuya etiqueta se conoce, se induce un modelo que relaciona la predicción en datos cuya etiqueta es desconocida. Esta forma de trabajar se conoce como aprendizaje supervisado.

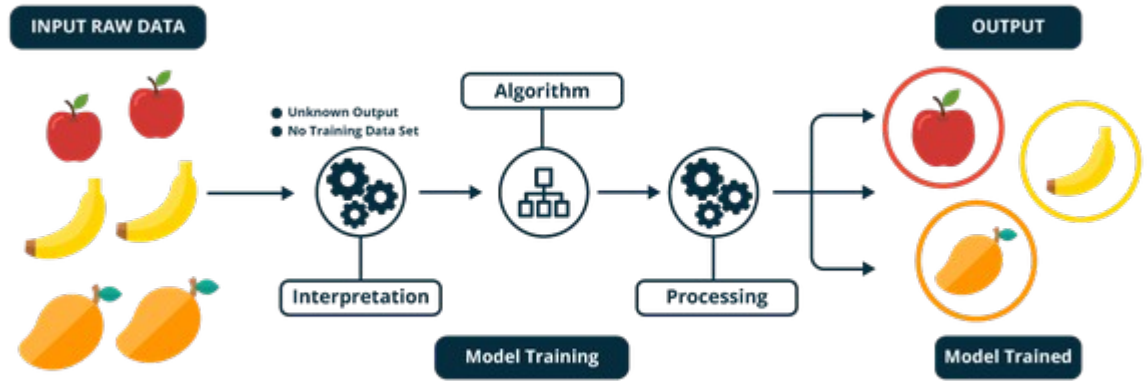


En este grupo se encuentran, por una parte, algoritmos que resuelven problemas de clasificación debido a que trabajan con etiquetas discretas (árboles de decisión, tablas de decisión, inducción neuronal, etc.) y, por otra, algoritmos que se utilizan en la predicción de valores continuos como son la regresión o las series temporales.

Técnicas de Aprendizaje No Supervisado

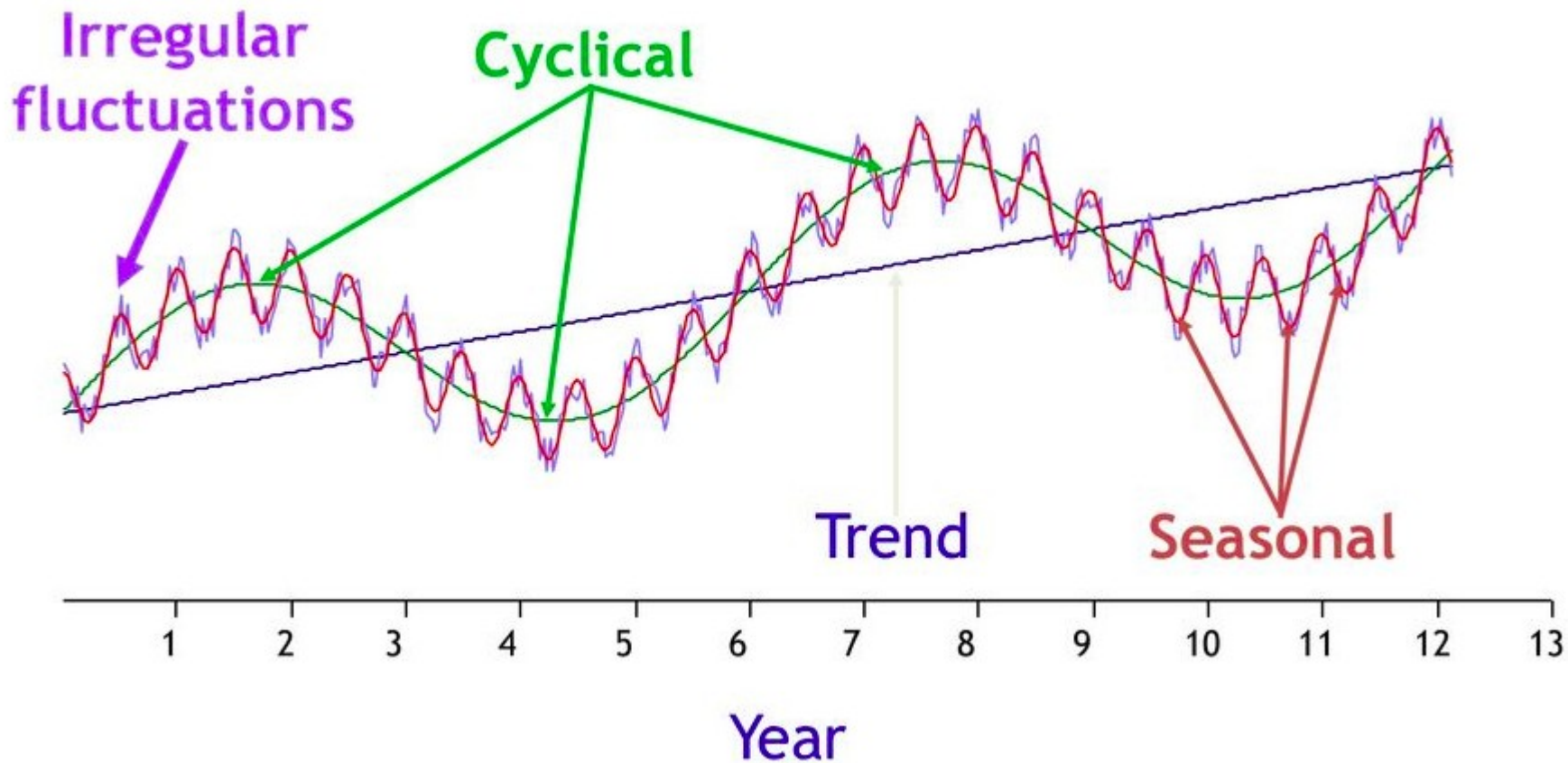
El aprendizaje no supervisado no emplea datos etiquetados, es decir, que no emplean datos históricos para su entrenamiento, pero sí utiliza modelos donde emplea problemas de clustering agrupando los datos para su análisis.

Los algoritmos no supervisados o de descubrimiento del conocimiento realizan tareas descriptivas como el descubrimiento de patrones y tendencias en los datos actuales (no utilizan datos históricos). El descubrimiento de esa información sirve para llevar a cabo acciones y obtener un beneficio científico o de negocio de ellas.

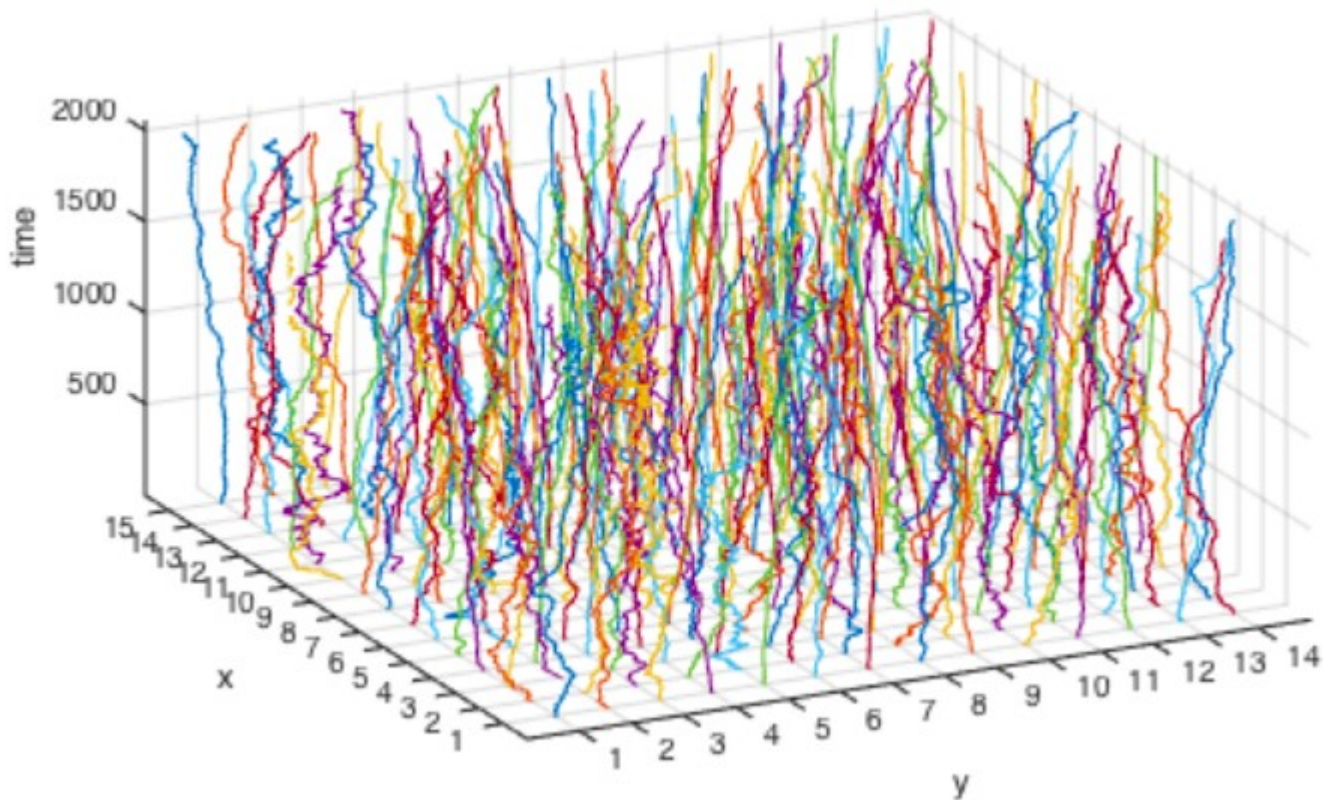


En el aprendizaje no supervisado no tenemos output predeterminado. Esto genera a su vez un gran reto porque es muy difícil saber si ya culminamos con el trabajo o podemos aun generar otro modelo con el que nos sintamos más satisfechos.

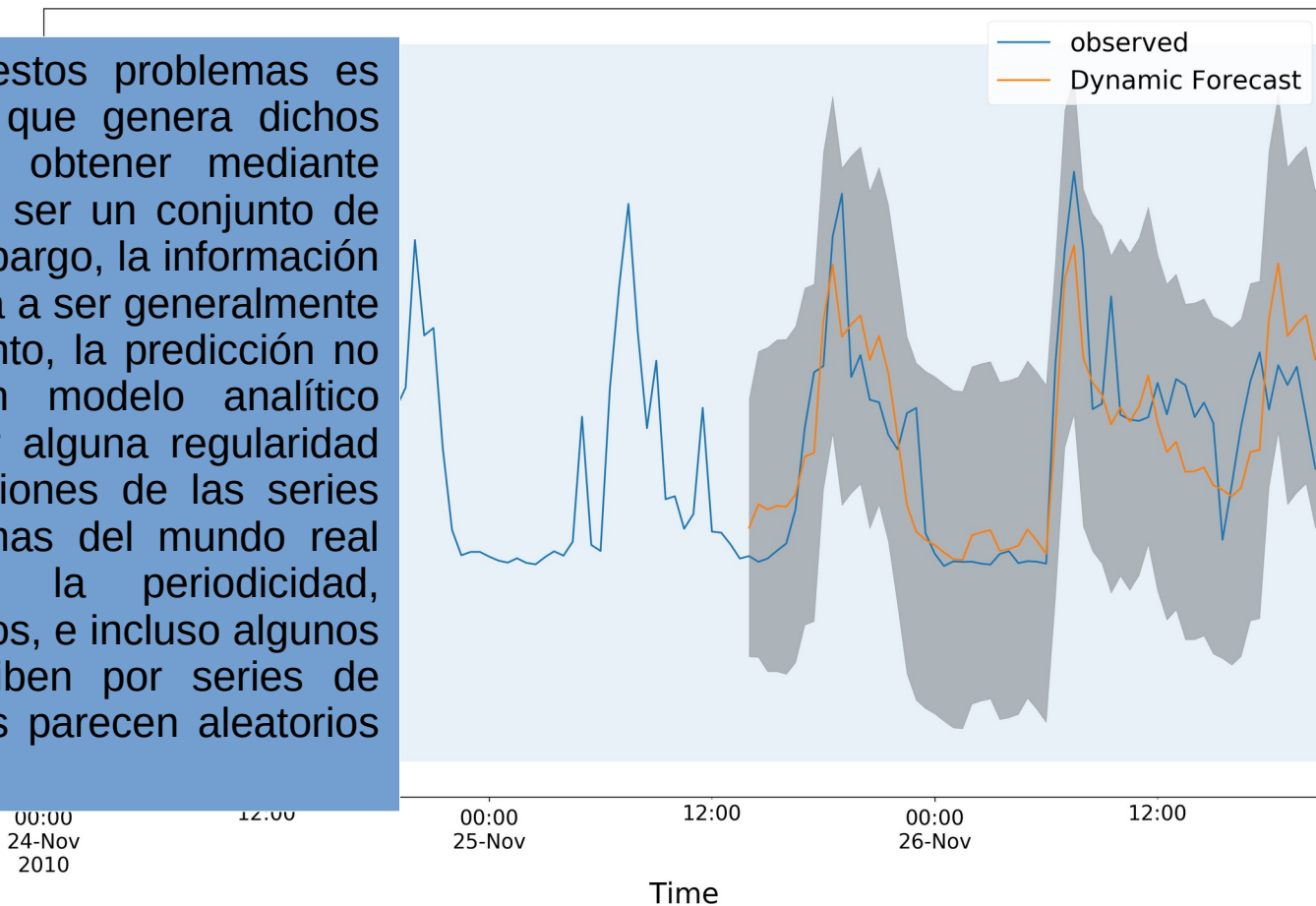
Para el pronóstico de ventas son importantes los intervalos de tiempo, ya que al utilizar datos históricos es necesario emplear la estacionalidad, los ciclos y las tendencias dentro de períodos determinados.



Una serie temporal consiste en una secuencia de valores de varias variables que evolucionan (van cambiando) en el tiempo. Se trata de predecir el comportamiento futuro del fenómeno o sistema dinámico que genera esos valores basándose en una colección de datos históricos. Por ejemplo, la predicción del consumo de energía eléctrica o la predicción del número de vacunas contra la gripe que se van a demandar en una región determinada.



La mejor manera de resolver estos problemas es encontrando la ley subyacente que genera dichos procesos. Esta ley se puede obtener mediante métodos analíticos, como puede ser un conjunto de ecuaciones diferenciales. Sin embargo, la información que vamos a tener del proceso va a ser generalmente parcial o incompleta y; por lo tanto, la predicción no se puede hacer mediante un modelo analítico conocido. Se intentará descubrir alguna regularidad empírica fuerte en las observaciones de las series temporales. En muchos problemas del mundo real algunas regularidades, como la periodicidad, aparecen enmascaradas por ruidos, e incluso algunos procesos dinámicos se describen por series de tiempo caóticas, donde los datos parecen aleatorios sin periodicidades aparente.

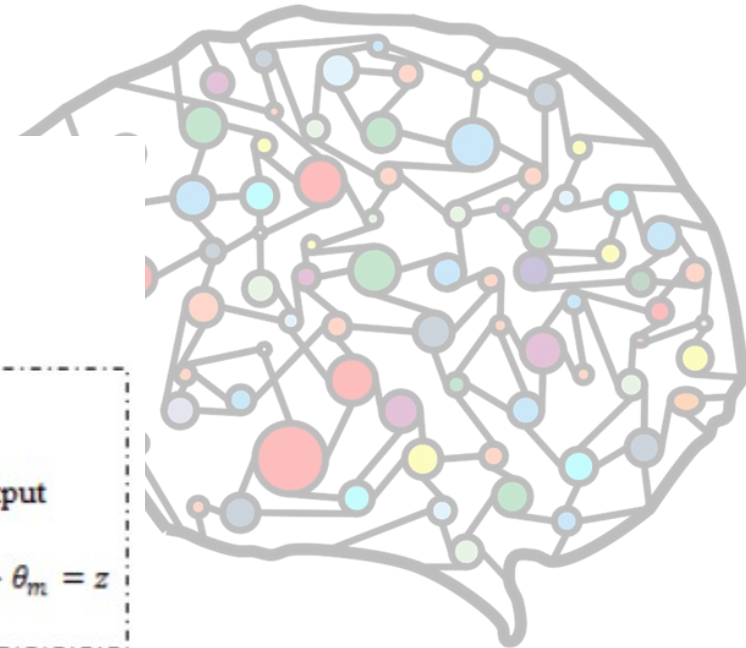
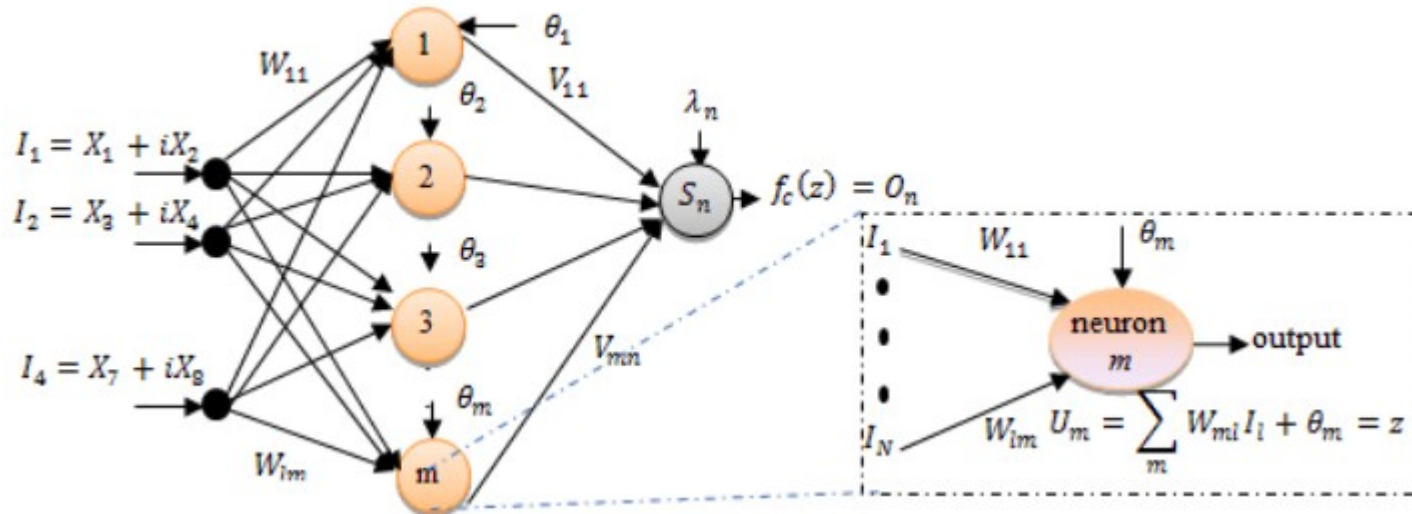


Redes Neuronales

Las ANN al margen de "parecerse" al cerebro presentan una serie de características propias del cerebro. Por ejemplo, las ANN aprenden de la experiencia, generalizan de ejemplos previos a ejemplos nuevos y abstraen las características principales de una serie de datos.

Tres idea fundamentales

- 1) Aprender
- 2) Generalizar
- 3) Abstraer



Aprender: adquirir el conocimiento de una cosa por medio del estudio, ejercicio o experiencia. Las ANN pueden cambiar su comportamiento en función del entorno. Se les muestra un conjunto de entradas y ellas mismas se ajustan para producir unas salidas Consistentes.

Generalizar: extender o ampliar una cosa. Las ANN generalizan automáticamente debido a su propia estructura y naturaleza. Estas redes pueden ofrecer, dentro de un margen, respuestas correctas a entradas que presentan pequeñas variaciones debido a los efectos de ruido o distorsión.

Abstraer: aislar mentalmente o considerar por separado las cualidades de un objeto. Algunas ANN son capaces de abstraer la esencia de un conjunto de entradas que aparentemente no presentan aspectos comunes o relativos.”



El conocimiento en los datos : una metodología

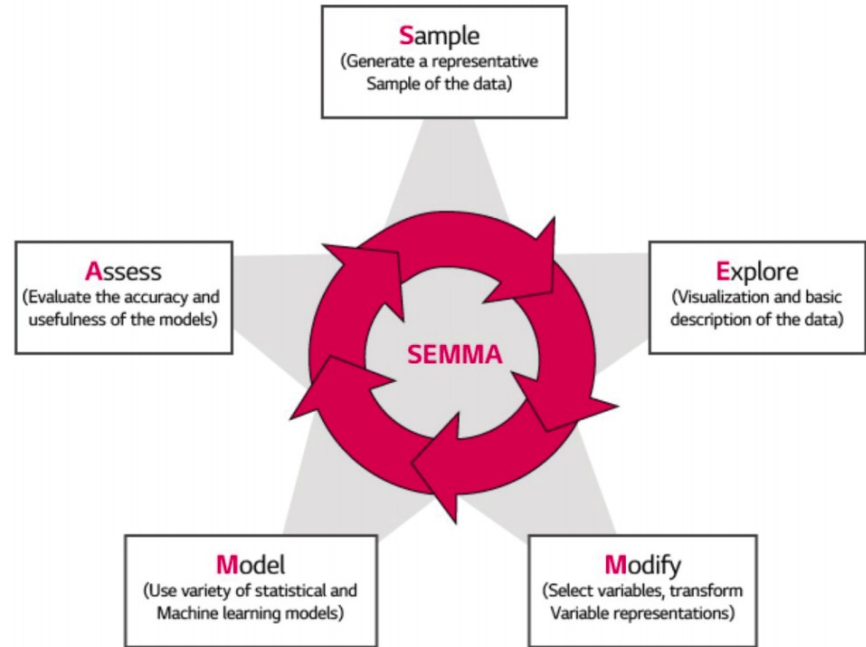
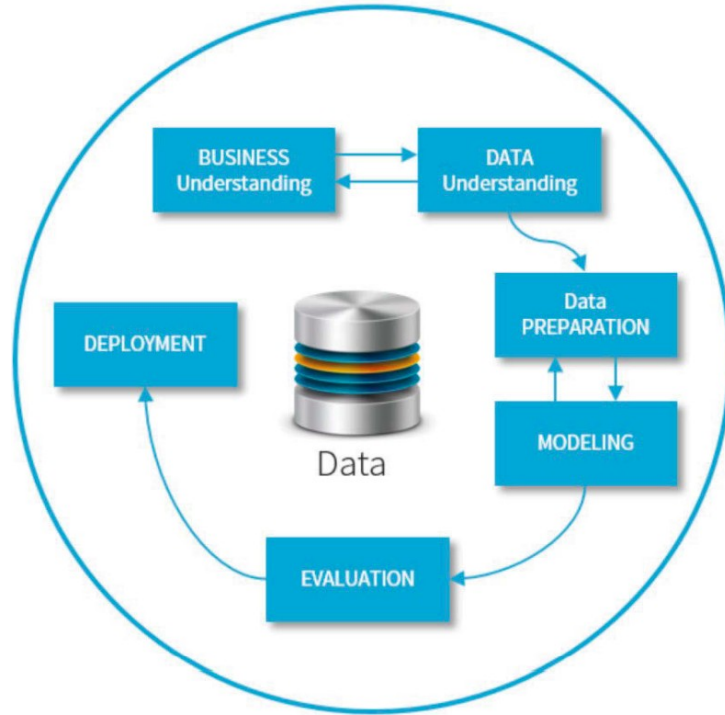
Conceptos básicos

CRISP-DM, que son las siglas de Cross-Industry Standard Process for Data Mining, es un método probado para orientar sus trabajos de minería de datos.

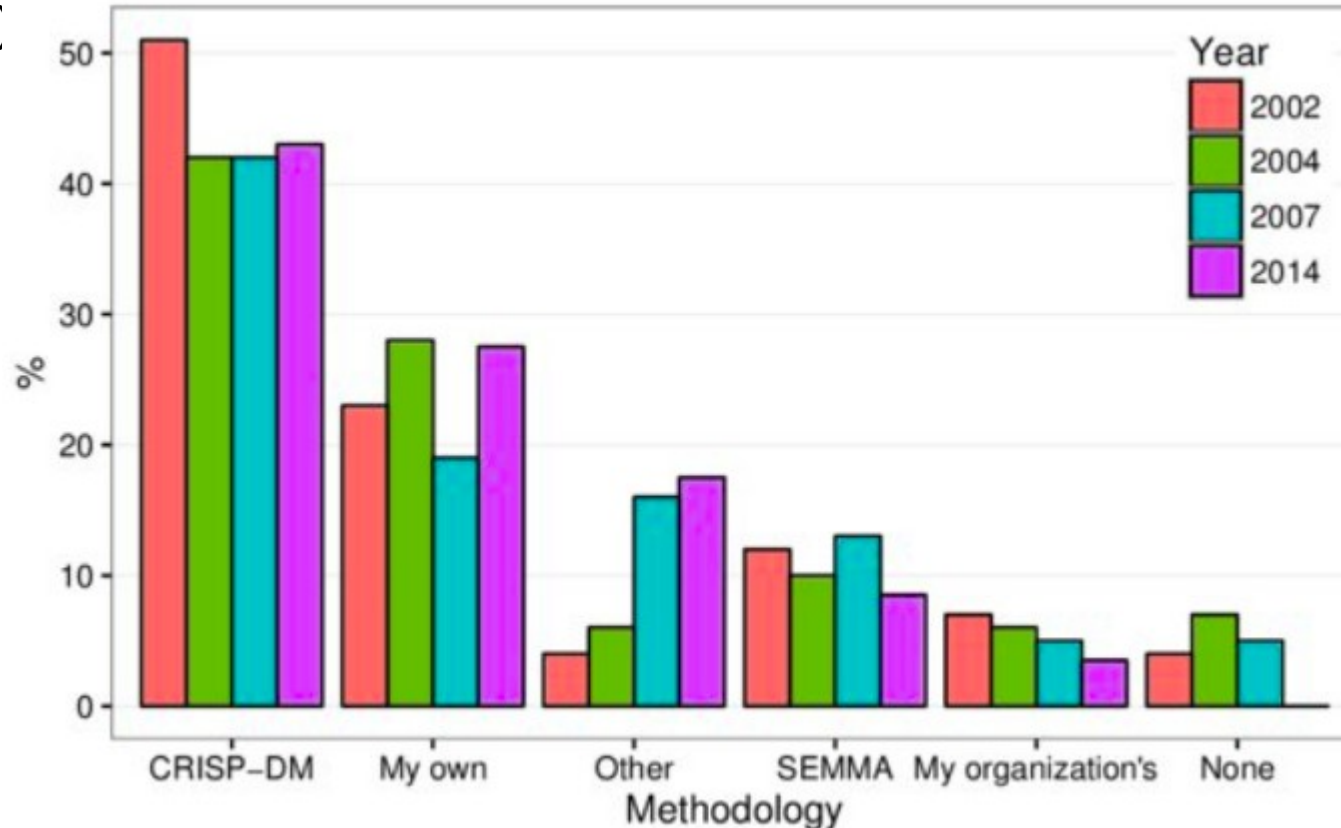
- Como metodología, incluye descripciones de las fases normales de un proyecto, las tareas necesarias en cada fase y una explicación de las relaciones entre las tareas.
- Como modelo de proceso, CRISP-DM ofrece un resumen del ciclo vital de minería de datos.

Las técnicas de Data Science o Data Analytics, que tanto interés despiertan hoy en día, en realidad surgieron en la década de los 90, cuando se usaba el término KDD (Knowledge Discovery in Databases) para referirse al (amplio) concepto de hallar conocimiento en los datos.

En un intento de normalización de este proceso de descubrimiento de conocimiento, de forma similar a como se hace en ingeniería software para normalizar el proceso de desarrollo software, surgieron a finales de los 90 dos metodologías principales: CRISP-DM (Cross Industry Standard Process for Data Mining) y SEMMA (Sample, Explore, Modify, Model, and Assess). Ambas especifican las tareas a realizar en cada fase descrita por el proceso, asignando tareas concretas y definiendo lo que es deseable obtener tras cada fase.



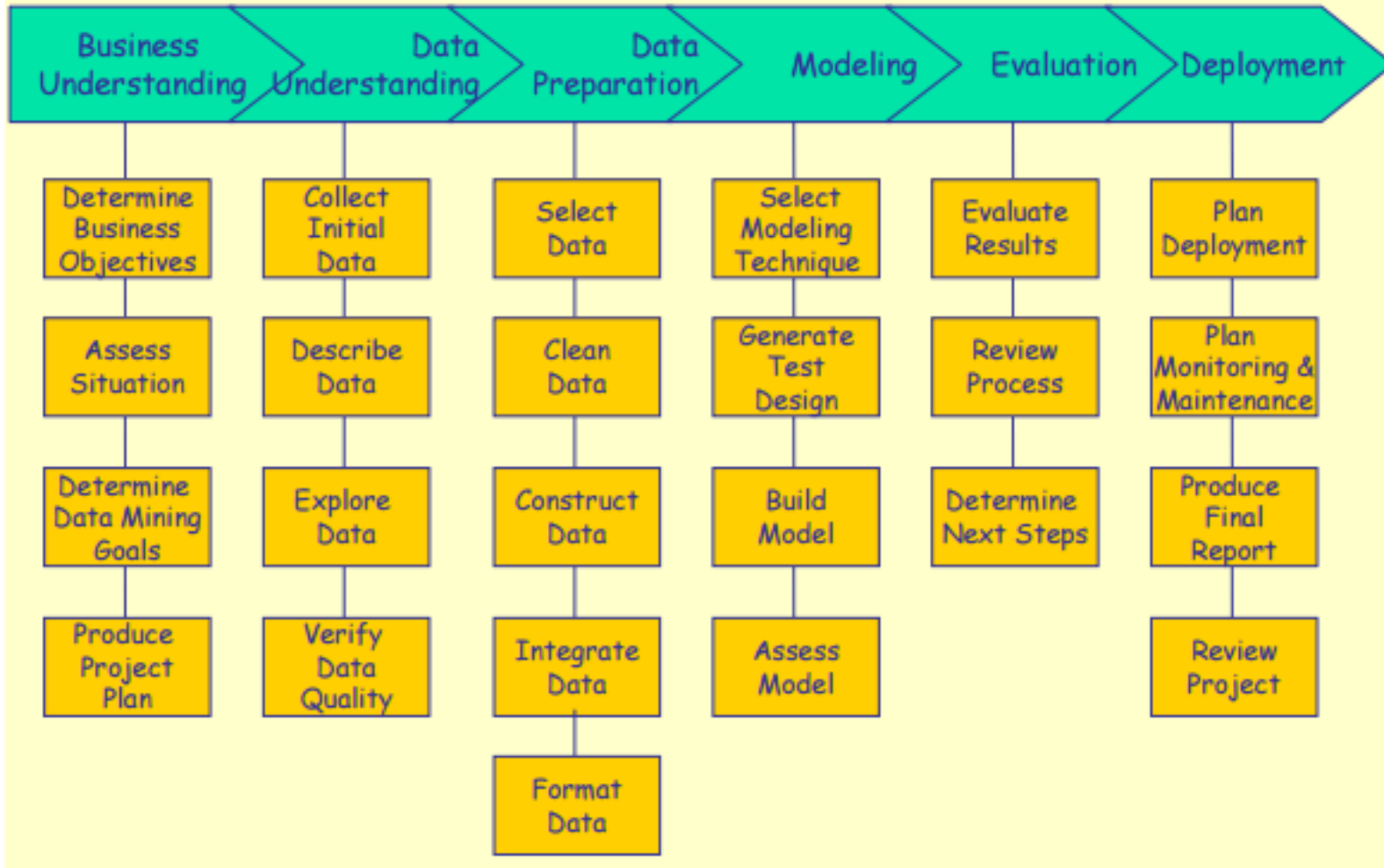
Al comparar las dos metodologías se llega a la conclusión de que, aunque se puede establecer un paralelismo claro entre ellas, CRISP-DM es más completo porque tiene en cuenta la aplicación al entorno de negocio de los resultados, y por ello es la que se adoptó popularmente (en encuestas realizadas en KDNuggets en 2002, 2004, 2007 y 2014 se comprobó que CRISP-DM era la principal metodología utilizada, 4 veces más que SEMMA).



La metodología CRISP-DM contempla el proceso de análisis de datos como un proyecto profesional, estableciendo así un contexto mucho más rico que influye en la elaboración de los modelos. Este contexto tiene en cuenta la existencia de un cliente que no es parte del equipo de desarrollo, así como el hecho de que el proyecto no sólo no acaba una vez se halla el modelo idóneo (ya que después se requiere un despliegue y un mantenimiento), sino que está relacionado con otros proyectos, y es preciso documentarlo de forma exhaustiva para que otros equipos de desarrollo utilicen el conocimiento adquirido y trabajen a partir de él.

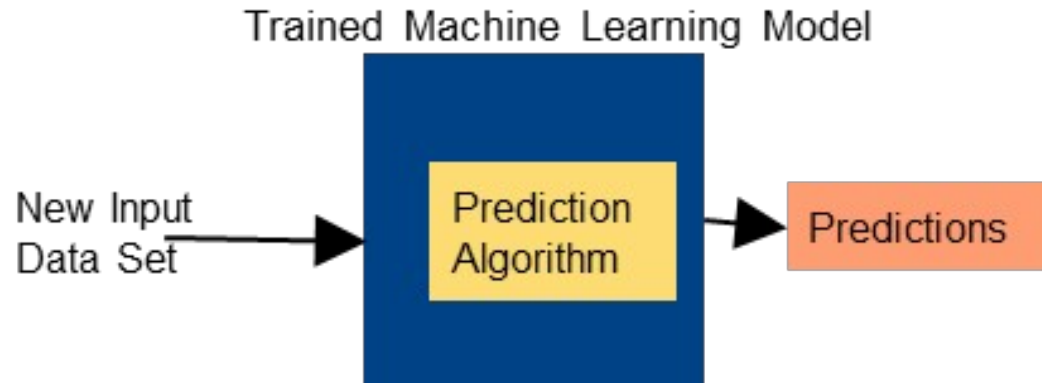
El ciclo de vida del proyecto de minería de datos consiste en seis fases mostradas en la figura siguiente.

Phases and Tasks



Rendimiento y métricas de un modelo de machine learning

Siempre debe evaluar un modelo para determinar si realizará un buen trabajo de predicción para nuevos y futuros datos de destino. Dado que las futuras instancias tienen valores de destino desconocidos, debe comprobar la métrica de precisión del modelo de ML en relación con los datos de los que ya sabe la respuesta de destino y utilizar esta comprobación como proxy de precisión predictiva para futuros datos.



Evaluar la precisión predictiva de un modelo de ML con los mismos datos que se han utilizado para el entrenamiento no es útil, ya que compensa a los modelos que pueden "recordar" los datos de entrenamiento en lugar de generalizar.

Una vez que haya entrenado el modelo de ML, envíe al modelo las observaciones separadas para las que conoce los valores de destino. A continuación, compare las predicciones devueltas por el modelo de ML con el valor de destino conocido. Por último, genere una métrica de resumen que indique la efectividad de la predicción y la coincidencia de valores reales. Una vez que haya entrenado el modelo de ML, envíe al modelo las observaciones separadas para las que conoce los valores de destino. A continuación, compare las predicciones devueltas por el modelo de ML con el valor de destino conocido. Por último, genere una métrica de resumen que indique la efectividad de la predicción y la coincidencia de valores reales.

Métricas de Rendimiento para problemas de clasificación

MATRIZ DE CONFUSION : Esta matriz se utiliza para evaluar la precisión de un clasificador y se presenta en la tabla a continuación.

		Resultado de la predicción		
		Positivo	Negativo	
Valor actual	Positivo	TP	FN	TP + FN
	Negativo	FP	TN	FP + TN

Métricas de Rendimiento para problemas de clasificación

Unos ejemplos

El **falso positivo (FP)** añade un correo electrónico de confianza a correos no deseados en un motor antispam.

El **falso negativo (FN)** en una evaluación médica puede mostrar incorrectamente la ausencia de enfermedad, cuando en realidad existe.

Error tipo I
(falso positivo)



Error tipo II
(falso negativo)



Métricas de Rendimiento para problemas de clasificación

MÉTRICA DE EXACTITUD

Esta métrica es la base uno. Indica el número de elementos clasificados correctamente en comparación con el número total de artículos.

Tenga en cuenta que la métrica de exactitud tiene limitaciones: no funciona bien con las clases desequilibradas que pueden tener muchos elementos de la misma clase e incluir algunas otras clases.

$$\text{accuracy} = \frac{TP + TN}{TP + TN + FP + FN}$$

Métricas de Rendimiento para problemas de clasificación

MÉTRICA DE EXHAUSTIVIDAD/SENSIBILIDAD

La métrica de exhaustividad muestra la cantidad de verdaderos positivos que el modelo ha clasificado en función del número total de valores positivos.

$$\text{recall} = \frac{TP}{TP + FN}$$

Métricas de Rendimiento para problemas de clasificación

MÉTRICA DE PRECISIÓN

Esta métrica representa el número de verdaderos positivos que son realmente positivos en comparación con el número total de valores positivos predichos.

$$\text{precision} = \frac{TP}{TP + FP}$$

Métricas de Rendimiento para problemas de clasificación

PUNTUACIÓN F1

Esta métrica es la combinación de las métricas de precisión y exhaustividad y sirve de compromiso entre ellas. La mejor puntuación F1 es igual a 1 y la peor a 0.

$$F1 = 2 * \frac{\text{precision} * \text{recall}}{\text{precision} + \text{recall}}$$

Métricas de Rendimiento para problemas de regresion

ERROR MEDIO ABSOLUTO (EMA)

Esta métrica de regresión es el valor medio de la diferencia absoluta entre el valor real y el valor predicho.

$$MAE = \frac{1}{n} \sum_{t=1}^n |e_t|, \text{ where } e_t = \text{original}_t - \text{predict}_t$$

Métricas de Rendimiento para problemas de regresion

ERROR CUADRÁTICO MEDIO (ECM)

El error cuadrático medio (ECM) calcula el valor medio de la diferencia al cuadrado entre el valor real y el predicho para todos los puntos de datos. Todos los valores relacionados se elevan a la segunda potencia, por lo tanto, todos los valores negativos no se compensan con los positivos. Además, debido a las características de esta métrica, el impacto de los errores es mayor. Por ejemplo, si el error en nuestros cálculos iniciales es de 1/2/3, el ECM será igual a 1/4/9 respectivamente. Cuanto menor sea el ECM, más precisas serán nuestras predicciones. ECM = 1 es el punto óptimo en el que nuestro pronóstico es perfectamente preciso.

$$MSE = \frac{1}{n} \sum_{t=1}^n e_t^2, \text{ where } e_t = \text{original}_t - \text{predict}_t$$

Métricas de Rendimiento para problemas de regresion

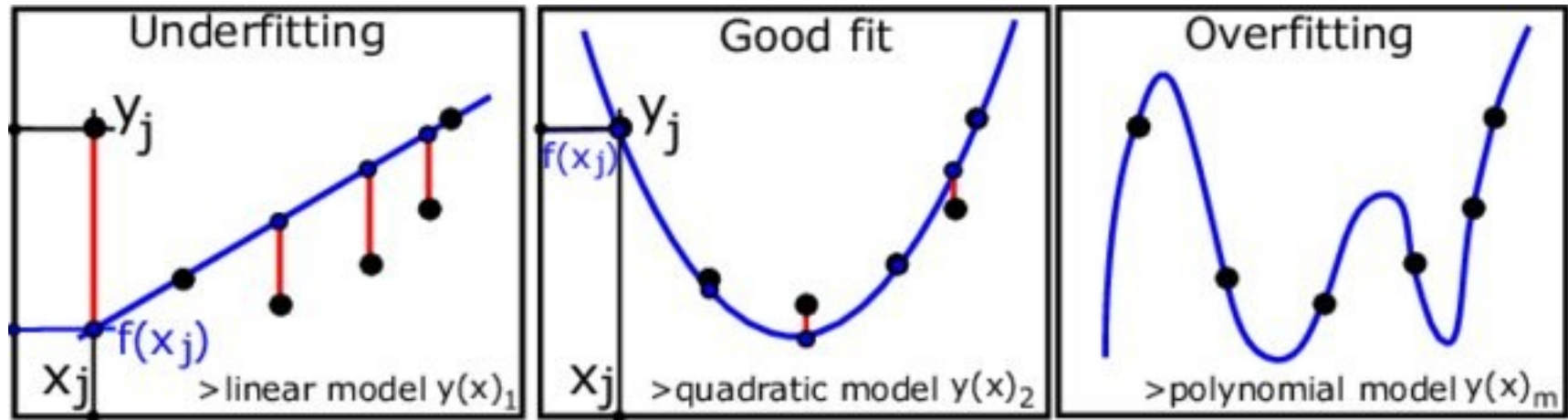
RAÍZ DEL ERROR CUADRÁTICO MEDIO (RECM)

El RECM es la raíz cuadrada del ECM. Es fácil de interpretar en comparación con el ECM y utiliza valores absolutos más pequeños, lo que es útil para los cálculos informáticos.

$$RMSE = \sqrt{\frac{1}{n} \sum_{t=1}^n e_t^2}, \text{ where } e_t = \text{original}_t - \text{predict}_t$$

Overfitting – Underfitting en el Aprendizaje Automatico

Las principales causas al obtener malos resultados en Machine Learning son el overfitting o el underfitting de los datos. Cuando entrenamos nuestro modelo intentamos “hacer encajar” -fit en inglés- los datos de entrada entre ellos y con la salida. Tal vez se pueda traducir overfitting como “sobreajuste” y underfitting como “subajuste” y hacen referencia al fallo de nuestro modelo al generalizar -encajar- el conocimiento que pretendemos que adquieran.



Underfitting

Entreno al modelo con
1 sólo raza de perro



Muestra nueva:
¿Es perro?



NO



La máquina fallará en reconocer al perro por falta de
suficientes muestras. No puede generalizar el conocimiento.

Overfitting

Entreno al modelo con
10 razas de perro color marrón



Muestra nueva:
¿Es perro?



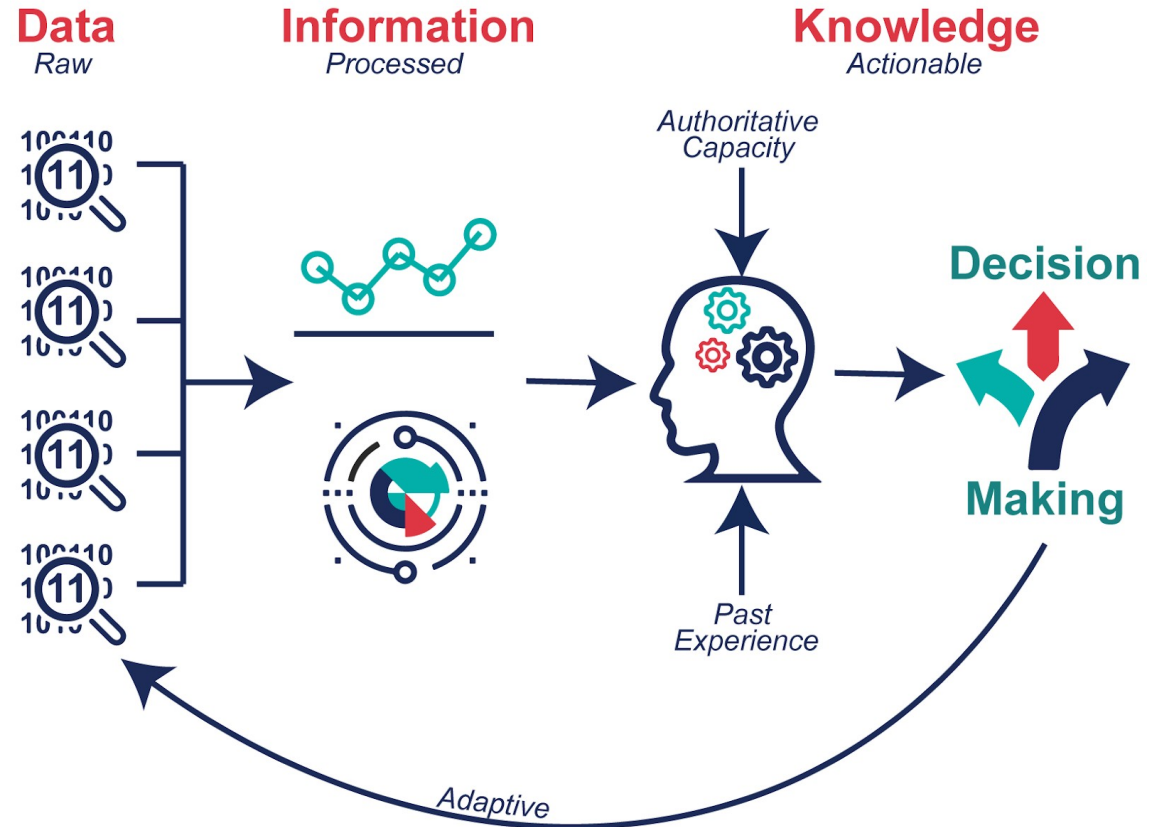
NO



La máquina fallará en reconocer un perro nuevo porque no tiene estrictamente los mismos valores de las muestras de entrenamiento.

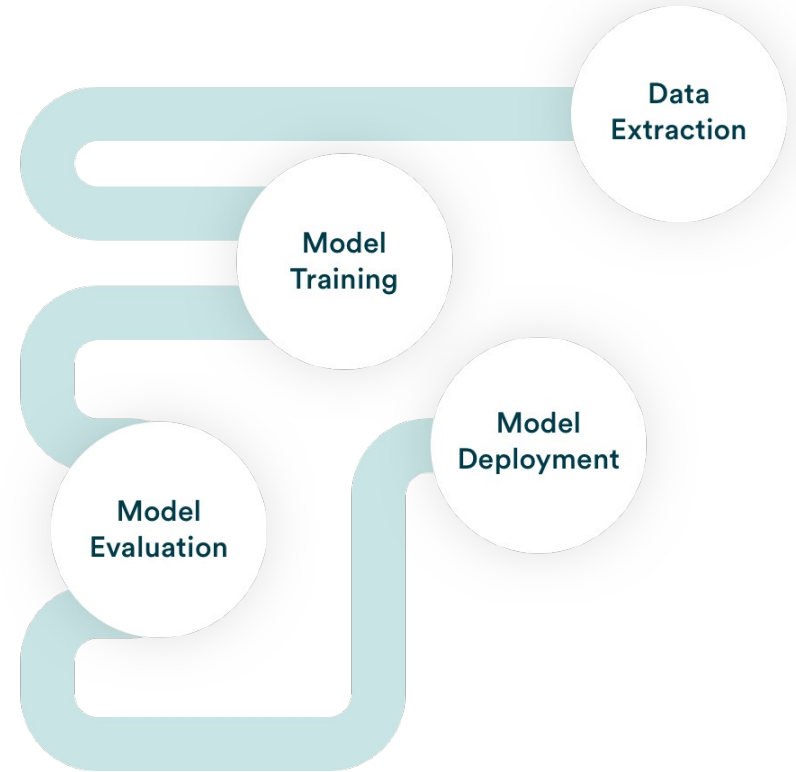
Generalización del conocimiento

Como si se tratase de un ser humano, las máquinas de aprendizaje deberán ser capaces de generalizar conceptos. Supongamos que vemos un perro Labrador por primera vez en la vida y nos dicen “eso es un perro”. Luego nos enseñan un perrito criollo y nos preguntan: ¿eso es un perro? Diremos “No”, pues no se parece en nada a lo que aprendimos anteriormente. Ahora imaginemos que nuestro tutor nos muestra un libro con fotos de 10 razas de perros distintas. Cuando veamos una raza de perro que desconocíamos seguramente seremos capaces de reconocer al cuadrúpedo canino al tiempo de poder discernir en que un gato no es un perro, aunque sea peludo y tenga 4 patas.



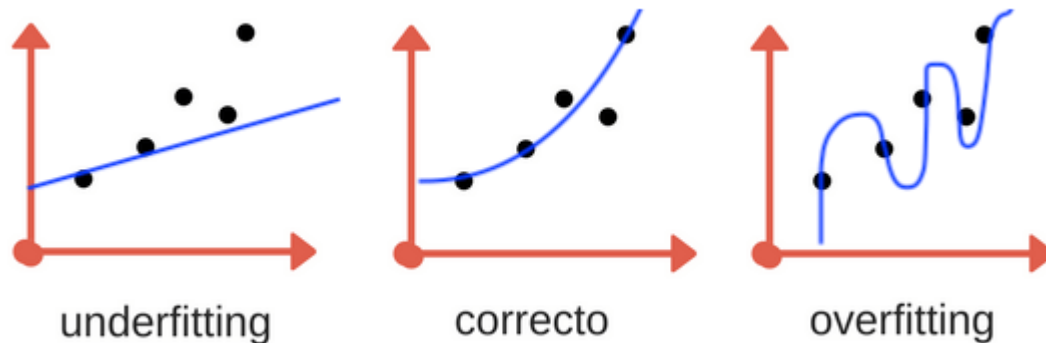
Generalización del conocimiento

Cuando entrenamos nuestros modelos computacionales con un conjunto de datos de entrada estamos haciendo que el algoritmo sea capaz de generalizar un concepto para que al consultarle por un nuevo conjunto de datos desconocido éste sea capaz de sintetizarlo, comprenderlo y devolvernos un resultado fiable dada su capacidad de generalización.



El problema de la Máquina al Generalizar

Si nuestros datos de entrenamiento son muy pocos nuestra máquina no será capaz de generalizar el conocimiento y estará incurriendo en underfitting. Este es el caso en el que le enseñamos sólo una raza de perros y pretendemos que pueda reconocer a otras 10 razas de perros distintas. El algoritmo no será capaz de darnos un resultado bueno por falta de “materia prima” para hacer sólido su conocimiento. También es ejemplo de “subajuste” cuando la máquina reconoce todo lo que “ve” como un perro, tanto una foto de un gato o un coche.



El problema de la Máquina al Generalizar

Por el contrario, si entrenamos a nuestra máquina con 10 razas de perros sólo de color marrón de manera rigurosa y luego enseñamos una foto de un perro blanco, nuestro modelo no podrá reconocerlo cómo perro por no cumplir exactamente con las características que aprendió (el color forzosamente debía ser marrón). Aquí se trata de un problema de overfitting.

Tanto el problema del ajuste “por debajo” como “por encima” de los datos son malos porque no permiten que nuestra máquina generalice el conocimiento y no nos darán buenas predicciones (o clasificación, o agrupación, etc.)

El problema de la Máquina al Generalizar

Overfitting en Machine Learning

Es muy común que al comenzar a aprender machine learning caigamos en el problema del Overfitting. Lo que ocurrirá es que nuestra máquina sólo se ajustará a aprender los casos particulares que le enseñamos y será incapaz de reconocer nuevos datos de entrada. En nuestro conjunto de datos de entrada muchas veces introducimos muestras atípicas (ó anomalas) o con “ruido/distorsión” en alguna de sus dimensiones, o muestras que pueden no ser del todo representativas. Cuando “sobre-entrenamos” nuestro modelo y caemos en el overfitting, nuestro algoritmo estará considerando como válidos sólo los datos idénticos a los de nuestro conjunto de entrenamiento –incluidos sus defectos– y siendo incapaz de distinguir entradas buenas como fiables si se salen un poco de los rangos ya preestablecidos.

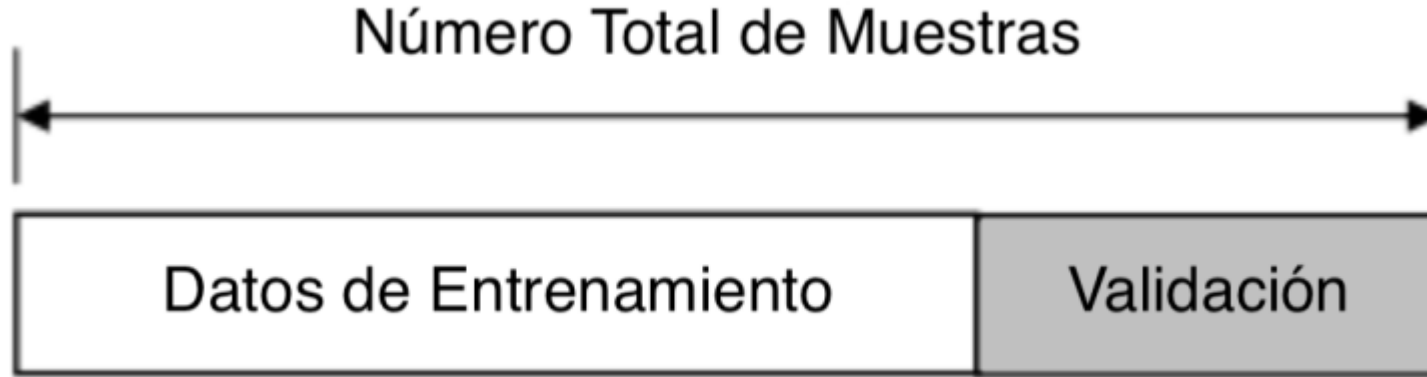
El problema de la Máquina al Generalizar

El equilibrio del Aprendizaje

Deberemos encontrar un punto medio en el aprendizaje de nuestro modelo en el que no estemos incurriendo en underfitting y tampoco en overfitting. A veces esto puede resultar una tarea muy difícil.

Para reconocer este problema deberemos subdividir nuestro conjunto de datos de entrada para entrenamiento en dos: uno para entrenamiento y otro para la Test que el modelo no conocerá de antemano. Esta división se suele hacer del 80% para entrenar y 20%. El conjunto de Test deberá tener muestras diversas en lo posible y una cantidad de muestras suficiente para poder comprobar los resultados una vez entrenado el modelo.

El problema de la Máquina al Generalizar



Cuando entrenamos nuestro modelo solemos parametrizar y limitar el algoritmo, por ejemplo la cantidad de iteraciones que tendrá o un valor de “tasa de aprendizaje” (learning-rate) por iteración y muchos otros. Para lograr que nuestro modelo dé buenos resultados iremos revisando y contrastando nuestro entrenamiento con el conjunto de Test y su tasa de errores, utilizando más o menos iteraciones, etc. hasta dar con buenas predicciones y sin tener los problemas de over-under-fitting.

Prevenir el Sobreajuste de datos

Para intentar que estos problemas nos afecten lo menos posible, podemos llevar a cabo diversas acciones.

- **Cantidad mínima de muestras** tanto para entrenar el modelo como para validarlo.
- **Clases variadas y equilibradas en cantidad**: En caso de aprendizaje supervisado y suponiendo que tenemos que clasificar diversas clases o categorías, es importante que los datos de entrenamiento estén balanceados. Supongamos que tenemos que diferenciar entre manzanas, peras y bananas, debemos tener muchas fotos de las 3 frutas y en cantidades similares. Si tenemos muy pocas fotos de peras, esto afectará en el aprendizaje de nuestro algoritmo para identificar esa fruta.

Prevenir el Sobreajuste de datos

- **Conjunto de Test de datos**. Siempre subdividir nuestro conjunto de datos y mantener una porción del mismo “oculto” a nuestra máquina entrenada. Esto nos permitirá obtener una valoración de aciertos/fallos real del modelo y también nos permitirá detectar fácilmente efectos del overfitting /underfitting.
- **Parameter Tuning o Ajuste de Parámetros**: deberemos experimentar sobre todo dando más/menos “tiempo/iteraciones” al entrenamiento y su aprendizaje hasta encontrar el equilibrio.

Prevenir el Sobreajuste de datos

- **Cantidad excesiva de Dimensiones (features)**, con muchas variantes distintas, sin suficientes muestras. A veces conviene eliminar o reducir la cantidad de características que utilizaremos para entrenar el modelo. Una herramienta útil para hacerlo es PCA.
- Quiero notar que si nuestro modelo es una red neuronal artificial –deep learning-, podemos caer en overfitting **si usamos capas ocultas en exceso**, ya que haríamos que el modelo memorice las posibles salidas, en vez de ser flexible y adecuar las activaciones a las entradas nuevas.

Prevenir el Sobreajuste de datos

Si el modelo entrenado con el conjunto de train tiene un 90% de aciertos y con el conjunto de test tiene un porcentaje muy bajo, esto señala claramente un problema de overfitting.

Si en el conjunto de Test sólo se acierta un tipo de clase (por ejemplo “peras”) o el único resultado que se obtiene es siempre el mismo valor será que se produjo un problema de underfitting.

- <https://towardsdatascience.com/various-ways-to-evaluate-a-machine-learning-models-performance-230449055f15>
- <https://towardsdatascience.com/metrics-to-evaluate-your-machine-learning-algorithm-f10ba6e38234>
- <https://heartbeat.fritz.ai/introduction-to-machine-learning-model-evaluation-fa859e1b2d7f>
- <https://machinelearningmastery.com/how-to-know-if-your-machine-learning-model-has-good-performance/>
- <https://www.datasource.ai/es/data-science-articles/metricas-de-evaluacion-de-modelos-en-el-aprendizaje-automatico>
- <https://cloud.ibm.com/docs/watson-knowledge-studio?topic=watson-knowledge-studio-evaluate-ml&locale=es>
- <https://stanford.edu/~shervine//es/teaching/cs-229/hoja-referencia-aprendizaje-automatico-consejos-trucos>
- <https://www.juanbarrios.com/la-matriz-de-confusion-y-sus-metricas/>
-
-