

Aprendizaje automático (Machine Learning)

Fundamentos del Machine
Learning

Clase 2

MAESTRÍA EN
SISTEMAS DE INFORMACIÓN
Mención Data Science

La excelencia no se improvisa



1. INTRODUCCIÓN DE LA CLASE

Machine learning o aprendizaje automático es un campo fundamental dentro de la inteligencia artificial, que permite a las máquinas aprender patrones y hacer predicciones basadas en datos sin intervención humana directa. Este proceso implica tres conceptos claves: **datos**, **modelos** y **algoritmos**. Los datos son el insumo que permite entrenar modelos matemáticos que, a través de algoritmos, pueden identificar patrones y tomar decisiones predictivas. Existen diferentes tipos de aprendizaje, como el **supervisado**, donde se entrenan los modelos con ejemplos etiquetados, el **no supervisado**, que busca descubrir patrones ocultos sin etiquetas previas, y el **aprendizaje por refuerzo**, donde los agentes aprenden a través de prueba y error (Goodfellow, Bengio & Courville, 2016). Estos conceptos son esenciales para el desarrollo de sistemas inteligentes que pueden aprender de la experiencia, sin requerir instrucciones explícitas.

Sin embargo, el desarrollo y la aplicación de modelos de *machine learning* también enfrentan varios problemas. Entre los más comunes se encuentran el **sobreajuste (*overfitting*)**, que ocurre cuando el modelo aprende demasiado bien los datos de entrenamiento, pero no generaliza adecuadamente a datos nuevos. Además, la calidad y la cantidad de los datos son fundamentales; la falta de datos etiquetados de alta calidad puede limitar la capacidad de los algoritmos para aprender correctamente. Un algoritmo ampliamente utilizado en *machine learning* es el de **k-vecinos más cercanos (k-NN)**, que es un enfoque de clasificación basado en la proximidad entre puntos de datos en un espacio de características. El algoritmo de k-NN es sencillo y eficiente, y se utiliza en una variedad de aplicaciones, como la clasificación de imágenes o la predicción de preferencias de usuarios (Hastie, Tibshirani & Friedman, 2009). Aunque es intuitivo, su rendimiento depende de la elección adecuada del valor de k y de las métricas de distancia utilizadas.

Clase 2. Fundamentos del *Machine Learning*

El *machine learning* (aprendizaje automático) es una rama de la inteligencia artificial que se centra en el desarrollo de algoritmos y modelos capaces de aprender y tomar decisiones a partir de datos. Sus fundamentos se basan en el análisis estadístico, la optimización y la computación, permitiendo a las máquinas identificar patrones y realizar predicciones sin ser programadas explícitamente para cada tarea. Entre las técnicas más utilizadas se encuentran el aprendizaje supervisado, no supervisado y por refuerzo, cada una con aplicaciones específicas en diversas áreas, como la medicina, la economía y el reconocimiento de imágenes. La capacidad del *machine learning* para generalizar a partir de datos previamente observados lo convierte en una herramienta esencial en el análisis de grandes volúmenes de datos en la actualidad (Goodfellow, Bengio & Courville, 2016; Bishop, 2006).

1.1. Conceptos claves del *machine learning*

El *machine learning*, también conocido como aprendizaje automático, es una disciplina que permite a las máquinas aprender y realizar predicciones basadas en datos. Para entender esta tecnología, es fundamental abordar los siguientes conceptos clave:

- **Aprendizaje supervisado.** Este enfoque consiste en entrenar un modelo con datos etiquetados, donde cada entrada tiene una salida conocida. Es especialmente útil para tareas como la clasificación y la regresión. Por ejemplo, la detección de correos electrónicos *spam* utiliza este tipo de aprendizaje (Bishop, 2006).

Para tareas de clasificación, la variable a predecir es una variable discreta; mientras que para tareas de regresión, la variable a predecir tiene valores continuos. La siguiente imagen ilustra un problema de clasificación básico, en donde se define un conjunto de entrenamiento, que será el *input* para el modelo (instancias de ejemplo y el *label* asociado) y un *classifier*, el cual entrena al modelo a base del *input* y permite generar la predicción o *label*.

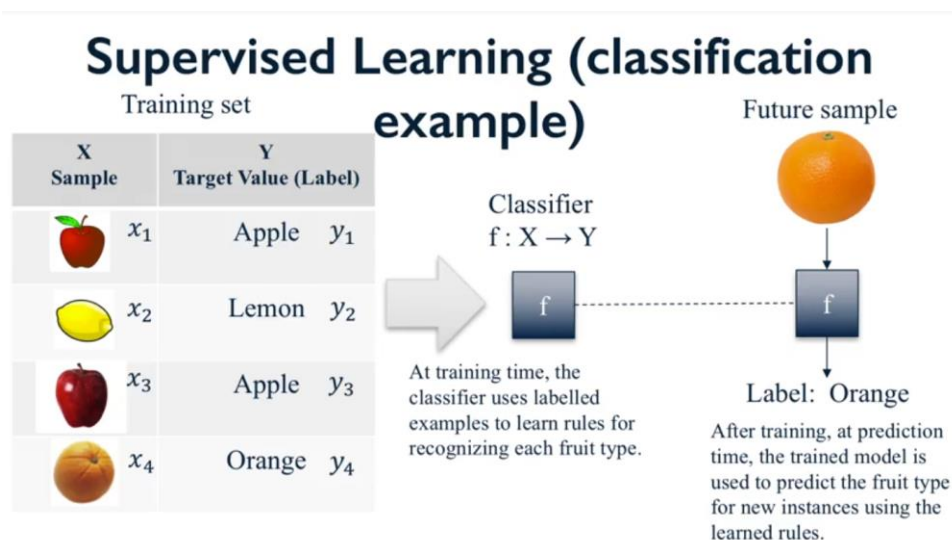


Figura N.º 1. Aprendizaje supervisado. Ejemplo de clasificación.

- **Aprendizaje no supervisado.** A diferencia del aprendizaje supervisado, este no requiere datos etiquetados. Se utiliza para descubrir patrones ocultos o estructuras intrínsecas en los datos, como la agrupación de clientes en función de su comportamiento de compra (Goodfellow, Bengio & Courville, 2016) o encontrar patrones inusuales (por ejemplo detección de *outliers*).

En otras palabras, consiste en encontrar una estructura de datos *no etiquetados*. La siguiente imagen ilustra el concepto básico de algoritmo de aprendizaje no supervisado.

Unsupervised learning: finding useful structure or knowledge in data when no labels are available

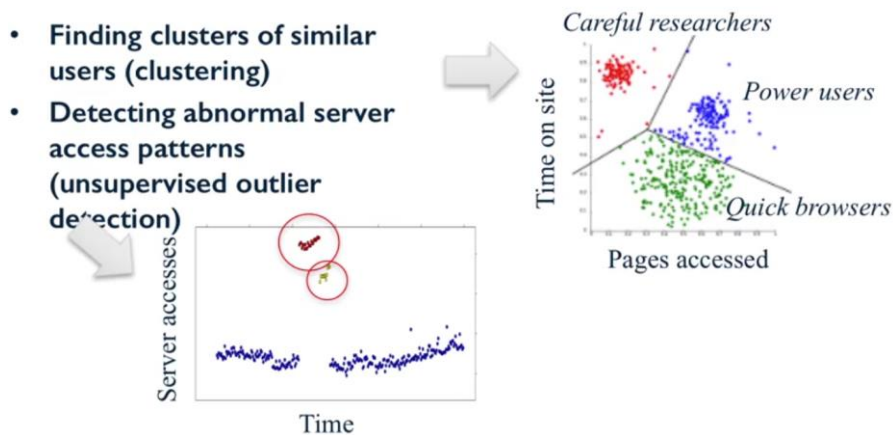


Figura N.º 2. Aprendizaje no supervisado. Buscar conocimiento en datos no etiquetados.

- **Aprendizaje por refuerzo.** Este tipo de aprendizaje se basa en la interacción con un entorno dinámico, donde el modelo aprende a tomar decisiones para maximizar una recompensa acumulativa. Es común en aplicaciones como los videojuegos y la robótica (Sutton & Barto, 2018).
- **Modelos de *machine learning*.** Los modelos representan la base del aprendizaje automático. Algunos de los más utilizados son los árboles de decisión, las máquinas de soporte vectorial (SVM), las redes neuronales y los modelos de ensemble. Cada uno de ellos tiene ventajas y limitaciones que los hacen adecuados para diferentes escenarios (Hastie, Tibshirani & Friedman, 2009).
- **Sobreajuste y subajuste.** Estos conceptos están relacionados con la capacidad de un modelo para generalizar. El sobreajuste ocurre cuando un modelo aprende demasiado bien los detalles del conjunto de entrenamiento, perdiendo capacidad de predicción en nuevos datos. El subajuste, por otro lado, ocurre cuando el modelo es demasiado simple y no logra capturar patrones importantes en los datos (Goodfellow et al., 2016).
- **Preprocesamiento de datos.** Antes de entrenar un modelo, es crucial realizar tareas de limpieza y transformación de datos. Esto incluye manejar valores nulos, normalizar variables y transformar datos categóricos en numéricos. Un preprocesamiento adecuado mejora el desempeño del modelo y la calidad de las predicciones (Han, Kamber & Pei, 2012).

- **Métricas de evaluación.** La efectividad de un modelo se mide mediante métricas como la precisión, la sensibilidad, la especificidad y el área bajo la curva ROC. La elección de la métrica adecuada depende del problema específico que se esté resolviendo (Hastie et al., 2009).
- **Conjuntos de datos para problemas de aprendizaje supervisado.** La librería Scikit-learn de Python provee un gran número de *datasets* que pueden ser usados para entender con mayor profundidad los algoritmos de aprendizaje supervisado. A continuación, se muestra un ejemplo:

```
import matplotlib.pyplot as plt
from sklearn.datasets import make_friedman1
plt.figure()
plt.title('Complex regression problem with one input variable')
X_F1, y_F1 = make_friedman1(n_samples = 100,
                             n_features = 7, random_state=0)

plt.scatter(X_F1[:, 2], y_F1, marker= 'o', s=50)
plt.show()
```

Figura N.º 3. Uso del *dataset* `make_friedman1`.

También existen otras páginas para trabajar con *datasets* para problemas de clasificación o regresión:

- UCI: <https://archive.ics.uci.edu/datasets>
- Kaggle: <https://www.kaggle.com/datasets>

1.2. Problemas del *machine learning*

El campo del *machine learning* ha avanzado significativamente en las últimas décadas, siendo crucial en diversas aplicaciones, desde la predicción de comportamientos hasta el desarrollo de sistemas autónomos. Sin embargo, existen problemas y desafíos que pueden afectar tanto a la eficacia como a la aplicabilidad de los modelos de *machine learning*. A continuación, se exploran algunos de los problemas más comunes.

- Generalización

Se refiere a la habilidad del algoritmo para devolver predicciones adecuadas para una instancia nueva o no conocida para el modelo. Para llegar a esto se parte de las siguientes suposiciones:

- Los datos que el modelo no conoce (*test data*) tendrán las mismas propiedades que el conjunto de datos usado para entrenar

- Si un modelo es preciso con datos de entrenamiento también lo será con datos de prueba, pero esto posiblemente no pase si el modelo entrenado es muy específico y se apega mucho a los datos de entrenamiento.

- **Sobreajuste (*overfitting*)**

Uno de los problemas más comunes en *machine learning* es el sobreajuste, que ocurre cuando un modelo aprende demasiado bien los detalles y el ruido de los datos de entrenamiento, en lugar de generalizar sobre los datos que no ha visto antes. Esto significa que el modelo puede tener un rendimiento excelente en los datos de entrenamiento, pero su desempeño en datos nuevos o no vistos es pobre. El sobreajuste es más probable cuando el modelo es demasiado complejo o cuando los datos de entrenamiento son insuficientes o no representativos (James, Witten, Hastie, & Tibshirani, 2013).

Cuando los modelos son demasiado complejos para la cantidad de datos que se tiene disponible tienden a sobreajustar y no generalizar correctamente nuevos ejemplos. La siguiente imagen ilustra el problema de *overfitting* en un ejercicio de regresión

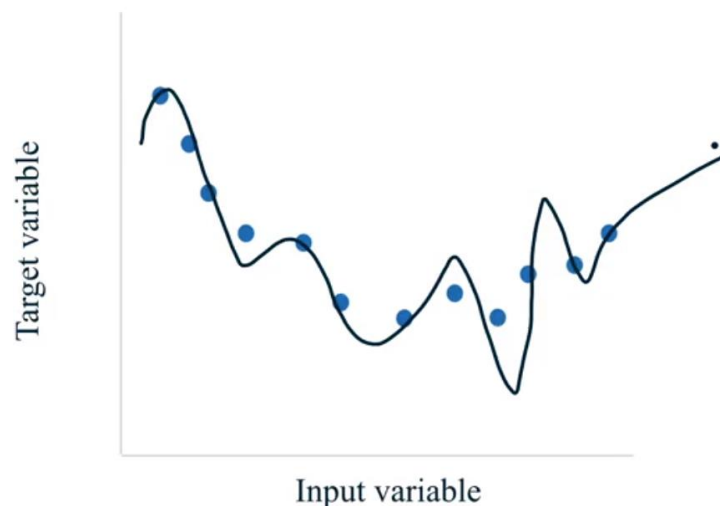


Figura N.º 4. Aprendizaje supervisado. Problemas de *overfitting*.

Si el modelo predice los mismos valores o valores muy parecidos a los que tiene la variable *target* y no puede generalizar correctamente, se dice que el modelo tiene problemas de *overfitting*.

- **Subajuste (*underfitting*)**

El *underfitting* o subajuste **ocurre cuando un modelo no logra captar la complejidad subyacente en los datos**. Este problema se presenta cuando el modelo es demasiado simple para representar adecuadamente las relaciones entre las variables. Como resultado, el modelo muestra un mal desempeño tanto en los datos de entrenamiento como en los datos de prueba. El subajuste es típico cuando se emplean modelos demasiado sencillos o cuando no se entrenan adecuadamente (Hastie, Tibshirani, & Friedman, 2009).

Cuando los modelos son tan simples, que no llegan ni siquiera a *performar* bien en los datos de entrenamiento, se dice que estos modelos tienen un subajuste y en consecuencia no generalizan bien. La siguiente imagen ilustra el problema del subajuste para un ejercicio básico de clasificación.

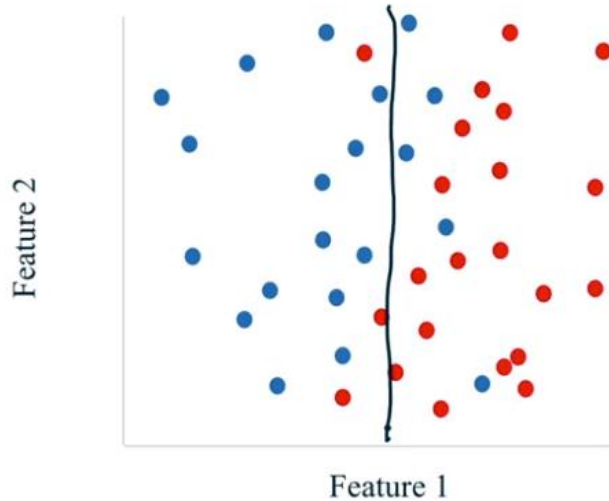


Figura N.º 5. Aprendizaje supervisado. Problemas de *underfitting*.

La siguiente imagen muestra una clasificación de los puntos bastante simple, ya que no llega a determinar los patrones en los datos. En este caso, la línea que separa los puntos debería tener cierto nivel de inclinación a la derecha.

En la siguiente figura se aprecia un modelo con una buena clasificación, pues logra encontrar los patrones escondidos en los datos y adicionalmente ignora el hecho de que pueden existir errores mínimos de clasificación (es el caso del punto rojo clasificado como negativo y algunos puntos azules clasificados como positivos).

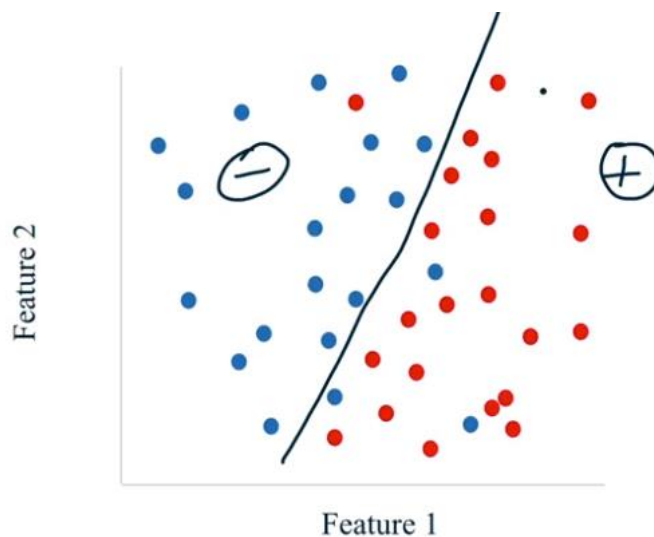


Figura N.º 6. Aprendizaje supervisado. Buena clasificación.

- **Falta de datos de calidad**

El rendimiento de un modelo de *machine learning* depende en gran medida de la calidad de los datos con los que se entrena. Si los datos están sesgados, incompletos o contienen errores, el modelo aprenderá patrones incorrectos o inapropiados. La falta de datos representativos o diversos también puede llevar a que el modelo no generalice bien a situaciones del mundo real, lo que limita su aplicabilidad y efectividad (Deng & Yu, 2014).

- **Datos faltantes en registros**

Un ejemplo común de falta de datos ocurre cuando algunas observaciones carecen de valores en ciertas características. Por ejemplo, en un conjunto de datos médicos es posible que las mediciones de presión arterial no estén registradas para algunos pacientes, debido a errores de registro o a condiciones específicas de los pacientes. Esta carencia puede llevar a problemas durante el entrenamiento, ya que el modelo no tiene información completa para aprender las relaciones entre las variables dependientes e independientes (García et al., 2022).

- **Distribución de datos no representativa**

Cuando los datos de entrenamiento no reflejan adecuadamente la población objetivo, el modelo puede ser parcial o ineficaz. Por ejemplo, un modelo de clasificación de correos electrónicos como *spam* entrenado exclusivamente con datos de usuarios de una región específica, como Estados Unidos, podría fallar al aplicarse en regiones con diferentes patrones lingüísticos o culturales. Este problema es particularmente relevante en aplicaciones globales de *machine learning* (Smith & Johnson, 2020).

- **Variables con alta proporción de valores ausentes**

En muchos *datasets*, algunas variables tienen una proporción significativa de datos faltantes. Por ejemplo, en conjuntos de datos inmobiliarios, la columna 'fecha de renovación' podría estar vacía en más del 50 % de los registros, debido a que no todas las propiedades se han renovado. Este escenario puede hacer que la variable sea inútil para el modelo o que se requieran técnicas complejas de imputación (Zhang et al., 2021).

- **Datos históricos incompletos**

En aplicaciones donde se analizan series temporales, como la previsión de demanda o análisis de tráfico, es común encontrar lagunas en los registros históricos. Por ejemplo, los datos de ventas de un comercio podrían estar incompletos debido a fallos en los sistemas de recolección de datos o errores humanos. Estas lagunas pueden complicar la identificación de patrones temporales y reducir la eficacia del modelo (Hernández et al., 2019).

- **Desbalanceo de clases**

En tareas de clasificación, como la detección de fraudes, el desbalanceo de clases es una forma de falta de datos que ocurre cuando una clase está representada significativamente menos que otra. Por ejemplo, en un conjunto de datos de transacciones bancarias, los casos

de fraude suelen ser escasos en comparación con las transacciones legítimas, lo que puede llevar a que el modelo tenga dificultades para identificar correctamente las instancias minoritarias (Fernández et al., 2020).

- **Problemas éticos y de privacidad**

Los sistemas de *machine learning* pueden generar preocupaciones éticas, especialmente en áreas como la toma de decisiones autónoma y el uso de datos personales. Por ejemplo, los algoritmos pueden aprender y perpetuar sesgos sociales y raciales presentes en los datos, lo que puede dar lugar a decisiones discriminatorias. Además, la recolección y el uso de grandes volúmenes de datos personales plantean serias inquietudes sobre la privacidad y la protección de la información (O'Neil, 2016).

- **Interpretabilidad**

Uno de los principales desafíos actuales en el campo de *machine learning* es la falta de interpretabilidad en muchos modelos, especialmente en los basados en redes neuronales profundas (*deep learning*). La 'caja negra' de estos modelos dificulta la comprensión de cómo se toman las decisiones, esto limita su implementación en sectores que requieren explicaciones claras, como la medicina y las finanzas. La falta de transparencia puede generar desconfianza entre los usuarios y reducir la aceptación de estas tecnologías en aplicaciones críticas (Ribeiro, Singh, & Guestrin, 2016).

En la siguiente figura se muestra un *boxplot*, es un gráfico muy ilustrativo y muy fácil de usar.

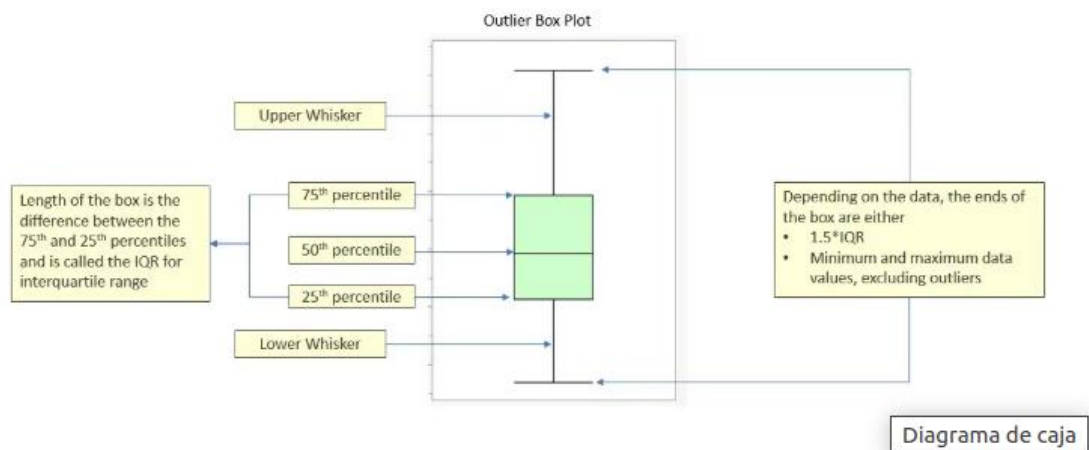


Figura N.º 7. Boxplot. Fuente https://www.jmp.com/es_mx/statistics-knowledge-portal/exploratory-data-analysis/box-plot.html

- **Escalabilidad y consumo de recursos**

Los modelos de *machine learning*, especialmente los más complejos, pueden requerir grandes cantidades de datos y recursos computacionales. El entrenamiento de modelos avanzados, como las redes neuronales profundas, demanda un poder de cómputo significativo y tiempo de procesamiento. Esto puede resultar en altos costos operativos y dificultar la implementación de modelos en entornos con recursos limitados (Goodfellow, Bengio, & Courville, 2016).

- **Generalización a nuevas situaciones**

La capacidad de un modelo para generalizar nuevas situaciones o datos no vistos es uno de los problemas clave en *machine learning*. A pesar de los avances en el desarrollo de algoritmos, muchos modelos aún tienen dificultades para adaptarse a cambios en los datos o a situaciones no representadas en los datos de entrenamiento. Esto es especialmente problemático en aplicaciones en las que las condiciones cambian rápidamente, como en la predicción de tendencias del mercado financiero o en el control de sistemas autónomos (Schmidhuber, 2015).

Aspectos interesantes el machine learning

[<https://homes.cs.washington.edu/~pedrod/papers/cacm12.pdf>]

- **El *trade-off* entre la capacidad predictiva y la capacidad descriptiva de un modelo de *machine learning***

La siguiente ilustración muestra los diferentes tipos de analítica en función del valor que brinda al usuario y de la complejidad.



Figura N.º 8. Tipos de analítica. Fuente: <https://gravitar.biz/bi/tipos-analitica/>

En el campo del aprendizaje automático (*machine learning*), uno de los dilemas fundamentales a la hora de construir un modelo es el *trade-off* entre la **capacidad predictiva** y la **capacidad descriptiva** del modelo. Este dilema implica una elección entre la precisión en la predicción de futuros resultados y la capacidad de **entender** y **explicar** los patrones subyacentes en los datos. En este contexto, es esencial comprender cómo estos

dos objetivos pueden influir en la construcción y evaluación de modelos de *machine learning*.

- **Capacidad predictiva**

La **capacidad predictiva** de un modelo se refiere a su habilidad para generar **predicciones precisas** sobre datos no observados previamente. Es decir, un modelo con alta capacidad predictiva es capaz de **generalizar** correctamente a nuevos datos, lo que significa que puede hacer predicciones con un **bajo error de generalización**. Este tipo de modelo es fundamental cuando el objetivo principal es **predecir** ciertos resultados de manera eficiente, sin importar si los datos subyacentes son complejos o no.

Modelos como las **redes neuronales profundas**, los **árboles de decisión** o los métodos de **ensemble** son ejemplos de modelos con alta capacidad predictiva, ya que tienen la capacidad de capturar patrones complejos en los datos. Sin embargo, uno de los costos de estos modelos más complejos es que suelen ser **menos interpretables**, lo que dificulta la comprensión directa de las relaciones entre las variables.

- **Capacidad descriptiva**

La **capacidad descriptiva** de un modelo se centra en la habilidad de explicar las **relaciones** entre las variables en un conjunto de datos. Un modelo descriptivo tiene como objetivo proporcionar una visión clara de cómo las diferentes características de los datos afectan a la variable objetivo. Esto es fundamental en contextos donde el objetivo es **comprender** los mecanismos detrás de los datos más que simplemente hacer predicciones.

Modelos como la **regresión lineal** o los **modelos de árboles de decisión simples** son ejemplos típicos de modelos descriptivos. Estos son fáciles de interpretar, ya que sus resultados pueden explicarse en términos de coeficientes o divisiones claras entre las características. Sin embargo, su capacidad para capturar patrones complejos en los datos es limitada, lo que puede afectar su rendimiento en términos de predicción.

- **El *trade-off*: descripción vs. predicción**

El ***trade-off*** entre la capacidad predictiva y la capacidad descriptiva surge cuando los modelos más complejos, que son mejores para la predicción, pierden **capacidad explicativa**. A medida que se aumenta la complejidad del modelo, su habilidad para **generalizar** y hacer **predicciones más precisas** mejora, pero la capacidad de describir e interpretar las relaciones subyacentes se reduce.

Por ejemplo, en modelos como las **redes neuronales profundas**, aunque se pueden obtener predicciones de alta calidad, es muy difícil entender cómo las características del modelo influyen en los resultados, lo que limita su **capacidad descriptiva**. En cambio, modelos más simples, como la **regresión lineal** o los **árboles de decisión pequeños**, pueden proporcionar una **interpretación clara** de las relaciones entre las variables, pero podrían no ser tan efectivos en capturar patrones complejos o no lineales en los datos.

Este *trade-off* también se ve reflejado en la problemática del **sobreajuste** (*overfitting*), donde un modelo demasiado complejo puede ajustarse demasiado bien a los datos de entrenamiento, mejorando su capacidad predictiva, pero perdiendo capacidad de **generalización** a nuevos datos.

1.3. K – vecinos más cercanos

Es un algoritmo de *machine learning* que puede ser usado tanto para clasificación como para regresión. Para modelos de clasificación, el algoritmo k-vecinos más cercanos funciona de la siguiente manera:

Dado un conjunto de entrenamiento X_{train} con *labels* y_{train} , y dada una instancia de datos a ser clasificada x_{test} , los pasos son los siguientes:

1. Buscar las instancias más similares (las cuales las llamaremos X_{NN}) a x_{test} que se encuentran en X_{train} .
2. Obtener los *labels* y_{NN} para cada una de las instancias en X_{NN} .
3. Predecir el *label* para x_{test} combinando los *labels* y_{NN} e.g. Por mayoría de votos.

El siguiente gráfico muestra un espacio de dos dimensiones para un problema de clasificación, si tenemos un registro nuevo, por ejemplo, las 'x' en el gráfico se le asignará la clasificación (0 o 1) de la instancia que se encuentre más próxima a ellos. En este caso, ambos puntos serán clasificados como clase 1. Este concepto aplica tanto para modelos de clasificación como para modelos de regresión.

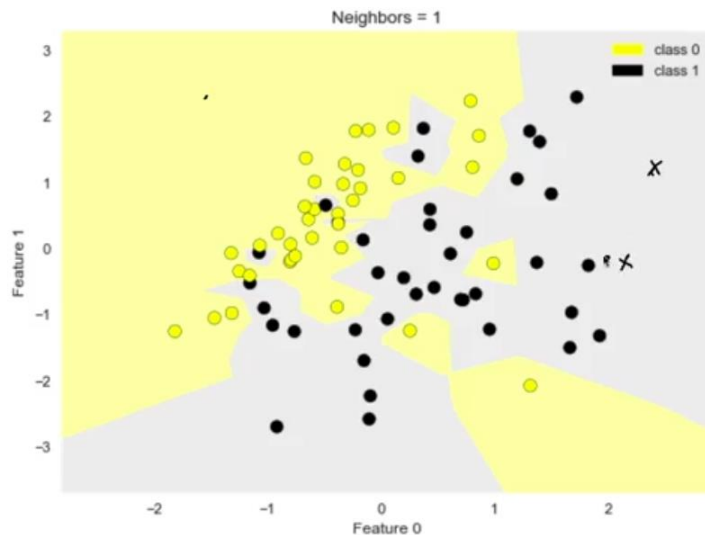


Figura N.º 9. K vecinos más cercanos. Ejemplo básico.

Algunos parámetros importantes para considerar tanto en modelos de clasificación como en regresión usando el algoritmo k – vecinos más cercanos:

La siguiente infografía muestra una estrategia para evitar la creación de modelos demasiado complejos:

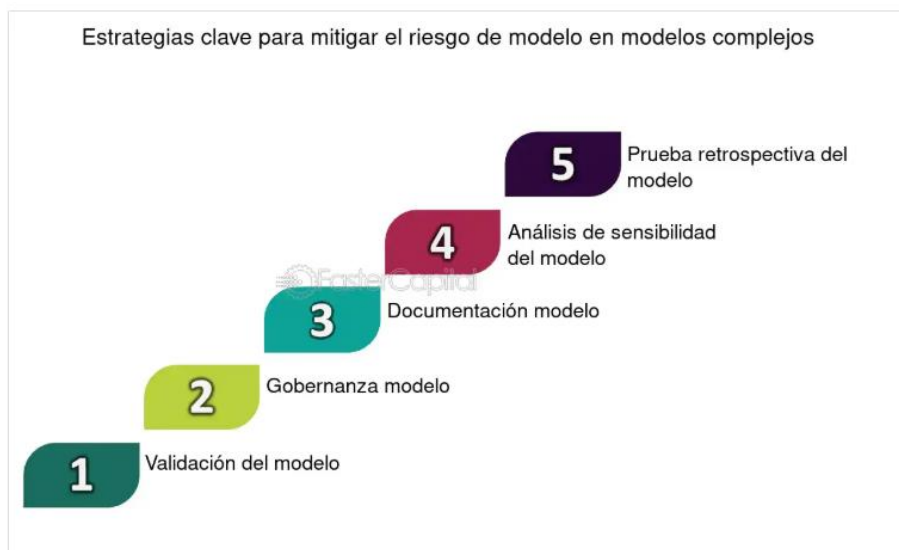


Figura N.º 10. Estrategias para evitar modelos complejos.

<https://fastercapital.com/es/contenido/Complejidad-del-modelo--gestion-del-riesgo-del-modelo-en-entornos-de-modelado-complejos.html>

Complejidad de los modelos

- n_neighbors: número de k-vecinos más cercanos a considerar, por defecto en la implementación de Python se usa el valor de 5.

Model fitting

- metric: función de distancia entre los puntos de datos: por defecto. en la implementación de Python se usa la distancia de Minkowski.

K vecinos más cercanos, aplicación [https://d1wqtxts1xzle7.cloudfront.net/34203825/e1-libre.pdf?1405405787=&response-content-disposition=inline%3B+filename%3DMineria_de_datos_prediccion_de_la_deserc.pdf&Expires=1738098637&Signature=Y2mSFkW36stTI0OD8gvgpZf-9oGOR2PnRWOKecXtcKEM~Z6WWiJS2o1zK0mMbTtwzfivkeP1JnErpUMlkkpKmWFceFwL81VRpovTSuMDD4bMm5Ii6Ux0UXEaeFusLPpIzEDZL39q9Syh~lN08lYmysnS9KIXOI55ySN NqolGrEuFF2CbvW-2Mta3CU8jOLNVrJ~pxIb6PwYtBaMHombLgJAZ-ZZlqpHaZxEca6YTf4RyMeayyu2Sa0GBTYAFZ~8Bg0W26xakn-hzevh0X6jBENwrS6CqmJTGKb4dsjZByK6S~AgaQAzp2f2VT5uSxt5RHHZXO34iqLlR1ukWAyhGA__&Key-Pair-Id=APKAJLOHF5GGSLRBV4ZA]

Referencias citadas en la Clase 2

- Bishop, C. M. (2006). *Pattern recognition and machine learning*. Springer.
- Goodfellow, I., Bengio, Y., & Courville, A. (2016). *Deep learning*. MIT Press.
- Han, J., Kamber, M., & Pei, J. (2012). *Data mining: Concepts and techniques*. Morgan Kaufmann.
- Hastie, T., Tibshirani, R., & Friedman, J. (2009). *The elements of statistical learning: Data mining, inference, and prediction*. Springer Science & Business Media.
- Deng, L., & Yu, D. (2014). *Deep learning: Methods and applications*. Foundations and Trends® in Signal Processing, 7(3–4), 197-387.
- James, G., Witten, D., Hastie, T., & Tibshirani, R. (2013). *An introduction to statistical learning: With applications in R*. Springer.
- O'Neil, C. (2016). *Weapons of math destruction: How big data increases inequality and threatens democracy*. Crown Publishing Group.
- Ribeiro, M. T., Singh, S., & Guestrin, C. (2016). "Why should I trust you?" Explaining the predictions of any classifier. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (pp. 1135-1144).

Definición de los términos citados en la Clase 2

Modelo complejo. Un modelo complejo en *machine learning* es aquel que tiene una estructura intrincada y usa un gran número de parámetros o capas (**como en redes neuronales profundas**), lo que le permite aprender patrones complejos en los datos. Estos modelos suelen ser más potentes en tareas difíciles, pero también son más propensos al **sobreajuste** (*overfitting*) y requieren más recursos computacionales para entrenar.

Modelo simple. Un modelo simple en *machine learning* es aquel que tiene una estructura sencilla y pocos parámetros, lo que facilita su **entrenamiento** e **interpretación**. Aunque es más fácil de ajustar y menos propenso al sobreajuste, su capacidad para aprender patrones complejos en los datos suele ser limitada, esto puede afectar su rendimiento en tareas más desafiantes.



La excelencia no se improvisa

síguenos

