

4. Antecedentes del Machine Learning (aprendizaje automático)

Desde hace décadas, se ha llegado a la conclusión de que **el aprendizaje es un elemento esencial de la inteligencia**. Esto se aplica tanto a la **inteligencia natural**, presente en los seres humanos y animales, como a la **inteligencia artificial**: en ambos casos, la capacidad de mejorar a partir de la experiencia es lo que permite adaptarse y tomar decisiones cada vez más acertadas.

El **aprendizaje automático** (*Machine Learning*) surge precisamente de esta idea. Se trata de un **conjunto de técnicas y métodos** que permiten que los ordenadores **aprendan a realizar tareas específicas** sin necesidad de ser programados explícitamente para cada una de ellas. En lugar de seguir instrucciones fijas, un sistema de aprendizaje automático **extrae patrones y regularidades** a partir de los datos, y los utiliza para **hacer predicciones o tomar decisiones**.

En términos generales, un **modelo de IA** aprende una **relación entre las entradas** (x_1, x_2, \dots) y **las salidas** (y) mediante un proceso de entrenamiento.

Dentro del Machine Learning existen diferentes tipos de aprendizaje, según cómo se entrena al modelo. Los principales son:

- **Aprendizaje supervisado**: el modelo aprende con datos etiquetados (se le dan ejemplos con las respuestas correctas).
- **Aprendizaje no supervisado**: el modelo aprende sin etiquetas, buscando patrones o grupos por sí mismo.

Como hemos visto en la parte anterior de la unidad, la evolución de la **Inteligencia Artificial (IA)** ha pasado por distintas etapas. En un principio, el objetivo era **imitar el comportamiento humano** mediante reglas lógicas y razonamiento simbólico. Sin embargo, con el tiempo, el enfoque se desplazó hacia la creación de sistemas capaces de **predecir el comportamiento más probable o más deseable** en función de los datos disponibles.

En este contexto, las técnicas más prometedoras han sido las que se basan en **modelos probabilísticos y estadísticos**, ya que permiten representar mejor la **incertidumbre y la variabilidad** de los fenómenos observados en el mundo real.

a. Estadística

La **estadística** desempeña un papel fundamental en el desarrollo del *Machine Learning*, ya que proporciona las herramientas necesarias para **extraer conocimiento a partir de los datos**.

Los modelos estadísticos permiten:

- Analizar y describir el comportamiento de los datos.
- Estimar relaciones entre variables.
- Medir la incertidumbre y la probabilidad de diferentes resultados.

De hecho, muchos de los algoritmos de aprendizaje automático actuales tienen su origen en **conceptos estadísticos clásicos**, como la **regresión**, la **inferencia bayesiana** o la **teoría de la probabilidad**.

Un ejemplo destacado es la **estadística bayesiana**, desarrollada hace más de dos siglos por el reverendo **Thomas Bayes**. Su principio central —la **actualización de las creencias** a medida que se obtiene nueva evidencia— sigue siendo **una base esencial del aprendizaje automático moderno**, especialmente en áreas como el filtrado de spam, la detección de anomalías o el reconocimiento de voz.

b. Probabilidad

¿En qué consiste la probabilidad?

El concepto de **probabilidad** está muy presente en nuestro lenguaje cotidiano. Habitualmente hablamos de “la probabilidad de que llueva mañana”, “la probabilidad de aprobar un examen” o “la probabilidad de que un equipo gane un partido”. Sin embargo, para comprender cómo funcionan los **algoritmos de aprendizaje automático basados en modelos probabilísticos**, necesitamos una definición **más formal y precisa**.

Definición formal

La **probabilidad** es una **medida numérica de la certeza o incertidumbre** asociada a la ocurrencia de un determinado suceso o evento. En otras palabras, cuantifica **qué tan probable es que ocurra algo**.

Se expresa mediante un **valor comprendido entre 0 y 1**:

- **0** significa que el evento es **imposible**.
- **1** significa que el evento es **seguro**.
- Los valores intermedios reflejan distintos grados de **verosimilitud o probabilidad**.

Aunque en la vida cotidiana solemos expresar las probabilidades en **porcentajes**, basta con multiplicar el valor decimal por 100. Por ejemplo, una probabilidad de 0,8 equivale al 80 %.

Valor de la probabilidad	Interpretación
0	Evento imposible
0,5	Evento equiprobable (mitad de las veces ocurre, mitad no)
1	Evento seguro

Origen

La teoría de la probabilidad surge del intento de **predecir y cuantificar la incertidumbre** en fenómenos del mundo real, especialmente en situaciones donde interviene el azar (por ejemplo, los juegos de azar, la meteorología o la genética). Su propósito es **estimar con**

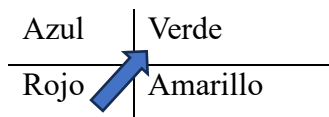
qué frecuencia puede ocurrir un evento, incluso cuando no se puede conocer el resultado exacto de antemano.

Importancia en el aprendizaje automático

En **Machine Learning**, los modelos probabilísticos utilizan este mismo principio para **estimar la probabilidad de diferentes resultados**. Por ejemplo, un modelo puede calcular la probabilidad de que un correo sea spam (0.95) o de que un paciente desarrolle una enfermedad cardíaca (0.30). El objetivo es **tomar decisiones informadas** basadas en el grado de certeza que ofrece cada predicción.

La **probabilidad** nos permite **cuantificar** la incertidumbre y razonar **matemáticamente** sobre eventos futuros. Esta capacidad es fundamental en la **Inteligencia Artificial**, ya que permite a los modelos **aprender de los datos** y **predecir comportamientos** en entornos donde la información es incompleta o incierta.

Ejemplo: Lanzamiento de flecha=experimento



Resultado Interés	Formas Resultado	Posibles Resultados	Probabilidad
Azul	1	4	1/4
...

Ejemplo: Lanzamiento dado



Probabilidad(5)=1/6
Probabilidad(PAR)=3/6

En los ejemplos anteriores lo que vemos es que todos los sucesos que tratamos de predecir son **igualmente probables**, es decir, que caiga una flecha en un color que en otro y que cuando yo lance un dado salga en un valor u otro.

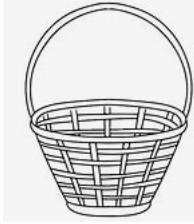
Sin embargo, nosotros podemos medir la probabilidad de que ocurra un suceso dentro de un conjunto de posibles sucesos que **no son igualmente probables**, es decir, vamos a tratar de predecir sucesos que **no tienen la misma probabilidad**.

Ejemplo: cesta de frutas

5 manzanas

3 peras

2 naranjas

Probabilidad(pera) = $3/10$ Probabilidad(naranja) = $2/10$ 

Todos estos experimentos los hemos visto por separado, como si no tuvieran relación entre sí. Pero en muchos casos es importante tener en cuenta que no es así, sino que dos resultados pueden estar conectados o influirse mutuamente. De aquí surge dos tipos de probabilidad:

- **Probabilidad condicional:** Eventos o sucesos que van a ser dependientes uno de otro. Trata de determinar la probabilidad de que se produzca un evento B a continuación de un evento A. Ejemplo: la probabilidad en el ejemplo de la cesta de frutas de sacar una manzana después de una pera.
- **Probabilidad conjunta:** Eventos independientes. Trata de determinar la probabilidad de que se produzcan un evento A y B al mismo tiempo. Ejemplo: que tire dos dados y me salgan dos cuatros.

Es importante comprender estos dos conceptos de probabilidad, ya que en muchos **problemas de Machine Learning** las variables o eventos que analizamos **no son independientes**, sino que **pueden influirse mutuamente**.

A continuación, veremos algunos ejemplos en los que será fundamental tener en cuenta esta **dependencia entre variables** para poder construir modelos más precisos y realistas.

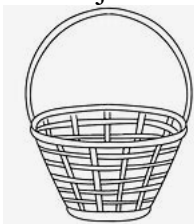
Ejemplo probabilidad condicional: cesto de frutas

3 manzanas

2 naranjas

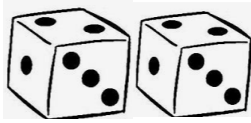
Probabilidad(naranja) = $2/5$

Ahora sacar otra naranja después de sacar previamente la primera naranja (dependiente):

Probabilidad(naranja) = $1/4$ 

Ejemplo probabilidad conjunta: lanzamiento de dos dados

Los eventos que vamos a ejecutar son independientes pero conjuntos:



Que salgan dos 5 a la vez en ambos dados:

Probabilidad(5)=1/6

Probabilidad(5)=1/6

Probabilidad(5,5)=Probabilidad(5)*Probabilidad(5)=1/6*1/6

Vamos a ver pues algoritmos basados en propiedades probabilísticas y vamos a hablar sobre la probabilidad condicional y conjunta.

Algoritmos probabilísticos

Los **algoritmos probabilísticos** son aquellos que construyen sus modelos basándose en **principios y propiedades de la probabilidad**. Su objetivo es estimar la **probabilidad de que un evento o clase ocurra** en función de los datos observados.

Este tipo de algoritmos se aplica en un **amplio rango de casos de uso**, especialmente en **problemas de clasificación**. Un ejemplo muy común es el **filtro de spam** en el correo electrónico: el modelo analiza las características de un mensaje (palabras utilizadas, remitente, formato, etc.) y calcula la **probabilidad de que sea spam o correo legítimo**, clasificándolo en consecuencia.

Una de las principales ventajas de estas técnicas es que **no requieren una gran capacidad computacional**. A diferencia de los modelos más complejos, como las redes neuronales profundas, los algoritmos probabilísticos suelen ser **más ligeros, rápidos de entrenar y eficientes en memoria**, lo que los hace ideales para tareas en las que se necesita **velocidad y simplicidad**.

Además, en muchos de estos algoritmos el proceso de entrenamiento es **muy sencillo o incluso inexistente** en el sentido tradicional: el modelo no aprende mediante iteraciones costosas, sino que **calcula directamente las probabilidades** a partir de los datos, basándose en las distribuciones observadas.

5. Teorema de Bayes y Naive Bayes.

El **Teorema de Bayes** permite calcular la probabilidad condicional de que ocurra un evento basándose en **información o conocimiento previo** sobre otros eventos relacionados.

En términos simples, este teorema nos ayuda a **actualizar nuestras creencias** acerca de la probabilidad de que algo ocurra, teniendo en cuenta evidencias o condiciones conocidas.

Por ejemplo, podemos querer conocer la probabilidad de que ocurra un evento **A**, dado que sabemos que ha ocurrido otro evento **B** que guarda alguna relación con él.

a. Aplicación del Teorema de Bayes en Machine Learning

En el ámbito del **aprendizaje automático**, el Teorema de Bayes se utiliza principalmente para **resolver problemas de clasificación**. En estos casos, nos permite determinar la **probabilidad condicional de que una observación (un ejemplo) pertenezca a una clase determinada**, dadas sus características o atributos.

Por ejemplo, en el caso del **filtro de correo electrónico**, el teorema permite estimar la probabilidad de que un mensaje sea **spam** o **legítimo** en función de las características del propio correo, tales como:

- El número de signos de interrogación o exclamación,
- La frecuencia de ciertas palabras clave (“gratis”, “promoción”, “urgente”, etc.),
- La cantidad de etiquetas HTML o enlaces incluidos.

De esta manera, el modelo calcula la probabilidad de que un correo con determinadas características **pertenezca a cada clase** y lo clasifica en función de la mayor probabilidad.

Limitaciones del Teorema de Bayes

En teoría, para aplicar correctamente el Teorema de Bayes sería necesario **calcular las relaciones de probabilidad entre todas las variables o características del conjunto de datos**.

Sin embargo, este proceso resulta **computacionalmente muy costoso**, ya que implica un número enorme de cálculos a medida que aumenta la cantidad de atributos.

Vamos a ver la fórmula de Bayes y con un ejemplo práctico:

Fórmula de Bayes

$$P(c \setminus x) = \frac{P(x \setminus c)P(c)}{P(x)}$$

Partimos de que el teorema de Bayes se basa en **aprendizaje supervisado**, con lo cual van a requerir un conjunto de datos etiquetados.

Vamos a explicar cada uno de sus términos, viendo que significan y como se calculan, lo vamos a ver a través del ejemplo de un filtro de spam de manera que cuando nos llegue un correo electrónico nosotros hagamos una extracción de determinadas características y en función de esas características nuestro algoritmo nos proporcione si ese correo electrónico es spam o es legítimo.

- **Probabilidad Posterior $P(c \setminus x)$** = (Resultado de la fórmula matemática). Probabilidad de que un ejemplo pertenezca a una clase determinada c dadas sus características (x_1, x_2, \dots, x_n)

Ejemplo: Se va a recibir un conjunto de datos etiquetados (aprendizaje supervisado). En concreto se va a basar en dos características de entrada concretamente: x_1 que será el número de signos de interrogación y exclamación y x_2 que será el número de tags HTML e Y va a ser la variable de salida (en este caso si es spam 1 y si es legítimo 0).

x_1 # ? !	x_2 # HTML	c Y
1	0	0
5	7	1
0	0	0
...

m =longitud de datos de entrenamiento, en este caso $m=3$.

$P(c \setminus x) \Rightarrow$ probabilidad condicional de la clase C en base a esas características x_1, x_2 . Por lo que nosotros podemos representar esto como:

$$P(c \setminus x_1, x_2)$$

Si yo recibiera un correo electrónico nuevo lo que yo voy a hacer es extraer de ese correo electrónico sus variables de entrada, es decir interrogaciones y exclamaciones para la primera característica y los tags HTML para la segunda característica. Imaginaos que la primera característica es igual 1 y la segunda es igual a 0.

Para ver si es spam(clase 1) para esas entradas:

$$P(1 \setminus 1, 0)$$

Veremos como entra en juego con la siguiente parte de la fórmula matemática:

- **Probabilidad de clase $P(c)$:** prevalencia de la clase c en el conjunto de datos. Es decir, que el conjunto de datos x_1, x_2 pertenezca a una clase determinada, en este caso a la 1 (spam).

Ejemplo:

x_1 # ? !	x_2 # HTML	c Y
2	2	1
0	0	0
1	0	0
...

Donde tengo 100 ejemplos de mi conjunto de datos de entrenamiento $\Rightarrow m=100$, de los cuales 40 pertenecen a la clase 1 (spam) y 60 que pertenecen a la clase 0 (legítimo). Bien, ¿qué quiere decir esta probabilidad de la clase? Pues para el ejemplo de antes de un nuevo ejemplo de correo electrónico que nos llegase, ¿Cuál era la probabilidad de que perteneciese a la clase 1 en base a esas características x_1 y x_2 que extraían de ese ejemplo que nosotros queríamos predecir?

Básicamente lo que vamos a calcular en este término que estamos viendo es esa probabilidad de la clase 1, en este caso sería 40/100. Para la clase 0, en este caso sería 60/100.

- **Probabilidad de predicción $P(x)$:** se refiere a la prevalencia de una característica x con un valor predeterminado en el conjunto de datos. (valor que tenga el conjunto de datos x_1, x_2, \dots, x_n)

Ejemplo

x_1 # ? i	x_2 # HTML	c Y
1	2	0
0	0	0
5	6	1

$$m=3$$

Cuando me llega un ejemplo nuevo de correo para clasificar como spam o legítimo, al extraer sus características de entrada obtengo por ejemplo 1 y 6.

EL termino sería : $P(1|1,6)$,es decir, la probabilidad conjunta de x_1 y x_2 , la probabilidad de x_1 por la probabilidad de x_2 :

Probabilidad de que x_1 sea 1 es: $1/3$

Probabilidad de que x_2 sea 6 es: $1/3$

Probabilidad(x_1, x_2) = $P(x_1) * P(x_2) = 1/3 * 1/3$

- **Probabilidad $P(x \setminus c)$:** probabilidad de encontrar una característica x con un valor determinado de la clase c .

Este término es más complejo y hace que aumente el poder computacional que se requiere para calcular el resultado de esta función matemática. Vamos a usar el ejemplo del correo electrónico salvo que en este caso voy a extraer 3 características de entrada de mi conjunto de datos x_1, x_2, x_3 .

x_1 # ? j	x_2 # HTML	x_3 # ::	c Y
1	2	0	0
0	0	0	0
5	6	1	1

Se puede definir este término como la probabilidad condicional de x respecto a la clase determinada, en este ejemplo como respecto a la clase determinada, en este caso x_1, x_2, x_3 a una clase determinada como:

$$P(1 \setminus x_1, x_2, x_3)$$

$$P(x \setminus c) = P(x_1, x_2, x_3 \setminus c) \Rightarrow \text{descomponiéndolo en:}$$

$$P(x_1 \setminus c) * P(x_2 \setminus c, x_1) * P(x_3 \setminus c, x_1, x_2)$$

La probabilidad de encontrar ese x_1 dentro de los datos de entrenamiento y además habiendo encontrado ese para valor para a clase en este caso 1

La probabilidad de encontrar ese valor x_2 habiendo encontrado previamente ese valor para la clase y además con relación a esa variable x_1 previa, es decir, la probabilidad conjunta de la probabilidad condicional de encontrar ese x_2 para esa clase c en particular y de x_1 .

La probabilidad de encontrar ese valor x_3 para ese valor de c de la clase en particular en este caso 1 para ese ejemplo nuevo con relación a x_1 y x_2 . Y esto muestra que el teorema de Bayes no considera independencia entre las características de nuestro conjunto de datos. Como no considera que sean independiente, es decir, considera relaciones de dependencia entre ellas este término de la formula se comienza a descomponer en una multiplicación de diferentes probabilidades condicionales y probabilidades conjuntas=>

$$P(x_1 \setminus c) * P(x_2 \setminus c, x_1) * P(x_3 \setminus c, x_1, x_2) =$$

$$P(x_1 \setminus c) * P(x_2 \setminus c) * P(x_1) * P(x_3 \setminus c) * P(x_1) * P(x_2)$$

Aquí es donde el teorema de Bayes con esas relaciones de dependencia entre nuestras características de nuestro conjunto de datos aumenta el poder computacional que requerimos para procesar cada uno de nuestros ejemplos, por lo que esa probabilidad resultante para un ejemplo nuevo aumentara muchísimo como consecuencia de calcular todas las relaciones de dependencia entre las características ya que normalmente no tenemos solo 3 sino que suelen ser 100 o más.

b. Surgimiento del algoritmo Naive Bayes

Para superar esta limitación, se desarrolló una versión simplificada del teorema: el **algoritmo Naive Bayes**. Este algoritmo parte del mismo principio probabilístico, pero introduce una **suposición de independencia condicional** entre las características del conjunto de datos. Es decir, asume que las variables o atributos son independientes entre sí, lo cual rara vez es cierto en la práctica, pero **reduce enormemente la complejidad de los cálculos** y hace posible aplicar el modelo de forma rápida y eficiente.

Características y aplicaciones de Naive Bayes

- Es un **algoritmo de aprendizaje supervisado**, ya que necesita un conjunto de **datos etiquetados** para entrenarse (con las clases o salidas conocidas).
- Calcula las **probabilidades de pertenencia** a cada clase y elige aquella con la mayor probabilidad.
- Es **rápido, escalable y preciso** en muchos escenarios, especialmente cuando se dispone de grandes volúmenes de datos o cuando se necesita una clasificación inmediata.

Ejemplos de aplicación:

- Clasificación de correos electrónicos (*spam* o *no spam*).
- Análisis de sentimientos (positivo, negativo o neutral).
- Clasificación de textos o noticias por temática.

Por lo tanto, el algoritmo de **Naive Bayes** simplifica Bayes y reduce el coste computacional aplicando para ello el concepto de independencia condicional en el cálculo del término de Probabilidad de la función del teorema de Bayes. Cada una de las características de entrada de nuestro conjunto de datos serán características independientes. De manera formal se traduce en estos términos:

$P(x_1, x_2, x_3, \dots, x_n \mid c) = P(x_i \mid c, x_j) = P(x_i \mid c)$, es decir, va a dejar solo ese término y va a olvidarse de esa relación de dependencia que existen con otras características en este caso x_j y va a eliminarlo: $P(x_2 \mid c, x_1) = P(x_2 \mid c)$ y eliminamos $P(x_1)$.

Por lo que para el caso del ejemplo de las tres características de entrada quedaría:

$$P(x_1, x_2, x_3 \mid c) = \frac{P(x_1 \mid c) * P(x_2 \mid c) * P(x_3 \mid c) * P(c)}{P(x_1) * P(x_2) * P(x_3)}$$

Ejemplo práctico completo : Nuevo correo electrónico con entrada 1 y 5.

$(x_1, x_2) = (1, 5)$

x_1 # ? i	x_2 # HTML	c Y
2	5	1
2	1	0
2	6	1
4	5	1
1	0	0

$m=5$

$(x_1, x_2) = (1, 5)$

$$P(1|1,5) = \frac{P(x_1|c) * P(x_2|c) * P(c)}{P(x_1) * P(x_2)} = \frac{0/3 * 2/3 * 3/5}{1/5 * 2/5}$$

- Para la clase 1 (es decir, el y como valor a 1), ¿cuántos valores tengo de que x_1 valga 1? Pues cero. Para la clase 1, ¿cuántos valores tengo de que x_2 valga 5? Tengo dos. Entre las 3 veces que sale el valor uno en nuestra clase.
- La probabilidad de la clase es 3 de 5 datos que tengo en el conjunto de datos.
- Probabilidad de x_1 es 1 de 5 y a probabilidad de x_2 es 2 de 5 datos.

$$= \frac{0 * 2/3 * 3/5}{1/5 * 2/5} = 0$$

Esto dará 0 (por el numerador) y para que no ocurra se aplica lo que se llama corrección de Laplace, que normalmente cuando el termino se hace cero en el numerador lo sustituimos por 1.

No voy a contar con el denominador porque si quiero calcular el mismo ejemplo para estos mismos valores pero para la clase 0 el denominador no va a variar.

$$= \frac{1 * 2/3 * 3/5}{1/5 * 2/5} = 0.41$$

$$P(0|1,5) = \frac{1/2 * 0 * 2/5}{1/5 * 2/5} =$$

$$= \frac{1/2 * 1 * 2/5}{1/5 * 2/5} = 0.2$$

Por lo que tiene más probabilidad de tener clase 1, ser spam, que clase 0 o legítimo. Por lo que se categorizaría como spam.

c. Conclusión.

El **Teorema de Bayes** proporciona la base teórica para calcular probabilidades condicionales y actualizar creencias con nueva información.

El **algoritmo Naive Bayes** simplifica su aplicación al suponer independencia entre las variables, lo que lo convierte en uno de los métodos **más utilizados en problemas de clasificación supervisada** dentro del *Machine Learning*.

6. KNN ((K-Nearest Neighbors)

El **método de los K vecinos más cercanos (K-Nearest Neighbors o K-NN)** es un **algoritmo de aprendizaje supervisado** que puede emplearse tanto para **clasificación** como para **regresión**. Aunque es un método sencillo, resulta **muy eficaz en numerosos casos prácticos**.

El principio fundamental del K-NN es que **los datos similares suelen encontrarse próximos entre sí** en el espacio de características. Por tanto, para clasificar o predecir un nuevo dato, el algoritmo busca los **K puntos más cercanos** del conjunto de entrenamiento y **realiza la predicción en función de ellos**.

- En **clasificación**, el nuevo dato se asigna a la **clase más frecuente** entre sus K vecinos más cercanos.
- En **regresión**, la predicción es el **promedio de los valores** de esos vecinos.

Funcionamiento

Imaginemos un conjunto de datos con animales (gatos y perros) representados por características físicas. Para clasificar un nuevo punto (un animal desconocido), el proceso sería:

1. **Elegir el valor de K:**
 - Es un **hiperparámetro** que indica cuántos vecinos se considerarán.
 - Valores típicos: 3, 5 o 7.
2. **Medir las distancias:**
 - Se calcula la distancia entre el nuevo punto y los datos existentes mediante una métrica, como la **distancia euclidiana** o la **distancia Manhattan**.
3. **Identificar los K vecinos más cercanos:**
 - Se seleccionan los K puntos del conjunto de entrenamiento más próximos al nuevo dato.

4. Realizar la predicción:

- **Clasificación:** se asigna la clase que más se repite entre los vecinos.
- **Regresión:** se calcula la media de los valores de los vecinos.

Ventajas del K-NN

- **Simplicidad:** fácil de entender e implementar.
- **No requiere entrenamiento explícito:** el modelo simplemente almacena los datos de entrenamiento.
- **Versatilidad:** puede aplicarse tanto a problemas de **clasificación** como de **regresión**.

Desventajas del K-NN

- **Baja eficiencia en grandes conjuntos de datos:** calcular las distancias a todos los puntos puede ser costoso en tiempo.
- **Sensibilidad al ruido:** los datos erróneos o mal etiquetados pueden afectar el resultado.
- **Dependencia de parámetros:** la elección de **K** y de la **métrica de distancia** es fundamental para obtener buenos resultados.

Usos comunes del K-NN

- **Detección de fraudes:** compara transacciones con otras previamente clasificadas como fraudulentas.
- **Reconocimiento de patrones:** por ejemplo, para identificar dígitos escritos a mano.
- **Sistemas de recomendación:** sugiere productos similares a los que han comprado otros usuarios.

Consideración final

Elegir el valor adecuado de **K** es **crítico**:

- Si **K** es demasiado pequeño, el modelo puede ser **sensible al ruido**.
- Si **K** es demasiado grande, puede **perder precisión** al incluir vecinos de clases diferentes.

Ejemplo:

Supón que deseas predecir si un nuevo animal es un gato o un perro. Si $K = 3$ y los tres vecinos más cercanos son: dos gatos y un perro, entonces el nuevo punto se **clasifica como gato** (la clase mayoritaria).

Vamos a desarrollar el ejemplo:

Primero vamos a ver que es la **distancia euclidiana**. Es simplemente la **distancia “normal” o recta** entre dos puntos en un espacio. Es la misma que usaríamos para medir con una regla la distancia entre dos puntos en un plano.

En el caso del K-NN, cada dato (animal, persona, etc.) se representa como un **punto** en un “espacio” formado por sus características.

Imagina que tienes dos características:

- Peso (eje X)
- Altura (eje Y)

Entonces cada animal se convierte en un punto en un **plano 2D**, como si fueran coordenadas en un gráfico:

- El **animal A (4 kg, 25 cm)** estaría en el punto (4, 25).
- El **nuevo animal X (5,5 kg, 28 cm)** estaría en el punto (5.5, 28).

La **distancia euclidiana** entre A y X se calcula igual que en geometría entre dos puntos (x_1, y_1) y (x_2, y_2) :

$$\text{Distancia} = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$$

Imaginemos que estas son las características de los animales que tenemos:

- **Peso (kg)**
- **Altura (cm)**

Animal	Peso (kg)	Altura (cm)	Clase
A	4	25	Gato
B	5	27	Gato
C	6	30	Perro
D	8	35	Perro

Ahora aparece un **nuevo animal (X)** con las siguientes características:

- Peso: **5,5 kg**
- Altura: **28 cm**

Queremos saber si **X es gato o perro** usando **K = 3** (es decir, miraremos los tres vecinos más cercanos).

Calcular las distancias

Usamos la **distancia euclidiana** entre X y cada animal del conjunto de entrenamiento:

Animal	Cálculo aproximado	Distancia
A	$\sqrt{((5.5-4)^2 + (28-25)^2)} = \sqrt{(2.25 + 9)}$	3.35
B	$\sqrt{((5.5-5)^2 + (28-27)^2)} = \sqrt{(0.25 + 1)}$	1.12
C	$\sqrt{((5.5-6)^2 + (28-30)^2)} = \sqrt{(0.25 + 4)}$	2.06
D	$\sqrt{((5.5-8)^2 + (28-35)^2)} = \sqrt{(6.25 + 49)}$	7.29

Elegir los 3 vecinos más cercanos

Los **3 más cercanos** son:

1. **B (Gato)** – Distancia 1.12
2. **C (Perro)** – Distancia 2.06
3. **A (Gato)** – Distancia 3.35

Entre los tres vecinos:

- 2 son **Gatos**
- 1 es **Perro**

Resultado: El nuevo animal **X se clasifica como Gato**