



Universidad Tecnológica de Panamá
Facultad de Ingeniería Eléctrica
Maestría en Ingeniería Eléctrica

Semana 1

Introducción a Machine Learning

Entrando en materia

Clasificación vs Regresión

Regresión

Predice valores continuos, es decir, variables numéricas que pueden tomar cualquier valor dentro de un rango.

- Ejemplo: Predecir el precio de una casa, la temperatura de una ciudad, el ingreso anual de una persona.

Clasificación

Asigna categorías o etiquetas a los datos, es decir, predice clases discretas.

- Ejemplo: Clasificar correos como spam o no spam, diagnosticar si un paciente tiene una enfermedad (sí/no), identificar el tipo de flor según sus características.

Diferencias clave

- Regresión: salida numérica continua.
- Clasificación: salida categórica/discreta.



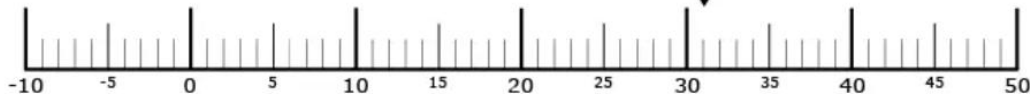
Regression

What temperature will there be tomorrow?

PREDICTION

31°

Celsius
°C



Classification

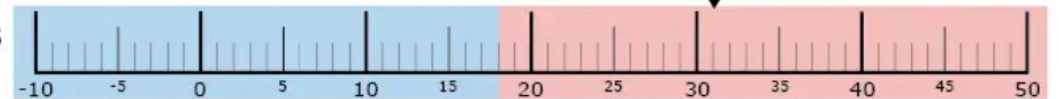
Will tomorrow be a cold or hot day?

Representación gráfica de la dif...

PREDICTION

Hot

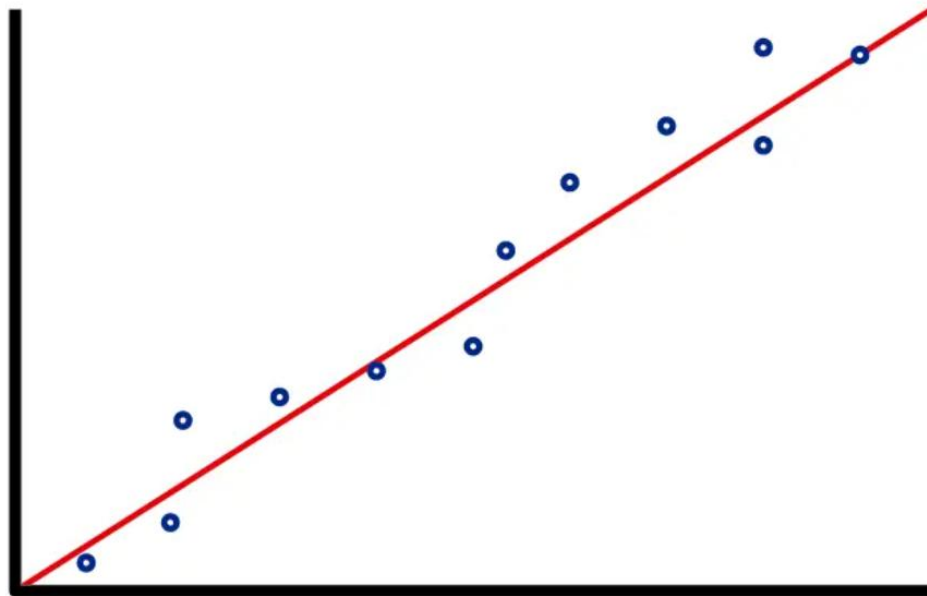
Celsius
°C



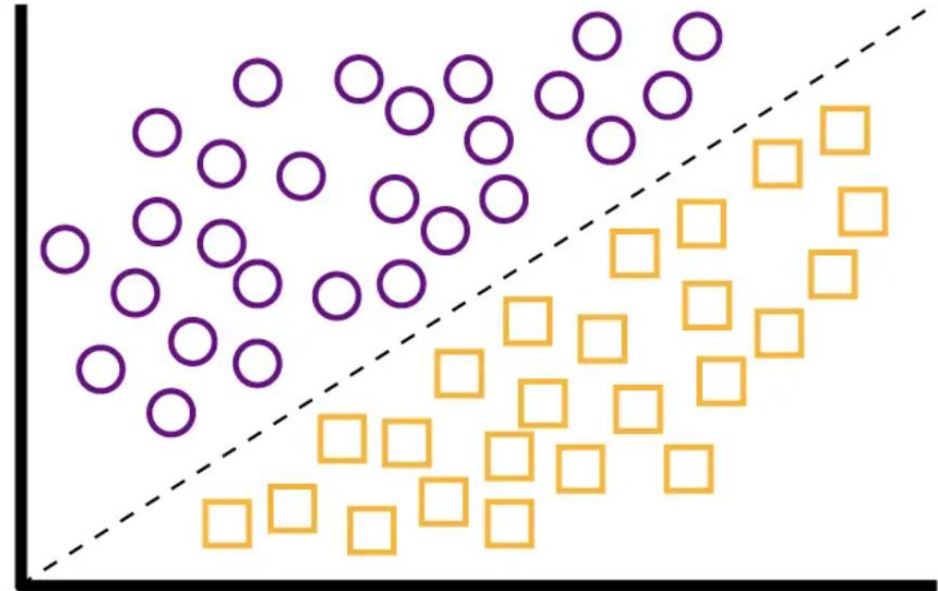
Diferencias clave

En regresión, los datos suelen representarse en gráficos de dispersión donde la variable objetivo es continua.

En clasificación, los datos pueden visualizarse en gráficos donde los puntos se agrupan por clases o colores.



Regression



Classification

Casos de uso

- **Regresión:** predicción de ventas, temperatura, demanda energética, precios de acciones.
- **Clasificación:** diagnóstico de enfermedades, reconocimiento de imágenes, detección de fraudes, análisis de sentimientos.

Tipos de aprendizaje

- ✓ Supervisado
- ✓ No supervisado
- ✓ Por reforzamiento

Aprendizaje supervisado



Utiliza datos etiquetados (con respuestas conocidas), es decir, cada ejemplo del conjunto de datos tiene una entrada y una salida esperada.



El objetivo es que el modelo aprenda la relación entre las entradas y las salidas para poder predecir la salida de nuevos datos.

Ejemplo de datos etiquetados: El MNIST dataset



Ejemplo de datos etiquetados: El MNIST dataset

- El MNIST (Modified National Institute of Standards and Technology) es un conjunto de datos clásico en machine learning que contiene 70,000 imágenes en escala de grises de dígitos manuscritos (0 al 9), cada una de 28x28 pixels (784 features).
- Está dividido en 60,000 muestras para entrenamiento y 10,000 para prueba.
- Las imágenes fueron recopiladas y normalizadas a partir de formularios escritos por estudiantes y empleados de la oficina de censos de EE.UU., y cada imagen está asociada a una etiqueta que indica el dígito correspondiente.

Dos conceptos importantes

Etiquetas

También llamadas "labels", son los valores o categorías que queremos predecir; en el caso de MNIST, el dígito manuscrito (0-9) asociado a cada imagen.

Features


También llamadas "características", son los atributos o variables de entrada que describen cada ejemplo; en MNIST, los valores de los píxeles de cada imagen.

 **Etiquetas**

Features

The image displays a Jupyter Notebook interface with a large grid of handwritten digits from the MNIST dataset. The grid is organized into 28 rows and 1000 columns. The first 10 columns contain the original digits, while the remaining 990 columns show the digits after a 28x28 pixel crop. The digits are primarily '0's and '1's, with some '4's and '9's visible. The grid is titled 'mnist_train'.

Datos etiquetados del MNIST en formato CSV



Ventajas del aprendizaje supervisado

- Suele ofrecer mayor precisión cuando se dispone de datos etiquetados de calidad, ya que el modelo aprende directamente de ejemplos reales y puede ser evaluado objetivamente.
- Permite medir el desempeño del modelo con métricas claras y ajustar los algoritmos para mejorar la exactitud.
- Es ideal para tareas donde se requiere una predicción específica y se cuenta con información histórica confiable.



Desventajas del aprendizaje supervisado

- Requiere un gran volumen de datos etiquetados, lo cual puede ser costoso o difícil de obtener, especialmente en dominios donde el etiquetado debe ser realizado por expertos o es un proceso manual.
- Si los datos de entrenamiento no son representativos o contienen sesgos, el modelo puede generalizar mal o perpetuar errores.
- El mantenimiento y actualización de los datos etiquetados también puede ser un reto a largo plazo.



Aprendizaje no supervisado

01

Trabaja con datos sin etiquetas, es decir, solo se dispone de las entradas y no de las salidas esperadas.

02

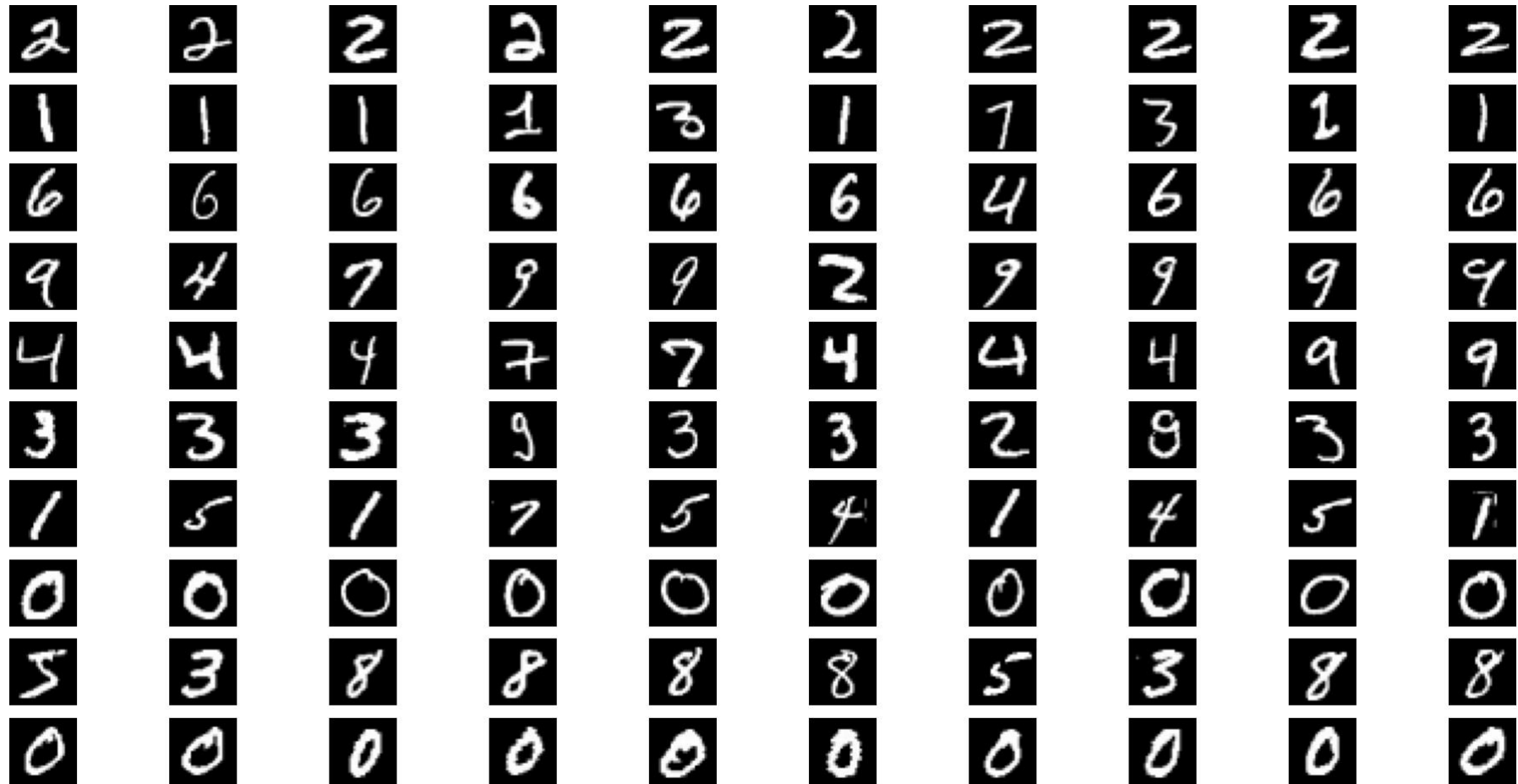
El objetivo es encontrar patrones, estructuras o relaciones ocultas en los datos.

Aprendizaje no supervisado

Técnicas como agrupamiento (clustering), reducción de dimensionalidad y detección de anomalías.

Ejemplos: Agrupar clientes según su comportamiento de compra, segmentar imágenes, identificar temas en textos, etc.

Clustering on MNIST (k-means)



Algoritmos comunes

- ✓ K-means
- ✓ Clustering jerárquico
- ✓ Análisis de componentes principales (PCA)
- ✓ Autoencoders

Ventajas del aprendizaje no supervisado

Útil cuando no se dispone de datos etiquetados y para explorar la estructura de los datos.

Permite descubrir patrones ocultos, segmentar datos de manera automática y encontrar relaciones inesperadas entre variables.

Es especialmente valioso en etapas iniciales de análisis, donde se busca entender la organización interna de los datos o identificar grupos naturales.

Puede reducir costos y tiempo al no requerir el proceso manual de etiquetado, y es aplicable a grandes volúmenes de datos no estructurados.

Desventajas del aprendizaje no supervisado

Los resultados pueden ser más difíciles de interpretar y validar, y no siempre hay una "respuesta correcta".

La evaluación de la calidad de los agrupamientos o patrones encontrados suele ser subjetiva y depende del contexto.

Existe el riesgo de identificar agrupaciones artificiales o irrelevantes si los parámetros no se eligen adecuadamente.

La falta de etiquetas dificulta la comparación objetiva entre diferentes modelos o configuraciones, y puede ser complicado traducir los hallazgos en acciones concretas.

Aprendizaje por reforzamiento

Reinforcement Learning

Reinforcement Learning



Es un tipo de aprendizaje automático donde un agente aprende a tomar decisiones mediante prueba y error, recibiendo recompensas o penalizaciones según sus acciones.



El objetivo del agente es maximizar la recompensa acumulada a lo largo del tiempo.

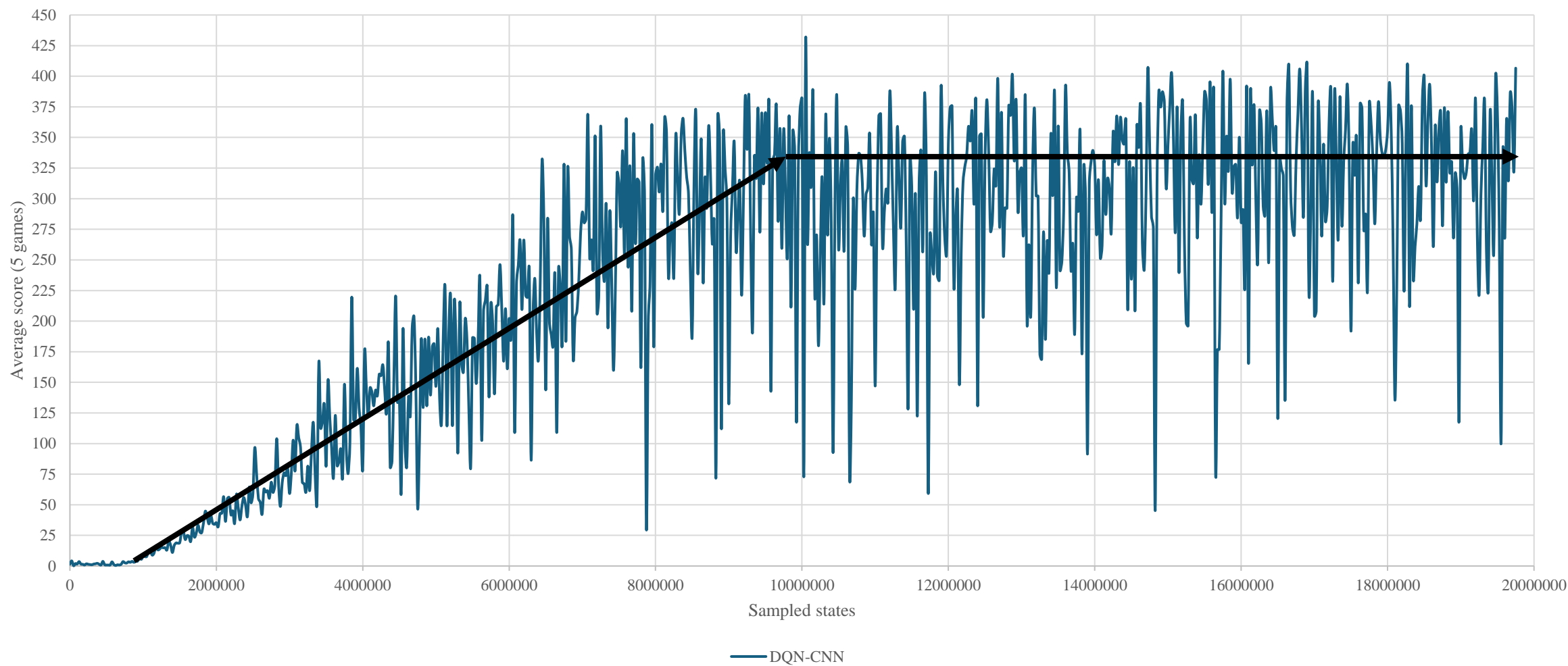


Se utiliza en problemas donde las decisiones afectan el entorno y las recompensas pueden ser diferidas, como en juegos, robótica o control automático.

Ejemplo de un
agente de
Reinforcement
learning



Average reward over samples states



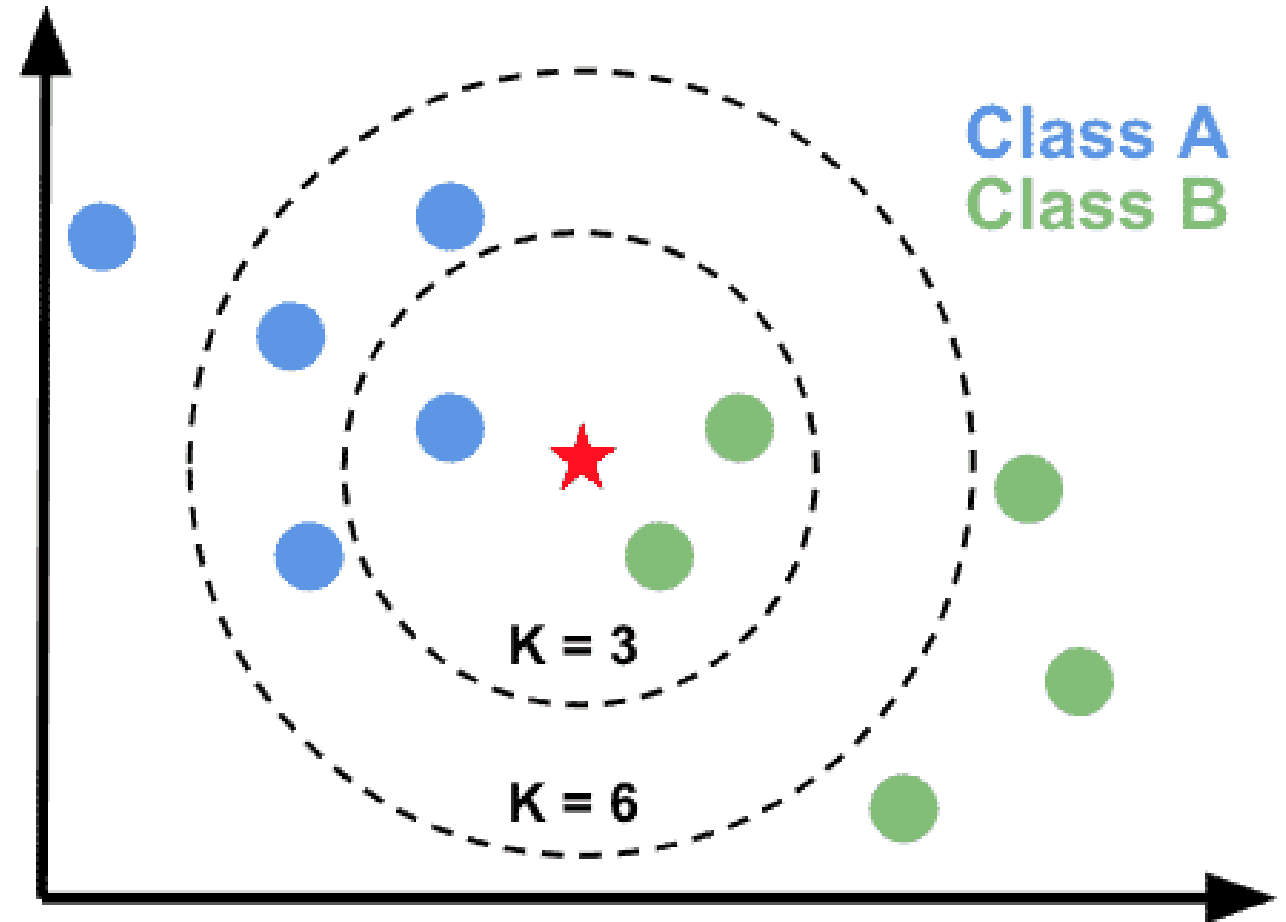
Algoritmos de Machine Learning para aprendizaje supervisado

K-Nearest Neighbors (KNN)

KNN (K-Nearest Neighbors)

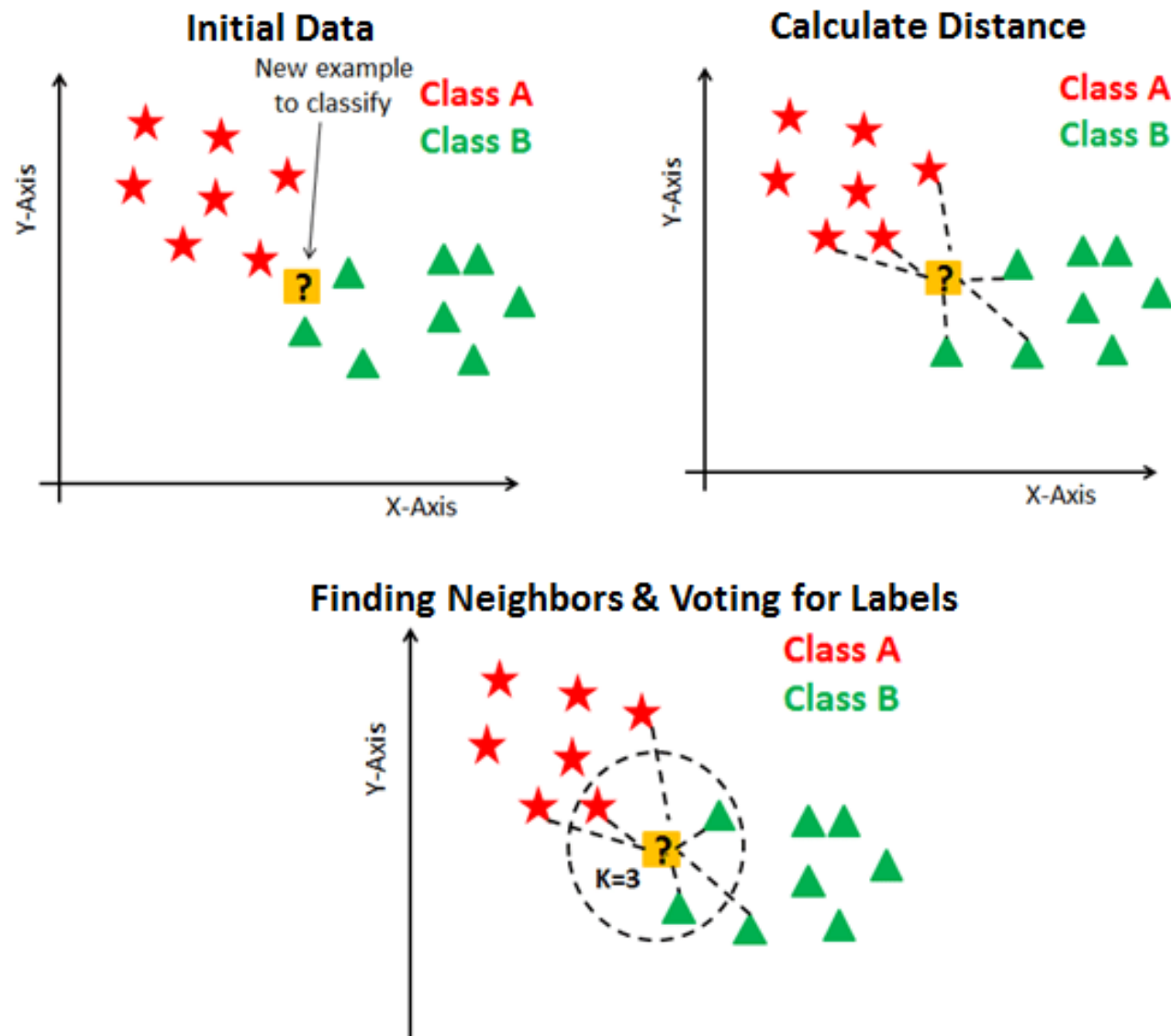
Es un algoritmo de aprendizaje supervisado utilizado tanto para clasificación como para regresión.

Su principio básico es que, para predecir la clase o valor de un nuevo dato, el algoritmo busca los "k" vecinos más cercanos en el conjunto de entrenamiento y toma una decisión basada en ellos.



KNN (K-Nearest Neighbors)

- En clasificación, la clase más común entre los vecinos determina la predicción.
- En regresión, se promedia el valor de los vecinos.



KNN (K-Nearest Neighbors)

Es un algoritmo no paramétrico y perezoso: no realiza un entrenamiento explícito, sino que almacena los datos y realiza los cálculos en el momento de la predicción.

La elección de la métrica de distancia y el valor de "k" son fundamentales para su desempeño.

Hyperparámetros

¿Qué son?

Son configuraciones externas al modelo que no se aprenden automáticamente durante el entrenamiento, sino que deben ser definidos antes de entrenar el modelo.

En KNN, los hiperparámetros controlan aspectos clave como el número de vecinos a considerar, la forma en que se calcula la distancia, el método de búsqueda de vecinos y cómo se ponderan las contribuciones de cada vecino.

Elegir los valores adecuados para estos hiperparámetros es fundamental para obtener un buen desempeño del model.

Hyperparámetros

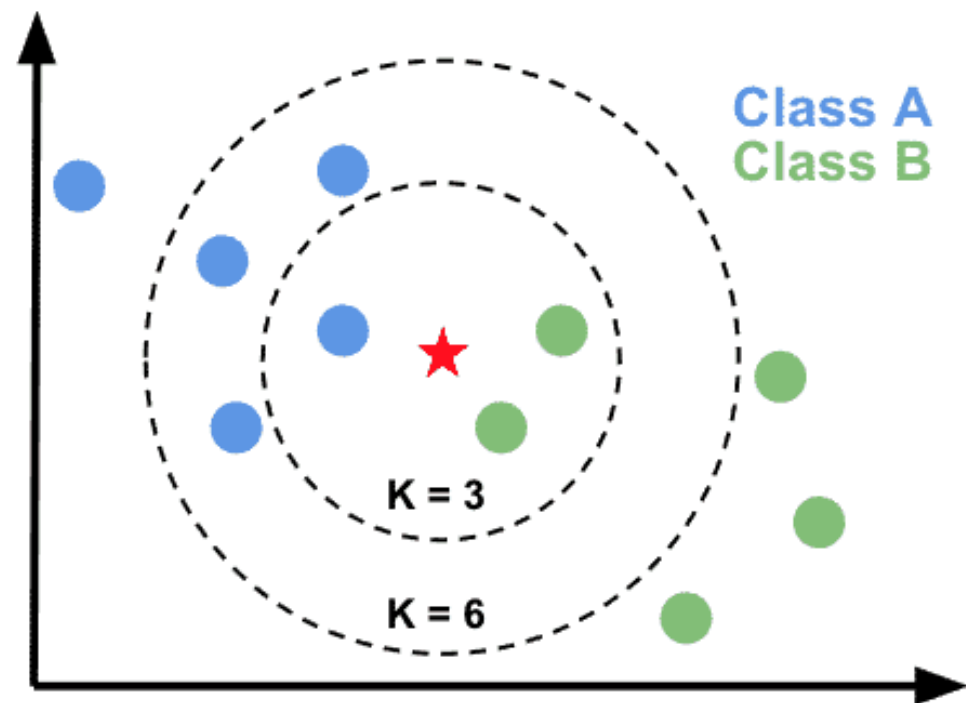
¿Cuáles son?

- ``n_neighbors``: Número de vecinos más cercanos que el algoritmo considera para hacer la predicción.
 - ``weights``: Cómo se pondera la contribución de cada vecino en la predicción
 - ``algorithm``: Método utilizado para buscar los vecinos más cercanos
 - ``metric``: Métrica utilizada para calcular la distancia entre los puntos
-

n_neighbors

Número de vecinos a considerar para la predicción.

Un valor bajo puede hacer el modelo sensible al ruido, mientras que un valor alto puede suavizar demasiado las fronteras de decisión.



weights

Función de pesos para los vecinos. Estos pueden ser:

- **uniform:** Todos los vecinos contribuyen por igual.
- **distance:** Los vecinos más cercanos tienen mayor peso en la predicción.
- También se puede usar una función definida por el usuario para personalizar los pesos.

algorithm

Algoritmo usado para buscar los vecinos más cercanos.

- **auto:** Selecciona automáticamente el algoritmo más adecuado según los datos y la métrica.
- **ball_tree:** Utiliza el algoritmo BallTree, eficiente para conjuntos de datos grandes y de alta dimensión. Divide el espacio en hiperesferas y permite búsquedas rápidas de vecinos.
- **kd_tree:** Utiliza el algoritmo KDTree, eficiente para datos de baja a moderada dimensión. Divide el espacio en hiperplanos y es rápido para dimensiones bajas.
- **brute:** Realiza una búsqueda por fuerza bruta, calculando todas las distancias posibles. Es útil para conjuntos de datos pequeños o cuando se usan métricas personalizadas.

metric

Métrica utilizada para calcular la distancia entre puntos (por defecto `'minkowski'`).

Se pueden utilizar métricas como:

- Euclidean distance
- Manhattan distance
- Chebysev
- Minkowski

Modelos matemáticos de distancia utilizados en el algoritmo KNN

- Euclidean distance

$$d(x, y) = \sqrt{\sum_{i=1}^n (x_i - y_i)^2}$$

- Manhattan distance

$$d(x, y) = \sum_{i=1}^n |x_i - y_i|$$

- Chebysev

$$d(x, y) = \max_i |x_i - y_i|$$

- Minkowski

$$d(x, y) = \left(\sum_{i=1}^n |x_i - y_i|^p \right)^{1/p}$$

Ejemplo

Suponiendo que tenemos dos muestras:

$$\text{Muestra A: } (x_1, y_1) = (2, 4)$$

$$\text{Muestra B: } (x_2, y_2) = (5, 1)$$

Ejemplo

Distancia Euclidiana

Modelo matemático:

$$d(x, y) = \sqrt{\sum_{i=1}^n (x_i - y_i)^2}$$

Aplicación:

$$d = \sqrt{(2 - 5)^2 + (4 - 1)^2} = \sqrt{(-3)^2 + (3)^2} = \sqrt{9 + 9} = \sqrt{18} \approx 4.24$$

Ejemplo

Distancia Manhattan

Modelo matemático:

$$d(x, y) = \sum_{i=1}^n |x_i - y_i|$$

Aplicación:

$$d = |2 - 5| + |4 - 1| = 3 + 3 = 6$$

Ejemplo

Distancia Chebyshev

Modelo matemático:

$$d(x, y) = \max_i |x_i - y_i|$$

Aplicación:

$$d = \max(|2 - 5|, |4 - 1|) = \max(3, 3) = 3$$

Ejemplo

Distancia de Minkowski (con $p = 3$)

Modelo matemático:

$$d(x, y) = \left(\sum_{i=1}^n |x_i - y_i|^p \right)^{1/p}$$

Aplicación:

$$d = (|2 - 5|^3 + |4 - 1|^3)^{1/3} = (27 + 27)^{1/3} = 54^{1/3} \approx 3.78$$

Ejemplo en
Python



[Notebook de ejemplo en Github](#)

Ventajas de KNN

Sencillo de entender e implementar.

No requiere entrenamiento explícito.

Puede adaptarse a problemas de clasificación y regresión.

Flexible en la elección de la métrica de distancia.

Desventajas de KNN

Computacionalmente costoso para grandes volúmenes de datos, ya que requiere calcular la distancia a todos los puntos de entrenamiento en cada predicción.

Sensible a la escala de las variables, por lo que es recomendable normalizar o estandarizar los datos.

Puede verse afectado por datos irrelevantes o ruido.

**Ejemplo
(Pima
Diabetes
Dataset)**

Variable	Muestra 1	Muestra 2
Pregnancies	1	5
Glucose	189	166
Blood Pressure	60	72
Skin Thickness	23	19
Insulin	846	175
BMI	30.1	25.8
Diabetes Pedigree	0.398	0.587
Age	59	51

Ejemplo

Distancia Euclidiana

Modelo matemático:

$$d(x, y) = \sqrt{\sum_{i=1}^n (x_i - y_i)^2}$$

Aplicación paso a paso:

$$\begin{aligned} d(\text{Muestra 1}, \text{Muestra 2}) &= \sqrt{(1 - 5)^2 + (189 - 166)^2 + (60 - 72)^2 + (23 - 19)^2 +} \\ &\quad + (846 - 175)^2 + (30.1 - 25.8)^2 + (0.398 - 0.587)^2 + (59 - 51)^2} \\ &= \sqrt{16 + 529 + 144 + 16 + 44944 + 18.49 + 0.035721 + 64} \\ &= \sqrt{45741.525721} \approx 213.98 \end{aligned}$$