RIESGOS FINANCIEROS

Notas Técnicas:

Machine Learning en la Gestión de Riesgos Financieros

Octubre 22, 2026



BANORTE

RIESGOS FINANCIEROS NOTAS TÉCNICAS

Machine Learning en la Gestión de Riesgos Financieros

CONTENIDO

OBJETIVO GENERAL 2
INTRODUCCIÓN 2
CONTEXTO REGULATORIO 2
APRENDIZAJE AUTOMÁTICO 3
1. APRENDIZAJE SUPERVISADO 5
1.1 Clasificador Bayesiano 8
1.2 K vecinos más cercanos 8
1.3 Árbol de decisión 8
1.4 Bosque aleatorio 9
1.5 Máquina de vector de soporte 9
1.6 Potenciación por gradiente 10
2. MODELOS ESTADÍSTICOS 11
2.1 Modelos de Regresión 11
2.2 Series temporales 13
3. REDES NEURONALES 14
3.1 Teorema de aproximación universal 15
3.2 Entrenamiento de una red neuronal 15
3.3 Tipos de redes neuronales 15
4. APRENDIZAJE NO SUPERVISADO 16
4.1 Agrupamiento de K medias 16
4.2 Modelo de mezcla gaussiana 17
4.3 Análisis de componentes principales 17
4.4 Modelo oculto de Markov 18
5. APRENDIZAJE POR REFUERZO 18
5.1 Aprendizaje Q 20
5.2 REINFORCE 20
6. APLICACIONES 20
6.1 Riesgo de crédito 20
6.2 Riesgo de contraparte 21
6.3 Riesgo de mercado 21
6.4 Riesgo de balance 22
7. LIMITACIONES 23
8. LÍNEAS EMERGENTES DE INVESTIGACIÓN 23
9. RETOS EN LA IMPLEMENTACIÓN EN LA BANCA MEXICANA 24
9.1 Entorno regulatorio 24
9.2 Gobernanza y validación interna 24
9.3 Infraestructura de datos 24
9.4 Talento especializado 24
10. CONCLUSIONES 24
11. REFERENCIAS 25

OBJETIVO GENERAL

El objetivo de esta nota técnica es presentar de manera estructurada los principales fundamentos, tipos de modelos y aplicaciones del aprendizaje automático en la gestión de riesgos financieros. El tratamiento se realiza de forma introductoria y no exhaustiva, con el propósito de ofrecer una visión general de las principales familias de algoritmos, su clasificación y algunos ejemplos representativos, sin desarrollar en profundidad todas sus variantes ni aspectos teóricos. Se consideran los principales paradigmas de aprendizaje automático, incluyendo aprendizaje supervisado, semisupervisado, no supervisado y por refuerzo, así como distintas familias de modelos estadísticos y redes neuronales que pueden emplearse dentro de dichos paradigmas. De esta manera, se busca proporcionar un panorama general que permita identificar qué tipos de modelos existen, cómo se clasifican y algunos contextos donde pueden resultar útiles.

INTRODUCCIÓN

La gestión de riesgos financieros depende fundamentalmente de la capacidad para identificar, medir y gestionar comportamientos adversos a partir de la información disponible. Tradicionalmente, esta tarea se ha sustentado en modelos estadísticos y econométricos que proporcionan un marco formal para describir la relación entre variables, estimar parámetros y realizar pronósticos. Sin embargo, el crecimiento en el volumen y diversidad de los datos financieros, junto con la presencia de relaciones no lineales y estructuras de alta dimensionalidad, ha impulsado la incorporación de técnicas de aprendizaje automático (*Machine Learning,* ML) como herramientas complementarias para el análisis de riesgos.

El aprendizaje automático ofrece un enfoque flexible para aprender patrones directamente a partir de los datos y utilizarlos con fines de predicción, clasificación, segmentación y toma de decisiones. Esta capacidad permite abordar problemas de distinta naturaleza dentro del riesgo y en general las finanzas, desde la estimación del riesgo de crédito y la volatilidad hasta la detección de anomalías, el análisis de información textual y la optimización de decisiones. Su utilidad resulta especialmente relevante cuando la relación entre las variables es compleja o cuando la información disponible no puede incorporarse fácilmente mediante especificaciones paramétricas convencionales.

No obstante, una mayor flexibilidad predictiva también introduce nuevos desafíos. La capacidad de un modelo para ajustarse a estructuras complejas debe acompañarse de una adecuada evaluación de su generalización, robustez e interpretabilidad, particularmente en aplicaciones donde las predicciones pueden intervenir directamente en procesos de gestión y control del riesgo. Por ello, el uso de aprendizaje automático no debe plantearse como una sustitución general de los métodos estadísticos tradicionales, sino como una ampliación del conjunto de herramientas cuantitativas disponibles, cuya conveniencia depende de las características del problema y de los datos.

CONTEXTO REGULATORIO

La adopción de modelos de aprendizaje automático en la banca no ocurre en un vacío regulatorio. Los marcos de gestión del riesgo de modelo (*model risk management*, MRM), como la guía SR 11-7 de la Reserva Federal de Estados Unidos (Board of Governors of Federal Reserve System, 2011), establecen principios de validación, documentación y monitoreo continuo cuya aplicación debe ser proporcional a la materialidad, complejidad y riesgo asociado a cada modelo. En el contexto específico de riesgo de tasa de interés en el libro bancario (*interest rate risk in the banking book,* IRRBB), el Comité de Supervisión de Basilea (*Basel Committee on Banking Supervision*, BCBS) establece que los supuestos utilizados para modelar partidas sin vencimiento contractual (como los depósitos a la vista) deben basarse en el comportamiento observado, ser prudentes y estar debidamente documentados y validados, lo cual aplica igualmente cuando se emplean técnicas de inteligencia artificial para dicha modelación (Basel Committee on Banking Supervision, 2016).

APRENDIZAJE AUTOMÁTICO

El aprendizaje automático puede entenderse como un conjunto de técnicas de optimización y ajuste de funciones que, a diferencia de la modelación clásica basada en leyes físicas o económicas conocidas a priori, aprende la relación entre variables directamente a partir de los datos, sacrificando en ocasiones interpretabilidad a cambio de flexibilidad y capacidad predictiva.

El objeto de estudio del aprendizaje automático son algoritmos capaces de aprender patrones, relaciones o reglas de decisión a partir de datos o de la experiencia. El objetivo es que el sistema pueda realizar predicciones, identificar estructuras o tomar decisiones sin que todas las reglas necesarias hayan sido programadas de forma explícita.

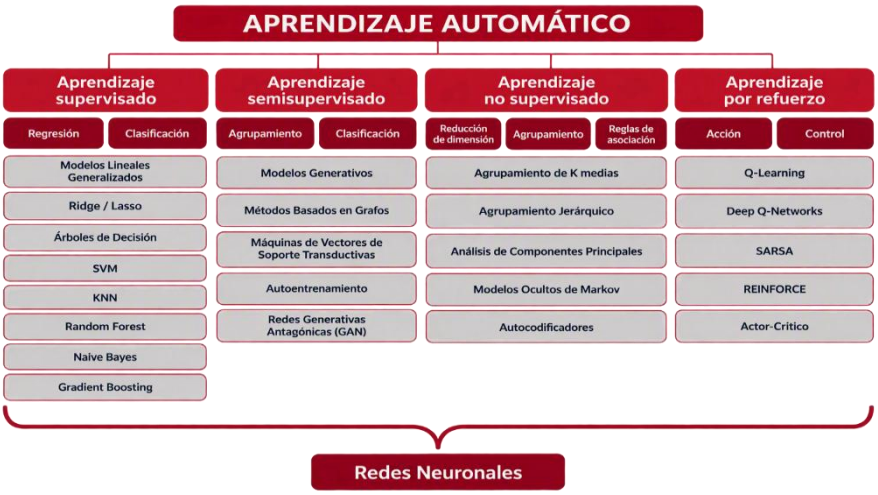


Ilustración 1. Algoritmos de aprendizaje automático.
Nota. Elaboración propia.

En la Ilustración 1 se presenta un esquema de los principales paradigmas del aprendizaje automático: aprendizaje supervisado, semisupervisado, no supervisado y por refuerzo. Para cada uno se muestran sus principales tipos de problemas, aplicaciones y algunos de los algoritmos más representativos.

De manera general, los métodos de aprendizaje automático pueden organizarse en cuatro paradigmas principales:

- Aprendizaje supervisado: el modelo se entrena a partir de datos previamente etiquetados, es decir, ejemplos que incluyen tanto la información de entrada como la respuesta esperada. Su finalidad es aprender la relación entre ambas para realizar predicciones sobre nuevos datos. Sus aplicaciones principales son la clasificación, cuando se desea asignar una categoría, y la regresión, cuando se busca predecir un valor numérico.
- Aprendizaje semisupervisado: combina una cantidad reducida de datos etiquetados con una mayor cantidad de datos sin etiquetar. Su objetivo es aprovechar la información disponible en ambos conjuntos para construir modelos más precisos cuando el proceso de etiquetado resulta costoso o limitado. Puede emplearse en tareas de clasificación, regresión y agrupamiento.
- Aprendizaje no supervisado: el modelo trabaja con datos que no poseen etiquetas ni respuestas previamente definidas. Su función principal consiste en descubrir de manera autónoma patrones, estructuras, relaciones o agrupaciones presentes en los datos. Entre sus aplicaciones más comunes se encuentran el agrupamiento, la reducción de dimensionalidad y la detección de anomalías.
- Aprendizaje por refuerzo: un agente aprende mediante la interacción continua con un entorno, tomando decisiones y recibiendo recompensas o penalizaciones según los resultados obtenidos. Su propósito es desarrollar una estrategia que permita maximizar la recompensa acumulada a lo largo del tiempo. Se utiliza principalmente en problemas de toma de decisiones, control, planificación y optimización de acciones.

Las categorías anteriores describen paradigmas de aprendizaje y no son mutuamente excluyentes de las familias de modelos presentadas posteriormente. Por ejemplo, una red neuronal puede utilizarse dentro de esquemas supervisados, no supervisados o de aprendizaje por refuerzo, mientras que diversos modelos estadísticos pueden emplearse con fines de aprendizaje supervisado.

En la Ilustración 2 se presenta de manera esquemática el ciclo de vida de un modelo de aprendizaje supervisado. Aunque el proceso puede variar según el tipo de problema y el algoritmo empleado, normalmente se compone de las siguientes etapas:

1. **Definición del problema**: consiste en establecer claramente qué se desea resolver y cuál será la salida esperada del modelo.

2. **Obtención de los datos**: es necesario recopilar los datos que podrían contener información relevante para resolverlo. La calidad, cantidad y representatividad de los datos tendrá una influencia importante sobre el desempeño del modelo.

3. **Preparación de los datos**: los datos obtenidos rara vez se encuentran completamente listos para utilizarse. Por ello, deben revisarse y prepararse. En esta etapa pueden realizarse tareas como: eliminar registros duplicados, tratar valores faltantes, corregir registros incorrectos y homogeneizar monedas o unidades. Debe distinguirse entre las tareas de depuración que no utilizan información estadística de la muestra y aquellas transformaciones cuyos parámetros se estiman a partir de los datos. Estas últimas, como la imputación estadística, normalización, estandarización o determinadas técnicas de codificación, deben estimarse únicamente con el conjunto de entrenamiento y aplicarse posteriormente a los conjuntos de validación y prueba, con objeto de evitar fugas de información (*data leakage*).

4. **División de datos**: una vez realizada la preparación inicial, los datos se dividen en subconjuntos con finalidades diferentes:
   - Conjunto de entrenamiento: utilizado para que el modelo aprenda.
   - Validación: utilizado para ajustar el modelo y sus hiperparámetros.
   - Prueba: utilizado para evaluar el desempeño final.

5. **Ingeniería de características**: una vez divididos los datos, se trabaja sobre las variables que serán utilizadas como entrada del modelo. Consiste en seleccionar, transformar o crear nuevas características a partir de las variables originales resultantes de la preparación de datos, que permitan representar mejor la información relevante para el problema. La selección de características puede realizarse mediante distintos métodos:
   - Métodos de filtrado (*filter methods*): seleccionan variables mediante criterios estadísticos independientes del modelo, como correlación, análisis de varianza (ANOVA) o la información mutua.
   - Métodos envolventes (*wrapper methods*): evalúan diferentes subconjuntos de variables utilizando el desempeño de un modelo como criterio de selección. Entre ellos se encuentran la eliminación recursiva de características (*recursive feature elimination*, RFE), la selección hacia adelante y la eliminación hacia atrás.
   - Métodos integrados (*embedded methods*): realizan la selección de variables durante el propio proceso de entrenamiento del modelo. Algunos ejemplos son la regresión LASSO (*least absolute shrinkage and selection operator*), los árboles de decisión y los bosques aleatorios (*random forests*).



Ilustración 2 . Ciclo de vida de un modelo de aprendizaje automático.
Nota. Elaboración propia.

6. **Entrenamiento del modelo**: se utiliza el conjunto de datos de entrenamiento, compuesto por las características previamente seleccionadas y, cuando corresponde, la variable objetivo que se desea predecir. Estos datos se introducen en un algoritmo de aprendizaje automático previamente seleccionado.

   Durante el entrenamiento, las predicciones del modelo se comparan con los valores observados mediante una función de pérdida, que cuantifica el error cometido. El objetivo es ajustar los parámetros del modelo de modo que el valor de esta función sea cada vez menor; es decir, minimizar la función de pérdida para que las predicciones se aproximen lo más posible a los valores reales.

7. **Validación del modelo:** una vez entrenado, se analiza qué tan bien funciona con datos que no utilizó directamente para aprender. El objetivo es comprobar su capacidad de generalización y detectar problemas como sobreajuste o subajuste. Para ello pueden utilizarse técnicas como la validación cruzada de k particiones, la validación cruzada estratificada y la validación cruzada para series temporales. La estrategia de validación debe respetar la estructura de dependencia de los datos. La validación cruzada convencional o estratificada resulta adecuada cuando las observaciones pueden considerarse independientes; cuando existen registros repetidos de un mismo cliente, contraparte o emisor, la división debe evitar que una misma entidad aparezca simultáneamente en entrenamiento y validación. Para series temporales debe preservarse el

orden cronológico mediante esquemas de ventana móvil o creciente, evitando particiones aleatorias que puedan incorporar información futura al proceso de entrenamiento.

El desempeño se evalúa mediante métricas adecuadas al tipo de tarea: error absoluto medio, error cuadrático medio, la raíz del error cuadrático medio, el coeficiente de determinación R^2, medida F1 y el área bajo la curva ROC. También pueden emplearse métricas como el coeficiente de Gini, el estadístico Kolmogorov-Smirnov (KS) y el *Brier Score* para evaluar el desempeño de modelos de clasificación. Las métricas deben seleccionarse de acuerdo con la naturaleza del problema. En regresión pueden emplearse MAE, MSE, RMSE o R^2. En clasificación, métricas como ROC-AUC, Gini, KS, precisión, exhaustividad o F1 evalúan fundamentalmente la capacidad de discriminación, mientras que medidas como el *Brier* Score y las curvas de calibración permiten evaluar la calidad de las probabilidades estimadas. Esta distinción resulta especialmente relevante cuando la salida del modelo se utiliza como una probabilidad de riesgo.

8. **Ajuste de los hiperparámetros:** los hiperparámetros son configuraciones que se establecen antes del entrenamiento y que controlan cómo aprende el modelo. El objetivo de esta etapa es encontrar la combinación de hiperparámetros que produzca el mejor desempeño sin generar sobreajuste. Para ello se prueban diferentes configuraciones mediante métodos como: búsqueda en cuadrícula (*Grid Search*), búsqueda aleatoria (*Random Search*), optimización bayesiana (*Bayesian Optimization*). Cada configuración suele evaluarse mediante validación cruzada y posteriormente se selecciona aquella que presenta el mejor desempeño promedio. En algunos problemas de optimización, la función objetivo puede evaluarse, pero su forma funcional no está disponible de manera explícita o no permite calcular gradientes. En este caso, el problema se trata como una caja negra. No pueden emplearse directamente métodos basados en gradientes. Se utilizan entonces métodos de optimización libres de derivadas. En aprendizaje automático, este problema aparece especialmente en la selección de hiperparámetros.

9. **Reentrenamiento del modelo final**: una vez seleccionados el algoritmo y los mejores hiperparámetros, se entrena nuevamente el modelo utilizando la configuración definitiva.

10. **Evaluación final**: el modelo final se evalúa utilizando el conjunto de prueba, que debe mantenerse separado durante todo el proceso anterior. Esta etapa proporciona una estimación más objetiva de cómo funcionará el modelo con datos nuevos.

11. **Uso con datos nuevos**: si los resultados son satisfactorios, el modelo se utiliza con nuevos datos para generar predicciones.

Una vez comprendido el ciclo de vida de un modelo de aprendizaje automático, a continuación, se presentan y describen algunos de los modelos más utilizados en este campo.

1. APRENDIZAJE SUPERVISADO

El aprendizaje supervisado busca aprender una función que relacione un conjunto de variables de entrada con una variable objetivo conocida. El objetivo es aprender una función: $f: \mathcal{X} \to \mathcal{Y}$, tal que, dada una observación representada por el vector de características $\mathbf{x}$, el modelo genere una predicción: $\hat{y} = f(\mathbf{x})$. El modelo aprende utilizando un conjunto de entrenamiento formado por N pares entrada–salida $\mathcal{D} = \{(\mathbf{x}_n, y_n)\}_{n=1}^{N}$, donde cada $\mathbf{x}_n$ contiene las características de la observación n mientras que y_n corresponde a su salida conocida. El objetivo es encontrar un mapeo f capaz de producir predicciones $\hat{y}$ que generalicen correctamente sobre nuevas observaciones.

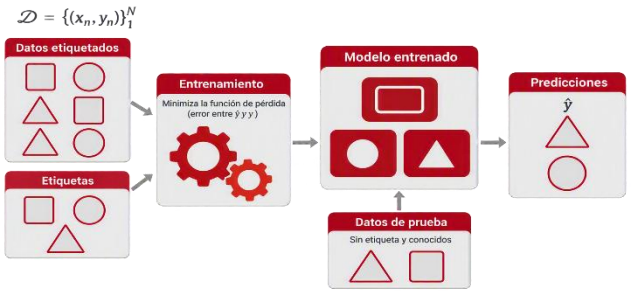


Ilustración 3. Mecanismo del aprendizaje automático.
Nota. Elaboración propia.

Una vez definido el mapeo, el entrenamiento consiste en encontrar los parámetros del modelo que hagan que sus predicciones sean lo más cercanas posible a los valores reales del conjunto de entrenamiento. Para cada observación n, el modelo genera

$$\hat{y}_n = f(\mathbf{x}_n; \boldsymbol{\theta}),$$

donde $\boldsymbol{\theta}$ es el vector de parámetros que el modelo debe aprender. El error de cada predicción se mide mediante una función de pérdida

$$\ell(y_n, \hat{y}_n),$$

la función utilizada depende del problema. El error total del modelo se obtiene promediando la pérdida sobre las N observaciones del conjunto de entrenamiento

$$\mathcal{L}(\boldsymbol{\theta}) = \frac{1}{N}\sum_{n=1}^{N} \ell\left(y_n, f(\mathbf{x_n}; \boldsymbol{\theta})\right),$$

a esta cantidad se le denomina **riesgo empírico**. Entrenar el modelo consiste entonces en encontrar los parámetros $\boldsymbol{\theta}$ que minimicen dicho error

$$\hat{\boldsymbol{\theta}} = \arg\min_{\boldsymbol{\theta}}\{\mathcal{L}(\boldsymbol{\theta})\}.$$

Este procedimiento se conoce como optimización. En los métodos iterativos se parte de un valor inicial $\boldsymbol{\theta}_0$ y los parámetros se actualizan sucesivamente según:

$$\boldsymbol{\theta}_{t+1} = \boldsymbol{\theta}_t + \eta_t \mathbf{d}_t,$$

donde $\mathbf{d}_t$ es la dirección de actualización y η_t es la tasa de aprendizaje, las actualizaciones continúan hasta satisfacer un criterio de convergencia, que puede estar basado en la magnitud del gradiente, la variación de la función objetivo, el cambio en los parámetros o un número máximo de iteraciones. Existen varios métodos de optimización:

- **Métodos de primer orden**: utilizan únicamente el gradiente de la función objetivo para determinar hacia dónde modificar los parámetros. El gradiente se define como
$$\nabla_{\boldsymbol{\theta}}\mathcal{L}(\boldsymbol{\theta}_t),$$
y señala la dirección de máximo crecimiento de $\mathcal{L}$. Por tanto, una dirección natural para minimizarla es la dirección contraria
$$\mathbf{d}_t = -\nabla_{\boldsymbol{\theta}}\mathcal{L}(\boldsymbol{\theta}_t),$$
estos métodos son computacionalmente económicos porque sólo requieren el gradiente, aunque pueden converger lentamente cuando la geometría de la función objetivo es desfavorable.

  Dentro de los métodos de primer orden, el gradiente puede calcularse utilizando todos los datos o aproximarse mediante un subconjunto de observaciones. El descenso de gradiente estocástico utiliza un minilote, $\mathfrak{B}_t$:
$$\mathbf{g}_t \approx \frac{1}{|\mathfrak{B}_t|}\sum_{n\in\mathfrak{B}_t} \nabla_{\boldsymbol{\theta}}\,\ell\big(y_n, f(\mathbf{x}_n; \boldsymbol{\theta}_t)\big),$$
posteriormente actualiza los parámetros recursivamente
$$\boldsymbol{\theta}_{t+1} = \boldsymbol{\theta}_t - \eta_t \mathbf{g}_t.$$
  Esto reduce el costo de cada iteración y hace que el descenso de gradiente estocástico sea especialmente útil para conjuntos de datos grandes. Esta aproximación mediante minilotes es ampliamente utilizada en aprendizaje automático.

- **Métodos de segundo orden**: incorporan, además del gradiente, información sobre la curvatura de la función objetivo mediante la matriz Hessiana[1]
$$\nabla^2_{\boldsymbol{\theta}}\mathcal{L}(\boldsymbol{\theta}_t),$$
o aproximaciones de ésta, estos métodos pueden converger más rápido que los de primer orden. Sin embargo, calcular y manipular la Hessiana puede resultar costoso cuando el número de parámetros es elevado.

El entrenamiento de un modelo supervisado permite obtener los parámetros

$$\hat{\boldsymbol{\theta}} = \arg\min_{\boldsymbol{\theta}}\{\mathcal{L}(\boldsymbol{\theta}; \mathcal{D}_{\text{entrenamiento}})\},$$

que hacen que sus predicciones se aproximen lo más posible a los valores observados en el conjunto de entrenamiento. Sin embargo, un modelo no resulta útil sólo por ajustarse bien a estos datos, sino por conservar un buen desempeño cuando recibe observaciones que no ha visto previamente; esta capacidad se le denomina **generalización** y supone que la pérdida esperada sobre datos futuros no observados sea lo más reducida posible.

[1] Para una función $f: \mathbb{R}^d \to \mathbb{R}$, la matriz Hessiana es la matriz de derivadas parciales de segundo orden, $H_f(\mathbf{x}) = \nabla^2 f(\mathbf{x}) = \left[\frac{\partial^2 f(\mathbf{x})}{\partial x_i\, \partial x_j}\right]_{i,j=1}^{d}$ y caracteriza la curvatura de f.

Si los datos provienen de una distribución verdadera desconocida $p(\boldsymbol{x}, y)$ se define el riesgo poblacional o pérdida esperada como

$$\mathcal{L}(\boldsymbol{\theta}; p) = \mathbb{E}_p[\ell(y, f(\mathbf{x}; \boldsymbol{\theta}))],$$

es el error esperado del modelo sobre nuevas observaciones provenientes de la misma distribución. La diferencia entre el riesgo poblacional y el riesgo obtenido sobre los datos de entrenamiento se denomina brecha de generalización:

$$\mathcal{L}(\boldsymbol{\theta}; p) - \mathcal{L}(\boldsymbol{\theta}; \mathcal{D}_{\text{entrenamiento}}),$$

cuando el modelo obtiene una pérdida muy baja sobre el conjunto de entrenamiento, pero una pérdida considerablemente mayor sobre datos nuevos, se presenta **sobreajuste**. Es decir, el modelo se adapta excesivamente a las particularidades de los datos utilizados para entrenarlo y pierde capacidad de generalización.

En la práctica, la distribución $p(\mathbf{x}, y)$ es desconocida. Por ello, los datos disponibles se dividen en tres conjuntos disjuntos:

$$\mathcal{D} = \mathcal{D}_{\text{entrenamiento}} \cup \mathcal{D}_{\text{validación}} \cup \mathcal{D}_{\text{prueba}}.$$

El conjunto de prueba no debe emplearse ni para ajustar los parámetros ni para seleccionar el modelo.

Un modelo con suficiente flexibilidad puede ajustarse excesivamente a los datos de entrenamiento, obteniendo una pérdida muy pequeña sin necesariamente generalizar bien a nuevas observaciones. Una de las principales estrategias para controlar este problema es la **regularización** (consultar la regresión regularizada de la Sección 2.1).

CLASIFICACIÓN

La **clasificación** es un problema de aprendizaje supervisado en el que la variable objetivo pertenece a un conjunto finito de categorías o clases. El propósito del modelo es aprender, a partir de las características disponibles, qué clase corresponde a cada observación.

En muchos casos, el modelo no asigna directamente una clase, sino que estima primero la probabilidad de pertenencia a cada una de las categorías posibles. La predicción final suele corresponder a la clase con la mayor probabilidad estimada.

REGRESIÓN

La **regresión** es un problema en el que la variable objetivo toma valores numéricos, generalmente continuos. En este caso, el modelo busca estimar una cantidad a partir de las características disponibles.

La diferencia fundamental entre ambos tipos de problemas radica en la naturaleza del resultado que se desea obtener: la clasificación predice una categoría, mientras que la regresión estima un valor numérico.

Una vez descritas estas dos tareas fundamentales del aprendizaje supervisado, se presentan a continuación los principales modelos utilizados para abordarlas.

1.1 Clasificador Bayesiano

El clasificador bayesiano ingenuo (*naive Bayes*) es un modelo de clasificación basado en el teorema de Bayes. Su funcionamiento consiste en estimar, para una nueva observación, qué tan probable es que pertenezca a cada una de las clases posibles a partir de las características observadas; posteriormente, se asigna la observación a la clase que presenta la mayor probabilidad.

La principal simplificación del modelo consiste en suponer que las características son independientes entre sí una vez conocida la clase. En otras palabras, el modelo analiza la información aportada por cada característica por separado y después combina dichas evidencias para determinar la clase más probable. Aunque este supuesto puede no cumplirse estrictamente en datos reales, permite construir modelos sencillos, rápidos y con un costo computacional reducido.

1.2 K vecinos más cercanos

El método de k vecinos más cercanos (*k-nearest neighbors*, KNN) clasifica una nueva observación comparándola con las observaciones conocidas que presentan características más similares. A diferencia de otros modelos, no construye previamente una función de clasificación; en su lugar, conserva los datos de entrenamiento y realiza la clasificación cuando recibe una nueva observación.

El procedimiento es intuitivo: primero se determina cuáles son las k observaciones más cercanas al nuevo dato y, posteriormente, se observa a qué clases pertenecen. La nueva observación se asigna a la clase que aparece con mayor frecuencia entre esos vecinos.

Como se muestra en la Ilustración 4, el resultado depende del número de vecinos considerado. Cuando $k = 1$, únicamente se utiliza el vecino más cercano; por tanto, si este pertenece a la clase azul, la nueva observación también se clasifica como azul. Cuando $k = 3$, se consideran los tres vecinos más próximos y se aplica una regla de mayoría: en el ejemplo, dos son azules y uno amarillo, por lo que la observación se clasifica como azul. En cambio, cuando $k = 4$, aparecen dos vecinos amarillos y dos azules, lo que produce un empate y dificulta la asignación de una clase.

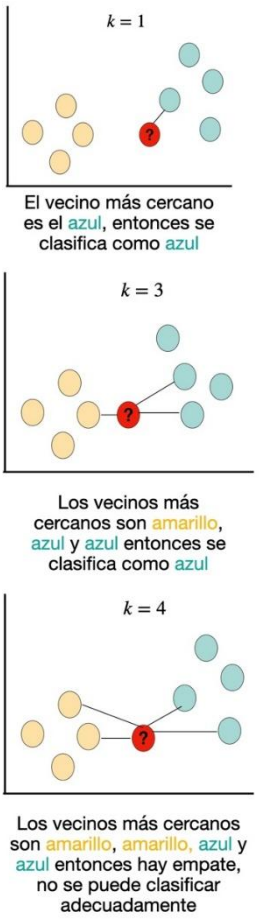
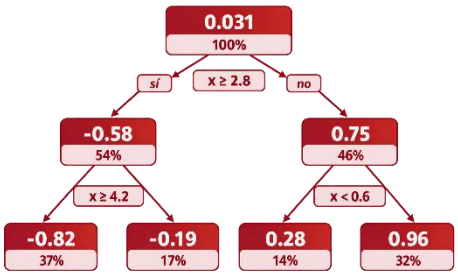


Ilustración 4. KNN.
Nota. Adaptado de Nami (2021).

Para determinar cuáles observaciones son "cercanas", KNN utiliza una medida de distancia. Entre las más comunes se encuentran la distancia euclidiana, que representa la distancia directa entre dos puntos; la distancia Manhattan, que suma las diferencias existentes en cada característica; la distancia de Mahalanobis, que considera también la variabilidad y relación entre las variables; y la distancia de Minkowski, que constituye una forma más general que incluye varios criterios de distancia.

Ilustración 5. Árbol de regresión.
Nota. Elaboración propia.



1.3 Árbol de decisión

Un árbol de decisión (*decision tree*) puede utilizarse tanto para problemas de clasificación como de regresión. En ambos casos, divide recursivamente el espacio de características mediante una secuencia de reglas. En cada nodo i se selecciona una característica $j(i)$ y un umbral u_i; las observaciones se dividen según cumplan o no $x_{j(i)} \leq u_i$.

La división se selecciona buscando reducir un criterio de impureza, como el índice de Gini

$$G = 1 - \sum_{k=1}^{K} p_k^2,$$

o la entropía

$$H = -\sum_{k=1}^{K} p_k \log\{p_k\},$$

donde p_k es la proporción de observaciones de la clase k en el nodo. En regresión puede utilizarse el error cuadrático medio.

Las observaciones que cumplen la condición se envían a una rama y las restantes a la otra. En problemas de clasificación, cada hoja puede definirse como $R_1 = \{\mathbf{x}: x_1 \leq u_1,\ x_2 \leq u_2\}$, contiene todas las observaciones que cumplen simultáneamente esas reglas. La representación general del árbol es:

$$f(\mathbf{x};\boldsymbol{\theta}) = \sum_{j=1}^{J} w_j \, \mathbb{1}_{\{\mathbf{x}\in R_j\}},$$

donde $\boldsymbol{\theta}$ es el conjunto de elementos que determinan la estructura y las predicciones del árbol; $\mathbb{1}_{\{\cdot\}}$ es la función indicadora. En el caso de regresión, w_j es la predicción asociada a esa región, obtenida como

$$w_j = \frac{\sum_{n=1}^{N} y_n \, \mathbb{1}_{\{\mathbf{x}_n\in R_j\}}}{\sum_{n=1}^{N} \mathbb{1}_{\{\mathbf{x}_n\in R_j\}}}.$$

En clasificación, cada hoja estima las proporciones p_{jk} de cada clase y la predicción corresponde a la clase mayoritaria. De esta forma, como las regiones R_j forman una partición del espacio de características, cada nueva observación pertenece a una única región terminal. Por tanto, sólo uno de los términos de la suma determina la predicción, la cual corresponde al valor asociado a dicha región, ya sea una clase en clasificación o un valor numérico en regresión.

1.4 Bosque aleatorio

Un bosque aleatorio es un modelo de ensamble formado por múltiples árboles de decisión. Puede considerarse no paramétrico, ya que su estructura no está determinada por un número fijo de coeficientes.

La idea parte de entrenar M árboles distintos $f_1(\mathbf{x}), f_2(\mathbf{x}), \ldots, f_M(\mathbf{x})$, pero introduciendo aleatoriedad de dos formas:

- Cada árbol se entrena con una muestra de los datos obtenida mediante muestreo con reemplazo.
- En cada nodo, el árbol considera únicamente un subconjunto aleatorio de las características disponibles. Esta segunda fuente de aleatoriedad busca reducir la correlación entre los árboles.

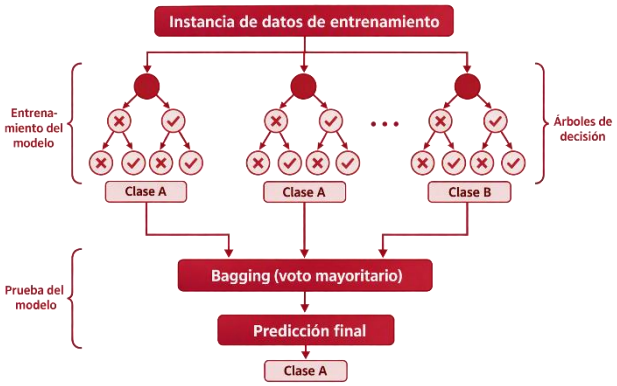


Ilustración 6. Bosque aleatorio.
Nota. Elaboración propia.

Para cada árbol m se genera una muestra *bootstrap* de tamaño N a partir del conjunto de entrenamiento, es decir, mediante muestreo con reemplazo. En cada nodo i, se selecciona aleatoriamente un subconjunto S_i de m_{try} características entre las D disponibles:

$$S_i \subset \{1, \ldots, D\}, \qquad |S_i| = m_{try} \leq D$$

y la división se realiza sólo entre ellas. La predicción final se obtiene combinando los M árboles: en regresión,

$$\hat{y}(\mathbf{x}) = \frac{1}{M} \sum_{m=1}^{M} \hat{f}_m(\mathbf{x}),$$

mientras que para clasificación se utiliza voto mayoritario entre las clases predichas por los árboles.

1.5 Máquina de vectores de soporte

La máquina de vectores de soporte es un modelo de clasificación no probabilístico que busca una frontera de decisión con el mayor margen posible entre las clases. Para clasificación binaria se suelen usar etiquetas $y_n \in \{-1,1\}$, la función de decisión lineal es

$$f(\mathbf{x}) = \mathbf{w}^{\top}\mathbf{x} + w_0,$$

donde $\mathbf{w}$ determina la orientación de la frontera y w_0 su desplazamiento. La distancia de una observación a la frontera está dada por:

$$|r| = \frac{|f(\mathbf{x})|}{\|\mathbf{w}\|},$$

el modelo busca maximizar el margen entre las clases. Las observaciones que determinan dicho margen se denominan vectores de soporte.

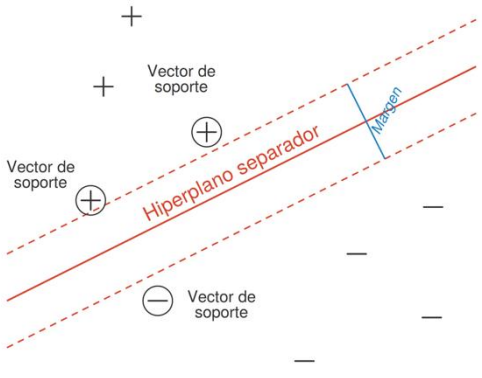


Ilustración 7. Hiperplano de la máquina de vectores de soporte.
Nota. Adaptado de MathWorks (s.f.).

El margen corresponde a la distancia entre los dos hiperplanos paralelos que delimitan la región alrededor de la frontera de decisión. Su amplitud es $\frac{2}{\|\mathbf{w}\|}$. Por tanto, maximizar el margen equivale a minimizar $\frac{1}{2}\|\mathbf{w}\|^2$.

Cuando los datos son separables, la optimización

$$\min_{\mathbf{w},w_0} \frac{1}{2}\|\mathbf{w}\|^2, \quad \text{sujeto a } y_n(\mathbf{w}^\top \mathbf{x}_n + w_0) \geq 1,$$

equivale a maximizar el margen. Cuando los datos no son linealmente separables, se permite cierto grado de violación mediante

$$\min_{\mathbf{w},w_0} \frac{1}{2}\|\mathbf{w}\|_2^2 + C\sum_{n=1}^{N} \xi_n, \quad \text{sujeto a } y_n(\mathbf{w}^\top \mathbf{x}_n + w_0) \geq 1 - \xi_n, \quad \xi_n \geq 0,$$

donde ξ_n mide cuánto viola el margen la observación n, mientras que C controla cuánto se penalizan esas violaciones.

La frontera lineal puede ser insuficiente cuando las clases presentan relaciones no lineales. Para estos casos se utiliza una función núcleo, también denominada **kernel**: $K(\mathbf{x}, \mathbf{x}')$, entonces, la función de decisión puede escribirse como

$$f(\mathbf{x}) = \sum_{n\in S} \alpha_n\, y_n K(\mathbf{x_n}, \mathbf{x}) + w_0,$$

donde α_n es el peso asociado al vector de soporte n y S es el conjunto de vectores de soporte. En la Tabla 1 se resumen los núcleos más utilizados.

Tabla 1. Funciones núcleo.

Núcleo	Expresión	Idea principal
Lineal	$K(\mathbf{x}, \mathbf{x}') = (\mathbf{x}^\top \mathbf{x}')$	No introduce una transformación no lineal; corresponde a una frontera de decisión lineal en el espacio original de características.
Polinómico	$K(\mathbf{x}, \mathbf{x}') = \left(\gamma \mathbf{x}^\top \mathbf{x}' + c\right)^d$	Permite relaciones polinómicas de grado *d*.
Gaussiano o de función de base radial	$K(\mathbf{x}, \mathbf{x}') = \exp\left(-\frac{\|\mathbf{x} - \mathbf{x}'\|^2}{2h^2}\right)$	Permite fronteras no lineales muy flexibles basadas en la distancia entre observaciones.

1.6 Potenciación por gradiente

La potenciación por gradiente (*gradient boosting*) es un método de ensamble que construye el modelo secuencialmente. Cada nuevo modelo débil intenta corregir los errores del modelo acumulado hasta ese momento. Es una forma general de realizar descenso por gradiente directamente en el espacio de funciones.

Cuando los modelos débiles son árboles de decisión, como ocurre habitualmente, el modelo resultante se suele considerar no paramétrico. Inicialmente, se considera un modelo constante $f_0(x)$, obtenido mediante la minimización de la función de pérdida

$$f_0(\mathbf{x}) = \arg\min_{c\in\mathbb{R}} \left\{ \sum_{n=1}^{N} \ell\,(y_n, c) \right\}.$$

Sea el modelo después de $m-1$ iteraciones, $f_{m-1}(\mathbf{x})$, en la iteración m, primero se calcula el residuo de gradiente

$$r_{n,m} = -\left[\frac{\partial \ell\big(y_n, f(\mathbf{x}_n)\big)}{\partial f(\mathbf{x}_n)}\right]_{f=f_{m-1}},$$

esto indica en qué dirección debe corregirse la predicción de la observación n para reducir la función de pérdida. Después se entrena un nuevo modelo débil $F_m(\mathbf{x})$ normalmente un árbol pequeño, para aproximar esos residuos

$$F_m = \arg\min_{F} \left\{ \sum_{n=1}^{N} \Big(r_{n,m} - F(\mathbf{x}_n)\Big)^2 \right\},$$

finalmente, el modelo acumulado se actualiza

$$f_m(\mathbf{x}) = f_{m-1}(\mathbf{x}) + \eta F_m(\mathbf{x}),$$

donde η es la tasa de aprendizaje que controla cuánto aporta cada nuevo modelo.

Las implementaciones más utilizadas son: *XGBoost*, *LightGBM* y *CatBoost*. Particularmente, *CatBoost* (*Categorical Boosting*) es una implementación de potenciación por gradiente diseñada para manejar de forma directa variables categóricas y reducir el sobreajuste asociado tanto a su codificación como al cálculo sesgado de los gradientes. El conjunto de observaciones $\mathcal{D} = \{(\mathbf{X}_i, Y_i)\}_{i=1}^{N}$, donde $\mathbf{X}_i$ contiene variables numéricas y categóricas. Para una variable categórica $x_{i,k}$ una codificación directa mediante la media de Y para cada categoría puede producir sobreajuste, pues utiliza la propia respuesta de la observación para construir su predictor. *CatBoost* evita este problema utilizando una permutación aleatoria $\sigma = (\sigma_1, \dots, \sigma_N)$, y, para la observación situada en la posición p, calcula el estadístico ordenado

$$\frac{\sum_{j=1}^{p-1} \mathbb{1}_{\{x_{\sigma_j,k} = x_{\sigma_p,k}\}} Y_{\sigma_j} + aP}{\sum_{j=1}^{p-1} \mathbb{1}_{\{x_{\sigma_j,k} = x_{\sigma_p,k}\}} + a},$$

donde P es un valor previo y $a > 0$ controla su peso. De esta manera, la representación de una categoría para la observación σ_p utiliza únicamente observaciones que aparecen antes de ella en la permutación, evitando emplear directamente su propia respuesta Y_{σ_p}.

Además, *CatBoost* utiliza *ordered boosting*, de modo que el gradiente de una observación se calcula utilizando un modelo construido únicamente con observaciones anteriores en la permutación σ,

$$g_i = \left. \frac{\partial \ell(y_i, f(\mathbf{x}_i))}{\partial f(\mathbf{x}_i)} \right|_{f = f_{<i}},$$

donde $f_{<i}$ representa el modelo entrenado sólo con las observaciones que preceden a i en la permutación σ. De esta forma, tanto la codificación de variables categóricas como el cálculo de gradientes evitan utilizar información de la propia observación.

Nótese que $g_i$representa el gradiente de la pérdida; por tanto, el correspondiente residuo de gradiente utilizado para actualizar el modelo es $-g_i$.

2. MODELOS ESTADÍSTICOS

En esta sección se presentan los principales modelos estadísticos de regresión y de series temporales. Aunque tienen su origen en la estadística, ambos pueden utilizarse con fines predictivos. Particularmente, los modelos de regresión pueden ser considerados dentro del aprendizaje supervisado.

La distinción adoptada en esta nota tiene un propósito expositivo y no implica que los modelos estadísticos constituyan un paradigma separado del aprendizaje automático. Diversos modelos de regresión pueden utilizarse dentro del aprendizaje supervisado cuando su objetivo principal es la predicción fuera de muestra.

2.1 Modelos de Regresión

Los modelos de regresión buscan aprender la relación entre un conjunto de variables explicativas $\mathbf{x}$ y una variable respuesta y. De forma general, buscan modelar una relación del tipo

$$y = f(\mathbf{x}) + \varepsilon,$$

donde $f(\mathbf{x})$ representa la componente sistemática del modelo y ε recoge la variabilidad no explicada. Dependiendo de su formulación, estos modelos pueden utilizarse tanto para inferencia, cuando el interés principal consiste en estudiar y cuantificar la relación entre las variables, como para predicción, cuando el objetivo es estimar el valor de y para nuevas observaciones.

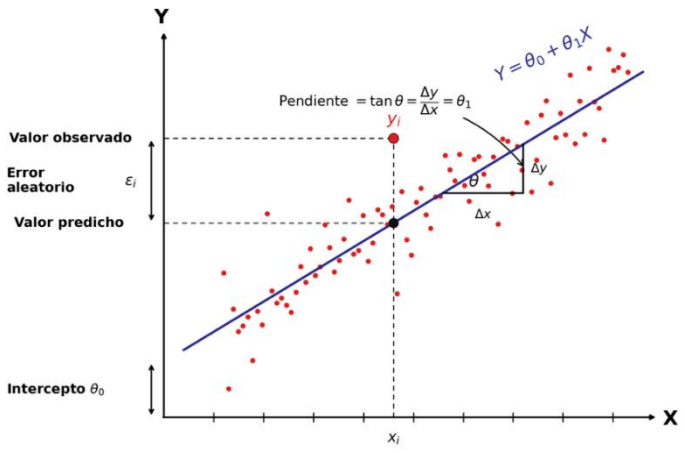


Ilustración 8. Regresión lineal.
Nota. Elaboración propia.

Esta última perspectiva establece una conexión directa con el aprendizaje automático supervisado. En ambos casos se dispone de un conjunto de observaciones $\mathcal{D} = \{(\mathbf{x}_n, \mathrm{y}_n)\}_{n=1}^{N}$ a partir del cual se estima un mapeo $f: \mathcal{X} \to \mathcal{Y}$ capaz de producir una predicción $\hat{y} = f(\mathbf{x})$. Por esta razón, modelos como la regresión lineal, los modelos lineales generalizados y sus extensiones pueden considerarse modelos estadísticos y, al mismo tiempo, utilizarse dentro del paradigma de aprendizaje supervisado.

La diferencia se encuentra principalmente en el enfoque: la estadística suele conceder especial importancia a la interpretación, la incertidumbre y las propiedades de los parámetros, mientras que el aprendizaje automático enfatiza la capacidad de generalización y el desempeño predictivo sobre datos no observados.

Los principales tipos de modelos de regresión son:

- Regresión lineal: propone que la media de la variable objetivo puede expresarse como una combinación lineal de las variables explicativas

$$y = \beta_0 + \boldsymbol{\beta}^\top \boldsymbol{x} + \varepsilon.$$

- Regresión polinómica: extiende la regresión lineal incorporando potencias de las variables para representar relaciones curvas

$$y = \sum_{m=0}^{M} \beta_m \, x^m + \varepsilon.$$

- Modelos lineales generalizados: extienden el modelo lineal permitiendo que y siga diferentes distribuciones de probabilidad, pertenecientes a la familia exponencial. La media condicional

$$\mu(\mathbf{x}) = \mathbb{E}[y \mid \mathbf{x}],$$

se relaciona con el predictor lineal mediante una función de enlace $g(\cdot)$:

$$g\big(\mu(\mathbf{x})\big) = \beta_0 + \boldsymbol{\beta}^\top \mathbf{x}.$$

Tipo	Distribución	Expresión característica
Poisson	$Y \mid \mathbf{x} \sim \text{Poisson}(\mu)$	$\log(\mu) = \beta_0 + \boldsymbol{\beta}^\top \mathbf{x}$
Binomial	$Y \mid \mathbf{x} \sim \text{Binomial}(n, p)$	$\log\left(\frac{p}{1-p}\right) = \beta_0 + \boldsymbol{\beta}^\top \mathbf{x}$
Gamma	$Y \mid \mathbf{x} \sim \text{Gamma}(\alpha, \mu/\alpha)$	$\frac{1}{\mu} = \beta_0 + \boldsymbol{\beta}^\top \mathbf{x}$
Gaussiana	$Y \mid \mathbf{x} \sim \mathcal{N}(\mu, \sigma^2)$	$\mu = \beta_0 + \boldsymbol{\beta}^\top \mathbf{x}$

Tabla 2. Distribuciones y expresiones características de modelos de regresión.

En particular, la regresión logística se emplea para problemas de clasificación binaria. Estima directamente la probabilidad de pertenecer a una clase. Para clasificación binaria:

$$\mathbb{P}(y = 1 \mid \mathbf{x}) = \rho\big(\beta_0 + \boldsymbol{\beta}^\top \mathbf{x}\big),$$

donde la función logística es

$$\rho(a) = \frac{1}{1 + e^{-a}}.$$

Se combinan linealmente las características (produce una puntuación) y posteriormente la función logística transforma ese valor en una probabilidad entre 0 y 1.

- Regresión regularizada: añade una penalización a la función de pérdida para controlar la magnitud o el número efectivo de coeficientes

$$\big(\hat{\boldsymbol{\beta}}_0, \hat{\boldsymbol{\beta}}\big) = \underset{\boldsymbol{\beta}_0, \boldsymbol{\beta}}{\arg\min} \left\{ \sum_{n=1}^{N} (y_n - \beta_0 - \boldsymbol{\beta}^\top \mathbf{x}_n)^2 + \lambda C(\boldsymbol{\beta}) \right\}.$$

Ridge	LASSO	Red elástica
$\mathcal{L}(\boldsymbol{\beta}) + \lambda \parallel \boldsymbol{\beta} \parallel_2^2$	$\mathcal{L}(\boldsymbol{\beta}) + \lambda \parallel \boldsymbol{\beta} \parallel_1$	$\mathcal{L}(\boldsymbol{\beta}) + \lambda_1 \parallel \boldsymbol{\beta} \parallel_1 + \lambda_2 \parallel \boldsymbol{\beta} \parallel_2^2$

Tabla 3. Métodos de regularización.

- Regularización LASSO: penaliza los parámetros del modelo con el propósito de reducir su complejidad. Su característica principal es que puede llevar algunos coeficientes exactamente a cero, de modo que las variables asociadas dejan de contribuir a la predicción. Por ello, además de controlar el sobreajuste, LASSO puede utilizarse como un método de selección de características.

- Regularización Ridge: también penaliza los parámetros del modelo, pero normalmente no los elimina por completo. En su lugar, reduce la magnitud de aquellos que tienen una influencia excesiva, distribuyendo de manera más equilibrada el peso entre las variables. De esta forma, ayuda a obtener modelos más estables y menos sensibles a los datos de entrenamiento.

- Red elástica: compagina LASSO y Ridge, es decir, selecciona variables relevantes y al mismo tiempo controla la complejidad del modelo para mejorar su estabilidad.

- Regresión mediante *splines*: ajusta funciones suaves por tramos. De forma general

$$f(x) = \sum_{j=1}^{J} \beta_j \, B_j(x),$$

donde $B_j(x)$ son funciones base y J es el número de funciones utilizadas. Un caso clásico es el spline cúbico.

- Regresión bayesiana: en lugar de considerar los parámetros como cantidades fijas desconocidas, se les asigna una distribución previa $\mathrm{p}(\boldsymbol{\theta})$. Después de observar los datos se obtiene la distribución posterior mediante

$$\mathrm{p}(\boldsymbol{\theta} \mid \mathcal{D}) = \frac{\mathrm{p}(\mathcal{D} \mid \boldsymbol{\theta})\mathrm{p}(\boldsymbol{\theta})}{\mathrm{p}(\mathcal{D})},$$

y la predicción se obtiene integrando sobre la incertidumbre de los parámetros

$$\mathrm{p}(y \mid \mathbf{x}, \mathcal{D}) = \int \mathrm{p}\,(y \mid \mathbf{x}, \boldsymbol{\theta})\, \mathrm{p}(\boldsymbol{\theta} \mid \mathcal{D})\, d\boldsymbol{\theta}.$$

2.2 Series temporales

Una serie de tiempo es una secuencia de observaciones registradas sucesivamente en el tiempo

$$\{y_t\}_{t=1}^{T} = \{y_1, y_2, \ldots, y_T\},$$

donde y_t representa el valor observado en el instante t. En una serie temporal, las observaciones suelen presentar dependencia temporal, por lo que valores cercanos en el tiempo pueden estar relacionados entre sí (Box, Jenkins, Reinsel, & Ljung, 2016).

La relación entre observaciones separadas en el tiempo se estudia mediante el concepto de rezago (*lag*). Un **rezago** indica cuántos periodos separan dos observaciones. Por ejemplo, en una serie mensual, un rezago de uno compara cada observación con la del mes anterior, mientras que un rezago de doce permite analizar su relación con el valor observado un año antes.

La **autocorrelación** mide el grado en que los valores de una serie están relacionados con sus propios valores pasados para un determinado rezago. Una autocorrelación positiva indica que valores altos tienden a estar seguidos por valores altos, y valores bajos por valores bajos; una autocorrelación negativa indica que valores altos tienden a ser seguidos por valores bajos,

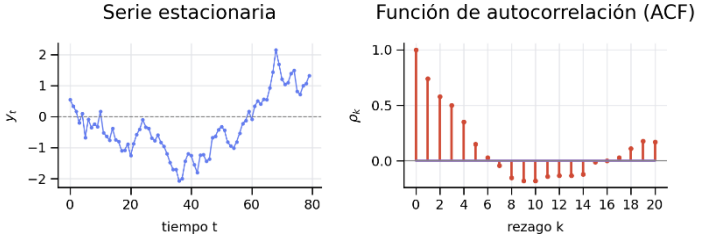


Ilustración 9. Serie de tiempo.
Nota. Elaboración propia.

o viceversa. El análisis de la autocorrelación permite, por tanto, identificar si la información pasada contiene patrones que pueden ser útiles para explicar o pronosticar el comportamiento futuro de la serie.

Una propiedad fundamental en el análisis de series de tiempo es la **estacionariedad**. Una serie débilmente estacionaria mantiene una media y una varianza constantes en el tiempo y una autocovarianza que depende del rezago entre observaciones, no del momento específico en el que fueron registradas. Esta propiedad es importante porque permite suponer que los patrones identificados en el pasado pueden seguir siendo informativos en otros momentos.

El análisis de series de tiempo busca describir y modelar estas relaciones temporales con el propósito de comprender la dinámica de una variable y, especialmente, realizar pronósticos. Para ello, los modelos utilizan información disponible hasta un momento determinado para estimar valores futuros.

Tabla 4 . Modelos de series temporales

Modelo	Descripción breve
AR — Autorregresivo *(Autoregressive)*	Explica el valor actual de una variable a partir de sus propios valores pasados. Es útil cuando las observaciones recientes contienen información sobre su comportamiento futuro.
MA — Media móvil *(Moving Average)*	Explica el valor actual a partir de los errores o perturbaciones ocurridos en periodos anteriores. Permite incorporar el efecto temporal de eventos no anticipados.
ARMA — Autorregresivo de medias móviles *(Autoregressive Moving Average)*	Combina los modelos AR y MA; utiliza tanto los valores pasados de la variable como los errores pasados para explicar su comportamiento actual.
ARIMA — Autorregresivo integrado de medias móviles *(Autoregressive Integrated Moving Average)*	Extiende el modelo ARMA mediante la diferenciación de la serie, lo que permite trabajar con series que presentan tendencias y no son estacionarias originalmente.
SARIMA — Autorregresivo integrado de medias móviles estacional *(Seasonal ARIMA)*	Extiende ARIMA incorporando patrones que se repiten periódicamente, como comportamientos mensuales, trimestrales o anuales.
ARIMAX — ARIMA con variables explicativas *(ARIMA with Exogenous Variables)*	Amplía ARIMA incorporando variables externas que pueden ayudar a explicar o pronosticar la variable de interés.
SARIMAX — SARIMA con variables explicativas *(SARIMA with Exogenous Variables)*	Combina los patrones autorregresivos, de medias móviles y estacionales con información proveniente de variables externas.
GARCH — Heterocedasticidad condicional autorregresiva generalizada *(Generalized Autoregressive Conditional Heteroskedasticity)*	Modela cambios en la variabilidad de una serie a lo largo del tiempo. Es especialmente utilizado en finanzas para representar periodos de alta y baja volatilidad.
EWMA — Media móvil ponderada exponencialmente *(Exponentially Weighted Moving Average)*	Es un método de suavizamiento que asigna mayor importancia a las observaciones recientes y reduce progresivamente el peso de las más antiguas. Se utiliza, entre otras aplicaciones, para estimar tendencias y volatilidad.

3. REDES NEURONALES

Una red neuronal artificial (*artificial neural network*) es un modelo formado por unidades de procesamiento denominadas neuronas, organizadas en capas y conectadas entre sí. Su finalidad es aprender relaciones entre las variables de entrada y la variable de salida mediante el ajuste de los pesos asociados a estas conexiones (Aggarwal, 2018). Su objetivo es aprender un mapeo f entre un vector de entrada $\mathbf{x}$ y una salida $\hat{y}$,

$$\hat{y} = f(\mathbf{x}; \boldsymbol{\theta}),$$

donde $\boldsymbol{\theta}$ representa el conjunto de pesos y sesgos de la red.

Como se muestra en la Ilustración 10, la información ingresa a través de la capa de entrada, atraviesa una o varias capas ocultas y finalmente llega a la capa de salida, donde se genera la predicción.

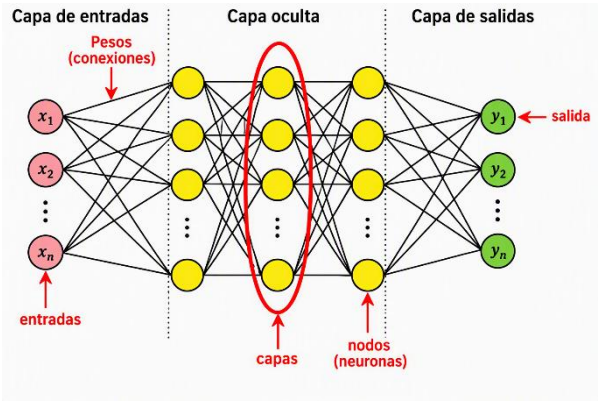


Ilustración 10. Red neuronal.
Nota. Elaboración propia.

Cada neurona recibe información procedente de la capa anterior, asigna distinta importancia a cada entrada mediante los pesos y combina esta información antes de transmitirla a la siguiente capa. Durante el entrenamiento, dichos pesos se ajustan progresivamente con el propósito de reducir el error de las predicciones.

Después de combinar la información recibida, cada neurona aplica una función de activación ϕ, que determina cómo se transforma la señal antes de continuar por la red. Estas funciones permiten introducir relaciones no lineales y, con ello, representar patrones más complejos.

Función de activación	Descripción
Lineal ***(linear)***	Transmite el valor recibido sin introducir una transformación no lineal. Se utiliza principalmente en la capa de salida de problemas de regresión, cuando se desea predecir un valor numérico sin restringirlo a un intervalo específico.
Escalonada ***(step function)***	Genera una salida distinta según se supere o no un determinado umbral. Se utiliza principalmente para representar decisiones binarias simples.
Unidad lineal rectificada ***(rectified linear unit, ReLU)***	Elimina los valores negativos y conserva los positivos. Es una de las funciones más utilizadas en las capas ocultas por su simplicidad y eficiencia computacional.
Sigmoide ***(sigmoid)***	Transforma la entrada en un valor entre cero y uno, por lo que suele utilizarse en la capa de salida de problemas de clasificación binaria.
Tangente hiperbólica ***(hyperbolic tangent, tanh)***	Transforma la entrada en valores comprendidos entre menos uno y uno, permitiendo representar tanto señales negativas como positivas.
Softsign	Comprime los valores dentro de un intervalo entre menos uno y uno mediante una transformación suave, con un comportamiento similar al de la tangente hiperbólica.
Softmax	Transforma las salidas de un modelo en probabilidades que suman uno, principalmente utilizada en clasificación multiclase.

Tabla 5. Funciones de activación.

En una red con múltiples capas, este proceso se repite sucesivamente. Si $h^{(\ell)}$ representa la salida de la capa ℓ, entonces

$$h^{(\ell)} = \phi^{(\ell)}\left(\mathbf{W}^{(\ell)} h^{(\ell-1)} + \mathbf{b}^{(\ell)}\right), \quad h^{(0)} = \mathbf{x},$$

donde $\mathbf{W}^{(\ell)}$ y $\mathbf{b}^{(\ell)}$ son la matriz de pesos y vector de sesgos de la capa (ℓ) respectivamente y $\phi^{(\ell)}$ es la función de activación.

3.1 Teorema de aproximación universal

El teorema de aproximación universal establece que una red neuronal de propagación hacia adelante, con al menos una capa oculta y un número suficiente de neuronas, puede aproximar una función continua $f(\mathbf{x})$ con una precisión arbitraria, bajo condiciones adecuadas sobre la función de activación. Si $f: \Omega \subset \mathbb{R}^n \to \mathbb{R}$, con Ω compacto, entonces, para cualquier $\varepsilon > 0$ existe una red neuronal $\hat{f}(\mathbf{x}, \boldsymbol{\theta})$ tal que

$$\sup_{\mathbf{x}\in\Omega} \left| f(\mathbf{x}) - \hat{f}(\mathbf{x}; \boldsymbol{\theta}) \right| < \varepsilon.$$

La red puede aproximar funciones no lineales $\hat{f}(\mathbf{x}; \boldsymbol{\theta}) \approx f(\mathbf{x})$ tan precisamente como se desee, siempre que tenga suficiente capacidad. No obstante, este teorema no proporciona una cota sobre el número de neuronas requerido, ni garantiza que el procedimiento de optimización permita alcanzar la aproximación establecida. Asimismo, el resultado no implica garantías sobre el desempeño de generalización de la red ante observaciones no utilizadas durante el entrenamiento.

3.2 Entrenamiento de una red neuronal

El entrenamiento de una red neuronal consiste en ajustar sus pesos y sesgos para reducir el error entre las predicciones y los valores reales. Para cada minilote, primero se realiza una propagación hacia adelante para obtener $\hat{y} = f(\mathbf{x}, \boldsymbol{\theta})$, y posteriormente se evalúa una función de pérdida $\mathcal{L}(\boldsymbol{\theta})$. Mediante retropropagación se calculan los gradientes de la pérdida respecto de los parámetros

$$\nabla_{\boldsymbol{\theta}} \mathcal{L}(\boldsymbol{\theta}),$$

y un algoritmo de optimización (primer orden o de segundo orden) los utiliza para actualizar dichos parámetros. Este procedimiento se repite durante múltiples iteraciones hasta obtener un ajuste adecuado.

3.3 Tipos de redes neuronales

Las redes neuronales pueden organizarse de distintas formas según la manera en que se conectan sus capas y el tipo de información que procesan. Entre las arquitecturas más utilizadas se encuentran las siguientes:

Tabla 6. Tipos de redes neuronales.

Tipo de red neuronal	Descripción
Red neuronal de propagación hacia adelante (*feedforward neural network*)	La información avanza únicamente desde la capa de entrada hasta la capa de salida, pasando por una o varias capas ocultas, sin regresar a capas anteriores. Es una de las arquitecturas más sencillas y se utiliza principalmente en problemas de clasificación y regresión.
Red neuronal convolucional (*convolutional neural network, CNN*)	Está diseñada para identificar patrones locales dentro de datos con estructura espacial. Mediante filtros, reconoce características como bordes, formas o texturas, por lo que se utiliza especialmente en el procesamiento de imágenes, aunque también puede aplicarse a otros tipos de datos estructurados.
Red neuronal recurrente (*recurrent neural network,* RNN)	Está orientada al procesamiento de datos secuenciales. Mantiene información de observaciones anteriores para utilizarla al procesar las siguientes, lo que permite representar dependencias a lo largo del tiempo. Se emplea, por ejemplo, en series temporales, texto y señales.
Red neuronal residual (*residual neural network,* ResNet)	Incorpora conexiones que permiten que la información pase directamente entre capas no consecutivas. Estas conexiones facilitan el entrenamiento de redes con muchas capas y reducen la pérdida de información durante su propagación.
Memoria a corto y largo plazo (long short-term memory, LSTM)	Es una variante de las RNN diseñada para capturar dependencias de largo plazo en datos secuenciales. Utiliza celdas de memoria y mecanismos de compuertas que permiten conservar o descartar información relevante a lo largo del tiempo, lo que reduce problemas como el desvanecimiento del gradiente. Se emplea ampliamente en procesamiento de lenguaje natural, reconocimiento de voz y predicción de series temporales.
Transformador	Arquitectura basada en mecanismos de atención (*attention*) que permite procesar simultáneamente todos los elementos de una secuencia, identificando las relaciones más relevantes entre ellos.

4. APRENDIZAJE NO SUPERVISADO

El aprendizaje no supervisado trabaja con datos sin etiquetas. El modelo recibe únicamente las observaciones x sin conocer una respuesta correcta y:

$$\mathcal{D} = \{\mathbf{x}_n\}_1^N,$$

donde cada $\mathbf{x}_n$ es un vector de características. No existe una variable objetivo y_n. Su objetivo es descubrir patrones o estructuras dentro de los datos. En aprendizaje supervisado, el modelo aprende una relación entre entrada y salida: $x \to y$. En aprendizaje no supervisado, intenta entender cómo están organizados los datos.

Desde una perspectiva probabilística, muchos métodos supervisados buscan estimar directa o indirectamente $\mathrm{p}(y \mid \boldsymbol{x})$, mientras que determinados métodos no supervisados modelan $\mathrm{p}(\boldsymbol{x})$ o estructuras latentes de los datos. Otros métodos no supervisados, como k-medias o PCA, se formulan mediante criterios geométricos o de reconstrucción sin especificar explícitamente una distribución probabilística.

Aunque no haya etiquetas, el modelo sí necesita una función objetivo

$$\hat{\boldsymbol{\theta}} = \arg\min_{\boldsymbol{\theta}}\{\mathcal{L}(\boldsymbol{\theta})\},$$

el entrenamiento consiste en ajustar los parámetros para minimizar una pérdida construida a partir de los propios datos.

4.1 Agrupamiento de k medias

El agrupamiento de k medias (*k-means*) tiene como objetivo dividir los datos en K grupos de manera que los elementos pertenecientes al mismo grupo sean similares entre sí. Cada grupo está representado por un centroide $\boldsymbol{\mu}_k$. Para cada observación $\mathbf{x}_n$, el algoritmo determina cuál de los K centroides se encuentra más cerca. Esta asignación se expresa como

$$z_n = \arg\min_{k\in\{1,\ldots,K\}}\{\|\mathbf{x}_n - \boldsymbol{\mu}_k\|^2\}.$$

Una vez realizadas las asignaciones, cada centroide se actualiza calculando la media de las observaciones que pertenecen a ese grupo

$$\boldsymbol{\mu}_k = \frac{1}{N_k} \sum_{n:z_n=k} \mathbf{x}_n \,.$$

La función objetivo es

$$\min_{\mu_k, z_n} \sum_{n=1}^{N} \left\| \mathbf{x}_n - \boldsymbol{\mu}_{z_n} \right\|^2 .$$

4.2 Modelo de mezcla gaussiana

El modelo de mezcla gaussiana (*Gaussian mixture model*, GMM) también busca grupos, pero no asigna necesariamente cada observación a un único grupo de forma absoluta. En su lugar, calcula una probabilidad de pertenencia.

Se supone que los datos proceden de una mezcla de K distribuciones gaussianas

$$\mathrm{p}(\mathbf{x} \mid \boldsymbol{\theta}) = \sum_{k=1}^{K} \pi_k \, \mathcal{N}(\mathbf{x} \mid \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k),$$

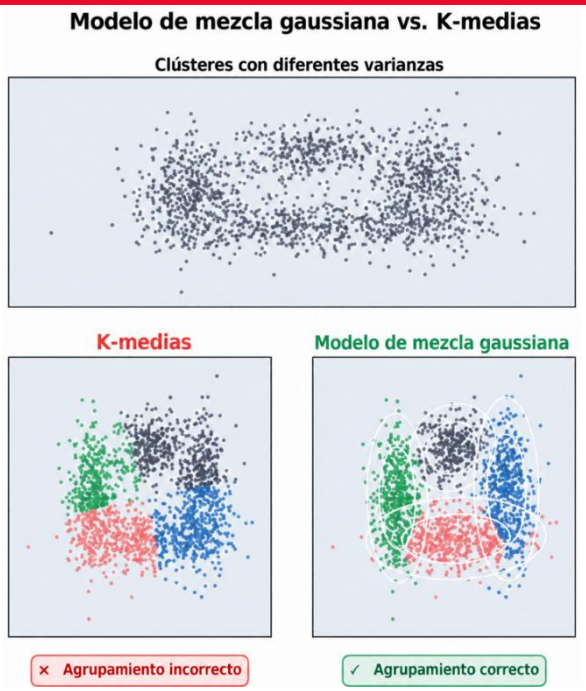


Ilustración 11. GMM vs. K medias.
Nota. Adaptado de Chawla (2024).

donde π_k es el peso o proporción del componente k, $\boldsymbol{\mu}_k$ es su vector de medias y $\boldsymbol{\Sigma}_k$ es su matriz de covarianza.

El objetivo del entrenamiento es encontrar los parámetros

$$\boldsymbol{\theta} = \{\pi_k, \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k\}_{k=1}^{K}$$

que hagan que los datos observados sean lo más probables posible. Por ello se maximiza la log-verosimilitud

$$\hat{\boldsymbol{\theta}} = \arg\max_{\boldsymbol{\theta}} \sum_{n=1}^{N} \log\{\mathrm{p}[\mathbf{x}_n \mid \boldsymbol{\theta}]\},$$

para entrenarlo se utiliza habitualmente el algoritmo esperanza maximización (*expectation-maximization*, EM).

4.3 Análisis de componentes principales

El análisis de componentes principales (*principal component analysis*, PCA) no busca grupos. Su objetivo es representar datos de dimensión alta mediante un número menor de variables. Se parte de $\mathbf{x} \in \mathbb{R}^D$ y se busca $\mathbf{z} \in \mathbb{R}^L$ tal que $L < D$. La representación $\mathbf{z}$ debe conservar la mayor cantidad posible de información relevante de $\mathbf{x}$.

Sea $\mathbf{X} = [\mathbf{x}_1, \ldots, \mathbf{x}_N] \in \mathbb{R}^{D \times N}$ la matriz de observaciones. Primero se centran los datos restando su media

$$\bar{x} = \frac{1}{N} \sum_{i=1}^{N} \mathbf{x}_i \,, \quad \mathbf{X}_0 = \mathbf{X} - \bar{x} \mathbf{1}_N^\top,$$

se buscan direcciones $\mathbf{p}_1, \mathbf{p}_2, \ldots, \mathbf{p}_D$ en las que los datos presentan la mayor varianza. Estas direcciones son los **eigenvectores** de la matriz $\mathbf{X}_0 \mathbf{X}_0^T$, ordenados según sus **eigenvalores** $\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_D$. Así

$$\frac{1}{N} \mathbf{X}_0 \mathbf{X}_0^\top = \mathbf{P} \Lambda \mathbf{P}^\top,$$

A las columnas de $\mathbf{P}$ se les denominan vectores de carga (*loading vectors)* y cada λ_i mide la varianza explicada por la componente correspondiente. La primera dirección se obtiene resolviendo el problema de máxima varianza:

$$\arg\max_{\mathbf{w}^\top \mathbf{w} = \mathbf{1}} \left\{ \mathbf{w}^\top \left(\frac{1}{N} \mathbf{X}_0 \mathbf{X}_0^\top \right) \mathbf{w} \right\}.$$

Para reducir de D a $L < D$ dimensiones, se conservan únicamente los primeros L vectores de carga (*loading vectors)*:

$$\mathbf{P}_L = [\mathbf{p}_1, \ldots, \mathbf{p}_L],$$

los datos reducidos son entonces

$$\mathbf{Z}_L = \mathbf{P}_L^\top \mathbf{X}_0.$$

El PCA también puede definirse mediante el problema de mínimo error de reconstrucción:

$$\min_{\mathbf{W}^\top \mathbf{W}=\mathbf{I}} \|\mathbf{X}_0 - \mathbf{W}\mathbf{W}^\top \mathbf{X}_0\|_F^2,$$

donde $\mathbf{W} \in \mathbb{R}^{D\times L}$.

4.4 Modelo oculto de Markov

Un modelo oculto de Markov (*hidden Markov model*, HMM) es un modelo para datos secuenciales en el que, en cada tiempo t, existe una variable oculta

$$z_t \in \{1, \dots, K\},$$

y una observación $\mathbf{y}_t$. Los estados z_t no se observan directamente. Un HMM se presenta como un modelo de espacio de estados con variables ocultas discretas. La distribución conjunta del modelo es

$$p(\mathbf{y}_{1:T}, \mathbf{z}_{1:T}) = p(z_1) \prod_{t=2}^{T} p\,(z_t \mid z_{t-1}) \prod_{t=1}^{T} p\,(\mathbf{y}_t \mid z_t).$$

El HMM supone que el estado actual depende sólo del estado inmediatamente anterior

$$p(z_t \mid z_{1:t-1}) = p(z_t \mid z_{t-1}),$$

también supone que, condicionado al estado actual, la observación no depende de los demás estados ni de observaciones pasadas

$$p(\mathbf{y}_t \mid \mathbf{z}_{1:T}, \mathbf{y}_{1:t-1}) = p(\mathbf{y}_t \mid z_t),$$

Las probabilidades de transición entre estados se representan mediante la matriz $\boldsymbol{A}$, cuyos elementos se definen como

$$A_{jk} = p(z_t = k | z_{t-1} = j),$$

donde A_{jk} representa la probabilidad de pasar del estado j al estado k. Esquemáticamente, $z_{t-1} \to z_t \to z_{t+1}$, y cada estado produce una observación $z_t \to \mathbf{y}_t$. El modelo de observación $p(\mathbf{y}_t \mid z_t)$ depende del tipo de dato observado. Algunos se presentan a continuación.

Tipo de observación	Modelo	Expresión
Categórica	Distribución categórica	$p(y_t = k \mid z_t = j) = B_{jk}$
Varias categóricas	Categóricas independientes	$p(\mathbf{y}_t \mid z_t = j) = \prod_{d=1}^{D} \mathrm{Cat}\left(y_{td} \mid \boldsymbol{\gamma}_{d,j,:}\right)$
Binaria	Bernoulli	$p(y_t \mid z_t = j) = \mathrm{Ber}\left(y_t \mid \theta_j\right)$
Continua	Gaussiana	$p(\mathbf{y}_t \mid z_t = j) = \mathcal{N}\left(\mathbf{y}_t \mid \boldsymbol{\mu}_j, \boldsymbol{\Sigma}_j\right)$
Continua autorregresiva de orden L	AR-HMM de orden L	$p(\mathbf{y}_t \mid \mathbf{y}_{t-L:t-1}, z_t = j) = \mathcal{N}\left(\mathbf{y}_t \mid \sum_{\ell=1}^{L} \mathbf{W}_{j,\ell}\, \mathbf{y}_{t-\ell} + \boldsymbol{\mu}_j, \boldsymbol{\Sigma}_j\right)$

Tabla 7. Modelos de observación para un HMM.

Nótese que el modelo AR-HMM constituye una extensión del HMM estándar y permite que $\mathbf{y}_t$ dependa de observaciones pasadas.

Los parámetros de un modelo oculto de Markov pueden estimarse mediante el algoritmo de Baum-Welch, mientras que el algoritmo de Viterbi permite obtener la secuencia conjunta de estados ocultos más probable condicionada a la secuencia de observaciones.

5. APRENDIZAJE POR REFUERZO

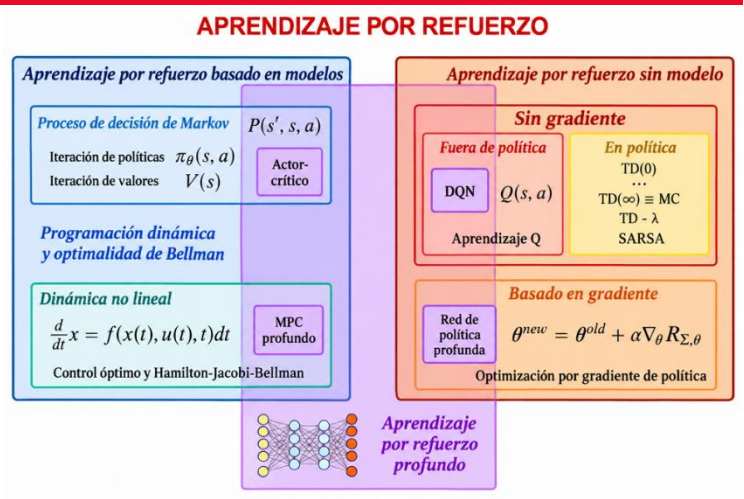


Ilustración 12. Modelos de aprendizaje por refuerzo.
Nota. Adaptado de Murphy (2023).

El aprendizaje por refuerzo (*reinforcement learning,* RL) es un paradigma de aprendizaje en el cual un agente interactúa de manera secuencial con un entorno inicialmente desconocido. En cada instante t, el agente observa un estado s_t, selecciona una acción a_t y recibe una recompensa r_t. Esta interacción genera una trayectoria

$$\tau = (s_0, a_0, r_0, s_1, a_1, r_1, \dots, s_T).$$

El objetivo del agente es aprender una política $\pi(a \mid s)$, que determine qué acción debe ejecutarse en cada estado, de manera que se maximice la recompensa acumulada. El retorno descontado se define como (Murphy K. P., 2023)

$$G_0 = \sum_{t=0}^{T-1} \gamma^t r_t, \quad \gamma \in [0,1],$$

donde γ es el factor de descuento. Este parámetro controla la importancia relativa de las recompensas futuras y permite definir recursivamente

$$G_t = r_t + \gamma G_{t+1}.$$

En consecuencia, el problema fundamental consiste en encontrar una política óptima que maximice la recompensa esperada

$$\pi^* = \arg\max_\pi \{\mathbb{E}_\pi[G_0]\}.$$

La formulación del aprendizaje por refuerzo suele basarse en un **proceso de decisión de Markov** (*Markov decision process*, MDP), compuesto por un conjunto de estados, acciones, probabilidades de transición y recompensas. El agente selecciona acciones mediante una política π, con el objetivo de maximizar la recompensa acumulada a lo largo del tiempo.

La calidad de una política puede evaluarse mediante la función de valor de estado

$$V^\pi(s) = \mathbb{E}_\pi[G_0 \mid s_0 = s],$$

y la función de valor de acción

$$Q^\pi(s, a) = \mathbb{E}_\pi[G_0 \mid s_0 = s, a_0 = a].$$

Cuando el funcionamiento del entorno es conocido, pueden emplearse métodos de planificación; cuando es desconocido, el agente aprende mediante la interacción con el entorno. En este último caso, los métodos pueden ser sin modelo, si aprenden directamente una política o función de valor, o basados en modelo, si primero estiman la dinámica del entorno.

Entre los métodos sin modelo destacan Monte Carlo y las diferencias temporales (*temporal-difference learning*), que actualizan las estimaciones a partir de la recompensa observada y del valor estimado del estado siguiente.

Los métodos de aprendizaje por refuerzo se clasifican según qué objeto aprende el agente: una función de valor, una política o un modelo del entorno.

Tabla 8. Principales familias de métodos de aprendizaje por refuerzo.

Familia	Qué aprende	Ejemplos
Métodos basados en valor	Una función de valor, principalmente *Q(s,a)*	SARSA, Q-learning, DQN
Métodos basados en política	Directamente la política *π(a|s)*	REINFORCE, TRPO, PPO
Actor-Crítico	Una política y una función de valor	A2C, A3C, DDPG, SAC
Métodos basados en modelo	El modelo de transición y recompensa del entorno	Model-based RL, MPC

5.1 Aprendizaje Q

El aprendizaje Q (*Q-learning*) es un método de aprendizaje por refuerzo basado en valor y fuera de política (*off-policy*). Su objetivo es aprender directamente la función de valor óptima de acción $Q^*(s,a)$ que representa el retorno esperado al ejecutar la acción a en el estado s y continuar posteriormente de forma óptima (Watkins & Dayan, 1992). A partir de una transición (s,a,r,s'), el aprendizaje Q actualiza la función de valor mediante

$$Q(s,a) \leftarrow Q(s,a) + \eta\left[r + \gamma \max_b\{Q(s',b)\} - Q(s,a)\right],$$

donde $\eta > 0$ es la tasa de aprendizaje y $\gamma \in [0,1]$ el factor de descuento. El término

$$\delta = r + \gamma \max_b\{Q(s',b)\} - Q(s,a),$$

es el error de diferencia temporal. La presencia del máximo sobre las acciones futuras permite aprender Q^* aun cuando las acciones observadas hayan sido generadas por una política distinta de la política óptima; por ello Q-learning es un método fuera de política.

La política puede obtenerse seleccionando la acción con mayor valor estimado

$$\pi(s) = \arg\max_a\{Q(s,a)\}.$$

5.2 REINFORCE

Es un método de aprendizaje por refuerzo basado en política. En lugar de aprender primero una función $Q(s,a)$, parametriza directamente la política como $\pi_{\boldsymbol{\theta}}(a \mid s)$ y busca maximizar el retorno esperado $J(\pi_{\boldsymbol{\theta}})$ (Williams, 1992). El algoritmo utiliza ascenso estocástico por gradiente. Para una trayectoria $\tau = (s_0, a_0, r_0, \dots, s_T)$, el gradiente puede aproximarse mediante (Murphy K. P., 2023)

$$\nabla_{\boldsymbol{\theta}} J(\pi_{\boldsymbol{\theta}}) \approx \sum_{t=0}^{T-1} \gamma^t\, G_t \nabla_{\boldsymbol{\theta}} \log\{\pi_{\boldsymbol{\theta}}(a_t \mid s_t)\},$$

para reducir la varianza puede introducirse una función $b(s_t)$

$$\boldsymbol{\theta} \leftarrow \boldsymbol{\theta} + \eta \sum_{t=0}^{T-1} \gamma^t \left(G_t - b(s_t)\right) \nabla_{\boldsymbol{\theta}} \log\{\pi_{\boldsymbol{\theta}}(a_t \mid s_t)\},$$

si $G_t - b(s_t) > 0$, aumenta la probabilidad de las acciones tomadas; si es negativo, dicha probabilidad disminuye.

6. APLICACIONES PRÁCTICAS

Las familias de modelos descritas en las secciones 1 a 5 tienen aplicación en prácticamente todos los tipos de riesgo que gestiona una institución financiera. Esta sección organiza dichas aplicaciones por tipo de riesgo (crédito, contraparte, mercado y balance) y presenta los principales problemas en los que estas técnicas pueden emplearse, los algoritmos utilizados y algunos antecedentes o resultados reportados en la literatura.

6.1 Riesgo de crédito

El riesgo de crédito es una de las áreas con mayor aplicación del aprendizaje supervisado. Técnicas como la regresión logística regularizada, los árboles de decisión, los bosques aleatorios, la potenciación por gradiente y las redes neuronales se utilizan en modelos de *scoring* y en la estimación de los componentes de la pérdida esperada: probabilidad de incumplimiento (PD), severidad de la pérdida (LGD) y exposición al momento del incumplimiento (EAD).

El uso de estas técnicas debe distinguir entre modelos empleados para gestión, alerta temprana o como modelos retadores, y aquellos cuyos resultados intervienen directamente en estimaciones contables, regulatorias o de capital, para los cuales los requisitos de gobierno, validación, estabilidad y explicabilidad pueden ser más exigentes.

Frente a los modelos tradicionales, los algoritmos de ensamble permiten identificar relaciones no lineales entre las variables crediticias y pueden mejorar la capacidad predictiva. Sin embargo, sus resultados son menos fáciles de interpretar, por lo que requieren técnicas complementarias de explicabilidad.

Por su parte, el aprendizaje no supervisado permite detectar anomalías y segmentar clientes con perfiles de riesgo similares. Estas técnicas pueden apoyar la identificación temprana de fraude y deterioro crediticio, así como el cumplimiento de IFRS 9. También pueden utilizarse para asignar modelos de scoring diferenciados a cada segmento.

Finalmente, para clientes con historial crediticio limitado, los datos alternativos, como patrones de gasto digital, uso de aplicaciones móviles o comportamiento de búsqueda, pueden complementar la información tradicional. Su utilización debe acompañarse de criterios adecuados de calidad y protección de datos.

6.2 Riesgo de contraparte

En riesgo de contraparte, el aprendizaje automático puede apoyar la estimación de la exposición potencial futura (PFE) y de la exposición esperada (EE), incorporando de manera flexible variables de mercado. Uno de los principales retos es la capacidad de cómputo requerida para calcular los perfiles de exposición EE y PFE, y en consecuencia el ajuste de valuación por riesgo de crédito (CVA) y sus sensibilidades. Cuando la valoración futura de un instrumento no dispone de una expresión cerrada, puede requerirse una simulación adicional en cada escenario, dando lugar a simulaciones de Monte Carlo anidadas. Como antecedente clásico, el método Longstaff y Schwartz emplea regresión dentro de un esquema de Monte Carlo para aproximar valores de continuación en opciones americanas, este enfoque puede extenderse mediante técnicas de aprendizaje automático.

Una alternativa consiste en utilizar modelos auxiliares que aproximen los resultados del modelo completo. Mediante técnicas como redes neuronales, *kriging*[2] o árboles de regresión, estos modelos se entrenan con una muestra representativa de escenarios y posteriormente aproximan los resultados para el portafolio completo. Su utilización puede reducir significativamente el tiempo de cómputo, siempre que el error de aproximación se cuantifique y permanezca dentro de tolerancias previamente definidas, incluyendo escenarios adversos y regiones del espacio de entrada con menor representación en la muestra de entrenamiento.

El aprendizaje automático también puede facilitar la identificación del riesgo de correlación adversa (*wrong-way risk*), que se presenta cuando la exposición aumenta al mismo tiempo que se deteriora la calidad crediticia de la contraparte. Su detección puede contribuir a definir medidas de mitigación, como ajustes a las líneas de crédito o requerimientos adicionales de colateral.

La identificación de dependencias estadísticas no implica por sí sola la existencia de una relación causal ni sustituye la especificación económica del mecanismo de WWR; por ello, sus resultados deben complementarse con análisis de escenarios, pruebas de estrés y revisión experta.

Finalmente, técnicas relacionadas con el aprendizaje por refuerzo y el control estocástico mediante redes neuronales pueden emplearse en la cobertura dinámica del CVA. En particular, enfoques de *Deep Hedging* parametrizan las estrategias de cobertura mediante redes neuronales y optimizan directamente una función de riesgo o pérdida sobre escenarios de mercado simulados, considerando fricciones como costos de transacción y restricciones de liquidez. La optimización de los parámetros puede realizarse mediante métodos de gradiente. Estos enfoques permiten ajustar dinámicamente las posiciones de cobertura conforme evolucionan la exposición, los factores de mercado y la calidad crediticia de las contrapartes, ofreciendo una alternativa flexible a los esquemas tradicionales de cobertura.

6.3 Riesgo de mercado

En riesgo de mercado, el aprendizaje automático se utiliza principalmente para pronosticar la volatilidad e identificar regímenes de riesgo mercado. Los modelos híbridos combinan métodos tradicionales, como GARCH, con redes neuronales o máquinas de vectores de soporte (SVM), con el objetivo de mejorar la estimación del valor en riesgo (VaR) condicional. En este contexto, cuando el modelo se utiliza para estimar VaR o medidas de cola, la evaluación debe complementarse con pruebas de cobertura y comportamiento de excepciones, además de las métricas predictivas generales.

La evidencia muestra que estos modelos pueden ofrecer mayor precisión y robustez, aunque su desempeño depende de las condiciones de mercado. Diversos estudios encuentran mejoras en determinados mercados y periodos mediante modelos SVM, LSTM o híbridos; sin embargo, su superioridad no es uniforme entre activos, horizontes ni regímenes. Durante episodios de estrés, modelos parsimoniosos como EWMA pueden mantener un desempeño competitivo, por lo que la selección debe sustentarse en evidencia fuera de muestra y pruebas específicas para distintos regímenes de mercado.

[2] El *kriging* es un método de interpolación estadística que, en el contexto de modelos sustitutos, permite aproximar la respuesta de un modelo computacional complejo a partir de un conjunto limitado de evaluaciones, utilizando la correlación entre puntos del espacio de entrada.

El aprendizaje automático también se aplica a la gestión del riesgo de modelo y la ejecución óptima de órdenes. Técnicas como el agrupamiento permiten identificar relaciones entre activos y diseñar estrategias para reducir los costos asociados con operaciones de gran volumen. Por su parte, el aprendizaje Q puede ajustar la ejecución de órdenes conforme cambian las condiciones del mercado, aunque sus resultados pueden presentar una mayor variabilidad.

Finalmente, los métodos de aprendizaje automático y aproximación estadística ofrecen herramientas para la valoración de instrumentos financieros cuyos pagos dependen de la trayectoria de los factores de mercado (*path-dependent derivatives*). Entre estos enfoques destaca la regresión por mínimos cuadrados sobre simulaciones de Monte Carlo, utilizada para aproximar los valores de continuación y determinar estrategias óptimas de ejercicio en opciones americanas. Por otra parte, el aprendizaje por refuerzo se ha aplicado a problemas de optimización secuencial, particularmente en la ejecución óptima de operaciones financieras. En este ámbito, se han desarrollado modelos de optimización que consideran los costos de transacción y el riesgo asociado a la volatilidad, así como extensiones basadas en *Q-learning* que permiten adaptar las estrategias de ejecución a las condiciones del mercado.

6.4 Riesgo de balance

El riesgo de balance constituye un ámbito especialmente relevante para la aplicación de técnicas de aprendizaje automático, debido a la presencia de opciones conductuales, relaciones no lineales y heterogeneidad entre clientes y productos. Estas técnicas pueden complementar los modelos tradicionales de gestión de activos y pasivos mediante la estimación de comportamientos que afectan la generación y temporalidad de los flujos de efectivo, el margen financiero y el valor económico.

Depósitos a la vista. En productos sin vencimiento contractual, los modelos de aprendizaje automático pueden utilizarse para analizar la estabilidad de los saldos, identificar patrones de permanencia y salida de depósitos, estimar la sensibilidad de las tasas pagadas frente a cambios en las tasas de referencia y segmentar clientes de acuerdo con su comportamiento. Modelos como árboles de decisión, bosques aleatorios y métodos de *gradient boosting* pueden capturar relaciones no lineales e interacciones entre variables, aunque su desempeño debe contrastarse frente a modelos de referencia y evaluarse bajo diferentes regímenes de tasas y condiciones de liquidez.

Prepago y ejercicio de opciones conductuales. En activos sujetos a prepago y otros productos con opcionalidad implícita, el aprendizaje automático puede utilizarse para estimar probabilidades de ejercicio condicionadas a características contractuales, comportamiento histórico del cliente, incentivos económicos y variables de mercado. Estas estimaciones pueden incorporarse posteriormente en la proyección de flujos de efectivo y en las métricas de sensibilidad del balance.

Riesgo de tasa de interés del libro bancario. En IRRBB, el aprendizaje automático puede complementar el proceso de medición mediante una estimación más granular de supuestos conductuales, incluyendo estabilidad y repricing de depósitos, prepagos y otras opciones de clientes. Los resultados de estos modelos no sustituyen las métricas de riesgo de tasa, sino que constituyen insumos para la generación de flujos de efectivo utilizados posteriormente en la estimación del valor económico y del margen financiero bajo los escenarios correspondientes.

La utilización de algoritmos más complejos sólo se justifica cuando la mejora en capacidad predictiva o segmentación es material frente a modelos más parsimoniosos y puede reconciliarse con relaciones económicas plausibles bajo distintos escenarios de tasas.

Liquidez y estructura de fondeo. Las técnicas de aprendizaje automático también pueden emplearse para modelar entradas y salidas de efectivo, analizar la estabilidad de las fuentes de fondeo e identificar cambios en los patrones de comportamiento de clientes y contrapartes. Los modelos de series temporales, aprendizaje supervisado y métodos de detección de anomalías pueden complementar los sistemas de alerta temprana y el análisis prospectivo de liquidez.

Dado que los comportamientos observados durante condiciones normales pueden modificarse de manera significativa bajo estrés, el desempeño predictivo fuera de muestra no es suficiente para validar estas aplicaciones. Los modelos deben someterse a análisis de estabilidad, sensibilidad y escenarios adversos, y complementarse con juicio experto y modelos de referencia. La consistencia entre los supuestos conductuales empleados en las distintas métricas de riesgo de balance resulta asimismo fundamental para evitar señales contradictorias entre la gestión del margen, el valor económico y la liquidez.

7. LIMITACIONES

El aprendizaje automático ofrece beneficios para la gestión de riesgos, pero también presenta limitaciones que deben considerarse desde el diseño y durante la operación de los modelos.

a) Generalización y datos

- Los modelos complejos pueden presentar sobreajuste, es decir, ajustarse excesivamente a la información histórica, pero perder capacidad predictiva fuera de muestra.
- Pueden generarse resultados artificialmente favorables por fugas de información o por no respetar el orden cronológico de los datos.
- Otros riesgos relevantes son la falta de calidad y representatividad de la información, los cambios estructurales en el mercado y la escasa disponibilidad de episodios de estrés.
- Finalmente, un buen desempeño a nivel individual no garantiza resultados adecuados a nivel portafolio.

b) Interpretabilidad y supervisión

- Algunos modelos, como las redes neuronales profundas y ciertos ensambles, funcionan como cajas negras, lo que dificulta explicar sus resultados y realizar su validación.
- Herramientas como LIME y SHAP facilitan la interpretación al estimar la influencia de cada variable en las predicciones. Sin embargo, no eliminan por completo esta limitación, ya que ofrecen explicaciones aproximadas de los resultados, sin revelar de manera integral el funcionamiento interno de los modelos complejos.
- También existe el riesgo de que los modelos reproduzcan sesgos presentes en los datos o utilicen variables relacionadas indirectamente con características protegidas. Por ello, su aplicación requiere criterios de equidad, supervisión humana y mecanismos claros de control.

c) Costo computacional y estabilidad

- La selección del modelo implica equilibrar precisión, complejidad, capacidad de cómputo e interpretabilidad. Arquitecturas como las redes neuronales recurrentes (RNN) y LSTM pueden requerir mayores volúmenes de información y tiempos de entrenamiento.
- La inicialización aleatoria puede generar variaciones entre ejecuciones, por lo que es necesario evaluar la estabilidad de los resultados.
- La robustez debe evaluarse también frente a distintas semillas aleatorias, muestras de entrenamiento, ventanas temporales, especificaciones e hiperparámetros, con objeto de identificar modelos cuyo desempeño dependa excesivamente de una configuración particular.

8. LÍNEAS EMERGENTES DE INVESTIGACIÓN

La investigación reciente busca atender varias de estas limitaciones. Entre las principales líneas se encuentran:

- **Aprendizaje federado:** permite entrenar modelos entre distintas instituciones sin intercambiar directamente los datos de los clientes.
- **Estimación de incertidumbre:** complementa las predicciones puntuales con medidas que reflejan su nivel de confianza.
- **Aprendizaje con datos escasos:** utiliza datos sintéticos o técnicas de aprendizaje con pocos ejemplos para atender historiales limitados.
- **Aprendizaje robusto y balanceado:** reduce la sensibilidad ante datos ruidosos, cambios estructurales y clases poco representadas.
- **Aprendizaje equitativo:** busca identificar y mitigar posibles sesgos o efectos discriminatorios en las decisiones del modelo.

Estas líneas pueden favorecer una aplicación más segura, explicable y adecuada del aprendizaje automático en la gestión de riesgos.

9. RETOS EN LA IMPLEMENTACIÓN EN LA BANCA MEXICANA

La adopción del aprendizaje automático en la banca mexicana enfrenta retos que van más allá de la capacidad predictiva de los modelos. Los principales se relacionan con la regulación, la gobernanza, la infraestructura de datos y la disponibilidad de talento especializado.

9.1 Entorno regulatorio

En México, la implementación de modelos de aprendizaje automático en la gestión del riesgo debe ajustarse a las disposiciones de la Comisión Nacional Bancaria y de Valores (CNBV). Asimismo, la Norma Internacional de Información Financiera 9 (IFRS 9) constituye un referente para la estimación de pérdidas crediticias esperadas mediante información histórica, condiciones actuales y proyecciones económicas. El principal reto consiste en garantizar que los modelos sean interpretables, estén debidamente documentados y validados, y cumplan con los requisitos regulatorios aplicables.

9.2 Gobernanza y validación interna

Las áreas de validación necesitan capacidades técnicas para evaluar modelos no paramétricos y algoritmos complejos. Esto requiere pruebas de robustez, estabilidad y desempeño fuera de muestra, así como herramientas de explicabilidad que permitan asegurar su trazabilidad y facilitar su revisión por los órganos de gobierno y las autoridades supervisoras. La validación de modelos de aprendizaje automático debe cubrir no sólo su desempeño predictivo, sino también la estabilidad frente a cambios en la muestra, hiperparámetros y condiciones de mercado; la reproducibilidad del proceso de entrenamiento; el linaje y calidad de los datos; la sensibilidad de las predicciones y explicaciones; y la existencia de límites de uso, criterios de recalibración o reentrenamiento y mecanismos de contingencia. Estos elementos deben aplicarse de manera proporcional a la materialidad y al uso del modelo.

9.3 Infraestructura de datos

La calidad, disponibilidad y profundidad histórica de los datos continúan siendo limitantes importantes. El uso de información no estructurada, como textos o redes sociales, requiere infraestructura adicional para su captura, procesamiento y monitoreo oportuno.

9.4 Talento especializado

La implementación de estas técnicas requiere perfiles que combinen conocimientos de ciencia de datos, regulación financiera y negocio bancario. La limitada disponibilidad de estos especialistas puede dificultar que los modelos evolucionen de proyectos piloto a soluciones integradas en la gestión cotidiana de riesgos.

10. CONCLUSIONES

El aprendizaje automático representa una extensión de las herramientas tradicionales de gestión de riesgos, no un sustituto. Sus algoritmos pueden mejorar la capacidad predictiva y aprovechar fuentes de información estructurada y no estructurada. Sin embargo, sus resultados dependen de la calidad de los datos, las condiciones del mercado y la adecuada selección y validación del modelo.

Su adopción responsable requiere un marco sólido de gobierno del riesgo de modelo, que incluya documentación de supuestos y limitaciones, comparación con metodologías tradicionales, pruebas de robustez, herramientas de explicabilidad y monitoreo continuo del desempeño y de la deriva.

Bajo este enfoque, el aprendizaje automático puede integrarse de manera sostenible a la gestión de riesgos, manteniendo un equilibrio entre innovación, precisión, interpretabilidad y cumplimiento regulatorio.

11. REFERENCIAS

Aggarwal, C. C. (2018). *Neural Networks and Deep Learning: A Textbook.* Cham, Switzerland: Springer International Publishing.

Almgren, R., & Chriss, N. (1999). *Optimal Execution of Portfolio Transactions.*

Basel Committee on Banking Supervision. (2016). *Interest Rate Risk in the Banking Book.* Bank for International Settlements.

Board of Governors of the Federal Reserve System. (2011). *Supervisory Guidance on Model Risk Management.* Board of Governors of the Federal Reserve Systems; Office of the Comptroller of the Currency.

Box, G. E., Jenkins, G. M., Reinsel, G. C., & Ljung, G. M. (2016). *Time Series Analysis: Forecasting and Control* (5 ed.). Hoboken, NJ: John Wiley & Sons.

Breiman, L. (2001). Random Forests. *Machine Learning, 45*, 5-32.

Buehler, H., Gonon, L., Teichmann, J., & Wood, B. (2019). Deep hedging. *Quantitative Finance*, 1271-1291.

Chawla, A. (Marzo de 2024). *KMeans vs Gaussian Mixture Models*. Obtenido de Daily Dose of Data Science: https://blog.dailydoseofds.com/p/kmeans-vs-gaussian-mixture-models

Chen, T., & Guestrin, C. (2016). XGBoost: A Scalable Tree Boosting System. *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (págs. 785-794). Association for Computing Machinery.

Cortes, C., & Vapnik, V. (1995). Support-Vector Networks. *Machine Learning, 20*, 273-297.

Dorogush, A. V., Ershov, V., & Gulin, A. (2018). CatBoost: Gradient Boosting with Categorical Features Support. *arXiv preprint arXiv:1810.11363*.

Friedman, J. H. (2001). Greedy Function Approximation: A Gradient Boosting Machine. *The Annals of Statistics, 29*, 1189-1232.

Hendricks, D., & Wilcox, D. (2014). A Reinforcement Learning Extension to the Almgren-Chriss Framework for Optimal Trade Execution. *2014 IEEE Conference on Computational Intelligence for Financial Engineering & Economics (CIFEr).* IEEE.

Hochreiter, S., & Schmidhuber, J. (1997). Long Short-Term Memory. *Neural Computation, 9*, 1735-1780.

Hornik, K., Stinchcombe, M., & White, H. (1989). Multilayer Feedforward Networks Are Universal Approximators. *Neural Networks, 2*, 359–366.

Ke, G., Meng, Q., Finley, T., Wang, T., Chen, W., Ma, W., . . . Liu, T.-Y. (2017). LightGBM: A Highly Efficient Gradient Boosting Decision Tree. *Advances in Neural Information Processing Systems. 30*, págs. 3146-3154. Curran Associates, Inc.

Longstaff, F. A., & Schwartz, E. S. (2001). Valuing American Options by Simulation: A simple Least-Squares Approach. *The Review of Financial Studies*, 113-147.

Lundberg, S. M., & Lee, S.-I. (2017). A Unified Approach to Interpreting Model Predictions. *Advances in Neural Information Processing Systems 30.* Long Beach, California: Curran Associates, Inc.

MathWorks. (s.f.). *Introduction to Support Vector Machines (SVM)*. Obtenido de MathWorks: https://la.mathworks.com/discovery/support-vector-machine.html

Murphy, K. P. (2022). *Probabilistic Machine Learning: An Introduction.* Cambridge, Massachusetts: MIT Press.

Murphy, K. P. (2023). *Probabilistic Machine Learning: Advanced Topics.* Cambridge, Massachusetts: The MIT Press.

Nami, Y. (6 de Noviembre de 2021). *Why Does Increasing K Decrease Variance in KNN?* Obtenido de Towards Data Science: https://towardsdatascience.com/why-does-increasing-k-decrease-variance-in-knn-9ed6de2f5061/

Pérez-Hernández, F., Arévalo-de-Pablos, A., & Camacho-Miñano, M.-d.-M. (2024). A Hybrid Model Integrating Artificial Neural Network with Multiple GARCH-Type Models and EWMA for Performing the Optimal Volatility Forecasting of Market Risk Factors. *Expert Systems With Applications*.

Prokhorenkova, L., Gusev, G., Vorobev, A., Dorogush, A. V., & Gulin, A. (2018). CatBoost: Unbiased Boosting with Categorical Features. *Advances in Neural Information Processing Systems. 31.* Curran Associates, Inc.

Ribeiro, M. T., Singh, S., & Guestrin, C. (2016). "Why Should I Trust You?": Explaining the Predictions of Any Classifier. *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining.* San Francisco, California: Association for Computing Machinery (ACM).

Salazar, G., Izquierdo, A. M., Grajales, M., & Echazarreta, S. (2026). *Social Media and Liquidity Risk Management: Applying a "Sentiment Analysis" Early Warning Indicator.* BTRM Ltd.

Son, Y., Byun, H., & Lee, J. (2016). Nonparametric Machine Learning Models for Predicting the Credit Default Swaps: An Empirical Study. *Expert Systems with Applications, 58*.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., . . . Polosukhin, I. (2017). Attention is All You Need. *Advances in Neural Information Processing Systems. 30*, págs. 5998-6008. Curran Associates, Inc.

Watkins, C. J., & Dayan, P. (1992). Q-learning. *Machine Learning, 8*, 279-292.

Williams, R. (1992). Simple Statistical Gradient-Following Algorithms for Connectionist Reinforcement Learning. *Machine Learning, 8*, 229-256.

Wilmott, P. (2019). *Machine Learning: An Applied Mathematics Introduction.* Panda Ohana Publishing.