HENRY

Data Science

M4L4 | Aprendizaje supervisado III: Modelos clasificación y métricas de evaluación

→ soyhenry.com

Objetivos

- Comparar los modelos clásicos de clasificación, incluyendo Árbol de Decisión, K-Vecinos, SVM y Bayesiano, evaluando sus principios y aplicaciones en distintos escenarios.
- Analizar métricas de evaluación como matriz de confusión, sensibilidad, precisión y exactitud, interpretando su impacto en la calidad de los modelos.
- Evaluar la curva ROC y el f-score para determinar el rendimiento y la capacidad predictiva de los modelos de clasificación

Agenda

COMENCEMOS →

#TEMAS

.01 K-Vecinos

.02 Árboles de decisión

.03 SVM y clasificación bayesiana

.04 Métricas de evaluación para clasificación

.05 Homework

<-->

¿Qué vimos en la lecture?

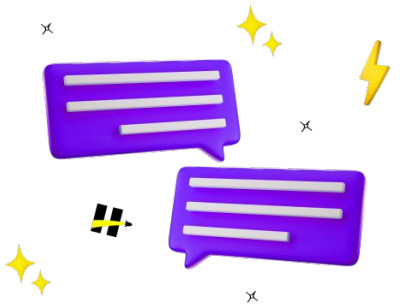
<01>

K-Vecinos

Introducción al modelo K-Vecinos

- El modelo de **K Vecinos más Cercanos (KNN)** es uno de los métodos de clasificación más **intuitivos y simples**.
- A diferencia de la regresión logística, **no aprende una fórmula matemática**: memoriza los datos de entrenamiento.
- Clasifica según la **similitud** entre observaciones previas y un nuevo caso.
- Se basa en la **distancia** entre puntos (por ejemplo, **Euclidiana**) para determinar los “vecinos” más parecidos.

Principio de funcionamiento



- Para una nueva observación (ej. un día nuevo en CityScoot):
  1️⃣ Calcula qué tan parecido es a los días anteriores.
  2️⃣ Encuentra los **k más cercanos**.
  3️⃣ Mira la clase de esos vecinos (alta o baja demanda).
  4️⃣ **Vota por mayoría** → si la mayoría tuvo alta demanda → predice alta demanda.

- KNN **no generaliza**: decide caso a caso, basándose en el contexto local.

El parámetro *k*

- **k pequeño (1 o 3):** muy sensible al ruido → puede sobreajustar.
- **k grande (15 o 25):** más estable → pierde detalle o mezcla patrones.
- La elección óptima se hace con **validación cruzada**, probando varios valores.
- Es esencial **escalar las variables**: si “marketing_spend” está en miles y “temp_c” en decenas, una dominará la distancia.
- Variables categóricas → usar **0/1** o **one-hot encoding**.

Limitaciones del KNN

- **Solo usa variables numéricas →** las categóricas deben codificarse.
- **Requiere normalización →** sin ella, las variables con mayor escala dominan.
- **Sensible al ruido y outliers →** un caso atípico puede alterar la clasificación.
- **Maldición de la dimensionalidad:** con muchas variables, las distancias pierden significado y el modelo se degrada.

Hiperparámetros clave

- **k→** número de vecinos a considerar (bajo → ruido; alto → pérdida de detalle).
- **metric →** cómo se mide la distancia (Euclidiana, Manhattan, etc.).
- **weights →** define si todos los vecinos pesan igual o los más cercanos influyen más.

Ajustar estos valores es esencial para mejorar el rendimiento del modelo.

<02>

Árboles de decisión

Qué son los árboles de decisión

- Son modelos **altamente interpretables** y **visuales**.
- Construyen **reglas jerárquicas** que dividen los datos hasta llegar a una predicción.
- A diferencia de KNN o regresión logística, **crean un "diagrama de flujo" lógico** con condiciones sucesivas.

Principio de funcionamiento

- El árbol plantea preguntas secuenciales:
  "¿Es fin de semana?", "¿Lluvia > 3 mm?", "¿Marketing > 900 USD?"

- Cada división genera nodos más **puros** (más homogéneos).

La predicción final se encuentra en las **hojas**.

Criterios de división

- **Pureza del nodo:** mide qué tan homogéneo es un grupo.
- **Índice Gini:**
  - Gini = 0 → nodo completamente puro.
  - Gini = 0.5 → mezcla perfecta (50/50).
- **Ganancia de información:** mejora lograda al dividir un nodo.
- El árbol elige siempre la división que **maximiza la ganancia** y reduce la impureza.

Ventajas y limitaciones

Ventajas

- Muy interpretables.
- No requieren escalado.
- Soportan valores faltantes.
- Capturan relaciones no lineales.

Limitaciones

- Alta tendencia al overfitting si no se limita su profundidad.
- Sensibles a pequeños cambios en los datos.
- Pueden tomar decisiones arbitrarias si hay variables correlacionadas.

Hiperparámetros principales

- **max_depth →** profundidad máxima → controla complejidad.
- **min_samples_leaf →** mínimo de ejemplos por hoja → previene sobreajuste.
- **max_features →** máx. de variables por división → añade aleatoriedad y reduce correlación.

Ajustar estos valores mejora la generalización del modelo.

<03>

SVM y clasificación bayesiana

Introducción a SVM

- Las Máquinas de Vectores de Soporte (SVM) buscan la **mejor frontera** entre clases.
- No cualquier frontera, sino la que **maximiza el margen** entre los puntos más cercanos de cada clase.
- Esto le da **robustez** y evita sobreajuste.

El truco del kernel

- Si las clases no se pueden separar con una línea, SVM usa el **kernel trick**.
- Transforma los datos a un espacio de mayor dimensión para hacerlos separables.
- Ejemplo metafórico: puntos en un círculo 2D → al pasar a 3D, se pueden separar con un plano.

Cuándo usar SVM

1️⃣ Clases mezcladas o sin frontera clara.

2️⃣ Separación no lineal → se beneficia del kernel.

3️⃣ Necesidad de modelo **robusto al ruido y overfitting**.

- En CityScoot, SVM maneja mejor las zonas grises (intermedias de temperatura o marketing).

Clasificación bayesiana

- Basada en probabilidades: predice la clase con mayor probabilidad según las características.
- Ejemplo más común: **Naive Bayes**, que asume independencia entre variables.
- Aun si esa suposición es “ingenua”, funciona bien con muchas variables categóricas y pocos datos.
- Es rápido, simple y útil cuando la velocidad prima sobre la precisión.

<04>

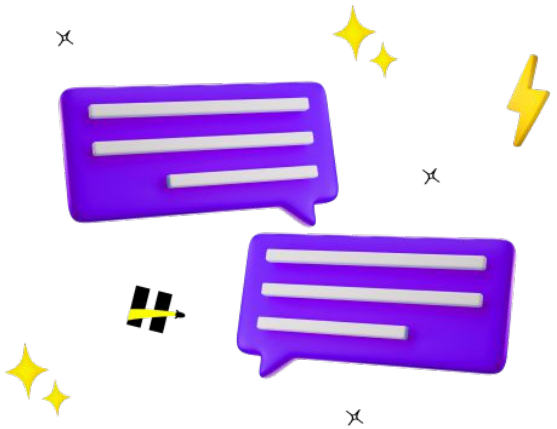
Métricas de evaluación para clasificación

Por qué necesitamos métricas más allá del accuracy

- El accuracy no basta cuando hay **clases desbalanceadas**.
- En CityScoot, si 90 % de los días son de baja demanda, un modelo que predice “baja” siempre tendrá 90 % de acierto pero 0 recall para alta demanda.
- Necesitamos entender **cómo se equivoca** el modelo y el impacto de cada error.

Matriz de confusión

- Resume los resultados en cuatro grupos:
    a. TP = verdaderos positivos.
    b. FP = falsos positivos.
    c. FN = falsos negativos.
    d. TN = verdaderos negativos.
- A partir de ella se calculan las principales métricas de rendimiento.



Métricas clave

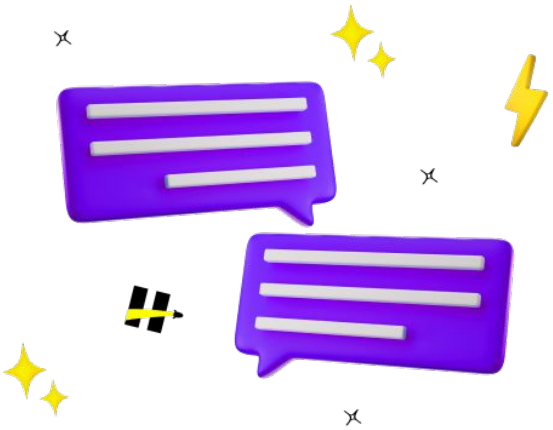
- **Precisión (Precision):** de los días predichos como alta demanda, ¿cuántos lo fueron realmente?
  → Alta precisión = pocos falsos positivos.

- **Recall (Sensibilidad):** de los días realmente de alta demanda, ¿cuántos detectó el modelo?
  → Alto recall = pocos falsos negativos.

- **F1-score:** media armónica entre precision y recall.
  → Equilibra ambas cuando hay desbalance.

Curva ROC y AUC

- La curva ROC muestra cómo varía recall vs falsos positivos según el umbral.
- El **AUC** (Área Bajo la Curva) resume su rendimiento en un número de 0 a 1.
- AUC≈1 → modelo excelente. AUC≈0.5 → como adivinar al azar.
- En CityScoot, sirve para comparar modelos (SVM, árbol, KNN) independientemente del umbral.



<DATA SCIENCE/>

Vayamos a la práctica

Homework

Consigna

1. Estás trabajando en el equipo de analítica de una fintech que busca optimizar su presupuesto de marketing. El objetivo es construir un modelo que **clasifique prospectos (leads)** en función de su probabilidad de conversión, para decidir **en qué contactos vale la pena invertir más recursos**.

2. Contás con un dataset histórico de usuarios que fueron impactados por campañas online. Algunas personas **compraron un producto financiero**, otras no. A partir de ese historial, deberás entrenar modelos que permitan predecir si un nuevo usuario **tiene alta o baja probabilidad de conversión**.

3. Este tipo de problema es típico de la industria Martech y es un caso real de aplicación de modelos de clasificación para **scoring de leads**.

Tareas a realizar

1. **Explorá el dataset**: revisá la distribución de las variables, su relación con la variable objetivo y posibles transformaciones necesarias (por ejemplo, codificación de variables categóricas o escalado).

2. **Entrená y compará al menos 3 modelos**:

    - K-Vecinos más cercanos (KNN)
    - Árbol de decisión
    - SVM (con kernel lineal o no lineal)
    - *(Opcional)* Agregá regresión logística.

Tareas a realizar

3. Evaluá el rendimiento con las métricas adecuadas:

- Accuracy
- Precisión
- Recall
- F1-score
- AUC y curva ROC

4. Seleccioná el mejor modelo según su rendimiento general y según el criterio de negocio: en este caso, es preferible **detectar bien los usuarios que sí convertirán**, incluso si eso implica asumir algunos falsos positivos.

5. Presentá tus conclusiones:

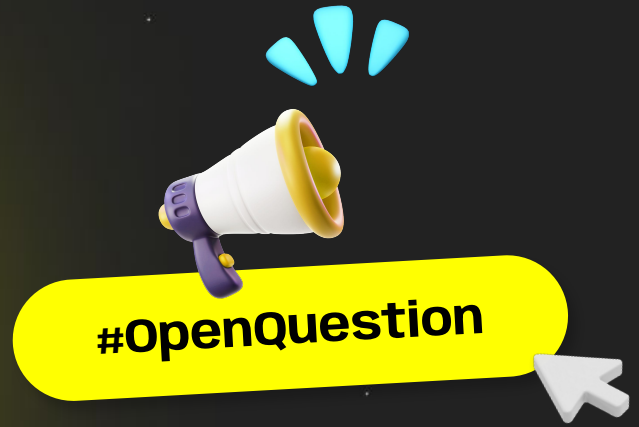
- ¿Qué modelo tuvo mejor desempeño?
- ¿Qué variable parece tener mayor peso en la clasificación?

¿Qué estrategia sugerirías al equipo de marketing con base en los resultados?

Tip final

No se espera que construyas el modelo perfecto, sino que puedas **comparar alternativas y justificar tu elección**. Este es el tipo de análisis que se espera de un/a Data Scientist en contextos reales de negocio.

HENRY



¿Preguntas?

HENRY

¡Muchas gracias!

→ soyhenry.com