Tarea 1 — Manejo de proyectos de ML: Conversión con descuentos y saturación

Equipo Docente - Deep Learning 2025-02

1) Enunciado de la Tarea 1

Modalidad de trabajo

Esta tarea es **grupal**. Se conformarán un total de 10 grupos: 6 grupos de 5 alumnos y 4 grupos de 4 alumnos.

Fechas Importantes

- Publicación: Miercoles 17 de septiembre de 2025.
- Fecha Limite entrega de grupos: Lunes 22 de septiembre. Posterior a eso se ordenarán de manera aleatoria.
- Entrega de data set a grupos: Lunes 22 de septiembre de 2025.
- Fecha Entrega: Domingo 5 de octubre de 2025

Contexto de Negocio

Una empresa de e-commerce evalúa una campaña de descuentos (0–30 %) para aumentar la probabilidad de compra. Debes apoyar una decisión operativa: "¿A quién ofrecer el descuento y con qué regla de decisión?". Trabajarás con un archivo CSV grupal (un registro por cliente con su descuento y si compró o no). La relación descuento → compra presenta rendimientos decrecientes (saturación).

Objetivo

Construir un mini-proyecto de ML *end-to-end* (sin MLP) que entregue una política operativa basada en probabilidades y umbrales, con control de capacidad y validación. Debes explicar y justificar cada decisión técnica, con foco en la decisión de negocio (E–T–P, generalización, regularización y métricas), consistente con lo visto en Bases de ML.

Variables Disponibles

Se ha generado un dataset sintético que contiene información de clientes, su respuesta ante descuentos, y algunas variables adicionales. El uso de estas variables es clave para la evaluación de su criterio.

- **Principales (mínimas necesarias):**
  - `discount` (numérica, 0–30): porcentaje de descuento ofrecido.
  - `segment` (categórica: A, B, C): segmento de clientes.
  - `age` (numérica): edad del cliente.
  - `tenure_months` (numérica): meses como cliente de la empresa.
  - `income_index` (numérica): proxy de nivel de ingresos (1 = bajo, 5 = alto).
  - `web_visits_30d` (numérica): número de visitas al sitio en los últimos 30 días.
  - `purchase` (binaria: 0/1): variable objetivo → si el cliente compró (1) o no (0).
- **Adicionales (opcionales, de uso crítico o estratégico):**
  - `id`: identificador único de cliente (no tiene valor predictivo directo).
  - `leak_after_offer`: variable de fuga que contiene información del futuro. Si se incluye en el modelo, puede inflar la *accuracy* artificialmente.
  - `unit_margin_if_buy`: margen económico esperado si el cliente compra (permite extender el análisis hacia métricas de negocio como *profit*).
  - `discount_bucket`: versión categórica de `discount` en intervalos.
- **Nota importante:** Recibirán un CSV grupal con el dataset. Cualquier variable cuyo nombre empiece con `leak_` es post-tratamiento y no debe usarse para entrenar (riesgo de *data leakage*).

Lo que debes desarrollar (paso a paso)

El trabajo debe seguir los siguientes puntos:

1. **E–T–P y framing (clasificación vs regresión)** Explica E (experiencia), T (tarea) y P (performance). Define tu *framing* principal: clasificación binaria de `purchase` con salida probabilística. Justifica si trabajas con tasas agregadas. Cierra con la decisión a soportar (política basada en umbral).
2. **Métricas y pérdida** Usa *log-loss* (entropía cruzada) como pérdida principal y reporta AUC y Brier. Explica por qué no usar MSE como objetivo principal para probabilidades. Relaciona con máxima verosimilitud.
3. **Diseño de validación y control de capacidad** Separa *train/valid/test* (70/15/15) o usa *k-fold* para elegir hiperparámetros y reentrena el modelo final con el mejor valor antes del *test*. Controla capacidad con L2 (`C`) y muestra curvas *train/valid* vs complejidad (o *learning curve*). Explica el *trade-off* sesgo–varianza.
4. **Preprocesamiento** Exploración inicial del dataset: revisar las variables, detectar posibles problemas (duplicaciones, variables irrelevantes, *leakage*). Preprocesamiento: codificación de variables, escalamiento si corresponde, manejo de *outliers*/nulos.
5. **Modelado predictivo** Entrenar al menos dos modelos de clasificación (ej. *logistic regression, árbol, random forest, XGBoost*, etc.) y comparar su desempeño inicial.
6. **Evaluación** Evalúa el desempeño de tus modelos usando métricas de clasificación: *accuracy, precision, recall, F1-score, AUC-ROC*.

7. **Discusión de resultados** Explica qué variables fueron relevantes para tus modelos. Justifica por qué ciertas columnas no debieron usarse. Propón *insights* accionables para la empresa basándote en los resultados obtenidos.

8. **Política operativa y sensibilidad** Entrega una regla clara: "contactar/ofrecer si $\hat{p} \geq t$". Muestra sensibilidad del resultado a t (ej., utilidad esperada por umbral) y discute implicancias.

9. **Riesgos y mitigación** Señala al menos tres: *leakage*, sesgo de muestreo por política de descuentos, *shift* temporal/segmento. Propón mitigaciones (auditoría de variables, validación por segmento o fuera de tiempo, *calibration*, A/B).

10. **Resultados y conclusiones** Resume hallazgos clave (saturación, desempeño en *test*, umbral recomendado) y cómo la empresa debe operar con tu modelo.

Lo que deben hacer [Opcional/Avanzado]

Se valorará positivamente el uso de las variables adicionales para los siguientes puntos:

- Mostrar cómo el uso indebido de `leak_after_offer` genera un modelo engañoso (*data leakage*).
- Calcular métricas de negocio como *expected profit* considerando `unit_margin_if_buy`.
- Comparar el uso de `discount` numérica vs. `discount_bucket`.

Formato y reproducibilidad (obligatorio)

- Incluye al inicio del *notebook* una celda con:

```
GLOBAL_SEED = <tu_entero>  # usa y declara una semilla
DATASET_ID = "<ID de tu archivo>"  # ej: '007' si tu archivo es T1_007_individual.c
```

- Fija la semilla para NumPy/sklearn y repórtala en tu *notebook*.
- Reporta versión de librerías (ej., `sklearn.__version__`) y *hash* SHA-256 del CSV.
- No regeneres datos: debes usar exactamente el archivo que te entregó ayudantía.
- Entrega resultados específicos de TU dataset (no "promedios de internet").
- (Esta exigencia de reproducibilidad y separación *train/test* está alineada con el enfoque del curso y la evaluación escrita 1.)

Entregables

- *Notebook* Colab (.ipynb) con código y explicaciones (en celdas Markdown) que siga el guion anterior.
- PDF exportado del *notebook*.
- JSON o tabla con tus métricas finales (*test*) y el umbral recomendado.
- README corto (3–5 líneas) con semilla, DATASET_ID, versiones y *hash* del archivo.

Nombres de archivo (obligatorio)

- `T1_<Grupo>_<NumeroGrupo>_<DATASET_ID>.ipynb`
- `T1_<Grupo>_<NumeroGrupo>_<DATASET_ID>.pdf`

Uso de herramientas de IA

Puedes usar GPT/Gemini u otras herramientas como apoyo. Debes entender y suscribir cada decisión y afirmación. No se aceptará "lo dijo la IA" como justificación. Tu nota dependerá principalmente de la calidad de tus explicaciones y de la consistencia técnica.

Restricciones

- No usar MLP ni arquitecturas profundas en esta tarea.
- No usar columnas `leak_*` para entrenar (anótalo explícitamente si las excluyes).

Criterios de corrección (resumen)

Framing y decisión de negocio (15 %) — Validación y detención (20 %) — Modelo base (15 %) — Pérdida y métricas (15 %) — Capacidad y sesgo–varianza (10 %) — Evidencia de generalización (10 %) — Riesgos/mitigación (10 %) — Claridad y reproducibilidad (5 %).
Bonus: Insights avanzados con variables adicionales (10 % extra)