AAI.EX.20240206

Ejercicios elaborados con fines educativos, inspirados en los contenidos evaluados en el exámen del 06/02/2024 (convocatoria Feb-2024) de Aprendizaje Automático I del MUICD de la UNED.

Este documento no es una copia ni una transcripción del examen oficial, sino una redacción propia de ejercicios conceptualmente equivalentes.

Test

Pregunta correcta: +0.5 puntos
Pregunta incorrecta: -0.1 puntos

AAI.EX.20240206.T.1

Enunciado AAI.EX.20240206.T.1
En un flujo típico de entrenamiento de modelos, ¿para qué se utiliza el conjunto de validación?

a) Para estimar el error de generalización final justo antes de desplegar en producción.

b) Para comparar candidatos y ajustar hiperparámetros sin tocar el test.

c) Para ambas cosas.

d) Ninguna de las anteriores.

Solución AAI.EX.20240206.T.1
Respuesta correcta: **b)**

Justificación: la validación se usa para **selección de modelo y ajuste de hiperparámetros**. La estimación "final" del error de generalización se reserva al conjunto de **test**.

AAI.EX.20240206.T.2

Enunciado AAI.EX.20240206.T.2
Entrenas una SVM con kernel RBF y observas *underfitting* incluso en entrenamiento. ¿Qué cambio suele ser el más razonable para aumentar la capacidad del modelo?

a) Aumentar γ y/o C.

b) Disminuir γ y/o C.

c) Aumentar γ y/o disminuir C.

d) Disminuir γ y/o aumentar C.

Solución AAI.EX.20240206.T.2
Respuesta correcta: **a)**

Justificación: subir γ hace el modelo más flexible (fronteras más "curvadas") y subir C reduce regularización (penaliza más los errores), ambos ayudan a combatir el subajuste.

AAI.EX.20240206.T.3

Enunciado AAI.EX.20240206.T.3
Al entrenar árboles de decisión, ¿es importante escalar/normalizar las características de entrada?

a) Sí, para reducir sobreajuste.

b) Sí, para reducir subajuste.

c) Sí, para que no "infravaloren" variables pequeñas.

d) No, en general no es necesario.

Solución AAI.EX.20240206.T.3
Respuesta correcta: **d)**

Justificación: los árboles toman decisiones por umbrales sobre una característica; en general son **invariantes a transformaciones monótonas** y no dependen de escalas como SVM o kNN.

AAI.EX.20240206.T.4

Enunciado AAI.EX.20240206.T.4
¿Cuál de las siguientes es una desventaja típica de aplicar reducción de dimensionalidad?

a) Puede perderse información útil y empeorar el rendimiento posterior.

b) Mejora la interpretabilidad al transformar las variables.

c) Aumenta el uso de memoria necesario para el procesamiento posterior.

d) Ninguna de las anteriores.

Solución AAI.EX.20240206.T.4
Respuesta correcta: **a)**

Justificación: al comprimir variables se puede perder señal relevante. Además, la interpretabilidad suele **empeorar** (componentes menos intuitivas) y normalmente se reduce, no aumenta, el coste posterior.

AAI.EX.20240206.T.5

Enunciado AAI.EX.20240206.T.5
En un MLP con 10 entradas, una capa oculta de 50 neuronas y una salida de 3 neuronas, ¿qué dimensiones tienen la matriz de pesos de salida W_o y el sesgo b_o?

a) W_o es 3×50 y b_o tiene longitud 3.

b) W_o es 50×3 y b_o tiene longitud 3.

c) W_o es 50×3 y b_o tiene longitud 50.

d) W_o es 3×50 y b_o tiene longitud 50.

Solución AAI.EX.20240206.T.5
Respuesta correcta: **a)**

Justificación:

La salida de la capa de salida de un MLP se calcula mediante la expresión:

$$z_o = W_o\, h + b_o$$

donde:

- z_o es el vector de preactivaciones de la capa de salida.
- W_o es la matriz de pesos que conecta la capa oculta con la capa de salida.
- h es el vector de activaciones de la capa oculta.
- b_o es el vector de sesgos (bias) de la capa de salida.

Sabemos que:

- La capa oculta tiene 50 neuronas, por lo que $h \in \mathbb{R}^{50}$.
- La capa de salida tiene 3 neuronas, por lo que $z_o \in \mathbb{R}^3$.

Para que la multiplicación matricial $W_o\, h$ esté bien definida y produzca un vector de dimensión 3, se requiere que:

- $W_o \in \mathbb{R}^{3\times 50}$.
- $b_o \in \mathbb{R}^3$, ya que hay un sesgo por cada neurona de salida.

Por tanto, la opción correcta es aquella en la que la matriz de pesos de salida tiene dimensiones 3×50 y el sesgo tiene longitud 3.

AAI.EX.20240206.T.6

Enunciado AAI.EX.20240206.T.6
Para elegir aproximadamente el número de componentes (clusters) en un Modelo de Mezcla Gaussiana (GMM), ¿qué criterio se suele emplear?

a) Elegir el número de componentes que maximiza BIC.

b) Elegir el número de componentes que minimiza BIC.

c) Elegir el número de componentes que maximiza AIC.

d) Ninguna de las anteriores.

Solución AAI.EX.20240206.T.6
Respuesta correcta: **b)**

Justificación: tanto AIC como BIC penalizan complejidad; se suelen comparar modelos y elegir el que **minimiza** el criterio (menor es mejor).

AAI.EX.20240206.T.7

Enunciado AAI.EX.20240206.T.7
¿Cuál de las siguientes es un hiperparámetro típico de un MLP?

a) Los pesos.

b) Los términos de sesgo (bias).

c) El número de neuronas en una capa oculta.

d) El número de neuronas de salida.

Solución AAI.EX.20240206.T.7
Respuesta correcta: **c)**

Justificación: pesos y sesgos son parámetros aprendidos. El número de unidades ocultas es una elección de arquitectura, es decir, un hiperparámetro. El número de salidas suele venir impuesto por el problema.

AAI.EX.20240206.T.8

Enunciado AAI.EX.20240206.T.8
En aprendizaje basado en instancias (p. ej., kNN), el modelo "recuerda" ejemplos para predecir nuevos casos. ¿Qué afirmación encaja mejor con este enfoque?

a) No requiere suposiciones y por eso no necesita decisiones de diseño.

b) Requiere conocimiento previo pero no suposiciones sobre similitud.

c) Requiere elegir una noción de similitud (por ejemplo, una métrica), lo que introduce supuestos sobre los datos.

d) Consiste en ajustar un modelo paramétrico como regresión logística o lineal.

Solución AAI.EX.20240206.T.8
Respuesta correcta: **c)**

Justificación: métodos basados en instancias dependen de una **métrica de distancia/similitud** (y a veces ponderaciones), lo cual es un supuesto clave.

AAI.EX.20240206.T.9

Enunciado AAI.EX.20240206.T.9
¿Qué preprocesado de entradas suele ser recomendable para entrenar un MLP de forma estable?

a) Escalar a $[0, 1]$.

b) Escalar a $[-1, 1]$.

c) Estandarizar a media 0 y desviación típica 1.

d) Todas las anteriores pueden ser adecuadas.

Solución AAI.EX.20240206.T.9
Respuesta correcta: **d)**

Justificación: lo importante es que las entradas queden en rangos comparables y "razonables". Normalización o estandarización suelen ayudar; la mejor opción depende de la activación, distribución y presencia de outliers.

AAI.EX.20240206.T.10

Enunciado AAI.EX.20240206.T.10
En una clasificación multiclase con un conjunto de tamaño medio y muchas variables numéricas/categóricas (ya codificadas), ¿qué algoritmo suele ser una opción sólida y flexible?

a) SVM.

b) Árboles de decisión.

c) Naive Bayes.

d) Redes neuronales.

Solución AAI.EX.20240206.T.10
Respuesta correcta: **a)**

Justificación: para tamaños pequeños/medios, las SVM (lineales o con kernel, según el caso) suelen ofrecer muy buen rendimiento y generalización. Redes neuronales suelen brillar más con mucho volumen de datos.

AAI.EX.20240206.D

AAI.EX.20240206.D.1

- Puntuación máxima: 5 puntos
- Extensión máxima orientativa: 2 caras

Enunciado AAI.EX.20240206.D.1
Una entidad financiera realiza cada año una campaña comercial en la que ofrece préstamos preaprobados a una parte de su cartera de clientes. El área de análisis de datos es la responsable de diseñar la estrategia de selección de clientes para la campaña actual, apoyándose en técnicas de aprendizaje automático.

La entidad dispone de dos conjuntos de datos. El primero contiene información financiera de los clientes en el año en curso, incluyendo un identificador de cliente y los saldos iniciales en cuenta corriente, depósitos y fondos de inversión. El segundo conjunto de datos corresponde a la campaña del año anterior e incluye las mismas variables financieras, junto con una etiqueta que indica si el cliente aceptó o no la oferta de préstamo.

El objetivo es construir un sistema que, a partir de la información histórica, estime la probabilidad de que cada cliente acepte la oferta en la campaña actual, de modo que el resultado del modelo sea fácilmente interpretable por el equipo encargado de lanzar la oferta comercial.

Responda de forma razonada a las siguientes cuestiones:

- ¿Qué tipo de algoritmos de aprendizaje automático serían adecuados para este problema? Proponga uno y explique con detalle su funcionamiento.
- Describa un proceso completo de aprendizaje automático que permita entrenar y aplicar el modelo propuesto, identificando sus principales etapas.
- ¿Es necesario ajustar algún parámetro o hiperparámetro del modelo? Indique cuáles y explique su influencia en el comportamiento del sistema.

Solución AAI.EX.20240206.D.1

Planteamiento del problema:

Se quiere predecir en 2023 la probabilidad de que un cliente acepte un préstamo preconcedido. Se dispone de:

- Datos de 2022: (ID, CC-22, De-22, FI-22) y etiqueta Pr-22 (aceptó/no aceptó).
- Datos de 2023: (ID, CC-23, De-23, FI-23) sin etiqueta (todavía no se ha lanzado la campaña).

El objetivo es entrenar un modelo con 2022 y aplicarlo a 2023 para obtener una probabilidad de aceptación por cliente. Además, el modelo debe ser interpretable para el equipo que decide a quién enviar la oferta.

Esto es un problema de **clasificación binaria supervisada** (acepta vs no acepta), con salida probabilística.

Qué tipo de algoritmos se podrían utilizar:

Hay varias opciones razonables y relativamente interpretables:

- Regresión logística (muy interpretable, probabilidades directas).
- Árbol de decisión poco profundo (reglas "si... entonces...", interpretable).
- Modelos lineales con regularización y transformaciones (interpretables si se controlan).
- Modelos de conjunto interpretables mediante explicaciones (p. ej. Gradient Boosting + SHAP), aunque ya no es "interpretabilidad nativa".

Dado que se pide que el modelo sea comprensible por el equipo de campaña y que proporcione probabilidades, una elección estándar y defendible es la **regresión logística**.

Cómo funciona la regresión logística:

La regresión logística modela:

$$P(Pr = 1 \mid x) = \sigma(z) = \frac{1}{1 + e^{-z}}$$

donde:

$$z = \beta_0 + \beta_1 CC + \beta_2 De + \beta_3 FI$$

En este contexto:

- $x = (CC, De, FI)$ son las variables financieras del año de referencia.
- El modelo aprende los coeficientes β para que las probabilidades predichas se ajusten a los resultados observados en 2022 (Pr-22).

Interpretación:

- Cada coeficiente β_j indica cómo cambia el log-odds de aceptación al aumentar la variable correspondiente (manteniendo el resto constante).
- Al transformar con la sigmoide, se obtiene una probabilidad entre 0 y 1, que es directamente utilizable para ranking de clientes.

Ventajas clave:

- Probabilidades bien definidas.
- Interpretabilidad mediante coeficientes.
- Robustez con regularización.
- Facilidad para explicar "qué factores impulsan la aceptación".

Proceso de aprendizaje automático propuesto:

1) **Definición del objetivo y del conjunto de entrenamiento**:
   - Variable objetivo: Pr-22 (0/1).
   - Features: CC-22, De-22, FI-22.
   - El conjunto de entrenamiento se construye con la base de 2022.
2) **Unión de datos y control de calidad**
   - Verificar consistencia de IDs.
   - Eliminar duplicados.
   - Gestionar valores faltantes: imputación simple (mediana) si procede.
   - Revisar outliers extremos (cantidades muy elevadas). Dependiendo del comportamiento, se pueden:
   - aplicar transformaciones logarítmicas,
   - recortar (winsorizar),
   - o simplemente dejarlo si el modelo y la regularización lo soportan.
3) **Ingeniería de características**

   Con solo tres variables se puede mejorar estabilidad e interpretabilidad con transformaciones sencillas:

   - Transformación log para variables monetarias, por ejemplo:

     $$x' = \log(1 + x)$$

     Esto reduce asimetría y hace el efecto más "lineal" para el modelo.
   - Ratios o composición patrimonial:

- total = CC + De + FI
- proporciones: CC/total, De/total, FI/total (si total > 0) Estas variables ayudan a capturar perfil financiero y preferencia por liquidez vs inversión.

4) **Partición entrenamiento–validación**

- Separar un conjunto de validación (por ejemplo 20%) para evaluar generalización.
- Si hay clases desbalanceadas (pocos aceptan), usar partición estratificada para mantener proporciones.
- Alternativa: validación cruzada estratificada si el volumen lo permite.

5) **Entrenamiento del modelo**

- Entrenar una regresión logística con regularización (por defecto L2 suele ir bien).
- Si hay desbalance:
- usar pesos de clase (`class_weight="balanced"`) o ajustar el umbral más tarde.
- Obtener probabilidades en validación.

6) **Evaluación**

Evaluar tanto la capacidad de ranking como la calidad probabilística:

- AUC-ROC para separabilidad.
- Precision/Recall y F1 si interesa priorizar clientes "muy probables".
- Log-loss o Brier score para calibración de probabilidades.
- Curvas de calibración para comprobar si "0.7" significa realmente ~70% en validación.

7) **Selección del umbral y estrategia de campaña**

- Convertir probabilidades en una lista priorizada.
- Elegir un umbral en función de:
- capacidad de contacto (número de ofertas que se pueden enviar),
- objetivo de negocio (maximizar aceptaciones, minimizar coste comercial, etc.).
- Por ejemplo, enviar ofertas al top X% de probabilidades.

8) **Aplicación a 2023**

- Preparar features con CC-23, De-23, FI-23 aplicando las mismas transformaciones usadas en 2022.
- Obtener $P(Pr-23=1 \mid x_{2023})$ por cliente.
- Entregar ranking con explicación:
- coeficientes del modelo,
- variables más influyentes,
- reglas simples derivadas (p. ej. "a mayor saldo total, mayor probabilidad", si aplica).

9) **Interpretabilidad y comunicación**

- Reporte sencillo:
- coeficientes con signo y magnitud,
- ejemplos de clientes con alta/baja probabilidad y las razones,
- calibración y rendimiento global. Esto permite que el equipo de campaña entienda por qué el modelo prioriza ciertos perfiles.

Hiperparámetros a afinar:

Sí, conviene afinar algunos hiperparámetros, especialmente para controlar sobreajuste, calibración y desbalance.

1) **Fuerza de regularización**

- En muchas implementaciones se usa C (inverso de la regularización):
- C pequeño = regularización fuerte = coeficientes más pequeños (modelo más estable).
- C grande = regularización débil = riesgo de sobreajuste. Se elige por validación (grid o búsqueda aleatoria).

2) **Tipo de penalización**

- L2 (ridge): estable, útil cuando hay correlación entre variables.
- L1 (lasso): puede poner coeficientes a cero, favoreciendo selección de variables y simplicidad.
- Elastic net: mezcla de L1 y L2 (si se quiere un equilibrio).

3) **Pesos de clase** Si la aceptación es rara, el modelo puede sesgarse a predecir "no acepta".

- `class_weight="balanced"` compensa el desbalance.
- Alternativamente, se fijan pesos manuales según coste de falsos negativos (dejar fuera un buen candidato) vs falsos positivos (enviar oferta a quien no aceptará).

4) **Umbral de decisión** Aunque no es un hiperparámetro del entrenamiento, sí es un parámetro operativo clave:

- Umbral bajo aumenta recall (capturas más aceptantes potenciales) pero baja precisión.
- Umbral alto aumenta precisión pero puedes perder aceptantes. Se ajusta con la curva Precision–Recall o según el número máximo de contactos.

5) **Transformaciones y selección de features** No es un hiperparámetro interno del algoritmo, pero es parte del ajuste del pipeline:

- usar o no usar log(1+x),
- usar totales y proporciones,
- tratar outliers. Se decide por validación midiendo mejora y estabilidad.

Con este enfoque se obtiene un modelo probabilístico, interpretable y accionable: entrenado con 2022, aplicado a 2023 para priorizar clientes con mayor probabilidad de aceptación, con explicaciones basadas en coeficientes y validación de calibración.