AAI.EX.20250204

Ejercicios elaborados con fines educativos, inspirados en los contenidos evaluados en el exámen del 04/02/2025 (convocatoria Feb-2024) de Aprendizaje Automático 1 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.20250204.T.1

Enunciado AAI.EX.20250204.T.1 En un árbol de decisión entrenado con un criterio como Gini, ¿cómo suele compararse la impureza de un nodo hijo con la impureza de su nodo padre?

a) Suele ser mayor.

b) Suele ser menor.

c) Es siempre menor.

d) No existe relación.

Solución AAI.EX.20250204.T.1 Respuesta correcta: **b)**

Justificación: los splits se eligen para **reducir** la impureza, por lo que normalmente los hijos tienen menor impureza que el padre (aunque no es una garantía absoluta para *todas* las situaciones/implementaciones).

AAI.EX.20250204.T.2

Enunciado AAI.EX.20250204.T.2 En una SVM de margen máximo (caso ideal lineal separable), ¿qué puntos del entrenamiento se consideran *vectores soporte*?

a) Los puntos situados justo en los bordes del margen.

b) Los puntos dentro del margen y también en sus bordes.

c) En el caso separable, los puntos estrictamente dentro del margen.

d) Ninguna de las anteriores.

Solución AAI.EX.20250204.T.2 Respuesta correcta: **a)**

Justificación: los vectores soporte son los ejemplos que **tocan** el margen y determinan la frontera óptima.

AAI.EX.20250204.T.3

Enunciado AAI.EX.20250204.T.3 Aproximadamente, ¿qué profundidad puede tener un árbol de decisión no restringido entrenado sobre $n = 10^6$ instancias con clases perfectamente balanceadas?

a) 15

b) 20

c) 25

d) 10

Solución AAI.EX.20250204.T.3 Respuesta correcta: **b)**

Justificación: un árbol aproximadamente balanceado tiene profundidad del orden de:

$$\log_2(n)$$

y $\log_2(10^6) \approx 20$.

AAI.EX.20250204.T.4

Enunciado AAI.EX.20250204.T.4 ¿Cuál de las siguientes afirmaciones **no** describe correctamente una propiedad típica de las SVM?

a) Pueden trabajar con muchas variables predictoras.

b) Pueden construir fronteras no lineales mediante kernels.

c) Solo sirven cuando las clases son perfectamente separables.

d) Pueden resolver multiclase mediante estrategias estándar (p. ej., one-vs-rest).

Solución AAI.EX.20250204.T.4 Respuesta correcta: **c)**

Justificación: las SVM con *soft margin* permiten errores y manejan datos **no separables** usando el parámetro C.

AAI.EX.20250204.T.5

Enunciado AAI.EX.20250204.T.5 En `sklearn.neighbors.KernelDensity`, ¿cuál es el kernel que se usa por defecto?

a) `tophat`

b) `epanechnikov`

c) `exponential`

d) `gaussian`

Solución AAI.EX.20250204.T.5 Respuesta correcta: **d)**

Justificación: el kernel por defecto en `KernelDensity` es el **gaussiano**.

AAI.EX.20250204.T.6

Enunciado AAI.EX.20250204.T.6 Para un perceptrón multicapa entrenado con retropropagación, con:

- n muestras,
- m características,
- k capas ocultas con h neuronas cada una,
- o neuronas de salida,
- i iteraciones,

¿qué expresión aproxima mejor su complejidad temporal?

a) $O(n \cdot m \cdot h \cdot k \cdot o \cdot i)$

b) $O(n \cdot m \cdot h^k \cdot o \cdot i)$

c) $O(n \cdot m \cdot \log(h \cdot k) \cdot o \cdot i)$

d) Ninguna de las anteriores

Solución AAI.EX.20250204.T.6 Respuesta correcta: **a)**

Justificación: La complejidad temporal de la retropropagación viene dada por la fórmula $O(i \cdot n(m \cdot h + (k-1) \cdot h \cdot h + h \cdot o))$, tal y como se indica en el apartado 1.17.6. "Complexity" de Skikit-learn, donde nm son las instancias de entrenamiento, n las características, k las capas ocultas, cada una de ellas conteniendo h neuronas (simplificando) y o neuronas de salida.

En conclusión, el coste por iteración escala aproximadamente con el número de operaciones asociadas a conexiones y activaciones y la dependencia es aproximadamente lineal en n, m, h, k, o e i.

AAI.EX.20250204.T.7

Enunciado AAI.EX.20250204.T.7 ¿Es necesario escalar/normalizar características de entrada para entrenar árboles de decisión de forma adecuada?

a) Sí, para evitar sobreajuste.

b) Sí, para evitar subajuste.

c) Sí, para que no se infravaloren variables de escala pequeña.

d) No, normalmente no es necesario.

Solución AAI.EX.20250204.T.7 Respuesta correcta: **d)**

Justificación: los árboles se basan en umbrales por característica y suelen ser **invariantes a la escala**, a diferencia de SVM o kNN.

AAI.EX.20250204.T.8

Enunciado AAI.EX.20250204.T.8 En clasificación multiclase con un conjunto pequeño, con atributos aproximadamente independientes y de distribución normal (tras preprocesado), ¿qué algoritmo suele ofrecer inferencia muy rápida y encajar bien con esos supuestos?

a) SVM

b) Árboles de decisión

c) Naive Bayes

d) Redes neuronales

Solución AAI.EX.20250204.T.8 Respuesta correcta: **c)**

Justificación: Naive Bayes (p. ej., `GaussianNB`) tiene inferencia muy rápida y se alinea con independencia condicional y normalidad de las variables.

AAI.EX.20250204.T.9

Enunciado AAI.EX.20250204.T.9 La regla de Hebb (inspiración histórica del perceptrón) se resume comúnmente como:

a) La conexión entre dos neuronas tiende a fortalecerse si se activan a la vez.

b) La conexión tiende a debilitarse si se activan a la vez.

c) La conexión se debilita si se activa la primera, independientemente de la segunda.

d) La conexión se fortalece solo si la segunda neurona conecta con salidas.

Solución AAI.EX.20250204.T.9 Respuesta correcta: **a)**

Justificación: la intuición hebbiana clásica es "neuronas que disparan juntas, se conectan más" (aumento del peso con co-activación).

AAI.EX.20250204.T.10

Enunciado AAI.EX.20250204.T.10 En una clasificación multiclase con un conjunto grande y mezcla de variables numéricas/categóricas ya codificadas, ¿qué opción suele ser una elección general adecuada para aprovechar el volumen de datos?

a) SVM

b) Árboles de decisión

c) Naive Bayes

d) Redes neuronales

Solución AAI.EX.20250204.T.10 Respuesta correcta: **d)**

Justificación: con muchos datos, las redes neuronales suelen escalar bien y aprovechar el volumen para aprender representaciones, mientras que SVM con kernels puede volverse costosa.

AAI.EX.20250204.D

AAI.EX.20250204.D.1

Puntuación máxima: 5 puntos Extensión máxima orientativa: 2 caras

Enunciado AAI.EX.20250204.D.1 En una red de metro urbana se han registrado durante el último mes varios robos en horario nocturno con un patrón muy similar, lo que sugiere la posible actuación coordinada de un mismo grupo. Los incidentes se concentran en un conjunto fijo de 10 estaciones situadas en el centro de la ciudad.

La unidad de análisis policial dispone de un único conjunto de datos agregado correspondiente a los últimos 30 días. Para cada una de las 10 estaciones se proporcionan las siguientes variables:

- número de personas que acceden a la estación durante el periodo,
- número de personas que salen de la estación durante el periodo,
- número de trenes que circulan por la estación durante el periodo,
- número total de delitos registrados en la estación durante el mismo periodo.

En total, se cuenta con una tabla de 10 registros (uno por estación) y cuatro variables. Se solicita proponer un enfoque de aprendizaje automático que permita:

- estimar el riesgo/probabilidad de que se produzcan delitos en las estaciones en el futuro próximo, o
- agrupar las estaciones según su nivel de riesgo estimado.

Responda de forma razonada:

- ¿Es más apropiado utilizar aprendizaje supervisado o no supervisado en este contexto? Justifique la elección.
- ¿Qué tipo de algoritmos podrían emplearse? Proponga uno y explique su funcionamiento.
- ¿Qué parámetros o hiperparámetros sería necesario ajustar en el modelo propuesto? Indique cuáles y describa su efecto.

Solución AAI.EX.20250204.D.1

Planteamiento:

Dispones de un único dataset del último mes, con 10 estaciones (X1. . . X10) y 4 variables por estación:

- Num-E: número de entradas
- Num-S: número de salidas
- Num-T: número de trenes
- Num-D: número de delitos

Por tanto, hay 10 observaciones (una por estación) y además Num-D es una medida agregada del mes. Con este tamaño muestral tan pequeño no se puede entrenar un modelo "estándar" con garantías estadísticas: cualquier algoritmo puede sobreajustar y la validación sería muy débil. Aun así, sí se puede plantear un enfoque razonable si se entiende como:

- una **estimación de riesgo relativa** entre estaciones (ranking), o
- una **agrupación por nivel de riesgo** (clustering), o
- un modelo supervisado muy simple si aceptamos fuertes supuestos o regularización extrema.

También es importante distinguir dos objetivos posibles:

1) "Predecir probabilidad de delito" (supervisado, requiere variable objetivo bien definida a nivel de ejemplo)
2) "Agrupar estaciones por riesgo" (no supervisado, no requiere etiquetas)

¿Supervisado o no supervisado?

Sí se puede usar **cualquiera de los dos**, pero la elección depende de cómo definamos el objetivo y de la granularidad de los datos.

1) **Aprendizaje supervisado**:

   - A favor: existe una variable relacionada con el delito (Num-D). Si definimos el objetivo como "riesgo de delito" y usamos Num-D como señal, podríamos ajustar un modelo que relacione (Num-E, Num-S, Num-T) con Num-D.
   - En contra (muy importante): con 10 observaciones es difícil estimar generalización. Además, Num-D es un conteo mensual agregado; no hay etiquetas a nivel de evento (día/hora), ni ejemplos negativos/positivos por periodo. Hablar de "probabilidad" en sentido estricto requeriría modelar el número de delitos como variable aleatoria condicionada a la exposición, o disponer de datos por día/turno.

   Conclusión: supervisado es posible si lo planteas como **modelado de conteos** (o tasa) con regularización fuerte y con el objetivo de comparar estaciones, no de producir probabilidades calibradas "de verdad".

2) **Aprendizaje no supervisado**:

   - A favor: si el objetivo es "agrupar estaciones por nivel de riesgo", no necesitas una etiqueta. Puedes agrupar según patrones de afluencia y operación (Num-E, Num-S, Num-T) y, opcionalmente, usar Num-D después para interpretar los grupos.
   - En contra: clustering no "predice delitos" directamente; produce grupos. El salto de grupo $\rightarrow$ probabilidad requiere una interpretación posterior (por ejemplo, asignar a cada cluster un riesgo medio observado).

Conclusión: no supervisado es especialmente defendible con tan pocos datos si el objetivo principal es **segmentar** estaciones y priorizar intervención.

¿Qué algoritmos se podrían utilizar? Propuesta y funcionamiento:

Propongo dos opciones típicas, una supervisada y otra no supervisada. En el examen puedes elegir una y desarrollarla a fondo; aquí describo ambas para cubrir la pregunta.

Opción A (supervisado): Regresión de Poisson (o Binomial Negativa) para conteos:

Si Num-D es el número de delitos del mes, es natural modelarlo como conteo. Un modelo clásico es la **regresión de Poisson**:

- Supuesto: $Y = \text{Num-D}$ sigue una distribución Poisson con media λ.
- Relación con variables explicativas: se usa un enlace logarítmico

$$\log(\lambda_i) = \beta_0 + \beta_1 \,\text{Num-E}_i + \beta_2 \,\text{Num-S}_i + \beta_3 \,\text{Num-T}_i$$

De ahí:

$$\lambda_i = \exp(\beta_0 + \beta_1 \,\text{Num-E}_i + \beta_2 \,\text{Num-S}_i + \beta_3 \,\text{Num-T}_i)$$

Interpretación:

- λ_i es el número esperado de delitos en la estación i (en el periodo).
- Si quieres "probabilidad de al menos un delito" en el próximo periodo, puedes aproximar:

$$P(Y \geq 1) = 1 - P(Y = 0) = 1 - e^{-\lambda}$$

Esto convierte un modelo de conteo en una probabilidad de ocurrencia (bajo el supuesto Poisson para ese horizonte).

Puntos clave para hacerlo más realista:

- Usar una **tasa** en lugar de conteo: delitos por "exposición". Por ejemplo, incluir un *offset* si tuvieras duración o afluencia total. Con lo dado, podrías aproximar exposición con Num-E+Num-S o con Num-T, aunque es un proxy.
- Si hay sobredispersión (varianza mayor que la media), Poisson puede fallar; en ese caso, **Binomial Negativa** suele ajustarse mejor.

Con 10 puntos, el objetivo práctico sería obtener un **ranking** de λ_i o de $P(Y \geq 1)$ para priorizar patrullaje y vigilancia.

Opción B (no supervisado): K-means o clustering jerárquico:

Si lo que se desea es agrupar estaciones por "nivel de riesgo probable", se puede hacer clustering usando variables que reflejen oportunidad/exposición:

- Num-E, Num-S, Num-T (y opcionalmente Num-D para interpretar, no para construir el grupo)

Un método simple es **K-means**:

1) Se elige un número de clusters k.
2) Se inicializan centroides.
3) Se asigna cada estación al centroide más cercano (distancia euclídea).
4) Se recalculan centroides como media de los puntos del cluster.
5) Se repite hasta convergencia.

Para que tenga sentido:

- Es imprescindible escalar variables (por ejemplo, estandarización) porque Num-E, Num-S y Num-T pueden tener magnitudes muy distintas.
- Tras obtener clusters, se puede evaluar el riesgo medio observado por cluster mirando Num-D y decir "Cluster A = alto riesgo", etc.

Con 10 estaciones, clustering jerárquico también es atractivo porque no exige fijar k al principio y permite visualizar agrupaciones con un dendrograma.

¿Es necesario ajustar parámetros o hiperparámetros? ¿Cuáles?

Sí. Con tan pocos datos, los hiperparámetros y decisiones de modelado son críticos.

Para regresión de Poisson / Binomial Negativa (supervisado):

1) **Regularización (L2/L1 o elastic net)**:
   - Con 10 observaciones, un modelo sin regularización puede ser inestable.
   - Regularización reduce varianza del estimador.
   - Hiperparámetro típico: fuerza de regularización (α o C según implementación).
2) **Offset o elección de exposición**:
   - No es un hiperparámetro "clásico", pero sí una decisión clave.
   - Si consideras que el riesgo crece con afluencia, puedes incluir un offset como $\log(\text{exposición}_i)$ para modelar una tasa.
   - Ejemplo conceptual:

$$\log(\lambda_i) = \log(\text{exposición}_i) + \beta^\top x_i$$

3) **Elección de familia: Poisson vs Binomial Negativa**:
   - Si observas sobredispersión (Num-D varía mucho más de lo esperado), Binomial Negativa añade un parámetro de dispersión.

- Parámetro relevante: dispersión (a veces denotado θ o k) que controla varianza.

4) **Selección de variables**:

- Con pocos datos, conviene limitar el número de features y usar transformaciones simples:
- ratios: Num-E/Num-T, (Num-E+Num-S)/Num-T
- log-transform si hay escalas muy grandes
- No es hiperparámetro formal, pero es parte del ajuste del modelo.

Para K-means / clustering jerárquico (no supervisado):

1) **Número de clusters k (K-means)**:

- Es el hiperparámetro principal.
- Con 10 estaciones, valores como $k = 2$ o $k = 3$ suelen ser razonables para "bajo/medio/alto".
- Se puede elegir por criterio de codo (*elbow*) o silhouette, aunque con tan pocos puntos estas heurísticas son inestables.

2) **Inicialización y número de reinicios (K-means)**:

- K-means puede converger a mínimos locales.
- Parámetros típicos: método de inicialización (k-means++) y número de inicializaciones (n_init).

3) **Métrica y enlace (clustering jerárquico)**:

- Elección de distancia (euclídea, manhattan).
- Elección de *linkage* (complete, average, ward).
- Con pocas observaciones, el resultado puede cambiar según estas elecciones; por eso se justifica escoger, por ejemplo, Ward + euclídea cuando se han escalado variables.

4) **Escalado de variables (imprescindible)**:

- No es hiperparámetro del algoritmo, pero es crítico para que la distancia sea significativa.
- StandardScaler (media 0, varianza 1) es el estándar.

Comentario final sobre viabilidad:

Con un dataset de 10 filas, cualquier conclusión debe presentarse como exploratoria y operativa (priorización y agrupación), no como un sistema de predicción robusto. Un paso clave para convertirlo en un problema de ML más sólido sería disponer de datos a nivel diario o por franja horaria durante varios meses, lo que permitiría modelos supervisados con validación temporal real.