

AAI.EX.20230904

Ejercicios elaborados con fines educativos, inspirados en los contenidos evaluados en el examen del 04/09/2024 (convocatoria Sep-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.20230904.T.1

Enunciado AAI.EX.20230904.T.1

En una SVM de margen máximo, ¿qué instancias del conjunto de entrenamiento reciben el nombre de *vectores soporte*?

- a) Las instancias situadas exactamente sobre los bordes del margen (las rectas/planos paralelos a la frontera).
- b) Las instancias situadas dentro de la franja del margen y también sobre sus bordes.
- c) En el caso perfectamente separable, las instancias estrictamente dentro de la franja del margen.
- d) Ninguna de las anteriores.

Solución AAI.EX.20230904.T.1

Respuesta correcta: **a)**

Justificación: los vectores soporte son los puntos que “tocan” el margen y determinan la posición de la frontera óptima (en el caso ideal separable).

AAI.EX.20230904.T.2

Enunciado AAI.EX.20230904.T.2

Si un árbol de decisión está sobreajustando, ¿qué cambio suele ayudar más a reducir la complejidad del modelo?

- a) Incrementar `max_depth`.
- b) Incrementar `min_samples_leaf`.

- c) Reducir `max_depth`.
- d) Reducir `min_samples_leaf`.

Solución AAI.EX.20230904.T.2

Respuesta correcta: **b)**

Justificación: aumentar `min_samples_leaf` fuerza a que exista un mínimo de ejemplos antes de crear una hoja para ese caso concreto, suaviza el árbol (evitando cambios bruscos para recoger instancias muy concretas) y reduce varianza (obteniendo un modelo similar si cambias pocos datos y repites el entrenamiento). Reducir `max_depth` también puede ayudar, pero aquí la opción más directamente orientada a “regularizar” hojas es `min_samples_leaf`.

AAI.EX.20230904.T.3

Enunciado AAI.EX.20230904.T.3

En una clasificación multiclase con muchas variables numéricas y categóricas (ya preprocesadas), si tu prioridad es que el modelo sea fácil de explicar a personas (evitar “caja negra”), ¿qué familia elegirías?

- a) SVM.
- b) Árboles de decisión.
- c) Naive Bayes.
- d) Redes neuronales.

Solución AAI.EX.20230904.T.3

Respuesta correcta: **b)**

Justificación: los árboles son altamente interpretables (reglas tipo “si-entonces”), especialmente si se limitan profundidad y se visualizan.

AAI.EX.20230904.T.4

Enunciado AAI.EX.20230904.T.4

¿Por qué funciones como la logística (sigmoide) fueron importantes para entrenar redes multicapa con retropropagación?

- a) Porque su derivada es siempre positiva.
- b) Porque su derivada está acotada entre 0 y 1.

- c) Porque su derivada nunca se hace cero.
- d) Porque es derivable.

Solución AAI.EX.20230904.T.4

Respuesta correcta: **d)**

Justificación: la retropropagación necesita calcular gradientes; para ello se requieren funciones de activación **derivables**. Que la derivada sea pequeña puede ser un problema, no una ventaja.

AAI.EX.20230904.T.5

Enunciado AAI.EX.20230904.T.5

Para un MLP que clasifica correos en dos clases (por ejemplo, “spam” vs “no spam”), ¿qué activación es típica en la neurona de salida si produces una probabilidad?

- a) ReLU.
- b) Sigmoide (logística).
- c) Softmax.
- d) Sin activación.

Solución AAI.EX.20230904.T.5

Respuesta correcta: **b)**

Justificación: en clasificación binaria se suele usar una única salida con sigmoide para obtener $p(y = 1 | x)$.

AAI.EX.20230904.T.6

Enunciado AAI.EX.20230904.T.6

En Naive Bayes, ¿qué parámetros se estiman a partir del conjunto de entrenamiento?

- a) $P(\text{clase})$ y $P(\text{clase} | \text{característica})$.
- b) $P(\text{clase})$ y $P(\text{característica} | \text{clase})$.
- c) Solo $P(\text{clase})$.
- d) Solo $P(\text{clase} | \text{característica})$.

Solución AAI.EX.20230904.T.6

Respuesta correcta: **b)**

Justificación: Naive Bayes aprende los *priors* $P(y)$ y las *verosimilitudes* $P(x_j | y)$ (asumiendo independencia condicional entre características dado y).

AAI.EX.20230904.T.7

Enunciado AAI.EX.20230904.T.7

¿Cuál de las siguientes afirmaciones es la incorrecta?

- a) Algunos algoritmos (p. ej., regresión logística, bosques aleatorios y SVM) pueden tratar multiclase de forma nativa o mediante estrategias estándar.
- b) Las SVM son versátiles: pueden hacer clasificación lineal/no lineal y regresión (SVR) lineal/no lineal.
- c) Las SVM suelen funcionar especialmente bien en problemas complejos de tamaño pequeño o mediano.
- d) Las SVM no son sensibles a la escala de las variables de entrada.

Solución AAI.EX.20230904.T.7

Respuesta correcta: **d)**

Justificación: las SVM **sí** son sensibles a la escala porque dependen de distancias/productos escalares; se recomienda estandarizar o normalizar.

AAI.EX.20230904.T.8

Enunciado AAI.EX.20230904.T.8

En una clasificación multiclase con un conjunto de datos grande (muchas instancias) y mezcla de variables numéricas/categóricas ya transformadas, ¿qué opción suele ser más apropiada como elección general?

- a) SVM.
- b) Árboles de decisión.
- c) Naive Bayes.
- d) Redes neuronales.

Solución AAI.EX.20230904.T.8

Respuesta correcta: **d)**

Justificación: con muchos datos, los modelos de redes neuronales suelen escalar bien y aprovechar el volumen para aprender representaciones, mientras que SVM con kernels puede ser costosa a gran escala.

AAI.EX.20230904.T.9

Enunciado AAI.EX.20230904.T.9

Considera un MLP entrenado por retropropagación con:

- n muestras,
- m características de entrada,
- k capas ocultas con h neuronas cada una,
- o neuronas de salida,
- i iteraciones (épocas o pasos equivalentes).

¿Cuál expresión se aproxima mejor a la 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.20230904.T.9

Respuesta correcta: **a)**

Justificación: el coste por pasada es proporcional al número de operaciones en las capas (en torno a conexiones), y escalando por n e iteraciones i se obtiene una dependencia aproximadamente lineal en n , m , h , k , o e i .

AAI.EX.20230904.T.10

Enunciado AAI.EX.20230904.T.10

Para que un robot aprenda a desplazarse en un entorno desconocido mediante interacción con el entorno y recompensas/penalizaciones, ¿qué tipo de aprendizaje es el más adecuado?

- a) Supervisado.
- b) No supervisado.
- c) Por refuerzo.
- d) Semi-supervisado.

Solución AAI.EX.20230904.T.10

Respuesta correcta: c)

Justificación: el aprendizaje por refuerzo modela decisiones secuenciales y aprendizaje por ensayo-error mediante recompensas, encajando con navegación en entornos desconocidos.

AAI.EX.20230904.D

AAI.EX.20230904.D.1

Puntuación máxima: 5 puntos

Extensión máxima orientativa: 2 caras

Enunciado AAI.EX.20230904.D.1

Se desea desarrollar una herramienta de análisis para un sistema de enseñanza en línea cuyo propósito sea anticipar el rendimiento académico final de los estudiantes. Esta estimación permitirá tanto al alumnado como al profesorado identificar con antelación si es necesario intensificar el estudio para superar la asignatura.

El análisis se centrará inicialmente en una materia que incluye cuatro evaluaciones parciales previas al examen final. Se dispone de información histórica de cursos anteriores que recoge:

- las calificaciones obtenidas en cada una de las cuatro pruebas,
- el nivel de participación del estudiante en el foro de la asignatura (número de mensajes),
- la calificación obtenida en el examen final.

A partir de esta información, responda de forma razonada a las siguientes cuestiones:

- Diseñe un proyecto de aprendizaje automático que permita estimar numéricamente la calificación final de nuevos estudiantes en dos momentos distintos:
 - a) antes de realizar la última prueba parcial,
 - b) antes de realizar el examen final.Indique qué algoritmo utilizaría en cada caso y justifique su elección.
- Diseñe un proyecto de aprendizaje automático que permita **clasificar a los estudiantes en tres categorías** (posible suspenso, dudoso, posible aprobado) o, alternativamente, estimar la probabilidad de pertenencia a cada una de ellas. Indique qué algoritmo emplearía y explique los motivos de su elección.

Solución AAI.EX.20230904.D.1

Planteamiento:

Se quiere construir una aplicación que estime el rendimiento final de estudiantes en educación a distancia. Se dispone de datos históricos con:

- Notas de 4 pruebas previas (P1, P2, P3, P4).
- Número de mensajes en el foro (Msg).
- Nota del examen final (Final).

Se piden diseños para:

- 1) Predicción numérica en dos momentos: (a) antes de realizar la última prueba (P4) y (b) antes de realizar el examen final.
- 2) Clasificación en tres clases (posible suspenso, dudoso, posible aprobado) o probabilidad de pertenencia a cada clase.

En todos los casos se asume aprendizaje supervisado, porque hay etiquetas históricas (Final o la clase derivada).

1) Estimación numérica en dos casos: diseño de proyecto y algoritmo:

Caso 1A: estimar la nota final antes de realizar la última prueba (P4):

Objetivo:

- Predecir una nota numérica (regresión) de la evaluación final con información parcial: P1, P2, P3 y Msg.

Variables de entrada:

- P1, P2, P3
- Msg (mensajes acumulados hasta ese momento, idealmente)

Variable objetivo:

- Final (nota del examen final) o nota final global si existiera una nota final distinta; como el enunciado da “evaluación en el examen final”, se toma esa como objetivo.

Diseño del proyecto:

- 1) Definición del dataset
 - Construir ejemplos históricos con features (P1, P2, P3, Msg) y target Final.
- 2) Limpieza y preprocesamiento
 - Control de nulos, coherencia de rangos de notas.
 - Escalado opcional (dependiendo del modelo).
 - Transformación de Msg si está muy sesgada: usar $\log(1+Msg)$ para reducir asimetría.
- 3) División entrenamiento/validación/test

- Si hay datos de varios años, separar por años (validación temporal) para simular generalización a cohortes nuevas.
- 4) Entrenamiento
 - Modelo propuesto: regresión lineal regularizada (Ridge o Elastic Net).
 - 5) Evaluación
 - MAE y RMSE, y análisis de residuales para detectar sesgo (p. ej. sobreestimar a estudiantes con pocos mensajes).
 - 6) Uso operativo
 - El sistema devuelve una nota esperada y, si interesa, un intervalo aproximado de incertidumbre (por ejemplo con cuantiles si se implementa).

Algoritmo recomendado y por qué: - Regresión lineal regularizada (Ridge o Elastic Net). Razones: - Interpretabilidad: se puede explicar el peso de P1, P2, P3 y Msg. - Estabilidad: la regularización reduce sobreajuste, especialmente si hay correlación entre pruebas. - Base sólida: suele rendir bien cuando la relación es aproximadamente monotónica y aditiva.

Alternativa si se sospechan no linealidades fuertes: - Árboles de boosting para regresión, pero sacrifican interpretabilidad nativa (aunque se puede recuperar con explicaciones).

Caso 1B: estimar la nota final antes de realizar el examen final (reescrito según temario):

Objetivo:

Predecir numéricamente la calificación del examen final utilizando toda la información disponible justo antes del examen:

- P1, P2, P3, P4
- nivel de participación en el foro (Msg)

Se trata de un problema de aprendizaje supervisado de regresión.

Variables:

- Entradas (X):
 - P1, P2, P3, P4
 - Msg (número de mensajes acumulados)
- Salida (y):
 - Final (nota del examen final)

Algoritmo elegido:

Árbol de decisión para regresión.

Justificación de la elección:

- Permite modelar relaciones no lineales entre las calificaciones parciales y la nota final.
- Captura efectos de umbral habituales en educación (por ejemplo, suspensos casi seguros por debajo de cierta nota).
- No requiere escalado de variables.
- Es interpretable: las reglas de decisión pueden explicarse a profesores y estudiantes.
- Funciona bien con conjuntos de datos de tamaño medio, habituales en contextos académicos.

Diseño del proyecto:

- 1) Definición del dataset
 - Construir ejemplos históricos con entradas (P1, P2, P3, P4, Msg) y salida Final.
- 2) Preprocesamiento
 - Comprobación de valores faltantes o fuera de rango.
 - Transformación opcional de Msg si presenta fuerte asimetría (por ejemplo, $\log(1 + \text{Msg})$).
 - No es necesario escalado.
- 3) División entrenamiento–validación
 - Preferiblemente separar por cursos académicos para evaluar generalización temporal.
 - Alternativamente, validación cruzada si no hay información temporal clara.
- 4) Entrenamiento
 - Ajustar un árbol de decisión de regresión minimizando el error cuadrático.
 - Controlar la complejidad mediante hiperparámetros como:
 - profundidad máxima,
 - número mínimo de muestras por hoja.
- 5) Evaluación
 - Métricas de regresión:
 - MAE
 - RMSE
 - Comparar el error con el modelo del caso 1A; se espera una mejora al disponer de P4.
- 6) Uso operativo
 - El sistema devuelve una nota final estimada.
 - Puede acompañarse de reglas interpretables (por ejemplo: “si $P4 < 4$, la nota final rara vez supera el aprobado”).

Conclusión:

Un árbol de decisión para regresión permite estimar la calificación final antes del examen de forma coherente con el temario, capturando relaciones no lineales y manteniendo interpretabilidad, lo que resulta especialmente adecuado en un entorno educativo.

2) Clasificar en tres clases o estimar probabilidades: diseño y algoritmo (según temario):

Se requiere asignar a cada estudiante una clase entre:

- posible suspenso
- dudoso
- posible aprobado
y/o estimar probabilidades de pertenencia a cada clase.

Definición de clases:

A partir de la nota histórica Final se definen etiquetas con umbrales (ajustables según criterio académico), por ejemplo:

- posible suspenso: $\text{Final} < 5$
- dudoso: $5 \leq \text{Final} < 6$
- posible aprobado: $\text{Final} \geq 6$

Diseño del proyecto:

- 1) Construcción de etiquetas
 - Convertir cada Final histórica en una clase (3 niveles) usando los umbrales definidos.
- 2) Selección de variables (según el momento)
 - Antes de P4: (P1, P2, P3, Msg)
 - Antes del examen: (P1, P2, P3, P4, Msg)
- 3) Preprocesamiento
 - Revisión de nulos y rangos de notas.
 - Transformación opcional de Msg si está muy sesgada (por ejemplo, $\log(1 + \text{Msg})$).
 - Escalado solo si el modelo lo requiere (por ejemplo, kNN o SVM). Para Naive Bayes no es imprescindible.
- 4) División entrenamiento–validación–test
 - Preferiblemente por cohortes (años/cursos), para evaluar generalización temporal.
- 5) Entrenamiento (modelo recomendado):
 - **Naive Bayes** para clasificación multiclase (por ejemplo, GaussianNB si las variables son numéricas).
 - Produce probabilidades por clase de forma natural mediante `predict_proba`.
- 6) Evaluación
 - Matriz de confusión y **F1-macro** (para no ignorar la clase “dudoso”).
 - Si interesa calidad de probabilidades: log-loss (opcional).
- 7) Uso operativo

- Mostrar:
 - clase predicha
 - probabilidades por clase, por ejemplo: $P(\text{posible suspenso})=0.15$, $P(\text{dudoso})=0.55$, $P(\text{posible aprobado})=0.30$
- Definir acciones: recomendar refuerzo cuando $P(\text{posible suspenso})+P(\text{dudoso})$ sea alta.

Algoritmo recomendado y por qué:

- **Naive Bayes**
 - Devuelve probabilidades de forma directa (`predict_proba`).
 - Entrena y predice muy rápido, útil si hay muchos estudiantes.
 - Suele funcionar bien como modelo base incluso con pocos datos.

Alternativa:

- **Árbol de decisión (clasificación)**
 - Muy interpretable por reglas (“si $P4 < x$ entonces...”).
 - También devuelve probabilidades (proporción por clase en la hoja).
 - Requiere limitar profundidad/`min_samples_leaf` para evitar sobreajuste.

Notas prácticas:

- Entrenar dos modelos distintos (antes de P4 y antes del examen), porque cambian las variables disponibles.
- Evitar fuga de información: `Msg` debe ser el acumulado hasta el momento de predicción, no el total del curso.