

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 preprocessado 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

Este es un problema de aprendizaje supervisado de clasificación binaria: la etiqueta histórica $y \in \{0, 1\}$ indica si el cliente aceptó (1) o no (0) la oferta. Además, se pide estimar una *probabilidad* de aceptación para cada cliente actual y que el resultado sea *fácilmente interpretable*.

Paso 1: Algoritmos adecuados y propuesta

Algoritmos adecuados (según el temario):

- *Naive Bayes* (clasificación probabilística e interpretable).
- *rboles de decisin* (muy interpretables; pueden producir probabilidades por proporción de clase en hojas).
- *K-Nearest Neighbors* (simple, pero menos interpretable globalmente; probabilidades por frecuencia de vecinos).
- (SVM aparece en temario, pero no es la opción más interpretable y la salida no es probabilística de forma nativa).

Propuesta: *Naive Bayes* (concretamente *Gaussian Naive Bayes*, apropiado para variables numéricas como saldos).

Funcionamiento detallado (Gaussian Naive Bayes):

- Datos de entrenamiento: para cada cliente del año anterior se dispone de un vector de características

$$\mathbf{x} = (x_1, x_2, x_3)$$

donde, por ejemplo, x_1 =saldo cuenta corriente, x_2 =depósitos, x_3 =fondos, y la etiqueta $y \in \{0, 1\}$.

- Hipótesis clave de Naive Bayes (independencia condicional):

$$P(\mathbf{x} | y) = \prod_{j=1}^d P(x_j | y)$$

Es decir, dado y , cada variable financiera aporta evidencia de forma “independiente” (supuesto simplificador que suele funcionar razonablemente bien).

- Regla de Bayes para clasificar:

$$P(y | \mathbf{x}) = \frac{P(y) P(\mathbf{x} | y)}{P(\mathbf{x})}$$

Como $P(\mathbf{x})$ es el mismo para todas las clases al comparar, se usa:

$$P(y | \mathbf{x}) \propto P(y) \prod_{j=1}^d P(x_j | y)$$

- En la variante Gaussiana se modela cada característica por clase como una Normal:

$$x_j | (y = c) \sim \mathcal{N}(\mu_{jc}, \sigma_{jc}^2)$$

donde $c \in \{0, 1\}$ y se estiman por máxima verosimilitud a partir del histórico:

$$\mu_{jc} = \frac{1}{n_c} \sum_{i:y_i=c} x_{ij}, \quad \sigma_{jc}^2 = \frac{1}{n_c} \sum_{i:y_i=c} (x_{ij} - \mu_{jc})^2$$

y el prior:

$$P(y = c) = \frac{n_c}{n}$$

- Predicción probabilística (lo que pide el enunciado): Para un cliente actual con \mathbf{x} , el modelo devuelve

$$\hat{p} = P(y = 1 | \mathbf{x})$$

y esa \hat{p} se puede interpretar como “probabilidad estimada de aceptar”.

Interpretabilidad:

- Se puede explicar como: “la probabilidad sale de combinar (i) la tasa histórica de aceptación $P(y = 1)$ y (ii) qué tan compatibles son los saldos del cliente con los patrones observados en aceptadores vs no aceptadores”.
- Además, se puede inspeccionar μ_{j1}, σ_{j1} frente a μ_{j0}, σ_{j0} para justificar por qué cierta variable empuja hacia aceptar/no aceptar (p.ej., aceptadores tienden a tener más depósitos).

Paso 2: Proceso completo de aprendizaje automático (entrenar y aplicar)

Etapas principales:

A) Comprensión del problema y definición de la métrica

- Objetivo: estimar $P(y = 1 | \mathbf{x})$ para priorizar clientes.
- Métricas para validar (ejemplos):
 - AUC-ROC (capacidad de ordenación por probabilidad).
 - Log-loss / entropía cruzada (calidad probabilística).
 - Accuracy/Precision/Recall si se fija un umbral de decisión.

- Decidir criterio operativo: seleccionar top- K clientes o seleccionar clientes con $\hat{p} \geq \tau$.

B) Preparación e integración de datos

- Dataset histórico (año anterior): contiene $\{id, x_1, x_2, x_3, y\}$.
- Dataset actual: contiene $\{id, x_1, x_2, x_3\}$.
- Separar identificador (no se usa como variable predictiva) y dejarlo solo para enlazar resultados.

C) Limpieza y preprocesado

- Tratamiento de valores ausentes:
 - imputación simple (media/mediana) o reglas de negocio (p.ej. saldo 0 si cuenta inexistente) según proceda.
- Detección de valores extremos:
 - opcionalmente winsorizar/capar para robustez si hay outliers muy fuertes.
- Escalado:
 - Para Gaussian NB no es estrictamente necesario, pero si hay magnitudes muy dispares y problemas numéricos, puede ayudar un reescalado o transformar con log:

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

(esto suele estabilizar distribuciones de saldos, que suelen ser asimétricas).

D) Partición de entrenamiento y validación (testeo y validación)

- Dividir el histórico en train/valid (por ejemplo 80/20) o usar validación cruzada.
- Mantener proporción de clases (estratificación) si la aceptación es poco frecuente.

E) Entrenamiento del modelo

- Ajustar Gaussian Naive Bayes en el conjunto de entrenamiento:
 - estimar $P(y)$, μ_{jc} y σ_{jc}^2 para cada clase c y variable j .

F) Evaluación y selección del umbral de decisión

- Calcular en validación: AUC, log-loss, etc.
- Elegir umbral τ según objetivo:
 - Maximizar retorno esperado (si se conoce beneficio/coste).
 - Controlar tasa de falsos positivos (no “molestar” a demasiados clientes con baja probabilidad).
- Si la campaña es “top- K ”: ordenar por \hat{p} y escoger los K mayores.

G) Entrenamiento final

- Reentrenar el modelo con todo el histórico (train+valid) tras fijar decisiones (preprocesado, hiperparámetros, umbral).

H) Aplicación al año actual (inferencia)

- Aplicar el mismo preprocesado a los datos actuales.
- Para cada cliente actual:
 - calcular $\hat{p}_i = P(y = 1 | \mathbf{x}_i)$.
- Generar salida interpretable para negocio:
 - tabla con id , \hat{p} , y opcionalmente explicaciones simples basadas en comparación de medias por clase: “saldo depósitos alto respecto a aceptadores” / “fondos muy bajos comparado con aceptadores”, etc.

I) Monitorización (tras campaña)

- Una vez se obtengan respuestas reales, medir:
 - calibración de probabilidades (si $\hat{p} \approx 0.3$, ~30% deberían aceptar).
 - deriva de datos (cambios en distribución de saldos).
- Incorporar los nuevos datos etiquetados al histórico para el siguiente año.

Paso 3: Parámetros e hiperparámetros: cuáles y cómo influyen

En Gaussian Naive Bayes el entrenamiento estima parámetros del modelo ($\mu_{jc}, \sigma_{jc}^2, P(y = c)$) automáticamente, pero sí hay elementos ajustables que influyen en el comportamiento:

a) Suavizado de varianza (hiperparámetro tipo ϵ)

- En implementaciones habituales (p.ej., scikit-learn) existe un término de estabilización para evitar varianzas casi cero:

$$\sigma_{jc,\text{ajustada}}^2 = \sigma_{jc}^2 + \epsilon$$

• Influencia:

- Si ϵ es muy pequeño, el modelo puede volverse demasiado “seguro” (probabilidades extremas) si alguna variable tiene varianza estimada muy baja.
- Si ϵ aumenta, se suaviza la densidad, reduciendo sobreajuste y mejorando estabilidad numérica, pero puede perder discriminación.

b) Transformaciones/preprocesado como “hiperdecisiones” Aunque no son hiperparámetros internos del modelo, en la práctica se ajustan como parte del sistema:

- Transformación logarítmica $x'_j = \log(1 + x_j)$:
 - Influencia: acerca las distribuciones a una forma más gaussiana, reduce asimetría y el efecto de outliers, lo que puede mejorar el ajuste del supuesto normal.
- Tratamiento de outliers (capado):

- Influencia: evita que pocos clientes con saldos extremos distorsionen μ y σ^2 .
- c) Umbral de decisión τ (parámetro operativo)
 - Si el sistema convierte probabilidad en “seleccionar/no seleccionar”,