AAI.EX.20230207

Ejercicios elaborados con fines educativos, inspirados en los contenidos evaluados en el exámen del 07/02/2023 (convocatoria Feb-2023) 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.20230207.T.1

Enunciado AAI.EX.20230207.T.1
La técnica conocida como LLE (Local Linear Embedding) se caracteriza por ser:

a) Un método lineal de reducción de dimensionalidad basado en proyecciones globales.

b) Un método lineal que preserva relaciones lineales globales entre instancias.

c) Un método no lineal de reducción de dimensionalidad que preserva relaciones locales y funciona especialmente bien con datos en variedades curvas como el "rollo suizo".

d) Un método no lineal basado en proyecciones ortogonales.

Solución AAI.EX.20230207.T.1
Respuesta correcta: **c)**

Justificación: LLE es una técnica **no lineal** que preserva las relaciones locales entre vecinos cercanos, siendo muy adecuada para datos que viven en variedades no lineales.

AAI.EX.20230207.T.2

Enunciado AAI.EX.20230207.T.2
¿Con qué objetivo se utiliza el conjunto de prueba (test) en un flujo estándar de aprendizaje automático?

a) Para estimar el error de generalización antes del despliegue en producción.

b) Para elegir modelos y ajustar hiperparámetros.

c) Para ambas cosas.

d) Para ninguno de los fines anteriores.

Solución AAI.EX.20230207.T.2
Respuesta correcta: **a)**

Justificación: el conjunto de test se reserva para **evaluar el rendimiento final** del modelo; no debe usarse para selección ni ajuste, que se hacen con el conjunto de validación.

AAI.EX.20230207.T.3

Enunciado AAI.EX.20230207.T.3
¿Por qué es importante escalar las variables de entrada al usar Máquinas de Vectores de Soporte (SVM)?

a) Para reducir el sobreajuste.

b) Porque SVM es invariante a la escala.

c) Para reducir el subajuste.

d) Para evitar que las variables con valores pequeños queden infravaloradas frente a otras de mayor escala.

Solución AAI.EX.20230207.T.3
Respuesta correcta: **d)**

Justificación: SVM depende de distancias y productos escalares; si no se escalan los datos, las características con valores grandes dominan el modelo.

AAI.EX.20230207.T.4

Enunciado AAI.EX.20230207.T.4
En un árbol de decisión, ¿cómo suele compararse la impureza de Gini de un nodo hijo con la de su nodo padre?

a) Normalmente es mayor.

b) Normalmente es menor.

c) Siempre es menor.

d) No existe relación entre ambas.

Solución AAI.EX.20230207.T.4
Respuesta correcta: **b)**

Justificación: los criterios de división buscan **reducir la impureza**, por lo que, en general, los nodos hijos son más puros que su padre (aunque no siempre estrictamente).

AAI.EX.20230207.T.5

Enunciado AAI.EX.20230207.T.5
¿Cuáles de las siguientes son razones habituales para aplicar reducción de dimensionalidad?

a) Acelerar y mejorar algoritmos posteriores.

b) Aumentar el coste computacional.

c) Facilitar la interpretación de los datos transformados.

d) a) y c) son correctas.

Solución AAI.EX.20230207.T.5
Respuesta correcta: **d)**

Justificación: reducir dimensionalidad puede **mejorar eficiencia**, reducir ruido y facilitar la **visualización e interpretación**.

AAI.EX.20230207.T.6

Enunciado AAI.EX.20230207.T.6
¿Qué algoritmos de clustering se basan en la detección de regiones de alta densidad?

a) BIRCH y K-means.

b) Mean-Shift y DBSCAN.

c) Mean-Shift y K-means.

d) DBSCAN y BIRCH.

Solución AAI.EX.20230207.T.6
Respuesta correcta: **b)**

Justificación: **DBSCAN** y **Mean-Shift** identifican clusters como regiones densas separadas por zonas de baja densidad.

AAI.EX.20230207.T.7

Enunciado AAI.EX.20230207.T.7
En un MLP con 10 entradas, una capa oculta de 50 neuronas y una capa de salida de 3 neuronas, ¿qué dimensiones tiene la matriz de salida Y?

a) 3×50

b) $M \times 3$, siendo M el tamaño del lote

c) $3 \times M$, siendo M el tamaño del lote

d) 3×10

Solución AAI.EX.20230207.T.7
Respuesta correcta: **b)**

Justificación: cada fila de Y corresponde a una instancia del lote y cada columna a una neurona de salida, por lo que $Y \in \mathbb{R}^{M \times 3}$.

AAI.EX.20230207.T.8

Enunciado AAI.EX.20230207.T.8
En la clase `KernelDensity` del paquete `sklearn.neighbors`, ¿cuál es el kernel que se utiliza por defecto?

a) `tophat`

b) `epanechnikov`

c) `exponential`

d) `gaussian`

Solución AAI.EX.20230207.T.8
Respuesta correcta: **d)**

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

AAI.EX.20230207.T.9

Enunciado AAI.EX.20230207.T.9
¿Bajar siempre en la dirección de mayor pendiente para llegar al fondo de un

valle es una analogía clásica de qué algoritmo de optimización?

a) Descenso del gradiente.

b) Algoritmo voraz.

c) Optimización aleatoria.

d) Algoritmo genético.

Solución AAI.EX.20230207.T.9
Respuesta correcta: **a)**

Justificación: el descenso del gradiente avanza iterativamente en la dirección de máxima pendiente descendente para minimizar una función.

AAI.EX.20230207.T.10

Enunciado AAI.EX.20230207.T.10
En un problema de clasificación multiclase con pocos datos, atributos independientes y distribuciones normales, ¿qué algoritmo es más adecuado si se prioriza una inferencia muy rápida?

a) SVM.

b) Árboles de decisión.

c) Naive Bayes.

d) Redes neuronales.

Solución AAI.EX.20230207.T.10
Respuesta correcta: **c)**

Justificación: **Naive Bayes** tiene inferencia extremadamente rápida y funciona bien bajo hipótesis de independencia y normalidad de las características.

AAI.EX.20230207.D

AAI.EX.20230207.D.1

Puntuación máxima: 5 puntos Extensión máxima orientativa: 2 caras

Enunciado AAI.EX.20230207.D.1
Una empresa dedicada a la comercialización mayorista de cítricos fija semanalmente el precio de venta de sus productos según el tipo de fruta: naranjas, mandarinas, limones y pomelos. Para cursos anteriores se dispone de un histórico

semanal que incluye, entre otros, el tipo de cítrico, la semana del año, el volumen producido, el productor, características físicas del fruto (como tamaño medio y color) y el precio final por kilogramo.

La política de precios de la empresa establece un único precio semanal por tipo de producto, de modo que todos los productores de una misma categoría reciben el mismo precio en una semana determinada, con independencia de su volumen de producción.

La empresa quiere desarrollar una herramienta que aprenda a partir de los datos históricos y sea capaz de proponer los precios para el año en curso. No es un requisito que el sistema sea explicable, pero sí que genere las predicciones de forma eficiente incluso con grandes volúmenes de datos.

Para el año actual se dispone de registros sin información del precio por kilo y sin conocer explícitamente el tipo de cítrico al que pertenecen.

Proponga uno o varios procesos de aprendizaje automático que permitan estimar el precio semanal por producto, aun cuando la categoría no esté disponible inicialmente. Justifique si sería necesario, o no, inferir primero la categoría del producto y explique el motivo.

Solución AAI.EX.20230207.D.1

Planteamiento:

Histórico (años pasados): categoría (4 clases), semana, kilos, productor, diámetro medio, color, precio/kg.
Regla de negocio: el precio depende solo de (categoría, semana); productor y kilos no cambian el precio "oficial" (aunque pueden aparecer en el histórico).
Año actual: tenemos semana, kilos, productor, diámetro, color, pero NO tenemos categoría ni precio.
Objetivo: proponer precio/kg por semana y categoría, siendo rápido con muchos datos.

Idea clave:

Como el precio "real" lo fija la empresa por (categoría, semana), lo más limpio es aprender dos cosas:

1) **Inferir la categoría** a partir de (diámetro, color, etc.)

2) **Predecir el precio** a partir de (semana, categoría inferida)

Esto es un pipeline de dos etapas (clasificación + regresión o tabla de precios).

Proceso 1 (recomendado): pipeline en 2 pasos:

```
**Paso A: modelo para predecir la categoría (clasificación supervisada)**
    Entradas:
```

```
    - diámetro medio (numérico)
    - color (categórico o numérico según cómo venga)
    - opcional: productor (si captura "especialización" de fruta), aunque puede introducir s
    - NO usar precio (sería fuga de información)

    Salida:
    - $\text{categoría} \in \{naranja, mandarina, limón, pomelo\}$

    Modelos típicos (rápidos y buenos en tabular):
    - Gradient Boosted Trees (p. ej. LightGBM/CatBoost o HistGradientBoostingClassifier)
    - Random Forest si se quiere algo simple
    - Si color es texto/categórico, CatBoost suele ir muy bien sin mucha ingeniería

    Entrenamiento:
    - split por años (si hay varios) o validación cruzada
    - métricas: accuracy y F1 macro (por si hay clases desbalanceadas)

**Paso B: calcular el precio por (categoría, semana)**
    Aquí hay dos alternativas:

    B1) **Tabla de precios (lo más coherente con la política de la empresa)**
        - Del histórico, agrupar por (categoría, semana) y estimar un precio "oficial":
        - media o mediana del precio/kg de esa semana y categoría
        - opcional: suavizado temporal (media móvil) para reducir ruido
        - Para 2023: precio_pred = tabla[cat_predicha, semana]

        Ventajas:
        - muy rápido en producción (solo lookup)
        - consistente con la regla de negocio
        - robusto aunque haya muchísimos registros por productor

    B2) **Modelo de regresión con features (categoría, semana)**
        - Entradas: semana (numérico/cíclico) + categoría (one-hot)
        - Salida: precio/kg
        - Modelo: Gradient Boosted Trees Regressor o regresión lineal con one-hot

        Ventajas:
        - generaliza mejor a semanas con pocos datos
        - permite suavizar y extrapolar ligeramente

    En la práctica, B1 suele ser suficiente y más "industrial" si hay histórico completo por

    **Predicción en el año actual**
    1) Con cada registro del año actual: predecir categoría con el clasificador.
    2) Asignar precio con la tabla (o el regresor) usando esa categoría y la semana.
    3) Si hay muchos registros, se puede:
```

```
    - predecir categoría para todos y luego agregar por (semana, categoría) si se quieren pr
```

Proceso 2 (alternativo): modelo único que predice precio sin categoría explícita:

```
Entradas:
- semana, diámetro, color (y opcional productor)
Salida:
- precio/kg

Modelo:
- Gradient Boosted Trees Regressor

Esto evita el paso de clasificación, pero tiene dos problemas:
- el modelo aprende una mezcla "implícita" de categorías (menos control)
- puedes acabar con precios inconsistentes con la regla "solo depende de semana y categoría'

Se puede usar si la clasificación es difícil o la categoría es ambigua, pero no es lo más al
```

Proceso 3 (si no hay etiquetas de categoría fiables o faltan en histórico): clustering + mapeo:

```
Si por algún motivo la categoría histórica fuese incompleta:

1) Agrupar productos en 4 clusters usando (diámetro, color) (K-means / GMM).
2) Asignar a cada cluster una categoría aproximada mirando qué categoría domina en histórico
3) Usar tabla de precios por (cluster, semana).

Es menos preciso que supervisado si sí tienes categoría en histórico.

**¿Es necesario inferir inicialmente la categoría?**

Para cumplir exactamente el objetivo "precio por semana y categoría" y mantener coherencia c

- el precio depende explícitamente de la categoría,
- sin categoría no puedes saber qué fila de la tabla (semana, categoría) usar,
- además te permite controlar errores: si la categoría tiene baja confianza, puedes marcar e

Matiz práctico:

- Si el negocio acepta "precio estimado" sin reportar categoría, podrías usar el modelo únic
- Pero si necesitan precio "por categoría" como salida formal, el paso de clasificación es l
```

Detalles prácticos para robustez (cortos):

- Codificar semana como variable cíclica: $sin(2\pi \cdot semana/52)$ y $cos(2\pi \cdot semana/52)$ (útil si usas un modelo lineal; en árboles es menos necesario).
- Usar mediana por (categoría, semana) para que outliers no distorsionen el precio.

- Añadir un umbral de confianza en la predicción de categoría: si $P(\text{cat})$ es baja, enviar a revisión o aplicar un precio "conservador".