

AAI.EX.20200901

Ejercicios elaborados con fines educativos, inspirados en los contenidos evaluados en el exámen del 01/09/2020 (convocatoria Sep-2020) 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.20200901.T.1

Enunciado AAI.EX.20200901.T.1

¿Cuáles son las dos tareas de aprendizaje supervisado más habituales?

- a) Regresión y clasificación.
- b) Regresión y agrupamiento.
- c) Reducción de dimensionalidad y clasificación.
- d) Ninguna de las anteriores.

Solución AAI.EX.20200901.T.1

Respuesta correcta: a)

Justificación:

En aprendizaje supervisado se dispone de etiquetas. Las dos tareas más comunes son la **clasificación** (salidas discretas) y la **regresión** (salidas continuas).

AAI.EX.20200901.T.2

Enunciado AAI.EX.20200901.T.2

2. ¿Qué enfoque puede utilizarse para dividir clientes en varios segmentos?

- a) Si no se conocen los grupos de antemano, se puede aplicar un algoritmo de clustering.
- b) Si se conocen los grupos deseados y se dispone de ejemplos etiquetados, se puede usar clasificación.
- c) Las opciones a) y b) son correctas.
- d) Ninguna de las opciones anteriores es correcta.

Solución AAI.EX.20200901.T.2**Respuesta correcta:** c)**Justificación:**

La segmentación puede abordarse como **aprendizaje no supervisado** (clustering) o como **aprendizaje supervisado** cuando los grupos están previamente definidos y etiquetados.

AAI.EX.20200901.T.3**Enunciado AAI.EX.20200901.T.3**

¿Garantizan todos los métodos de descenso del gradiente llegar al mismo modelo final?

- a) Sí, siempre.
- b) Sí, si el problema es convexo.
- c) Sí, si el problema es convexo y la tasa de aprendizaje no es demasiado grande.
- d) A lo anterior hay que añadir que, en algunos métodos, la tasa de aprendizaje debe disminuirse progresivamente.

Solución AAI.EX.20200901.T.3**Respuesta correcta:** d)**Justificación:**

En problemas convexos existe un único mínimo global, pero para alcanzarlo de forma estable es necesario un learning rate adecuado y, en métodos como descenso de gradiente estocástico (SGD), suele ser necesario reducirlo gradualmente.

AAI.EX.20200901.T.4**Enunciado AAI.EX.20200901.T.4**

¿Qué implementaciones de SVM en scikit-learn presentan menor coste computacional para grandes conjuntos de datos?

- a) LinearSVC.
- b) SVC.
- c) SGDClassifier.
- d) Las opciones a) y c) son correctas.

Solución AAI.EX.20200901.T.4**Respuesta correcta:** d)**Justificación:**

SVC con kernels tiene un coste computacional elevado. **LinearSVC** y **SGD-Classifier** están optimizados para problemas lineales y escalan mucho mejor con grandes volúmenes de datos.

AAI.EX.20200901.T.5**Enunciado AAI.EX.20200901.T.5**

¿Cuál es la profundidad aproximada de un árbol de decisión sin restricciones entrenado con un millón de instancias y clases balanceadas?

- a) 15.
- b) 20.
- c) 25.
- d) 10.

Solución AAI.EX.20200901.T.5**Respuesta correcta:** b)**Justificación:**

La profundidad de un árbol de decisión balanceado crece aproximadamente como:

$$\log_2(n)$$

Para $n = 10^6$, se obtiene un valor cercano a 20. El ejercicio requiere que el alumno recuerde que $10^6 \approx 20^2$.

AAI.EX.20200901.T.6**Enunciado AAI.EX.20200901.T.6**

Al entrenar árboles de decisión, ¿qué efecto tiene activar la opción **presort=True**?

- a) Acelera el entrenamiento cuando el número de instancias es pequeño (del orden de miles).
- b) Reduce la velocidad de entrenamiento en todos los casos.
- c) Aumenta la velocidad de entrenamiento en todos los casos.
- d) No tiene ningún impacto en la velocidad.

Solución AAI.EX.20200901.T.6**Respuesta correcta:** a)**Justificación:**

La ordenación previa de las características puede reducir el tiempo de entrenamiento en conjuntos pequeños, pero **no escala bien** para conjuntos de datos grandes.

AAI.EX.20200901.T.7**Enunciado AAI.EX.20200901.T.7**

¿Cuál de los siguientes métodos corresponde a aprendizaje semi-supervisado?

- a) Clustering jerárquico aglomerativo.
- b) BIRCH.
- c) Propagación de etiquetas.
- d) Mean-Shift.

Solución AAI.EX.20200901.T.7**Respuesta correcta:** c)**Justificación:**

La **propagación de etiquetas** utiliza una pequeña cantidad de datos etiquetados junto con muchos datos no etiquetados, lo que la clasifica como aprendizaje semi-supervisado.

AAI.EX.20200901.T.8**Enunciado AAI.EX.20200901.T.8**

¿Qué hipótesis hacen los Modelos de Mezcla Gaussiana (GMM) sobre la forma de los clusters?

- a) Son circulares y de tamaño similar.
- b) No hacen ninguna suposición.
- c) Son elipsoidales y de tamaño y densidad similares.
- d) Son elipsoidales y pueden tener distintos tamaños y densidades.

Solución AAI.EX.20200901.T.8**Respuesta correcta:** d)**Justificación:**

Los GMM modelan los datos como una combinación de distribuciones gaus-

sianas, permitiendo clusters **elipsoidales**, con **diferentes tamaños y densidades**.

AAI.EX.20200901.T.9

Enunciado AAI.EX.20200901.T.9

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 pesos W_h y el vector de sesgos b_h de la capa oculta?

- a) W_h es 10×50 y b_h tiene tamaño 10.
- b) W_h es 50×10 y b_h tiene tamaño 10.
- c) W_h es 10×50 y b_h tiene tamaño 50.
- d) W_h es 3×50 y b_h tiene tamaño 10.

Solución AAI.EX.20200901.T.9

Respuesta correcta: c)

Justificación:

En una capa oculta de un MLP el cálculo es:

$$h = f(XW_h + b_h)$$

donde:

- X : vector de entrada
- W_h : matriz de pesos de la capa oculta
- b_h : vector de sesgos (*bias*) de la capa oculta
- h : salidas de las neuronas ocultas

Suponiendo que hubiera una sola instancia, para que la multiplicación matricial tenga sentido:

$$(1 \times 10) W_h = (1 \times 50)$$

se tiene que cumplir que:

- $W_h \in \mathbb{R}^{10 \times 50}$

Por otro lado, tiene que haber un sesgo por cada neurona oculta. Además, por la fórmula tiene que tener el mismo ancho que W_h .

- $b_h \in \mathbb{R}^{50}$

AAI.EX.20200901.T.10

Enunciado AAI.EX.20200901.T.10

¿Qué función de activación es la más adecuada en la capa de salida de un MLP para un problema de regresión?

- a) ReLU.
- b) Sigmoide.
- c) Softmax.
- d) Ninguna función de activación.

Solución AAI.EX.20200901.T.10

Respuesta correcta: d)

Justificación:

En regresión se desean salidas reales sin restricciones, por lo que la capa de salida suele ser **lineal**, es decir, sin función de activación.

AAI.EX.20200901.D

AAI.EX.20200901.D.1

Puntuación máxima: 5 puntos

Extensión máxima orientativa: 2 caras

Enunciado AAI.EX.20200901.D.1

Explica el funcionamiento de los árboles de decisión aplicados a problemas de clasificación. En tu respuesta deben tratarse los siguientes aspectos:

- Descripción general del método y del proceso de construcción del árbol, así como sus principales usos.
- Algoritmo de entrenamiento empleado para generar el árbol a partir de los datos.
- Cómo se realizan las predicciones una vez entrenado el modelo.
- Concepto de pureza de un nodo, principales medidas utilizadas y comparación entre ellas.
- Forma en que el modelo permite estimar probabilidades de pertenencia a cada clase.
- Principales hiperparámetros del modelo y su influencia en el comportamiento del árbol.

Solución AAI.EX.20200901.D.1

Los árboles de decisión son modelos de aprendizaje supervisado que representan un proceso de decisión mediante una estructura jerárquica de nodos.

Cada nodo interno corresponde a una condición sobre una característica, cada rama representa el resultado de dicha condición y cada hoja asigna una clase (o distribución de clases). En clasificación, su objetivo es dividir el espacio de características en regiones lo más homogéneas posible respecto a la variable objetivo.

Son especialmente útiles por su **interpretabilidad**, su capacidad para manejar datos numéricos y categóricos y porque apenas requieren preprocesamiento (no necesitan escalado de variables).

Proceso de creación del árbol y usos:

La construcción del árbol se realiza de forma recursiva, comenzando en el nodo raíz con todo el conjunto de entrenamiento. En cada paso se selecciona la característica y el umbral que mejor separan los datos según un criterio de pureza. El proceso continúa hasta que se cumple un criterio de parada (por ejemplo, profundidad máxima o nodos puros).

Usos principales:

- Clasificación interpretable en dominios donde la explicación es importante.
- Modelos base para métodos de conjunto como Random Forests o Gradient Boosting.
- Problemas con relaciones no lineales entre variables.

Algoritmo de entrenamiento:

El algoritmo de entrenamiento es un enfoque **greedy y top-down** (de arriba hacia abajo):

1. En un nodo dado, se evalúan todas las posibles divisiones candidatas.
2. Para cada división se calcula la reducción de impureza obtenida.
3. Se elige la división que maximiza dicha reducción.
4. El conjunto de datos se divide según esa regla y el proceso se repite en cada nodo hijo.

Este procedimiento no garantiza encontrar el árbol globalmente óptimo, pero es computacionalmente eficiente y funciona bien en la práctica.

Predicciones:

Para clasificar una nueva instancia, se recorre el árbol desde la raíz hasta una hoja siguiendo las condiciones de los nodos internos. La clase asignada es normalmente la clase mayoritaria en la hoja alcanzada.

Este proceso es rápido, ya que su coste es proporcional a la profundidad del árbol.

Pureza de un nodo y comparación de medidas:

La **pureza de un nodo** mide cuán homogéneas son las clases de las muestras que contiene. Cuanto más puro es un nodo, más concentradas están las muestras en una sola clase.

Las medidas más comunes son:

- **Índice Gini**

Mide la probabilidad de clasificar incorrectamente una instancia si se asigna aleatoriamente según la distribución de clases del nodo. Favorece particiones donde una clase domina claramente.

- **Entropía**

Mide el nivel de desorden o incertidumbre del nodo. Se basa en la teoría de la información y penaliza más las distribuciones equilibradas entre clases.

Comparación:

- Ambas medidas suelen producir árboles muy similares.
- Gini es ligeramente más rápido de calcular.
- Entropía tiende a generar árboles algo más equilibrados. En la práctica, la diferencia entre ambas es pequeña y la elección rara vez es crítica.

Estimación de probabilidades:

Además de predecir una clase, los árboles de decisión pueden estimar **probabilidades de pertenencia**. En una hoja, la probabilidad de una clase se calcula como la proporción de muestras de esa clase respecto al total de muestras del nodo.

Estas probabilidades son fáciles de interpretar, aunque pueden ser poco fiables en hojas con pocas muestras. Por este motivo, a menudo se prefiere usar árboles dentro de métodos de conjunto para mejorar la calibración.

Hiperparámetros principales:

Los hiperparámetros controlan la complejidad del árbol y ayudan a evitar el sobreajuste:

- **max_depth**: profundidad máxima del árbol.
- **min_samples_split**: número mínimo de muestras para dividir un nodo.
- **min_samples_leaf**: número mínimo de muestras en una hoja.
- **max_features**: número de características consideradas en cada división.
- **criterion**: medida de pureza utilizada (Gini o entropía).

Árboles muy profundos tienden a sobreajustar, mientras que árboles demasiado restringidos pueden infraajustar. El ajuste adecuado de estos hiperparámetros es clave para obtener buen rendimiento.

Conclusión:

Los árboles de decisión para clasificación son modelos flexibles, interpretables y fáciles de usar. Su entrenamiento se basa en divisiones sucesivas que maximizan la pureza de los nodos, y permiten tanto predicciones de clase como estimaciones de probabilidad. Aunque pueden sobreajustar si no se controlan, constituyen un pilar fundamental del aprendizaje automático y la base de muchos modelos más avanzados.