**Métodos de ensamble**

1)\_ Si ha entrenado cinco modelos diferentes en los mismos datos de entrenamiento y todos logran una precisión del 95%, ¿existe alguna posibilidad de combinar estos modelos para obtener mejores resultados? Si es así, ¿cómo? Si no, ¿por qué?

Respuesta: si, es posible mediante los métodos de ensamblaje, ya que la idea principal es combinar las predicciones de múltiples modelos para obtener una predicción final más precisa y robusta. Las técnicas son:

* Voating: los modelos individuales votan por la clase de cada instancia y la clase con más votos se selecciona como la predicción final. Tenemos la "hard voting", donde cada modelo tiene el mismo peso en la votación, o "soft voting" donde los modelos pueden tener diferentes pesos según su confianza en sus predicciones
* Stacking: se entrena un modelo adicional llamado "meta-modelo" para combinar las predicciones de los modelos base. El meta-modelo se entrena para aprender como combinar las predicciones de los modelos base de manera óptima. El proceso de stacking generalmente implica dividir los datos de entrenamiento en conjuntos de entrenamiento y validación, donde los modelos base se entrenan en el conjunto de entrenamiento y se generan predicciones en el conjunto de validación. Luego, estas predicciones se utilizan como características para entrenar el meta-modelo en el conjunto de validación.
* Boosting: se basa en entrenar modelos en serie, donde cada modelo se enfoca en corregir los errores del modelo anterior. Los modelos base se entrenan de forma secuencial, dándole más peso a las instancias clasificadas incorrectamente por los modelos anteriores. El modelo final es una combinación ponderada de todos los modelos base.

2)\_ ¿Cuál es la diferencia entre los clasificadores de votación hard y de votación soft?

Respuesta:

* Votación hard: los clasificadores base emiten sus predicciones y se selecciona la clase que obtiene la mayoría de votos (predicciones individuales). En otras palabras, la clase que es elegida por la mayoría de los clasificadores se considera como la predicción final. Si hay empate, se puede utilizar alguna estrategia de desempate, como seleccionar la clase con el índice más bajo. Este es adecuado para problemas de clasificación binaria o multiclase.
* Votación soft: utiliza la información de probabilidad proporcionada por los clasificadores base para tomar una decisión más informada y precisa, en donde se calcula la probabilidad promedio de cada clase para una instancia en particular, y la clase con la mayor probabilidad promedio se selecciona como la predicción final. Es especialmente útil cuando los clasificadores base pueden proporcionar estimaciones de probabilidad confiables.

\_ En términos de rendimiento, la votación soft tiende a ser más precisa que la votación hard, ya que aprovecha la información adicional proporcionada por las probabilidades de los clasificadores base. Sin embargo, la votación soft puede ser más costosa computacionalmente, ya que implica el cálculo y promedio de las probabilidades.

3)\_ ¿Es posible acelerar el entrenamiento de un conjunto de bagging distribuyéndolo en varios servidores? ¿Qué pasa con los conjuntos de pasting, los conjuntos de boosting, los Random Forest o los ensambles Stacking?

Respuesta: Si, es posible acelerar el entrenamiento de conjuntos de bagging distribuyéndolos en varios servidores. El bagging es un enfoque que consiste en entrenar múltiples modelos de manera independiente en subconjuntos de datos generados aleatoriamente y luego combinar sus predicciones. Cada modelo se puede entrenar en un servidor diferente, lo que permite realizar el entrenamiento en paralelo y acelerar el proceso.

\_ El mismo enfoque se puede aplicar a conjuntos de pasting, que es similar al bagging pero sin reemplazo de las muestras. En este caso, los modelos también se pueden entrenar en servidores diferentes para aprovechar la distribución y acelerar el proceso.

\_ En el caso de los conjuntos de boosting, como AdaBoost o Gradient Boosting, el entrenamiento no se presta tan fácilmente a la distribución en múltiples servidores. Esto se debe a que los modelos se construyen de forma secuencial, donde cada modelo se entrena en función de los errores del modelo anterior. Sin embargo, existen variaciones como XGBoost y LightGBM que permiten la distribución y paralelización del entrenamiento en múltiples servidores.

\_ En el caso de los Random Forest, cada árbol se puede entrenar de forma independiente, lo que facilita la distribución en múltiples servidores para acelerar el proceso de entrenamiento.

\_ En cuanto a Stacking, donde los modelos de nivel inferior se entrenan en base a las predicciones de otros modelos, la distribución en múltiples servidores puede ser más complicada debido a la dependencia entre los modelos. Sin embargo, se pueden aplicar estrategias de paralelización para el entrenamiento de los modelos de nivel inferior si es posible dividir los datos o las características de manera adecuada.

4)\_ ¿Cuál es el beneficio de la evaluación out-of-bag (OOB)?

Repuesta: el proceso de Bagging implica entrenar múltiples modelos utilizando diferentes subconjuntos de datos seleccionados aleatoriamente con reemplazo. Esto significa que para cada modelo individual, hay instancias que no se incluyeron en su conjunto de entrenamiento debido a la selección aleatoria. Estas instancias no utilizadas se conocen como instancias out-of-bag (OOB). La evaluación OOB aprovecha estas instancias OOB para evaluar el rendimiento del modelo. Después de entrenar los modelos, se puede calcular el rendimiento promedio utilizando las instancias OOB. Estas instancias OOB se consideran como un conjunto de validación interno y proporcionan una estimación imparcial del rendimiento del modelo en datos no vistos.

\_ El beneficio clave de la evaluación OOB es:

* Permite obtener una estimación imparcial y sin necesidad de un conjunto de validación adicional del rendimiento del modelo en datos no vistos.
* Evita la necesidad de un conjunto de validación adicional. Tradicionalmente, separamos los datos en conjuntos de entrenamiento y validación, lo que reduce la cantidad de datos disponibles para entrenar el modelo. Con la evaluación OOB, podemos aprovechar todas las instancias de datos para el entrenamiento y obtener una estimación confiable del rendimiento del modelo sin requerir un conjunto de validación separado.
* Proporciona una evaluación más robusta del rendimiento del modelo, ya que utiliza instancias que no han sido vistas por el modelo durante el entrenamiento. Esto puede ayudar a evitar el sobreajuste y proporcionar una estimación más realista del rendimiento en datos nuevos.

5)\_ ¿Qué hace que los Extra-Trees sean más aleatorios que los Random Forest regulares? ¿Cómo puede esta aleatoriedad adicional ayudar? ¿Son los Extra-Trees más lentos o más rápidos que los Random Forest regulares?

Respuesta: Los Extra-Trees son más aleatorios que los RandomForest regulares en dos aspectos:

* Aleatoriedad en la selección de características: tanto los RandomForest como los Extra-Trees seleccionan características aleatorias para cada división en los árboles de decisión. Sin embargo, los Extra-Trees toman esta aleatoriedad un paso más allá al permitir divisiones aleatorias en puntos de corte predefinidos en lugar de buscar la mejor división.
* Aleatoriedad en los datos de entrenamiento: mientras que los RandomForest utilizan subconjuntos de datos de entrenamiento muestreados con reemplazo (bootstrapping), los Extra-Trees también introducen aleatoriedad adicional al considerar subconjuntos de datos de entrenamiento más pequeños y seleccionados aleatoriamente para cada árbol individual.

\_ La aleatoriedad adicional en los Extra-Trees puede proporcionar algunos beneficios:

* Mayor diversidad: al ser más aleatorios en la selección de características y puntos de corte, los Extra-Trees generan arboles más diversos en comparación con los RandomForest. Esta diversidad puede ayudar a reducir el sesgo del modelo y aumentar la capacidad de generalización.
* Reducción del sobreajuste: la aleatoriedad adicional puede ayudar a reducir el sobreajuste al disminuir la dependencia en características específicas y puntos de corte óptimos. Esto permite que los Extra-Trees se adapten mejor a los datos ruidosos o con características irrelevantes, evitando asi un ajuste excesivo a los detalles de los datos de entrenamiento.

\_ En términos de velocidad, los Extra-Trees suelen ser más rápidos que los Random Forest regulares. Esto se debe a que la aleatoriedad adicional en los Extra-Trees permite construir arboles de decisión más rápidamente, ya que no es necesario buscar los mejores puntos de corte en cada nodo. Sin embargo, la velocidad puede depender de la implementación especifica y del tamaño y la complejidad del conjunto de datos.

6)\_ Si su conjunto de AdaBoost no se ajusta lo suficientemente bien a los datos de entrenamiento, ¿qué hiperparámetros debe ajustar y cómo?

Respuesta: los modelos de base en AdaBoost son clasificadores débiles, como arboles de decisión débiles, utilizados individualmente en cada iteración del algoritmo para construir un conjunto de clasificadores más fuertes. Si el conjunto de AdaBoost no se ajusta lo suficientemente bien a los datos de entrenamiento, hay varios hiperparámetros que se pueden ajustar para mejorar su rendimiento, en donde los más importantes son:

* Numero de estimadores (n\_estimators):  determina el número de modelos de base (por ejemplo, arboles de decisión débiles) que se utilizan en el conjunto de AdaBoost. Aumentar el número de estimadores puede mejorar el ajuste del modelo, pero también puede aumentar el riesgo de sobreajuste.
* Tasa de aprendizaje (learning rate): controla la contribución de cada modelo de base al conjunto. Un valor más bajo de la tasa de aprendizaje reduce la contribución de cada modelo y puede ayudar a evitar el sobreajuste.
* Complejidad del modelo base: el tipo y la complejidad del modelo de base utilizado en AdaBoost también pueden afectar el rendimiento. En general, los modelos más simples, como los árboles de decisión débiles, se utilizan comúnmente en AdaBoost, pero se puede aumentar la complejidad del modelo base.
* Ponderación de instancias (sample\_weight): AdaBoost asigna pesos a las instancias en cada iteración para enfocarse en las instancias difíciles de clasificar.

7)\_ Si su conjunto de Gradient Boosting sobre ajusta el conjunto de entrenamiento, ¿debería aumentar o disminuir la tasa de aprendizaje?

Respuesta: si el conjunto de Gradient Boosting sobre ajusta el conjunto de entrenamiento, es recomendable disminuir la tasa de aprendizaje. La tasa de aprendizaje controla la contribución de cada árbol en el conjunto a la corrección de los errores de los árboles anteriores. Una tasa de aprendizaje alta permite que cada árbol tenga un impacto más significativo en el modelo final, lo que puede llevar a un sobreajuste, especialmente si se combinan con un gran número de estimadores.

\_ Al reducir la tasa de aprendizaje, se reduce la importancia de cada árbol en el conjunto, lo que puede ayudar a evitar el sobreajuste. Una tasa de aprendizaje más baja permite que el modelo se ajuste de manera más gradual y refinada, lo que puede mejorar la generalización y reducir el sobreajuste. Pero si la tasa de aprendizaje se establece demasiado baja, el modelo puede requerir un numero considerablemente mayor de estimadores para converger y puede llevar más tiempo entrenar el modelo, por lo que se debe buscar un equilibrio.

**Métodos de ensamble**

1)\_ ¿Cuáles son las principales motivaciones para reducir la dimensionalidad de un conjunto de datos? ¿Cuáles son las principales desventajas?

Respuesta: las principales motivaciones para reducir la dimensionalidad de un conjunto de datos son:

* Simplificación: se simplifica la representación del conjunto de datos, lo que facilita su comprensión y visualización. Esto es útil cuando se trabaja con conjuntos de datos complejos con muchas variables.
* Eficiencia computacional: se reduce la cantidad de cálculos necesarios para analizar y procesar los datos, ahorrando tiempo y recursos computacionales.
* Eliminación de características irrelevantes: es posible identificar y eliminar características que no aportan información relevante para el análisis. Esto ayuda a eliminar el ruido y la redundancia en los datos, mejorando la calidad de los resultados.
* Mejora de la generalización: se reduce el riesgo de sobreajuste, ya que se disminuye la complejidad del modelo al eliminar características redundantes o irrelevantes. Esto puede mejorar la capacidad del modelo para generalizar y aplicarse a nuevos datos.

\_ Algunas desventajas en la reducción de dimensionalidad son:

* Perdida de información: es inevitable que se pierda cierta información. Al eliminar características, se puede perder parte de la variabilidad de los datos, lo que puede afectar la precision y el rendimiento del modelo.
* Interpretabilidad: es posible que la interpretación de los resultados se vuelva más difícil. Al eliminar características, se puede perder la capacidad de comprender y explicar las relaciones entre las variables originales.
* Riesgo de sesgo: si la reducción de dimensionalidad se realiza de manera inapropiada, puede introducir sesgos en el análisis. Es importante seleccionar y aplicar métodos de reducción de dimensionalidad de manera cuidadosa y considerar las implicaciones en los resultados.

2)\_ ¿A qué se denomina la maldición de la dimensionalidad?

Respuesta: la "maldición de la dimensionalidad" se refiere a los desafíos y problemas que surgen al tratar con conjuntos de datos de alta dimensionalidad (conjuntos de datos que tienen un gran número de características o variables). A medida que el número de características o variables en un conjunto de datos aumenta, se dice que el conjunto de datos sufre de la maldición de la dimensionalidad. La maldición de la dimensionalidad puede manifestarse de varias formas:

* Esparsidad de los datos: A medida que aumenta la dimensionalidad, la cantidad de datos necesarios para tener una cobertura representativa de todo el espacio de características aumenta exponencialmente. Por ende es más probable que los datos sean escasos y dispersos, lo que dificulta el análisis y el modelado preciso.
* Aumento de la complejidad: con un mayor número de características, la complejidad del modelo aumenta. Esto puede llevar a un mayor riesgo de sobreajuste, donde el modelo se ajusta demasiado a los datos de entrenamiento y no puede generalizar bien a nuevos datos.
* Reducción de la relación señal-ruido: a medida que aumenta la dimensionalidad, la relación señal-ruido en los datos puede deteriorarse. Las características irrelevantes o ruido pueden dominar sobre las características relevantes, lo que dificulta la identificación de patrones y relaciones significativas en los datos.
* Necesidad de más datos: debido a la esparsidad de los datos en alta dimensionalidad, se requiere una cantidad mucho mayor de datos para construir modelos precisos y confiables. Esto puede ser costoso en términos de tiempo y recursos necesarios para recopilar y etiquetar grandes volúmenes de datos.

3)\_ Una vez que se ha reducido la dimensionalidad de un conjunto de datos, ¿es posible revertir la operación? Si es así, ¿cómo? Si no, ¿por qué no?

Respuesta: la reducción de la dimensionalidad implica la transformación de un conjunto de datos de alta dimensionalidad a uno de menor dimensionalidad. Dependiendo del método utilizado para la reducción de la dimensionalidad, es posible o no revertir la operación y recuperar la dimensionalidad original Existen dos enfoques principales para la reducción de la dimensionalidad:

* Selección de características (Feature Selection): se selecciona un subconjunto de características del conjunto de datos original. Esto implica descartar algunas características y conservar solo las más relevantes o informativas. En este caso, es posible revertir la operación si se mantiene un registro de las características seleccionadas.
* Extracción de características (Feature Extraction): se crean nuevas características que son combinaciones lineales o no lineales de las características originales. Algunos algoritmos comunes para la extracción de características son PCA (Análisis de Componentes Principales) y t-SNE (t-Distributed Stochastic Neighbor Embedding). En este caso, la transformación es irreversible, ya que se pierde información durante la extracción de características, por lo que no se puede recuperar la dimensionalidad original de forma precisa.

4)\_ ¿Se puede utilizar PCA para reducir la dimensionalidad de un conjunto de datos altamente no lineal?

Respuesta: si, si se puede usar PCA para reducir la dimensionalidad de un conjunto de datos, incluso cuando estos datos son altamente no lineales. Sin embargo, es importante tener en cuenta que PCA es un método lineal y no captura de manera eficiente las estructuras no lineales en los datos. Cuando se aplica PCA a un conjunto de datos no lineales, es posible que se pierda información importante y que la representación resultante no capture correctamente las relaciones no lineales entre las variables. Esto significa que la reducción de dimensionalidad con PCA en conjuntos de datos altamente no lineales puede no ser tan efectiva como en conjuntos de datos lineales. En tales casos, es recomendable considerar técnicas de reducción de dimensionalidad no lineales, como t-SNE.

* En un conjunto de datos no lineal, la relación entre las variables no se puede describir mediante una línea recta o una función lineal. En cambio, las relaciones pueden ser curvas, irregulares o complejas.

5)\_ Suponga que realiza PCA en un conjunto de datos de 1,000 dimensiones, estableciendo la relación de varianza explicada en un 95%. ¿Cuántas dimensiones tendrá el conjunto de datos resultante?

Respuesta: si realizamos PCA en un conjunto de datos de 1,000 dimensiones y establecemos una relación de varianza explicada en un 95%, el conjunto de datos resultante tendrá un número reducido de dimensiones. El objetivo de PCA es encontrar las componentes principales que capturan la mayor parte de la varianza en los datos. Al establecer una relación de varianza explicada del 95%, significa que el conjunto de datos resultante retendrá el 95% de la varianza total de los datos originales. Creamos el objeto PCA, ajustamos y transformamos con fit y despues vemos el número de dimensiones

6)\_ ¿En qué casos utilizaría PCA simple, PCA incremental, PCA aleatorio o kernel PCA?

Respuesta: el uso de diferentes variantes de PCA depende del contexto y los requisitos específicos del problema:

* PCA simple: implementación estándar de PCA y se utiliza cuando el conjunto de datos se puede cargar completamente en la memoria. Es adecuado para conjuntos de datos de tamaño moderado y es útil cuando se requiere una reducción de dimensionalidad lineal.
* PCA incremental: se utiliza cuando el conjunto de datos es demasiado grande para caber en la memoria o cuando se requiere un procesamiento en tiempo real. Este divide el conjunto de datos en lotes más pequeños y realiza el cálculo de PCA de manera incremental en cada lote.
* PCA aleatorio: se utiliza cuando el conjunto de datos es muy grande y no se puede cargar en la memoria. En lugar de calcular PCA en todo el conjunto de datos, selecciona aleatoriamente una muestra más pequeña y realiza PCA en esa muestra. Proporciona una aproximación rápida de PCA en grandes conjuntos de datos, aunque puede haber una pequeña perdida de precision.
* Kernel PCA: se utiliza cuando los datos tienen una estructura no lineal y se requiere una reducción de dimensionalidad no lineal. Transforma los datos a un espacio de mayor dimensión utilizando una función de kernel y luego aplica PCA en ese espacio de mayor dimensión. Se utiliza además en problemas como el reconocimiento de imágenes y la clasificación de texto.

7)\_ ¿Cómo se puede evaluar el rendimiento de un algoritmo de reducción de dimensionalidad en su conjunto de datos?

Respuesta: para evaluar el rendimiento de un algoritmo de reducción de dimensionalidad en nuestro conjunto de datos, podemos utilizar diversas métricas y técnicas. A continuación, se presentan algunas opciones comunes:

* Preservación de la varianza: evaluar cuanta varianza o información se ha preservado despues de la reducción de dimensionalidad. Esto se puede medir utilizando la relación de varianza explicada (RVE), en donde, cuanto mayor sea la varianza preservada, mejor será el rendimiento del algoritmo.
* Visualización de datos: ver los datos en un espacio de menor dimensión y verificar si se mantiene la estructura o la separabilidad de las clases, mediante gráficos de dispersión o gráficos en 2D/3D, para observar cómo se distribuyen los datos despues de la reducción de dimensionalidad.
* Evaluación de modelos posteriores: evaluamos el rendimiento del modelo resultante de la reducción en términos de métricas de evaluación, como accuracy, recall, F1-score, etc. Si el modelo posterior mejora su rendimiento con la reducción de dimensionalidad, indica que el algoritmo ha sido efectivo.
* Tiempo de ejecución: si el algoritmo es capaz de reducir eficientemente la dimensionalidad de los datos en un tiempo razonable, puede considerarse como un buen desempeño.

8)\_ ¿Tiene sentido encadenar dos algoritmos de reducción de dimensionalidad diferentes?

Respuesta: Si tiene sentido encadenar dos algoritmos de reducción de dimensionalidad diferentes, y a veces puede ser beneficioso. Esta técnica se conoce como encadenamiento de dimensiones o apilamiento de dimensiones. La idea principal de esto es que cada algoritmo puede capturar diferentes aspectos de la estructura de los datos. Al aplicar múltiples algoritmos en secuencia, podemos aprovechar las fortalezas individuales de cada uno para obtener una representación de menor dimensionalidad más efectiva.

* Por ejemplo, primero podemos aplicar PCA para reducir la dimensionalidad inicial y luego aplicar otro algoritmo de reducción de dimensionalidad, como t-SNE, para capturar relaciones no lineales o estructuras más complejas. Esta combinación de algoritmos puede ofrecer una representación de los datos con una estructura más significativa y una mejor capacidad de visualización.

\_ Sin embargo, es importante tener en cuenta que encadenar múltiples algoritmos de reducción de dimensionalidad puede aumentar la complejidad computacional y potencialmente introducir cierto grado de pérdida de información en cada paso.

**Técnicas de aprendizaje no supervisado**

1)\_ ¿Cómo definirías el agrupamiento (clustering)? ¿Puedes nombrar algunos algoritmos de clustering?

Respuesta: el agrupamiento es una técnica no supervisada para organizar conjuntos de datos en grupos con características similares. Algunos algoritmos pueden ser Kmenas, DBSCAN, Gaussian Mixture Models.

2)\_ ¿Cuáles son algunas de las principales aplicaciones de los algoritmos de clustering?

Respuesta: algunas de las principales aplicaciones de los algoritmos de clustering son reconocimiento de patrones, compresión de datos, análisis de redes sociales y análisis de textos.

3)\_ Describe dos técnicas para seleccionar el número adecuado de clusters al utilizar KMeans.

Respuesta: las técnicas son:

* Elbow Method: consiste en calcular la suma de las distancias al cuadrado de cada punto dentro de su respectivo cluster y luego trazar el gráfico de la suma de las distancias en función del número de clusters. Se busca punto donde las distancias se estabilizan, es decir, donde la curva se aplana más.
* Silhouette Coefficient: se calcula para cada punto y representa la similitud del punto con su cluster en comparación con otros clusters cercanos. Se elige el número de clusters que maximice el coeficiente.

4)\_ ¿Qué es la propagación de etiquetas (label propagation)? ¿Por qué la implementarías y cómo?

Respuesta: la propagación de etiquetas es un algoritmo semi supervisado para asignar etiquetas a instancias no etiquetadas. Se implementa cuando el conjunto de datos es limitado y se quiere etiquetar más datos a partir de los pocos etiquetados que existen. Para implementarlo se preparan los datos con algunas instancias etiquetadas, se construye la matriz de afinidad (captura similitud entre instancias cercanas), se propaga las etiquetas (se etiqueta según la matriz), se evalúa y ajustan parámetros.

5)\_ ¿Puedes nombrar dos algoritmos de clustering que puedan escalar a conjuntos de datos grandes? ¿Y dos que busquen regiones de alta densidad?

Respuesta:

* Clustering para conjuntos de grandes datos: KMeans++ (inicializa centroides eficientemente), DBSCAN (agrupa según la densidad de los puntos).
* Clustering para regiones de alta densidad: DBSCAN, OPTICS (extensión de DBSCAN, ordena según conectividad y densidad relativa de instancias).

6)\_ ¿Puedes pensar en un caso de uso donde el aprendizaje activo sería útil? ¿Cómo lo implementarías?

Respuesta: el aprendizaje activo es útil en situaciones donde tenemos un conjunto de datos etiquetados limitado y queremos maximizar la precisión del modelo a través de una selección inteligente de las instancias a etiquetar. Un caso de ejemplo donde se utilizaría es en la clasificación de documentos por categorías. Para implementarlo primero procesamos los datos, luego entrenamos un clasificador inicial, despues realizamos el aprendizaje activo, por siguiente actualizamos el modelo y por último repito los últimos 2 pasos hasta terminar.

7)\_ ¿Cuál es la diferencia entre la detección de anomalías y la detección de novedades?

Respuesta: aunque ambos se centran en identificar instancias atípicas la diferencia entre ellos es que la detección de anomalías se centra en identificar desviaciones negativas con respecto al patrón general de los datos existentes, mientras que la detección de novedades busca identificar instancias que son completamente nuevas o desconocidas en relación con los datos existentes.

8)\_ ¿Qué es un modelo de Gaussian Mixture? ¿Para qué tareas se puede usar?

Respuesta: un modelo de Gaussian Mixture se basa en la suposición de que los datos provienen de una combinación de varias distribuciones gaussianas. Se estima la estructura subyacente de los datos mediante la asignación de una probabilidad a cada punto de datos para pertenecer a cada componente gaussiano. Estas probabilidades pueden utilizarse para realizar tareas como la clasificación, la agrupación o la generación de nuevos datos.

9)\_ ¿Puedes nombrar dos técnicas para encontrar el número adecuado de clusters al usar un modelo de Gaussian Mixture?

Respuesta: las técnicas para encontrar el número adecuado de clusters al usar un modelo de Gaussian Mixture son:

* Criterio de Información de Akaike (AIC) y Criterio de Información Bayesiano (BIC).
* Elbow Method.