TEMA-0-1-2-3-4-5.pdf

BeaSalga

Aprenentatge Computacional

3º Grado en Ingeniería de Datos

Escuela de Ingeniería
Universidad Autónoma de Barcelona

TEMA 0:

Diferencia entre IA fuerte y IA débil:

La diferencia entre la inteligencia artificial fuerte (IA fuerte) y la inteligencia artificial débil (IA débil) radica en el nivel de autonomía y capacidad cognitiva. La IA fuerte busca replicar la inteligencia humana completa, incluyendo conciencia y comprensión del mundo, mientras que la IA débil se centra en realizar tareas específicas sin tener una comprensión global. En la práctica, la mayoría de las aplicaciones de inteligencia artificial en la actualidad son ejemplos de IA débil, diseñadas para tareas específicas.

¿Qué es un modelo Naive?

Un modelo ingenuo (naive) se caracteriza por hacer suposiciones simplificadas para facilitar su implementación y comprensión, aunque estas suposiciones pueden no reflejar completamente la complejidad del mundo real. El término "ingenuo" no se utiliza en el sentido de falta de sofisticación, sino más bien en el sentido de simplicidad asumida para facilitar el análisis o la implementación.

¿Diferencia entre modelos generativos y modelos discriminativos?

En resumen, la principal diferencia radica en el enfoque y la tarea principal de cada tipo de modelo. Los modelos generativos se centran en entender cómo se generan los datos, mientras que los modelos discriminativos se centran en discriminar entre diferentes clases basándose en las características observadas. Ambos tipos de modelos tienen aplicaciones específicas en el campo del aprendizaje automático.

TEMA 1 : EVALUACIÓN DE MODELOS

¿Cuándo nos interesa priorizar la precisión de un modelo?

Cuando los Falsos Positivos son Costosos: Si los errores de clasificación positivos incorrectos son costosos o tienen consecuencias significativas, entonces es crucial priorizar la precisión. Por ejemplo, en un sistema de detección de fraudes, donde un falso positivo podría bloquear una cuenta legítima, la precisión sería una métrica importante.

¿Cuándo nos interesa priorizar el recall de un modelo?

Cuando los Falsos Negativos son Costosos: Si los errores de clasificación negativos incorrectos son costosos o tienen consecuencias significativas, entonces es importante priorizar el recall. En aplicaciones médicas, por ejemplo, donde la detección de enfermedades es crítica, se prefiere tener menos falsos negativos, incluso si esto significa más falsos positivos.

¿Qué métrica sería más relevante si queremos minimizar la posibilidad de que un paciente con cáncer sea enviado a casa sin ser detectado, incluso a expensas de tener algunos casos de falsas alarmas? ¿Debería ser alta o baja?

En este caso, la métrica más relevante sería la sensibilidad (recall). Queremos minimizar los falsos negativos, por lo que la sensibilidad debe ser alta.

En el contexto de la detección de cáncer, ¿qué métrica sería más crítica si queremos asegurarnos de que la mayoría de los casos reales de cáncer sean identificados, incluso si esto significa tener más casos de falsas alarmas? ¿Debería ser alta o baja?

La sensibilidad (recall) sería la métrica más crítica en este caso, y queremos que sea alta para asegurarnos de identificar la mayoría de los casos reales de cáncer, aunque pueda haber más falsos positivos.

Supongamos que el costo de no detectar un caso de cáncer es significativamente mayor que el costo de realizar pruebas adicionales o consultas innecesarias. ¿Qué métrica deberíamos priorizar en este escenario? ¿Debería ser alta o baja?

En este escenario, la sensibilidad (recall) debería ser priorizada y queremos que sea alta, ya que el costo de no detectar un caso de cáncer real es alto.

Si nos preocupa tanto la detección de casos positivos como evitar diagnósticos erróneos en pacientes sanos, ¿qué métrica podría proporcionar un equilibrio entre ambas consideraciones? ¿Debería ser alta o baja?

En este caso, podríamos considerar utilizar el F1-score, que proporciona un equilibrio entre precisión y recall. Queremos un F1-score alto para lograr un equilibrio adecuado.

Imagina que, en este sistema de detección de cáncer, se prefiere errar en el lado de la precaución y enviar a más pacientes a pruebas adicionales para confirmar o descartar la presencia de cáncer. ¿Qué métrica sería más relevante en este caso? ¿Debería ser alta o baja?**

En este caso, podríamos priorizar la precisión, ya que queremos minimizar los falsos positivos. La precisión debería ser alta para evitar diagnósticos incorrectos en pacientes sanos.

¿Qué puede suponer un sesgo y una varianza muy elevada?

Un sesgo elevado indica falta de capacidad del modelo para capturar la complejidad de los datos (underfitting), mientras que una varianza elevada sugiere un modelo demasiado complejo que se ajusta demasiado a los datos de entrenamiento (overfitting).

¿Cómo puede ayudarnos el CROSS VALIDATION? ¿Qué motivo puede llevarnos a utilizarlo?**

La validación cruzada ayuda a evaluar el rendimiento del modelo al dividir los datos en conjuntos de entrenamiento y prueba múltiples. Se utiliza para evitar problemas como el sobreajuste y proporciona una estimación más robusta del rendimiento del modelo.

¿**Qué hace que LEAVE-ON-OUT resulte muy rentable?**

Leave-One-Out (LOO) Cross Validation utiliza un solo ejemplo como conjunto de prueba y el resto como conjunto de entrenamiento en cada iteración. Puede ser rentable en conjuntos de datos pequeños, ya que utiliza la mayoría de los datos para entrenar en cada iteración.

¿Cómo afecta reducir el threshold en la especificidad y sensibilidad? ¿Y aumentarlo?**

Reducir el umbral aumenta la sensibilidad y disminuye la especificidad. Aumentar el umbral tiende a aumentar la especificidad y disminuir la sensibilidad.

¿Qué significa si tenemos una curva ROC que es la recta y=x (x>0), es decir, la diagonal positiva del primer cuadrante?

Una curva ROC que es una línea diagonal indica rendimiento aleatorio, donde el modelo no puede discriminar entre las clases.

¿Qué significa tener un AUC-ROC de: a) aproximadamente 1, b) 0.5, c) 0?

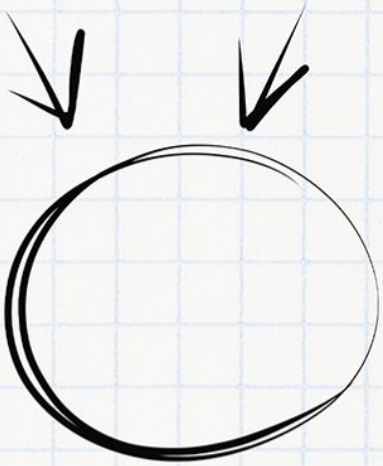
- a) Aproximadamente 1: Indica un modelo excelente con capacidad de discriminación perfecta.
- b) 0.5: Indica rendimiento aleatorio o nulo.
- c) 0: Indica que el modelo está prediciendo las clases de manera inversa.

Imagínate aprobando el examen

Necesitas tiempo y concentración

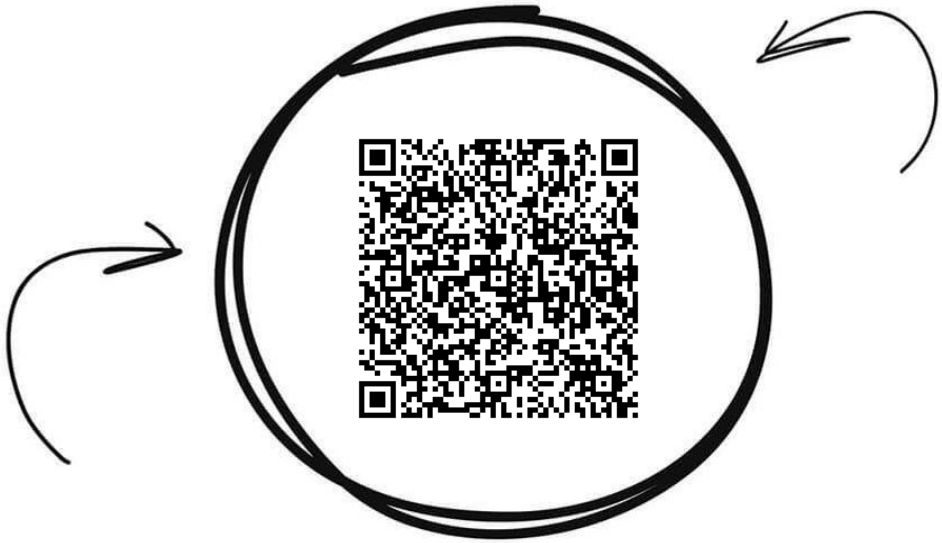
Planes	PLAN TURBO	PLAN PRO	PLAN PRO+
Descargas sin publi al mes	10	40	80
Elimina el video entre descargas	✓	✓	✓
Descarga carpetas	✗	✓	✓
Descarga archivos grandes	✗	✓	✓
Visualiza apuntes online sin publi	✗	✓	✓
Elimina toda la publi web	✗	✗	✓
Precios Anual	0,99 € / mes	3,99 € / mes	7,99 € / mes

Ahora que puedes conseguirlo, ¿Qué nota vas a sacar?



WUOLAH

Aprenentatge Computacional



Banco de apuntes de la

WUOLAH

Comparte estos flyers en tu clase y consigue más dinero y recompensas

1. Imprime esta hoja
2. Recorta por la mitad
3. Coloca en un lugar visible para que tus compis puedan escanar y acceder a apuntes
4. Llévate dinero por cada descarga de los documentos descargados a través de tu QR

¿Qué significa tener un AUC-ROC de 0.7?

Significa que la probabilidad de que se clasifique bien es del 70%.

TEMA 2: KNN

¿Qué pasa si K es demasiado pequeño? Por otro lado, ¿qué pasa si es demasiado grande?

Si K es demasiado pequeño (por ejemplo, 1), el modelo será muy sensible al ruido y a los valores atípicos, pudiendo conducir a un sobreajuste (overfitting). Si K es demasiado grande, el modelo puede perder sensibilidad a las pequeñas variaciones en los datos, lo que podría conducir a un subajuste (underfitting)

Con un K bajo, el modelo tiene una varianza alta (sensible al ruido), pero un sesgo bajo (sensible a las pequeñas variaciones). Con un K alto, el modelo tiene una varianza baja (menos sensible al ruido), pero un sesgo alto (puede perder detalles en los datos).

Explica en tres pasos el algoritmo k-NN.

1. **Paso 1: Almacenamiento de Datos.** Guarda los puntos con sus respectivas etiquetas en el conjunto de entrenamiento.

2. **Paso 2: Cálculo de Distancia.** Calcula la distancia entre el punto de consulta y todos los puntos del conjunto de entrenamiento (utilizando una medida de distancia como la euclidiana).

3. **Paso 3: Clasificación/Regresión.** Selecciona los K vecinos más cercanos y predice la etiqueta (clasificación) o el valor (regresión) basándose en la mayoría o el promedio de los vecinos, respectivamente.

Enumera tres formas que pueden servirnos para calcular la distancia entre dos puntos para el algoritmo k-NN.

1. **Distancia Euclidiana.**: Ventaja de la Distancia Euclidiana:

   Ventaja: Versatilidad Geométrica: La distancia euclidiana tiene una interpretación geométrica intuitiva

   Desventaja: sensibilidad a escala

2. **Distancia de Manhattan.**

   Ventaja: Robusta delante de Outliers

   Desventaja: Restricción a movimientos ortogonales

3. **Distancia Minkowski.**

   Ventaja: Flexibilidad paramétrica

   Desventaja: Sensibilidad al parámetro p

¿Qué problema nos puede dar utilizar la distancia euclidiana en el algoritmo k-NN? ¿Qué solución podemos aplicar?

La distancia euclidiana puede ser sensible a características con diferentes escalas, dándoles un peso excesivo. Una solución es normalizar las características antes de calcular la distancia, asegurándose de que todas tengan la misma influencia en el cálculo.

Qué tipo de aprendizaje es aquel en el que se desconocen las etiquetas/clases de los datos?

Este tipo de aprendizaje se conoce como "aprendizaje no supervisado". Aquí, el modelo debe encontrar patrones y estructuras en los datos sin tener etiquetas predefinidas.

¿Qué significa el parámetro K y cómo puede afectar a las predicciones del algoritmo?

El parámetro K en k-NN representa el número de vecinos considerados en la predicción. Un K pequeño puede hacer que el modelo sea sensible al ruido, mientras que un K grande puede hacer que se pierda detalle y se genere un modelo más suavizado. La selección de K depende del tipo de datos y el problema específico.

¿Cómo afecta la elección de un valor adecuado de k en k-NN?

La elección de un valor adecuado de k en k-NN tiene un impacto en la sensibilidad del modelo. Un valor pequeño de k puede hacer que el modelo sea más sensible al ruido y a los valores atípicos, llevando a un sobreajuste. En contraste, un valor grande de k suaviza el modelo, haciendo que sea menos sensible a variaciones locales, lo que puede llevar a un subajuste. En resumen, elegir un valor adecuado de k es crucial para lograr un equilibrio entre sesgo y varianza.

¿En qué situaciones k-NN puede tener dificultades o no funcionar tan bien?

k-NN puede tener dificultades en situaciones donde la dimensionalidad es alta debido a la maldición de la dimensionalidad. También puede no funcionar bien en presencia de características irrelevantes o ruido significativo en los datos. Además, cuando la distribución de las clases es desigual, k-NN puede sesgarse hacia la clase dominante, afectando la capacidad de generalización del modelo.

¿Cuáles son los desafíos computacionales asociados con la implementación de k-NN en conjuntos de datos grandes?

Los desafíos computacionales en k-NN con conjuntos de datos grandes radican en el cálculo de distancias entre el punto de consulta y todos los puntos del conjunto de entrenamiento. Este proceso puede volverse costoso en términos de tiempo y recursos computacionales a medida que el tamaño del conjunto de datos aumenta. Estrategias como el uso de estructuras de datos eficientes, como árboles KD, pueden mitigar estos desafíos.

Pregunta: ¿Dirías que k-NN es adecuado para conjuntos de datos con muchas dimensiones? ¿Por qué?

No, generalmente no se considera que k-NN sea adecuado para conjuntos de datos con muchas dimensiones. Esta afirmación se basa en el fenómeno conocido como la "maldición de la dimensionalidad". A medida que la dimensionalidad aumenta, la distancia entre los puntos se vuelve menos significativa, ya que todos los puntos tienden a estar equidistantes entre sí. Esto afecta la capacidad de k-NN para identificar vecinos más cercanos de manera efectiva, lo que puede conducir a un rendimiento deficiente en espacios de alta dimensionalidad. En tales casos, se prefieren algoritmos más avanzados y técnicas de reducción de dimensionalidad para mejorar la eficacia del modelo.

Pregunta: ¿Por qué se dice que k-NN es un modelo de aprendizaje perezoso?

Respuesta: Se dice que k-NN es un modelo de aprendizaje perezoso porque posterga el proceso de aprendizaje hasta el momento de la predicción. En lugar de aprender un modelo durante la fase de entrenamiento, k-NN almacena los datos de entrenamiento y realiza la clasificación o regresión en el momento de la predicción, basándose en la proximidad de los vecinos más cercanos en ese momento.

Pregunta: ¿Qué efecto tiene utilizar como peso para los k neighbors w=1/d^2 en lugar de solamente w=d?

Utilizar pesos ponderados como w=1/d^2 en lugar de simplemente w=d asigna más importancia a los vecinos más cercanos y reduce la influencia de los vecinos más lejanos en la predicción. Esto puede ser beneficioso cuando se desea dar más peso a los vecinos cercanos, ya que reflejará más fielmente la contribución relativa de cada vecino en función de su inverso cuadrado de la distancia. Sin embargo, la elección del peso depende del problema específico y la interpretación deseada.

TEMA 3: REGRESIÓN

¿Qué pasa si el learning rate es demasiado alto? ¿D'altra banda, si és massa baix? I per què?

Si el learning rate es demasiado alto, el algoritmo de optimización puede converger rápidamente, pero existe el riesgo de pasar por alto el mínimo global y oscilar alrededor de él, o incluso diverger. Si es demasiado bajo, la convergencia puede ser lenta y el algoritmo puede quedarse atrapado en mínimos locales. La elección adecuada del learning rate es crucial para garantizar una convergencia eficiente y estable.

¿Qué pasa en el Gradient Descent cuando nos aproximamos a 0?

Cuando nos aproximamos a un mínimo local o global, la derivada de la función de costo se acerca a cero. En este punto, el algoritmo de Gradient Descent disminuirá los pasos y ajustará lentamente los parámetros. Si se ha alcanzado un mínimo, el descenso de gradiente se detendrá, ya que la derivada será exactamente cero en el mínimo.

¿Porqué es tan importante la mean normalization, para tener una media de 0 y una desviación estándar de 1?

Al normalizar las características para tener una media de cero y una desviación estándar de uno, se facilita la convergencia de los algoritmos de optimización, como el Gradient Descent. Esto es especialmente relevante en algoritmos sensibles a las diferencias de escala, donde una convergencia más rápida puede mejorar la eficiencia del proceso de aprendizaje

¿Cuándo se utiliza el Gradient Descent y cuándo el Normal equation?

El Gradient Descent se utiliza comúnmente cuando hay un gran número de datos o características, y la Normal equation se prefiere cuando el conjunto de datos es pequeño ya que involucra el cálculo directo de los coeficientes. Gradient Descent es más escalable y eficiente para grandes conjuntos de datos.

De cada característica del conjunto de datos, ¿qué relación tiene el valor de su parámetro en la función de hipótesis con la influencia que supone para representar el modelo de regresión? ¿Cuanto más alto, más importante?

Sí, la relación es directa. El valor del parámetro asociado a cada característica en la función de hipótesis indica cuánto afecta esa característica específica al modelo de regresión. Cuanto mayor sea el valor del parámetro, mayor será la influencia de la característica en la predicción del modelo.

Si la función de coste es conocida en todo momento, ¿por qué el algoritmo que la minimiza es iterativo y no se aplica cálculo multivariante básico para encontrar directamente los coeficientes que minimizan el modelo?

Aunque la función de coste es conocida, el algoritmo iterativo, como el Gradient Descent, es necesario cuando se trata con conjuntos de datos grandes o modelos complejos donde el cálculo directo de los coeficientes puede ser computacionalmente costoso o incluso impracticable. El enfoque iterativo permite realizar ajustes paso a paso, lo que facilita la convergencia hacia el mínimo de la función de coste.

¿Cómo puede afectar un valor alto en la función de coste?

Un valor alto en la función de coste indica que las predicciones del modelo están lejos de las etiquetas reales. Esto puede deberse a un mal ajuste del modelo o a la presencia de outliers. Un alto costo sugiere que el modelo necesita ajustes para mejorar su rendimiento y reducir la discrepancia entre las predicciones y los valores reales.

¿Cuál es el objetivo que tenemos respecto a la función de coste? ¿Qué conseguimos cumpliéndolo?

El objetivo es minimizar la función de coste. Al hacerlo, logramos ajustar el modelo de manera óptima a los datos de entrenamiento, mejorando su capacidad predictiva en nuevos datos. Minimizar la función de coste implica encontrar los valores óptimos de los parámetros del modelo que mejor se ajustan a los datos observados.

¿Cuál es el impacto de los outliers en la función de coste en la regresión lineal?

Los outliers pueden tener un impacto significativo en la función de coste en la regresión lineal, ya que los cuadrados en la fórmula del MSE amplifican las diferencias entre las predicciones y los valores reales. Los outliers pueden aumentar considerablemente el valor de la función de coste y afectar negativamente la calidad del ajuste del modelo.

¿Qué quiere decir si la función de coste de un modelo es 0?

Si la función de coste de un modelo es 0, significa que las predicciones del modelo coinciden exactamente con las etiquetas reales en el conjunto de entrenamiento. Esto indica un ajuste perfecto del modelo a los datos de entrenamiento, pero también puede ser un indicativo de sobreajuste si no generaliza bien a nuevos datos.

¿Cuál es la importancia de evaluar el rendimiento de un modelo mediante la medición de su error?

Calcular el error de un modelo es fundamental para evaluar su rendimiento y su capacidad para hacer predicciones precisas. Proporciona una medida cuantitativa de qué tan bien el modelo se ajusta a los datos de entrenamiento o a nuevos datos. Evaluar el error permite identificar áreas de mejora, ajustar los parámetros del modelo y garantizar que el modelo sea capaz de generalizar de manera efectiva a datos no vistos, contribuyendo así a un desempeño óptimo y confiable en aplicaciones del mundo real.

¿Cuál es la importancia del learning rate en el descenso de gradiente?

El learning rate es crucial en el descenso de gradiente, ya que determina el tamaño de los pasos que el algoritmo toma en la dirección opuesta al gradiente. Un learning rate adecuado asegura la convergencia eficiente del modelo, mientras que valores demasiado grandes o pequeños pueden llevar a problemas como convergencia lenta, oscilaciones o incluso divergencia.

¿Cómo puede afectar una tasa de aprendizaje demasiado grande?

Una tasa de aprendizaje demasiado grande puede causar que el descenso de gradiente converja rápidamente, pero existe el riesgo de pasar por alto el mínimo global y oscilar alrededor de él, o incluso diverger. Puede resultar en pasos demasiado grandes que "sobrepasan" el punto óptimo, afectando negativamente la convergencia del modelo.

¿Cómo puede afectar una tasa de aprendizaje demasiado pequeña?

Una tasa de aprendizaje muy pequeña puede conducir a una convergencia lenta del modelo, ya que los pasos son demasiado pequeños para alcanzar eficientemente el mínimo global. Esto puede resultar en tiempos de entrenamiento prolongados y la posibilidad de quedar atrapado en mínimos locales subóptimos.

¿Qué significa que el descenso de gradiente ha convergido?

La convergencia en el descenso de gradiente significa que el algoritmo ha alcanzado un mínimo de la función de coste y ha ajustado los parámetros del modelo de manera óptima. En este punto, los cambios en los parámetros son muy pequeños, y el modelo ha alcanzado un rendimiento aceptable.

¿Cuándo es útil aplicar un algoritmo de descenso de gradiente?

El descenso de gradiente es útil en problemas de optimización donde se busca minimizar una función de coste. Es especialmente útil en conjuntos de datos grandes y para modelos con muchos parámetros, ya que su enfoque iterativo facilita la adaptabilidad a diversas condiciones de optimización.

¿Qué pasa en el punto en el que la derivada del gradiente es 0? Nos interesa?

Cuando la derivada del gradiente es cero, estamos en un punto crítico donde la pendiente es horizontal. Este punto podría ser un mínimo local, máximo local o un punto de inflexión. Nos interesa si es un mínimo local, ya que representa una posible solución al problema de optimización que buscamos resolver.

Nombra tres algoritmos que podemos utilizar para minimizar funciones de coste.

Algunos algoritmos para minimizar funciones de coste son:

- Gradient Descent (Descenso de Gradiente)
- Stochastic Gradient Descent (Descenso de Gradiente Estocástico)
- L-BFGS (Limited-memory Broyden-Fletcher-Goldfarb-Shanno)

¿En relación con el tamaño de los datos, cuándo es mejor utilizar un Gradient Descent Algorithm o un Batch Gradient Descent? ¿Por qué?

El Batch Gradient Descent es más adecuado para conjuntos de datos pequeños debido a que calcula el gradiente utilizando todos los datos en cada iteración. El Gradient Descent (mini-batch) es preferible para conjuntos de datos grandes, ya que utiliza subconjuntos aleatorios de datos en cada iteración, lo que mejora la eficiencia computacional y la escalabilidad.

¿En relación con la convergencia, cuándo es mejor utilizar un Gradient Descent Algorithm o un Batch Gradient Descent?

La elección entre Gradient Descent y Batch Gradient Descent en términos de convergencia depende del problema. Batch Gradient Descent generalmente converge más suavemente ya que utiliza el gradiente preciso en cada iteración. Sin embargo, Gradient Descent puede ser más rápido en converger en ciertos casos debido a su naturaleza estocástica.

¿Cuál es el propósito de incluir términos polinomiales en la regresión polinomial?

Incluir términos polinomiales en la regresión polinomial permite modelar relaciones no lineales entre las características y la variable objetivo. Esto aumenta la flexibilidad del modelo para capturar patrones más complejos en los datos y mejorar la capacidad de ajuste a curvas no lineales.

¿Cómo afecta el grado del polinomio a la flexibilidad del modelo en la regresión polinomial?

Aumentar el grado del polinomio aumenta la flexibilidad del modelo. Grados más altos permiten al modelo ajustarse más a los detalles y variaciones en los datos, pero también pueden llevar a sobreajuste si no se controlan adecuadamente. La elección del grado del polinomio es crucial para lograr un equilibrio entre sesgo y varianza.

¿Cómo puede ayudar la validación cruzada a la elección del grado del polinomio?

La validación cruzada ayuda a evaluar el rendimiento del modelo con diferentes grados de polinomios en conjuntos de datos no utilizados durante el entrenamiento. Permite seleccionar el grado del polinomio que generaliza mejor a nuevos datos y evita el sobreajuste, contribuyendo así a una elección más informada del modelo.

¿Cuál es el propósito principal de normalizar características en un conjunto de datos?

El propósito principal de normalizar características es asegurar que todas las características contribuyan de manera equitativa al modelo, independientemente de sus escalas originales. La normalización facilita la convergencia de algoritmos, mejora la interpretación de los coeficientes y ayuda a evitar problemas relacionados con diferencias de escala.

¿Cuáles son las ventajas de normalizar características antes de aplicar algoritmos de aprendizaje automático?

Normalizar características facilita la convergencia de algoritmos, mejora la interpretación de los coeficientes, evita problemas de sensibilidad a la escala y contribuye a un rendimiento más consistente y equitativo del modelo en diferentes conjuntos de datos.

¿**En qué situaciones podría ser especialmente importante normalizar las características?**

La normalización es crucial cuando las características tienen escalas muy diferentes, en algoritmos sensibles a la escala (como el descenso de gradiente) y cuando se utilizan medidas de distancia en algoritmos como k-NN. También es esencial en problemas donde la escala de las características afecta significativamente el rendimiento del modelo.

¿Cómo afecta la normalización de características a algoritmos sensibles a la escala, como el descenso del gradiente?

La normalización evita que las características con magnitudes más grandes dominen la contribución al modelo. En algoritmos como el descenso de gradiente, la normalización facilita la convergencia al garantizar que los pasos de actualización sean proporcionales y equitativos para todas las características.

¿**Es necesario normalizar características en todos los casos o hay situaciones en las que puede ser prescindible?**

Si bien no siempre es obligatorio, la normalización suele ser recomendable, especialmente en algoritmos sensibles a la escala. En casos donde las características tienen magnitudes similares y la escala no afecta significativamente el rendimiento, la normalización puede ser prescindible.

¿**Cómo seleccionarías entre diferentes técnicas de normalización de características, según el tipo de datos y el algoritmo de aprendizaje usado?**

La elección de la técnica de normalización depende del tipo de datos y el algoritmo. La Z-score (estandarización) es común para datos con distribución normal, mientras que la escala min-max puede ser útil cuando se desea limitar el rango de valores. La elección debe basarse en el conocimiento del dominio y la naturaleza de los datos.

¿Cuál es el propósito de la ecuación normal en regresión lineal?

La ecuación normal proporciona una solución cerrada y analítica para encontrar los coeficientes óptimos en un modelo de regresión lineal. Permite calcular directamente los valores de los parámetros que minimizan la función de coste, eliminando la necesidad de un enfoque iterativo como el descenso de gradiente.

¿**Cuál es la ventaja de usar la ecuación normal en comparación con otros modelos de optimización como el descenso de gradiente? ¿Y en qué casos produce una desventaja?**

La ventaja de la ecuación normal es su eficiencia en conjuntos de datos pequeños y el hecho de proporcionar una solución analítica. Sin embargo, puede volverse computacionalmente costosa en conjuntos de datos grandes debido al cálculo de la inversa de matrices. Además, no es adecuada cuando la matriz \(X^T * X\) no es invertible o es singular.

¿En qué situaciones la ecuación normal podría ser preferible sobre el descenso de gradiente para encontrar los parámetros de un modelo de regresión lineal?

La ecuación normal es preferible en conjuntos de datos pequeños donde la inversión de matrices es computacionalmente viable. También es útil cuando se busca una solución analítica precisa y la matriz \(X^T * X\) es invertible.

¿**Qué consideraciones se deben tener en cuenta al aplicar la ecuación normal a conjuntos de datos grandes?**

Al aplicar la ecuación normal a conjuntos de datos grandes, la inversión de matrices puede volverse costosa. Es esencial verificar la invertibilidad de la matriz \(X^T * X\) y considerar alternativas como el descenso de gradiente para mejorar la eficiencia computacional en estos casos.

¿Qué pasa si \(X^T * X\) no es cuadrada o tiene determinante igual a 0? ¿Cómo lo puedes solucionar fácilmente?

Si \(X^T * X\) no es cuadrada o tiene determinante igual a 0, la matriz no es invertible, y la ecuación normal no se puede aplicar directamente. Una solución común es utilizar técnicas de regularización, como la regularización de Ridge (L2), para hacer que la matriz sea invertible y evitar problemas de singularidad.

¿Cuál es la importancia de evaluar el rendimiento de un modelo mediante la medición de su error?

Evaluar el rendimiento de un modelo mediante la medición de su error es crucial para cuantificar la calidad del modelo, identificar áreas de mejora, ajustar hiperparámetros, comparar modelos, validar la generalización y tomar decisiones fundamentadas en aplicaciones del mundo real. Proporciona una base cuantitativa para comprender cómo el modelo se desempeña en comparación con los datos reales y orienta el proceso de desarrollo hacia mejoras continuas.

TEMA 4: REGRESIÓN LOGÍSTICA

¿Qué controla el coeficiente de regularización?

El coeficiente de regularización controla la magnitud de las penalizaciones aplicadas a los parámetros del modelo durante el proceso de entrenamiento. En modelos de regresión logística, la regularización ayuda a evitar el sobreajuste al penalizar coeficientes grandes, contribuyendo así a la generalización del modelo a nuevos datos.

¿Qué implica una regularización baja o alta?

- Regularización Baja: Permite a los coeficientes del modelo tener valores más grandes, lo que puede conducir a un mayor riesgo de sobreajuste.

- Regularización Alta: Impone fuertes penalizaciones a los coeficientes, limitando su magnitud y promoviendo modelos más simples y generalizables al costo de una posible pérdida de ajuste a los datos de entrenamiento.

¿Cuáles son las diferencias en el efecto que tienen el underfitting y el overfitting en el ajuste de modelos? ¿Cómo se soluciona cada caso?

- Underfitting: Ocurre cuando el modelo es demasiado simple y no puede capturar la complejidad de los datos. Se soluciona aumentando la complejidad del modelo, agregando características o utilizando modelos más avanzados.

- Overfitting: Ocurre cuando el modelo se ajusta demasiado a los datos de entrenamiento, perdiendo capacidad de generalización. Se soluciona mediante la simplificación del modelo, reduciendo la complejidad, o mediante el uso de técnicas como la regularización.

¿Cómo afecta el overfitting al ajuste de un modelo y cómo solucionarlo?

- Efecto del Overfitting: Resulta en un modelo que se ajusta demasiado a los datos de entrenamiento y tiene dificultades para generalizar a nuevos datos.

- Solución: Reducir la complejidad del modelo, utilizar técnicas de regularización, aumentar el tamaño del conjunto de datos de entrenamiento o aplicar validación cruzada para una evaluación más robusta.

¿Cómo afecta el underfitting al ajuste de un modelo y cómo solucionarlo?

- Efecto del Underfitting: El modelo es demasiado simple y no puede capturar la estructura subyacente de los datos.

- Solución: Aumentar la complejidad del modelo, agregar características relevantes o utilizar modelos más avanzados.

¿Cómo influye la complejidad del modelo en el ajuste y la presencia de overfitting o underfitting?

- Influencia de la Complejidad: A medida que la complejidad aumenta, el modelo se vuelve más propenso a sobreajustarse a los datos de entrenamiento y menos propenso al subajuste.

- Overfitting y Underfitting: Modelos demasiado complejos pueden conducir a overfitting, mientras que modelos demasiado simples pueden resultar en underfitting.

¿Cómo podemos darnos cuenta de que tenemos overfitting fijándonos en las métricas de evaluación? Y underfitting?

- Overfitting: Se observa cuando el modelo tiene un rendimiento excelente en el conjunto de entrenamiento pero un rendimiento deficiente en datos no vistos (conjunto de prueba o validación).

- Underfitting: Se evidencia cuando el rendimiento del modelo es insatisfactorio tanto en el conjunto de entrenamiento como en datos no vistos.

¿Cómo influye el coeficiente de regularización en la complejidad del modelo de regresión logística?

El coeficiente de regularización en la regresión logística controla la magnitud de las penalizaciones aplicadas a los parámetros del modelo. Un coeficiente más alto resulta en un modelo más regularizado y menos propenso al overfitting.

¿Cómo influye el coeficiente de regularización a la tendencia de sobreajuste?

Un coeficiente de regularización más alto tiende a reducir la tendencia de sobreajuste al imponer penalizaciones más fuertes a los coeficientes del modelo, lo que favorece la simplicidad y la generalización.

¿Qué pasa si el coeficiente de regularización es demasiado bajo? ¿Y si es demasiado alto?

- Coeficiente Bajo: Puede permitir que el modelo se sobreajuste a los datos de entrenamiento.

- Coeficiente Alto: Puede llevar a un modelo demasiado regularizado que no se ajusta bien a los datos de entrenamiento.

¿Cuándo preferimos un coeficiente de regularización alto o bajo?

- Coeficiente Alto: Prefiere cuando se busca evitar el overfitting y se prefiere la simplicidad del modelo.

- Coeficiente Bajo: Prefiere cuando se permite un mayor grado de complejidad y se busca ajustar

más a los datos de entrenamiento.

¿Qué problema puede tener la regresión logística sin regularización?

Sin regularización, la regresión logística puede ser propensa al overfitting, especialmente cuando hay muchas características en el conjunto de datos. Esto puede conducir a un modelo que se ajusta demasiado a la variabilidad de los datos de entrenamiento y tiene dificultades para generalizar a nuevos datos.

¿Qué objetivo tiene la función logística (Sigmoid)?

La función logística, o Sigmoid, se utiliza en la regresión logística para transformar la salida del modelo en probabilidades. Su objetivo es mapear los valores continuos a un rango entre 0 y 1, facilitando la interpretación de las probabilidades de pertenencia a una clase.

¿Qué pasa con la decision boundary si se añade una variable predictora adicional?

La decisión boundary se convierte en un hiperplano en un espacio de mayor dimensión. La adición de una variable predictora adicional introduce una dimensión adicional en el espacio, lo que afecta la forma y la complejidad de la decision boundary.

¿Cómo afecta la inclinación de la decision boundary al rendimiento del modelo?

La inclinación de la decision boundary puede afectar la sensibilidad y especificidad del modelo. Una inclinación inadecuada puede llevar a errores de clasificación y afectar el rendimiento general del modelo.

¿Puede la decisión boundary no ser lineal en regresión logística?

Sí, la decisión boundary en la regresión logística puede ser no lineal, especialmente cuando se introducen términos no lineales o interacciones entre variables predictoras. Esto permite al modelo capturar relaciones más complejas entre las características y la variable objetivo.

¿Cómo afecta al sesgo(bias) y a la varianza la forma de la decision boundary?

Decision boundaries más complejas pueden reducir el sesgo pero aumentar la varianza, lo que puede llevar a overfitting.

¿Qué papel juega la regularización en la determinación de la forma de la decisión boundary?

La regularización influye en la forma de la decision boundary al controlar la complejidad del modelo y prevenir el overfitting, lo que puede resultar en decision boundaries más suaves.

¿La decision boundary puede cambiar si se utiliza un umbral diferente para la clasificación?

Sí, cambiar el umbral de clasificación puede desplazar la decision boundary, afectando la sensibilidad y la especificidad del modelo.

¿Por qué puede ser necesario utilizar una non-linear decision boundary en lugar de una línea recta?

Una non-linear decision boundary es necesaria cuando la relación entre las variables predictoras y la variable objetivo no puede ser bien modelada por una línea recta. Situaciones más complejas pueden requerir fronteras de decisión no lineales.

¿Cómo afecta la complejidad de la non-linear decision boundary al sesgo y a la varianza del modelo?

Decision boundaries más complejas pueden reducir el sesgo pero aumentar la varianza, incrementando el riesgo de overfitting.

¿Cómo nos puede ayudar la validación cruzada al trabajar con modelos que incluyen non-linear decision boundaries?

La validación cruzada proporciona una evaluación robusta del rendimiento del modelo al considerar múltiples particiones de datos, ayudando a evitar la sobreoptimización y proporcionando una visión más realista del rendimiento del modelo.

Encuentra una función que dé a los puntos verdes un valor más alto que a los rojos o viceversa (algoritmo, dadas estas funciones, dime cuál es mejor en este caso)

Sin las funciones mencionadas, no puedo proporcionar una evaluación precisa de cuál sería mejor. Necesitaría información específica sobre las funciones o algoritmos para hacer una comparación.

¿Qué significa que la función de coste es convexa en el modelo de regresión logística?

La convexidad implica que la función de coste tiene un único mínimo global, facilitando la convergencia de los algoritmos de optimización hacia una solución óptima.

¿Cómo influye la regularización en la función de coste en la regresión logística?

La regularización añade términos adicionales a la función de coste para penalizar parámetros grandes, controlando así la complejidad del modelo y evitando el overfitting.

¿Cuáles son algunas técnicas comunes para abordar problemas de clasificación multiclase?

Algunas técnicas comunes incluyen One-vs-One, One-vs-All (One-vs-Rest), y técnicas avanzadas como clasificadores basados en árboles, máquinas de soporte vectorial, y redes neuronales.

¿Cuál es la ventaja del enfoque One-vs-One en la clasificación multiclase?

One-vs-One construye un clasificador binario para cada par de clases, lo que puede ser eficaz en conjuntos de datos desequilibrados y proporcionar mayor robustez.

¿Cuándo preferimos usar One vs One a One vs All?

One-vs-One puede ser preferido en situaciones con conjuntos de datos desequilibrados, mientras que One-vs-All es más simple y puede funcionar bien con conjuntos de datos balanceados.

¿Cuál es el impacto de la dimensionalidad de los datos en modelos de clasificación multiclase?

El aumento de dimensionalidad puede aumentar la complejidad del problema y requerir modelos más sofisticados, pero también puede aumentar el riesgo de overfitting.

¿Cómo afecta el tamaño del conjunto de datos en la elección del modelo en clasificación multiclase?

Conjuntos de datos más grandes pueden permitir el uso de modelos más complejos y la obtención de decisiones más informadas sobre la elección del modelo.

¿Es más probable que haya overfitting en datasets grandes o pequeños? ¿Por qué?

El overfitting es más probable en datasets pequeños, ya que los modelos pueden ajustarse demasiado a la variabilidad inherente de los datos limitados.

¿Cómo podemos evitar el overfitting?

Se puede evitar el overfitting mediante técnicas como regularización, validación cruzada, aumento de datos, reducción de la complejidad del modelo, y uso de conjuntos de datos más grandes.

¿Cómo afecta la complejidad del modelo al riesgo de underfitting y overfitting?

Modelos más complejos pueden reducir el riesgo de underfitting pero aumentan el riesgo de overfitting, siendo crucial encontrar un equilibrio adecuado.

Cuando un modelo sufre overfitting, ¿cómo es la complejidad, el sesgo y la varianza?

En un modelo con overfitting, la complejidad es alta, la varianza es alta, y el sesgo puede ser bajo en el conjunto de entrenamiento, pero alto en el conjunto de prueba.

Cuando un modelo sufre underfitting, ¿cómo es la complejidad, el sesgo y la varianza?
Cuando un modelo sufre underfitting, la complejidad del modelo es baja, lo que significa que no es capaz de capturar la estructura subyacente de los datos de manera adecuada. El sesgo tiende a ser alto, ya que el modelo es demasiado simple para ajustarse a la complejidad de los datos. Por otro lado, la varianza también suele ser baja, ya que el modelo no es lo suficientemente complejo como para variar significativamente en diferentes conjuntos de datos. En resumen, el underfitting se caracteriza por un modelo insuficientemente complejo que no puede ajustarse bien a los datos.

TEMA 5: SVM

Di tres ventajas de SVM(modelo de clasificación) delante de modelos de generativos

Robustez ante Datos Ruidosos: SVM es menos afectado por datos ruidosos o atípicos, ya que su enfoque en vectores de soporte proporciona una mayor resistencia a perturbaciones en los datos.

Manejo Eficiente de Conjuntos de Datos de Alta Dimensión: SVM es eficiente en espacios de alta dimensión, superando la maldición de la dimensionalidad, gracias a la capacidad de usar kernels que facilitan la transformación de características.

Fácil Adaptación a Problemas de Clasificación Binaria: SVM destaca en problemas de clasificación binaria, simplificando la formulación del problema y logrando buen rendimiento sin la necesidad de modelar la distribución conjunta de todas las clases, a diferencia de los modelos generativos.

En qué situaciones podría ser preferible utilizar un modelo generativo SVM?

Esta pregunta también parece contener un error tipográfico o de redacción. Si se refiere a cuándo podría ser preferible utilizar un modelo generativo en lugar de un modelo SVM, se podría reformular para mayor claridad.

Diferencia principal entre linear classifier, maximal margin y support vectors?

La diferencia principal radica en que un linear classifier es un modelo que divide el espacio en clases linealmente, maximal margin se refiere a la maximización del margen entre las clases en SVM, y support vectors son los puntos de datos fundamentales para definir la decision boundary en SVM.

Qué le pasa a la decision boundary si los vectores de soporte se mueven?

Si los vectores de soporte se mueven, la decision boundary de SVM también se verá afectada. La ubicación de los vectores de soporte es crítica para la determinación de la boundary, y su movimiento puede alterar la posición y la orientación de dicha boundary.

Los valores atípicos no deberían ser vectores de soporte. Qué puede ocurrir para que lo sean?

En circunstancias normales, los valores atípicos no deberían ser vectores de soporte en SVM. Sin embargo, podrían convertirse en vectores de soporte si afectan significativamente la posición de la decision boundary y son necesarios para mantener el margen máximo.

Cómo pueden afectar los valores atípicos (outliers) a la solución de SVM?

Los valores atípicos pueden influir en la solución de SVM al afectar la posición de la decision boundary. Pueden tirar de la boundary hacia ellos, alterando el margen y la clasificación de otros puntos.

Qué tipo de margen tenemos cuando los valores atípicos son parte de los vectores de soporte? ¿Qué significa tener este margen?

Cuando los valores atípicos son parte de los vectores de soporte, se habla de un "margen suave" en SVM. Esto significa que se permite cierto grado de error en la clasificación, permitiendo que algunos puntos estén dentro del margen o incluso en el lado incorrecto de la decision boundary.

Qué función desempeñan las variables de holgura en SVM (slack vars)?

Las variables de holgura (slack variables) en SVM permiten la flexibilidad en la clasificación, especialmente en casos donde los datos no son linealmente separables. Estas variables representan el grado de violación de las restricciones y permiten algunos errores de clasificación dentro del margen o en el lado incorrecto de la boundary.

¿Qué pasa si los outliers son support vectors? ¿No debería pasar eso? ¿Cómo se puede solucionar?

Si los outliers son support vectors, puede afectar negativamente el rendimiento del modelo SVM. Esto no debería suceder idealmente, y para solucionarlo, se pueden ajustar los parámetros del modelo, como la penalización (C) y la elección de kernel.

¿Cómo trabaja SVM si las clases no son linealmente separables?

SVM utiliza un truco de kernel para mapear los datos a un espacio de mayor dimensión donde las clases pueden ser separables linealmente.

Qué solución se propone al problema de clases no linealmente separables?

La solución propuesta es usar el kernel trick, que permite manejar casos donde las clases no son linealmente separables en el espacio original.

Cuál es el propósito del kernel trick en SVM?

El kernel trick permite transformar los datos a un espacio de mayor dimensión de manera implícita, facilitando la separación lineal de clases que no serían separables en el espacio original.

¿Cómo evita el Kernel Trick la necesidad de calcular explícitamente la transformación de características?

El Kernel Trick evita calcular explícitamente la transformación de características al realizar operaciones en el espacio de características transformado mediante el kernel, sin necesidad de conocer la transformación exacta.

En qué situaciones el kernel trick puede ser NO beneficioso?

El kernel trick puede no ser beneficioso en conjuntos de datos grandes, ya que el cálculo implícito en un espacio de mayor dimensión puede volverse computacionalmente costoso.

Cuál es el propósito principal de la función de similitud?

La función de similitud mide la similitud entre dos instancias de datos y es fundamental para el cálculo de las transformaciones no lineales en SVM mediante el kernel trick.

Cuál es la relación entre la función de similitud y el kernel trick?

La función de similitud es esencial para el kernel trick, ya que define la medida de similitud utilizada para realizar transformaciones no lineales sin necesidad de conocer la transformación explícita.

Ejemplos típicos de función de similitud?

Ejemplos de funciones de similitud incluyen el kernel lineal, polinómico y radial (RBF).

Cuál es el objetivo de la intuición?

La intuición en SVM busca comprender cómo el modelo clasifica las instancias, especialmente enfocándose en los vectores de soporte y el margen.

Qué papel juegan los vectores de soporte en la intuición de SVM?

Los vectores de soporte son instancias de datos que determinan la posición y orientación del hiperplano de decisión, siendo fundamentales para la intuición en SVM.

Qué indica la geometría cuando el modelo es linearmente separable?

La geometría indica que existe un hiperplano de decisión que puede separar perfectamente las clases en el espacio de características.

Por qué es importante que el plano de decisión sea equidistante de los vectores de soporte?

Es importante para maximizar el margen y mejorar la generalización del modelo, evitando el sobreajuste a los datos de entrenamiento.

Cuál es el objetivo principal de la función de optimización en SVM?

El objetivo principal es maximizar el margen entre las clases y minimizar la clasificación errónea, expresado a través de la función de pérdida y los términos de penalización.

Cuál es la importancia de la regularización en la función de optimización?

La regularización, controlada por el parámetro C, equilibra la maximización del margen y la minimización de errores, evitando el sobreajuste.

Cuál es la principal diferencia entre SVM y Regresión logística en relación con los valores de salida?

Mientras que la Regresión Logística produce probabilidades directas, SVM produce decisiones basadas en la posición de las instancias en relación con el hiperplano de decisión.

En qué tipo de problemas son especialmente útiles las SVM en comparación con la Regresión logística?

Las SVM son especialmente útiles en problemas de clasificación con espacios de características de alta dimensión y conjuntos de datos más pequeños.

Cómo afecta la presencia de valores atípicos a SVM y Regresión logística?

Los valores atípicos pueden tener un impacto significativo en la SVM debido a su sensibilidad a los vectores de soporte. La Regresión Logística es menos afectada porque utiliza probabilidades.

Cuándo es preferible usar SVM en lugar de Regresión logística?

Se prefiere SVM cuando se enfrenta a problemas de clasificación no lineal, especialmente cuando el número de dimensiones es alto y el conjunto de datos no es muy grande.

Cómo afecta al SVM un parámetro de regularización C alto o bajo?

Un valor alto de C hace que el modelo SVM sea menos tolerante a errores de clasificación, mientras que un valor bajo permite más errores pero busca maximizar el margen.

Qué son las funciones kernel y cómo se relacionan con la Representación dual?

Las funciones kernel son funciones que cuantifican la similitud entre instancias de datos. Se relacionan con la Representación dual al permitir el cálculo eficiente del producto punto en el espacio de características transformado.

Cuándo es especialmente útil la Representación dual en función del tamaño de datos y por qué?

La Representación dual es útil en conjuntos de datos grandes, ya que permite trabajar con productos punto sin la necesidad de calcular explícitamente la transformación de características.

Cómo afecta el parámetro de regularización C al uso de las slack variables?

Un parámetro C más alto penalizará más fuertemente las slack variables, lo que resulta en un modelo SVM más estricto, mientras que un C más bajo permite más violaciones de margen.