
RESULTADO DE APRENDIZAJE

RdA de la asignatura:

- **RdA 3:** Resolver problemas prácticos mediante el uso de modelos de aprendizaje automático, ajustándolos para la mejora de su rendimiento y precisión.

RdA de la actividad:

- Comprender los conceptos de validación cruzada y su importancia en la evaluación de modelos.
- Aplicar técnicas de optimización de hiperparámetros (Grid Search y Random Search) en modelos de aprendizaje automático.
- Implementar el guardado y carga de modelos entrenados para reutilización.
- Diseñar un esquema de trabajo estructurado para proyectos de aprendizaje automático.

INTRODUCCIÓN

Pregunta inicial: ¿Cómo evaluamos correctamente un modelo para asegurarnos de que no está sobreajustado o subajustado?

DESARROLLO

Actividad 1: Validación cruzada

Esta actividad se centra en comprender la validación cruzada y su importancia en la evaluación de modelos. Se explorarán diferentes tipos de validación cruzada y su impacto en las métricas de desempeño del modelo.

¿Cómo lo haremos?

- **Clase magistral:** Se cubrirán los siguientes conceptos clave:
 - Propósito de la validación cruzada.
 - Tipos de validación cruzada: hold-out, K-Fold y LOOCV.
 - Ventajas y desventajas de cada tipo.
- **Implementación en Python:** Los estudiantes accederán a un cuaderno de Jupyter preparado con ejemplos sobre validación cruzada.

Enlace al cuaderno: [21-Validacion-Cruzada.ipynb](https://colab.research.google.com/github/andresmerino/21-Validacion-Cruzada.ipynb).

Verificación de aprendizaje:

- ¿Qué ventajas tiene el uso de K-Fold sobre el hold-out?
- ¿Cómo afecta el tamaño de k en la validación cruzada?
- ¿Por qué es importante medir la variabilidad en los resultados de validación cruzada?

Actividad 2: Optimización de hiperparámetros

Esta actividad introduce las técnicas de optimización de hiperparámetros, como Grid Search y Random Search. Los estudiantes aprenderán a configurar búsquedas de hiperparámetros y a evaluar sus resultados.

¿Cómo lo haremos?

- **Clase magistral:** Se cubrirán los siguientes temas:
 - Diferencia entre parámetros aprendidos e hiperparámetros.
 - Importancia de la optimización de hiperparámetros.
 - Grid Search: concepto y configuración.
 - Random Search: concepto, ventajas y configuración.
 - Comparación entre ambos métodos: eficiencia y uso.
- **Implementación en Python:** Los estudiantes explorarán un cuaderno de Jupyter con ejemplos prácticos de Grid Search y Random Search.

Enlace al cuaderno: [22-Optimizacion-Hiperparametros.ipynb](#).

Verificación de aprendizaje:

- ¿Cuáles son las principales diferencias entre Grid Search y Random Search?
- ¿Cómo afecta la validación cruzada en la búsqueda de hiperparámetros?
- ¿En qué casos sería más eficiente usar Random Search en lugar de Grid Search?

Actividad 3: Guardado y lectura de modelos

Esta actividad aborda la importancia de guardar y cargar modelos entrenados para evitar costos computacionales y asegurar reproducibilidad.

¿Cómo lo haremos?

- **Clase magistral:** Se discutirán los siguientes temas:
 - Beneficios de guardar modelos: eficiencia y despliegue.
 - Métodos de serialización: `pickle` y `joblib`.
 - Buenas prácticas al guardar modelos (compatibilidad y versiones de librerías).

- Precauciones al cargar modelos en diferentes entornos.

- **Implementación en Python:** Los estudiantes practicarán con dos cuadernos de Jupyter que incluyen ejemplos sobre cómo guardar y cargar modelos entrenados.

Enlace al cuaderno 1: [23-Guardado-Modelos.ipynb](#).

Enlace al cuaderno 2: [24-Lectura-Modelos.ipynb](#).

Verificación de aprendizaje:

- ¿Cuáles son las diferencias entre `pickle` y `joblib`?
- ¿Qué problemas pueden surgir al cargar un modelo en otro entorno?
- ¿Cómo podrías manejar cambios de versiones en las librerías usadas para entrenar un modelo?

Actividad 4: Esquema de trabajo para proyectos de Aprendizaje Automático

Esta actividad guía a los estudiantes en el diseño de un flujo de trabajo completo para proyectos de Aprendizaje Automático, destacando las mejores prácticas y cada etapa clave del proceso.

¿Cómo lo haremos?

- **Clase magistral:** Se cubrirá el flujo de trabajo típico en Aprendizaje Automático, del Cuadro 1.

Verificación de aprendizaje:

- ¿Qué pasos son imprescindibles en un flujo de trabajo de Aprendizaje Automático?
- ¿Cómo garantizas la reproducibilidad en cada etapa del proceso?
- ¿Qué herramientas y librerías usarías para automatizar este esquema?

CIERRE

Tarea: Desarrollar los ejercicios planteados en el siguiente cuaderno y entregarlo por el aula virtual:

Enlace al cuaderno: [10-Optimizacion-Hiperparametros.ipynb](#).

Pregunta de investigación:

1. ¿Qué métodos avanzados de optimización de hiperparámetros existen (como Bayesian Optimization) y en qué casos podrían ser útiles?

Para la próxima clase: Ya no hay próxima clase :’C

| Paso | Detalle | Datos |
|------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------|--------------|
| Limpieza de datos | Manejar valores faltantes, duplicados y atípicos según el contexto del problema. | Completo |
| Análisis exploratorio | Visualiza distribuciones, correlaciones y patrones en los datos para entender su estructura. | Completo |
| Selección de atributos (opcional) | Elige atributos relevantes usando técnicas estadísticas, de importancia de modelos o reducción de dimensionalidad. Puede ubicarse aquí o en otros puntos. | Completo |
| Separar train-test | Divide los datos en entrenamiento y prueba. | Completo |
| Benchmark inicial | Entrena varios modelos usando train y evalúa su desempeño en test. | Train y Test |
| Seleccionar modelos | Elige los mejores modelos según el benchmark inicial. | |
| Validación cruzada y ajuste de hiperparámetros | Realiza búsqueda de hiperparámetros junto con validación cruzada para los modelos seleccionados. | Train |
| Seleccionar mejor modelo | Escoge el modelo más prometedor tras la validación cruzada. | |
| Evaluar en test | Evalúa el modelo seleccionado en el conjunto de prueba. | Test |
| Reentrenar con todos | Reentrena el modelo final con todos los datos (train + test combinados). | Completo |
| Guardar modelo | Serializa el modelo para despliegue en producción. | |

Cuadro 1: Esquema de trabajo para proyectos de Aprendizaje Automático.