

---

## **RESULTADO DE APRENDIZAJE**

---

### **RdA de la asignatura:**

- **RdA 1:** Plantear los conceptos fundamentales del aprendizaje automático, incluyendo los principios básicos, técnicas de preprocesado de datos, métodos de evaluación y ajuste de modelos, destacando su importancia en el análisis y resolución de problemas de datos.

### **RdA de la clase:**

- Comprender las técnicas principales de preprocesado de datos: limpieza, normalización, discretización y reducción de dimensionalidad.
- Aplicar métodos de partición de conjuntos de datos: leave-one-out, leave-p-out y k-fold cross-validation.
- Identificar la importancia de los conjuntos de validación en el ajuste y evaluación de modelos.

---

## **INTRODUCCIÓN**

---

**Pregunta inicial:** Si desarrollas un algoritmo para predecir comportamientos futuros a partir de datos históricos, ¿cómo comprobarías que sus aciertos no se deben únicamente a que «vio» demasiado los ejemplos originales?

---

## **DESARROLLO**

---

### **Actividad 1: Preprocesado de Datos**

En esta actividad los estudiantes aprenderán las técnicas básicas de preprocesado de datos mediante clase magistral y práctica en cuaderno de Jupyter, incluyendo limpieza, normalización, discretización y reducción de dimensionalidad para preparar datos de aprendizaje automático.

### **¿Cómo lo haremos?**

- **Conceptos fundamentales:** Se presentarán las técnicas de preprocesado: normalización, discretización, one-hot-encoding y reducción de dimensionalidad.
- **Implementación en Python:** Los estudiantes accederán a un cuaderno de Jupyter previamente preparado.

Enlace al cuaderno: [03-1-Preprocesado-Datos.ipynb](#).

- **Experimentación:** Realizar los ejercicios propuestos en el cuaderno.

### Actividad 2: Conjuntos de Entrenamiento y Test

En esta actividad se abordará la importancia de dividir un conjunto de datos en entrenamiento y prueba mediante exploración de cuaderno de Jupyter.

#### ¿Cómo lo haremos?

- **Métodos de partición:** Se presentarán las estrategias de división de datos.
- **Implementación en Python:** Los estudiantes accederán a un cuaderno de Jupyter previamente preparado.

Enlace al cuaderno: [03-2-Conjuntos-Entrenamiento-Prueba.ipynb](#).

- **Experimentación:** Realizar los ejercicios propuestos en el cuaderno.

---

### CIERRE

---

#### Verificación de aprendizaje:

1. ¿En qué rangos queda una variable luego de aplicar cada tipo de normalización?
2. ¿Cuál es la diferencia entre discretizar una variable usando igual amplitud o igual frecuencia?
3. ¿Dónde ingresan el concepto de valores propios en el ACP?

#### Preguntas tipo entrevista:

1. Describe la diferencia entre discretización y one-hot encoding.
2. ¿Qué método de normalización utilizarías para variables categóricas antes de entrenar un modelo?

**Tarea:** Realizar el procesamiento de datos necesarios al conjunto de datos seleccionado en la clase anterior.

#### Pregunta de investigación:

1. ¿Qué estrategias existen para manejar conjuntos de datos desbalanceados al realizar particiones de entrenamiento y prueba?
2. ¿Realizar one-hot encoding agrega colinealidad a las características? Ver: [The Problem With One-Hot Encoding](#)

**Para la próxima clase:** Se debe tener el conjunto de datos preprocesados y listos en el repositorio.