

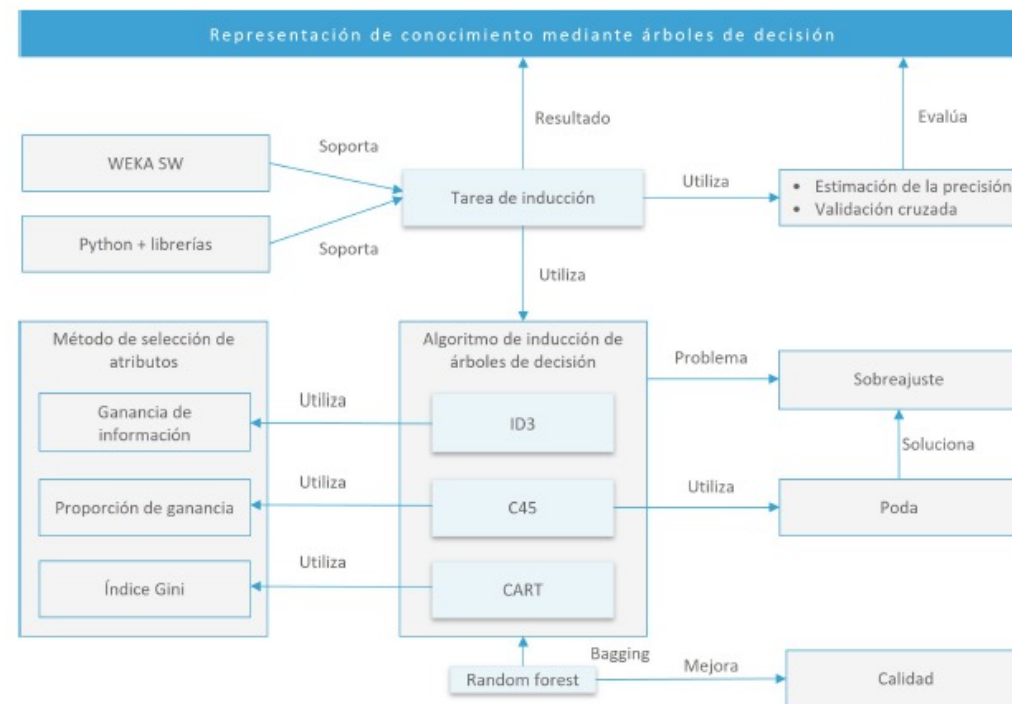
Técnicas de Inteligencia artificial

Tema 3. Árboles de decisión II

00 – ¿De qué hablamos en nuestra clase anterior?

- ✓ Representación del conocimiento mediante árboles de decisión
- ✓ Cómo aprenden los árboles de decisión
- ✓ El Algoritmo básico de árboles de decisión: ID3
- ✓ Problema : Resolución de un problema de clasificación con árboles de decisión

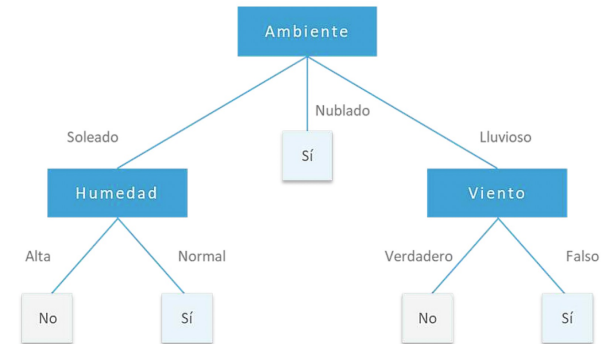
00 – Visión general del tema



00 – Recordando

Aprendizaje de árboles de decisión

- Es una técnica **de aprendizaje supervisado**
- Permite **clasificar** instancias cuya clase sea desconocida.
- Se utiliza tanto para **regresión** como para **clasificación**.
- Funciona mejor con **variables categóricas**.



Ejemplo de árbol de decisión para el aprendizaje del concepto «Jugar al aire libre».

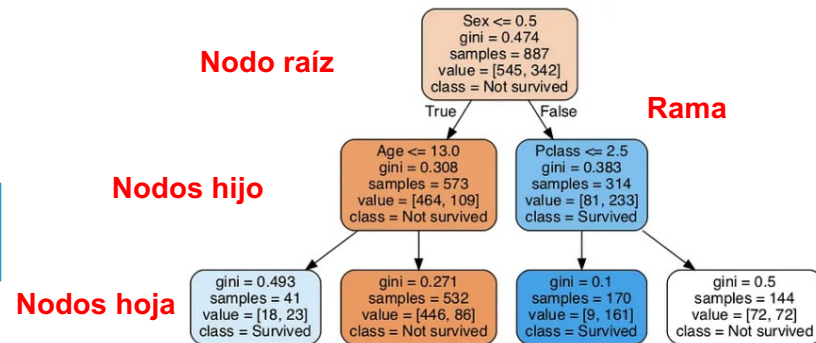
00 – Recordando

¿De qué están compuestos los árboles de decisión?

- Están compuestos por **nodos** y **ramas**.
- Un nodo representa un conjunto o subconjunto de datos.
- Una rama representa la condición **True/False**.



Los datos de cada nodo deben ser lo más homogéneos posibles (misma clase)



Ejemplo de un árbol de decisión. Fuente: towardsdatascience.com

00 – Recordando

Aspectos clave en los árboles de decisión

- Elegir el método de selección de atributos: **Gini, Ganancia de Información.**
- Aplicar técnicas que eviten el sobreajuste: **Validación cruzada, Prepoda y Postpoda.**
- Evaluar la precisión o rendimiento del algoritmo: **Matriz de confusión y Curva ROC-AUC.**

¿Os acordáis en qué consistía la selección de atributos?



00 – Recordando

La selección de atributos

- Busca hacer la **mejor división** (*Best Split*).
- Utiliza las siguientes medidas y técnicas:
 - ✓ **Entropía:** Mide la homogeneidad de los datos.
 - ✓ **Índice de Gini:** Mide la homogeneidad de los datos.
 - ✓ **Ganancia de información:** Mide la información que proporciona un atributo.



Existen otras técnicas de selección de atributos como la Proporción de ganancia de Información que se utiliza junto con la Ganancia de Información

00 – Recordando

La entropía

- Mide el grado de incertidumbre o impureza.
- Se utiliza para **medir la homogeneidad** de los datos.
- La utiliza la Ganancia de información en la selección de atributos.
- La incluyen los algoritmos **ID3** y **C4.5**.



Una entropía cercana a 0 indica homogeneidad (misma clase) mientras que una cercana a 1 implica heterogeneidad)

00 – Recordando

Gini index

- Mide la **probabilidad** de que una característica se clasifique incorrectamente.
- Se utiliza también para **medir la homogeneidad** de los datos.
- La incluye el algoritmo **CART**.



Un índice cercano a 0 indica que la muestra es homogénea mientras que uno cercano a 1 indica que es heterogénea

00 –Recordando

La Ganancia de Información

- Mide cómo se reduce la incertidumbre después de dividir los datos.
- **Utiliza la entropía** para medir la homogeneidad de los datos.
- La utilizan los algoritmos **ID3** y **C4.5**.



Una ganancia de Información alta implica una mayor homogeneidad en los datos mientras que una más baja conlleva una mayor heterogeneidad

00 – Recordando

Sobreajuste u *overfitting*

- El modelo se ajusta demasiado a los datos de entrenamiento.
- Técnicas para evitar este sobreajuste que veremos:
 - ✓ Utilizar la técnica de **validación cruzada**.
 - ✓ Utilizar **técnicas de ensemble**.
 - ✓ Aplicar técnicas de **Pre-poda** y **Post-poda**.

¿Os acordáis de alguna otra técnica para reducir el sobreajuste?



00 – Recordando

Pre-poda

- Consiste en establecer criterios para limitar la expansión del árbol.
- Estos límites se pueden establecer configurando **hiperparámetros**:
 - ✓ **Min_samples_leaf**: n° mínimo de muestras de una hoja.
 - ✓ **Min_samples_Split**: n° mínimo de observaciones para dividir un nodo.
 - ✓ **Max_features**: n° máximo de variables para considerar la mejor división.
 - ✓ **Max_Depth**: n° máximo de niveles del árbol (profundidad)



Los hiperparámetros que más se suelen configurar son Max_Depth y Min_samples_split

00 – ¿De qué vamos a hablar hoy?

- ✓ Técnicas para evitar el sobreajuste: Poda de árboles
- ✓ Técnicas para evaluar la precisión de la clasificación
- ✓ Ensamble learning y algoritmo Random Forest
- ✓ Problema : Resolver un problema de clasificación con árboles de decisión en Python
- ✓ Presentación de la Actividad 1. Árboles de decisión, reglas y ensemble learning

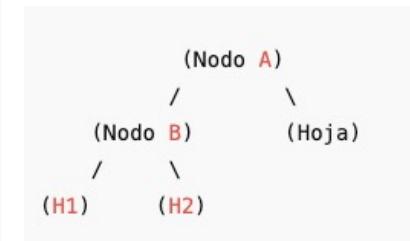
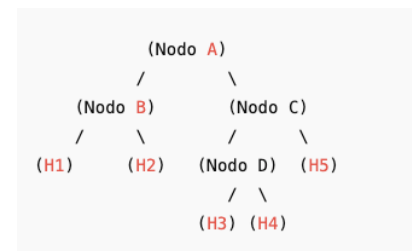
01 –Sobreaajuste y poda de árboles

¿Qué es la poda de árboles?

- Consiste en reducir el tamaño del árbol una vez construido.
- Se puede llevar a cabo de dos formas:
 - ✓ **Poda directa:** Se eliminan subárboles y se reemplazan por nodos hoja.
 - ✓ **Poda por reglas:** Se eliminan condiciones de las reglas.



La poda directa se realiza con técnica como CCP (coste complejidad)



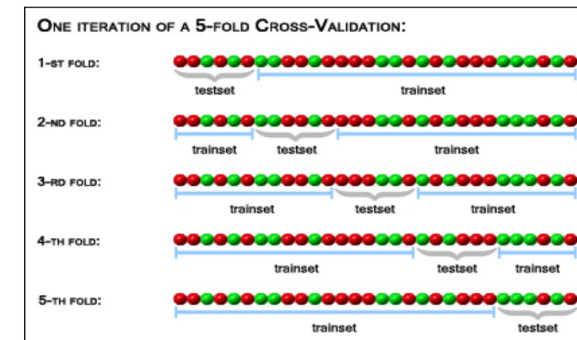
01 – Sobreajuste y poda de árboles

¿Cómo se puede saber el tamaño adecuado de un árbol?

- Utilizando Validación cruzada (*Cross-Validation*).
- Haciendo una poda pesimista de C4.5
- Simplificando mediante poda de reglas.



Hablaremos de reglas en el Tema 4, nuestra próxima clase .



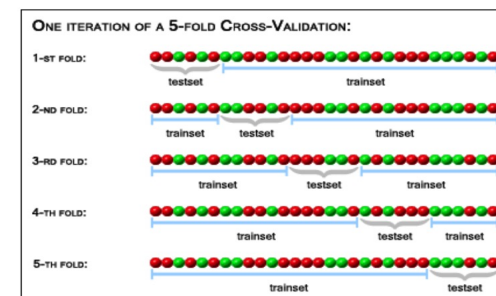
01 – Sobreajuste y poda de árboles

¿En qué consiste la técnica de Cross-Validation?

- Evalúa la capacidad de **generalizar** del modelo en nuevos datos.
- Evita el **sobreajuste** u *overfitting*.
- Divide el conjunto de datos en **k-folds** (subconjuntos).
- Habitualmente en 5 o 10 folds.



Por ejemplo, si divide en 5 folds, 4 se utilizan para entrenamiento y 1 para test. El proceso se repite 5 veces



02 – Medidas de precisión de la clasificación

La matriz de confusión

- Indica el número de **predicciones correctas e incorrectas** para cada clase.
- Las instancias clasificadas correctamente caen en la diagonal de la matriz.

		Predicción	
		Positivos	Negativos
Observación	Positivos	Verdaderos Positivos (VP)	Falsos Negativos (FN)
	Negativos	Falsos Positivos (FP)	Verdaderos Negativos (VN)

Matriz de confusión. Fuente: (Rpubs.com, 2017)



La matriz de confusión evalúa tanto clasificación binaria como multiclase.

02 – Medidas de precisión de la clasificación

La matriz de confusión

- Tasa (VP/TP): Instancias positivas correctamente clasificadas.
- Tasa (FN/FN): Instancias negativas incorrectamente clasificadas.
- Tasa (VN/TN): Instancias negativas correctamente clasificadas.
- Tasa (FP/FP): Instancias positivas incorrectamente clasificadas.

		Predicción	
		Positivos	Negativos
Observación	Positivos	Verdaderos Positivos (VP)	Falsos Negativos (FN)
	Negativos	Falsos Positivos (FP)	Verdaderos Negativos (VN)

02 – Medidas de precisión de la clasificación

Métricas de la matriz de confusión

Accuracy

- Proporción de predicciones correctas entre el n° total de predicciones.

$$accuracy = \frac{TP + TN}{TP + TN + FP + FN}$$

Precision

- Proporción de ejemplos verdaderamente positivos entre los positivos.

$$precision = \frac{TP}{TP + FP}$$

02 – Medidas de precisión de la clasificación

Métricas de la matriz de confusión

Recall

- Proporción de ejemplos correctamente clasificados.

$$recall = \frac{TP}{TP + FN}$$

F-measure (F1)

- Combina precisión y recall utilizando la media armónica.

$$F1 = \frac{2 * precision * recall}{recall + precision} = \frac{2 * TP}{2 * TP + FP + FN}$$

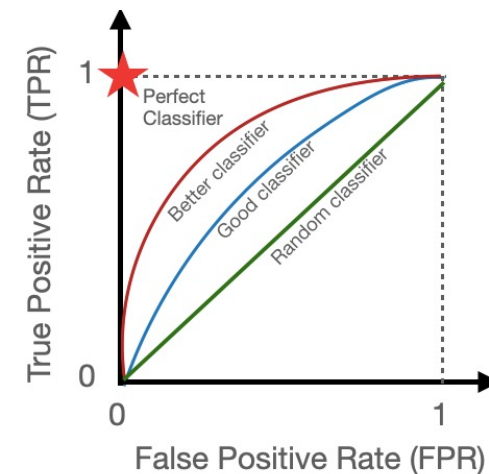
02 – Medidas de precisión de la clasificación

Curva ROC-AUC

- Se utiliza en problemas de **clasificación binaria**.
- ROC es la curva que representa la **relación entre TP y FP**.
- AUC es el área que hay por debajo de la curva:
 - ✓ Un valor AUC de 0,5 implica una capacidad predictiva baja.
 - ✓ Un valor AUC de 1 una capacidad predictiva óptima.



A diferencia de la matriz de confusión, la curva ROC-AUC nos proporciona solo información sobre TP y FP



02 – Medidas de precisión de la clasificación

Vamos a verlo con un ejemplo

Imaginad que tenemos que **predecir el riesgo de sufrir un problema cardiovascular** de un paciente (problema de clasificación binaria). Tras dividir los datos disponibles en entrenamiento y test utilizamos la matriz de confusión para evaluar el rendimiento del modelo y tenemos el siguiente resultado:

	Predicted negative	Predicted positive
Negative class	390	13
Positive class	24	23

- **390 (TN)**: Casos que el algoritmo ha **predicho que no** tiene riesgo y **no** lo tiene.
- **13 (FP)**: Casos que ha **predicho que sí** lo tiene, pero **no** lo tiene.
- **24 (FN)**: Casos que ha **predicho que no** lo tiene, pero **sí** lo tiene.
- **23 (VP)**: Casos que ha **predicho que sí** lo tiene y **sí** lo tiene.

02 – Medidas de precisión de la clasificación

¿Cómo calcularíamos estas métricas y cual es su significado?

- Accuracy : $TP+TN/TP+TN+FP+FN = 413/450 = 0,917$
- Precision: $TP/TP+FP=23/36 = 0,63$
- Recall: $TP/TP+FN=23/47 = 0,48$

	Predicted negative	Predicted positive
Negative class	390	13
Possitive class	24	23

¿Qué implica una Accuraccy de 0,917? ¿Y una precisión de 0,63? ¿Y un Recall de 0,48?



02 – Medidas de precisión de la clasificación

Cómo interpretar las métricas que hemos obtenido

Accuracy del 0,91

El modelo acierta en un 91% de las veces a la hora de predecir si un paciente tendrá riesgo o no de sufrir una enfermedad cardiovascular.

Precisión del 0,63

De todas los pacientes que el modelo predijo que tendrían riesgo cardiovascular ha acertado en el 63% de los casos. Esto implica que hay un 37% de FP (pacientes a los que le que le han dicho que tienen riesgo, pero no lo tienen)

Recall del 0,48

De todos los pacientes que el modelo predijo que tendría riesgo cardiovascular (TP y FN) solo ha acertado en el 48% de los casos. Esto implica que hay un 52% de FN (pacientes que les han dicho que no tienen riesgo, pero si lo tienen)

¿Diríais que este modelo tiene un buen rendimiento?



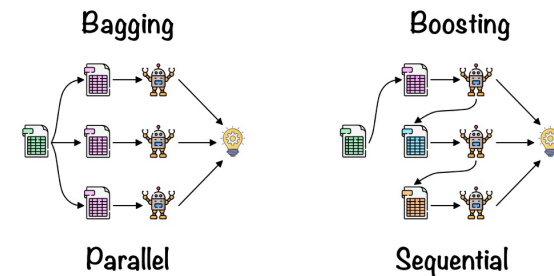
03 – Ensemble learning y algoritmo Random Forest

¿Qué son los métodos de ensemble?

- Se utilizan para **mejorar la capacidad predictiva** de los modelos.
- Combinan distintos algoritmos de machine learning para crear modelos.
- Los métodos de ensemble pueden ser de dos tipos:
 - ✓ **Boosting:** Usan algoritmos de forma secuencial.
 - ✓ **Bagging:** Usan algoritmos en paralelo.



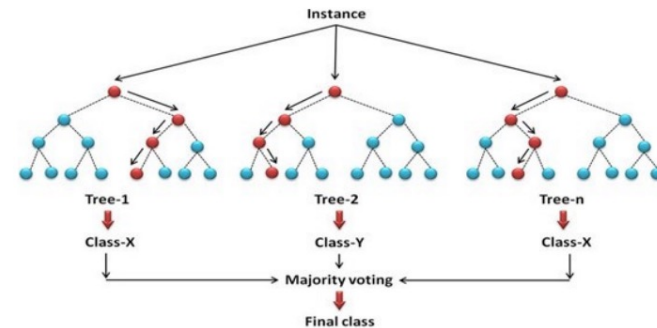
El algoritmo Random Forest utiliza Bagging y los algoritmos AdaBoost y Gradient Boosting utilizan boosting.



03 – Ensemble learning y algoritmo Random forest

Algoritmo Random Forest

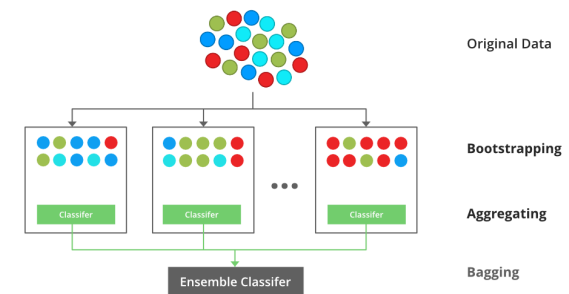
- Es un método basado en **ensembles**.
- Entrena **árboles de decisión**.
- Combina los resultados de esos árboles individuales utilizando **bagging**.
- Para crear los conjuntos de datos de cada árbol utiliza **bootstrapping**.



03 – Ensemble learning y algoritmo Random Forest

Etapa1. Creación de los conjuntos de datos

- Para crear los conjuntos de datos de cada árbol se utiliza **bootstrapping**.
- Cada conjunto tiene el mismo tamaño que la muestra original.
- Las instancias que no se seleccionan pueden usarse para calcular el **out-of-bag**.
- Se persigue conseguir variabilidad en el modelo.



Bagging Fuente: Medium.com

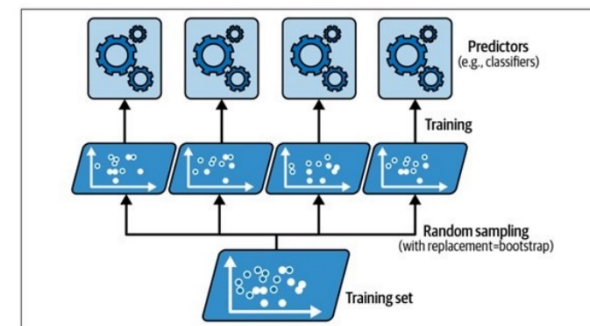
03 – Ensemble learning y algoritmo Random Forest

Etapa 2. Entrenamiento de cada árbol

- Cada árbol se entrena con un subconjunto de datos.
- **Las variables de cada split se seleccionan de forma aleatoria.**
- Esta aleatoriedad consigue decorrelar los árboles generados.
- En esta etapa se ajustan los **hiperparámetros** y se busca optimizar.



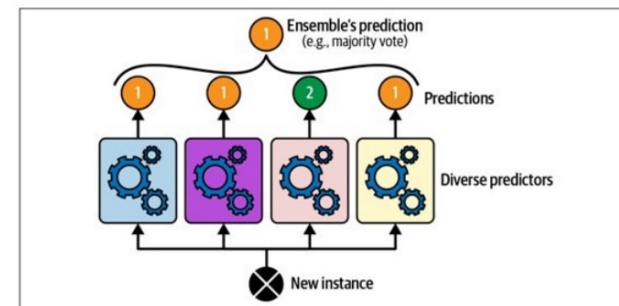
En un árbol de decision no hay aleatoriedad (se tienen en cuenta todas las variables). En un Random Forest hay aleatoriedad (hay variables que no tienen por que salir)



03 – Ensemble learning y algoritmo Random Forest

Etapa 3. Obtención de la predicción

- Cada árbol realiza una predicción de forma independiente.
- La forma de combinar las predicciones dependerá del problema a resolver:
 - ✓ **Clasificación:** Se vota por mayoría (la clase más predicha).
 - ✓ **Regresión:** Se calcula el promedio de las predicciones.



Votación de las predicciones del clasificador. Fuente: Geron, 2023.

03 – Ensemble learning y algoritmo Random Forest

Implementaciones de Random Forest

Forests of randomized trees

- <https://scikit-learn.org/stable/modules/ensemble.html#forests-of-randomized-trees>

Tensor Flow Decision Forest

- https://www.tensorflow.org/decision_forests?hl=es-419
- https://keras.io/examples/structured_data/classification_with_tfdf/

04 – Problema. Resolver un problema de clasificación con Python

Tema 3. ArbolesDecision.ipynb ☆ ⚙ Guardando...

Archivo Editar Ver Insertar Entorno de ejecución Herramientas Ayuda

Comandos + Código + Texto ▶ Ejecutar todas

Técnicas de Inteligencia Artificial

MU en Análisis y Visualización de datos masivos

▼ Árboles de decisión para clasificación

¿Qué problema vamos a resolver?

Resolveremos un **problema de clasificación** utilizando el algoritmo de aprendizaje supervisado **Árboles de decisión**. El problema específico que queremos resolver es **predecir si un pasajero sobrevivirá al naufragio o no**. Es decir, predecir la probabilidad de que un pasajero pertenezca a la Clase 0 (NO sobrevive) y 1 (Sobrevive).

Descripción de los datos

- **PassengerId** = identificador único de cada pasajero
- **Name** = nombre del pasajero
- **Sex** = factor, con niveles (masculino y femenino)
- **Age** = edad del pasajero
- **Pclass** = clase en la que viajaba el pasajero embarked = lugar en el que embarcó el pasajero
- **Ticket** = número de ticket del pasajero (na para la tripulación)
- **Fare** = precio del ticket (na para la tripulación, músicos, empleados y otros)
- **SibSp** = número de hermanos/familiares
- **Cabin** = cabina que ocupa cada pasajero
- **Parch** = número de padres e hijos a bordo
- **Survived** = especifica si el pasajero sobrevivió al hundimiento

05 – Presentación Actividad 1. Laboratorio

Asignatura	Datos del alumno	Fecha
Técnicas de Inteligencia Artificial	Apellidos:	
	Nombre:	

Actividad 1. Laboratorio (individual): Árboles de decisión, reglas y *ensemble learning*

Objetivo

Esta actividad te permitirá profundizar en la importación y manejo de *datasets*, así como su posterior aplicación de técnicas de aprendizaje supervisado (clasificación) basadas en árboles de decisión, reglas de clasificación y técnicas *ensemble learning* utilizando librerías como Scikit-learn sobre Python. Para ello, se te proporcionará un *dataset* determinado (en un archivo CSV) y deberás elegir al menos **dos algoritmos de clasificación** eligiendo de entre los árboles de decisión (como *decision tree classifier* o CART, por sus siglas en inglés), las reglas de clasificación y las técnicas de *ensemble learning* (como *random forest*).

Descripción

En primer lugar, repasa los siguientes contenidos teórico-prácticos de la asignatura:

- ▶ El tema «Python para la implementación de técnicas de inteligencia artificial» y haciendo hincapié en los siguientes apartados:
 - «Librerías útiles para el análisis de datos», en particular, los apartados sobre



En la descarga de la actividad se adjunta un notebook que os puede servir de base

00 – ¿De qué vamos a hablar en nuestra próxima clase?

- Reglas de clasificación y asociación
- Algoritmos de aprendizaje y reglas de clasificación
- Algoritmos de aprendizaje y reglas de asociación.
- Problema: Resolución de problemas de clasificación usando la biblioteca Scikit-learn.

**Muchas gracias por
vuestra atención**



www.unir.net