Datos del estudiante

**Nombre y apellidos**

**Fecha de entrega**

Trabajando con árboles de decisión y técnicas de *ensemble learning*

Objetivos de la actividad

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 habrás de elegir al menos dos algoritmos de clasificación eligiendo de entre los árboles de decisión (como *Decision Tree Classifier* o CART), las reglas de clasificación y las técnicas de *ensemble learning* (como *random forest*).

Pautas de elaboración

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

* Tema: Python para la implementación de técnicas de inteligencia artificial, haciendo especial hincapié en las secciones:
* Librerías útiles para el análisis de datos (apartados Scikit-learn, Matplotlib y Pandas).
* Importación de datos.
* Introducción a *Machine Learning* con librerías en Python.
* Tema: Árboles de decisión, especialmente:
* Medidas de precisión de la clasificación. Curva ROC.
* Aplicaciones y ejemplos de implementación.
* Tema: Reglas, especialmente:
* Ejemplos de implementación.

El anterior repaso y la ejecución de los ejemplos incluidos en los diferentes apartados te permitirá asegurar que cuentas en tu máquina computadora con el entorno de ejecución de Python y con todas las librerías necesarias para llevar a cabo esta actividad, además de haber realizado varios ejemplos comparados de clasificación con la librería *scikit-learn*. Del mismo modo, permitirá asegurar que estás familiarizado con los conceptos teóricos de los algoritmos de clasificación a emplear.

* Has de utilizar, **al menos**, las siguientes librerías:
* **pandas** para la importación de datos.
* **scikit-learn** para al menos alguno de los ejemplos.
* **matplotlib** o **seaborn** para la muestra de gráficas.
* Utiliza Python y la librería **pandas** para la importación del dataset. **No es posible adjuntar ningún archivo adicional**. El propio código Python del archivo ha de ingerir automáticamente el *dataset* proporcionado como archivo CSV encontrándose en la ruta actual de ejecución y sin emplear rutas absolutas o relativas que solo funcionen en la computadora del alumno. **Preprocesa los datos si fuera necesario.**
* Mediante Python y las librerías que consideres (para la parte gráfica utiliza **matplotlib y/o seaborn**), analiza el *dataset* proporcionando una caracterización del mismo, mostrando **al menos** algunas de sus características en modo texto (mediante tablas o prosa) y **al menos** algunas de ellas en modo gráfico (por ejemplo: histogramas, diagramas de dispersión, diagramas de cajas y bigotes, etc.). **Las características y las gráficas incluidas han de provenir de la ejecución del código en Python que se aporte como respuesta.**
* Además, dispondrás de un ejemplo de cuaderno de Python que te facilitaremos para que puedas tomarlo como ejemplo de la estructura que podrías seguir.
* A partir del código anterior, describe el conjunto de datos utilizado. Para ello, **incluye en el informe** una descripción de los datos del problema que contenga:
* Número de clases, indicando qué representan dichas clases y el tipo de valor que toman.
* Número de instancias en total.
* Número de instancias pertenecientes a cada clase.
* Número de atributos de entrada, su significado y tipo.
* ¿Hay algún valor de atributo desconocido?
* **Elige al menos dos algoritmos de clasificación** eligiendo de entre los árboles de decisión (como *Decision Tree Classifier* o CART), las reglas de clasificación y las técnicas de *ensemble learning* (como *random forest*).
* Mediante Python y las librerías que consideres, entrena los modelos escogidos, dividiendo el *dataset* en datos de entrenamiento y datos de test previamente en base a tu criterio y pruébalos frente a dichos datos de test. **Incluye en el informe** **una explicación de los parámetros que consideres relevantes en cada ejecución.**
* Mediante Python y **matplotlib y/o seaborn**, **muestra los resultados** obtenidos por los diferentes algoritmos escogidos de forma gráfica y comparada/superpuesta.
* A la hora de escoger una técnica adecuada para generar un clasificador, es muy importante la validación del modelo. Realiza una comparativa de los modelos en lo que respecta a la efectividad en la predicción de la clase de futuras nuevas instancias. Básate **al menos** en los siguientes valores habituales:
* Instancias clasificadas correctamente.
* Instancias clasificadas incorrectamente.
* TP Rate.
* FP Rate.
* Matriz de confusión.
* **Incluye en el informe** una **discusión sobre los resultados obtenidos**, argumenta con qué técnica se obtienen mejores resultados en base a las diferentes métricas que hayas escogido (instancias clasificadas correctamente, TP Rate, etc.) y **explica cómo se podrían mejorar los resultados obtenidos en ambos algoritmos**.
* Recuerda, incluye en el informe a entregar al menos lo siguiente:
* Descripción de la fuente de datos empleada.
* Caracterización del *dataset* utilizado en modo texto y gráfico.
* Parámetros relevantes utilizados en los diferentes algoritmos.
* Resultados obtenidos por los diferentes algoritmos escogidos de forma gráfica y comparada/superpuesta.
* Discusión de los resultados obtenidos y argumentos sobre cómo mejorar de dichos resultados.
* **NOTA:** no está permitido el plagio entre compañeros del mismo o anteriores cursos, esta no es una actividad grupal. Tampoco está permitido el plagio de fuentes de Internet, libros, manuales o de cualquier otro tipo que no se citen apropiadamente. **Entregar un ejercicio que incurra en los anteriores escenarios implicará automáticamente un 0 como calificación de la actividad, sin perjuicio de que se tomen medidas disciplinarias de acuerdo con la correspondiente normativa de la UNIR.** Está permitido utilizar código, fracciones de código o ejemplos existentes en Internet u otras fuentes **siempre que se citen apropiadamente y su uso se limite a partes minoritarias de la actividad**.

Extensión y formato

Habrás de entregar un único informe de 4 páginas como máximo de extensión (Arial 11, interlineao 1,5) y un único archivo Python. **El informe no debe incluir anexos, portada ni repetir el enunciado o la rúbrica de la actividad.** Cualquier página que incluyas será computada dentro de la extensión entregada. No se evaluará ninguna página a partir más allá de la cuarta página. **No es posible adjuntar ningún archivo adicional.** El propio código Python del archivo ha de ingerir automáticamente el *dataset* proporcionado como archivo CSV encontrándose en la ruta actual de ejecución y sin emplear rutas absolutas o relativas que sólo funcionen en la computadora del alumno. Puedes elegir libremente cómo repartes las 4 páginas de extensión del informe para cada uno de los apartados.