

Guion de prácticas

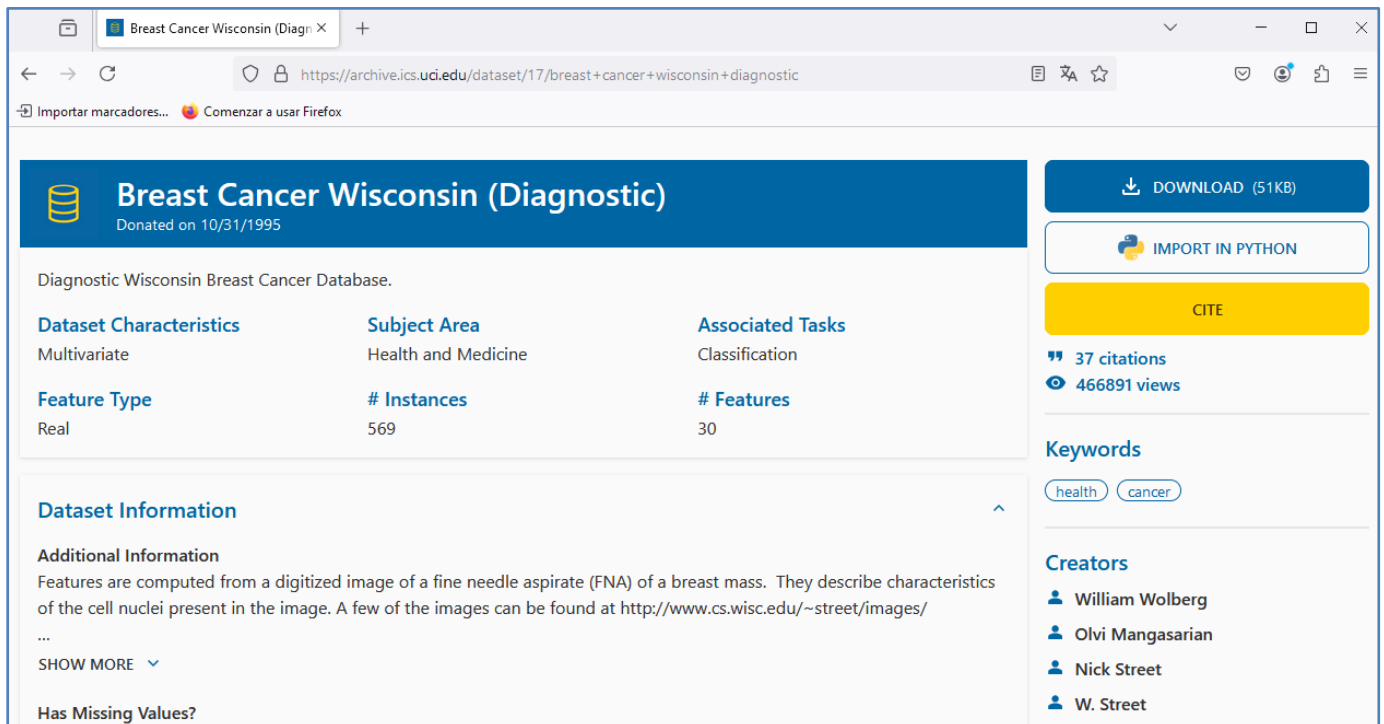
P1. Árboles de Decisión y Selección de Modelos

Introducción

En esta práctica trabajaréis el ciclo completo del **análisis inteligente de datos**, abarcando todas sus fases, que van desde el acceso y pre-procesado de los datos hasta la creación, validación y selección de los modelos. El objetivo de la práctica es que completéis y entendáis el ciclo de análisis en su integridad y, especialmente, comparéis el funcionamiento y rendimiento de varios modelos de predicción (prestando especial atención a los Árboles de Decisión y su comparativa con K-NN). Os resultará necesario, para ello, realizar el **ajuste de hiper-parámetros** de los modelos, para tratar de conseguir las mejores tasas de predicción según varias métricas, de forma consistente (evitando, por tanto, el sobreajuste). Consideraremos además la comparación de modelos por test estadísticos y la selección del mejor modelo por la regla de 1 desviación.

Trabajaréis sobre dos problemas de **clasificación binaria**:

- En el primer problema, el objetivo es diferenciar tumores malignos (cancerosos) y benignos (no cancerosos) en masas mamarias. El cáncer de mama comienza cuando las células de la mama empiezan a crecer de forma descontrolada. Es el tipo de cáncer más común entre las mujeres, representando el 25% de todos los casos de cáncer a nivel mundial. En esta práctica, emplearéis un conjunto de datos (*dataset*) que está basado en el [Breast Cancer Dataset](https://doi.org/10.24432/C5DW2B) de la Universidad de Wisconsin (<https://doi.org/10.24432/C5DW2B>).



The screenshot shows the UCI Machine Learning Repository page for the 'Breast Cancer Wisconsin (Diagnostic)' dataset. The page includes a title bar, a description, and various statistics.

Dataset Characteristics	Subject Area	Associated Tasks
Multivariate	Health and Medicine	Classification

Feature Type	# Instances	# Features
Real	569	30

Dataset Information

Additional Information
Features are computed from a digitized image of a fine needle aspirate (FNA) of a breast mass. They describe characteristics of the cell nuclei present in the image. A few of the images can be found at <http://www.cs.wisc.edu/~street/images/>

...
SHOW MORE

Has Missing Values?

Actions: DOWNLOAD (51KB), IMPORT IN PYTHON, CITE

Citations: 37 citations, 466891 views

Keywords: health, cancer

Creators: William Wolberg, Olvi Mangasarian, Nick Street, W. Street

En este *dataset*, se proporcionan ejemplos anotados de tumores malignos y benignos de más de 400 pacientes. La primera columna en el archivo de datos contiene el identificador de la paciente, mientras que la segunda indica el diagnóstico ("M" = maligno, "B" = benigno). Además, cada tumor se describe mediante 30 atributos o características (las columnas restantes) construidas a partir de una imagen

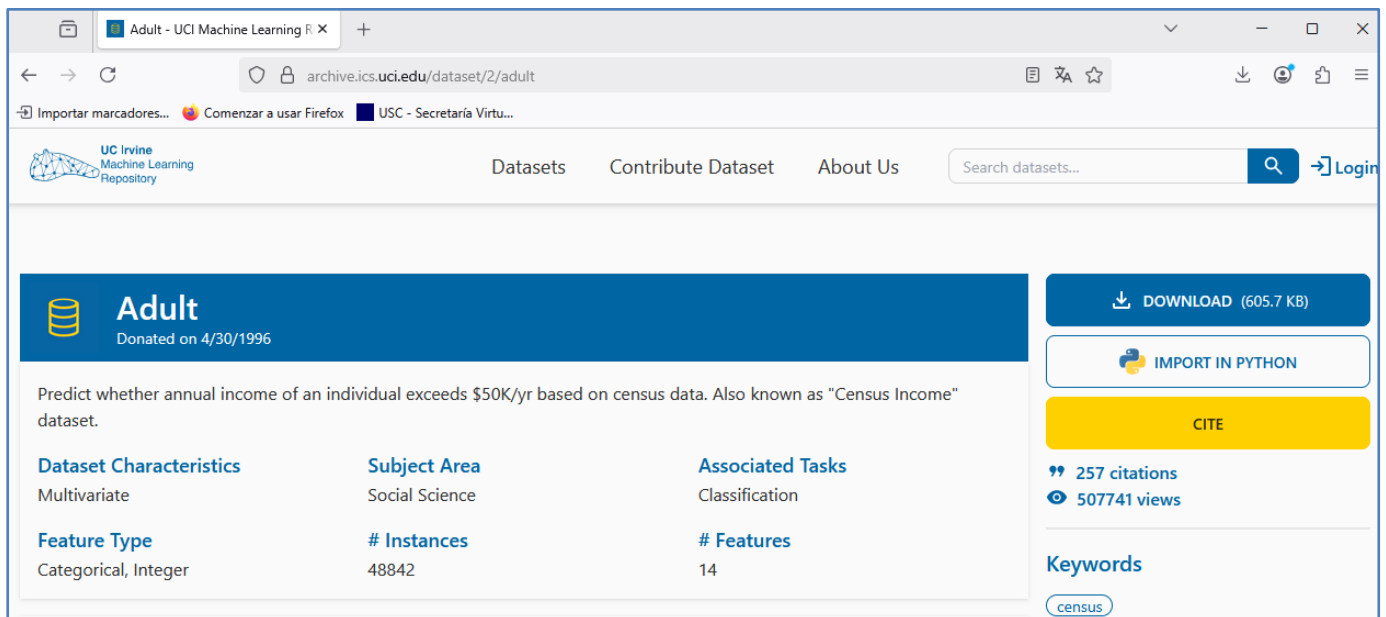
Guion de prácticas

P1. Árboles de Decisión y Selección de Modelos

digitalizada de la masa mamaria. Las 30 variables describen distintas **propiedades de los núcleos celulares** presentes en la imagen, incluyendo: a) Radio (distancia del centro a los puntos del perímetro); b) Textura (desviación estándar de los valores en escala de grises); c) Perímetro; d) Área; e) Suavidad (variación local de las longitudes de los radios); f) Compactación ($\text{perímetro}^2 / \text{área} - 1.0$); g) Concavidad (gravedad de las partes cóncavas del contorno); h) Puntos cóncavos (número de porciones cóncavas del contorno); i) Simetría; j) Dimensión fractal.

Para cada imagen se calcularon el **valor medio** (mean), **la desviación típica** (sd) **y el peor valor** (worst) de cada una de las propiedades anteriores, dando lugar a las 30 variables numéricas que contiene el *dataset*. Por ejemplo, la columna 3 es el valor medio del radio, la 13 es su desviación típica y la 23 es el peor radio.

- En el segundo problema, se tratar de determinar si una persona ganará (o no) más de 50k dolares al año, según sus datos censales. En esta práctica, emplearéis un conjunto de datos (*dataset*) que está basado en “Adult” (<https://doi.org/10.24432/C5XW20>). Este es un problema de clasificación binaria con 14 atributos (edad, ocupación, sexo, etc.) y más de 48k casos.



The screenshot shows the UC Irvine Machine Learning Repository page for the 'Adult' dataset. The page includes a header with navigation links (Datasets, Contribute Dataset, About Us) and a search bar. The main content area displays the dataset name 'Adult', its donation date (4/30/1996), and a brief description: 'Predict whether annual income of an individual exceeds \$50K/yr based on census data. Also known as "Census Income" dataset.' Below this, there are three columns of metadata: Dataset Characteristics (Multivariate), Subject Area (Social Science), Associated Tasks (Classification), Feature Type (Categorical, Integer), # Instances (48842), and # Features (14). On the right side, there are buttons for 'DOWNLOAD (605.7 KB)', 'IMPORT IN PYTHON', and 'CITE'. At the bottom right, it shows '257 citations' and '507741 views'. A 'Keywords' section at the bottom lists 'census'.

Planteamiento y desarrollo de la práctica

1. Pre-procesado de los datos

En primer lugar, debéis descargar del aula virtual los ficheros csv con los datos de entrenamiento para los dos problemas bajo estudio (*BreastCancer* y *Adult*).

El siguiente paso es cargar los datos en vuestro JupyterNotebook (P1.ipynb) con **pandas** (`read_csv`) y realizar una tarea de inspección/análisis (podéis usar `pairplot`) para gestionar **posibles errores** que pueda haber en el *dataset* (valores repetidos/duplicados, valores faltantes, etc.). Si fuera el caso,

Guion de prácticas

P1. Árboles de Decisión y Selección de Modelos

tendréis que aplicar las siguientes estrategias de gestión de **valores faltantes** (“*missing values*”) basadas en `SimpleImputer` (mean / median) y `KNNImputer`. Si algún atributo tiene 500 o más casos faltantes podéis eliminarlo del `DataFrame`. También podéis explorar la opción de **normalizar los datos**, es decir, escalarlos a $[0, 1]$. Recordad eliminar variables innecesarias (por ejemplo, “id” en *BreastCancer*). Además, la variable a predecir debe separarse de los atributos de entrenamiento y si es textual debería renombrarse como una variable categórica con 0 asociado a la clase minoritaria y 1 asociado a la clase mayoritaria. Se recomienda usar diccionarios para sustituir las etiquetas textuales de variables categóricas por valores numéricos correlativos (por ejemplo, sex: {Female, Male} se podría reemplazar por sex: {0, 1}).

2. Creación y validación de modelos

En esta etapa deberéis crear, ajustar y validar los **modelos de predicción** (cada uno asociado a una configuración de parámetros diferente para KNN o Árboles de Decisión) para los dos problemas considerados.

Podéis utilizar las funciones que proporciona *sklearn*, para ajustar los siguientes hiper-parámetros de KNN (el resto se dejan con el valor por defecto de *sklearn* que sea consistente para los valores seleccionados):

- El número de vecinos (`n_neighbors`).
- El peso que pondera la importancia de los vecinos (`weights`): uniform, distance.

En el caso de los árboles de decisión, trabajad con `sklearn.tree.DecisionTreeClassifier` (usando los valores por defecto de los parámetros, excepto para *criterion* que debe tomar el valor ‘entropy’) y utilizad como hiper-parámetros las variables `min_samples_split` y `max_depth` (que permitirá ajustar el tamaño del árbol).

Debéis realizar el **ajuste de hiper-parámetros** de los modelos simultáneamente con la prueba de estos (“Test”) mediante **validación cruzada**, utilizando el número de particiones (“*folds*”) que consideréis oportuno. Es recomendable que antes de aplicar la validación cruzada dividáis el *dataset* en *training* y *test*. Y repitáis el procedimiento varias veces para estimar la generalidad del mejor modelo según su media y desviación sobre los datos de prueba.

Antes de empezar, se recomienda leer la documentación de *sklearn* asociada a las siguientes funciones: `train_test_split`, `cross_validate`, `StratifiedKFold`, `GridSearchCV`.

Una vez ajustados todos los modelos, tenéis que seleccionar el modelo que consideráis mejor para cada problema, según la regla de 1 desviación. Discutid qué modelo es el mejor atendiendo a distintas métricas de calidad. Dibujad la gráfica del error de entrenamiento con validación cruzada frente al valor de cada hiper-parámetro: ¿Cuál es el menor error de validación cruzada, su desviación estándar y el valor del hiper-parámetro para el que se consigue? ¿Cuál es el valor del hiper-parámetro si se aplicase la regla de una desviación estándar? Recordad que en caso de que haya varios modelos con error mínimo, debe seleccionarse siempre el más simple.

Además, podéis aplicar validación estadística para verificar si entre los distintos modelos existen diferencias significativas (para seleccionar los algoritmos apropiados podéis seguir los consejos de <https://tec.citius.usc.es/stac/>).

Guion de prácticas

P1. Árboles de Decisión y Selección de Modelos

Después, debéis hacer un último entrenamiento del modelo seleccionado con el conjunto de datos completo para cada uno de los problemas estudiados, con lo que tendréis el modelo final. Dejad el código preparado para que sea sencillo cargar y ejecutar ese modelo con un **nuevo conjunto de prueba** que se os facilitará el día de la evaluación práctica. Cuanto más modular (organizado, documentado y estructurado en funciones) sea el código desarrollado, mejor.

Por último, escribid una sección de discusión y conclusiones en la que resumáis lo que habéis aprendido. Analizad la relación entre precisión e interpretabilidad (tamaño y profundidad) en los modelos de árbol construidos. Por último, comparad los resultados obtenidos por KNN y árboles en los dos problemas considerados, y justificad si los resultados se ajustan a lo que esperabais o no.

Realización, entrega y evaluación de la práctica

- Realizaréis el trabajo en **equipos de tres personas**, que trabajaréis conjuntamente según las pautas siguientes:
 - La entrega será única, siendo responsables por igual todos los integrantes del equipo. Es suficiente con que uno de los miembros del equipo suba la entrega al campus virtual.
 - Debéis explicar y justificar suficientemente las diferentes tareas realizadas y cada decisión tomada a lo largo del desarrollo de la práctica. En particular, debéis aportar información suficiente para que se pueda entender para cada problema cómo se realizó primero el pre-procesado de los datos, después el ajuste de los hiper-parámetros de los modelos, y finalmente la selección del mejor modelo en cada caso.
- Dedicaremos a esta práctica las cuatro primeras sesiones interactivas.
- La fecha límite de entrega será la indicada en el aula virtual.
- La entrega consistirá en un único fichero comprimido en **formato .zip** que contenga una memoria (en PDF) describiendo el trabajo realizado, justificando cada decisión tomada, y todos los ficheros con la resolución de los ejercicios indicados en el guion:
 - El fichero fuente y el HTML del notebook Python con la realización de la práctica, y todo el código debidamente documentado.
 - Todos los ficheros auxiliares necesarios.
- La entrega se calificará sobre un máximo de 10 puntos. En la evaluación tendremos en cuenta:
 - La claridad y calidad de las descripciones, explicaciones y justificaciones en el informe.
 - Calidad de los modelos construidos y claridad del código desarrollado.
 - La calificación de la entrega tiene dos componentes:
 - **C1 (70%):** Una componente común idéntica para todos los miembros del equipo, que se corresponde con la parte conjunta del trabajo, la memoria escrita y el código desarrollado.
 - **C2 (30%):** Una componente individual, que se corresponde con el resultado de las respuestas al test de evaluación de la práctica 1 (que incluirá cuestiones asociadas a los ejercicios realizados en las 4 sesiones interactivas correspondientes a la P1).