# Pandas

Un rápido y eficiente objeto DataFrame para la manipulación de datos con indexación integrada;

Herramientas para leer y escribir datos entre estructuras de datos en memoria y diferentes formatos: CSV y archivos de texto, Microsoft Excel, bases de datos SQL y el rápido formato HDF5;

Repositorios

<https://www.kaggle.com/datasets>

Se trata de un recurso muy útil donde puedes encontrar tanto ficheros de datos, normalmente en formato CSV, como notebooks, uno de los más actuales esta relacionado con el coronavirus.

Además esta página hace competiciones con premios, donde puedes subir tus modelos y notebooks y ver en qué posición te encuentras.

Por ejemplo el primero de ellos para detectar caras falsas tiene un premio de 1 millon.

Google:

Con Búsqueda de Datasets, los usuarios pueden buscar conjuntos de datos almacenados en la Web por palabras clave. Esta herramienta muestra información sobre conjuntos de datos alojados en miles de repositorios de Internet; de esta forma, cualquier usuario puede acceder a ellos y aprovechar la información que contienen.

UCI

Actualmente tienen 497 conjuntos de datos. En este repositorio aparece el nombre del dataset, el tipo de datos que tiene, la tarea para que se a subido (clasificación, regresión, recomendación, clustering, etc), el tipo de los atributos, el número de instancias, el número de atributos y el año.

# Fichero de datos

https://drive.google.com/drive/folders/1AUrXVNigp8vQdiwgfRYe1eYdvR0oHQJt?usp=sharing

<https://archive.ics.uci.edu/ml/datasets/Iris>

Este es un ejemplo clásico de clasificación. Este conjunto de datos es sobre las características de un tipo de flor (iris).

En los atributos de entrada tiene:

1 . Longitud del sépalo en cm

2. Anchura del sépalo en cm

3. Longitud del pétalo en cm

4. Anchura del pétalo en cm

5. clase:

-- Iris Setosa

-- Iris Versicolor

-- Iris Virginica

# Scikitlearn

Es una libreria de código abierto para el aprendizaje automático que soporta el aprendizaje supervisado y no supervisado. Esta librería también proporciona varias herramientas para el ajuste de modelos, preprocesamiento de datos, selección y evaluación de modelos, y muchas otras utilidades.

Lo primero que vamos a importar es un árbol de decisión.

Se trata de un algoritmo de clasificación, que tiene representación en forma de árbol. En este caso vamos a usar el DecisionTreeClassifier

Graphviz es un software de código abierto de visualización de gráficos. La visualización de gráficos es una forma de representar información estructural como diagramas de gráficos abstractos y redes. Tiene importantes aplicaciones en redes, bioinformática, ingeniería de software, diseño de bases de datos y de páginas web, aprendizaje automático y en interfaces visuales para otros dominios técnicos.

Esta forma de mostrar los datos no solo podemos ver como de bueno es el árbol.

Validación.

Para saber si los modelos que estamos creando son buenos es necesario testearlo de alguna manera.   
La forma más sencilla es dividir el conjunto de datos entre entrenamiento y test.

Cuando hacemos el *predict()* sobre el conjunto de test y obtenemos las predicciones, las podemos comprobar y **contrastar con los valores reales** almacenados en *y\_test* y hallar así la métrica que usamos. Los resultados que nos puede dar serán:

1. Si el accuracy en Test es <<cercano>> al de Entrenamiento (dijimos 75%) por ejemplo en este caso si estuviera entre 65 ú 85% quiere decir que nuestro modelo entrenado está generalizando bien y lo podemos dar por bueno (siempre y cuando estemos conformes con las métricas obtenidas).
2. Si el Accuracy en Test es muy distinto al de Entrenamiento tanto por encima como por debajo, nos da un 99% ó un 25% (lejano al 75%) entonces es un indicador de que nuestro modelo no ha entrenado bien y no nos sirve. De hecho este podría ser un indicador de [Overfitting](https://www.aprendemachinelearning.com/que-es-overfitting-y-underfitting-y-como-solucionarlo/).

Problemas:

Tenemos pocos datos, la división no es justa respecto a a algún criterio. Para crear bien los modelos tenemos que

### Cross-Validation: K-fold con 5 splits

Lo que hacemos normalmente al entrenar el modelo es pasarle los 8.000 registros y que haga el fit(). Con [K-Folds](https://scikit-learn.org/stable/modules/cross_validation.html#k-fold) -en este ejemplo de 10 splits- para entrenar, en vez de pasarle todos los registros directamente al modelo, haremos así:

* Iterar 10 veces:
  1. Apartaremos 1/5 de muestras, es decir 1600.
  2. Entrenamos al modelo con el restante 4/5 de muestras = 6400.
  3. Mediremos el accuracy obtenido sobre las 1600 que habíamos apartado.
* Esto quiere decir que hacemos 5 entrenamientos independientes.
* El Accuracy final será el promedio de las 5 accuracies anteriores.

# Recapitulación de clasificación:

Hemos visto como importar datos de un fichero, ya sea de manera online o subiéndolos a notebook. Hemos visto como pandas nos puede ayudar con el almacenamiento, transformación y visualización de los datos.

Los métodos que nos proporciona Scikit para importar algoritmos de clasificación, como podemos visualizar los arboles de decisión y hemos aprendido como evaluarlos.

Además hemos visto el problema de las clases desbalanceadas y cómo podemos solucionarlo para disminuir el número de errores.

¿Qué pasa si no tenemos variables nominales? Y queremos predecir tiempo, o dinero

# Regresion

Medidas adicionales

En [estadística](https://es.wikipedia.org/wiki/Estad%C3%ADstica), el **coeficiente de correlación de**[**Pearson**](https://es.wikipedia.org/wiki/Karl_Pearson) es una medida lineal entre dos [variables aleatorias](https://es.wikipedia.org/wiki/Variable_aleatoria) [cuantitativas](https://es.wikipedia.org/wiki/Variable_estad%C3%ADstica#Seg%C3%BAn_la_medici%C3%B3n). A diferencia de la [covarianza](https://es.wikipedia.org/wiki/Covarianza), la correlación de [Pearson](https://es.wikipedia.org/wiki/Karl_Pearson) es independiente de la escala de medida de las variables.

La **curtosis** de una variable estadística/aleatoria es una característica de forma de su distribución de frecuencias/probabilidad.

Según su concepción clásica, una curtosis grande implica una mayor concentración de valores de la variable tanto muy cerca de la media de la distribución (pico) como muy lejos de ella (colas), al tiempo que existe una relativamente menor frecuencia de valores intermedios. Esto explica una forma de la distribución de frecuencias/probabilidad con colas más gruesas, con un centro más apuntado y una menor proporción de valores intermedios entre el pico y colas.

Abrir ficheros

Explicar

# Recapitulación de Regresión: