Desafío - Regularización

- Para realizar este desafío debes haber revisado la lectura y videos correspondiente a la unidad.
- Crea una carpeta de trabajo y guarda todos los archivos correspondientes (notebook y csv).
- Una vez terminado el desafío, comprime la carpeta y sube el `.zip` a la sección correspondiente.

Descripción

- En esta ocasión utilizaremos datos de la composición química de muestras de vino.
- Nuestro propósito será tratar de predecir su calidad a partir de su composición química.
- Comience su estudio realizando un breve análisis exploratorio y descriptivo:
  - Identifique el tipo de datos de cada columna.
  - Identifique outliers univariados y multivariados. Los primeros los puede encontrar realizando boxplots de cada atributo, los segundos estudiando los scatterplot entre pares de atributos.
  - Encuentre la cantidad de valores faltantes en cada columna.
  - Grafique la distribución empírica de cada atributo.

Ejercicio 1: Preparación del Ambiente de Trabajo

- Importe las librerías básicas
- Importe los métodos asociados a regularización
- Importe la base de datos

Ejercicio 2: Análisis exploratorio

- Identifique el tipo de datos de cada columna.
- Identifique outliers univariados y multivariados. Los primeros los puede encontrar realizando boxplots de cada atributo, los segundos estudiando los scatterplot entre pares de atributos.

Ejercicio 3: Regularización de modelos

- Genere tres modelos, correspondientes a `RidgeCV`, `LassoCV` y `ElasticNetCV`. Para cada uno de ellos implemente los siguientes pasos:
    - Entrene los modelos con 10 validaciones cruzadas.
    - Extraiga los coeficientes estimados, el mejor valor alpha, MAE (`median_absolute_error`) y RMSE (`mean_squared_error`) asociado a cada modelo.
    - En base a la información solicitada, responda lo siguiente:
    - ¿Qué modelo presenta un mejor desempeño en base a las métricas?
    - ¿Qué atributos mantienen un comportamiento similar a lo largo de cada modelo?

Sobre los métodos `RidgeCV`, `LassoCV` y `ElasticNetCV`

- Para implementar la búsqueda de hiperparámetros con validación cruzada, utilizaremos la clase `sklearn.linear_model.{RidgeCV, LassoCV, ElasticNetCV}`.
- Por defecto, la clase necesita de una lista de valores `alphas` que representa el hiperparámetro de validación. Si no se define la lista, el método incorporará un array con valores por defecto.
- Por defecto la clase tendrá una opción `scoring` que permitirá definir la métrica con la cual se evaluará el mejor hiperparámetro. Para el caso de un problema de regresión, si no se define, la métrica será el error cuadrático promedio negativo (más alto es mejor).
- Una vez entrenado el modelo, se puede reportar el mejor alpha con `modelo_entrenado.alpha_`, el valor de los coeficientes con `modelo_entrenado.coef_` y los valores de la métrica de desempeño con `modelo_entrenado.cv_values_`