Boletín 1: evaluación y selección de modelos
Boletín 2: métodos basados en vecinos más próximos

Para la realización de las prácticas de esta segunda parte de la materia se utilizará scikit-learn, una librería de aprendizaje estadístico en Python, a través de **Jupyter Notebooks**. **La ejecución se realizará en el CESGA** siguiendo los pasos indicados en el archivo CESGA.pdf.

introduction.ipynb

En primer lugar, abre mediante *ipython notebook* el fichero **introduction.ipynb**, donde se describen algunas de las operaciones básicas necesarias para trabajar con scikit-learn: aprenderás a cargar los datos, realizar operaciones básicas con matrices, y representaciones gráficas.

knn.ipynb

A continuación abre el fichero **knn.ipynb**. En este archivo se realiza la experimentación con un algoritmo sobre un conjunto de datos. Concretamente, se ha escogido el método de vecinos más cercanos, y un archivo con un problema muy simple (*toyExample.data*). Los pasos que se realizan son los siguientes:

- Carga de datos y preprocesado básico.
- División del conjunto de datos en entrenamiento y test.
- Generación de los datos sobre los que se harán las representaciones gráficas.
- Búsqueda de los mejores valores para los hiper-parámetros mediante validación cruzada.
- Generación del modelo final, test y representación gráfica.
- Guardar el modelo aprendido.

Instrucciones para la experimentación en TODOS los boletines de prácticas

En los diferentes ejercicios que se realizarán durante el curso, existen una serie de operaciones con una componente aleatoria: la división en entrenamiento y test, el aprendizaje de un modelo o incluso, en algunos casos, el test del modelo. Como norma general de experimentación es interesante asegurar la repetibilidad de los experimentos, eliminando la aleatoriedad, puesto que nos permite depurar errores, comparar modelos, etc. Además, para la evaluación de los boletines también es imprescindible eliminar esa aleatoriedad.

Para ello vamos a fijar la semilla del generador de números aleatorios, de tal manera que su secuencia sea siempre la misma. La semilla se establece mediante el comando **np.random.seed(SEED_VALUE)**, y en este boletín utilizaremos un **SEED_VALUE=1**. Será necesario utilizar este comando inmediatamente antes de cualquier operación con un componente aleatorio. Esto incluye: train_test_split(), fit(), predict(), etc. En aquellas funciones que lo admitan, sustituiremos el comando np.random.seed(SEED_VALUE) por el argumento **random_state=SEED_VALUE**.

Boletín

1. Dado el siguiente conjunto de datos de clasificación con 6 observaciones, 3 variables de entrada y una variable de salida:

Observación	X_1	X_2	X_3	Y
1	0	3	2	1
2	3	0	3	0
3	0	3	-1	0
4	3	0	0	1
5	1	2	1	1
6	2	1	0	0

   Suponiendo que se quiere hacer la predicción de la variable de salida para X_1=0, X_2=0, X_3=0 mediante KNN.

   a. Computar la distancia entre cada observación y el punto de test.

   b. ¿Cuál es la predicción para K=1? ¿Por qué?

   c. ¿Cuál es la predicción para K=3? ¿Por qué?

   Nota: este ejercicio debe hacerse sin utilizar ninguna función de scikit-learn. No es necesario estandarizar las variables.

2. Dado el problema de clasificación Blood Transfusion Service Center:

   a. Analiza las características del conjunto de datos: número y tipo de variables de entrada y salida, número de instancias, número de clases y distribución de las mismas, correlación entre las variables, valores perdidos, etc.

   b. Una de las clases que implementa el algoritmo KNN en scikit-learn es *sklearn.neighbors.KNeighborsClassifier*. Revisa los parámetros y métodos que tiene.

   c. Divide los datos en entrenamiento (80%) y test (20%).

   d. Realiza la experimentación con KNN (*KNeighborsClassifier*) usando como hiper-parámetro el número de vecinos.

      Muestra la gráfica del error de entrenamiento con validación cruzada (5-CV) frente al valor del 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? En caso de que haya varios modelos con error mínimo, debe seleccionarse siempre el más simple.

      Muestra la gráfica del error de test frente al valor del hiper-parámetro, y valora si la gráfica del error de entrenamiento con validación cruzada ha hecho una buena estimación del

error de test. ¿Cuál es el error de test para el valor del hiper-parámetro seleccionado por la validación cruzada?

3. Repite el ejercicio 2 pero para el problema de regresión Energy Efficiency con la variable de salida *cooling load*. Al ser un problema de regresión deberás utilizar *KNeighborsRegressor*, y como medida de error de entrenamiento y test el MSE.

   Nota. Al ser un problema de regresión, para estimar tanto el error de entrenamiento como el de test (MSE) es necesario *desestandarizar* los errores calculados. Para desestandarizar el campo `mean_test_score`, únicamente será necesario multiplicar cada valor por la varianza (cuadrado de la desviación estándar) de las observaciones de Y del conjunto de entrenamiento. No se debe restar la media, ya que los campos `splitX_test_score` se calculan como la diferencia entre el valor de *groundtruth* y la predicción para cada dato de test, por lo que todas las operaciones de adición o substracción ya se han tenido en cuenta. De forma similar, para desestandarizar el campo `std_test_score`, únicamente será necesario multiplicar cada valor por la varianza de las observaciones de Y del conjunto de entrenamiento.

Entregable

Se debe entregar un único fichero comprimido con el nombre *PrimerApellido_SegundoApellido.zip* (también son válidos los formatos .rar y .7z), que contenga dos archivos:

- El primer archivo debe ser de tipo pdf, y contendrá exclusivamente las respuestas a los ejercicios (incluyendo las gráficas necesarias para justificar dichas respuestas). No se incluirá en este archivo ningún otro tipo de texto.

- El segundo archivo será de tipo ipynb, y permitirá reproducir toda la experimentación realizada en el boletín.

Ejercicio 1

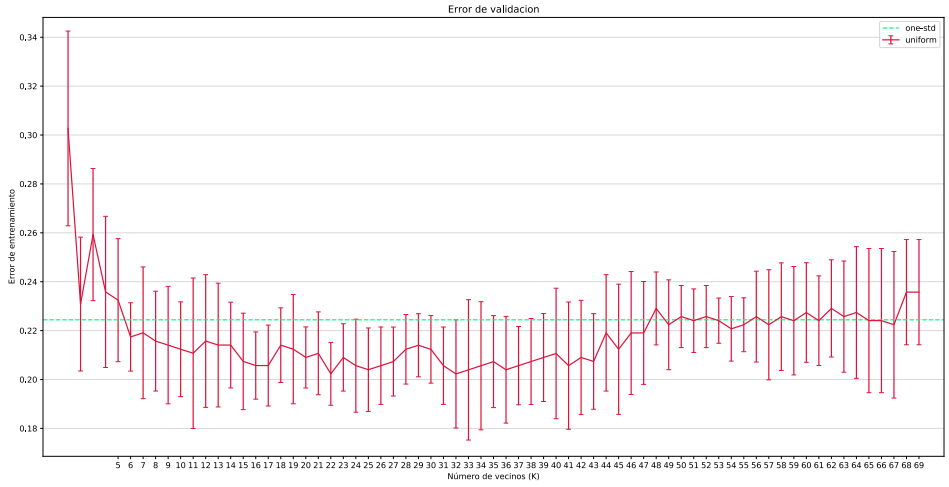
- Distancias:

X_1	X_2	X_3	X_4	X_5	X_6
$\sqrt{13} \approx 3{,}605551$	$\sqrt{18} \approx 4{,}242641$	$\sqrt{10} \approx 3{,}162278$	3	$\sqrt{6} \approx 2{,}449490$	$\sqrt{5} \approx 2{,}236068$

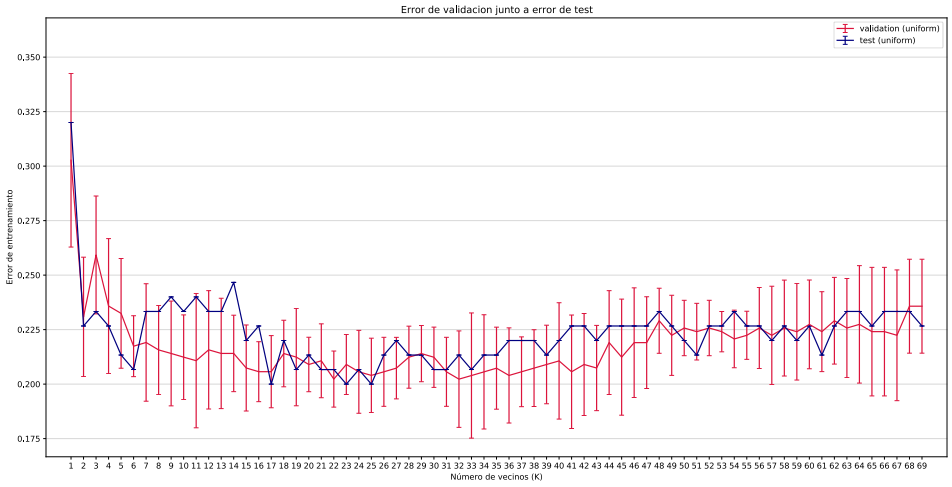
- **Para $K = 1$:** $Y = 0$ (muestra 6).
- **Para $K = 3$:** $Y = 1$ (muestras $6, 5, 4$).

Ejercicio 2

- **Menor error de validación cruzada, su desviación estándar y valor de K:** $\Delta = 0{,}202283$, $\sigma = 0{,}022126$, $K = 32$.
- **Con regla de una desviación estándar:** $\Delta = 0{,}219034$, $\sigma = 0{,}021042$, $K = 47$.

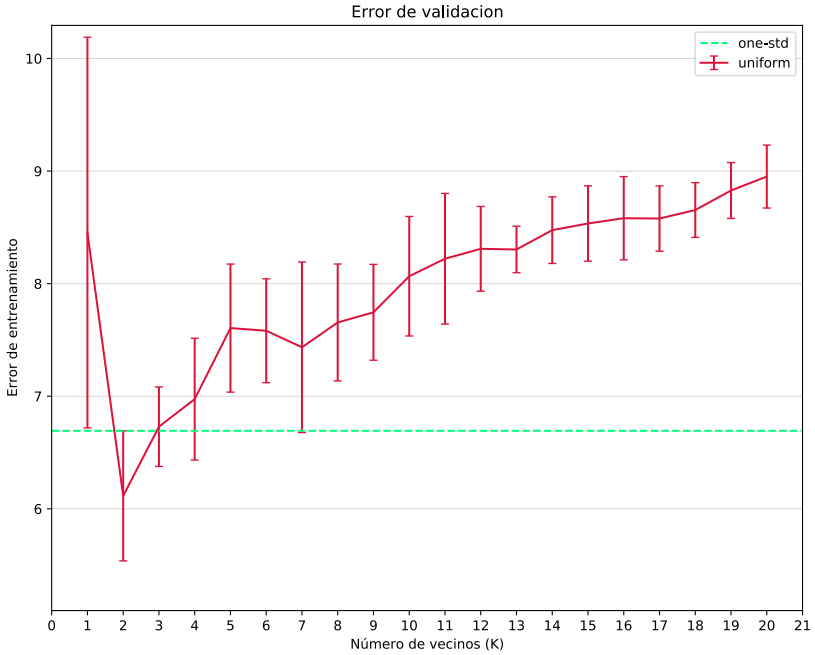


- **Error de test para el K de validación cruzada:** $\Delta = 0{,}213333$, $K = 32$.



Ejercicio 3

- **Menor error de validación cruzada, su desviación estándar y valor de K:** MSE $=$ 6,115019, $\sigma = 0{,}577814$, $K = 2$.
- **Con regla de una desviación estándar:** MSE $= 6{,}115019$, $\sigma = 0{,}577814$, $K = 2$.



- **Error de test para el K de validación cruzada:** MSE $= 12{,}588656$, $K = 2$.