TALLER DE PROCESAMIENTO DE SEÑALES

1er Cuatrimestre 2026 - Trabajo Práctico Nº 6 - KMeans

IMPORTANTE: En este ejercicio solamente se podrá importar numpy y matplotlib.pyplot(el resto deberá ser implementación propia).

---

1. Se desea comprimir la imagen `goku_vs_piccolo.jpeg` a 16 colores, utilizando el algoritmo K-Means.

   (**a**) Graficar la imagen en cuestión.

   (**b**) Tomando cada pixel como muestras diferentes, implementar un K-means de 16 clusters. El código debe estar estructurado de la siguiente manera:

   ```
   class Kmeans:
       # Inicializar atributos y declarar hiperparámetros
       def __init__(self,...

       # Etapa de entrenamiento
       def fit(self,X):

       # Etapa de testeo
       def predict(self,X):
   ```

   A su vez, debe poder extraer el atributo `kmeans.cluster_centers_` (centroides).

   (**c**) Utilizar los centroides como diccionario, para convertir cada pixel en un centroide (utilizando el algoritmo previamente entrenado). Graficar la imagen ya codificada.

   (**d**) Calcular la cantidad de bits necesarios para guardar la imagen antes y después de comprimirla (teniendo en cuenta tanto el etiquetado como los centroides).

2. Se desea caracterizar distintos tipos de consumos eléctricos. La base de datos `consumo_electrico.npy` contiene datos diarios de consumo de diferentes hogares de la ciudad de Buenos Aires.

   (**a**) Graficar las series de tiempo.

   (**b**) Tomando señales de 24hs de duración, entrenar un algoritmo K-Means para agrupar los consumidores en tres clusters. Debe reutilizar el código definido en el punto 1.

   (**c**) Graficar las series de tiempo utilizando un color diferente para cada cluster.

   (**d**) Graficar los centroides. ¿Como los interpreta?