



**Universidad Nacional Autónoma de México**  
**Posgrado en Ciencia e Ingeniería de la Computación**

**Asignatura: Reconocimiento de Patrones**

Práctica 3: Segmentación de patrones usando el algoritmo de  
Linde-Buzo-Gray

**Artemio Baños Gallardo**

**Profesor: Jesús Savage**

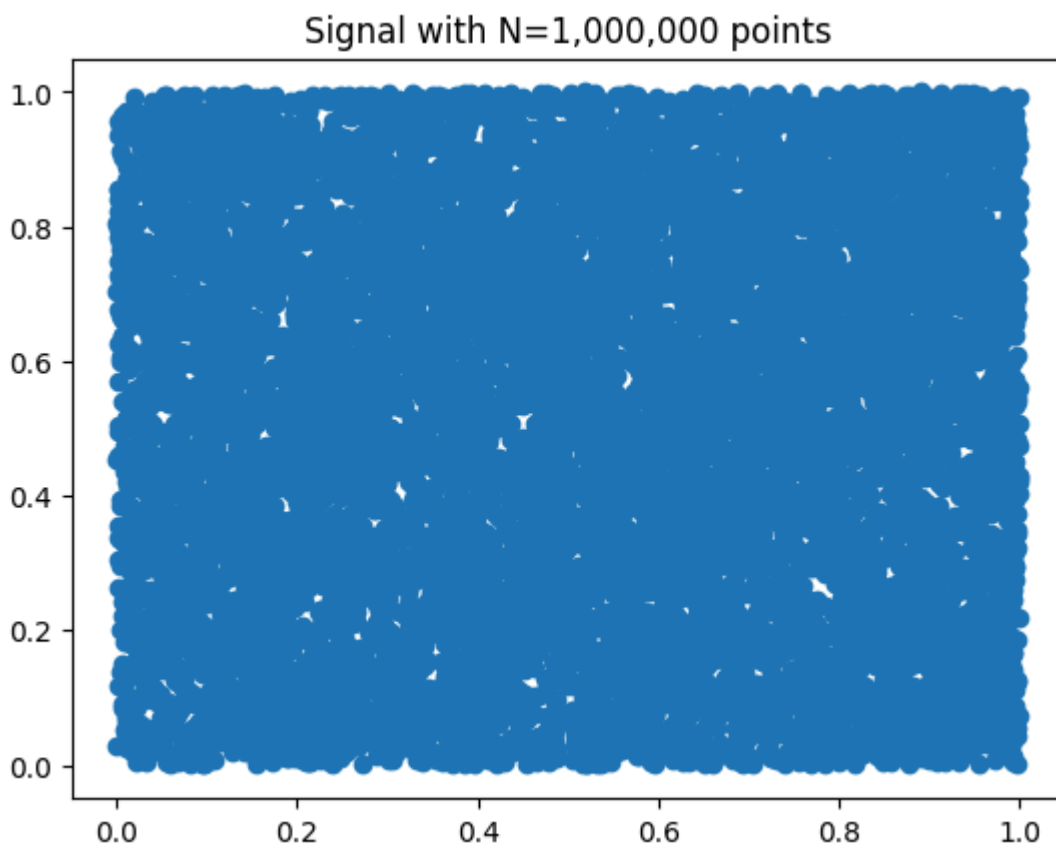
## Introducción

Una de las formas que existen para la reducción de información para transmisión sin pérdida importante es usando cuantización vectorial. En esta práctica se hará uso del algoritmo de Linde-Buzo-Gray para obtener cuantizadores variables de vectores de dos dimensiones. Después se intenta aplicar este mismo algoritmo para predicción y reducción de dimensionalidad de archivos de voz usando además los coeficientes de Wiener para reducir aún más la cantidad de información que se envíe a través de cables de comunicación o incluso antenas.

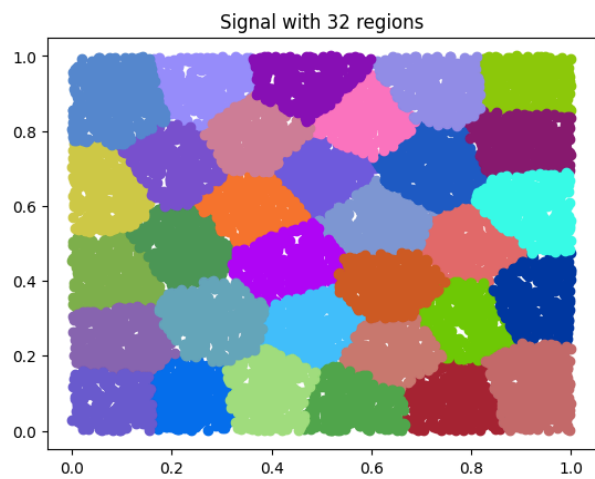
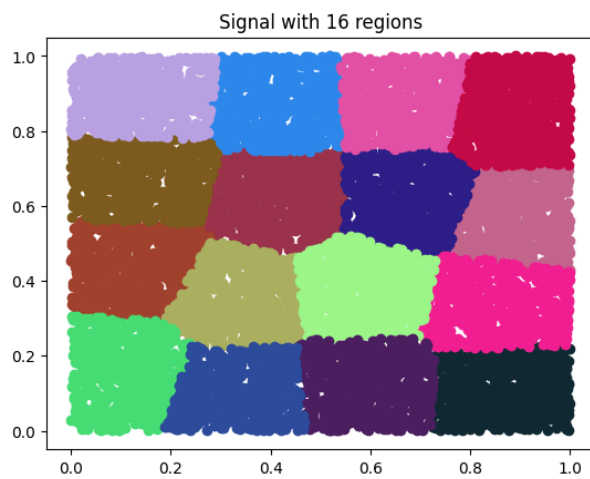
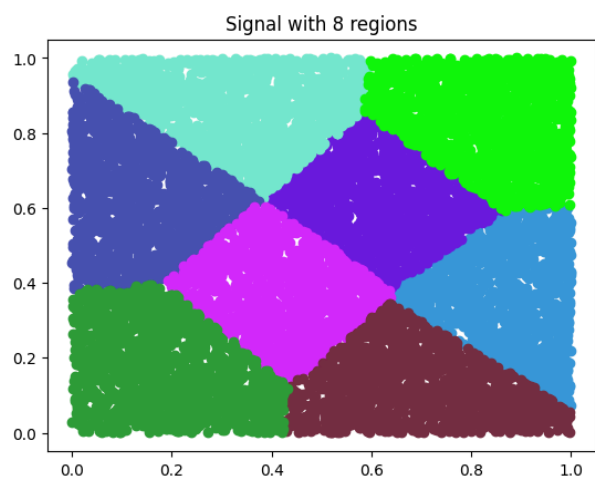
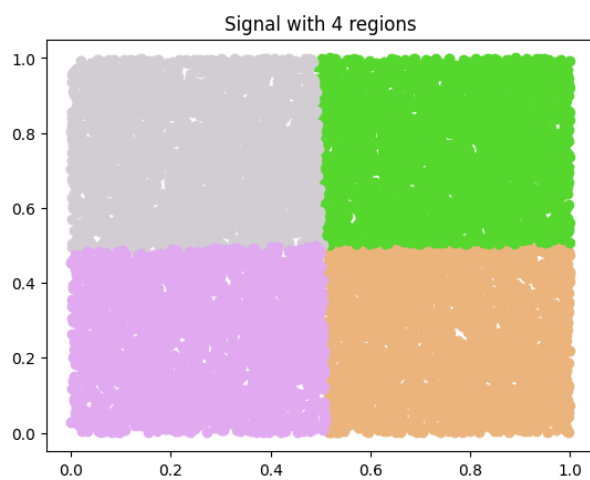
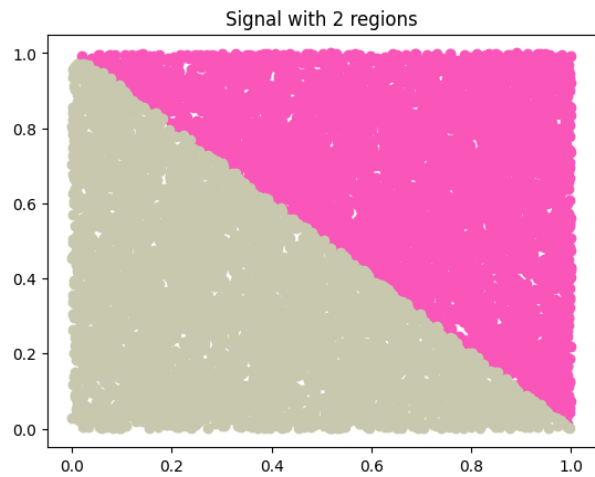
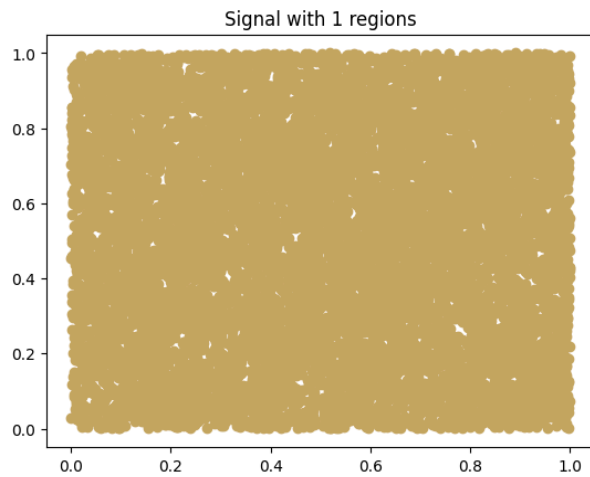
## Desarrollo y Resultados

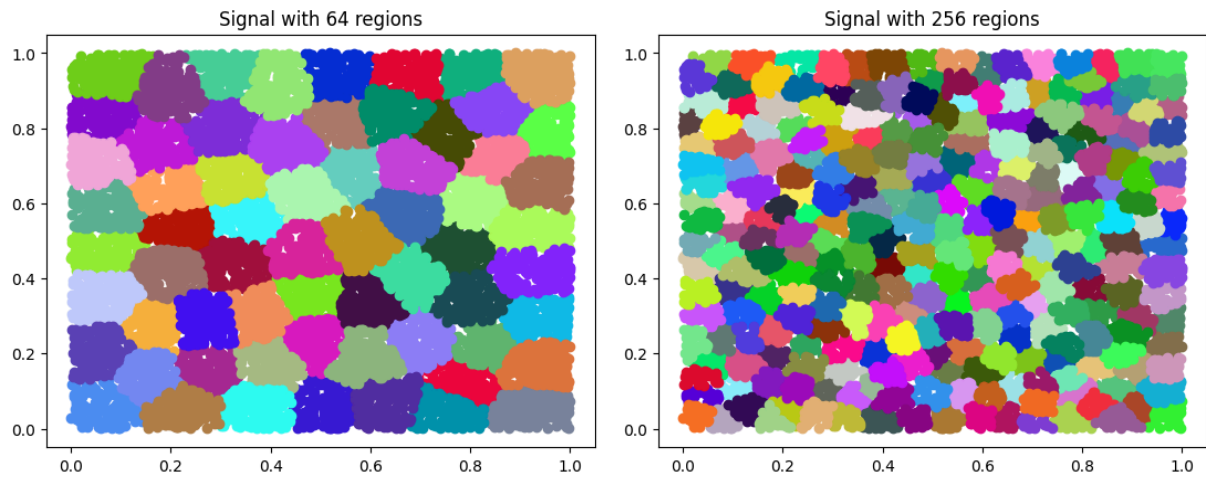
1. Generar 1,000,000 de vectores con una función de densidad uniforme de cada componente entre 0 y 1.

Nota: Para este ejercicio use el algoritmo ***lgb.py*** una implementación del algoritmo de Linde-Buzo-Gray con la función de distancia/error euclidiana de N dimensiones (en este caso a 2 dimensiones). Este se encuentra en los archivos de entrega con el mismo nombre.

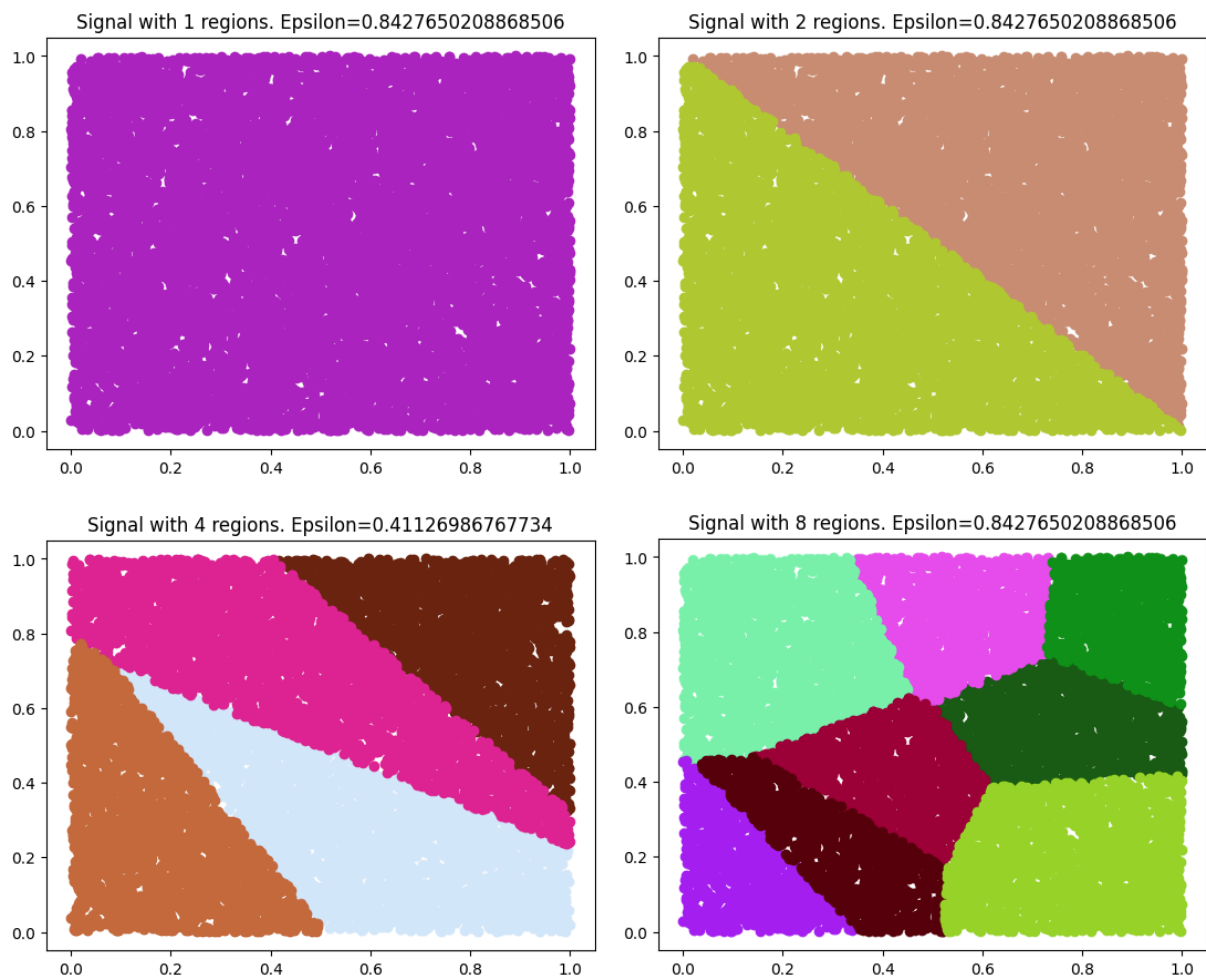


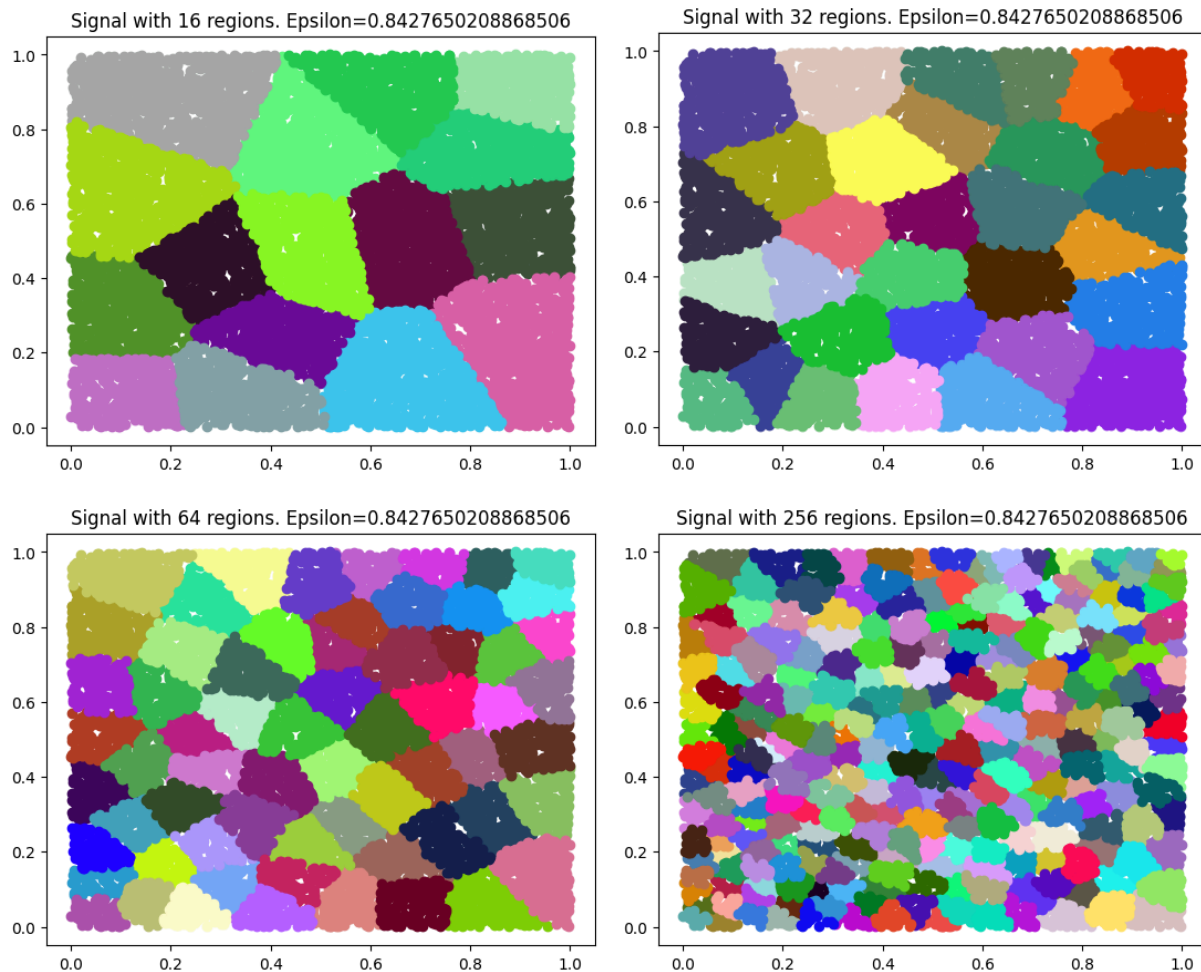
- A. Encuentre un **cuantizador vectorial usando el algoritmo de Linde-Buzo-Gray** para 1, 2, 4, 8, 16, 64 y 256 regiones. **Gráfica con colores la posición de los puntos en las diferentes regiones del cuantificador.**  $e1 = \{0.0, 0.001\}$  y una  $e2 = \{0.0, -0.001\}$





B. Repita el punto anterior modificando los centroides  $e_1$  y  $e_2$  generados aleatoriamente



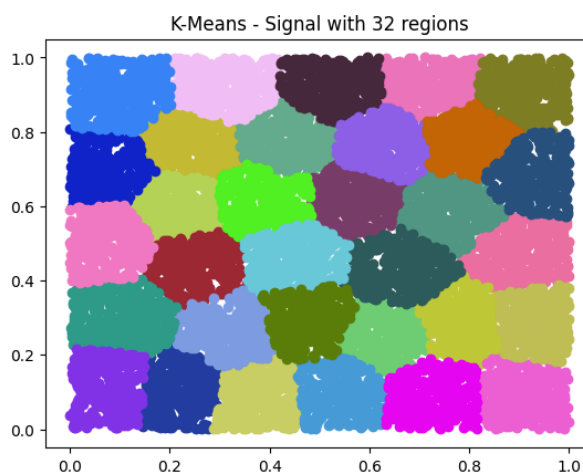
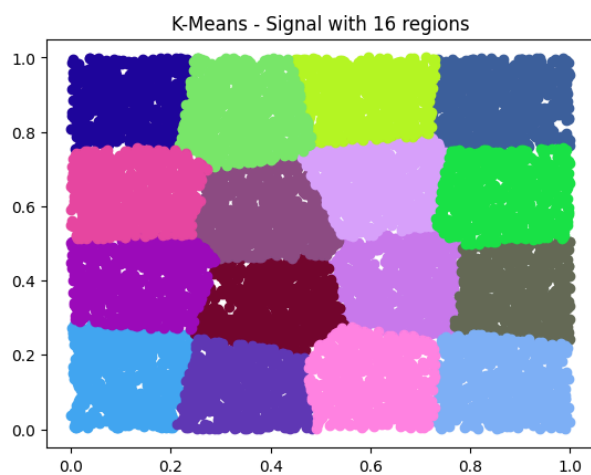
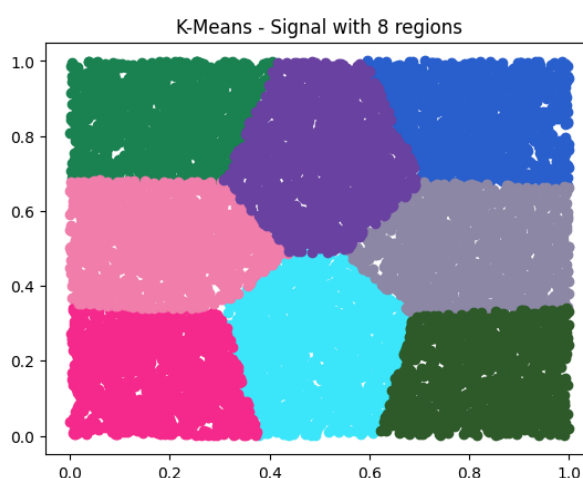
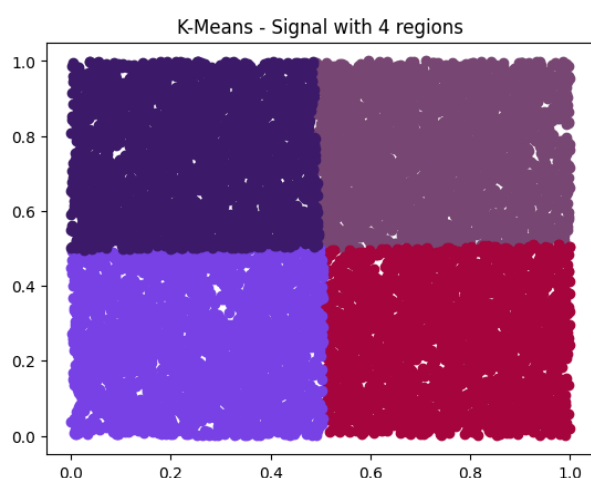
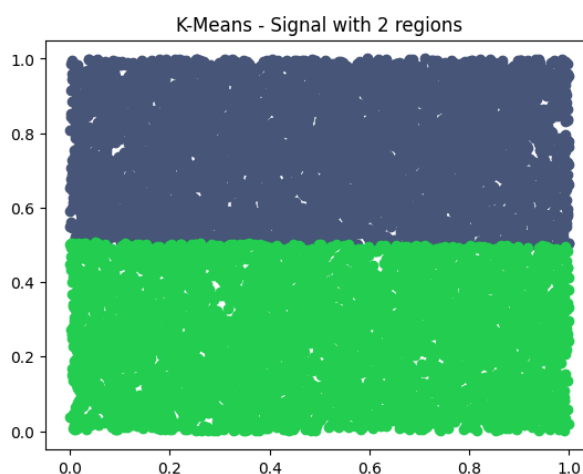
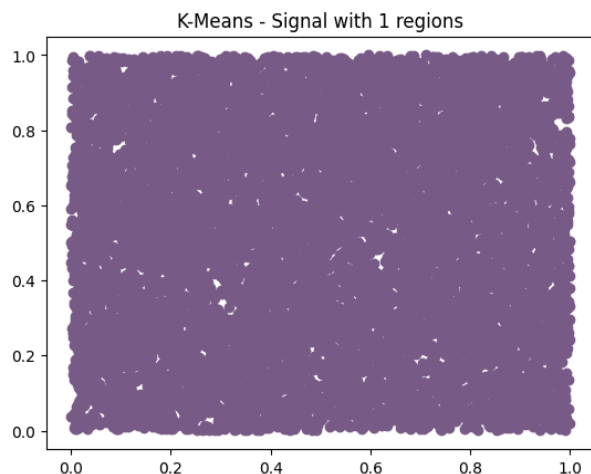


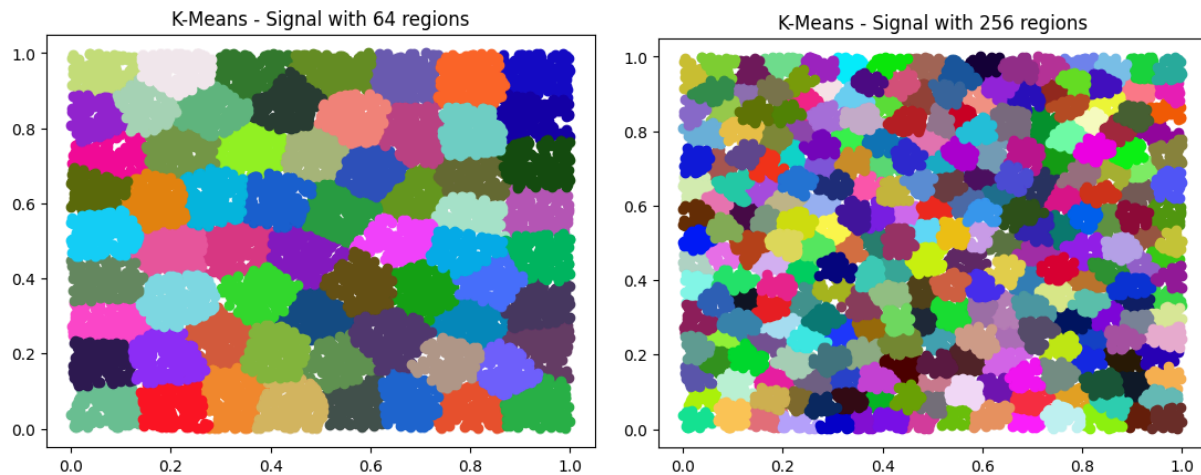
**Conclusiones 1:** Se nota que para  $\epsilon=0.001$  el tiempo de procesamiento se incrementa en 3 o 4 veces. Por el contrario para  $\epsilon=0.8428$  (Aquí redondeado por simplicidad) el tiempo de procesamiento se reduce contundentemente. Esto puede significar que el reprocesamiento de los vectores y reajuste de centroides lleva una menor cantidad de operaciones a errores cercanos a 1 o el rango de la señal, aquí  $\{0,1\}$ .

2. Con los mismos puntos generados en el punto anterior encuentre las mismas regiones con el algoritmo de K-medias, grafique estas y comparé con los obtenidos en el punto anterior.

Nota: Para este ejercicio use la implementación de K-Means [2] de SKLearn con los siguientes parámetros

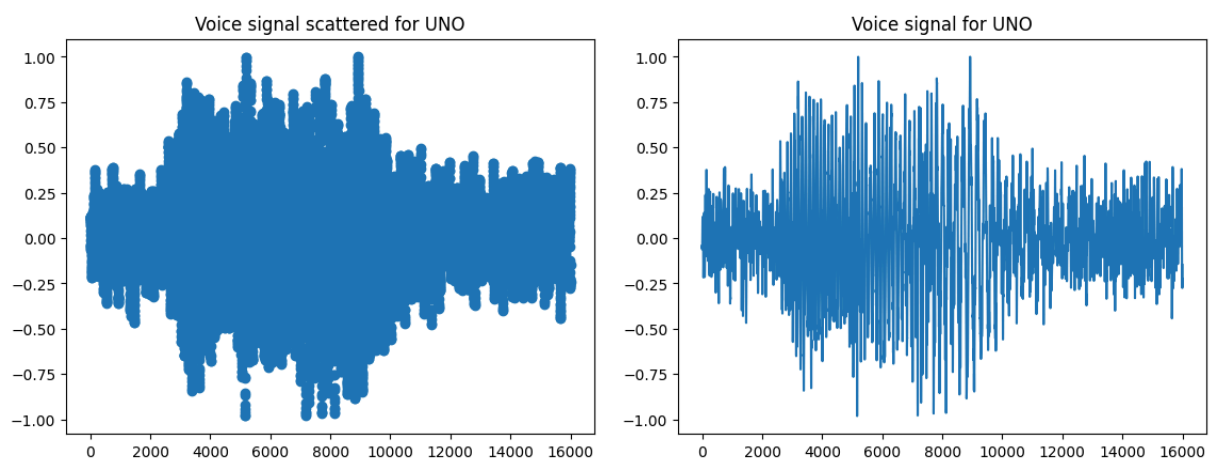
```
KMeans(n_clusters=number_of_regions,
        random_state=0, n_init="auto")
```

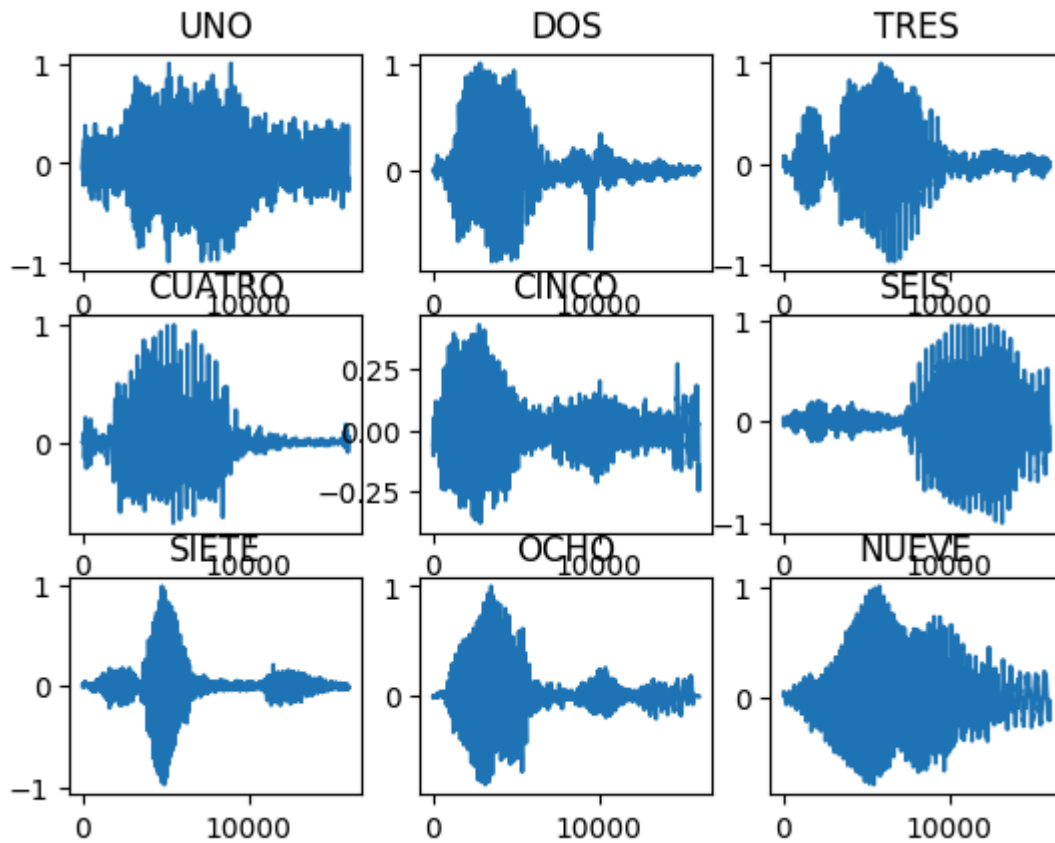




**Observaciones:** En las pruebas que realice este algoritmo es ligeramente más rapido en tiempo respecto a mi implementación del lgb.py con  $\epsilon=0.001$ . Sin embargo, para un  $\epsilon>0.001$ , el tiempo es muy similar. Por otro lado, en cuanto a la forma de las regiones este se comporta muy similar al inciso 1a con  $\epsilon=0.001$ . Pero, el cambio es notable entre ambas implementaciones para un epsilon de mayor tamaño, como se puede ver en el ejercicio 1b, y número de regiones=4

### 3. Filtro de preénfasis

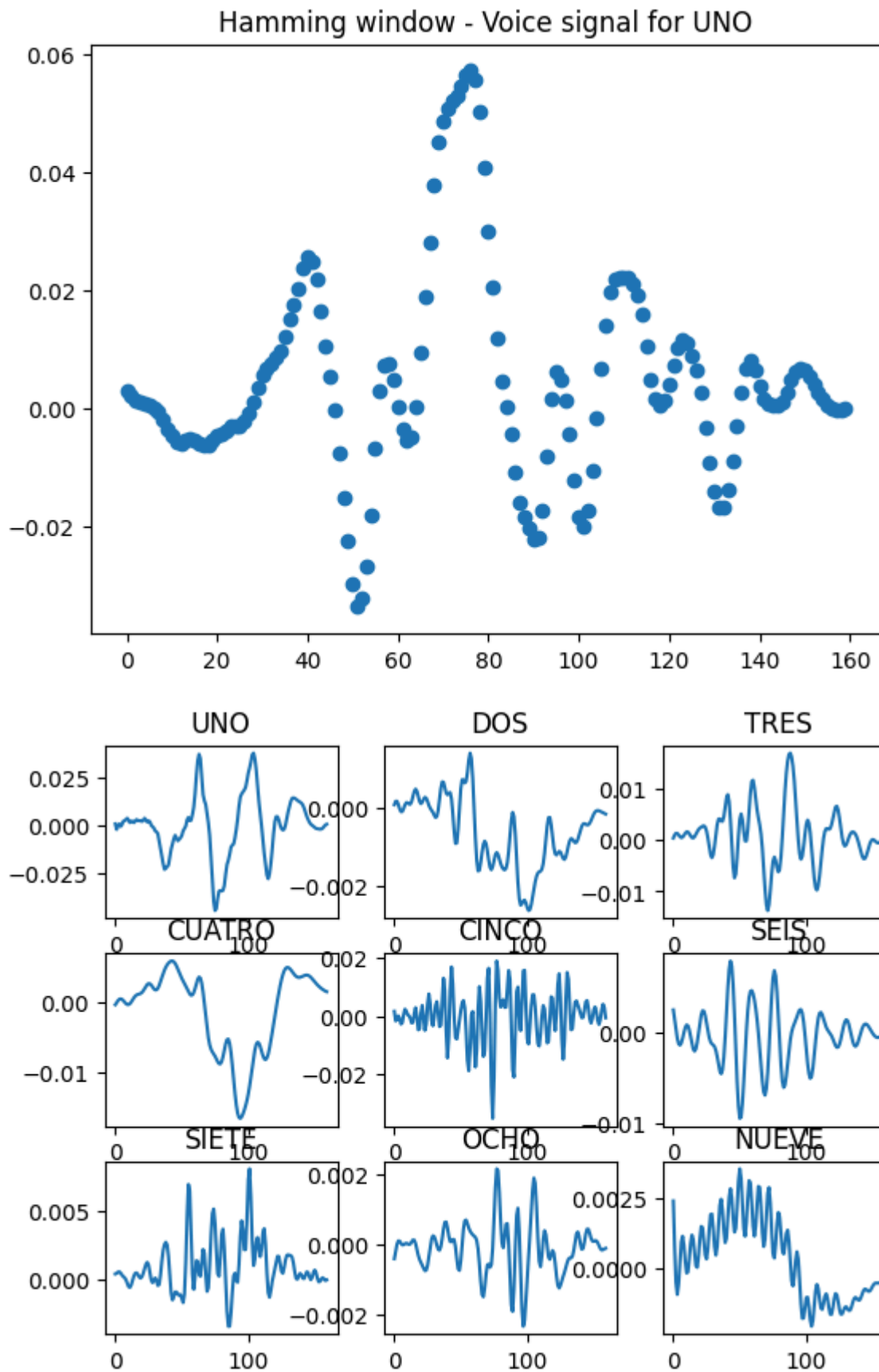




Filtro de preénfasis aplicado a todas las señales de voz

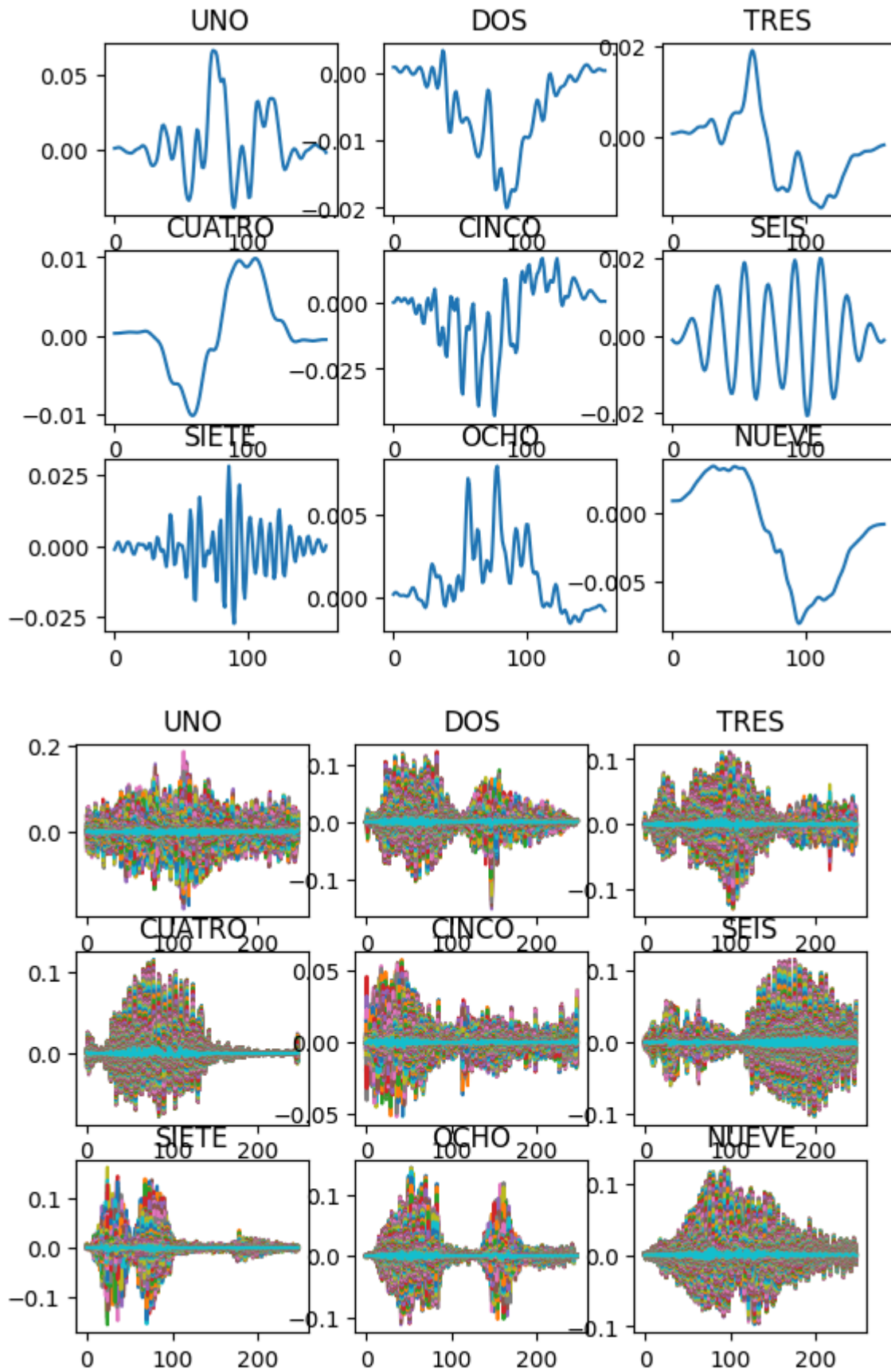
#### 4. Ventana de Hamming para las señales de voz





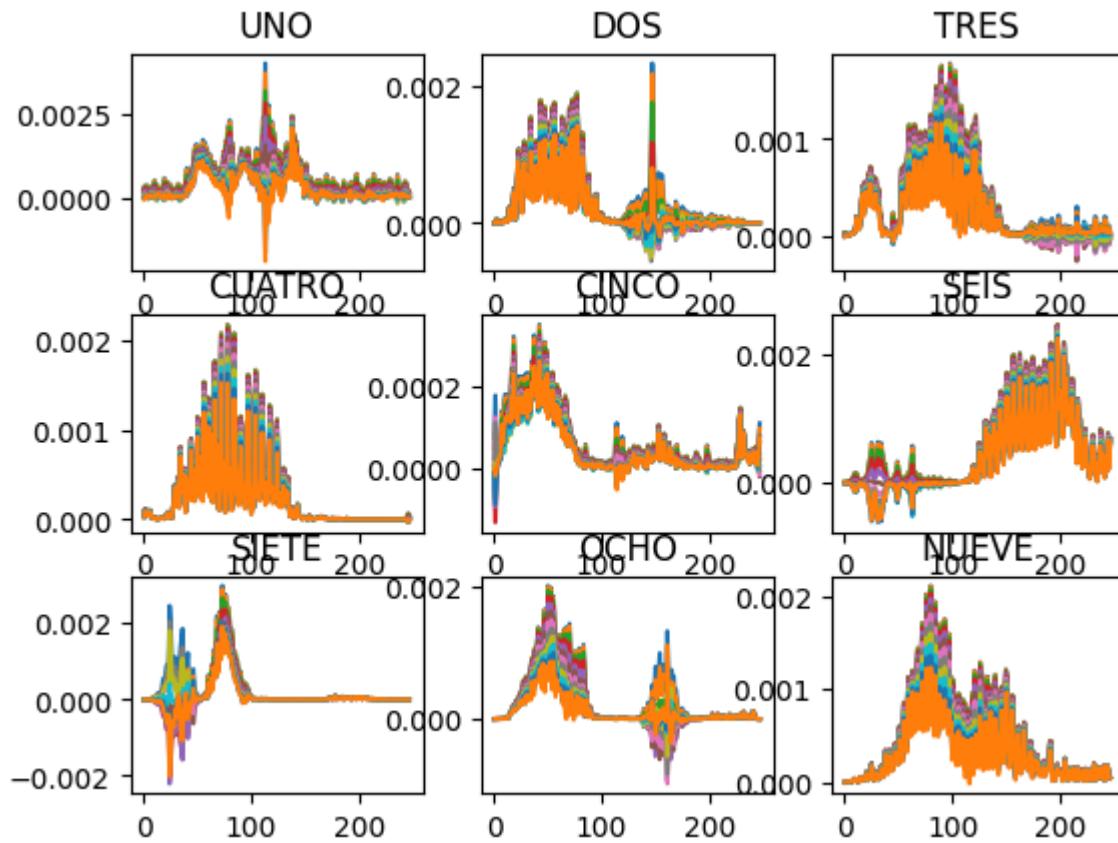
Ventana de hamming aplicado a todas las señales de voz. Aquí se puede observar la primera

ventana. Abajo se puede observar la ventana 10



En esta última foto podemos observar todas las ventanas de hamming unidas de cada una de las señales de voz.

5. Vector de correlación de orden 12 para cada señal de voz  
En estas gráficas se muestran todos los coeficientes de orden 12 para la señal. Es decir cada uno de los 12 en cada ventana de Hamming.



## Conclusiones

Las técnicas de cuantización pueden reducir enormemente el tamaño de un set de información mejorando así la transmisión de información de información cómo se pudo visualizar en la práctica puede llegarse a describir una señal de manera muy precisa usando el algoritmo de Linde-Buzo-Gray.

## Referencias:

1. Jesús Savage, Carlos Rivera [2024]. “Lección 5: Cuantización Vectorial”. Clase Procesamiento de señales. UNAM, Primavera 2024.
2. SKLearn: KMeans [2024]. Se encuentra en: [KMeans — scikit-learn 1.5.0 documentation](#)