Clase 095 — Clustering K-Means: selección de K, MiniBatch

Parte: 1 — Machine Learning Clásico · Fuente: Géron, cap. 9. Duración estimada: 70 min.

Clase 095 — Clustering K-Means: selección de K, MiniBatch

Parte: 1 — Machine Learning Clásico · Fuente: Géron, cap. 9. Duración estimada: 70 min.

Objetivo

Aplicar K-Means para segmentar datos no etiquetados, elegir el número de clusters K con criterios reproducibles (elbow, silhouette) y escalar el algoritmo con MiniBatchKMeans cuando el dataset no entra cómodo en memoria.

Resultados de aprendizaje

Al finalizar, vas a poder:

- Explicar el algoritmo de Lloyd y por qué K-Means++ mejora la inicialización aleatoria.
- Ajustar KMeans de scikit-learn fijando n_init y random_state para resultados estables.
- Elegir K combinando elbow method (inercia vs K) y silhouette score.
- Reemplazar KMeans por MiniBatchKMeans y discutir el trade-off velocidad / calidad.
- Diagnosticar por qué K-Means falla sin escalado o frente a clusters no esféricos.

Temas

1. Clustering no supervisado: planteo del problema.
2. Algoritmo de Lloyd: asignación + actualización de centroides.
3. Inicialización: random vs K-Means++; rol de n_init.
4. Inercia (within-cluster sum of squares) como objetivo.
5. Selección de K: elbow method, silhouette score, gap statistic (mención).
6. MiniBatchKMeans para datasets grandes / streaming.
7. Limitaciones: clusters no convexos, densidades distintas, sensibilidad al escalado.

Definiciones y características

- K-Means — algoritmo iterativo que parte el espacio de features en K regiones de Voronoi minimizando la suma de distancias cuadradas a los centroides.
- K-Means++ — esquema de inicialización que elige centroides iniciales separados entre sí; baja la varianza entre corridas y suele converger en menos iteraciones.
- Inertia — suma de distancias cuadradas de cada punto a su centroide asignado (model.inertia_). Monótona decreciente con K; sirve para el elbow.
- Elbow method — graficar inertia_ vs K y elegir el K donde la curva "se quiebra" (la mejora marginal se aplana).
- Silhouette score — métrica en [-1, 1] que combina cohesión intra-cluster y separación inter-cluster. Más alto = mejor; útil para comparar K.
- MiniBatchKMeans — variante que actualiza centroides con mini-lotes en vez de pasar por todo X en cada iteración. ~5-10× más rápido, inercia levemente peor.
- Centroides — vectores (K, n_features) que representan el "centro" de cada cluster; en KMeans es la media de los puntos asignados.

- n_init — cantidad de corridas con distintas semillas; scikit-learn se queda con la de menor inercia. Default 10 (sklearn ≥1.4: "auto").

Dataset / recursos

- sklearn.datasets.make_blobs(n_samples=2000, centers=5, cluster_std=0.8, random_state=42) para los ejercicios de exploración.
- sklearn.datasets.load_digits() (1797 × 64) para el ejercicio de MiniBatchKMeans.
- Opcional: dataset Mall_Customers.csv o cualquier CSV tabular numérico ya escalado.

Ejercicios

1. Generá blobs con 5 centros, ajustá KMeans(n_clusters=5, n_init=10, random_state=42) y graficá los puntos coloreados por etiqueta junto con cluster_centers_.
2. Para K en range(2, 11), calculá inertia_ y silhouette_score. Graficá ambas curvas y justificá el K elegido.
3. Repetí el ajuste sin escalar un dataset donde una feature tenga escala 100× mayor que la otra. Compará con StandardScaler previo y comentá el cambio en las etiquetas.
4. Sobre load_digits(), ajustá KMeans(n_clusters=10) y MiniBatchKMeans(n_clusters=10, batch_size=256). Cronometrá ambos con %timeit y compará inercias.
5. Probá K-Means sobre make_moons(noise=0.05). Mostrá visualmente por qué falla y proponé qué algoritmo usarías en su lugar (te lo vamos a contestar en la clase 086).

Homework verificable

Entregá un script homework_085.py que:

1. Cargue make_blobs con random_state=42, 4 centros reales.
2. Recorra K = 2..8 y guarde inercia + silhouette en un DataFrame.
3. Imprima el K óptimo según silhouette y genere elbow.png + silhouette.png.

Criterio de aceptación: el script corre sin errores con python homework_085.py, imprime K óptimo = 4, y produce los dos PNG con ejes y título legibles.

Errores comunes

1. No escalar features — K-Means usa distancia euclídea; una feature en miles domina a otra en décimas. Aplicá StandardScaler salvo que tengas razón explícita para no hacerlo.
2. No fijar n_init (o dejarlo en 1) — una sola corrida puede caer en un mínimo local malo. Mantené n_init >= 10 y random_state fijo para reproducibilidad.
3. Elegir K mirando solo la inercia — la inercia siempre baja al subir K. Sin elbow visible ni silhouette, el criterio queda arbitrario.
4. Asumir clusters esféricos — K-Means no sirve para "lunas", anillos o densidades muy distintas; ahí necesitás DBSCAN o GMM.
5. Usar MiniBatchKMeans con batch_size muy chico — la varianza entre updates explota y los centroides oscilan. Mantenelo en 256–1024 salvo benchmarks específicos.

Preguntas frecuentes

1. ¿Elbow o silhouette? Silhouette es más objetivo cuando no hay codo claro; usá ambos y, si discrepan, priorizá silhouette + criterio de negocio.
2. ¿Cuántas iteraciones máximas? El default max_iter=300 alcanza casi siempre. Si convergés antes, sklearn corta solo.
3. ¿K-Means es determinístico? No: depende de la inicialización. Fijá random_state y n_init para resultados reproducibles.
4. ¿Cuándo usar MiniBatchKMeans? Cuando n_samples > 10^5 o el dataset llega en streaming. Para datasets chicos, el K-Means clásico es más preciso y suficientemente rápido.
5. ¿Sirve para detectar outliers? No directamente; los outliers distorsionan los centroides. Para outliers usá DBSCAN o IsolationForest.

Referencias

- Géron, A. Hands-On Machine Learning (3ra ed.), cap. 9 — sección "K-Means".
- scikit-learn user guide: Clustering — K-Means.
- Arthur, D. & Vassilvitskii, S. (2007). k-means++: The Advantages of Careful Seeding.
- Sculley, D. (2010). Web-Scale K-Means Clustering (paper original de MiniBatchKMeans).

Siguiente clase

Clase 096 — DBSCAN

Apéndice: notebook (primer bloque)

Primera celda ejecutable del notebook de la clase.

```
# Imports y configuración inicial
```

Archivos complementarios

- notebook.ipynb