Clustering no supervisado para detectar patrones y riesgo de abandono en clientes bancarios

Jose Morales Rendón

Facultad de Ciencias Físico-Matemáticas, UANL

Abstract

Aplicamos DBSCAN a variables transaccionales y de crédito para identificar estructuras subyacentes. Se observa un clúster dominante de comportamiento típico y un conjunto de observaciones atípicas. Llama la atención que la proporción de outliers se aproxima al churn real del conjunto de datos, sugiriendo que el abandono presenta un patrón conductual detectable sin etiquetas.

Keywords: Clustering, DBSCAN, Churn, Banca, Aprendizaje no supervisado

1. Introducción

El algoritmo DBSCAN en Python se utiliza para el agrupamiento basado en densidad y se implementa comúnmente con la biblioteca scikit-learn. DBSCAN agrupa puntos de datos cercanos, identifica valores atípicos y puede descubrir clústeres de formas arbitrarias sin necesidad de especificar el número de grupos de antemano. Los dos parámetros principales para configurar son eps (el radio de la vecindad) y min samples (el número mínimo de puntos necesarios para formar un clúster). Funcionamiento principal Identificación de núcleos: Un punto se considera un "núcleo" si tiene al menos min samples puntos dentro de su vecindad de radio eps. Expansión de clústeres: Los clústeres se expanden comenzando desde los núcleos. Si un punto es vecino de un núcleo, se agrega al clúster y se buscan sus vecinos. Este proceso continúa hasta que no se puedan agregar más puntos al clúster. Detección de ruido: Los puntos que no están dentro de la vecindad de ningún núcleo (es decir, no son directamente accesibles desde un núcleo ni alcanzan el umbral min samples) se clasifican como ruido o valores atípicos. [1, 2].

2. Datos y metodología

2.1. Datos

Describir *BankChurners* (10,127 registros, 19 variables). Variables usadas: Total_Trans_Ct, Total_Trans_Amt, Credit_Limit, Avg_Utilization_Ratio, Months_Inactive_12_mon, Contacts_Count_12_mon, Total_Relationship_Count, Months_on_book.

2.2. Preprocesamiento

Estandarización ($z = (x - \mu)/\sigma$). Selección de ε con k-distance y $minPts$=8.

2.3. Modelo

DBSCAN: definiciones de vecindad $N_\varepsilon(x)$, puntos núcleo, alcanzabilidad por densidad y ruido.

3. Resultados

3.1. Selección de ε

Describir el codo observado (p.ej., $\varepsilon \approx 1.1$).

3.2. Segmentación obtenida

Distribución de clústeres (cluster 0 ∼81%, ruido ∼15%). Perfiles: mayoría estable; nichos intensivos; dormidos; estrés crediticio.

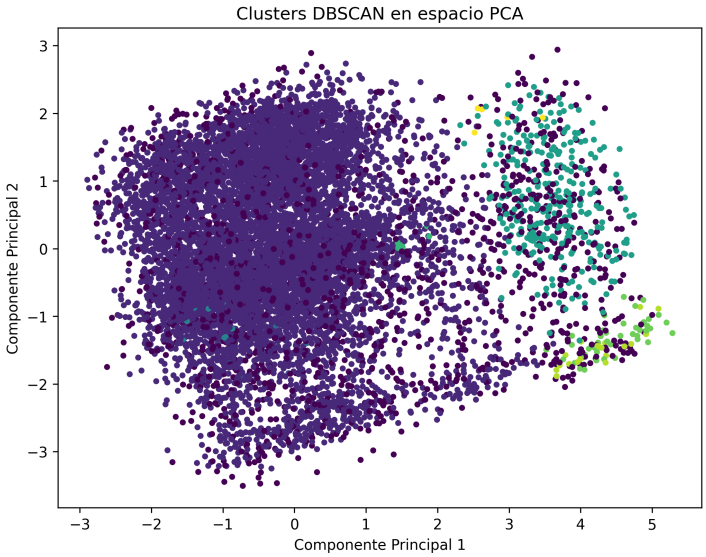

Figure 1: Clusters DBSCAN visualizados en espacio PCA (2D).

4. Discusión

Implicaciones: el churn no es aleatorio; utilidad para retención y riesgo.

5. Conclusiones

DBSCAN descubre estructuras sin fijar k. La coincidencia ruido–churn valida su uso en monitoreo temprano.

Table 1: Medias por clúster (variables seleccionadas).

Cluster	TTC	TTA	CL	AUR	MI12	CC12	TRC	MOB
0	61.76	3544.12	7612.86	0.29	2.27	2.48	3.98	35.85
-1	70.48	6425.99	12628.31	0.22	2.78	2.45	3.40	36.24
4	109.55	14596.45	9548.85	0.21	2.07	2.03	1.95	36.22
6	111.53	14465.58	33791.86	0.04	2.37	1.72	1.70	35.19
7	110.52	14571.00	33777.38	0.04	1.05	2.52	2.24	37.90

Agradecimientos

(José Alberto Benavides, por mostrarnos herramientas como latex y retar a su alumnado día a día)

References

[1] Ester, M., Kriegel, H.-P., Sander, J., & Xu, X. (1996). A density-based algorithm for discovering clusters in large spatial databases. In *KDD*.

[2] (Autor(es)). (2024). Unsupervised contrastive clustering via density-based representation learning. *Expert Systems with Applications*.