Aplicación de aprendizaje de máquina para segmentación de clientes

Gabriela Rocío Rivero
regaby@gmail.com
Agosto de 2021

Resumen

En este trabajo se analiza el desempeño de un algoritmo de cluster de aprendizaje de máquina aplicado a un escenario de segmentación de clientes. En la primera parte se presenta la metodología de segmentación a utilizar, como así también se expone el proceso de preprocesamiento aplicado al dataset. En la segunda parte se ahonda en el enfoque a cluster aplicado. Para culminar se exponen los test llevados a cabo y las conclusiones.

Introducción

En este trabajo se busca realizar una segmentación de clientes usando la base de datos real de una empresa dedicada a la venta de artículos para el hogar (bazar, blanquería, productos electrónicos, etc), cuenta con 9 locales y ante este contexto de pandemia han implementado también un sitio de e-commerce.
Se propone usar el enfoque propuesto por el modelo RFM [1, 2] (Recency Frecuency Monetary) para analizar el comportamiento de los compradores, usando la información de las transacciones de compras para segmentar y explicar las tendencias de compras. En donde las variables (features) son:

- Recency: número de días desde la última compra.
- Frecuency: número de transacciones hechas en un periodo.
- Monetary: la cantidad de dinero gastado en un periodo de tiempo.

Los clientes más valiosos son aquellos que tienen mayor frecuency/monetary y menor antigüedad (recency).

Metodología

El trabajo fue desarrollado utilizando python utilizando Visual Studio Code en un entorno Linux, con el set de librerías de scikit-learn para aplicar los distintos modelos de cluster y Pandas para el manejo de datos.
El código fuente desarrollado se puede encontrar en el repositorio de git https://github.com/regaby/tp_ml, más específicamente el archivo *kmeans.py*. También se puede encontrar una implementación en Google Colaboratory (*ML_docinf_Rivero.ipynb*). El archivo del dataset se denomina *data_ba.zip*, el mismo es un archivo csv que está comprimido.
A continuación se presentará el trabajo desarrollado organizado en:

- Presentación del dataset y el preprocesamiento del mismo
- Enfoque del método de cluster a aplicar
- Testeo de los diferentes modelos

Dataset

Para la realización del presente trabajo se cuenta con una base de datos postgres del sistema ERP de la empresa en donde en la tabla 1 se detallan las tablas y datos de las mismas

Tabla	Descripción	Rows
res_partner	Clientes	243574
account_invoice	Cabecera de facturas	423441
account_invoice_line	Líneas de factura	877379
product_product	Productos	40935

Tabla 1: Descripción de tablas postgreSQL

A efectos de poder manipular los datos más fácilmente se aplicó una *query select sql* para volcar los datos más relevantes en un archivo csv. Se aprovechó dicha query para excluir los registros con cantidades negativas y precios unitarios negativos o mayores a $250.000 ya que se determinó que dichas facturas no tenían relación con los productos a la venta.
El dataset contiene 877379 productos vendidos y 10 características que se listan en la tabla 2. Las ventas comprenden una ventana del tiempo desde el 11/08/2005 al 17/06/2021. Además el dataset no contiene valores nulos en la característica CustomerID, que es la más relevante ya que se utilizará para agrupar las ventas según los criterios RFM.

Característica	Descripción
InvoiceNo	Número de factura
StockCode	Código de producto
Description	Nombre del producto
Category	Categoría del producto
ParentCategory	Categoría padre del producto
Quantity	Cantidad de cada producto por transacción
InvoiceDate	Fecha de la transacción
UnitPrice	Precio unitario del producto
CustomerID	Número de cliente
City	Ciudad del cliente

Tabla 2: Características del dataset

La información de los tipos de datos se muestra en la tabla 3. Una vista preliminar de los datos se muestran en la tabla 4.

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 877379 entries, 0 to 877378
Data columns (total 10 columns):

#	Column	Non-Null Count	Dtype
0	InvoiceNo	878845 non-null	int64
1	StockCode	878845 non-null	int64
2	Description	878828 non-null	object
3	Category	878845 non-null	object
4	ParentCategory	877493 non-null	object
5	Quantity	878845 non-null	float64
6	InvoiceDate	878816 non-null	object
7	UnitPrice	878845 non-null	float64
8	CustomerID	878845 non-null	int64
9	City	722421 non-null	object

dtypes: float64(2), int64(3), object(5)

Tabla 3: Información del dataset

	InvoiceNo	StockCode	Description	Category	...	InvoiceDate	UnitPrice	CustomerID	City
0	7	12160	ALFOMBRA BA¥O SESAMO 0,40	Blanqueria.Com	…	2005-08-18	16.53	120	Neuquen
1	7	12160	ALFOMBRA P/BAÐO USA CANNO	Blanqueria.Com	…	2005-08-18	11.16	120	Neuquen
2	7	12160	ALFOMBRA P/BAÐO USA CANNO	Blanqueria.Com	…	2005-08-18	11.16	120	Neuquen
3	13	7020	SIMPLEX COLCHON 0.80 x 18 x 1.90	Linea Simplex	…	2005-11-17	140.50	7190	Neuquen
4	18	177	TRISET ARTEX PERCAL 180 2	Coteminas	…	2005-11-17	69.33	7190	Neuquen

[5 rows x 10 columns]

Tabla 4: vista preliminar del dataset

En la tabla 5 se exponen algunas estadísticas del dataset

	InvoiceNo	StockCode	Quantity	UnitPrice	CustomerID
count	877378.000000	877378.000000	877378.000000	877378.000000	877378.000000
mean	221452.806663	15302.544501	1.409282	1808.423981	80957.662880
std	125640.236239	9273.020636	3.692097	5187.756229	62520.448473
min	3.000000	80.000000	0.100000	0.000000	115.000000
25%	111776.000000	8876.000000	1.000000	115.350000	27666.000000
50%	225116.500000	12998.000000	1.000000	350.620000	69398.000000
75%	331915.750000	17508.000000	1.000000	1218.650000	123433.000000
max	431737.000000	41020.000000	504.000000	249697.890000	249205.000000

Tabla 5: Estadísticas del dataset

Preprocesamiento de la información

A efectos de agrupar este dataset según el modelo RFM. Se ordenarán los clientes en base a los valores de recency, frecuency y monetary. Para calcular recency se tomará un día después de la fecha de la última factura del dataset, es decir 18/06/2021. La diferencia de fechas mostrará que tan reciente fue la última transacción. Con esto podemos agrupar el dataframe por CustomerID para el preprocesamiento de datos.

En la tabla 6 se muestra una vista preliminar del preprocesamiento de datos, en donde se ha realizado la agrupación por CustomerID. El feature recency se explicó en el párrafo anterior como se obtuvo. Frecuency corresponde al total de facturas de dicho cliente y monetaryValue corresponde a la suma total de venta (precio unitario * cantidad).

CustomerID	Recency	Frequency	MonetaryValue
115	1350.0	12	9364.72
119	5784.0	3	2168.92
120	5782.0	8	3171.16
121	1212.0	5	4750.82
122	5782.0	1	693.80
142,481 rows; 3 columns			

Tabla 6: Vista preliminar del preprocesamiento de datos

Se obtuvieron 142481 agrupamientos por la actualidad, frecuencia y valor monetario de las compras. En la figura 1 se muestra la distribución de estos datos, se puede observar que los mismo están sesgados a la izquierda. En la tabla 7 se exponen estadísticas del preprocesamiento de datos.

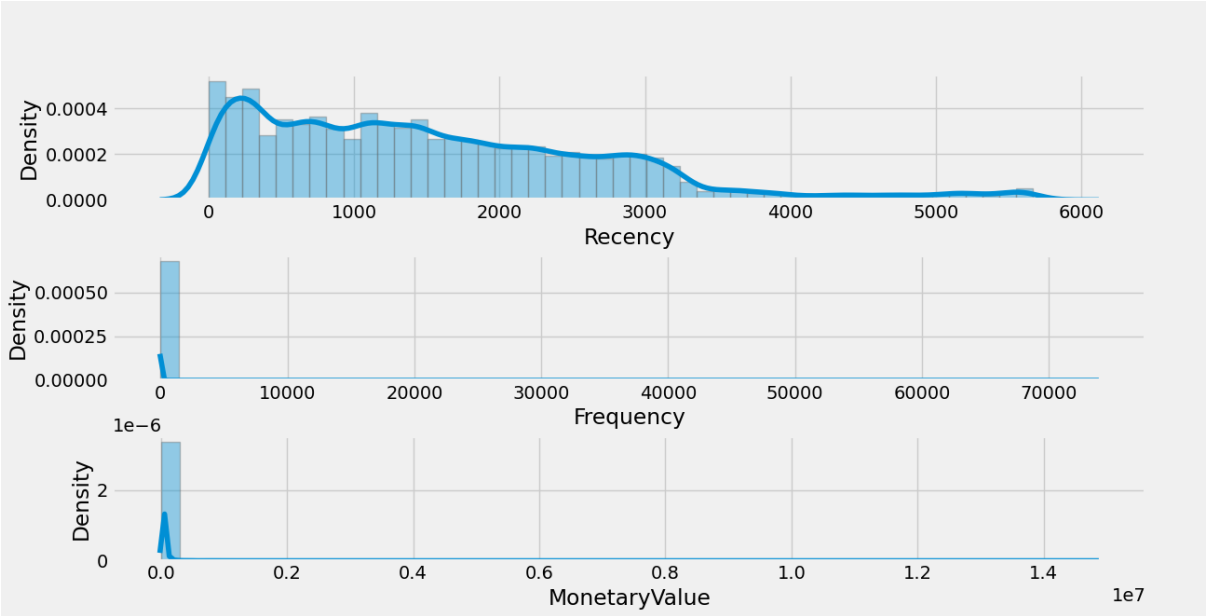


Figura 1: Distribución RFM sin normalizar

	Recency	Frequency	MonetaryValue
count	142480.000000	142481.00000	1.424810e+05
mean	1569.697172	6.15786	1.301598e+04
std	1200.910935	196.08050	5.932287e+04
min	1.000000	1.00000	0.000000e+00
25%	606.000000	2.00000	1.039600e+03
50%	1361.000000	3.00000	3.983660e+03
75%	2306.000000	7.00000	1.347408e+04
max	5784.000000	73889.00000	1.485737e+07

Tabla 7: Estadística RFM

Clustering

Para la segmentación de los clientes se aplicará el algoritmo k-means, determinando la cantidad de K con el método elbow tal como se propone en [1, 2].
K-means es un algoritmo de agrupación en clusters de aprendizaje automàtico no supervisado que utiliza múltiples iteraciones para segmentar los puntos de datos sin etiquetar en diferentes clústeres "k" de manera que cada punto de datos pertenezca a un solo grupo que tenga propiedades similares. Estos puntos son más similares entre ellos que a los puntos que pertenecen a otros grupos. El agrupamiento basado en la distancia agrupa los puntos en cierto número de grupos, de modo que las distancias dentro del grupo deben ser pequeñas, mientras que las distancias entre los grupos deben ser grandes.
K-means utiliza la distancia euclidiana como métrica de distancia para calcular la distancia entre cada punto y el centroide.
K-means da el mejor resultado en las siguientes condiciones:

- La distribución de datos no está sesgada
- Los datos están estandarizados (es decir, media de 0 y desviación estándar de 1).

Como se puede ver en la figura 1 la distribución de los datos está sesgada. Por lo tanto se procedió con su transformación a escala logarítmica y luego su respectiva normalización (Figura 2).

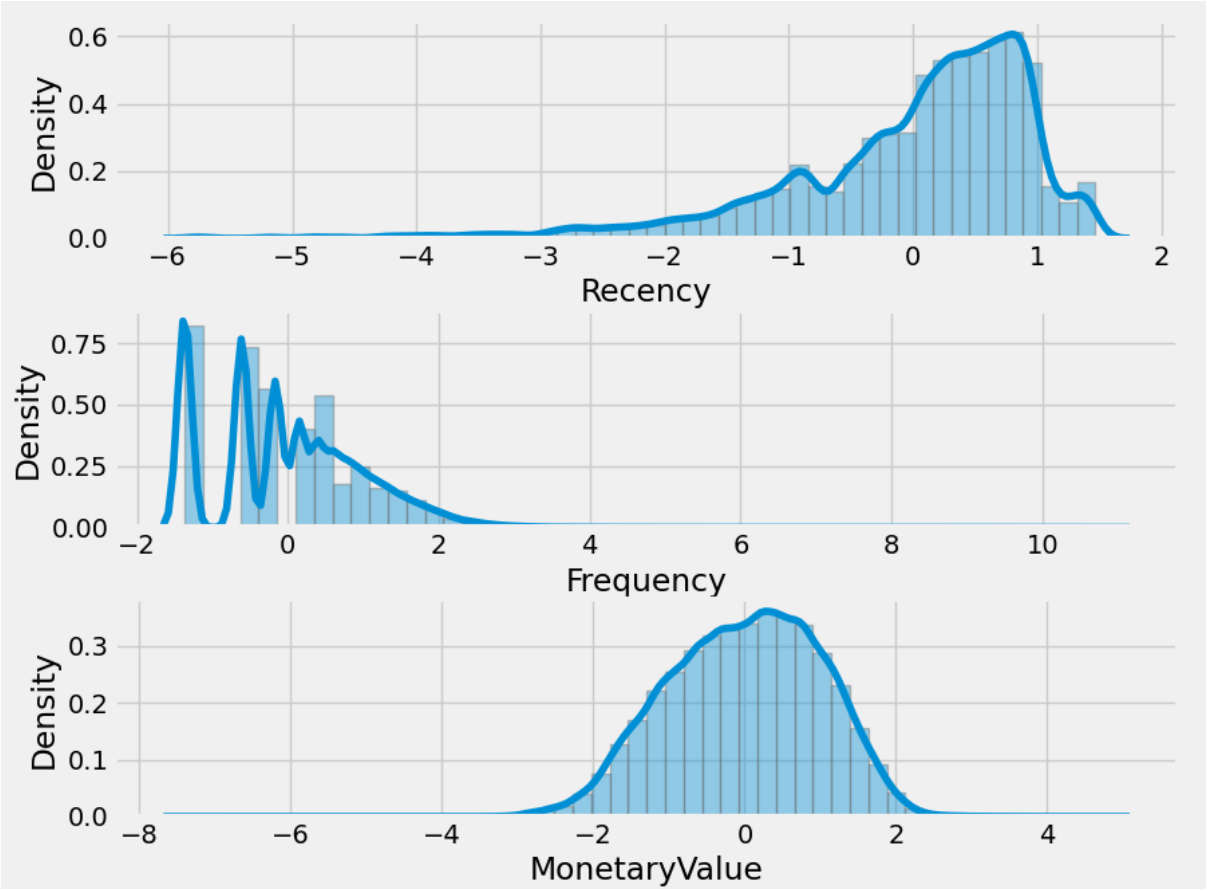


Figura 2: Distribución RFM normalizada

Se probará el algoritmo con los datos sin normalizar y normalizados, aplicando diferentes métricas de calidad para comparar el desempeño de ambos dataset.

Determinación de K

El siguiente paso es seleccionar el número correcto de clústeres k. Como no se sabe de antemano este número, para obtenerlo se utilizó el método elbow (codo en inglés).

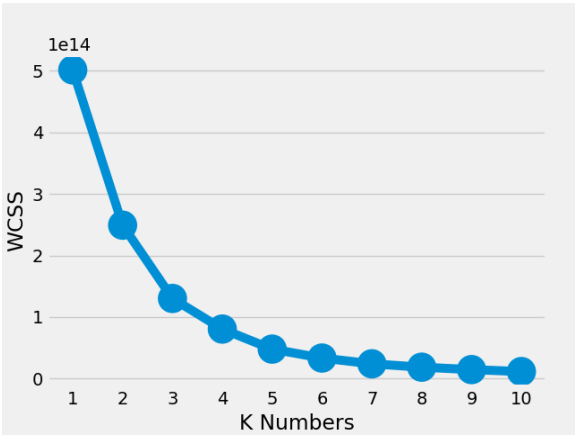


Figura 3.a:
Dataset sin normalizar

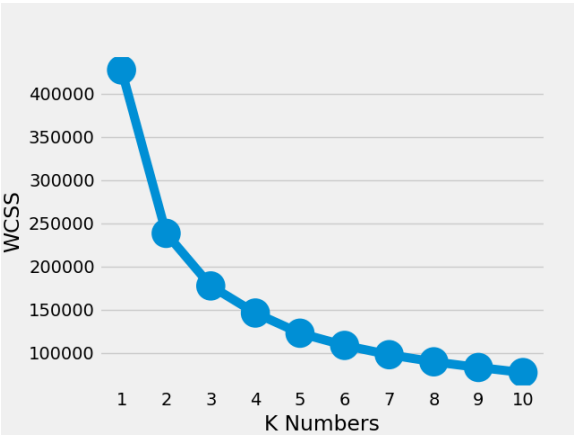


Figura 3.b:
Dataset normalizado

Se construyó un modelo iterativo de clústeres del 1 al 10, y luego se obtuvo el valor WSS (*Within clusters sum of squares*) para cada modelo. El mismo se aplicó al dataset sin normalizar y normalizado (Figura 3.a y 3.b). Como se puede ver en dichas figuras, a medida que aumenta el número de clústeres el valor de WCSS disminuye. Esto es porque a mayor cantidad de grupos, el tamaño de cada uno disminuye como así también la suma de las distancias. Además se observa que la amplitud de valores para WCSS es mayor en la figura 3.a dado que en dicho dataset los datos están dispersos (sin normalizar).
El número óptimo de K es a partir de donde WCSS empieza a disminuir más paulatinamente, es decir en donde se forma un “codo”. En ambas figuras se determina que esto sucede cuando k=3.

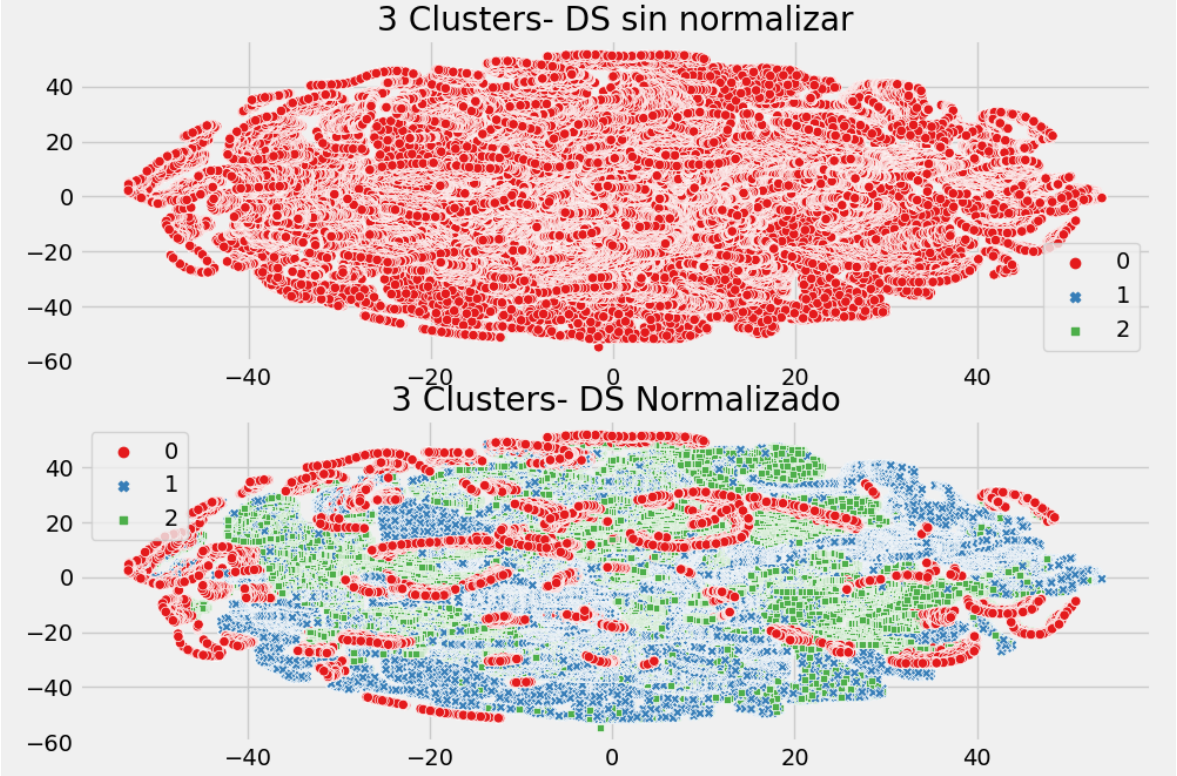
Métricas

Para evaluar los distintos modelos se utilizaron distintas métricas de calidad de cluster:

- Homogeneity Score (homo)
- Completeness Score (compl)
- V measure (v-meas)
- Adjusted Rand index (ARI)
- Adjusted mutual information (AMI)
- Silhouette coefficient (silhouette)

Testeo de los Dataset

Como se mencionó anteriormente se realizó una comparativa de los dos dataset con k=3 e inicializando con el parámetro k-means++. En la figura 4 se muestran los resultados.



init	time	inertia	homo	compl	v-meas	ARI	AMI	silhouette
DS s/norm.	1.883s	125795	0.109	0.000	0.000	-0.000	0.000	0.030
DS Norma.	4.296s	177581	0.999	0.999	0.999	1.000	0.999	0.341

Figura 4: K-means con k=3 con dataset sin normalizar y normalizado

La técnica de visualización que se usó se denomina t-Distributed Stochastic Neighbor Embedding, es un método estadístico para visualizar datos de alta dimensión dando a cada punto de datos una ubicación en un mapa bidimensional.

En la tabla 8 se muestra la cantidad de resultados para cada agrupación.

K	DS S/normalizar	DS normalizado
0	142462	58978
1	18	54813
2	1	28690

Tabla 8: Valores obtenidos

Posteriormente se probaron diferentes parámetros de inicialización sobre el dataset normalizado con k=3: k-means++, random y PCA. La comparativa de métricas se muestra en la tabla 9.

init	time	inertia	homo	compl	v-meas	ARI	AMI	silhouette
K=3 DSN km++	4.289s	125794	0.109	0.000	0.000	-0.002	0.000	0.051
K=3 DSN random	1.346s	306534	0.226	0.044	0.073	0.063	0.073	-0.247
K=3 DSN PCA	0.315s	313270	0.570	0.123	0.202	-1.160	0.202	0.075

Conclusiones

En la primera prueba realizada se observa que usando un dataset se observan los mejores resultados en cuanto a la segmentación, ya que cuando no se realizó la normalización de datos casi la totalidad de los puntos se agruparon en el cluster 0.
En cuanto a los parámetros de inicialización, el insume mayor tiempo y capacidad de cómputo fue k-means++ sobre random y PCA. Este último tuvo los mayores valores en la métricas.

Referencias

[1] M Dachyar, F M Esperanca and R Nurcahyo. *Loyalty Improvement of Indonesian Local Brand Fashion Customer Based on Customer Lifetime Value (CLV) Segmentation*, 2019

[2] Sari Hartini, Windu Gata, Sigit Kurniawan, Hendra Setiawan and Kadinar Novel. *Cosmetics Customer Segmentation and Profile in Indonesia Using Clustering and Classification Algorithm*. 2020