

Parte A: K-Means – Segmentación de clientes

Preprocesamiento de datos:

Datos originales

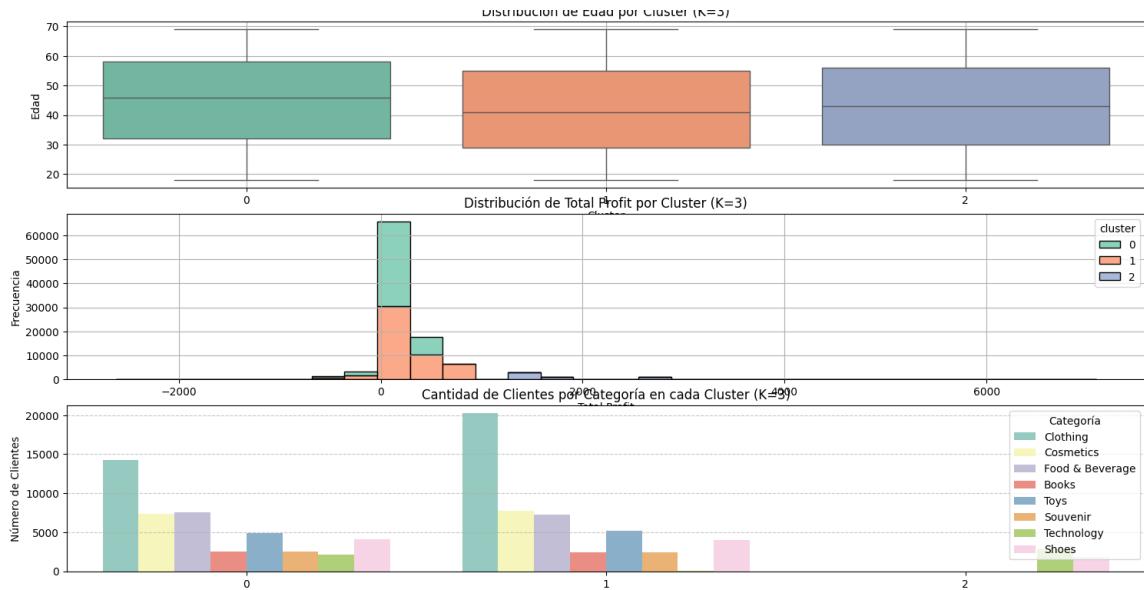
1	customer_id	gender	age	category	quantity	selling_price_per_unit	total_profit	payment_method	region	state	shopping_mall
2	C241288	Female	28	Clothing	5	15004	3751	CreditCard	South	Kentucky	Kanyon
3	C111565	Male	21	Shoes	3	180051	540153	Debit Card	South	Kentucky	Viaport Outlet
4	C266599	Male	20	Clothing	1	30008	502	Cash	West	California	Metrocity
5	C988172	Female	66	Shoes	5	300085	500425	Credit Card	South	Florida	MetropolAVM
6	C189076	Female	53	Books	4	606	606	Cash	South	Florida	Kanyon
7	C657758	Female	28	Clothing	5	15004	1251	Credit Card	West	Oregon	Viaport Outlet
8	C151197	Female	49	Cosmetics	1	4066	5198	Cash	West	California	Istinye Park
9	C176086	Female	32	Clothing	2	60016	30008	Credit Card	West	California	Mall of Istanbul
10	C159642	Male	69	Clothing	3	90024	67518	Credit Card	West	California	Metrocity
11	C283361	Female	60	Clothing	2	60016	30008	Credit Card	West	California	Kanyon
12	C240286	Female	36	Food & Beverage	2	1046	4276	Cash	West	California	Metrocity
13	C191708	Female	29	Books	8	1515	-36	Credit Card	West	Idaho	Zorlu Center
14	C225330	Female	67	Toys	4	14336	2688	Debit Card	South	North Carolina	Metrocity
15	C312861	Male	25	Clothing	2	60016	10008	Cash	West	Washington	Istinye Park
16	C555402	Female	67	Clothing	2	60016	3888	Credit Card	Central	Texas	Kanyon
17	C362288	Male	24	Shoes	5	300085	1500425	Credit Card	Central	Texas	Viaport Outlet
18	C300786	Male	65	Books	2	303	618	Debit Card	Central	Iowa	Metrocity

Datos procesados, se realiza un distincion con categoricos y numericos. Para los numericos se los escala con la instancia StandardScaler y los categoricos con OneHotEncoder y el argumento drop='first' que se encarga de eliminar una columna para evitar la multicolinealidad

```
Iniciando preprocesamiento...
Preprocesamiento completado. Primeras filas del resultado:
   age  quantity  total_profit  gender_Male  category_Clothing  ...
0 -1.029160  1.410072    0.339583      0.0          1.0  ...
1 -1.496139 -0.003126    0.727874      1.0         0.0  ...
2 -1.562850 -1.416324   -0.531039      1.0          1.0  ...
3  1.505867  1.410072    0.634413      0.0         0.0  ...
4  0.638621  0.703473   -0.400285      0.0         0.0  ...
```

Clusters por K-means

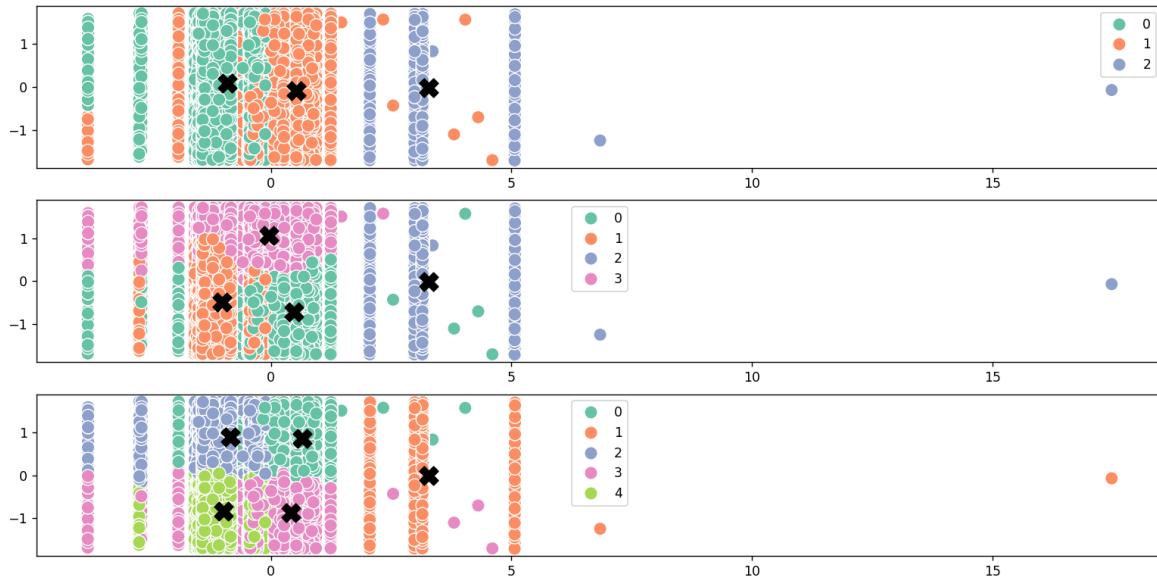
Se realizan pruebas para 3, 4 y 5 clusters y se grafican los cluster por edades, el beneficio de la compra y el articulo comprado.



Andres Felipe Marcillo



Además, se grafican los clusters en todas las categorías y se simplifica con la instancia PCA para reducir a 2 dimensiones la clasificación. Se puede evidenciar una buena agrupación a pesar de que no es perfecta y a medida que se aumentan los clusters, no afecta la dispersión de los datos, simplemente se crean mas conjuntos de datos. Las X representan el centroide de cada cluster el cual gráficamente es adecuado para identificar.



Características

A continuación se ilustran las estadísticas por medio de la instancia describe y se logra obtener un bajo desempeño de diferenciación en las clasificaciones tenidas en cuenta de: edad, genero y cantidad, pero en la clasificación de categoría y total_profit si se ve una gran diferencia de los agrupamientos de los clusters.

Interpretación de clusters (K=3)										
cluster	age		gender		category	quantity		total_profit		
	mean	median	mean	<lambda>		<lambda>	mean	sum	mean	sum
0	44.860104	46.0	Female		Clothing	1.696685	77038	69.641096	3.162054e+06	
1	42.117545	41.0	Female		Clothing	4.074630	201193	235.815927	1.164388e+07	
2	43.340535	43.0	Female		Technology	4.402139	20580	1742.014231	8.143917e+06	

Cantidad de clientes por cluster (K=3):										
cluster	count									
0	45405									
1	49377									
2	4675									
Name: count, dtype: int64										

		Estadísticas detalladas por cluster (describe) (K=3):			
cluster		0	1	2	
age	count	45405.000000	49377.000000	4675.000000	
	mean	44.860104	42.117545	43.340535	
	std	14.933020	14.933197	14.875362	
	min	18.000000	18.000000	18.000000	
	25%	32.000000	29.000000	30.000000	
	50%	46.000000	41.000000	43.000000	
	75%	58.000000	55.000000	56.000000	
	max	69.000000	69.000000	69.000000	
quantity	count	45405.000000	49377.000000	4675.000000	
	mean	1.696685	4.074630	4.402139	
	std	0.696511	0.788440	0.819893	
	min	1.000000	3.000000	3.000000	
	25%	1.000000	3.000000	4.000000	
	50%	2.000000	4.000000	5.000000	
	75%	2.000000	5.000000	5.000000	
	max	5.000000	12.000000	15.000000	
total_profit	count	45405.000000	49377.000000	4675.000000	
	mean	69.641096	235.815927	1742.014231	
	std	197.982021	261.526845	452.454354	
	min	-2625.000000	-2625.000000	1050.280000	
	25%	10.752000	73.312500	1500.425000	
	50%	48.792000	203.300000	1500.425000	
	75%	105.000000	375.100000	1680.000000	
	max	420.000000	750.200000	7087.500000	

Interpretación de clusters (K=4)											
cluster	age			gender		category	quantity		total_profit		
	mean	median	<lambda>	<lambda>	<lambda>		mean	sum	mean	sum	
0	32.639985	33.0	Female	Clothing	4.056406	Clothing	4.056406	128008	210.873907	6654547.868	
1	35.961241	36.0	Female	Clothing	1.464580	Clothing	1.464580	41452	74.135885	2098267.954	
2	43.340535	43.0	Female	Technology	4.402139	Technology	4.402139	20580	1742.014231	8143916.530	
3	59.237157	60.0	Female	Clothing	3.114684	Clothing	3.114684	108771	173.332603	6053121.164	

🕒 Cantidad de clientes por cluster (K=4):

```
cluster
0    31557
1    28303
2     4675
3    34922
Name: count, dtype: int64
```

Estadísticas detalladas por cluster (describe) (K=4):					
cluster		0	1	2	3
age	count	31557.000000	28303.000000	4675.000000	34922.000000
	mean	32.639985	35.961241	43.340535	59.237157
	std	8.892751	10.946592	14.875362	6.439689
	min	18.000000	18.000000	18.000000	44.000000
	25%	25.000000	27.000000	30.000000	54.000000
	50%	33.000000	36.000000	43.000000	60.000000
	75%	40.000000	45.000000	56.000000	65.000000
	max	67.000000	58.000000	69.000000	69.000000
quantity	count	31557.000000	28303.000000	4675.000000	34922.000000
	mean	4.056406	1.464580	4.402139	3.114684
	std	0.815919	0.526395	0.819893	1.262011
	min	3.000000	1.000000	3.000000	1.000000
	25%	3.000000	1.000000	4.000000	2.000000
	50%	4.000000	1.000000	5.000000	3.000000
	75%	5.000000	2.000000	5.000000	4.000000
	max	12.000000	3.000000	15.000000	8.000000
total_profit	count	31557.000000	28303.000000	4675.000000	34922.000000
	mean	210.873907	74.135885	1742.014231	173.332603
	std	273.737501	161.805759	452.454354	262.664010
	min	-2625.000000	-1417.500000	1050.280000	-2625.000000
	25%	46.920000	6.276000	1500.425000	31.671000
	50%	179.200000	48.792000	1500.425000	105.000000
	75%	375.100000	105.000000	1680.000000	300.080000
	max	750.200000	420.000000	7087.500000	675.180000

Interpretación de clusters (K=5)										
cluster	age			gender		category	quantity	total_profit		
	mean	median	<lambda>	mean	<lambda>			sum	mean	sum
0	56.294914	56.0	Female	Clothing	4.201946	93695	266.248981	5.936820e+06		
1	43.337826	43.0	Female	Technology	4.401583	20573	1742.162227	8.142866e+06		
2	56.674501	57.0	Female	Clothing	1.851046	45756	63.149151	1.560984e+06		
3	30.383746	30.0	Female	Clothing	3.972183	107527	206.983333	5.603039e+06		
4	30.821318	31.0	Female	Clothing	1.510437	31260	82.438384	1.706145e+06		

👤 Cantidad de clientes por cluster (K=5):

cluster

0 22298

1 4674

2 24719

3 27070

4 20696

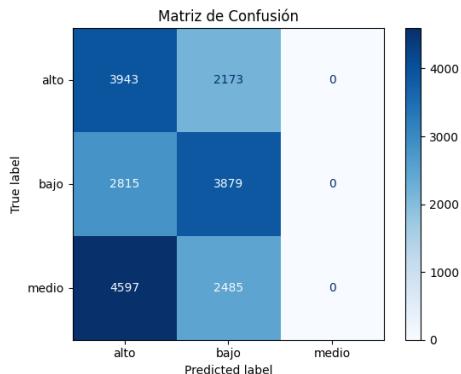
Name: count, dtype: int64

📈 Estadísticas detalladas por cluster (describe) (K=5):						
cluster		0	1	2	3	4
age	count	22298.000000	4674.000000	24719.000000	27070.000000	20696.000000
	mean	56.294914	43.337826	56.674501	30.383746	30.821318
	std	7.648594	14.875801	7.420232	7.439327	7.697146
	min	42.000000	18.000000	41.000000	18.000000	18.000000
	25%	50.000000	30.000000	50.000000	24.000000	24.000000
	50%	56.000000	43.000000	57.000000	30.000000	31.000000
	75%	63.000000	56.000000	63.000000	37.000000	37.000000
	max	69.000000	69.000000	69.000000	47.000000	45.000000
	total_profit	22298.000000	4674.000000	24719.000000	27070.000000	20696.000000
quantity	count	22298.000000	4674.000000	24719.000000	27070.000000	20696.000000
	mean	4.201946	4.401583	1.851046	3.972183	1.510437
	std	0.737262	0.819100	0.771214	0.813188	0.530024
	min	3.000000	3.000000	1.000000	3.000000	1.000000
	25%	4.000000	4.000000	1.000000	3.000000	1.000000
	50%	4.000000	5.000000	2.000000	4.000000	1.000000
	75%	5.000000	5.000000	2.000000	5.000000	2.000000
	max	12.000000	15.000000	5.000000	12.000000	3.000000
	total_profit	22298.000000	4674.000000	24719.000000	27070.000000	20696.000000
total_profit	count	22298.000000	4674.000000	24719.000000	27070.000000	20696.000000
	mean	266.248981	1742.162227	63.149151	206.983333	82.438384
	std	255.152435	452.389581	202.585623	278.392342	167.462646
	min	-1500.425000	1417.500000	-2625.000000	-2625.000000	-1417.500000
	25%	94.687500	1500.425000	10.752000	40.905000	6.276000
	50%	240.064000	1500.425000	48.792000	130.112000	48.792000
	75%	375.100000	1680.000000	91.485000	375.100000	150.042500
	max	1050.280000	7087.500000	420.000000	750.200000	420.000000

Clasificación de clientes según gasto >> Maquina de soporte vectorial – SVM

Se puede observar que para la categoría de medio gasto, no se logra obtener un buen desempeño, no se logra identificar este grupo de personas, cosa diferente con las personas que gastan alto y bajo, que a pesar de no ser perfecta si hay una clasificación que se logra obtener y entre estas dos la mejor es la de bajo gasto con un aproximado de 0.52 como f1_score lo cual nos indica que el modelo tiene una precisión y recall regulares y por tal motivo se debe mejorar. Se hicieron dos pruebas en las cuales la primera es con 30% de los datos puestos a prueba y un 70% para el entrenamiento del modelo mientras que en la siguiente se usa un 20% y 80% respectivamente.

	precision	recall	f1-score	support		precision	recall	f1-score	support
alto	0.30	0.40	0.35	9186	alto	0.35	0.64	0.45	6116
bajo	0.41	0.72	0.52	10054	bajo	0.45	0.58	0.51	6694
medio	0.00	0.00	0.00	10598	medio	0.00	0.00	0.00	7882
accuracy					accuracy				19892
macro avg	0.24	0.37	0.29	29838	macro avg	0.27	0.41	0.32	19892
weighted avg	0.23	0.37	0.28	29838	weighted avg	0.26	0.39	0.31	19892
Distribución de clases en el conjunto de entrenamiento:									
gasto_nivel									
medio	24749				medio	28265			
bajo	23378				bajo	26738			
alto	21492				alto	24562			
Name: count, dtype: int64					Name: count, dtype: int64				
					[[3943 2173 0]]				
					[2815 3879 0]				
					[4597 2485 0]]]				



De acuerdo con la matriz de confusión:

Clase "medio" nunca fue predicha, esto ya que la columna "medio" tiene todo ceros, lo que significa que tu modelo nunca predijo la clase "medio".

Lo anterior puede ser por: Clase desequilibrada, el modelo no aprendió bien sus patrones y hay posibilidad de que haya habido algún error en el etiquetado o preprocesamiento.

Hay bastante confusión entre "alto" y "bajo" 2173 instancias reales de clase alto fueron clasificadas como bajo.

2815 de clase bajo fueron clasificadas como alto.

Red Neuronal con la biblioteca KERAS:

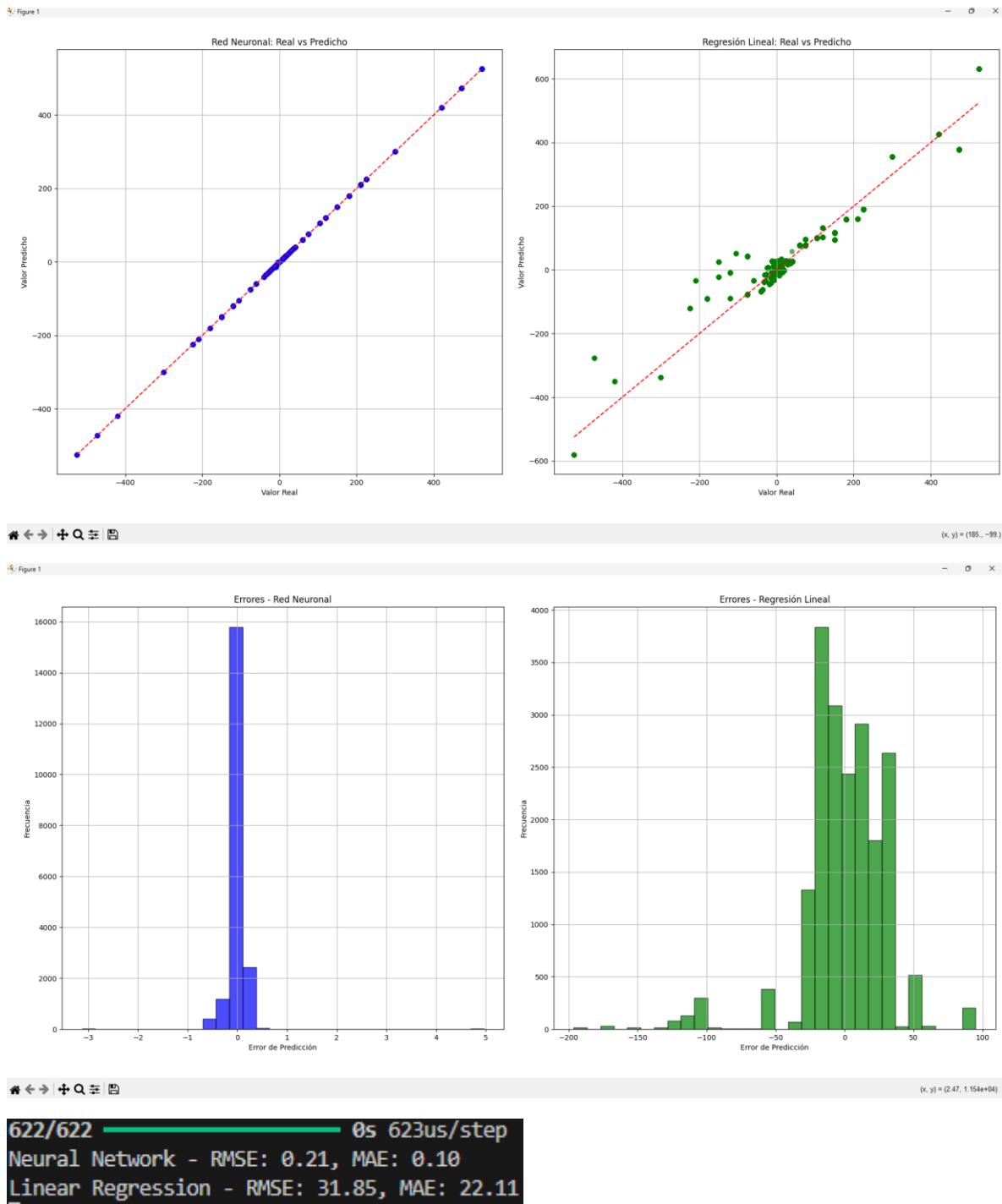
Se crea una red neuronal usando 20% de los datos de entrenamiento como validación y el 80% para entrenamiento. 100 épocas en un inicio y luego se determinó que se la red trabaja bien con 69 épocas, esto por medio de la supervisión de la perdida en validación, si no mejora en 10 épocas cada vez que aumenta, se detiene el entrenamiento.

Se realizan 5 pruebas de entrenamiento de la red y el resultado muy bueno para la red neuronal, mucho mejor que para la regresión lineal. La quinta vez se determinó que con 69 épocas era suficiente el entrenamiento.

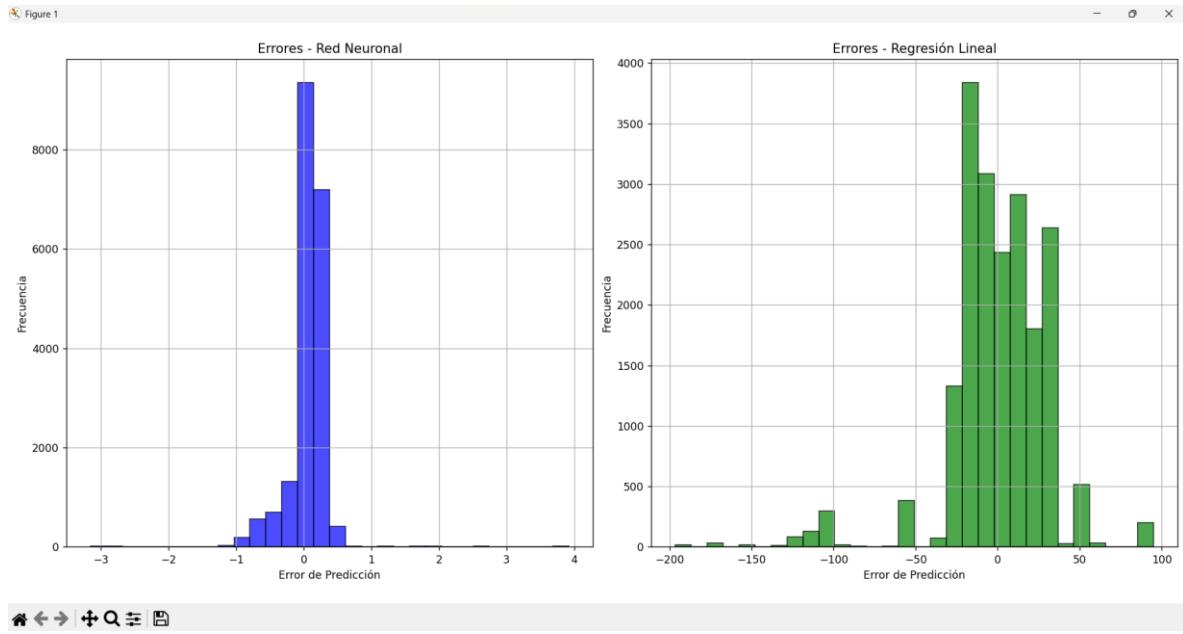
También cabe aclarar que se trabaja con un modelo de tipo secuencial, en la capa de entrada se tiene una neurona, dos capas ocultas en seguida, la primera de 16 neuronas y la siguiente de 8 antes de la capa de salida con una neurona de la salida como predicción del gasto predicho con el algoritmo de Adam y un agrupamiento de 16 datos.

A continuación, se obtiene los valores predichos de la red neuronal a comparación con la regresión y los errores RMSE y MAE para cada uno, donde siempre se tiene un RMSE mas alto siempre debido a que es cuadrático y esto hace que penalice mas las diferencias altas que existan en la validación predicho respecto al real.

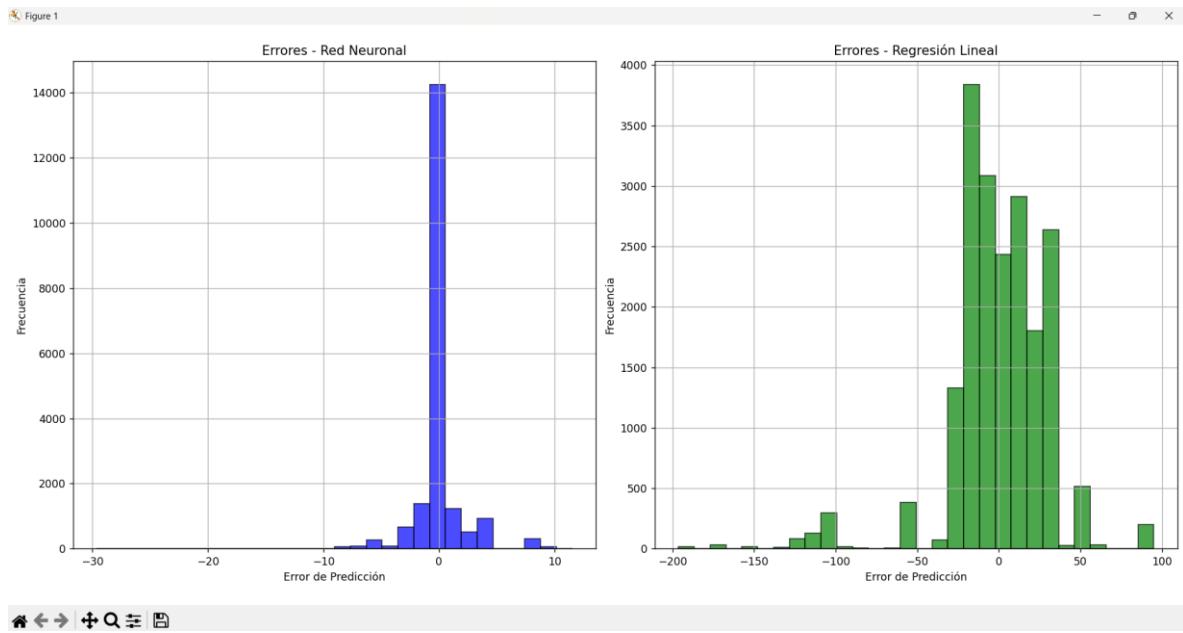
Andres Felipe Marcillo



Andres Felipe Marcillo

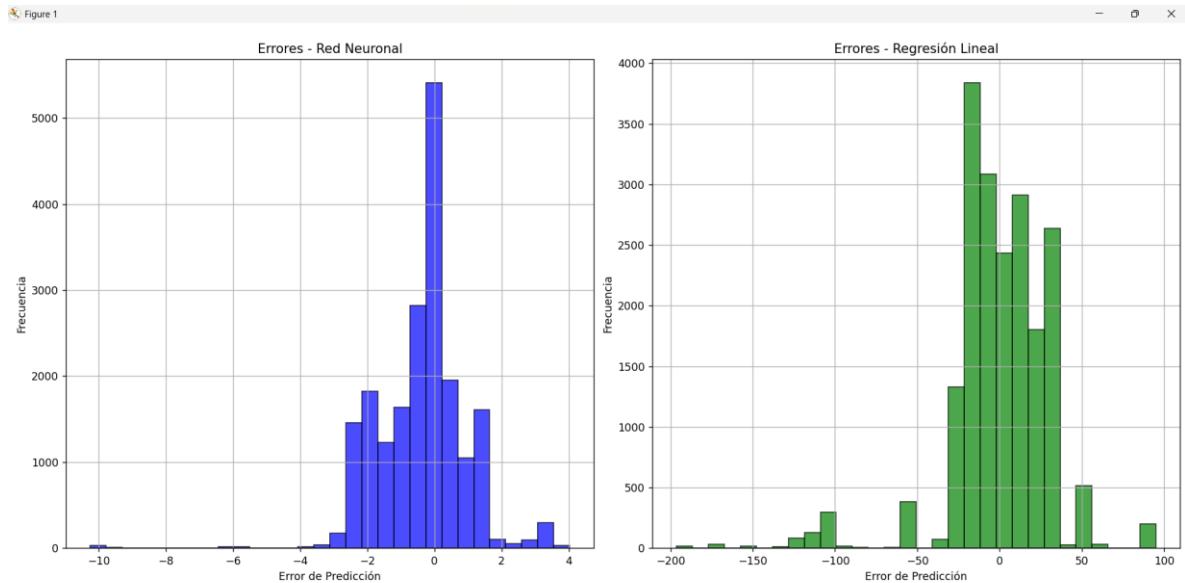


622/622 ━━━━━━ 1s 937us/step
Neural Network - RMSE: 0.33, MAE: 0.20
Linear Regression - RMSE: 31.85, MAE: 22.11



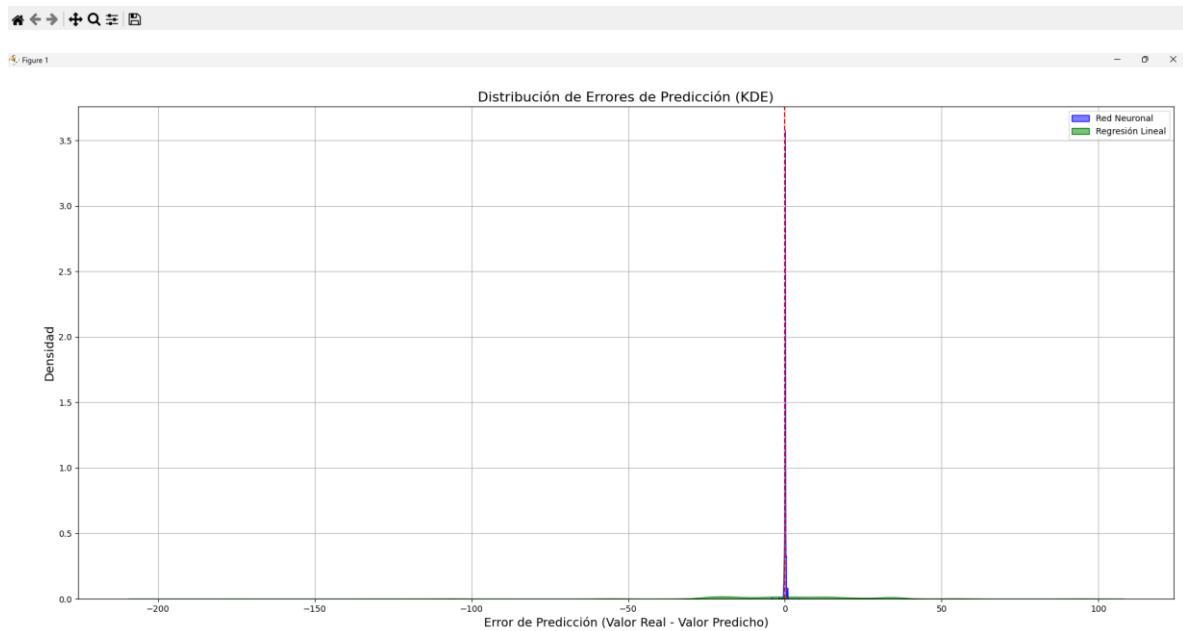
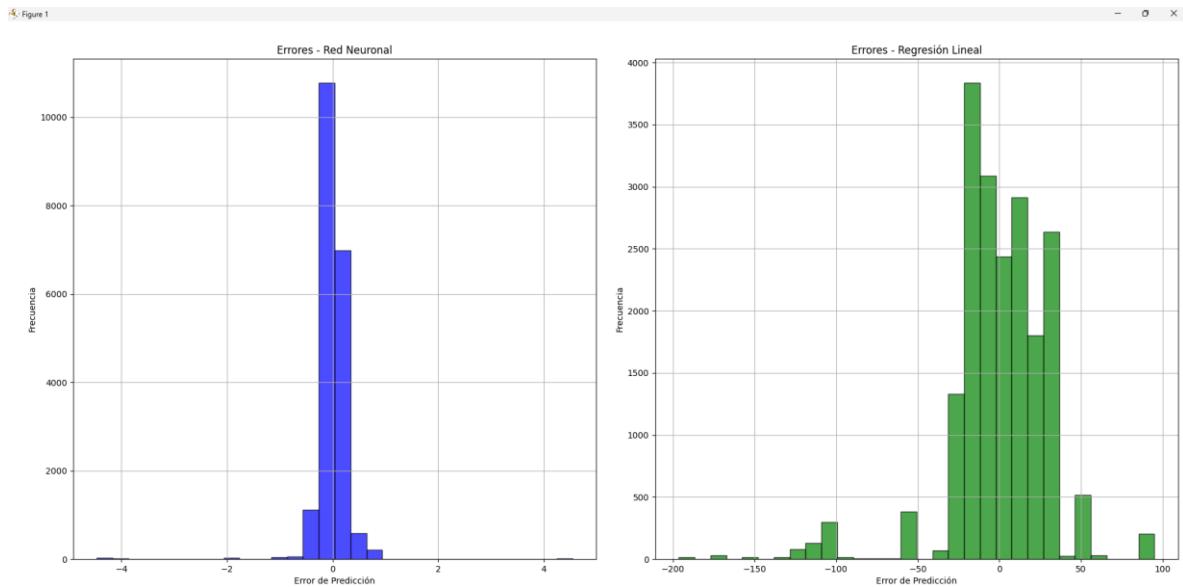
622/622 ━━━━━━ 0s 631us/step
Neural Network - RMSE: 2.16, MAE: 1.00
Linear Regression - RMSE: 31.85, MAE: 22.11

Andres Felipe Marcillo



622/622 0s 584us/step
Neural Network - RMSE: 1.36, MAE: 0.97
Linear Regression - RMSE: 31.85, MAE: 22.11

Andres Felipe Marcillo



```
Entrenamiento detenido después de 59 épocas.  
622/622 ━━━━━━ 1s 768us/step  
Neural Network - RMSE: 0.31, MAE: 0.14  
Linear Regression - RMSE: 31.85, MAE: 22.11
```

1. ¿Qué técnica fue más efectiva para cada problema?

SVM – Clasificación de clientes según nivel de gasto:

Accuracy: 39%

F1-score para clase "medio": 0.00

El modelo no logra clasificar la clase "medio" en absoluto, hay un desbalance en las clases que el modelo no está manejando bien. SVM tiende a verse afectado negativamente por lo anterior.

Red Neuronal y regresión lineal – Predicción de gasto:

Red Neuronal:

- **RMSE: 0.31**
- **MAE: 0.14**

Regresión Lineal:

- **RMSE: 31.85**
- **MAE: 22.11**

La red neuronal es mucho más precisa, con errores de predicción muy pequeños comparados con la regresión lineal. Se observa también en los histogramas: los errores de la red están centrados en 0 y son muy estrechos.

Conclusión:

SVM no fue efectiva y falló particularmente con la clase "medio". La red neuronal fue muy superior a la regresión lineal y esta si fue efectiva.

2. ¿Qué ventajas/desventajas tiene cada una?

Técnica	Ventajas	Desventajas
SVM	Funciona bien con pocos datos.	Mala con datos desbalanceados. Escala mal con datasets grandes. Difícil interpretación y alto consumo computacional
Red Neuronal	Aprende relaciones complejas no lineales. Muy precisa en regresión como se evidencio en este caso	Requiere más datos y tiempo de entrenamiento. Alto consumo computacional

Reg. Lineal	Fácil de interpretar. Rápida.	Mal desempeño si hay no linealidades.
--------------------	----------------------------------	---------------------------------------

3. ¿Cómo se pueden aplicar estos modelos en la vida real?

SVM (Clasificación de nivel de gasto):

Siempre y cuando se mejore el modelo usando técnicas para tratar el desbalance de clases, (clasificadores como random, forest o XGBoost) se puede clasificar clientes en segmentos de alto, medio o bajo para marketing personalizado y promocionar a los clientes prioritarios.

Red Neuronal (Predicción de gasto):

Predecir cuanto gasta un cliente en función de sus características o comportamiento.

Planear stock, promociones o ingresos futuros