Limpieza inicial

Ventas

Ventas tiene 4 columnas de la cual a ninguna le hace falta datos, para poder agregar más a estos datos usamos el ProductCategory de productos para poder agregarlos junto con nuestras ventas.

	CustomerId	material	calmonth	uni_box	productCategory
0	499920078	9151	201909	0.4364	CATEGORIAS EN EXPANSION
1	499920078	2287	201909	3.1701	REFRESCOS
2	499920078	4526	201909	0.2818	LACTEOS
3	499920078	14050	201909	0.2642	LACTEOS
4	499920078	1333	201909	2.1134	CATEGORIAS EN EXPANSION

Después usamos un pivot para reorganizar el DataFrame ventas de modo que las categorías de producto se conviertan en columnas y las cantidades vendidas en unidades por caja (uni_box) se sumen para cada cliente.

Al final esto lo pasamos a un excel y nos da el siguiente resultado:

A	B	C	D	E	F	G	H
	CustomerId	AGUA	MINERAL CON S	EBIDAS CON ALCOH	GORIAS EN EXPANS	LACTEOS	REFRESCOS
0	5E+08	782.7386	4.438	3.751	367.3702	53.5409	4952.596763
1	5E+08	739.0422	12.5744	0.3751	659.4166	13.2088	10518.70494
2	5E+08	1465.562	20.8164	11.253	724.9306	67.0312	21934.32205
3	5E+08	2051.024	29.3755	0	2727.7082	180.1915	18018.79165
4	5E+08	2403.561	6.8684	0.7502	1451.125121	86.0534	12437.76925
5	5E+08	297.0157	0	0	524.613925	0.4227	8886.44652
6	5E+08	200.7292	0	0	171.8432666	20.7827	6390.737394
7	5E+08	419.341	0.634	4.5012	52.77405	33.2863	6211.715013
8	5E+08	206.7525	2.2191	0	419.8455	1.9021	10172.65255
9	5E+08	415.224	10.0384	0	598.7994	28.5317	8965.736453
10	5E+08	2034.775	15.216	13.1285	1518.563079	73.6009	17580.21288
11	5E+08	678.1715	15.4275	4.8763	485.4312	28.7781	7922.8335
12	5E+08	174.951	1.268	0	20.5750422	1.5499	3567.062328
13	5E+08	573.0874	8.2422	0	977.4064916	23.2822622	14012.52706
14	5E+08	1506.892	13.6311	0.7502	1347.7332	95.0187	12521.8026
15	5E+08	2158.489	34.1304	10.1277	1963.5729	263.2279	26180.48619
16	5E+08	3300.255	22.0844	0	2351.117996	180.540175	85991.17382
17	5E+08	618.7049	6.0231	0	318.1638274	20.6766	5022.00183
18	5E+08	593.8157	0.5284	0	1926.508937	68.9875	7792.632397
19	5E+08	215.2011	2.536	0	293.8220405	8.982	3863.644538
20	5E+08	182.8834	1.902	0	62.272029	5.0719811	1986.709489
21	5E+08	989.7031	8.8762	4.8763	497.0936	42.3915	15811.3681
22	5E+08	1539.734	58.2228	5.2514	1799.2964	138.4303	18185.46636

Productos
Los productos tienen 22 columnas algunas de estas columnas son descriptivas sobre el tipo de producto, envase, tamaño, etc. En nuestros datos de productos no se tienen datos faltantes más que en algunos casos donde faltan 2 datos. Los datos que nos interesan para los productos son el material, cantidad, tipo, contenedor, tamaño, categoría, entre otros.

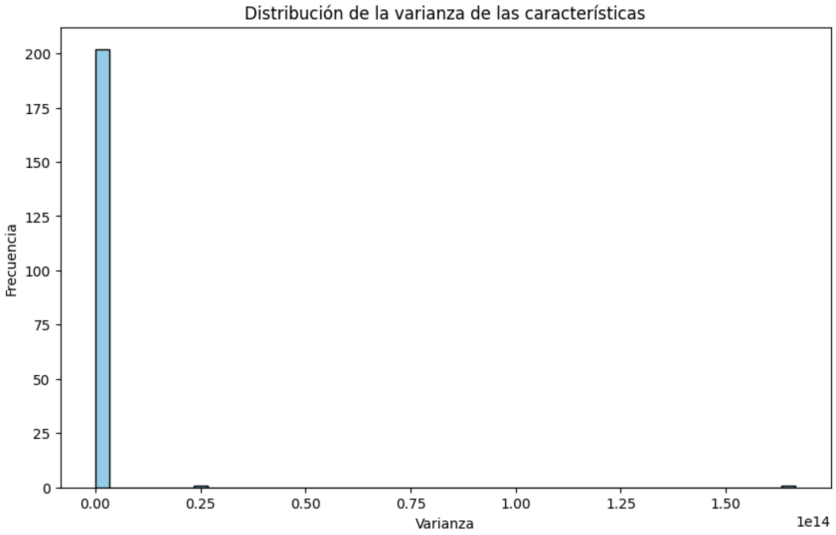
Uno de nuestros datos faltantes en MLSize lo cual es una columna que queremos utilizar, se crea una función donde se puede conseguir el MLSize desde la columna de presentation, una vez ya corremos esto y los cambios se guardan, ya tenemos nuestra tabla con los datos que se utilizaran.

792	14426	COSTA - MOCHA ITAL 500 GRS NR BOL 12	12	COSTA 500 G NO RETORNABLE	NOT_SUGGESTED	COSTA	BOLSA 500 GR.	0	NO RETORNABLE	500 GR.	...
792	14426	COSTA - MOCHA ITAL 500 GRS NR BOL 12	12	COSTA 500 G NO RETORNABLE	NOT_SUGGESTED	COSTA	BOLSA 500 GR.	500	NO RETORNABLE	500 GR.	...

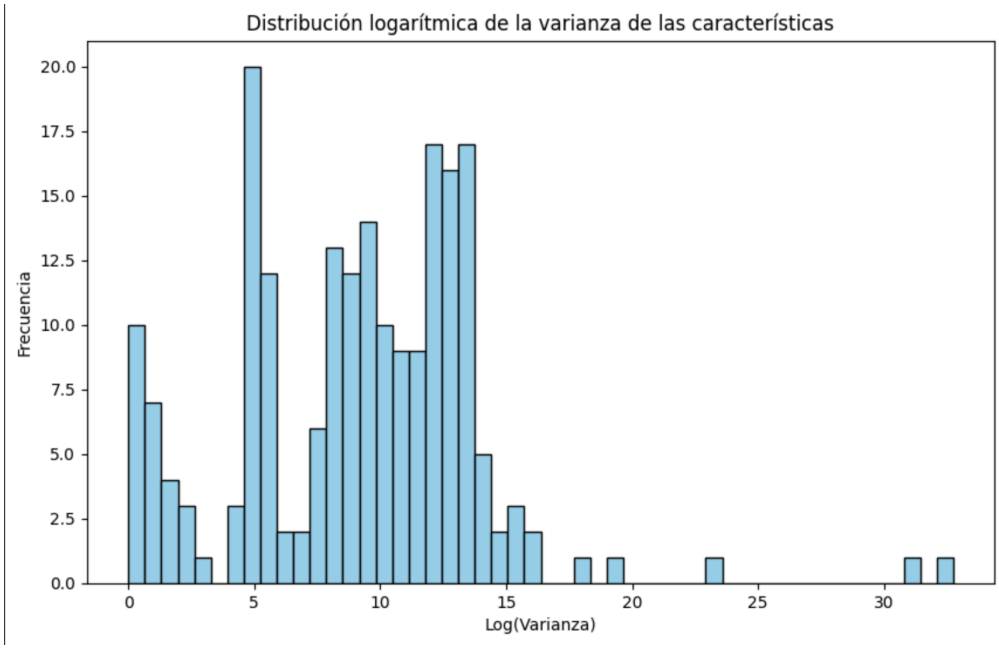
Después de esto eliminamos las columnas que no se van a utilizar y esto nos deja con 10 columnas importantes como las cuales mencionamos anteriormente.

Clientes
Los datos de clientes es donde se tiene mayor número de columnas, para revisar mejor estas columnas se utiliza análisis para revisar datos nulos, para esto escogimos utilizar selección umbral para así revisar los datos.

La mayoría de las columnas tienen varianzas muy pequeñas, lo que indica que los valores en esas columnas son muy similares entre sí o que están casi constantes. Mientras que otras (pocas) tienen una varianza extremadamente alta.



Las diferencias en las magnitudes de las varianzas pueden ser tan grandes que las varianzas pequeñas están aplastadas contra el eje, por lo que aplicamos una transformación logarítmica para visualizar mejor las diferencias.



Hay varias características con una varianza muy baja (valores de log cercanos a 0), lo que indica que estas características tienen poca variabilidad y podrían ser redundantes o irrelevantes para algunos modelos. Y nuevamente se observan algunas características con varianzas muy grandes (mayores de 25). Estas son variables con

una dispersión muy alta y podrían requerir atención, ya que pueden afectar el desempeño de algunos modelos.

Todos los datos se revisan por sección para saber mejor el uso que se puede tener así como las escuelas, horarios, servicios, etc.

- Por encima de 60-70% de ceros: Si una columna tiene más del 60-70% de ceros, es probable que la información en esa columna no sea relevante o útil para la mayoría de los registros. En este caso, es recomendable eliminarla, ya que podría introducir ruido en el modelo.
- Entre 30% y 60% de ceros: En este rango, puede ser útil revisar si esos ceros representan datos faltantes o tienen algún significado específico. Si la columna es relevante para el problema, puedes considerar técnicas como imputación o transformación.
- Menos del 30% de ceros: Normalmente, las columnas con un bajo porcentaje de ceros no deberían afectar significativamente los modelos de clustering.

Algoritmos sensibles a las distancias (como K-Means): Las columnas con muchos ceros pueden desbalancear las distancias entre puntos, lo que puede llevar a agrupamientos subóptimos. En estos casos, eliminar columnas con más del 60% de ceros es aconsejable.

Algoritmos que no dependen de distancias métricas (como DBSCAN): Pueden ser más robustos ante columnas con ceros, pero aún así, eliminar columnas con muchos ceros puede mejorar la eficiencia y claridad del modelo.

Llevamos a cabo un análisis inicial en el cual examinamos los datos desde distintas perspectivas para comprender mejor su estructura y utilidad para nuestro modelo. Durante este proceso, identificamos cuáles datos son relevantes y podían aportar valor al análisis, además de detectar que varias columnas presentaban datos faltantes, lo cual podría influir en el desempeño y precisión del modelo.

Features Seleccionados

Movilidad

- mov_domingo
- mov_lunes
- mov_martes
- mov_miercoles
- mov_jueves

- mov_viernes
- mov_sabado
- MovMañana
- MovTardeMovNoche

Lugares Cercanos
- viviendas_300m
- escuelas
- Servicios
- Entretenimiento
- Infra

Economía
- socioeconomic_status_rgm
- NivelE

Edades
- NiñosyAdolescentes
- JovenesyAdultos
- AdultosyMayores

Cluster
- ClusterTipo

Categorías de productos

- AGUA
- AGUA MINERAL CON SABOR
- BEBIDAS CON ALCOHOL
- CATEGORIAS EN EXPANSION
- LACTEOS
- REFRESCOS

Hemos estado usando features que describen a nuestros clientes en función de lo que los rodea y del tipo de clientes que llegan a su negocio, sin embargo no hemos puesto features que describen al cliente en función de lo que ofrece a sus compradores. Para poder incluir esta información en nuestro modelo, creamos estas features, que señalan el porcentaje de volumen de ventas por categoria, esto nos ayuda a entender que es lo que el cliente suele comprar.

Descripción de producto

- Brand
- MLSize
- Returnability
- Size
- Flavor
- Container
- ProductCategory
- SegDet

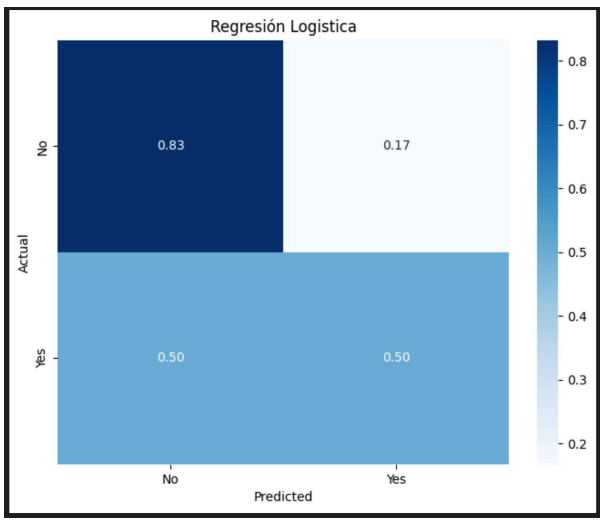
'industry_customer_size' 'sub_canal_comercial'

Recursive Feature Elimination

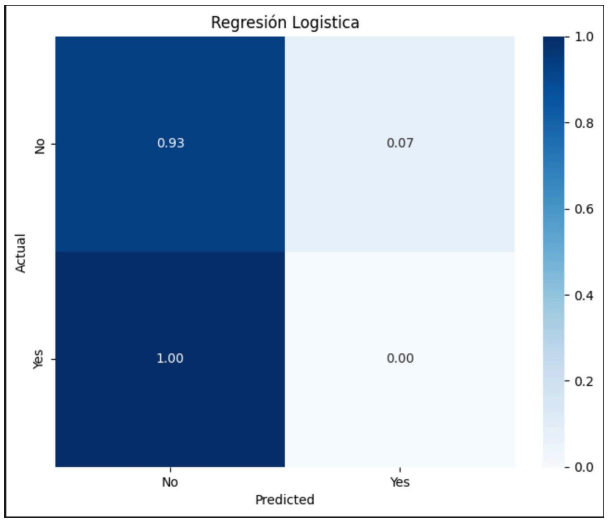
Se utiliza este método para seleccionar las variables más importantes para el modelo, esté método se encarga de entrenar modelos y evaluarlos utilizando distintas variables. A las variables las califica de acuerdo a una métrica y elimina las peores, en este caso se ajustó el parámetro para que seleccionara solamente 25 features.

```python
from sklearn.feature_selection import RFECV
from sklearn.linear_model import LogisticRegression
from sklearn.feature_selection import RFE
from sklearn.pipeline import Pipeline
from imblearn.pipeline import Pipeline as ImbPipeline  # For SMOTE with scikit-learn Pipeline

# Define the model
model = LogisticRegression(random_state=42, max_iter= 10000)

# Define RFE with the model and number of features to select
rfe = RFE(estimator=model, n_features_to_select=25)

# Create the pipeline
pipeline = ImbPipeline(steps=[
    ('preprocessor', preprocessor),
    ('smote', SMOTE(sampling_strategy=1, k_neighbors=3, random_state=42)),
    ('rfe', rfe),
    ('classifier', model)
])

# Fit the pipeline on the training data
pipeline.fit(X_train, y_train)

# Make predictions
y_pred = pipeline.predict(X_test)

# Evaluate the model
from sklearn.metrics import accuracy_score, classification_report

print("Accuracy:", accuracy_score(y_test, y_pred))
print(classification_report(y_test, y_pred))
```

Resultados (Regresión Logística)

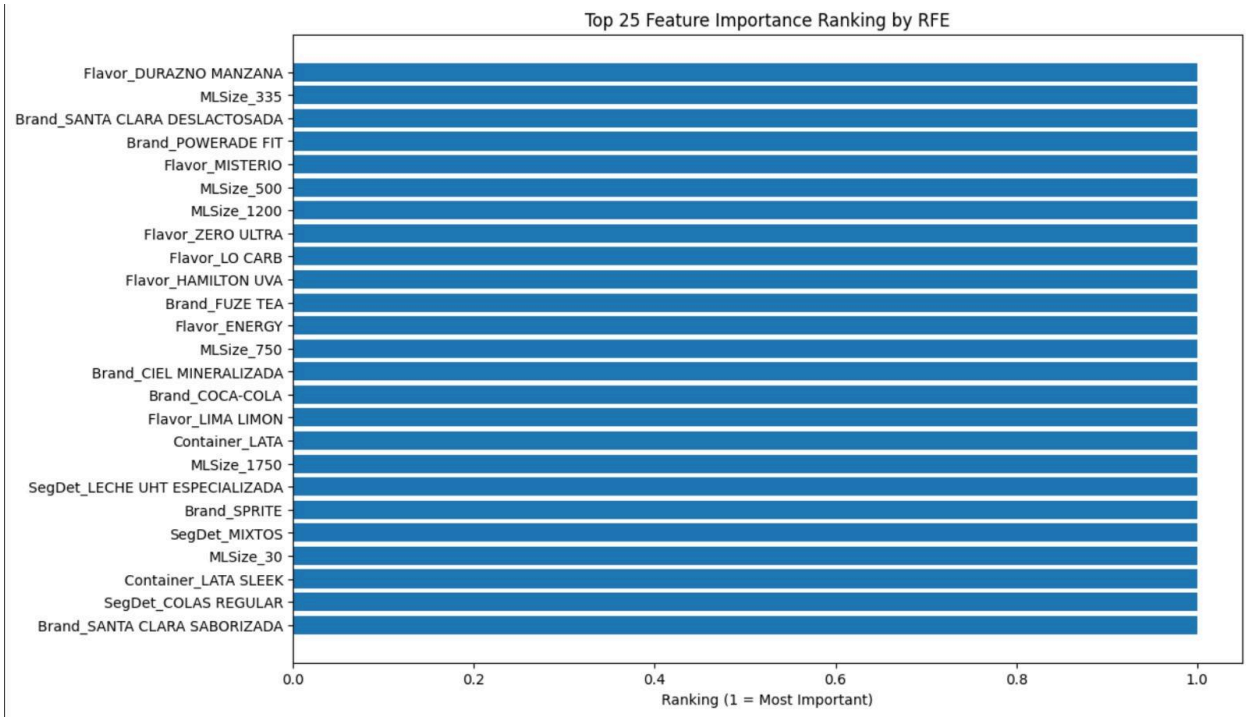


Sin RFE



Con RFE

Top 25 Features



Como se puede observar las matrices de confusión, la solución obtenida por el RFE tiene más accuracy en general, sin embargo, no categoriza ningún producto de lanzamiento exitoso correctamente, mientras que el modelo sin RFE sí logra clasificar la mitad correctamente. Debido a que este es el grupo más importante, se considerarán todas las variables seleccionadas antes de utilizar el RFE.