

PROYECTO APLICADO EN ANALÍTICA DE DATOS - MIAD

ANOMALIES DETECTION AND MARKET BASKET ANALYSIS - FRUBANA

ANÁLISIS EXPLORATORIO DE DATOS:

Órdenes de venta: Se realizó la lectura de 12 archivos .pkl con las ordenes de ventas mensuales de 1 año, tras la unificación de estos archivos, se tienen 502.154 registros y 14 columnas obteniendo el dataframe “**ventas_año_completo**”.

El mes con menor número de registros es abril (32.672) y el de mayor número de registros es marzo (46.944). Volumen y tipos de datos originales:

(502154, 14)	nro_orden	object
month	conteo	datetime64[ns]
0	1	40680
1	2	39292
2	3	46944
3	4	32672
4	5	40304
5	6	42228
6	7	41294
7	8	45546
8	9	43207
9	10	41839
10	11	43538
11	12	44610

Productos: Se realizó la lectura de “Productos_BAQ.csv”. La tabla originalmente contiene 137 registros y 11 columnas. Se agregó la columna “producto_general”:

product_id	int64
sku	object
name	object
producto_general	object
category	object
region_code	object
product_category_id	int64
mean_shelf_life	int64
promised_lead_time	float64
purchasing_unit	float64
buy_unit	object
weight_parameter_apricot	float64

Luego de depuraciones y transformaciones, se obtuvo el df “**productos_final**”, que contiene 238 registros y 4 columnas: product_id, name, producto_general, category.

Principales validaciones y transformaciones aplicadas a los datos:

- Se realizó cambio de “product_id” a tipo entero en df “ventas_año_completo”.
- La columna “producto_general” fue creada localmente para agrupar productos del mismo tipo. Ejemplo, todos los tipos de tomate tienen producto general “Tomate”.

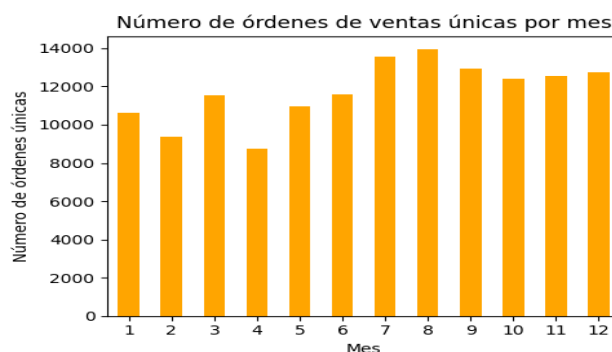
- Se realizó el cruce del df “ventas_año_completo”. vs. La tabla de productos recibida originalmente, identificando 101 códigos de productos que figuraban en las órdenes de venta, pero no en la tabla de productos.
- Se completó la tabla de productos, incluyendo los “producto_id” faltantes asignándoles su respectivo “category” y “producto_general”, como resultado se obtuvo el df “productos_final”.
- Unión de “ventas_año_completo” y “productos_final” usando la llave “producto_id”.
- Se validó que no existen valores nulos en el df resultante.
- Se eliminan registros duplicados usando como llave la combinación de los 5 campos: nro_orden, fecha, cantidad, customer_id, Producto_id.
El df resultante pasó de contener 502.154 a 501.636.
- No se realiza tratamiento de datos atípicos de ventas, dado que en caso de que existan se espera que el modelo de anomalías a implementar, logre alertarlos.

Dataframe final: corresponde al dataframe resultante “df” que unifica “ventas_año_completo” y “productos_final” luego de las depuraciones y transformaciones aplicadas, el cual será usado como insumo para los algoritmos a implementar.

```
Index(['nro_orden', 'fecha', 'producto', 'cantidad', 'precio', 'descuento',
      'customer_id', 'sku', 'producto_id', 'product_quantity_x_step_unit',
      'product_step_unit', 'product_unit', 'sku_parent', 'month', 'Nombre',
      'producto_general', 'Categoria', 'fecha_str'],
      dtype='object')
```

Este df contiene 501.636 registros y 18 columnas. Con un total de ordenes únicas de venta (sin duplicados por nro_orden) de 140.989 órdenes.

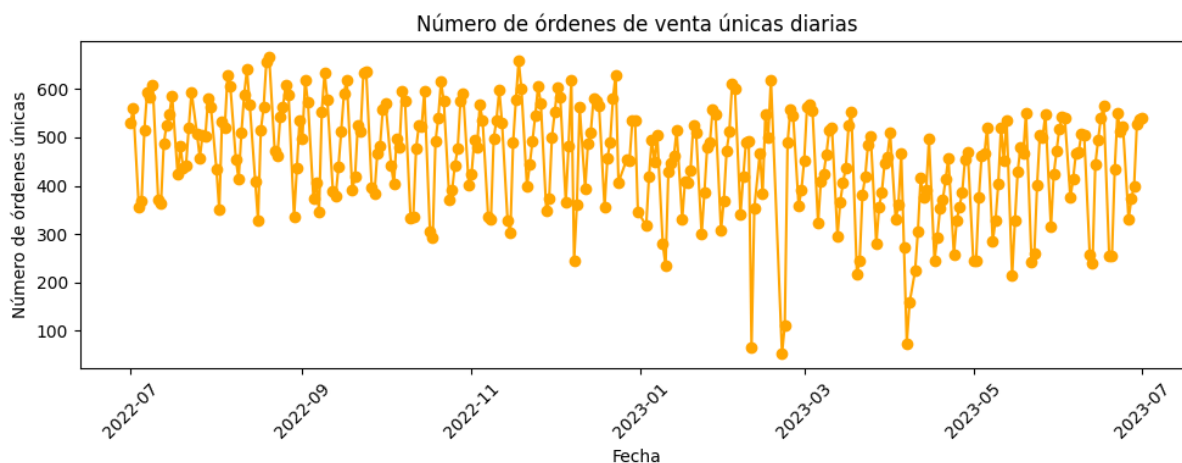
Comportamiento mensual del número de órdenes únicas de venta:



El mes con menor número de órdenes únicas es abril (8.769) y el de mayor número de es agosto (13.932).

Comportamiento diario del número de órdenes única de ventas:

Se cuenta con información de ventas desde julio 2022 hasta junio 2023.



Top 5 de días con menor y mayor número de órdenes únicas de venta:

2023-02-21	53	2022-08-20	668
2023-02-10	66	2022-11-18	659
2023-04-07	74	2022-08-19	656
2023-02-22	112	2022-08-12	641
2023-04-08	158	2022-09-24	637

- El día con mayor número de órdenes únicas de venta registradas fue el 20ago2022 (sábado) con 668 órdenes.
- El día con menor número de órdenes únicas de venta registradas fue el 21feb2023 (martes) con 53 órdenes.

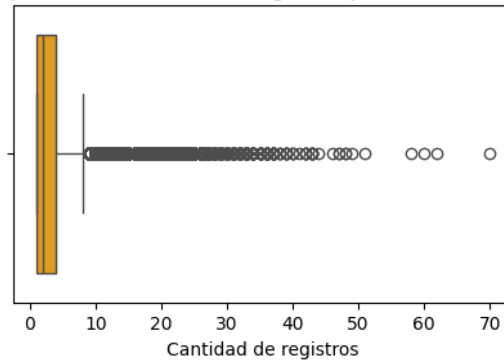
Total de órdenes únicas de venta según día de la semana:

	Número de órdenes únicas	Porcentaje
Lunes	16773	11.9
Martes	18923	13.4
Miércoles	23633	16.8
Jueves	25637	18.2
Viernes	28033	19.9
Sábado	27990	19.9

No se registran ventas los días domingo. Los días de la semana con mayor número de órdenes únicas de venta registradas son: viernes y sábado (19,9%). El día de menor volumen es el lunes (11,9%).

Registros por orden de venta:

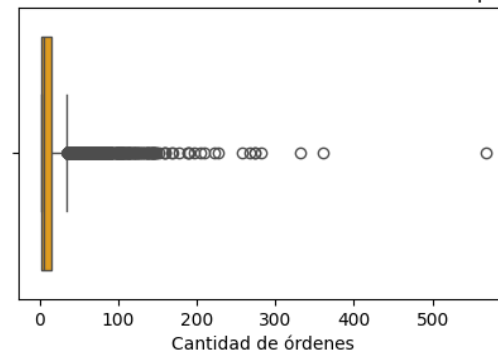
Boxplot de la cantidad de registros por orden de venta



- Una orden de venta tiene en promedio 3,6 registros distintos (productos).
- La orden de venta con mayor número de registros corresponde a 14483185 (nro_orden), la cual tiene 70 registros distintos.

Órdenes de venta únicas por cliente:

Boxplot de la cantidad de órdenes de venta únicas por cliente

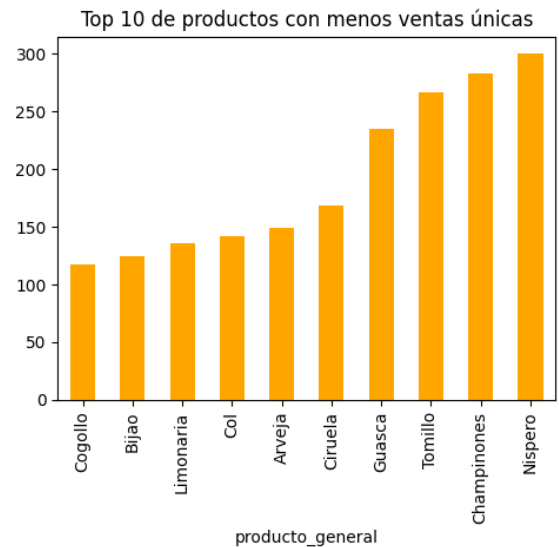
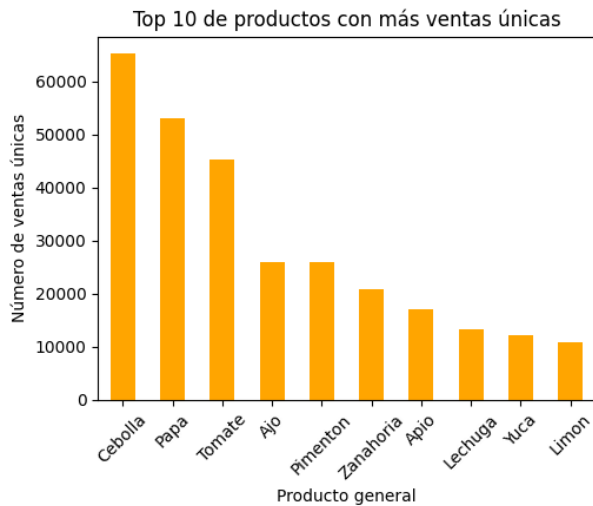


- El número promedio de registros únicos de órdenes de venta por cliente es 12.
- Existen 11.759 clientes con registro de al menos 1 orden de venta en el periodo.
- Top 5 de clientes que registran un mayor número de ordenes únicas de venta:

customer_id	
c1f7d038-3744-4dae-b2df-486895bb6837	568
b3795e9d-4775-4452-98f4-8be08474ceb1	361
2e1565b3-120a-49b5-ad4d-e18d0f8ed68b	331
807837bb-b93b-4f06-b442-0f5f7ae577a5	282
55940636-955c-40db-8c28-83f8dd0e18c7	273

Top 10 de productos generales con mayor y menor número de ordenes únicas de venta:

Existen registros de ventas para un total de 71 productos generales, correspondientes a 232 producto_id distintos.



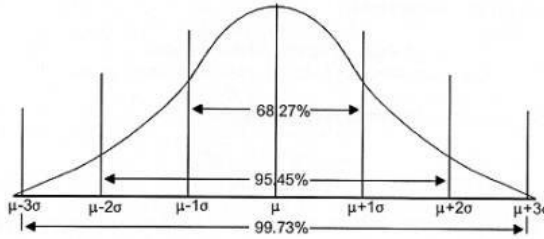
- El producto general que registra un mayor número de ordenes únicas de venta es: Cebolla. Figurando en un total de 65.189 órdenes.
- El producto general que registra un menor número de ordenes únicas de venta es: Cogollo. Figurando en un total de 117 órdenes.

Producto_general con mayor número de ordenes únicas de ventas, por producto_id:

El producto_general “Cebolla” fue incluido en un total de 65.189 órdenes únicas de venta. A continuación, el detalle según producto_id y Nombre:

producto_id	Nombre	ventas_unicas
258690	Cebolla Cabezona Blanca Sin Pelar Mixta - Desde 1Kg	25748
26107	Cebolla Larga Junca Atado - Unidad	17494
258696	Cebolla Cabezona Blanca Sin Pelar Mixta Al por mayor	10634
1848	Cebolla Roja Mixta Mixta Desde 1Kg	10153
171	Cebolla Roja Mixta Mixta - Desde 5kg	9827
62904	Cebolla Roja Mixta Mixta Al por mayor	5189
73211	Cebolla Cabezona Blanca Sin Pelar Grande - Kg	1872
724	Cebolla Roja Mixta Pequeña Kg	1790
107	Cebolla Cabezona Blanca Limpia Mixta - Kg	1722
36878	Cebolla Puerro Estándar - Unidad	1432
287738	Cebollín Limpio Atado Al por mayor	1356
287277	Cebollín Sucio Estándar Al por mayor	1063
613087	Cebolla Roja Mixta Estandar Grande	400
258790	Cebolla Cabezona Roja Grande Cero (Grande) - Kg	398
616670	Cebolla Roja Grande Cero (Grande) Grande	382
62906	Cebolla Roja Mixta Pequeña Al por mayor	380
62896	Cebolla Cabezona Blanca Limpia Mixta Al por mayor	315
609343	Cebolla Blanca Baby Estandar Pequeña	133
1849	Cebolla Roja Mixta Pequeña Kg (Tamaño 8YB)	107
611088	Cebolla Cabezona Blanca Sin Pelar Mixta Kg - 8YB (Insuperable)	40

VALIDACIÓN DE SUPUESTOS:



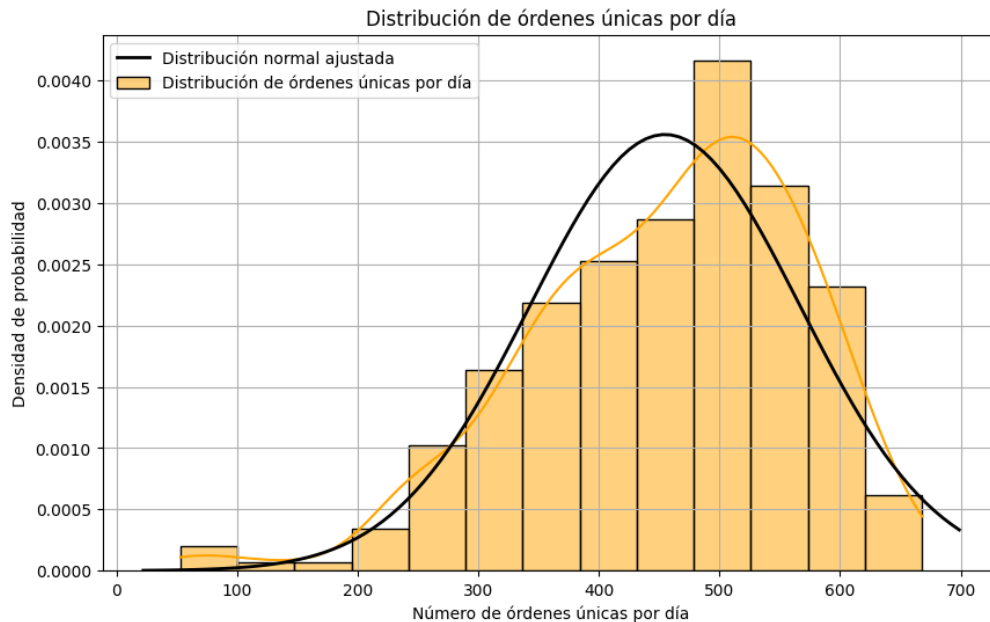
Distribución normal de los datos: También conocida como distribución gaussiana, es una de las distribuciones más importantes en estadística y se caracteriza por su forma de campana simétrica. En una distribución normal, la media, la mediana y la moda son iguales y se encuentran en el centro de la distribución. La mayoría de los datos (alrededor del 68 %) están dentro de una desviación estándar de la media, y aproximadamente el 95 % de los datos están dentro de dos desviaciones estándar de la media. La distribución normal está completamente definida por dos parámetros: la media (μ) y la desviación estándar (σ).

Sobre las ventas únicas diarias se realizaron pruebas de ajuste de diferentes tipos de distribución y la más semejante corresponde a la distribución normal la cual tiene el p valor más alto, aquí los resultados:

H0 Los datos se ajustan a una distribución normal, H1 no se ajusta, cuando el valor p es < que (0.05) se rechaza la hipótesis nula, en este caso $p=0.08$ es mayor a 0.05, por lo tanto, se rechaza la hipótesis nula y se asemeja a una distribución normal.

```
norm: Estadístico KS=0.0714406171791897, Valor p=0.08043897332311678
expon: Estadístico KS=0.378228696901925, Valor p=2.334680177140878e-40
gamma: Estadístico KS=0.08048882312830624, Valor p=0.03404217674093479
lognorm: Estadístico KS=0.7651218273252687, Valor p=2.6575125290679956e-190
chi2: Estadístico KS=0.07667670967721035, Valor p=0.04951937302347387
La distribución que mejor se ajusta es: norm, Valor p=0.08043897332311678
```

Se corrobora la semejanza con el siguiente gráfico de distribución:



ALGORITMO DE DETECCIÓN DE ANOMALÍAS:

Alternativas de algoritmos/modelos validados y descartados:

1. **Isolation Forest:** Un algoritmo que utiliza árboles de decisión para aislar las anomalías en el conjunto de datos.

Ventajas: Eficiente para conjuntos de datos de gran tamaño y no requiere supuestos sobre la distribución de los datos.

Desventajas: Puede tener dificultades con datos de alta dimensionalidad y no es tan efectivo para detectar anomalías en datos densamente agrupados.

2. **One-Class SVM:** Un tipo de SVM que se entrena en datos normales y luego puede detectar anomalías como puntos que están lejos del límite de decisión del modelo.

Ventajas: Eficaz para detectar anomalías en datos de alta dimensión y puede manejar bien conjuntos de datos desequilibrados.

Desventajas: Sensible a la elección de hiperparámetros y puede ser computacionalmente costoso para conjuntos de datos grandes.

Algoritmo seleccionado: alertamiento basado en desviaciones estándar:

Se selecciona este tipo de algoritmo para identificar el top de productos que están por encima o por debajo de los umbrales definidos respecto al comportamiento de ventas (cantidades vendidas y número de ventas) de los últimos 6 meses. Es eficiente al partir de una distribución normal de los datos, sin embargo, no es bloqueante en su aplicación.

Ventajas:

Simplicidad: Es un método sencillo y fácil de entender.

Rápido de implementar: No requiere un conocimiento profundo del dominio o del algoritmo en sí mismo para implementarlo.

Robusto: Funciona bien para datos que se distribuyen normalmente y cuando las desviaciones del comportamiento normal pueden ser medidas con precisión.

Interpretación clara: Las alertas se basan en umbrales fácilmente interpretables (número de desviaciones estándar de la media).

Desventajas:

Supuestos sobre la distribución de los datos: Este método asume que los datos siguen una distribución normal, lo cual puede no ser cierto en todos los casos.

Sensibilidad a cambios en la distribución: Si los datos no siguen una distribución normal o si la distribución cambia con el tiempo, este método puede generar falsas alarmas.

Falta de sensibilidad a patrones complejos: Este método puede no detectar anomalías que no se ajusten a un patrón de desviación estándar.

Necesidad de ajustar manualmente los umbrales: A menudo, los umbrales de desviación estándar deben ajustarse manualmente, lo que puede ser subjetivo y consumir tiempo.

Cálculo de umbrales alertamiento:

Anomalías por disminución: Promedio (número de ventas o cantidades vendidas) - desviación estándar definida * desviación del número de ventas o cantidades vendidas.

Anomalías por aumento: Promedio (número de ventas o cantidades vendidas) + desviación estándar definida * desviación del número de ventas o cantidades vendidas.

Cálculo de flags alertamiento:

Anomalías por disminución: Indica 1 si el valor de número de ventas o cantidades vendidas del último mes es inferior al umbral de disminución definido en el anterior cálculo, 0 de lo contrario.

Anomalías por aumento: Indica 1 si el valor de número de ventas o cantidades vendidas del último mes es superior al umbral de aumento definido en el anterior cálculo, 0 de lo contrario.

Calibración de parámetros:

Se realiza la calibración de parámetros modificando el valor de la desviación estándar para considerar anomalías.

A continuación, tenemos los resultados con el valor de desviación estándar **1**:

Número de productos generales únicos en el consolidado ventas es: 71

Resumen anomalías identificadas:

Productos con anomalía por disminución número de ventas: ['Acelga', 'Apio', 'Arracacha', 'Champinones', 'Durazno', 'Guasca', 'Habichuela', 'Jengibre', 'Kiwi', 'Limonaria', 'Manzana', 'Melon', 'Nispero', 'Papaya', 'Pepino', 'Pera', 'Pina', 'Zucchini'] - Número de alertas: 18

Productos con anomalía por disminución cantidades vendidas: ['Acelga', 'Apio', 'Arracacha', 'Champinones', 'Jengibre', 'Kiwi', 'Limonaria', 'Manzana', 'Melon', 'Nispero', 'Patilla', 'Pera', 'Pina'] - Número de alertas: 13

Productos con anomalía por aumento número de ventas: ['Aguacate', 'Ahuyama', 'Ajo', 'Bijao', 'Coco', 'Cogollo', 'Lulo', 'Mandarina', 'Maracuya', 'Tomate', 'Uchuva', 'Yuca'] - Número de alertas: 12

Productos con anomalía por aumento cantidades vendidas: ['Aguacate', 'Ajo', 'Berenjena', 'Bijao', 'Cebollin', 'Coco', 'Cogollo', 'Guayaba', 'Hierbabuena', 'Lechuga', 'Lulo', 'Mandarina', 'Mazorca', 'Pimenton', 'Platano', 'Tomate', 'Uchuva', 'Uva'] - Número de alertas: 18

Total de alertas: 61

Total de productos generales únicos alertados: 40

Tasa de alertas (producto_general): 56.34 %

Además, tenemos los resultados con el valor de desviación estándar 2:

Número de productos generales únicos en el consolidado ventas es: 71

Resumen anomalías identificadas:

Productos con anomalía por disminución número de ventas: ['Apio', 'Kiwi', 'Nispero', 'Pera'] - Número de alertas: 4

Productos con anomalía por disminución cantidades vendidas: ['Nispero'] - Número de alertas: 1

Productos con anomalía por aumento número de ventas: ['Aguacate', 'Ajo', 'Bijao', 'Lulo', 'Maracuya', 'Tomate', 'Yuca'] - Número de alertas: 7

Productos con anomalía por aumento cantidades vendidas: ['Aguacate', 'Bijao', 'Cogollo', 'Lulo', 'Tomate'] - Número de alertas: 5

Total de alertas: 17

Total de productos generales únicos alertados: 12

Tasa de alertas (producto_general): 16.9 %

Finalmente, los resultados con el valor de desviación estándar 3:

Número de productos generales únicos en el consolidado ventas es: 71

Resumen anomalías identificadas:

Productos con anomalía por disminución número de ventas: ['Nispero'] - Número de alertas: 1

Productos con anomalía por disminución cantidades vendidas: ['Nispero'] - Número de alertas: 1

Productos con anomalía por aumento número de ventas: ['Ajo', 'Bijao', 'Yuca'] - Número de alertas: 3

Productos con anomalía por aumento cantidades vendidas: ['Bijao', 'Cogollo', 'Tomate'] - Número de alertas: 3

Total de alertas: 8

Total de productos generales únicos alertados: 6


Tasa de alertas (producto_general): 8.45 %

Según la calibración de parámetros realizada para el algoritmo de alertamiento y por la cantidad de alertas generadas, se utilizará el valor de desviación estándar de 2. Esto dado que al utilizar el valor 1 se genera alertamiento para el 56% de los productos, siendo un valor bastante alto y tomaría la mayor cantidad de datos como anómalos; por su parte, el valor 3 genera alertamiento para solo el 8% de los productos, lo cual, no favorece la creación del ranking de alertamiento propuesto al ser demasiado baja la cantidad de productos alertados. Lo anterior, teniendo en cuenta que los datos de ventas únicas diarias siguen una distribución normal.

Resultados:


Productos alertados por disminución en número de ventas:

Los productos con anomalía por disminución en número de ventas en el último mes son: 4

	producto_general c	promedio_conteo...	desviacion_cont...	umbral_inf_cont...	ult_mes_conteo f	anomalía_dism_c...	 Visualize	diferencia_conteo f...
7	Apio	1566.3	98	1370.4	1275	1		-291.3
34	Kiwi	45.3	10.4	24.4	21	1		-24.3
49	Nispero	25.3	3.8	17.8	3	1		-22.3
54	Pera	57	15.2	26.7	24	1		-33


Productos alertados por aumento en número de ventas:

Los productos con anomalía por aumento en número de ventas en el último mes son: 7

	producto_...	promedio_conteo...	desviacion_conteo...	umbral_sup_conteo f	ult_mes_conteo flo...	anomalía_aum_co...	 Visualize	diferencia_conteo f...
63	Tomate	4568.5	337.5	5243.5	5517	1		948.5
43	Maracuya	451.2	432.6	1316.4	1387	1		935.8
5	Ajo	2098.2	71.8	2241.8	2716	1		617.8
2	Aguacate	281.3	89.9	461.1	534	1		252.7
67	Yuca	921.7	46.4	1014.4	1100	1		178.3
39	Lulo	187.7	60.4	308.5	328	1		140.3
13	Bijao	9	4.9	18.7	37	1		28


Productos alertados por disminución en cantidades vendidas:

Los productos con anomalía por disminución de cantidades vendidas en el último mes son: 1

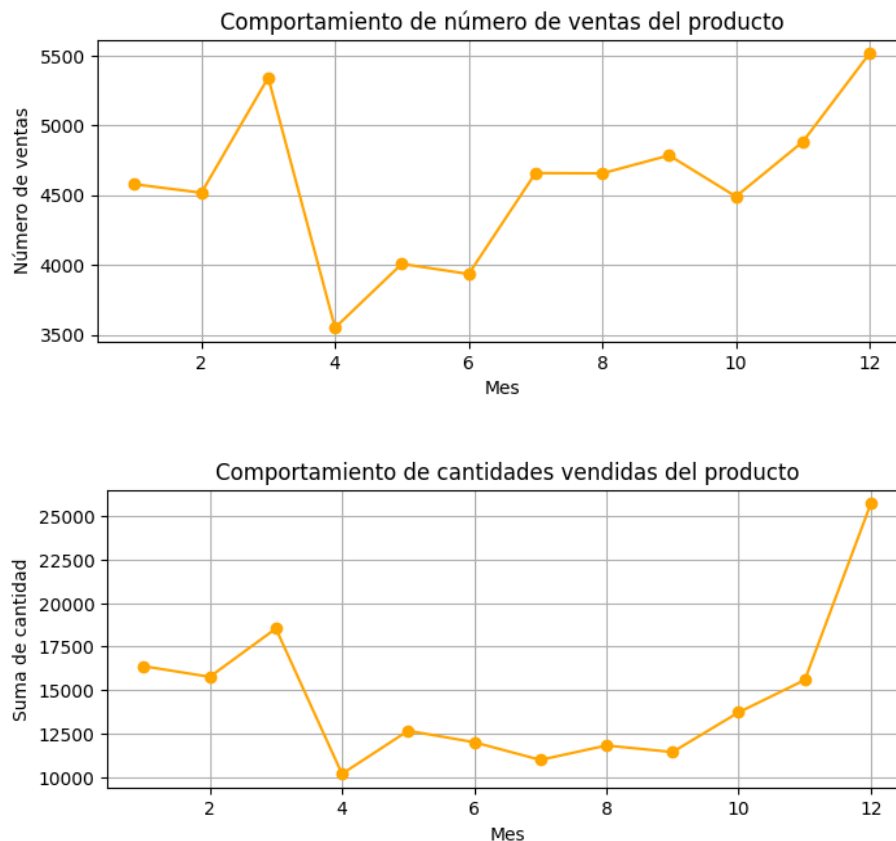
	producto_...	promedio_suma_c...	desviacion_suma...	umbral_inf_cantid...	ult_mes_canti...	anomalía_dism_ca...	 Visualize	diferencia_cantidaa f
49	Nispero	39.8	7.9	24	3	1		-36.8

Productos alertados por aumento en cantidades vendidas:

Los productos con anomalía por aumento de cantidades vendidas en el último mes son: 5

	producto_...	promedio_suma_c...	desviacion_suma...	umbral_sup_cant...	ult_mes_cantidad f...	anomalía_aum_ca...	 Visualize	diferencia_cantidad f
63	Tomate	12585.3	1736.9	16059.2	25730	1		13144.7
2	Aguacate	797.7	299.3	1396.3	1687	1		889.3
39	Lulo	458.5	194.8	848.1	982	1		523.5
13	Bijao	16.8	10.8	38.4	86	1		69.2
21	Cogollo	16.3	8.5	33.3	45	1		28.7

Comportamiento general de un producto específico (Tomate):



MODELO MARKET BASKET:

Alternativas de algoritmos/modelos validados y descartados:

1. Descomposición de valores singulares (SVD): En el análisis de cesta de compra, los datos suelen representarse como una matriz binaria o de frecuencia. Cada fila de la matriz representa una transacción y cada columna representa un artículo en el inventario. Los elementos de la matriz indican si un artículo está presente en una transacción (1 si está presente, 0 si no lo está) o la cantidad de veces que aparece un artículo en una transacción. Posteriormente se aplica SVD, y se hace reducción de dimensionalidad manteniendo los conceptos latentes más importantes (Es decir los k primeros valores singulares). Lo anterior permite identificar comportamientos latentes de diferentes categorías de productos

2. Dismiliaridad del Coseno: Para la recomendación de análisis de canasta de compra también se puede utilizar la técnica de asociación por disimilaridad del coseno, calculando una matriz binaria o de frecuencia como en el SVD, donde cada fila representa una transacción y los valores de 1 identifican una compra de un producto realizada, mediante el cálculo de distancia de coseno se identifican las transacciones más similares a la de un usuario interesado y se realizan sugerencias según estas.

Algoritmo seleccionado: Apriori (Análisis de canasta de compra):

El análisis de asociación en minería de datos se fundamenta en tres medidas clave: soporte, confianza y lift. El soporte representa la probabilidad de que un artículo o conjunto de artículos aparezca en la canasta de compra. Por ejemplo, el soporte de un producto X sería la probabilidad de comprar ese producto. Además, se puede calcular el soporte conjunto de dos productos, X e Y, representando la probabilidad de comprar ambos productos simultáneamente.

La confianza, por otro lado, indica la frecuencia con la que una regla es cierta. Se basa en la probabilidad condicional y mide la probabilidad de que un producto Y haya sido comprado dado que se compró el producto X. En esencia, la confianza evalúa la fuerza de la asociación entre dos productos, X e Y, en términos de probabilidad de compra conjunta respecto a la probabilidad de comprar el primer producto solo.

El lift, siendo el tercer componente, es un indicador crucial en el análisis de asociación. Este ratio compara la confianza observada de una regla $X \rightarrow Y$ con el soporte del producto Y. Su rango va desde 0 hasta infinito y ofrece una medida relativa del aumento en la probabilidad conjunta de compra en comparación con lo que se esperaría si los productos fueran independientes. Un lift de 1 indica ausencia de correlación, mientras que un lift mayor a 1 sugiere una asociación positiva entre los productos, y un lift menor a 1 señala una asociación negativa. En resumen, el lift es esencial para comprender cómo la compra de un producto afecta la probabilidad de compra de otro producto.

Para el modelo de Análisis de canasta de compra primero se convirtieron los datos en el formato de Listas para el algoritmo Apriori y posteriormente se procedió a calcular las reglas de asociación con base a los hiperparametros establecidos.

Ventajas:

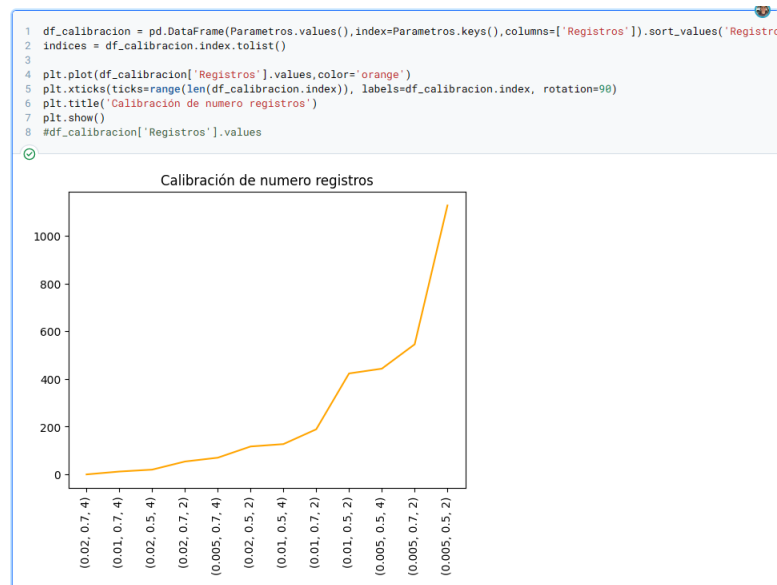
- Fácil de implementar: El algoritmo Apriori es relativamente sencillo de entender e implementar, lo que lo hace accesible e interpretable para todas las partes interesadas.
- Eficiente para conjuntos de datos pequeños a medianos: En conjuntos de datos de tamaño moderado, el algoritmo Apriori puede ser bastante eficiente y proporcionar resultados rápidos.
- Identificación de patrones de asociación: Apriori es útil para identificar relaciones de asociación entre elementos en conjuntos de datos transaccionales, lo que puede ayudar a comprender mejor el comportamiento de los usuarios, patrones de compra o preferencias de los clientes.

Desventajas:

- Costo computacional elevado: El principal inconveniente del algoritmo Apriori es su costo computacional, especialmente en conjuntos de datos grandes. El tiempo de ejecución puede aumentar significativamente a medida que crece el número de elementos únicos y el tamaño de los conjuntos de datos.
- Sensibilidad a la métrica de soporte: El rendimiento del algoritmo Apriori puede depender en gran medida del valor del umbral de soporte utilizado para identificar

conjuntos de elementos frecuentes. En algunos casos, la elección de un valor adecuado puede requerir cierta experimentación y ajuste.

Calibración de parámetros:



df_calibracion utilizando los valores de un diccionario llamado Parametros, con las claves como índices y los valores como datos, luego se ordena este DataFrame por los valores en la columna 'Registros'. Se genera una lista de índices a partir de este DataFrame. Luego, se realiza una visualización mediante un gráfico de líneas de los valores de 'Registros' del DataFrame utilizando matplotlib.

Se identifica que con la calibración de números de registros, la mejor combinación de hiperparametros es **min_support=0.01, min_confidence=0.5, min_lift=4**

```
1 Datos_para_apriori = df[['nro_orden', 'producto_general']].drop_duplicates()
2
3 records = []
4 for i in Datos_para_apriori['nro_orden'].unique():
5     records.append(list(Datos_para_apriori[Datos_para_apriori['nro_orden'] == i]['producto_general'].values))
6 records
```

El código comienza extrayendo un subconjunto de datos de un DataFrame llamado `df`, que contiene columnas denominadas **'nro_orden'** y **'producto_general'**. Este subconjunto de datos se crea seleccionando solo las columnas **'nro_orden'** y **'producto_general'** del DataFrame original y eliminando las filas duplicadas. Esto resulta en un DataFrame llamado **'Datos_para_apriori'** que contiene únicamente los pares únicos de **'nro_orden'** y **'producto_general'**.

Luego, se inicializa una lista vacía llamada **'records'**. A continuación, el código itera sobre los valores únicos de la columna **'nro_orden'** en el DataFrame **'Datos_para_apriori'**. En cada iteración, se seleccionan los productos asociados con el número de orden actual y se agregan a la lista **'records'** como una lista de valores. Finalmente, el código devuelve la lista **'records'**, que contiene listas de productos asociados con cada número de orden único en el

DataFrame original. En resumen, este código prepara los datos en un formato adecuado para el análisis de reglas de asociación utilizando el algoritmo **Apriori**.

Resultados:

Reglas de asociación generadas:

```
1 association_rules = apriori(records, min_support=0.01, min_confidence=0.5, min_lift=4, max_length=3)
2 association_results = list(association_rules)
3 association_results
4
5 print("Derivamos {} reglas de asociación.".format(len(association_results)))
```

Derivamos 127 reglas de asociación.

El fragmento de código presenta el proceso de generación y evaluación de reglas de asociación utilizando el algoritmo **Apriori**. Primero, se aplica el algoritmo **Apriori** a los datos preparados en la lista `records`, con ciertos criterios de soporte mínimo (`min_support`), confianza mínima (`min_confidence`), lift mínimo (`min_lift`) y longitud máxima de las reglas (`max_length`). El resultado se almacena en `association_rules`.

```
1 print(association_results[0])
RelationRecord(items=frozenset({'Mazorca', 'Apio'}), support=0.015838115030250587, ordered_statistics=[OrderedStatistic(items_base=frozenset(
```

En este caso las Zanahorias se compran a menudo con Habichuelas: - El valor de soporte es 0.02, que nos dice que el 2% de los ordenes compraron Zanahorias. - El nivel de confianza es .62 nos dice que la probabilidad de que los usuarios Compren Zanahorias dado que compraron Habichuelas es del 62%. - Finalmente, el lift de 4.21 nos dice que la probabilidad de que alguien compre Zanahorias dado que compro Habichuelas es 4.2 veces más altas a la probabilidad de comprarlos de manera independiente

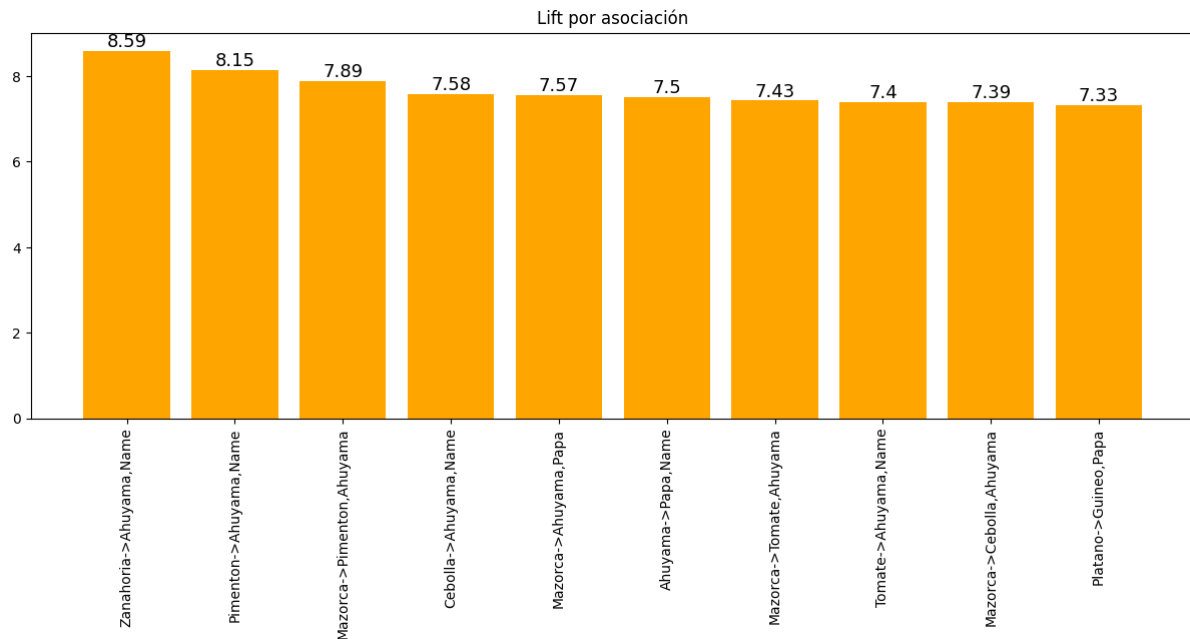
Para ver ver reglas de asociación usamos la función reglas

```
1 def ver_reglas(num):
2     #Primer índice de la lista interna:
3     #Contiene elemento base y el adicional
4     item = association_results[num]
5     pair = item[0]
6     items = [x for x in pair]
7     print("Regla: " + items[0] + " -> " + items[1])
8
9     #Segundo índice de la lista interna
10    print("Soporte: " + str(item[1]))
11
12    #Tercer índice de la lista ubicada en 0th
13    #del tercer índice de la lista interna
14
15    print("Confianza: " + str(item[2][0][2]))
16    print("Lift: " + str(item[2][0][3]))
17    print("=====")
18 ver_reglas(2)
```

Regla: Zanahoria -> Mazorca
Soporte: 0.019192986686904653
Confianza: 0.6079532689283307
Lift: 4.115755470706636
=====

Lift por asociación de productos

Se presenta el top10 reglas con mayor lift de las 127 reglas obtenidas por el algoritmo Apriori. Todas las reglas tienen un lift mayor a 4, y la más alta es de 8.59



PLAN DE IMPLEMENTACIÓN:

1. Preparación de los datos (Ejecutado):

- **Cargar los archivos CSV:** Importar los archivos CSV que se generan con python que contienen los datos necesarios para el análisis. Estos archivos tienen contener información sobre las anomalías y Market basket:

Anomalías:

df_alertamiento

Con (71 filas, y 17 columnas)

Las columnas son:

producto_general	object
promedio_conteo_registros	float64
desviacion_conteo_registros	float64
promedio_suma_cantidad	float64
desviacion_suma_cantidad	float64
umbral_inf_conteo	float64
umbral_sup_conteo	float64
umbral_inf_cantidad	float64
umbral_sup_cantidad	float64
ult_mes_conteo	float64
ult_mes_cantidad	float64
anomalia_dism_conteo	int64
anomalia_aum_conteo	int64
anomalia_dism_cantidad	int64

anomalia_aum_cantidad	int64
diferencia_conteo	float64

Market basket:

df_Reglas_asociación
Con(127 filas y 4 columnas)
Las columnas son:

Regla	object
Soporte	float64
Confianza	float64
Lift	float64

- **ETL (Ejecutado):** Realizar el proceso de ETL para limpiar y transformar los datos según sea necesario. Esto puede incluir la eliminación de datos duplicados, corrección de errores, unificación de formatos de datos, este paso se realiza antes de importar los datos a power bi, en python se realizan todos esos pasos, garantizando el ETL.

Contenido (Por ejecutar):

- **Matriz de frecuencia:** Utilizar las funciones de análisis de Power BI para crear una matriz de frecuencia que muestre la frecuencia de ocurrencia de diferentes variables, como productos, meses, etc. Esto puede ayudar a identificar tendencias y patrones en los datos.
- **Top de productos con mayor lit:** Utilizar las funciones de agregación en Power BI para calcular la suma de cantidades vendidas por producto y luego mostrar los productos con las mayores sumas. Esto puede proporcionar información sobre los productos más populares o los que generan más ingresos.
- **Anomalías:** realizar el ranking de los productos con aumento y disminución con comportamientos anómalos detectados de ventas en el último mes con base en el comportamiento de los últimos 6 meses. Realizar las gráficas que permitan visualizar este comportamiento mensual.

3. Creación del tablero de control (Por ejecutar):

- **Diseño del tablero:** Diseñar el tablero de control en Power BI utilizando visualizaciones como gráficos de barras, gráficos circulares, tablas dinámicas, etc. Organizar las visualizaciones de manera que proporcionen una vista clara y concisa de los datos relevantes para la empresa.
- **Interactividad:** Aprovechar las capacidades interactivas de Power BI para permitir a los usuarios explorar los datos de manera dinámica. Esto puede incluir filtros interactivos, segmentación de datos, etc.
- **Publicación y distribución:** Publicar el tablero de control en Power BI Service para que los usuarios puedan acceder a él a través de la web o dispositivos móviles. Establecer permisos de acceso según sea necesario para garantizar la seguridad de los datos.

4. Monitoreo y mantenimiento (Por ejecutar):

- **Monitoreo del rendimiento:** Supervisar el rendimiento del tablero de control y realizar ajustes según sea necesario para garantizar que siga siendo útil y efectivo para los usuarios.
- **Actualización de datos (Sujeto a ejecución por Frubana):** Programar actualizaciones periódicas de los datos para garantizar que la información en el tablero de control esté siempre actualizada y refleje la situación más reciente de la empresa.

Siguiendo estos pasos, se podrá implementar un tablero de control en Power BI para Frubana que proporcione información valiosa para la toma de decisiones empresariales.