

PONTIFÍCIA UNIVERSIDADE CATÓLICA DE MINAS GERAIS
NÚCLEO DE EDUCAÇÃO A DISTÂNCIA
Pós-graduação *Lato Sensu* em Ciência de Dados e Big Data

João Rafael Quadros dos Santos

**ANÁLISE DE REGRESSÃO E ASSOCIAÇÃO PARA GERAÇÃO DE
INDICADORES NO CONTEXTO DE MARKET BASKET ANALYSIS.**

Belo Horizonte

2024

João Rafael Quadros dos santos

**ANÁLISE DE REGRESSÃO E ASSOCIAÇÃO PARA GERAÇÃO DE
INDICADORES NO CONTEXTO DE MARKET BASKET ANALYSIS.**

Trabalho de Conclusão de Curso apresentado
ao Curso de Especialização em Ciência de
Dados e Big Data como requisito parcial à
obtenção do título de especialista.

Belo Horizonte
2024

SUMÁRIO

1. Introdução.....	4
1.1. Contextualização	4
1.1. O problema proposto	4
2. Coleta de Dados	4
3. Processamento/Tratamento de Dados	6
4. Análise e Exploração dos Dados	11
5. Criação de Modelos de Machine Learning	15
6. Apresentação dos Resultados	24
7. Links	26

1. Introdução

1.1. Contextualização

O trabalho apresentado visa demonstrar a aplicação prática de técnicas de Data Science e Big Data no contexto do varejo, especificamente na análise de cestas de compras (Market Basket Analysis). Em um mundo cada vez mais conectado, compreender o comportamento de compra dos consumidores é crucial para otimizar estratégias de marketing e melhorar a experiência do cliente. A análise de cestas de compras permite identificar padrões de consumo, possibilitando que redes de supermercados desenvolvam promoções mais eficazes e personalizadas. Utilizando um dataset público, a pesquisa visa não apenas a replicabilidade do estudo, mas também a validação dos resultados, contribuindo para o avanço das práticas de análise de dados na área do varejo.

1.2. O problema proposto

O objetivo central deste trabalho é investigar as relações entre produtos no setor de varejo global, utilizando dados de compras para desenvolver indicadores que sustentem decisões estratégicas de vendas e recomendações de produtos. A identificação de padrões de consumo é essencial, pois não apenas enriquece a experiência do cliente, mas também impulsiona as receitas por meio de promoções direcionadas. O dataset analisado é de domínio público e reflete transações de redes de supermercados em todo o mundo, abrangendo registros de compras dos anos de 2010 e 2011, com uma perspectiva geográfica abrangente.

2. Coleta de Dados

Os dados utilizados neste trabalho foram obtidos a partir de um dataset público disponível no Kaggle (<https://www.kaggle.com/datasets/aslanahmedov/market-basket-analysis/data>). O dataset contém informações sobre transações de compras em redes de supermercados ao redor do mundo, abrangendo os anos de 2010 e 2011. A

estrutura do dataset é composta por várias colunas que descrevem os itens comprados, suas quantidades e preços. A seguir, apresentamos uma tabela com a descrição de cada campo/coluna do dataset:

Nome da Coluna	Descrição	Tipo
BillNo	Número de 6 dígitos atribuído a cada transação.	Nominal
Itemname	Nome do produto.	Nominal
Quantity	Quantidade de cada produto por transação.	Numérico
Date	Data e hora em que cada transação foi gerada.	Numérico
Price	Preço do produto.	Numérico
CustomerID	Número de 5 dígitos atribuído a cada cliente.	Nominal
Country	Nome do país onde reside cada cliente.	Nominal

Figura 1 – Exemplo de registros do dataset.

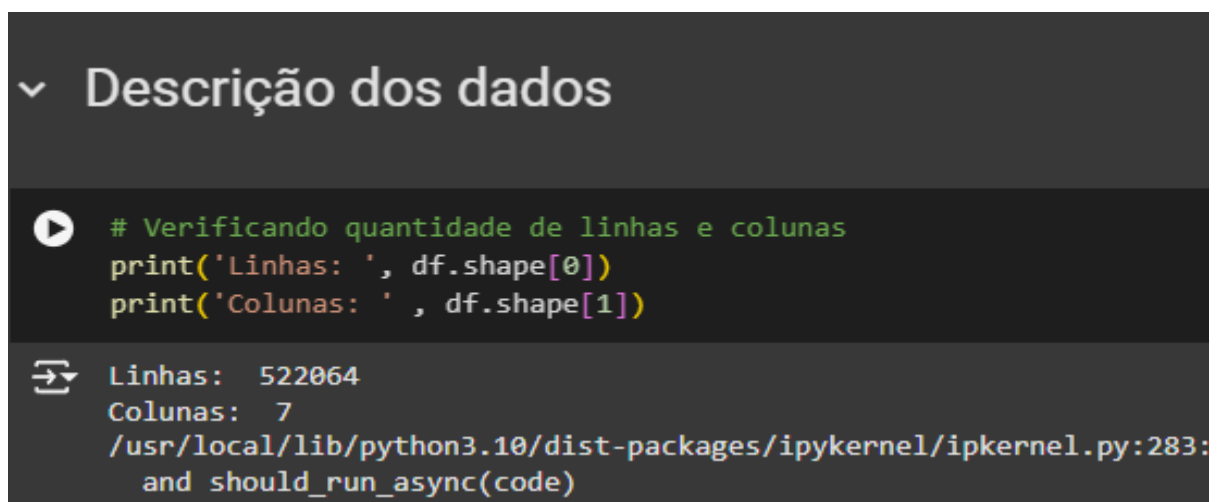
 /usr/local/lib/python3.10/dist-packages/ipykernel/ipkernel.py:283: DeprecationWarning: `should_run_async` will not call and should_run_async(code)

	BillNo	Itemname	Quantity	Date	Price	CustomerID	Country
0	536365	WHITE HANGING HEART T-LIGHT HOLDER	6	2010-12-01 08:26:00	2.55	17850.00	United Kingdom
1	536365	WHITE METAL LANTERN	6	2010-12-01 08:26:00	3.39	17850.00	United Kingdom
2	536365	CREAM CUPID HEARTS COAT HANGER	8	2010-12-01 08:26:00	2.75	17850.00	United Kingdom
3	536365	KNITTED UNION FLAG HOT WATER BOTTLE	6	2010-12-01 08:26:00	3.39	17850.00	United Kingdom
4	536365	RED WOOLLY HOTTIE WHITE HEART.	6	2010-12-01 08:26:00	3.39	17850.00	United Kingdom

3. Processamento/Tratamento de Dados

O conjunto de dados analisado possui 522.065 registros e 7 colunas. Inicialmente, alteramos os cabeçalhos das colunas para facilitar a leitura, renomeando-os para 'Pedido', 'Produto', 'Quantidade', 'Data', 'Preco', 'Id_Cliente' e 'Pais'.

Figura 2 – Descrição dos dados.

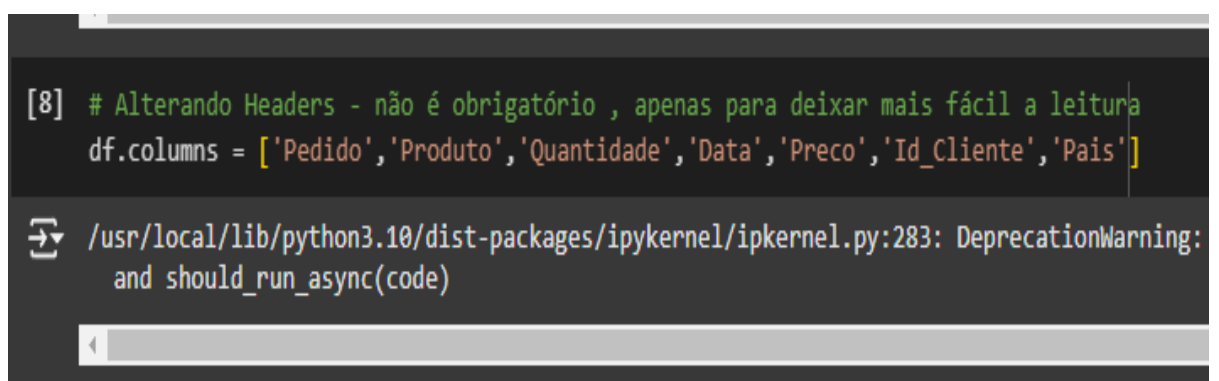


A screenshot of a Jupyter Notebook interface. The top section is titled 'Descrição dos dados' with a dropdown arrow. Below the title, there is a code cell with a play icon. The code checks the number of rows and columns of a DataFrame. The output cell shows the results: 522064 lines and 7 columns. A warning message is also visible at the bottom of the output cell.

```
# Verificando quantidade de linhas e colunas
print('Linhas: ', df.shape[0])
print('Colunas: ', df.shape[1])
```

```
Linhas: 522064
Colunas: 7
/usr/local/lib/python3.10/dist-packages/ipykernel/ipkernel.py:283:
and should_run_async(code)
```

Figura 3 – Alteração de cabeçalho.



A screenshot of a Jupyter Notebook interface. The code cell shows the modification of DataFrame headers. The output cell shows a DeprecationWarning message. A scrollbar is visible at the bottom of the output cell.

```
[8] # Alterando Headers - não é obrigatório , apenas para deixar mais fácil a leitura
df.columns = ['Pedido', 'Produto', 'Quantidade', 'Data', 'Preco', 'Id_Cliente', 'Pais']
```

```
/usr/local/lib/python3.10/dist-packages/ipykernel/ipkernel.py:283: DeprecationWarning:
and should_run_async(code)
```

Figura 4 – Análise Descritiva.



Durante a análise descritiva, observou-se que os campos 'Quantidade' e 'Preco' contêm valores negativos, o que é anômalo e requer tratamento. Valores negativos em 'Quantidade' podem indicar erros de entrada de dados, enquanto os negativos em 'Preco' podem resultar de retornos ou ajustes incorretos. Esses dados inválidos precisam ser tratados para garantir a integridade das análises subsequentes.

A seguir, fizemos uma análise da distribuição dos países, produtos e pedidos, bem como verificações para identificar valores nulos e duplicados. Observamos que os campos 'Produto' e 'Id_Cliente' apresentavam dados ausentes, e o conjunto

continha 5.286 registros duplicados. Diante dos dados negativos e nulos, consideramos estratégias para o tratamento. Inicialmente, ponderamos substituir os valores negativos na 'Quantidade' e 'Preco' pela média ou mediana dessas colunas.

Figura 5 – Valores para produto.

<pre>df['Produto'].value_counts().sort_values(ascending=True) # verificando a distribuição dos produtos e a quantidade de registros de cada um. # df['Produto'].value_counts().sort_values(ascending=False) # mesma situação que a distribuição acima, porém, do maior para o menor.</pre>		
DOORMAT KEEP CALM AND COME IN	727	
LUNCH BOX I LOVE LONDON	729	
ROUND SNACK BOXES SET OF 4 WOODLAND	737	
RED HANGING HEART FLIGHT HOLDER	740	
RECYCLING BAG RETROSPOT	750	
LUNCH BAG ALPHABET DESIGN	750	
GIN + TONIC DIET METAL SIGN	753	
CHARLOTTE BAG PINK POLKADOT	755	
CLOTHES PEGS RETROSPOT PACK 24	759	
PINK REGENCY TEACUP AND SAUCER	764	
GARDENERS KNEELING PAD CUP OF TEA	769	
ALARM CLOCK BAKELIKE PINK	769	
SCOTTIE DOG HOT WATER BOTTLE	779	
HOME BUILDING BLOCK WORD	783	
PAPER BUNTING RETROSPOT	793	
HOT WATER BOTTLE KEEP CALM	794	
POPCORN HOLDER	803	
JUMBO BAG STRAWBERRY	827	
PAPER CHAIN KIT VINTAGE CHRISTMAS	829	
60 TEATIME FAIRY CAKE CASES	832	
SET OF 6 SPICE TINS PANTRY DESIGN	837	
SET OF 3 REGENCY CAKE TINS	841	
WOODLAND CHARLOTTE BAG	842	

Figura 6 – Valores para pedido.

```
df['Pedido'].value_counts().sort_values(ascending=True) # verificando a distribuição dos pedidos e a quantidade de registros de cada um
# df['Produto'].value_counts().sort_values(ascending=False) # mesma situação que a distribuição acima, porém, do maior para o menor.
```

559055	487
538524	490
578065	494
540551	502
575875	503
579512	503
539958	512
575477	515
575176	518
576329	518
539437	518
578827	520
575930	526
577768	526
578844	527
536544	527
580727	529

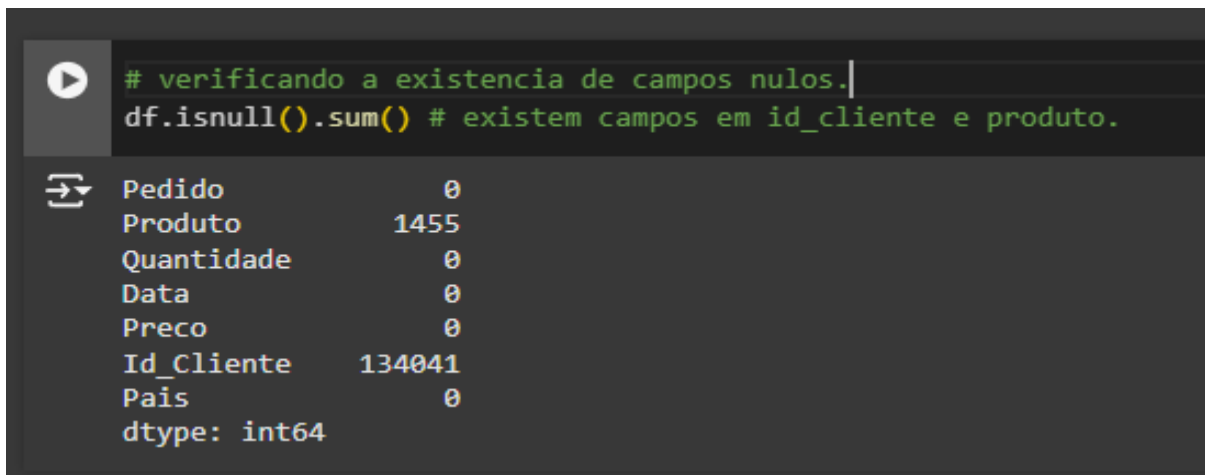
Figura 7 – Valores para quantidades.

```
df['Quantidade'].value_counts().sort_values(ascending=True) # verificando a distribuição de quantidade e a quantidade de registros de cada um.
# df['Pais'].value_counts().sort_values(ascending=False) # mesma situação que a distribuição acima, porém, do maior para o menor.
```

/usr/local/lib/python3.10/dist-packages/ipykernel/ipkernel.py:283: DeprecationWarning: `should_run_async` will not call `transform_cell` automatically and should_run_async(code)

count	
Quantidade	
80995	1
-2003	1
-2618	1
-1671	1
77	1
-657	1
-900	1
148	1
177	1
-905	1
1992	1
912	1
-1000	1
4800	1
490	1

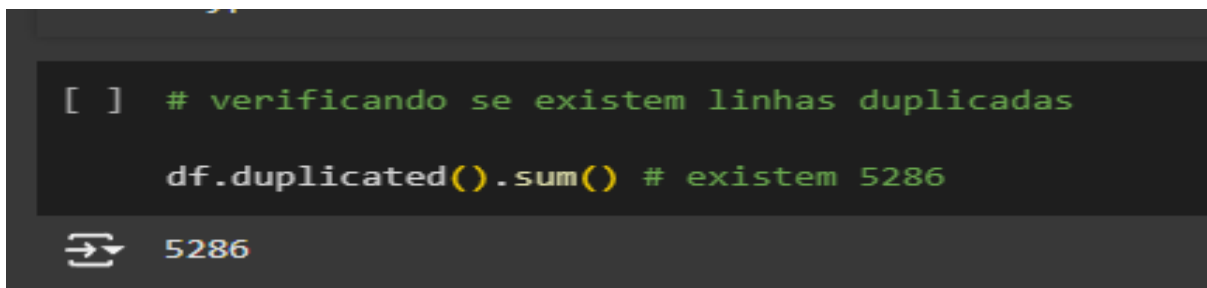
Figura 8 – Valores Nulos.



```
# verificando a existencia de campos nulos.
df.isnull().sum() # existem campos em id_cliente e produto.
```

Pedido	0
Produto	1455
Quantidade	0
Data	0
Preco	0
Id_Cliente	134041
País	0
dtype:	int64

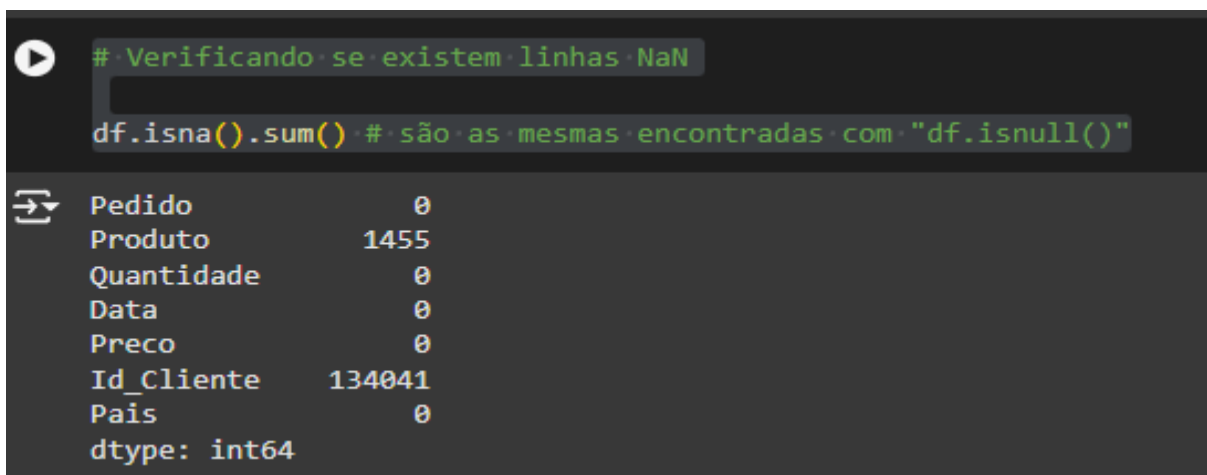
Figura 9 – Valores Duplicados.



```
[ ] # verificando se existem linhas duplicadas
df.duplicated().sum() # existem 5286
```

5286

Figura 10 – Valores “NaN”.



```
# Verificando se existem linhas NaN
df.isna().sum() # são as mesmas encontradas com "df.isnull()"
```

Pedido	0
Produto	1455
Quantidade	0
Data	0
Preco	0
Id_Cliente	134041
País	0
dtype:	int64

No entanto, após uma análise mais aprofundada, decidimos pela exclusão de registros com dados inválidos ou ausentes, dado que a preservação da qualidade dos dados é fundamental para a validade da análise.

Figura 11 – Dropando negativos e duplicados.

```

# Dropar valores NaN
df.dropna(inplace=True)
# Dropando valores negativos para quantidade
df = df[df["Quantidade"] > 0]

[ ] # Na coluna preco, iremos transformar os valores para numéricos e então sim, dropar os negativos.
df = df[df["Preco"] > 0]

[ ] # Dropando duplicados

df=df.drop_duplicates()

```

4. Análise e Exploração dos Dados

Após análise inicial do dataset, foi abordada a totalização das vendas por cliente, criando uma nova coluna chamada 'Preco_Total', que resultou da multiplicação entre o 'Preco' e a 'Quantidade'. Em seguida, os dados foram agrupados por 'Id_Cliente' e Pais, permitindo identificar os clientes com as maiores vendas.

Figura 12 – Criando Preco_Total.

```

# Total de vendas por Cliente
total_vendas = df
total_vendas["Preco_Total"] = total_vendas["Preco"] * total_vendas["Quantidade"] #Criando coluna a partir da multiplicacao entre preco e quantidade
total_vendas_cliente = total_vendas.groupby(["Id_Cliente", "Pais"]).agg({"Preco_Total": "sum"}) # Usando essa coluna criada para visualizar a soma desses precos totalizados agrupados por id_cliente e Pais.
total_vendas_cliente.head(25)

```

Id_Cliente	Pais	Preco_Total
12346.00	United Kingdom	77183.60
12347.00	Iceland	4310.00
12349.00	Italy	1757.55
12350.00	Norway	334.40
12352.00	Norway	2506.04
12353.00	Bahrain	89.00
12354.00	Spain	1079.40
12355.00	Bahrain	459.40
12356.00	Portugal	2811.43
12357.00	Switzerland	6207.67
12358.00	Austria	1168.06
12360.00	Austria	2662.06
12361.00	Belgium	189.90
12362.00	Belgium	5226.23
12363.00	Unspecified	552.00
12364.00	Belgium	1313.10

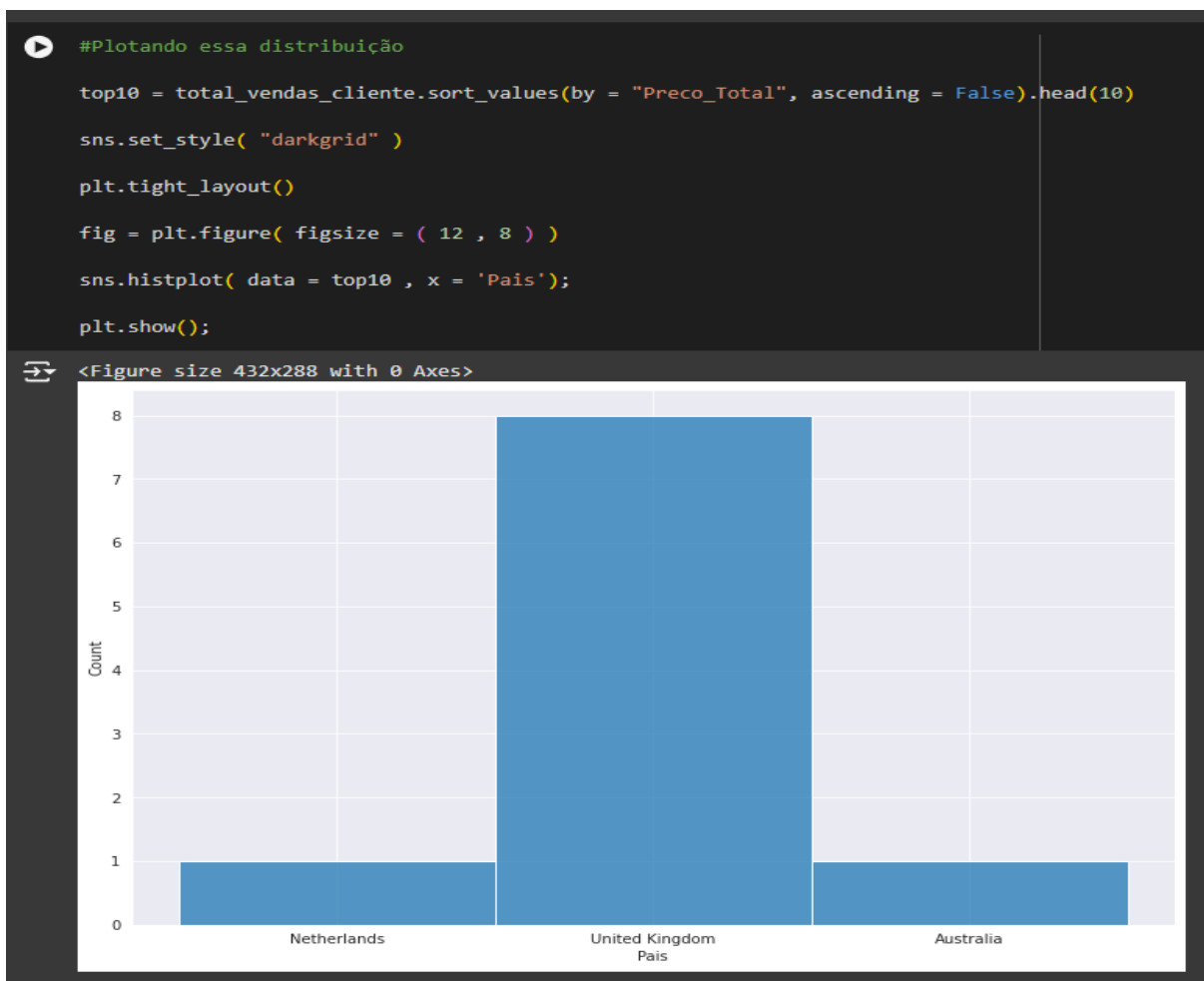
Figura 13 – Maiores Vendas.

```
# top 10 clientes
total_vendas_cliente.reset_index(inplace=True)
total_vendas_cliente.sort_values(by = "Preco_Total", ascending = False).head(10)
```

a partir do dataframe e sua visualização criada anteriormente, é possível organizar os valores de forma decendente
visualizar os 10 maiores valores

	Id_Cliente	Pais	Preco_Total
1663	14646.00	Netherlands	280206.02
4162	18102.00	United Kingdom	259657.30
3690	17450.00	United Kingdom	194390.79
2972	16446.00	United Kingdom	168472.50
44	12415.00	Australia	124914.53
3733	17511.00	United Kingdom	91062.38
2666	16029.00	United Kingdom	80850.84
0	12346.00	United Kingdom	77183.60
3140	16684.00	United Kingdom	66653.56
1266	14096.00	United Kingdom	65164.79

Figura 14 – Distribuição dos 10 maiores clientes por país.



A análise da distribuição das vendas revela que 80% das dez maiores vendas estão concentradas no Reino Unido. Esse insight é significativo, pois indica um mercado forte e potencialmente leal, onde os clientes podem ter uma maior disposição para gastar. Essa concentração também sugere que estratégias de marketing e promoção podem ser especialmente eficazes nesse país, permitindo

que as redes de supermercados aproveitem esse comportamento para otimizar suas vendas e atender melhor às necessidades dos consumidores locais.

Figura 15 – Total de vendas por país.

```
# Total de vendas agrupados por país
total_vendas_cliente.groupby(["País"]).agg({"Preco_Total": "sum").reset_index().sort_values(by="Preco_Total", ascending=False)
```

	País	Preco_Total
27	United Kingdom	7284789.39
15	Netherlands	285446.34
6	Germany	228678.40
5	France	208934.31
0	Australia	138453.81
22	Spain	61558.56
24	Switzerland	56443.95
3	Belgium	41196.34
23	Sweden	38367.83
11	Japan	37416.37
16	Norway	36165.44
18	Portugal	33375.84
21	Singapore	21279.29
10	Italy	17483.24
1	Austria	10198.68
17	Poland	7334.65
9	Israel	7205.84
7	Greece	4760.52
8	Iceland	4310.00
25	USA	3580.39
14	Malta	2725.59
28	Unspecified	2660.77
26	United Arab Emirates	1902.28
12	Lebanon	1693.88
13	Lithuania	1661.06
4	Brazil	1143.60
19	RSA	1002.31
2	Bahrain	548.40
20	Saudi Arabia	145.92

Figura 16 – Total de vendas por país 2.



Figura 17 – Distribuição das maiores vendas por país com trendline.



A visualização das 50 maiores vendas mostra que, de maneira geral, as vendas seguem uma tendência próxima à linha de tendência (trendline), indicando um comportamento consistente entre os clientes. No entanto, o cliente da "Netherlands" se destaca como um outlier, apresentando vendas significativamente mais altas do que a média. Esse ponto fora da curva merece atenção, pois pode indicar um cliente especial ou uma situação particular que influenciou suas compras, sugerindo oportunidades para explorar essa dinâmica no mercado local.

O dataset não contém um valor total de vendas, o que limita as inferências sobre a relação entre variáveis. Portanto, foram levantadas duas hipóteses de análise futura:

1. **Regressão Supervisionada:** Criar um sub-dataset que inclua o total de vendas, permitindo prever vendas futuras com base em variáveis como país e cliente.
2. **Aprendizado Não Supervisionado:** Aplicar algoritmos de clusterização ou associação para explorar padrões de compra e identificar segmentos de clientes.

Essas abordagens visam enriquecer a análise e fornecer insights mais profundos sobre o comportamento de compra no contexto do varejo.

5. Criação de Modelos de Machine Learning

Nesta seção, propomos duas abordagens distintas para a criação de modelos de machine learning, focando na previsão do valor total de vendas. A primeira hipótese envolve a criação de um sub-dataset que agrupa os dados por país e cliente.

A segunda abordagem considera o uso de aprendizado não supervisionado, empregando algoritmos de clusterização ou associação.

5.1 Hipótese 1 - Sub-dataset para Previsão de Preço Total

Nesta etapa, a criação de modelos de machine learning visa prever o 'Preço_Total', utilizando um sub-dataset que agrupa dados por país e cliente. A análise exploratória inicial, incluindo histogramas e gráficos de dispersão, proporciona uma compreensão da distribuição e das relações entre as variáveis.

Os modelos a serem testados incluem Random Forest e Linear Regression, selecionados por sua capacidade de capturar diferentes padrões nos dados. Essa abordagem permitirá comparar o desempenho e a eficácia na previsão do valor total de vendas, fundamentando decisões estratégicas para o negócio.

Figura 18 – Definição das variáveis e divisão em treino e teste.

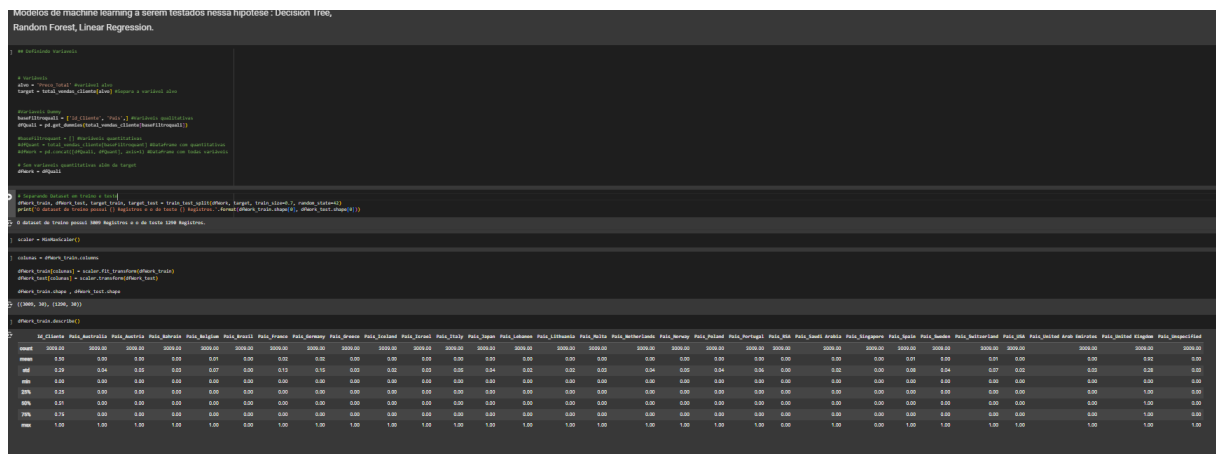


Figura 19 – Correlações.

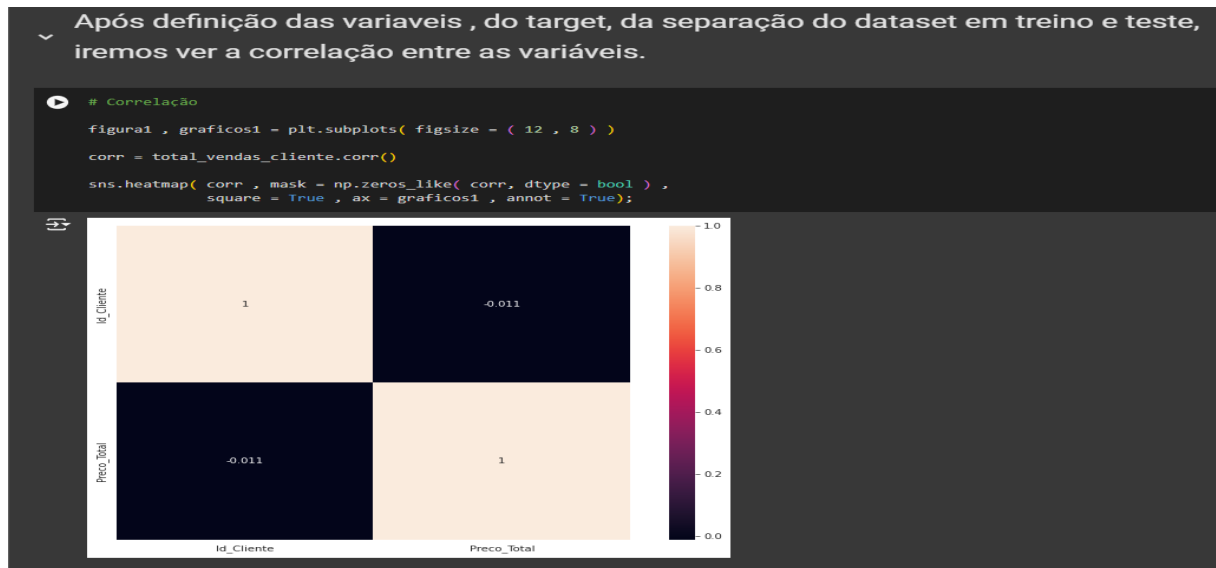


Figura 20 – Regressão Linear.

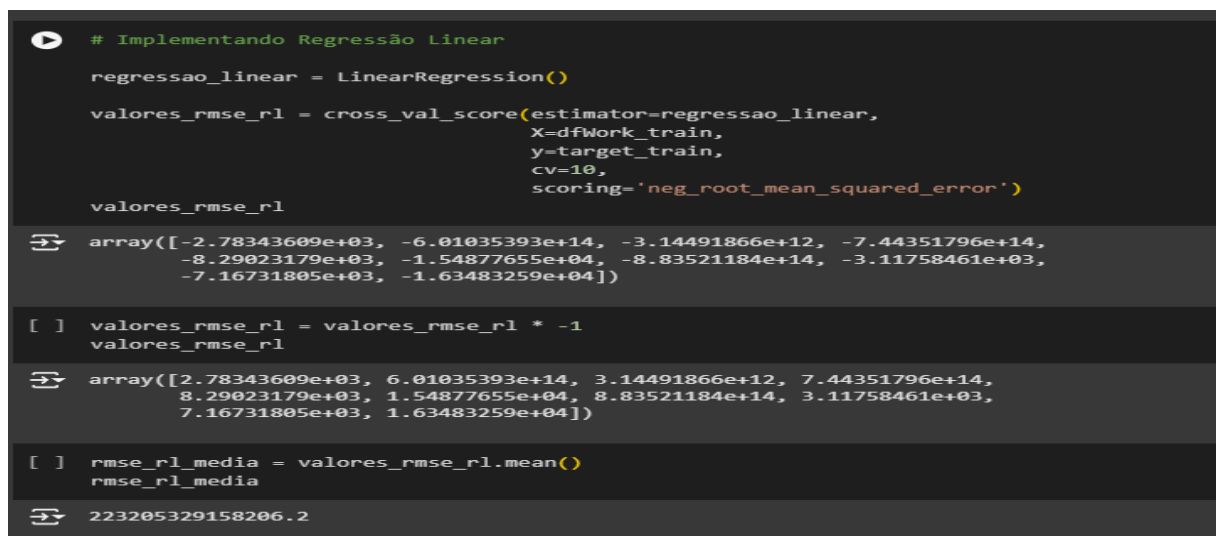


Figura 21 – Random Forest.

```
[ ] rfr_regressao = RandomForestRegressor()

valores_rfr = cross_val_score(estimator=rfr_regressao,
                              X=dfwork_train,
                              y=target_train,
                              cv=10,
                              scoring='neg_root_mean_squared_error')

valores_rfr

array([ -3368.67569346,  -4460.48589729,  -4704.29950939,  -8455.13841508,
        -7549.33711835,  -15403.19259907,  -11880.45277109,  -4906.9200927 ,
        -14507.51380609,  -16563.57269112])

[ ] rfr_regressao.fit(dfwork_train, target_train)

RandomForestRegressor()

for feature, importancia in zip(dfwork.columns, rfr_regressao.feature_importances_):
    print("{}:({}) %".format(feature, (importancia)*100))

Id_Cliente:92.2668350225803 %
Pais_Australia:0.0011405380220192408 %
Pais_Austria:0.0011022057119994988 %
Pais_Bahrain:0.0015574336492341421 %
Pais_Belgium:0.004387112168531182 %
Pais_Brazil:0.0 %
Pais_France:0.017537488189096288 %
Pais_Germany:0.042819237709763334 %
Pais_Greece:0.00010971998304974503 %
Pais_Iceland:0.001465119212715168 %
Pais_Israel:0.007215808182533839 %
Pais_Italy:0.0011958569639073477 %
Pais_Japan:0.11912548573808689 %
Pais_Lebanon:0.0002024157866325293 %
Pais_Lithuania:0.00019739539804103116 %
Pais_Malta:0.0003851882414054139 %
Pais_Netherlands:6.859388937663804 %
Pais_Norway:0.015708240960188642 %
Pais_Poland:0.0025079116977512594 %
Pais_Portugal:0.0015463880813938382 %
Pais_RSA:0.0 %
Pais_Saudi Arabia:6.329774757907154e-05 %
Pais_Singapore:0.0 %
Pais_Spain:0.007096177271539773 %
Pais_Sweden:0.29962779568102854 %
Pais_Switzerland:0.03196812779890294 %
Pais_USA:0.00011228873001327472 %
Pais_United Arab Emirates:0.00020951557508715969 %
Pais_United Kingdom:0.31628495642073556 %
Pais_Unspecified:0.00021025483467448718 %

[ ] valores_rfr_media = valores_rfr.mean() * -1

[ ] valores_rfr_media

9179.958859364298
```

Figura 22 – Comparando modelos.



A partir da comparação dos resultados dos modelos, optou-se por utilizar a regressão linear.

Figura 24 – Resultados Regressão Linear.

predicoes_vs_real.head(20)

	predicao	real	diferenca_abs
0	1732.53	447.68	1284.85
1	2101.66	113.50	1988.16
2	1770.09	627.13	1142.96
3	1612.56	4147.96	2535.40
4	2347.38	1043.10	1304.28
5	2165.56	108.50	2057.06
6	2209.66	1740.48	469.18
7	1469.56	560.47	909.09
8	1561.66	1095.08	466.58
9	2013.34	217.90	1795.44
10	2040.00	309.54	1730.46
11	1963.22	5360.63	3397.41
12	1657.09	3701.44	2044.35
13	2106.97	348.91	1758.06
14	1466.41	689.90	776.51
15	1573.19	1016.14	557.05
16	1866.88	301.32	1565.56
17	1546.38	91.80	1454.58
18	1509.91	207.80	1302.11
19	2153.00	157.90	1995.10

[] r2_score(y_true = target_test , y_pred = precos_preditos) *100

-1.9441913069271958e+20

5.2 Hipótese 2: Aprendizado Não Supervisionado

Nesta hipótese, propomos a utilização de técnicas de aprendizado não supervisionado, como algoritmos de clusterização e associação, para explorar o dataset. O objetivo é identificar grupos de produtos ou clientes com características similares, permitindo a descoberta de padrões ocultos que podem não ser evidentes em análises tradicionais. A abordagem de associação pode, por exemplo, revelar quais produtos são frequentemente comprados juntos, oferecendo insights valiosos para a criação de promoções e estratégias de marketing direcionadas. Esta análise contribuirá para um melhor entendimento do comportamento do consumidor e para a otimização das operações de vendas. Nesse estudo aplicamos duas abordagens, k-means e apriori.

Figura 25 – Ajustando dataset para kmeans

```

# Data
data_k = df

# Variável Target
target_k = df['Produto']

# Convertendo valores não numericos para numericos
le = LabelEncoder()
data_k['Produto'] = le.fit_transform(data_k['Produto'])
target_k = le.transform(target_k)

# printing the dataset
data_k.head()

```

	Pedido	Produto	Quantidade	Preco	Id_Cliente	Pais
0	536365	3667	6	2.55	17850.00	27
1	536365	3675	6	3.39	17850.00	27
2	536365	843	8	2.75	17850.00	27
3	536365	1783	6	3.39	17850.00	27
4	536365	2734	6	3.39	17850.00	27

```

# Aplicando k means com 2 clusters

kmeans = KMeans(n_clusters=2, random_state=0)

kmeans.fit(data_k)

kmeans.cluster_centers_

```

```

array([[5.70944723e+05, 2.01599319e+03, 1.26825670e+01, 2.97316241e+00,
        1.53095502e+04, 2.55388837e+01],
       [5.47957023e+05, 1.97463906e+03, 1.33686465e+01, 3.23007531e+00,
        1.53115718e+04, 2.54509113e+01]])

```

Figura 26 – Resultados com 2 clusters

```

labels = kmeans.labels_

# Verificando quantidade de amostras classificadas corretamente
correct_labels = sum(y == labels)

# resultados
print("Resultado: %d de %d amostras foram corretamente rotuladas." % (correct_labels, y.size))

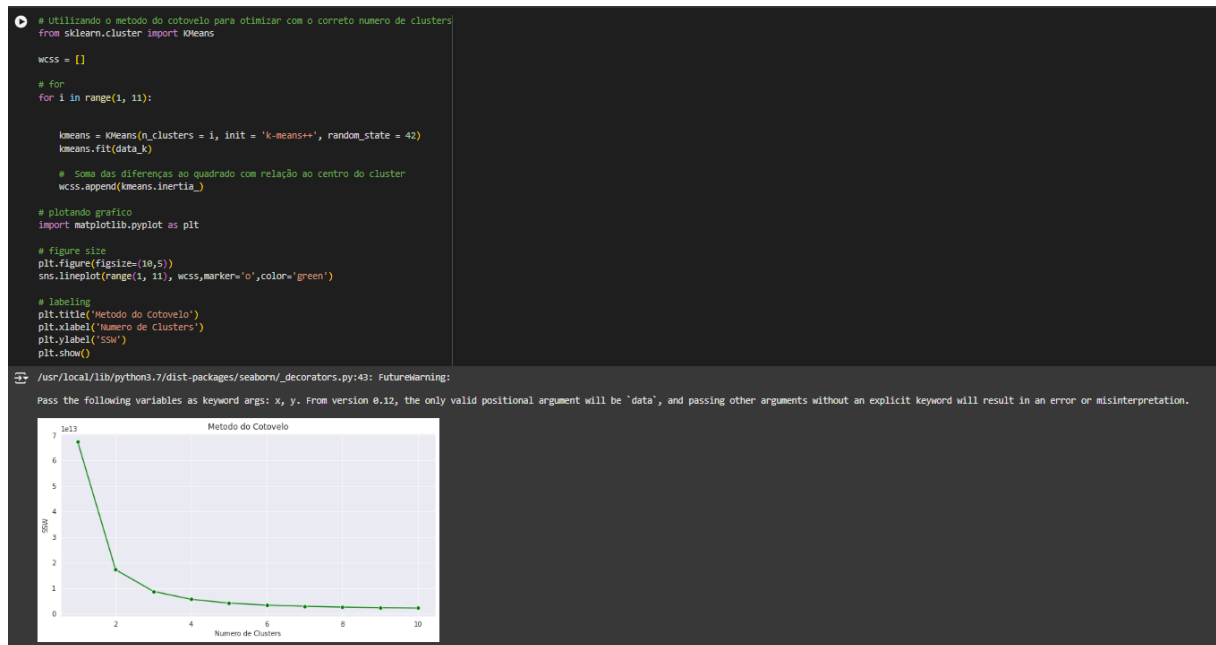
```

```

Resultado: 191 de 38272 amostras foram corretamente rotuladas.

```

Figura 27 – Aplicando método do cotovelo para otimização.



O método do cotovelo é uma maneira prática de escolher o número certo de clusters ao usar o K-Means. A ideia é simples: você plota a soma dos erros quadráticos (SSE) para diferentes números de clusters e observa o gráfico. Conforme você adiciona mais clusters, a SSE diminui, mas chega um ponto onde essa diminuição desacelera—é o "cotovelo". Esse ponto sugere um bom equilíbrio, evitando a complexidade desnecessária e garantindo que os clusters sejam significativos e úteis para a análise.

Figura 30 – Regras de associação.

```
#Regras de associação
produtos_frequentes = apriori(df_matriz, min_support=0.01, use_colnames=True)
produtos_frequentes.sort_values("support", ascending=False)
```

o ideal era utilizar o mínimo support possível, mas devido a quantidade de linhas poderá levar muito tempo para realizar o processo

	support	itemsets
586	0.11	((Quantidade, WHITE HANGING HEART FLIGHT HOLD...))
411	0.09	((Quantidade, REGENCY CAKESTAND 3 TIER))
244	0.09	((Quantidade, JUMBO BAG RED RETROSPOT))
36	0.07	((Quantidade, ASSORTED COLOUR BIRD ORNAMENT))
394	0.07	((Quantidade, PARTY BUNTING))
278	0.07	((Quantidade, LUNCH BAG RED RETROSPOT))
465	0.06	((Quantidade, SET OF 3 CAKE TINS PANTRY DESIGN))
370	0.06	((Quantidade, POSTAGE))
270	0.06	((Quantidade, LUNCH BAG BLACK SKULL))
320	0.05	((Quantidade, PACK OF 72 RETROSPOT CAKE CASES))
529	0.05	((Quantidade, SPOTTY BUNTING))
279	0.05	((Quantidade, LUNCH BAG SPACEBOY DESIGN))
328	0.05	((Quantidade, PAPER CHAIN KIT 50'S CHRISTMAS))
301	0.05	((Quantidade, NATURAL SLATE HEART CHALKBOARD))
273	0.05	((Quantidade, LUNCH BAG CARS BLUE))
203	0.05	((Quantidade, HEART OF WICKER SMALL))
277	0.05	((Quantidade, LUNCH BAG PINK POLKADOT))
280	0.05	((Quantidade, LUNCH BAG SUKI DESIGN))
603	0.05	((Quantidade, WOODEN PICTURE FRAME WHITE FINISH))
242	0.05	((Quantidade, JUMBO BAG PINK POLKADOT))
426	0.05	((Quantidade, REX CASH+CARRY JUMBO SHOPPER))
30	0.05	((Quantidade, ALARM CLOCK BAKELIKE RED))
272	0.05	((Quantidade, LUNCH BAG APPLE DESIGN))
481	0.05	((Quantidade, SET OF 4 PANTRY JELLY MOULDS))
229	0.05	((Quantidade, JAM MAKING SET WITH JARS))
228	0.05	((Quantidade, JAM MAKING SET PRINTED))
40	0.05	((Quantidade, BAKING SET 9 PIECE RETROSPOT))
283	0.04	((Quantidade, LUNCH BAG WOODLAND))
565	0.04	((Quantidade, VICTORIAN GLASS HANGING FLIGHT))
424	0.04	((Quantidade, RETROSPOT TEA SET CERAMIC 11 PC))
377	0.04	((Quantidade, RECIPE BOX PANTRY YELLOW DESIGN))

Figura 31 – Métricas e produtos com maior confiança.

```
# Verificando as regras , antecedentes, consequentes, supports, confiança , convicção, etc.
regras = association_rules(produtos_frequentes, metric="support", min_threshold=0.01)
regras.sort_values("support", ascending=False).head(10)
```

	antecedents	consequents	antecedent support	consequent support	support	confidence	lift	leverage	conviction
217	((Quantidade, JUMBO BAG RED RETROSPOT))	((Quantidade, JUMBO BAG PINK POLKADOT))	0.09	0.05	0.03	0.34	7.22	0.03	1.45
216	((Quantidade, JUMBO BAG PINK POLKADOT))	((Quantidade, JUMBO BAG RED RETROSPOT))	0.05	0.09	0.03	0.62	7.22	0.03	2.44
24	((Quantidade, ALARM CLOCK BAKELIKE GREEN))	((Quantidade, ALARM CLOCK BAKELIKE RED))	0.04	0.05	0.03	0.67	14.25	0.03	2.90
25	((Quantidade, ALARM CLOCK BAKELIKE RED))	((Quantidade, ALARM CLOCK BAKELIKE GREEN))	0.05	0.04	0.03	0.61	14.25	0.03	2.47
408	((Quantidade, LUNCH BAG RED RETROSPOT))	((Quantidade, LUNCH BAG PINK POLKADOT))	0.07	0.05	0.03	0.40	8.00	0.02	1.59
409	((Quantidade, LUNCH BAG PINK POLKADOT))	((Quantidade, LUNCH BAG RED RETROSPOT))	0.05	0.07	0.03	0.56	8.00	0.02	2.10
330	((Quantidade, LUNCH BAG BLACK SKULL))	((Quantidade, LUNCH BAG RED RETROSPOT))	0.06	0.07	0.03	0.49	7.01	0.02	1.82
331	((Quantidade, LUNCH BAG RED RETROSPOT))	((Quantidade, LUNCH BAG BLACK SKULL))	0.07	0.06	0.03	0.40	7.01	0.02	1.57
116	((Quantidade, ROSES REGENCY TEACUP AND SAUCER))	((Quantidade, GREEN REGENCY TEACUP AND SAUCER))	0.04	0.04	0.03	0.69	19.17	0.03	3.10
117	((Quantidade, GREEN REGENCY TEACUP AND SAUCER))	((Quantidade, ROSES REGENCY TEACUP AND SAUCER))	0.04	0.04	0.03	0.77	19.17	0.03	4.26

```
# 10 registros com maiores subidas, e bons valores de confidence e conviction
regras_ordenadas = regras.sort_values("lift", ascending=False)
regras.sort_values("lift", ascending=False).head(10)
```

	antecedents	consequents	antecedent support	consequent support	support	confidence	lift	leverage	conviction
528	((Quantidade, REGENCY SUGAR BOWL GREEN))	((Quantidade, REGENCY MILK JUG PINK))	0.01	0.01	0.01	0.76	56.59	0.01	4.17
529	((Quantidade, REGENCY MILK JUG PINK))	((Quantidade, REGENCY SUGAR BOWL GREEN))	0.01	0.01	0.01	0.75	56.59	0.01	3.96
530	((Quantidade, REGENCY TEA PLATE ROSES))	((Quantidade, REGENCY TEA PLATE GREEN))	0.02	0.01	0.01	0.69	50.76	0.01	3.16
531	((Quantidade, REGENCY TEA PLATE GREEN))	((Quantidade, REGENCY TEA PLATE ROSES))	0.01	0.02	0.01	0.84	50.76	0.01	6.20
505	((Quantidade, POPPY'S PLAYHOUSE LIVINGROOM))	((Quantidade, POPPY'S PLAYHOUSE BEDROOM))	0.01	0.02	0.01	0.81	48.00	0.01	5.21
504	((Quantidade, POPPY'S PLAYHOUSE BEDROOM))	((Quantidade, POPPY'S PLAYHOUSE LIVINGROOM))	0.02	0.01	0.01	0.64	48.00	0.01	2.78
561	((Quantidade, SET/6 RED SPOTTY PAPER PLATES))	((Quantidade, SET/6 RED SPOTTY PAPER CUPS))	0.02	0.02	0.01	0.72	47.34	0.01	3.58
560	((Quantidade, SET/6 RED SPOTTY PAPER CUPS))	((Quantidade, SET/6 RED SPOTTY PAPER PLATES))	0.02	0.02	0.01	0.82	47.34	0.01	5.57
506	((Quantidade, POPPY'S PLAYHOUSE KITCHEN))	((Quantidade, POPPY'S PLAYHOUSE LIVINGROOM))	0.02	0.01	0.01	0.62	46.07	0.01	2.59
507	((Quantidade, POPPY'S PLAYHOUSE LIVINGROOM))	((Quantidade, POPPY'S PLAYHOUSE KITCHEN))	0.01	0.02	0.01	0.85	46.07	0.01	6.65

Figura 32 – Testes de recomendação.

```
[ ] # Testes para fazer e obter recomendacoes

# ('Quantity', 'REGENCY SUGAR BOWL GREEN')
# ('Quantity', 'REGENCY MILK JUG PINK')
# ('Quantity', 'REGENCY TEA PLATE ROSES')
# ('Quantity', 'REGENCY TEA PLATE GREEN')

recomencacoes = []

for i, produto in regras_ordenadas["antecedents"].items():
    for j in list(produto):
        if j == ('Quantity', 'REGENCY MILK JUG PINK'):
            recomendacoes.append(list(regras_ordenadas.iloc[i]["consequents"]))

[ ] recomendacoes

[ ] [['Quantity', 'LUNCH BAG RED RETROSPOT'],
      ['Quantity', 'JUMBO BAG RED RETROSPOT']]
```

6. Apresentação dos Resultados

Nesta seção, apresentamos os resultados da análise e dos modelos aplicados, refletindo sobre as descobertas significativas e sua relevância.

6.1 Avaliação da Hipótese 1

Ao examinar a primeira hipótese, a previsão do valor total de vendas com um modelo de regressão, observamos que o modelo apresentou um R^2 score de $-1.94e+20$, o que indica um grave problema de underfitting. Isso sugere que as variáveis selecionadas não possuem correlação suficiente para explicar a variabilidade dos dados, levando à conclusão de que o sub-dataset gerado não foi eficaz. Essa falha nos levou a descartar a abordagem de regressão e a buscar novas alternativas.

6.2 Análise por Associação com Apriori

Dando continuidade à nossa investigação, aplicamos o algoritmo Apriori para realizar uma análise por associação, aproveitando os registros de compra que envolvem múltiplos produtos por cliente. O objetivo foi identificar quais itens tendem a ser comprados juntos.

Os resultados das regras de associação revelaram as seguintes informações valiosas:

Figura 33 – Melhores resultados dos produtos.

Antecedentes	Consequentes	Confiança	Lift
JUMBO BAG RED RETROSPOT	JUMBO BAG PINK POLKADOT	34%	7.22
ALARM CLOCK BAKELIKE GREEN	ALARM CLOCK BAKELIKE RED	67%	14.25
LUNCH BAG RED RETROSPOT	LUNCH BAG PINK POLKADOT	40%	8.00
REGENCY SUGAR BOWL GREEN	REGENCY MILK JUG PINK	76%	56.59
POPPY'S PLAYHOUSE LIVINGROOM	POPPY'S PLAYHOUSE BEDROOM	81%	48.00

Essas regras demonstram que a maioria dos itens com altos valores de confiança ($> 70\%$) e lift são produtos que podem ser estrategicamente agrupados em promoções. Por exemplo, a associação entre "JUMBO BAG RED RETROSPOT" e "JUMBO BAG PINK POLKADOT" poderia ser utilizada para oferecer um desconto na compra conjunta, incentivando o cliente a adicionar ambos os itens ao carrinho.

6.3 Implicações para o Negócio

Os insights obtidos a partir da análise por associação oferecem uma oportunidade significativa para melhorar as estratégias de marketing e vendas. A identificação de itens que costumam ser comprados juntos permite a implementação de recomendações personalizadas e promoções direcionadas. Isso pode não apenas aumentar a taxa de conversão, mas também melhorar a experiência do cliente ao facilitar escolhas relevantes durante o processo de compra.

Essas descobertas contribuem para a nossa missão inicial de entender melhor o comportamento do consumidor e maximizar as vendas através de insights orientados por dados.

Data Science Workflow Canvas*

Start here. The sections below are ordered intentionally to make you state your goals first, followed by steps to achieve those goals. You're allowed to switch orders of these steps!

Title:

1 Problem Statement What problem are you trying to solve? What larger issues do the problem address? Identificar padrões de compra em dados de vendas e desenvolver recomendações de produtos.	2 Outcomes/Predictions What prediction(s) are you trying to make? Identify applicable predictor (x) and/or target (y) variables. Prever vendas futuras e determinar associações entre produtos para impulsionar as vendas.	3 Data Acquisition Where are you sourcing your data from? Is there enough data? Can you work with it? Coletar dados de vendas e registros de compras, incluindo informações de clientes.
4 Modeling What models are appropriate to use given your outcomes? Aplicar modelos de regressão, K-Means e Apriori para análise e recomendações.	5 Model Evaluation How can you evaluate your model's performance? Avaliar modelos utilizando métricas como R ² Score, suporte e confiança.	6 Data Preparation What do you need to do to your data in order to run your model and achieve your outcomes? Limpar e transformar os dados, criando sub-datasets relevantes para a análise.

✓ Activation

When you finish filling out the canvas above, now you can begin implementing your data science workflow in roughly this order:

1 Problem Statement → 2 Data Acquisition → 3 Data Prep → 4 Modeling → 5 Outcomes/Preds → 6 Model Eval

* Note: This canvas is intended to be used as a starting point for your data science projects. Data science workflows are typically nonlinear.

7. Links

<https://github.com/JRafaQuadros91/TCC-2024>

<https://www.youtube.com/watch?v= ZrEfSkO3IQ>