Data Science Bootcamp - WoMakersCode

Análise de consumidor em e-commerce utilizando um conjunto de dados público da Olist Store

Desafio 4 - Kaggle #1 - Brazilian E-Commerce Public Dataset by Olist

Grupo 1 - Atena:

MURTINHO, A. B.[1]; LEITE, M. F.[2]; MONACHESI, P. P[3].; PINHEIRO, T[4].

RESUMO: Este trabalho documenta a coleta, transformação, análise e modelagem de um conjunto de dados de e-commerce por meio de práticas de Ciência de Dados, como o tratamento estatístico por linhas de código em Python e a clusterização. O RFV - Recência, Frequência e Valor, método de segmentação comumente utilizado na análise de consumidores, foi também adotado para nortear o entendimento do comportamento desses clientes no período estudado, levando à conclusão de que um planejamento de marketing mais assertivo e estratégico, considerando-se um particionamento maior dos grupos, poderá surtir efeito positivo na conversão e manutenção desse público.

Palavras-chave: *Análise de Dados*; *Tratamento Estatístico*; *Data Science*; *Comércio Eletrônico*.

Introdução

O presente trabalho documenta a análise de dados realizada por estudantes do Grupo 1, denominado Atena, para a apresentação do *Hack Day* promovido pela *WoMakersCode* como trabalho final decorrente do *bootcamp* (curso intensivo) de

[1] Amanda de Britto Murtinho, e-mail: ab.murtinho@hotmail.com, LinkedIn: https://www.linkedin.com/in/amandareznor/.

[2] Michele Francisca Leite, e-mail: michele.fl18@gmail.com, LinkedIn: https://www.linkedin.com/in/michele-leite/.

[3] Paola Picolo Monachesi, e-mail: paolapicolomonachesi@gmail.com, LinkedIn: https://www.linkedin.com/in/paola-monachesi/.

[4] Taís Pinheiro, e-mail: tais141@gmail.com, LinkedIn: https://www.linkedin.com/in/taispin/.

Data Science (Clência de Dados) ministrado entre 18 de outubro e 17 de dezembro de 2019 em laboratórios da Universidade São Judas Tadeu - campus da Avenida Angélica, próximo à estação Paulista de metrô.

A análise visou o tratamento e a inferência estatística de um *dataset* (conjunto de dados) disponibilizado pela *Olist Store* (uma empresa de tecnologia SaaS - *Software as a Service* - Olist, 2019), e que configura-se como um dos desafios com dados publicados pela plataforma Kaggle (2019).

Esse dataset é composto por um conjunto de oito tabelas com diferentes informações relacionadas a ordens de venda geradas por diferentes comércios eletrônicos (dentre eles as Lojas Americanas, Submarino, entre outras), incluindo data de compra, tipo de produto, prazo de entrega, análise do consumidor, origem das solicitações, dentre outras variáveis.

Dentro disso, questionou-se a possível relação entre essas oito tabelas com o objetivo principal de gerar uma análise do perfil dos consumidores, exercitando as diversas ferramentas e técnicas que foram expostas durante as aulas do bootcamp como objetivo secundário.

Metodologia

Para dar início à análise, inicialmente os dados foram coletados em formato “csv.” (dados tabulados e divididos por vírgula, conforme disponibilizados pela plataforma Kaggle) e tratados por meio do programa Microsoft Power BI, que conta com uma interface e ferramentas voltadas para *business intelligence* (inteligência analítica - Microsoft, 2019).

Em sequência, os dados foram exportados para um Jupyter Notebook (aplicação que permite criar e compartilhar documentos com linhas executáveis de códigos, visualizações e textos, facilitando a modelagem estatística e a visualização de dados, dentre outras funcionalidades - Project Jupyter, 2019). A linguagem de programação adotada foi Python - versão 3 - e diversas bibliotecas correlacionadas, como Pandas (2019), NumPy (2019) e Seaborn (WASKOM, 2018).

Optou-se pelo método de segmentação para proceder à análise do perfil de consumidores, uma vez que essa prática é amplamente utilizada na área de marketing e *business analytics*, sendo conhecida pela sigla RFV, de “Recência, Frequência e Valor” (GRIGSBY, 2016).

Após importar as bibliotecas do *NumPy*, *Pandas* e *datetime*, procedeu-se ao tratamento dos dados, em *Python*, e à análise de dados pelo método RFV, sendo que a biblioteca do *Seaborn* permitiu plotar um gráfico inicial a partir dos resultados alcançados. Para melhor efeito de visualização, os dados foram exportados em ".csv" para o programa Microsoft Power BI, onde foi gerado o gráfico com o resultado final da análise, e o corpo do código foi compartilhado no GitHub - plataforma de desenvolvimento que hospeda códigos-fonte com controle de versão dos envios (GITHUB, 2019).

Desenvolvimento

A Ciência de Dados é uma área dedicada ao estudo e à análise de conjuntos de dados, de modo a extrair informações relevantes que auxiliem na tomada estratégica de decisões ou nos resultados de pesquisa dos mais diversos campos de atuação. A coleta, o tratamento e a transformação seriam etapas anteriores à análise dos dados em si, envolvendo conhecimentos de computação, matemática, estatística e de negócios para guiar o processo de interpretação dos dados (GODOI, 2017).

Além disso, a visualização da análise é uma etapa posterior à análise e também muito importante, uma vez que permitirá comunicar os resultados com maior (ou menor) eficiência. Dentro disso, há uma enorme diversidade de métodos e ferramentas que podem ser empregados nas diferentes etapas da análise, sendo que a linguagem *Python* possui muitas bibliotecas associadas que são específicas ou mesmo excelentes opções para uma análise, permitindo desde o tratamento até a transformação e visualização dos dados (MCKINNEY, 2018).

A partir da matemática e da estatística, abre-se um leque de possibilidades de tratamento e interpretação de uma base de dados, abarcando desde aferições básicas (como média, moda e mediana) até análises de dispersão, correlação, causalidade, probabilidades e distribuições que permitem criar inferências e testar hipóteses (GRUS, 2016).

A segmentação ou agrupamento entra como um dos muitos métodos possíveis de análise (em inglês também conhecida como *clustering*, ou clusterização), sendo que esse particionamento serve como base para processos de taxonomia ou para encontrar uma divisão mais homogênea dentre os elementos de um grupo (GRIGSBY, 2016).

Ao clusterizar uma base de dados (ou seja, ao dividir um conjunto em subconjuntos com elementos mais homogêneos entre si), pode-se aplicar desde algoritmos

estatísticos mais avançados até regras simples de agrupamento (como uma separação entre os extremos), o que permite criar ou enxergas diferentes categorias (taxonomias) dentro de um *dataset*, de modo que muitas empresas recorrem a esse tipo de análise em seus negócios (GRIGSBY, 2016).

O RFV é um formato simples de clusterização que permite separar e agrupar variáveis relacionadas aos consumidores, sendo elas:

a) Recência: baseada na data da última compra;
b) Frequência: quantas vezes um consumidor comprou em um determinado período; e
c) Valor: quanto o consumidor costuma investir em suas compras.

A partir do RFV, assim, é possível traçar diferentes estratégias para interpretar o perfil do consumidor, com a identificação de perfis similares e um planejamento de marketing mais adequado, para tratar cada segmento de forma diferenciada.

Como o *dataset* fornecido pela *Olist Store* fornece todos os dados necessários para a melhor compreensão do perfil dos consumidores, a clusterização e, mais especificamente, o RFV foram os métodos escolhidos para a análise desses dados de e-commerce neste trabalho.

Discussão e resultados

Após a coleta dos dados da *Olist Store* na plataforma *Kaggle*, as tabelas foram previamente exploradas pelo Microsoft Power BI e, dentre as oito tabelas disponíveis, foram selecionadas as que traziam informações sobre pedidos, itens e clientes, uma vez que elas continham as colunas necessárias para a análise RFV.

Em seguida, essas três tabelas ("olist_orders_dataset".csv, "olist_order_items_dataset.csv" e "olist_customers_dataset.csv") foram importadas para um Jupyter Notebook e tratadas via código, usando a linguagem Python em versão 3, conforme demonstrado na Figura 1.

Figura 1: Importando as tabelas selecionadas para tratamento

```
import pandas as pd
import numpy as np
import datetime
from datetime import datetime, timedelta
import seaborn as sns
import matplotlib.pyplot as plt
```

```
# Utilizando apenas três datasets, pois são os que contêm dados importantes e necessários para o cálculo RFV.

pedidos  = pd.read_csv('https://raw.githubusercontent.com/paolamonachesi/teste/master/olist_orders_dataset.csv')
itens_pedido  = pd.read_csv('https://raw.githubusercontent.com/paolamonachesi/teste/master/olist_order_items_dataset
clientes  = pd.read_csv('https://raw.githubusercontent.com/paolamonachesi/teste/master/olist_customers_dataset.csv')
```

Fonte: autoras.

Em seguida, visualizou-se o cabeçalho das tabelas para verificar o relacionamento entre elas (essa etapa também havia sido anteriormente visualizada pelo programa Microsoft Power BI). Essa visualização é possível por meio do comando "nome_da_tabela.head()", conforme se nota pela Figura 2.

Figura 2: Conferindo o cabeçalho de uma tabela

```
pedidos.head()
```

	order_id	customer_id	order_status	order_purchase_timestamp
0	e481f51cbdc54678b7cc49136f2d6af7	9ef432eb6251297304e76186b10a928d	delivered	2017-10-02 10:56:33
1	53cdb2fc8bc7dce0b6741e2150273451	b0830fb4747a6c6d20dea0b8c802d7ef	delivered	2018-07-24 20:41:37
2	47770eb9100c2d0c44946d9cf07ec65d	41ce2a54c0b03bf3443c3d931a367089	delivered	2018-08-08 08:38:49
3	949d5b44dbf5de918fe9c16f97b45f8a	f88197465ea7920adcdbec7375364d82	delivered	2017-11-18 19:28:06
4	ad21c59c0840e6cb83a9ceb5573f8159	8ab97904e6daea8866dbdbc4fb7aad2c	delivered	2018-02-13 21:18:39

Fonte: autoras.

Uma vez identificados os relacionamentos entre as tabelas - que seriam suas colunas identificadoras (coluna "customer_id" e "order_id"), por meio do comando “merge” essas três tabelas foram unificadas em uma única tabela, respeitando o relacionamento existente ao indicar suas colunas identificadoras, o que gerou a tabela “df_olist”, apresentada na Figura 3.

Figura 3: Tabela unificada com os dados de pedidos, itens e consumidores

```
# Conectando as tabelas para que podemos formar um dataframe único.

clientes_pedidos = pd.merge(pedidos, clientes, left_on='customer_id', right_on='customer_id', how='inner')

df_olist = pd.merge(clientes_pedidos, itens_pedido, left_on='order_id', right_on='order_id', how='inner')
```

```
df_olist.head()
```

	order_id	customer_id	order_status	order_purchase_timestamp
0	e481f51cbdc54678b7cc49136f2d6af7	9ef432eb6251297304e76186b10a928d	delivered	2017-10-02 10:56:33
1	53cdb2fc8bc7dce0b6741e2150273451	b0830fb4747a6c6d20dea0b8c802d7ef	delivered	2018-07-24 20:41:37
2	47770eb9100c2d0c44946d9cf07ec65d	41ce2a54c0b03bf3443c3d931a367089	delivered	2018-08-08 08:38:49
3	949d5b44dbf5de918fe9c16f97b45f8a	f88197465ea7920adcdbec7375364d82	delivered	2017-11-18 19:28:06
4	ad21c59c0840e6cb83a9ceb5573f8159	8ab97904e6daea8866dbdbc4fb7aad2c	delivered	2018-02-13 21:18:39

Fonte: autoras.

A partir dessa tabela unificada, foi possível tratar os dados para identificar quais valores seriam mais importantes para a análise, o que foi feito filtrando-se valores únicos existentes, o que retornou os dados apresentados na Figura 4.

Figura 4: Valores únicos identificados a partir da tabela unificada

```
# Contagem de valores únicos
def count_unique(df):
  print("Quantidade de valores únicos para cada feature no conjunto de treinamento")
  for i in df.columns:
    print(f"{i}: {df[i].nunique()}")
```

```
count_unique(df_olist)
```

```
Quantidade de valores únicos para cada feature no conjunto de treinamento
order_id: 98666
customer_id: 98666
order_status: 7
order_purchase_timestamp: 98112
order_approved_at: 90174
order_delivered_carrier_date: 81017
order_delivered_customer_date: 95664
order_estimated_delivery_date: 450
customer_unique_id: 95420
customer_zip_code_prefix: 14976
customer_city: 4110
customer_state: 27
order_item_id: 21
product_id: 32951
seller_id: 3095
shipping_limit_date: 93318
price: 5968
freight_value: 6999
```

Fonte: autoras.

Essa contagem permitiu filtrar os dados que seriam mais importantes para dar prosseguimento à análise. Assim sendo, as colunas com identificação do consumidor, da compra, da data de compra, da quantidade de itens e do preço dos itens ("customer_unique_id", "customer_id", "order_purchase_timestamp", "order_item_id" e "price") seriam as únicas necessárias para a análise RFV, de modo que as demais colunas foram desconsideradas nas etapas seguintes. As colunas selecionadas foram então tratados para verificar a existência de nulos (dados ausentes) e, a partir disso, iniciou-se o cálculo da recência, que é o primeiro elemento da análise RFV (Recência, Frequência e Valor).

Para encontrar a Recência, transformou-se a coluna de data de compra, que estava no formato de "timestamp", em uma coluna com o formato "datetime", de modo a permitir operações de cálculo usando os dados dessa coluna. Já para criar o elemento Valor, multiplicou-se a coluna de itens ("order_item_id") pela coluna de preço ("price"), gerando a coluna "total_pedido", sendo que essas colunas de preço e de itens foram então excluídas da tabela unificada, deixando apenas a coluna com os valores desejados ("total_pedido"). O trecho do código utilizado para essas transformações pode ser conferido na Figura 5.

Figura 5: Início da análise RFV

```
# Para calcular a Recência, é necessário transformar a coluna da data de compra de 'timestamp' para 'datetime'

df_olist['order_purchase_timestamp'] = pd.to_datetime(df_olist['order_purchase_timestamp'])

df_olist['order_purchase_timestamp'].head(3)

# Para calcular o Valor, é necessário multiplicar o valor dos itens pela quantidade.

df_olist['total_pedido'] = df_olist['order_item_id'] * df_olist['price']

df_olist.head()

# Já que temos o total dos pedidos, podemos excluir as colunas com a quantidade de itens e preço.

df_olist = df_olist.drop(['order_item_id', 'price'], axis=1)
```

Fonte: autoras.

A função *timedelta* foi utilizada em sequência, de modo a calcular a diferença de dias e chegar ao valor da Recência (que é a data da última compra realizada no período analisado). Em posse desse valor, foi gerada uma tabela (*dataframe*) com os valores desejados, renomeando a coluna para efeito de melhor compreensão, como explicitado na Figura 6.

Figura 6: Criando um *dataframe* com os valores de RFV

```
# Renomeando as colunas do dataframe para facilitar entendimento

df_olist.rename(columns = {'order_purchase_timestamp': 'Recência',
                           'customer_id': 'Frequência', 'total_pedido': 'Valor'}, inplace=True)
```

```
df_olist.head(2)
```

customer_unique_id	Recência	Frequência	Valor
0000366f3b9a7992bf8c76cfdf3221e2	116	1	129.9
0000b849f77a49e4a4ce2b2a4ca5be3f	119	1	18.9

Fonte: autoras.

O próximo passo foi realizar a distribuição em quantis ou quartis (*quartiles*) para o cálculo RFV de cada cliente. O quantil serve como medida para definir a posição em que um conjunto de dados será amostrado, ou seja, é um parâmetro para a segmentação dos dados de acordo com a distribuição probabilística dentro de um percentual. A Figura 7 mostra como isso foi feito dentro da execução do código.

Figura 7: Segmentação dos dados por meio de quartis

```
# Recência
# Utilizando a função qcut do pandas que permite realizar a distribuição em quantis.
recencia_labels = range(4, 0, -1)
recencia_quartiles = pd.qcut(df_olist['Recência'], 4, labels = recencia_labels)
df_olist = df_olist.assign(R = recencia_quartiles.values)

# Frequência
# Utilizando a função cut que permite segmentar e classificar dados em posições
df_olist['F'] = pd.cut(df_olist['Frequência'], [0, 1, 2, df_olist['Frequência'].max()], labels=[1, 2, 3])

# Valor
valor_labels = range(1, 4)
valor_quartiles = pd.qcut(df_olist['Valor'], 3, labels = valor_labels)
df_olist = df_olist.assign(V = valor_quartiles.values)
```

Fonte: autoras.

O último passo da análise foi a classificação dos segmentos por meio de pesos (ou *scores*) que identificam os consumidores em quatro possíveis categorias:

a) *Premium* (score de 9 a 10): apresentam Recência baixa (ou seja, com menor número de dias desde a última compra), maior frequência de pedidos e maior valor investido.
b) *Master* (score de 6 a 8): com Recência média, frequência de pedidos e valor investido médios.

c) *Business* (score de 4 a 5): com Recência alta (maior número de dias desde a última compra), frequência de pedidos baixa e baixo valor de investimento; e
d) *Inativos* (score de até 4): grupo com menor número de compras e demais atividades.

O notebook com o código completo foi compartilhado no GitHub para futuros estudos, e os dados obtidos por meio da divisão com pesos (*scores*) podem ser observado na Tabela 1.

Tabela 1: Classificação dos consumidores por RFV

customer_unique_id	Recência	Frequência	Valor	R	F	V	Classificação
0000366f3b9a7992bf8c76cfdf3221e2	116	1	129.90	4	1	2	Master
0000b849f77a49e4a4ce2b2a4ca5be3f	119	1	18.90	4	1	1	Master
0000f46a3911fa3c0805444483337064	542	1	69.00	1	1	2	Business
0000f6ccb0745a6a4b88665a16c9f078	326	1	25.99	2	1	1	Business
0004aac84e0df4da2b147fca70cf8255	293	1	180.00	2	1	3	Master

Fonte: autoras.

Ainda foi realizado agrupamento por meio de categorização, explorando a média de cada variável e a contagem de registros, o que gerou a tabela exibida na Figura 8.

Figura 8: Distribuição da média RFV para cada categoria

```
# Agrupando por categorização, com a média de cada variável e a contagem de registros.

df_olist_class = df_olist.groupby(['Classificação']).agg({ 'Recência': 'mean', 'Frequência': 'mean', 'Valor':
                                                           ['mean','count'] }).round(1)
```

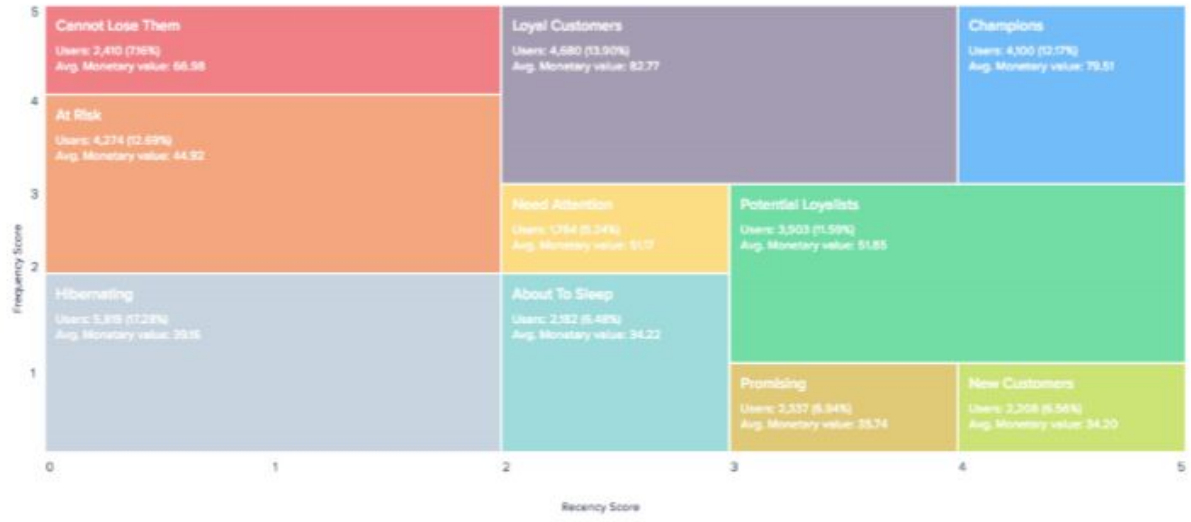
```
df_olist_class
```

	Recência	Frequência	Valor	
	mean	mean	mean	count
Classificação				
Business	329.7	1.0	118.5	39133
Inativo	458.8	1.0	34.9	7875
Master	139.9	1.0	213.5	47767
Premium	72.9	2.3	445.2	645

Fonte: autoras.

Com os dados assim obtidos, foi gerado um gráfico do tipo RFV dentro do Microsoft Power BI. O *RFM grid*, ou gráfico de RFV, tem uma estrutura baseada em quadrantes (áreas quadrangulares), conforme o padrão exibido na Figura 9.

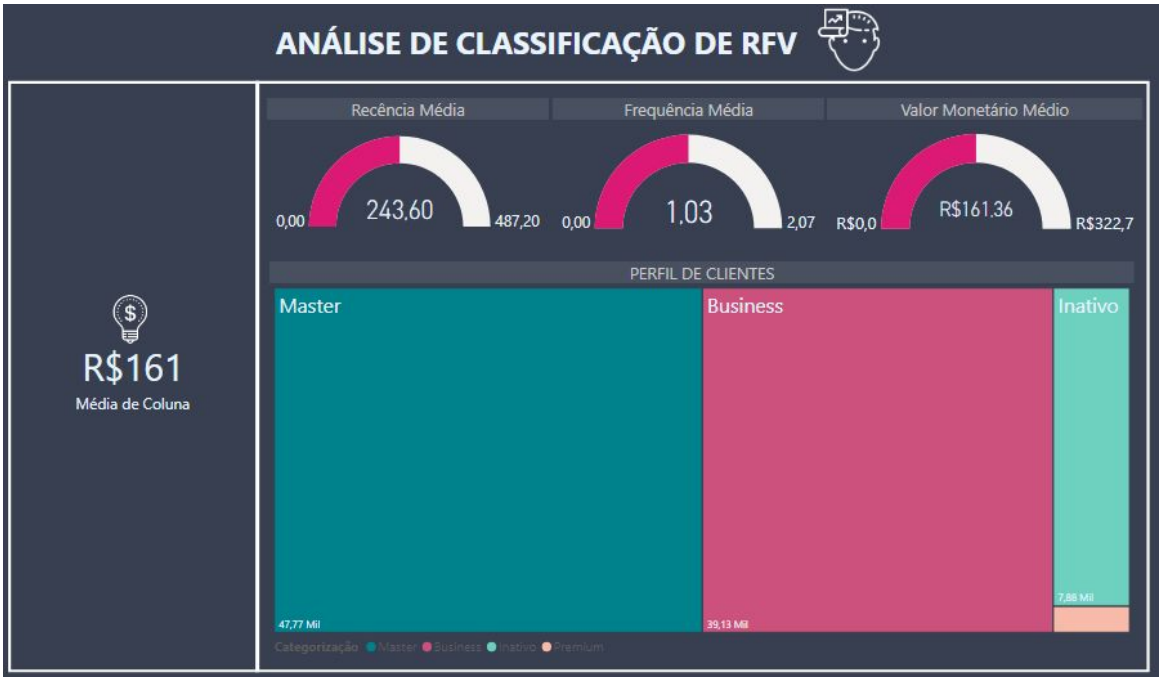
Figura 9: Exemplo de um gráfico para visualização de análise RFV



Fonte: CleverTap, 2013.

Esse padrão de gráfico exibe os usuários mais ativos no quadrante superior direito, enquanto que os usuários mais inativos ficam no quadrante inferior esquerdo (CleverTap, 2013). A visualização dos dados da análise da *Olist* transposta com base nesse formato de gráfico RFV é exibida no Gráfico 1.

Gráfico 1: Resultado da análise em um gráfico RFV



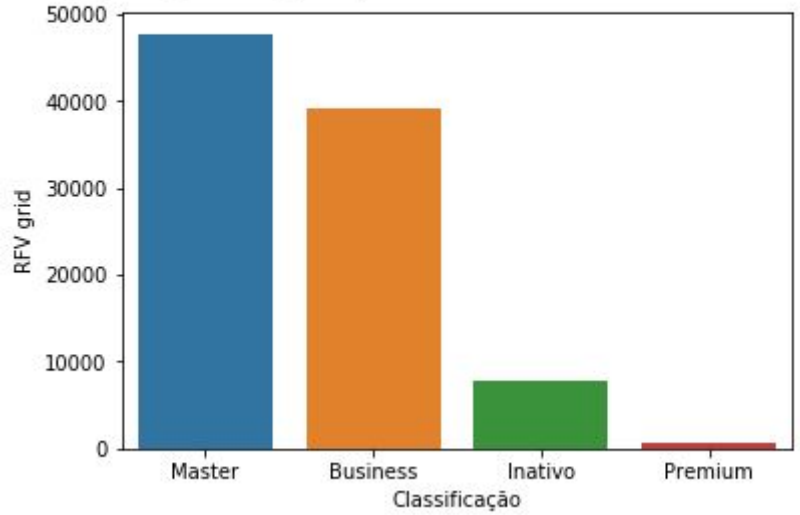
Fonte: autoras.

O gráfico em formato de barras também foi plotado para visualização por meio de código, conforme exibido na Figura 10.

Figura 10:

```
sns.countplot(df_olist['Classificação'])
plt.ylabel('RFV grid')
```

```
Text(0, 0.5, 'RFV grid')
```



Fonte: autoras.

De acordo com o guia de marketing com base em RFV elaborado pela CleverTap (2013), há três possíveis questionamentos para definir uma estratégia para cada perfil de consumidor, a citar:

a) Consumidores mais ativos: como manter o engajamento do consumidor?
b) Usuários de média atividade: como encorajar novas comprar e conversões?
c) Perfis inativos: como garantir que essas conversões não sejam perdidas?

Tais questionamentos induzem a uma série de possíveis estratégias para manter, envolver ou conquistar um consumidor novo ou antigo. A partir do resultado da análise, apontado no Gráfico 1 e na Figura 10, é possível interpretar que há uma escala crescente de consumidores entre “inativos” e “master”, com uma concentração maior entre os perfis “business” e “master” e uma concentração menor entre “inativos” e “premium”.

Considerações finais

A análise dos resultados permite chegar à interpretação de que a maioria dos consumidores apresenta uma tendência ao engajamento e ao aumento da atividade

de compra, com tendência à conversão caso sejam adotadas as estratégias mais adequadas para cada segmento.

O número pequeno de usuários na categoria "premium" e "inativo" e a grande faixa de usuários com atividade entre média e alta ("business" e "master") pode indicar, também, que é possível que já existam estratégias de marketing ativas para esses consumidores, mas que essas estratégias precisariam ser realinhadas de modo a melhor atender cada segmento.

Ademais, um marketing mais assertivo, baseado em uma segmentação maior para subdividir os perfis apresentados, pode ajudar a alcançar resultados melhores de conversão, uma vez que existem estratégias que conseguem abarcar maiores especificidades de alcance e engajamento de público.

Referências

CLEVERTAP. ***A Quick-Start Guide to Automated Segmentation with RFM Analysis***. CleverTap, Whitepaper. US, 2013. Disponível em: <https://clevertap.com/l/wp-rfm-quick-start-guide/>. Acesso em: 06 dez. 2019.

GITHUB. ***GitHub is how people build software***. GitHub, About. São Francisco, 2019. Disponível em: <https://github.com/about>. Acesso em: 06 dez. 2019.

GODOI, D. **Data Science:** o que é, conceito e definição. Cetax, Artigos de Data Science, 2017. Disponível em: <https://www.cetax.com.br/blog/data-science/>. Acesso em: 6 dez. 2019.

GRIGSBY, M. ***Advanced Customer Analytics:*** Targeting, Valuing, Segmenting and Loyalty. Londres: Kogan Page, 2016.

GRUS, J. **Data Science do Zero:** primeiras regras como Python. Rio de Janeiro: Alta Books, 2016.

KAGGLE. ***Brazilian E-Commerce Public Dataset by Olist***. Kaggle, Datasets. São Francisco, 2019. Disponível em: <https://www.kaggle.com/olistbr/brazilian-ecommerce/#olist_products_dataset.csv>. Acesso em: 6 dez. 2019.

MCKINNEY, W. ***Python for Data Analysis****: Data Wrangling with Pandas, NumPy, and IPython*. California: O'Reilly, 2018.

MICROSOFT. ***Turn data into opportunity.*** Microsoft, Power BI. Washington, 2019. Disponível em: <https://powerbi.microsoft.com/pt-br/>. Acesso em: 6 dez. 2019.

NUMPY. ***The NumPy community***. NumPy developers, Community, 2019. Disponível em: <https://numpy.org/community.html>. Acesso em: 6 dez. 2019.

OLIST. **Dúvidas Frequentes**. Olist, FAQ. Curitiba, 2019. Disponível em: <https://olist.com/faq/>. Acesso em: 6 dez. 2019.

PANDAS. ***The pandas project***. Pandas, About, nov. 2019. Disponível em: <https://pandas.pydata.org/about.html>. Acesso em: 6 dez. 2019.

PROJECT JUPYTER. ***About us***. Jupyter, About. U.S., 14 nov. 2019. Disponível em: <https://jupyter.org/about>. Acesso em: 6 dez. 2019.

WASKOM, M. ***Seaborn:*** *statistical data visualization*. Seaborn, 2012-2018. Disponível em: <https://seaborn.pydata.org/>. Acesso em: 6 dez. 2019.