Link apresentacao: https://youtu.be/_yJE3Zj4OKE

# PROJETO APLICADO I


Descobrindo Padrões e Insights em compras Olist store

Enzo Vemado

# Objetivo da Análise

01 Encontrar padrões

02 Definir valores de RFM

03 Encontrar o Churn dos clientes

# Metodologia

Ferramentas utilizadas:

1. Python
2. Visual Studio Code
3. Jupyter

# Análise Sumarizada

- **order_id** (O indicador único do pedido);
- **customer_id** (O indicador de qual cliente);
- **order_status** (O indicador se o pedido tinha sido finalizado ou não);
- **order_purchase_timestamp;**
- **order_approved_at** (Quanto o pedido foi aprovado);
- **order_delivered_carrier_date** (Quando o pedido foi despachado);
- **order_delivered_customer_date** (Quando o pedido foi entregue);
- **order_estimated_delivery_date** (Previsão de entrega);
- **payment_sequential** (Quantidade de tipo de pagamento);
- **payment_type** (Tipo de pagamento);
- **payment_installments** (Quantidade de parcelas);
- **payment_value** (Pagamento total por transação);
- **price** (Valor da venda);
- **freight_value** (Valor do frete).

# Visualização dos datasets

Etapa onde os datasets são visualizados e analisados para verificar a qualidade e determinar se as informações coletadas estão bem distribuídas, identificar problemas e fazer as relações e ajustes necessários.

| | dataset | n_rows | n_cols | null_amount | qtd_null_columns | columns_name | null_columns |
|---|---|---|---|---|---|---|---|
| 0 | df_customers | 99441 | 5 | 0 | 0 | customer_id, customer_unique_id, customer_zip_code_prefix, customer_city, customer_state | |
| 1 | df_geolocation | 1000163 | 5 | 0 | 0 | geolocation_zip_code_prefix, geolocation_lat, geolocation_lng, geolocation_city, geolocation_state | |
| 2 | df_orders | 99441 | 8 | 4908 | 3 | order_id, customer_id, order_status, order_purchase_timestamp, order_approved_at, order_delivered_carrier_date, order_delivered_customer_date, order_estimated_delivery_date | order_approved_at, order_delivered_carrier_date, order_delivered_customer_date |
| 3 | df_items | 112650 | 7 | 0 | 0 | order_id, order_item_id, product_id, seller_id, shipping_limit_date, price, freight_value | |
| 4 | df_payment | 103886 | 5 | 0 | 0 | order_id, payment_sequential, payment_type, payment_installments, payment_value | |
| 5 | df_reviews | 99224 | 7 | 145903 | 2 | review_id, order_id, review_score, review_comment_title, review_comment_message, review_creation_date, review_answer_timestamp | review_comment_title, review_comment_message |
| 6 | df_products | 32951 | 9 | 2448 | 8 | product_id, product_category_name, product_name_lenght, product_description_lenght, product_photos_qty, product_weight_g, product_length_cm, product_height_cm, product_width_cm | product_category_name, product_name_lenght, product_description_lenght, product_photos_qty, product_weight_g, product_length_cm, product_height_cm, product_width_cm |
| 7 | df_sellers | 3095 | 4 | 0 | 0 | seller_id, seller_zip_code_prefix, seller_city, seller_state | |

# Verificações

- Transações são distintas
- Intervalo da base
- Status das transações
- Valores nulos

```
#verificando se as transações são distintas
duplicateRowsOrders = df_orders[df_orders.duplicated()]
print("Número de transações: ",df_orders.order_id.nunique())
print("Transações duplicadas: ",duplicateRowsOrders.shape[0])
```

```
Número de transações:  99441
Transações duplicadas:  0
```

```
#verificado intervalo da base
print("Data da primeira transação: ",df_orders.order_purchase_timestamp.min())
print("Data da última transação: ",df_orders.order_purchase_timestamp.max())
```

```
Data da primeira transação:  2016-09-04 21:15:19
Data da última transação:  2018-10-17 17:30:18
```

```
#status das transações
status_order = df_orders.groupby('order_status')['order_id'].nunique().sort_values(ascending=False).reset_index()
status_order.columns = ['status','qtd_transações']
status_order
```

| | status | qtd_transações |
|---|---|---|
| 0 | delivered | 96478 |
| 1 | shipped | 1107 |
| 2 | canceled | 625 |

# Análises das transações

- clientes por transação
- produtos por transação
- pagamento por transação
- frete por transação
- transações onde os produtos não tem nome

```
#tabela com as transações e clientes relacionados e valores referentes

# clientes por transação
df_transactions = pd.merge(df_orders[['order_id','order_status','customer_id','order_purchase_timestamp']],df_customers,on='customer_id')

# produtos por transação
df_transactions = pd.merge(df_transactions,produtos_por_compra, on = 'order_id')

# pagamento por transação
df_transactions = pd.merge(df_transactions, paid, on='order_id')

# frete por transação
df_transactions = pd.merge(df_transactions,df_frete, how = 'left',on = 'order_id')


df_transactions.drop(['customer_zip_code_prefix','customer_city','customer_state','customer_id'],axis=1,inplace=True)
df_transactions
```

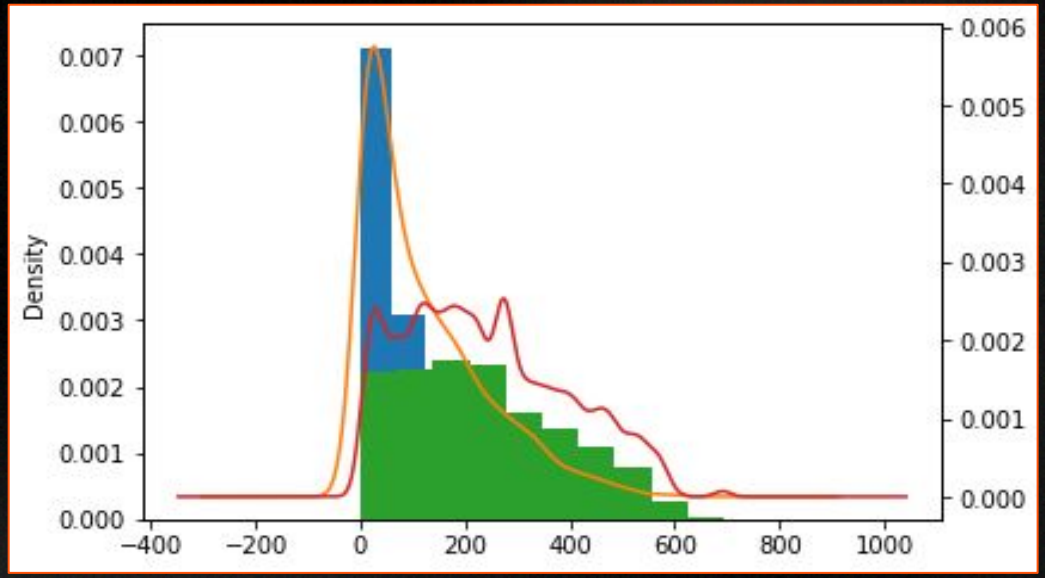
# RFM

O RFM foi calculado separadamente, tendo como base a frequência, recência e o monetário. Sendo utilizado como análise, não como predição.

```
global_recency = pd.DataFrame(df_transactions.groupby('customer_unique_id').order_purchase_timestamp.max())
global_recency.columns = ['ultima_compra']
global_recency

global_recency['recencia'] = global_recency['ultima_compra'].max() - global_recency['ultima_compra']
global_recency['recencia']= global_recency['recencia'].dt.days
global_recency

global_frequency=pd.DataFrame(df_transactions.groupby('customer_unique_id')['order_id'].count())
global_frequency.columns = ['qtd_transacoes']
global_frequency

#gasto total de cada cliente
global_monetary=pd.DataFrame(df_transactions[['customer_unique_id','receita_liquida']].groupby('customer_unique_id')['receita_liquida'].sum())
global_monetary

df_rfm=pd.merge(global_recency,global_frequency,on='customer_unique_id')
df_rfm=pd.merge(df_rfm,global_monetary,on='customer_unique_id')

df_rfm.drop(['ultima_compra'],axis=1,inplace=True)
df_rfm.columns=['rfm_recency','rfm_frequency','rfm_monetary']
df_rfm.reset_index()
```

# Churn

O Churn foi o método usado para determinar se o cliente ainda está vivo na base ou não, baseado no cruzamento da recência máxima de clientes recorrentes e da recência histórica da base completa.




```
#criterio do churn (if recency > 139)
#churn = 1, morto
#churn = 0, vivo
df_client['Churn'] = np.where(df_client['rfm_recency'] >= 139, 1, 0)
```

# Resultados Obtidos

- Recência
- Frequência
- Valor Monetário
- Churn

```
#criterio do churn (if recency > 139)
df_client['Churn'] = np.where(df_client['rfm_recency'] >= 139, 1, 0)
df_client
```

| | customer_unique_id | customer_zip_code_prefix | customer_city | customer_state | primeira_compra | ultima_compra | rfm_recency | rfm_frequency | rfm_monetary | ticket_medio | retencao | Churn |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 861eff4711a542e4b93843c6dd7febb0 | 14409 | franca | SP | 2017-05-16 15:05:35 | 2017-05-16 15:05:35 | 469 | 1 | 124.99 | 124.99 | 0 | 1 |
| 1 | 290c77bc529b7ac935b93aa66c333dc3 | 9790 | sao bernardo do campo | SP | 2018-01-12 20:48:24 | 2018-01-12 20:48:24 | 228 | 1 | 289.00 | 289.00 | 0 | 1 |
| 2 | 060e732b5b29e8181a18229c7b0b2b5e | 1151 | sao paulo | SP | 2018-05-19 16:07:45 | 2018-05-19 16:07:45 | 101 | 1 | 139.94 | 139.94 | 0 | 0 |
| 3 | 259dac757896d24d7702b9acbbff3f3c | 8775 | mogi das cruzes | SP | 2018-03-13 16:06:38 | 2018-03-13 16:06:38 | 168 | 1 | 149.94 | 149.94 | 0 | 1 |
| 4 | 345ecd01c38d18a9036ed96c73b8d066 | 13056 | campinas | SP | 2018-07-29 09:51:30 | 2018-07-29 09:51:30 | 31 | 1 | 230.00 | 230.00 | 0 | 0 |
| ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... |
| 93352 | 1a29b476fee25c95fbafc67c5ac95cf8 | 3937 | sao paulo | SP | 2018-04-07 15:48:17 | 2018-04-07 15:48:17 | 143 | 1 | 74.90 | 74.90 | 0 | 1 |
| 93353 | d52a67c98be1cf6a5c84435bd38d095d | 6764 | taboao da serra | SP | 2018-04-04 08:20:22 | 2018-04-04 08:20:22 | 147 | 1 | 114.90 | 114.90 | 0 | 1 |
| 93354 | e9f50caf99f032f0bf3c55141f019d99 | 60115 | fortaleza | CE | 2018-04-08 20:11:50 | 2018-04-08 20:11:50 | 142 | 1 | 37.00 | 37.00 | 0 | 1 |
| 93355 | 73c2643a0a458b49f58cea58833b192e | 92120 | canoas | RS | 2017-11-03 21:08:33 | 2017-11-03 21:08:33 | 298 | 1 | 689.00 | 689.00 | 0 | 1 |
| 93356 | 84732c5050c01db9b23e19ba39899398 | 6703 | cotia | SP | 2017-12-19 14:27:23 | 2017-12-19 14:27:23 | 253 | 1 | 13.99 | 13.99 | 0 | 1 |

93357 rows × 12 columns

Obrigado!