# Olist Order Analysis

## Marketing Department

Leveraging Power BI and Python

for valuable insights

Ricardo Schmid

Data Scientist

2024

# Table of Contents

## 1. Introduction

Olist is an e-commerce platform headquartered in Brazil. It operates as a marketplace connecting sellers offering a diverse range of products and customers looking for a convenient shopping experience across the entire country.

The primary objective of the project was to conduct a comprehensive analysis on Power BI while incorporating some Python codes. The dataset was obtained on Kaggle and represents Olist's 2016, 2017, and 2018 sales data. The focus was on generating valuable insights beneficial to the company, particularly for its marketing department.

## 2. Objectives

The project aimed to uncover patterns and trends that could guide marketing strategies. The specific objectives are as follows:

1. **Find the best time to advertise:** Utilize historical data to identify peak periods for advertising campaigns to optimize resource allocation and effectiveness of marketing campaigns.
2. **Segment customers for targeted campaigns:** Employ RFM (Recency, Frequency, Monetary) analysis to categorize customers into distinct groups allowing for targeted marketing campaigns directed to each customer segment to maximize engagement, retention, and profitability.

## 3. Data Model

Figure 1 below illustrates the data model constructed in Power BI. Certain columns from the original dataset were omitted during import since they were not intended for use and a Calendar dimension table was created to facilitate time-based analysis. Additionally, a new csv file named "State Region" was generated to facilitate the representation of each state's region in the visualizations.

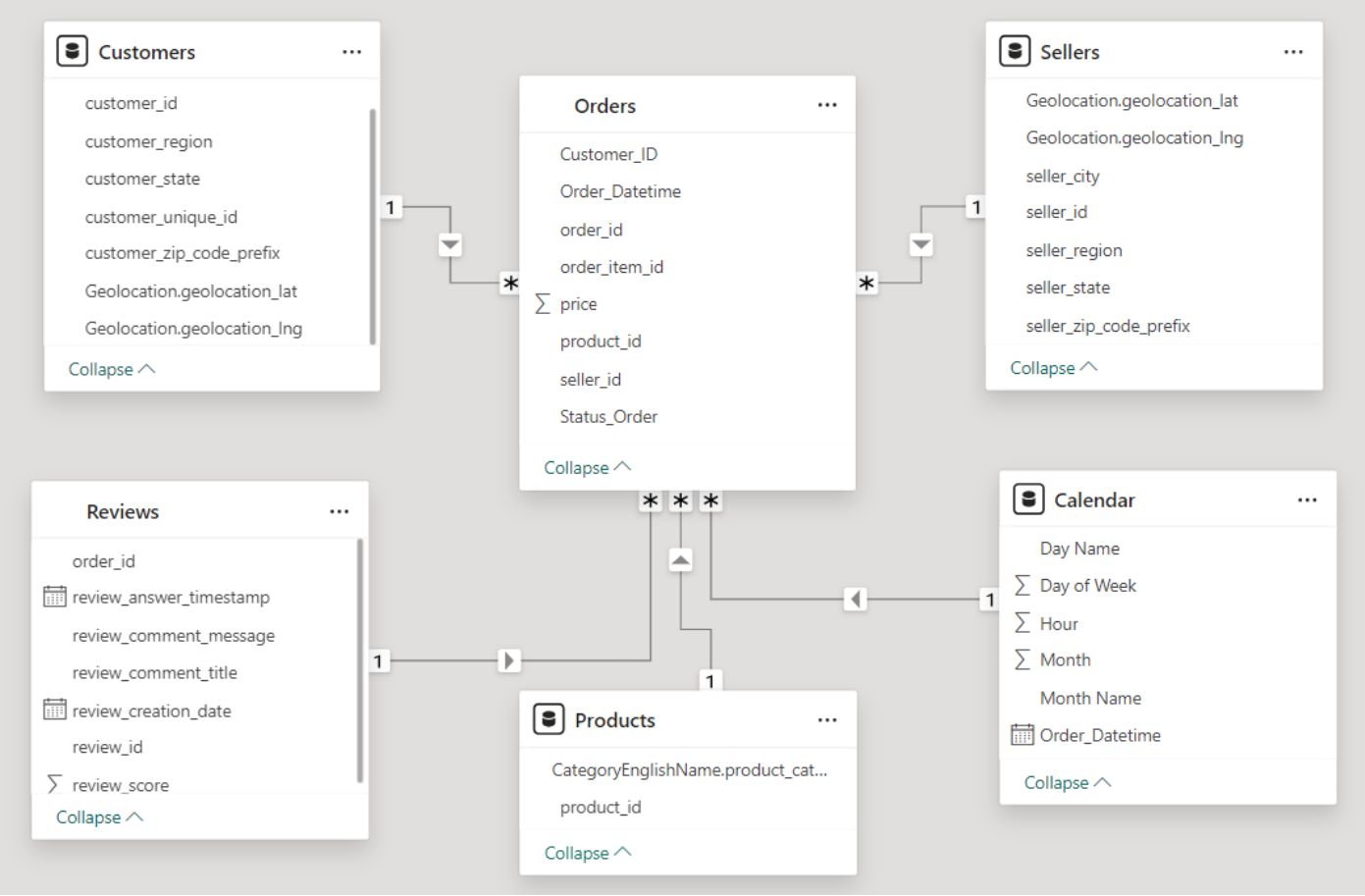



Image 1 – Data Model

## 4. Customers Overview

### 4.1. Customer distribution

Figure 2 showcases the geographic distribution of customers across Brazil's map.


Image 2 – Distribution of Customers

The company's customer base extends to every state in Brazil, with a notable concentration in the Southeast and South regions, as depicted in Figure 3.

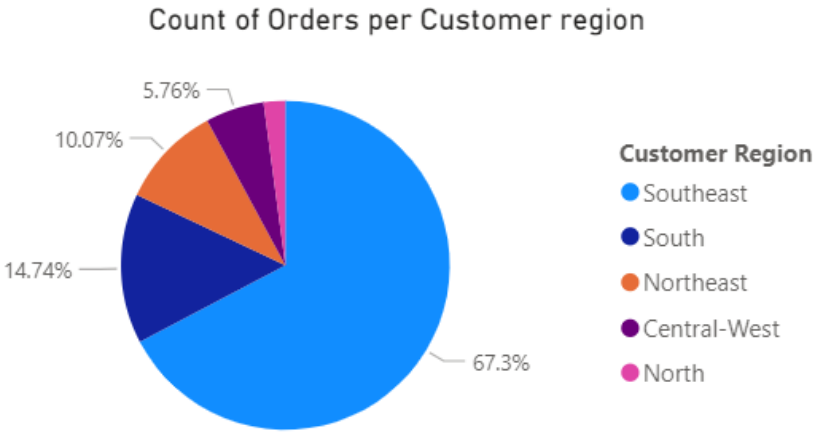



Image 3 – Order per customer region

Analyzing customer geolocation can be crucial for several reasons. It helps the company understand its market penetration in various regions, identify areas with high order volumes, and

strategize partnerships with local sellers. This knowledge ultimately enhances customer satisfaction by optimizing logistics, improving product availability, and tailoring services to regional preferences.

## 4.2. Product Categories

Understanding customer preferences is paramount for successful marketing campaigns, personalized recommendations, and targeted offers. Figure 4 depicts the top 15 most ordered product categories, highlighting the platform's extensive variety and showcasing the diverse range of options available for purchase. This insight is crucial for customer engagement with marketing campaigns and offers.

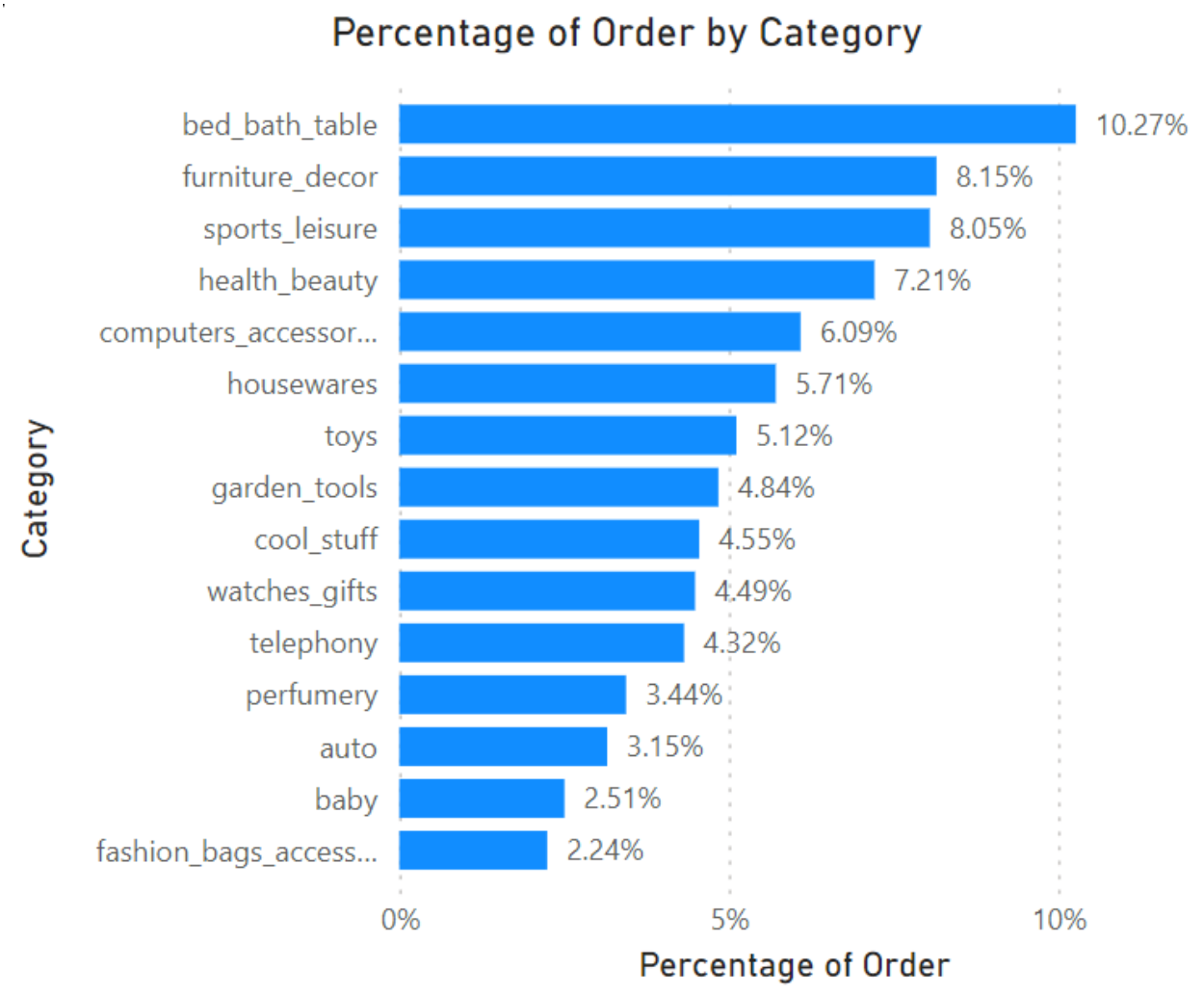



Image 4 – Percentage of Orders by Category

## 4.3. Review scores

Review scores play a critical role in evaluating the company's performance from the customers' perspective, assessing seller performance, and informing marketing campaigns. They directly impact customers' trust in the platform. Image 5 visually represents the distribution of orders across different review scores. To generate this visualization, a Python code was integrated into Power BI, as demonstrated in Image 6.

```
import pandas as pd
import matplotlib.pyplot as plt

# Group by review_score and count distinct order_ids
orders_per_review = dataset.groupby('review_score')['order_id'].nunique().reset_index()
orders_per_review = orders_per_review.sort_values(by='review_score', ascending=False)
review_categories = {1: 'Very Bad', 2: 'Bad', 3: 'Good', 4: 'Very Good', 5: 'Excellent'}
orders_per_review['review_score'] =
orders_per_review['review_score'].replace(review_categories)

# Plotting
plt.figure(figsize=(10, 5))
plt.bar(orders_per_review['review_score'], orders_per_review['order_id'])
plt.xlabel('Review Score')
plt.ylabel('Count of Order IDs')
plt.title('Count of Order IDs per Review Score')
plt.xticks(ha='center')
plt.grid(axis='y')
plt.tight_layout()

# Show plot
plt.show()
```

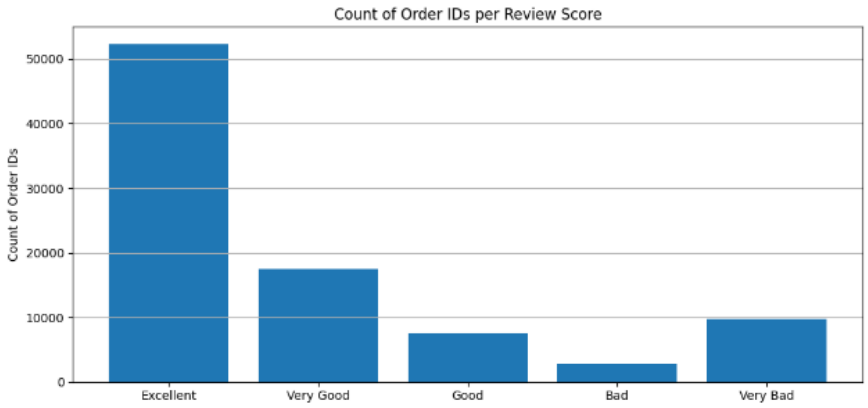



Image 5 – Count of orders per Review Score

```
import pandas as pd

import matplotlib.pyplot as plt

# Group by review_score and count distinct order_ids
orders_per_review = dataset.groupby('review_score')['order_id'].nunique().reset_index()
orders_per_review = orders_per_review.sort_values(by='review_score', ascending=False)

# Replace numeric review scores with corresponding strings
review_categories = {1: 'Very Bad', 2: 'Bad', 3: 'Good', 4: 'Very Good', 5: 'Excellent'}
orders_per_review['review_score'] =
orders_per_review['review_score'].replace(review_categories)

# Calculate the percentage of orders for each review score category
total_orders = orders_per_review['order_id'].sum()
orders_per_review['order_percentage'] = (orders_per_review['order_id'] / total_orders) *
100

# Plotting the pie chart
plt.figure(figsize=(7, 7))
plt.pie(orders_per_review['order_percentage'], labels=orders_per_review['review_score'],
autopct='%1.0f%%', startangle=140, textprops={'fontsize': 15})
plt.title('Percentage of Orders for Each Review Score Category', fontsize=18, pad=20)
plt.axis('equal')
plt.show()
```

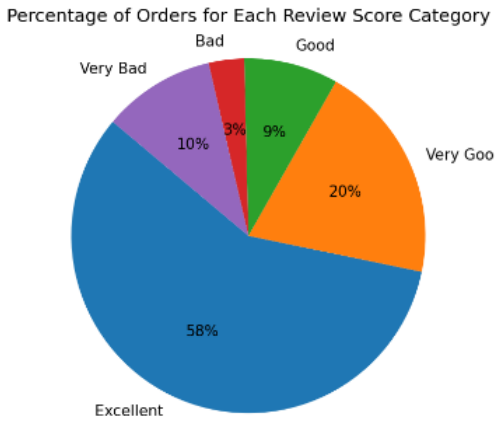



Image 6 – Percentage of each review score

```
import pandas as pd
import matplotlib.pyplot as plt

# Group customer_region and calculate average review_scores
df_unique = dataset.drop_duplicates(subset=['order_id'])
avg_review_per_region =
df_unique.groupby('customer_region')['review_score'].mean().reset_index()

# Plotting
plt.figure(figsize=(10, 5))
plt.bar(avg_review_per_region['customer_region'],
avg_review_per_region['review_score'], color='navy')
plt.xlabel('Customer Region', fontsize=14)
plt.ylabel('Average Review Score', fontsize=14)
plt.title('Average Review Score per Customer Region', fontsize=16)
plt.xticks(rotation=0, ha='center', fontsize=12)
plt.yticks(fontsize=12)
plt.grid(axis='y')

# Show plot
plt.show()
```

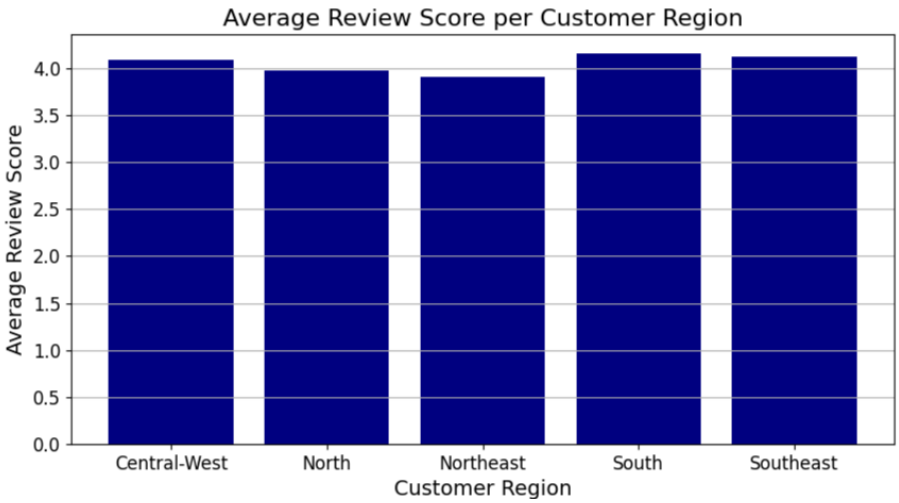



Image 7 – Review Score per Customer Region

## 4.4. Number of orders and spending per order

Image 8 presents data on the number of orders per month and the average order value per customer throughout the year 2017. The analysis reveals a dual trend: an increase in both the volume of orders and the average spending per customer. This trend signifies a growing trust and confidence among customers toward the platform.

```
import pandas as pd
import matplotlib.pyplot as plt

# Calculate outliers and filter out
Q1 = dataset['price'].quantile(0.25)
Q3 = dataset['price'].quantile(0.75)
IQR = Q3 - Q1
a = Q1 - 1.5 * IQR
b = Q3 + 1.5 * IQR
dataset_filtered = dataset[(dataset['price'] > a) & (dataset['price'] < b)]

# Calculate the average price and number of orders per month and merge
average_price_per_month = dataset_filtered.groupby('Month
Name')['price'].mean().reset_index()
orders_per_month = dataset.groupby('Month Name')['order_id'].count().reset_index()
merged_df = pd.merge(orders_per_month, average_price_per_month, on='Month Name')

# Convert 'Month' column to categorical with month order
month_order = ['January', 'February', 'March', 'April', 'May', 'June', 'July', 'August',
'September', 'October', 'November', 'December']
merged_df['Month Name'] = pd.Categorical(merged_df['Month Name'], categories=month_order,
ordered=True)
merged_df = merged_df.sort_values(by='Month Name')

# Change month names
month_short = {'January': 'Jan','February': 'Feb','March': 'Mar','April': 'Apr','May':
'May','June': 'Jun','July': 'Jul','August': 'Aug','September': 'Sep','October': 'Oct',
'November': 'Nov','December': 'Dec'}
merged_df['Month Name'] = merged_df['Month Name'].map(month_short)

# Plotting the bar chart
fig, ax1 = plt.subplots(figsize=(18,7))
ax1.bar(merged_df['Month Name'], merged_df['order_id'], color='b', label='Number of
Orders')
ax2 = ax1.twinx()
ax2.plot(merged_df['Month Name'], merged_df['price'], color='r', marker='o',
label='Average Price')
ax1.set_xlabel('Month Name', fontsize=20)
ax1.set_ylabel('Number of Orders', color='b', fontsize=20)
ax2.set_ylabel('Average Price', color='r', fontsize=20)
plt.title('Number of Orders and Average Price per Month', fontsize=20)
ax1.legend(loc='upper left')
ax2.legend(loc='upper right')
ax1.set_xticklabels(merged_df['Month Name'], rotation=45)

# Display the plot
plt.show()
```

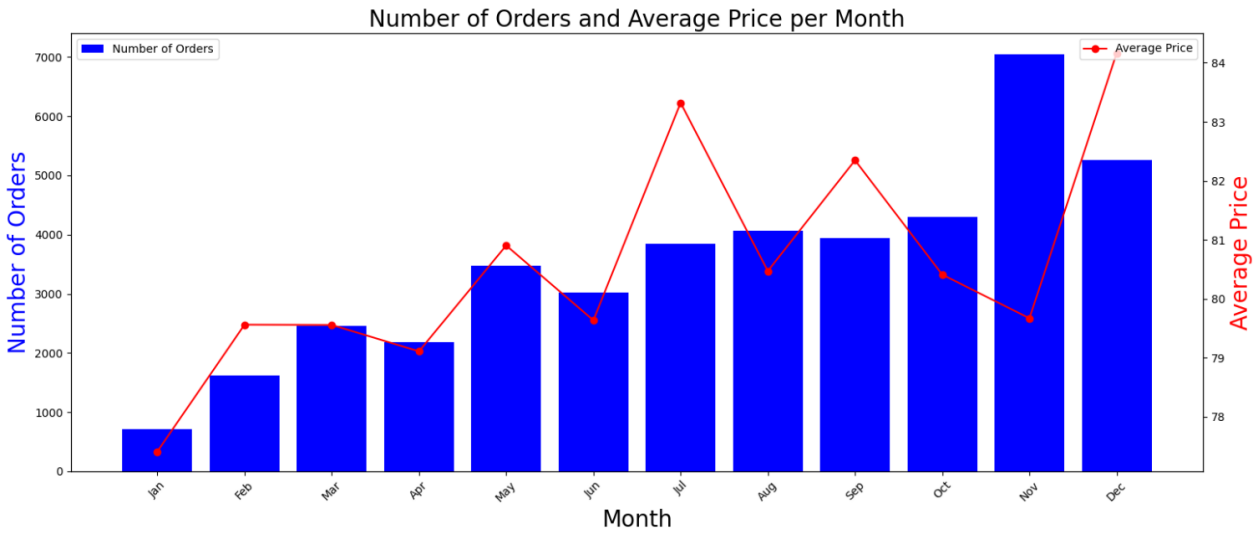



Image 8 – Number of Orders and Average Price per Month

The calculation mentioned above involved identifying and excluding outliers in order prices from the graph. This approach aimed to mitigate bias caused by exceptionally high-priced orders, which constitute a small fraction of the overall orders. By excluding outliers, the analysis yielded a more reliable and trustworthy result for the average spending per order.

## 5. Finding the Best Time to Advertise

Determining the optimal timing for launching marketing campaigns is pivotal for enhancing their effectiveness, as they can be rolled out when customers are most active on the platform.

### 5.1. Day of the month evaluation

An analysis of customer orders throughout the month is crucial for identifying trends. As depicted in Image 9, there is a noticeable decrease in engagement during the last 3 or 4 days of the month. Additionally, it's worth noting that the 24th stands out due to Black Friday, indicating a significant increase in activity on that day.

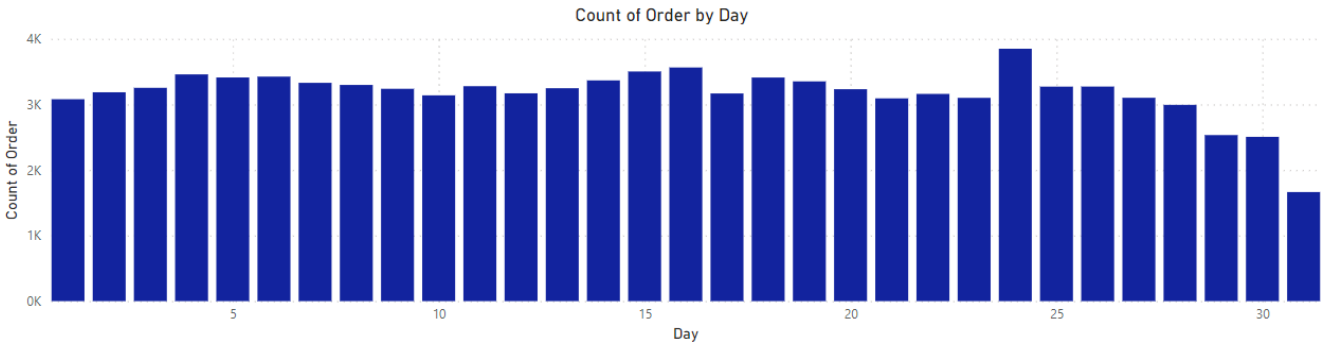



Image 9 – Count of Order by Day of the Month

As evidenced in Image 10 below, the 24th of November saw a remarkable spike in orders, with over 1,000 orders compared to less than 500 on other days of the month. This substantial difference highlights the significance of leveraging marketing efforts during promotional and festive dates to capitalize on heightened customer engagement.

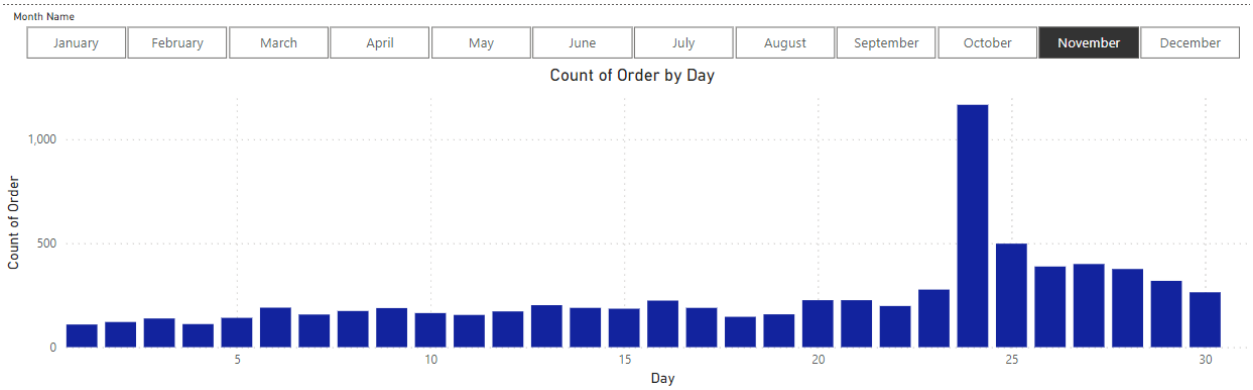



Image 10 – Count ot Order by Day (November)

## 5.2. Day of the Week Analysis

Upon analyzing the number of orders on each day of the week, it's noticeable that there are fewer orders during the weekends. Weekdays exhibit higher purchase activity, gradually declining as the week progresses.

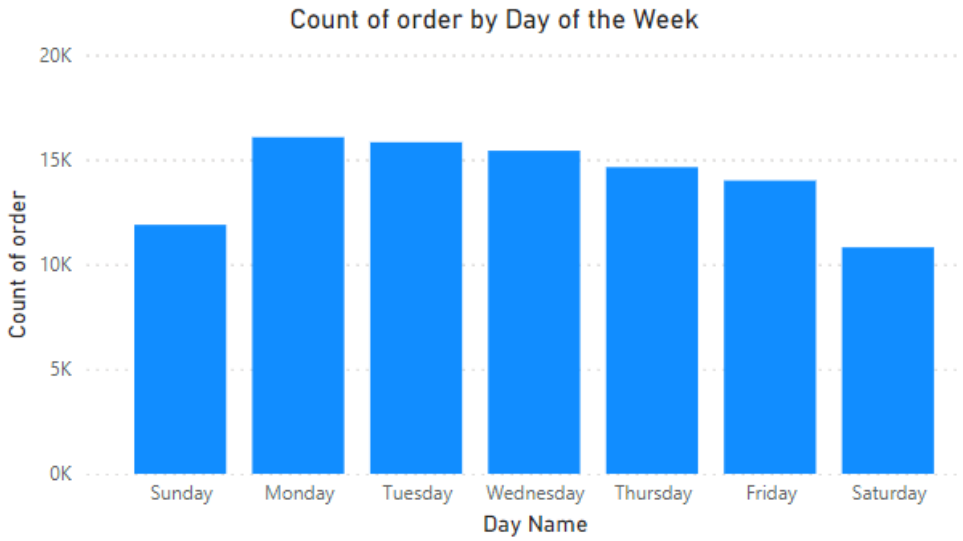



Image 11 – Orders by Day of the Week

## 5.3. Order per hour of the day

The busiest hours for advertising span from 10 am to 10 pm. However, the optimal focus for advertising lies within the time frame from 10 am to 4 pm, which appears to yield the most favorable results. Conversely, the platform experiences slower hours from 11 pm to 9 am.

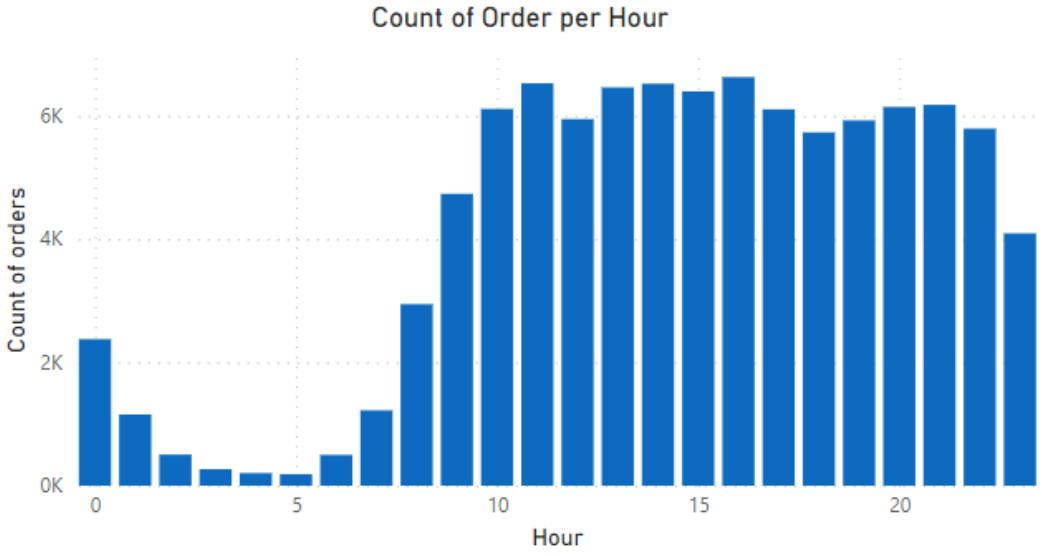



Image 12 – Count of Order per Hour

## 6. Customer Segmentation

RFM (Recency, Frequency, Monetary) analysis was applied to categorize customers into distinct groups allowing for targeted marketing campaigns directed to each customer segment to maximize engagement, retention, and profitability. Scores were given to each customer based on their last purchase, frequency of purchases, and the monetary value they spent in the company. Based on the customer's score they were divided into categories.

Image 13 below shows the scores based on the customer Recency, Frequency, and Monetary values.

| Monetary | Interval |
|---|---|
| Quartile 1 | 0 to 64 |
| Quartile 2 | 64 to 112 |
| Quartile 3 | 112 to 192 |
| Quartile 4 | > 192 |

| Recency | Score |
|---|---|
| <= 2 months | 5 |
| <= 6 months | 4 |
| <= 8 months | 3 |
| <= 1 year | 2 |
| > 1 year | 1 |

| Frequency | Score |
|---|---|
| >=5 | 5 |
| 4 | 4 |
| 3 | 3 |
| 2 | 2 |
| 1 | 1 |

Image 13 – Recency, Frequency and Monetary Analysis

In conducting the Monetary analysis, the price column underwent quartile division, with values above Quartile 4 earmarked for the Top-Value customers segment. This categorization resulted in six distinct customer segments, as illustrated in Figure 14.

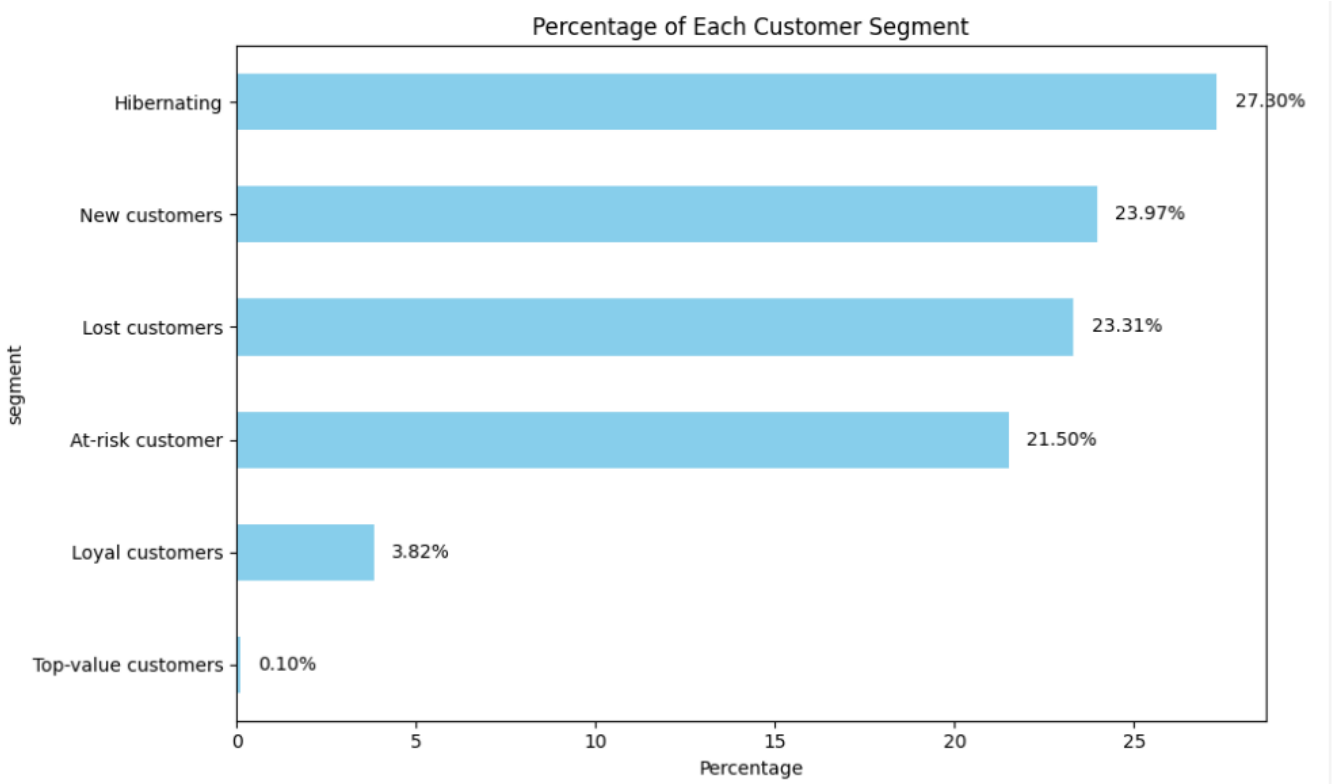
| Customer Segment | Recency | Frequency | Monetary ($) |
|---|---|---|---|
| Hibernating | <= 8 months | 1 | NA |
| At-risk customer | <= 1 year | 1 | NA |
| Loyal customer | <= 1 year | 2 | NA |
| Top-value customer | <= 6 months | 3 | > 192 |
| New customer | <= 6 months | 1 | NA |
| Lost customer | > 1 year | NA | NA |

Figure 14 – Customer segments and scores

The Python code implemented in this analysis segments customers based on their scores, allocating them into distinct segments as depicted in Figure 15, showcasing the percentage distribution within each segment.

```
import pandas as pd
import matplotlib.pyplot as plt

# Calculate recency (days since last purchase)
dataset['Order_Datetime'] = pd.to_datetime(dataset['Order_Datetime'])
max_date = dataset['Order_Datetime'].max()
dataset['recency'] = (max_date - dataset['Order_Datetime']).dt.days

# Calculate total purchases per customer (frequency)
dataset['frequency'] =
dataset.groupby('customer_unique_id')['customer_unique_id'].transform('count')
agg_funcs = {'Order_Datetime': 'first','price': 'sum','recency': 'first','frequency':
'first'}
grouped_dataset = dataset.groupby('customer_unique_id').agg(agg_funcs).reset_index()

# Formula for recency, frequency and monetary scores
def calculate_recency_score(recency_days):
    if recency_days <= 60:
        return 5
    elif recency_days <= 120:
        return 4
    elif recency_days <= 240:
        return 3
    elif recency_days <= 365:
        return 2
    elif recency_days > 365:
        return 1
    else:
        return 1

def calculate_frequency_score(frequency):
    if frequency >= 5:
        return 5
    elif frequency == 4:
        return 4
    elif frequency == 3:
        return 3
    elif frequency == 2:
        return 2
    elif frequency == 1:
        return 1

monetary_quartiles = pd.qcut(grouped_dataset['price'], q=4, labels=False)
```

```
# Create columns R, F and M
grouped_dataset['F'] = grouped_dataset['frequency'].apply(calculate_frequency_score)
grouped_dataset['R'] = grouped_dataset['recency'].apply(calculate_recency_score)
grouped_dataset['M'] = monetary_quartiles + 1

# Segmentation criteria
def assign_segment(row):
    if row['R'] >= 4 and row['F'] <= 1:
        return 'New customers'
    elif row['R'] >= 4 and row['F'] >= 3 and row['M'] >= 4:
        return 'Top-value customers'
    elif row['R'] >= 2 and row['F'] >= 2:
        return 'Loyal customers'
    elif row['R'] == 3 and row['F'] >= 1:
        return 'Hibernating'
    elif row['R'] == 1:
        return 'Lost customers'
    else:
        return 'At-risk customer'

# Create segment column and calculate the percentage of each
grouped_dataset['segment'] = grouped_dataset.apply(assign_segment, axis=1)
segment_counts = grouped_dataset['segment'].value_counts()
segment_percentage = (segment_counts / segment_counts.sum()) * 100

# Plotting the bar chart
plt.figure(figsize=(10, 6))
bars = segment_percentage.plot(kind='barh', color='skyblue')
plt.xlabel('Percentage')
plt.title('Percentage of Each Customer Segment')
plt.gca().invert_yaxis()
```




Possible approaches for each customer segment are described below:

**Hibernating Customers:** recommendations on products, targeted promotional campaigns, and highlighting the benefits of engagement to rekindle interest.

**At-Risk Customers:** targeted product recommendations, time-sensitive offers, and highlighting the perks of loyalty to retain their engagement.

**Loyal Customers:** tailored product recommendations based on their preferences, and exclusive loyalty rewards to enhance retention.

**Top-Value Customers:** exclusive offers, personalized communication addressing their unique needs, and incentives aimed at fostering long-term loyalty and repeat purchases.

**New Customers:** personalized onboarding messages, guidance in selecting products/sellers to facilitate initial purchases, and encouragement for subsequent transactions.

**Lost Customers:** standard communications with engaging offers, and presenting relevant product deals to recover engagement and encourage their return.

## 7. Conclusion

The project's comprehensive analysis using Power BI and Python codes on Olist's sales data spanning 2016 to 2018 resulted in valuable insights crucial for enhancing the company's marketing strategies. By aligning advertising efforts with peak periods and implementing targeted campaigns, the company would be able to improve resource allocation and drive growth with enhanced marketing strategies.

This strategy could be further enhanced by incorporating website tracking to monitor engagement with marketing campaigns and gain deeper insights into customer behavior. Enhancements to the website's functionality and user experience would also increase the effectiveness of these efforts. Moreover, leveraging machine learning algorithms for generating personalized recommendations would add a layer of sophistication, enriching the overall customer experience and driving greater engagement and conversion rates.