Data Analytics

CUSTOMER RETENTION

E-commerce Olist (Brazil)

Thi Hai Yen LE

Mars, 2023

Table of content

I. Introduction: Business case study

With a background in finance and experience working in commerce companies, I have always been fascinated by the role of customer retention in the success of a business. What is customer retention?

Customer retention refers to a company's ability to retain existing customers over a period. It is the ability of a company to keep its customers engaged with the brand, to maintain their loyalty, and to encourage repeat purchases or continued use of its products or services. In other words, customer retention is the ability of a company to ensure that its customers are satisfied with their experience and to encourage them to stay engaged with the company over time. This is important because it is generally more expensive to acquire new customers than it is to retain existing ones, and loyal customers are more likely to make repeat purchases, provide positive feedback and reviews, and refer others to the brand. It leads to increased profitability, brand loyalty, better customer insights, and a competitive advantage.

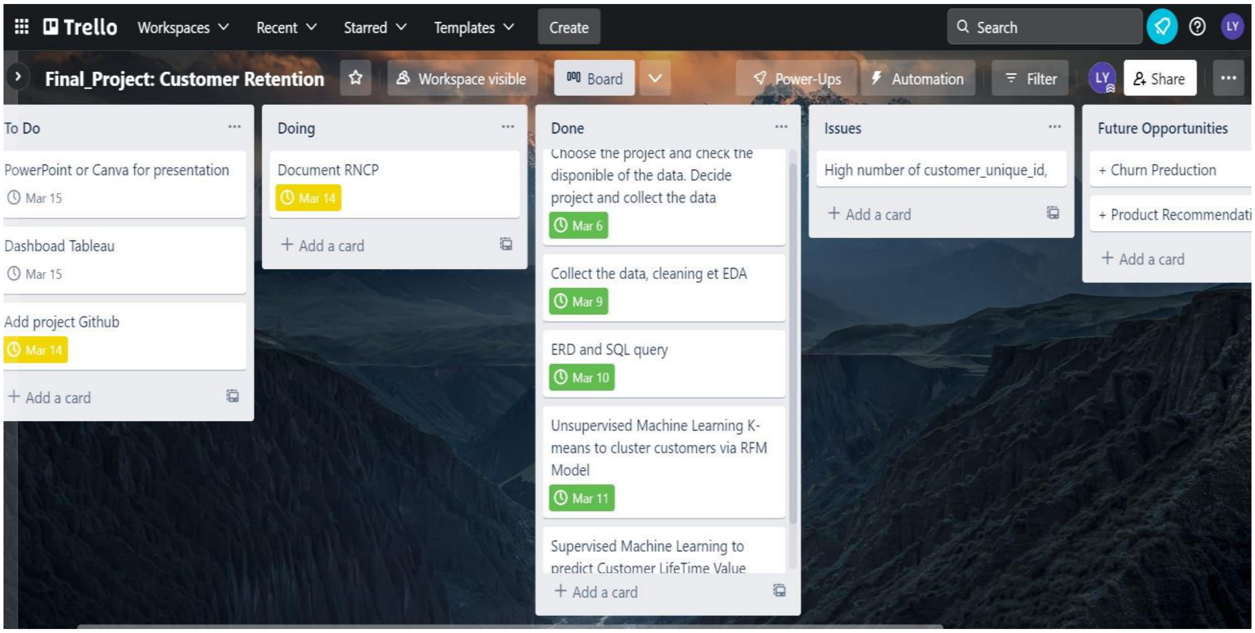
To further explore this topic, I decided to undertake a data analysis project focused on customer retention in the commerce sector. I used the Brazilian E-Commerce Public Dataset by Olist from Kaggle.

The goal of this project is to analyze an e-commerce dataset to provide valuable insights into customer behavior and develop strategies for customer retention. Using unsupervised machine learning, specifically K-means clustering via the RFM model (Recency - Frequency - Monetary), I segment customers to better understand their preferences. Additionally, I employ supervised learning models to predict customer lifetime value.

My ultimate objective is to provide actionable recommendations to improve customer retention and increase customer loyalty. I believe that my expertise in finance and commerce will be valuable in providing information to companies looking to optimize their customer retention strategies. I hope that my work will help companies better understand the drivers of customer churn and provide them with actionable recommendations to improve their retention rates.

After retrieving the necessary information and raw data that I need, I try to receive an overview of data I have imported, clean and perform exploratory data analysis. Then, I create an entity-relationship model and explore some queries with MySQL to show the insights. Last, suitable machine learning model will be deployed on the main dataset to cluster customers and predict the customer lifetime value.

- Plan Trello Project Management



Final_Project: Customer Retention

To Do

PowerPoint or Canva for presentation
Mar 15

Dashboad Tableau
Mar 15

Add project Github
Mar 14

+ Add a card

Doing

Document RNCP
Mar 14

+ Add a card

Done

Choose the project and check the disponible of the data. Decide project and collect the data
Mar 6

Collect the data, cleaning et EDA
Mar 9

ERD and SQL query
Mar 10

Unsupervised Machine Learning K-means to cluster customers via RFM Model
Mar 11

Supervised Machine Learning to predict Customer LifeTime Value

+ Add a card

Issues

High number of customer_unique_id,

+ Add a card

Future Opportunities

+ Churn Preduction

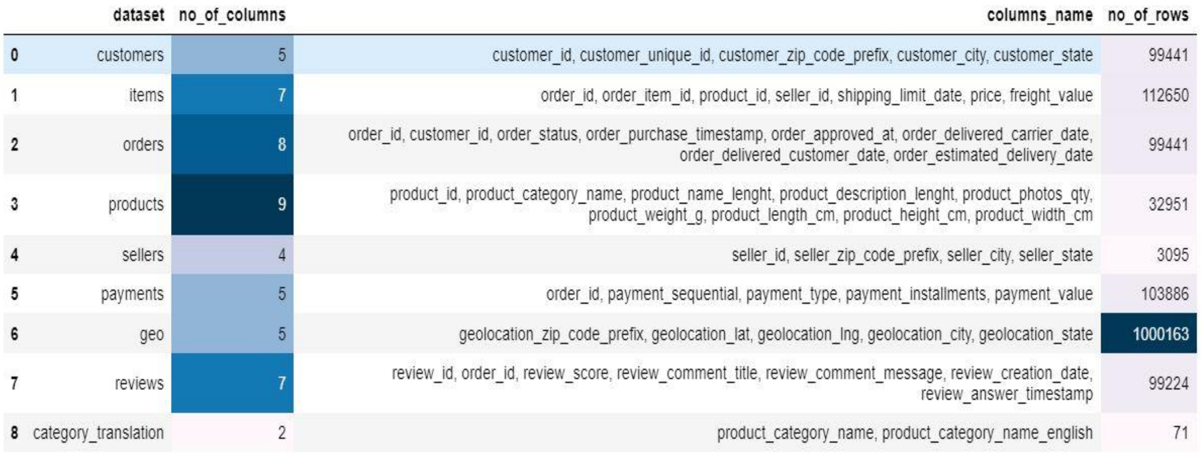
+ Product Recommendati

+ Add a card

II. Data: Brazilian E-Commerce Dataset by Olist

First, it is necessary to mention that Kaggle is a great platform for learning data science and machine learning, and it offers a wide range of real-world datasets in various industries and sectors. It has an amazing API, free, friendly user - you can find the documentation here https://www.kaggle.com/docs/api and also a python library (kaggle).

The dataset has information of 100k orders from 2016 to 2018 made at multiple marketplaces in Brazil at Olist Store. Its features allow viewing an order from multiple dimensions: from order status, price, payment and freight performance to customer location, product attributes and reviews with 9 csv file.



	dataset	no_of_columns	columns_name	no_of_rows
0	customers	5	customer_id, customer_unique_id, customer_zip_code_prefix, customer_city, customer_state	99441
1	items	7	order_id, order_item_id, product_id, seller_id, shipping_limit_date, price, freight_value	112650
2	orders	8	order_id, customer_id, order_status, order_purchase_timestamp, order_approved_at, order_delivered_carrier_date, order_delivered_customer_date, order_estimated_delivery_date	99441
3	products	9	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	32951
4	sellers	4	seller_id, seller_zip_code_prefix, seller_city, seller_state	3095
5	payments	5	order_id, payment_sequential, payment_type, payment_installments, payment_value	103886
6	geo	5	geolocation_zip_code_prefix, geolocation_lat, geolocation_lng, geolocation_city, geolocation_state	1000163
7	reviews	7	review_id, order_id, review_score, review_comment_title, review_comment_message, review_creation_date, review_answer_timestamp	99224
8	category_translation	2	product_category_name, product_category_name_english	71

The Olist E-commerce dataset is a comprehensive dataset containing information on multiple items within an order. It is important to note that each order is assigned a unique customer_id, which means that the same customer can have different ids for different orders. However, to identify customers who have made repurchases at the store, the dataset includes a customer_unique_id field. This allows analysts to track customer behavior and identify patterns of repeat purchases over time. This information is critical for e-commerce companies as it helps them to understand customer loyalty and engagement, and to tailor their marketing strategies accordingly. With this dataset, analysts can explore customer behavior and gain insights into the factors that drive repeat purchases and customer retention.

I have selected 6 tables from the Olist E-commerce dataset, as well as specific columns within those tables that are useful for the purpose of this project.

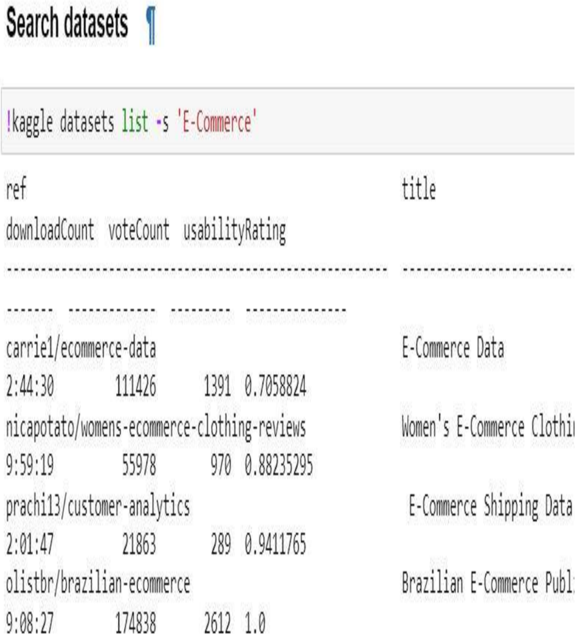
Following you can find the features explanation:

Dataset	Column Name	Description
customers	customer_id	key to the orders dataset. Each order has a unique customer_id.
	customer_unique_id	unique identifier of a customer.
	customer_city	customer city name
	customer_state	customer state
items	order_id	unique identifier of an order
	order_item_id	sequential number identifying number of items included in the same order.
	product_id	product unique identifier
	price	item price
	freight_value	item freight value item (if an order has more than one item the freight value is splitted between items)
orders	order_id	unique identifier of an order
	customer_id	key to the customer dataset. Each order has a unique customer_id.
	order_purchase_ timestamp	Shows the purchase timestamp.
reviews	order_id	unique identifier of an order
	review_score	Note from 1 to 5 given by the customer on a satisfaction survey.
category_ translation	product_category_name	category name in Portuguese
	product_category_name _english	category name in English

III. Data collection

I used the Kaggle API to search, download the data and unzip the data with command line and library kaggle of Python with following steps:

- Step 1: Login the kaggle and download kaggle API (username and password)
- Step 2: Install kaggle (!pip install kaggle)
- Step 3: Search datasets e-commerce (!kaggle datasets list -s 'E-Commerce')
- Step 4: Choose the dataset and download (!kaggle datasets download -d "olistbr/brazilian-ecommerce")
- Step 5: Unzip file and save its in the folder: raw_data



Search datasets ¶

```
!kaggle datasets list -s 'E-Commerce'
```

```
ref                                                          title
downloadCount  voteCount  usabilityRating
------------------------------------------------------------ ---------------------------
-------  -----------  --------  -------------
carrie1/ecommerce-data                                       E-Commerce Data
2:44:30         111426       1391  0.7058824
nicapotato/womens-ecommerce-clothing-reviews                 Women's E-Commerce Clothi
9:59:19          55978        970  0.88235295
prachi13/customer-analytics                                   E-Commerce Shipping Data
2:01:47          21863        289  0.9411765
olistbr/brazilian-ecommerce                                  Brazilian E-Commerce Publ
9:08:27         174838       2612  1.0
```

Then I imported them to Jupiter notebook using Pandas read csv function. From here I start to clean the data, enrich data, explore in visualization and do machine learning.

IV. Data cleaning and Exploratory data analysis (EDA)

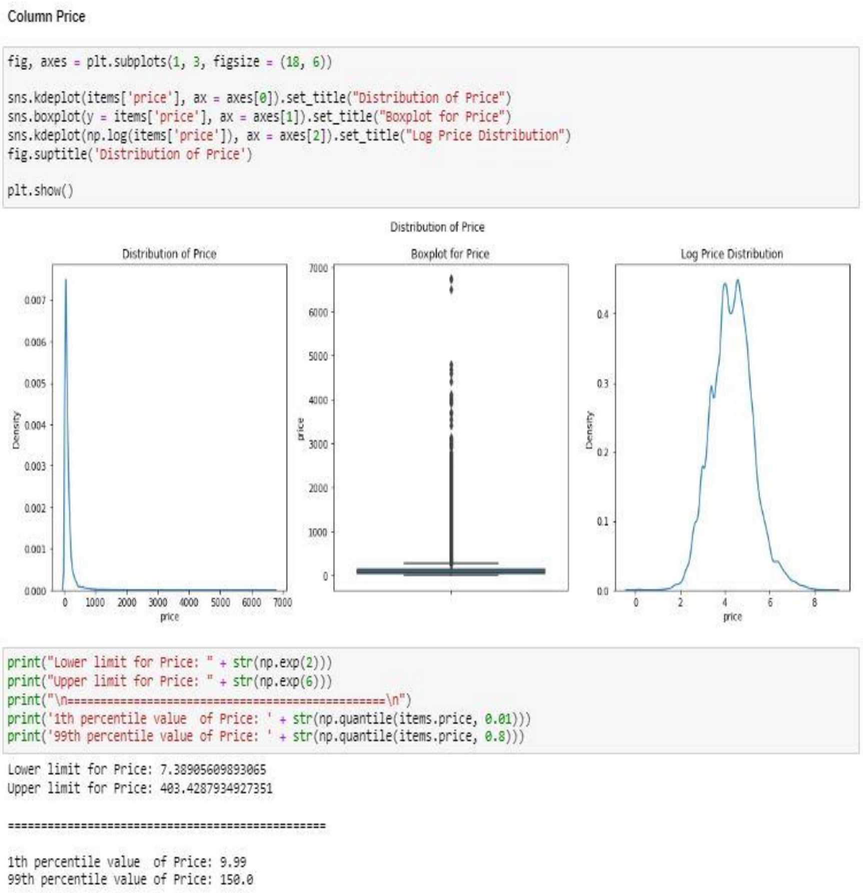
1. Data cleaning (Collecting_cleaning_data.ipynb)

Firstly, I imported all the necessary libraries I need to work with on my Jupiter notebook such as: NumPy, pandas and matplotlib, seaborn for visualization, …

Secondly, I imported 9 .csv data files and began reviewing them to determine the number of tables and columns in each, as well as their corresponding column names. This analysis will enable me to select the appropriate tables and columns required for my task.

After checking data (.isnull().sum()), I noticed that there were several missing values in the dataset. However, these missing values were found in columns that were not useful for my analysis.

Therefore, I made the decision to drop all missing values, except for those found in the "produit_category_name" column, as it is a crucial factor for my analysis. For this column, there are 610 rows from the products dataframe (products = products.dropna()) that contain null values because I had almost 32.341 rows in this table, so it was nothing.



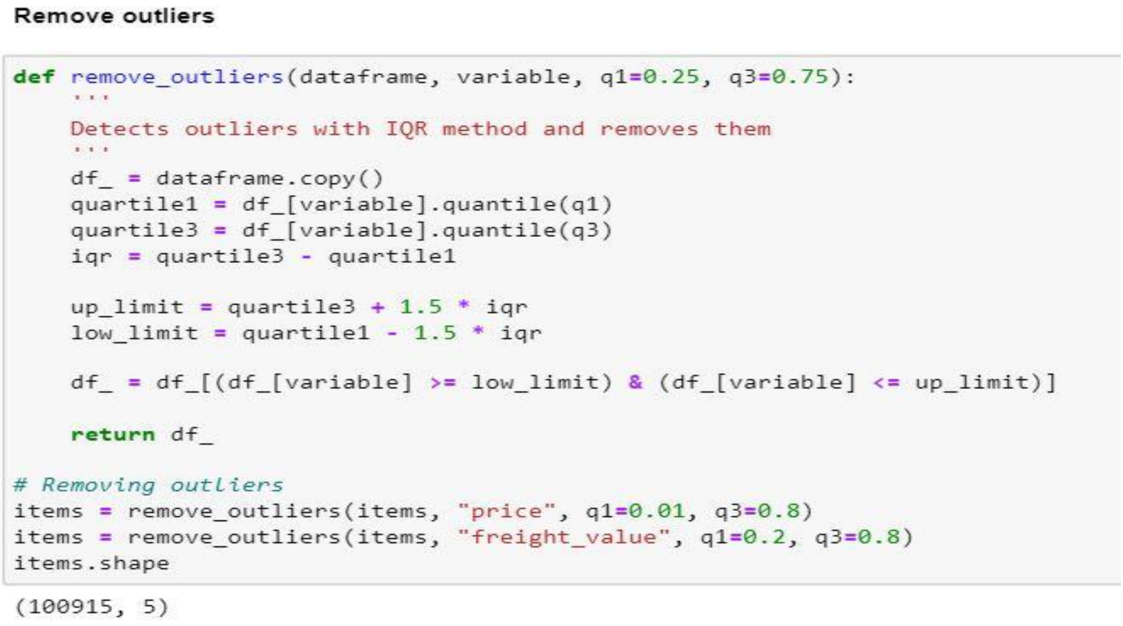
Column Price

```
fig, axes = plt.subplots(1, 3, figsize = (18, 6))

sns.kdeplot(items['price'], ax = axes[0]).set_title("Distribution of Price")
sns.boxplot(y = items['price'], ax = axes[1]).set_title("Boxplot for Price")
sns.kdeplot(np.log(items['price']), ax = axes[2]).set_title("Log Price Distribution")
fig.suptitle('Distribution of Price')

plt.show()
```

```
print("Lower limit for Price: " + str(np.exp(2)))
print("Upper limit for Price: " + str(np.exp(6)))
print("\n==================================================\n")
print('1th percentile value  of Price: ' + str(np.quantile(items.price, 0.01)))
print('99th percentile value of Price: ' + str(np.quantile(items.price, 0.8)))
```

```
Lower limit for Price: 7.38905609893065
Upper limit for Price: 403.4287934927351

==================================================

1th percentile value  of Price: 9.99
99th percentile value of Price: 150.0
```

I remove the outliers of the columns price and freight_value:

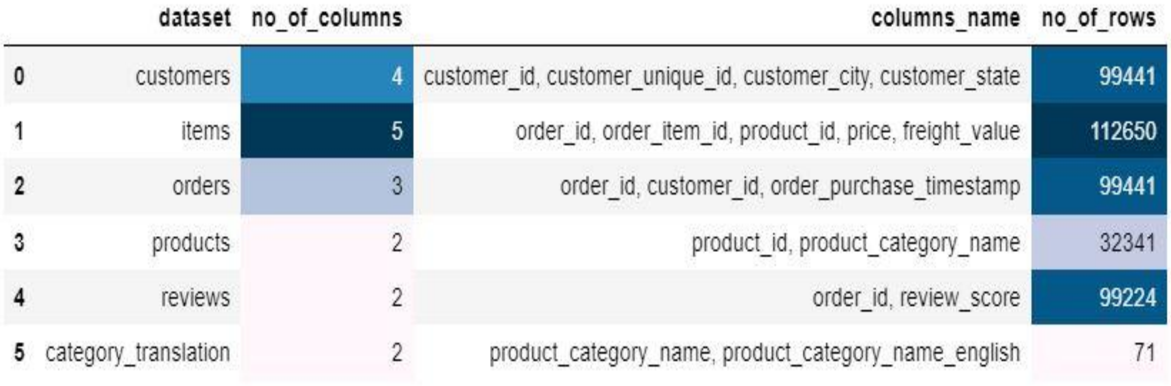


Remove outliers

```
def remove_outliers(dataframe, variable, q1=0.25, q3=0.75):
    '''
    Detects outliers with IQR method and removes them
    '''
    df_ = dataframe.copy()
    quartile1 = df_[variable].quantile(q1)
    quartile3 = df_[variable].quantile(q3)
    iqr = quartile3 - quartile1

    up_limit = quartile3 + 1.5 * iqr
    low_limit = quartile1 - 1.5 * iqr

    df_ = df_[(df_[variable] >= low_limit) & (df_[variable] <= up_limit)]

    return df_

# Removing outliers
items = remove_outliers(items, "price", q1=0.01, q3=0.8)
items = remove_outliers(items, "freight_value", q1=0.2, q3=0.8)
items.shape
```

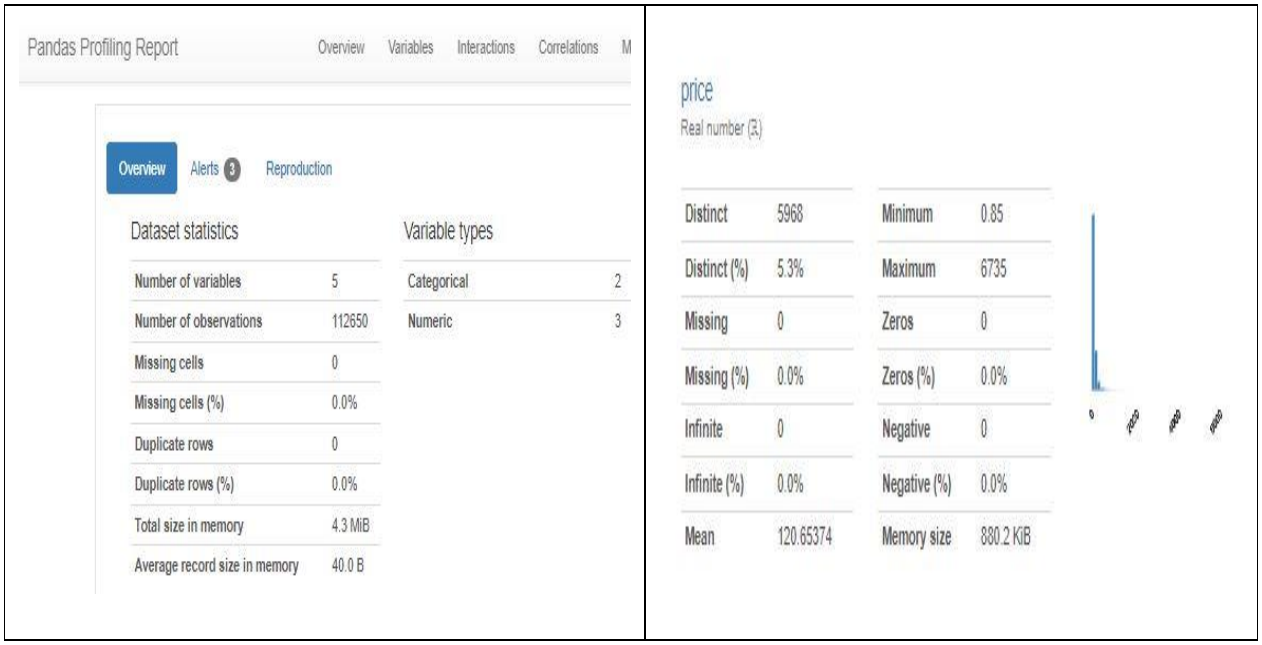
```
(100915, 5)
```

So, I have the dataset as bellow:



	dataset	no_of_columns	columns_name	no_of_rows
0	customers	4	customer_id, customer_unique_id, customer_city, customer_state	99441
1	items	5	order_id, order_item_id, product_id, price, freight_value	112650
2	orders	3	order_id, customer_id, order_purchase_timestamp	99441
3	products	2	product_id, product_category_name	32341
4	reviews	2	order_id, review_score	99224
5	category_translation	2	product_category_name, product_category_name_english	71

Then I used library from pandas_profiling import ProfileReport to have an overview for the items table.



Dataset statistics

Number of variables	5
Number of observations	112650
Missing cells	0
Missing cells (%)	0.0%
Duplicate rows	0
Duplicate rows (%)	0.0%
Total size in memory	4.3 MiB
Average record size in memory	40.0 B

Variable types

Categorical	2
Numeric	3

price
Real number (ℝ)

Distinct	5968	Minimum	0.85
Distinct (%)	5.3%	Maximum	6735
Missing	0	Zeros	0
Missing (%)	0.0%	Zeros (%)	0.0%
Infinite	0	Negative	0
Infinite (%)	0.0%	Negative (%)	0.0%
Mean	120.65374	Memory size	880.2 KiB

To analyze customer retention, I selected a 12-month period from the available data spanning from 2016 to 2018. However, since the data for 2016 and 2018 are incomplete, I decided to only use the data from 2017 (orders = orders[orders['order_purchase_timestamp'].dt.year == 2017])

I identified a key column that is common across tables and used a join operation to combine the 6 tables into one based on this key column.

After merging the tables, it is important to remove any duplicate rows to ensure that the data is accurate and not redundant. To do this, you used the "drop_duplicates" function of pandas. In this case, I had 233 duplicate rows and removed them from the dataset.

Next, added a new column called "total_amount" that calculates the sum of the "price" and "freight_value" columns. This new column will provide additional insight into the total cost of each order (data['total_amount'] = data['price'] + data['freight_value']).

I reordered the columns so that they are arranged in a more logical way (data = data.reindex(columns = new_col, inplace = True).

Finally, I have a final data with 12 columns and 44.943 rows. I exported the cleaned dataset as separate dataframes and a final dataset to CSV files in the "cleaned_data" folder. This will allow others to easily access and analyze the data for next steps.

2. Exploratory data analysis (EDA):

2.1. Cohort analysis and customer behavior

Dealing with low retention rate can be just as challenging as handling customer churn for any organization. For customer success and product teams, it can be frustrating when customers do not return to make repeat purchases, even when they may be willing to do so.

One effective way to address this issue is by using cohort analysis. This technique provides valuable insights into customer behavior, enabling you to identify trends and patterns and understand how they affect the business.

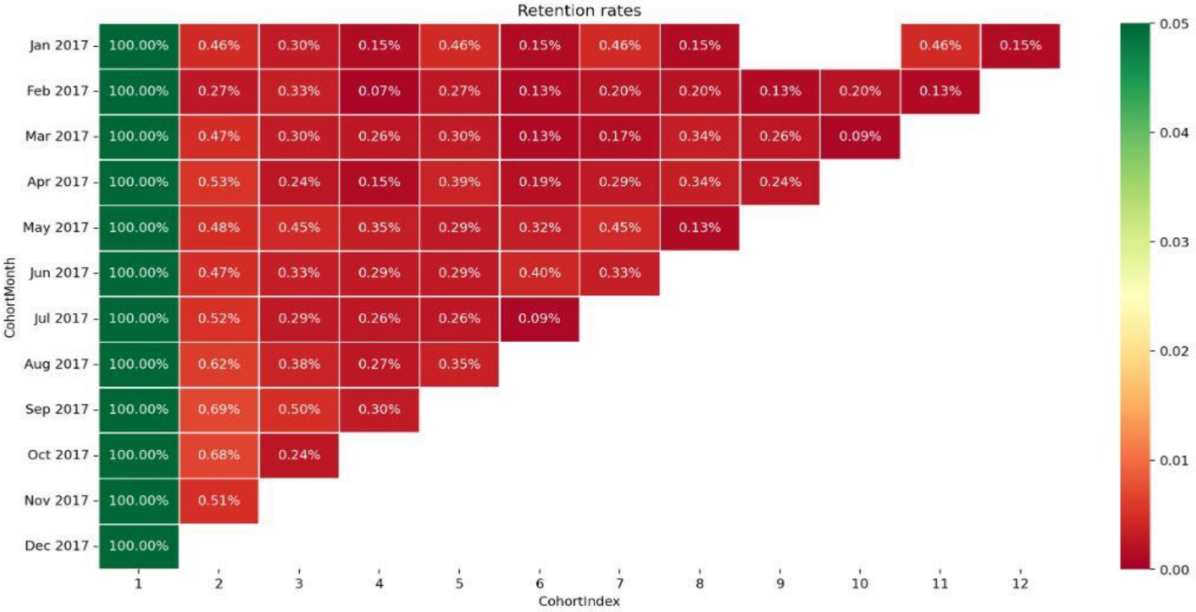
By analyzing cohorts, I can gain a deeper understanding of customer behavior over time and identify opportunities to improve retention and increase repeat purchases. Cohort analysis offers in-depth insights into customer and user behavior, as well as product or business performance.

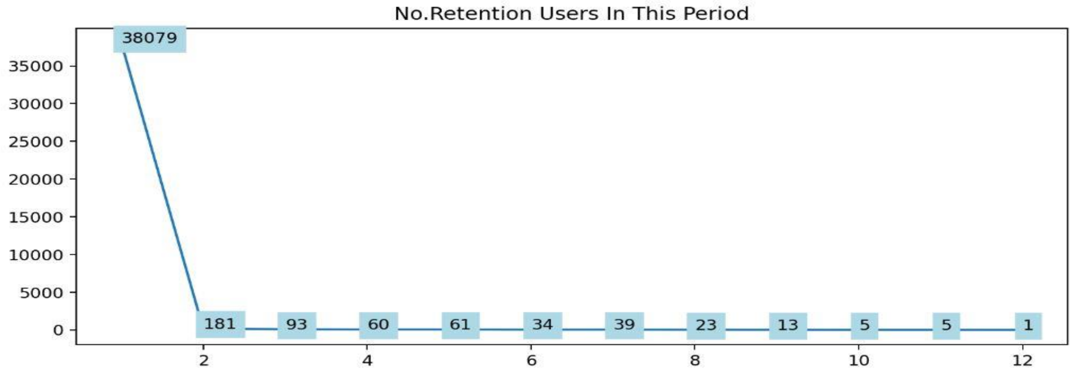
In my project, I chose a 12-month period. This allows me to better understand customer behavior over time and identify opportunities to improve retention and loyalty.

Here are my summary steps to make the cohort analysis:

- Define a function to extract the month from the "order_purchase_timestamp" column using the .datetime() method.

- Add a new column called "order_month" to the dataframe to represent the month in which each order was placed. Use the function created in step 1 to extract the month from the "order_purchase_timestamp" column.
- Create a new column called "CohortMonth" to represent the month in which each customer made their first purchase. This can be done by grouping the data by customer ID and finding the earliest order month for each.
- Create a new column called "CohortIndex" to indicate the number of months that have passed since each customer's first purchase. This can be calculated by finding the difference between the order month and the cohort month for each.



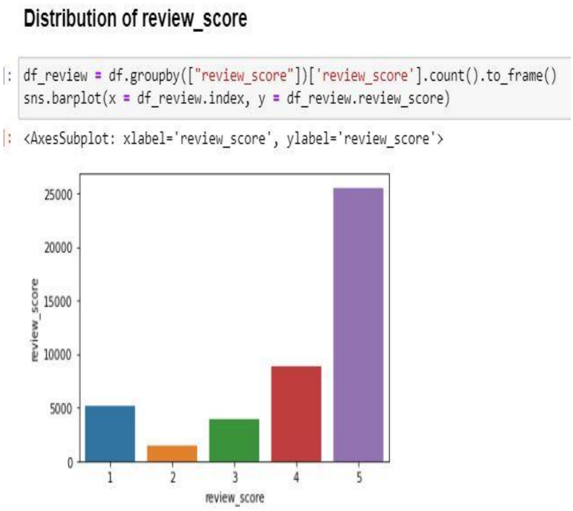


It seems that the number of returning customers is significantly low, less than 1%. This could indicate that there may be issues with customer satisfaction or loyalty. It may be necessary to investigate the reasons why customers are not returning to Olist to make improvements in customer retention.

I make some more analyze to know this situation of the Olist:

I first focused on analyzing customer feedback and then revenue. Understanding customer feedback is crucial as it can help identify areas for improvement in the customer experience, which can lead to increased customer loyalty.

Most of the review scores are either 4/5 or 5/5, indicating that the low retention rate is not necessarily due to customers having a negative experience with Olist.

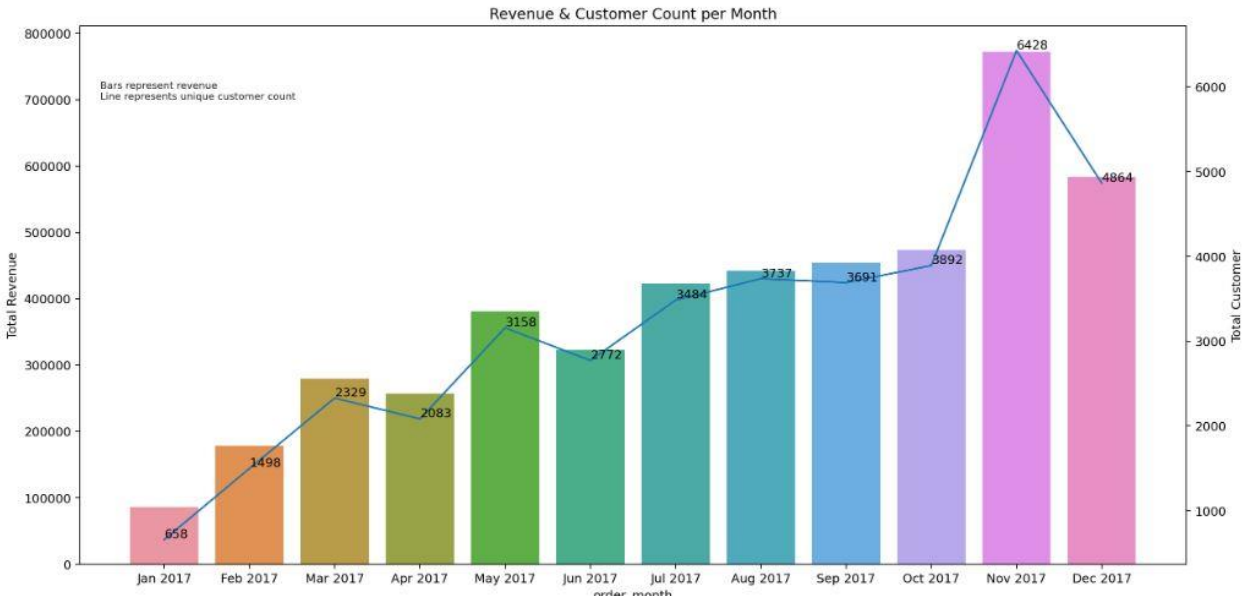


Distribution of review_score

```
df_review = df.groupby(["review_score"])['review_score'].count().to_frame()
sns.barplot(x = df_review.index, y = df_review.review_score)
```

```
<AxesSubplot: xlabel='review_score', ylabel='review_score'>
```

Then, I continued to focus on the revenue.



The revenue increases with an increase in the number of customers. The revenue in November is significantly higher than other months due to the occurrence of Black Friday, where a lot of promotional offers and discounts are provided to customers.

However, this chart does not distinguish between revenue from new customers and revenue from returning customers.

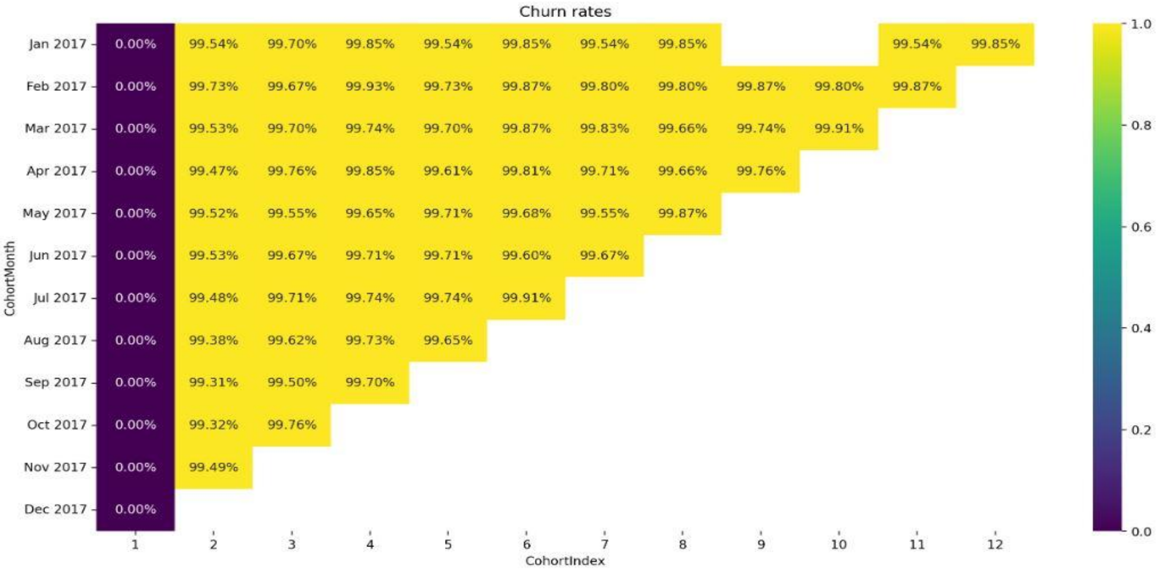
According to research, it is much less expensive to retain an existing customer than to acquire a new one. Therefore, investing in strategies to improve customer retention using cohort analysis can have a significant impact on the business.

2.2. Customer retention KPIs:

Next to several additional customer retention KPIs:

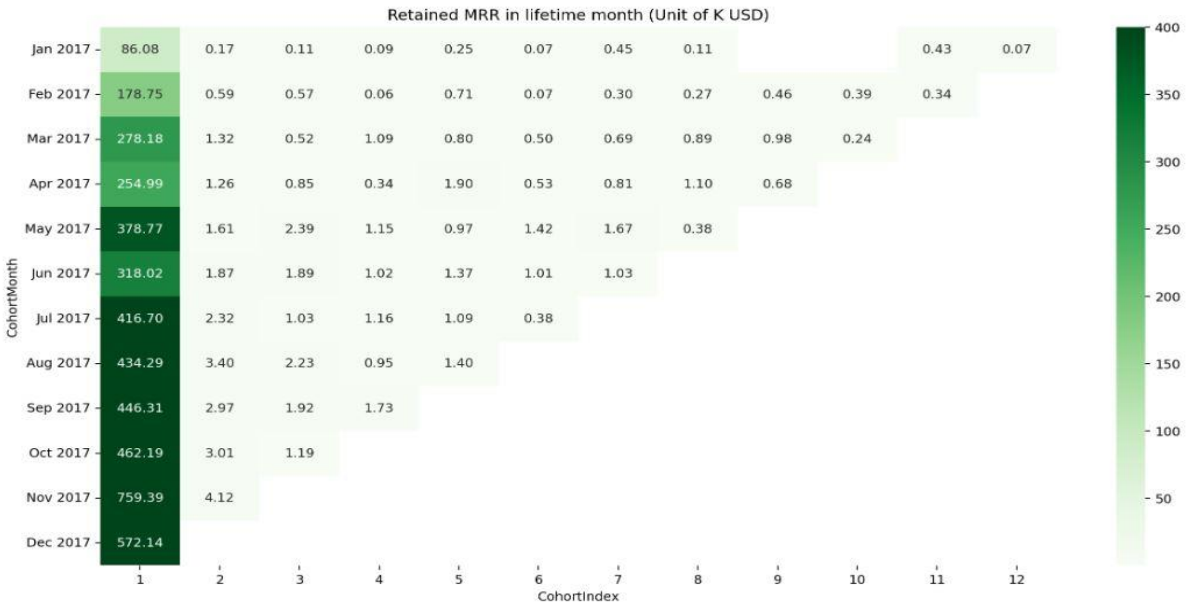
- Churn Rate:

Churn rate refers to the rate at which customers or subscribers discontinue their relationship with a company or service over a given period.



- Monthly Recurring Revenue:

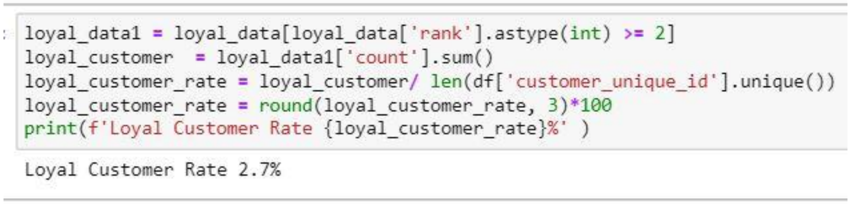
Monthly Recurring Revenue is a key metric used by businesses that offer subscription-based services or products. It refers to the amount of revenue that a company can expect to receive on a monthly basis from its customers



The main revenue comes from new customers, while revenue from existing customers accounts for a very small proportion.

- Loyal Customer Rate:

Loyal customer rate refers to the percentage of customers who have made repeat purchases from a business over a specific period.

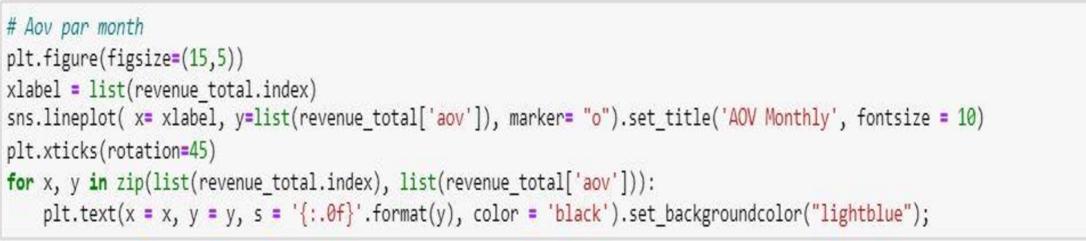


```
loyal_data1 = loyal_data[loyal_data['rank'].astype(int) >= 2]
loyal_customer  = loyal_data1['count'].sum()
loyal_customer_rate = loyal_customer/ len(df['customer_unique_id'].unique())
loyal_customer_rate = round(loyal_customer_rate, 3)*100
print(f'Loyal Customer Rate {loyal_customer_rate}%' )
```

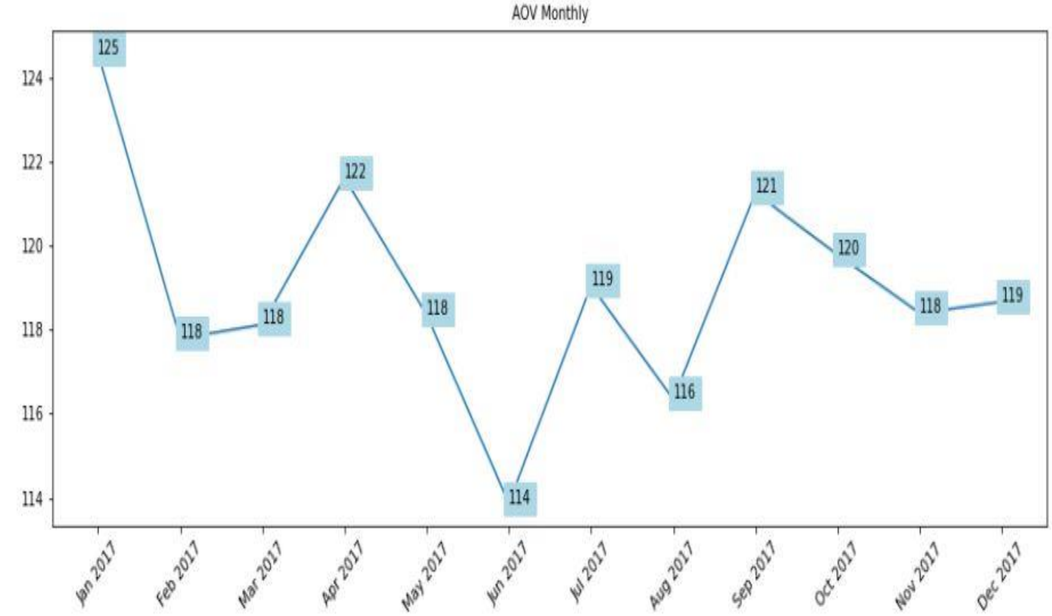
```
Loyal Customer Rate 2.7%
```

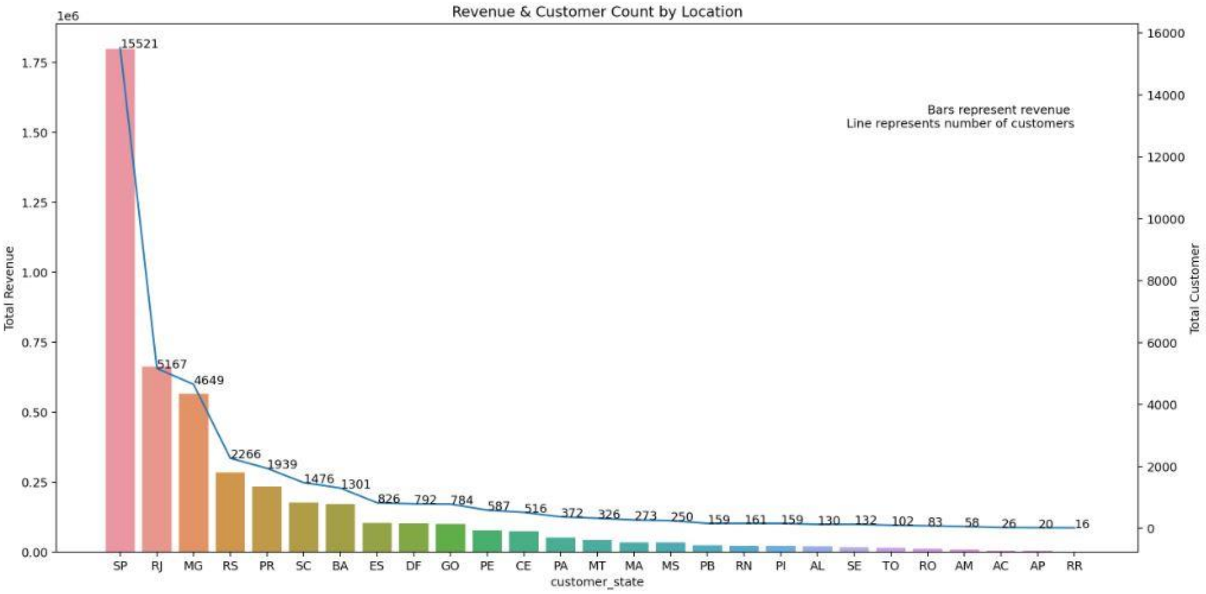
It means that only 2.7% of the total customers who made purchases from Olist during the defined period have made repeat purchases. This suggests that Olist may have a relatively low level of customer loyalty or repeat business and may need to focus on improving its customer retention strategies to encourage more customers to make repeat purchases.

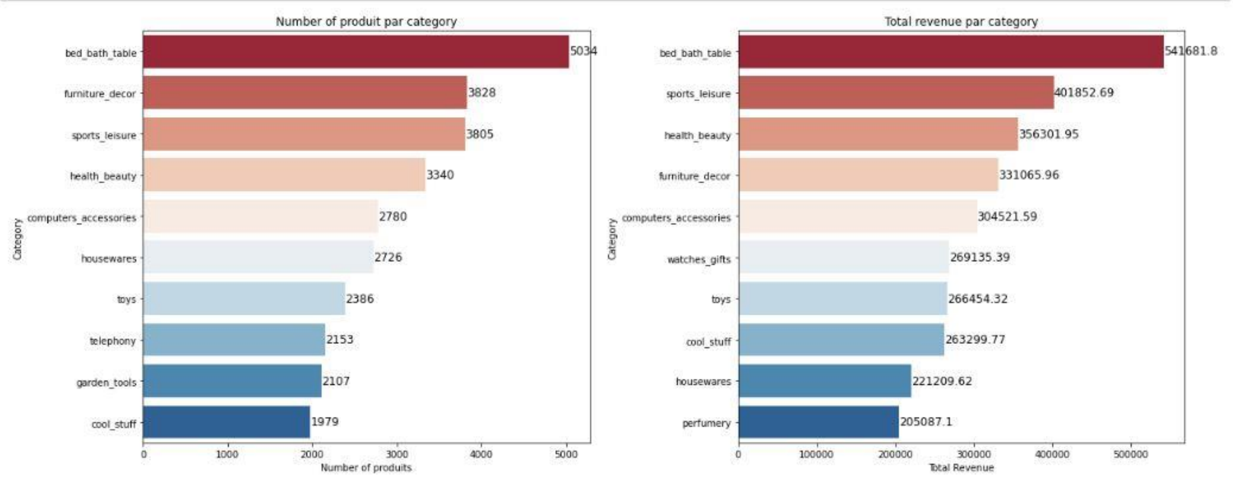
More insights:



```
# Aov par month
plt.figure(figsize=(15,5))
xlabel = list(revenue_total.index)
sns.lineplot( x= xlabel, y=list(revenue_total['aov']), marker= "o").set_title('AOV Monthly', fontsize = 10)
plt.xticks(rotation=45)
for x, y in zip(list(revenue_total.index), list(revenue_total['aov'])):
    plt.text(x = x, y = y, s = '{:.0f}'.format(y), color = 'black').set_backgroundcolor("lightblue");
```







V. SQL

To proceed with my project, the next step is to create a database. After careful consideration, I have decided to use SQL due to its ability to import my cleaned and converted data frame in a structured manner and its capacity to create and inform my queries.

1. Database type

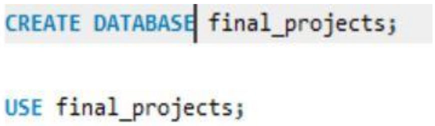
To proceed with my project, the next step is to create a database. After careful consideration, I have decided to use SQL due to its ability to import my cleaned and converted data frame in a structured manner and its capacity to create and inform my queries.

The following are the five key differences between SQL and NoSQL:

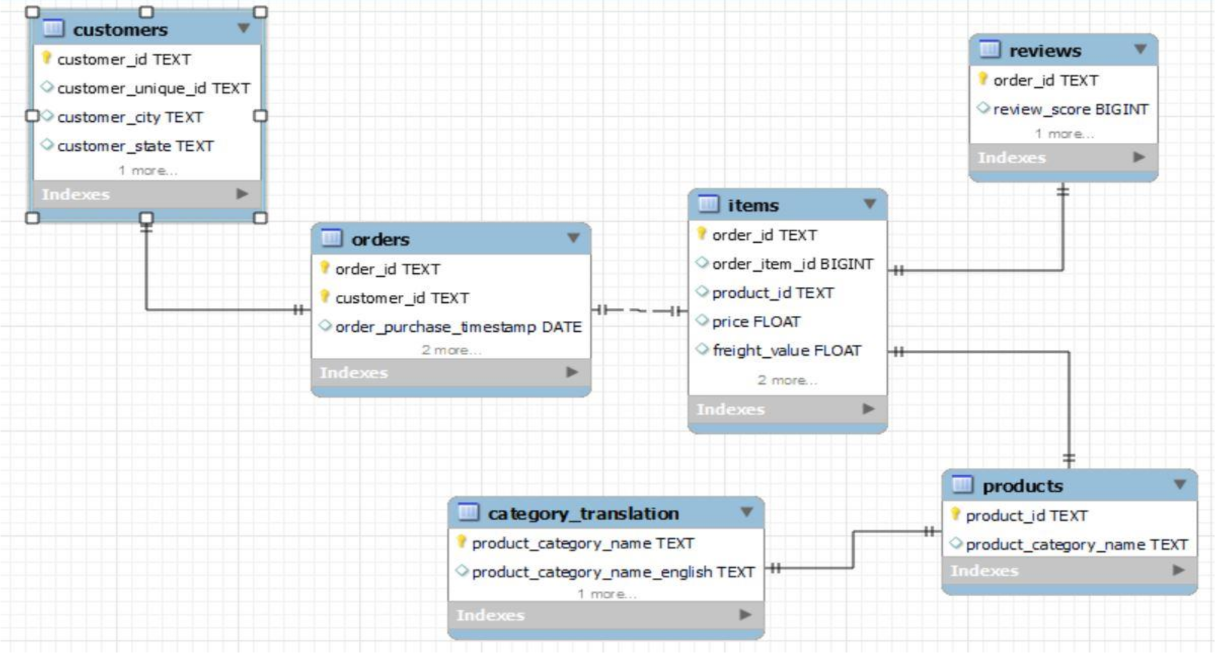
SQL	**NoSQL**
- Relational - Use structured query language and predefined schema - Are vertically scalable - Are table-based -Are better for multi-row transactions	- Non-relational - Have dynamic schemas for unstructured or semi structured data - Are horizontally scalable - Databases are document, key-value, graph, or wide-column stores - better for unstructured data like documents or JSON

2. Entity-relationship diagram (ERD)

The entities of the diagram represent my datasets after the cleaning process. First, I had my six tables. The diagram for ERD can be seen as follows:



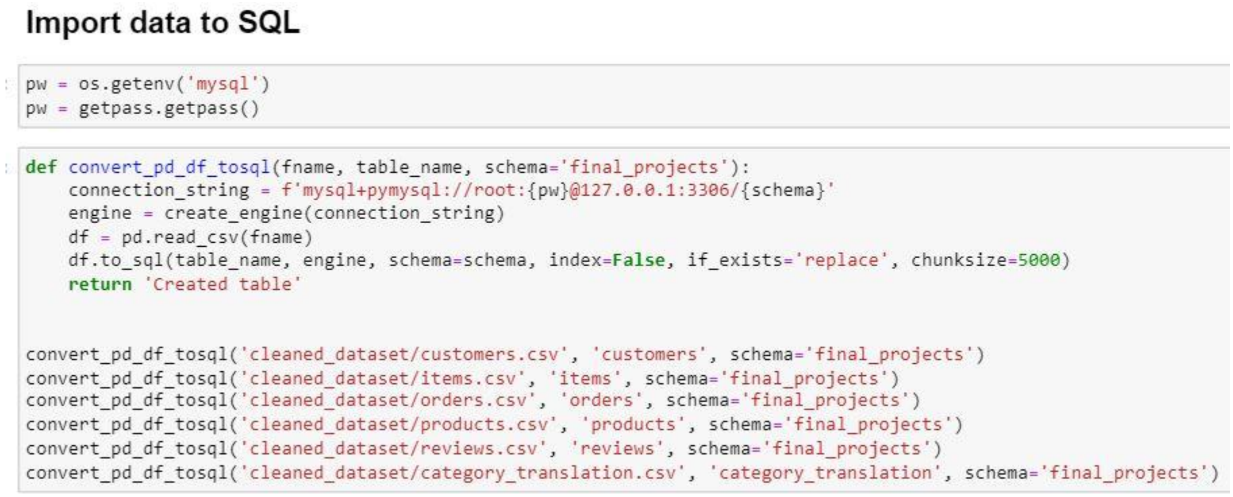
```
CREATE DATABASE final_projects;

USE final_projects;
```



3. Creation of the database and data importation:

Once I finalized my decision on the type of database to use, I proceeded to create a relational database on MySQL Workbench using:

To import data, I connected my Python file to the MySQL database using SQLalchemy.



Import data to SQL

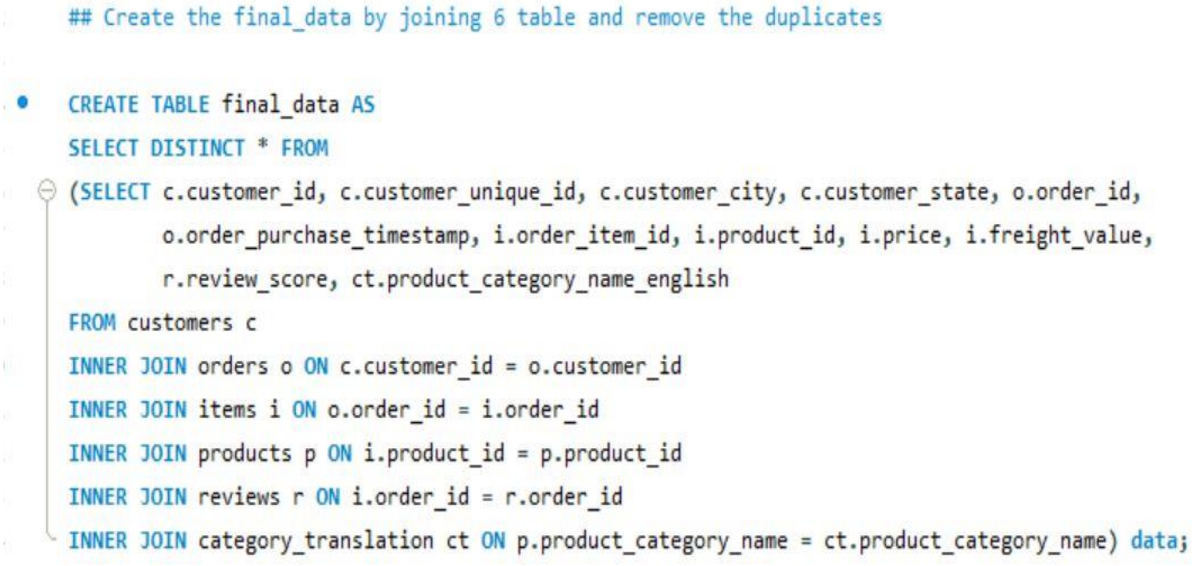
```
pw = os.getenv('mysql')
pw = getpass.getpass()
```

```
def convert_pd_df_tosql(fname, table_name, schema='final_projects'):
    connection_string = f'mysql+pymysql://root:{pw}@127.0.0.1:3306/{schema}'
    engine = create_engine(connection_string)
    df = pd.read_csv(fname)
    df.to_sql(table_name, engine, schema=schema, index=False, if_exists='replace', chunksize=5000)
    return 'Created table'


convert_pd_df_tosql('cleaned_dataset/customers.csv', 'customers', schema='final_projects')
convert_pd_df_tosql('cleaned_dataset/items.csv', 'items', schema='final_projects')
convert_pd_df_tosql('cleaned_dataset/orders.csv', 'orders', schema='final_projects')
convert_pd_df_tosql('cleaned_dataset/products.csv', 'products', schema='final_projects')
convert_pd_df_tosql('cleaned_dataset/reviews.csv', 'reviews', schema='final_projects')
convert_pd_df_tosql('cleaned_dataset/category_translation.csv', 'category_translation', schema='final_projects')
```

After import successfully, I verified the data with SELECT * FROM table_name;

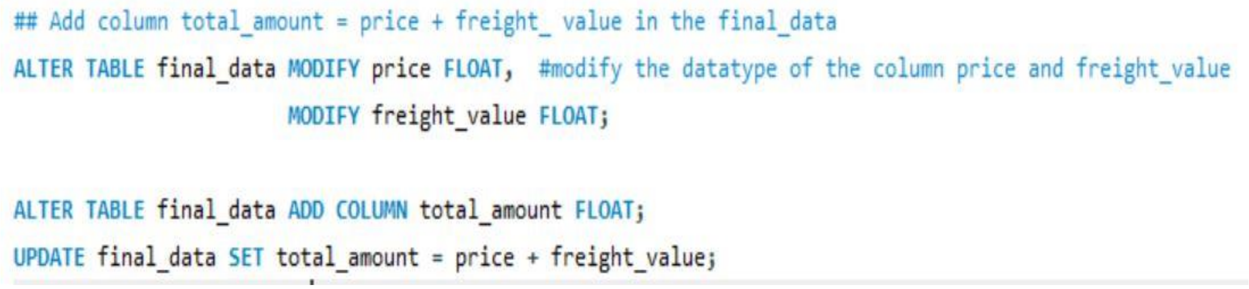
4. SQL Queries:

To facilitate analysis, I combined six tables into a single table through the process of joining.



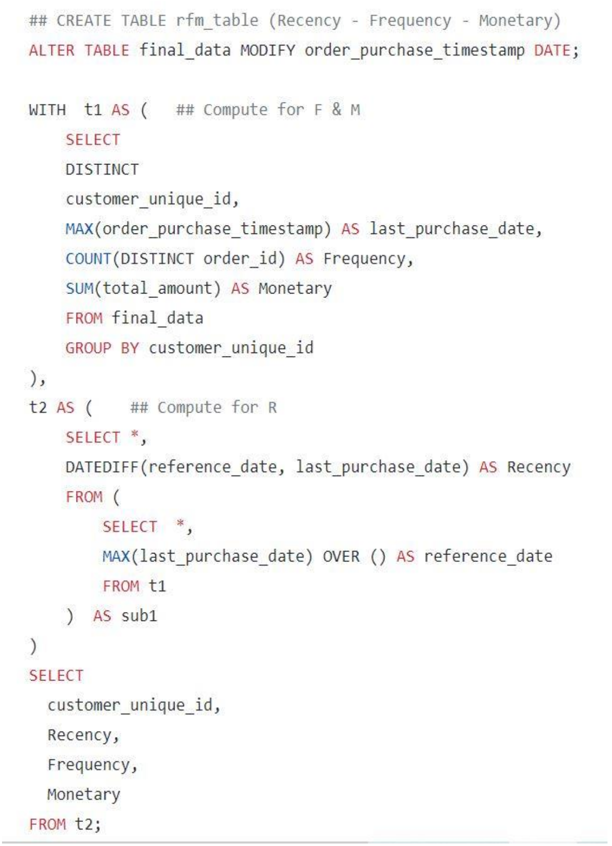
```sql
## Create the final_data by joining 6 table and remove the duplicates

CREATE TABLE final_data AS
SELECT DISTINCT * FROM
(SELECT c.customer_id, c.customer_unique_id, c.customer_city, c.customer_state, o.order_id,
        o.order_purchase_timestamp, i.order_item_id, i.product_id, i.price, i.freight_value,
        r.review_score, ct.product_category_name_english
FROM customers c
INNER JOIN orders o ON c.customer_id = o.customer_id
INNER JOIN items i ON o.order_id = i.order_id
INNER JOIN products p ON i.product_id = p.product_id
INNER JOIN reviews r ON i.order_id = r.order_id
INNER JOIN category_translation ct ON p.product_category_name = ct.product_category_name) data;
```

I included an additional column called "total_amount," which represents the overall cost of each order.

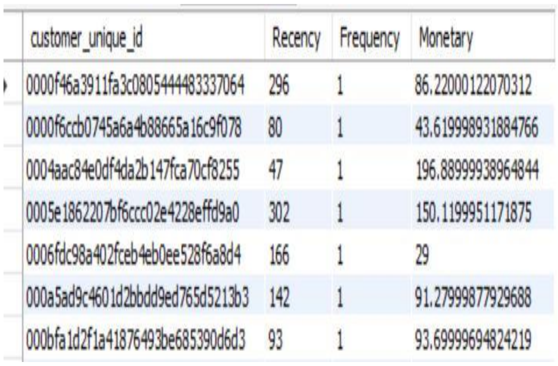


```sql
## Add column total_amount = price + freight_ value in the final_data
ALTER TABLE final_data MODIFY price FLOAT,  #modify the datatype of the column price and freight_value
                       MODIFY freight_value FLOAT;

ALTER TABLE final_data ADD COLUMN total_amount FLOAT;
UPDATE final_data SET total_amount = price + freight_value;
```

I calculated the RFM Model to prepare the data for the machine learning model.



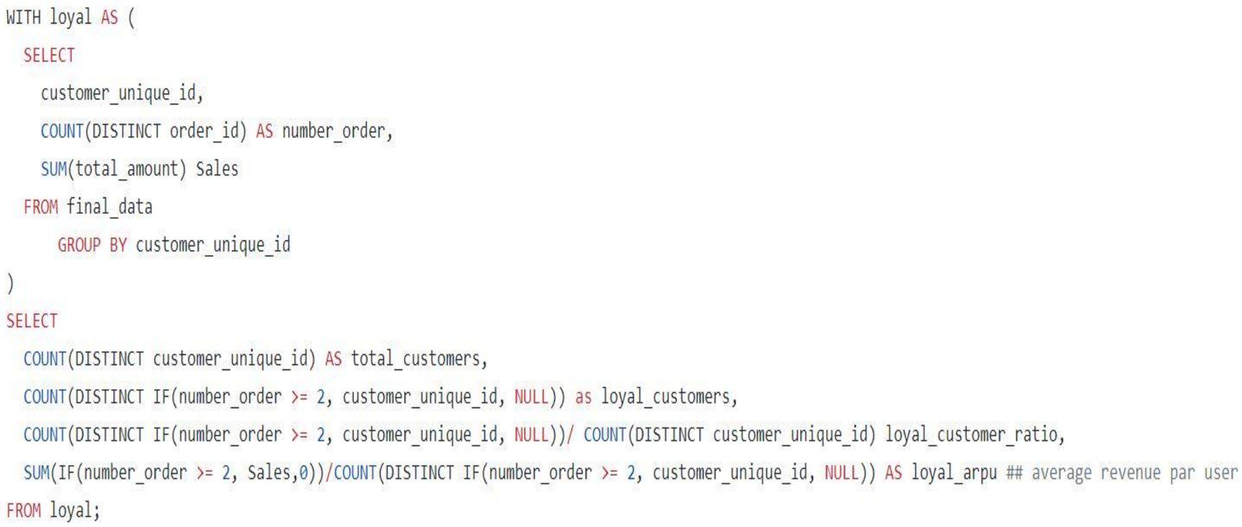
```
## CREATE TABLE rfm_table (Recency - Frequency - Monetary)
ALTER TABLE final_data MODIFY order_purchase_timestamp DATE;

WITH  t1 AS (    ## Compute for F & M
    SELECT
    DISTINCT
    customer_unique_id,
    MAX(order_purchase_timestamp) AS last_purchase_date,
    COUNT(DISTINCT order_id) AS Frequency,
    SUM(total_amount) AS Monetary
    FROM final_data
    GROUP BY customer_unique_id
),
t2 AS (     ## Compute for R
    SELECT *,
    DATEDIFF(reference_date, last_purchase_date) AS Recency
    FROM (
        SELECT  *,
        MAX(last_purchase_date) OVER () AS reference_date
        FROM t1
    )  AS sub1
)
SELECT
  customer_unique_id,
  Recency,
  Frequency,
  Monetary
FROM t2;
```



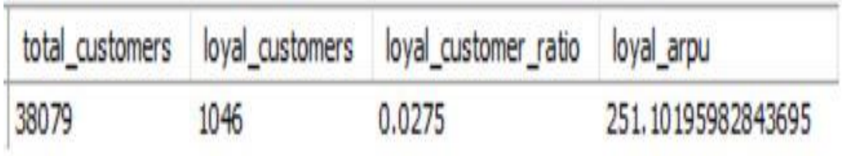
customer_unique_id	Recency	Frequency	Monetary
0000f46a3911fa3c0805444483337064	296	1	86.22000122070312
0000f6ccb0745a6a4b88665a16c9f078	80	1	43.619998931884766
0004aac84e0df4da2b147fca70cf8255	47	1	196.88999938964844
0005e1862207bf6ccc02e4228effd9a0	302	1	150.1199951171875
0006fdc98a402fceb4eb0ee528f6a8d4	166	1	29
000a5ad9c4601d2bbdd9ed765d5213b3	142	1	91.27999877929688
000bfa1d2f1a41876493be685390d6d3	93	1	93.69999694824219

Then, I create some queries to get some insights.

- Calculate Loyal Customer Ratio:



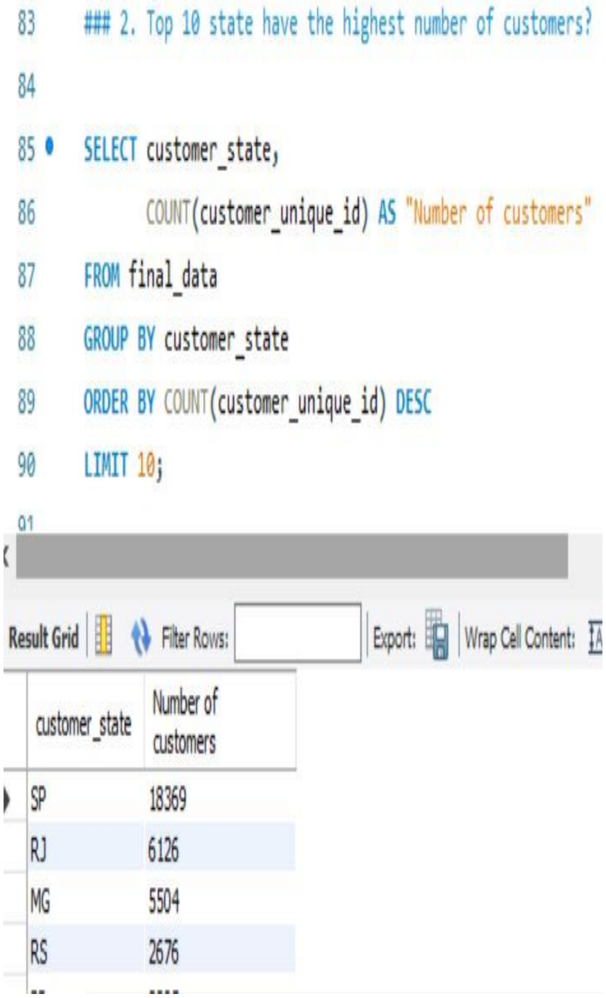
```
WITH loyal AS (
  SELECT
    customer_unique_id,
    COUNT(DISTINCT order_id) AS number_order,
    SUM(total_amount) Sales
  FROM final_data
      GROUP BY customer_unique_id
)
SELECT
  COUNT(DISTINCT customer_unique_id) AS total_customers,
  COUNT(DISTINCT IF(number_order >= 2, customer_unique_id, NULL)) as loyal_customers,
  COUNT(DISTINCT IF(number_order >= 2, customer_unique_id, NULL))/ COUNT(DISTINCT customer_unique_id) loyal_customer_ratio,
  SUM(IF(number_order >= 2, Sales,0))/COUNT(DISTINCT IF(number_order >= 2, customer_unique_id, NULL)) AS loyal_arpu ## average revenue par user
FROM loyal;
```



total_customers	loyal_customers	loyal_customer_ratio	loyal_arpu
38079	1046	0.0275	251.10195982843695

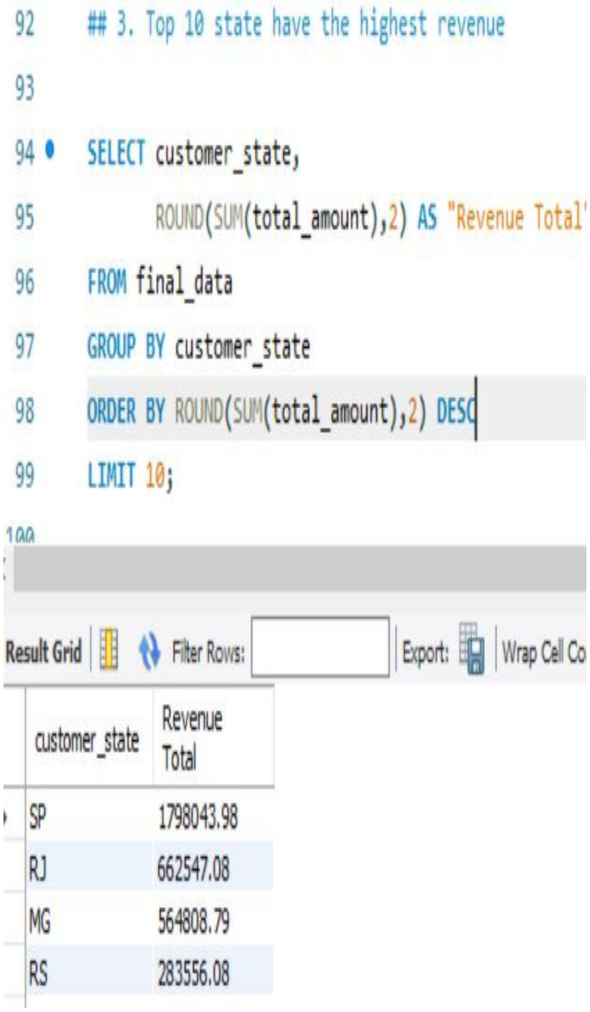
The loyal customer ratio of 2.75% indicates the percentage of customers who have made repeated purchases or have shown loyalty towards the business. This ratio suggests that a small fraction of the customer base is loyal to the business, and there may be a need to focus on strategies to increase customer loyalty and retention.

- Top 10 state have the highest number of customers and revenue.



```
### 2. Top 10 state have the highest number of customers?

SELECT customer_state,
        COUNT(customer_unique_id) AS "Number of customers"
FROM final_data
GROUP BY customer_state
ORDER BY COUNT(customer_unique_id) DESC
LIMIT 10;
```

customer_state	Number of customers
SP	18369
RJ	6126
MG	5504
RS	2676

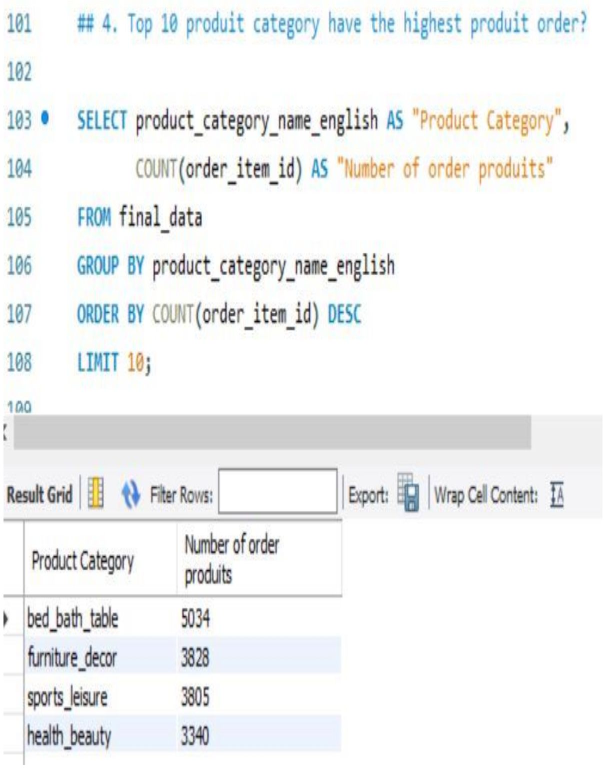


```
## 3. Top 10 state have the highest revenue

SELECT customer_state,
        ROUND(SUM(total_amount),2) AS "Revenue Total'
FROM final_data
GROUP BY customer_state
ORDER BY ROUND(SUM(total_amount),2) DESC
LIMIT 10;
```

customer_state	Revenue Total
SP	1798043.98
RJ	662547.08
MG	564808.79
RS	283556.08

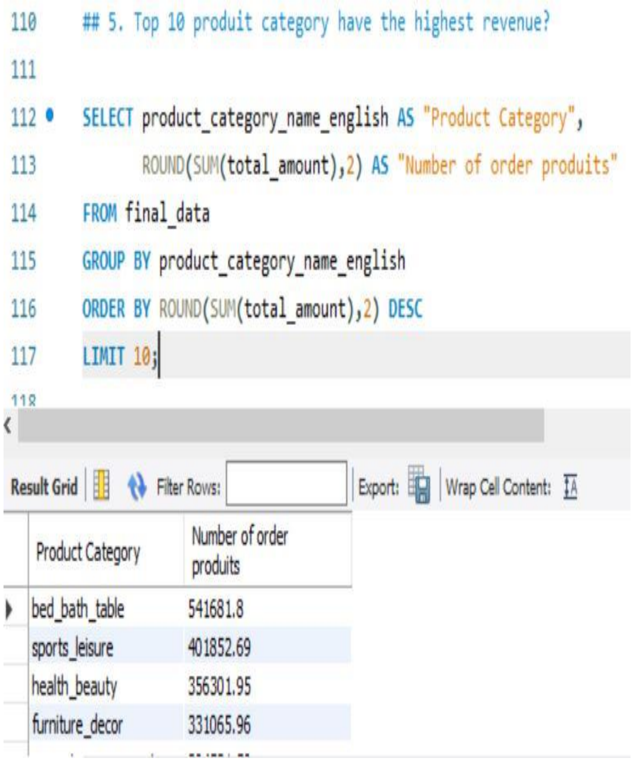
Sao Paolo and Rio de Janeiro have the highest number of customers and generate the most revenue, likely because they are the two biggest states in Brazil.

- Top 10 product category have the highest product order and revenue.



```
## 4. Top 10 produit category have the highest produit order?

SELECT product_category_name_english AS "Product Category",
        COUNT(order_item_id) AS "Number of order produits"
FROM final_data
GROUP BY product_category_name_english
ORDER BY COUNT(order_item_id) DESC
LIMIT 10;
```

Product Category	Number of order produits
bed_bath_table	5034
furniture_decor	3828
sports_leisure	3805
health_beauty	3340

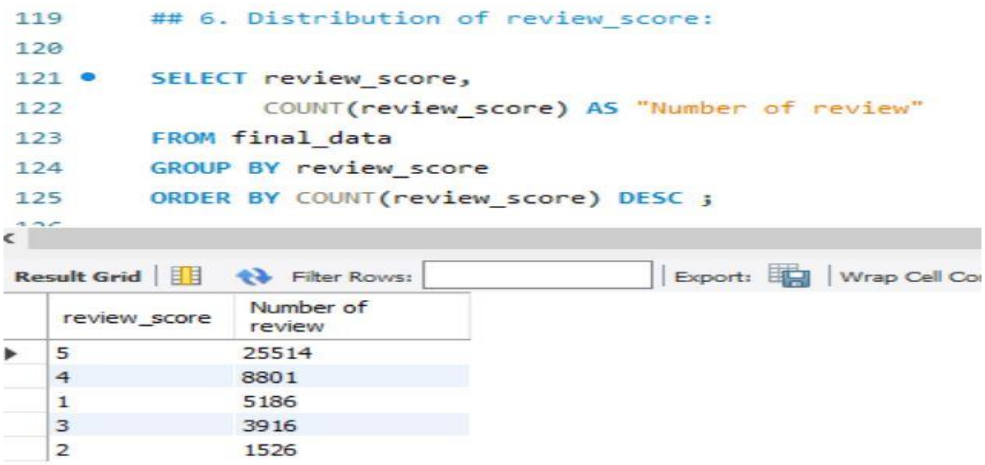


```
## 5. Top 10 produit category have the highest revenue?

SELECT product_category_name_english AS "Product Category",
        ROUND(SUM(total_amount),2) AS "Number of order produits"
FROM final_data
GROUP BY product_category_name_english
ORDER BY ROUND(SUM(total_amount),2) DESC
LIMIT 10;
```

Product Category	Number of order produits
bed_bath_table	541681.8
sports_leisure	401852.69
health_beauty	356301.95
furniture_decor	331065.96

Based on my results, I have found that the products that are purchased the most have a usage lifespan longer than one year. This may be one of the reasons why the customer retention rate is low within a year. To better analyze this index, we need to have a longer period of analysis, for example, two years.

- Distribution of the review_score



```
## 6. Distribution of review_score:

SELECT review_score,
        COUNT(review_score) AS "Number of review"
FROM final_data
GROUP BY review_score
ORDER BY COUNT(review_score) DESC ;
```

review_score	Number of review
5	25514
4	8801
1	5186
3	3916
2	1526

The fact that a company has mostly positive reviews (4/5 and 5/5) but the loyal customer ratio is low, so the reviews may not translate to customer loyalty. Positive reviews could be a result of several factors such as quality of products, prompt delivery, user experience, and customer service.

Moreover, the loyal customer ratio of 2.75% suggests that only a small fraction of customers is loyal to the company, which may be due to several reasons such as a lack of personalized customer experience, competitors offering better deals or services, or a lack of targeted marketing efforts towards building customer loyalty.

The company must increase customer loyalty by providing exceptional customer service, personalized experiences, and relevant marketing.

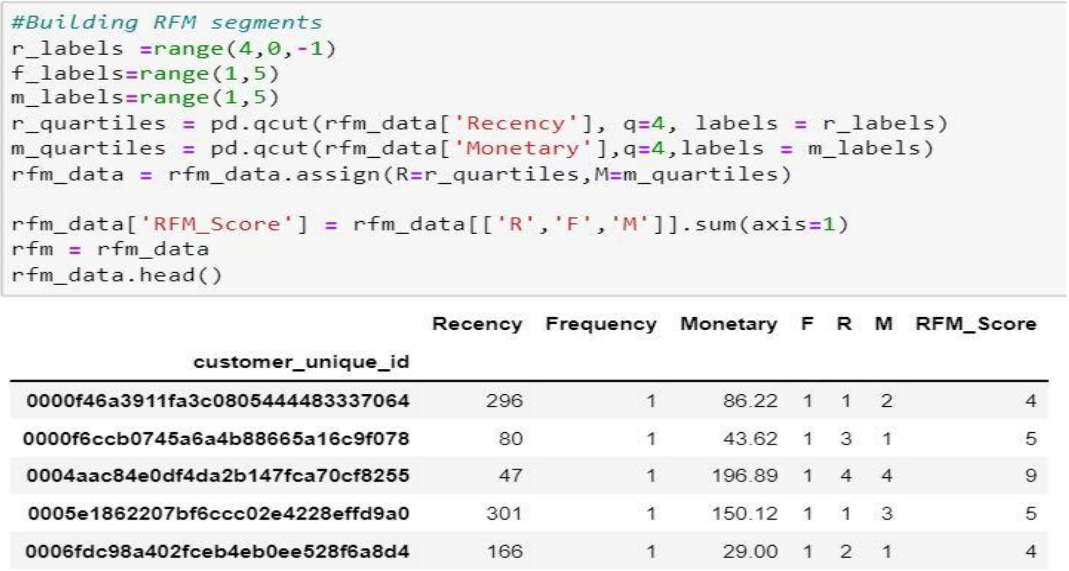
VI. Machine Learning Model

I used machine learning models to gain further insights into customer behavior and predict customer lifetime value (CLV).

1. Unsupervised Learning Model:

I used a K-means clustering model with k=4 to cluster customers based on their RFM metric (Recency - Frequency - Monetary) and gain insights into different customer segments.

Step 1: Build the RFM Metric and RFM Score



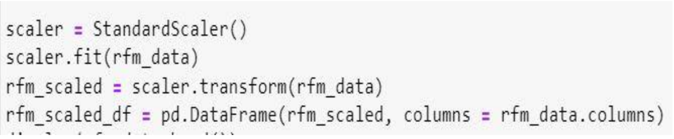
```
#Building RFM segments
r_labels =range(4,0,-1)
f_labels=range(1,5)
m_labels=range(1,5)
r_quartiles = pd.qcut(rfm_data['Recency'], q=4, labels = r_labels)
m_quartiles = pd.qcut(rfm_data['Monetary'],q=4,labels = m_labels)
rfm_data = rfm_data.assign(R=r_quartiles,M=m_quartiles)

rfm_data['RFM_Score'] = rfm_data[['R','F','M']].sum(axis=1)
rfm = rfm_data
rfm_data.head()
```

	Recency	Frequency	Monetary	F	R	M	RFM_Score
customer_unique_id							
0000f46a3911fa3c0805444483337064	296	1	86.22	1	1	2	4
0000f6ccb0745a6a4b88665a16c9f078	80	1	43.62	1	3	1	5
0004aac84e0df4da2b147fca70cf8255	47	1	196.89	1	4	4	9
0005e1862207bf6ccc02e4228effd9a0	301	1	150.12	1	1	3	5
0006fdc98a402fceb4eb0ee528f6a8d4	166	1	29.00	1	2	1	4

It's the data for my machine learning model.

Step 2 : Cleaning the data rfm_data

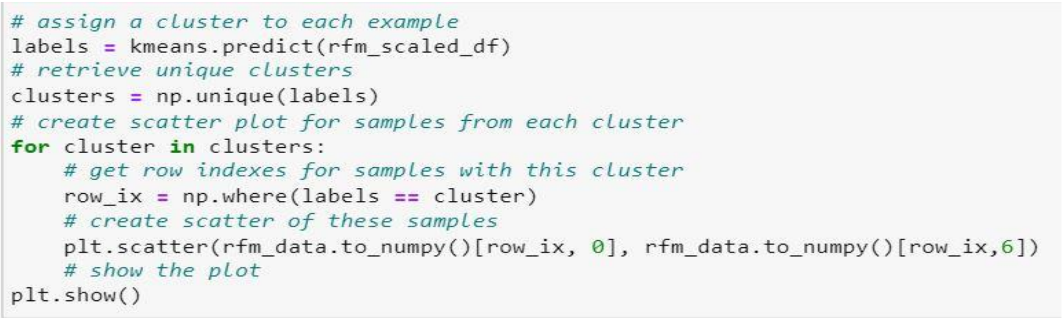
- Remove the outliers
- Remove the duplicates



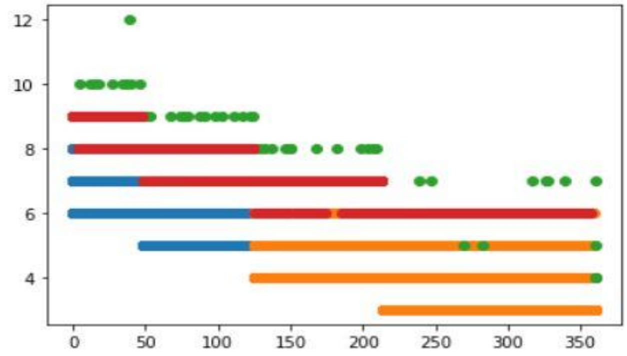
```
scaler = StandardScaler()
scaler.fit(rfm_data)
rfm_scaled = scaler.transform(rfm_data)
rfm_scaled_df = pd.DataFrame(rfm_scaled, columns = rfm_data.columns)
```

Step 3: Standardize the data using standardscaler

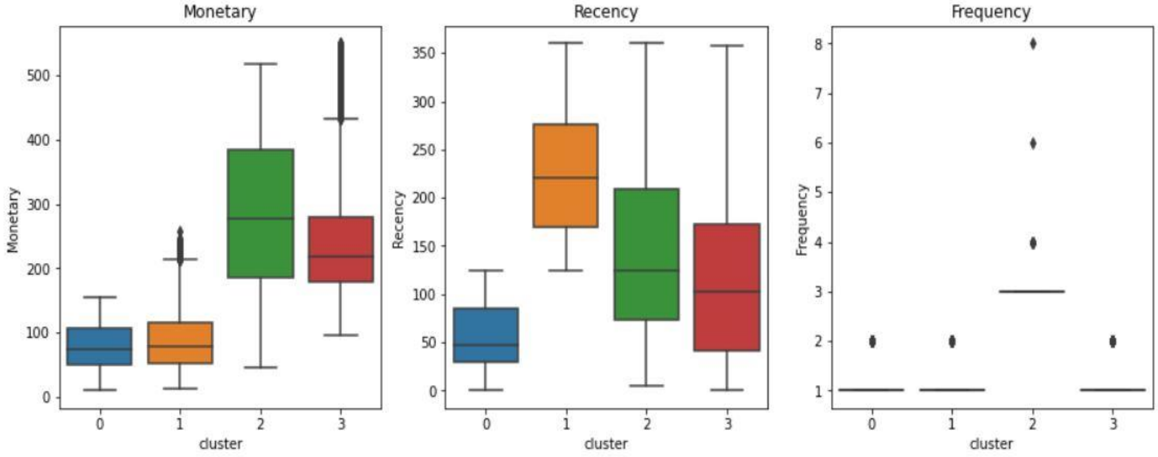
Step 4: Find the k optimal and cluster customer with k = 4



```
# assign a cluster to each example
labels = kmeans.predict(rfm_scaled_df)
# retrieve unique clusters
clusters = np.unique(labels)
# create scatter plot for samples from each cluster
for cluster in clusters:
    # get row indexes for samples with this cluster
    row_ix = np.where(labels == cluster)
    # create scatter of these samples
    plt.scatter(rfm_data.to_numpy()[row_ix, 0], rfm_data.to_numpy()[row_ix,6])
    # show the plot
plt.show()
```

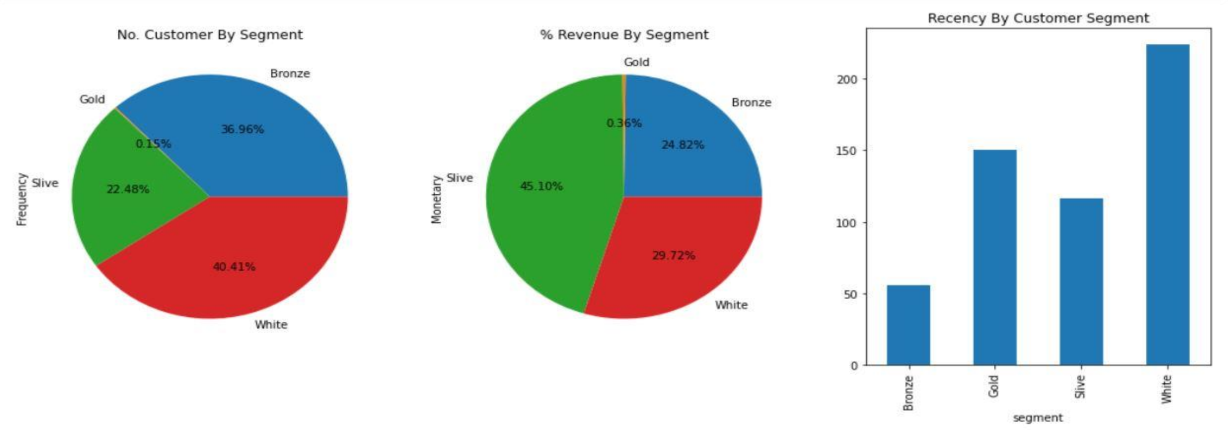


Step 5: Use boxplot to understand the behavior of customers



Step 6: Segment customers and insights

- Cluster 0: Bronze
- Cluster 1: White
- Cluster 2: Gold
- Cluster 3: Slive



One of the key customer segments identified in the Olist e-commerce project is the segment "sliver". This segment is comprised of customers who exhibit average spending and average time between purchases. Although the silver segment represents only 22.48% of the total customer base, it generates the highest profitability for the business.

The segment "white" accounts for the majority of customers, but generates low profits due to their long time since last purchase. Olist need to develop appropriate strategies such as promotions to encourage them to make purchases again.

The segment "gold" represents a small proportion, but they are the customers who are willing to spend the most, with the average time between purchases being moderate. Businesses can have policies such as loyal customer programs and offer suitable discounts to increase the number of customers in this segment.

Frequency charts can be used to analyze the purchasing habits of customers, allowing businesses to identify patterns and trends that can be used to encourage more purchases. For example, if a customer hasn't made a purchase in a while, a targeted promotion or personalized communication could be sent to encourage them to return and make a purchase.

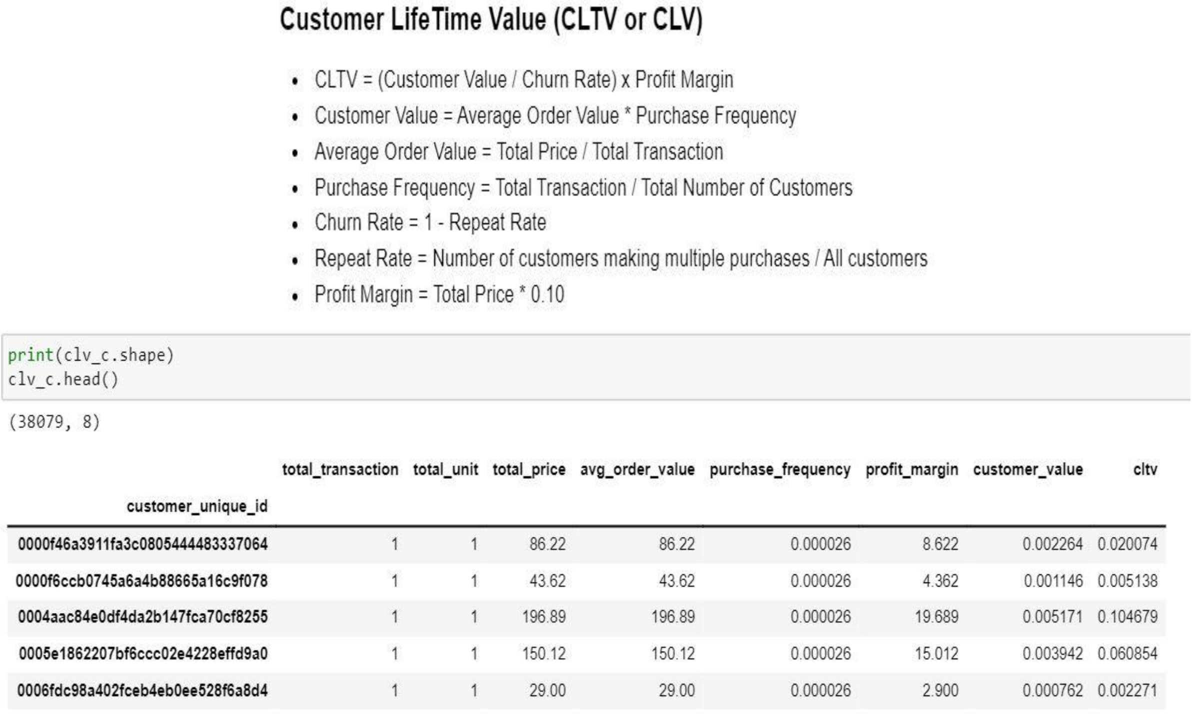
Average spending provides insight into how much customers are willing to spend on products or services and can be used to suggest suitable products at different price points. By recommending products that fit within a customer's spending range, businesses can increase the likelihood of repeat purchases and strengthen customer loyalty.

Overall, by leveraging customer segmentation, businesses can develop effective retention strategies that resonate with their customers and drive repeat purchases, ultimately leading to increased revenue and profitability.

2. Supervised Learning Model:

I used this model to predict the customer lifetime value. The purpose is to estimate the total amount of revenue a customer will generate for a business over their lifetime. By calculating CLV, businesses can make informed decisions about how much they should invest in acquiring and retaining customers, and how much they can expect to earn from each customer. It is a valuable tool for businesses to allocate resources to maximize revenue and profitability.

Step 1: Build the data for the Model:



Customer LifeTime Value (CLTV or CLV)

- CLTV = (Customer Value / Churn Rate) x Profit Margin
- Customer Value = Average Order Value * Purchase Frequency
- Average Order Value = Total Price / Total Transaction
- Purchase Frequency = Total Transaction / Total Number of Customers
- Churn Rate = 1 - Repeat Rate
- Repeat Rate = Number of customers making multiple purchases / All customers
- Profit Margin = Total Price * 0.10

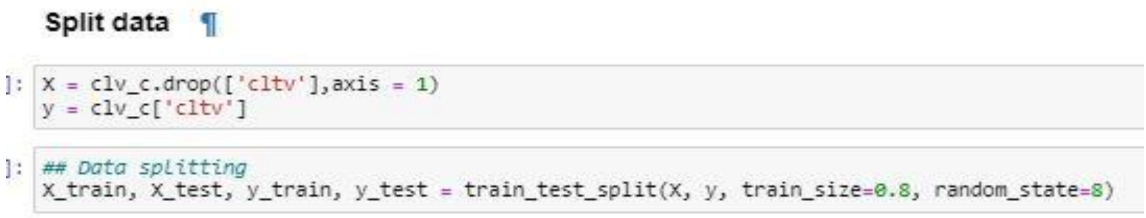
```
print(clv_c.shape)
clv_c.head()
```

```
(38079, 8)
```

	total_transaction	total_unit	total_price	avg_order_value	purchase_frequency	profit_margin	customer_value	cltv
customer_unique_id								
0000f46a3911fa3c0805444483337064	1	1	86.22	86.22	0.000026	8.622	0.002264	0.020074
0000f6ccb0745a6a4b88665a16c9f078	1	1	43.62	43.62	0.000026	4.362	0.001146	0.005138
0004aac84e0df4da2b147fca70cf8255	1	1	196.89	196.89	0.000026	19.689	0.005171	0.104679
0005e1862207bf6ccc02e4228effd9a0	1	1	150.12	150.12	0.000026	15.012	0.003942	0.060854
0006fdc98a402fceb4eb0ee528f6a8d4	1	1	29.00	29.00	0.000026	2.900	0.000762	0.002271

The data have 38079 rows and 8 columns

Step 2: Split the data and fit the model training

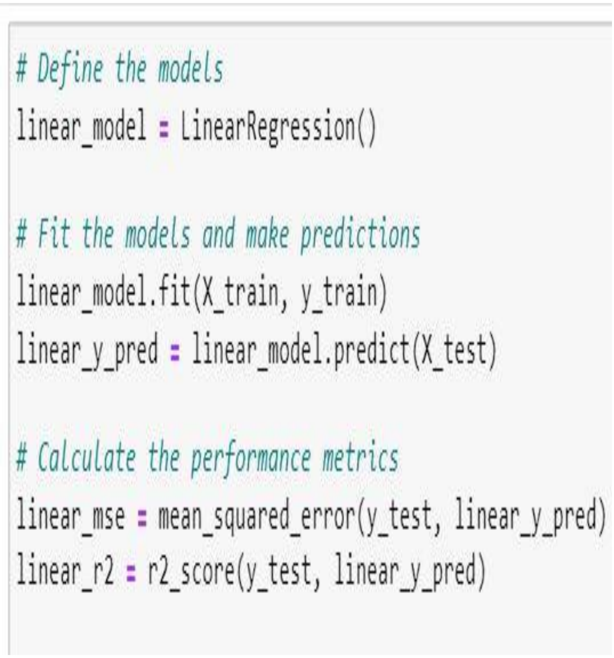


Split data ¶

```
]: X = clv_c.drop(['cltv'],axis = 1)
   y = clv_c['cltv']

]: ## Data splitting
   X_train, X_test, y_train, y_test = train_test_split(X, y, train_size=0.8, random_state=8)
```

I divided the dataset into train and test sets, with 80% of the data used for training and 20% for testing. I then used various supervised machine learning models to predict the customer lifetime value (CLV):

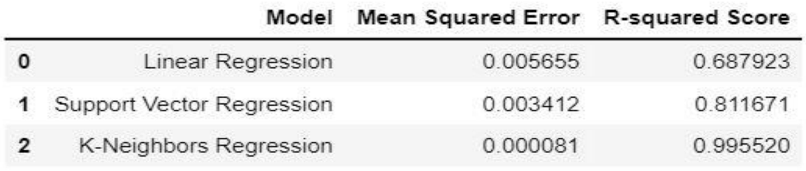
- Linear Regression: find the best-fit line that can describe the relationship between the independent variables and the dependent variable.
- Support vector regression (SVR): find a function that best fits the training data.
- K-Neighbors Regression (KNN): find the K-nearest neighbors to a data point in the training set and using their average value as the predicted value for the new data point.



```
# Define the models
linear_model = LinearRegression()

# Fit the models and make predictions
linear_model.fit(X_train, y_train)
linear_y_pred = linear_model.predict(X_test)

# Calculate the performance metrics
linear_mse = mean_squared_error(y_test, linear_y_pred)
linear_r2 = r2_score(y_test, linear_y_pred)
```

Step 3: Compare the performance of the models

By comparing the performance of these models with Mean Squared Error R-squared Score , I was able to select the best-fitted model for our dataset.



	Model	Mean Squared Error	R-squared Score
0	Linear Regression	0.005655	0.687923
1	Support Vector Regression	0.003412	0.811671
2	K-Neighbors Regression	0.000081	0.995520

My analysis shows that K-Neighbors Regression is the best model for predicting customer lifetime value (CLV). This is based on two key metrics - the Mean Squared Error (MSE) and the R-squared score. The MSE measures the average difference between the predicted CLV and the actual CLV for each customer in the test dataset, while the R-squared score measures the percentage of variance in the CLV that is explained by the model.

K-Neighbors Regression had the lowest MSE and highest R-squared score compared to other models, indicating that it is the most accurate and reliable model for predicting CLV. This suggests that K-Neighbors Regression is able to capture the underlying patterns and trends in customer data to accurately estimate the total amount of revenue a customer will generate for a business over their lifetime.

VII. Conclusion

Overall, it's clear that improving customer retention should be a priority for Olist. By addressing any issues with customer satisfaction or loyalty and implementing targeted retention strategies, Olist can increase the number of returning customers and improve its overall performance as an e-commerce platform.

It's important for businesses to understand the reasons behind low customer retention rates to identify areas for improvement and take appropriate actions to increase customer loyalty.

It's also important to analyze the customer data to identify any patterns or trends in customer behavior. Cohort analysis, for example, can help to understand how customer retention changes over time and identify any specific cohorts that may have a higher likelihood of returning to Olist. This information can then be used to develop targeted retention strategies for different customer segments.

Another strategy could be to offer incentives for returning customers, such as discounts or loyalty programs. This could encourage customers to return and make repeat purchases, increasing the overall retention rate.

Links:
Github: https://github.com/yenle2092/Final_Project_Customer_Retention_E-Commerce.git

Trello: https://trello.com/b/0eXVBfhA/finalproject-customer-retention