Kingdom of Saudi Arabia
Ministry of Education
Umm Al-Qura University
031

المملكة العربية السعودية
وزارة التعليم
جامعة أم القرى
٠٣١

Name	id
Nora Tarek al-suhaibi	444002730
Raya abdullah alsaab	444000732

Task2 Report

Title: Brazilian E-Commerce Public Dataset by Olist

The List Brazilian E-Commerce Dataset captures details of 100,000 orders from 2016 to 2018, including payment, shipping, customer reviews, and regional data, all anonymized for privacy. This dataset is ideal for analyzing customer behavior, logistics, and regional trends in Brazil, and can be combined with marketing data for a full customer journey view.

Introduction :

The Brazilian E-Commerce Public Dataset by List provides insights into 100,000 orders made between 2016 and 2018. It covers aspects such as order status, pricing, shipping, customer locations, and reviews, with anonymized data for privacy protection. This dataset can also be combined with Marketing Funnel data for an expanded analysis.

Objectives:

•Increase Sales and Cross-Selling Opportunities:

By identifying products that are often purchased together, businesses can create targeted promotions, bundles, or recommendations to encourage customers to buy more.

• Improve Product Placement and Store Layout:

Understanding item associations can help optimize the physical arrangement of products in stores or the layout of e-commerce websites to make related products more accessible, thus increasing the likelihood of additional purchases.

•Enhance Marketing Strategies:

Market basket analysis can help segment customer based on their buying habits, allowing businesses to create personalized marketing campaigns that appeal to different customer groups.

•Inventory Management:

It provides insights into stocking strategies by identifying high-demand product combinations, helping businesses better manage their inventory levels.

•Identify Customer Behavior Patterns:

It helps in understanding how customers behave during shopping, such as seasonal trends or time based purchasing habits, which can inform long term business strategies.

Importing Libraries:

The following libraries are imported for data manipulation, visualization, and analysis:

numpy and pandas: For data manipulation.

matplotlib.pyplot and seaborn: For visualizing data.

warnings: To ignore warning messages.

Matplotlib's figure size is set to [10,5] for consistent plot dimensions.

Loading the Datasets:

Multiple CSV files are loaded into separate pandas DataFrames, including datasets related to customers, geolocation, orders, reviews, sellers, products, and payments.

Merging the Datasets:

A series of DataFrame merges are performed to consolidate the information from multiple datasets into a single DataFrame:

df1: Merges olist_orders with olist_order_payments.

df2: Merges df1 with olist_customers.

df3: Merges df2 with olist_order_reviews.

df4: Merges df3 with olist_order_items.

df5: Merges df4 with olist_products.

df6: Merges df5 with olist_sellers.

The final merged DataFrame, named df, contains data from all the source datasets, providing a complete view of the order information.

Handling Missing Values:

Missing values in the columns related to product dimensions (product_weight_g, product_length_cm, product_height_cm, product_width_cm) are replaced with the mean of the respective columns.

Handling Date Columns:

Several columns with date and time information are converted to datetime format to facilitate date manipulation and analysis.

Dropping Unnecessary Columns:

The columns customer_zip_code_prefix, seller_zip_code_prefix, review_creation_date, review_answer_timestamp, and shipping_limit_date are dropped from the DataFrame as they may not be necessary for analysis.

Market Basket Analysis Setup:

The data is transformed to prepare for market basket analysis:

Grouping is done by order_id and product_category_name, then counts are pivoted using unstack(), resulting in a matrix format where rows are orders and columns are product categories.

Each value in the matrix is replaced with 1 if the count is greater than 0, indicating the presence of the product category in the order, otherwise replaced with 0.

Market Basket Analysis:

The apriori algorithm from the mlxtend library is used to find frequent itemsets:

min_support=0.01: Only itemsets appearing in at least 1% of transactions are considered.

Identified Problems and Recommendations:

Download data:

While downloading the data, it did not download with me, because there was data applied by a system other than the normal system.

Solution: I used encoding to make sure that the data is read with proper encryption and supports all characters, also “utf-8” was chosen because it can represent most of the letters and symbols used, which is the most appropriate choice and very secure and can handle text data.

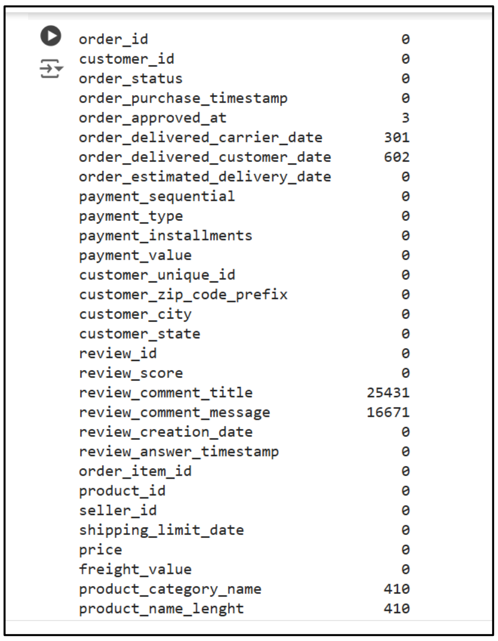
Null values:

It appeared to us that there are null values in the data I am working on and they should be discarded.

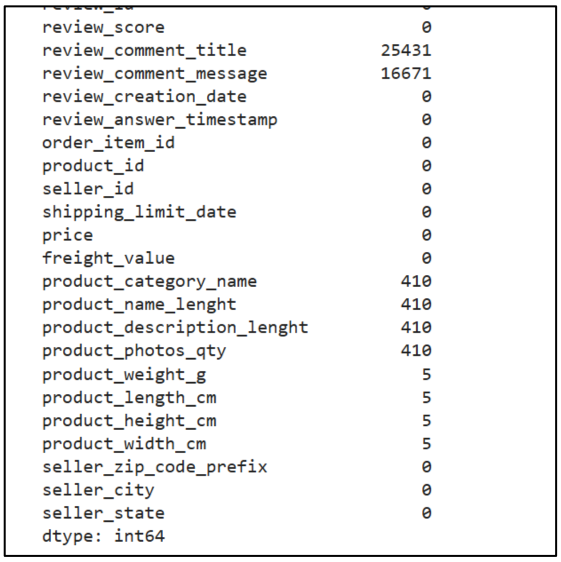
Solution: We must delete each existing null value , and put in its place the calculation of the average for the column in which there are null values.

output about cood:

Null values:



```
order_id                          0
customer_id                       0
order_status                      0
order_purchase_timestamp          0
order_approved_at                 3
order_delivered_carrier_date    301
order_delivered_customer_date   602
order_estimated_delivery_date     0
payment_sequential                0
payment_type                      0
payment_installments              0
payment_value                     0
customer_unique_id                0
customer_zip_code_prefix          0
customer_city                     0
customer_state                    0
review_id                         0
review_score                      0
review_comment_title          25431
review_comment_message        16671
review_creation_date              0
review_answer_timestamp           0
order_item_id                     0
product_id                        0
seller_id                         0
shipping_limit_date               0
price                             0
freight_value                     0
product_category_name           410
product_name_lenght             410
```



```
review_score                      0
review_comment_title          25431
review_comment_message        16671
review_creation_date              0
review_answer_timestamp           0
order_item_id                     0
product_id                        0
seller_id                         0
shipping_limit_date               0
price                             0
freight_value                     0
product_category_name           410
product_name_lenght             410
product_description_lenght      410
product_photos_qty              410
product_weight_g                  5
product_length_cm                 5
product_height_cm                 5
product_width_cm                  5
seller_zip_code_prefix            0
seller_city                       0
seller_state                      0
dtype: int64
```

We must delete each existing empty value , and put in its place the calculation of the average for the column in which there are null values, and so we get rid of the null values.

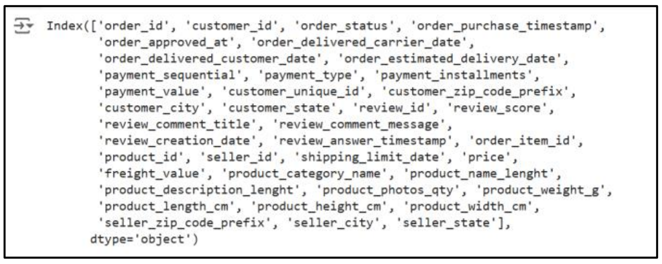
Rows Recurring:

We used the duplicated () function to see if there are duplicate rows or not, and it turned out that there are no duplicate rows.

```
0
```

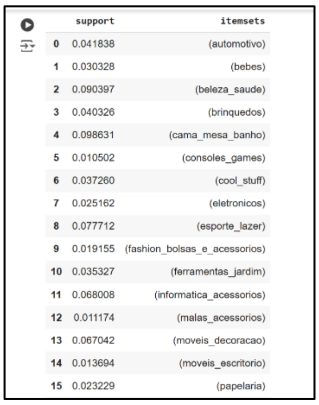
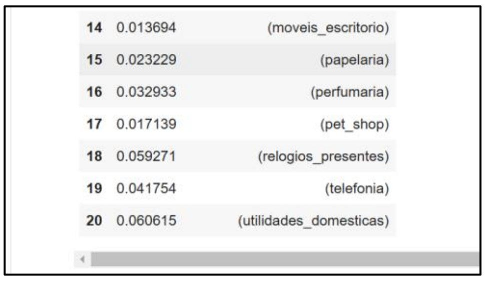
Columns used in the data:

We used the columns function to show me all the columns used in the data I was working on.



```
Index(['order_id', 'customer_id', 'order_status', 'order_purchase_timestamp',
       'order_approved_at', 'order_delivered_carrier_date',
       'order_delivered_customer_date', 'order_estimated_delivery_date',
       'payment_sequential', 'payment_type', 'payment_installments',
       'payment_value', 'customer_unique_id', 'customer_zip_code_prefix',
       'customer_city', 'customer_state', 'review_id', 'review_score',
       'review_comment_title', 'review_comment_message',
       'review_creation_date', 'review_answer_timestamp', 'order_item_id',
       'product_id', 'seller_id', 'shipping_limit_date', 'price',
       'freight_value', 'product_category_name', 'product_name_lenght',
       'product_description_lenght', 'product_photos_qty', 'product_weight_g',
       'product_length_cm', 'product_height_cm', 'product_width_cm',
       'seller_zip_code_prefix', 'seller_city', 'seller_state'],
      dtype='object')
```

Support – itemsets:

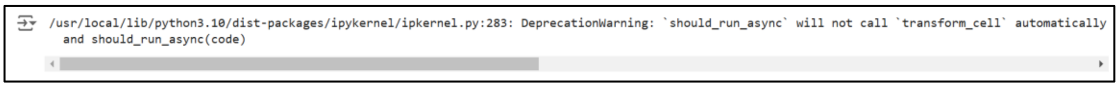



	support	itemsets
0	0.041838	(automotivo)
1	0.030328	(bebes)
2	0.090397	(beleza_saude)
3	0.040326	(brinquedos)
4	0.098631	(cama_mesa_banho)
5	0.010502	(consoles_games)
6	0.037260	(cool_stuff)
7	0.025162	(eletronicos)
8	0.077712	(esporte_lazer)
9	0.019155	(fashion_bolsas_e_acessorios)
10	0.035327	(ferramentas_jardim)
11	0.068008	(informatica_acessorios)
12	0.011174	(malas_acessorios)
13	0.067042	(moveis_decoracao)
14	0.013694	(moveis_escritorio)
15	0.023229	(papelaria)
16	0.032933	(perfumaria)
17	0.017139	(pet_shop)
18	0.059271	(relogios_presentes)
19	0.041754	(telefonia)
20	0.060615	(utilidades_domesticas)

Support: Represents the percentage of orders per product group.

itemsets: Contains the category name of the products that are repeated in the student data.

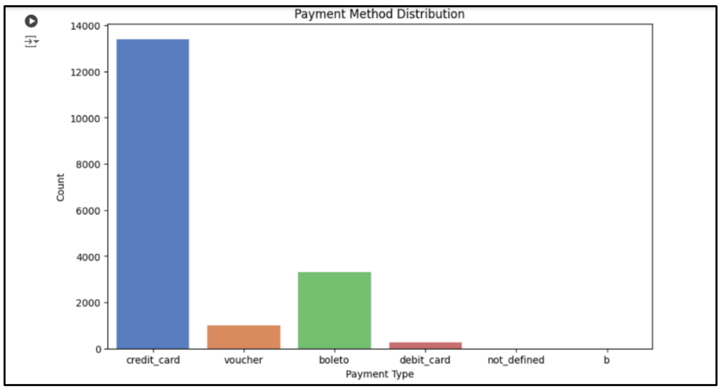
Warning message:



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

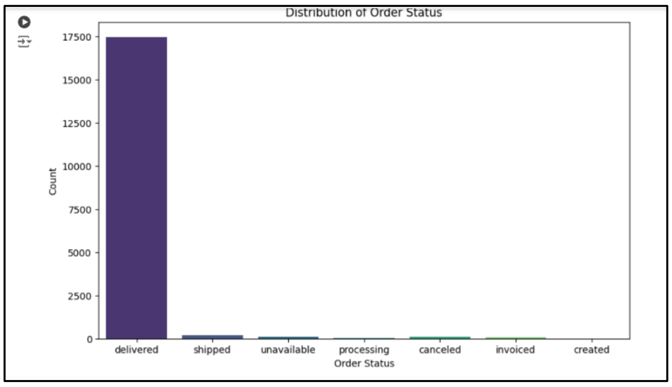
A warning message has appeared to us and indicates future updates in how Python deals with code that contains asynchronous operations.

Payment Method Distribution:



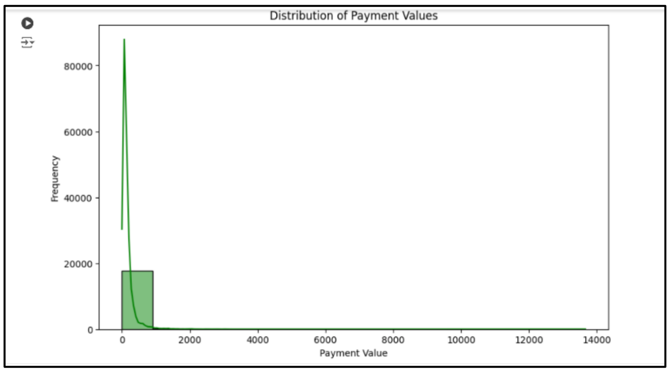
Credit card is greater in payment method.

Distribution of Order Status:



Delivered is greater in order status.

Distribution is Payment Values:



Payment is greater in range between 0 to 1000.

References:

Link about data:

https://www.kaggle.com/datasets/olistbr/brazilian-ecommerce?select=product_category_name_translation.csv