Power BI Project: Olist E-commerce Dataset Analysis

Objective:

Develop a comprehensive Power BI report that provides valuable insights into Olist's e-commerce performance, addressing key business questions.

Olist Data Source

- **Provider**: Olist
- **Dataset Name**: Olist Store Dataset
- **Source**: https://www.kaggle.com/datasets/olistbr/brazilian-ecommerce/data

License

The dataset is available under the **Open Database License (ODbL)**, which allows you to:

- Share: Copy and redistribute the material in any medium or format.
- Adapt: Remix, transform, and build upon the material for any purpose, even commercially.

The Olist dataset is a comprehensive collection of data from Olist, a Brazilian e-commerce marketplace. It contains several tables, each providing insights into different aspects of the business operations, including orders, products, customers, payments, reviews, and logistics. Here's a brief explanation of the key tables and their primary columns:

1. ORDERS TABLE (`OLIST_ORDERS_DATASET`)

Columns:

- `order_id`: Unique identifier for each order.
- `customer_id`: Unique identifier for each customer.
- `order_status`: Current status of the order (e.g., delivered, shipped, canceled).
- `order_purchase_timestamp`: Timestamp when the order was placed.
- `order_approved_at`: Timestamp when the order was approved.
- `order_delivered_carrier_date`: Date when the order was handed to the carrier.
- `order_delivered_customer_date`: Date when the order was delivered to the customer.
- `order_estimated_delivery_date`: Estimated delivery date for the order.

2. ORDER ITEMS TABLE (`OLIST_ORDER_ITEMS_DATASET`)

Columns:

- `order_id`: Unique identifier for each order.
- `order_item_id`: Identifier for each item in an order.

- `product_id`: Unique identifier for each product.
- `seller_id`: Unique identifier for each seller.
- `shipping_limit_date`: Latest date by which the seller must ship the item.
- `price`: Price of the item.
- `freight_value`: Freight cost for the item.

3. CUSTOMERS TABLE (`OLIST_CUSTOMERS_DATASET`)

Columns:

- `customer_id`: Unique identifier for each customer.
- `customer_unique_id`: Unique identifier that groups multiple orders from the same customer.
- `customer_zip_code_prefix`: First five digits of the customer's zip code.
- `customer_city`: City of the customer.
- `customer_state`: State of the customer.

4. SELLERS TABLE (`OLIST_SELLERS_DATASET`)

Columns:

- `seller_id`: Unique identifier for each seller.
- `seller_zip_code_prefix`: First five digits of the seller's zip code.
- `seller_city`: City of the seller.
- `seller_state`: State of the seller.

5. PRODUCTS TABLE (`OLIST_PRODUCTS_DATASET`)

Columns:

- `product_id`: Unique identifier for each product.
- `product_category_name`: Category name of the product.
- `product_name_lenght`: Length of the product name.
- `product_description_lenght`: Length of the product description.
- `product_photos_qty`: Number of photos for the product.
- `product_weight_g`: Weight of the product in grams.
- `product_length_cm`: Length of the product in centimeters.
- `product_height_cm`: Height of the product in centimeters.
- `product_width_cm`: Width of the product in centimeters.

6. PAYMENTS TABLE (`OLIST_ORDER_PAYMENTS_DATASET`)

Columns:

- `order_id`: Unique identifier for each order.
- `payment_sequential`: Sequential number for the payment.
- `payment_type`: Type of payment (e.g., credit card, boleto).
- `payment_installments`: Number of payment installments.
- `payment_value`: Value of the payment.

7. REVIEWS TABLE (OLIST_ORDER_REVIEWS_DATASET)

Columns:

- review_id: Unique identifier for each review.
- order_id: Unique identifier for each order.
- review_score: Score given by the customer (1-5).
- review_comment_title: Title of the review comment.
- review_comment_message: Message of the review comment.
- review_creation_date: Date when the review was created.
- review_answer_timestamp: Timestamp when the review was answered.

8. GEOLOCATION TABLE (OLIST_GEOLOCATION_DATASET)

Columns:

- geolocation_zip_code_prefix: First five digits of the zip code.
- geolocation_lat: Latitude coordinate.
- geolocation_lng: Longitude coordinate.
- geolocation_city: City corresponding to the geolocation.
- geolocation_state: State corresponding to the geolocation.

These tables can be joined using common identifiers like order_id, customer_id, product_id, and seller_id to create a comprehensive view of the e-commerce operations and analyze various aspects such as sales performance, customer behavior, and delivery logistics.

TARGET AUDIENCE FOR POWER BI REPORTS:

- **Olist Higher Management**: The reports aim to provide a comprehensive overview of the company's financial and commercial performance.
- **Marketing Team**: To analyze product performance and understand customer preferences.
- **Logistic Team**: To improve their operations and make informed decisions based on the provided data.

BUSINESS QUESTIONS RELATED TO E-COMMERCE GROWTH AND OPTIMIZATION

1. **Sales Performance:**
   - What are the top-selling products and categories?
   - What are the revenue trends over time (monthly, quarterly, annually)?
   - How do sales vary across different regions or demographics?
2. **Customer Segmentation:**
   - How are customers segmented based on their purchasing behavior?

    - What percentage of customers fall into high-value frequent, high-value infrequent, low-value frequent, and low-value infrequent segments?
3. **Order Fulfillment:**
    - What is the average order cancellation rate?
    - What is the on-time delivery rate and delayed delivery rate?
4. **Payment Preferences:**
    - What are the preferred payment methods among customers?
    - How many installments do customers prefer when making a purchase?
5. **Customer Feedback:**
    - What is the Net Promoter Score (NPS)?
    - How many reviews have been provided, and how many customers are promoters or detractors?
6. **Category Preferences:**
    - Which product categories are most preferred by customers?
    - What are the significant drivers for these preferences?
7. **Monthly and Weekly Preferences:**
    - What are the monthly and weekly purchasing trends?

DATA CLEANING AND TRANSFORMATION DOCUMENTATION FOR OLIST DATASETS

1. CUSTOMERS TABLE (`CUSTOMERS`)

- **Remove Duplicates**: Ensure that there are no duplicate customer entries.
- **Standardize Customer IDs**: Verify that all customer IDs are unique and properly formatted.
- **Validate Geolocation Data**: Cross-check the geolocation data for any missing or incorrect values.
- **Clean Column Names**: Standardize column names to follow a consistent naming convention (e.g., `customer_id`, `customer_unique_id`, `customer_zip_code_prefix`).

2. GEOLOCATION TABLE (`GEOLOCATION`)

- **Remove Duplicates**: Ensure that there are no duplicate geolocation entries.
- **Handle Missing Values**: Fill missing geolocation coordinates (latitude and longitude) where possible, or remove records if they are incomplete.
- **Clean Column Names**: Standardize column names (e.g., `geolocation_zip_code_prefix`, `geolocation_lat`, `geolocation_lng`).

3. ORDER ITEMS TABLE (`ORDER_ITEMS`)

- **Remove Duplicates**: Ensure there are no duplicate order item entries.
- **Validate Order IDs**: Ensure that all order IDs are valid and exist in the orders table.
- **Clean Product IDs**: Verify that all product IDs are valid and exist in the products table.
- **Standardize Column Names**: (e.g., `order_id`, `order_item_id`, `product_id`, `seller_id`, `price`, `freight_value`).

4. ORDER PAYMENTS TABLE (`ORDER_PAYMENTS`)

- **Remove Duplicates**: Ensure there are no duplicate payment entries.
- **Validate Order IDs**: Ensure that all order IDs are valid and exist in the orders table.
- **Handle Missing Values**: Fill or remove any missing payment details.

- **Standardize Payment Types**: Ensure consistency in payment types (e.g., `credit_card`, `boleto`, `voucher`).
- **Clean Column Names**: (e.g., `order_id`, `payment_sequential`, `payment_type`, `payment_installments`, `payment_value`).

5. ORDER REVIEWS TABLE (`ORDER_REVIEWS`)

- **Remove Duplicates**: Ensure there are no duplicate review entries.
- **Validate Order IDs**: Ensure that all order IDs are valid and exist in the orders table.
- **Handle Missing Review Scores**: Fill missing review scores with a default value or remove the records.
- **Standardize Column Names**: (e.g., `review_id`, `order_id`, `review_score`, `review_comment_message`).

6. ORDERS TABLE (`ORDERS`)

- **Remove Duplicates**: Ensure there are no duplicate order entries.
- **Validate Customer IDs**: Ensure that all customer IDs are valid and exist in the customers table.
- **Handle Missing Dates**: Fill missing dates (order purchase date, estimated delivery date) where possible.
- **Standardize Order Status**: Ensure consistency in order status values (e.g., `delivered`, `shipped`, `canceled`).
- **Clean Column Names**: (e.g., `order_id`, `customer_id`, `order_status`, `order_purchase_timestamp`).

7. PRODUCTS TABLE (`PRODUCTS`)

- **Remove Duplicates**: Ensure there are no duplicate product entries.
- **Validate Product IDs**: Ensure that all product IDs are unique and properly formatted.
- **Handle Missing Values**: Fill missing product details (e.g., `product_category_name`, `product_description_length`).
- **Standardize Column Names**: (e.g., `product_id`, `product_category_name`, `product_weight_g`, `product_length_cm`).

8. SELLERS TABLE (`SELLERS`)

- **Remove Duplicates**: Ensure there are no duplicate seller entries.
- **Validate Seller IDs**: Ensure that all seller IDs are unique and properly formatted.
- **Handle Missing Values**: Fill missing geolocation data where possible.
- **Clean Column Names**: (e.g., `seller_id`, `seller_zip_code_prefix`, `seller_city`, `seller_state`).

9. PRODUCT CATEGORY NAME TRANSLATION TABLE (`PRODUCT_CATEGORY_NAME_TRANSLATION`)

- **Remove Duplicates**: Ensure there are no duplicate category name entries.
- **Validate Translations**: Ensure that all translations are accurate and cover all necessary categories.
- **Standardize Column Names**: (e.g., `product_category_name`, `product_category_name_english`).

GENERAL DATA TRANSFORMATION STEPS

1. **Data Type Conversion**: Ensure all columns have appropriate data types (e.g., dates, numerical values, categorical values).
2. **Date/Time Formatting**: Standardize date and time formats across all tables.
3. **Column Renaming**: Rename columns for consistency and readability.
4. **Creating New Columns**: Derive new columns where necessary for analysis (e.g., calculating delivery time, total order value).

DATA MODELING AND RELATIONSHIPS

1. ESTABLISH RELATIONSHIPS BETWEEN TABLES BASED ON DATA STRUCTURE

- **One-to-One Relationships**: These are rare but can occur in certain scenarios. For example, each `customer_id` in the `customer` table might have a unique corresponding entry in an `orders` table .
- **One-to-Many Relationships**: Common in e-commerce data, where one entity is related to multiple records in another table. Examples:
  - `orders` to `order_items`: One order can contain multiple items.
  - `products` to `order_items`: One product can be part of multiple order items.

2. CREATE A STAR SCHEMA OR FOR EFFICIENT DATA MODELING

- **Star Schema**: Central fact table surrounded by dimension tables. Efficient for querying and analysis.
  - **Fact Table**: merge of `orders` & `order_items`
  - **Dimensions**:
    - `customers`
    - `products`
    - `sellers`
    - `order_payments`
    - `order_reviews`
    - `geolocation`

3. CONSIDER CARDINALITY

- **One-to-Many (1**

  ): Commonly seen relationships in e-commerce data.

  - Example: `customers` (1) to `orders` (N)
- **Many-to-One (N:1)**: Reverse of the one-to-many relationship.
  - Example: `order_items` (N) to `products` (1)
- **Many-to-Many (N**

  ): Should be carefully modeled using bridge tables to avoid performance issues.

  - Example: If we had a `promotions` table and an `orders` table linked through an `order_promotions` bridge table.

4. VALIDATE DATA MODEL INTEGRITY

- **Data Preview**: Regularly preview data in Power BI to ensure transformations and relationships are working as expected.
- **Cross-Filtering**: Test cross-filtering in Power BI to verify relationships. Ensure that selections in one table filter data appropriately in related tables.
- **Data Integrity Checks**: Perform data validation checks to ensure referential integrity:
    - Check for orphaned records (e.g., orders without corresponding customers).
    - Ensure that all foreign keys (e.g., `product_id` in `order_items`) exist in their respective primary key tables (e.g., `products`).

STEPS TO ESTABLISH RELATIONSHIPS IN POWER BI

1. **Load Data**: Import all tables into Power BI.
2. **Manage Relationships**:
    - Navigate to the "Model" view.
    - Automatically detect relationships using Power BI's "Manage Relationships" feature.
    - Manually adjust or create relationships as necessary by dragging and dropping fields between tables.
3. **Define Relationships**:
    - Specify the relationship type (one-to-one, one-to-many).
    - Set cross-filter direction (single, both).
    - Ensure cardinality is correctly set to optimize performance.
4. **Create Hierarchies** (optional):
    - Create hierarchies within dimension tables to enhance drill-down capabilities (e.g., Year > Quarter > Month for dates).
5. **Test Relationships**:
    - Use Power BI visuals (e.g., tables, charts) to test if filters and slicers are working correctly across related tables.
    - Validate with sample queries to ensure data consistency and integrity.

EXAMPLE RELATIONSHIPS IN POWER BI MODEL VIEW

1. **Customers to Orders**:
    - `customers.customer_id` (1) to `orders.customer_id` (N)
2. **Orders to Order Items**:
    - `orders.order_id` (1) to `order_items.order_id` (N)
3. **Products to Order Items**:
    - `products.product_id` (1) to `order_items.product_id` (N)
4. **Sellers to Order Items**:
    - `sellers.seller_id` (1) to `order_items.seller_id` (N)
5. **Orders to Order Payments**:
    - `orders.order_id` (1) to `order_payments.order_id` (N)
6. **Orders to Order Reviews**:
    - `orders.order_id` (1) to `order_reviews.order_id` (N)
7. **Geolocation to Customers**:
    - `geolocation.geolocation_zip_code_prefix` (1) to `customers.customer_zip_code_prefix` (N)

FINAL VALIDATION AND TESTING

- **Data Preview**: Regularly check the data preview to ensure that transformations and relationships are correctly applied.

- **Cross-Filtering**: Verify that cross-filtering and interactions between visuals work correctly, reflecting accurate data.
- **Performance Testing**: Monitor performance metrics in Power BI to ensure the data model is optimized for query performance.

Important DAX Calculations

- Average order value = DIVIDE([total revenue],CALCULATE(COUNTROWS('orders')))

- payment value MoM% =

```
IF(
  ISFILTERED('orders'[order_purchase_timestamp]),
  ERROR("Time intelligence quick measures can only be grouped or filtered by the Power
BI-provided date hierarchy or primary date column."),
  VAR __PREV_MONTH =
    CALCULATE(
      SUM('order payments'[payment_value]),
      DATEADD('orders'[order_purchase_timestamp].[Date], -1, MONTH)
    )
  RETURN
    DIVIDE(
      SUM('order payments'[payment_value]) - __PREV_MONTH,
      __PREV_MONTH
    )
)
```

- payment value QoQ% =

```
IF(
  ISFILTERED('orders'[order_purchase_timestamp]),
  ERROR("Time intelligence quick measures can only be grouped or filtered by the Power
BI-provided date hierarchy or primary date column."),
  VAR __PREV_QUARTER =
    CALCULATE(
      SUM('order payments'[payment_value]),
      DATEADD('orders'[order_purchase_timestamp].[Date], -1, QUARTER)
    )
  RETURN
```

```
        DIVIDE(
            SUM('order payments'[payment_value]) - __PREV_QUARTER,
            __PREV_QUARTER
        )
)
```

The **Net Promoter Score (NPS)** is a metric used to gauge customer loyalty

```
Detractors =
CALCULATE(
    COUNTROWS('order reviews'),
     'order reviews'[review_score]<3
)
DetractorPercentage =
DIVIDE(
    [Detractors],
    [TotalReviews]
)
Promoters =
CALCULATE(
    COUNTROWS('order reviews'),
    'order reviews'[review_score] >= 3
)
PromoterPercentage =
DIVIDE(
    [Promoters],
    [TotalReviews]
)

NPS =
([PromoterPercentage] - [DetractorPercentage]) *100
```

```
Average Order Cancellation Rate =
        DIVIDE(
          CALCULATE(COUNTROWS(orders),
              orders[order_status] = "canceled"),
                              COUNTROWS(orders))
```

Customer segmentation is the process of dividing a customer base into distinct groups based on shared characteristics. It helps businesses target specific customer groups more effectively, customize their marketing strategies, and improve customer satisfaction and retention.

```
Customer Monetary Value =
CALCULATE(
    SUM('order payments'[payment_value]),
    ALLEXCEPT('customers',
'customers'[customer_unique_id])
)
Customer Order Frequency =
CALCULATE(
    COUNTROWS('customers'),
    ALLEXCEPT('customers', 'customers'[customer_unique_id])
)
Customer_Segment =
SWITCH(
    TRUE(),
    [Customer Order Frequency] > 1 && [Customer Monetary Value] > 200, "High Value Frequent",
    [Customer Order Frequency] > 1 && [Customer Monetary Value] <= 200, "Frequent Low Value",
    [Customer Order Frequency] <= 1 && [Customer Monetary Value] > 200, "High Value Infrequent",
    "Low Value Infrequent"
)
Avg Orders per Seller =
DIVIDE(
    COUNTROWS('order items'),
    [Total Sellers]
)
```

```
Avg Orders per Seller =
DIVIDE(
    COUNTROWS('order items'),
   [Total Sellers]
)


ontime count days =
CALCULATE(COUNT(orders[ontime/delay]),FILTER(orders,orders[ontime/delay]="on time"))

On-Time Delivery Rate = DIVIDE([ontime count days],[ontime/delay count])

delay count days =
CALCULATE(COUNT(orders[ontime/delay]),FILTER(orders,orders[ontime/delay]="delay"))

Delayed Delivery Rate = DIVIDE([delay count days],[ontime/delay count])


Active_Sellers = CALCULATE(
                DISTINCTCOUNT('order items'[seller_id]),
                   'orders'[order_status] = "delivered",
                        'orders'[order_purchase_timestamp] >= MIN('orders'[order_purchase_timestamp])
- 30,
                        'orders'[order_purchase_timestamp] <= MAX('orders'[order_purchase_timestamp])


               )


Avg Orders per Seller =
DIVIDE(
   COUNTROWS('order items'),
   [Total Sellers]
)
```

Conclusions

1. **Customer Segmentation and Preferences**
   - The majority of customers (74%) are categorized as "Low Value Infrequent," indicating that they make small and infrequent purchases.
   - "High Value Infrequent" customers constitute 19% of the customer base, suggesting potential for targeted marketing to increase their purchase frequency.

- The "High Value Frequent" and "Frequent Low Value" segments are smaller, but still significant, representing 4% and 3%, respectively.

2. **Order Fulfillment and Cancellations**
    - The average order cancellation rate is relatively low at 0.63%, indicating a high level of order completion.
    - On-time delivery rates and delayed delivery rates are critical metrics that need continuous monitoring to ensure customer satisfaction.
3. **Payment and Installment Preferences**
    - Credit cards are the most preferred payment method, as indicated by the word cloud.
    - A significant number of customers (52.5K) prefer to pay in one installment, while others opt for multiple installments, reflecting diverse financial flexibility among customers.
4. **Customer Feedback and Satisfaction**
    - The Net Promoter Score (NPS) of 71 is a positive indicator of customer satisfaction and loyalty.
    - A high number of promoters (84,649) compared to detractors (14,575) suggests strong overall customer approval.
5. **Category Preferences**
    - Categories such as "Electronics," "Beauty," and "Fashion" are highly preferred, indicating areas where marketing efforts can be concentrated.
    - Significant drivers behind category preferences provide insights into customer needs and buying behavior.
6. **Monthly and Weekly Trends**
    - Purchase preferences vary by month and day, with noticeable peaks and troughs, which can guide inventory and promotional planning.

Recommendations

1. **Targeted Marketing**
    - Focus on increasing the purchase frequency of "High Value Infrequent" customers through personalized offers and loyalty programs.
    - Develop campaigns to convert "Low Value Infrequent" customers into more frequent buyers.
2. **Order Fulfillment Improvement**
    - Maintain the low order cancellation rate by continuously improving the accuracy and efficiency of the fulfillment process.
    - Monitor and enhance on-time delivery rates to ensure customer satisfaction.
3. **Payment Options**
    - Continue to offer diverse payment and installment options to cater to different customer preferences.
    - Highlight the convenience and benefits of preferred payment methods in marketing materials.
4. **Enhance Customer Satisfaction**
    - Utilize feedback from promoters to understand what drives customer satisfaction and apply these insights to broader customer service strategies.

    - Address the concerns of detractors promptly to convert them into promoters and reduce negative feedback.

5. **Product Category Focus**
    - Allocate marketing resources to high-preference categories like "Electronics," "Beauty," and "Fashion" to maximize sales.
    - Analyze the significant drivers for category preferences to refine product offerings and marketing messages.
6. **Seasonal and Trend Analysis**
    - Use monthly and weekly purchasing trends to plan promotions, inventory, and staffing effectively.
    - Capitalize on peak buying periods with targeted campaigns to boost sales during those times.

By implementing these recommendations, Olist can enhance its e-commerce operations, increase customer satisfaction, and drive growth through data-driven decision-making.