Ecommerce Data Analysis Report

Overview:

- The ecommerce project involves the analysis of various datasets related to an online retail business taken from Kaggle.com (Brazilian E-Commerce Public Dataset by Olist)
- The datasets typically include information about orders, customers, products, payments, sellers, and geolocation.

Introduction:

- The ecommerce industry has witnessed significant growth in recent years, with more consumers opting for online shopping due to convenience and a wide range of options.
- My analysis focuses on a specific ecommerce business, which operates in various regions and offers a diverse range of products.
- The datasets which i analyze comprises several tables containing information about orders, customers, products, payments, sellers, and geolocation.

Objectives:

- The primary objectives of the analysis are to gain actionable insights into various aspects of the ecommerce business, including customer behaviour, sales performance, product trends, geographic distribution, seller performance, and payment trends.
- By analyzing these key areas, we aim to provide valuable insights that can inform strategic decision-making, optimize operational processes, enhance customer experience, and drive business growth.

Data Overview:

Datasets Description:

Dataset - Link

1. **Order Items:** This dataset contains information about individual items within orders, including details such as order ID, product ID, seller ID, price, and shipping details.
2. **Customers:** This dataset provides information about the customers, including unique identifiers, zip code, city, and state.
3. **Orders:** The Orders dataset includes information about orders, such as order ID, customer ID, order status, and timestamps for various stages of the order process (purchase, approval, delivery, etc.).
4. **Sellers:** This dataset contains information about sellers, including seller ID, zip code, city, and state.
5. **Product Categories:** Provides information about product categories, including category names and their English translations.
6. **Geolocation:** This dataset contains geographic information related to zip codes, including latitude, longitude, city, and state.
7. **Payments:** Dataset containing payment details, including order ID, payment sequential, payment type, installments, and payment value.
8. **Products:** Information about products, including product ID, category name, name length, description length, number of photos, weight, length, height, and width.

Key Fields/Columns:

Order Items:

- order_id: Unique identifier for each order.
- product_id: Unique identifier for each product.
- seller_id: Unique identifier for each seller.
- price: Price of the item.
- shipping_limit_date: Deadline for shipping the item.

Customers:

- customer_id: Unique identifier for each customer.
- customer_unique_id: Unique identifier for each customer (potentially a hashed value for privacy).
- customer_zip_code_prefix: First part of the customer's zip code.
- customer_city: City where the customer is located.
- customer_state: State where the customer is located.

Orders:

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

Sellers:

- seller_id: Unique identifier for each seller.
- seller_zip_code_prefix: First part of the seller's zip code.
- seller_city: City where the seller is located.
- seller_state: State where the seller is located.

Product Categories:

- product_category_name: Name of the product category.
- product_category_name_english: English translation of the product category name.

Payments:

- order_id: Unique identifier for each order.
- payment_sequential: Sequential number of the payment.
- payment_type: Type of payment method used.
- payment_installments: Number of installments for the payment.
- payment_value: Value of the payment.

Products:

- product_id: Unique identifier for each product.
- product_category_name: Name of the product category.
- product_name_length: Length of the product name.
- product_description_length: Length of the product description.
- product_photos_qty: Quantity of product photos.
- 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.

Geolocation:

- geolocation_zip_code_prefix: First part of the zip code.
- geolocation_lat: Latitude coordinate of the location.
- geolocation_lng: Longitude coordinate of the location.
- geolocation_city: City name.
- geolocation_state: State abbreviation.

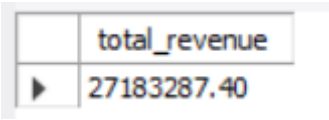
Analysis Findings:

1.Total Revenue Analysis:

This query calculates the total revenue generated from all orders.

```
SELECT SUM(order_items.price) AS total_revenue
FROM order_items;
```

OUTPUT:



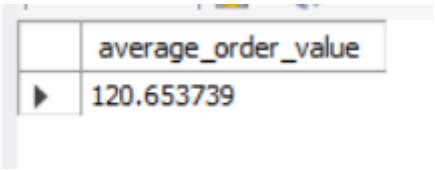
	total_revenue
▶	27183287.40

2.Average Order Value Analysis:

This query computes the average order value, which indicates the average amount spent per order.

```
SELECT AVG(order_items.price) AS average_order_value
FROM order_items;
```

OUTPUT:



	average_order_value
▶	120.653739

3.Customer Distribution by State Analysis:

This query provides a breakdown of the number of customers by state.

```
SELECT customer_state, COUNT(*) AS customer_count
FROM customers
GROUP BY customer_state
ORDER BY customer_count DESC;
```

OUTPUT:

	customer_state	customer_count
	SP	41746
	RJ	12852
	MG	11635
▶	RS	5466
	PR	5045
	SC	3637
	BA	3380
	DF	2140
	ES	2033
	GO	2020
	PE	1652
	CE	1336
	PA	975

Result 47 ×

4.Top Selling Products Analysis:

This query identifies the top 10 best-selling products.

```
SELECT
    p.product_id,
    p.product_category_name,
    COUNT(*) AS total_sales
FROM
    products p
JOIN
    order_items oi ON p.product_id = oi.product_id
GROUP BY
    p.product_id
ORDER BY
    total_sales DESC
LIMIT
    10;
```

OUTPUT:

product_id	product_category_name	total_sales
aca2eb7d00ea1a7b8ebd4e68314663af	moveis_decoracao	1054
99a4788cb24856965c36a24e339b6058	cama_mesa_banho	976
422879e10f46682990de24d770e7f83d	ferramentas_jardim	968
389d119b48cf3043d311335e499d9c6b	ferramentas_jardim	784
368c6c730842d78016ad823897a372db	ferramentas_jardim	776
53759a2ecddad2bb87a079a1f1519f73	ferramentas_jardim	746
d1c427060a0f73f6b889a5c7c61f2ac4	informatica_acessorios	686
53b36df67ebb7c41585e8d54d6772e08	relogios_presentes	646
154e7e31ebfa092203795c972e5804a6	beleza_saude	562
3dd2a17168ec895c781a9191c1e95ad7	informatica_acessorios	548

5.Payment Method Analysis:

This query provides insights into payment methods used by customers and their corresponding payment counts and total amounts.

```
SELECT
    payment_type,
    COUNT(*) AS payment_count,
    SUM(payment_value) AS total_payment_amount
FROM
    payments
GROUP BY
    payment_type;
```

OUTPUT:

payment_type	payment_count	total_payment_amount
credit_card	76795	12542084.19
boleto	19784	2869361.27
voucher	5775	379436.87
debit_card	1529	217989.79

6.Order Status Analysis:

This query analyzes the distribution of orders by status, helping to understand order fulfillment efficiency.

```
SELECT
    order_status,
    COUNT(*) AS order_count
FROM
    orders
GROUP BY
    order_status;
```

OUTPUT:

order_status	order_count
delivered	96455
canceled	6

7.Geolocation Analysis:

This query explores the geographic distribution of customers or sellers by state.

```
    SELECT
    geolocation_state,
    COUNT(*) AS location_count
FROM
    geolocation
GROUP BY
    geolocation_state
ORDER BY
    location_count DESC;
```

OUTPUT:

	geolocation_state	location_count
▶	SP	1212804
	MG	79008
	RJ	363507
	RS	185553
	PR	173577
	SC	114984
	BA	108135
	GO	60417
	ES	50244
	PE	49296
	DF	38958
	MT	36093
	CE	35022
	PA	32559
	MS	31293
	MA	23559
	PB	16614
	RN	15123

Result 51 ×

8.Sales Performance Analysis:

This query retrieves the total sales for each month by joining the orders and order_items tables.

```
SELECT
    DATE_FORMAT(o.order_purchase_timestamp, '%Y-%m') AS order_month,
    SUM(oi.price) AS total_sales
FROM
    orders o
JOIN
    order_items oi ON o.order_id = oi.order_id
GROUP BY
    order_month
ORDER BY
    order_month;
```

OUTPUT:

order_month	total_sales
2016-09	269.94
2016-10	81882.60
2016-12	21.80
2017-01	223424.94
2017-02	465277.72
2017-03	718397.70
2017-04	681339.36
2017-05	978318.50
2017-06	843846.74
2017-07	963209.04
2017-08	1109399.40
2017-09	1214441.34
2017-10	1296495.30
2017-11	1975296.14
2017-12	1452066.38
2018-01	1849290.00
2018-02	1652934.24
2018-03	1906712.50

esult 52 ×

9.Top Selling Categories Analysis:

This query identifies the top-selling product categories by joining the product_categories, products, and order_items tables.

```
SELECT
    pc.product_category_name,
    COUNT(oi.product_id) AS total_sales
FROM
    product_categories pc
LEFT JOIN
    products p ON pc.product_category_name = p.product_category_name
LEFT JOIN
    order_items oi ON p.product_id = oi.product_id
GROUP BY
    pc.product_category_name
ORDER BY
    total_sales DESC
LIMIT 10;
```

OUTPUT:

product_category_name	total_sales
cama_mesa_banho	22230
beleza_saude	19340
esporte_lazer	17282
moveis_decoracao	16668
informatica_acessorios	15654
utilidades_domesticas	13928
relogios_presentes	11982
telefonia	9090
ferramentas_jardim	8694
automotivo	8470

9. Order Approval Time Analysis:

This query calculates the average time taken for order approval by subtracting the order_purchase_timestamp from the order_approved_at, using TIMESTAMPDIFF function.

```
SELECT
    AVG(TIMESTAMPDIFF(HOUR, o.order_purchase_timestamp, o.order_approved_at)) AS avg_approval_time_hours
FROM
    orders o;
```

OUTPUT:

avg_approval_time_hours
9.9267

11.Average Order Value by State Analysis:

This query calculates the average order value for each state by joining the customers, orders, and order_items tables.

```
SELECT
    c.customer_state,
    AVG(oi.price) AS avg_order_value
FROM
    customers c
JOIN
    orders o ON c.customer_id = o.customer_id
JOIN
    order_items oi ON o.order_id = oi.order_id
GROUP BY
    c.customer_state;
```

OUTPUT:

customer_state	avg_order_value
RJ	124.432195
SP	109.092213
MG	120.200637
PR	117.908216
GO	124.214625
BA	134.016872
AL	184.673817
MS	142.330419
CE	154.159572
DF	125.901660
RS	118.774340
PE	144.266604
SC	123.754790
ES	120.738629
MA	146.344668
PA	165.531869
MT	146.761446
PR	192.127679

Result 55

12. Average Payment Installments by Product Category:

This query calculates the average number of payment installments for each product category, providing insights into customer payment behaviour.

```sql
SELECT
    pc.product_category_name,
    AVG(pmt.payment_installments) AS avg_payment_installments
FROM
    product_categories pc
LEFT JOIN
    products p ON pc.product_category_name = p.product_category_name
LEFT JOIN
    order_items oi ON p.product_id = oi.product_id
LEFT JOIN
    payments pmt ON oi.order_id = pmt.order_id
GROUP BY
    pc.product_category_name
ORDER BY
    avg_payment_installments DESC;
```

OUTPUT:

product_category_name	avg_payment_installments
pcs	6.0091
portateis_casa_forno_e_cafe	5.4872
la_cuisine	4.2500
eletrodomesticos_2	4.1045
moveis_sala	4.0265
casa_conforto	3.9937
moveis_escritorio	3.8172
fashion_roupa_feminina	3.7917
relogios_presentes	3.6730
construcao_ferramentas_construcao	3.6572
cama_mesa_banho	3.5532
eletroportateis	3.5435
moveis_cozinha_area_de_servico_j...	3.5369
casa_construcao	3.5158
construcao_ferramentas_seguranca	3.4523
instrumentos_musicais	3.4485

13.Order Status Analysis by Customer City:

This query analyzes the distribution of order statuses (e.g., delivered, processing) by customer city, helping identify areas with potential delivery or processing issues.

```sql
SELECT
    c.customer_city,
    o.order_status,
    COUNT(o.order_id) AS order_count
FROM
    customers c
JOIN
    orders o ON c.customer_id = o.customer_id
GROUP BY
    c.customer_city, o.order_status
ORDER BY
    c.customer_city, order_count DESC;
```

OUTPUT:

customer_city	order_status	order_count
abadia dos dourados	delivered	3
abadiania	delivered	1
abaete	delivered	11
abaetetuba	delivered	11
abaiara	delivered	2
abaira	delivered	2
abare	delivered	2
abatia	delivered	3
abdon batista	delivered	1
abelardo luz	delivered	6
abrantes	delivered	1
abre campo	delivered	5
abreu e lima	delivered	11
acaiaca	delivered	2
acailandia	delivered	7
acajutiba	delivered	1
acarau	delivered	8
acari	delivered	1
acegua	delivered	2

14. Product Return Rate Analysis:

This query calculates product category to the total number of orders.

```
SELECT
    pc.product_category_name,
    COUNT(DISTINCT o.order_id) AS total_orders
FROM
    product_categories pc
LEFT JOIN
    products p ON pc.product_category_name = p.product_category_name
LEFT JOIN
    order_items oi ON p.product_id = oi.product_id
LEFT JOIN
    orders o ON oi.order_id = o.order_id
GROUP BY
    pc.product_category_name;
```

OUTPUT:

product_category_name	total_orders
agro_industria_e_comercio	177
alimentos	441
alimentos_bebidas	221
artes	195
artes_e_artesanato	23
artigos_de_festas	38
artigos_de_natal	125
audio	348
automotivo	3809
bebes	2808
bebidas	287

15. Customer Lifetime Value Analysis:

This query calculates the total amount spent, total number of orders, and average order value for each customer, allowing for the analysis of customer lifetime value.

```sql
SELECT
    c.customer_id,
    SUM(oi.price) AS total_spent,
    COUNT(DISTINCT o.order_id) AS total_orders,
    AVG(oi.price) AS avg_order_value
FROM
    customers c
JOIN
    orders o ON c.customer_id = o.customer_id
JOIN
    order_items oi ON o.order_id = oi.order_id
GROUP BY
    c.customer_id
ORDER BY
    total_spent DESC;
```

OUTPUT:

customer_id	total_spent	total_orders	avg_order_value
1617b1357756262bfa56ab541c47bc16	26880.00	1	1680.000000
ec5b2ba62e574342386871631fafd3fc	14320.00	1	1790.000000
c6e2731c5b391845f6800c97401a43a9	13470.00	1	6735.000000
f48d464a0baaea338cb25f816991ab1f	13458.00	1	6729.000000
3fd6777bbce08a352fddd04e4a7cc8f6	12998.00	1	6499.000000
05455dfa7cd02f13d132aa7a6a9729c6	11869.20	1	989.100000
df55c14d1476a9a3467f131269c2477f	9598.00	1	4799.000000
24bbf5fd2f2e1b359ee7de94defc4a15	9380.00	1	4690.000000
3d979689f636322c62418b6346b1c6d2	9180.00	1	4590.000000
cc803a2c412833101651d3f90ca7de24	8800.00	1	2200.000000
1afc82cd60e303ef09b4ef9837c9505c	8799.74	1	4399.870000
35a413c7ca3c69756cb75867d6311c0d	8199.98	1	4099.990000

16. Popular Product Categories by Customer City:

This query identifies the most popular product categories for each customer city based on the number of orders, helping to understand regional preferences.

```sql
SELECT
    c.customer_city,
    pc.product_category_name,
    COUNT(DISTINCT oi.order_id) AS order_count
FROM
    customers c
JOIN
    orders o ON c.customer_id = o.customer_id
JOIN
    order_items oi ON o.order_id = oi.order_id
JOIN
    products p ON oi.product_id = p.product_id
JOIN
    product_categories pc ON p.product_category_name = pc.product_category_name
GROUP BY
    c.customer_city, pc.product_category_name
ORDER BY
    order_count DESC;
```

OUTPUT:

customer_city	product_category_name	order_count
sao paulo	cama_mesa_banho	1623
sao paulo	beleza_saude	1501
sao paulo	esporte_lazer	1229
sao paulo	utilidades_dom	3
sao paulo	informatica_acessorios	980
sao paulo	moveis_decoracao	973
sao paulo	relogios_presentes	760
rio de janeiro	cama_mesa_banho	704
sao paulo	brinquedos	594
sao paulo	telefonia	554
rio de janeiro	beleza_saude	504
rio de janeiro	esporte_lazer	496
sao paulo	automotivo	492

esporte_lazer

17.Top Selling Products:

This query identifies the top-selling products by counting the number of order items associated with each product, providing insights into product popularity.

```
SELECT
    p.product_id,
    p.product_category_name,
    COUNT(oi.order_item_id) AS total_sales
FROM
    products p
JOIN
    order_items oi ON p.product_id = oi.product_id
GROUP BY
    p.product_id, p.product_category_name
ORDER BY
    total_sales DESC
LIMIT 10;
```

OUTPUT:

product_id	product_category_name	total_sales
aca2eb7d00ea1a7b8ebd4e68314663af	moveis_decoracao	1054
99a4788cb24856965c36a24e339b6058	cama_mesa_banho	976
422879e10f46682990de24d770e7f83d	ferramentas_jardim	968
389d119b48cf3043d311335e499d9c6b	ferramentas_jardim	784
368c6c730842d78016ad823897a372db	ferramentas_jardim	776
53759a2ecddad2bb87a079a1f1519f73	ferramentas_jardim	746
d1c427060a0f73f6b889a5c7c61f2ac4	informatica_acessorios	686
53b36df67ebb7c41585e8d54d6772e08	relogios_presentes	646
154e7e31ebfa092203795c972e5804a6	beleza_saude	562
3dd2a17168ec895c781a9191c1e95ad7	informatica_acessorios	548

18.Sales Distribution by Month:

This query calculates the total sales for each month, allowing for the analysis of sales trends over time.

```
SELECT
    YEAR(o.order_purchase_timestamp) AS year,
    MONTH(o.order_purchase_timestamp) AS month,
    SUM(oi.price) AS total_sales
FROM
    orders o
JOIN
    order_items oi ON o.order_id = oi.order_id
GROUP BY
    YEAR(o.order_purchase_timestamp), MONTH(o.order_purchase_timestamp)
ORDER BY
    year, month;
```

OUTPUT:

year	month	total_sales
2016	9	269.94
2016	10	81882.60
2016	12	223.80
2017	1	223424.94
2017	2	465277.72
2017	3	718397.70
2017	4	681339.36
2017	5	978318.50
2017	6	843846.74
2017	7	963209.04
2017	8	1109399.40
2017	9	1214441.34
2017	10	1296495.30

19. Average Order Value by Customer State:

This query calculates the average order value for each customer state, providing insights into regional spending habits.

```
    SELECT
    c.customer_state,
    AVG(oi.price) AS avg_order_value
FROM
    customers c
JOIN
    orders o ON c.customer_id = o.customer_id
JOIN
    order_items oi ON o.order_id = oi.order_id
GROUP BY
    c.customer_state
ORDER BY
    avg_order_value DESC;
```

OUTPUT:

customer_state	avg_order_value
PB	192.127679
AL	184.673817
AC	175.065604
RO	167.336117
PA	165.531869
AP	165.121111
PI	161.990440
RN	157.592438
TO	156.137129
CE	154.159572
RR	153.423261
SE	150.864507
MT	146.761446
MA	146.344668
PE	144.266604
MS	142.330419
AM	135.925399

20.Seller Performance Analysis:

This query provides insights into seller performance by calculating the total number of orders and total sales generated by each seller.

```
    SELECT
    s.seller_id,
    s.seller_city,
    s.seller_state,
    COUNT(o.order_id) AS total_orders,
    SUM(oi.price) AS total_sales
FROM
    sellers s
LEFT JOIN
    order_items oi ON s.seller_id = oi.seller_id
LEFT JOIN
    orders o ON oi.order_id = o.order_id
GROUP BY
    s.seller_id, s.seller_city, s.seller_state
ORDER BY
    total_sales DESC;
```

OUTPUT:

seller_id	seller_city	seller_state	total_orders	total_sales
4869f7a5dfa277a7dca6462dcf3b52b2	guariba	SP	2296	458945.26
53243585a1d6dc2643021fd1853d8905	lauro de freitas	BA	800	445552.10
4a3ca9315b744ce9f8e9374361493884	ibitinga	SP	3898	400945.84
fa1c13f2614d7b5c4749cbc52fecda94	sumare	SP	1158	388084.06
7c67e1448b00f6e969d365cea6b010ab	itaquaquecetuba	SP	2710	375847.78
7e93a43ef30c4f03f38b393420bc753a	barueri	SP	644	352863.74
da8622b14eb17ae2831f4ac5b9dab84a	piracicaba	SP	3096	320473.14
7a67c85e85bb2ce8582c35f2203ad736	sao paulo	SP	2304	283491.06
1025f0e2d44d7041d6cf58b6550e0bfa	sao paulo	SP	2840	277937.10
955fee9216a65b617aa5c0531780ce60	sao paulo	SP	2944	270343.40
46dc3b2cc0980fb8ec44634e21d2718e	rio de janeiro	RJ	1048	256222.38
6560211a19b47992c3666cc44a7e94c0	sao paulo	SP	3992	246609.66
620c87c171fb2a6dd6e8bb4dec959fc6	petropolis	RJ	1562	229549.00
7d13fca15225358621be4086e1eb0964	ribeirao preto	SP	1142	227257.94
5dceca129747e92ff8ef7a997dc4f8ca	santa barbara d...	SP	686	224311.06
1f50f920176fa81dab994f9023523100	sao jose do rio ...	SP	3852	213878.42
cc419e0650a3c5ba77189a1882b7556a	santo andre	SP	3438	208576.84

21.Average Shipping Time by Seller State:

This query calculates the average shipping time for orders fulfilled by sellers in each state, helping to identify regions with faster or slower delivery times.

```
SELECT
    s.seller_state,
    AVG(DATEDIFF(o.order_delivered_customer_date, o.order_approved_at)) AS avg_shipping_time
FROM
    sellers s
JOIN
    order_items oi ON s.seller_id = oi.seller_id
JOIN
    orders o ON oi.order_id = o.order_id
WHERE
    o.order_delivered_customer_date IS NOT NULL
GROUP BY
    s.seller_state
ORDER BY
    avg_shipping_time ASC;
```

OUTPUT:

seller_state	avg_shipping_time
RS	10.9719
RJ	11.4609
PB	11.6486
SP	11.6853
SE	11.7000
MS	11.8800
DF	11.9060
PE	12.2090
MG	12.2483
ES	12.2747
GO	12.2756
RN	12.3036
PA	12.7500
PR	12.8115

22. Product Category Sales Analysis:

This query provides insights into sales performance across different product categories, including the total number of items sold and total revenue generated for each category.

```
SELECT
    pc.product_category_name_english,
    COUNT(oi.order_item_id) AS total_items_sold,
    SUM(oi.price) AS total_revenue
FROM
    product_categories pc
JOIN
    products p ON pc.product_category_name = p.product_category_name
JOIN
    order_items oi ON p.product_id = oi.product_id
GROUP BY
    pc.product_category_name_english
ORDER BY
    total_revenue DESC;
```

OUTPUT:

product_category_name_english	total_items_sold	total_revenue
health_beauty	19340	2517362.68
watches_gifts	11982	2410011.36
bed_bath_table	22230	2073977.36
sports_leisure	17282	1976097.94
computers_accessories	15654	1823908.64
furniture_decor	16668	1459524.98
cool_stuff	7592	1270581.70
housewares	13928	1264497.32
auto	8470	1185440.22
garden_tools	8694	970512.92
toys	8234	967893.20
baby	6128	819661.78
perfumery	6838	798249.74
telephony	9090	647335.06
office_furniture	3382	547921.40
stationery	5034	461886.46
computers	406	445926.26

23.Customer State Analysis:

This query analyzes customer spending and order frequency by state, helping to identify regions with high-value customers and potential opportunities.

```
SELECT
    c.customer_state,
    COUNT(DISTINCT o.order_id) AS total_orders,
    SUM(oi.price) AS total_spent
FROM
    customers c
JOIN
    orders o ON c.customer_id = o.customer_id
JOIN
    order_items oi ON o.order_id = oi.order_id
GROUP BY
    c.customer_state
ORDER BY
    total_spent DESC;
```

OUTPUT:

customer_state	total_orders	total_spent
SP	40489	10131830.22
RJ	12351	3519689.06
MG	11352	3104542.04
RS	5342	1456410.96
PR	4923	1332127.02
SC	3547	1014294.26
BA	3256	987168.28
DF	2080	592996.82
GO	1957	565673.40
ES	1995	537286.90
PE	1593	503778.98
CE	1278	439354.78
PA	946	348941.18
MT	886	304383.24
MA	716	233858.78
MS	701	230859.94

24.Order Fulfillment Efficiency:

This query tracks order fulfillment efficiency over time by analyzing the total number of orders, items sold, and revenue generated on each day.

```
    SELECT
    DATE(o.order_approved_at) AS order_date,
    COUNT(DISTINCT o.order_id) AS total_orders,
    COUNT(DISTINCT oi.order_item_id) AS total_items_sold,
    SUM(oi.price) AS total_revenue
FROM
    orders o
JOIN
    order_items oi ON o.order_id = oi.order_id
GROUP BY
    order_date
ORDER BY
    order_date;
```

OUTPUT:

order_date	total_orders	total_items_sold	total_revenue
2016-09-15	1	3	269.94
2016-10-04	18	2	4507.34
2016-10-05	7	4	1673.36
2016-10-06	93	5	29107.24
2016-10-07	47	6	15610.44
2016-10-08	23	3	11075.82
2016-10-09	9	2	2178.36
2016-10-10	40	6	11277.14
2016-10-11	26	1	4864.32
2016-10-12	4	1	389.20
2016-10-13	3	2	1199.38
2016-12-23	1	1	21.80
2017-01-05	3	1	171.60
2017-01-06	3	1	1811.78
2017-01-07	33	2	3326.00
2017-01-08	2	1	397.78

Recommendations:

Improving Sales Performance:

- **Targeted Marketing Campaigns:** Utilize insights from the product category sales analysis to design targeted marketing campaigns for top-performing categories. For example, allocate a higher marketing budget to promote products in these categories through social media ads or email marketing.

- **Product Bundling:** Identify complementary products within top-selling categories and create bundled offerings to encourage higher average order value (AOV). For instance, offer discounts for purchasing related items together, such as pairing a camera with memory cards and tripods.

- **Seasonal Promotions:** Leverage seasonal trends identified in the sales data to launch time-sensitive promotions and capitalize on peak demand periods. For instance, offer discounts on winter clothing during the colder months to stimulate sales.

Enhancing Customer Engagement:

- **Personalized Recommendations:** Utilize customer state analysis insights to tailor product recommendations based on regional preferences and behaviors. Implement recommendation algorithms that suggest products similar to those previously purchased by customers in the same state.

- **Reward Programs:** Implement a loyalty program to incentivize repeat purchases and foster customer loyalty. Offer rewards such as discounts, exclusive access to new products, or free shipping for members who reach certain spending thresholds.

- **Interactive Content:** Create interactive content such as quizzes or polls on social media platforms to engage

customers and gather feedback about their preferences. Use insights from customer interactions to refine product offerings and marketing strategies.

Optimizing Operational Efficiency:

- **Real-Time Inventory Management:** Implement an inventory management system that provides real-time updates on stock levels and automates replenishment processes. Use insights from order fulfillment efficiency analysis to forecast demand accurately and maintain optimal inventory levels to prevent stockouts or overstocking.

- **Streamlined Order Processing:** Utilize automation tools to streamline order processing and reduce manual errors. Implement order tracking systems that provide customers with real-time updates on the status of their orders, enhancing transparency and improving the overall shopping experience.

- **Supplier Collaboration:** Collaborate closely with suppliers to optimize logistics and minimize shipping delays. Negotiate favorable terms with shipping partners and explore alternative delivery options, such as local fulfillment centers, to reduce shipping times and costs.

Conclusion:

Summary of Key Insights and Recommendations:

- The analysis of the ecommerce dataset provided valuable insights into various aspects of the business, from sales performance across product categories to customer behaviour and order fulfillment efficiency.

Appendix:

Detailed SQL queries used for the analysis - Link