WALMART EXECUTIVE DASHBOARD

Presented by: Shalini Raj

Date:25-08-25

SQL PROJECT OVERVIEW

Purpose:

To simulate real-world analytics at Walmart and extract actionable insights using advanced SQL.

Focus Areas:

- Sales performance
- Customer behavior
- Operational efficiency
- Regional analysis
- Time-based trends

Tools Used: SQL

Calculate total sales revenue and quantity sold by product category and customer_state.

```sql
select products.product_category, customers.customer_state,
round(sum(order_items.price),2) as sales_revenue,
sum(order_items.order_item_id) as quant_sold from
order_items join orders on order_items.order_id =orders.order_id
join customers ON orders.customer_id = customers.customer_id
JOIN products ON order_items.product_id = products.product_id
GROUP BY products.product_category, customers.customer_state
ORDER BY sales_revenue DESC;
```

product_category	customer_state	sales_revenue	quant_sold
Furniture Decora...	MG	637.75	32
Furniture Decora...	SP	1145.44	31
Cool Stuff	SP	16898.09	30
bed table bath	SP	1691.98	20
Furniture office	RJ	319.96	16
sport leisure	SP	1214.87	12
HEALTH BEAUTY	SP	981.7	11
Cool Stuff	RJ	4947.89	9
housewares	SP	275.79	9

Identify the top 5 products by total sales revenue across all Walmart regions.

```sql
with a as (SELECT p.product_id,
SUM(oi.price) AS total_sales_rev
FROM order_items oi
JOIN products p ON oi.product_id = p.product_id
GROUP BY 1)
select * from
(select * ,dense_rank() over(order by total_sales_rev desc)
 rnk from a) b where rnk <=5;
```

product_id	total_sales_rev	rnk
5f504b3a1c75b73d6151be81eb05bdc9	37733.90000000001	1
1e5428c428e0f783acd6e3d94ba4ee2a	4792	2
0d009643171aee696f4733340bc2fdd0	3581.9	3
2bfb0bdbcea77da2879f0ef3be9a219e	2801.6	4
732bd381ad09e530fe0a5f457d81becb	1926	5

Compute average delivery time (in days) by seller state

```sql
select s.seller_state,
avg(DATEDIFF(o.order_delivered_customer_date,
o.order_purchase_timestamp)) as avg_del_days
from orders o
join order_items oi on o.order_id = oi.order_id
join sellers s on oi.seller_id = s.seller_id
where o.order_delivered_customer_date IS NOT NULL
group by s.seller_state order by avg_del_days;
```

seller_state	avg_del_days
AC	NULL
RS	11.4910
RJ	11.9396
SP	12.2149
MS	12.3000
PB	12.3784
DF	12.4371
SE	12.6000
MG	12.7463
GO	12.7677

seller_state	avg_del_days
PE	12.8494
ES	12.8571
RN	12.9286
PA	13.2500
PR	13.3220
SC	13.5195
PI	13.7273
BA	13.7708
MT	14.6875

List the top 5 sellers based on total revenue earned.

```sql
with a as (SELECT s.seller_id,s.seller_state,
round(SUM(oi.price),2) AS total_revenue
FROM sellers s JOIN
order_items oi ON s.seller_id = oi.seller_id
GROUP BY s.seller_id,s.seller_state)
select * from(
select *, dense_rank() over(order by total_revenue desc)
 rnk from a) b where rnk <=5;
```

seller_id	seller_state	total_revenue
4869f7a5dfa277a7dca6462dcf3b52b2	SP	229472.63
53243585a1d6dc2643021fd1853d8905	BA	222776.05
4a3ca9315b744ce9f8e9374361493884	SP	200472.92
fa1c13f2614d7b5c4749cbc52fecda94	SP	194042.03
7c67e1448b00f6e969d365cea6b010ab	SP	187923.89

Analyze the monthly revenue trend over the last 12 months to track Walmart's growth.

```sql
select
date_format(o.order_purchase_timestamp, '%Y/%m') as months,
round(sum(oi.price),2) as monthly_revenue from orders o
join order_items oi on o.order_id = oi.order_id
where o.order_purchase_timestamp >= date_sub(
(select max(order_purchase_timestamp) from orders),
interval 12 month)
group by months order by months;
```

months	monthly_revenue
2017/10	296896.33
2017/11	1010271.37
2017/12	743914.17
2018/01	950030.36
2018/02	844178.71
2018/03	983213.44
2018/04	996647.75
2018/05	996517.68
2018/06	865124.31
2018/07	895507.22
2018/08	854686.33

2018/07	895507.22
2018/08	854686.33
2018/09	145

List the top 5 sellers based on total revenue earned.

```sql
select
date_format(first_buy_date, '%Y/%m') as months,
count(*) as new_customers from
(select c.customer_unique_id,
min(o.order_purchase_timestamp) as first_buy_date
from customers c
join orders o on c.customer_id = o.customer_id
group by c.customer_unique_id) as first_orders
group by 1 order by 1;
```

months	new_customers
2016-09	4
2016-10	321
2016-12	1
2017-01	764
2017-02	1752
2017-03	2636
2017-04	2352
2017-05	3596
2017-06	3139
2017-07	3894
2017-08	4184
2017-09	4130
2017-10	4470

Rank customers by lifetime spend within each customer state using SQL window functions.

```sql
select *,
rank() over (partition by customer_state
order by total_spend desc) as rnk from (
select c.customer_state, c.customer_unique_id,
SUM(oi.price) as total_spend 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, c.customer_unique_id)
as b;
```

customer_state	customer_unique_id	total_spend	rnk
AC	62a459e5629b03dd73134964df732077	1200	1
AC	086d6b5b5ba195a91aa0a6ec8e75d1a4	961.6	2
AC	3947ca729a860c522a64a49d762baada	839.99	3
AC	3e5c928acf49c4b95e57af1f350d3493	809.1	4
AC	28989ef45087c96e5a4346e88216c2ba	589.6	5
AC	22c739518f5240ffd2f80e0087aebe26	549	6
AC	c6bc2bf4b75f3f9da5ce95971a2f463b	527.9	7
AC	0845d810b482f4d469d3c88380f4a7d5	524.9	8
AC	2ec67750cd5b985538c03061ebee01ef	510	9
AC	12e92c0f870fc6941c2bfafae6357c4b	479.4	10
AC	8c792a4fb7f5d708257abf55fa899ce8	399	11
AC	85c3726baee0e11d07b88888586cb59f	388	12
AC	aab7b3f97f05bbb173c7d408f8ee96a0	369.8	13

Compute the rolling 3-month average revenue trend, to visualize sales momentum

```sql
select month, rev, ROUND(avg(rev)
over (order by month rows between 2 preceding and current row), 2)
 as rolling_3_mon_avg from (
select DATE_FORMAT(o.order_purchase_timestamp, '%Y/%m') as month,
round(SUM(oi.price),2) as rev from orders o
join order_items oi on o.order_id = oi.order_id
group by month) AS monthly_revenue order by month;
```

month	rev	rolling_3_mon_avg
2016/09	267.36	267.36
2016/10	49507.66	24887.51
2016/12	10.9	16595.31
2017/01	120312.87	56610.48
2017/02	247303.02	122542.26
2017/03	374344.3	247320.06
2017/04	359927.23	327191.52
2017/05	506071.14	413447.56
2017/06	433038.6	433012.32
2017/07	498031.48	479047.07

Find customers with the highest number of orders and total spend, ranking them as Walmart's most valuable customers

```sql
SELECT  c.customer_id, c.customer_unique_id,
COUNT(o.order_id) AS total_orders,
SUM(oi.price) AS total_spend 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, c.customer_unique_id
ORDER BY total_spend desc,total_orders desc
LIMIT 10;
```

customer_id	customer_unique_id	total_orders	total_spend
1617b1357756262bfa56ab541c47bc16	0a0a92112bd4c708ca5fde585afaa872	8	13440
ec5b2ba62e574342386871631fafd3fc	763c8b1c9c68a0229c42c9fc6f662b93	4	7160
c6e2731c5b391845f6800c97401a43a9	dc4802a71eae9be1dd28f5d788ceb526	1	6735
f48d464a0baaea338cb25f816991ab1f	459bef486812aa25204be022145caa62	1	6729
3fd6777bbce08a352fddd04e4a7cc8f6	ff4159b92c40ebe40454e3e6a7c35ed6	1	6499
05455dfa7cd02f13d132aa7a6a9729c6	4007669dec559734d6f53e029e360987	6	5934.6
df55c14d1476a9a3467f131269c2477f	da122df9eeddfedc1dc1f5349a1a690c	1	4799
24bbf5fd2f2e1b359ee7de94defc4a15	eebb5dda148d3893cdaf5b5ca3040ccb	1	4690
e0a2412720e9ea4f26c1ac985f6a7358	5d0a2980b292d049061542014e8960bf	2	4599.9
3d979689f636322c62418b6346b1c6d2	48e1ac109decbb87765a3eade6854098	1	4590

Identify the top 5 products by total sales revenue across all Walmart regions.

```sql
SELECT distinct customer_state, ROUND(AVG(order_value)
over (PARTITION BY customer_state), 2) AS AOV from
( SELECT o.order_id, c.customer_state,
SUM(oi.price) AS order_value
    from orders o
    join customers c on o.customer_id = c.customer_id
    join order_items oi on o.order_id = oi.order_id
    GROUP BY o.order_id, c.customer_state
) AS state_orders
ORDER BY 1 desc;
```

customer_state	AOV
PB	216.67
AP	198.15
AC	197.32
AL	195.41
RO	186.8
PA	184.48
TO	177.86
PI	176.3

PI	176.3
MT	173.26
RN	172.27
CE	171.25
SE	170.79
RR	170.2
MS	164.76
MA	161.69
PE	159.46

POWER BI PROJECT OVERVIEW

Objective: To provide Walmart’s executive team with a comprehensive, interactive dashboard for monitoring and optimizing operations across sales, customers, regions, and time.

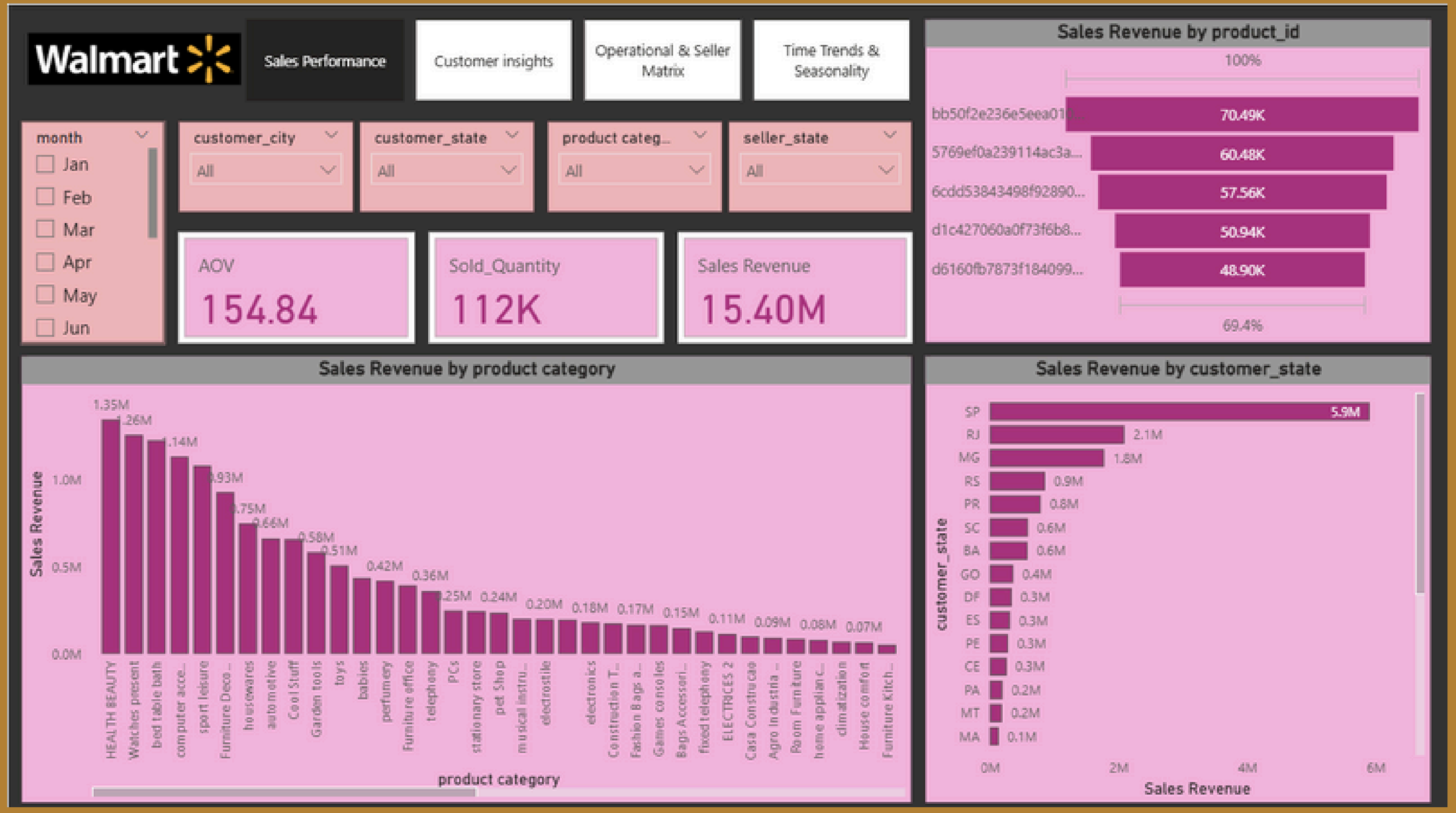
Data Sources Used:

- Customers
- Orders & Order Items
- Products
- Payments
- Sellers
- Geolocation

DASHBOARD HIGHLIGHTS / FEATURES

- Sales & Revenue Insights
- Customer Behavior & Segmentation
- Regional & Seller Performance
- Operational Efficiency
- Time-Based Trends & Seasonality
- Interactivity: Filters(Customer StateProduct, Category, Seller State, Time Period), Drill-downs to inspect specific customers, products, or sellers.

SALES PERFORMANCE



CUSTOMER INSIGHTS

Walmart

Sales Performance | Customer insights | Operational & Seller Matrix | Time Trends & Seasonality

seller_state: All

product category: All

customer_state: All

Loyal Customer %: 4.32%

month: Jan, Feb

customer_city: All

Customer Lifetime Value: 13.59M

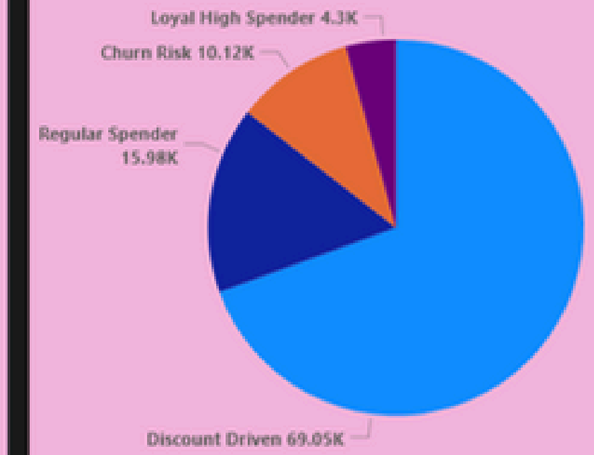
No of Loyal Customers: 4K

Top 10 Loyal Customer

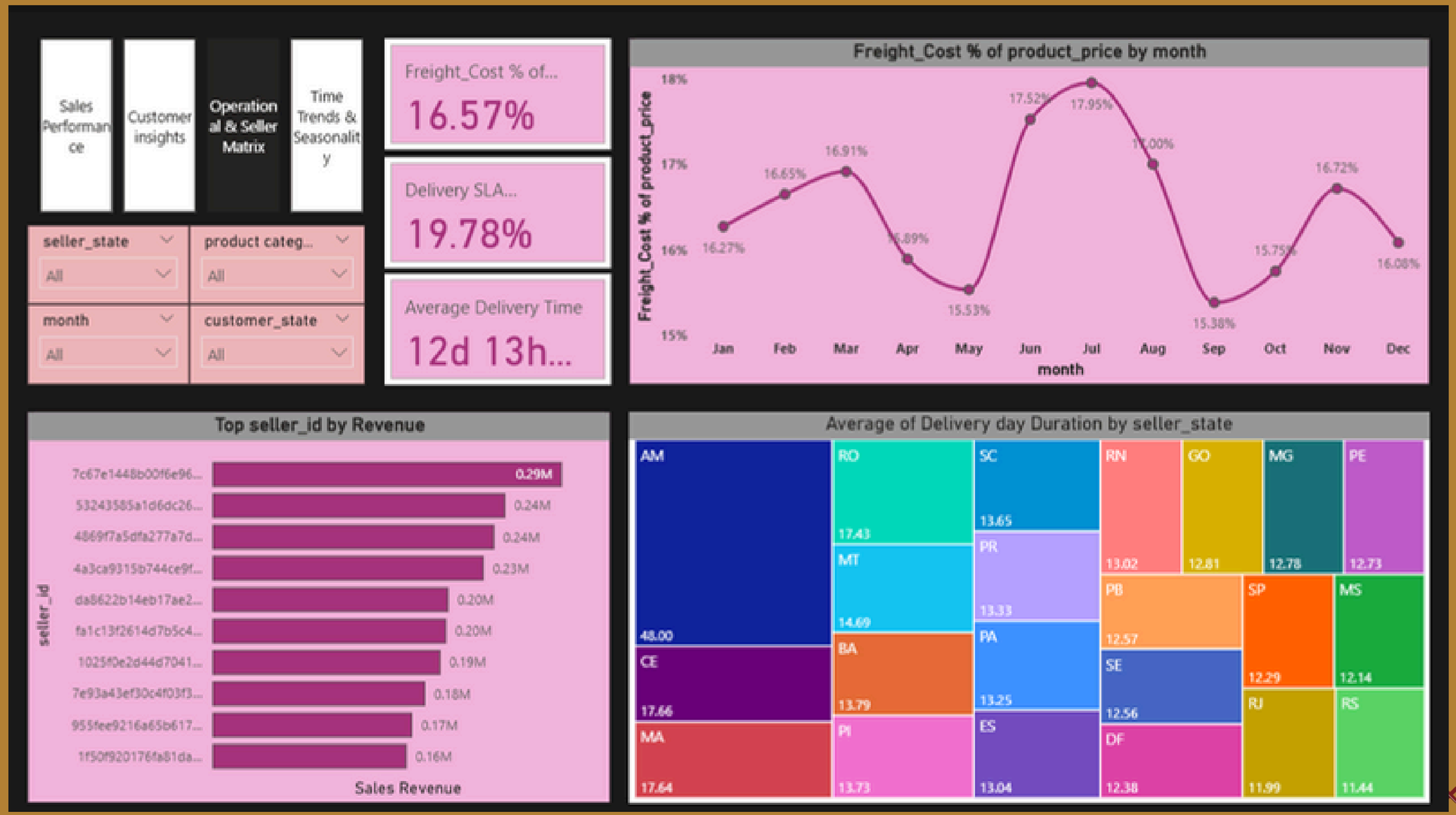
customer_id	Customer Total Spend	Customer Segment
05455dfa7cd02f13d132aa7a6a9729c6	6,081.54	Loyal High Spender
1617b1357756262bfa56ab541c47bc16	13,664.08	Loyal High Spender
24bbf5fd2f2e1b359ee7de94defc4a15	4,764.34	Loyal High Spender
3d979689f636322c62418b6346b1c6d2	4,681.78	Loyal High Spender
3fd6777bbce08a352fddd04e4a7cc8f6	6,726.66	Loyal High Spender
c6e2731c5b391845f6800c97401a43a9	6,929.31	Loyal High Spender
df55c14d1476a9a3467f131269c2477f	4,950.34	Loyal High Spender
e0a2412720e9ea4f26c1ac985f6a7358	4,809.44	Loyal High Spender
ec5b2ba62e574342386871631fafd3fc	7,274.88	Loyal High Spender
f48d464a0baaea338cb25f816991ab1f	6,922.21	Loyal High Spender
Total	**66,804.58**	

AOV by customer_state

Count of customer_id by Customer Segment



OPERATIONAL & SELLER MATRIX



TIME TRENDS & SEASONALITY

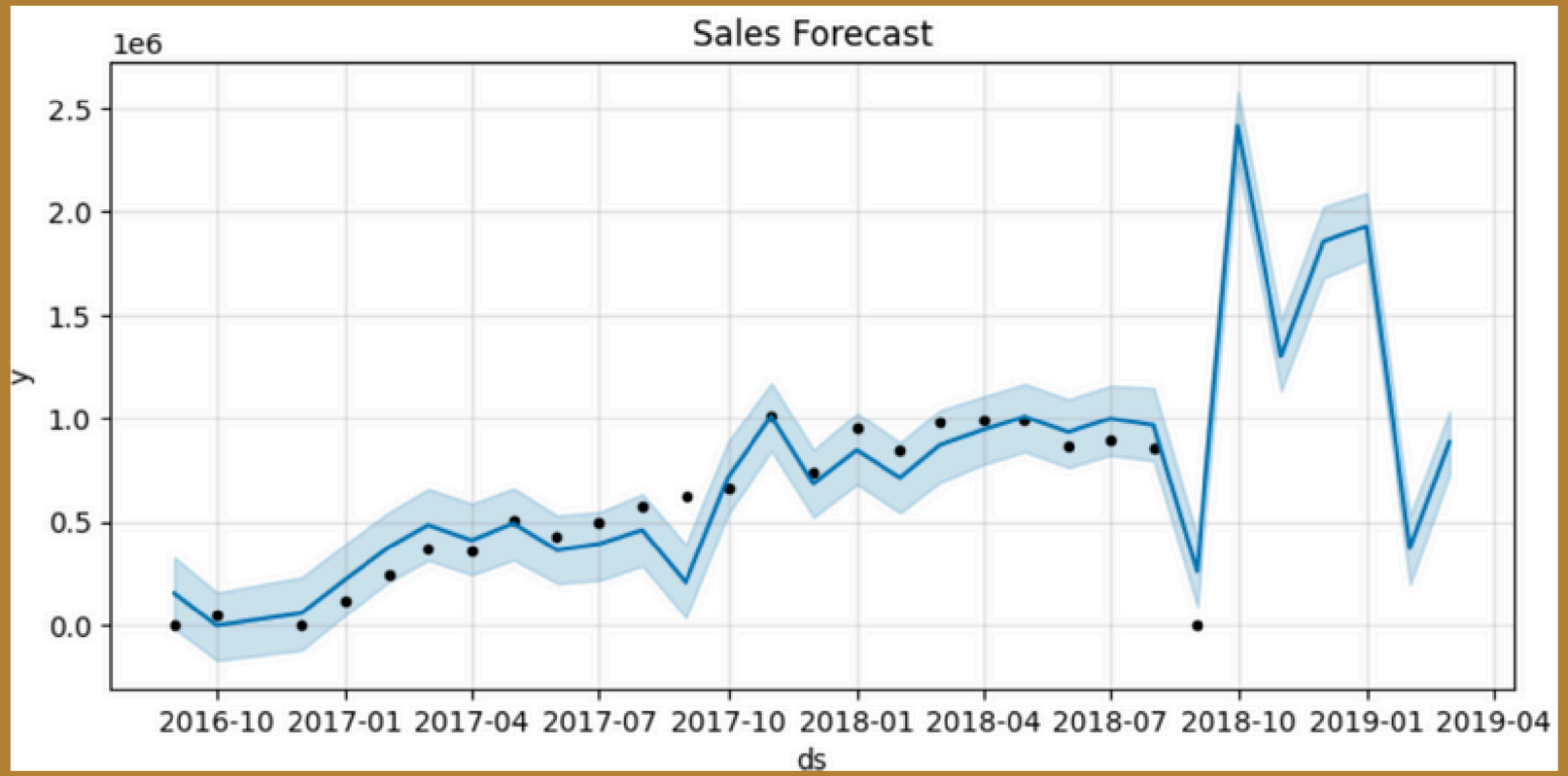
ML PROJECT OVERVIEW

- **Objective:**
- Analyze e-commerce sales data
- Forecast future sales and customer growth
- Segment customers using RFM
- Predict product returns using machine learning

Data Sources Used:

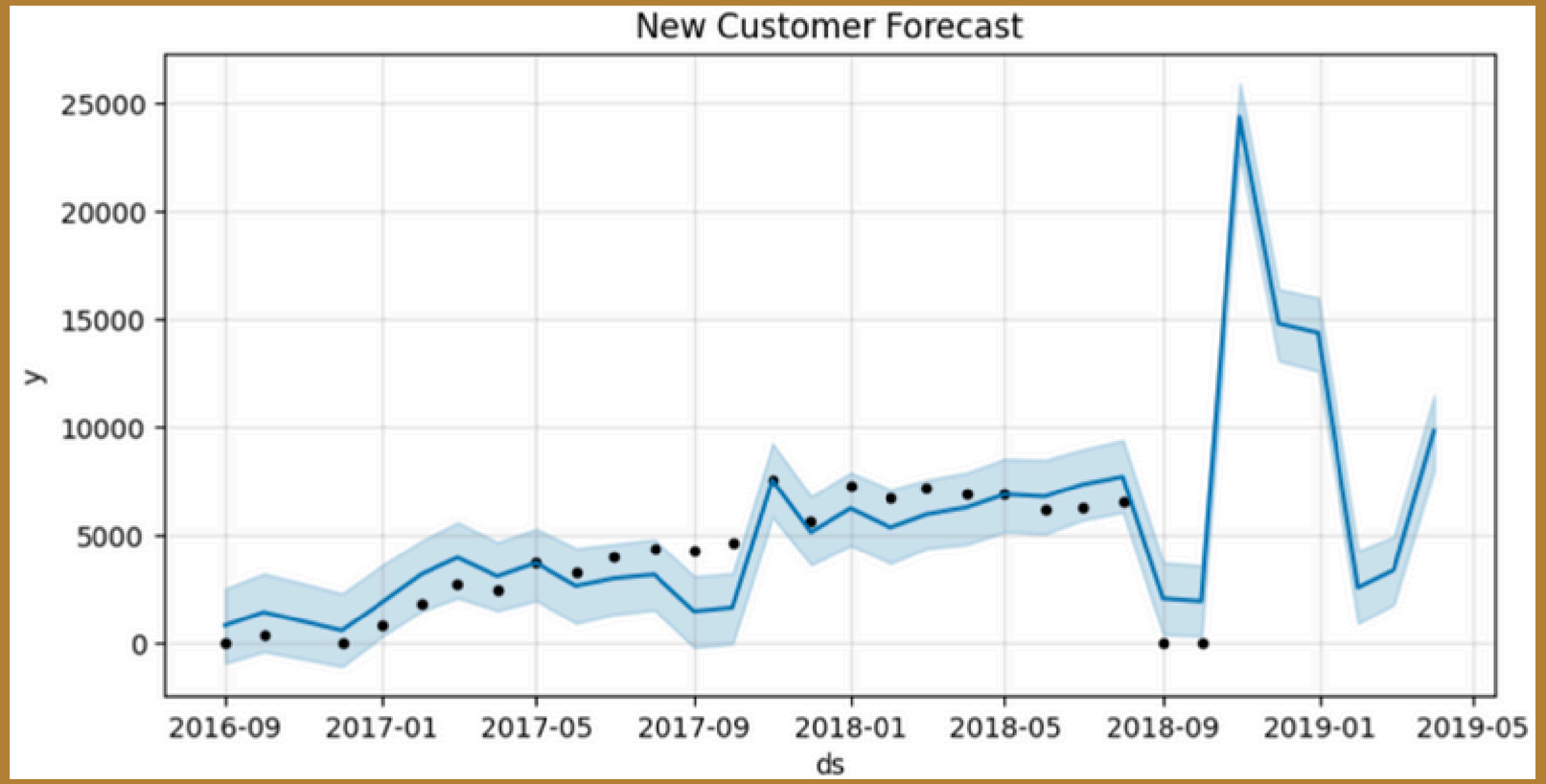
- Customers
- Orders & Order Items
- Products
- Sellers

SALES FORECASTING



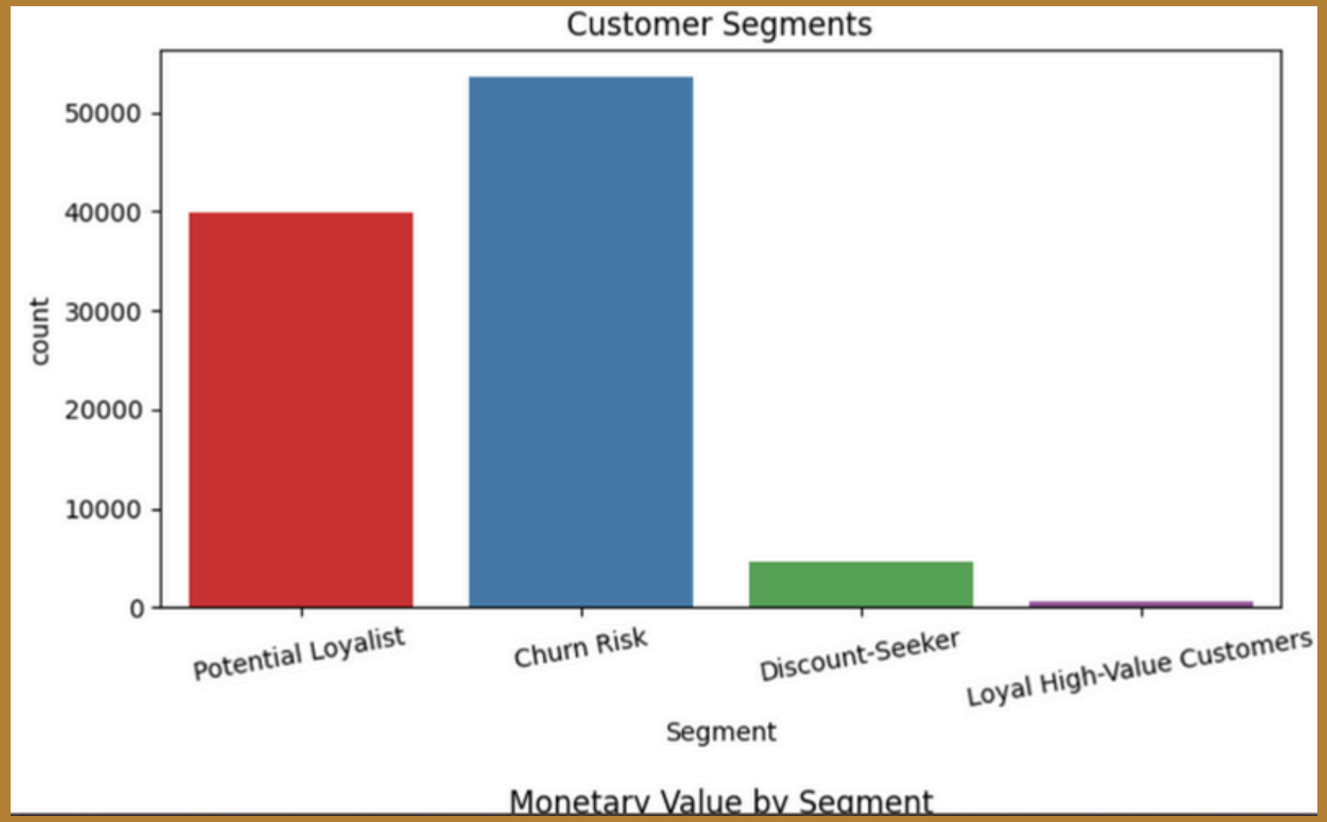
- Created monthly aggregated sales
- Used Prophet for 6-month forecast
- Insights: Seasonal patterns & expected revenue

CUSTOMER GROWTH FORECAST



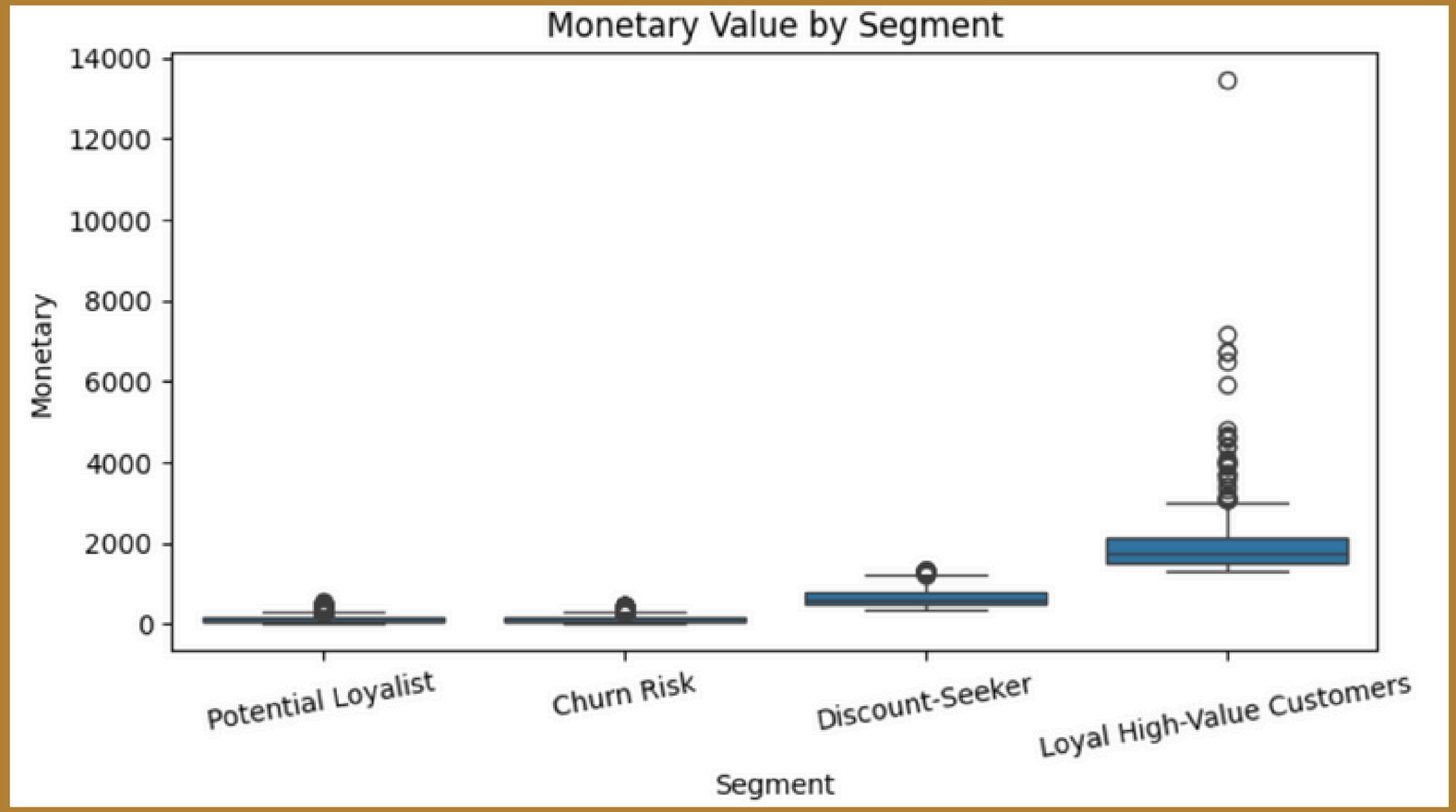
- Tracked unique monthly customers
- Used same Prophet approach
- Anticipated growth helps in marketing/resource planning

RFM ANALYSIS



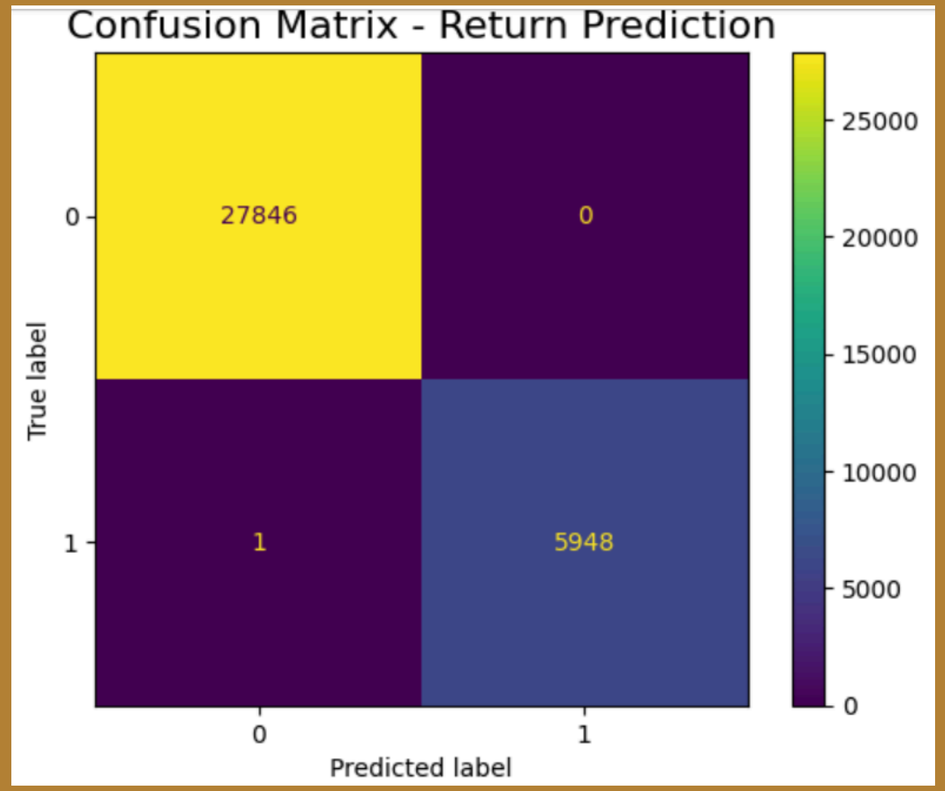
- Calculated Recency, Frequency, Monetary
- Used KMeans (4 clusters)
- Segments: Loyalists, Potential Loyalists, Discount Seekers, Churn Risk

CUSTOMER SEGMENTS VISUALIZATION



- Loyal customers have higher monetary value
- Churn risks are low-spending and infrequent

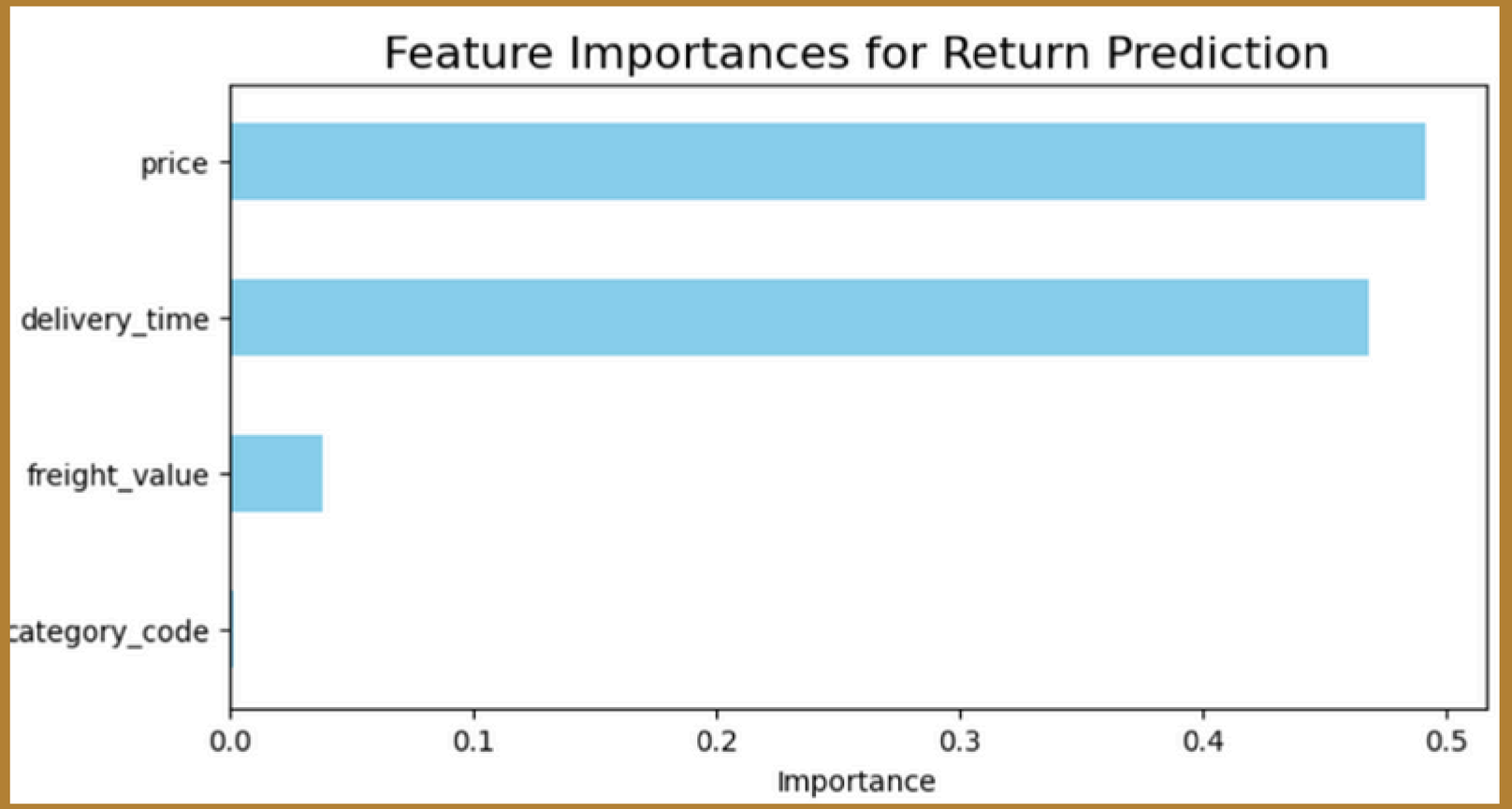
RETURN PREDICTION



	precision	recall	f1-score	support
0	1.00	1.00	1.00	27846
1	1.00	1.00	1.00	5949
accuracy			1.00	33795
macro avg	1.00	1.00	1.00	33795
weighted avg	1.00	1.00	1.00	33795

- RandomForest Classifier
- Features: Price, delivery time, freight, product category
- Accuracy metrics and confusion matrix

FEATURE IMPORTANCE



- Delivery time and product price are strong return predictors
- Helps in optimizing shipping and pricing strategies

SUMMARY & FUTURE WORK

- **Bullet Points:**
- Accurate forecasting with Prophet
- Actionable customer segmentation
- Effective return prediction using ML
- Future: Improve product-level forecasting, add real-time analytics

THANK YOU