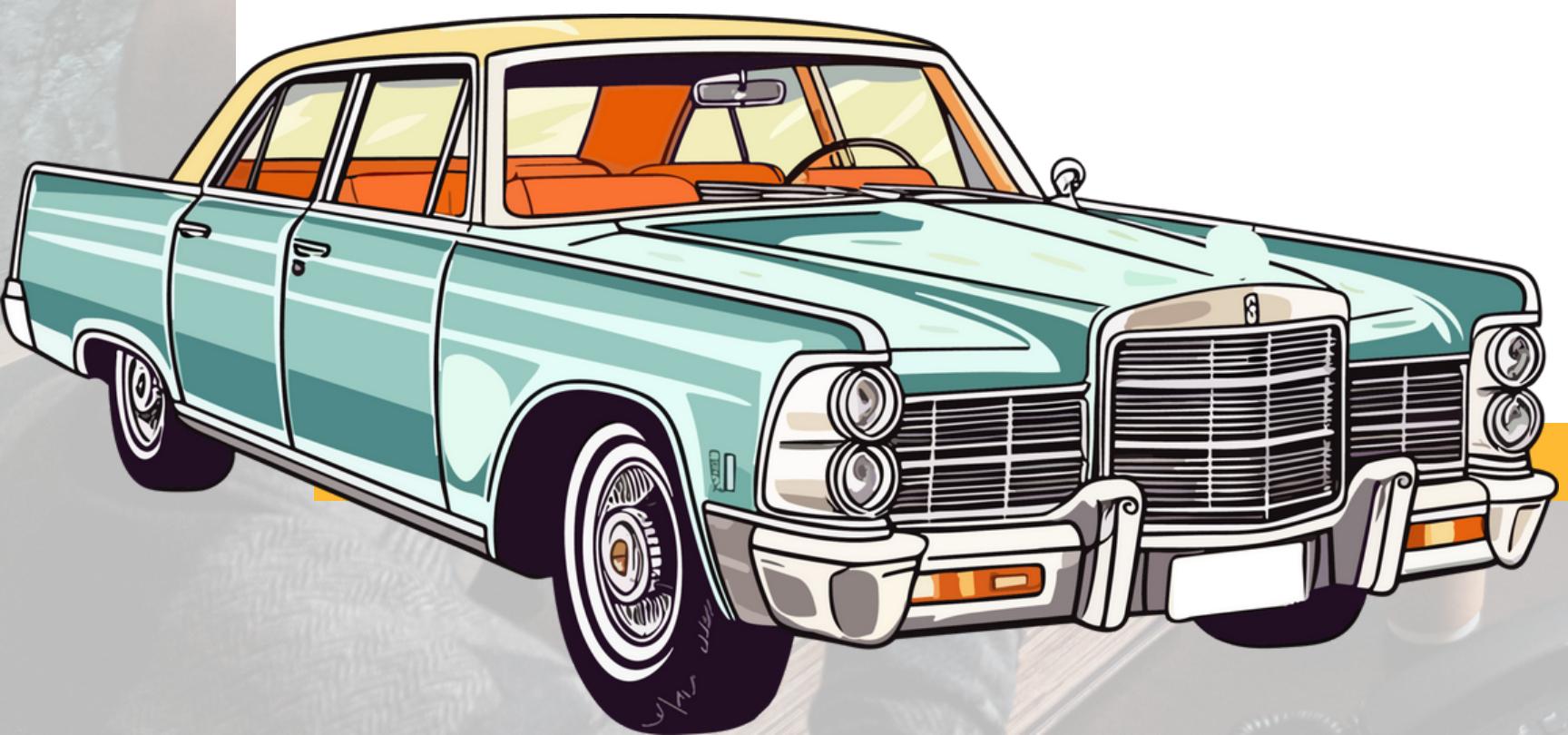


CLASSIC MODELS PROJECT ANALYSIS

BY ARYAN SHARMA



AGENDA

01

INTRODUCTION TO PROJECT

02

OBJECTIVE

03

DATA EXTRACTING

04

DATA ANALYSIS

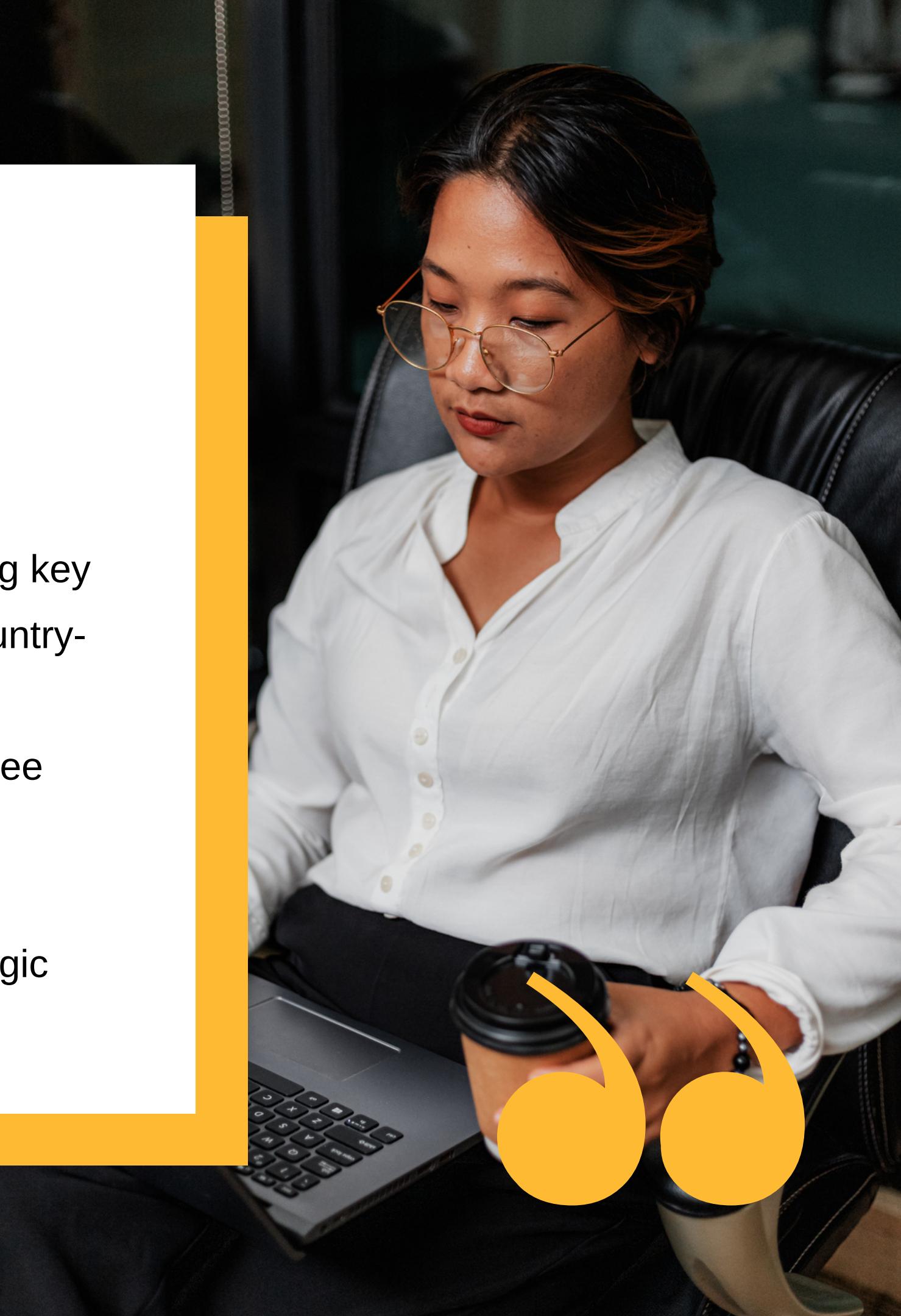
05

CONCLUSIONS



INTRODUCTION

Welcome, ladies and gentlemen. In this project, I leveraged Kaggle's classicmodels dataset, extracting and analyzing data in MySQL. Unveiling key insights, I identified the top 10 customers and employees, scrutinized country-wise distribution, assessed sales trends across years, and highlighted maximum sales offices. Delving deeper, I explored customer and employee density, pinpointed top and bottom products, and meticulously calculated margins. This presentation encapsulates a comprehensive exploration of Classic Models' business dynamics, providing valuable insights for strategic decision-making.



OBJECTIVE OF ANALYSIS



The objective of this project is to extract, analyze, and derive valuable insights from Kaggle's classicmodels dataset, meticulously organized in MySQL. By identifying the top 10 customers and employees, exploring country-wise distributions, dissecting sales trends across years, and scrutinizing office densities, this endeavor aims to provide a comprehensive understanding of Classic Models' operational landscape. Through detailed examination of product rankings and profit margins, the project seeks to offer strategic insights for optimizing business performance and decision-making processes.

Q1. TOP 10 CUSTOMERS OF COMPANY



```
select (customers.customerName) as Customer_name, (customers.contactFirstName) as Contact(firstName,  
       (count(orderdetails.quantityOrdered)) as Quantity_ordered  
  from customers inner join orders  
  on customers.customerNumber = orders.customerNumber  
  inner join  
  orderdetails on  
  orders.orderNumber = orderdetails.orderNumber  
 group by customers.customerName, customers.contactFirstName  
 order by count(orderdetails.quantityOrdered) desc limit 10;
```

Result Grid			
	Customer_name	Contact(firstName	Quantity_ordered
▶	Euro+ Shopping Channel	Diego	259
	Mini Gifts Distributors Ltd.	Susan	180
	Australian Collectors, Co.	Peter	55
	La Rochelle Gifts	Janine	53
	AV Stores, Co.	Rachel	51
	Land of Toys Inc.	Kwai	49
	Rovelli Gifts	Giovanni	48
	Musde Machine Inc	Jeff	48
	Kelly's Gift Shop	Tony	48
	Anna's Decorations, Ltd	Anna	46

Q2 TOP 10 EMPLOYEES OF COMPANY

```
-- Q2 top 10 employees of company
select employees.employeeNumber, employees.firstName, count(customers.customerNumber)
from employees inner join customers
on employees.employeeNumber = customers.salesRepEmployeeNumber
group by employees.employeeNumber, employees.firstName order by count(customers.customerNumber) desc limit 10;
```

Result Grid | Filter Rows: _____

	Employee_no	firstName	Total_customer_no
▶	1401	Pamela	10
	1504	Barry	9
	1323	George	8
	1501	Larry	8
	1286	Foon Yue	7
	1370	Gerard	7
	1165	Leslie	6
	1166	Leslie	6
	1188	Julie	6
	1216	Steve	6

Q3 EMPLOYEES AND CUSTOMERS DISTRIBUTION ON COUNTRY

-- Q3 Employees and customers distribution on country

```
select customers.country,customers.city, count(customers.customerNumber), count(customers.salesRepEmployeeNumber)
from customers group by customers.country,customers.city order by count(customers.salesRepEmployeeNumber) desc;
```

	country	city	total_customer_no	total_Employee_no
▶	USA	NYC	5	5
	Spain	Madrid	5	3
	France	Paris	3	3
	USA	Brickhaven	3	3
	New Zealand	Auckland	3	3
	France	Nantes	2	2
	USA	New Bedford	2	2
	USA	Glendale	2	2
	USA	New Haven	2	2
	USA	Philadelphia	2	2
	USA	Boston	2	2
	UK	London	2	2
	USA	San Francisco	2	2
	USA	Cambridge	2	2
	Singapore	Singapore	3	2
	USA	Las Vegas	1	1
	Australia	Melbourne	1	1
	Norway	Stavern	1	1
	USA	San Rafael	1	1

Q4. SALES DISTRIBUTION IN YEARS

```
-- Q4 Sales distribution in years
select year(orders.orderdate), count(orderdetails.quantityOrdered)
from orders inner join orderdetails
on orders.orderNumber = orderdetails.orderNumber
group by year(orders.orderdate) order by count(orderdetails.quantityOrdered) desc;
```

Result Grid | Filter Rows:

	years	quantityOrdered
▶	2004	1421
	2003	1052
	2005	523

Q5. MAX SALES OFFICES

```
-- Q5 Max sales offices  
  
select customers.city, (count(orderdetails.  
quantityOrdered)) as Total_orders  
  
from orderdetails inner join orders on  
orderdetails.orderNumber = orders.orderNumber  
  
inner join customers  
on customers.customerNumber = orders.customerNumber  
  
group by customers.city;
```

city	Total_orders
NYC	152
Madrid	304
Luleå	19
Kobenhavn	36
Lyon	41
Singapore	79
Allentown	31
Burlingame	34
Bergen	29
Lille	20
Paris	70
Cambridge	38
Bridgewater	25
Kita-ku	20
Helsinki	30
Manchester	51
Dublin	16
Brickhaven	47
Liverpool	29
Vancouver	22
Pasadena	30
Strasbourg	19
Central Hong...	16
Barcelona	23
Glendale	22

Q6 CUSTOMES' DENSITY ON COUNTRY'S BASIS

```
-- Q6 Customers' density on country's basis  
select customers.country, (count(customers.customerNumber)) as Total_customer  
from customers  
group by customers.country order by count(customers.customerNumber) desc ;
```

Result Grid		Filter Rows:
	country	Total_customer
▶	USA	36
	Germany	13
	France	12
	Spain	7
	Australia	5
	UK	5
	Italy	4
	New Zealand	4
	Norway	3
	Singapore	3
	Finland	3
	Canada	3
	Switzerland	3
	Sweden	2
	Denmark	2
	Portugal	2
	Japan	2
	Ireland	2

Q7. EMPLOYEES' DENSITY ON OFFICE BASIS

```
select employees.employeeNumber, employees.firstName, offices.officeCode, offices.city, (count(employees.employeeNumber)
over(partition by offices.city)) AS City_offices
from employees inner join offices
on employees.officeCode = offices.officeCode
group by employees.employeeNumber, employees.firstName, offices.officeCode, offices.city order by City_offices
```

	employeeNumber	firstName	officeCode	city	City_offices
▶	1002	Diane	1	San Francisco	6
	1056	Mary	1	San Francisco	6
	1076	Jeff	1	San Francisco	6
	1143	Anthony	1	San Francisco	6
	1165	Leslie	1	San Francisco	6
	1166	Leslie	1	San Francisco	6
	1102	Gerard	4	Paris	5
	1337	Loui	4	Paris	5
	1370	Gerard	4	Paris	5
	1401	Pamela	4	Paris	5
	1702	Martin	4	Paris	5
	1088	William	6	Sydney	4
	1611	Andy	6	Sydney	4
	1612	Peter	6	Sydney	4
	1619	Tom	6	Sydney	4
	1188	Julie	2	Boston	2
	1216	Steve	2	Boston	2
	1501	Larry	7	London	2
	1504	Barry	7	London	2
	1286	Foon Yue	3	NYC	2
	1323	George	3	NYC	2
	1621	Mami	5	Tokyo	2
	1625	Yoshimi	5	Tokyo	2

Q8. TOP 10 PRODUCTS

```
-- Q8 TOP 10 products
select productlines.productLine, products.productName, (count(orderdetails.quantityOrdered)) as QuantityOrdered
from productlines inner join products on
productlines.productLine = products.productLine
inner join orderdetails
on orderdetails.productCode = products.productCode
group by productlines.productLine, products.productName order by count(orderdetails.quantityOrdered) desc limit
10;
```

	productLine	productName	QuantityOrdered
▶	Classic Cars	1992 Ferrari 360 Spider red	53
	Classic Cars	1952 Alpine Renault 1300	28
	Classic Cars	1972 Alfa Romeo GTA	28
	Classic Cars	1962 Lancia Delta 16V	28
	Classic Cars	1958 Chevy Corvette Limited Edition	28
	Vintage Cars	1930 Buick Marquette Phaeton	28
	Motorcycles	1996 Moto Guzzi 1100i	28
	Classic Cars	1970 Dodge Coronet	28
	Motorcycles	1969 Harley Davidson Ultimate Chopper	28
	Classic Cars	1982 Camaro Z28	28

Q9. TOP 10 BOTTOM PRODUCTS

```
select productlines.productLine, products.productName, (count(orderdetails.quantityOrdered)) as quantityOrdered  
from productlines inner join products on  
productlines.productLine = products.productLine  
inner join orderdetails  
on orderdetails.productCode = products.productCode  
group by productlines.productLine, products.productName order by count(orderdetails.quantityOrdered) limit 10;
```

	productLine	productName	quantityOrdered
▶	Classic Cars	1957 Ford Thunderbird	24
	Classic Cars	1952 Citroen-15CV	24
	Classic Cars	1969 Chevrolet Camaro Z28	25
	Classic Cars	1999 Indy 500 Monte Carlo SS	25
	Classic Cars	1965 Aston Martin DB5	25
	Classic Cars	1949 Jaguar XK 120	25
	Classic Cars	2002 Chevy Corvette	25
	Classic Cars	1970 Chevy Chevelle SS 454	25
	Classic Cars	1948 Porsche Type 356 Roadster	25
	Classic Cars	1966 Shelby Cobra 427 S/C	25

Q10 TOP 10 MARGINAL PRODUCTS

NOTE: As there was no column of margin in the data. Therefore I made a Querry to create a new column in exisiting data as "Profit_margin"

```
select productLine, avg(MSRP - buyPrice) as pm from products  
group by productLine;
```

Querry:

```
select products.productLine, products.productName,  
products.MSRP, products.buyPrice, (MSRP - buyPrice)  
as Profit_margin  
from products order by products.profit_margin desc  
limit 10;
```

	productLine	productName	quantityOrdered
▶	Classic Cars	1957 Ford Thunderbird	24
	Classic Cars	1952 Citroen-15CV	24
	Classic Cars	1969 Chevrolet Camaro Z28	25
	Classic Cars	1999 Indy 500 Monte Carlo SS	25
	Classic Cars	1965 Aston Martin DB5	25
	Classic Cars	1949 Jaguar XK 120	25
	Classic Cars	2002 Chevy Corvette	25
	Classic Cars	1970 Chevy Chevelle SS 454	25
	Classic Cars	1948 Porsche Type 356 Roadster	25
	Classic Cars	1966 Shelby Cobra 427 S/C	25

CONCLUSION

1. Strategic Marketing:

- Tailor marketing efforts based on hot-selling products and top customer preferences

2. Customer Engagement:

- Implement customer-centric strategies for the top 10 customers to enhance loyalty

3. Supply Chain Optimization:

- Streamline inventory management based on product profitability and sales trends

4. Forecasting Accuracy:

- Refine forecasting models for better accuracy, reducing the risk of stockouts or overstock

5. Pricing Strategy Refinement:

- Adjust pricing strategies to maximize profit margins and maintain competitiveness

6. Product Portfolio Management:

- Optimize the product mix by focusing on the top 5 products with the highest margins

THANK YOU!

DO
LIKE,
COMMENT
AND SHARE



www.linkedin.com/in/aryan-sharma14



aarryansha1407@gmail.com



<https://github.com/AryanSharma1407>