SUPER MARKET SALES ANALYSIS USING SQL

AHILA SASEENDRAN

INTRODUCTION

This report details an in-depth Exploratory Data Analysis (EDA) of supermarket sales data, utilizing SQL to efficiently extract, query, and analyse large datasets for meaningful insights. Supermarkets generate vast amounts of data across various categories, including product sales, customer demographics, transaction details, and seasonal trends. Through SQL-based analysis, this report aims to systematically investigate these datasets, identifying patterns, trends, and anomalies that could influence business strategies. By analysing sales performance across different product categories, regions, and time frames, this EDA provides a robust understanding of customer purchasing behaviour, peak sales periods, and high-demand products. These insights can serve as a foundation for targeted marketing, inventory optimization, and strategic planning, ultimately helping the supermarket to enhance profitability and customer satisfaction. Additionally, this report demonstrates the effectiveness of SQL as a tool for conducting data analysis in a business context, emphasizing its capability to handle complex queries and large-scale data efficiently.

Exploratory Data Analysis (EDA)

Exploratory Data Analysis (EDA) is an approach in data science used to analyse datasets to summarize their main characteristics, often with the help of visualizations and descriptive statistics. The primary goal of EDA is to uncover patterns, detect anomalies, test hypotheses, and check assumptions, providing a foundational understanding of the data before proceeding with more complex modelling or data processing tasks.

EDA involves several key steps:

1.Data Collection And Understanding
First, we gather data from relevant sources and examine its structure, types, and dimensions. Understanding the context and variables within the data is crucial, as it helps shape the direction of analysis and decision-making.

2.Data Cleaning
This step includes handling missing values, correcting data types, dealing with outliers, and ensuring data consistency.

Data cleaning is essential to enhance the quality and reliability of insights derived from the data.

3.Feature Engineering

It involves transforming raw data into meaningful features that improve the performance of machine learning models and enhances the dataset by creating, modifying, or selecting variables that reveal patterns and improve the interpretability of data.

4.Univariate Analysis

This involves examining each variable individually, often through visualizations like histograms, box plots, and bar charts, to understand its distribution and key characteristics.

5.Bivariate Analysis

It is a statistical method used to explore the relationship between two variables. It aims to understand how the two variables interact, influence each other, or show correlations.

6.Multivariate Analysis

It is a statistical approach used to examine more than two variables simultaneously to understand complex relationships within a dataset. Unlike univariate or bivariate analysis, which focuses on one or two variables, respectively, multivariate analysis considers multiple variables to gain deeper insights, detect patterns, and capture interactions that may be missed when examining variables individually.

DATA SET DESCRIPTION

Data set source : Kaggle
Rows :1001
Columns :17
The data set contains the following feature :
Invoice ID : It is a unique identifier assigned to each invoice generated by a business. It helps in tracking, managing, and organizing invoices within accounting systems and is essential for accurate financial record-keeping.

City : It refers to the location where a supermarket or retail store operates. Analysing data based on city can reveal valuable insights into regional sales patterns, customer preferences, and purchasing behaviours.

Customer Type : It shows that whether the customer is member of the super market who has membership or normal customers who visit the store occasionally.

Gender : The female and male customers are identifies in order to know who makes shopping more.

Product Line : It is a group of related products that are marketed under a single brand or category. These products share common characteristics, functions, or target consumers, allowing for streamlined marketing and inventory management.

Unit Price : It is the cost per single unit of a product, allowing consumers to compare prices of items sold in different quantities or sizes more easily.

Quantity : It is the amount or number of items, units, or measurements of a particular product or substance.

Tax : It is a mandatory financial charge or levy imposed by a government on individuals, businesses, and other entities to fund public services and infrastructure.

Total : It refers to the overall revenue generated from the sale of goods or services within a specified period. It is a key performance metric for businesses, including supermarkets, and is often used to evaluate financial performance, assess market demand, and inform strategic decisions.

Date : It refer to several important aspects related to inventory management, product quality, sales transactions, and customer service.

Time : It is a crucial factor that influences various operations, customer experiences, and inventory management.

Payment : It shows the method of payment done by the customer via credit card E Wallet, Cash

Cogs (Cost of goods sold) : It is to the direct costs associated with the production of goods that a company sells during a specific period.

Gross percentage : It is commonly referred to as *gross profit percentage* or *gross margin percentage*, is a financial metric that indicates the percentage of revenue that exceeds the cost of goods sold (COGS).

Gross Income : It refers to the total revenue earned by a business from its operations before any expenses, taxes, or deductions are subtracted. It represents the income generated from sales of goods or services and is a critical metric for assessing the overall financial performance of a company, including supermarkets.

Rating : It refers to the assessment or evaluation of a product, service, or overall customer experience based on certain criteria.

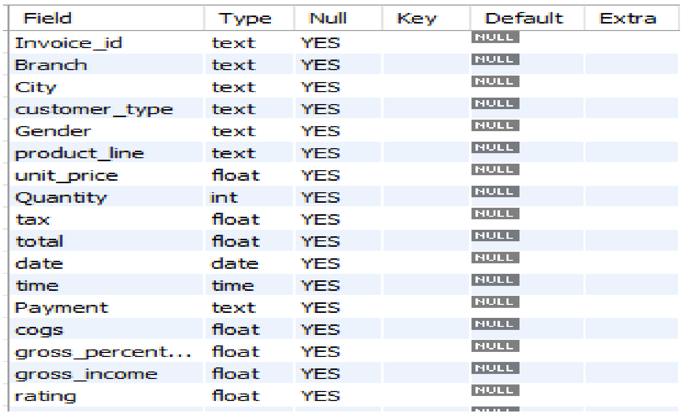
DATA CLEANING AND PREPROCESSING

```
alter table supermarket rename column `Invoice ID` to Invoice_id;
alter table supermarket rename column `Tax 5%` to Tax;
alter table supermarket rename column `customer type` to customer_type;
alter table supermarket rename column `product line` to product_line;
alter table supermarket rename column `unit price` to unit_price;
alter table supermarket rename column `gross margin percentage` to gross_percentage;
alter table supermarket rename column `gross income` to gross_income;
```

Rename id done to access the column easily with the column name the space between the Invoice ID.

```
alter table supermarket modify unit_price float;
alter table supermarket modify tax float;
alter table supermarket modify total float;
alter table supermarket modify cogs float;
alter table supermarket modify unit_price float;
alter table supermarket modify gross_percentage float;
alter table supermarket modify gross_income float;
alter table supermarket modify rating float;
alter table supermarket modify date date;
alter table supermarket modify time time;
```

```
describe supermarket;
```



Field	Type	Null	Key	Default	Extra
Invoice_id	text	YES		NULL	
Branch	text	YES		NULL	
City	text	YES		NULL	
customer_type	text	YES		NULL	
Gender	text	YES		NULL	
product_line	text	YES		NULL	
unit_price	float	YES		NULL	
Quantity	int	YES		NULL	
tax	float	YES		NULL	
total	float	YES		NULL	
date	date	YES		NULL	
time	time	YES		NULL	
Payment	text	YES		NULL	
cogs	float	YES		NULL	
gross_percent...	float	YES		NULL	
gross_income	float	YES		NULL	
rating	float	YES		NULL	

The data type for the date, time was in the form of text it was removed and converted into date, time. The decimal values was given as int data type it has been converted into float.

Checking Missing Values :

```
select count(*)-count(invoice_id) from supermarket group by invoice_id ;
select count(*)-count(branch) from supermarket group by branch ;
select count(*)-count(city) from supermarket group by city ;
select count(*)-count(customer_type) from supermarket group by customer_type ;
select count(*)-count(gender) from supermarket group by gender ;
select count(*)-count(product_line) from supermarket group by product_line ;
select count(*)-count(unit_price) from supermarket group by unit_price ;
select count(*)-count(quantity) from supermarket group by quantity ;
select count(*)-count(tax) from supermarket group by tax;
select count(*)-count(total) from supermarket group by total;
select count(*)-count(date) from supermarket group by date;
select count(*)-count(time) from supermarket group by time ;
select count(*)-count(payment) from supermarket group by payment;
select count(*)-count(cogs) from supermarket group by cogs ;
select count(*)-count(gross_percentage) from supermarket group by gross_percentage ;
select count(*)-count(gross_income) from supermarket group by gross_income;
select count(*)-count(rating) from supermarket group by rating ;
select count(distinct(Invoice_id)) from supermarket;
```

Checking NULL values :

```
select count(*) from supermarket where Invoice_id is null;
select count(*) from supermarket where Branch is null;
select count(*) from supermarket where city is null;
select count(*) from supermarket where customer_type is null;
select count(*) from supermarket where gender is null;
select count(*) from supermarket where product_line is null;
select count(*) from supermarket where unit_price is null;
```

```sql
select count(*) from supermarket where Quantity is null;
select count(*) from supermarket where tax is null;
select count(*) from supermarket where total is null;
select count(*) from supermarket where date is null;
select count(*) from supermarket where time is null;
select count(*) from supermarket where payment is null;
select count(*) from supermarket where cogs is null;
select count(*) from supermarket where gross_percentage is null;
select count(*) from supermarket where gross_income is null;
select count(*) from supermarket where rating is null;
```

Checking Outliers :

```sql
select avg(gross_income) as avg_income,stddev(gross_income) as
stddev_income from supermarket;
select * from supermarket where gross_income > (select avg(gross_income) + 3
* stddev(gross_income) from supermarket)
or gross_income < (select avg(gross_income) - 3 * stddev(gross_income) from
supermarket);
select * from supermarket where unit_price > (select avg(unit_price) + 3 *
stddev(unit_price) from supermarket)
or unit_price < (select avg(unit_price) - 3 * stddev(unit_price) from
supermarket);
```

RESULT :
Data cleaning is completed all the missing values and null values is checked and verified that there are no missing values and NULL values and the data type for all the column has been converted into INT, TEXT , FLOAT .

Ensured consistent formats for dates, times, and other fields is essential for accurate analysis. For example, dates should be standardized to a specific format (e.g., YYYY-MM-DD).

Duplicate entries skew analysis and lead to inaccurate results. Identified and removing duplicates ensures data integrity verified that there are no duplicate entries.

FEATURE ENGINEERING

```
alter table supermarket add column revenue_per_item float;
SET SQL_SAFE_UPDATES =0;
update supermarket set revenue_per_item = total / quantity; #add a column revenue (total_sales)
alter table supermarket add column original_price FLOAT;
update supermarket set original_price = unit_price / (1 - 0.1); # Assuming a 10% average discount
alter table supermarket add column discount_percentage FLOAT;
UPDATE supermarket
SET discount_percentage =
   CASE
      WHEN customer_type = 'Member' THEN ((original_price - unit_price) / original_price) * 100
      ELSE 0
   END; #the members are given 10% discount others have no discount
alter table supermarket add column day_part VARCHAR(10);
update supermarket
set day_part =
   case
      when hour(Time) between 5 and 12 then 'Morning'
      when hour(Time) between 12 and 17 then 'Afternoon'
      else 'Evening'
   end;
```

Adding new columns during feature engineering is a critical step in preparing your data for analysis and modeling. In the context of a supermarket, this can involve creating derived features that enhance the dataset's informative value, helping to improve predictions and insights.

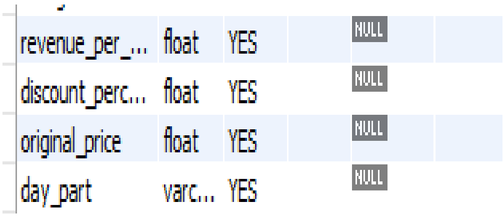
Revenue : It refers to the total amount of money generated by a supermarket from its operations over a specific period.

Discount Percentage : It is given 10% discount for the member who has membership and normal customer who visits the store occasionally is not given any discount.

Orginal Price : The price by subtracting the discount is the orginal price.

Day Part : The day time which represents the morning , afternoon , evening where most sales takes place and to analyse it.

describe supermarket;



revenue_per_...	float	YES		NULL	
discount_perc...	float	YES		NULL	
original_price	float	YES		NULL	
day_part	varc...	YES		NULL	

Result :

By adding these new columns revenue, discounted percentage , orginal price Day time. enhance the dataset's analytical capabilities, allowing for deeper insights into sales performance, customer purchasing behaviour, and pricing strategies within the supermarket context. These features can support more informed decision-making and strategy formulation to improve business outcomes.

UNIVARIATE ANALYSIS

Univariate analysis refers to the statistical examination of a single variable to understand its distribution, central tendency, and variability. In the context of a supermarket, univariate analysis is essential for gaining insights into individual features within the dataset. This analysis helps supermarkets make informed decisions regarding inventory management, pricing strategies, customer behavior, and marketing efforts.

```
select * from supermarket order by Invoice_ID limit 5 ;
select * from supermarket order by rand() limit 5;
selectcount(gross_income),min(gross_income),max(gross_income),avg(gross_income),std(gross_income) from supermarket;
select distinct(City) from supermarket ;
select Payment,count(*) as counts from supermarket group by Payment order by counts desc;
select product_line,avg(rating) from supermarket group by product_line order by avg(rating) desc;
select product_line,sum(Quantity) as sum from supermarket
group by product_line order by sum desc;
select distinct(branch) from supermarket;
select customer_type from supermarket where branch='A';
select customer_type from supermarket where branch='B';
select customer_type from supermarket where branch='C';
select count(customer_type) from supermarket  where gender='female';
select count(customer_type) from supermarket where gender='male';
select distinct(payment) from supermarket;
select count(Invoice_id) as counts from supermarket where payment='cash';
select count(Invoice_id) as counts from supermarket where payment='credit card';
select count(Invoice_id) as counts from supermarket where payment='ewallet';
select count(customer_type) as customer_count from supermarket where customer_type='Member';
select count(customer_type) as customer_count from supermarket where customer_type='normal';
select product_line,sum(tax) as total_tax from supermarket group by
product_line order by sum(tax) desc;
select product_line, avg(Quantity) as average_quantity from supermarket group
by product_line;
select gender, count(distinct Invoice_id) AS unique_customers from
supermarket group by gender;
select branch,count(Invoice_id) as sales_count from supermarket group by
branch;
select city,count(Invoice_id) as sales_count from supermarket group by city;
select product_line,count(*) as transaction_count from supermarket group by
product_line order by transaction_count desc;
select product_line,count(*) as transaction_count from supermarket group by
product_line order by transaction_count desc;
select avg(cogs) from supermarket;
select customer_type from supermarket;
select customer_type from supermarket where discount_percentage !=0;
select avg(gross_income) from supermarket;
select Invoice_id,sum(quantity) as total_quantity from supermarket group by
Invoice_id;
```

Result :

1. The super market is present in cities named Yangon ,Naypyitaw, Mandalay which has branch named A,B,C.

2. Mostly the payment is done through E Wallet and the least payment is done through credit card.

3. The product line Electronic accessories have high sales and the product line sales and beauty have low sales.

4. There are 2 distinct customer type in super market member and normal ,member have the membership.

5. The female customers is higher than male customers.

6. The branch A has highest sales and branch C has lowest sales.

BIVARIATE ANALYSIS

Bivariate analysis in the context of a supermarket involves examining the relationship between two variables to identify patterns, correlations, or trends that can influence business decisions. This analysis helps supermarket managers or data analysts understand how different factors interact and affect outcomes like sales, customer preferences, inventory turnover, and more.

```
select Branch,sum(gross_income) as sum_gross_income from supermarket
group by Branch order by sum_gross_income desc;
select dayname(date),dayofweek(date),sum(Total) from supermarket group by
dayname(date),dayofweek(date)
order by sum(total) desc;
select monthname(date) as name,month(date) as month,sum(Total) as total
from supermarket group by name,month order by total desc;
select hour(Time) as hour,sum(Total) as total from supermarket
group by hour order by total desc;
select Gender,avg(gross_income) from supermarket group by gender;
select gender,count(*),product_line from supermarket where
product_line='Health and beauty' group  by gender ;
select count(gender) as customer_count,product_line from supermarket group
by product_line;
select product_line,avg(unit_price) as average_unit_price from supermarket
group by product_line order by average_unit_price asc;
select distinct(product_line),gross_percentage from supermarket;
select product_line,avg(tax) as average_tax from supermarket group by
product_line;
select product_line,min(tax) as minimum_tax from supermarket group by
product_line order by minimum_tax asc;
SELECT city,
   SUM(CASE WHEN Payment = "Cash" THEN 1 ELSE 0 END) AS "Cash",
   SUM(CASE WHEN Payment = "Ewallet" THEN 1 ELSE 0 END) AS "Ewallet",
   SUM(CASE WHEN Payment = "Credit card" THEN 1 ELSE 0 END) AS "Credit
card"
FROM supermarket GROUP BY City;
```

```
select product_line,rating from supermarket; #checks rating based on product
line
select distinct(product_line),rating from supermarket where rating>9;
select product_line,avg(gross_income) as average_gross from supermarket
group by product_line;
select city,product_line,avg(cogs) from supermarket group by city,product_line;
select city,product_line,min(cogs) from supermarket group by city,product_line;
select city,product_line,max(cogs) from supermarket group by city,product_line
order by max(cogs) desc;
SELECT Branch, customer_type, AVG(gross_income) AS average_gross_income
FROM supermarket GROUP BY Branch, customer_type ORDER BY
average_gross_income DESC;
SELECT Gender, product_line, SUM(Total) AS total_sales FROM supermarket
GROUP BY Gender, product_line ORDER BY total_sales DESC;
SELECT product_line, city, AVG(tax) AS average_tax
FROM supermarket GROUP BY product_line, city ORDER BY average_tax ASC;
SELECT product_line, MONTH(date) AS month, SUM(Quantity) AS total_quantity
FROM supermarket
GROUP BY product_line, month ORDER BY total_quantity DESC;
SELECT city, Payment, SUM(Total) AS total_sales_amount FROM supermarket
GROUP BY city, Payment ORDER BY total_sales_amount DESC;
SELECT Branch, product_line, AVG(rating) AS average_rating
FROM supermarket GROUP BY Branch, product_line ORDER BY average_rating
DESC;
```

Result :

1. The highest sales takes place on 7th day of every week (Saturday).

2. January, February, March are the 3 months where highest sales takes place in the year 2019.

3. The highest sales takes during morning 10 AM to night 8 PM.

4. Female customers spends more money in shopping.

5. The product line health and beauty is the lowest sales count where male =88 and female =64.

6. The highest unit price is for fashion accessories

7. Health and beauty product line have highest tax.

8. Sports and travel has highest rating and Fashion accessories have lowest rating.

9. In Yangon the average of goods sold for the product line is less compared to other branches of the super market.

10. Female customers spends more in product line food and beverages but less in health and beauty.Male customers spends more in health and beauty.

11. Sports and travel ,Home and life style and Fashion accessories has higher amount of sales during the month of January and at the start of every year.

12. Fashion accessories have average highest rating in the branch C.

MULTIVARIATE ANALYSIS

Multivariate analysis in a supermarket involves examining multiple variables simultaneously to understand complex relationships that can influence store performance, customer behaviour, and inventory management. Unlike bivariate analysis, which looks at only two variables at a time, multivariate analysis considers the interplay among three or more variables, giving a more comprehensive view of patterns and trends. This type of analysis is essential for identifying deeper insights that are not apparent when analysing variables in isolation.

```
SELECT date,product_line,SUM(quantity) AS total_quantity FROM supermarket
GROUP BY date, product_line;
SELECT customer_type, Payment, AVG(Total) AS average_purchase
FROM supermarket GROUP BY customer_type, Payment ORDER BY
average_purchase DESC;
SELECT city,product_line,SUM(Quantity) AS total_quantity FROM supermarket
GROUP BY city, product_line ORDER BY city, total_quantity DESC;
SELECT customer_type, Payment, AVG(Total) AS average_purchase
FROM supermarket GROUP BY customer_type, Payment ORDER BY
average_purchase DESC;
SELECT Date,time, product_line, SUM(Total) AS total_sales
FROM supermarket GROUP BY Date,time, product_line ORDER BY Date ASC;
#Trend over time
SELECT branch, product_line, revenue_per_item FROM supermarket ORDER BY
revenue_per_item DESC;
SELECT City, Branch, SUM(gross_income) AS sum FROM supermarket GROUP
BY City, Branch;
SELECT customer_type, COUNT(*) AS visit_count,
   CASE
      WHEN COUNT(*) >= 10 THEN 'Frequent'
      WHEN COUNT(*) BETWEEN 5 AND 9 THEN 'Moderate'
      ELSE 'Occasional'
   END AS frequency_category
FROM supermarket
GROUP BY customer_type;
```

```
select Invoice_id,product_line,customer_type,(original_price-
discount_percentage) as price_after_discount from supermarket
where customer_type='Member' or customer_type='Normal' ;
select distinct(city),total as total_sales from supermarket where
day_part='Morning';
SELECT branch,product_line,customer_type,SUM(gross_income) AS
total_gross_income
FROM supermarket GROUP BY branch, product_line, customer_type ORDER BY
total_gross_income DESC;
SELECT HOUR(time) AS hour_of_day,product_line,branch,SUM(Total) AS
total_sales FROM supermarket
GROUP BY hour_of_day, product_line, branch ORDER BY hour_of_day,
total_sales DESC;
```

Result :

1. Health and beauty has highest quantity of sales on 2019-01-05

2. Super market located at city Mandalay has highest sales on product line Sports and travel

3. During the first month of year highest sales takes place on Food and Beverages.

4. Sports and travel has highest revenue in the branch B.

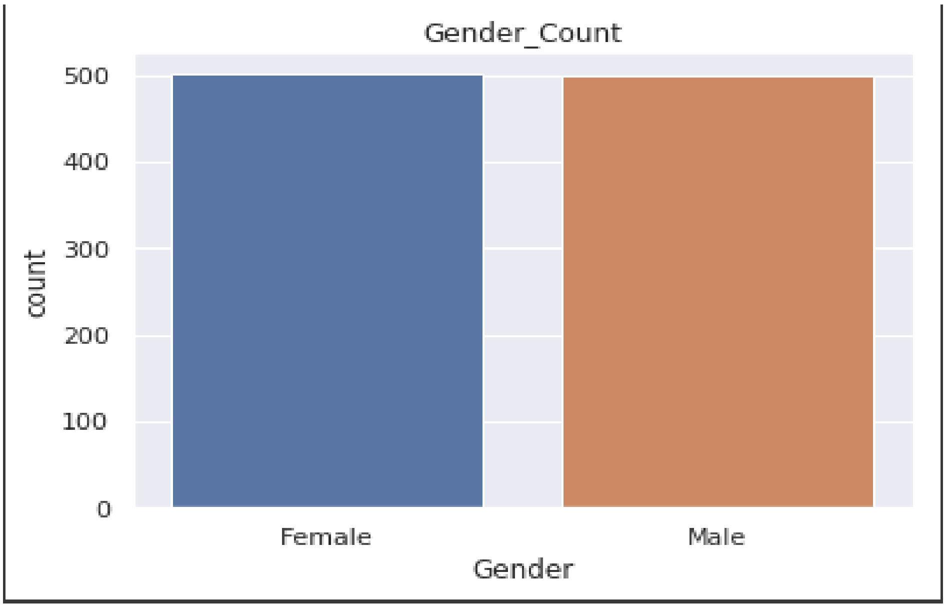
5. The customer having the membership visits the store frequently than normal customers.

6.Highest sales takes place at Naypyitaw where branch is C.

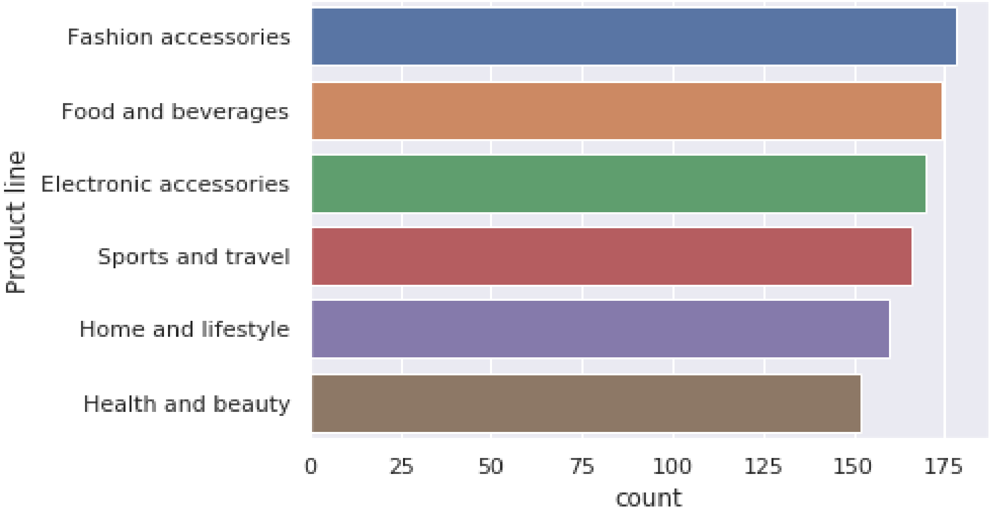
7.Highest sales takes placed at the branch C during the hour 10 for fashion accessories and least in branch A for fashion accessories .

VISUALISATION

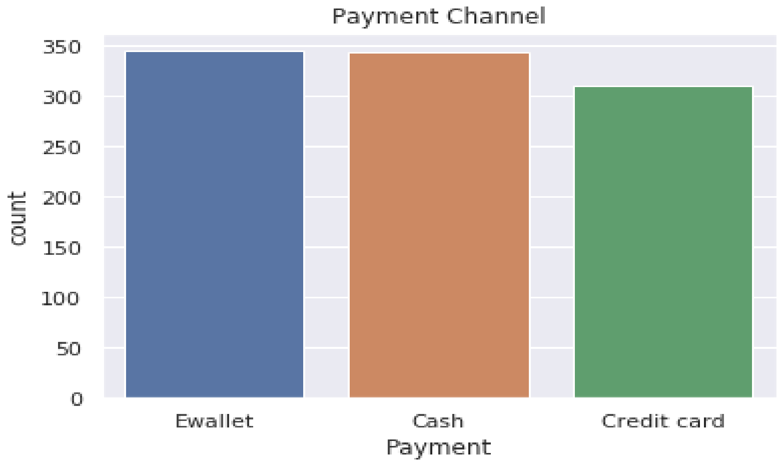
COUNT PLOT :



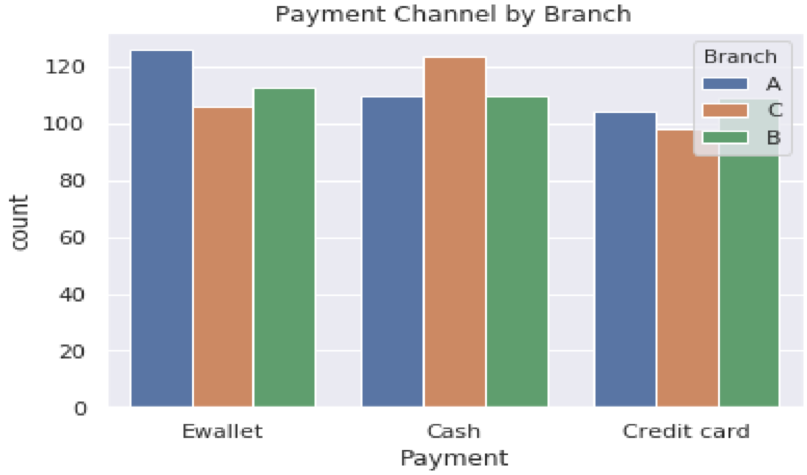
From the above plot we can understand that female customers is little higher than male customers.



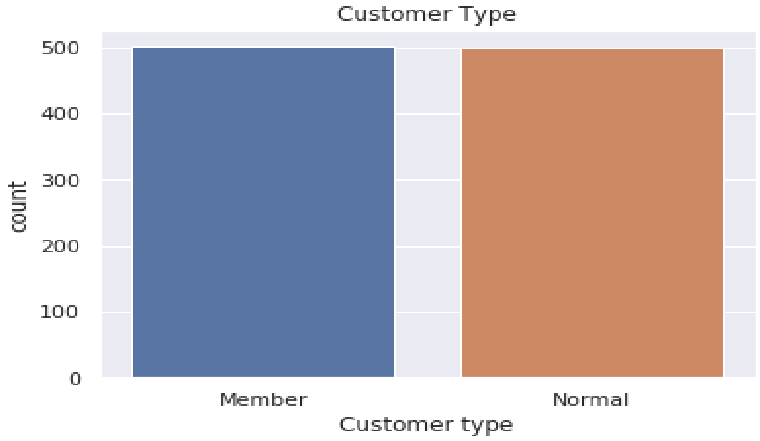
Fashion accessories has highest amount of sales than health and beauty.



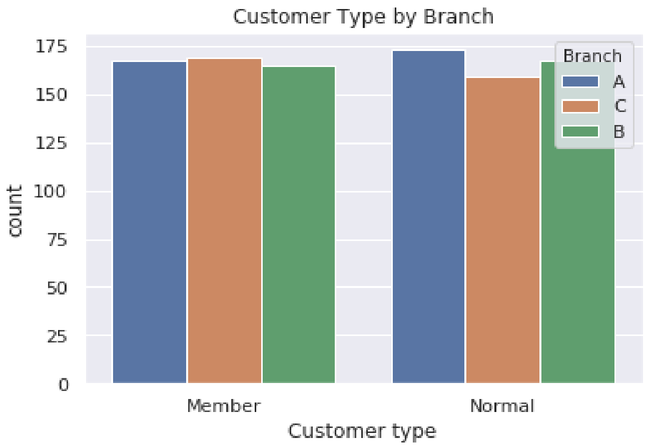
E Wallet is the most used payment method



Cash Payment is done mostly in branch C

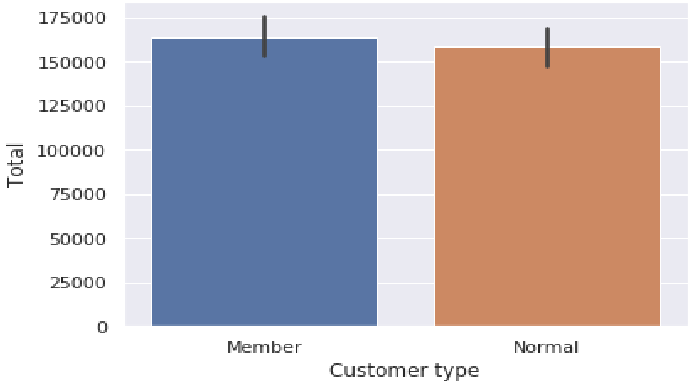


Member Customers is little more higher than normal customers



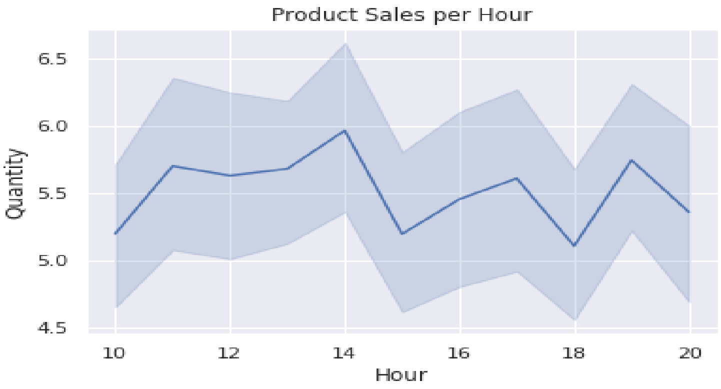
Branch A has more normal customers who visits the store occasionally

BAR PLOT :



Member customers does total quandity of sales than normal customers

Line Plot :



CONCLUSION

- Health and beauty has least sales if discount is given for both member and customer there is a chance to increase the sales by 50 % .
- If fashion and accessories rating got increased there is chance for the customers to get attracted easily.
- The maximum sales takes place during first three month of the year, if there is offer during the first month January the sales may get double.
- Providing advertisement based on the offer to customers who has membership through contact information will make them more attracted.
- Morning 10 AM to 8 PM is the time where customers chooses mostly for purchase
- Female customers purchases more than male customer if offer is given for food and beverages, health and beauty it makes female customers attract easily.
- Branch C has lowest sales if special offers mainly concentrated on that branch there is the chance for the sales to get double.
- Since sales peak during the first three months, introduce January sales events or “New Year Discounts”
- Encourage current members to refer friends by providing them with referral rewards, such as discounts on their next purchase. Promote these efforts on social media channels and encourage members to share promotional posts for extra loyalty points.
- Set up seasonal display sections to promote products like winter health items in Health and Beauty and new fashion arrivals, drawing attention from customers in-store.
- Host flash sales or “Tech Tuesdays” with special discounts on electronic accessories for a limited time makes gain more profit for the electronic accessories as they have more sales.
- Ensuring a variety of male-focused products, such as grooming kits, beard care products, skincare essentials, and hair styling items. Highlight products geared towards men's skincare, haircare, and body care to increase their appeal.
- After a customer makes a purchase, send them a thank-you message with a request for feedback, whether in-store or online. This reminder encourages them to leave a review which can increase rating.