ScalerMart Sales Decline: A Data-Powered Prescription for Recovery

G L Katyayini K

Date : 10th May 2024

Abstract

ScalerMart, a leading global electronics retailer, has grappled with a significant decline in sales, experiencing a nearly 50% revenue drop in 2020 compared to the previous year. To address this challenge and regain its market footing, this study delves into customer-level transactional data. By leveraging data analysis techniques, the study aims to identify the root causes behind the sales slump.

1. Problem Statement:

This report addresses the concerning decline in sales experienced by ScalerMart, highlighting a nearly 50% revenue drop in 2020 compared to 2019.

2. Objective

The objective of the study is to leverage customer-level transactional data to identify the root causes behind the substantial sales decrease. The aim is to uncover customer buying patterns, product trends, and potential market shifts contributing to the decline.

3. Importing Required Libraries

The report starts by importing essential libraries such as:

- Pandas
- Numpy
- matplotlib.pyplot
- seaborn

to facilitate data analysis and visualization.

4. Data Loading

Customer, product, and sales datasets were collected from relevant sources to analyse transactional data and customer behaviour.

Loading Customer, Product and Sales Data Sets

```
cust = pd.read_csv('Customers.csv', encoding='unicode_escape')
prod = pd.read_csv('Products.csv')
sales = pd.read_csv('Sales.csv')
```

5. Data Cleaning

Missing values for statecode in the customer data set is filled with 'NA' as there is no applicable state code and delivery date in sales data set is filled with forward and backwars fill from grouped data of OrderDate and DeliveryDate.

```
cust.isnull().sum()
```

```
CustomerKey     0
Gender          0
Name            0
City            0
State Code     10
State           0
Zip Code        0
Country         0
Continent       0
Birthday        0
dtype: int64
```

Filling null values with 'NA' as they are state codes.

```
cust.fillna('NA',inplace=True)
```

```
cust.isnull().sum()
```

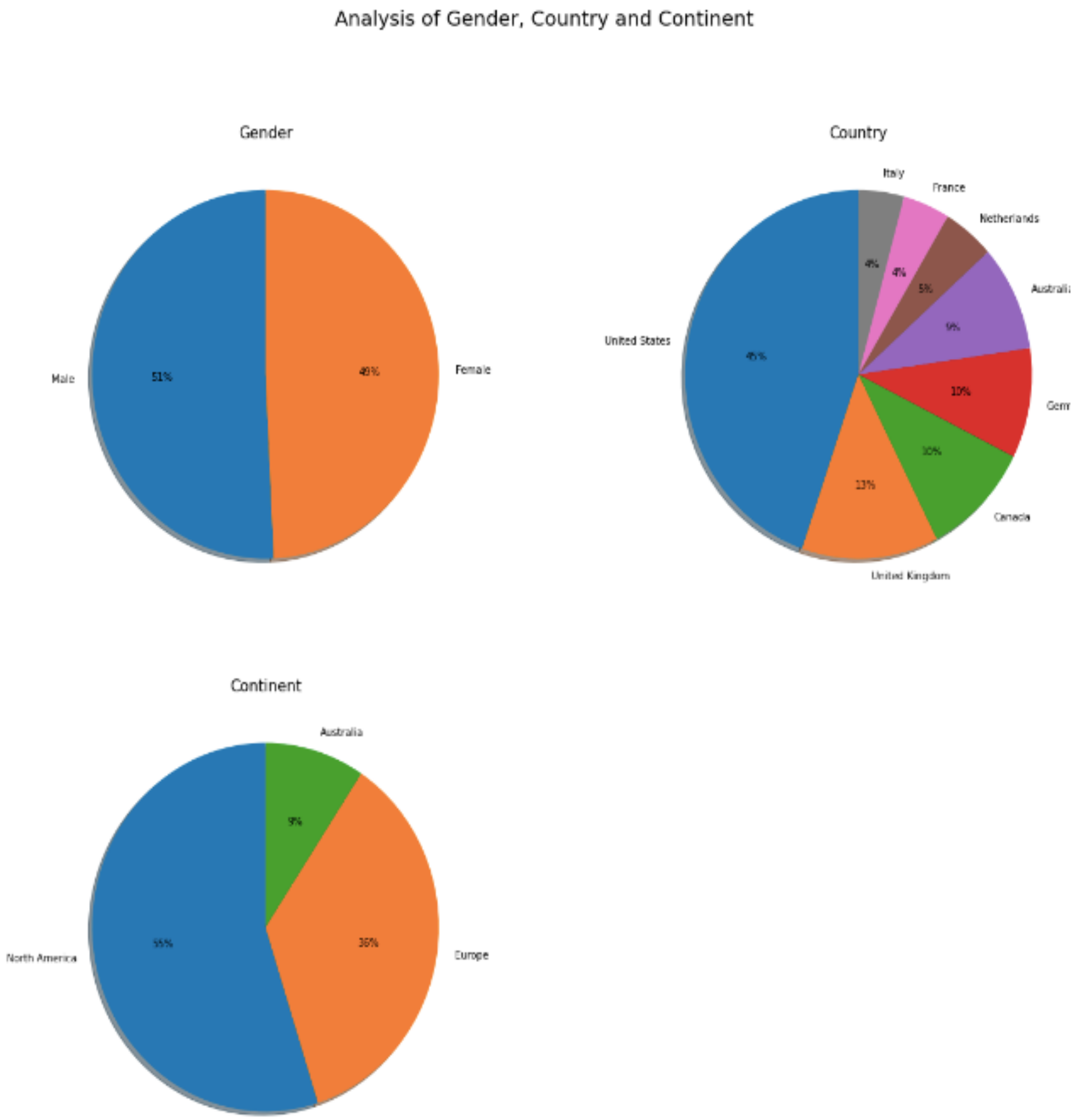
```
CustomerKey    0
Gender         0
Name           0
City           0
State Code     0
State          0
Zip Code       0
Country        0
Continent      0
Birthday       0
year           0
dtype: int64
```

6. Data Analysis

6.1 Customer Data Analysis(EDA):

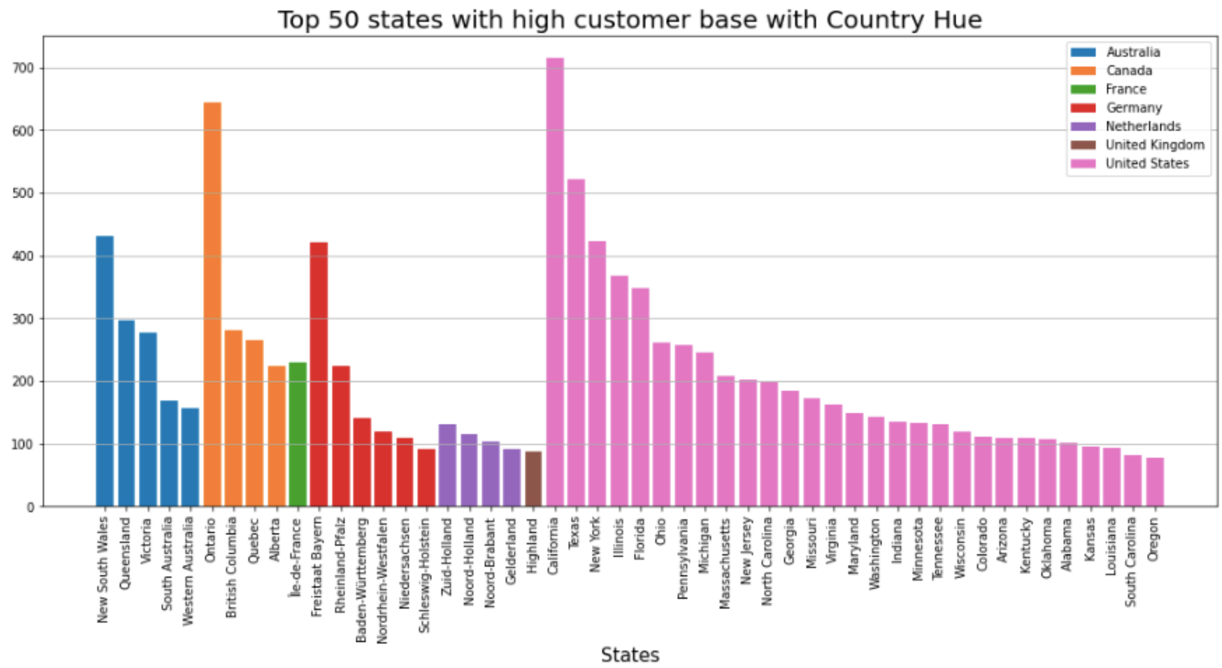
Analysis of customer demographics, including gender distribution and geographic location.

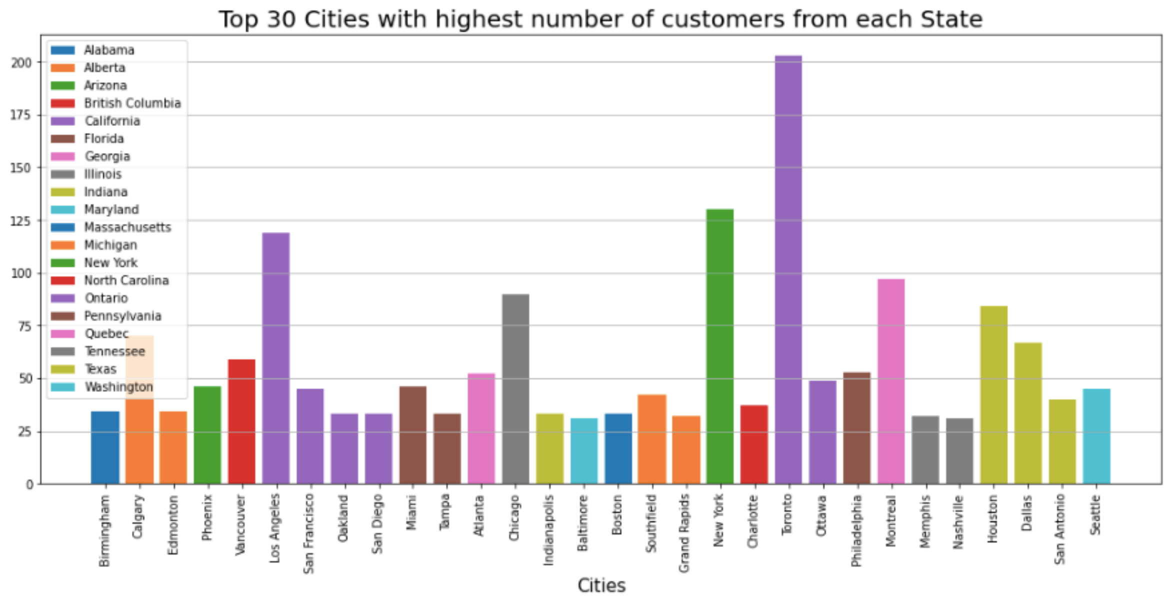
- 51% of the customers are Male and 49% are Female.
- 45% of the customers are from United states, 13% from UK, 10% Canada and 10% Germany.
- 55% of customrs are from NorthAmerica, 36% from Europe and 9% from Australia



Identification of top states with the highest number of customers from different countries.

- US,Canada,Germany and Australia are the countryies having states with highest customer base.
- California is the top state with more than 700 customers, next comes Ontario with 650 customers and next comes texas with more than 500 customers.
- Follows other states Newyork, Frelsaat Bayern,New South Wales,Illions, Florida,Qeensland, British Columbia.
- Italy and Uk are having most of the least customer states where the organization can concentrate to increase their sales.





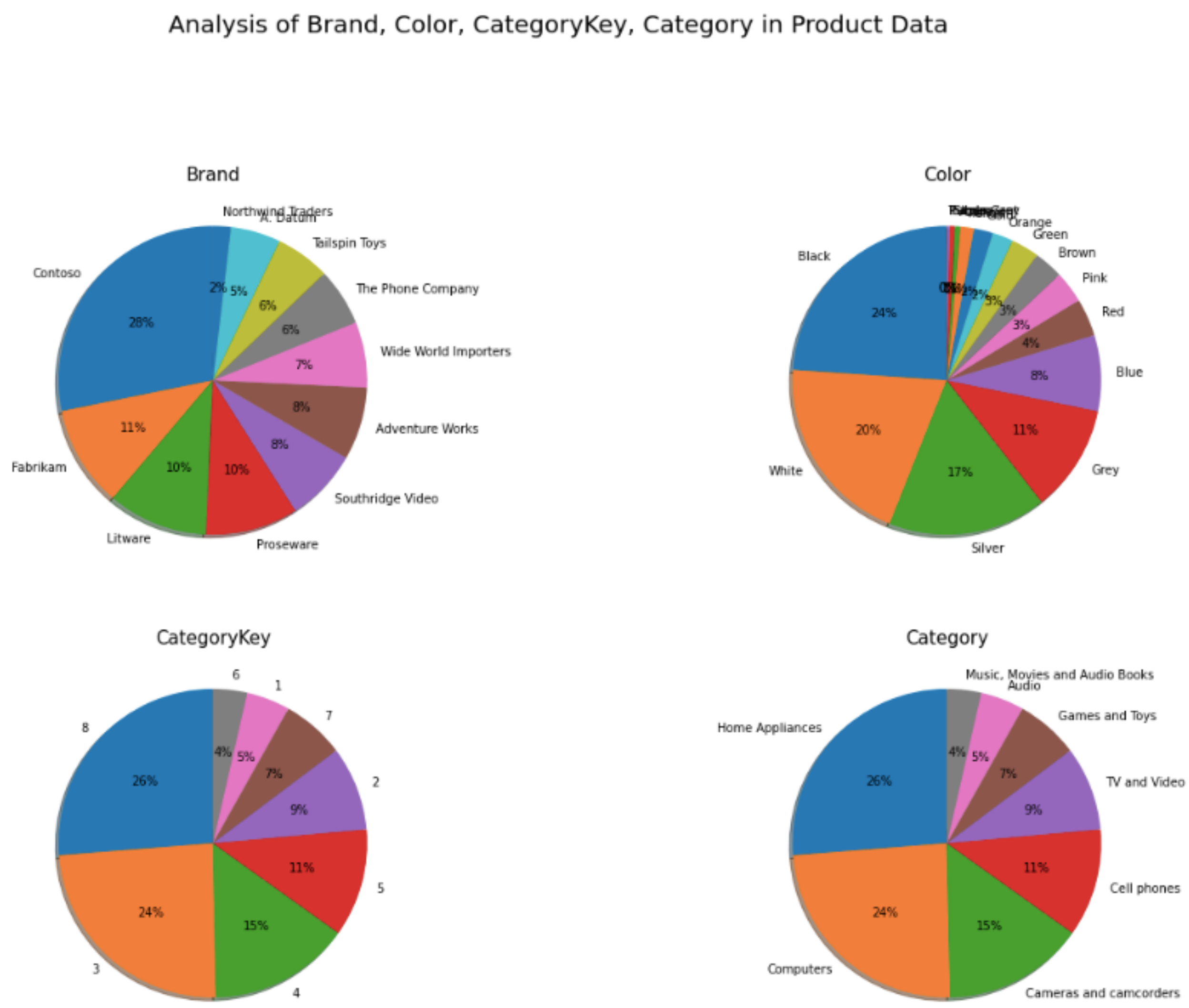
- Toranto from state Ontrio and Losangeles from state California are the cities with highest customer base.

Segmentation of customers based on various criteria for targeted marketing strategies.

6.2 Product Data Analysis

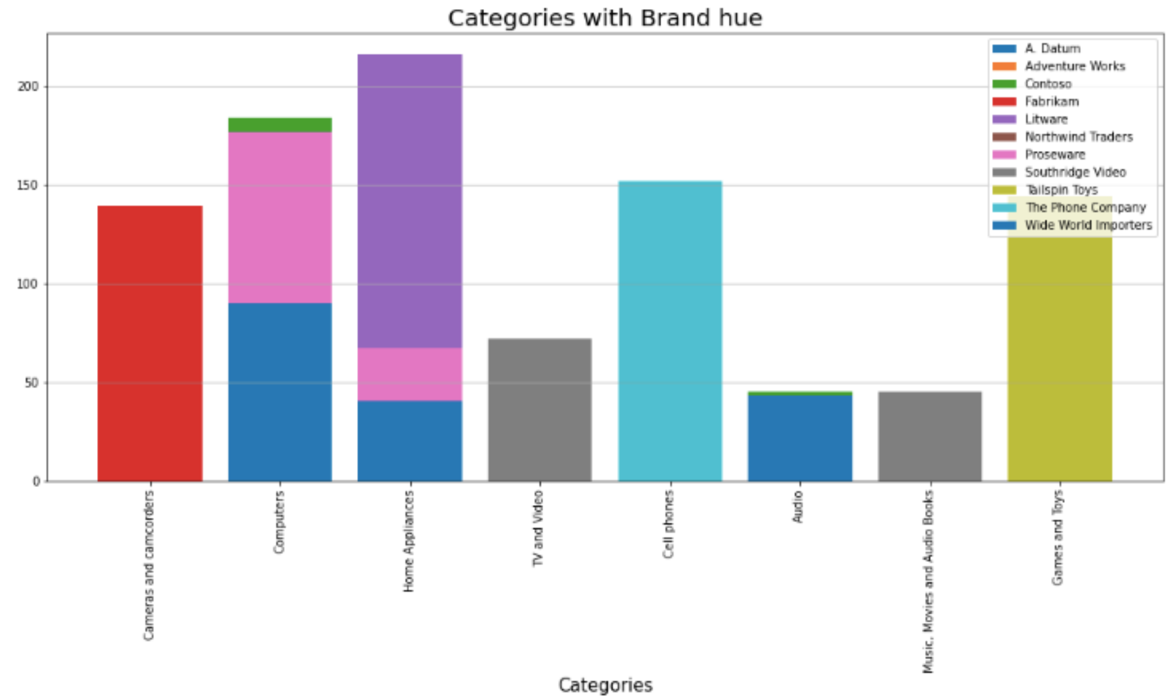
Examination of product attributes such as brand, color, unit cost, unit price, and category.

- Contoso, Fabrikam, Litware and Poseware are the leading product brands in the product data with 28, 11 10 and 10 Percent of overall products respectively.
- Black, White,Silver, Grey are the more produced colors.
- Home Appliances, Computers, Cameras & Camcorders and Cell phones are the highest percent occupiers with 26, 24, 15, 11 percent of overall product Category respectively.



Bivariate Analysis of Product Data

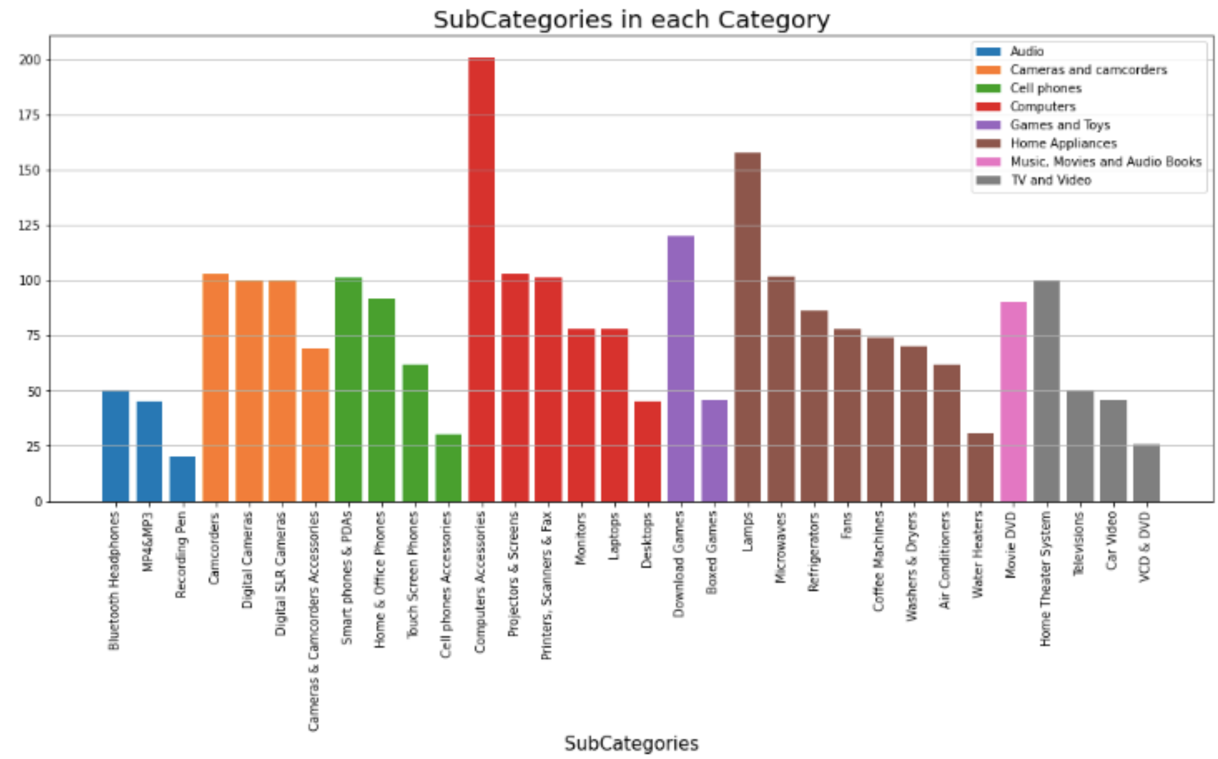
```
## grouping data by Brand and Category
group_subcat_data = prod.groupby(['Brand','Category']).size().sort_values(ascending=False).reset_index(name='count')

plt.figure(figsize=(17,7))

for brand,group in group_subcat_data.groupby('Brand'):
    plt.bar(group['Category'],group['count'],label=brand)
plt.grid(True, axis='y')
plt.xlabel('Categories', fontsize=15)
plt.title('Categories with Brand hue', fontsize='20')
plt.xticks(rotation=90)
plt.legend()
plt.show()
```



From the above plot:

- Even if Contoso is the brand selling highest number of products it has limited categories like computers and audio.
- Litware is the brand exclusive for Home Appliances.
- The Phone Company is exclusive Cellphone brand.
- Fabricam is the brand selling Cameras and camcorders.
- WorldWideImporters ia the brand with categories like Computers, Audio and HomeAppliances.
- Tailspin Toys is an exclusive Games and Toy brand.
- Porsware is a brand selling Computers and HomeAppliances.
- NorthWindTraders is an exclusive HomAppliances seller.

```python
## grouping data by Category and subcategory , counting each type in each group
group_subcat_data = prod.groupby(['Category','Subcategory']).size().sort_values(ascending=False).reset_index(name='c

plt.figure(figsize=(17,7))

for cat,group in group_subcat_data.groupby('Category'):
    plt.bar(group['Subcategory'],group['count'],label=cat)
plt.grid(True, axis='y')
plt.xlabel('SubCategories', fontsize=15)
plt.title('SubCategories in each Category', fontsize='20')
plt.xticks(rotation=90)
plt.legend()
plt.show()
```



From the above plot:

- HomeAppliances and computers are the highest sold categories with includes brands like WorldWideImporters, Contoso and Litware.
- Computers&Accessories, Lamps and DownloadedGames are the subcategories sold more from the categories Computers, HomeAppliances and Games & Toys respectively.

Evaluation of product performance in terms of sales, profitability, and demand.

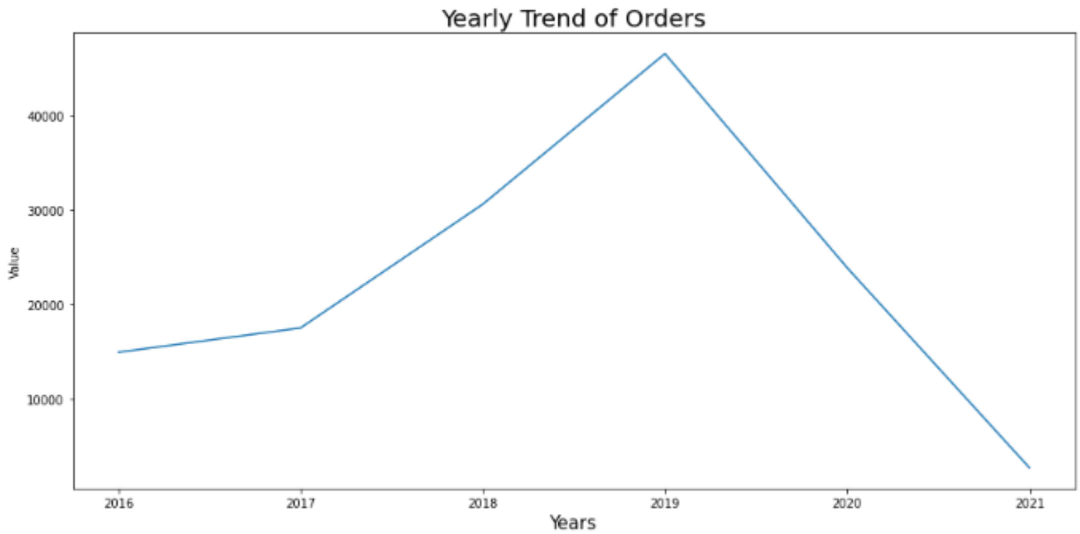
Identification of top-selling products and categories contributing to revenue.

6.3 Sales Data Analysis:

Analysis of sales transactions, including order dates, delivery dates.

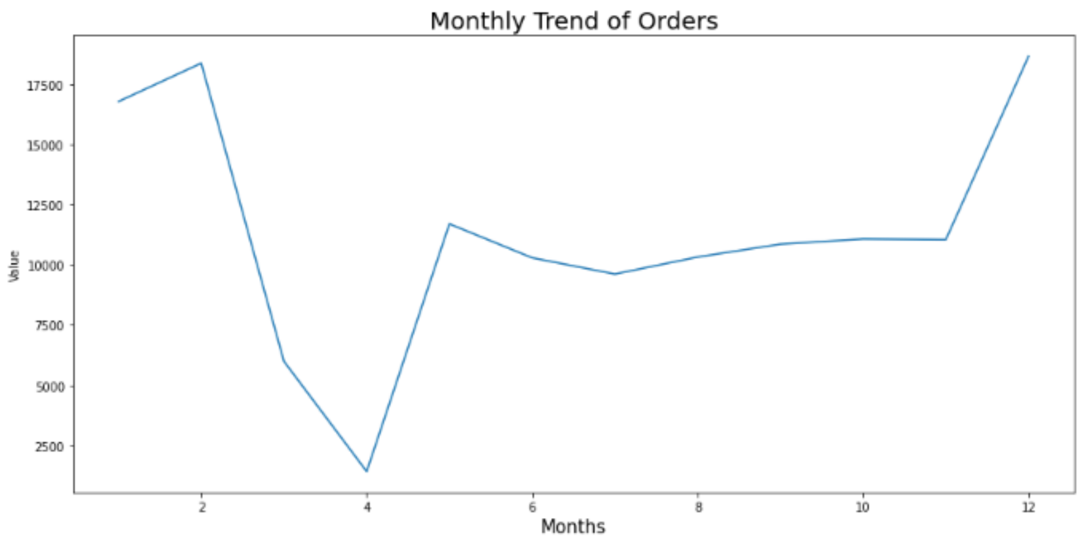
- Highest sales happened in the year 2019 and 2020, there is a drastic decrease after 2020.

```
#Yearly Trend of Orders
grp_data_yr= merged_data.groupby('OrderYear')['Line Item'].sum().reset_index(name='count')
plt.figure(figsize=(15,7))
sns.lineplot(x=grp_data_yr['OrderYear'], y=grp_data_yr['count'])
plt.xlabel("Years", fontsize=15)
plt.ylabel("Value")
plt.title("Yearly Trend of Orders", fontsize='20')
plt.show()
```



- There is trend of decrease in orders in the 4th month(April) hike in 2nd and 12th months(Feb and Dec).

```
#Monthly Trend of Orders
grp_data_mnth= merged_data.groupby('OrderMonth')['Line Item'].sum().reset_index(name='count')
plt.figure(figsize=(15,7))
sns.lineplot(x=grp_data_mnth['OrderMonth'], y=grp_data_mnth['count'])
plt.xlabel("Months", fontsize=15)
plt.ylabel("Value")
plt.title("Monthly Trend of Orders", fontsize='20')
plt.show()
```



6.4 Feature Engineering

Features like OrderYear, OrderMonth, TotalItems and DaysForDelivery are extracted from the Sales Dataset.

Extracting Features like OrderYear, OrderMonth , TotalItems, DaysForDelivery

```
sales['OrderYear']= sales['Order Date'].dt.year
```

```
sales['OrderMonth']= sales['Order Date'].dt.month
```

```
sales['Total Items']= sales['Line Item']* sales['Quantity']
```

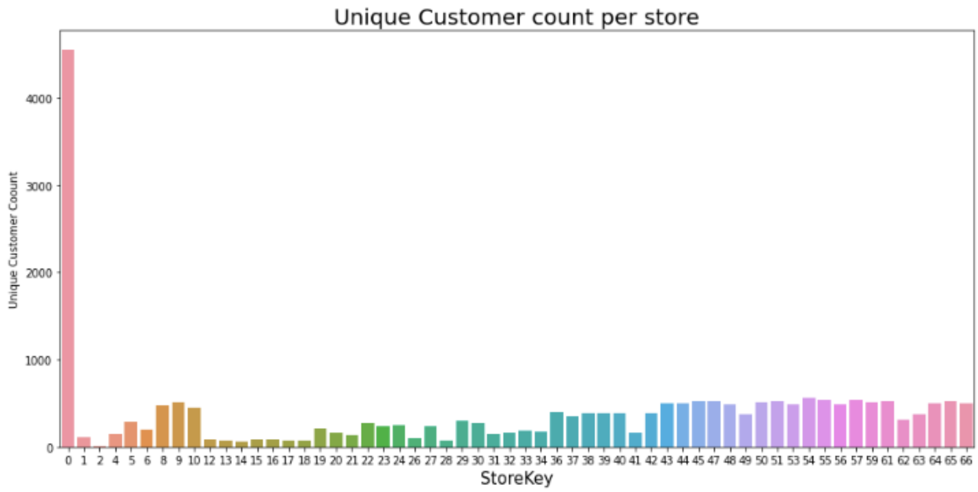
```
sales['DaysForDelivery']= (sales['Delivery Date']- sales['Order Date']).dt.days
```

6.5 Merged data Analysis

Customer, Sales and Product data sets are merged and analyzed.
Below are the insights from the analysis after merging sales and customer:

- Male orders are slightly more than female orders.
- UnitedStates is in the top of list of total order per country.. Next is UK, Germany and Canada.
- California is the state with highest number of orders and next to it is Texas and Ontario
- Store 0 is having highest number of customers.
- The number of days reuired for delivery from 2016 gradually decreased saying that the access for delivery became easy.
- Till 2019 the average days of delivery per year decreased but increased in the year 2020 due to the pandemic stituations which are again set back in the year 2021.

```
#Store vs customers
grp_store_cust= merged_data.groupby('StoreKey')['CustomerKey'].nunique().reset_index(name='count')
plt.figure(figsize=(15,7))
sns.barplot(x=grp_store_cust['StoreKey'], y=grp_store_cust['count'])
plt.xlabel("StoreKey", fontsize=15)
plt.ylabel("Unique Customer Coount")
plt.title("Unique Customer count per store", fontsize='20')
plt.show()
```




```
Insights:

    Store 0 is having highest number of customers.
```

6.6 Profit Analysis:

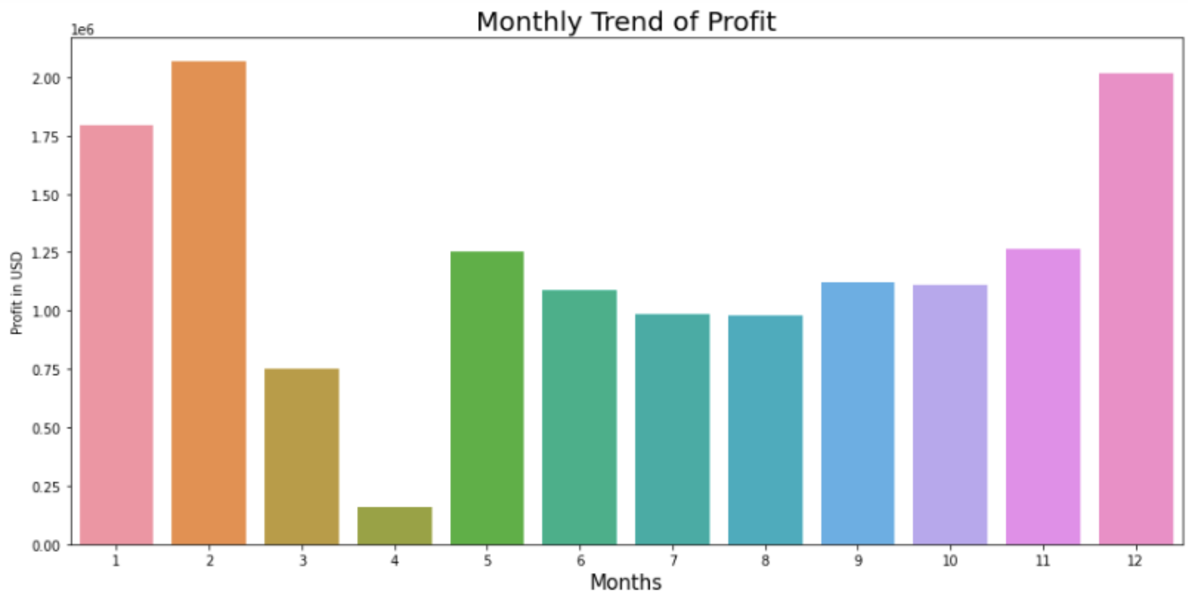
Exploration of sales trends over time to understand fluctuations in revenue.

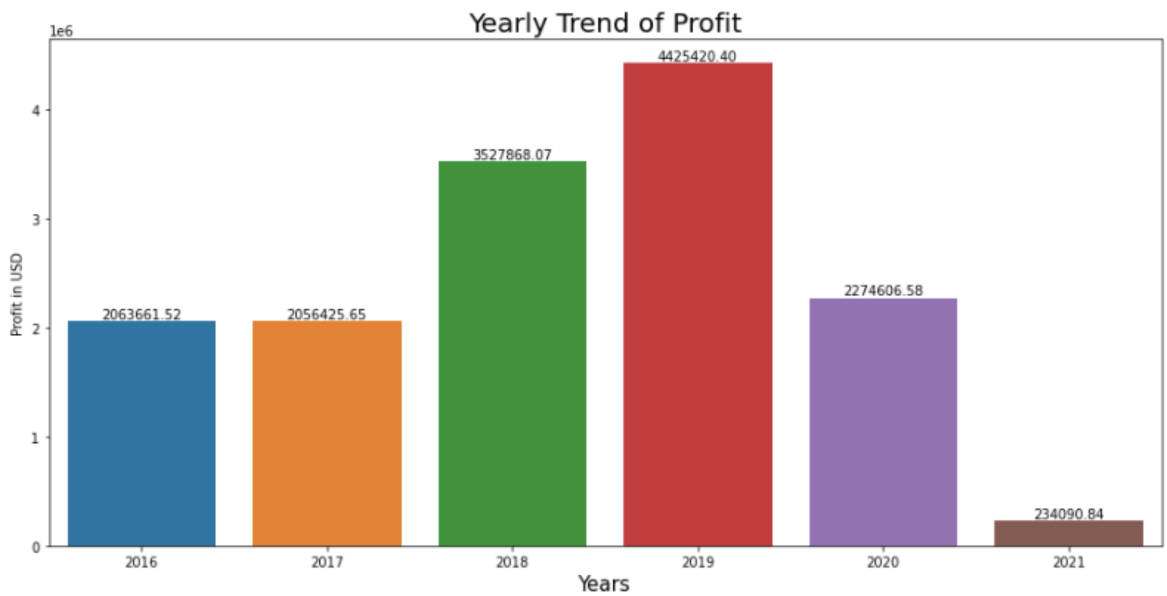
Calculating totalPurchase column from TotalItems and UnitPriceUSD . TotalProfit from profit and TotalItems columns.

```
sale_prod_data['TotalPurchase']=sale_prod_data['Total Items']*sale_prod_data['Unit Price USD']
```

```
sale_prod_data['TotalProfit']= sale_prod_data['profit']*sale_prod_data['Total Items']
```

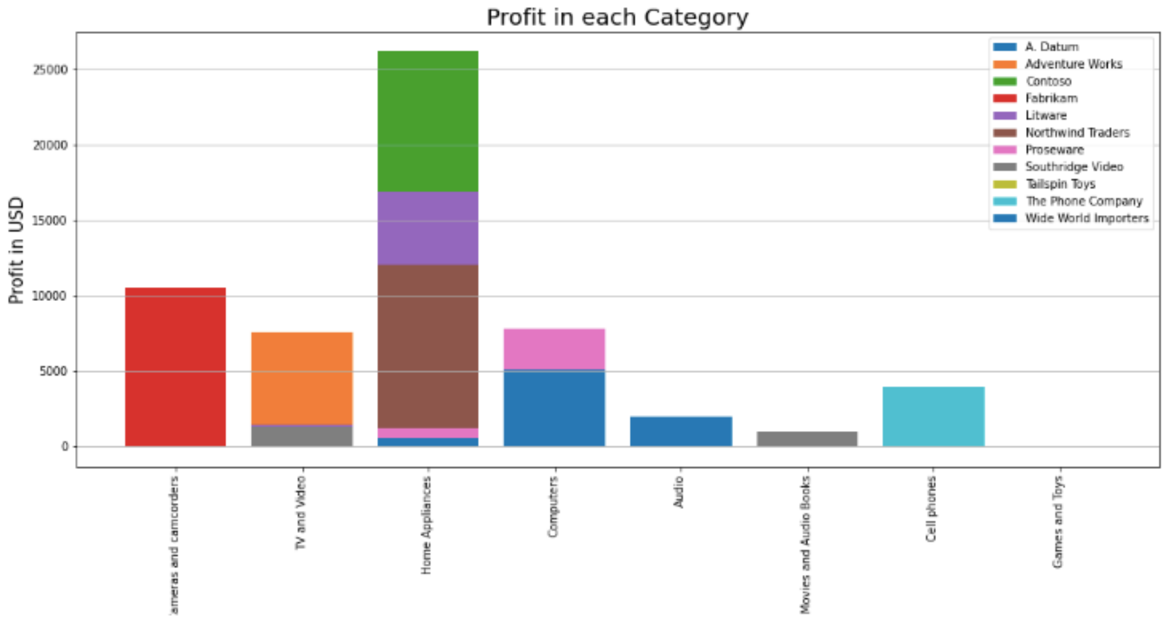
- Like the sales the trend of profits is also high in the year 2019 and almost halved in 2020 and nearly 5% of the profit of 2019.
- The trend of profits follows the orders. The Profits are least in the month of April and highest in the month of Dec and Feb.
- The category wise sales for the Category 'Computers' decreased from 35000 items to 2100 from year 2019 to 2021. The sales almost decreases by 16 times.



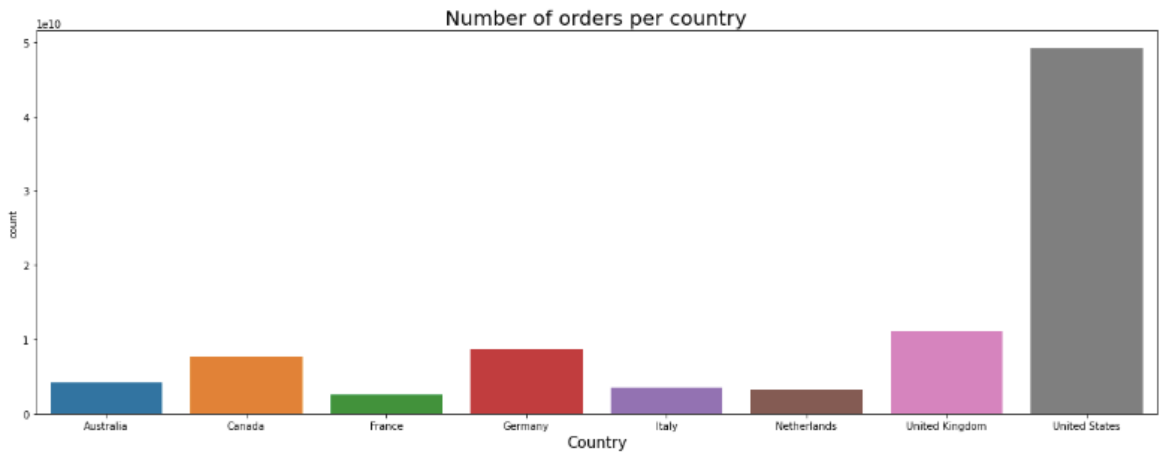


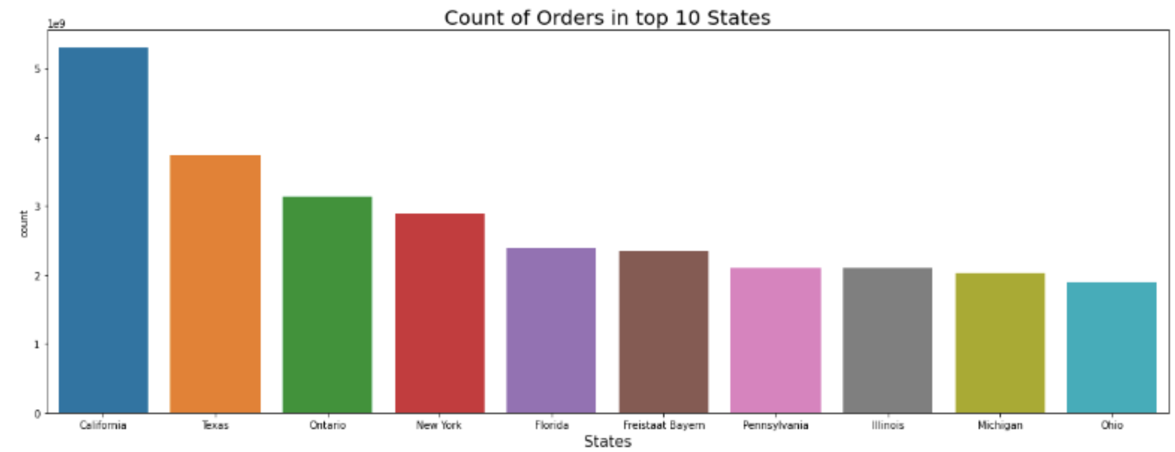
Comparison of sales performance across different product categories and regions.

```
## grouping data by Category and subcategory , counting each type in each group
group_profit_data = prod.groupby(['Brand','Category'])['profit'].sum().sort_values(ascending=False).reset_index(name

plt.figure(figsize=(17,7))

for brand,group in group_profit_data.groupby('Brand'):
    plt.bar(group['Category'],group['count'],label=brand)
plt.grid(True, axis='y')
plt.xlabel('Categories', fontsize=15)
plt.ylabel('Profit in USD', fontsize=15)
plt.title('Profit in each Category', fontsize='20')
plt.xticks(rotation=90)
plt.legend()
plt.show()
```



- Home Appliances is the category with highest profits with 2.5k USD.
- Next to it is Camreras and Camrecorders fom the brand Fabrikam.
- Least profit category is Muisc,movies and Audiobooks from SouthRidge Video which inturn is the least profit brand.



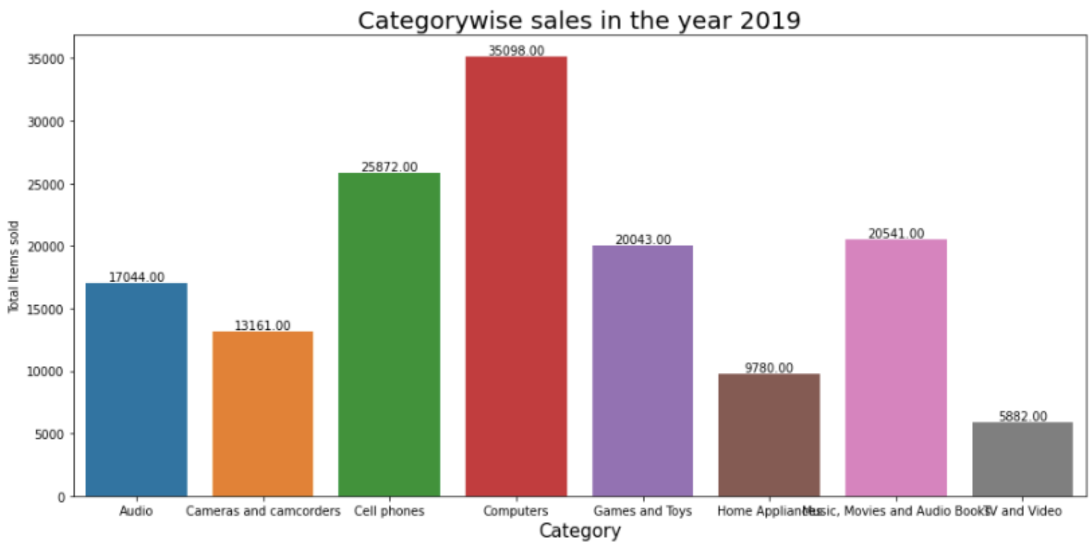


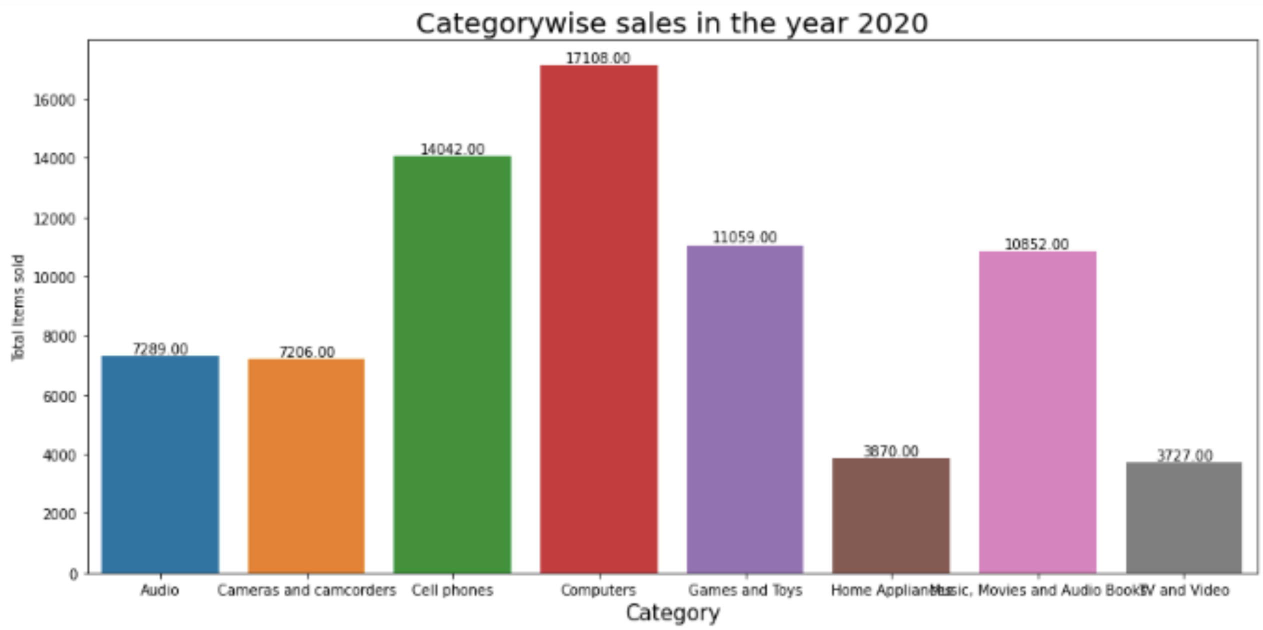
- UnitedStates is in the top of list of total order per country.. Next is UK, Germany and Canada.
- California is the state with highest number of orders and next to it is Texas and Ontario.

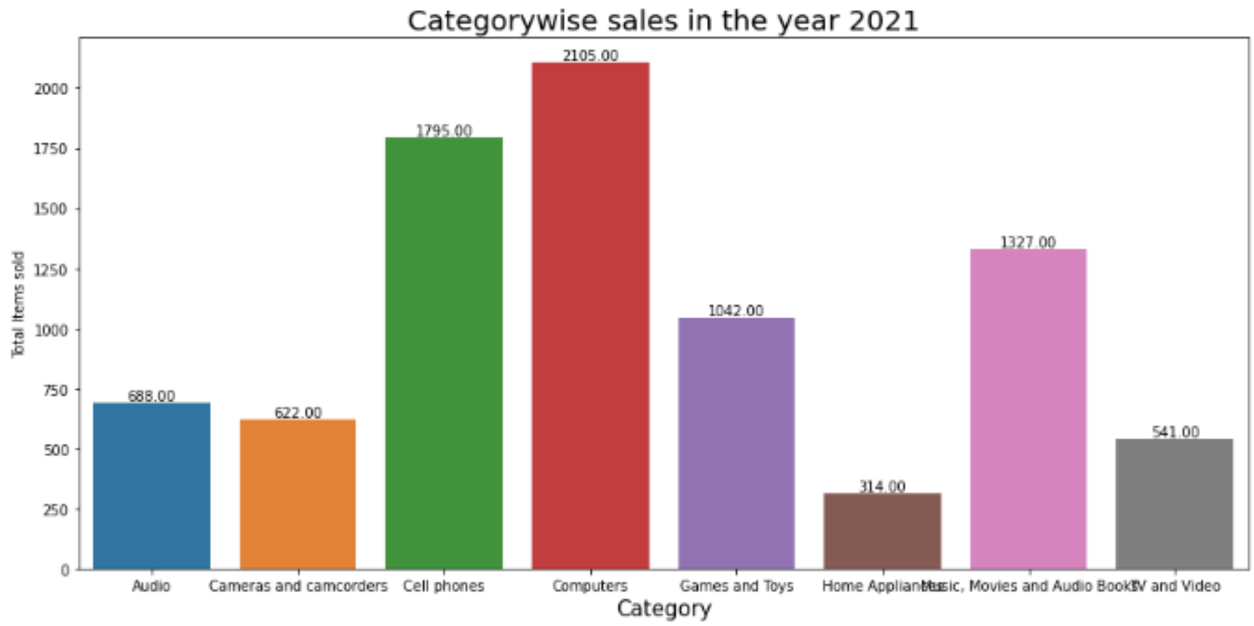
6.7 Trend Analysis:

Examination of trends in customer purchasing behaviour over time in all categories.

```
#Categorywise sales for years 2019, 2020, 2021
for i in [2019,2020,2021]:
    data = sale_prod_data[sale_prod_data['OrderYear']==i]
    plt.figure(figsize=(15,7))
    ax=sns.barplot(x=data.groupby('Category')['Total Items'].sum().index, y=data.groupby('Category')['Total Items'].
    container = ax.containers[0]
    ax.bar_label(container, fmt='{:.2f}')
    plt.xlabel("Category", fontsize=15)
    plt.ylabel("Total Items sold")
    plt.title(f"Categorywise sales in the year {i}", fontsize='20')
    plt.show()
```





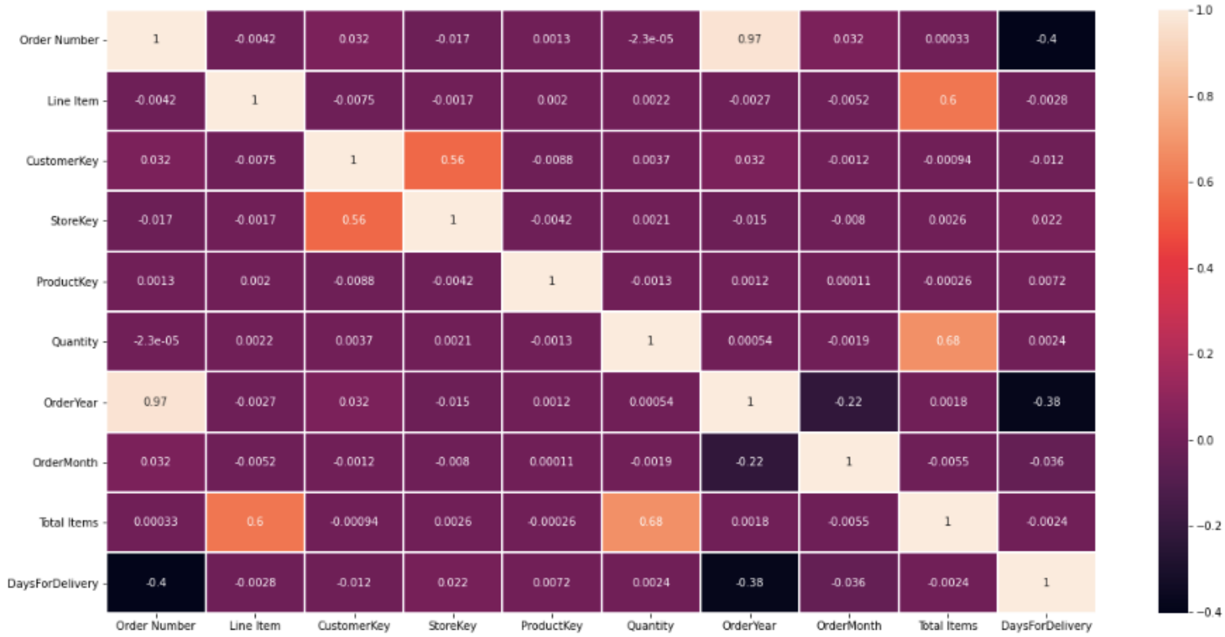


- The category wise sales for the Category 'Computers' decreased from 35000 items to 2100 from year 2019 to 2021. The sales almost decreases by 16 times.

6.8 Correlation Analysis

Exploration of relationships between different variables such as customer demographics, product attributes, and sales performance.

```
#Correlation matrix for merged data of sales and customers
plt.figure(figsize=(20,10))
sns.heatmap(merged_data.corr(),annot=True,linewidths=0.2, linecolor='white')
plt.show()
```

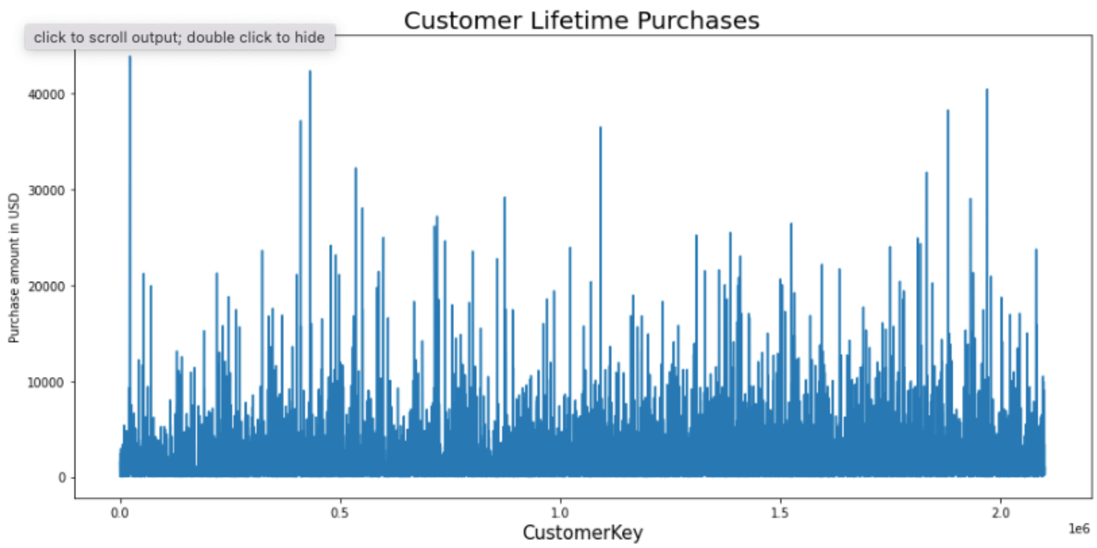


- Correlation matrix of sales and customer data gives strong relation for customerkey and storekey, saying Customers are store centric.

6.9 Customer Segmentation and Loyality Analysis

Customers were segmented based on various criteria such as buying behaviour.

```
grp_data_cust= sale_prod_data.groupby('CustomerKey')['TotalPurchase'].sum().reset_index(name='count')
plt.figure(figsize=(15,7))
sns.lineplot(x=grp_data_cust['CustomerKey'], y=grp_data_cust['count'])
plt.xlabel("CustomerKey", fontsize=15)
plt.ylabel("Purchase amount in USD")
plt.title("Customer Lifetime Purchases", fontsize='20')
plt.show()
```



- This is the graph for Customer life time purchases based on which we can segment the customers into loyal and random customers.

```
#let us take customers whose life time purchase is >10000 USD
cust_lf_pur = sale_prod_data.groupby('CustomerKey')['TotalPurchase'].sum().sort_values(ascending=False).reset_index(
```

```
#loyal or high valued customers whose lifetimepurchase>10000 USD
loyal_customers=cust_lf_pur[cust_lf_pur.amount>10000].CustomerKey
len(loyal_customers)
```

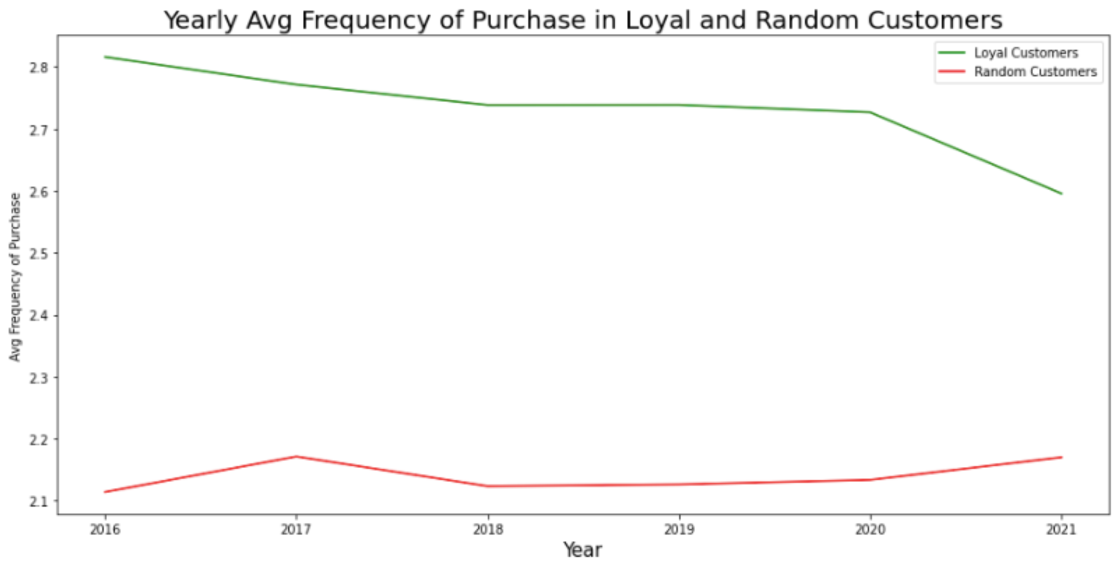
```
263
```

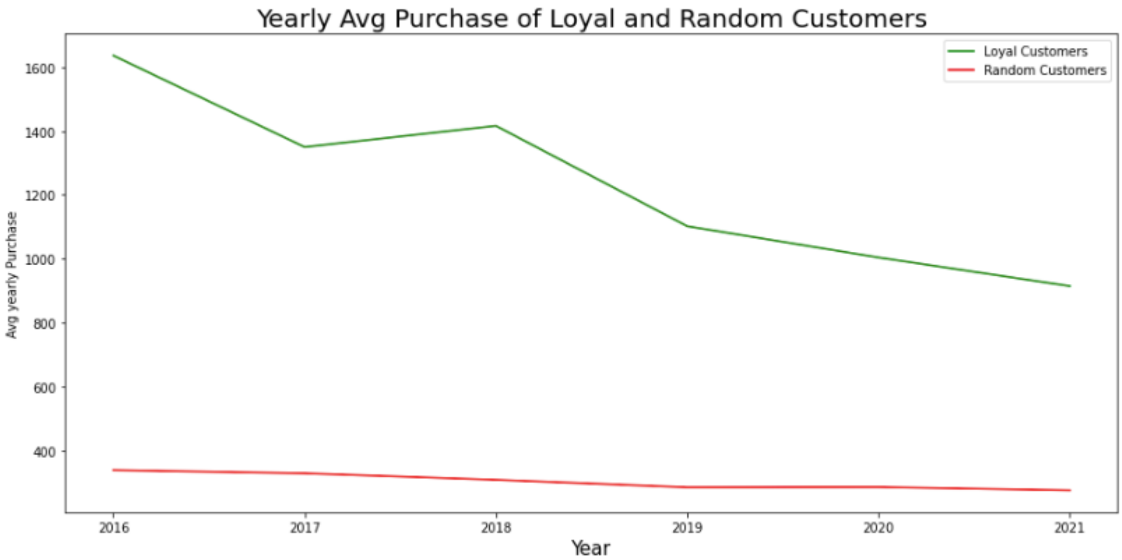
```
#loyal customers whose lifetimepurchase>10000 USD
random_customers=cust_lf_pur[cust_lf_pur.amount<10000].CustomerKey
len(random_customers)
```

```
11624
```

```
#Extracting Loyal customer data bbased on customer keys from loyal customers
loyal_customer_data = sale_prod_data[sale_prod_data['CustomerKey'].isin(loyal_customers)]
```

```
#Extracting Random customer data bbased on customer keys from random customers
random_customer_data = sale_prod_data[sale_prod_data['CustomerKey'].isin(random_customers)]
```





- The yearly average purchases in loyal customers decrease from yaer 2020 saying that it will be the reason for decline in the revenue genereation after 2020.
- From the above plot of customer loyality defined in the terms of amount of purchase over the year, it is observed that the trend of purchases got declined after 2020 from 1600 to 1000 which will be an amount of $600*263(loyalcustomers)= $157800 decrease in the revenue.

```
#taking highest avg for loyal customers
high_avg = loyal_customer_data.groupby('OrderYear')['TotalPurchase'].mean().sort_values(ascending=False).reset_index
high_avg
```

```
1636.748404494382
```

```
#taking lowest avg for loyal customers
low_avg = loyal_customer_data.groupby('OrderYear')['TotalPurchase'].mean().sort_values(ascending=True).reset_index(n
low_avg
```

```
914.5095505617978
```

```
#Contribution of loyal cutomers for purchases per year
loyal_pur_contrbution = (high_avg-low_avg)*len(loyal_customers)
loyal_pur_contrbution
```

```
189948.8185842696
```

```
#Percentage of loyal cutomers for purchases per year
prcnt_role_loyl_cust = (loyal_pur_contrbution/profit_decline_2020)*100
prcnt_role_loyl_cust
```

```
9.308862002761538
```

- There is approximately 9.3% decrease in profits due to loyal customers. That means there should be some measures taken for loyal customer engagement so that the number of avg purchases increases, increasing the revenue from the loyal customer end.

```
#taking highest avg for random customers
high_avg_ran = random_customer_data.groupby('OrderYear')['TotalPurchase'].mean().sort_values(ascending=False).reset_
high_avg_ran
```

```
338.1163405572714
```

```
#taking lowest avg for random customers
low_avg_ran = random_customer_data.groupby('OrderYear')['TotalPurchase'].mean().sort_values(ascending=True).reset_in
low_avg_ran
```

```
274.6891718610862
```

```
#Contribution of random cutsomers for purchases per year
ran_pur_contrbution = (high_avg_ran-low_avg_ran)*len(random_customers)
ran_pur_contrbution
```

```
737277.408924457
```

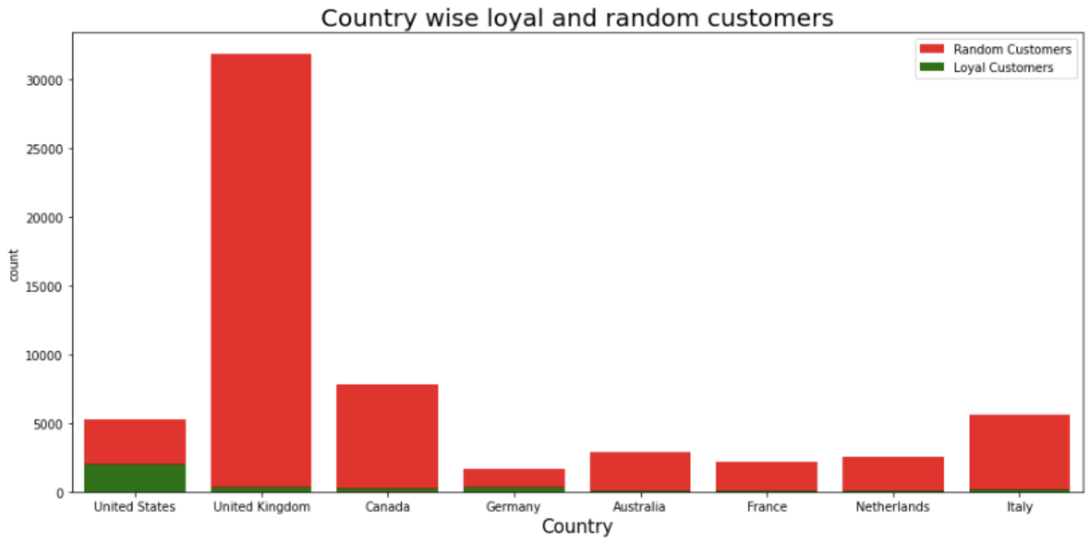
```
#Percentage of random cutsomers for purchases per year
prcnt_role_ran_cust = (ran_pur_contrbution/profit_decline_2020)*100
prcnt_role_ran_cust
```

```
36.13191020920478
```

- There is approximately 36% decrease in profits due to random customers. That means there should be some measures taken for random customer retention so that the number of avg purchases increases, increasing the revenue from the random customer end.

6.10 Geographic Sales Analysis

Sales patterns across different regions and countries were analyzed to identify geographical trends, customer preferences, and opportunities for market expansion.

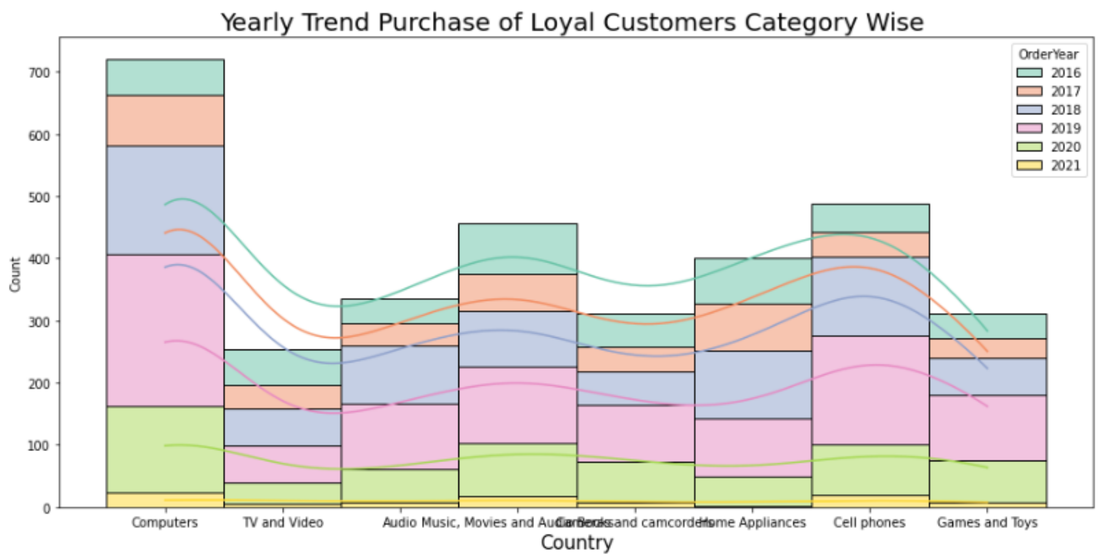


- US has a near equal number of loyal and random customers. and all the other countries are dominated with random customers.

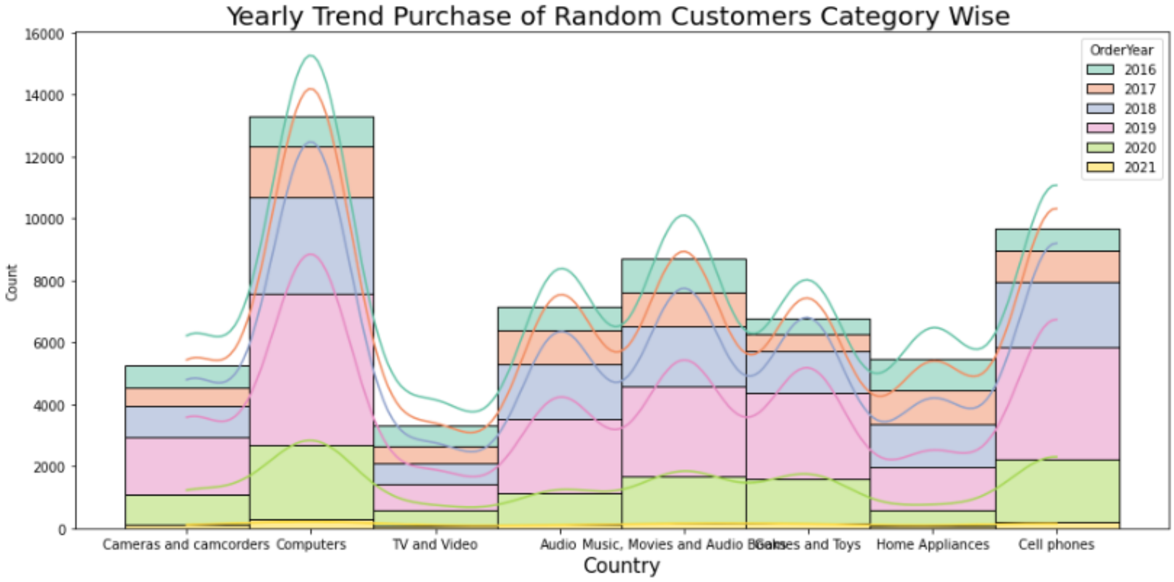
6.11 Category wise Sales Trends Analysis:

Sales trends across different product categories were examined to understand which categories were driving sales and which ones were underperforming, guiding strategic decisions.

- The yearly trend of loyal and random customers based on category is shown in the below plots.



- From below plot it is observed that the purchases in all the categories decreased in case of both loyal and random customers.



7. Data-Driven Recommendations

Customer-Centric Strategies:

- **Segmentation and Personalization:** Analyze customer data to segment customers based on demographics, purchase history, and preferences. Develop targeted marketing campaigns, promotions, and product recommendations for each segment.
- **Loyalty Programs:** Implement a loyalty program to reward repeat customers and incentivize them to continue shopping at ScalerMart. Offer points, discounts, or exclusive benefits to loyal customers.

Product Portfolio Optimization:

- **Product Analysis:** Analyze sales data to identify top-selling products and categories. Focus on stocking and promoting these profitable products.
- **Customer Preferences:** Use customer data to understand product preferences and emerging trends. Introduce new products or variants that align with customer needs and market demands.

Price Optimization and Value Proposition:

- **Competitive Analysis:** Research competitor pricing strategies and market trends. Set competitive prices while maintaining profitability.
- **Value Communication:** Clearly communicate the unique value proposition of ScalerMart's products. Highlight features, benefits, and any advantages over competitors.

By focusing on these simplified recommendations and leveraging data-driven insights, ScalerMart can improve sales, regain market share, and achieve sustainable growth.

8. GitHub Link

Detailed report and code are available in the following github link:

https://github.com/katyayini0583/ScalerMatch