FEZZARI SALES

by KAVIYARASAN.S

1.Introduction

In today's competitive business environment, leveraging data to inform sales strategies is essential for achieving success. Sales datasets provide invaluable insights into customer behaviors, sales trends, and market dynamics. These datasets can be used to identify opportunities, optimize sales processes, and enhance decision-making.

Importance of Sales Datasets

Sales datasets encompass various types of information, including transaction records, customer demographics, product performance, and sales channels. By analyzing these datasets, businesses can:

1.Understand Customer Preferences*: Identify the products or services that are most popular among different customer segments.

2.Track Sales Performance*: Monitor sales figures over time to evaluate the effectiveness of sales strategies and marketing campaigns.

3.Optimize Inventory Management*: Ensure the right products are available at the right time to meet customer demand.

4.Forecast Sales Trends*: Predict future sales patterns to inform budget planning and resource allocation.

5.Enhance Customer Experience*: Tailor sales approaches to meet the needs and expectations of various customer groups.

Structure of Sales Datasets

Sales datasets typically include several key components:

1. Transaction Data*: Details of individual sales transactions, including date, time, product or service sold, quantity, price, and payment method.

2. Customer Data*: Information about customers, such as demographics, purchasing history, and contact details.

3. Product Data*: Descriptions of products or services, including categories, pricing, and inventory levels.

4. Sales Channel Data*: Information about the channels through which sales are made, such as online platforms, physical stores, or third-party vendors.

5. Time Series Data*: Sales figures over different periods, allowing for trend analysis and seasonal adjustments.

Applications of Sales Data Analysis

Businesses can apply sales data analysis in various ways to drive growth and improve efficiency:

1.Sales Performance Analysis: Evaluate the effectiveness of sales teams and individual sales representatives.

2.Market Segmentation: Identify and target specific market segments with tailored marketing and sales strategies.

3.Product Development: Use customer feedback and sales trends to guide product innovation and development.

4.Pricing Strategies: Analyze the impact of pricing changes on sales volumes and profitability.

5.Customer Retention: Develop strategies to retain high-value customers based on purchasing behavior and satisfaction levels.

2.Project Description

Project Overview:

The objective of this project is to analyze and visualize sales data to derive actionable insights that can inform business strategies and decisions. The project involves cleaning, processing, and analyzing historical sales data from various sources to identify trends, patterns, and opportunities for growth. The final deliverable will include a comprehensive report and an interactive dashboard showcasing key findings.

1.Project Goals:

Data Collection and Cleaning:

- Gather sales data from different sources such as POS systems, e-commerce platforms, and CRM systems.

- Clean the data to handle missing values, duplicates, and inconsistencies.

2. Data Analysis:

- Perform exploratory data analysis (EDA) to understand the data distribution and key metrics.

- Identify sales trends over time (daily, monthly, yearly).

- Analyze sales performance across different regions, products, and customer segments.

- Calculate key performance indicators (KPIs) such as average order value, customer lifetime value, and sales growth rate.

3. Data Visualization:

- Create visualizations to represent sales trends, geographic distribution, and product performance.

- Develop an interactive dashboard that allows users to filter and explore the data dynamically.

4. Insights and Recommendations:

- Derive actionable insights from the analysis to help in strategic decision-making.
- Provide recommendations for improving sales performance based on the analysis.

Scope of Work:

1. Data Collection:

- Collect historical sales data from various sources.
- Integrate data from different formats and platforms into a unified dataset.

2.Data Cleaning and Preparation:

- Handle missing data, duplicates, and outliers.
- Standardize data formats for consistency.

3. Exploratory Data Analysis:

- Perform statistical analysis to understand data distribution.
- Visualize data using graphs and charts to identify patterns.

4. Sales Trend Analysis:

- Analyze time-series data to identify sales trends and seasonality.

- Compare sales performance across different periods.

5. Segmentation Analysis:

- Segment data by region, product, and customer demographics.
- Analyze the performance of different segments to identify high-performing areas.

6. Reporting and Presentation:

- Compile findings into a comprehensive report.
- Present key insights and recommendations to stakeholders.

This project description outlines the objectives, goals, scope, tools, and timeline, providing a clear roadmap for analyzing and visualizing sales data.

3. Business problem

Develop a predictive model to identify customers who are likely to cancel their subscription in the near future. Use historical data to understand the factors that contribute to churn and implement strategies to retain high-risk customers.

Data Collection:

1. Age, gender, location, income level, etc.

2.Subscription Details: Subscription type, start date, renewal date, payment method, etc.

3.Usage Data: Frequency of service usage, duration of sessions, feature utilization, etc.

4.Customer Interactions: Customer support tickets, service feedback, complaints, etc.

5.Transaction History: Payment history, missed payments, refunds, etc.

Analytical Approach:

1. Data Cleaning and Preprocessing: Handle missing values, normalize data, and encode categorical variables.

2. Exploratory Data Analysis (EDA): Identify patterns and relationships within the data.

3. Feature Engineering: Create new features based on domain knowledge and data insights.

4. Model Selection: Choose appropriate models (e.g., logistic regression, decision trees, random forests, gradient boosting machines, neural networks).

5. Model Training and Validation: Split the data into training and validation sets, train models, and evaluate performance using metrics like accuracy, precision, recall, F1-score, and ROC-AUC.

6. Hyperparameter Tuning: Optimize model parameters to improve performance.

7. Model Interpretation: Use techniques like feature importance, SHAP values, or LIME to understand model predictions.

8. Implementation: Deploy the model into a production environment and integrate it with the customer management system.

9. Monitoring and Maintenance: Continuously monitor model performance and update it as needed.

Business Actions:

1.Targeted Retention Campaigns: Offer personalized discounts, incentives, or tailored communication to high-risk customers.

2.Enhanced Customer Support: Provide proactive support to customers showing signs of dissatisfaction.

3.Service Improvement: Identify common pain points from churned customers and improve the service accordingly.

By addressing customer churn, businesses can not only retain more customers but also create a more loyal and engaged customer base, leading to sustainable growth and profitability.

4.Analysis :

```
[5]: df = pd.read_csv("Fezzari Sales.csv")
```

```
[6]: df
```

[6]:

	Date	Time	Year	Customer Age	Customer Gender	Country	State	Product Category	Sub Category	Quantity	Unit Cost	Revenue	Payment	Rating
0	02-19-16	13:08	2016.0	29.0	F	United States	Washington	Accessories	Tires and Tubes	1.0	80.00	109.00	Credit Card	6.5
1	2-20-16	10:29	2016.0	29.0	F	United States	Washington	Clothing	Gloves	2.0	24.50	57.00	Credit Card	5.0
2	2-27-16	13:23	2016.0	29.0	F	United States	Washington	Accessories	Tires and Tubes	3.0	3.67	15.00	Cash	7.5
3	12-03-2016	20:33	2016.0	29.0	F	United States	Washington	Accessories	Tires and Tubes	2.0	87.50	233.00	Cash	6.5
4	12-03-2016	10:37	2016.0	29.0	F	United States	Washington	Accessories	Tires and Tubes	3.0	35.00	125.00	Cash	8.0
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
34862	07-02-2016	11:55	2016.0	38.0	M	France	Hauts de Seine	Bikes	Mountain Bikes	2.0	1160.00	1971.00	Ewallet	NaN
34863	3-13-15	11:42	2015.0	38.0	M	France	Hauts de Seine	Bikes	Mountain Bikes	1.0	2049.00	1583.00	Cash	9.5
34864	05-04-2015	14:30	2015.0	38.0	M	France	Hauts de Seine	Bikes	Mountain Bikes	3.0	683.00	1682.00	Cash	7.0
34865	8-30-15	15:11	2015.0	38.0	M	France	Hauts de Seine	Bikes	Mountain Bikes	1.0	2320.00	1568.00	Cash	9.5
34866	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	641.53	Cash	9.5

34867 rows × 14 columns

To cheak null values

```
df.isnull()
```

	Date	Time	Year	Customer Age	Customer Gender	Country	State	Product Category	Sub Category	Quantity	Unit Cost	Revenue	Payment	Rating
0	False	False	False	False	False	False	False	False	False	False	False	False	False	False
1	False	False	False	False	False	False	False	False	False	False	False	False	False	False
2	False	False	False	False	False	False	False	False	False	False	False	False	False	False
3	False	False	False	False	False	False	False	False	False	False	False	False	False	False
4	False	False	False	False	False	False	False	False	False	False	False	False	False	False
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
34862	False	False	False	False	False	False	False	False	False	False	False	False	False	True
34863	False	False	False	False	False	False	False	False	False	False	False	False	False	False
34864	False	False	False	False	False	False	False	False	False	False	False	False	False	False
34865	False	False	False	False	False	False	False	False	False	False	False	False	False	False
34866	True	True	True	True	True	True	True	True	True	True	True	False	False	False

34867 rows × 14 columns

To find null value counts

```
[8]: df.isnull().sum()
```

```
[8]: Date                1
     Time                1
     Year                1
     Customer Age        3
     Customer Gender     3
     Country             1
     State               3
     Product Category    4
     Sub Category        3
     Quantity            1
     Unit Cost           3
     Revenue             1
     Payment             0
     Rating              1
     dtype: int64
```

Finding the null values

```
new_df = df[df.isna().any(axis=1)]
new_df
```

	Date	Time	Year	Customer Age	Customer Gender	Country	State	Product Category	Sub Category	Quantity	Unit Cost	Revenue	Payment	Rating
19	9-24-15	15:30	2015.0	29.0	F	United States	Washington	NaN	Tires and Tubes	1.0	64.00	74.00	Credit Card	6.0
21	07-10-2015	10:40	2015.0	NaN	F	United States	Washington	Accessories	Tires and Tubes	1.0	125.00	136.00	Cash	8.0
58	3-21-16	15:55	2016.0	19.0	M	United States	California	Clothing	Jerseys	1.0	NaN	985.00	Credit Card	10.0
63	8-21-15	12:27	2015.0	19.0	M	United States	California	Clothing	Jerseys	2.0	NaN	1139.00	Credit Card	9.0
66	1-25-16	15:43	2016.0	19.0	F	United States	California	NaN	Helmets	3.0	46.67	155.00	Credit Card	8.5
77	8-14-15	15:48	2015.0	24.0	F	United States	NaN	Accessories	Bike Stands	3.0	530.00	1588.00	Cash	5.5
110	08-08-2015	11:32	2015.0	19.0	F	United States	California	NaN	Tires and Tubes	3.0	1.67	NaN	Credit Card	7.0
140	8-25-15	13:00	2015.0	NaN	M	United States	California	Bikes	Mountain Bikes	3.0	773.33	2046.00	Credit Card	8.0
144	4-16-16	16:37	2016.0	25.0	F	United States	NaN	Clothing	Socks	1.0	207.00	265.00	Credit Card	9.0
151	07-02-2016	16:07	2016.0	53.0	NaN	United States	Washington	Accessories	NaN	2.0	20.00	47.00	Credit Card	7.5

To clean the null values

10]:
```
df=df.dropna(axis=0,how='any')
df
```

10]:

	Date	Time	Year	Customer Age	Customer Gender	Country	State	Product Category	Sub Category	Quantity	Unit Cost	Revenue	Payment	Rating
0	02-19-16	13:08	2016.0	29.0	F	United States	Washington	Accessories	Tires and Tubes	1.0	80.00	109.0	Credit Card	6.5
1	2-20-16	10:29	2016.0	29.0	F	United States	Washington	Clothing	Gloves	2.0	24.50	57.0	Credit Card	5.0
2	2-27-16	13:23	2016.0	29.0	F	United States	Washington	Accessories	Tires and Tubes	3.0	3.67	15.0	Cash	7.5
3	12-03-2016	20:33	2016.0	29.0	F	United States	Washington	Accessories	Tires and Tubes	2.0	87.50	233.0	Cash	6.5
4	12-03-2016	10:37	2016.0	29.0	F	United States	Washington	Accessories	Tires and Tubes	3.0	35.00	125.0	Cash	8.0
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
34860	2-24-16	13:24	2016.0	37.0	M	Germany	Bayern	Bikes	Mountain Bikes	2.0	384.50	1072.0	Cash	7.0
34861	3-22-15	18:06	2015.0	38.0	M	France	Charente-Maritime	Bikes	Mountain Bikes	1.0	2049.00	1487.0	Cash	9.0
34863	3-13-15	11:42	2015.0	38.0	M	France	Hauts de Seine	Bikes	Mountain Bikes	1.0	2049.00	1583.0	Cash	9.5
34864	05-04-2015	14:30	2015.0	38.0	M	France	Hauts de Seine	Bikes	Mountain Bikes	3.0	683.00	1682.0	Cash	7.0
34865	8-30-15	15:11	2015.0	38.0	M	France	Hauts de Seine	Bikes	Mountain Bikes	1.0	2320.00	1568.0	Cash	9.5

After we clean the all null values

```
df.isna().sum()
```

```
Date                0
Time                0
Year                0
Customer Age        0
Customer Gender     0
Country             0
State               0
Product Category    0
Sub Category        0
Quantity            0
Unit Cost           0
Revenue             0
Payment             0
Rating              0
dtype: int64
```

To getting the data type

```
[12]: df.dtypes
```

```
[12]: Date                 object
      Time                 object
      Year                float64
      Customer Age        float64
      Customer Gender      object
      Country              object
      State                object
      Product Category     object
      Sub Category         object
      Quantity            float64
      Unit Cost           float64
      Revenue             float64
      Payment              object
      Rating              float64
      dtype: object
```

```
df.describe()
```

	Year	Customer Age	Quantity	Unit Cost	Revenue	Rating
count	34853.000000	34853.000000	34853.000000	34853.000000	34853.000000	34853.000000
mean	2015.569277	36.384902	2.002496	349.886978	640.863340	7.491177
std	0.495185	11.111604	0.813929	490.055370	736.644793	1.578697
min	2015.000000	17.000000	1.000000	0.670000	2.000000	5.000000
25%	2015.000000	28.000000	1.000000	45.000000	102.000000	6.000000
50%	2016.000000	35.000000	2.000000	150.000000	319.000000	7.500000
75%	2016.000000	44.000000	3.000000	455.000000	902.000000	9.000000
max	2016.000000	87.000000	3.000000	3240.000000	5082.000000	10.000000

Additional column creating for month

df['Month'] = pd.to_datetime(df['Date']).dt.month

	Date	Time	Year	Customer Age	Customer Gender	Country	State	Product Category	Sub Category	Quantity	Unit Cost	Revenue	Payment	Rating	Month
0	02-19-16	13:08	2016	29	F	United States	Washington	Accessories	Tires and Tubes	1	80.00	109.0	Credit Card	6.5	2
1	2-20-16	10:29	2016	29	F	United States	Washington	Clothing	Gloves	2	24.50	57.0	Credit Card	5.0	2
2	2-27-16	13:23	2016	29	F	United States	Washington	Accessories	Tires and Tubes	3	3.67	15.0	Cash	7.5	2
3	12-03-2016	20:33	2016	29	F	United States	Washington	Accessories	Tires and Tubes	2	87.50	233.0	Cash	6.5	12
4	12-03-2016	10:37	2016	29	F	United States	Washington	Accessories	Tires and Tubes	3	35.00	125.0	Cash	8.0	12
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
34860	2-24-16	13:24	2016	37	M	Germany	Bayern	Bikes	Mountain Bikes	2	384.50	1072.0	Cash	7.0	2
34861	3-22-15	18:06	2015	38	M	France	Charente-Maritime	Bikes	Mountain Bikes	1	2049.00	1487.0	Cash	9.0	3
34863	3-13-15	11:42	2015	38	M	France	Hauts de Seine	Bikes	Mountain Bikes	1	2049.00	1583.0	Cash	9.5	3
34864	05-04-2015	14:30	2015	38	M	France	Hauts de Seine	Bikes	Mountain Bikes	3	683.00	1682.0	Cash	7.0	5
34865	8-30-15	15:11	2015	38	M	France	Hauts de Seine	Bikes	Mountain Bikes	1	2320.00	1568.0	Cash	9.5	8

34853 rows × 15 columns

To find Which one is the best month for sales? How much was earned this month?

```
df['Sales Cost'] = df['Quantity']* df['Unit Cost']

df
```

	Date	Time	Year	Customer Age	Customer Gender	Country	State	Product Category	Sub Category	Quantity	Unit Cost	Revenue	Payment	Rating	Month	Sales Cost
0	02-19-16	13:08	2016	29	F	United States	Washington	Accessories	Tires and Tubes	1	80.00	109.0	Credit Card	6.5	2	80.00
1	2-20-16	10:29	2016	29	F	United States	Washington	Clothing	Gloves	2	24.50	57.0	Credit Card	5.0	2	49.00
2	2-27-16	13:23	2016	29	F	United States	Washington	Accessories	Tires and Tubes	3	3.67	15.0	Cash	7.5	2	11.01
3	12-03-2016	20:33	2016	29	F	United States	Washington	Accessories	Tires and Tubes	2	87.50	233.0	Cash	6.5	12	175.00
4	12-03-2016	10:37	2016	29	F	United States	Washington	Accessories	Tires and Tubes	3	35.00	125.0	Cash	8.0	12	105.00
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
34860	2-24-16	13:24	2016	37	M	Germany	Bayern	Bikes	Mountain Bikes	2	384.50	1072.0	Cash	7.0	2	769.00
34861	3-22-15	18:06	2015	38	M	France	Charente-Maritime	Bikes	Mountain Bikes	1	2049.00	1487.0	Cash	9.0	3	2049.00
34863	3-13-15	11:42	2015	38	M	France	Hauts de Seine	Bikes	Mountain Bikes	1	2049.00	1583.0	Cash	9.5	3	2049.00
34864	05-04-2015	14:30	2015	38	M	France	Hauts de Seine	Bikes	Mountain Bikes	3	683.00	1682.0	Cash	7.0	5	2049.00
34865	8-30-15	15:11	2015	38	M	France	Hauts de Seine	Bikes	Mountain Bikes	1	2320.00	1568.0	Cash	9.5	8	2320.00

34853 rows × 16 columns

Grouping Month by Sales Cost

```
]: df_temp = df.groupby('Month').sum()['Sales Cost'].reset_index()
   df_temp
```

]:

	Month	Sales Cost
0	1	1734134.19
1	2	1612798.61
2	3	1735518.24
3	4	1779628.02
4	5	2014706.84
5	6	2130992.54
6	7	1431118.20
7	8	1430123.61
8	9	1409997.43
9	10	1385913.93
10	11	1525403.64
11	12	1883753.77

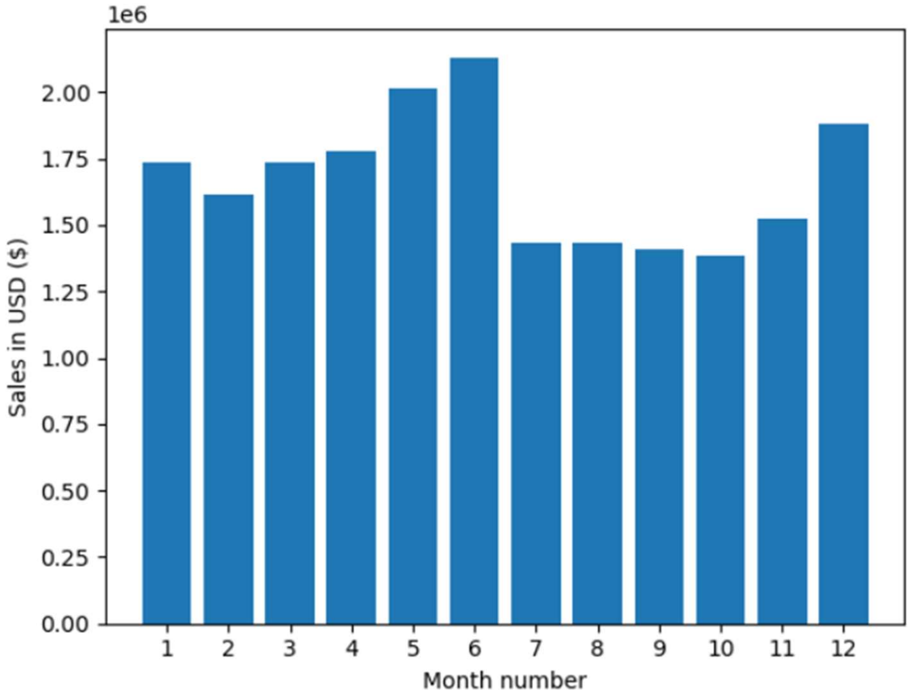
Plotting Sales by Month

```
import matplotlib.pyplot as plt

months = range(1,13)
print(months)

plt.bar(months,df.groupby(['Month']).sum()['Sales Cost'])
plt.xticks(months)
plt.ylabel('Sales in USD ($)')
plt.xlabel('Month number')
plt.show()
```

```
range(1, 13)
```



inference:

Here in Month of 6 has high Sales. So June is the best month for sales with 2 USD to 3 USD

Sorting Product Category and Subcategory based on revenue

```
cat_subcat = pd.DataFrame(df.groupby(['Product Category', 'Sub Category']).sum()['Revenue'])
cat_subcat.sort_values(['Product Category','Revenue'], ascending=False)
```

Product Category	Sub Category	Revenue
Clothing	Jerseys	1831986.0
	Shorts	689184.0
	Vests	368681.0
	Caps	255992.0
	Gloves	228353.0
	Socks	60972.0
Bikes	Mountain Bikes	5172439.0
	Road Bikes	3921989.0
	Touring Bikes	2387910.0
Accessories	Tires and Tubes	2865498.0
	Helmets	2738055.0
	Bottles and Cages	709407.0
	Hydration Packs	403276.0
	Fenders	329204.0
	Bike Stands	149323.0
	Bike Racks	140854.0

Inference:

i)In Clothing,jerseys are more profitable

ii)In Bikes,Mountain Bikes are more profitable

iii)In Accessories Tires and Tubes are more profitable

Sorting top 10 Products by Sales Cost

```
product_sales = pd.DataFrame(df.groupby('Sub Category').sum()['Sales Cost'])
product_sales.sort_values(by=['Sales Cost'], inplace=True, ascending=False)
product_sales.head(10)
```

	Sales Cost
Sub Category	
Mountain Bikes	5027183.58
Road Bikes	3823818.35
Tires and Tubes	2353428.93
Touring Bikes	2293103.10
Helmets	2219595.58
Jerseys	1531295.87
Shorts	602139.99
Bottles and Cages	579840.55
Hydration Packs	330935.06
Vests	310337.04

Which is the most sold product?

Grouping and Sorting products based on Quantity

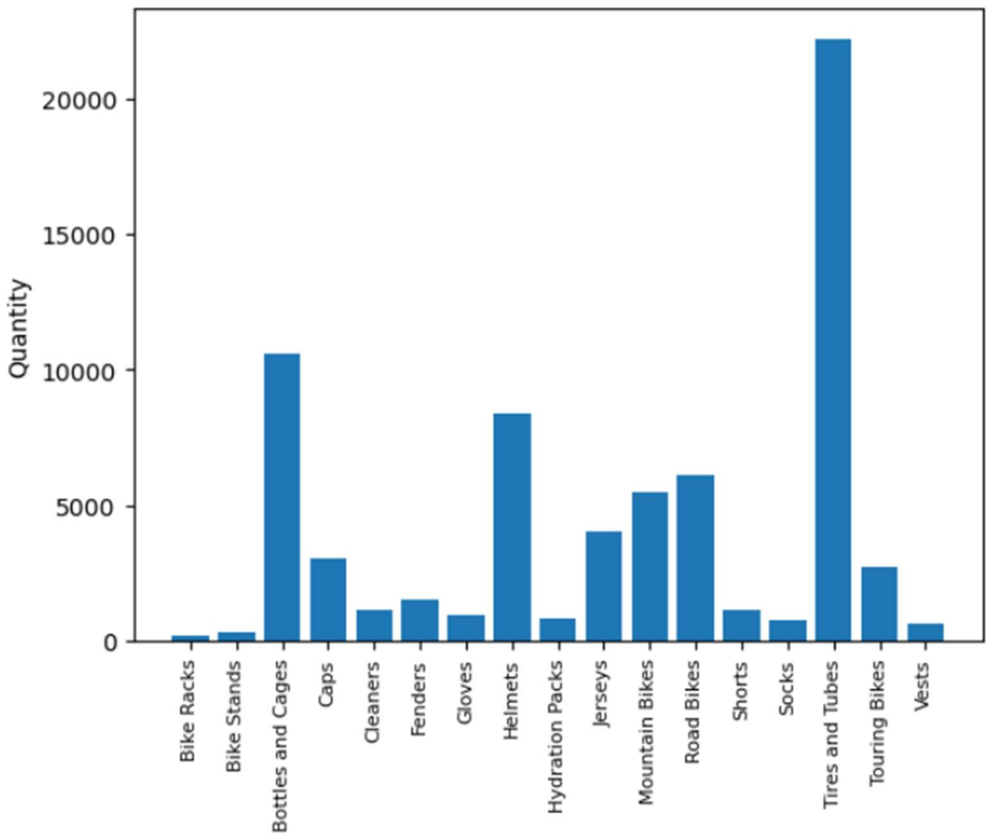
```
most_selling_products = pd.DataFrame(df.groupby('Sub Category').sum()['Quantity'])
most_selling_products.sort_values(by=['Quantity'], inplace=True, ascending=False)
most_selling_products[:10]
```

	Quantity
Sub Category	
Tires and Tubes	22201
Bottles and Cages	10558
Helmets	8384
Road Bikes	6119
Mountain Bikes	5494
Jerseys	4030
Caps	3020
Touring Bikes	2673
Fenders	1494
Shorts	1129

Plotting most sold products based on Quantity

```
product_group = df.groupby('Sub Category')
quantity = product_group.sum()['Quantity']

keys = [pair for pair, df in product_group]
plt.bar(keys, quantity)
plt.xticks(keys, rotation='vertical', size=8)
plt.ylabel('Quantity')
plt.xlabel('Sub Category')
plt.show()
```



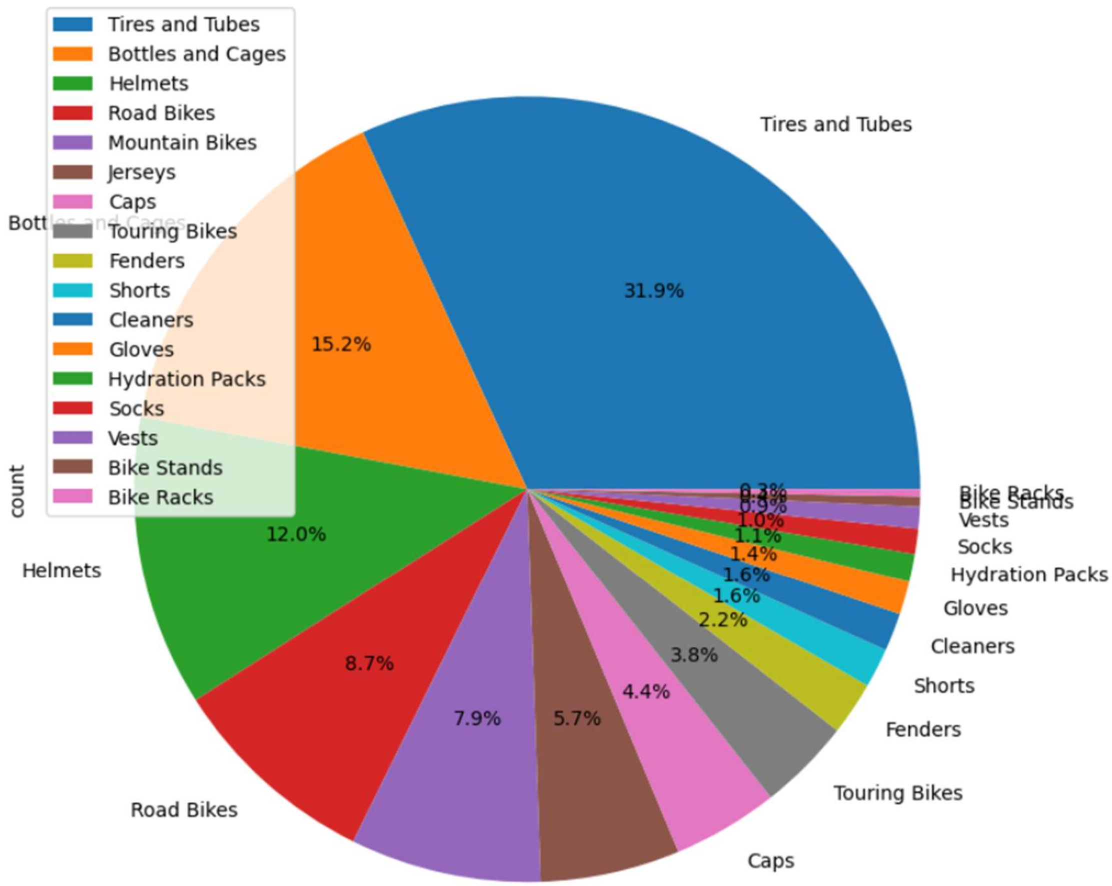
Inference:

maximum of the customer bought tires and tubes

category_chart=df['Sub Category'].value_counts()

category_chart.plot(kind = 'pie', autopct='%1.1f%%', figsize=(9, 15)).legend()

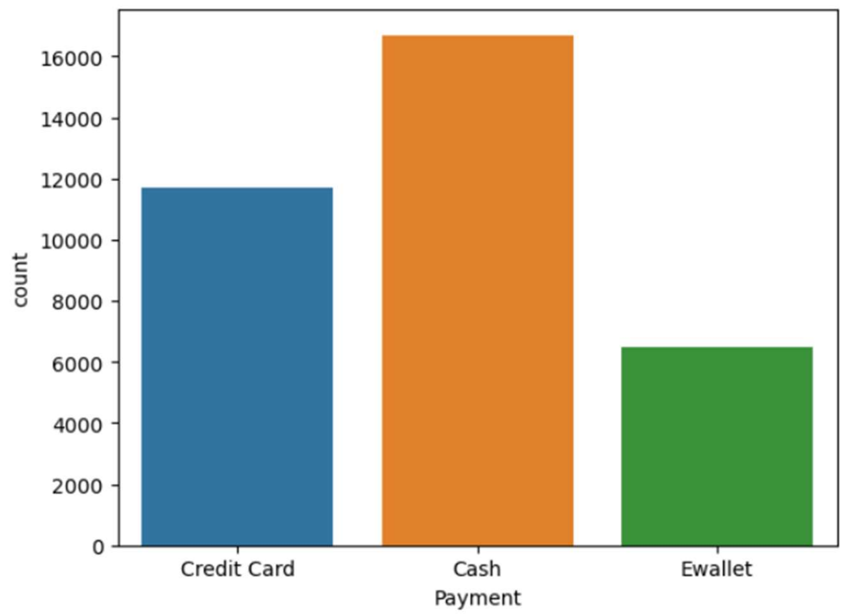
<matplotlib.legend.Legend at 0x1abc2b48c10>



Counting Payment modes

```
df['Payment'].value_counts()
```

```
Payment
Cash           16699
Credit Card    11688
Ewallet         6466
Name: count, dtype: int64
```



1.most of the customers using cash payment hiked upto count of 16000

2.customers using credit card are in the range from 11500 to 12000

3.least amount of customer using ewallet upto 6000

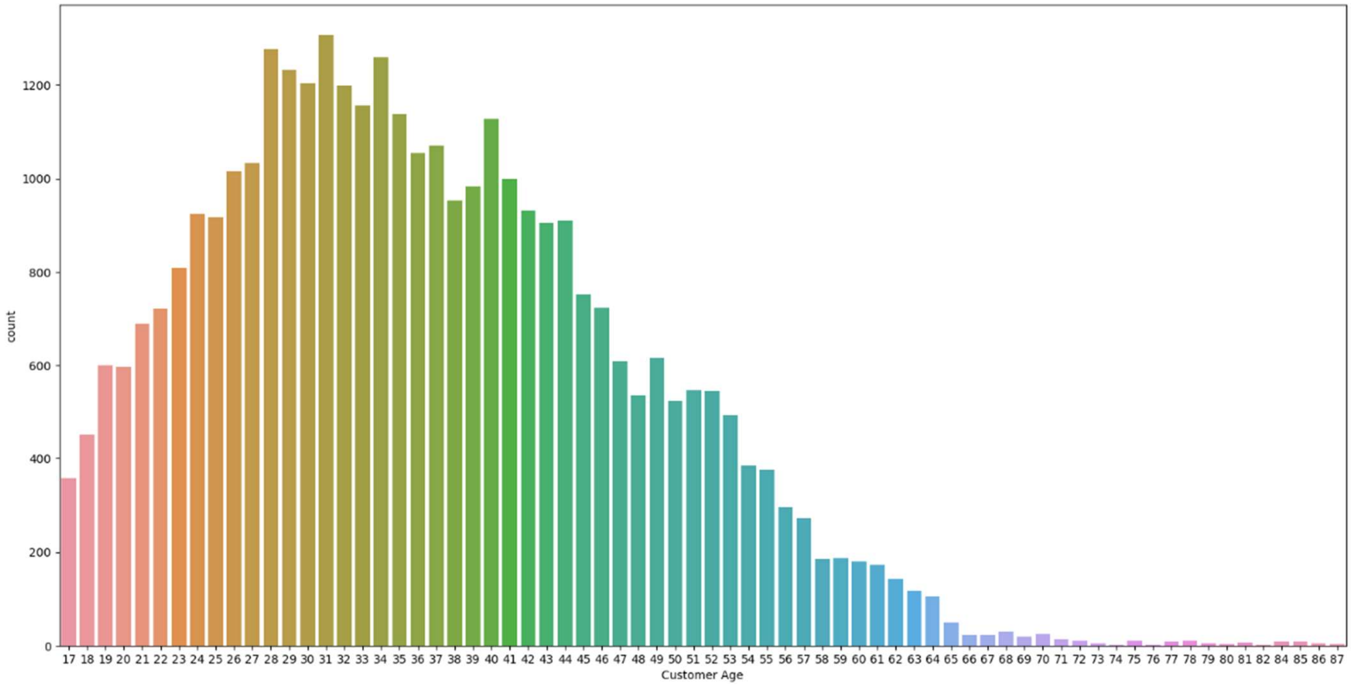
From this we can know that people are using cash payment which are more convienient

Grouping the customer age and quantity

```
Age=df.groupby('Customer Age').sum()['Quantity']
Age=Age[Age == Age.max()]
Age
```

```
Customer Age
31    2630
Name: Quantity, dtype: int32
```

```
import seaborn as sns
plt.figure(figsize=(20, 10))
sns.countplot(x='Customer Age', data=df)
plt.show()
```

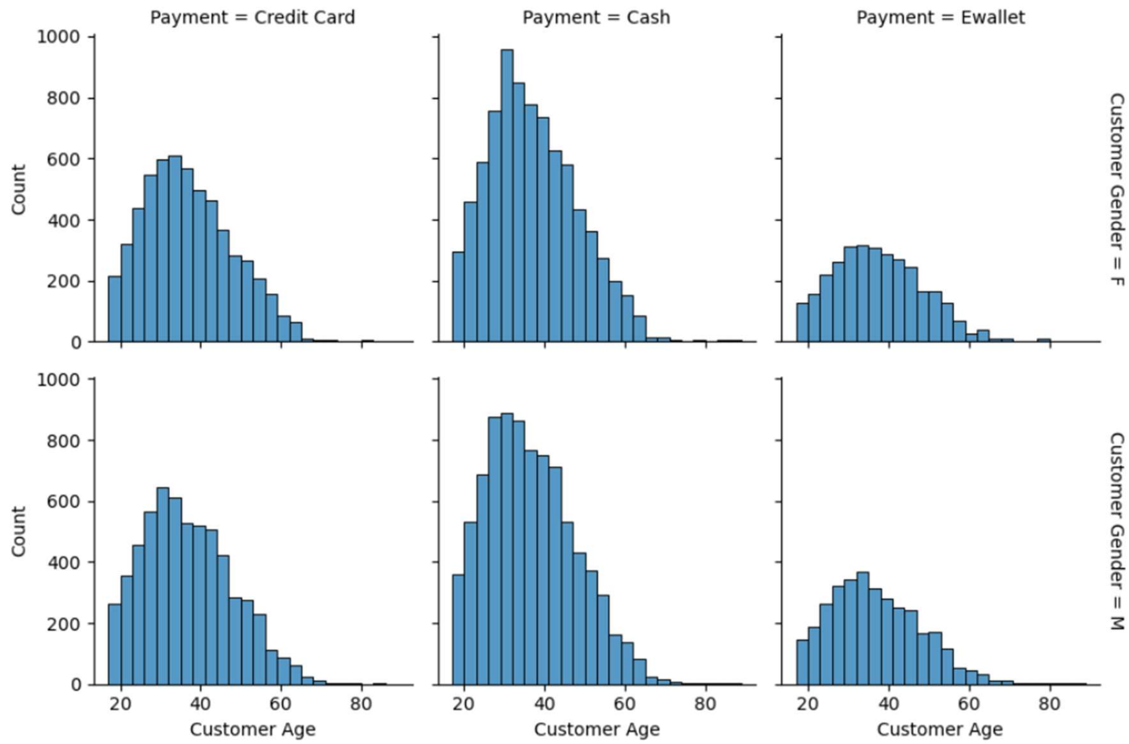


Inference :

The age category of 31 have done more purchase when compare to other age categories.

[36]:

```
<seaborn.axisgrid.FacetGrid at 0x23d7e0ee350>
```



Inference :

1.comparatively female using hand cash more than male

2.women use hand cash count upto 900

3.male use hand cash count nearly 900

1.comparativly male using credit card more than female

2.male using credit card count is 650

3.female using count is nearly 610

1.least amount of customers using ewallet

2.using count of female is nearly 300

3.and the male count is upto 350¶

One sample T-test

```
# from scipy.stats import ttest_1samp
from scipy.stats import ttest_1samp

hm = 36.40  # HYPOTHESISED MEAN
t_stat, p_value = ttest_1samp(df['Customer Age'],hm)
alpha = 0.05 # acceptance error percent
print(f"T-statistic:{t_stat}")
print(f"P-value:{p_value}")
if p_value<alpha:
    print(f"Reject null hypothesis at alpha ={alpha}")
else:
    print(f"Failed to reject null hypothesis at alpha = {alpha}")
```

```
T-statistic:-0.25366118574988133
P-value:0.7997588189996673
Failed to reject null hypothesis at alpha = 0.05
```

Given hypothysis Mean is 36.40 for group of AGE with significance level of error is 5 % allowed

In this OneSample T-test We can confidently tell 95% the given mean value is FAILD TO REJECT NULL HYPOTHISIS because of the p_value is less than significant level ¶

Conclusion

Sales datasets are a critical asset for any business seeking to thrive in a competitive marketplace. By effectively collecting, analyzing, and utilizing sales data, companies can make informed decisions that drive revenue growth, enhance customer satisfaction, and ensure long-term success.