**Supermarket Sales Data:-**

**1. Aim of the Project:**

To enhance business performance, focus on understanding customer behavior by segmenting data by type, gender, and city. Analyze product line profitability to prioritize high-margin items and assess branch and city performance to allocate resources effectively. Optimize sales through pricing and inventory adjustments, improve customer satisfaction by addressing feedback, and streamline payment processes based on preferences**.**

### 1. Customer Insights

**Aim:** Understand customer behavior and preferences.

* **Action:** Segment by type, gender, and city to tailor marketing and offers.

### 2. Product Line Performance

**Aim:** Identify top-performing and profitable products.

* **Action:** Focus on high-margin and high-volume products for promotions.

### 3. Branch and City Analysis

**Aim:** Evaluate branch and city performance.

* **Action:** Allocate resources and adjust strategies based on performance data.

### 4. Sales Optimization

**Aim:** Increase sales and revenue through pricing and inventory management.

* **Action:** Adjust pricing and manage inventory based on sales data.

### 5. Customer Satisfaction

**Aim:** Improve satisfaction and loyalty.

* **Action:** Enhance service quality based on customer ratings.

### 6. Payment Preferences

**Aim:** Streamline payment processes.

* **Action:** Optimize and promote preferred payment methods.

These aims will help drive targeted strategies and improve overall business performance.

**2. Business Problem or Problem Statement:**

The company faces challenges in optimizing its sales and marketing strategies due to a lack of detailed insights into customer behavior, product performance, and branch efficiency. There is a need to better understand customer preferences, identify profitable product lines, evaluate branch and city performance, and streamline payment processes to enhance overall business performance and drive revenue growth.

* **Lack of Customer Insights:** Inadequate understanding of customer preferences and behavior hampers targeted marketing and personalization.
* **Inefficient Product Line Management:** Difficulty in identifying which product lines are most profitable and popular affects inventory and promotional strategies.
* **Branch and City Performance Issues:** Variability in performance across branches and cities makes it challenging to allocate resources effectively.
* **Payment Process Optimization:** Unclear payment method preferences impact the efficiency and convenience of the payment process.

**3. Project Description:**

The project aims to analyze sales data to improve business operations by gaining deeper insights into customer behavior, product performance, and branch efficiency. The analysis will segment customers to enhance targeted marketing, evaluate product lines to optimize inventory and promotions, and assess branch and city performance to allocate resources more effectively. Additionally, the project will examine payment method preferences to streamline and improve the payment process, ultimately driving increased revenue and customer satisfaction.

1. **Customer Segmentation:**

2. **Product Line Evaluation:**

3**.Branch and City Performance Analysis:**

4. **Payment Method Analysis:**

5. **Revenue Optimization:**

**4. Functionalities:**

• **Customer Segmentation and Insights:** Provide detailed customer profiles and segmentation based on behavior, demographics, and purchase history to enable targeted marketing and personalized offers.

1. **Product Line Performance Analysis:** Generate reports and dashboards to evaluate the profitability, sales volume, and performance of different product lines, helping to optimize inventory and promotional strategies.
2. **Branch and City Performance Tracking:** Develop tools to compare and analyze sales performance, customer ratings, and other key metrics across various branches and cities for better resource allocation.
3. **Payment Method Analysis:** Offer insights into payment method preferences and usage patterns, facilitating improvements in the payment process and offering targeted promotions based on preferred payment options.
4. **Revenue and Sales Optimization:** Implement features to analyze pricing strategies, identify opportunities for revenue growth, and adjust inventory management practices to enhance overall profitability.

**5. Input Versatility with Error Handling and Exception Handling:**

* **Diverse Input Formats:** The system will handle various input formats for data, such as CSV, Excel, and database queries, ensuring flexibility in data integration and processing. Users will be able to upload and import data from multiple sources with ease.
* **Validation of Input Data:** Robust validation mechanisms will be implemented to check for data integrity and accuracy. This includes verifying data types, ranges, and required fields to prevent incorrect or incomplete data from being processed.
* **Error Handling:** The system will include comprehensive error handling to manage common issues, such as missing or corrupted files, invalid data formats, and connection errors. User-friendly error messages will guide users to resolve issues efficiently.
* **Exception Management:** Exceptions will be captured and logged systematically to track and address unexpected issues. The system will provide detailed error logs and alerts to assist in diagnosing and correcting problems quickly.
* **User Feedback and Recovery:** In cases of input errors or exceptions, the system will offer clear feedback and recovery options, such as data correction prompts and retry mechanisms, to ensure smooth user experience and minimal disruption.

**6. Code Implementation:**

* **Description:**

The code handles diverse input formats by using pandas to read CSV and Excel files, and provides error messages for unsupported formats. It validates data to ensure all required columns are present and data types are correct, reporting issues if validation fails. Error handling is integrated to manage issues like missing files or incorrect data, while exception management ensures the system can handle errors such as division by zero without crashing. The user interface allows for clear feedback and recovery from errors during data processing and input operations.

**Importing Pakages:-**

import pandas as pd

import numpy as np

import matplotlib.pyplot as plt

import plotly.express as px

import plotly.graph\_objects as go

import seaborn as sns

import warnings

warnings.filterwarnings('ignore')

**Importing Dataset:-**

sales=pd.read\_csv("supermarket\_sales.csv")

pd.set\_option('display.max\_columns', None)

sales

**Renaming columns:-**

sales.rename(columns={'Product line':'Products','Payment':'Payment methods'},inplace=True)

sales

**Droping columns:-**

sales = sales.drop("Invoice ID",axis = 1)

sales

**Initial Data Exploration:-**

sales.info()

sales.describe()

sales.shape

sales.isnull().sum().sum()

**Checking for outliers:-**

import matplotlib.pyplot as plt

sales.boxplot(column=['Unit price','Quantity','Tax 5%','Total'])

plt.ylabel('Values')

plt.title('Box plot of Four Columns')

plt.xticks(rotation=30)

plt.show()

**Removing Outliers:-**

import numpy as np

Q1=sales['Total'].quantile(0.25)

Q3=sales['Total'].quantile(0.75)

IQR=Q3-Q1

lower\_bound = Q1 - 1.5 \*IQR

Upper\_bound = Q3 + 1.5 \*IQR

print(f"Lower Bound: {lower\_bound}")

print(f"Upper Bound: {Upper\_bound}")

median\_value = sales['Total'].median()

print(f"Median value: {median\_value}")

sales['Total']=np.where((sales['Total']< lower\_bound) | (sales['Total']> Upper\_bound),median\_value,sales['Total'])

print("Data after Replacing outliers with Median:\n",sales)

**Exploratory Data Analysis(EDA):-**

* Barplot
* Countplot
* Histplot
* Histogram
* Linechart
* Violinplot
* Pie chart
* FacetGrid
* Jointplot
* Pairplot
* Heatmap

**Statistics Test:-**

* One Way Annova

import pandas as pd

import scipy.stats as statsanova\_result = stats.f\_oneway(

sales[sales['Product line'] == 'Health and beauty']['Total'],

sales[sales['Product line'] == 'Electronic accessories']['Total'],

sales[sales['Product line'] == 'Home and lifestyle']['Total'],

sales[sales['Product line'] == 'Sports and travel']['Total'],

sales[sales['Product line'] == 'Food and beverages']['Total'],

sales[sales['Product line'] == 'Fashion accessories']['Total']

)

alpha = 0.05

print("ANOVA F-statistic:", anova\_result.statistic)

print("ANOVA p-value:", anova\_result.pvalue)

if anova\_result.pvalue < alpha:

print("Reject the null hypothesis.")

else:

print("Fail to reject the null hypothesis.")

**7. Results and Outcomes:**

* **Improved Data Integration:** The system successfully handles various input formats (CSV, Excel), allowing for seamless integration of data from multiple sources.
* **Enhanced Data Quality:** Data validation processes ensure that all required columns are present and data types are accurate, reducing errors and improving the reliability of subsequent analyses.
* **Robust Error Handling:** The system effectively manages common errors, such as unsupported file formats and data inconsistencies, providing clear feedback to users and preventing data processing failures.
* **Effective Exception Management:** The system handles exceptions, such as division by zero, gracefully, ensuring stability and continuity of operations without crashes.
* **User-Friendly Experience:** The interface provides clear guidance and recovery options, enhancing the overall user experience and allowing for efficient data processing and interaction.

**8. Conclusion:**

The implementation successfully addresses key challenges in data handling and processing. By supporting diverse input formats and incorporating rigorous data validation, the system ensures reliable data integration and quality. Robust error and exception management mechanisms prevent disruptions and maintain system stability, while user-friendly feedback enhances the overall experience. These functionalities collectively improve data accuracy, streamline operations, and facilitate effective decision-making, ultimately driving better business outcomes.