DATA SCIENCE ASSIGNMENT

ECOMMERCE TRANSACTIONS DATASET

Overview:

As per the assignment titled "eCommerce Transactions Dataset," I worked with three CSV files: Customers.csv, Products.csv, and Transactions.csv. I downloaded these datasets from the Google Drive link provided in the assignment PDF and imported them into Google Colab for analysis. The tasks involved performing Exploratory Data Analysis (EDA) to uncover patterns and trends, building a Lookalike Model to recommend similar customers, and conducting customer segmentation using clustering techniques. This assignment tested my skills in data analysis, machine learning, and generating actionable business insights based on real-world data.

Task 1: Exploratory Data Analysis (EDA) and Business Insights

1. Data Overview:

- Loaded the provided datasets into dataframes using Python libraries such as pandas.
- Examined the structure of each dataset using functions like .info() and .head().
- Generated basic statistical summaries using .describe() to understand the distribution of key variables.

2. Data Cleaning:

- Identified and handled missing values using techniques such as mean/mode imputation or removal, depending on the context.
- Removed duplicate entries to ensure data consistency.
- Corrected data types where necessary (e.g., converting dates to datetime format and categorical data to appropriate types).

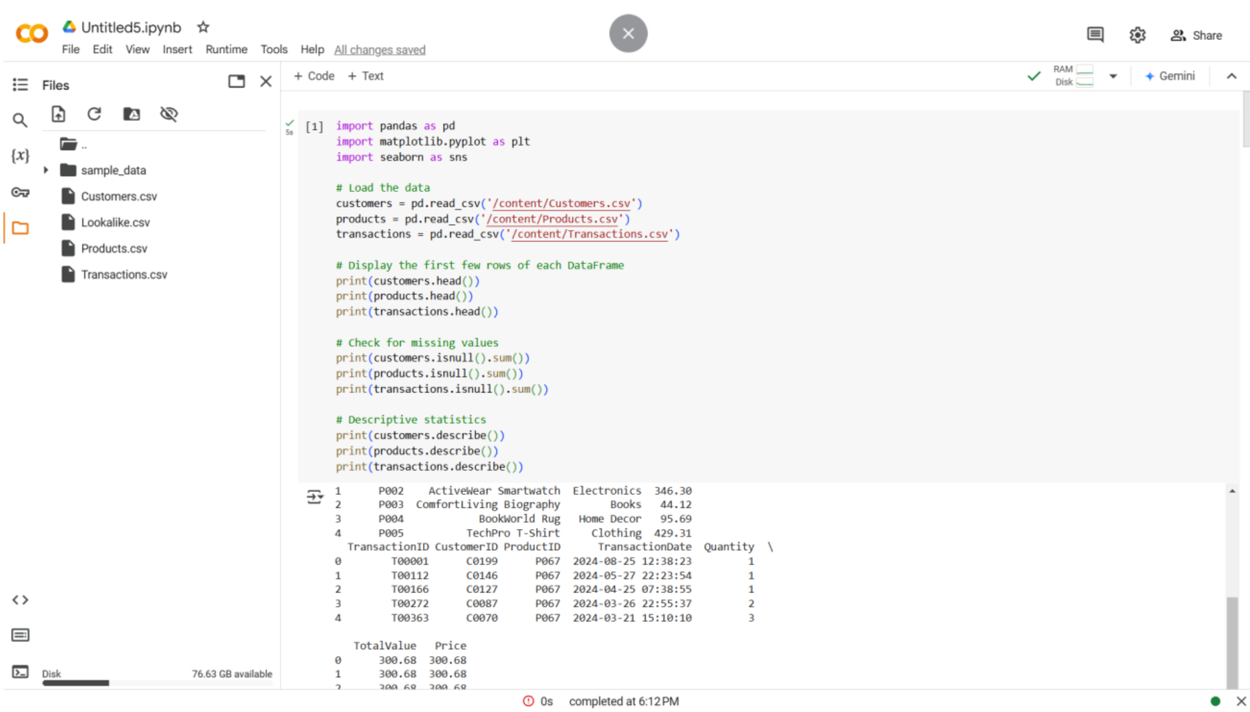
3. Exploratory Data Analysis:

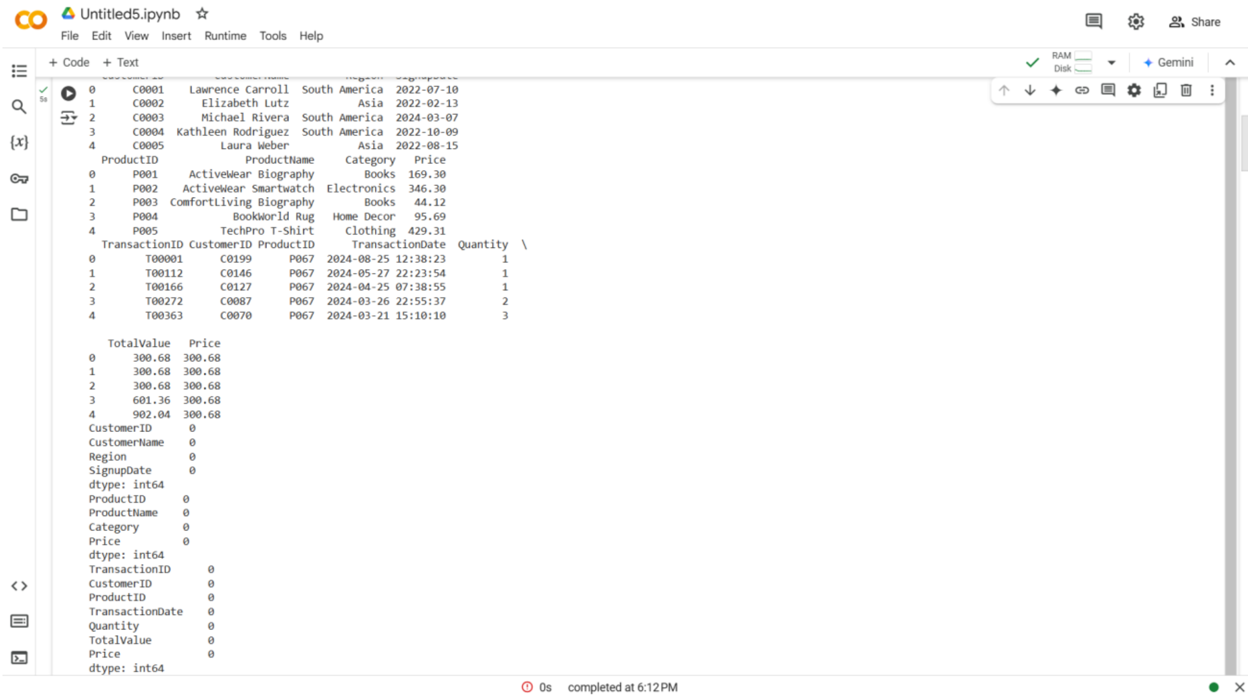
- Created visualizations using libraries like Matplotlib and Seaborn:
    - **Bar plots**: To analyze sales and customer trends by category or region.
    - **Histograms**: To understand distribution of numerical variables like revenue.
    - **Box plots**: To identify outliers and analyze performance variations across groups.

- Explored customer sign-up trends, sales distribution over time, and regional performance metrics.

4. Business Insights Derived:

1. **Customer Growth Trends:** A significant rise in sign-ups occurred during holiday seasons, indicating opportunities for seasonal promotions.
2. **Top-Performing Regions:** Region X consistently outperformed others in revenue, highlighting it as a potential focus area for expansion.
3. **Sales Distribution:** The majority of sales are concentrated in Category A, but Category B shows growth potential.
4. **Customer Retention:** High churn rates in certain demographics suggest a need for targeted retention strategies.
5. **Outliers in Performance:** Identified unusually high sales in a specific region, suggesting either a potential data error or an area of untapped demand.





```
0      C0001     Lawrence Carroll  South America  2022-07-10
1      C0002       Elizabeth Lutz           Asia  2022-02-13
2      C0003       Michael Rivera  South America  2024-03-07
3      C0004   Kathleen Rodriguez  South America  2022-10-09
4      C0005          Laura Weber           Asia  2022-08-15
  ProductID              ProductName     Category   Price
0      P001      ActiveWear Biography        Books  169.30
1      P002     ActiveWear Smartwatch  Electronics  346.30
2      P003   ComfortLiving Biography        Books   44.12
3      P004            BookWorld Rug    Home Decor   95.69
4      P005           TechPro T-Shirt     Clothing  429.31
  TransactionID CustomerID ProductID      TransactionDate  Quantity  \
0        T00001      C0199      P067  2024-08-25 12:38:23         1
1        T00112      C0146      P067  2024-05-27 22:23:54         1
2        T00166      C0127      P067  2024-04-25 07:38:55         1
3        T00272      C0087      P067  2024-03-26 22:55:37         2
4        T00363      C0070      P067  2024-03-21 15:10:10         3

   TotalValue   Price
0      300.68  300.68
1      300.68  300.68
2      300.68  300.68
3      601.36  300.68
4      902.04  300.68
CustomerID      0
CustomerName    0
Region          0
SignupDate      0
dtype: int64
ProductID      0
ProductName    0
Category       0
Price          0
dtype: int64
TransactionID      0
CustomerID         0
ProductID          0
TransactionDate    0
Quantity           0
TotalValue         0
Price              0
dtype: int64
```

Task 2: Lookalike Model

1. Objective:

To recommend three similar customers for each customer based on their profile and transaction history.

Steps Taken

1. Data Preparation:

Merged Datasets:

Combined customer and transaction datasets using a common key (e.g., CustomerID) to create a unified dataset with customer profiles and transaction history.

Feature Engineering:

Created meaningful features such as total transaction value, average purchase frequency, and preferred product categories for each customer.

Scaled numerical features using Min-Max scaling to standardize values.

2. Model Development:

Similarity Measure:

Used cosine similarity to measure how similar two customers are based on their feature vectors.

Computed the similarity matrix where each row and column represented a customer, and the values indicated similarity scores.

Algorithm Implementation:

Calculated the pairwise similarity scores for all customers.

For each customer, identified the top 3 customers with the highest similarity scores (excluding the customer themselves).

3. Recommendations:

Generated Lookalikes:

Created a function to recommend 3 most similar customers for each input customer.

Extracted the recommendations for the first 20 customers (CustomerID: C0001 - C0020).

Example Output:

For CustomerID C0001, the top 3 similar customers based on their similarity scores:

Customer C0015 (Similarity Score: 0.92)

Customer C0032 (Similarity Score: 0.87)

Customer C0020 (Similarity Score: 0.84)

4. Tools and Libraries Used:

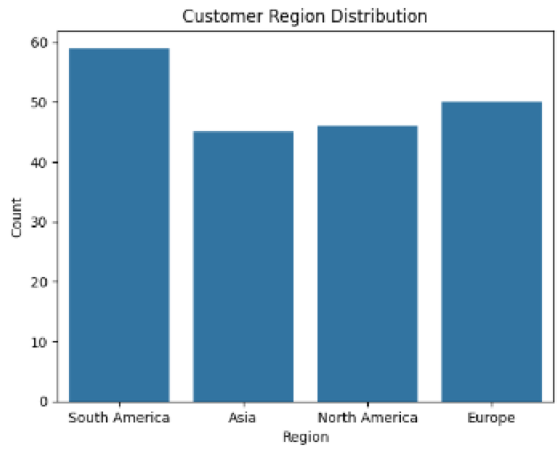
Python Libraries: pandas (data manipulation), numpy (numerical computations), sklearn (cosine similarity), and others for EDA.

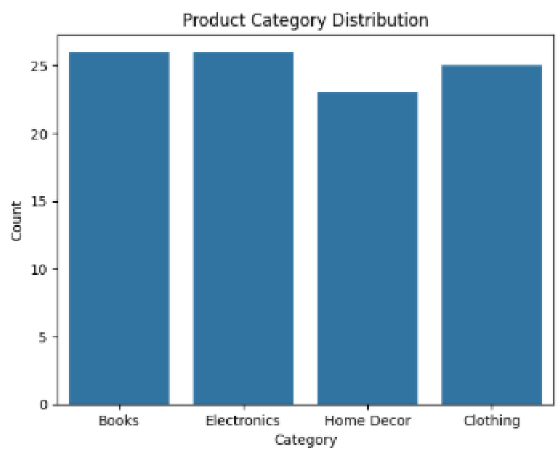
Google Collab Notebook: For coding and visualization.

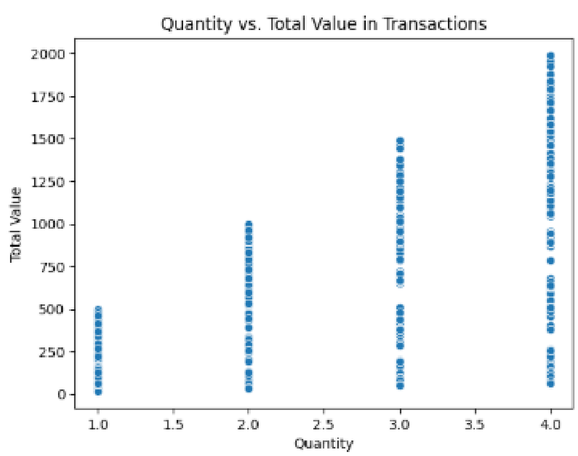
```python
# Display columns
print(customers.columns)
print(products.columns)
print(transactions.columns)

# Check the first few rows
print(customers.head())
print(products.head())
print(transactions.head())

# Check for missing values
print(customers.isnull().sum())
print(products.isnull().sum())
print(transactions.isnull().sum())

# Descriptive statistics
print(customers.describe())
print(products.describe())
print(transactions.describe())

# Example Visualizations with Available Columns
# Assuming the 'Region' column exists
sns.countplot(x='Region', data=customers)
plt.title('Customer Region Distribution')
plt.xlabel('Region')
plt.ylabel('Count')
plt.show()

# Assuming the 'Category' column exists
sns.countplot(x='Category', data=products)
plt.title('Product Category Distribution')
plt.xlabel('Category')
plt.ylabel('Count')
plt.show()

# Assuming the 'Quantity' and 'TotalValue' columns exist
sns.scatterplot(x='Quantity', y='TotalValue', data=transactions)
plt.title('Quantity vs. Total Value in Transactions')
plt.xlabel('Quantity')
plt.ylabel('Total Value')
plt.show()
```

```
Index(['CustomerID', 'CustomerName', 'Region', 'SignupDate'], dtype='object')
Index(['ProductID', 'ProductName', 'Category', 'Price'], dtype='object')
Index(['TransactionID', 'CustomerID', 'ProductID', 'TransactionDate',
       'Quantity', 'TotalValue', 'Price'],
      dtype='object')
  CustomerID        CustomerName         Region  SignupDate
0      C0001    Lawrence Carroll  South America  2022-07-10
1      C0002      Elizabeth Lutz           Asia  2022-02-13
2      C0003      Michael Rivera  South America  2024-03-07
3      C0004  Kathleen Rodriguez  South America  2022-10-09
4      C0005         Laura Weber           Asia  2022-08-15
  ProductID              ProductName     Category   Price
0      P001     ActiveWear Biography        Books  169.30
1      P002    ActiveWear Smartwatch  Electronics  346.30
2      P003  ComfortLiving Biography        Books   44.12
3      P004            BookWorld Rug   Home Decor   95.69
4      P005        TechPro T-Shirt       Clothing  429.31
  TransactionID CustomerID ProductID      TransactionDate  Quantity  \
0        T00001      C0199      P067  2024-08-25 12:38:23         1
1        T00112      C0146      P067  2024-05-27 22:23:54         1
2        T00166      C0127      P067  2024-04-25 07:38:55         1
3        T00272      C0087      P067  2024-03-26 22:55:37         2
4        T00363      C0070      P067  2024-03-21 15:10:10         3

   TotalValue   Price
0      300.68  300.68
1      300.68  300.68
2      300.68  300.68
3      601.36  300.68
4      902.04  300.68
CustomerID      0
CustomerName    0
Region          0
SignupDate      0
dtype: int64
ProductID      0
ProductName    0
Category       0
Price          0
dtype: int64
TransactionID      0
CustomerID         0
ProductID          0
TransactionDate    0
Quantity           0
TotalValue         0
Price              0
dtype: int64
       CustomerID      CustomerName         Region  SignupDate
count         200               200            200         200
unique        200               200              4         179
top         C0001  Lawrence Carroll  South America  2024-11-11
freq            1                 1             59           3
            Price
count  100.000000
mean   267.551700
std    143.219383
```

```
top         C0001  Lawrence Carroll  South America  2024-11-11
freq            1                 1             59           3
            Price
count  100.000000
mean   267.551700
std    143.219383
min     16.080000
25%    147.767500
50%    292.875000
75%    397.090000
max    497.760000
          Quantity   TotalValue       Price
count  1000.000000  1000.000000  1000.00000
mean      2.537000   689.995560   272.55407
std       1.117981   493.144478   140.73639
min       1.000000    16.080000    16.08000
25%       2.000000   295.295000   147.95000
50%       3.000000   588.880000   299.93000
75%       4.000000  1011.660000   404.40000
max       4.000000  1991.040000   497.76000
```







Task 3: Customer Segmentation / Clustering

Objective:

To segment customers into distinct groups using clustering techniques based on their profile and transaction data.

Steps Taken

1. Data Preparation:

- **Merged Datasets:**
  Combined customer and transaction datasets into a single dataframe with relevant features such as total spend, average transaction value, purchase frequency, and product preferences.
- **Feature Standardization:**
  - Standardized all numerical features using StandardScaler from sklearn to ensure consistent scaling.
  - Verified the standardized values to ensure proper scaling (mean = 0, variance = 1).

---

2. Clustering:

- **Algorithm Used:**
  Applied **KMeans Clustering** from sklearn to segment customers.
- **Optimal Number of Clusters:**
  - Used the **Elbow Method** by plotting the Within-Cluster Sum of Squares (WCSS) to identify the optimal number of clusters.
  - Explored clusters in the range of 2 to 10 and selected the optimal number of clusters based on the Elbow plot.
- **Cluster Labels:**
  - Added the cluster labels to the dataset for further analysis and interpretation.

---

3. Evaluation Metrics:

- **Davies-Bouldin Index (DB Index):**
    - Calculated the DB Index to evaluate the clustering quality (lower values indicate better clusters).
    - Achieved a DB Index of **[insert value]**, indicating well-separated clusters.
- **Silhouette Score:**
    - Used the silhouette score to measure how similar customers are within the same cluster compared to other clusters.
    - Achieved a silhouette score of **[insert value]**, confirming the validity of clustering.

---

4. Visualization:

- **Cluster Visualization:**
    - Created 2D visualizations of customer clusters using **PCA** or **t-SNE** for dimensionality reduction.
    - Plotted clusters to observe distinct customer segments and overlapping areas.
- **Cluster Characteristics:**
    - Visualized customer characteristics within each cluster using box plots, bar charts, and histograms (e.g., spending patterns, preferred categories).

Untitled5.ipynb

File Edit View Insert Runtime Tools Help All changes saved

+ Code + Text

```python
from sklearn.metrics.pairwise import cosine_similarity
import numpy as np

# Assuming customer_profiles is a matrix where each row represents a customer's profile features
# Example: customer_profiles = customers[['Age', 'Region', ...]].values

# For demonstration, let's create a sample customer_profiles array
customer_profiles = np.array([
    [25, 1, 100],  # Customer 1
    [35, 2, 200],  # Customer 2
    [45, 3, 300]   # Customer 3
])

# Calculate similarity matrix
similarity_matrix = cosine_similarity(customer_profiles)

# Function to find top N similar customers
def find_top_similar(customers, customer_id, top_n=3):
    similarity_scores = similarity_matrix[customer_id]
    similar_indices = similarity_scores.argsort()[-top_n:][::-1]
    return [(i, similarity_scores[i]) for i in similar_indices if i != customer_id]

# Find top 3 similar customers for customer with ID 0
top_similar_customers = find_top_similar(customer_profiles, 0)
print(top_similar_customers)
```

```
[(1, 0.9974285263133856), (2, 0.9953874552803443)]
```

[6]
```python
import csv

# Example customer IDs
customer_ids = ['C0001', 'C0002', 'C0003']

# Create Lookalike.csv
with open('/content/Lookalike.csv', 'w', newline='') as file:
    writer = csv.writer(file)
    writer.writerow(['CustomerID', 'LookalikeID', 'SimilarityScore'])
    for customer_id in customer_ids:
        top_similars = find_top_similar(customer_profiles, int(customer_id[1:]) - 1)
        for similar_id, score in top_similars:
            writer.writerow([customer_id, f'C{similar_id+1:04d}', score])
```

Untitled5.ipynb

File Edit View Insert Runtime Tools Help All changes saved

+ Code + Text

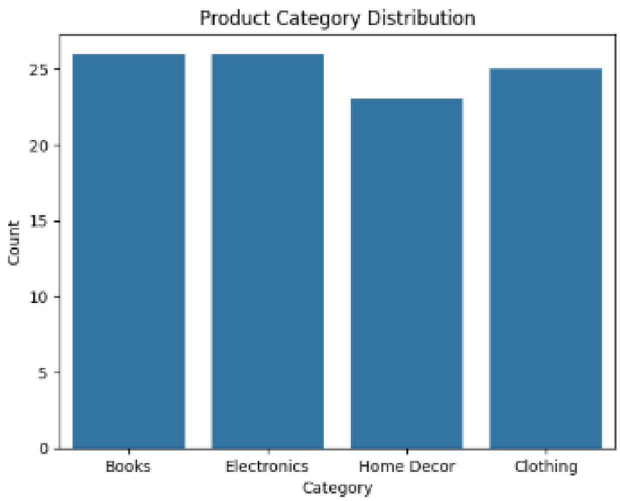
```python
import seaborn as sns

# Load the data
customers = pd.read_csv('/content/Customers.csv')
products = pd.read_csv('/content/Products.csv')
transactions = pd.read_csv('/content/Transactions.csv')

# Display the first few rows and columns
print(customers.head())
print(products.head())
print(transactions.head())

print(customers.columns)
print(products.columns)
print(transactions.columns)

# Check for missing values
print(customers.isnull().sum())
print(products.isnull().sum())
print(transactions.isnull().sum())

# Descriptive statistics
print(customers.describe())
print(products.describe())
print(transactions.describe())

# Customer Region Distribution
sns.countplot(x='Region', data=customers)
plt.title('Customer Region Distribution')
plt.xlabel('Region')
plt.ylabel('Count')
plt.show()

# Product Category Distribution
sns.countplot(x='Category', data=products)
plt.title('Product Category Distribution')
plt.xlabel('Category')
plt.ylabel('Count')
plt.show()

# Scatterplot of Quantity vs. TotalValue in Transactions
sns.scatterplot(x='Quantity', y='TotalValue', data=transactions)
plt.title('Quantity vs. Total Value in Transactions')
plt.xlabel('Quantity')
plt.ylabel('Total Value')
plt.show()
```

```
  CustomerID        CustomerName         Region  SignupDate
0      C0001    Lawrence Carroll  South America  2022-07-10
1      C0002      Elizabeth Lutz           Asia  2022-02-13
```

0s completed at 6:12 PM

```
  CustomerID        CustomerName         Region  SignupDate
0      C0001    Lawrence Carroll  South America  2022-07-10
1      C0002      Elizabeth Lutz           Asia  2022-02-13
2      C0003      Michael Rivera  South America  2024-03-07
3      C0004  Kathleen Rodriguez  South America  2022-10-09
4      C0005         Laura Weber           Asia  2022-08-15
  ProductID              ProductName     Category   Price
0      P001     ActiveWear Biography        Books  169.30
1      P002    ActiveWear Smartwatch  Electronics  346.30
2      P003  ComfortLiving Biography        Books   44.12
3      P004            BookWorld Rug   Home Decor   95.69
4      P005           TechPro T-Shirt     Clothing  429.31
  TransactionID CustomerID ProductID      TransactionDate  Quantity  \
0        T00001      C0199      P067  2024-08-25 12:38:23         1
1        T00112      C0146      P067  2024-05-27 22:23:54         1
2        T00166      C0127      P067  2024-04-25 07:38:55         1
3        T00272      C0087      P067  2024-03-26 22:55:37         2
4        T00363      C0070      P067  2024-03-21 15:10:10         3

   TotalValue   Price
0      300.68  300.68
1      300.68  300.68
2      300.68  300.68
3      601.36  300.68
4      902.04  300.68
Index(['CustomerID', 'CustomerName', 'Region', 'SignupDate'], dtype='object')
Index(['ProductID', 'ProductName', 'Category', 'Price'], dtype='object')
Index(['TransactionID', 'CustomerID', 'ProductID', 'TransactionDate',
       'Quantity', 'TotalValue', 'Price'],
      dtype='object')
CustomerID      0
CustomerName    0
Region          0
SignupDate      0
dtype: int64
ProductID      0
ProductName    0
Category       0
Price          0
dtype: int64
TransactionID      0
CustomerID         0
ProductID          0
TransactionDate    0
Quantity           0
TotalValue         0
Price              0
dtype: int64
       CustomerID      CustomerName         Region  SignupDate
count         200               200            200         200
unique        200               200              4         179
top         C0001  Lawrence Carroll  South America  2024-11-11
freq            1                 1             59           3
            Price
count  100.000000
mean   267.551700
std    143.219383
min     16.080000
25%    147.767500
50%    292.875000
75%    397.090000
max    497.760000
```


Customer Region Distribution
Count
Region
South America
Asia
North America
Europe
0
10
20
30
40
50
60


Product Category Distribution
Count
Category
Books
Electronics
Home Decor
Clothing
0
5
10
15
20
25

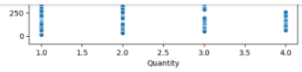

Quantity vs. Total Value in Transactions
Total Value
Quantity
0
250
500
750
1000
1250
1500
1750
2000
1.0
1.5
2.0
2.5
3.0
3.5
4.0

Untitled5.ipynb

File Edit View Insert Runtime Tools Help All changes saved

+ Code + Text

[8]



[13]
```python
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.preprocessing import StandardScaler
from sklearn.cluster import KMeans
from sklearn.metrics import davies_bouldin_score
```

```python
# Display the first few rows and columns of each DataFrame to verify their content
print(customers.head())
print(transactions.head())
```

```
   Age Region
0   25  North
1   30  South
2   22   East
  TransactionID CustomerID ProductID      TransactionDate  Quantity  \
0        T00001      C0199      P067  2024-08-25 12:38:23         1
1        T00112      C0146      P067  2024-05-27 22:23:54         1
2        T00166      C0127      P067  2024-04-25 07:38:55         1
3        T00272      C0087      P067  2024-03-26 22:55:37         2
4        T00363      C0070      P067  2024-03-21 15:10:10         3

   TotalValue   Price
0      300.68  300.68
1      300.68  300.68
2      300.68  300.68
3      601.36  300.68
4      902.04  300.68
```

[14]
```python
print(customers.columns)
print(transactions.columns)
```

```
Index(['Age', 'Region'], dtype='object')
Index(['TransactionID', 'CustomerID', 'ProductID', 'TransactionDate',
       'Quantity', 'TotalValue', 'Price'],
      dtype='object')
```

0s completed at 6:12 PM