Python programming Lab(23CP301P)

Name: Krishika Vansh

Semester: V

Roll No: 23BCP448

Faculty: Mr. Davinder Singh

Division: VII Batch G13

Branch: Computer Engineering

School of Technology

November 2025

Experiment No: 9

Basic Data Analysis using pandas and matplotlib

Objective: To analyze customer transaction data and segment customers based on their shopping behavior using data analysis and clustering techniques in Python.

Install pip install numpy pandas matplotlib scikit-learn

Code:

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
from sklearn.cluster import KMeans
from sklearn.preprocessing import StandardScaler

# 1. Load dataset
data = pd.read_csv("customerTransactions.csv")
print("Dataset loaded successfully!")
print(data.head())

# 2. Data Cleaning
print("\nMissing values before cleaning:")
print(data.isnull().sum())

data.dropna(subset=["Customer ID"], inplace=True)
```

```
data.drop_duplicates(inplace=True)


print("\nData cleaned successfully!")
print(f"Total records after cleaning: {len(data)}")


# 3. Descriptive Statistics
print("\nDescriptive Statistics:")
print(data[["Total Amount Spent", "Total Items Purchased"]].describe())


# 4. Clustering Preparation
X = data[["Total Amount Spent", "Total Items Purchased", "Average Purchase Value"]]
scaler = StandardScaler()
X_scaled = scaler.fit_transform(X)


# 5. Apply K-Means
kmeans = KMeans(n_clusters=3, random_state=42)
data["Cluster"] = kmeans.fit_predict(X_scaled)


# 6. Visualization
plt.figure(figsize=(8, 6))
plt.scatter(data["Total Amount Spent"], data["Total Items Purchased"],
        c=data["Cluster"], cmap="viridis", s=100, edgecolors='k')
plt.title("Customer Segmentation based on Spending and Purchase Behavior")
plt.xlabel("Total Amount Spent")
```

```
plt.ylabel("Total Items Purchased")

plt.colorbar(label="Cluster")

plt.show()


# 7. Segment Summary

print("\nCustomer Segment Summary:")

segment_summary = data.groupby("Cluster")[["Total Amount Spent", "Total Items Purchased", "Average Purchase Value"]].mean()

print(segment_summary)


# 8. Assign segment labels

cluster_labels = {

   0: "Low-Value Customers (Inactive/Occasional Shoppers)",

   1: "Mid-Tier Customers (Moderate Shoppers)",

   2: "High-Value Customers (Frequent & High Spenders)"

}

data["Segment"] = data["Cluster"].map(cluster_labels)


# 9. Insights & Recommendations

print("\nCustomer Engagement Recommendations:")

print("""

 High-Value Customers:

   - Offer loyalty rewards, exclusive previews, or premium deals.

   - Personalized product recommendations.
```

Mid-Tier Customers:

- Targeted marketing emails with discounts to boost spending.
- Encourage subscription or bundle offers.

Low-Value/Inactive Customers:

- Send reactivation offers, reminders, and special limited-time discounts.
- Analyze their drop-off reasons (delivery time, pricing, etc.)

""")

print("\nSegmentation complete!")

Output:

```
Dataset loaded successfully!
   Customer ID  Total Amount Spent  Total Items Purchased Last Purchase Date  Average Purchase Value
0        10001                2500                     50         2025-09-20                    50.0
1        10002               12000                    200         2025-09-18                    60.0
2        10003                 800                     20         2025-08-25                    40.0
3        10004                3500                     70         2025-09-30                    50.0
4        10005                 500                     10         2025-07-10                    50.0

Missing values before cleaning:
Customer ID              0
Total Amount Spent       0
Total Items Purchased    0
Last Purchase Date       0
Average Purchase Value   0
dtype: int64

Data cleaned successfully!
Total records after cleaning: 20

Descriptive Statistics:
       Total Amount Spent  Total Items Purchased
count           20.000000              20.000000
mean          4895.000000              89.400000
std           5670.489165              97.093769
min            250.000000               5.000000
25%            912.500000              20.000000
50%           2300.000000              45.000000
75%           7625.000000             142.500000
max          20000.000000             350.000000
```

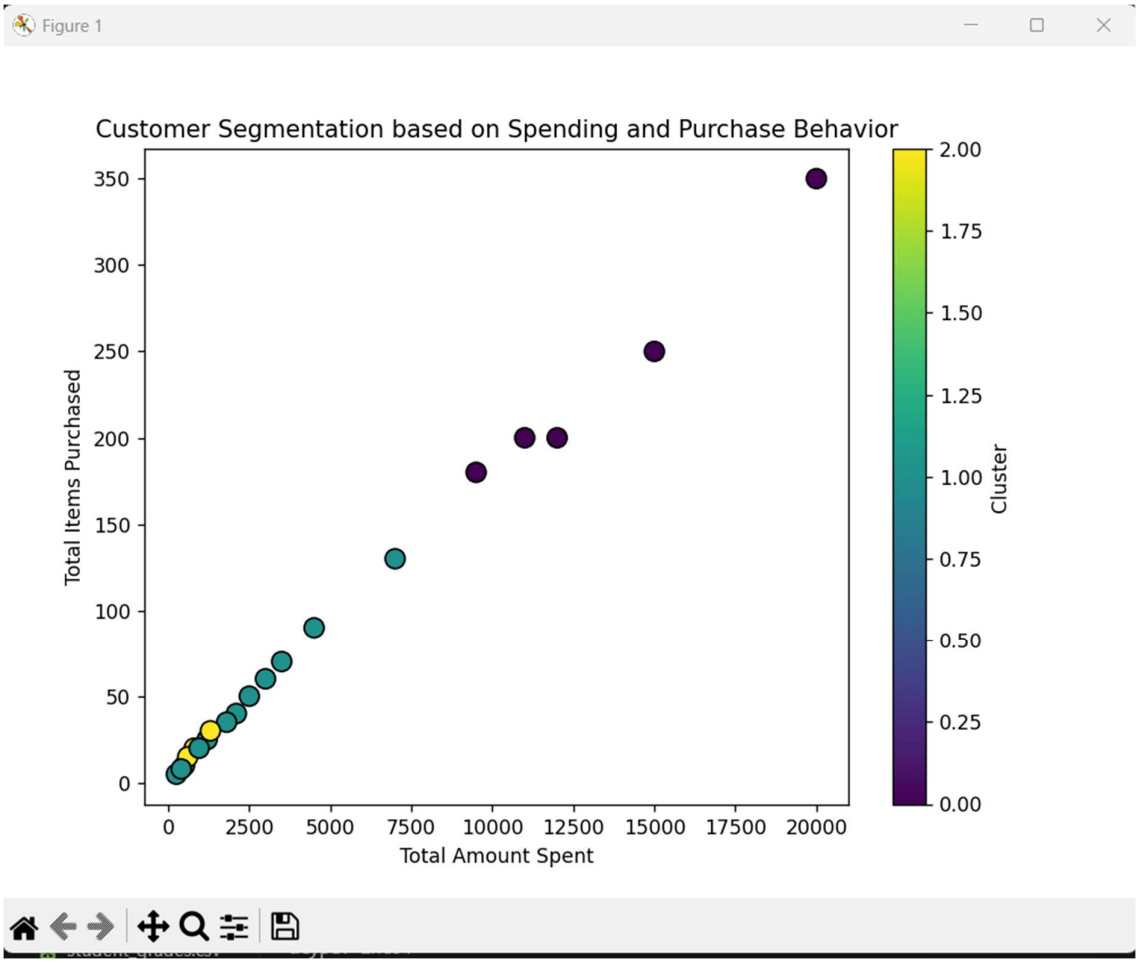
Figure 1
Customer Segmentation based on Spending and Purchase Behavior
Total Items Purchased
Total Amount Spent
Cluster
0
50
100
150
200
250
300
350
0
2500
5000
7500
10000
12500
15000
17500
20000
0.00
0.25
0.50
0.75
1.00
1.25
1.50
1.75
2.00

Experiment No: 10

Reading, Creating, and Modifying PDF Files in Python

Objective: To analyze customer transaction data and segment customers based on their shopping behavior using data analysis and clustering techniques in Python.

pip install pandas numpy matplotlib scikit-learn reportlab PyPDF2

Code:

```
import pandas as pd

from sklearn.cluster import KMeans

import matplotlib.pyplot as plt

from reportlab.lib.pagesizes import letter

from reportlab.pdfgen import canvas

from PyPDF2 import PdfMerger

from datetime import datetime

import os


print("=== CUSTOMER SEGMENTATION ===")


# Load customer data

customer_data = pd.read_csv("customers.csv")

print("Customer Data Loaded Successfully!\n")

print(customer_data.head())


X = customer_data[["Total Amount Spent", "Total Items Purchased",
"Average Purchase Value"]]
```

```
# Apply K-Means clustering
kmeans = KMeans(n_clusters=3, random_state=42)
customer_data["Cluster"] = kmeans.fit_predict(X)

print("\nCustomer Clustering Completed. Cluster Centers:")
print(kmeans.cluster_centers_)

# Visualize clusters
plt.figure(figsize=(8, 5))
plt.scatter(customer_data["Total Amount Spent"], customer_data["Total Items Purchased"],
        c=customer_data["Cluster"], cmap='viridis', s=100)
plt.title("Customer Segments Based on Spending and Purchase Frequency")
plt.xlabel("Total Amount Spent")
plt.ylabel("Total Items Purchased")
plt.grid(True)
plt.show()

# Save clustered data
customer_data.to_csv("customer_segments.csv", index=False)
print("\nClustered customer data saved to 'customer_segments.csv'\n")


print("=== INVOICE GENERATION ===")
```

```
# Load order data
orders = pd.read_csv("orders.csv")
print("Order Data Loaded Successfully!\n")
print(orders.head())

# Create directory for invoices
if not os.path.exists("invoices"):
    os.makedirs("invoices")

# Generate PDF invoices for each order
for _, order in orders.iterrows():
    order_id = str(order["Order ID"])
    customer_name = order["Customer Name"]
    product_name = order["Product Name"]
    quantity = order["Quantity"]
    unit_price = order["Unit Price"]
    total_amount = quantity * unit_price
    date = datetime.now().strftime("%Y-%m-%d")

    filename = f"invoices/{order_id}.pdf"
    c = canvas.Canvas(filename, pagesize=letter)

    c.setFont("Helvetica-Bold", 16)
    c.drawString(200, 750, "INVOICE")
```

```
    c.setFont("Helvetica", 12)
    c.drawString(50, 700, f"Invoice Number: {order_id}")
    c.drawString(50, 680, f"Date of Purchase: {date}")
    c.drawString(50, 660, f"Customer Name: {customer_name}")
    c.drawString(50, 640, f"Product Name: {product_name}")
    c.drawString(50, 620, f"Quantity: {quantity}")
    c.drawString(50, 600, f"Unit Price: ₹{unit_price}")
    c.drawString(50, 580, f"Total Amount: ₹{total_amount:.2f}")

    c.showPage()
    c.save()
    print(f"Invoice generated for Order ID {order_id}")

# Merge all invoices into one PDF
merger = PdfMerger()
for file in sorted(os.listdir("invoices")):
    if file.endswith(".pdf"):
        merger.append(os.path.join("invoices", file))

merged_filename = "All_Invoices.pdf"
merger.write(merged_filename)
merger.close()

print(f"\nAll invoices merged into '{merged_filename}' successfully!")
```

Output:

```
   Customer ID  Total Amount Spent  Total Items Purchased Last Purchase Date  Average Purchase Value
0        10001                2500                     50         2025-09-20                    50.0
1        10002               12000                    200         2025-09-18                    60.0
2        10003                 800                     20         2025-08-25                    40.0
3        10004                3500                     70         2025-09-30                    50.0
4        10005                 500                     10         2025-07-10                    50.0

Customer Clustering Completed. Cluster Centers:
[[17500.           300.            58.57      ]
 [ 1671.42857143    34.14285714    48.05428571]
 [ 9875.           177.5           55.4075    ]]

Clustered customer data saved to 'customer_segments.csv'

=== INVOICE GENERATION ===
Order Data Loaded Successfully!

  Order ID   Customer Name    Product Name  Quantity  Unit Price
0   ORD001  Krishika Vansh  Wireless Mouse         2         500
1   ORD002     Aryan Patel      Laptop Bag         1        1200
2   ORD003     Neha Sharma        Keyboard         1         700
3   ORD004    Ananya Desai       USB Cable         3         150
4   ORD005       Raj Mehta         Monitor         1        8500
Invoice generated for Order ID ORD001
Invoice generated for Order ID ORD002
Invoice generated for Order ID ORD003
Invoice generated for Order ID ORD004
Invoice generated for Order ID ORD005

All invoices merged into 'All_Invoices.pdf' successfully!
```