Customer & Revenue Intelligence System

Abstract:

This project analyzes retail transactional data to generate business insights and predict customer purchasing behavior using SQL-style aggregation and Python analytics.

Introduction:

Understanding customer behavior is critical for revenue growth. This project transforms raw transaction data into actionable business intelligence.

Problem Statement:

1. Analyze total revenue.
2. Study monthly revenue trends.
3. Identify high-value customers.
4. Perform customer segmentation.
5. Predict future purchases.

Dataset Description:

The Online Retail dataset includes InvoiceNo, Quantity, UnitPrice, CustomerID, InvoiceDate, and Country.

Tools & Technologies:

Python, Pandas, NumPy, Matplotlib, Seaborn, Scikit-learn, Google Colab.

Data Preprocessing:

- Removed missing CustomerID values.
- Removed negative quantities.
- Converted InvoiceDate to datetime.

- Created TotalPrice column.

Exploratory Data Analysis:

Total revenue and monthly trends were analyzed to identify patterns and seasonality.

RFM Analysis:

Customers were segmented using Recency, Frequency, and Monetary values.

Clustering:

KMeans clustering grouped customers into meaningful behavioral segments.

Predictive Modeling:

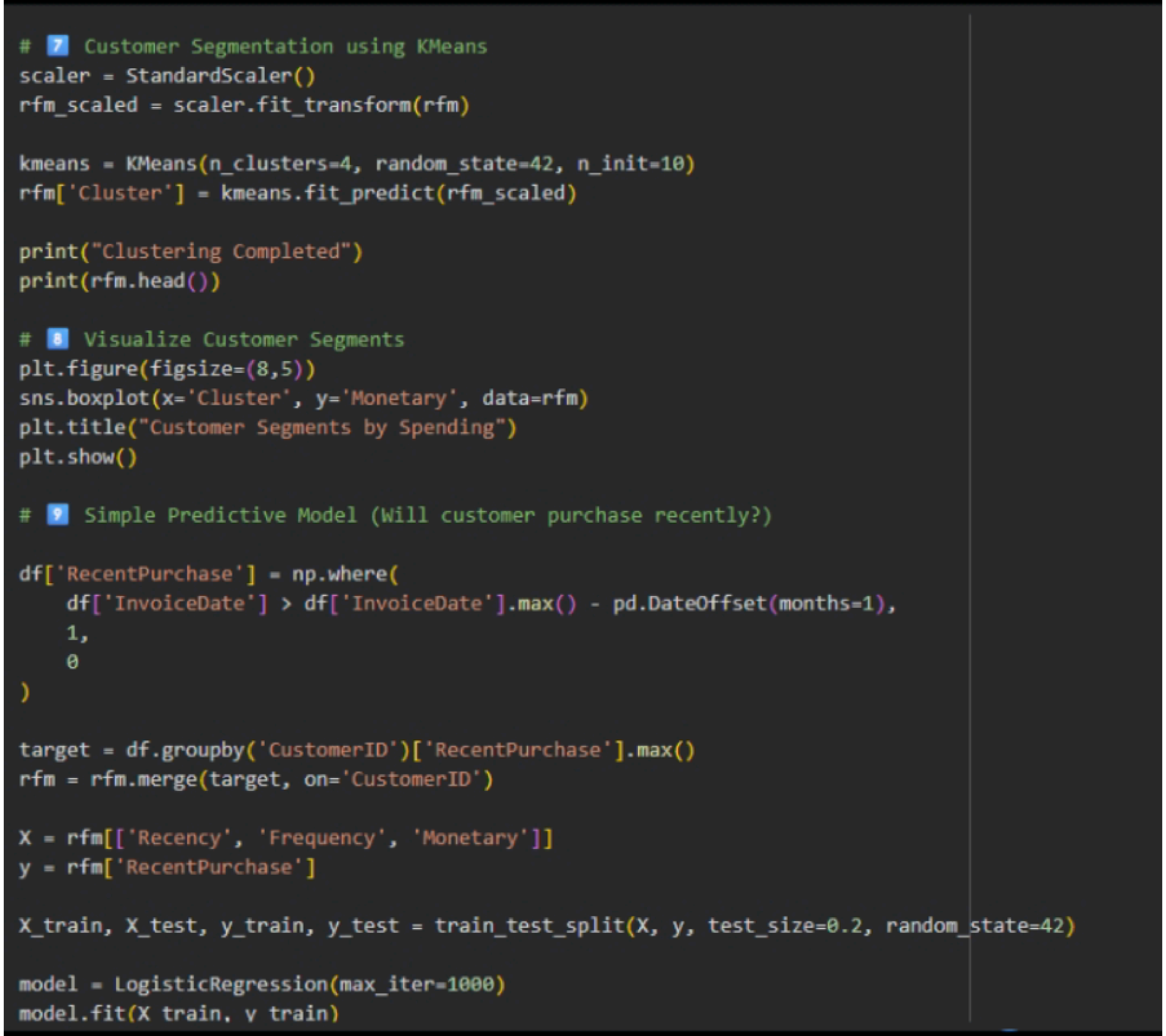
Logistic Regression predicted customer future purchases using RFM features.

Business Insights:

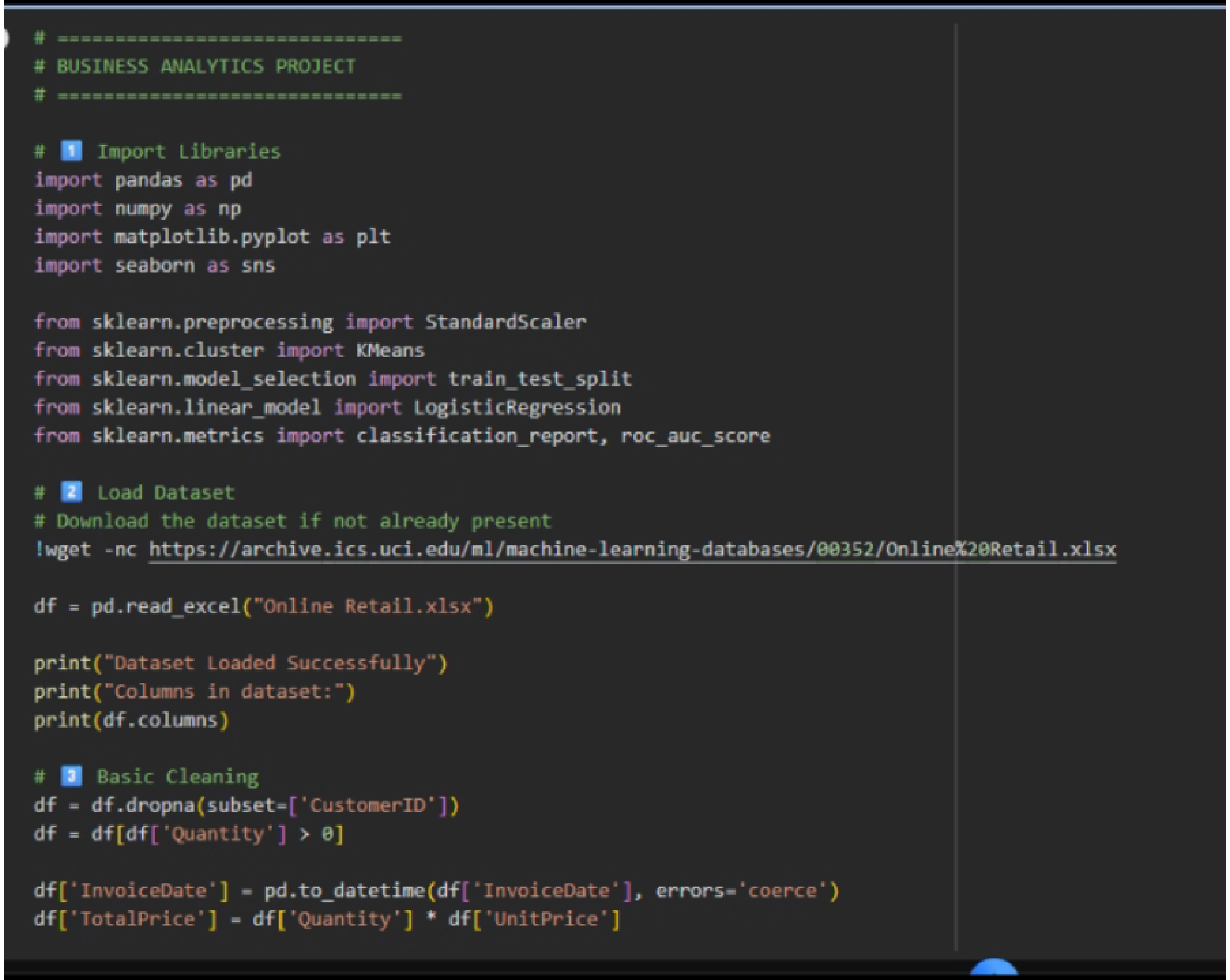
- High-value customers contribute major revenue.
- Recent customers are more likely to repurchase.
- Segmentation improves targeted marketing.

Conclusion:

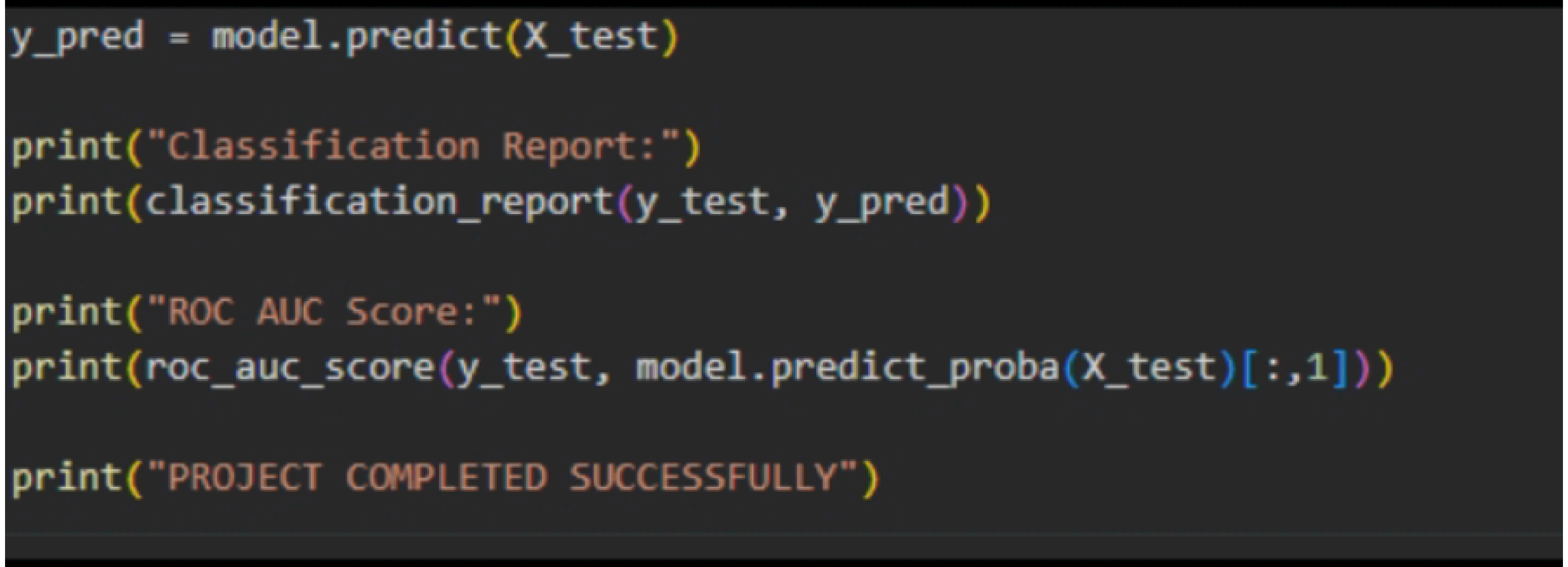
The project demonstrates a complete business analytics pipeline from data cleaning to predictive modeling and actionable insights.



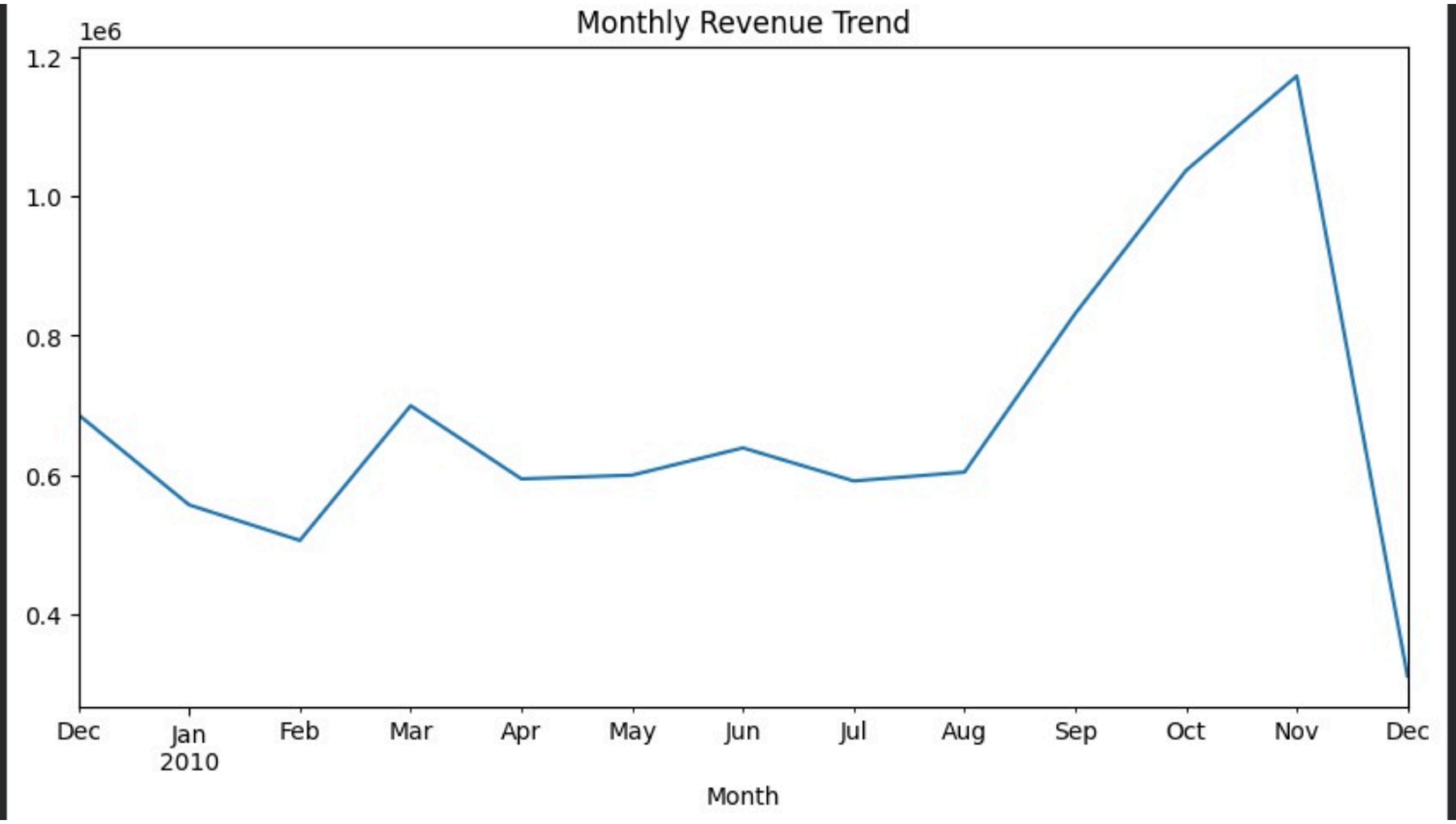
```
# 7️⃣ Customer Segmentation using KMeans
scaler = StandardScaler()
rfm_scaled = scaler.fit_transform(rfm)

kmeans = KMeans(n_clusters=4, random_state=42, n_init=10)
rfm['Cluster'] = kmeans.fit_predict(rfm_scaled)

print("Clustering Completed")
print(rfm.head())

# 8️⃣ Visualize Customer Segments
plt.figure(figsize=(8,5))
sns.boxplot(x='Cluster', y='Monetary', data=rfm)
plt.title("Customer Segments by Spending")
plt.show()

# 9️⃣ Simple Predictive Model (Will customer purchase recently?)

df['RecentPurchase'] = np.where(
    df['InvoiceDate'] > df['InvoiceDate'].max() - pd.DateOffset(months=1),
    1,
    0
)

target = df.groupby('CustomerID')['RecentPurchase'].max()
rfm = rfm.merge(target, on='CustomerID')

X = rfm[['Recency', 'Frequency', 'Monetary']]
y = rfm['RecentPurchase']

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

model = LogisticRegression(max_iter=1000)
model.fit(X_train, y_train)
```

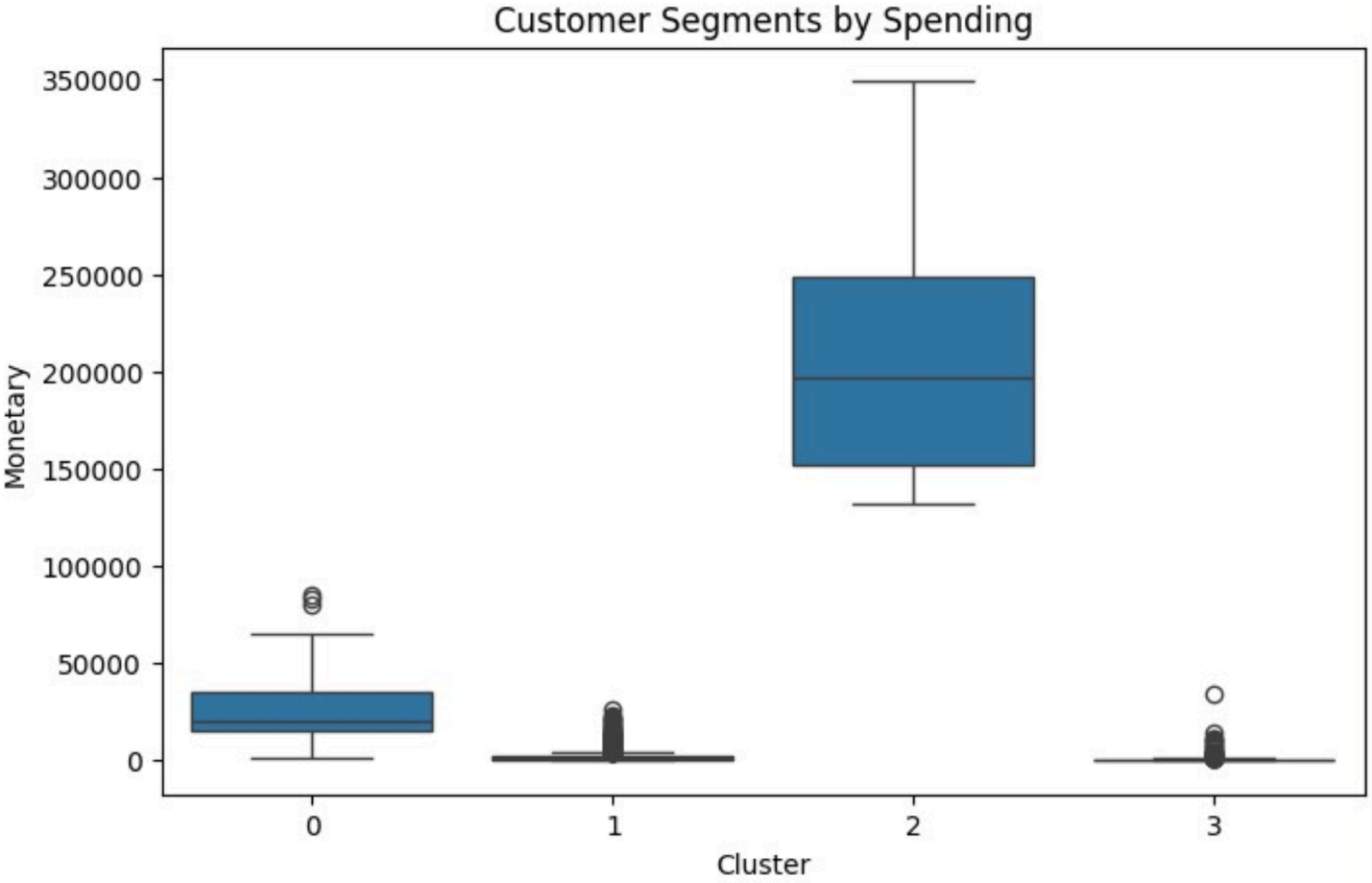


```
# ==============================
# BUSINESS ANALYTICS PROJECT
# ==============================

# 1️⃣ Import Libraries
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.model_selection import train_test_split
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import classification_report, roc_auc_score

# 2️⃣ Load Dataset
# Download the dataset if not already present
!wget -nc https://archive.ics.uci.edu/ml/machine-learning-databases/00352/Online%20Retail.xlsx

df = pd.read_excel("Online Retail.xlsx")

print("Dataset Loaded Successfully")
print("Columns in dataset:")
print(df.columns)

# 3️⃣ Basic Cleaning
df = df.dropna(subset=['CustomerID'])
df = df[df['Quantity'] > 0]

df['InvoiceDate'] = pd.to_datetime(df['InvoiceDate'], errors='coerce')
df['TotalPrice'] = df['Quantity'] * df['UnitPrice']
```



```python
y_pred = model.predict(X_test)

print("Classification Report:")
print(classification_report(y_test, y_pred))

print("ROC AUC Score:")
print(roc_auc_score(y_test, model.predict_proba(X_test)[:,1]))

print("PROJECT COMPLETED SUCCESSFULLY")
```


Monthly Revenue Trend
1e6
1.2
1.0
0.8
0.6
0.4
Dec
Jan
2010
Feb
Mar
Apr
May
Jun
Jul
Aug
Sep
Oct
Nov
Dec
Month


Customer Segments by Spending
Monetary
350000
300000
250000
200000
150000
100000
50000
0
0
1
2
3
Cluster