INTERNSHALA TRAININGS

Career Ki Guarantee

CAPSTONE PROJECT

Data Science

Web Scraping and Analysis of Online Book Data for Pricing, Ratings, and Market Segmentation

Tools Used:
Python, Pandas, BeautifulSoup, Power BI, Machine Learning (K-Means)

PROJECT OBJECTIVE

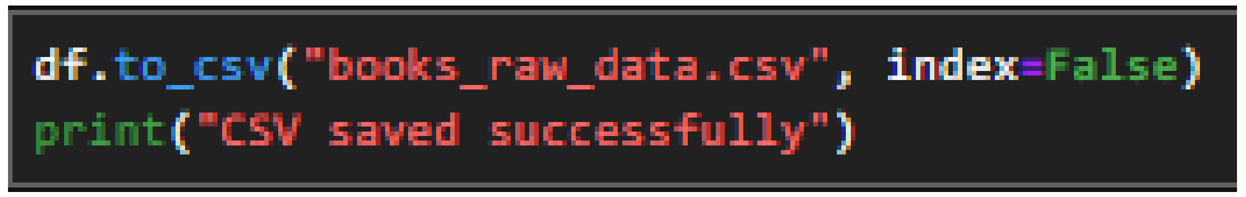
The objective of this project is to collect real-world online book data through web scraping and transform it into structured, decision-ready insights.
The project focuses on analyzing pricing patterns, customer ratings, and popularity indicators, and applying machine learning techniques to segment books into meaningful groups.
An interactive Power BI dashboard is created to communicate insights effectively and support data-driven decision-making.

INTERNSHALA TRAININGS
Career Ki Guarantee

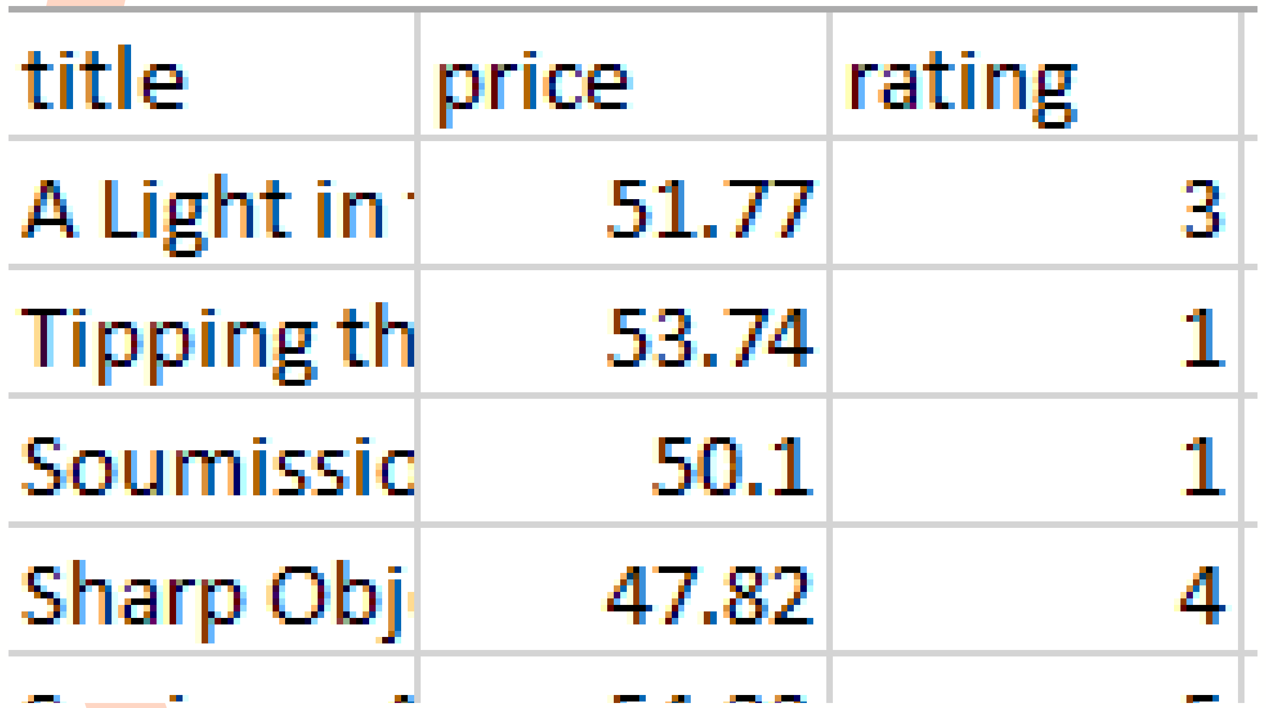
DATA COLLECTION (WEB SCRAPING)

- **Website name: Books to Scrape**
- **Technique: Python + BeautifulSoup**
- **Pages scraped: 50**
- **Records collected: ~1000 books**

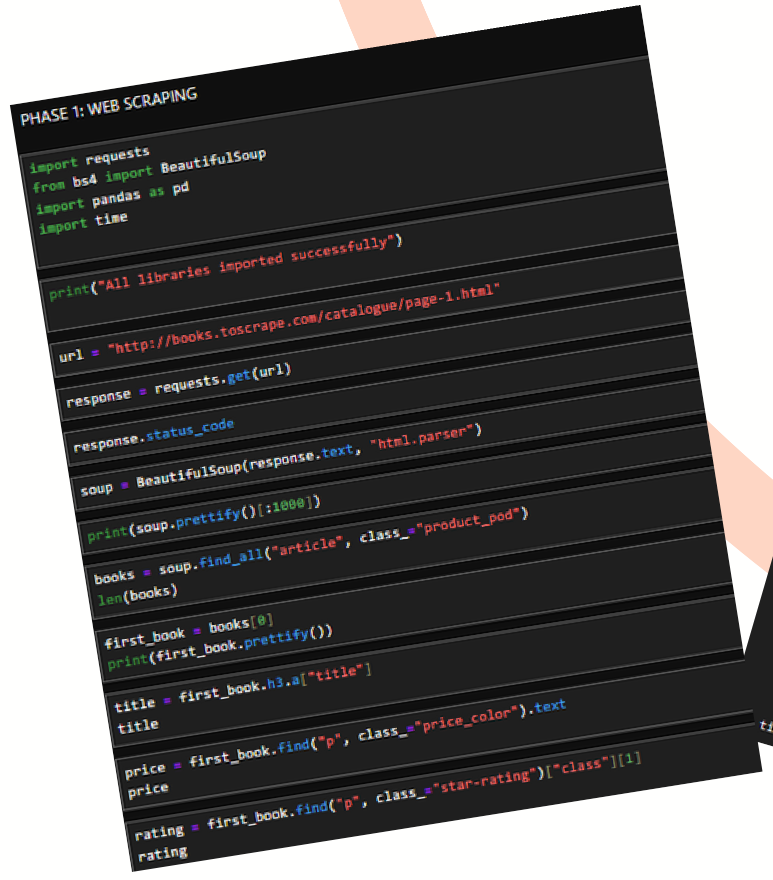
A reproducible Python script was developed to scrape book titles, prices, and ratings from a publicly available website. Data was collected across multiple pages to ensure sufficient volume for analysis.



```python
df.to_csv("books_raw_data.csv", index=False)
print("CSV saved successfully")
```

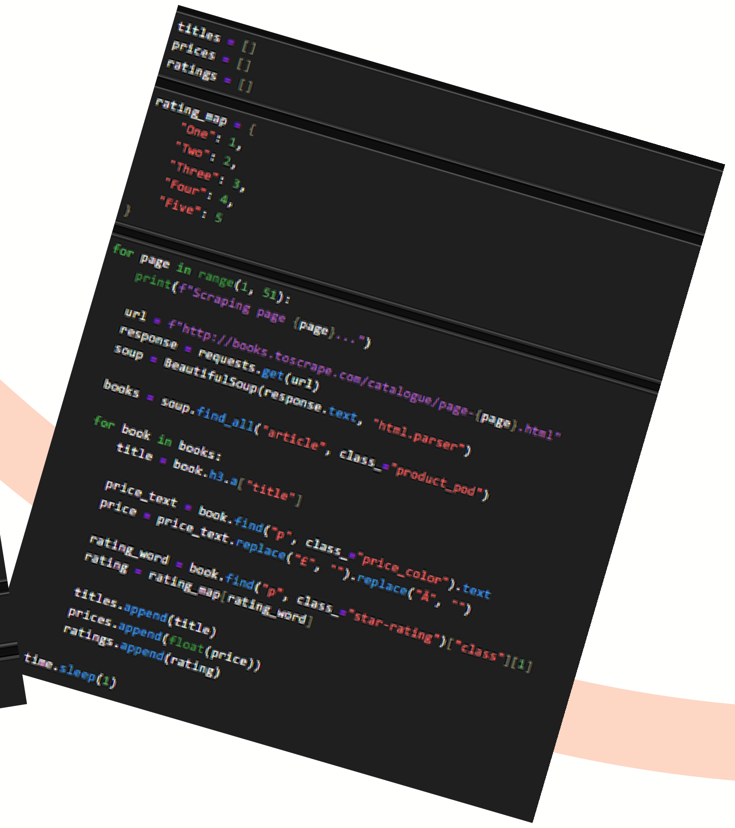


title	price	rating
A Light in	51.77	3
Tipping th	53.74	1
Soumissic	50.1	1
Sharp Obj	47.82	4



PHASE 1: WEB SCRAPING

```python
import requests
from bs4 import BeautifulSoup
import pandas as pd
import time

print("All libraries imported successfully")

url = "http://books.toscrape.com/catalogue/page-1.html"

response = requests.get(url)

response.status_code

soup = BeautifulSoup(response.text, "html.parser")

print(soup.prettify()[:1000])

books = soup.find_all("article", class_="product_pod")
len(books)

first_book = books[0]
print(first_book.prettify())

title = first_book.h3.a["title"]
title

price = first_book.find("p", class_="price_color").text
price

rating = first_book.find("p", class_="star-rating")["class"][1]
rating
```



```python
titles = []
prices = []
ratings = []

rating_map = {
    "One": 1,
    "Two": 2,
    "Three": 3,
    "Four": 4,
    "Five": 5
}

for page in range(1, 51):
    print(f"Scraping page {page}...")

    url = f"http://books.toscrape.com/catalogue/page-{page}.html"
    response = requests.get(url)
    soup = BeautifulSoup(response.text, "html.parser")

    books = soup.find_all("article", class_="product_pod")

    for book in books:
        title = book.h3.a["title"]

        price_text = book.find("p", class_="price_color").text
        price = price_text.replace("£", "").replace("Â", "")

        rating_word = book.find("p", class_="star-rating")["class"][1]
        rating = rating_map[rating_word]

        titles.append(title)
        prices.append(float(price))
        ratings.append(rating)

    time.sleep(1)
```

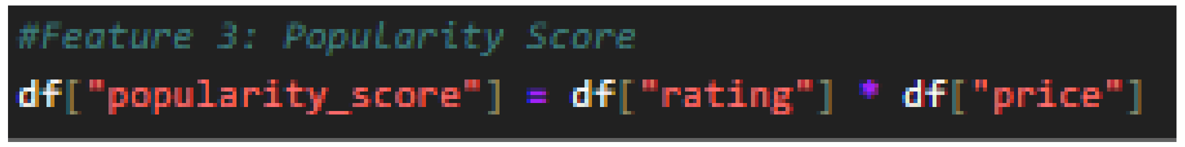
DATA CLEANING & PREPROCESSING

- Duplicate removal
- Data type correction
- Encoding issues handled
- Feature engineering

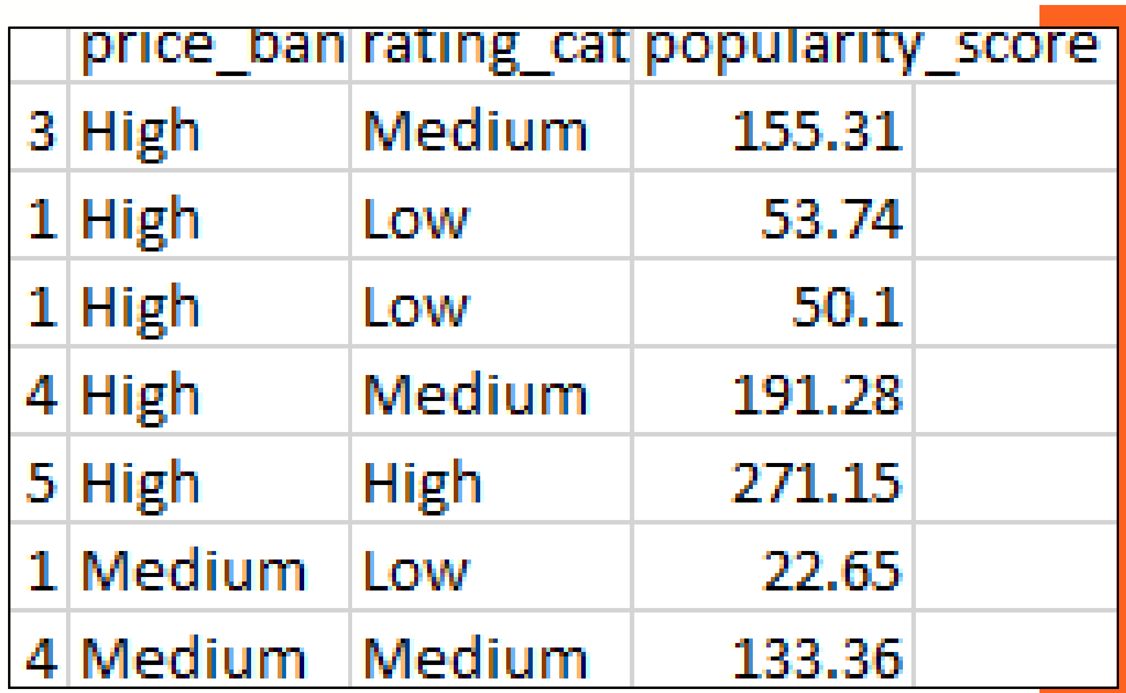
During data extraction, encoding-related inconsistencies were observed due to currency symbols. These were handled through preprocessing and exception handling to ensure data integrity.

CREATED FEATURE:

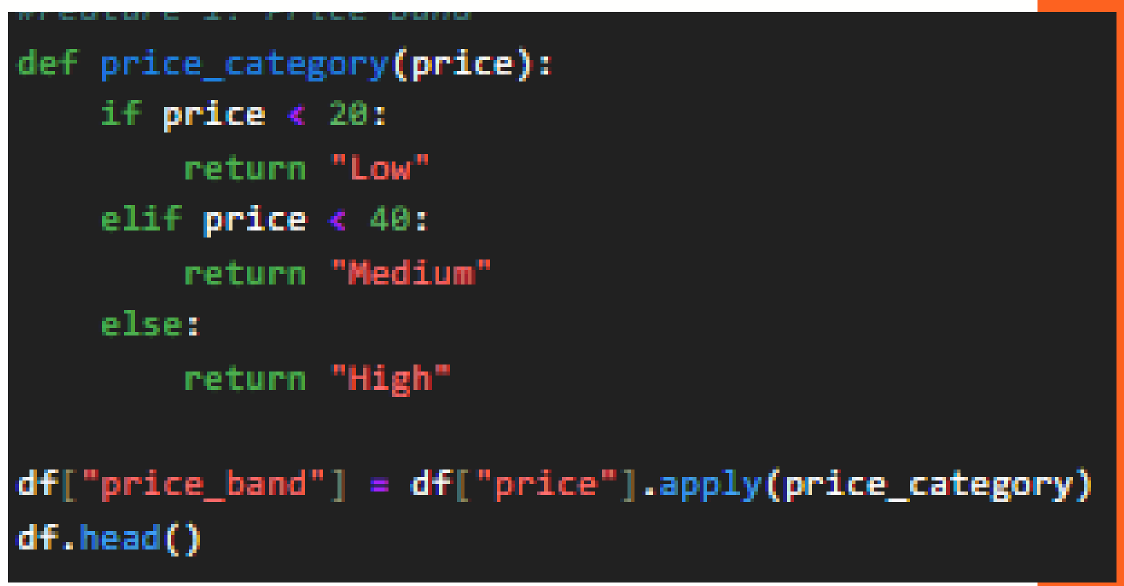
- **price_band**
- **rating_category**
- **popularity_score**



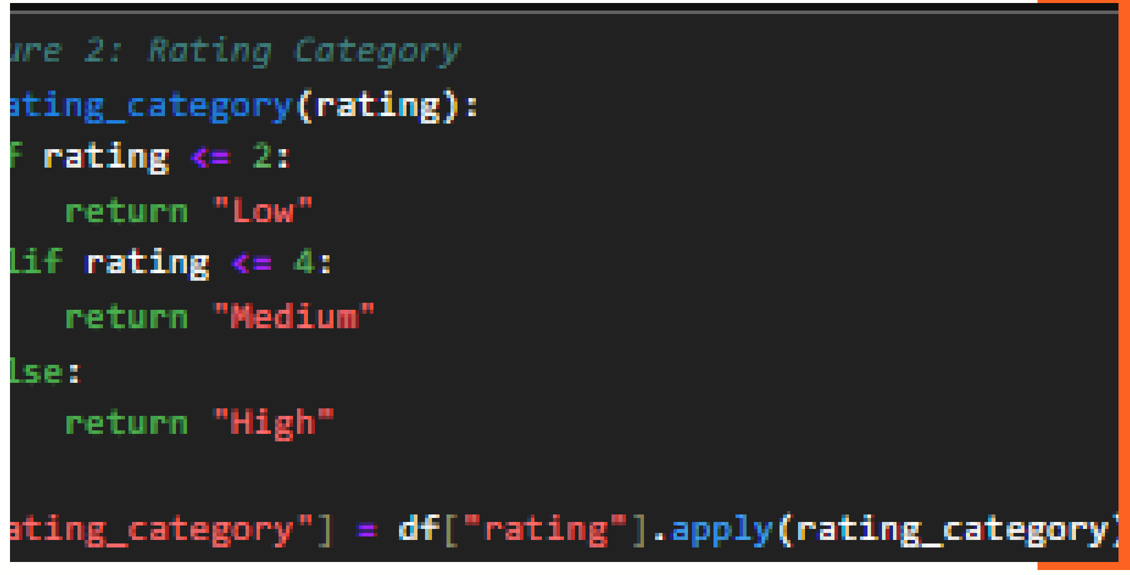
```
#Feature 3: Popularity Score
df["popularity_score"] = df["rating"] * df["price"]
```



	price_ban	rating_cat	popularity_score
3	High	Medium	155.31
1	High	Low	53.74
1	High	Low	50.1
4	High	Medium	191.28
5	High	High	271.15
1	Medium	Low	22.65
4	Medium	Medium	133.36

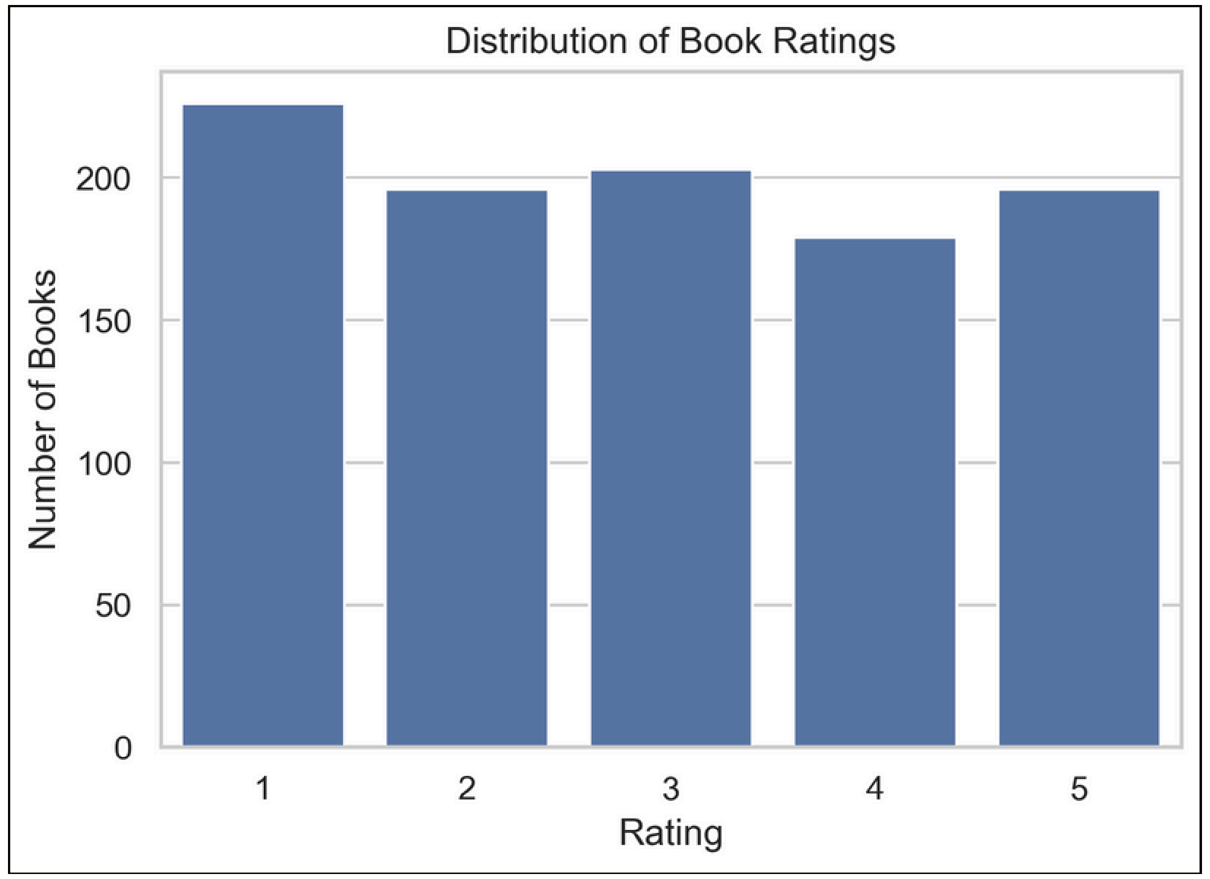


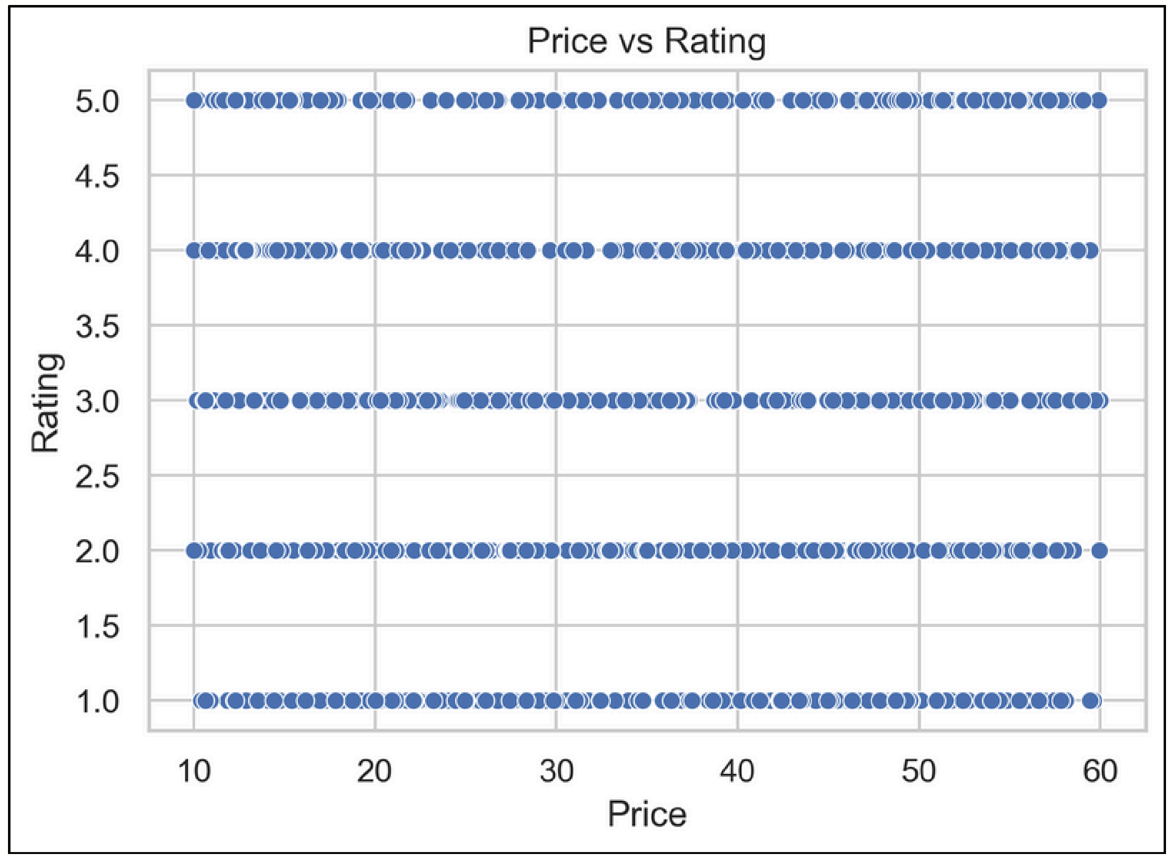
```
def price_category(price):
    if price < 20:
        return "Low"
    elif price < 40:
        return "Medium"
    else:
        return "High"

df["price_band"] = df["price"].apply(price_category)
df.head()
```

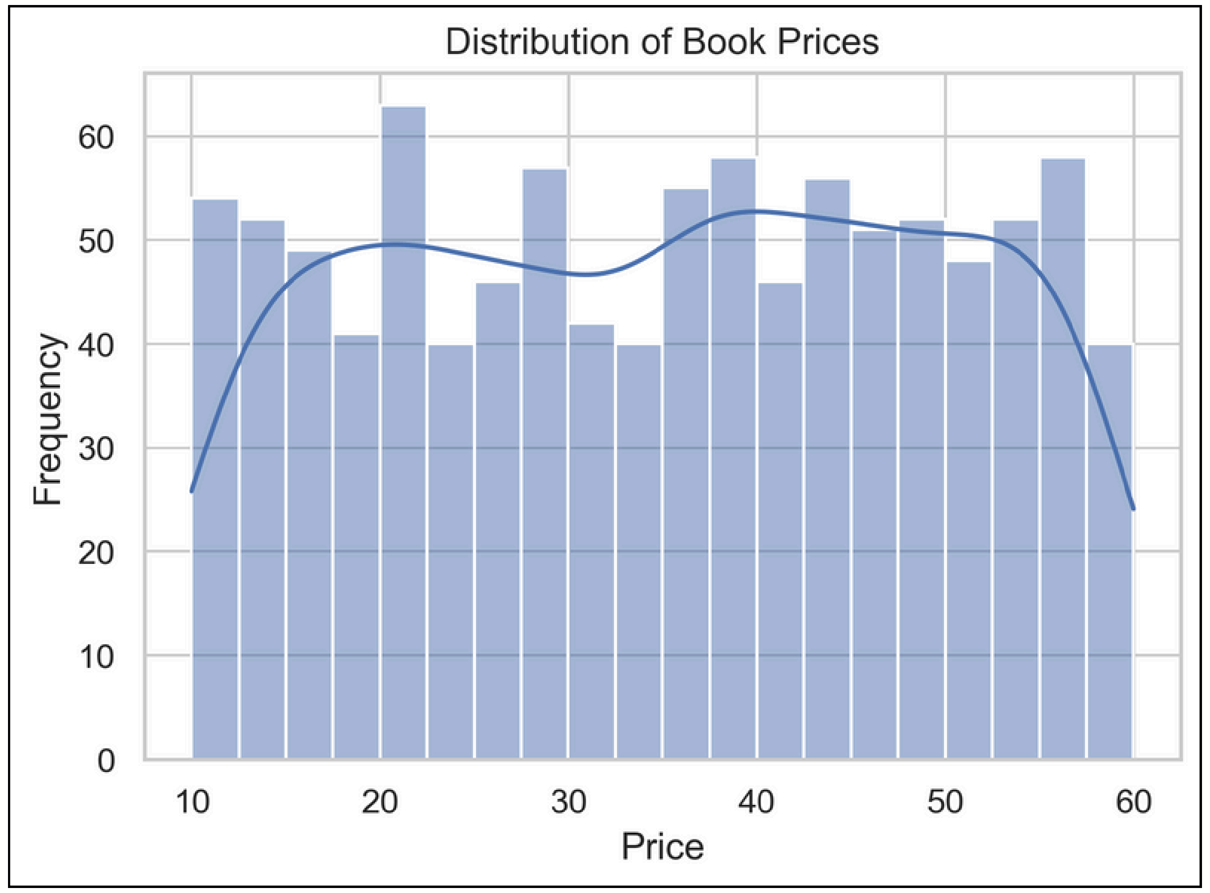


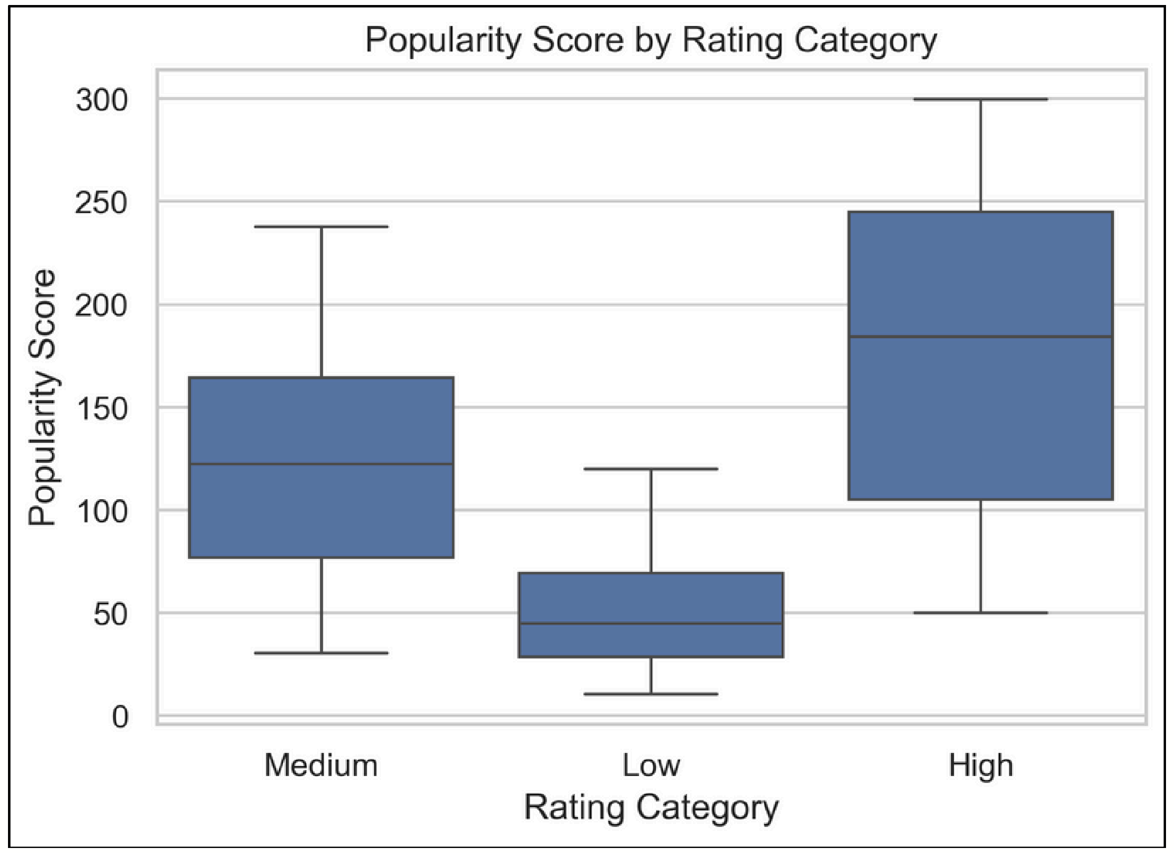
```
ure 2: Rating Category
ating_category(rating):
f rating <= 2:
   return "Low"
lif rating <= 4:
   return "Medium"
lse:
   return "High"

ating_category"] = df["rating"].apply(rating_category)
```

Exploratory Data Analysis & Visualization

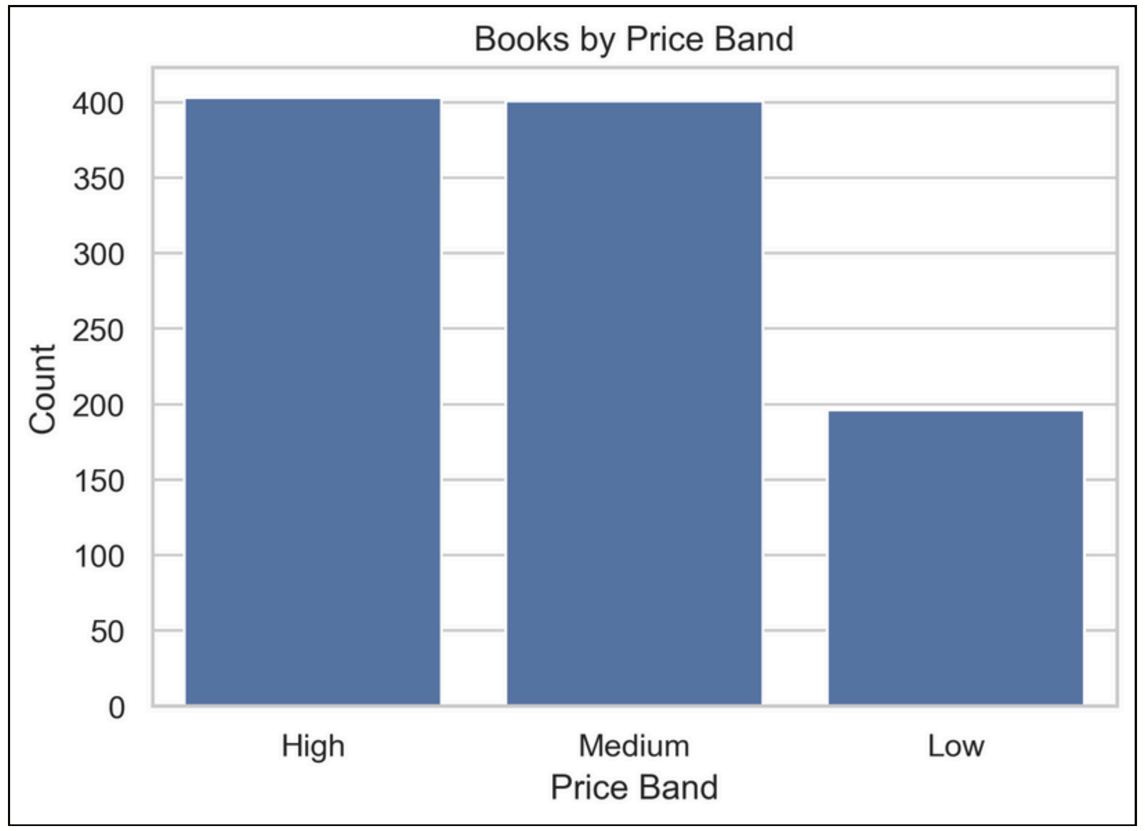


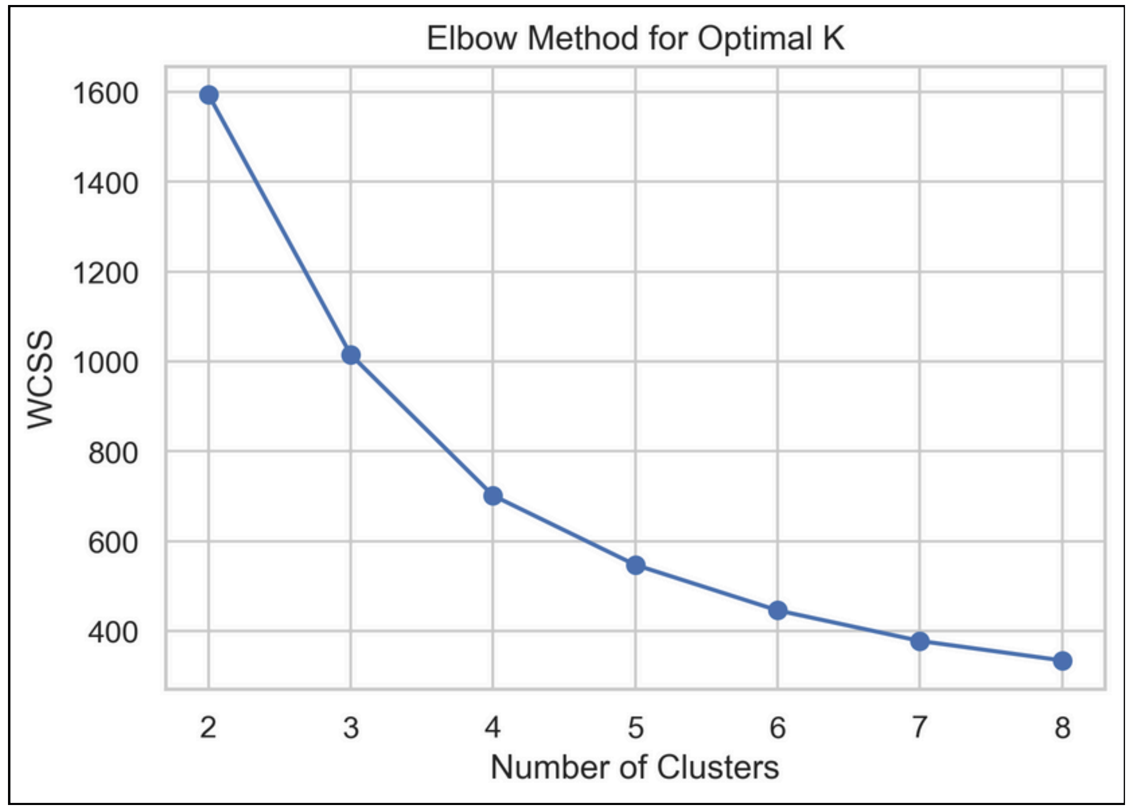


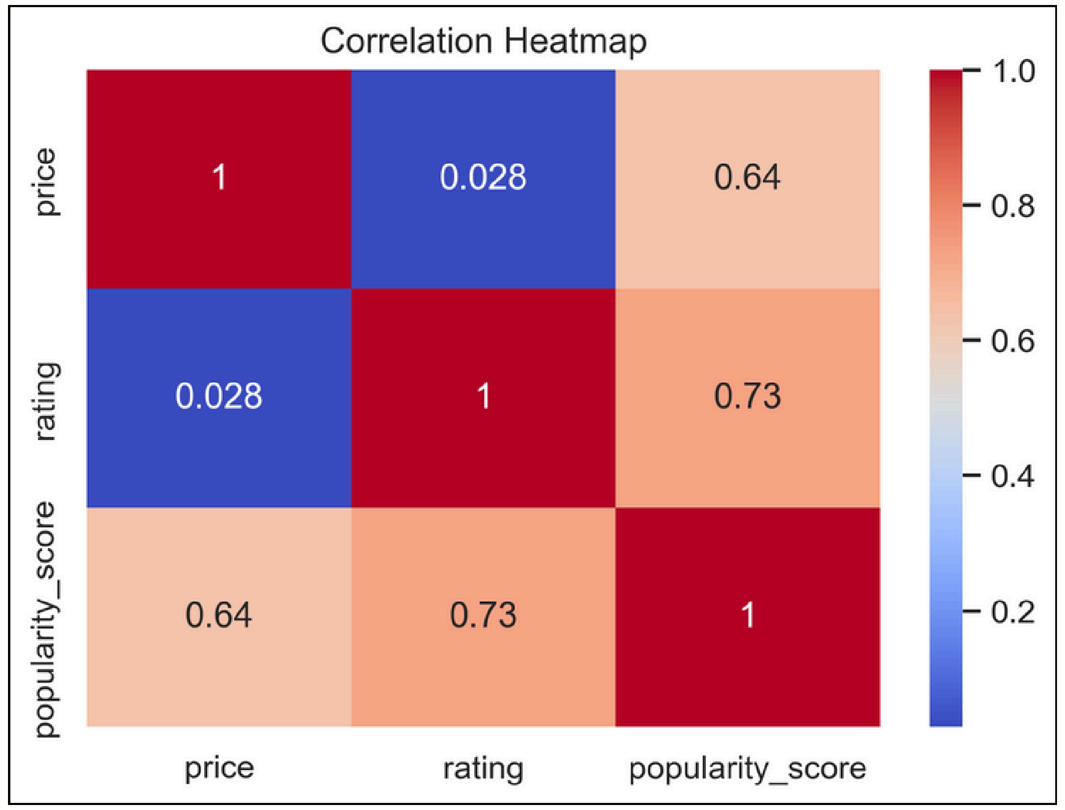




Exploratory Data Analysis & Visualization







INSIGHTS:

Most books are concentrated around mid to high ratings, indicating generally positive customer feedback across the platform.

Book prices are right-skewed, with most books priced in the lower to medium range and fewer high-priced books.

There is no strong linear relationship between price and rating, suggesting that higher-priced books are not necessarily rated higher.

Most books fall under the low and medium price bands, indicating affordability-focused pricing strategies.

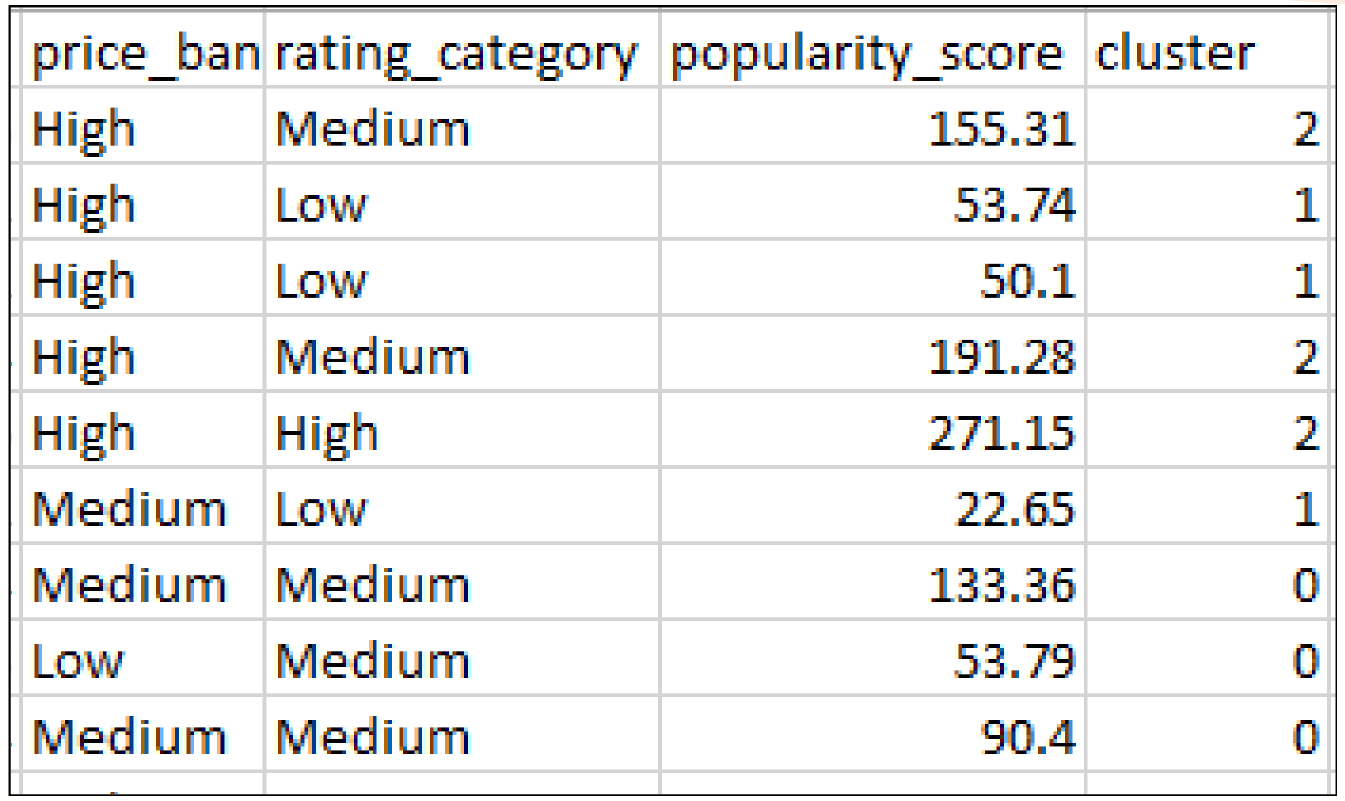
Books with higher ratings generally show higher popularity scores, reinforcing the importance of customer satisfaction.

Popularity score is positively correlated with rating, while price shows weak correlation with rating.

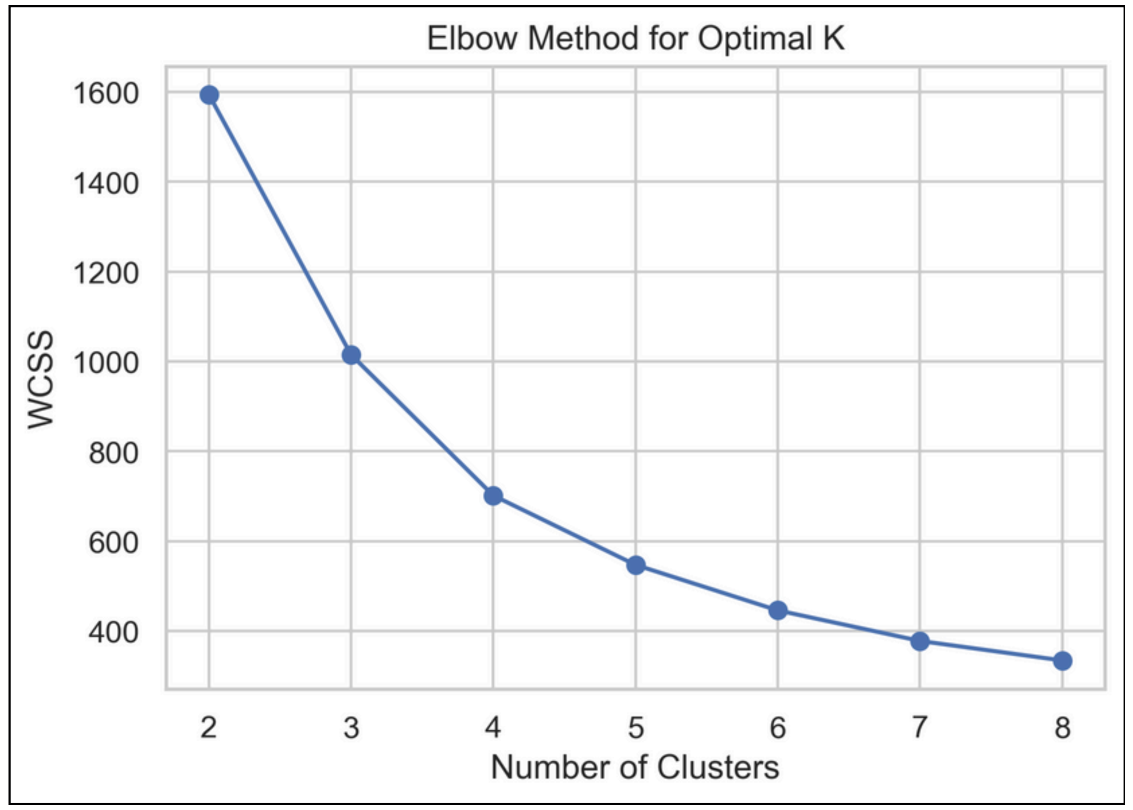
Model Building & Evaluation

- Features used: price, rating, popularity_score
- Scaling applied
- Elbow method → K = 3
- Silhouette score (mention value)

K-Means clustering was applied to segment books into three distinct groups based on pricing, ratings, and popularity. The silhouette score indicates reasonable cluster separation.

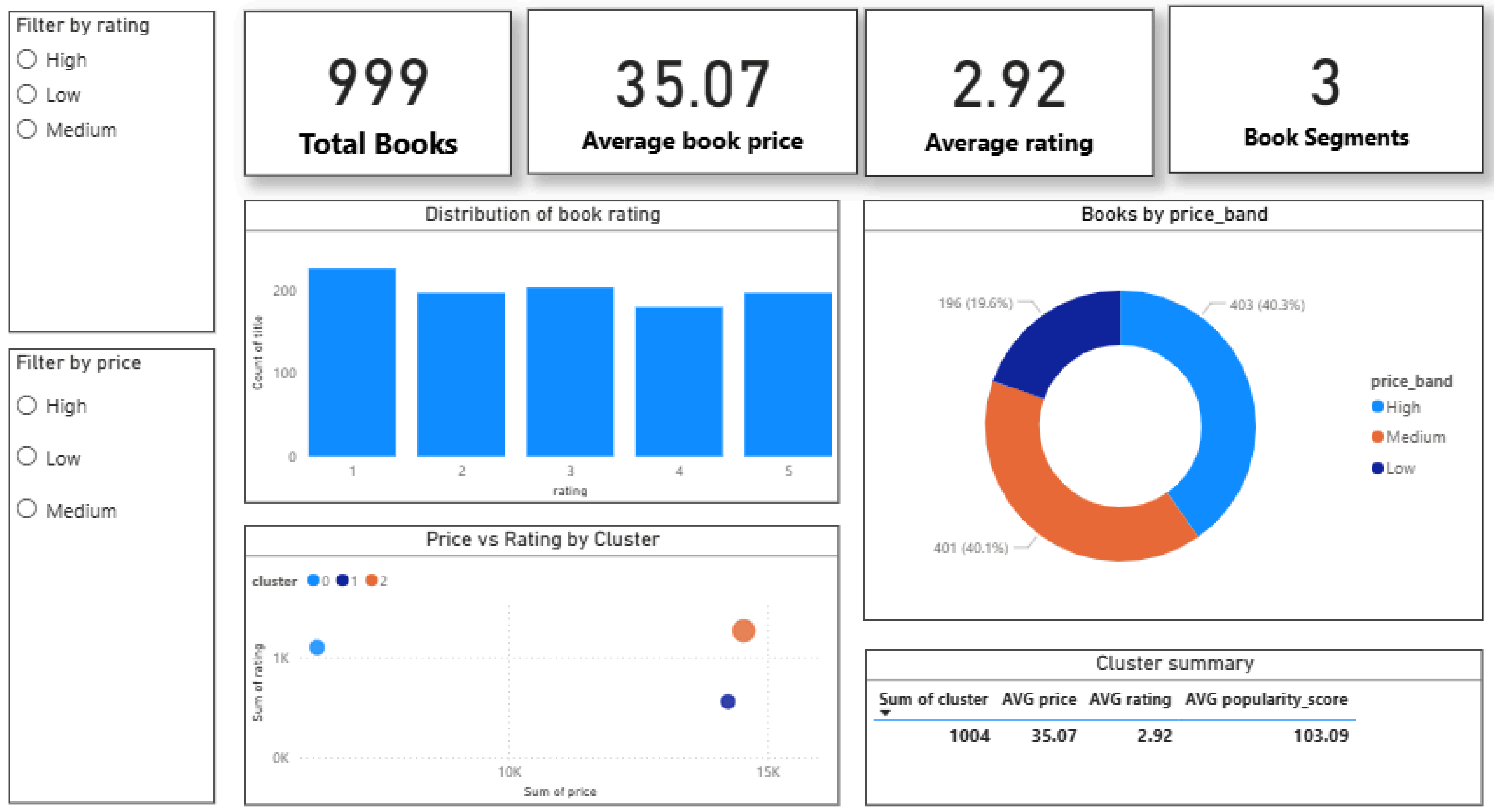


price_ban	rating_category	popularity_score	cluster
High	Medium	155.31	2
High	Low	53.74	1
High	Low	50.1	1
High	Medium	191.28	2
High	High	271.15	2
Medium	Low	22.65	1
Medium	Medium	133.36	0
Low	Medium	53.79	0
Medium	Medium	90.4	0



Interactive Dashboard

An interactive Power BI dashboard was developed to visualize key metrics and allow users to explore insights through filters and drill-downs.


Filter by rating
- ○ High
- ○ Low
- ○ Medium

Filter by price
- ○ High
- ○ Low
- ○ Medium

999 Total Books

35.07 Average book price

2.92 Average rating

3 Book Segments

Cluster summary

Sum of cluster	AVG price	AVG rating	AVG popularity_score
1004	35.07	2.92	103.09

BUSINESS INSIGHTS & RECOMMENDATIONS

Summary of Methodology, Insights, Model Results, and Recommendations

Methodology Summary

This project followed an end-to-end data science workflow using real-world web data. Book-related data was collected from a publicly accessible online source through a Python-based web scraping script developed using Requests and BeautifulSoup. Approximately 1,000 book records were extracted across multiple pages to ensure sufficient data volume.

The raw scraped data was cleaned and preprocessed by handling encoding issues, correcting data types, removing inconsistencies, and engineering additional features such as price categories, rating categories, and a popularity score. Exploratory Data Analysis (EDA) was performed using Python visualization libraries to identify trends, distributions, and relationships within the data.

To segment books into meaningful groups, an unsupervised machine learning approach using K-Means clustering was applied after feature scaling. Finally, an interactive Power BI dashboard was created to present key metrics, trends, and segmentation results in a user-friendly and decision-oriented format.

Key Insights from Data Analysis

- The majority of books are priced within the low to medium price range, indicating a market focus on affordability rather than premium pricing.
- Most books receive medium to high ratings, suggesting generally positive customer sentiment across the dataset.
- There is no strong linear relationship between book price and customer rating, implying that higher-priced books do not necessarily receive better ratings.
- Popularity scores tend to increase with higher ratings, highlighting customer satisfaction as a key driver of perceived book popularity.
- Correlation analysis shows that ratings have a stronger influence on popularity compared to price.

Machine Learning Model Results

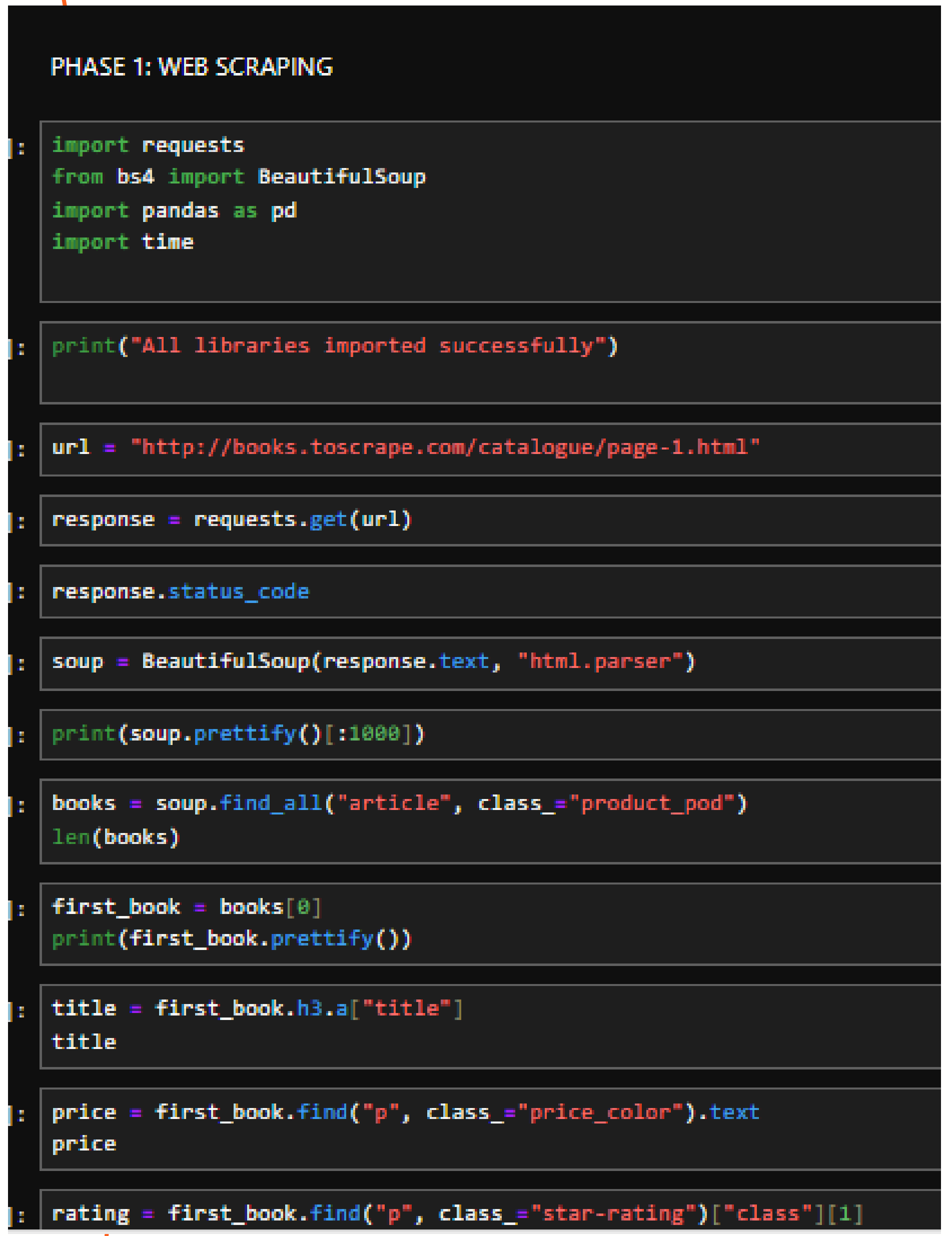
K-Means clustering was used to segment books based on price, rating, and popularity score. The elbow method was applied to determine the optimal number of clusters, and three clusters were selected for final modeling. Feature scaling was performed prior to model training to ensure accurate distance-based clustering.

The silhouette score indicates reasonable separation between clusters, validating the effectiveness of the segmentation. The resulting clusters represent distinct groups of books with different pricing and rating characteristics, such as value-oriented books, moderately priced popular books, and higher-priced niche offerings.

Business Recommendations

- Pricing strategies should not rely solely on premium pricing, as higher prices do not guarantee better customer ratings.
- Improving customer satisfaction and content quality can have a stronger impact on popularity than price adjustments.
- Segmentation insights can be used to tailor marketing strategies for different book groups, such as promoting affordable high-rated books or positioning premium books for niche audiences.
- Businesses can leverage clustering to identify underperforming segments and optimize inventory, promotions, and pricing strategies accordingly.
- Interactive dashboards should be used by decision-makers to continuously monitor trends and adapt strategies based on real-time insights.

ALL CODES PERFORM IN PYTHON:



PHASE 1: WEB SCRAPING

```python
import requests
from bs4 import BeautifulSoup
import pandas as pd
import time
```

```python
print("All libraries imported successfully")
```

```python
url = "http://books.toscrape.com/catalogue/page-1.html"
```

```python
response = requests.get(url)
```

```python
response.status_code
```

```python
soup = BeautifulSoup(response.text, "html.parser")
```

```python
print(soup.prettify()[:1000])
```

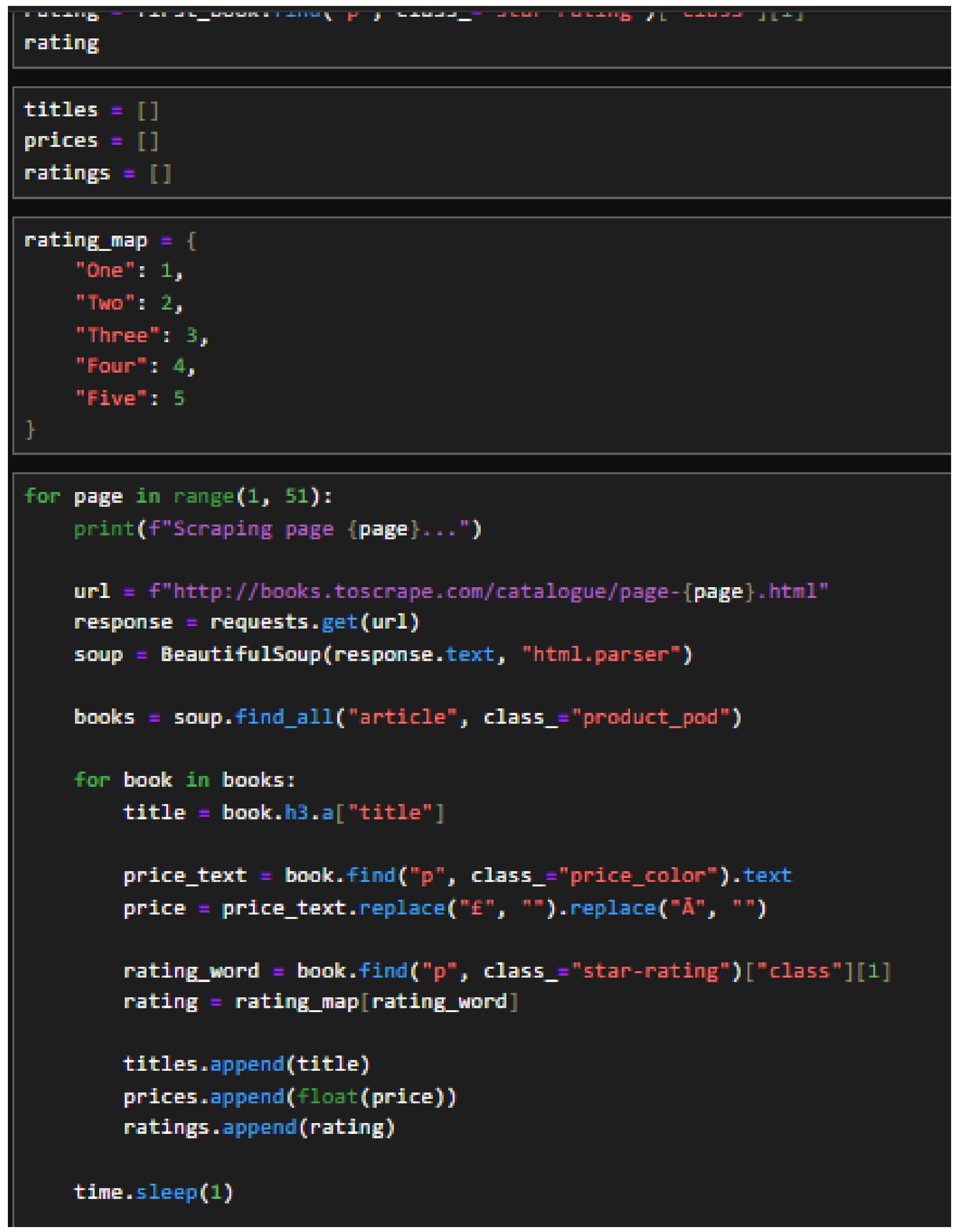
```python
books = soup.find_all("article", class_="product_pod")
len(books)
```

```python
first_book = books[0]
print(first_book.prettify())
```

```python
title = first_book.h3.a["title"]
title
```

```python
price = first_book.find("p", class_="price_color").text
price
```

```python
rating = first_book.find("p", class_="star-rating")["class"][1]
```

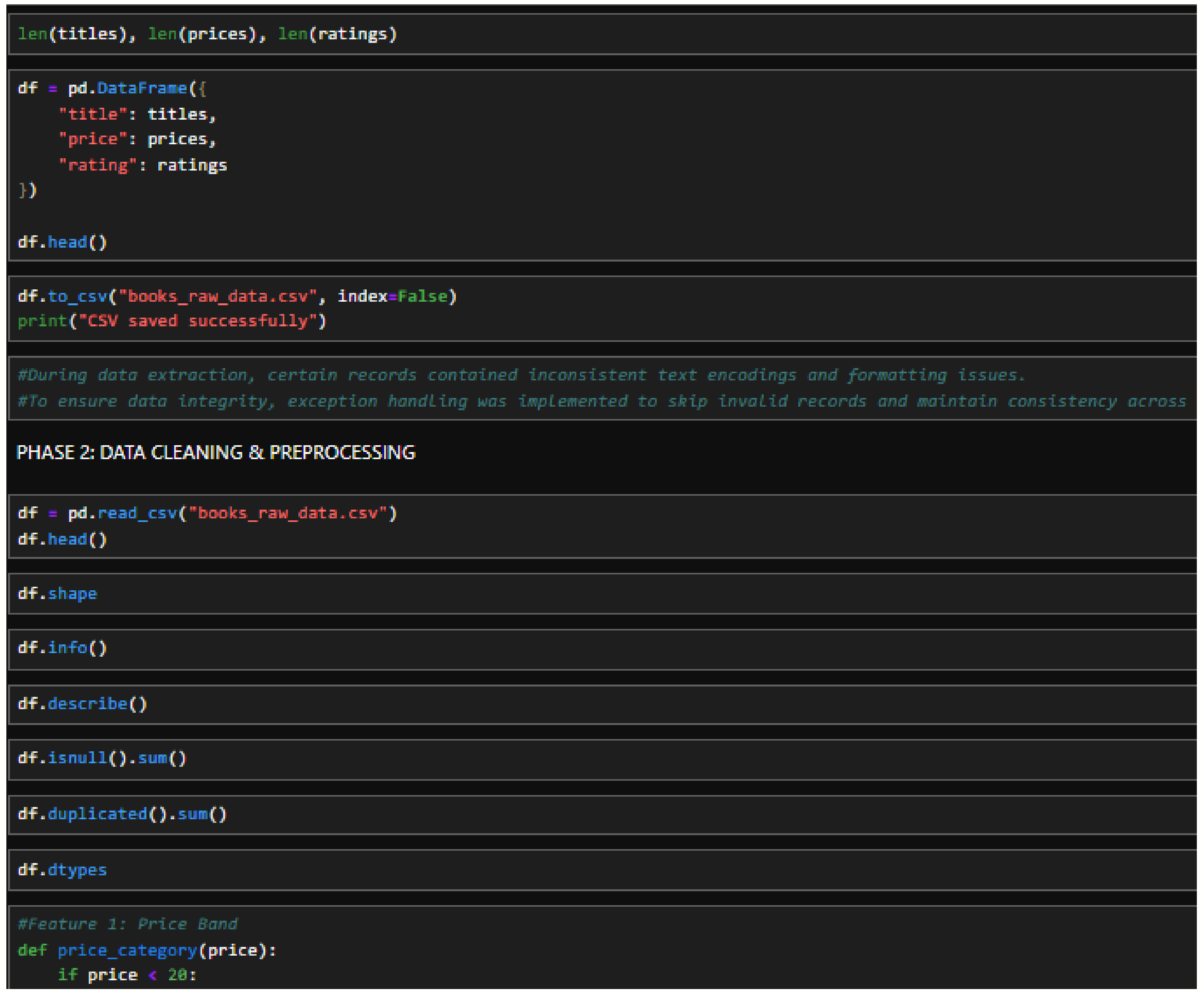


```python
rating
```

```python
titles = []
prices = []
ratings = []
```

```python
rating_map = {
    "One": 1,
    "Two": 2,
    "Three": 3,
    "Four": 4,
    "Five": 5
}
```

```python
for page in range(1, 51):
    print(f"Scraping page {page}...")

    url = f"http://books.toscrape.com/catalogue/page-{page}.html"
    response = requests.get(url)
    soup = BeautifulSoup(response.text, "html.parser")

    books = soup.find_all("article", class_="product_pod")

    for book in books:
        title = book.h3.a["title"]

        price_text = book.find("p", class_="price_color").text
        price = price_text.replace("£", "").replace("Â", "")

        rating_word = book.find("p", class_="star-rating")["class"][1]
        rating = rating_map[rating_word]

        titles.append(title)
        prices.append(float(price))
        ratings.append(rating)

    time.sleep(1)
```



```python
len(titles), len(prices), len(ratings)
```

```python
df = pd.DataFrame({
    "title": titles,
    "price": prices,
    "rating": ratings
})

df.head()
```

```python
df.to_csv("books_raw_data.csv", index=False)
print("CSV saved successfully")
```

```python
#During data extraction, certain records contained inconsistent text encodings and formatting issues.
#To ensure data integrity, exception handling was implemented to skip invalid records and maintain consistency across
```

PHASE 2: DATA CLEANING & PREPROCESSING

```python
df = pd.read_csv("books_raw_data.csv")
df.head()
```

```python
df.shape
```

```python
df.info()
```

```python
df.describe()
```

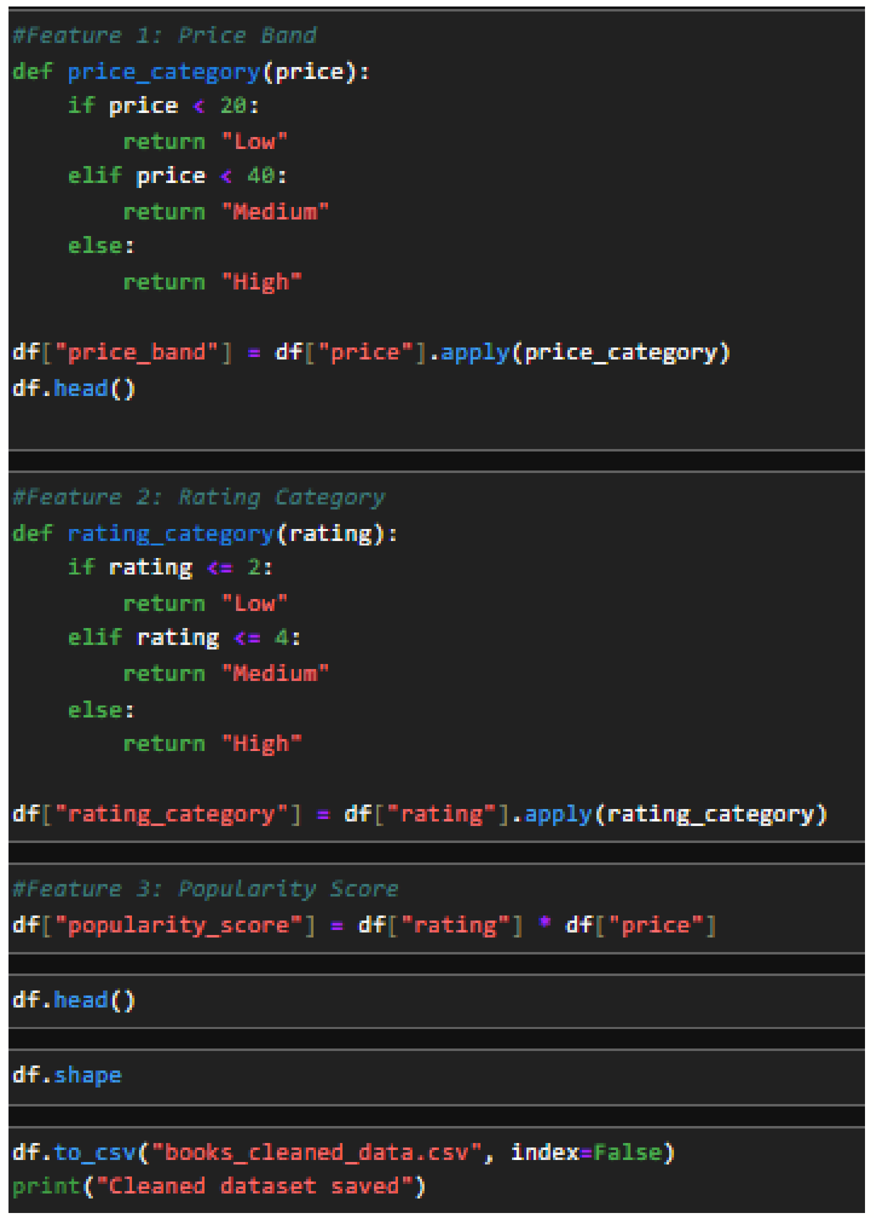
```python
df.isnull().sum()
```

```python
df.duplicated().sum()
```

```python
df.dtypes
```

```python
#Feature 1: Price Band
def price_category(price):
    if price < 20:
```

ALL CODES PERFORM IN PYTHON:



```python
#Feature 1: Price Band
def price_category(price):
    if price < 20:
        return "Low"
    elif price < 40:
        return "Medium"
    else:
        return "High"

df["price_band"] = df["price"].apply(price_category)
df.head()
```

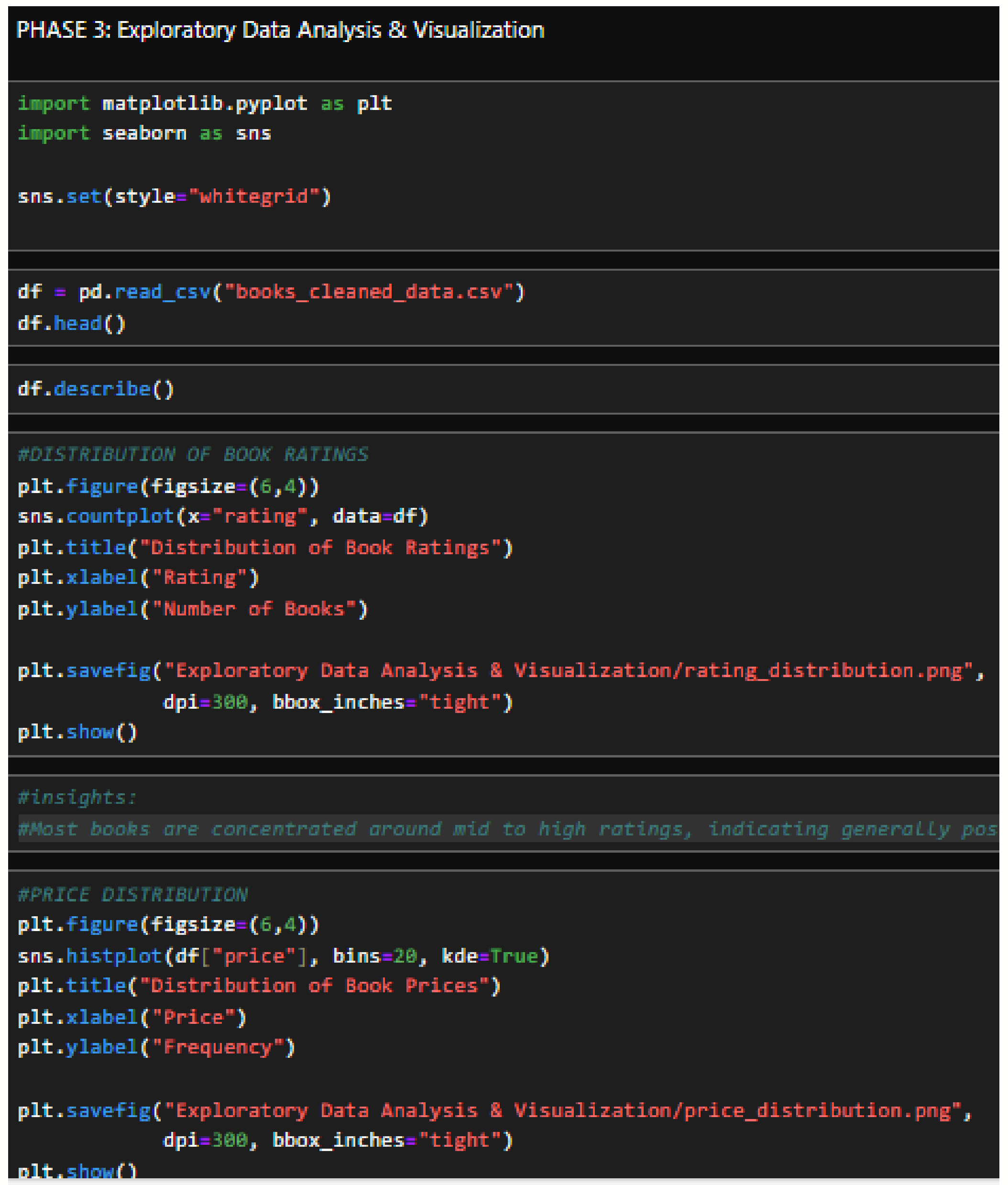
```python
#Feature 2: Rating Category
def rating_category(rating):
    if rating <= 2:
        return "Low"
    elif rating <= 4:
        return "Medium"
    else:
        return "High"

df["rating_category"] = df["rating"].apply(rating_category)
```

```python
#Feature 3: Popularity Score
df["popularity_score"] = df["rating"] * df["price"]
```

```python
df.head()
```

```python
df.shape
```

```python
df.to_csv("books_cleaned_data.csv", index=False)
print("Cleaned dataset saved")
```



PHASE 3: Exploratory Data Analysis & Visualization

```python
import matplotlib.pyplot as plt
import seaborn as sns

sns.set(style="whitegrid")
```

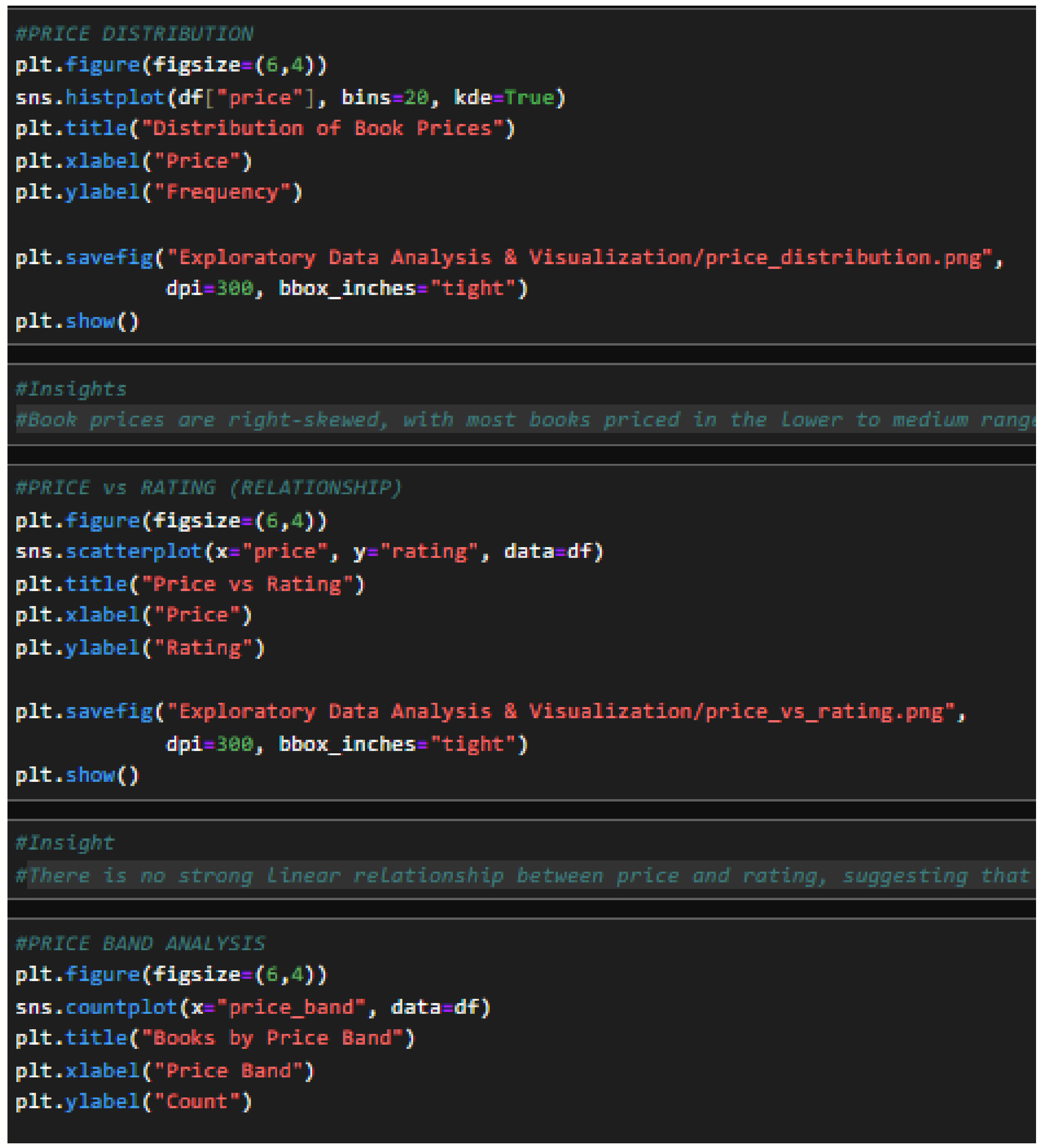
```python
df = pd.read_csv("books_cleaned_data.csv")
df.head()
```

```python
df.describe()
```

```python
#DISTRIBUTION OF BOOK RATINGS
plt.figure(figsize=(6,4))
sns.countplot(x="rating", data=df)
plt.title("Distribution of Book Ratings")
plt.xlabel("Rating")
plt.ylabel("Number of Books")

plt.savefig("Exploratory Data Analysis & Visualization/rating_distribution.png",
            dpi=300, bbox_inches="tight")
plt.show()
```

```python
#insights:
#Most books are concentrated around mid to high ratings, indicating generally pos
```

```python
#PRICE DISTRIBUTION
plt.figure(figsize=(6,4))
sns.histplot(df["price"], bins=20, kde=True)
plt.title("Distribution of Book Prices")
plt.xlabel("Price")
plt.ylabel("Frequency")

plt.savefig("Exploratory Data Analysis & Visualization/price_distribution.png",
            dpi=300, bbox_inches="tight")
plt.show()
```



```python
#PRICE DISTRIBUTION
plt.figure(figsize=(6,4))
sns.histplot(df["price"], bins=20, kde=True)
plt.title("Distribution of Book Prices")
plt.xlabel("Price")
plt.ylabel("Frequency")

plt.savefig("Exploratory Data Analysis & Visualization/price_distribution.png",
            dpi=300, bbox_inches="tight")
plt.show()
```

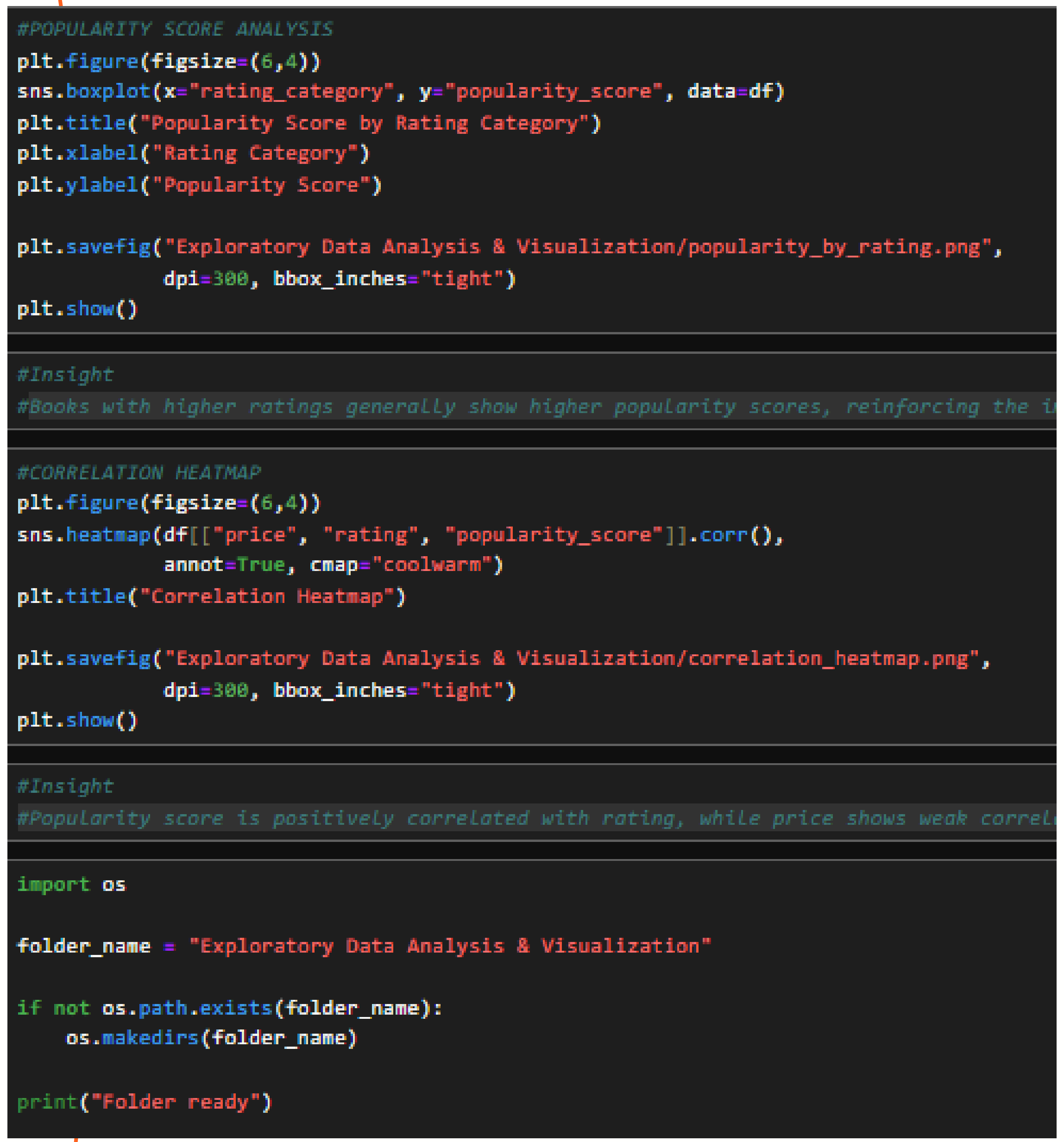
```python
#Insights
#Book prices are right-skewed, with most books priced in the lower to medium range
```

```python
#PRICE vs RATING (RELATIONSHIP)
plt.figure(figsize=(6,4))
sns.scatterplot(x="price", y="rating", data=df)
plt.title("Price vs Rating")
plt.xlabel("Price")
plt.ylabel("Rating")

plt.savefig("Exploratory Data Analysis & Visualization/price_vs_rating.png",
            dpi=300, bbox_inches="tight")
plt.show()
```

```python
#Insight
#There is no strong linear relationship between price and rating, suggesting that
```

```python
#PRICE BAND ANALYSIS
plt.figure(figsize=(6,4))
sns.countplot(x="price_band", data=df)
plt.title("Books by Price Band")
plt.xlabel("Price Band")
plt.ylabel("Count")
```

ALL CODES PERFORM IN PYTHON:



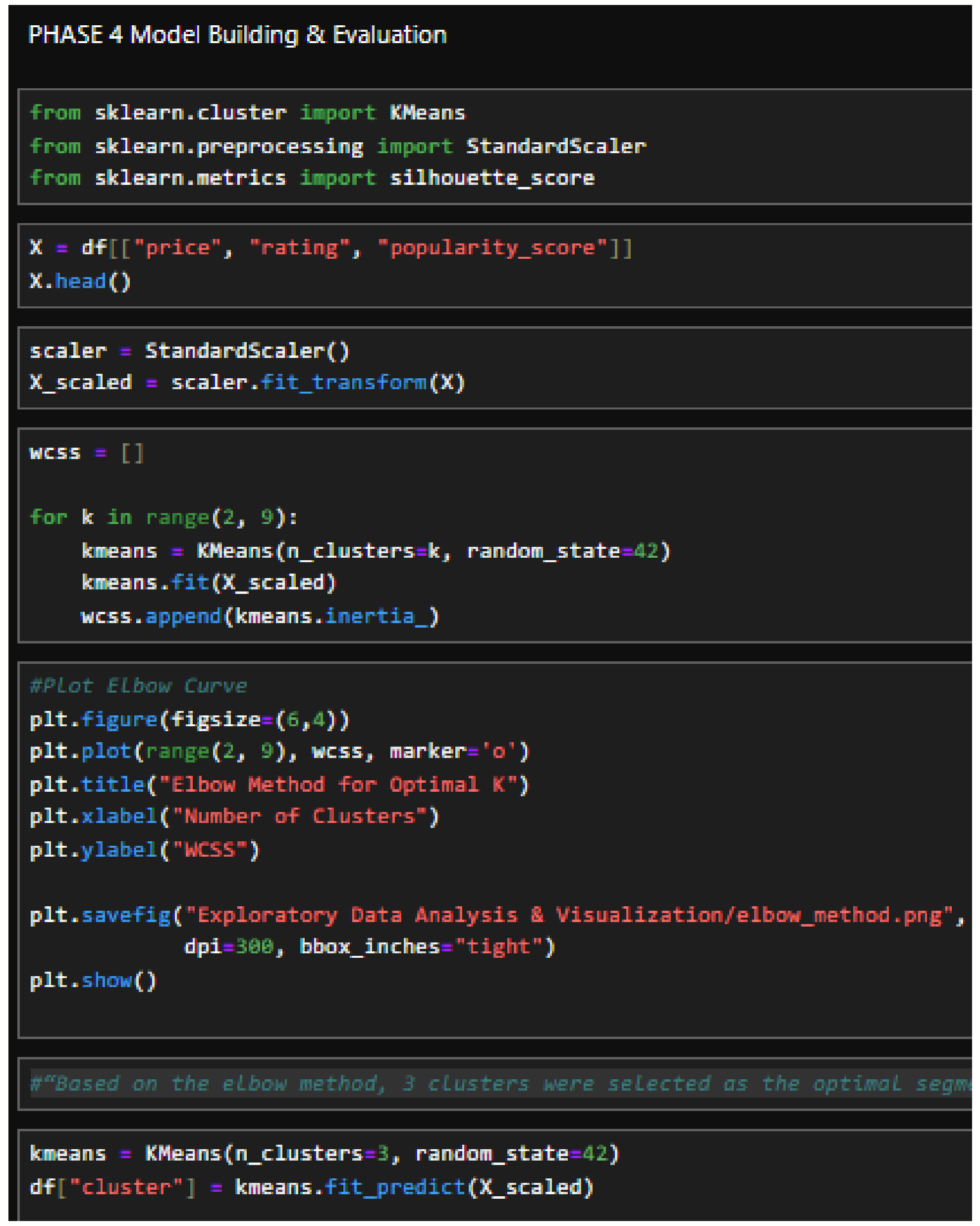
```python
#POPULARITY SCORE ANALYSIS
plt.figure(figsize=(6,4))
sns.boxplot(x="rating_category", y="popularity_score", data=df)
plt.title("Popularity Score by Rating Category")
plt.xlabel("Rating Category")
plt.ylabel("Popularity Score")

plt.savefig("Exploratory Data Analysis & Visualization/popularity_by_rating.png",
            dpi=300, bbox_inches="tight")
plt.show()
```

```python
#Insight
#Books with higher ratings generally show higher popularity scores, reinforcing the i
```

```python
#CORRELATION HEATMAP
plt.figure(figsize=(6,4))
sns.heatmap(df[["price", "rating", "popularity_score"]].corr(),
            annot=True, cmap="coolwarm")
plt.title("Correlation Heatmap")

plt.savefig("Exploratory Data Analysis & Visualization/correlation_heatmap.png",
            dpi=300, bbox_inches="tight")
plt.show()
```

```python
#Insight
#Popularity score is positively correlated with rating, while price shows weak correl
```

```python
import os

folder_name = "Exploratory Data Analysis & Visualization"

if not os.path.exists(folder_name):
    os.makedirs(folder_name)

print("Folder ready")
```



PHASE 4 Model Building & Evaluation

```python
from sklearn.cluster import KMeans
from sklearn.preprocessing import StandardScaler
from sklearn.metrics import silhouette_score
```

```python
X = df[["price", "rating", "popularity_score"]]
X.head()
```

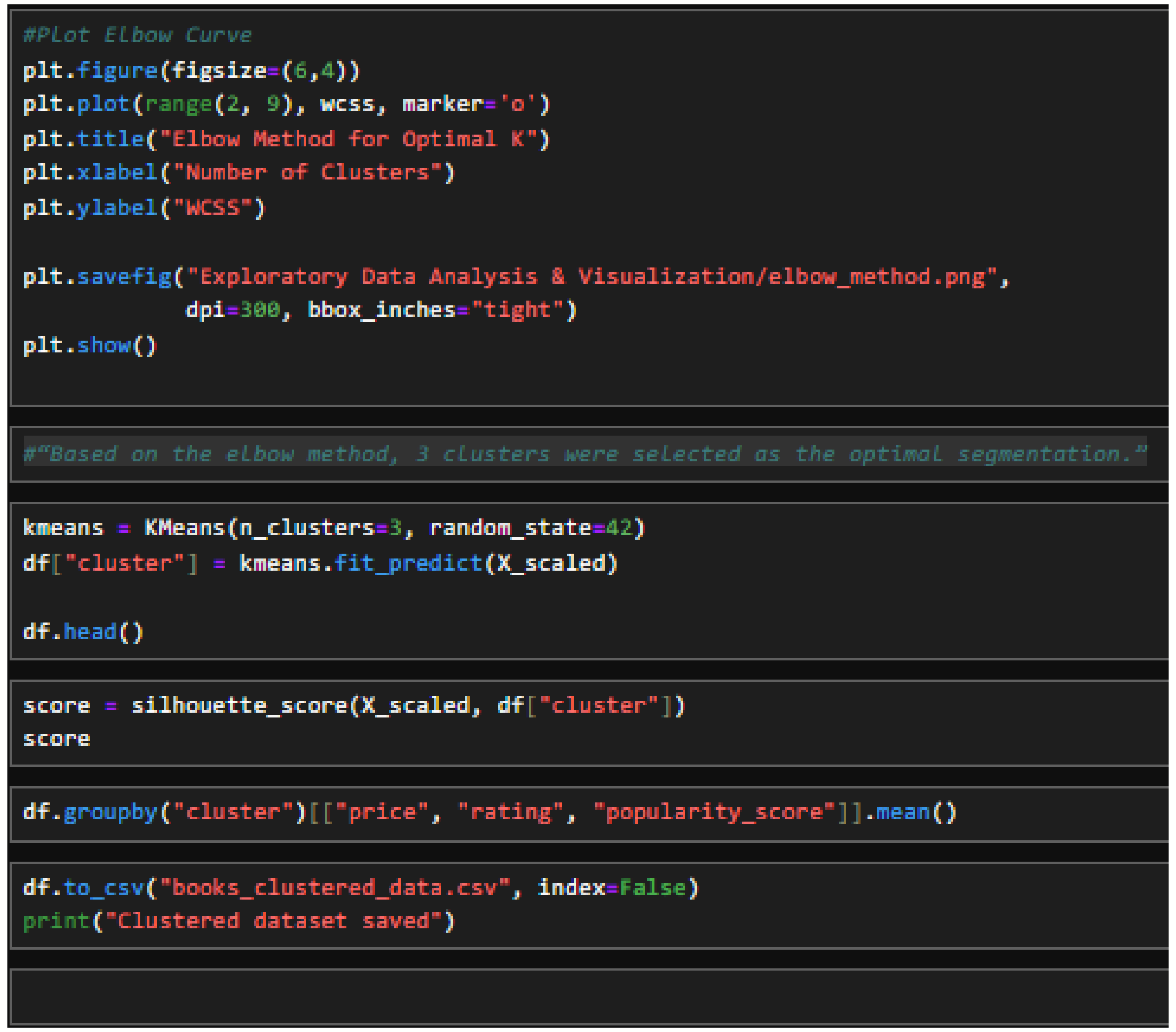
```python
scaler = StandardScaler()
X_scaled = scaler.fit_transform(X)
```

```python
wcss = []

for k in range(2, 9):
    kmeans = KMeans(n_clusters=k, random_state=42)
    kmeans.fit(X_scaled)
    wcss.append(kmeans.inertia_)
```

```python
#Plot Elbow Curve
plt.figure(figsize=(6,4))
plt.plot(range(2, 9), wcss, marker='o')
plt.title("Elbow Method for Optimal K")
plt.xlabel("Number of Clusters")
plt.ylabel("WCSS")

plt.savefig("Exploratory Data Analysis & Visualization/elbow_method.png",
            dpi=300, bbox_inches="tight")
plt.show()
```

```python
#"Based on the elbow method, 3 clusters were selected as the optimal segm
```

```python
kmeans = KMeans(n_clusters=3, random_state=42)
df["cluster"] = kmeans.fit_predict(X_scaled)
```



```python
#Plot Elbow Curve
plt.figure(figsize=(6,4))
plt.plot(range(2, 9), wcss, marker='o')
plt.title("Elbow Method for Optimal K")
plt.xlabel("Number of Clusters")
plt.ylabel("WCSS")

plt.savefig("Exploratory Data Analysis & Visualization/elbow_method.png",
            dpi=300, bbox_inches="tight")
plt.show()
```

```python
#"Based on the elbow method, 3 clusters were selected as the optimal segmentation."
```

```python
kmeans = KMeans(n_clusters=3, random_state=42)
df["cluster"] = kmeans.fit_predict(X_scaled)

df.head()
```

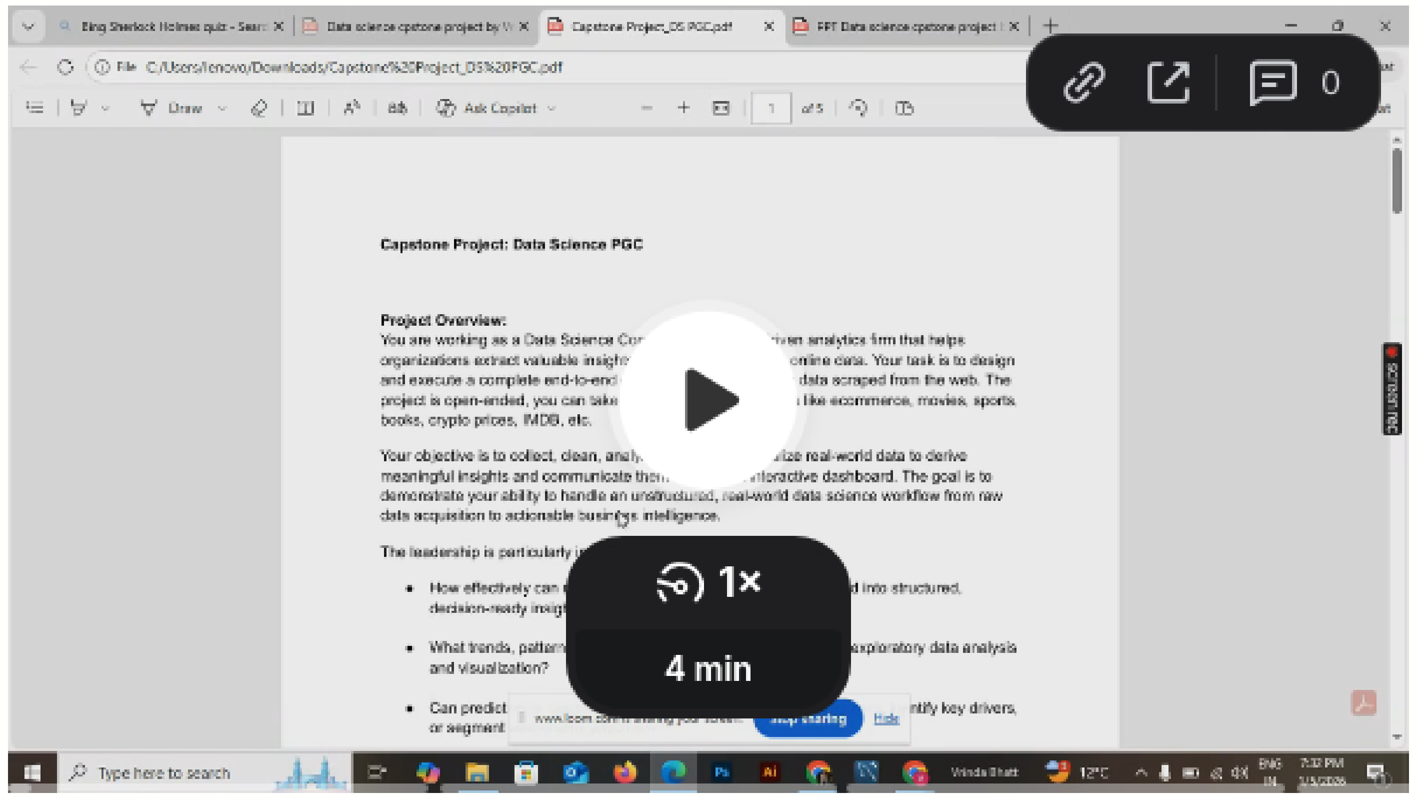
```python
score = silhouette_score(X_scaled, df["cluster"])
score
```

```python
df.groupby("cluster")[["price", "rating", "popularity_score"]].mean()
```

```python
df.to_csv("books_clustered_data.csv", index=False)
print("Clustered dataset saved")
```

Thank you

VIDEO LINK



Capstone Project: Data Science PGC

Project Overview:
You are working as a Data Science Co[illegible] ...iven analytics firm that helps organizations extract valuable insigh[illegible] ...online data. Your task is to design and execute a complete end-to-end [illegible] ...data scraped from the web. The project is open-ended, you can take [illegible] ...like ecommerce, movies, sports, books, crypto prices, IMDB, etc.

Your objective is to collect, clean, analy[illegible] ...lize real-world data to derive meaningful insights and communicate them [illegible] ...nteractive dashboard. The goal is to demonstrate your ability to handle an unstructured, real-world data science workflow from raw data acquisition to actionable business intelligence.

The leadership is particularly i[illegible]

- How effectively can [illegible] ...d into structured, decision-ready insig[illegible]
- What trends, pattern[illegible] ...exploratory data analysis and visualization?
- Can predict[illegible] ...ntify key drivers, or segment[illegible]

Video link