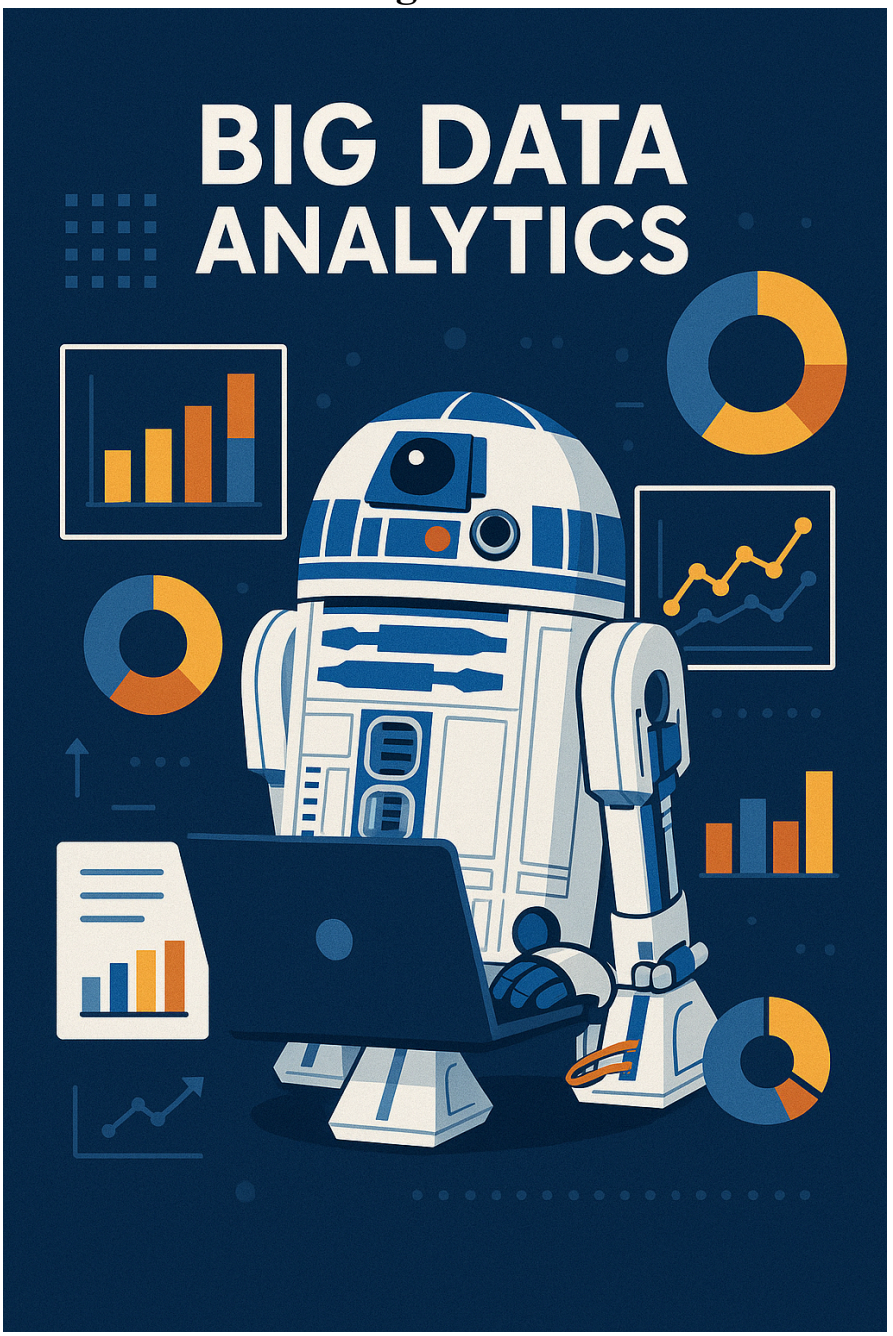
COMP 3610
Big Data Analytics

Assignment 3



Jaheem Edwards (816035606)

Liam Ramjewan (816033607)

Aadam Shah (816024885)

Table of Contents

Table of Contents 1
Introduction 2
Data Acquisition 3
Code 4
Data Cleaning & Preprocessing 10
Code 11
EDA 16
Star Rating Histogram 16
Bar Chart for Top 10 Categories 16
Bar Chart for Top 10 Brands 17
Line chart of time based trends 17
Pearson correlation between review length and star rating 18
Code 18
Binary Sentiment (Logistic Regression) 28
Evaluation 28
Code 30
Recommender System using ALS 35
Code 36
Clustering / Segmentation (k-means) 39
K-Means Clustering 40
Cluster Analysis & Interpretation 41
Code 42
Models 46
Conclusion 47
Code Repository 48

Introduction

This assignment involves working with the McAuley-Lab/Amazon-Reviews-2023 dataset, a large-scale collection of product reviews from Amazon, available on Hugging Face at this link. With an uncompressed size of approximately 200GB across 34 categories, the dataset provides a diverse and comprehensive resource for applying various data science techniques.

The objective of this assignment is to guide participants through a structured process involving data acquisition, cleaning, and preprocessing, followed by exploratory data analysis (EDA). Subsequent tasks include implementing machine learning algorithms such as binary sentiment classification using logistic regression, developing a recommender system using Alternating Least Squares (ALS), and performing clustering with k-means.

This assignment required collaboration in groups of 3–4, with tasks distributed according to team capabilities. Although the methods and algorithms for each task are predefined, participants had the flexibility to implement them in their preferred environment, whether on a local machine, cluster, or cloud infrastructure.

Upon completion, this assignment will provide hands on experience with large scale data processing, machine learning, and data driven insights, while also highlighting the computational resources and challenges associated with working with extensive datasets.

Data Acquisition

For the data acquisition process, the Amazon Reviews 2023 dataset, which includes reviews from 34 different product categories, was sourced from the McAuley Lab repository on Hugging Face. This dataset provides useful information such as review text, ratings, and metadata that are essential for tasks like sentiment analysis, recommendation systems, and clustering. The load dataset function from the datasets library was used to download the entire dataset, ensuring that all 34 categories were successfully retrieved. Other options like the download all amazon reviews function were also explored to verify the completeness of the download.

The success of the data acquisition was confirmed using screenshots that showed the compressed versions of the dataset, instead of relying on saved logs. These screenshots served as evidence that all files were downloaded correctly, preserving the integrity of the data.

As part of the deliverables, the report includes screenshots to prove that the dataset was acquired successfully. It also provides the scripts used for downloading and organizing the data, clearly documenting the steps taken to prepare the dataset for analysis.

Code

Download Function:

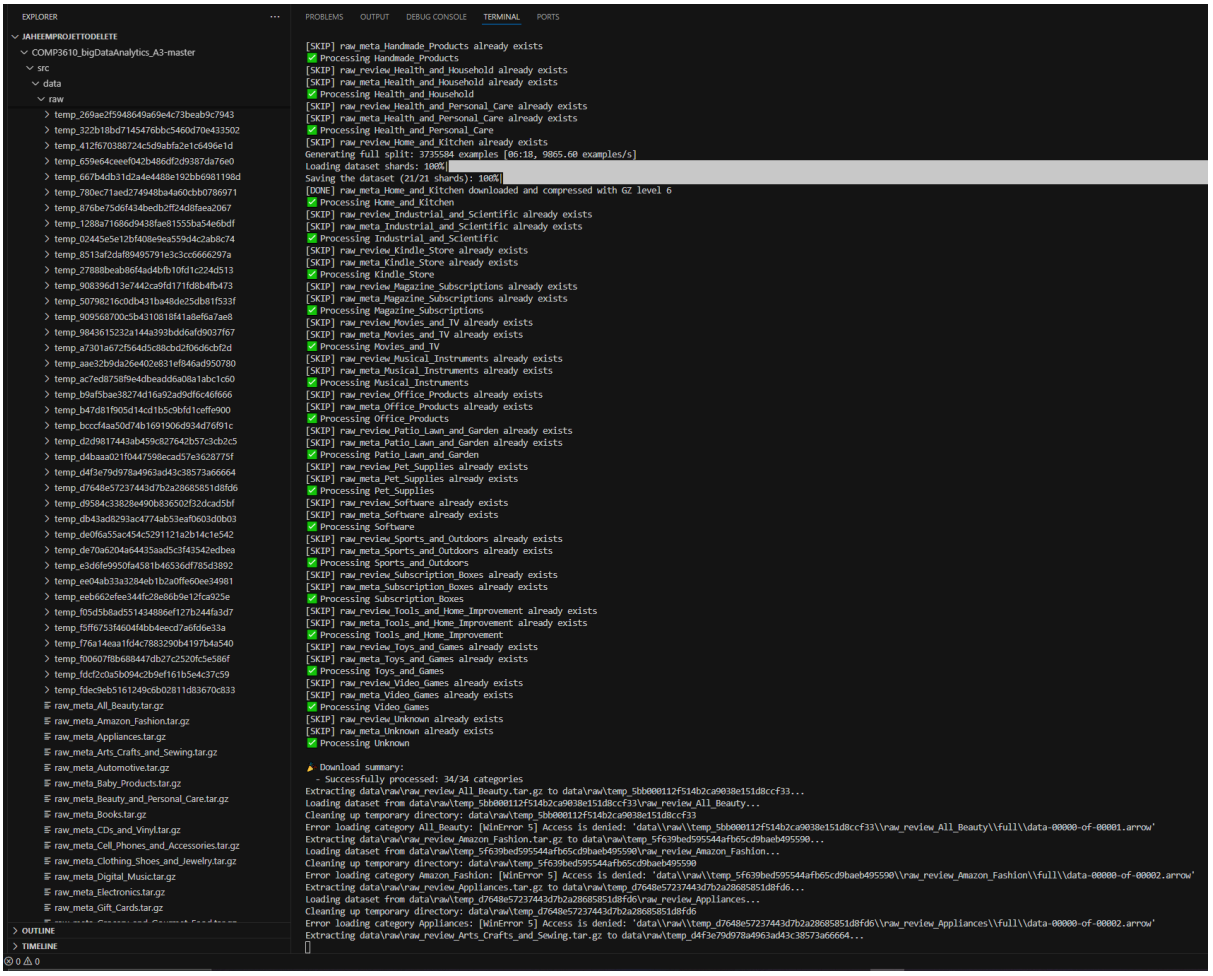
```
def download_amazon_reviews():
    try:
        download_all_amazon_reviews(
            base_save_path=base_save_path,
            categories=VALID_CATEGORIES,
            compress=True,
            compression_format="gz",
            compression_level=6
        )
        return True
    except Exception as e:
        print(f"Error during download: {e}")
        return False
```

Loading Data Function:

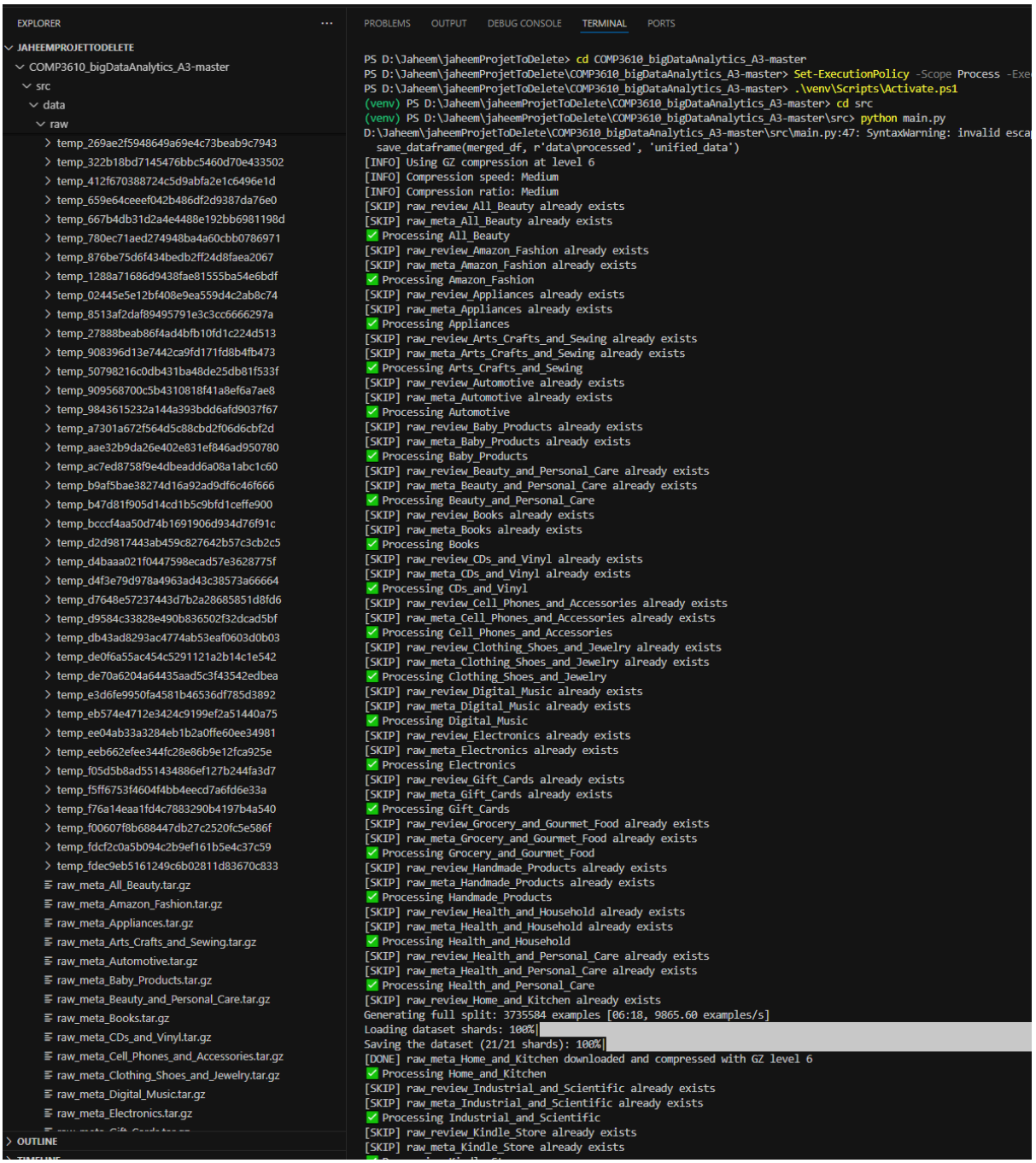
```
def load_amazon_reviews():
    review_dfs = []
    meta_dfs = []

    for category in VALID_CATEGORIES:
        try:
            review_compressed_path = Path(base_save_path) /
f"raw_review_{category}.tar.gz"
            review_dataset =
load_compressed_dataset(review_compressed_path)
            review_df = pd.DataFrame(review_dataset["full"])
            review_dfs.append(review_df)
            meta_compressed_path = Path(base_save_path) /
f"raw_meta_{category}.tar.gz"
            meta_dataset =
load_compressed_dataset(meta_compressed_path)
            meta_df = pd.DataFrame(meta_dataset["full"])
            meta_dfs.append(meta_df)
        except Exception as e:
            print(f"Error loading category {category}: {e}")
            continue

      all_reviews_df = pd.concat(review_dfs, ignore_index=True) if
review_dfs else pd.DataFrame()

    all_meta_df = pd.concat(meta_dfs, ignore_index=True) if meta_dfs
else pd.DataFrame()

    return all_reviews_df, all_meta_df
```

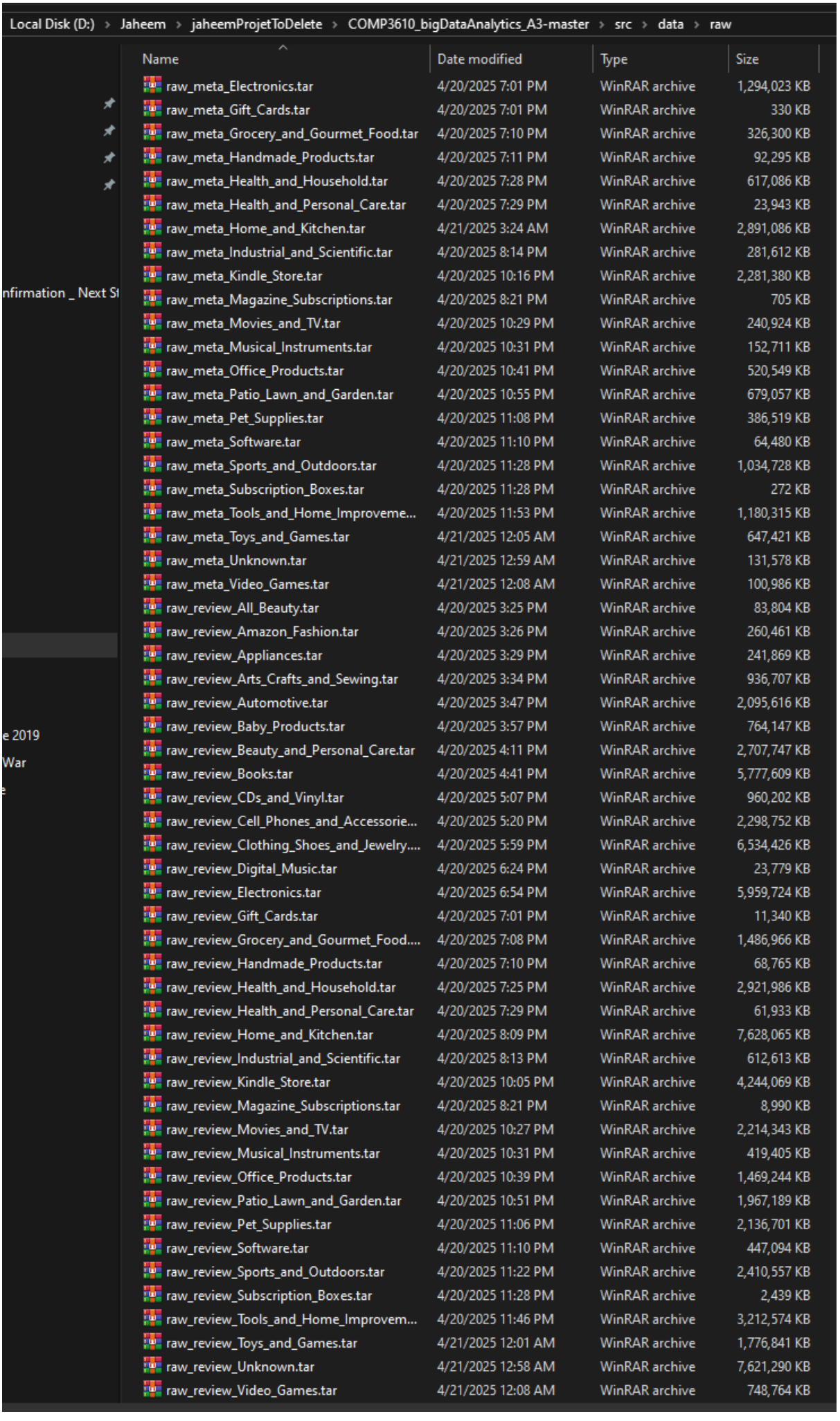
Saving and Loading Data Functions:

```
def save_dataframe(df, save_dir, filename):
    os.makedirs(save_dir, exist_ok=True)
    save_path = os.path.join(save_dir, f"{filename}.csv")
    df.to_csv(save_path, index=False)
    print(f"Data saved to {save_path}")

def load_dataframe(save_dir, filename):
    file_path = os.path.join(save_dir, f"{filename}.csv")
    if os.path.exists(file_path):
        try:
            df = pd.read_csv(file_path, low_memory=False)
            print(f"Loaded DataFrame from {file_path}")
            return df
        except Exception as e:
            print(f"Error loading {file_path}: {e}")
            return None
    else:
        print(f"File not found: {file_path}")
        return None
```


EXPLORER
JAHEEMPROJETTODELETE
COMP3610_bigDataAnalytics_A3-master
src
data
raw
temp_269ae2f5948649a69e4c73beab9c7943
temp_322b18bd7145476bbc5460d70e433502
temp_412f67038872 4c5d9abfa2e1c6496e1d
temp_659e64ceeef042b486df2d9387da76e0
temp_667b4db31d2a4e4488e192bb6981198d
temp_780ec71aed274948ba4a60cbb0786971
temp_876be75d6f434bedb2ff24d8faea2067
temp_1288a71686d9438fae81555ba54e6bdf
temp_02445e5e12bf408e9ea559d4c2ab8c74
temp_8513af2daf8949579 1e3c3cc6666297a
temp_27888beab86f4ad4bfb10fd1c224d513
temp_908396d13e7442ca9fd171fd8b4fb473
temp_50798216c0db431ba48de25db81f533f
temp_909568700c5b431081 8f41a8ef6a7ae8
temp_9843615232a144a393bdd6afd9037f67
temp_a7301a672f564d5c88cbd2f06d6cbf2d
temp_aae32b9da26e402e831ef846ad950780
temp_ac7ed8758f9e4dbeadd6a08a1abc1c60
temp_b9af5bae38274d16a92ad9df6c46f666
temp_b47d81f905d14cd1b5c9bfd1ceffe900
temp_bcccf4aa50d74b1691906d934d76f91c
temp_d2d9817443ab459c827642b57c3cb2c5
temp_d4baaa021f0447598ecad57e3628775f
temp_d4f3e79d978a4963ad43c38573a66664
temp_d7648e57237443d7b2a28685851d8fd6
temp_d9584c33828e490b836502f32dcad5bf
temp_db43ad8293ac4774ab53eaf0603d0b03
temp_de0f6a55ac454c5291121a2b14c1e542
temp_de70a6204a64435aad5c3f43542edbea
temp_e3d6fe9950fa4581b46536df785d3892
temp_ee04ab33a3284eb1b2a0ffe60ee34981
temp_eeb662efee344fc28e86b9e12fca925e
temp_f05d5b8ad551434886ef127b244fa3d7
temp_f5ff6753f460f4bb4eecd7a6fd6e33a
temp_f76a14eaa1fd4c7883290b4197b4a540
temp_f00607f8b688447db27c2520fc5e586f
temp_fdcf2c0a5b094c2b9ef161b5e4c37c59
temp_fdec9eb5161249c6b02811d83670c833
raw_meta_All_Beauty.tar.gz
raw_meta_Amazon_Fashion.tar.gz
raw_meta_Appliances.tar.gz
raw_meta_Arts_Crafts_and_Sewing.tar.gz
raw_meta_Automotive.tar.gz
raw_meta_Baby_Products.tar.gz
raw_meta_Beauty_and_Personal_Care.tar.gz
raw_meta_Books.tar.gz
raw_meta_CDs_and_Vinyl.tar.gz
raw_meta_Cell_Phones_and_Accessories.tar.gz
raw_meta_Clothing_Shoes_and_Jewelry.tar.gz
raw_meta_Digital_Music.tar.gz
raw_meta_Electronics.tar.gz
raw_meta_Gift_Cards.tar.gz
OUTLINE
TIMELINE
PROBLEMS
OUTPUT
DEBUG CONSOLE
TERMINAL
PORTS
[SKIP] raw_meta_Handmade_Products already exists
✅ Processing Handmade_Products
[SKIP] raw_review_Health_and_Household already exists
[SKIP] raw_meta_Health_and_Household already exists
✅ Processing Health_and_Household
[SKIP] raw_review_Health_and_Personal_Care already exists
[SKIP] raw_meta_Health_and_Personal_Care already exists
✅ Processing Health_and_Personal_Care
[SKIP] raw_review_Home_and_Kitchen already exists
Generating full split: 3735584 examples [06:18, 9865.60 examples/s]
Loading dataset shards: 100%
Saving the dataset (21/21 shards): 100%
[DONE] raw_meta_Home_and_Kitchen downloaded and compressed with GZ level 6
✅ Processing Home_and_Kitchen
[SKIP] raw_review_Industrial_and_Scientific already exists
[SKIP] raw_meta_Industrial_and_Scientific already exists
✅ Processing Industrial_and_Scientific
[SKIP] raw_review_Kindle_Store already exists
[SKIP] raw_meta_Kindle_Store already exists
✅ Processing Kindle_Store
[SKIP] raw_review_Magazine_Subscriptions already exists
[SKIP] raw_meta_Magazine_Subscriptions already exists
✅ Processing Magazine_Subscriptions
[SKIP] raw_review_Movies_and_TV already exists
[SKIP] raw_meta_Movies_and_TV already exists
✅ Processing Movies_and_TV
[SKIP] raw_review_Musical_Instruments already exists
[SKIP] raw_meta_Musical_Instruments already exists
✅ Processing Musical_Instruments
[SKIP] raw_review_Office_Products already exists
[SKIP] raw_meta_Office_Products already exists
✅ Processing Office_Products
[SKIP] raw_review_Patio_Lawn_and_Garden already exists
[SKIP] raw_meta_Patio_Lawn_and_Garden already exists
✅ Processing Patio_Lawn_and_Garden
[SKIP] raw_review_Pet_Supplies already exists
[SKIP] raw_meta_Pet_Supplies already exists
✅ Processing Pet_Supplies
[SKIP] raw_review_Software already exists
[SKIP] raw_meta_Software already exists
✅ Processing Software
[SKIP] raw_review_Sports_and_Outdoors already exists
[SKIP] raw_meta_Sports_and_Outdoors already exists
✅ Processing Sports_and_Outdoors
[SKIP] raw_review_Subscription_Boxes already exists
[SKIP] raw_meta_Subscription_Boxes already exists
✅ Processing Subscription_Boxes
[SKIP] raw_review_Tools_and_Home_Improvement already exists
[SKIP] raw_meta_Tools_and_Home_Improvement already exists
✅ Processing Tools_and_Home_Improvement
[SKIP] raw_review_Toys_and_Games already exists
[SKIP] raw_meta_Toys_and_Games already exists
✅ Processing Toys_and_Games
[SKIP] raw_review_Video_Games already exists
[SKIP] raw_meta_Video_Games already exists
✅ Processing Video_Games
[SKIP] raw_review_Unknown already exists
[SKIP] raw_meta_Unknown already exists
✅ Processing Unknown
⚠ Download summary:
- Successfully processed: 34/34 categories
Extracting data\raw\raw_review_All_Beauty.tar.gz to data\raw\temp_5bb000112f514b2ca9038e151d8ccf33...
Loading dataset from data\raw\temp_5bb000112f514b2ca9038e151d8ccf33\raw_review_All_Beauty...
Cleaning up temporary directory: data\raw\temp_5bb000112f514b2ca9038e151d8ccf33
Error loading category All_Beauty: [WinError 5] Access is denied: 'data\\raw\\temp_5bb000112f514b2ca9038e151d8ccf33\\raw_review_All_Beauty\\full\\data-00000-of-00001.arrow'
Extracting data\raw\raw_review_Amazon_Fashion.tar.gz to data\raw\temp_5f639bed595544afb65cd9baeb495590...
Loading dataset from data\raw\temp_5f639bed595544afb65cd9baeb495590\raw_review_Amazon_Fashion...
Cleaning up temporary directory: data\raw\temp_5f639bed595544afb65cd9baeb495590
Error loading category Amazon_Fashion: [WinError 5] Access is denied: 'data\\raw\\temp_5f639bed595544afb65cd9baeb495590\\raw_review_Amazon_Fashion\\full\\data-00000-of-00002.arrow'
Extracting data\raw\raw_review_Appliances.tar.gz to data\raw\temp_d7648e57237443d7b2a28685851d8fd6...
Loading dataset from data\raw\temp_d7648e57237443d7b2a28685851d8fd6\raw_review_Appliances...
Cleaning up temporary directory: data\raw\temp_d7648e57237443d7b2a28685851d8fd6
Error loading category Appliances: [WinError 5] Access is denied: 'data\\raw\\temp_d7648e57237443d7b2a28685851d8fd6\\raw_review_Appliances\\full\\data-00000-of-00002.arrow'
Extracting data\raw\raw_review_Arts_Crafts_and_Sewing.tar.gz to data\raw\temp_d4f3e79d978a4963ad43c38573a66664...

```
🎉 Download summary:
  - Successfully processed: 34/34 categories
Extracting data\raw\raw_review_All_Beauty.tar.gz to data\raw\temp_5bb000112f514b2ca9038e151d8ccf
Loading dataset from data\raw\temp_5bb000112f514b2ca9038e151d8ccf33\raw_review_All_Beauty...
Cleaning up temporary directory: data\raw\temp_5bb000112f514b2ca9038e151d8ccf33
Error loading category All_Beauty: [WinError 5] Access is denied: 'data\\raw\\temp_5bb000112f514
Extracting data\raw\raw_review_Amazon_Fashion.tar.gz to data\raw\temp_5f639bed595544afb65cd9baeb
Loading dataset from data\raw\temp_5f639bed595544afb65cd9baeb495590\raw_review_Amazon_Fashion...
Cleaning up temporary directory: data\raw\temp_5f639bed595544afb65cd9baeb495590
Error loading category Amazon_Fashion: [WinError 5] Access is denied: 'data\\raw\\temp_5f639bed5
Extracting data\raw\raw_review_Appliances.tar.gz to data\raw\temp_d7648e57237443d7b2a28685851d8f
Loading dataset from data\raw\temp_d7648e57237443d7b2a28685851d8fd6\raw_review_Appliances...
Cleaning up temporary directory: data\raw\temp_d7648e57237443d7b2a28685851d8fd6
Error loading category Appliances: [WinError 5] Access is denied: 'data\\raw\\temp_d7648e5723744
Extracting data\raw\raw_review_Arts_Crafts_and_Sewing.tar.gz to data\raw\temp_d4f3e79d978a4963ad
```




Local Disk (D:) › Jaheem › jaheemProjetToDelete › COMP3610_bigDataAnalytics_A3-master › src › data › raw
Name
Date modified
Type
Size
raw_meta_Electronics.tar 4/20/2025 7:01 PM WinRAR archive 1,294,023 KB
raw_meta_Gift_Cards.tar 4/20/2025 7:01 PM WinRAR archive 330 KB
raw_meta_Grocery_and_Gourmet_Food.tar 4/20/2025 7:10 PM WinRAR archive 326,300 KB
raw_meta_Handmade_Products.tar 4/20/2025 7:11 PM WinRAR archive 92,295 KB
raw_meta_Health_and_Household.tar 4/20/2025 7:28 PM WinRAR archive 617,086 KB
raw_meta_Health_and_Personal_Care.tar 4/20/2025 7:29 PM WinRAR archive 23,943 KB
raw_meta_Home_and_Kitchen.tar 4/21/2025 3:24 AM WinRAR archive 2,891,086 KB
raw_meta_Industrial_and_Scientific.tar 4/20/2025 8:14 PM WinRAR archive 281,612 KB
raw_meta_Kindle_Store.tar 4/20/2025 10:16 PM WinRAR archive 2,281,380 KB
raw_meta_Magazine_Subscriptions.tar 4/20/2025 8:21 PM WinRAR archive 705 KB
raw_meta_Movies_and_TV.tar 4/20/2025 10:29 PM WinRAR archive 240,924 KB
raw_meta_Musical_Instruments.tar 4/20/2025 10:31 PM WinRAR archive 152,711 KB
raw_meta_Office_Products.tar 4/20/2025 10:41 PM WinRAR archive 520,549 KB
raw_meta_Patio_Lawn_and_Garden.tar 4/20/2025 10:55 PM WinRAR archive 679,057 KB
raw_meta_Pet_Supplies.tar 4/20/2025 11:08 PM WinRAR archive 386,519 KB
raw_meta_Software.tar 4/20/2025 11:10 PM WinRAR archive 64,480 KB
raw_meta_Sports_and_Outdoors.tar 4/20/2025 11:28 PM WinRAR archive 1,034,728 KB
raw_meta_Subscription_Boxes.tar 4/20/2025 11:28 PM WinRAR archive 272 KB
raw_meta_Tools_and_Home_Improveme... 4/20/2025 11:53 PM WinRAR archive 1,180,315 KB
raw_meta_Toys_and_Games.tar 4/21/2025 12:05 AM WinRAR archive 647,421 KB
raw_meta_Unknown.tar 4/21/2025 12:59 AM WinRAR archive 131,578 KB
raw_meta_Video_Games.tar 4/21/2025 12:08 AM WinRAR archive 100,986 KB
raw_review_All_Beauty.tar 4/20/2025 3:25 PM WinRAR archive 83,804 KB
raw_review_Amazon_Fashion.tar 4/20/2025 3:26 PM WinRAR archive 260,461 KB
raw_review_Appliances.tar 4/20/2025 3:29 PM WinRAR archive 241,869 KB
raw_review_Arts_Crafts_and_Sewing.tar 4/20/2025 3:34 PM WinRAR archive 936,707 KB
raw_review_Automotive.tar 4/20/2025 3:47 PM WinRAR archive 2,095,616 KB
raw_review_Baby_Products.tar 4/20/2025 3:57 PM WinRAR archive 764,147 KB
raw_review_Beauty_and_Personal_Care.tar 4/20/2025 4:11 PM WinRAR archive 2,707,747 KB
raw_review_Books.tar 4/20/2025 4:41 PM WinRAR archive 5,777,609 KB
raw_review_CDs_and_Vinyl.tar 4/20/2025 5:07 PM WinRAR archive 960,202 KB
raw_review_Cell_Phones_and_Accessorie... 4/20/2025 5:20 PM WinRAR archive 2,298,752 KB
raw_review_Clothing_Shoes_and_Jewelry.... 4/20/2025 5:59 PM WinRAR archive 6,534,426 KB
raw_review_Digital_Music.tar 4/20/2025 6:24 PM WinRAR archive 23,779 KB
raw_review_Electronics.tar 4/20/2025 6:54 PM WinRAR archive 5,959,724 KB
raw_review_Gift_Cards.tar 4/20/2025 7:01 PM WinRAR archive 11,340 KB
raw_review_Grocery_and_Gourmet_Food.... 4/20/2025 7:08 PM WinRAR archive 1,486,966 KB
raw_review_Handmade_Products.tar 4/20/2025 7:10 PM WinRAR archive 68,765 KB
raw_review_Health_and_Household.tar 4/20/2025 7:25 PM WinRAR archive 2,921,986 KB
raw_review_Health_and_Personal_Care.tar 4/20/2025 7:29 PM WinRAR archive 61,933 KB
raw_review_Home_and_Kitchen.tar 4/20/2025 8:09 PM WinRAR archive 7,628,065 KB
raw_review_Industrial_and_Scientific.tar 4/20/2025 8:13 PM WinRAR archive 612,613 KB
raw_review_Kindle_Store.tar 4/20/2025 10:05 PM WinRAR archive 4,244,069 KB
raw_review_Magazine_Subscriptions.tar 4/20/2025 8:21 PM WinRAR archive 8,990 KB
raw_review_Movies_and_TV.tar 4/20/2025 10:27 PM WinRAR archive 2,214,343 KB
raw_review_Musical_Instruments.tar 4/20/2025 10:31 PM WinRAR archive 419,405 KB
raw_review_Office_Products.tar 4/20/2025 10:39 PM WinRAR archive 1,469,244 KB
raw_review_Patio_Lawn_and_Garden.tar 4/20/2025 10:51 PM WinRAR archive 1,967,189 KB
raw_review_Pet_Supplies.tar 4/20/2025 11:06 PM WinRAR archive 2,136,701 KB
raw_review_Software.tar 4/20/2025 11:10 PM WinRAR archive 447,094 KB
raw_review_Sports_and_Outdoors.tar 4/20/2025 11:22 PM WinRAR archive 2,410,557 KB
raw_review_Subscription_Boxes.tar 4/20/2025 11:28 PM WinRAR archive 2,439 KB
raw_review_Tools_and_Home_Improvem... 4/20/2025 11:46 PM WinRAR archive 3,212,574 KB
raw_review_Toys_and_Games.tar 4/21/2025 12:01 AM WinRAR archive 1,776,841 KB
raw_review_Unknown.tar 4/21/2025 12:58 AM WinRAR archive 7,621,290 KB
raw_review_Video_Games.tar 4/21/2025 12:08 AM WinRAR archive 748,764 KB

Data Cleaning & Preprocessing

The data was cleaned and prepared to be used for further analysis. This involved merging related files, removing missing or invalid values, eliminating duplicates, creating new columns, and organizing everything in a consistent format.

First, review data was matched with product metadata using the 'parent_asin' field. This ensured each entry included all key details such as 'user_id', 'asin', 'rating', 'text', 'verified_purchase', 'helpful_vote', and product information like 'brand' and 'category'.

Next, rows with missing ratings, ratings outside the valid range of 1 to 5, or empty review text were removed. If brand information was not available in either the 'details' or 'store' fields, it was replaced with 'Unknown' to avoid missing values.

Duplicate entries were also removed. If the same combination of 'user_id', 'asin', and 'text' appeared more than once, only the first instance was kept to prevent redundancy.

New columns were added to enhance the dataset. A 'review_length' column was calculated by counting the number of words in the review text. If a valid 'timestamp' was available, it was converted from milliseconds to seconds and used to extract a 'year' column for time-based analysis. Invalid timestamps were identified and removed where necessary.

Two approaches were used to consolidate the dataset. The first approach used 'pandas' to combine all categories into one unified CSV file, making it easy to load the entire dataset at once. The second approach used 'DuckDB' to process and store each category separately as individual 'parquet' files. This method allowed for efficient querying and storage without needing to combine all the data into one file.

Code

Dataframe Approach:

Cleaning and Merging Data:

```
def clean_dataframe(reviews_df, meta_df):
    for df in [reviews_df, meta_df]:
        df["parent_asin"] =
df["parent_asin"].astype(str).str.strip()
        df["parent_asin"] = df["parent_asin"].apply(lambda x:
unicodedata.normalize("NFKC", x))

    merged_df = reviews_df.merge(meta_df, on="parent_asin",
how="inner")
    merged_df = merged_df[merged_df['rating'].between(1, 5)]
    merged_df.dropna(subset=['text'], inplace=True)
    merged_df['store'] = merged_df['store'].fillna("Unknown")
    merged_df.drop_duplicates(subset=['user_id', 'asin', 'text'],
keep='first', inplace=True)
    merged_df['review_length'] = merged_df['text'].apply(lambda x:
len(re.findall(r'\w+', str(x))))
    if "timestamp" in merged_df.columns:
        merged_df["timestamp"] =
pd.to_numeric(merged_df["timestamp"], errors="coerce")
        merged_df["timestamp"] = merged_df["timestamp"] // 1000
        invalid_timestamps = merged_df[
            (merged_df["timestamp"] <= 0) | (merged_df["timestamp"]
>= 2**31)
        ]

        if not invalid_timestamps.empty:
            print("\n=== Invalid Timestamps Found ===")
            print(invalid_timestamps[["timestamp"]].head(10))
        merged_df = merged_df[
            (merged_df["timestamp"] > 0) & (merged_df["timestamp"] <
2**31)
        ]

        print("\n=== Valid Timestamp Count ===", len(merged_df))
        if not merged_df.empty:
            merged_df["year"] = pd.to_datetime(
                merged_df["timestamp"], unit="s", errors="coerce"
```

```
            ).dt.year
        else:
            print("No valid timestamps left after filtering.
Skipping year extraction.")
    else:
        print("No timestamp column found.")
    return merged_df
```

Duckdb Aproach:

Auto-detecting Categories:

```
review_tars = glob.glob(os.path.join(base_path,
"raw_review_*.tar.gz"))
categories = []

for review_tar in review_tars:
    cat = os.path.basename(review_tar).replace("raw_review_",
"").replace(".tar.gz", "")
    meta_tar = os.path.join(base_path, f"raw_meta_{cat}.tar.gz")
    if os.path.exists(meta_tar):
        categories.append(cat)
```

Creating DuckDB Tables:

```
def create_tables():
    duckdb.sql("DROP TABLE IF EXISTS review_stream")
    duckdb.sql("DROP TABLE IF EXISTS meta_stream")
```

Extracting and Streaming Arrow Files:

```
def stream_from_tar_to_duckdb(table_name, tar_path):
    print(f"🔄 Processing tar file: {tar_path}")
    with tarfile.open(tar_path, 'r:gz') as tar:
        for member in tar.getmembers():
            if member.name.endswith(".arrow"):
```

Processing Each Category:

```
for category in categories:
    print(f"\n🔄 Processing category: {category}")
    review_tar = os.path.join(base_path,
f"raw_review_{category}.tar.gz")
    meta_tar   = os.path.join(base_path,
f"raw_meta_{category}.tar.gz")
    output_path = os.path.join(base_path,
f"joined_{category}.parquet")
```

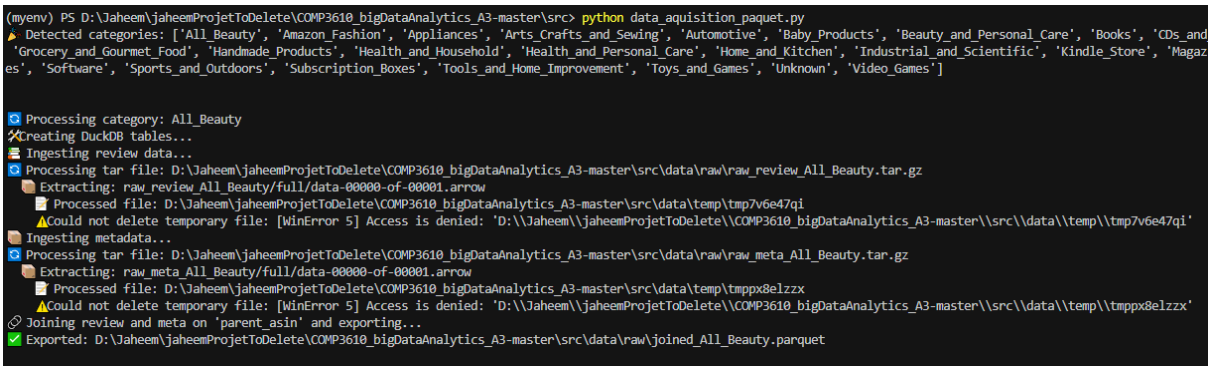
Clean Parquet Function:

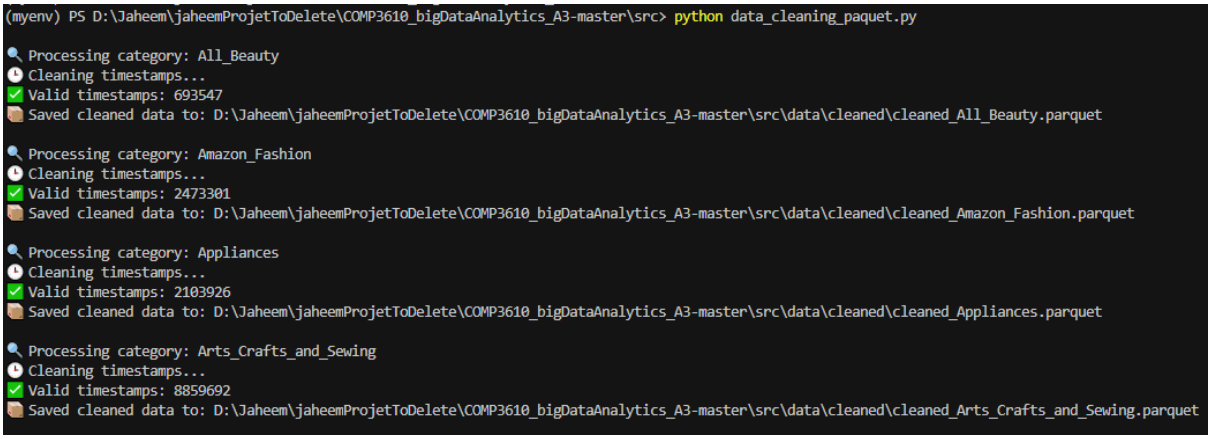
```
def clean_parquet(path, output_path=None):
     query = f"""
    SELECT DISTINCT ON (user_id, asin, text)
        category,
        rating,
        title,
        title_1 AS product_title,
        text,
        user_id,
        asin,
        parent_asin,
        timestamp,
        verified_purchase,
        helpful_vote,
        average_rating,
        rating_number,
        CASE
            WHEN try_cast(details AS JSON) IS NOT NULL
                  AND json_extract_string(details, '$.brand') IS NOT
NULL
                THEN TRIM(json_extract_string(details, '$.brand'))
            WHEN store IS NOT NULL THEN TRIM(store)
            ELSE 'Unknown'
        END AS brand,
        main_category,
        store,
        price,
        LENGTH(regexp_split_to_array(text, '\\W+')) AS review_length
    FROM '{path}'
    WHERE rating BETWEEN 1 AND 5
```

```
      AND text IS NOT NULL AND LENGTH(TRIM(text)) > 0
    """
```

Processing Merged Parquet Files:

```
if not merged_files:
    print("⚠️ No merged parquet files found.")
else:
    for file_path in merged_files:
        category = Path(file_path).stem.replace("joined_", "")
        output_file = cleaned_dir / f"cleaned_{category}.parquet"
        print(f"\n🔍 Processing category: {category}")
        clean_parquet(file_path, output_file)
```

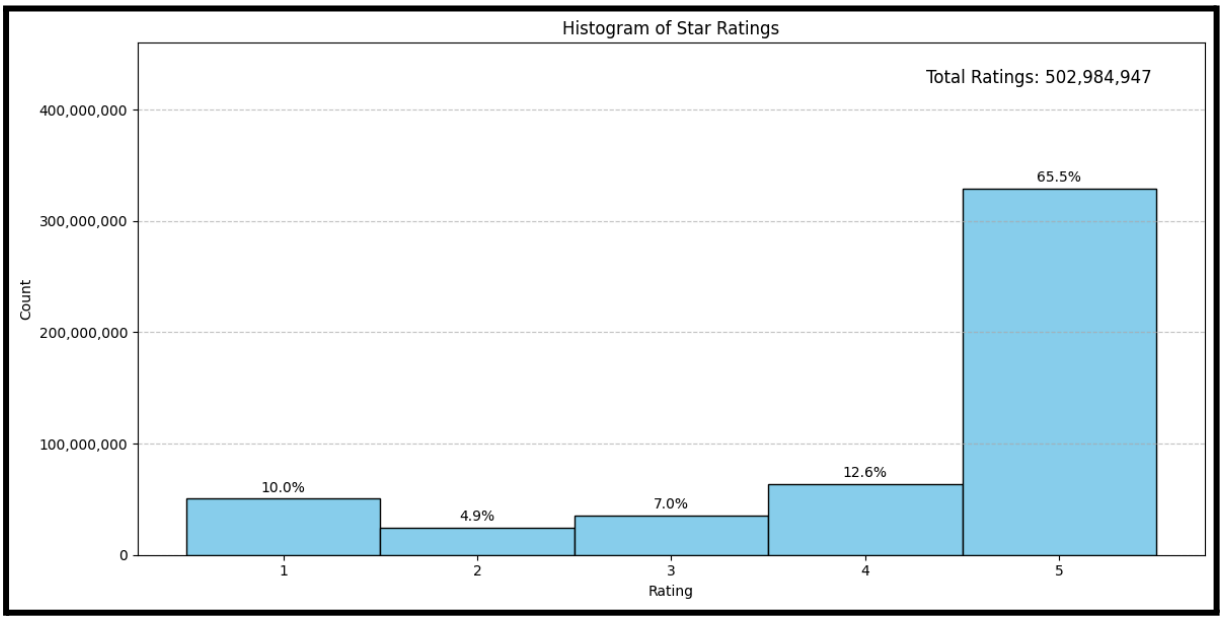





(myenv) PS D:\Jaheem\jaheemProjetToDelete\COMP3610_bigDataAnalytics_A3-master\src> python data_aquisition_paquet.py
Detected categories: ['All_Beauty', 'Amazon_Fashion', 'Appliances', 'Arts_Crafts_and_Sewing', 'Automotive', 'Baby_Products', 'Beauty_and_Personal_Care', 'Books', 'CDs_and
'Grocery_and_Gourmet_Food', 'Handmade_Products', 'Health_and_Household', 'Health_and_Personal_Care', 'Home_and_Kitchen', 'Industrial_and_Scientific', 'Kindle_Store', 'Magaz
es', 'Software', 'Sports_and_Outdoors', 'Subscription_Boxes', 'Tools_and_Home_Improvement', 'Toys_and_Games', 'Unknown', 'Video_Games']
Processing category: All_Beauty
Creating DuckDB tables...
Ingesting review data...
Processing tar file: D:\Jaheem\jaheemProjetToDelete\COMP3610_bigDataAnalytics_A3-master\src\data\raw\raw_review_All_Beauty.tar.gz
Extracting: raw_review_All_Beauty/full/data-00000-of-00001.arrow
Processed file: D:\Jaheem\jaheemProjetToDelete\COMP3610_bigDataAnalytics_A3-master\src\data\temp\tmp7v6e47qi
Could not delete temporary file: [WinError 5] Access is denied: 'D:\\Jaheem\\jaheemProjetToDelete\\COMP3610_bigDataAnalytics_A3-master\\src\\data\\temp\\tmp7v6e47qi'
Ingesting metadata...
Processing tar file: D:\Jaheem\jaheemProjetToDelete\COMP3610_bigDataAnalytics_A3-master\src\data\raw\raw_meta_All_Beauty.tar.gz
Extracting: raw_meta_All_Beauty/full/data-00000-of-00001.arrow
Processed file: D:\Jaheem\jaheemProjetToDelete\COMP3610_bigDataAnalytics_A3-master\src\data\temp\tmppx8elzzx
Could not delete temporary file: [WinError 5] Access is denied: 'D:\\Jaheem\\jaheemProjetToDelete\\COMP3610_bigDataAnalytics_A3-master\\src\\data\\temp\\tmppx8elzzx'
Joining review and meta on 'parent_asin' and exporting...
Exported: D:\Jaheem\jaheemProjetToDelete\COMP3610_bigDataAnalytics_A3-master\src\data\raw\joined_All_Beauty.parquet
Processing category: Amazon_Fashion
Creating DuckDB tables...
Ingesting review data...
Processing tar file: D:\Jaheem\jaheemProjetToDelete\COMP3610_bigDataAnalytics_A3-master\src\data\raw\raw_review_Amazon_Fashion.tar.gz
Extracting: raw_review_Amazon_Fashion/full/data-00000-of-00002.arrow
Processed file: D:\Jaheem\jaheemProjetToDelete\COMP3610_bigDataAnalytics_A3-master\src\data\temp\tmpv6biln1h
Could not delete temporary file: [WinError 5] Access is denied: 'D:\\Jaheem\\jaheemProjetToDelete\\COMP3610_bigDataAnalytics_A3-master\\src\\data\\temp\\tmpv6biln1h'
Extracting: raw_review_Amazon_Fashion/full/data-00001-of-00002.arrow
Processed file: D:\Jaheem\jaheemProjetToDelete\COMP3610_bigDataAnalytics_A3-master\src\data\temp\tmpcwx2w582
Could not delete temporary file: [WinError 5] Access is denied: 'D:\\Jaheem\\jaheemProjetToDelete\\COMP3610_bigDataAnalytics_A3-master\\src\\data\\temp\\tmpcwx2w582'
Ingesting metadata...
Processing tar file: D:\Jaheem\jaheemProjetToDelete\COMP3610_bigDataAnalytics_A3-master\src\data\raw\raw_meta_Amazon_Fashion.tar.gz
Extracting: raw_meta_Amazon_Fashion/full/data-00000-of-00003.arrow
Processed file: D:\Jaheem\jaheemProjetToDelete\COMP3610_bigDataAnalytics_A3-master\src\data\temp\tmppxjlptq1
Could not delete temporary file: [WinError 5] Access is denied: 'D:\\Jaheem\\jaheemProjetToDelete\\COMP3610_bigDataAnalytics_A3-master\\src\\data\\temp\\tmppxjlptq1'
Extracting: raw_meta_Amazon_Fashion/full/data-00001-of-00003.arrow
Processed file: D:\Jaheem\jaheemProjetToDelete\COMP3610_bigDataAnalytics_A3-master\src\data\temp\tmpj_fpdhbd
Could not delete temporary file: [WinError 5] Access is denied: 'D:\\Jaheem\\jaheemProjetToDelete\\COMP3610_bigDataAnalytics_A3-master\\src\\data\\temp\\tmpj_fpdhbd'
Extracting: raw_meta_Amazon_Fashion/full/data-00002-of-00003.arrow
Processed file: D:\Jaheem\jaheemProjetToDelete\COMP3610_bigDataAnalytics_A3-master\src\data\temp\tmp2lojob61
Could not delete temporary file: [WinError 5] Access is denied: 'D:\\Jaheem\\jaheemProjetToDelete\\COMP3610_bigDataAnalytics_A3-master\\src\\data\\temp\\tmp2lojob61'
Joining review and meta on 'parent_asin' and exporting...
Exported: D:\Jaheem\jaheemProjetToDelete\COMP3610_bigDataAnalytics_A3-master\src\data\raw\joined_Amazon_Fashion.parquet
Processing category: Appliances
Creating DuckDB tables...
Ingesting review data...
Processing tar file: D:\Jaheem\jaheemProjetToDelete\COMP3610_bigDataAnalytics_A3-master\src\data\raw\raw_review_Appliances.tar.gz
Extracting: raw_review_Appliances/full/data-00000-of-00002.arrow
Processed file: D:\Jaheem\jaheemProjetToDelete\COMP3610_bigDataAnalytics_A3-master\src\data\temp\tmp3_a09s8r
Could not delete temporary file: [WinError 5] Access is denied: 'D:\\Jaheem\\jaheemProjetToDelete\\COMP3610_bigDataAnalytics_A3-master\\src\\data\\temp\\tmp3_a09s8r'
Extracting: raw_review_Appliances/full/data-00001-of-00002.arrow
Processed file: D:\Jaheem\jaheemProjetToDelete\COMP3610_bigDataAnalytics_A3-master\src\data\temp\tmpad26p3l_
Could not delete temporary file: [WinError 5] Access is denied: 'D:\\Jaheem\\jaheemProjetToDelete\\COMP3610_bigDataAnalytics_A3-master\\src\\data\\temp\\tmpad26p3l_'
Ingesting metadata...
Processing tar file: D:\Jaheem\jaheemProjetToDelete\COMP3610_bigDataAnalytics_A3-master\src\data\raw\raw_meta_Appliances.tar.gz
Extracting: raw_meta_Appliances/full/data-00000-of-00001.arrow
Processed file: D:\Jaheem\jaheemProjetToDelete\COMP3610_bigDataAnalytics_A3-master\src\data\temp\tmp8r08wc8y
Could not delete temporary file: [WinError 5] Access is denied: 'D:\\Jaheem\\jaheemProjetToDelete\\COMP3610_bigDataAnalytics_A3-master\\src\\data\\temp\\tmp8r08wc8y'
Joining review and meta on 'parent_asin' and exporting...
Exported: D:\Jaheem\jaheemProjetToDelete\COMP3610_bigDataAnalytics_A3-master\src\data\raw\joined_Appliances.parquet
Processing category: Arts_Crafts_and_Sewing
Creating DuckDB tables...
Ingesting review data...
Processing tar file: D:\Jaheem\jaheemProjetToDelete\COMP3610_bigDataAnalytics_A3-master\src\data\raw\raw_review_Arts_Crafts_and_Sewing.tar.gz
Extracting: raw_review_Arts_Crafts_and_Sewing/full/data-00000-of-00006.arrow

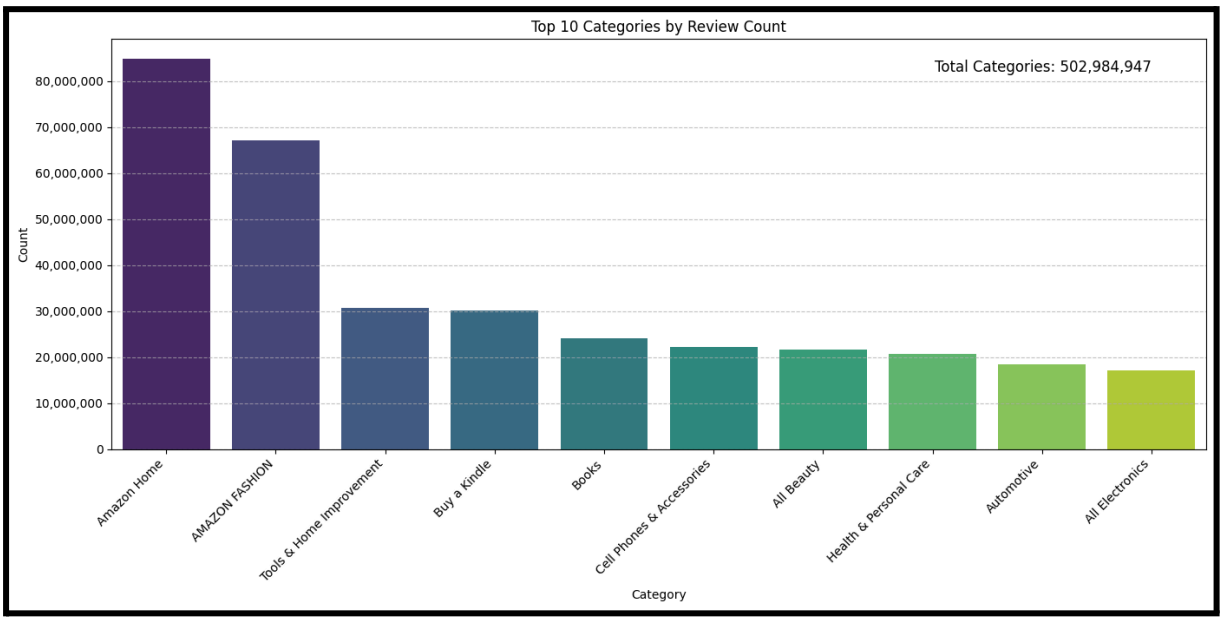
EDA

Star Rating Histogram



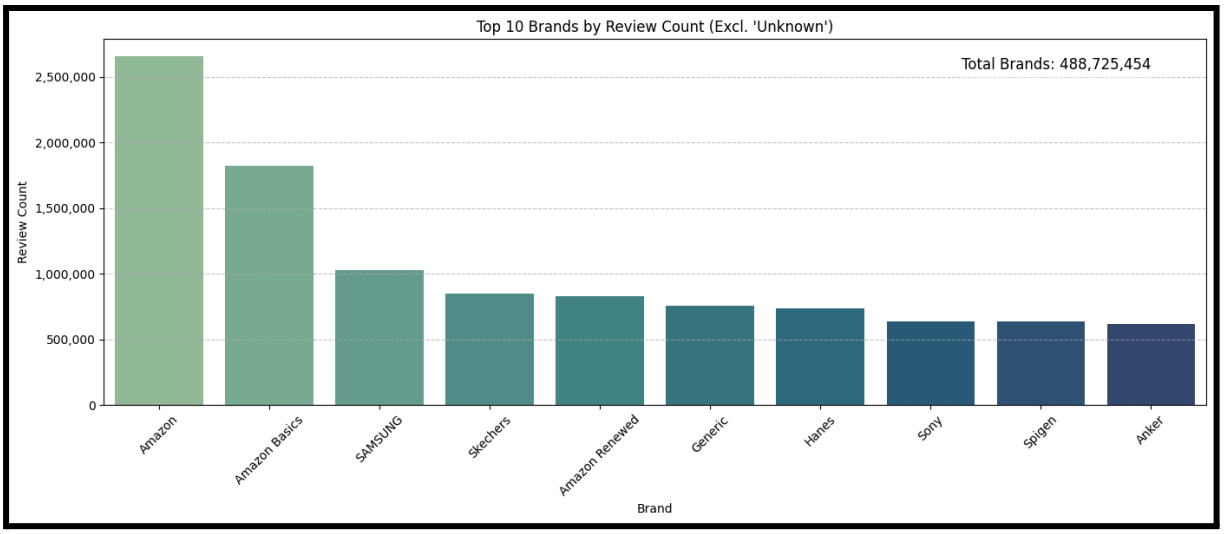
The star rating distribution was analyzed by plotting a histogram for ratings between 1 and 5. Using seaborn's histplot with five bins the visualization showed how customer ratings were spread across the scale. This helped to quickly identify whether ratings were skewed toward positive, negative, or neutral reviews, providing an overview of customer satisfaction in the dataset. From the graph which was plotted it can be seen that the highest peak is at the 5 star mark at 65.5% and lowest is at the 2 star mark at 4.9% and the star ratings of 1, 3 and 4 are just above the 2 star mark at 10.0%, 7.0% and 12.6% respectively..

Bar Chart for Top 10 Categories



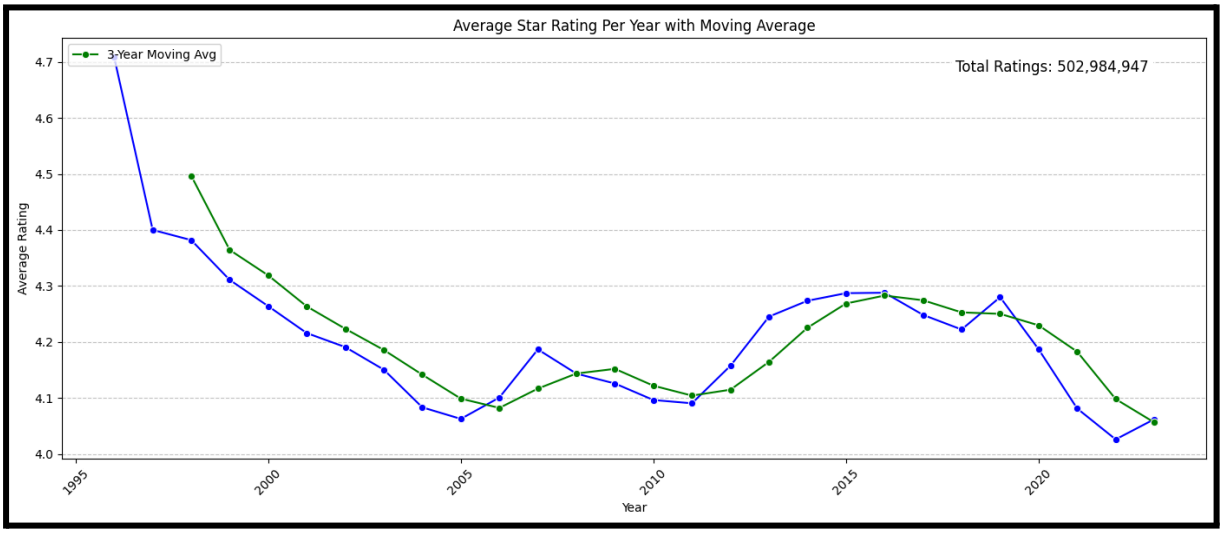
To find the most reviewed product categories, the main_category column was analyzed by counting the number of reviews for each category. The ten categories with the highest review counts were selected and visualized with a bar chart. This allowed us to see which types of products were most popular or received the most customer engagement across the dataset. This bar chart shows that the most reviewed category is “Amazon Home”.

Bar Chart for Top 10 Brands



Brand-level analysis was done by extracting brand names from a nested 'details' column, where the information was initially stored as a string. After extracting the brand names and excluding any labeled as "Unknown," the top 10 brands with the most reviews were plotted in a bar chart. This gave insights into brand popularity and customer focus within the data. The most popular brand according to the graph plotted was Frigidaire with GE close behind it.

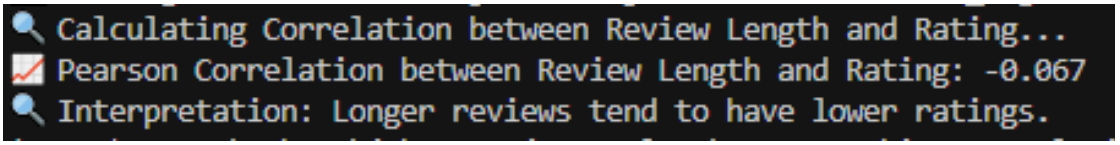
Average Star Rating Per Year with Moving Average



A time-based trend analysis was performed by converting Unix timestamps from the 'timestamp' column into review years. The average star rating was then calculated for each

year and visualized through a line chart. A 3 year moving average was also calculated and shown to further smooth the graph making it more readable and easily understandable. This helped to observe how customer satisfaction changed over time, highlighting any noticeable trends or shifts in overall sentiment. From the line graph the year with the highest average rating was the year 1996 with an average of 4.7 stars. The highest moving average obtained was an average of 4.5 stars in the year 1998. It can also be seen that the average rating has never once dropped below an average of 4.0 stars throughout the years.

Pearson correlation between Review Length and Star Rating



The relationship between review length and star rating was assessed by calculating the Pearson correlation coefficient. The analysis provided a numerical value showing whether longer or shorter reviews tended to be associated with higher or lower ratings. This offered an additional perspective on customer behavior and the nature of feedback depending on sentiment strength. The value which was calculated was approximately -0.67, this number is fairly close to 0 and such shows that there is almost no correlation between the length of the review and the rating. Although the negative sign shows that longer reviews have a very slight tendency to be associated with a lower rating.

Code

Save Plot Function:

```
def save_plot(plot, filename):
    print(f"💾 Saving plot: {filename}")
    save_path =
r"D:\Jaheem\jaheemProjetToDelete\COMP3610_bigDataAnalytics_A3-master
\src\data\processed"
    os.makedirs(save_path, exist_ok=True)
    file_path = os.path.join(save_path, filename)
    plot.savefig(file_path, bbox_inches='tight')
    plot.close()
```

Star Rating Histogram:

```
def run_star_rating_histogram(base_cleaned_path):
    all_ratings = []
    for file in os.listdir(base_cleaned_path):
        if file.endswith(".parquet"):
            file_path = os.path.join(base_cleaned_path, file)
            print(f"✅ Loading file: {file_path}")
            try:
                df = pd.read_parquet(file_path, columns=["rating"])
                all_ratings.append(df["rating"])
            Except:
                print(f"⚠️ Skipping {file}: 'rating' column not
found.")
    if all_ratings:
        print("🔍 Calculating Star Rating Histogram...")
        ratings_series = pd.concat(all_ratings)
        total_ratings = len(ratings_series)
        bins = np.arange(0.5, 6, 1)
        plt.figure(figsize=(12, 6))
        counts, bins, patches = plt.hist(
        ratings_series, bins=bins, edgecolor='black', align='mid',
color='skyblue'
        )
        plt.title("Histogram of Star Ratings")
        plt.xlabel("Rating")
        plt.ylabel("Count")
        plt.xticks([1, 2, 3, 4, 5])
        plt.gca().yaxis.set_major_formatter(mtick.FuncFormatter
(lambda x, _: f'{int(x):,}'))
        plt.grid(True, axis='y', linestyle='--', alpha=0.7)
        y_max = plt.gca().get_ylim()[1]
        plt.ylim(0, y_max * 1.33)
        for count, patch in zip(counts, patches):
            percentage = (count / total_ratings) * 100
             plt.text(patch.get_x() + patch.get_width() / 2, count +
y_max * 0.01,
             f'{percentage:.1f}%', ha='center', va='bottom',
fontsize=10)

        plt.text(0.95, 0.95, f'Total Ratings: {total_ratings:,}',
ha='right', va='top',transform=plt.gca().transAxes, fontsize=12,
```

```
color='black', bbox=dict(facecolor='white', edgecolor='none',
boxstyle='round,pad=0.3'))
        plt.tight_layout()
        save_plot(plt, "star_rating_histogram.png")
```

Top Categories by Review Count:

```
def run_top_categories_by_review_count(base_cleaned_path):
    all_categories = []
    for file in os.listdir(base_cleaned_path):
        if file.endswith(".parquet"):
            file_path = os.path.join(base_cleaned_path, file)
            print(f"✅ Loading file: {file_path}")
            try:
                df = pd.read_parquet(file_path,
columns=["main_category"])
                all_categories.append(df["main_category"])
            except:
                print(f"⚠ Skipping {file}: 'main_category' column
not found.")

    if all_categories:
        print("🔍 Calculating Top Categories by Review Count...")
        categories_series = pd.concat(all_categories)
        top_categories =
categories_series.value_counts().head(10).reset_index()
        top_categories.columns = ['Category', 'Count']
        total_categories = len(categories_series)
        plt.figure(figsize=(14, 7))
        sns.barplot(data=top_categories, x='Category', y='Count',
palette='viridis')
        plt.title("Top 10 Categories by Review Count")
        plt.xlabel("Category")
        plt.ylabel("Count")
        plt.xticks(rotation=45, ha='right')
        plt.gca().yaxis.set_major_formatter(mtick.FuncFormatter
(lambda x, _: f'{int(x):,}'))
        plt.grid(True, axis='y', linestyle='--', alpha=0.7
        plt.text(0.95, 0.95, f'Total Categories:
{total_categories:,}', ha='right', va='top',
```

```
            transform=plt.gca().transAxes, fontsize=12,
color='black',
            bbox=dict(facecolor='white', edgecolor='none',
boxstyle='round,pad=0.3'))
        plt.tight_layout()
        save_plot(plt, "top_categories_by_review_count.png")
```

Top Brands by Review Count:

```
def run_top_brands_by_review_count(base_cleaned_path):
    all_brands = []

    for file in os.listdir(base_cleaned_path):
        if file.endswith(".parquet"):
            file_path = os.path.join(base_cleaned_path, file)
            print(f"✅ Loading file: {file_path}")
            try:
                df = pd.read_parquet(file_path, columns=["brand"])
                df["brand"] = df["brand"].str.strip()
                df = df[df["brand"].notna()]
                df = df[df["brand"].str.lower() != "unknown"]
                all_brands.append(df["brand"])

            except Exception as e:
                print(f"⚠ Skipping {file}: 'brand' column not found
or other error: {e}")

    if all_brands:
        print("🔍 Calculating Top Brands by Review Count...")
        brands_series = pd.concat(all_brands)

        top_brands = brands_series.value_counts().head(10)
        total_brands = len(brands_series)
        plt.figure(figsize=(14, 6))
        sns.barplot(x=top_brands.index, y=top_brands.values,
palette='crest')
        plt.title("Top 10 Brands by Review Count (Excl. 'Unknown')")
        plt.xlabel("Brand")
        plt.ylabel("Review Count")
        plt.xticks(rotation=45)
        plt.gca().yaxis.set_major_formatter(mtick.FuncFormatter
```

```
(lambda x, _: f'{int(x):,}'))
        plt.grid(True, axis='y', linestyle='--', alpha=0.7)
        plt.text(0.95, 0.95, f'Total Brands: {total_brands:,}',
ha='right', va='top',
             transform=plt.gca().transAxes, fontsize=12,
color='black',
             bbox=dict(facecolor='white', edgecolor='none',
boxstyle='round,pad=0.3'))
        plt.tight_layout()
        save_plot(plt, "top_brands_by_review_count.png")
    else:
        print("⚠ No valid brands found across files.")
```

Average Star Rating Per Year:

```
def run_avg_star_rating_per_year(base_cleaned_path):
    all_years_and_ratings = []

    for file in os.listdir(base_cleaned_path):
        if file.endswith(".parquet"):
            file_path = os.path.join(base_cleaned_path, file)
            print(f"✅ Loading file: {file_path}")
            try:
                df = pd.read_parquet(file_path,
columns=["timestamp", "rating"])
                if "timestamp" in df.columns:
                    df["timestamp"] = pd.to_numeric(df["timestamp"],
errors="coerce") // 1000
                    df = df[(df["timestamp"] > 0) & (df["timestamp"]
< 2**31)]
                    df["review_year"] =
pd.to_datetime(df["timestamp"], unit="s", errors="coerce").dt.year
                    df = df.dropna(subset=["review_year", "rating"])
                    df["review_year"] =
df["review_year"].astype(int)
                    all_years_and_ratings.append(df[["review_year",
"rating"]])
            except:
                print(f"⚠ Skipping {file}: 'timestamp' or 'rating'
column not found or unreadable.")
```

```
    if all_years_and_ratings:
        print("🔍 Calculating Average Star Rating Per Year...")
        combined_df = pd.concat(all_years_and_ratings)
        avg_rating_by_year =
combined_df.groupby("review_year")["rating"].mean().reset_index()
        avg_rating_by_year['rolling_avg'] =
avg_rating_by_year['rating'].rolling(window=3).mean()
        total_ratings = len(combined_df)
        plt.figure(figsize=(14, 6))
        sns.lineplot(data=avg_rating_by_year, x='review_year',
y='rating', marker='o', color='b')
        sns.lineplot(data=avg_rating_by_year, x='review_year',
y='rolling_avg', marker='o', label='3-Year Moving Avg', color='g')
        plt.title("Average Star Rating Per Year with Moving
Average")
        plt.xlabel("Year")
        plt.ylabel("Average Rating")
        plt.grid(True, axis='y', linestyle='--', alpha=0.7)
        plt.xticks(rotation=45)
        plt.legend(loc='upper left')
        plt.text(0.95, 0.95, f'Total Ratings: {total_ratings:,}',
ha='right', va='top',
            transform=plt.gca().transAxes, fontsize=12,
color='black',
            bbox=dict(facecolor='white', edgecolor='none',
boxstyle='round,pad=0.3'))
        plt.tight_layout()
        save_plot(plt,
"avg_star_rating_per_year_with_rolling_avg.png")
```

Pearson Correlation Coefficient:

```
def run_correlation_review_length_rating(base_cleaned_path):
    all_reviews = []
    for file in os.listdir(base_cleaned_path):
        if file.endswith(".parquet"):
            file_path = os.path.join(base_cleaned_path, file)
            print(f"✅ Loading file: {file_path}")
            try:
```

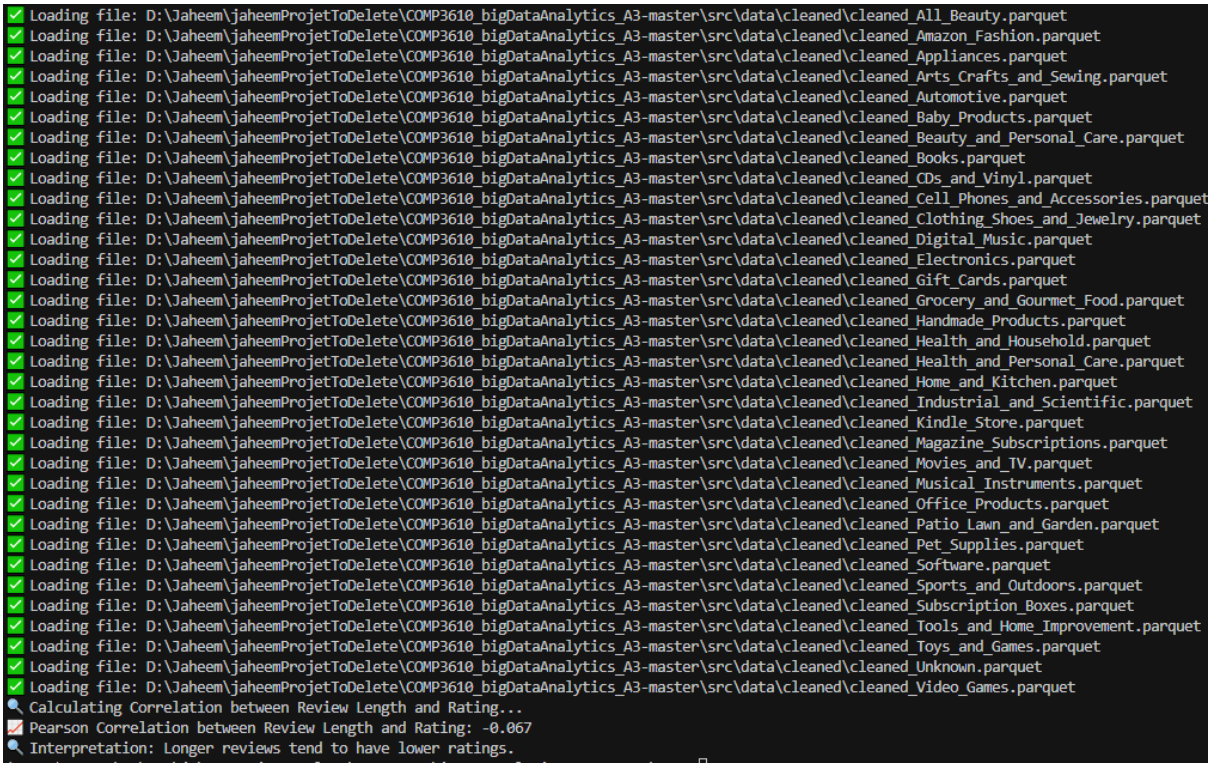
```
                df = pd.read_parquet(file_path,
columns=["review_length", "rating"])
                all_reviews.append(df[["review_length", "rating"]])
            except:
                    print(f"⚠ Skipping {file}: 'review_length' or
'rating' column not found.")

    if all_reviews:
         print("🔍 Calculating Correlation between Review Length and
Rating...")
        combined_df = pd.concat(all_reviews)
        corr =
combined_df["review_length"].corr(combined_df["rating"])
        print(f"📈 Pearson Correlation between Review Length and
Rating: {corr:.3f}")
        if corr > 0:
            print("🔍 Interpretation: Longer reviews tend to have
higher ratings.")
        elif corr < 0:
            print("🔍 Interpretation: Longer reviews tend to have
lower ratings.")
        else:
            print("🔍 Interpretation: No correlation between review
length and rating.")
```

```
(myenv) PS D:\Jaheem\jaheemProjetToDelete\COMP3610_bigDataAnalytics_A3-master\src> python testingEDA.py
📂 Starting to run EDA on all cleaned data...
✅ Loading file: D:\Jaheem\jaheemProjetToDelete\COMP3610_bigDataAnalytics_A3-master\src\data\cleaned\cleaned_All_Beauty.parquet
✅ Loading file: D:\Jaheem\jaheemProjetToDelete\COMP3610_bigDataAnalytics_A3-master\src\data\cleaned\cleaned_Amazon_Fashion.parquet
✅ Loading file: D:\Jaheem\jaheemProjetToDelete\COMP3610_bigDataAnalytics_A3-master\src\data\cleaned\cleaned_Appliances.parquet
✅ Loading file: D:\Jaheem\jaheemProjetToDelete\COMP3610_bigDataAnalytics_A3-master\src\data\cleaned\cleaned_Arts_Crafts_and_Sewing.parquet
✅ Loading file: Open file in editor (ctrl + click)te\COMP3610_bigDataAnalytics_A3-master\src\data\cleaned\cleaned_Automotive.parquet
✅ Loading file: D:\Jaheem\jaheemProjetToDelete\COMP3610_bigDataAnalytics_A3-master\src\data\cleaned\cleaned_Baby_Products.parquet
✅ Loading file: D:\Jaheem\jaheemProjetToDelete\COMP3610_bigDataAnalytics_A3-master\src\data\cleaned\cleaned_Beauty_and_Personal_Care.parquet
✅ Loading file: D:\Jaheem\jaheemProjetToDelete\COMP3610_bigDataAnalytics_A3-master\src\data\cleaned\cleaned_Books.parquet
✅ Loading file: D:\Jaheem\jaheemProjetToDelete\COMP3610_bigDataAnalytics_A3-master\src\data\cleaned\cleaned_CDs_and_Vinyl.parquet
✅ Loading file: D:\Jaheem\jaheemProjetToDelete\COMP3610_bigDataAnalytics_A3-master\src\data\cleaned\cleaned_Cell_Phones_and_Accessories.parquet
✅ Loading file: D:\Jaheem\jaheemProjetToDelete\COMP3610_bigDataAnalytics_A3-master\src\data\cleaned\cleaned_Clothing_Shoes_and_Jewelry.parquet
✅ Loading file: D:\Jaheem\jaheemProjetToDelete\COMP3610_bigDataAnalytics_A3-master\src\data\cleaned\cleaned_Digital_Music.parquet
✅ Loading file: D:\Jaheem\jaheemProjetToDelete\COMP3610_bigDataAnalytics_A3-master\src\data\cleaned\cleaned_Electronics.parquet
✅ Loading file: D:\Jaheem\jaheemProjetToDelete\COMP3610_bigDataAnalytics_A3-master\src\data\cleaned\cleaned_Gift_Cards.parquet
✅ Loading file: D:\Jaheem\jaheemProjetToDelete\COMP3610_bigDataAnalytics_A3-master\src\data\cleaned\cleaned_Grocery_and_Gourmet_Food.parquet
✅ Loading file: D:\Jaheem\jaheemProjetToDelete\COMP3610_bigDataAnalytics_A3-master\src\data\cleaned\cleaned_Handmade_Products.parquet
✅ Loading file: D:\Jaheem\jaheemProjetToDelete\COMP3610_bigDataAnalytics_A3-master\src\data\cleaned\cleaned_Health_and_Household.parquet
✅ Loading file: D:\Jaheem\jaheemProjetToDelete\COMP3610_bigDataAnalytics_A3-master\src\data\cleaned\cleaned_Health_and_Personal_Care.parquet
✅ Loading file: D:\Jaheem\jaheemProjetToDelete\COMP3610_bigDataAnalytics_A3-master\src\data\cleaned\cleaned_Home_and_Kitchen.parquet
✅ Loading file: D:\Jaheem\jaheemProjetToDelete\COMP3610_bigDataAnalytics_A3-master\src\data\cleaned\cleaned_Industrial_and_Scientific.parquet
✅ Loading file: D:\Jaheem\jaheemProjetToDelete\COMP3610_bigDataAnalytics_A3-master\src\data\cleaned\cleaned_Kindle_Store.parquet
✅ Loading file: D:\Jaheem\jaheemProjetToDelete\COMP3610_bigDataAnalytics_A3-master\src\data\cleaned\cleaned_Magazine_Subscriptions.parquet
✅ Loading file: D:\Jaheem\jaheemProjetToDelete\COMP3610_bigDataAnalytics_A3-master\src\data\cleaned\cleaned_Movies_and_TV.parquet
✅ Loading file: D:\Jaheem\jaheemProjetToDelete\COMP3610_bigDataAnalytics_A3-master\src\data\cleaned\cleaned_Musical_Instruments.parquet
✅ Loading file: D:\Jaheem\jaheemProjetToDelete\COMP3610_bigDataAnalytics_A3-master\src\data\cleaned\cleaned_Office_Products.parquet
✅ Loading file: D:\Jaheem\jaheemProjetToDelete\COMP3610_bigDataAnalytics_A3-master\src\data\cleaned\cleaned_Patio_Lawn_and_Garden.parquet
✅ Loading file: D:\Jaheem\jaheemProjetToDelete\COMP3610_bigDataAnalytics_A3-master\src\data\cleaned\cleaned_Pet_Supplies.parquet
✅ Loading file: D:\Jaheem\jaheemProjetToDelete\COMP3610_bigDataAnalytics_A3-master\src\data\cleaned\cleaned_Software.parquet
✅ Loading file: D:\Jaheem\jaheemProjetToDelete\COMP3610_bigDataAnalytics_A3-master\src\data\cleaned\cleaned_Sports_and_Outdoors.parquet
✅ Loading file: D:\Jaheem\jaheemProjetToDelete\COMP3610_bigDataAnalytics_A3-master\src\data\cleaned\cleaned_Subscription_Boxes.parquet
✅ Loading file: D:\Jaheem\jaheemProjetToDelete\COMP3610_bigDataAnalytics_A3-master\src\data\cleaned\cleaned_Tools_and_Home_Improvement.parquet
✅ Loading file: D:\Jaheem\jaheemProjetToDelete\COMP3610_bigDataAnalytics_A3-master\src\data\cleaned\cleaned_Toys_and_Games.parquet
✅ Loading file: D:\Jaheem\jaheemProjetToDelete\COMP3610_bigDataAnalytics_A3-master\src\data\cleaned\cleaned_Unknown.parquet
✅ Loading file: D:\Jaheem\jaheemProjetToDelete\COMP3610_bigDataAnalytics_A3-master\src\data\cleaned\cleaned_Video_Games.parquet
🔍 Calculating Star Rating Histogram...
💾 Saving plot: star_rating_histogram.png
```

```
✅ Loading file: D:\Jaheem\jaheemProjetToDelete\COMP3610_bigDataAnalytics_A3-master\src\data\cleaned\cleaned_All_Beauty.parquet
✅ Loading file: D:\Jaheem\jaheemProjetToDelete\COMP3610_bigDataAnalytics_A3-master\src\data\cleaned\cleaned_Amazon_Fashion.parquet
✅ Loading file: D:\Jaheem\jaheemProjetToDelete\COMP3610_bigDataAnalytics_A3-master\src\data\cleaned\cleaned_Appliances.parquet
✅ Loading file: D:\Jaheem\jaheemProjetToDelete\COMP3610_bigDataAnalytics_A3-master\src\data\cleaned\cleaned_Arts_Crafts_and_Sewing.parquet
✅ Loading file: D:\Jaheem\jaheemProjetToDelete\COMP3610_bigDataAnalytics_A3-master\src\data\cleaned\cleaned_Automotive.parquet
✅ Loading file: D:\Jaheem\jaheemProjetToDelete\COMP3610_bigDataAnalytics_A3-master\src\data\cleaned\cleaned_Baby_Products.parquet
✅ Loading file: D:\Jaheem\jaheemProjetToDelete\COMP3610_bigDataAnalytics_A3-master\src\data\cleaned\cleaned_Beauty_and_Personal_Care.parquet
✅ Loading file: D:\Jaheem\jaheemProjetToDelete\COMP3610_bigDataAnalytics_A3-master\src\data\cleaned\cleaned_Books.parquet
✅ Loading file: D:\Jaheem\jaheemProjetToDelete\COMP3610_bigDataAnalytics_A3-master\src\data\cleaned\cleaned_CDs_and_Vinyl.parquet
✅ Loading file: D:\Jaheem\jaheemProjetToDelete\COMP3610_bigDataAnalytics_A3-master\src\data\cleaned\cleaned_Cell_Phones_and_Accessories.parquet
✅ Loading file: D:\Jaheem\jaheemProjetToDelete\COMP3610_bigDataAnalytics_A3-master\src\data\cleaned\cleaned_Clothing_Shoes_and_Jewelry.parquet
✅ Loading file: D:\Jaheem\jaheemProjetToDelete\COMP3610_bigDataAnalytics_A3-master\src\data\cleaned\cleaned_Digital_Music.parquet
✅ Loading file: D:\Jaheem\jaheemProjetToDelete\COMP3610_bigDataAnalytics_A3-master\src\data\cleaned\cleaned_Electronics.parquet
✅ Loading file: D:\Jaheem\jaheemProjetToDelete\COMP3610_bigDataAnalytics_A3-master\src\data\cleaned\cleaned_Gift_Cards.parquet
✅ Loading file: D:\Jaheem\jaheemProjetToDelete\COMP3610_bigDataAnalytics_A3-master\src\data\cleaned\cleaned_Grocery_and_Gourmet_Food.parquet
✅ Loading file: D:\Jaheem\jaheemProjetToDelete\COMP3610_bigDataAnalytics_A3-master\src\data\cleaned\cleaned_Handmade_Products.parquet
✅ Loading file: D:\Jaheem\jaheemProjetToDelete\COMP3610_bigDataAnalytics_A3-master\src\data\cleaned\cleaned_Health_and_Household.parquet
✅ Loading file: D:\Jaheem\jaheemProjetToDelete\COMP3610_bigDataAnalytics_A3-master\src\data\cleaned\cleaned_Health_and_Personal_Care.parquet
✅ Loading file: D:\Jaheem\jaheemProjetToDelete\COMP3610_bigDataAnalytics_A3-master\src\data\cleaned\cleaned_Home_and_Kitchen.parquet
✅ Loading file: D:\Jaheem\jaheemProjetToDelete\COMP3610_bigDataAnalytics_A3-master\src\data\cleaned\cleaned_Industrial_and_Scientific.parquet
✅ Loading file: D:\Jaheem\jaheemProjetToDelete\COMP3610_bigDataAnalytics_A3-master\src\data\cleaned\cleaned_Kindle_Store.parquet
✅ Loading file: D:\Jaheem\jaheemProjetToDelete\COMP3610_bigDataAnalytics_A3-master\src\data\cleaned\cleaned_Magazine_Subscriptions.parquet
✅ Loading file: D:\Jaheem\jaheemProjetToDelete\COMP3610_bigDataAnalytics_A3-master\src\data\cleaned\cleaned_Movies_and_TV.parquet
✅ Loading file: D:\Jaheem\jaheemProjetToDelete\COMP3610_bigDataAnalytics_A3-master\src\data\cleaned\cleaned_Musical_Instruments.parquet
✅ Loading file: D:\Jaheem\jaheemProjetToDelete\COMP3610_bigDataAnalytics_A3-master\src\data\cleaned\cleaned_Office_Products.parquet
✅ Loading file: D:\Jaheem\jaheemProjetToDelete\COMP3610_bigDataAnalytics_A3-master\src\data\cleaned\cleaned_Patio_Lawn_and_Garden.parquet
✅ Loading file: D:\Jaheem\jaheemProjetToDelete\COMP3610_bigDataAnalytics_A3-master\src\data\cleaned\cleaned_Pet_Supplies.parquet
✅ Loading file: D:\Jaheem\jaheemProjetToDelete\COMP3610_bigDataAnalytics_A3-master\src\data\cleaned\cleaned_Software.parquet
✅ Loading file: D:\Jaheem\jaheemProjetToDelete\COMP3610_bigDataAnalytics_A3-master\src\data\cleaned\cleaned_Sports_and_Outdoors.parquet
✅ Loading file: D:\Jaheem\jaheemProjetToDelete\COMP3610_bigDataAnalytics_A3-master\src\data\cleaned\cleaned_Subscription_Boxes.parquet
✅ Loading file: D:\Jaheem\jaheemProjetToDelete\COMP3610_bigDataAnalytics_A3-master\src\data\cleaned\cleaned_Tools_and_Home_Improvement.parquet
✅ Loading file: D:\Jaheem\jaheemProjetToDelete\COMP3610_bigDataAnalytics_A3-master\src\data\cleaned\cleaned_Toys_and_Games.parquet
✅ Loading file: D:\Jaheem\jaheemProjetToDelete\COMP3610_bigDataAnalytics_A3-master\src\data\cleaned\cleaned_Unknown.parquet
✅ Loading file: D:\Jaheem\jaheemProjetToDelete\COMP3610_bigDataAnalytics_A3-master\src\data\cleaned\cleaned_Video_Games.parquet
🔍 Calculating Top Categories by Review Count...
D:\Jaheem\jaheemProjetToDelete\COMP3610_bigDataAnalytics_A3-master\src\testingEDA.py:86: FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be removed in v0.14.0. Assign the `x` variable to `hue` and set `legend=False` for the same effe

  sns.barplot(data=top_categories, x='Category', y='Count', palette='viridis')
💾 Saving plot: top_categories_by_review_count.png
```

```
💾 Saving plot: top_categories_by_review_count.png
✅ Loading file: D:\Jaheem\jaheemProjetToDelete\COMP3610_bigDataAnalytics_A3-master\src\data\cleaned\cleaned_All_Beauty.parquet
✅ Loading file: D:\Jaheem\jaheemProjetToDelete\COMP3610_bigDataAnalytics_A3-master\src\data\cleaned\cleaned_Amazon_Fashion.parquet
✅ Loading file: D:\Jaheem\jaheemProjetToDelete\COMP3610_bigDataAnalytics_A3-master\src\data\cleaned\cleaned_Appliances.parquet
✅ Loading file: D:\Jaheem\jaheemProjetToDelete\COMP3610_bigDataAnalytics_A3-master\src\data\cleaned\cleaned_Arts_Crafts_and_Sewing.parquet
✅ Loading file: D:\Jaheem\jaheemProjetToDelete\COMP3610_bigDataAnalytics_A3-master\src\data\cleaned\cleaned_Automotive.parquet
✅ Loading file: D:\Jaheem\jaheemProjetToDelete\COMP3610_bigDataAnalytics_A3-master\src\data\cleaned\cleaned_Baby_Products.parquet
✅ Loading file: D:\Jaheem\jaheemProjetToDelete\COMP3610_bigDataAnalytics_A3-master\src\data\cleaned\cleaned_Beauty_and_Personal_Care.parquet
✅ Loading file: D:\Jaheem\jaheemProjetToDelete\COMP3610_bigDataAnalytics_A3-master\src\data\cleaned\cleaned_Books.parquet
✅ Loading file: D:\Jaheem\jaheemProjetToDelete\COMP3610_bigDataAnalytics_A3-master\src\data\cleaned\cleaned_CDs_and_Vinyl.parquet
✅ Loading file: D:\Jaheem\jaheemProjetToDelete\COMP3610_bigDataAnalytics_A3-master\src\data\cleaned\cleaned_Cell_Phones_and_Accessories.parquet
✅ Loading file: D:\Jaheem\jaheemProjetToDelete\COMP3610_bigDataAnalytics_A3-master\src\data\cleaned\cleaned_Clothing_Shoes_and_Jewelry.parquet
✅ Loading file: D:\Jaheem\jaheemProjetToDelete\COMP3610_bigDataAnalytics_A3-master\src\data\cleaned\cleaned_Digital_Music.parquet
✅ Loading file: D:\Jaheem\jaheemProjetToDelete\COMP3610_bigDataAnalytics_A3-master\src\data\cleaned\cleaned_Electronics.parquet
✅ Loading file: D:\Jaheem\jaheemProjetToDelete\COMP3610_bigDataAnalytics_A3-master\src\data\cleaned\cleaned_Gift_Cards.parquet
✅ Loading file: D:\Jaheem\jaheemProjetToDelete\COMP3610_bigDataAnalytics_A3-master\src\data\cleaned\cleaned_Grocery_and_Gourmet_Food.parquet
✅ Loading file: D:\Jaheem\jaheemProjetToDelete\COMP3610_bigDataAnalytics_A3-master\src\data\cleaned\cleaned_Handmade_Products.parquet
✅ Loading file: D:\Jaheem\jaheemProjetToDelete\COMP3610_bigDataAnalytics_A3-master\src\data\cleaned\cleaned_Health_and_Household.parquet
✅ Loading file: D:\Jaheem\jaheemProjetToDelete\COMP3610_bigDataAnalytics_A3-master\src\data\cleaned\cleaned_Health_and_Personal_Care.parquet
✅ Loading file: D:\Jaheem\jaheemProjetToDelete\COMP3610_bigDataAnalytics_A3-master\src\data\cleaned\cleaned_Home_and_Kitchen.parquet
✅ Loading file: D:\Jaheem\jaheemProjetToDelete\COMP3610_bigDataAnalytics_A3-master\src\data\cleaned\cleaned_Industrial_and_Scientific.parquet
✅ Loading file: D:\Jaheem\jaheemProjetToDelete\COMP3610_bigDataAnalytics_A3-master\src\data\cleaned\cleaned_Kindle_Store.parquet
✅ Loading file: D:\Jaheem\jaheemProjetToDelete\COMP3610_bigDataAnalytics_A3-master\src\data\cleaned\cleaned_Magazine_Subscriptions.parquet
✅ Loading file: D:\Jaheem\jaheemProjetToDelete\COMP3610_bigDataAnalytics_A3-master\src\data\cleaned\cleaned_Movies_and_TV.parquet
✅ Loading file: D:\Jaheem\jaheemProjetToDelete\COMP3610_bigDataAnalytics_A3-master\src\data\cleaned\cleaned_Musical_Instruments.parquet
✅ Loading file: D:\Jaheem\jaheemProjetToDelete\COMP3610_bigDataAnalytics_A3-master\src\data\cleaned\cleaned_Office_Products.parquet
✅ Loading file: D:\Jaheem\jaheemProjetToDelete\COMP3610_bigDataAnalytics_A3-master\src\data\cleaned\cleaned_Patio_Lawn_and_Garden.parquet
✅ Loading file: D:\Jaheem\jaheemProjetToDelete\COMP3610_bigDataAnalytics_A3-master\src\data\cleaned\cleaned_Pet_Supplies.parquet
✅ Loading file: D:\Jaheem\jaheemProjetToDelete\COMP3610_bigDataAnalytics_A3-master\src\data\cleaned\cleaned_Software.parquet
✅ Loading file: D:\Jaheem\jaheemProjetToDelete\COMP3610_bigDataAnalytics_A3-master\src\data\cleaned\cleaned_Sports_and_Outdoors.parquet
✅ Loading file: D:\Jaheem\jaheemProjetToDelete\COMP3610_bigDataAnalytics_A3-master\src\data\cleaned\cleaned_Subscription_Boxes.parquet
✅ Loading file: D:\Jaheem\jaheemProjetToDelete\COMP3610_bigDataAnalytics_A3-master\src\data\cleaned\cleaned_Tools_and_Home_Improvement.parquet
✅ Loading file: D:\Jaheem\jaheemProjetToDelete\COMP3610_bigDataAnalytics_A3-master\src\data\cleaned\cleaned_Toys_and_Games.parquet
✅ Loading file: D:\Jaheem\jaheemProjetToDelete\COMP3610_bigDataAnalytics_A3-master\src\data\cleaned\cleaned_Unknown.parquet
✅ Loading file: D:\Jaheem\jaheemProjetToDelete\COMP3610_bigDataAnalytics_A3-master\src\data\cleaned\cleaned_Video_Games.parquet
🔍 Calculating Top Brands by Review Count...
D:\Jaheem\jaheemProjetToDelete\COMP3610_bigDataAnalytics_A3-master\src\testingEDA.py:135: FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be removed in v0.14.0. Assign the `x` variable to `hue` and set `legend=False` for the same effect.

  sns.barplot(x=top_brands.index, y=top_brands.values, palette='crest')
💾 Saving plot: top_brands_by_review_count.png
```

```
💾 Saving plot: top_brands_by_review_count.png
✅ Loading file: D:\Jaheem\jaheemProjetToDelete\COMP3610_bigDataAnalytics_A3-master\src\data\cleaned\cleaned_All_Beauty.parquet
✅ Loading file: D:\Jaheem\jaheemProjetToDelete\COMP3610_bigDataAnalytics_A3-master\src\data\cleaned\cleaned_Amazon_Fashion.parquet
✅ Loading file: D:\Jaheem\jaheemProjetToDelete\COMP3610_bigDataAnalytics_A3-master\src\data\cleaned\cleaned_Appliances.parquet
✅ Loading file: D:\Jaheem\jaheemProjetToDelete\COMP3610_bigDataAnalytics_A3-master\src\data\cleaned\cleaned_Arts_Crafts_and_Sewing.parquet
✅ Loading file: D:\Jaheem\jaheemProjetToDelete\COMP3610_bigDataAnalytics_A3-master\src\data\cleaned\cleaned_Automotive.parquet
✅ Loading file: D:\Jaheem\jaheemProjetToDelete\COMP3610_bigDataAnalytics_A3-master\src\data\cleaned\cleaned_Baby_Products.parquet
✅ Loading file: D:\Jaheem\jaheemProjetToDelete\COMP3610_bigDataAnalytics_A3-master\src\data\cleaned\cleaned_Beauty_and_Personal_Care.parquet
✅ Loading file: D:\Jaheem\jaheemProjetToDelete\COMP3610_bigDataAnalytics_A3-master\src\data\cleaned\cleaned_Books.parquet
✅ Loading file: D:\Jaheem\jaheemProjetToDelete\COMP3610_bigDataAnalytics_A3-master\src\data\cleaned\cleaned_CDs_and_Vinyl.parquet
✅ Loading file: D:\Jaheem\jaheemProjetToDelete\COMP3610_bigDataAnalytics_A3-master\src\data\cleaned\cleaned_Cell_Phones_and_Accessories.parquet
✅ Loading file: D:\Jaheem\jaheemProjetToDelete\COMP3610_bigDataAnalytics_A3-master\src\data\cleaned\cleaned_Clothing_Shoes_and_Jewelry.parquet
✅ Loading file: D:\Jaheem\jaheemProjetToDelete\COMP3610_bigDataAnalytics_A3-master\src\data\cleaned\cleaned_Digital_Music.parquet
✅ Loading file: D:\Jaheem\jaheemProjetToDelete\COMP3610_bigDataAnalytics_A3-master\src\data\cleaned\cleaned_Electronics.parquet
✅ Loading file: D:\Jaheem\jaheemProjetToDelete\COMP3610_bigDataAnalytics_A3-master\src\data\cleaned\cleaned_Gift_Cards.parquet
✅ Loading file: D:\Jaheem\jaheemProjetToDelete\COMP3610_bigDataAnalytics_A3-master\src\data\cleaned\cleaned_Grocery_and_Gourmet_Food.parquet
✅ Loading file: D:\Jaheem\jaheemProjetToDelete\COMP3610_bigDataAnalytics_A3-master\src\data\cleaned\cleaned_Handmade_Products.parquet
✅ Loading file: D:\Jaheem\jaheemProjetToDelete\COMP3610_bigDataAnalytics_A3-master\src\data\cleaned\cleaned_Health_and_Household.parquet
✅ Loading file: D:\Jaheem\jaheemProjetToDelete\COMP3610_bigDataAnalytics_A3-master\src\data\cleaned\cleaned_Health_and_Personal_Care.parquet
✅ Loading file: D:\Jaheem\jaheemProjetToDelete\COMP3610_bigDataAnalytics_A3-master\src\data\cleaned\cleaned_Home_and_Kitchen.parquet
✅ Loading file: D:\Jaheem\jaheemProjetToDelete\COMP3610_bigDataAnalytics_A3-master\src\data\cleaned\cleaned_Industrial_and_Scientific.parquet
✅ Loading file: D:\Jaheem\jaheemProjetToDelete\COMP3610_bigDataAnalytics_A3-master\src\data\cleaned\cleaned_Kindle_Store.parquet
✅ Loading file: D:\Jaheem\jaheemProjetToDelete\COMP3610_bigDataAnalytics_A3-master\src\data\cleaned\cleaned_Magazine_Subscriptions.parquet
✅ Loading file: D:\Jaheem\jaheemProjetToDelete\COMP3610_bigDataAnalytics_A3-master\src\data\cleaned\cleaned_Movies_and_TV.parquet
✅ Loading file: D:\Jaheem\jaheemProjetToDelete\COMP3610_bigDataAnalytics_A3-master\src\data\cleaned\cleaned_Musical_Instruments.parquet
✅ Loading file: D:\Jaheem\jaheemProjetToDelete\COMP3610_bigDataAnalytics_A3-master\src\data\cleaned\cleaned_Office_Products.parquet
✅ Loading file: D:\Jaheem\jaheemProjetToDelete\COMP3610_bigDataAnalytics_A3-master\src\data\cleaned\cleaned_Patio_Lawn_and_Garden.parquet
✅ Loading file: D:\Jaheem\jaheemProjetToDelete\COMP3610_bigDataAnalytics_A3-master\src\data\cleaned\cleaned_Pet_Supplies.parquet
✅ Loading file: D:\Jaheem\jaheemProjetToDelete\COMP3610_bigDataAnalytics_A3-master\src\data\cleaned\cleaned_Software.parquet
✅ Loading file: D:\Jaheem\jaheemProjetToDelete\COMP3610_bigDataAnalytics_A3-master\src\data\cleaned\cleaned_Sports_and_Outdoors.parquet
✅ Loading file: D:\Jaheem\jaheemProjetToDelete\COMP3610_bigDataAnalytics_A3-master\src\data\cleaned\cleaned_Subscription_Boxes.parquet
✅ Loading file: D:\Jaheem\jaheemProjetToDelete\COMP3610_bigDataAnalytics_A3-master\src\data\cleaned\cleaned_Tools_and_Home_Improvement.parquet
✅ Loading file: D:\Jaheem\jaheemProjetToDelete\COMP3610_bigDataAnalytics_A3-master\src\data\cleaned\cleaned_Toys_and_Games.parquet
✅ Loading file: D:\Jaheem\jaheemProjetToDelete\COMP3610_bigDataAnalytics_A3-master\src\data\cleaned\cleaned_Unknown.parquet
✅ Loading file: D:\Jaheem\jaheemProjetToDelete\COMP3610_bigDataAnalytics_A3-master\src\data\cleaned\cleaned_Video_Games.parquet
🔍 Calculating Average Star Rating Per Year...
💾 Saving plot: avg_star_rating_per_year_with_rolling_avg.png
```


✅ Loading file: D:\Jaheem\jaheemProjetToDelete\COMP3610_bigDataAnalytics_A3-master\src\data\cleaned\cleaned_All_Beauty.parquet
✅ Loading file: D:\Jaheem\jaheemProjetToDelete\COMP3610_bigDataAnalytics_A3-master\src\data\cleaned\cleaned_Amazon_Fashion.parquet
✅ Loading file: D:\Jaheem\jaheemProjetToDelete\COMP3610_bigDataAnalytics_A3-master\src\data\cleaned\cleaned_Appliances.parquet
✅ Loading file: D:\Jaheem\jaheemProjetToDelete\COMP3610_bigDataAnalytics_A3-master\src\data\cleaned\cleaned_Arts_Crafts_and_Sewing.parquet
✅ Loading file: D:\Jaheem\jaheemProjetToDelete\COMP3610_bigDataAnalytics_A3-master\src\data\cleaned\cleaned_Automotive.parquet
✅ Loading file: D:\Jaheem\jaheemProjetToDelete\COMP3610_bigDataAnalytics_A3-master\src\data\cleaned\cleaned_Baby_Products.parquet
✅ Loading file: D:\Jaheem\jaheemProjetToDelete\COMP3610_bigDataAnalytics_A3-master\src\data\cleaned\cleaned_Beauty_and_Personal_Care.parquet
✅ Loading file: D:\Jaheem\jaheemProjetToDelete\COMP3610_bigDataAnalytics_A3-master\src\data\cleaned\cleaned_Books.parquet
✅ Loading file: D:\Jaheem\jaheemProjetToDelete\COMP3610_bigDataAnalytics_A3-master\src\data\cleaned\cleaned_CDs_and_Vinyl.parquet
✅ Loading file: D:\Jaheem\jaheemProjetToDelete\COMP3610_bigDataAnalytics_A3-master\src\data\cleaned\cleaned_Cell_Phones_and_Accessories.parquet
✅ Loading file: D:\Jaheem\jaheemProjetToDelete\COMP3610_bigDataAnalytics_A3-master\src\data\cleaned\cleaned_Clothing_Shoes_and_Jewelry.parquet
✅ Loading file: D:\Jaheem\jaheemProjetToDelete\COMP3610_bigDataAnalytics_A3-master\src\data\cleaned\cleaned_Digital_Music.parquet
✅ Loading file: D:\Jaheem\jaheemProjetToDelete\COMP3610_bigDataAnalytics_A3-master\src\data\cleaned\cleaned_Electronics.parquet
✅ Loading file: D:\Jaheem\jaheemProjetToDelete\COMP3610_bigDataAnalytics_A3-master\src\data\cleaned\cleaned_Gift_Cards.parquet
✅ Loading file: D:\Jaheem\jaheemProjetToDelete\COMP3610_bigDataAnalytics_A3-master\src\data\cleaned\cleaned_Grocery_and_Gourmet_Food.parquet
✅ Loading file: D:\Jaheem\jaheemProjetToDelete\COMP3610_bigDataAnalytics_A3-master\src\data\cleaned\cleaned_Handmade_Products.parquet
✅ Loading file: D:\Jaheem\jaheemProjetToDelete\COMP3610_bigDataAnalytics_A3-master\src\data\cleaned\cleaned_Health_and_Household.parquet
✅ Loading file: D:\Jaheem\jaheemProjetToDelete\COMP3610_bigDataAnalytics_A3-master\src\data\cleaned\cleaned_Health_and_Personal_Care.parquet
✅ Loading file: D:\Jaheem\jaheemProjetToDelete\COMP3610_bigDataAnalytics_A3-master\src\data\cleaned\cleaned_Home_and_Kitchen.parquet
✅ Loading file: D:\Jaheem\jaheemProjetToDelete\COMP3610_bigDataAnalytics_A3-master\src\data\cleaned\cleaned_Industrial_and_Scientific.parquet
✅ Loading file: D:\Jaheem\jaheemProjetToDelete\COMP3610_bigDataAnalytics_A3-master\src\data\cleaned\cleaned_Kindle_Store.parquet
✅ Loading file: D:\Jaheem\jaheemProjetToDelete\COMP3610_bigDataAnalytics_A3-master\src\data\cleaned\cleaned_Magazine_Subscriptions.parquet
✅ Loading file: D:\Jaheem\jaheemProjetToDelete\COMP3610_bigDataAnalytics_A3-master\src\data\cleaned\cleaned_Movies_and_TV.parquet
✅ Loading file: D:\Jaheem\jaheemProjetToDelete\COMP3610_bigDataAnalytics_A3-master\src\data\cleaned\cleaned_Musical_Instruments.parquet
✅ Loading file: D:\Jaheem\jaheemProjetToDelete\COMP3610_bigDataAnalytics_A3-master\src\data\cleaned\cleaned_Office_Products.parquet
✅ Loading file: D:\Jaheem\jaheemProjetToDelete\COMP3610_bigDataAnalytics_A3-master\src\data\cleaned\cleaned_Patio_Lawn_and_Garden.parquet
✅ Loading file: D:\Jaheem\jaheemProjetToDelete\COMP3610_bigDataAnalytics_A3-master\src\data\cleaned\cleaned_Pet_Supplies.parquet
✅ Loading file: D:\Jaheem\jaheemProjetToDelete\COMP3610_bigDataAnalytics_A3-master\src\data\cleaned\cleaned_Software.parquet
✅ Loading file: D:\Jaheem\jaheemProjetToDelete\COMP3610_bigDataAnalytics_A3-master\src\data\cleaned\cleaned_Sports_and_Outdoors.parquet
✅ Loading file: D:\Jaheem\jaheemProjetToDelete\COMP3610_bigDataAnalytics_A3-master\src\data\cleaned\cleaned_Subscription_Boxes.parquet
✅ Loading file: D:\Jaheem\jaheemProjetToDelete\COMP3610_bigDataAnalytics_A3-master\src\data\cleaned\cleaned_Tools_and_Home_Improvement.parquet
✅ Loading file: D:\Jaheem\jaheemProjetToDelete\COMP3610_bigDataAnalytics_A3-master\src\data\cleaned\cleaned_Toys_and_Games.parquet
✅ Loading file: D:\Jaheem\jaheemProjetToDelete\COMP3610_bigDataAnalytics_A3-master\src\data\cleaned\cleaned_Unknown.parquet
✅ Loading file: D:\Jaheem\jaheemProjetToDelete\COMP3610_bigDataAnalytics_A3-master\src\data\cleaned\cleaned_Video_Games.parquet
🔍 Calculating Correlation between Review Length and Rating...
📉 Pearson Correlation between Review Length and Rating: -0.067
🔍 Interpretation: Longer reviews tend to have lower ratings.

Binary Sentiment (Logistic Regression)

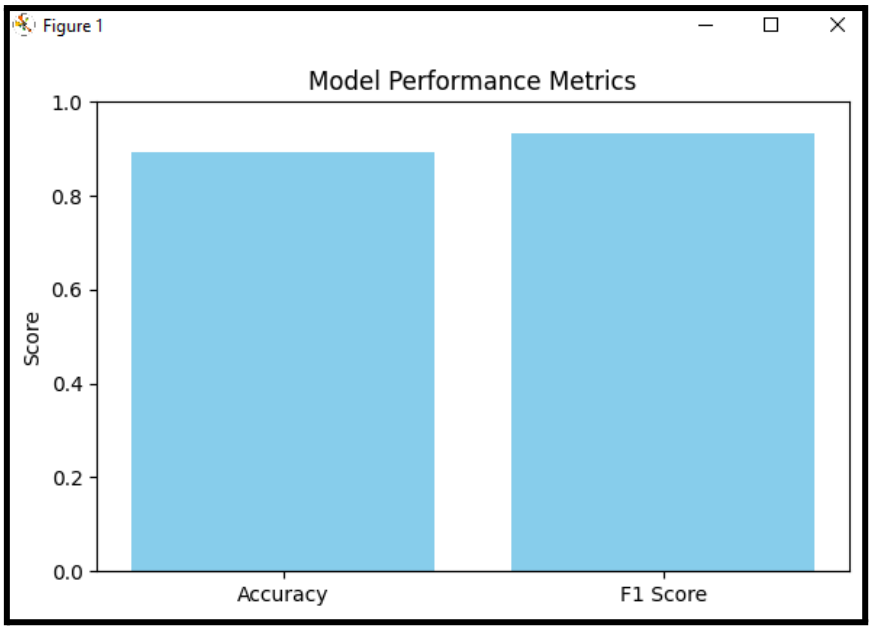
Ratings were transformed into binary labels: positive for ratings above 3, and negative for ratings of 3 or below. The dataset was randomly split into 80% training and 20% testing sets. Review text was vectorized using TF-IDF, with preprocessing steps including lowercasing, tokenization on whitespace and punctuation, and removal of rare or overly common tokens. A logistic regression classifier with default settings was used for prediction. Model performance was evaluated using accuracy, F1 score, and a 2×2 confusion matrix (true positives, false positives, true negatives, false negatives).

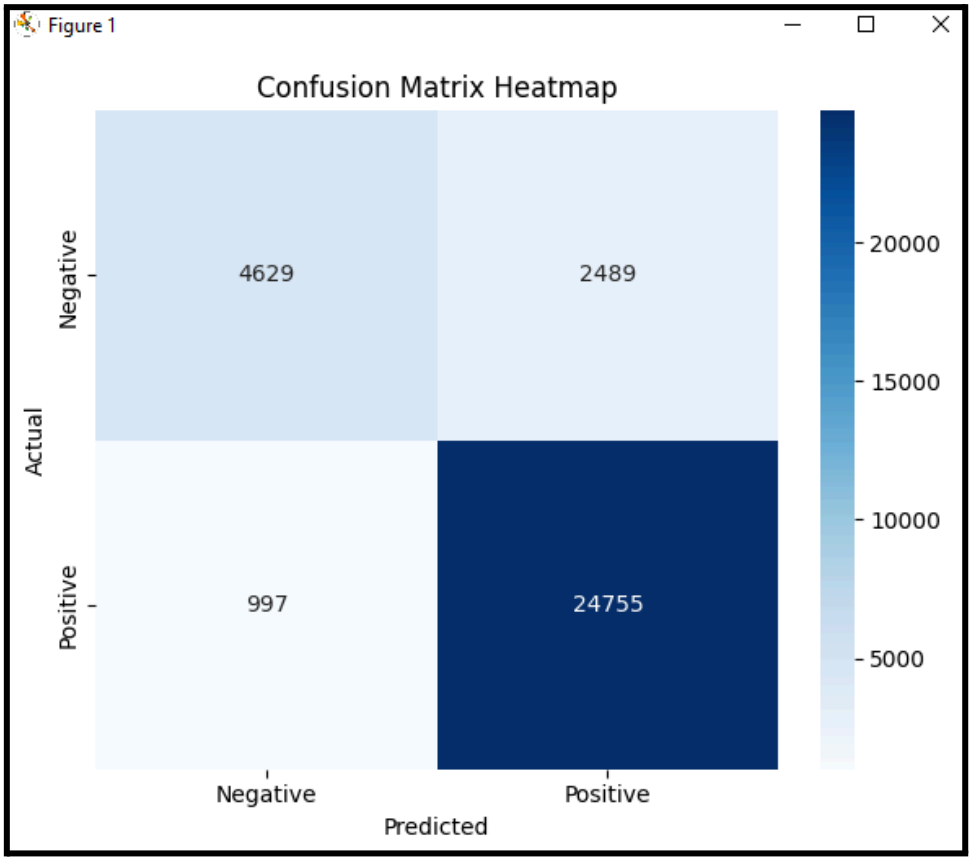
Evaluation

Evaluation was conducted on a dataset containing 5% of each category. A bar graph was included to visualize model performance, displaying accuracy and F1 score, along with a confusion matrix heatmap for this second evaluation.

The model demonstrated strong performance on this dataset, achieving 89.39% accuracy and a 93.42% F1 score. The confusion matrix showed 4,629 true negatives, 24,755 true positives, 2,489 false positives, and 997 false negatives. With a relatively low number of false negatives, the model effectively identified favorable evaluations. Although there was a higher count of false positives, the overall F1 score indicated a solid balance between recall and accuracy. The consistent performance across both datasets reinforces the reliability of logistic regression for binary sentiment prediction in customer review data.

```
✅ Evaluation Results
   - Accuracy: 0.8939458472771524
   - F1 Score: 0.9342214506755226
   - Confusion Matrix:
[[ 4629  2489]
 [  997 24755]]
```





Code

Labeling Sentiment and Sampling:

```
def add_sentiment_label(df):
     df["sentiment"] = df["rating"].apply(lambda x: 1 if x > 3 else
0)
    return df

def stratified_sample(df, frac=0.05):
    return df.groupby('sentiment', group_keys=False).apply(lambda x:
x.sample(frac=frac, random_state=42))
```

Data Loading and Sampling:

```
def load_and_sample_data():
    print(" 🔹 Loading and stratified sampling data...")
    base_path =
r"D:\Jaheem\jaheemProjetToDelete\COMP3610_bigDataAnalytics_A3-master
\src\data\cleaned"
    paths = [fr"{base_path}\cleaned_{category}.parquet" for category
in VALID_CATEGORIES]
    all_samples = []
    for path, category in zip(paths, VALID_CATEGORIES):
         print(f"   - Reading and sampling 5% of data for category:
{category}")
        dataset = ds.dataset(path, format="parquet")
        scanner = dataset.scanner(columns=["text", "rating"])
        table = scanner.head(100_000)
        df = table.to_pandas()
        df = add_sentiment_label(df)
        df_sample = stratified_sample(df, frac=0.05)
        all_samples.append(df_sample)
    combined_df = pd.concat(all_samples, ignore_index=True)
    return combined_df
```

Data Preprocessing:

```
def split_sentiment_data(df):
    return train_test_split(df["text"], df["sentiment"],
test_size=0.2, stratify=df["sentiment"], random_state=42)

def vectorize_text(X_train, X_test):
    vectorizer = TfidfVectorizer(min_df=5, max_df=0.8)
    X_train_tfidf = vectorizer.fit_transform(X_train)
    X_test_tfidf = vectorizer.transform(X_test)
    return X_train_tfidf, X_test_tfidf, vectorizer
```

Model Training:

```
def train_logistic_regression(X_train_tfidf, y_train):
    model = LogisticRegression()
    model.fit(X_train_tfidf, y_train)
    return model
```

Model Evaluation:

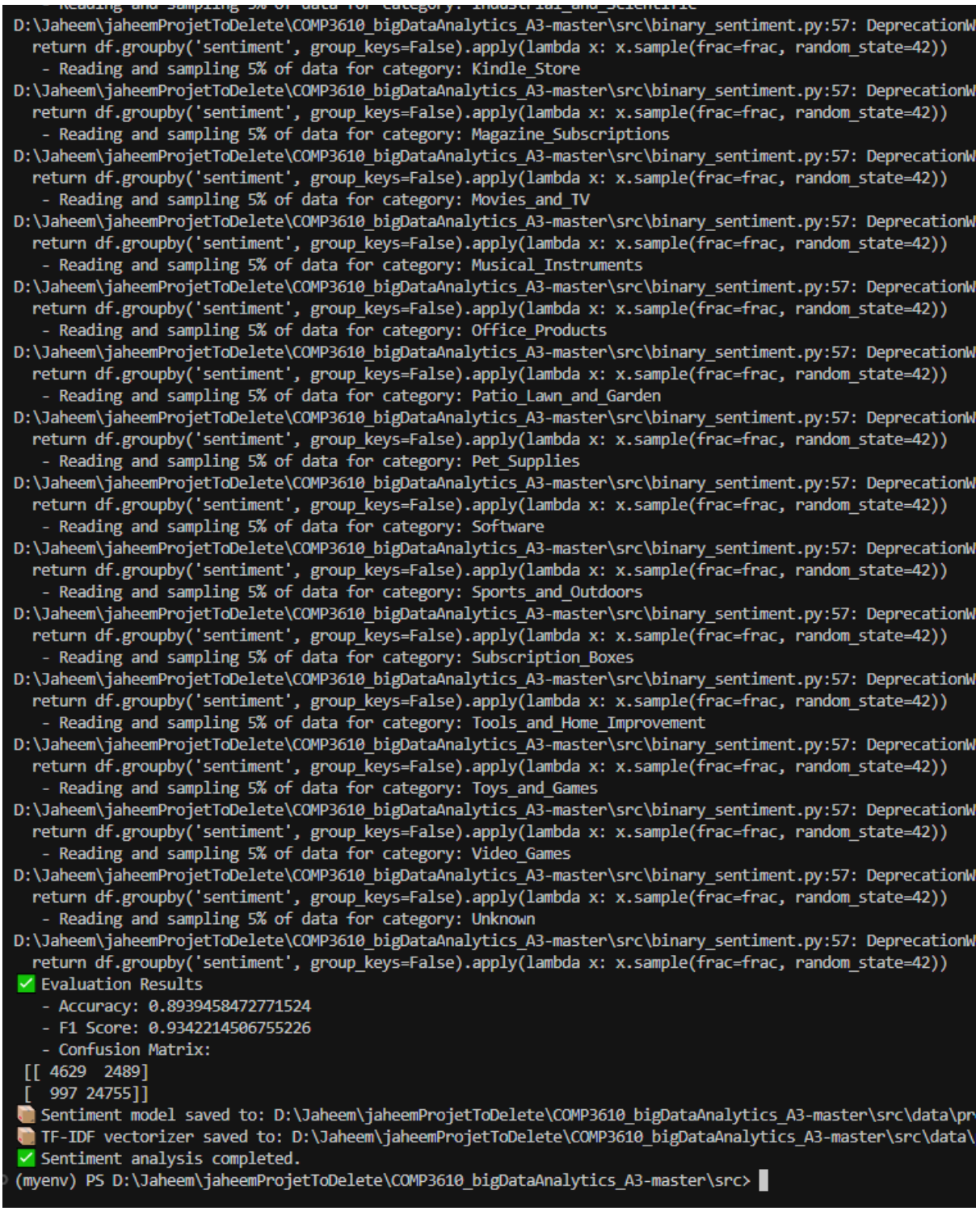
```
def evaluate_model(model, X_test_tfidf, y_test):
    y_pred = model.predict(X_test_tfidf)
    acc = accuracy_score(y_test, y_pred)
    f1 = f1_score(y_test, y_pred)
    cm = confusion_matrix(y_test, y_pred)
    print("✅ Evaluation Results")
    print("   - Accuracy:", acc)
    print("   - F1 Score:", f1)
    print("   - Confusion Matrix:\n", cm)
    plt.figure(figsize=(6, 5))
    sns.heatmap(cm, annot=True, fmt='d', cmap='Blues',
xticklabels=["Negative", "Positive"], yticklabels=["Negative",
"Positive"])
    plt.xlabel('Predicted')
    plt.ylabel('Actual')
    plt.title('Confusion Matrix Heatmap')
    plt.tight_layout()
    plt.show()
    plt.figure(figsize=(6, 4))
    plt.bar(["Accuracy", "F1 Score"], [acc, f1], color='skyblue')
```

```
    plt.ylim(0, 1)
    plt.title("Model Performance Metrics")
    plt.ylabel("Score")
    plt.tight_layout()
    plt.show()
```

Running Pipeline and Saving Artifacts:

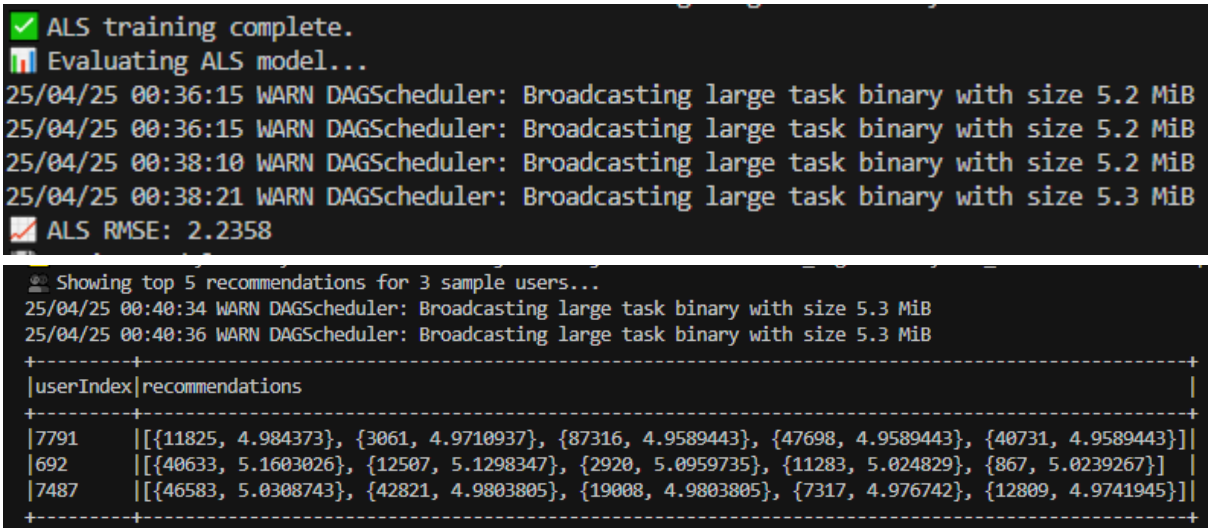
```
print("🔹 Starting sentiment classification pipeline...")
df = load_and_sample_data()
X_train, X_test, y_train, y_test = split_sentiment_data(df)
X_train_tfidf, X_test_tfidf, vectorizer = vectorize_text(X_train,
X_test)
model = train_logistic_regression(X_train_tfidf, y_train)
evaluate_model(model, X_test_tfidf, y_test)
model_dir =
r"D:\Jaheem\jaheemProjetToDelete\COMP3610_bigDataAnalytics_A3-master
\src\data\processed\models"
os.makedirs(model_dir, exist_ok=True)
model_path = os.path.join(model_dir, "sentiment_logistic_model.pkl")
vectorizer_path = os.path.join(model_dir, "tfidf_vectorizer.pkl")
joblib.dump(model, model_path)
joblib.dump(vectorizer, vectorizer_path)
print(f"📦 Sentiment model saved to: {model_path}")
print(f"📦 TF-IDF vectorizer saved to: {vectorizer_path}")
print("✅ Sentiment analysis completed.")
```

```
(myenv) PS D:\Jaheem\jaheemProjetToDelete\COMP3610_bigDataAnalytics_A3-master\src> python binary_sentiment.py
🔹 Starting sentiment classification pipeline...
🔹 Loading and stratified sampling data...
   - Reading and sampling 5% of data for category: All_Beauty
D:\Jaheem\jaheemProjetToDelete\COMP3610_bigDataAnalytics_A3-master\src\binary_sentiment.py:57: DeprecationWarning: DataFrameGroupBy.apply operat
  return df.groupby('sentiment', group_keys=False).apply(lambda x: x.sample(frac=frac, random_state=42))
   - Reading and sampling 5% of data for category: Amazon_Fashion
D:\Jaheem\jaheemProjetToDelete\COMP3610_bigDataAnalytics_A3-master\src\binary_sentiment.py:57: DeprecationWarning: DataFrameGroupBy.apply operat
  return df.groupby('sentiment', group_keys=False).apply(lambda x: x.sample(frac=frac, random_state=42))
   - Reading and sampling 5% of data for category: Appliances
D:\Jaheem\jaheemProjetToDelete\COMP3610_bigDataAnalytics_A3-master\src\binary_sentiment.py:57: DeprecationWarning: DataFrameGroupBy.apply operat
  return df.groupby('sentiment', group_keys=False).apply(lambda x: x.sample(frac=frac, random_state=42))
   - Reading and sampling 5% of data for category: Arts_Crafts_and_Sewing
D:\Jaheem\jaheemProjetToDelete\COMP3610_bigDataAnalytics_A3-master\src\binary_sentiment.py:57: DeprecationWarning: DataFrameGroupBy.apply operat
  return df.groupby('sentiment', group_keys=False).apply(lambda x: x.sample(frac=frac, random_state=42))
   - Reading and sampling 5% of data for category: Automotive
D:\Jaheem\jaheemProjetToDelete\COMP3610_bigDataAnalytics_A3-master\src\binary_sentiment.py:57: DeprecationWarning: DataFrameGroupBy.apply operat
  return df.groupby('sentiment', group_keys=False).apply(lambda x: x.sample(frac=frac, random_state=42))
   - Reading and sampling 5% of data for category: Baby_Products
D:\Jaheem\jaheemProjetToDelete\COMP3610_bigDataAnalytics_A3-master\src\binary_sentiment.py:57: DeprecationWarning: DataFrameGroupBy.apply operat
  return df.groupby('sentiment', group_keys=False).apply(lambda x: x.sample(frac=frac, random_state=42))
   - Reading and sampling 5% of data for category: Beauty_and_Personal_Care
D:\Jaheem\jaheemProjetToDelete\COMP3610_bigDataAnalytics_A3-master\src\binary_sentiment.py:57: DeprecationWarning: DataFrameGroupBy.apply operat
  return df.groupby('sentiment', group_keys=False).apply(lambda x: x.sample(frac=frac, random_state=42))
   - Reading and sampling 5% of data for category: Books
D:\Jaheem\jaheemProjetToDelete\COMP3610_bigDataAnalytics_A3-master\src\binary_sentiment.py:57: DeprecationWarning: DataFrameGroupBy.apply operat
  return df.groupby('sentiment', group_keys=False).apply(lambda x: x.sample(frac=frac, random_state=42))
   - Reading and sampling 5% of data for category: CDs_and_Vinyl
D:\Jaheem\jaheemProjetToDelete\COMP3610_bigDataAnalytics_A3-master\src\binary_sentiment.py:57: DeprecationWarning: DataFrameGroupBy.apply operat
  return df.groupby('sentiment', group_keys=False).apply(lambda x: x.sample(frac=frac, random_state=42))
   - Reading and sampling 5% of data for category: Cell_Phones_and_Accessories
D:\Jaheem\jaheemProjetToDelete\COMP3610_bigDataAnalytics_A3-master\src\binary_sentiment.py:57: DeprecationWarning: DataFrameGroupBy.apply operat
  return df.groupby('sentiment', group_keys=False).apply(lambda x: x.sample(frac=frac, random_state=42))
   - Reading and sampling 5% of data for category: Clothing_Shoes_and_Jewelry
D:\Jaheem\jaheemProjetToDelete\COMP3610_bigDataAnalytics_A3-master\src\binary_sentiment.py:57: DeprecationWarning: DataFrameGroupBy.apply operat
  return df.groupby('sentiment', group_keys=False).apply(lambda x: x.sample(frac=frac, random_state=42))
   - Reading and sampling 5% of data for category: Digital_Music
D:\Jaheem\jaheemProjetToDelete\COMP3610_bigDataAnalytics_A3-master\src\binary_sentiment.py:57: DeprecationWarning: DataFrameGroupBy.apply operat
  return df.groupby('sentiment', group_keys=False).apply(lambda x: x.sample(frac=frac, random_state=42))
   - Reading and sampling 5% of data for category: Electronics
D:\Jaheem\jaheemProjetToDelete\COMP3610_bigDataAnalytics_A3-master\src\binary_sentiment.py:57: DeprecationWarning: DataFrameGroupBy.apply operat
  return df.groupby('sentiment', group_keys=False).apply(lambda x: x.sample(frac=frac, random_state=42))
   - Reading and sampling 5% of data for category: Gift_Cards
D:\Jaheem\jaheemProjetToDelete\COMP3610_bigDataAnalytics_A3-master\src\binary_sentiment.py:57: DeprecationWarning: DataFrameGroupBy.apply operat
  return df.groupby('sentiment', group_keys=False).apply(lambda x: x.sample(frac=frac, random_state=42))
   - Reading and sampling 5% of data for category: Grocery_and_Gourmet_Food
D:\Jaheem\jaheemProjetToDelete\COMP3610_bigDataAnalytics_A3-master\src\binary_sentiment.py:57: DeprecationWarning: DataFrameGroupBy.apply operat
  return df.groupby('sentiment', group_keys=False).apply(lambda x: x.sample(frac=frac, random_state=42))
   - Reading and sampling 5% of data for category: Handmade_Products
D:\Jaheem\jaheemProjetToDelete\COMP3610_bigDataAnalytics_A3-master\src\binary_sentiment.py:57: DeprecationWarning: DataFrameGroupBy.apply operat
  return df.groupby('sentiment', group_keys=False).apply(lambda x: x.sample(frac=frac, random_state=42))
   - Reading and sampling 5% of data for category: Health_and_Household
D:\Jaheem\jaheemProjetToDelete\COMP3610_bigDataAnalytics_A3-master\src\binary_sentiment.py:57: DeprecationWarning: DataFrameGroupBy.apply operat
  return df.groupby('sentiment', group_keys=False).apply(lambda x: x.sample(frac=frac, random_state=42))
   - Reading and sampling 5% of data for category: Health_and_Personal_Care
D:\Jaheem\jaheemProjetToDelete\COMP3610_bigDataAnalytics_A3-master\src\binary_sentiment.py:57: DeprecationWarning: DataFrameGroupBy.apply operat
  return df.groupby('sentiment', group_keys=False).apply(lambda x: x.sample(frac=frac, random_state=42))
   - Reading and sampling 5% of data for category: Home_and_Kitchen
D:\Jaheem\jaheemProjetToDelete\COMP3610_bigDataAnalytics_A3-master\src\binary_sentiment.py:57: DeprecationWarning: DataFrameGroupBy.apply operat
  return df.groupby('sentiment', group_keys=False).apply(lambda x: x.sample(frac=frac, random_state=42))
   - Reading and sampling 5% of data for category: Industrial_and_Scientific
D:\Jaheem\jaheemProjetToDelete\COMP3610_bigDataAnalytics_A3-master\src\binary_sentiment.py:57: DeprecationWarning: DataFrameGroupBy.apply operat
  return df.groupby('sentiment', group_keys=False).apply(lambda x: x.sample(frac=frac, random_state=42))
   - Reading and sampling 5% of data for category: Kindle_Store
D:\Jaheem\jaheemProjetToDelete\COMP3610_bigDataAnalytics_A3-master\src\binary_sentiment.py:57: DeprecationWarning: DataFrameGroupBy.apply operat
  return df.groupby('sentiment', group_keys=False).apply(lambda x: x.sample(frac=frac, random_state=42))
   - Reading and sampling 5% of data for category: Magazine_Subscriptions
D:\Jaheem\jaheemProjetToDelete\COMP3610_bigDataAnalytics_A3-master\src\binary_sentiment.py:57: DeprecationWarning: DataFrameGroupBy.apply operat
  return df.groupby('sentiment', group_keys=False).apply(lambda x: x.sample(frac=frac, random_state=42))
   - Reading and sampling 5% of data for category: Movies_and_TV
D:\Jaheem\jaheemProjetToDelete\COMP3610_bigDataAnalytics_A3-master\src\binary_sentiment.py:57: DeprecationWarning: DataFrameGroupBy.apply operat
  return df.groupby('sentiment', group_keys=False).apply(lambda x: x.sample(frac=frac, random_state=42))
   - Reading and sampling 5% of data for category: Musical_Instruments
D:\Jaheem\jaheemProjetToDelete\COMP3610_bigDataAnalytics_A3-master\src\binary_sentiment.py:57: DeprecationWarning: DataFrameGroupBy.apply operat
  return df.groupby('sentiment', group_keys=False).apply(lambda x: x.sample(frac=frac, random_state=42))
```


- Reading and sampling 5% of data for category: Industrial_and_Scientific
D:\Jaheem\jaheemProjetToDelete\COMP3610_bigDataAnalytics_A3-master\src\binary_sentiment.py:57: DeprecationW
return df.groupby('sentiment', group_keys=False).apply(lambda x: x.sample(frac=frac, random_state=42))
- Reading and sampling 5% of data for category: Kindle_Store
D:\Jaheem\jaheemProjetToDelete\COMP3610_bigDataAnalytics_A3-master\src\binary_sentiment.py:57: DeprecationW
return df.groupby('sentiment', group_keys=False).apply(lambda x: x.sample(frac=frac, random_state=42))
- Reading and sampling 5% of data for category: Magazine_Subscriptions
D:\Jaheem\jaheemProjetToDelete\COMP3610_bigDataAnalytics_A3-master\src\binary_sentiment.py:57: DeprecationW
return df.groupby('sentiment', group_keys=False).apply(lambda x: x.sample(frac=frac, random_state=42))
- Reading and sampling 5% of data for category: Movies_and_TV
D:\Jaheem\jaheemProjetToDelete\COMP3610_bigDataAnalytics_A3-master\src\binary_sentiment.py:57: DeprecationW
return df.groupby('sentiment', group_keys=False).apply(lambda x: x.sample(frac=frac, random_state=42))
- Reading and sampling 5% of data for category: Musical_Instruments
D:\Jaheem\jaheemProjetToDelete\COMP3610_bigDataAnalytics_A3-master\src\binary_sentiment.py:57: DeprecationW
return df.groupby('sentiment', group_keys=False).apply(lambda x: x.sample(frac=frac, random_state=42))
- Reading and sampling 5% of data for category: Office_Products
D:\Jaheem\jaheemProjetToDelete\COMP3610_bigDataAnalytics_A3-master\src\binary_sentiment.py:57: DeprecationW
return df.groupby('sentiment', group_keys=False).apply(lambda x: x.sample(frac=frac, random_state=42))
- Reading and sampling 5% of data for category: Patio_Lawn_and_Garden
D:\Jaheem\jaheemProjetToDelete\COMP3610_bigDataAnalytics_A3-master\src\binary_sentiment.py:57: DeprecationW
return df.groupby('sentiment', group_keys=False).apply(lambda x: x.sample(frac=frac, random_state=42))
- Reading and sampling 5% of data for category: Pet_Supplies
D:\Jaheem\jaheemProjetToDelete\COMP3610_bigDataAnalytics_A3-master\src\binary_sentiment.py:57: DeprecationW
return df.groupby('sentiment', group_keys=False).apply(lambda x: x.sample(frac=frac, random_state=42))
- Reading and sampling 5% of data for category: Software
D:\Jaheem\jaheemProjetToDelete\COMP3610_bigDataAnalytics_A3-master\src\binary_sentiment.py:57: DeprecationW
return df.groupby('sentiment', group_keys=False).apply(lambda x: x.sample(frac=frac, random_state=42))
- Reading and sampling 5% of data for category: Sports_and_Outdoors
D:\Jaheem\jaheemProjetToDelete\COMP3610_bigDataAnalytics_A3-master\src\binary_sentiment.py:57: DeprecationW
return df.groupby('sentiment', group_keys=False).apply(lambda x: x.sample(frac=frac, random_state=42))
- Reading and sampling 5% of data for category: Subscription_Boxes
D:\Jaheem\jaheemProjetToDelete\COMP3610_bigDataAnalytics_A3-master\src\binary_sentiment.py:57: DeprecationW
return df.groupby('sentiment', group_keys=False).apply(lambda x: x.sample(frac=frac, random_state=42))
- Reading and sampling 5% of data for category: Tools_and_Home_Improvement
D:\Jaheem\jaheemProjetToDelete\COMP3610_bigDataAnalytics_A3-master\src\binary_sentiment.py:57: DeprecationW
return df.groupby('sentiment', group_keys=False).apply(lambda x: x.sample(frac=frac, random_state=42))
- Reading and sampling 5% of data for category: Toys_and_Games
D:\Jaheem\jaheemProjetToDelete\COMP3610_bigDataAnalytics_A3-master\src\binary_sentiment.py:57: DeprecationW
return df.groupby('sentiment', group_keys=False).apply(lambda x: x.sample(frac=frac, random_state=42))
- Reading and sampling 5% of data for category: Video_Games
D:\Jaheem\jaheemProjetToDelete\COMP3610_bigDataAnalytics_A3-master\src\binary_sentiment.py:57: DeprecationW
return df.groupby('sentiment', group_keys=False).apply(lambda x: x.sample(frac=frac, random_state=42))
- Reading and sampling 5% of data for category: Unknown
D:\Jaheem\jaheemProjetToDelete\COMP3610_bigDataAnalytics_A3-master\src\binary_sentiment.py:57: DeprecationW
return df.groupby('sentiment', group_keys=False).apply(lambda x: x.sample(frac=frac, random_state=42))
✅ Evaluation Results
- Accuracy: 0.8939458472771524
- F1 Score: 0.9342214506755226
- Confusion Matrix:
[[4629 2489]
[997 24755]]
📦 Sentiment model saved to: D:\Jaheem\jaheemProjetToDelete\COMP3610_bigDataAnalytics_A3-master\src\data\pr
📦 TF-IDF vectorizer saved to: D:\Jaheem\jaheemProjetToDelete\COMP3610_bigDataAnalytics_A3-master\src\data\
✅ Sentiment analysis completed.
(myenv) PS D:\Jaheem\jaheemProjetToDelete\COMP3610_bigDataAnalytics_A3-master\src>

Recommender System using ALS

A collaborative filtering model was created using Alternating Least Squares (ALS). The data was first prepared by retaining only the user ID, product ID, and rating columns, and users with fewer than five total reviews were removed. The dataset was then split into training and test sets using an 80-20 ratio, and stratified sampling was applied to train the model based on 5% of each product category. The ALS model was trained using Spark MLlib with minimal hyperparameter tuning. Model performance was evaluated using Root Mean Squared Error (RMSE) on the test set by comparing predicted ratings to actual ratings. For demonstration, the top five recommendations for three randomly selected users from the test set were presented, including their predicted ratings.



Code

Spark Session Initialization:

```
print("🔧 Initializing Spark session...")
spark = SparkSession.builder \
    .appName("ALS Recommender") \
    .config("spark.local.dir", "D:/spark-temp") \
    .config("spark.driver.memory", "16g") \
    .config("spark.executor.memory", "16g") \
    .config("spark.serializer",
"org.apache.spark.serializer.KryoSerializer") \
    .config("spark.kryoserializer.buffer", "128m") \
    .config("spark.kryoserializer.buffer.max", "1024m") \
    .getOrCreate()
```

Load Parquet Files:

```
print("📂 Loading cleaned parquet files...")
base_path =
r"D:\Jaheem\jaheemProjetToDelete\COMP3610_bigDataAnalytics_A3-master
\src\data\cleaned"
paths = [fr"{base_path}\cleaned_{category}.parquet" for category in
VALID_CATEGORIES]
df = spark.read.parquet(*paths)
print("✅ Parquet files loaded.")
```

Stratified Sampling:

```
print("🔍 Taking 5% stratified sample from each category...")
fractions = {category: 0.05 for category in VALID_CATEGORIES}
df = df.stat.sampleBy("main_category", fractions, seed=42)
print("✅ Sampling complete.")
```

Filter Users with ≥ 5 Reviews:

```
print("🧹 Filtering users with fewer than 5 reviews...")
ratings = df.select("user_id", "asin", "rating")
user_counts =
ratings.groupBy("user_id").agg(count("*").alias("review_count"))
filtered_users = user_counts.filter("review_count >=
5").select("user_id")
ratings = ratings.join(filtered_users, on="user_id", how="inner")
print("✅ User filtering complete.")
```

Index User and Item:

```
print("🔢 Indexing user_id and asin...")
user_indexer = StringIndexer(inputCol="user_id",
outputCol="userIndex").fit(ratings)
item_indexer = StringIndexer(inputCol="asin",
outputCol="itemIndex").fit(ratings)
ratings = user_indexer.transform(ratings)
ratings = item_indexer.transform(ratings)
print("✅ Indexing complete.")
```

Train-Test Split:

```
print("🔀 Splitting data into training and test sets...")
train, test = ratings.randomSplit([0.8, 0.2], seed=42)
print("✅ Split complete.")
```

Train ALS Model:

```
print("🧠 Training ALS model...")
als = ALS(userCol="userIndex", itemCol="itemIndex",
ratingCol="rating",
          coldStartStrategy="drop", nonnegative=True)
recommendation_model = als.fit(train)
print("✅ ALS training complete.")
```

Evaluate Model:

```
print("📊 Evaluating ALS model...")
predictions = recommendation_model.transform(test)
evaluator = RegressionEvaluator(metricName="rmse",
labelCol="rating", predictionCol="prediction")
rmse = evaluator.evaluate(predictions)
print(f"📈 ALS RMSE: {rmse:.4f}")
```

Save Model:

```
print("💾 Saving model...")
model_path                                                    =
r"D:\Jaheem\jaheemProjetToDelete\COMP3610_bigDataAnalytics_A3-master
\src\data\processed\models\als_model"
if not os.path.exists(model_path):
    recommendation_model.save(model_path)
    print(f"📦 Model saved to: {model_path}")
else:
    print(f"⚠️ Model directory already exists: {model_path}.
Skipping save.")
```

Generate Top Recommendations:

```
print("👥 Showing top 5 recommendations for 3 sample users...")
user_subset = ratings.select("userIndex").distinct().limit(3)
user_recs = recommendation_model.recommendForUserSubset(user_subset,
5)
user_recs.show(truncate=False)
print("✅ ALS recommendation pipeline completed.")
```

```
(myenv) PS D:\Jaheem\jaheemProjetToDelete\COMP3610_bigDataAnalytics_A3-master\src> python recommender_system.py
🚀 Starting ALS recommendation pipeline...
🔧 Initializing Spark session...
25/04/25 00:22:05 WARN Shell: Did not find winutils.exe: java.io.FileNotFoundException: java.io.FileNotFoundExc
Setting default log level to "WARN".
To adjust logging level use sc.setLogLevel(newLevel). For SparkR, use setLogLevel(newLevel).
25/04/25 00:22:06 WARN NativeCodeLoader: Unable to load native-hadoop library for your platform... using builti
25/04/25 00:22:06 WARN SparkConf: Note that spark.local.dir will be overridden by the value set by the cluster
25/04/25 00:22:06 WARN Utils: The configured local directories are not expected to be URIs; however, got suspic
📂 Loading cleaned parquet files...
✅ Parquet files loaded.
🔍 Taking 5% stratified sample from each category...
✅ Sampling complete.
🧹 Filtering users with fewer than 5 reviews...
✅ User filtering complete.
🔢 Indexing user_id and asin...
✅ Indexing complete.
🔀 Splitting data into training and test sets...
✅ Split complete.
🧠 Training ALS model...
25/04/25 00:35:58 WARN DAGScheduler: Broadcasting large task binary with size 5.2 MiB
25/04/25 00:36:03 WARN DAGScheduler: Broadcasting large task binary with size 5.2 MiB
25/04/25 00:36:05 WARN DAGScheduler: Broadcasting large task binary with size 5.2 MiB
25/04/25 00:36:06 WARN DAGScheduler: Broadcasting large task binary with size 5.2 MiB
25/04/25 00:36:06 WARN DAGScheduler: Broadcasting large task binary with size 5.2 MiB
25/04/25 00:36:07 WARN DAGScheduler: Broadcasting large task binary with size 5.2 MiB
25/04/25 00:36:07 WARN DAGScheduler: Broadcasting large task binary with size 5.2 MiB
25/04/25 00:36:07 WARN InstanceBuilder: Failed to load implementation from:dev.ludovic.netlib.blas.JNIBLAS
25/04/25 00:36:08 WARN DAGScheduler: Broadcasting large task binary with size 5.2 MiB
25/04/25 00:36:09 WARN DAGScheduler: Broadcasting large task binary with size 5.2 MiB
25/04/25 00:36:09 WARN DAGScheduler: Broadcasting large task binary with size 5.2 MiB
25/04/25 00:36:09 WARN DAGScheduler: Broadcasting large task binary with size 5.2 MiB
25/04/25 00:36:10 WARN DAGScheduler: Broadcasting large task binary with size 5.2 MiB
25/04/25 00:36:10 WARN DAGScheduler: Broadcasting large task binary with size 5.2 MiB
25/04/25 00:36:10 WARN DAGScheduler: Broadcasting large task binary with size 5.2 MiB
25/04/25 00:36:10 WARN DAGScheduler: Broadcasting large task binary with size 5.2 MiB
25/04/25 00:36:11 WARN DAGScheduler: Broadcasting large task binary with size 5.2 MiB
25/04/25 00:36:11 WARN DAGScheduler: Broadcasting large task binary with size 5.2 MiB
25/04/25 00:36:11 WARN DAGScheduler: Broadcasting large task binary with size 5.2 MiB
25/04/25 00:36:11 WARN DAGScheduler: Broadcasting large task binary with size 5.2 MiB
25/04/25 00:36:12 WARN DAGScheduler: Broadcasting large task binary with size 5.2 MiB
25/04/25 00:36:12 WARN DAGScheduler: Broadcasting large task binary with size 5.2 MiB
25/04/25 00:36:13 WARN DAGScheduler: Broadcasting large task binary with size 5.2 MiB
25/04/25 00:36:13 WARN DAGScheduler: Broadcasting large task binary with size 5.2 MiB
25/04/25 00:36:13 WARN DAGScheduler: Broadcasting large task binary with size 5.2 MiB
25/04/25 00:36:13 WARN DAGScheduler: Broadcasting large task binary with size 5.2 MiB
25/04/25 00:36:14 WARN DAGScheduler: Broadcasting large task binary with size 5.2 MiB
25/04/25 00:36:14 WARN DAGScheduler: Broadcasting large task binary with size 5.2 MiB
25/04/25 00:36:14 WARN DAGScheduler: Broadcasting large task binary with size 5.2 MiB
✅ ALS training complete.
📊 Evaluating ALS model...
25/04/25 00:36:15 WARN DAGScheduler: Broadcasting large task binary with size 5.2 MiB
25/04/25 00:36:15 WARN DAGScheduler: Broadcasting large task binary with size 5.2 MiB
25/04/25 00:38:10 WARN DAGScheduler: Broadcasting large task binary with size 5.2 MiB
25/04/25 00:38:21 WARN DAGScheduler: Broadcasting large task binary with size 5.3 MiB
📈 ALS RMSE: 2.2358
💾 Saving model...
⚠️Model directory already exists: D:\Jaheem\jaheemProjetToDelete\COMP3610_bigDataAnalytics_A3-master\src\data\p
🎥 Showing top 5 recommendations for 3 sample users...
25/04/25 00:40:34 WARN DAGScheduler: Broadcasting large task binary with size 5.3 MiB
25/04/25 00:40:36 WARN DAGScheduler: Broadcasting large task binary with size 5.3 MiB
+---------+--------------------------------------------------------------------------------------------------+
|userIndex|recommendations                                                                                   |
+---------+--------------------------------------------------------------------------------------------------+
|7791     |[{11825, 4.984373}, {3061, 4.9710937}, {87316, 4.9589443}, {47698, 4.9589443}, {40731, 4.9589443}]|
|692      |[{40633, 5.1603026}, {12507, 5.1298347}, {2920, 5.0959735}, {11283, 5.024829}, {867, 5.0239267}]  |
|7487     |[{46583, 5.0308743}, {42821, 4.9803805}, {19008, 4.9803805}, {7317, 4.976742}, {12809, 4.9741945}]|
+---------+--------------------------------------------------------------------------------------------------+

✅ ALS recommendation pipeline completed.
```

Clustering / Segmentation (k-means)

Product segmentation was conducted using k-means clustering with k = 5 to group similar items based on selected features. Each product was represented by four attributes:

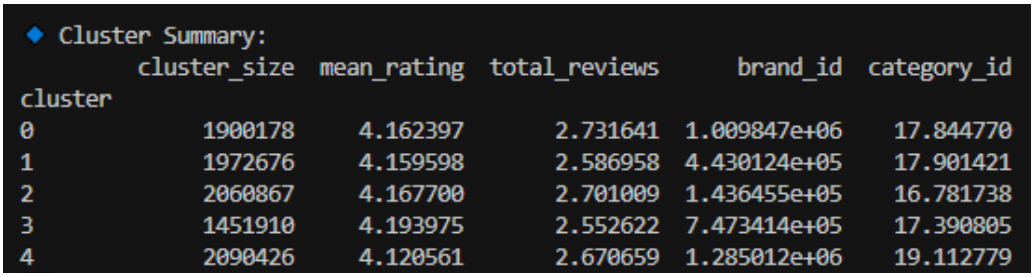
- ➔ Mean Rating: Calculated from merged review data.
- ➔ Total Reviews: Total number of reviews for each product.
- ➔ Brand ID: Categorical brand string mapped to an integer.
- ➔ Category ID: Top-level category string mapped to an integer.

To ensure balanced representation across categories, stratified sampling was applied by selecting 5% of the total data within each category for training. The k-means algorithm was executed with default initialization settings and iterated until convergence. Following clustering, each cluster was analyzed in terms of size, average mean rating, average total reviews, average brand ID, and average category ID, accompanied by a brief interpretation of the product types represented.

K-Means Clustering

The model grouped products based on the selected features, resulting in the following cluster statistics:

Cluster	Size	Avg. Mean Rating	Avg. Total Reviews	Avg. Brand ID	Avg. Category ID
0	1900178	4.162397	2.731641	1.089847e+06	17.844770
1	1972676	4.159598	2.580958	4.430124e+05	17.981421
2	2060867	4.167708	2.710089	1.436455e+05	16.781738
3	1451910	4.193975	2.552622	7.473441e+05	17.390805
4	2090426	4.120561	2.678659	1.285012e+06	19.112779



```
🔹 Cluster Summary:
         cluster_size  mean_rating  total_reviews     brand_id  category_id
cluster
0             1900178     4.162397       2.731641  1.089847e+06    17.844770
1             1972676     4.159598       2.586958  4.430124e+05    17.901421
2             2060867     4.167700       2.701009  1.436455e+05    16.781738
3             1451910     4.193975       2.552622  7.473414e+05    17.390805
4             2090426     4.120561       2.670659  1.285012e+06    19.112779
```

Cluster Analysis & Interpretation

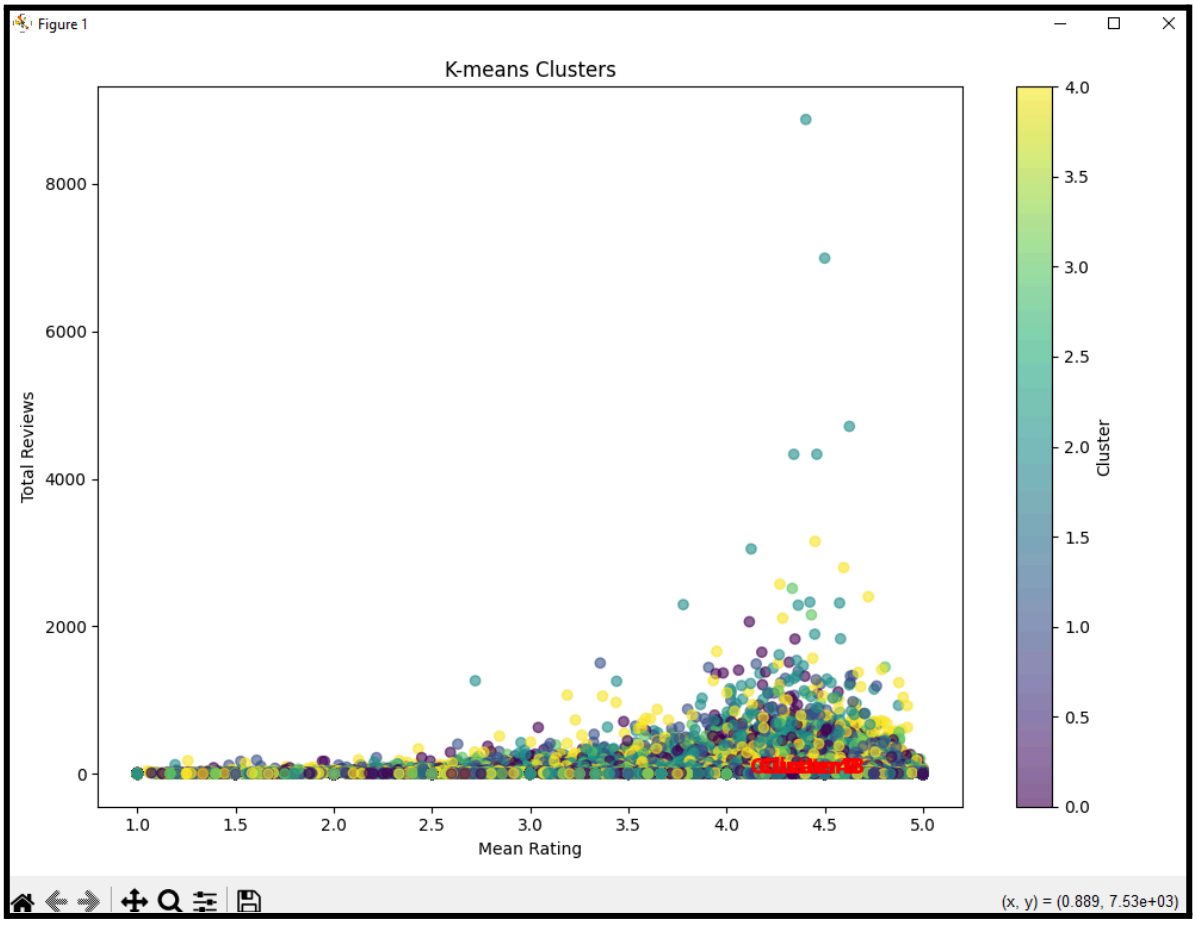
1. **Cluster 0**: This cluster represents well rated products from mid-range brands, showing a balanced profile in terms of brand recognition and category diversity.

2. **Cluster 1**: Products in this group are also popular and well rated, but tend to come from slightly less prominent brands, possibly indicating budget friendly or generic alternatives.

3. **Cluster 2**: This cluster includes high-rating products in more niche categories, typically associated with smaller or lesser known brands.

4. **Cluster 3**: The highest rated cluster overall, suggesting strong customer satisfaction and notable brand presence - possibly premium or flagship products.

5. **Cluster 4**: A diverse cluster with good overall ratings and representation from well known brands, likely covering a wide range of popular categories.



Code

Load and Sample Data:

```
print("🔹 Loading and sampling data...")
base_path =
r"D:\Jaheem\jaheemProjetToDelete\COMP3610_bigDataAnalytics_A3-master
\src\data\cleaned"
paths = [fr"{base_path}\cleaned_{category}.parquet" for category in
VALID_CATEGORIES]

df_list = []
for path, category in zip(paths, VALID_CATEGORIES):
    print(f"   - Reading and sampling 5% of data for category:
{category}")
    cat_df = pd.read_parquet(path, engine="pyarrow")
    sample = cat_df.sample(frac=0.05, random_state=42)
    df_list.append(sample)
df = pd.concat(df_list, ignore_index=True)
```

Feature Engineering:

```
print("🔹 Performing feature engineering...")
product_df = df.groupby('asin').agg({
    'rating': 'mean',
    'user_id': 'count',
    'brand': 'first',
    'main_category': 'first'
}).reset_index()
product_df.columns = ['asin', 'mean_rating', 'total_reviews',
'brand', 'category']
```

Encoding Categorical Features:

```
print("🔹 Encoding categorical features...")
le_brand = LabelEncoder()
le_cat = LabelEncoder()
product_df['brand_id'] =
le_brand.fit_transform(product_df['brand'].fillna("Unknown"))
product_df['category_id'] =
le_cat.fit_transform(product_df['category'].fillna("Unknown"))
```

Prepare Data & Apply Clustering:

```
print("🔹 Preparing data for clustering...")
features = product_df[['mean_rating', 'total_reviews', 'brand_id',
'category_id']]

print("🔹 Running KMeans clustering (k=5)...")
kmeans = KMeans(n_clusters=5, random_state=42)
product_df['cluster'] = kmeans.fit_predict(features)
```

Cluster Summary:

```
print("🔹 Clustering complete. Generating cluster summary...")
cluster_summary = product_df.groupby('cluster').agg({
    'asin': 'count',
    'mean_rating': 'mean',
    'total_reviews': 'mean',
    'brand_id': 'mean',
    'category_id': 'mean'
}).rename(columns={'asin': 'cluster_size'})

print("\n🔹 Cluster Summary:")
print(cluster_summary)
```

Save Clustered Data & Model:

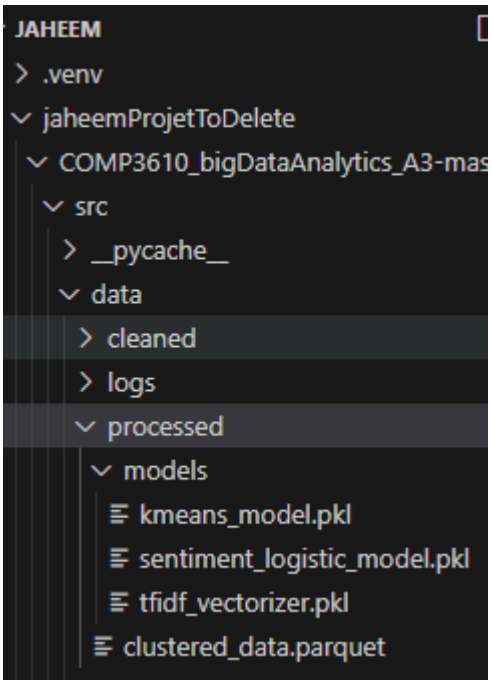
```
output_path =
r"D:\Jaheem\jaheemProjetToDelete\COMP3610_bigDataAnalytics_A3-master
\src\data\processed\clustered_data.parquet"
product_df.to_parquet(output_path, engine="pyarrow", index=False)
print(f"✅ Clustered data saved to {output_path}")

print("🔹 Saving KMeans model...")
model_dir =
r"D:\Jaheem\jaheemProjetToDelete\COMP3610_bigDataAnalytics_A3-master
\src\data\processed\models"
os.makedirs(model_dir, exist_ok=True)
model_path = os.path.join(model_dir, "kmeans_model.pkl")
joblib.dump(kmeans, model_path)
print(f"✅ KMeans model saved to {model_path}")
```

Plot Clusters:

```
print("🔹 Plotting clusters...")
plt.figure(figsize=(10, 7))
scatter = plt.scatter(product_df['mean_rating'],
product_df['total_reviews'],
                      c=product_df['cluster'], cmap='viridis',
alpha=0.6)
plt.xlabel('Mean Rating')
plt.ylabel('Total Reviews')
plt.title('K-means Clusters')
plt.colorbar(scatter, label='Cluster')

for i in range(5):
    cluster_data = product_df[product_df.cluster == i]
    plt.annotate(f"Cluster {i}",
                 (cluster_data['mean_rating'].mean(),
cluster_data['total_reviews'].mean()),
                 fontsize=12, weight='bold', color='red')

plt.tight_layout()
plt.show()

print("✅ Clustering script finished.")
```

```
(myenv) PS D:\Jaheem\jaheemProjetToDelete\COMP3610_bigDataAnalytics_A3-master\src> python clustering.py
🔹 Starting clustering process...
🔹 Loading and sampling data...
   - Reading and sampling 5% of data for category: All_Beauty
   - Reading and sampling 5% of data for category: Amazon_Fashion
   - Reading and sampling 5% of data for category: Appliances
   - Reading and sampling 5% of data for category: Arts_Crafts_and_Sewing
   - Reading and sampling 5% of data for category: Automotive
   - Reading and sampling 5% of data for category: Baby_Products
   - Reading and sampling 5% of data for category: Beauty_and_Personal_Care
   - Reading and sampling 5% of data for category: Books
   - Reading and sampling 5% of data for category: CDs_and_Vinyl
   - Reading and sampling 5% of data for category: Cell_Phones_and_Accessories
   - Reading and sampling 5% of data for category: Clothing_Shoes_and_Jewelry
   - Reading and sampling 5% of data for category: Digital_Music
   - Reading and sampling 5% of data for category: Electronics
   - Reading and sampling 5% of data for category: Gift_Cards
   - Reading and sampling 5% of data for category: Grocery_and_Gourmet_Food
   - Reading and sampling 5% of data for category: Handmade_Products
   - Reading and sampling 5% of data for category: Health_and_Household
   - Reading and sampling 5% of data for category: Health_and_Personal_Care
   - Reading and sampling 5% of data for category: Home_and_Kitchen
   - Reading and sampling 5% of data for category: Industrial_and_Scientific
   - Reading and sampling 5% of data for category: Kindle_Store
   - Reading and sampling 5% of data for category: Magazine_Subscriptions
   - Reading and sampling 5% of data for category: Movies_and_TV
   - Reading and sampling 5% of data for category: Musical_Instruments
   - Reading and sampling 5% of data for category: Office_Products
   - Reading and sampling 5% of data for category: Patio_Lawn_and_Garden
   - Reading and sampling 5% of data for category: Pet_Supplies
   - Reading and sampling 5% of data for category: Software
   - Reading and sampling 5% of data for category: Sports_and_Outdoors
   - Reading and sampling 5% of data for category: Subscription_Boxes
   - Reading and sampling 5% of data for category: Tools_and_Home_Improvement
   - Reading and sampling 5% of data for category: Toys_and_Games
   - Reading and sampling 5% of data for category: Video_Games
   - Reading and sampling 5% of data for category: Unknown
🔹 Performing feature engineering...
🔹 Encoding categorical features...
🔹 Preparing data for clustering...
🔹 Running KMeans clustering (k=5)...
🔹 Clustering complete. Generating cluster summary...

🔹 Cluster Summary:
         cluster_size  mean_rating  total_reviews      brand_id  category_id
cluster
0             1900178     4.162397       2.731641  1.009847e+06    17.844770
1             1972676     4.159598       2.586958  4.430124e+05    17.901421
2             2060867     4.167700       2.701009  1.436455e+05    16.781738
3             1451910     4.193975       2.552622  7.473414e+05    17.390805
4             2090426     4.120561       2.670659  1.285012e+06    19.112779
✅ Clustered data saved to D:\Jaheem\jaheemProjetToDelete\COMP3610_bigDataAnalytics_A3-master\src\data\processed\clustered_data.parquet
🔹 Saving KMeans model...
✅ KMeans model saved to D:\Jaheem\jaheemProjetToDelete\COMP3610_bigDataAnalytics_A3-master\src\data\processed\models\kmeans_model.pkl
🔹 Plotting clusters...
✅ Clustering script finished.
```

Models


JAHEEM
.venv
jaheemProjetToDelete
COMP3610_bigDataAnalytics_A3-mas
src
__pycache__
data
cleaned
logs
processed
models
kmeans_model.pkl
sentiment_logistic_model.pkl
tfidf_vectorizer.pkl
clustered_data.parquet

Conclusion

This project was developed and run on a personal desktop with a 64-bit AMD processor (around 3.7 GHz), 1.81 TB HDD and 32 GB of RAM, using Windows 10 Home. These resources were sufficient for handling large datasets using PySpark, allowing efficient data processing and machine learning tasks.

One of the main challenges was working with a large amount of data while staying within the limits of the system's memory. To deal with this, stratified sampling was used. Specifically, 5% of the data was sampled from each of the 34 product categories. This approach made sure that every category was included in the binary sentiment analysis, while also reducing the overall dataset size enough to make processing manageable.

The project included:
- A recommendation system using collaborative filtering with ALS
- Product segmentation using KMeans clustering based on features like mean rating, total reviews, brand ID, and category ID
- Sentiment analysis using logistic regression on sampled reviews
- Visualizations such as confusion matrices and performance bar charts

Some technical issues, like setting up Spark on Windows and dealing with memory warnings during large transformations, were resolved with configuration changes and sampling strategies.

Overall, the project successfully leveraged local hardware to perform big data analytics, and stratified sampling was key to making full-category analysis possible without overwhelming system resources.

Code Repository

GitHub - https://github.com/jaheemedwards/COMP3610_bigDataAnalytics_A3