**Project Name: Sentiment Analysis of Customer Reviews**

**(Amazon Customer Reviews with Sentiment)**

***Url:*** <https://www.kaggle.com/datasets/thedevastator/amazon-customer-reviews-with-2013-2019-sentiment>

***Date:*** 06.04.2024

***Participants:*** Arijit kr. Sarkar, Subha Poddar

***Technology used:*** MS Excel, CSV, My SQL, Microsoft SSMS

**Project description:**

This data analytics project will analyze Amazon customer reviews across diverse product categories using SQL. Sentiment analysis has been performed, categorizing reviews into positive, negative, or neutral sentiments. Insights will be derived from the dataset, uncovering trends and patterns in customer preferences and sentiments over a six-year period. Various SQL queries will be applied to extract meaningful information, such as identifying top-rated products, most frequently mentioned features, and overall sentiment distribution. The project aims to provide valuable insights for businesses to understand customer feedback and improve product offerings, enhancing customer satisfaction and retention.

**The rationale of this project:**

A customer review dataset from Amazon will be analyzed to gain insights into customer sentiment and product reception. The data includes review text, ratings, categories, and sentiment labels. This analysis aims to uncover trends in customer preferences and identify areas for product improvement across various categories. By examining review statistics, average ratings, and so on, the project seeks to understand customer satisfaction and guide data-driven decisions.

**Significance of this project:**

This project holds significant value for future real-life scenarios in multiple ways:

***Business Decision Making:*** By analyzing customer reviews and sentiments, businesses can gain actionable insights into customer preferences, enabling them to make informed decisions about product development, marketing strategies, and customer service improvements.

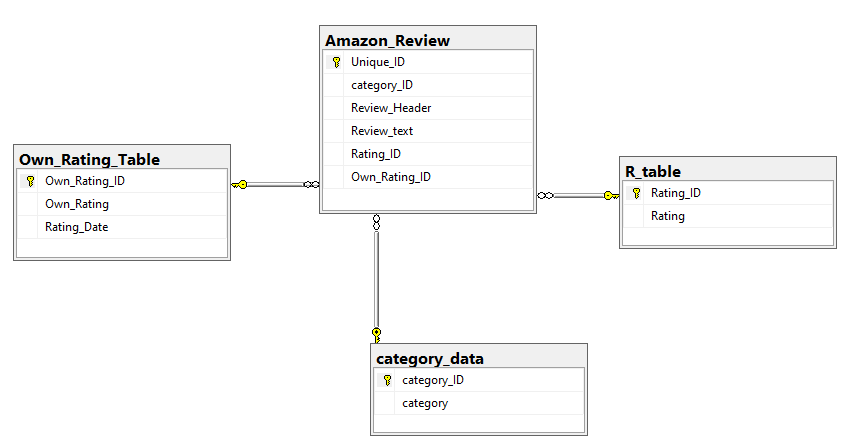
***Customer Satisfaction and Loyalty:*** Addressing issues highlighted in customer reviews can lead to improved customer satisfaction and loyalty, as businesses demonstrate responsiveness to customer feedback and strive to meet their expectations.

***Product Enhancement:*** Understanding the strengths and weaknesses of products as perceived by customers can guide companies in refining existing products and designing new ones that better meet consumer needs and preferences.

***Marketing and Communication Strategies:***

Insights derived from customer reviews can inform marketing and communication strategies, allowing businesses to tailor their messaging and promotions to resonate with target audiences effectively.

**ER Diagram:**



**CREATE Scripts**

***R\_table***

CREATE TABLE R\_table(

Rating\_ID int NOT NULL Primary Key,

Rating int NOT NULL);

***category\_data***

CREATE TABLE category\_data(

category\_ID int NOT NULL Primary Key,

category varchar(50));

***Own\_Rating\_Table***

create table Own\_Rating\_Table

(Own\_Rating\_ID int primary key,

Own\_Rating nvarchar(10),

Rating\_Date Date);

***Amazon\_Review***

CREATE TABLE Amazon\_Review(

Unique\_ID INT PRIMARY KEY,

Category\_ID INT NOT NULL,

Review\_Header NVARCHAR(MAX),

Review\_text NVARCHAR(MAX),

Rating\_ID INT NOT NULL,

Own\_Rating\_ID INT NOT NULL,

FOREIGN KEY (Category\_ID) REFERENCES category\_data(Category\_ID),

FOREIGN KEY (Rating\_ID) REFERENCES R\_table(Rating\_ID),

FOREIGN KEY (Own\_Rating\_ID) REFERENCES Own\_Rating\_Table(Own\_Rating\_ID)

);

**INSERT Scripts**

***R\_table***

INSERT INTO R\_table (Rating\_ID, Rating)

SELECT Rating\_ID, Rating

FROM Rating\_table;

***category\_data***

INSERT INTO category\_data( category\_ID, category)

SELECT category\_ID, category

FROM Category\_table;

***Own\_Rating\_Table***

Insert into Own\_Rating\_Table (Own\_Rating\_ID, Own\_Rating, Rating\_Date)

select Own\_Rating\_ID, Own\_Rating, Rating\_Date from

Own\_Rating;

***Amazon\_Review***

INSERT INTO Amazon\_Review (Unique\_ID, category\_ID, Review\_Header, Review\_text, Rating\_ID, Own\_Rating\_ID)

SELECT Unique\_ID, category\_ID, Review\_Header, Review\_text, Rating\_ID, Own\_Rating\_ID

FROM Amazon\_Review\_Data;

**Analysis**

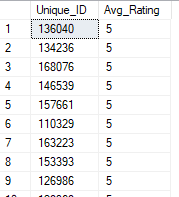
***1. Problem statement***

***Finding the average rating for each product***

**Query**

| **--the average rating for each product**  **SELECT ar.Unique\_ID, AVG(r.Rating) AS Avg\_Rating**  **FROM amazon\_review ar**  **JOIN r\_table r ON ar.Rating\_ID = r.Rating\_ID**  **GROUP BY ar.Unique\_ID**  **ORDER BY Avg\_Rating DESC;** |
| --- |

**Output**



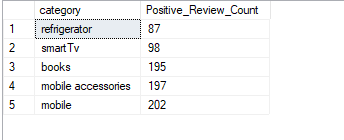
***2. Problem statement***

***Recognizing the number of reviews with positive sentiment for each category***

**Query**

| **SELECT c.category, COUNT(\*) AS Positive\_Review\_Count**  **FROM amazon\_review ar**  **JOIN category\_data c ON ar.category\_ID = c.category\_ID**  **JOIN Own\_Rating\_Table ort ON ar.Own\_Rating\_ID = ort.Own\_Rating\_ID**  **WHERE ort.Own\_Rating = 'Positive'**  **GROUP BY c.category**  **ORDER BY Positive\_Review\_Count;** |
| --- |

***Output***



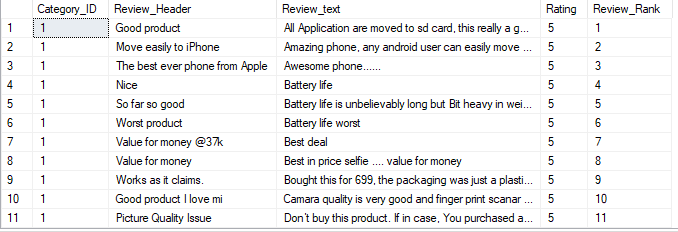
3. ***Problem statement***

***Rank Reviews Within Each Category by Rating***

**Query**

| **SELECT ar.Category\_ID, ar.Review\_Header, ar.Review\_text, r.Rating,**  **ROW\_NUMBER() OVER(PARTITION BY ar.Category\_ID ORDER BY r.Rating DESC) AS Review\_Rank**  **FROM amazon\_review ar**  **JOIN r\_table r ON ar.Rating\_ID = r.Rating\_ID;** |
| --- |

***Output***



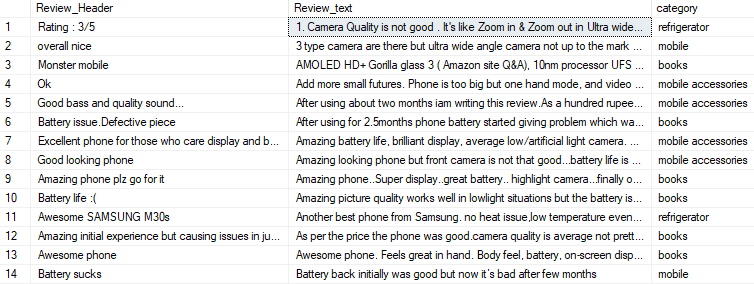
4. ***Problem statement***

***Identifying all reviews containing the word "battery" in the review text***

**Query**

| **SELECT ar.Review\_Header, ar.Review\_text, c.category**  **FROM amazon\_review ar**  **JOIN category\_data c ON ar.category\_ID = c.category\_ID**  **WHERE ar.Review\_text LIKE '%battery%';** |
| --- |

***Output***



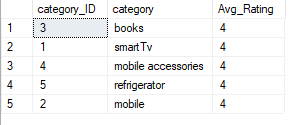
5. ***Problem statement***

***Identifying Top-Rated Amazon Products with Average Rating of Higher than 3***

**Query**

| **SELECT ar.category\_ID, c.category, AVG(r.Rating) AS Avg\_Rating**  **FROM amazon\_review ar**  **JOIN category\_data c ON ar.category\_ID = c.category\_ID**  **JOIN r\_table r ON ar.Rating\_ID = r.Rating\_ID**  **GROUP BY ar.category\_ID, c.category**  **HAVING AVG(r.Rating) > 3**  **ORDER BY Avg\_Rating DESC;** |
| --- |

***Output***



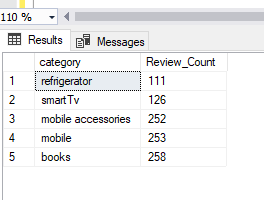
***6. Problem statement***

***Find the Most Reviewed Category***

**Query**

| **SELECT c.category, COUNT(\*) AS Review\_Count**  **FROM amazon\_review ar**  **JOIN category\_data c ON ar.Category\_ID = c.Category\_ID**  **GROUP BY c.Category**  **ORDER BY Review\_Count;** |
| --- |

***Output***



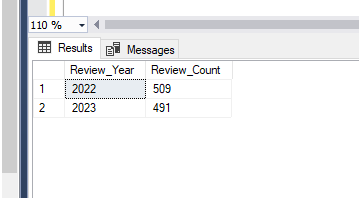
***7. Problem statement***

***Count the Number of Reviews per Year***

**Query**

| **SELECT YEAR(Rating\_Date) AS Review\_Year, COUNT(\*) AS Review\_Count**  **FROM Own\_Rating\_table**  **GROUP BY YEAR(Rating\_Date)**  **ORDER BY Review\_Year;** |
| --- |

***Output***



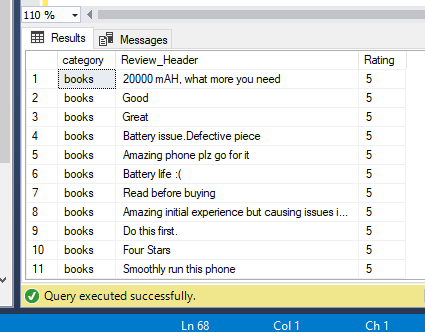
***8. Problem statement***

***Find the Products with Ratings Greater Than 4***

**Query**

| **SELECT c.category, ar.Review\_Header, r.Rating**  **FROM amazon\_review ar**  **JOIN category\_data c ON ar.category\_ID = c.category\_ID**  **JOIN R\_table r ON ar.Rating\_ID = r.Rating\_ID**  **WHERE r.Rating > 4**  **order by c.category;** |
| --- |

***Output***

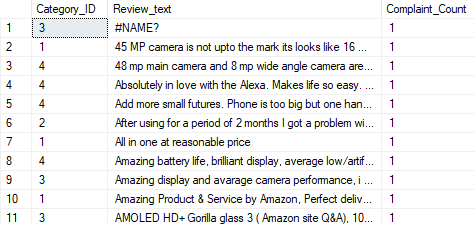
******

***9. Problem statement***

***Identify Most Common Complaints Across Categories***

**Query**

| **SELECT ar.Category\_ID, ar.Review\_text, COUNT(\*) AS Complaint\_Count**  **FROM amazon\_review ar**  **WHERE ar.Own\_Rating\_ID IN (SELECT Own\_Rating\_ID FROM Own\_Rating\_table WHERE Own\_Rating = 'Negative')**  **GROUP BY ar.Category\_ID, ar.Review\_text**  **ORDER BY Complaint\_Count DESC;** |
| --- |



***10. Problem statement***

***Show the 10 most recent reviews with their corresponding ratings and categories and rating date***

**Query**

| **SELECT TOP 10 ar.Review\_Header, r.Rating, c.category, ort.Rating\_Date**  **FROM amazon\_review ar**  **JOIN r\_table r ON ar.Rating\_ID = r.Rating\_ID**  **JOIN category\_data c ON ar.category\_ID = c.category\_ID**  **JOIN Own\_Rating\_Table ort ON ar.Own\_Rating\_ID = ort.Own\_Rating\_ID**  **ORDER BY ort.Rating\_Date DESC;** |
| --- |

