



e-ISSN: 2319-8753 | p-ISSN: 2347-6710

# IJRSET

International Journal of Innovative Research in  
**SCIENCE | ENGINEERING | TECHNOLOGY**



# INTERNATIONAL JOURNAL OF INNOVATIVE RESEARCH

IN SCIENCE | ENGINEERING | TECHNOLOGY

Volume 13, Issue 3, March 2024

**ISSN** INTERNATIONAL  
STANDARD  
SERIAL  
NUMBER  
INDIA

Impact Factor: 8.423

9940 572 462

6381 907 438

[ijrset@gmail.com](mailto:ijrset@gmail.com)

[www.ijrset.com](http://www.ijrset.com)

# Building a Comprehensive E-commerce Analytics Ecosystem

Dr. T. Sudhir<sup>\*1</sup>, Mutyala Harini<sup>2</sup>, Kolipakula Dhanalakshmi<sup>3</sup>, Yallanki Pavithra Lakshmi<sup>4</sup>, Joshna Gummadi<sup>5</sup>, Kakarla Prasanna<sup>6</sup>

Professor, Department of Artificial Intelligence & Data Science, Vasireddy Venkatadri Institute of Technology, Guntur, Andhra Pradesh, India<sup>\*1</sup>

UG Students, Department of Artificial Intelligence & Data Science, Vasireddy Venkatadri Institute of Technology, Guntur, Andhra Pradesh, India<sup>2,3,4,5,6</sup>

**ABSTRACT:** In the highly competitive world of business-to-consumer e-commerce, knowing how customers behave is essential. This concept describes an ecosystem for advanced analytics that incorporates several aspects of e-commerce data. By methodically gathering product information from various platforms such as Flipkart, it facilitates a thorough comprehension of the product catalog. It offers insightful sentiment research on customer reviews by using natural language processing. For effective storing and retrieval of dynamic data, a NoSQL database is used. It additionally simplifies data structure and classifies products. The project's main feature is a strong recommendation system that uses user preferences and sentiment analysis to provide tailored recommendations. This system improves user experience and increases business profitability, demonstrating the power of NoSQL databases in contemporary e-commerce analytics.

**KEYWORDS:** B2C e-commerce, Consumer behavior, Sentiment analysis, NoSQL database, Product categorization.

## I. INTRODUCTION

Understanding customer behavior and preferences is essential for corporate success in the ever-changing world of business-to-consumer (B2C) e-commerce. By building a comprehensive analytics platform that smoothly combines many aspects of e-commerce data, our project aims to satisfy this requirement. The project guarantees a comprehensive comprehension of the vast online product catalog by methodically extracting product details from many B2C websites. By utilizing cutting-edge natural language processing methods, the project delves into sentiment analysis of client comments, providing organizations with priceless insights into customer happiness. A NoSQL database, which prioritizes efficiency and adaptability, is used to handle the large volume and dynamic nature of e-commerce data by providing scalability and flexibility for both storage and retrieval.

### A. Sentiment Analysis

Sentiment analysis, sometimes referred to as opinion mining, is a natural language processing method for figuring out the sentiment or emotional tone that is expressed in a document. Sentiment analysis in e-commerce examines user reviews, comments on social media, and other textual data to determine how customers feel about certain companies, goods, or services. To give organizations useful information about client preferences, satisfaction levels, and market trends, the primary goal is to categorize these feelings as positive, negative, or neutral.

Sentiment analysis plays a critical role in e-commerce. Businesses may use it to obtain insightful information about client attitudes and use that information to inform data-driven decisions that will improve customer satisfaction and increase revenue. E-commerce platforms may determine their offers' advantages and disadvantages by examining product reviews, which helps inform future product development and marketing initiatives. Additionally, sentiment analysis supports reputation management by quickly identifying and resolving unfavorable attitudes, protecting a brand's reputation, and building consumer confidence. Sentiment research also provides guidance for focused marketing efforts and individualized suggestions, guaranteeing that consumers are presented with interesting and

pertinent material. In general, sentiment research gives e-commerce companies the ability to maintain their competitiveness, adjust to changing consumer preferences, and ultimately increase client loyalty and happiness.

### B. Doc-Based

Document-based databases, often referred to as document stores, are a subset of NoSQL databases used for managing, storing, and retrieving documents containing semi-structured data. Document databases store data as adaptable, self-describing documents, usually in formats like JSON or BSON, as opposed to conventional relational databases, which arrange data into tables and rows. Because nested data structures may be found in any page, they are perfect for storing intricate, hierarchical data. Since document-based databases do not rely on schemas, individual documents can have distinct structures that can be dynamically adjusted to meet evolving data needs. Because of their excellent performance managing unstructured or dynamic data, these databases are becoming more and more common in contemporary application development situations, such as e-commerce.

Document-based databases are essential for handling a variety of dynamic data sets in the e-commerce environment. They allow e-commerce systems to store pertinent data, such as transaction details, customer profiles, and product information, in an organized, flexible, and scalable manner. Product catalogs, for example, may be seen as documents in e-commerce systems, where each document has all of the information on a product, such as its name, description, price, photos, and variants. This makes it simple to access and modify product data, which supports customized product suggestions, focused advertising efforts, and efficient inventory control.

Additionally, document-based databases enable e-commerce companies to provide customers with smooth and customized purchasing experiences. E-commerce systems may customize product suggestions, promotions, and content according to individual consumer preferences by maintaining customer profiles as documents that include their browsing history, purchase behavior, and preferences. Furthermore, real-time analysis of client interactions is made possible by document-based databases, which gives organizations the ability to spot market trends, understand consumer behavior, and make data-driven decisions that improve the customer experience and increase sales.

E-commerce platforms that aim to effectively handle vast amounts of heterogeneous and constantly changing data might benefit greatly from the utilization of document-based databases. Document databases are ideal for storing and retrieving complicated, semi-structured data because they offer a flexible schema-less data format that can be tailored to meet the specific needs of e-commerce applications. In the cutthroat world of e-commerce, companies may increase productivity, make better decisions, and provide consumers with individualized experiences by utilizing document-based databases. These actions will eventually increase customer engagement, loyalty, and income.

The following sections include literature, followed by the implementation of proposed model

## II. LITERATURE

### A. SQL VS NoSQL

SQL (Structured Query Language) and NoSQL (Not Only SQL) databases represent two distinct approaches to data management, each offering unique advantages and challenges. This literature review synthesizes research comparing SQL and NoSQL databases, elucidating their differences, strengths, and application contexts.

1. Comparative Performance Analysis  
Li and Manoharan (2013) conducted a comparative performance analysis of SQL and NoSQL databases, focusing on factors such as data retrieval speed, scalability, and concurrency control. Their study provided empirical insights into the relative efficiency and effectiveness of SQL and NoSQL technologies in diverse use cases.
2. Database Management System Comparison  
Boicea et al. (2012) undertook a comprehensive comparison between specific database management systems, namely MongoDB (NoSQL) and Oracle (SQL). Their study examined various aspects including data storage model, query language, transaction support, and scalability, offering insights into the practical differences between SQL and NoSQL implementations.
3. Advantages and Disadvantages

Mohammed (2015) discussed the respective advantages and disadvantages of SQL and NoSQL databases. The review highlighted NoSQL's benefits such as schema flexibility and horizontal scalability, alongside challenges like lack of standardization and reporting tools. Through this comparison, Mohammed provided a nuanced understanding of the suitability of each approach in different scenarios.

#### 4. CAP Theorem and Database Design

Sabri (2015) explored the implications of the CAP (Consistency, Availability, Partition tolerance) theorem on database design, comparing SQL and NoSQL databases in light of this theoretical framework. The review elucidated how NoSQL prioritizes availability and partition tolerance over strong consistency, contrasting with SQL databases' adherence to the ACID properties. Sabri's analysis shed light on the architectural distinctions between SQL and NoSQL databases.

While SQL databases offer strong consistency and transactional guarantees, NoSQL databases excel in scalability, flexibility, and performance in distributed environments. The choice between SQL and NoSQL hinges on factors such as data structure, scalability requirements, and application complexity, emphasizing the importance of understanding the trade-offs between these two paradigms.

### B. MongoDB over other NoSQL Databases

MongoDB is a leader in the field of NoSQL databases, surpassing its competitors in a number of areas. The comparison study that is provided shows that MongoDB is superior, particularly when taking into account popularity, performance, and sentiment handling appropriateness. Because of its ability to handle massive amounts of changing data and its concurrent behavior similar to that of SQL, MongoDB is a popular option among researchers and data practitioners. MongoDB has in-memory capabilities, which sets it apart from certain other NoSQL databases like CouchDB. This makes MongoDB an appealing choice for applications that require high speed and real-time data processing.

Moreover, MongoDB's success and broad acceptance demonstrate how well it works to solve the problems that the modern data science period presents. A database that can effectively manage such dynamic data is essential since social network attitudes are changing at an exponential rate. Because of its flexibility and scalability, MongoDB is an excellent choice for modeling feelings and doing system behavioral investigations. Because of its document-based methodology, which makes it easier to read and analyze large sentiment data, practitioners may get important insights and make predictions that will improve system performance and profitability.

In the field of sentiment analysis and behavioral research, MongoDB has clearly outperformed alternative NoSQL databases. It is a strong option for managing changing data types and meeting the demands of contemporary data science applications because of its strong characteristics, which include concurrent behavior, scalability, and in-memory capabilities. MongoDB is a strong competitor because it provides the flexibility, performance, and dependability needed to handle the needs of the always changing data landscape, even while sentiments continue to play a crucial part in understanding system behavior and guiding predictive analytics.

## III. PROPOSED WORK

### A. Why

Understanding customer behavior and preferences is essential for businesses to succeed in the ever-changing world of e-commerce. Finding relevant insights from B2C websites is difficult, though, because of the abundance and variety of data present. Current systems frequently have issues with real-time updates, scalable data, and tailored suggestions, which makes it harder for companies to remain competitive. In order to provide organizations with actionable insights and improve the user experience, a sophisticated analytics ecosystem that seamlessly integrates data retrieval, sentiment analysis, and recommendation systems is therefore desperately needed.

### B. Model Explanation

For this model, the e-commerce website <https://www.flipkart.com/> was used, and we used certain tags on the website's inspect page to extract all of the product data's information.

The proposed model takes Flipkart URL as input from the user and extracts the information as mentioned earlier as shown in figure 4.



As seen in Figure 1, a database called Flipkart Item is constructed, and inside the database, many collections representing different product categories are made. Figure 2 displays the many Musical Instrument goods that are kept in the Musical Instrument collection.

From the website based on the Product name, product id, price, image URLs, reviews: rating and review content of each product data is created as a document sample in the collection as shown in Figure 3

In order to determine how well a product meets the demands of the consumer, Sentiment Analysis is carried out on customer-provided reviews and ratings that are kept in their relevant product category collections in MongoDB.

Lastly, it offers a selection of products based on ratings, sentiment scores, and the user's chosen product category. For the purpose of gaining a competitive edge, company owners might modify their goods by comparing them to those of their competitors.

The output is depicted in figure 5.

#### IV. IMPLEMENTATION

We utilized Spyder (Python 3.7) to implement the code for this suggested model. The code that has been built below facilitates the extraction of material from the Flipkart website and its subsequent storage in the Mongo dB database.

Each product field is saved as a document within its corresponding collection, which functions as a product category.

The code that follows is used to retrieve data from a website and save it in a MongoDB database.

A.Code Snippet:

```
Code for extracting Data from website
response = requests.get(url, headers=headers)
soup = BeautifulSoup(response.content, 'html.parser')
title = soup.find('span', class_="B_NuCI").text.strip()
price = soup.find('div', class_="_30jeq3_16Jk6d").text.strip()
breadcrumbs = soup.find_all('a', class_="_2whKao")
categories = [breadcrumb.text for breadcrumb in breadcrumbs if breadcrumb.text != "Home"]
# Find all elements with class 'q6DCIP'
elements_with_q6DCIP_class = soup.find_all(class_='q6DCIP')
# Extract src attribute of each element
image_urls = [element['src'] for element in elements_with_q6DCIP_class if 'src' in element.attrs]
product_info={
    "Title": title,
    "ProductId":pid,
    "Price": price,
    "Categories": categories
    "Images": image_urls,
}
category = product_info["Categories"][0]
collection = db[category]
# Insert the product information into the collection
collection.insert_one(product_info)
print("Product information stored successfully in MongoDB!")
```

When this code is implemented, it creates a database with the specified name, collections, and an object id for each document in the collection if the database is not already in the MongoDB databases list. This code yields the following outcome.

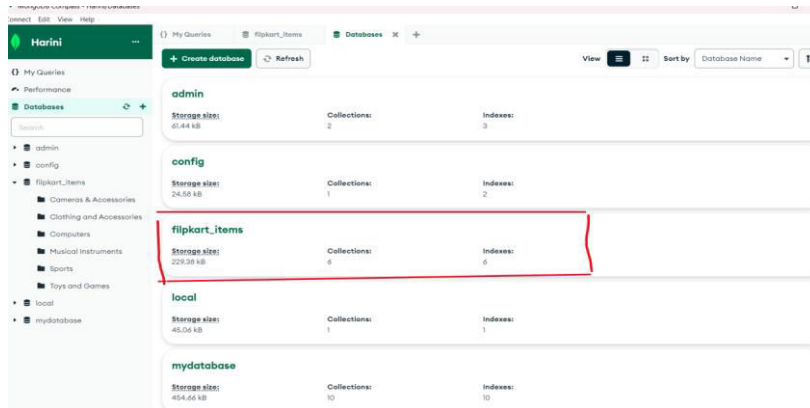


Figure 1: Database in MongoDB Server

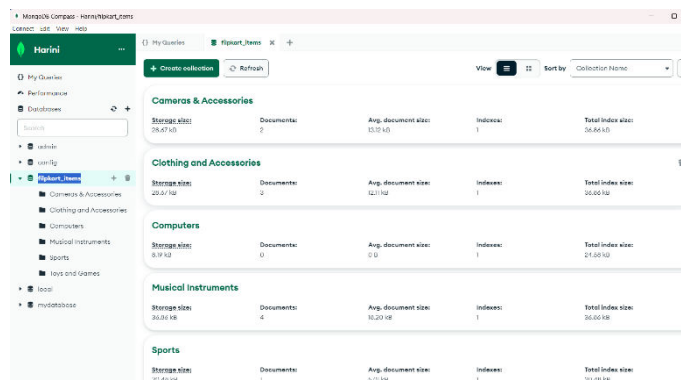


Figure 2: Collections in Flipkart\_items after execution of code

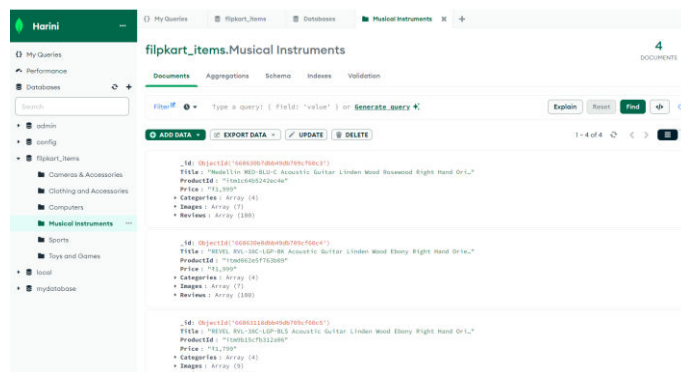


Figure 3: Each collection containing the data of each product

Sentiment analysis is performed on user-submitted evaluations and ratings that are stored in their respective product category collections in MongoDB. The objective of this analysis is to ascertain the extent to which a product satisfies customer requests.

The code iterates through the "Reviews" array within existing\_product, calculates sentiment scores for each review using analyze\_sentiment, and computes the overall average rating and sentiment score for all reviews. The results are stored in overall\_rating and overall\_sentiment\_score.

**B. Code Snippet**

```

total_rating = 0
total_sentiment_score = 0
num_reviews = len(reviews)
if num_reviews == 0:
    
```



```

return None
for review in reviews:
    review_rating = review["review_rating"]
    review_content = review["review_content"]
    # Calculate sentiment score for review content
    sentiment_score=analyze_sentiment(review_content)
    # Convert sentiment score to equivalent rating (assuming linear mapping for simplicity)
    sentiment_rating = (sentiment_score + 1) * 5 / 2
    # Handle cases where "Rating not available" is present in reviews
    if review_rating != "Rating not available":
        total_rating += int(review_rating)
        total_sentiment_score += sentiment_score
    else:
        total_rating += sentiment_rating
        total_sentiment_score += sentiment_score
# Calculate average rating
average_rating = total_rating / num_reviews
# Calculate average sentiment score
average_sentiment_score = total_sentiment_score / num_reviews
return average_rating, average_sentiment_score
    
```

The Execution of code also categorizes the sentiment scores into Best, good, neutral, bad, worst.

### C. Code Snippet

```

good_threshold = 0.6 # Adjust as needed
bad_threshold = 0.4 # Adjust as needed
worst_threshold = 0.2 # Adjust as needed

# Initialize counters for good, bad, and worst reviews
num_good_reviews = 0
num_bad_reviews = 0
num_worst_reviews = 0
num_best_reviews = 0
num_neutral_reviews = 0
for review in reviews:
    review_content = review["review_content"]
    # Calculate sentiment score for review content    sentiment_score=analyze_sentiment(review_content)
    # Categorize the review based on sentiment
    score
    if sentiment_score >= 0.75:
        num_best_reviews += 1
    elif sentiment_score >= 0.4:
        num_good_reviews += 1
    elif sentiment_score >= 0.25:
        num_neutral_reviews += 1
    elif sentiment_score >= 0.1:
        num_bad_reviews += 1
    else:
        num_worst_reviews += 1
    
```

Finally, We implemented a recommendation system that aggregates different MongoDB operations inorder to recommend products based on user product category preference , rating and sentiment score.

### D. Code Snippet

The sample code is given below:  
 pipeline = [  
 {



```
"$addFields": {
  "num_reviews": {"$size": "$Reviews"},
  "overall_rating": {
    "$avg": {
      "$map": {
        "input": "$Reviews",
        "as": "review",
        "in": {
          "$toInt": "$$review.review_rating"
        }
      }
    }
  }
},
{
  "$match": {
    "overall_rating": {"$gte": min_rating}
  }
},
{
  "$sort": {"overall_rating": -1}
}
] items=db[your_collection_name].aggregate(pipeline)
```

We finally added HTML and CSS to code for user Interaction and also For the Visualization of output.

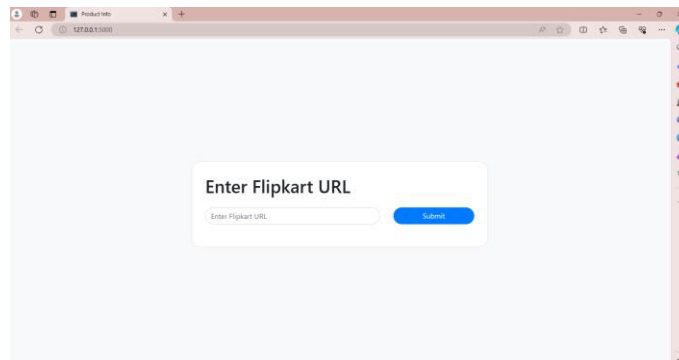


Figure 4: User Interface



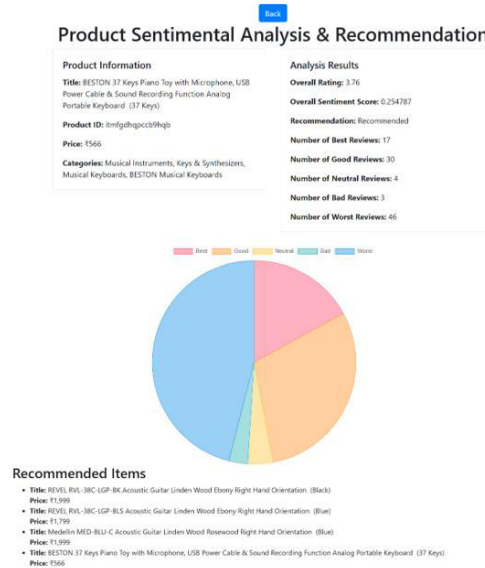


Figure 5: Output of overall code execution

### V. ADVANTAGES

- Extensive data retrieval guarantees current product insights.
- Sentiment analysis offers insightful commentary on client happiness.
- Scalability and flexibility are provided by NoSQL databases for effective data storage.
- Simplified product classification improves accessibility and data organization.
- A personalized recommendation system increases revenue for businesses and increases user engagement.

### VI. CONCLUSION

The Proposed analytics ecosystem provides a comprehensive answer to the problems associated with business-to-consumer e-commerce. Businesses may obtain important insights into customer behavior, improve user experience, and boost profitability by combining sentiment analysis, extensive data retrieval, and a strong recommendation engine driven by a NoSQL database. This strategy opens the door for increased market success and competitiveness by demonstrating the value of data-driven decision-making in contemporary e-commerce contexts.

### REFERENCES

- [1] Arun Monicka, M. Raja, Winster S. Godfrey and S. Swamynathan, "Review Analyzer: Analyzing Consumer Product Reviews from Review Collections", IEEE International Conference on Recent Advances in Computing and Software Systems, 2012.
- [2] Geng Cui, Hun-Kwong Lui and Xiaoning Guo, "Online Reviews as a Driver of New Product Sales", IEEE International Conference on Management of e-Commerce and e-Government, 2010.
- [3] Mustafa Shuaieb Sabri, Harrisburg University "COMPARATIVE STUDY OF DISTRIBUTED, SCALABLE, & HIGH PERFORMANCE NOSQL DATABASES", January 2015SSRN Electronic Journal 3(1):28-35
- [4] Innocent Mapanga, Prudence Kadebu "Database Management Systems: A NoSQL Analysis", International Journal of Modern Communication Technologies & Research (IJMCTR) ISSN: 2321- 0850, Volume-1, Issue-7, September 2013
- [5] C.M. Umair, "The Trend of Online Shopping in 21st Century: Impact of Enjoyment in TAM Model", Asian Journal of Empirical Research, pp. 131-141, 2013.



- [6] Rupali Arora, Rinkle Rani Aggarwal “Modeling and Querying Data in MongoDB”, International Journal of Scientific & Engineering Research, Volume 4, Issue 7, July-2013,141 ISSN 2229-5518
- [7] G. Dubey, S. Chawla and K. Kaur, "Social media opinion analysis for indian political diplomats", Proceedings of the 2017 7th International Conference on Cloud Computing Data Science Engineering, pp. 681-686, 12–13 January 2017.
- [8] Geng Cui, Hun-Kwong Lui and Xiaoning Guo, "Online Reviews as a Driver of New Product Sales", IEEE International Conference on Management of e-Commerce and e-Government, 2010.
- [9] Chintan R. Varnagar, Nirali N. Madhak, Trupti M. Kodinariya, Jayesh N. Rathod, "Web usage mining: A review on process, methods and techniques", 2013 International Conference on Information Communication and Embedded Systems (ICICES), pp.40-46, 2013.



INTERNATIONAL  
STANDARD  
SERIAL  
NUMBER  
INDIA



# INTERNATIONAL JOURNAL OF INNOVATIVE RESEARCH

IN SCIENCE | ENGINEERING | TECHNOLOGY

 9940 572 462  6381 907 438  [ijirset@gmail.com](mailto:ijirset@gmail.com)



[www.ijirset.com](http://www.ijirset.com)

Scan to save the contact details