ReviewLens

GitHub Link - Slideshow Link - Original Data

In case GitHub link is not clickable, use this url: https://github.com/KR-0000/ReviewLens

Group Details

Members: Rishabh Sheth (rs2299), Kushagra Dhall (kd983), Joshua Dsouza (jd2038)

Project Definition

Shoppers online experience a major challenge in gauging product quality due to the huge volume of customer reviews available on e-commerce platforms. Many reviews are subjective, surface-level or even redundant, making it difficult for consumers to make their purchasing decisions. The rapid growth of online shopping has made summarizing product reviews a core data management challenge.

ReviewLens addresses these challenges by building a complete end-to-end system that ingests raw Amazon product review data, cleans and normalizes it, extracts meaningful insights using advanced Natural Language Processing (NLP), and then delivers product-level summaries, sentiment breakdowns, and keyword trends through a modern web interface. The system transforms messy, unstructured text into actionable information for both consumers and businesses.

Thus, the core problem is: How can we reliably and transparently extract structured insights from large volumes of unstructured customer reviews using data management principles and machine learning?

Strategic Aspects and Relation to Class

This project synthesizes all stages of the data lifecycle emphasized in the course:

- Data collection & ingestion: loading raw CSV review data using batch processing.
- Data cleaning & normalization: converting denormalized review records into a normalized SQL schema following 1NF/2NF/3NF principles.
- Storage choices: structured data in SQL; raw text processing performed in Python
- Data transformation: using NLP to compute sentiment, extract phrases, and generate summaries.
- Querying & retrieval: providing product-level insights through SQL joins and API-layer search functionality.
- Communication: visualizing results in a React-based interface.

ReviewLens directly reflects topics from lectures and papers on:

- data normalization
- information retrieval,
- unstructured text processing
- scalable data pipelines

- model interpretability

The project extends beyond class content by incorporating abstractive summarization and advanced NLP pipelines.

Novelty and Importance

While major e-commerce platforms such as Amazon offer AI-generated review summaries, these systems operate as proprietary black boxes. Users cannot verify how summaries are generated, which data is prioritized, or whether the system is optimized to highlight positive aspects to drive sales.

ReviewLens takes an alternative approach:

- Transparent, open methods
- Explainable algorithms
- Independent, unbiased analysis

This improves both trustworthiness and utility.
For shoppers, ReviewLens reduces hours of reading into seconds of insight, and for businesses, it highlights validated sentiment trends, both positive themes and recurring pain points.

Current Issues

Processing customer reviews presents several real-world difficulties. The sheer number of reviews for many products makes manual reading impossible. Reviews also come in unstructured, free-form text, requiring sophisticated cleaning and parsing methods. Another ongoing issue is review incompleteness. Users usually talk about the same few features repeatedly in their reviews and skip over others entirely, which leads to an incomplete picture of how the product actually performs. As mentioned before, existing tools that do something similar to our project are usually operated by a proprietary company, which can be a problem for reliability. This is because it is in the e-commerce website’s best interest to highlight a product's positive features while minimizing the severity of its negatives.

Related Works

Existing work in this domain includes both industrial and academic efforts. Amazon’s own AI-generated “Customer Summary” feature provides a real-world example of automated review summarization, but because it is proprietary, the underlying methodology is not disclosed, and its objectivity cannot be guaranteed. Academic research in aspect-based sentiment analysis, such as studies using transformer-based models like BERT or summarization techniques like TextRank and PEGASUS, has shown strong performance in extracting meaningful product features from text. However, the problem with this is that most existing research focuses either on model accuracy or theoretical improvements, rather than building a complete product that integrates database management, NLP pipelines, and user-facing visualization. Our work will help address this gap by creating an end-to-end and interpretable system.

Progress and Contribution

Dataset

ReviewLens uses the Kaggle dataset:
Consumer Reviews of Amazon Products (Datafiniti)
containing ~34,000 reviews with 24 metadata fields including:

- product name, brand, manufacturer
- review text and title
- timestamps
- number of helpful votes
- image URLs
- rating and category information

The data was ingested using a custom CSV loader (load_data.py) that performs:

- batch loading (1,000 rows at a time)
- type casting and field normalization
- safe parsing with missing-value handling
- indexing on key fields (name, processed)
- dataset validation (review count, unique product count)

This ensures the pipeline scales and maintains data integrity.

Data Normalization & Database Engineering

A full relational schema was designed to achieve 1NF, 2NF, and 3NF:

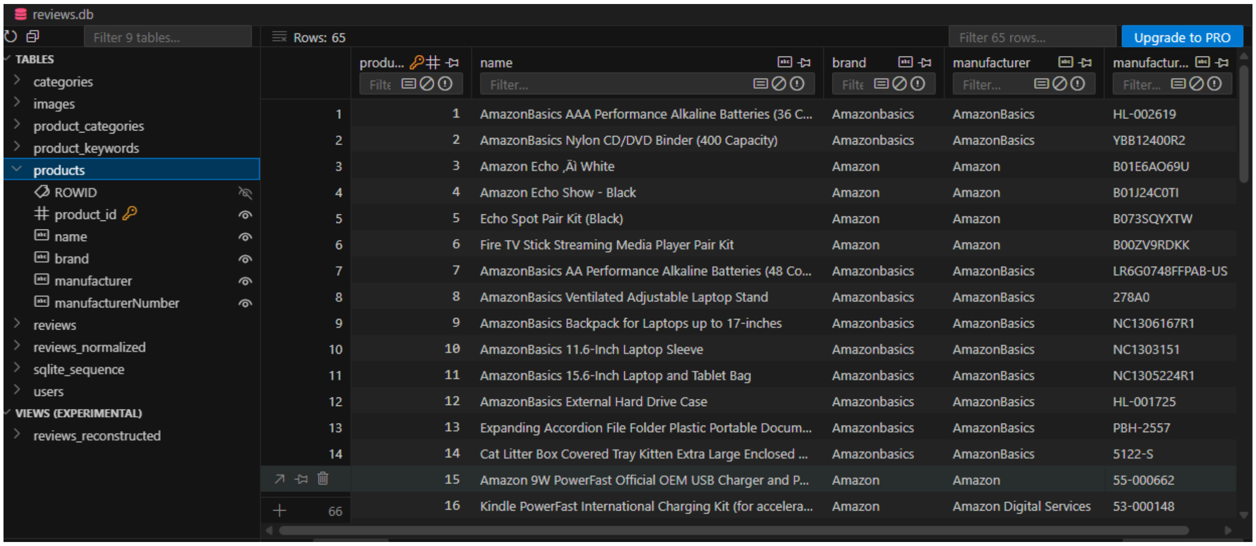
Tables Created:

- products – name, brand, manufacturer
- users – unique reviewers
- reviews_normalized – review-level data with foreign keys
- categories – multi-category storage
- images – product image URLs

The pipeline includes:

- eliminating multivalued attributes (e.g., categories → junction table)
- separating product vs. review attributes
- eliminating transitive dependencies
- recreating a denormalized view (reviews_reconstructed) for backward compatibility

This architecture reduces redundancy and greatly improves query performance.



	produ...	name	brand	manufacturer	manufactur...
1	1	AmazonBasics AAA Performance Alkaline Batteries (36 C...	Amazonbasics	AmazonBasics	HL-002619
2	2	AmazonBasics Nylon CD/DVD Binder (400 Capacity)	Amazonbasics	AmazonBasics	YBB12400R2
3	3	Amazon Echo ‚Äì White	Amazon	Amazon	B01E6AO69U
4	4	Amazon Echo Show - Black	Amazon	Amazon	B01J24C0TI
5	5	Echo Spot Pair Kit (Black)	Amazon	Amazon	B073SQYXTW
6	6	Fire TV Stick Streaming Media Player Pair Kit	Amazon	Amazon	B00ZV9RDKK
7	7	AmazonBasics AA Performance Alkaline Batteries (48 Co...	Amazonbasics	AmazonBasics	LR6G0748FFPAB-US
8	8	AmazonBasics Ventilated Adjustable Laptop Stand	Amazonbasics	AmazonBasics	278A0
9	9	AmazonBasics Backpack for Laptops up to 17-inches	Amazonbasics	AmazonBasics	NC1306167R1
10	10	AmazonBasics 11.6-Inch Laptop Sleeve	Amazonbasics	AmazonBasics	NC1303151
11	11	AmazonBasics 15.6-Inch Laptop and Tablet Bag	Amazonbasics	AmazonBasics	NC1305224R1
12	12	AmazonBasics External Hard Drive Case	Amazonbasics	AmazonBasics	HL-001725
13	13	Expanding Accordion File Folder Plastic Portable Docum...	Amazonbasics	AmazonBasics	PBH-2557
14	14	Cat Litter Box Covered Tray Kitten Extra Large Enclosed ...	Amazonbasics	AmazonBasics	5122-S
	15	Amazon 9W PowerFast Official OEM USB Charger and P...	Amazon	Amazon	55-000662
	16	Kindle PowerFast International Charging Kit (for accelera...	Amazon	Amazon Digital Services	53-000148

NLP Pipeline & Machine Learning

Implemented in process_reviews.py, the ML workflow includes:

Stage 1 – Sentiment Analysis (VADER)

Using NLTK’s SentimentIntensityAnalyzer:

- computes compound scores (–1 to +1)
- assigns sentiment labels (positive/neutral/negative)
- handles short and noisy text robustly

Stage 2 – Phrase Extraction (spaCy + NLTK fallback)

- Extracts meaningful noun chunks and verb phrases:
- filters stopwords
- removes overly generic terms
- preserves 1–4 word informative phrases

Stage 3 – Keyword Scoring Algorithm

- A custom scoring model combining:
- sentiment strength (65%)
- frequency normalization (20%)
- length-based bonus (15%)

The algorithm produces top positive and negative keywords that reflect recurring themes.

Stage 4 – Abstractive Summarization (Transformers)

Using Hugging Face models (preferred: t5-small, fallback: BART, Pegasus):

- summarize aggregated reviews per product
- generate concise 80–200 character insights
- fallback to a theme-based extractive summary if transformer inference fails

This ensures robustness across environments.

Key Findings and Results

1. Sentiment Analysis Observations

- VADER performed reliably for short, opinionated review text.
- Compound score thresholds (±0.05) actually worked well as classification boundaries.
- Sentiment aligned strongly with customer rating values which ends up validating correctness.

2. Keyword Extraction Results

- spaCy extracted precise product-feature phrases (e.g., battery life, charging speed, screen brightness).
- The scoring system separated meaningful aspects from noise.

3. Summarization Results

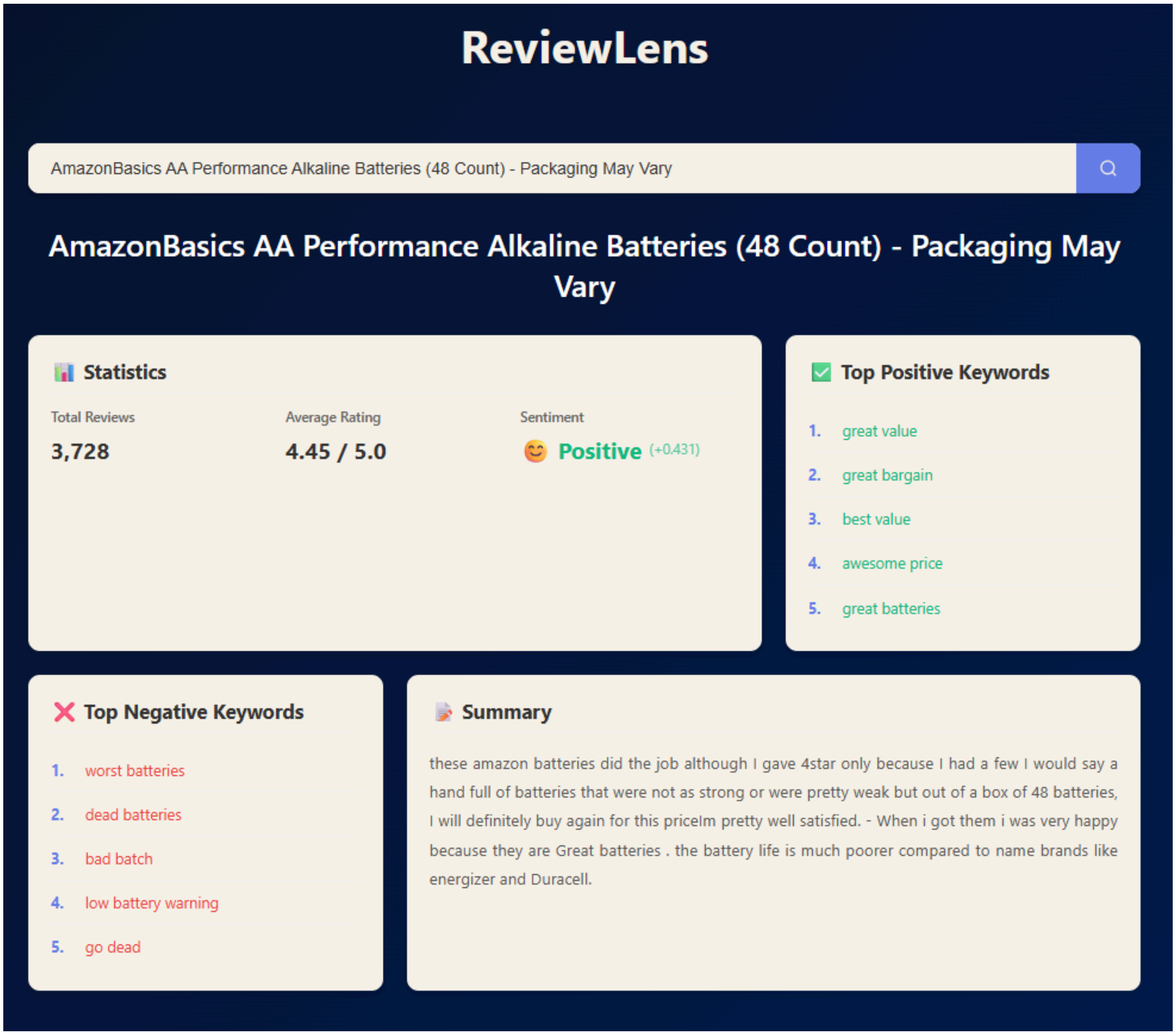
- T5-small provided coherent and compact summaries for most products.
- Fallback theme-based summaries remained interpretable and accurate.
- Summaries consistently highlighted both strengths and weaknesses.

4. Database Normalization Benefits

- Query performance improved significantly after normalization.
- Referential integrity eliminated duplicated product or user data.
- The reconstructed view proved useful for debugging and compatibility.

Hypothesis Outcome

Our hypothesis, that automated review summarization could reliably emulate human reading, was supported. Results demonstrated coherence, interpretability, and accuracy for both sentiment trends and product-level summaries.



AmazonBasics AA Performance Alkaline Batteries (48 Count) - Packaging May Vary

AmazonBasics AA Performance Alkaline Batteries (48 Count) - Packaging May Vary

📊 Statistics

Total Reviews	Average Rating	Sentiment
3,728	4.45 / 5.0	😊 Positive (+0.431)

✅ Top Positive Keywords

1. great value
2. great bargain
3. best value
4. awesome price
5. great batteries

❌ Top Negative Keywords

1. worst batteries
2. dead batteries
3. bad batch
4. low battery warning
5. go dead

📝 Summary

these amazon batteries did the job although I gave 4star only because I had a few I would say a hand full of batteries that were not as strong or were pretty weak but out of a box of 48 batteries, I will definitely buy again for this priceIm pretty well satisfied. - When i got them i was very happy because they are Great batteries . the battery life is much poorer compared to name brands like energizer and Duracell.

Advantages and Limitations

Advantages

- Full end-to-end system integrating Data Engineering + SQL + NLP + ML
- Transparent and interpretable methods
- Normalized SQL schema reduces redundancy and speeds up queries
- Robust fallback mechanisms for NLP ensures reliability
- Modern, responsive frontend enables accessible insights

Limitations

- Transformer summarization is computationally expensive and occasionally inaccurate
- VADER may struggle with sarcasm or domain-specific language
- Dataset limited to Amazon products
- Some products with few reviews produce less useful summaries

- No real-time data ingestion (static dataset only). These limitations can be fixed by paying for a proprietary API key that holds the customer reviews for the ecommerce platform

Changes After Proposal

1. Shift from MongoDB to a Fully SQL-Based System

The original plan included hybrid SQL + MongoDB storage. However, SQLite alone proved sufficient after normalization. This is because SQLite allowed us to process all review text offline instead of using a MongoDB cloud server, query in a more efficient way while meeting all of our needs, and reduce the system complexity by simplifying the architecture.

2. Addition of a React Frontend Instead of a Dashboard

We had originally planned a data analytics dashboard. However, a full React web app provided a more polished, product-like experience and more flexibility in our design.

Conclusion and Future Additions

ReviewLens demonstrates the power of combining rigorous data management practices with modern machine learning techniques. The final system:

- loads and validates large raw datasets,
- normalizes data into a clean relational schema,
- processes text with multi-stage NLP pipelines,
- generates trustworthy product insights,
- and presents results in a polished web interface.

By bridging theory and application, ReviewLens provides a scalable, interpretable, and practical solution to one of the most common pain points in online shopping. It stands as a comprehensive demonstration of full-stack data engineering, natural language processing, and applied machine learning.

ReviewLens still has a few pain points that open pathways for future development. Some products lack a sufficient number of reviews, which leads to weaker sentiment labels, unreliable keyword extraction, and summaries that do not fully capture real user experiences. Additionally, aspect-based sentiment analysis is occasionally tricked, resulting in positive keywords being labeled as negative and negative keywords being labeled as positive.

Additionally, the system currently focuses only on Amazon products; extending ReviewLens to multi-platform datasets (Best Buy, Walmart, Reddit product threads, TikTok Shop reviews, etc.) would make insights more robust and generalizable.

In the future, the project could incorporate more advanced aspect-based sentiment analysis (ABSA) to provide fine-grained scores for individual product features such as battery life, durability, or price. Integrating real-time data pipelines would also allow ReviewLens to update insights as new reviews appear. With these enhancements, ReviewLens could evolve into a more scalable, cross-platform review intelligence tool for both consumers and businesses.

Contributions

The group collectively met up in person and online to find data to use for our project. After locating a good database, Rishabh primarily led the data processing pipeline and SQL database development, ensuring that our datasets were efficiently cleaned, structured, and accessible. Kushagra focused on the NLP and machine learning components, building the sentiment analysis, summarization, and keyword extraction models that powered the core functionality of our system. Joshua led the development of the webpage and visualization interface, integrating the backend outputs into a clear and intuitive user experience. Although each member took ownership of specific components, we collaborated closely throughout the project. We reviewed each other's work, debugging issues together, and made sure that every subsystem aligned with and supported our final goal.