

Amazon Product Review Analysis Service

- *Datacenter Scale Computing - Fall 2024*

Group 14
Trapti Damodar Balgi
Nidhi Choudhary

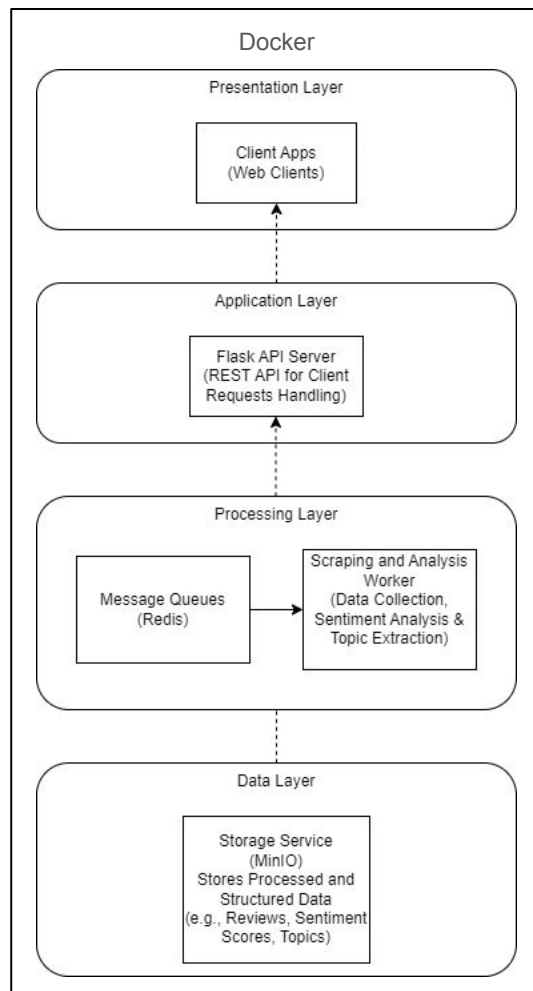
Project Goals

- Develop a service to collect and analyze Amazon product reviews.
- Extract product names and summarize customer feedback.
- Perform sentiment analysis to assess customer satisfaction (% positive, negative, neutral).
- Identify and store key topics from reviews in a data storage system to enhance decision-making for consumers and businesses.

Software Components

1. **REST API:** Flask-based API for user interactions (submit URLs, retrieve insights).
2. **Message Queues:** Redis to distribute tasks across worker nodes.
3. **Storage:** MinIO for storing processed review data.
4. **Text Analysis:** Pre-trained models for sentiment analysis, topic modeling, and summarization. *[Pre-trained models like "facebook/bart-large-cnn", "joeddav/distilbert-base-uncased-go-emotions-student", and Latent Dirichlet Allocation (LDA) models for summarization, sentiment analysis, and topic modeling, respectively.]*
5. **Docker:** Scalable, containerized application.

Architectural Diagram



Interaction between different Software Components

Presentation Layer:

- Users interact with the service through a web client interface, submitting Amazon product URLs and viewing analysis results, which are processed and delivered via the Flask REST API.

Application Layer:

- Flask API processes user requests and forwards tasks to Redis queues.

Processing Layer:

- Redis queues tasks for the worker.
- Workers handle scraping, sentiment analysis, topic extraction, and summarization using pre-trained models.

Data Layer:

- MinIO stores processed data (reviews, sentiment scores, topics).

Deployment:

- Docker containerizes the application, ensuring consistent environments and simplifying the deployment process.

Debugging and Testing

Developed and tested each component separately to ensure functionality:

- Scraping: Validated correct data extraction from Amazon product pages.
- UI and REST API: Tested API endpoints and UI for seamless interaction.
- Summarization: Verified proper summarization of reviews using the pre-trained model.
- Product Name Extraction: Checked accuracy in extracting product names.
- Topic Extraction and Sentiment Analysis: Tested accuracy of topic modeling and sentiment classification.
- Callback Feature: Ensured correct callback responses.

After individual tests, we integrated the components, checked the logs for errors, and verified the output through MinIO UI for proper data storage.

Debugging and Testing Snapshots

The screenshot shows the Docker Desktop interface. The 'Containers' tab is active, displaying a list of containers. The container 'k8s_proj-worker-proj-wrkr-8a793b97a4b6' is selected. The terminal window shows the following output:

```
2024-12-07 09:23:36,097 - INFO - Scraping reviews for URL: https://a.co/d/anzhWpH
2024-12-07 09:23:37,046 - INFO - CSV file created: /tmp/dc958a5-a3a-4a29-baac-f32f8a19f0fe_reviews.csv
(base) nldh@choudhary@nldh-ls-kadbook-Pro nldh:~$ kubectl logs proj-worker-07b1040f-hdzc
2024-12-07 09:23:37,096 - INFO - Worker is listening for tasks in the Redis queue...
2024-12-07 09:23:38,093 - INFO - Received task for review with hash: dc958a5-a3a-4a29-baac-f32f8a19f0fe
2024-12-07 09:23:38,093 - INFO - Starting task for reviewhash: dc958a5-a3a-4a29-baac-f32f8a19f0fe
2024-12-07 09:23:38,097 - INFO - Scraping reviews for URL: https://a.co/d/anzhWpH
2024-12-07 09:23:37,046 - INFO - CSV file created: /tmp/dc958a5-a3a-4a29-baac-f32f8a19f0fe_reviews.csv
2024-12-07 09:25:17,492 - INFO - Summary file created: /tmp/dc958a5-a3a-4a29-baac-f32f8a19f0fe_summary_output.csv
2024-12-07 09:25:17,589 - INFO - Uploaded summary to MinIO: dc958a5-a3a-4a29-baac-f32f8a19f0fe/summary_output.csv
2024-12-07 09:25:17,589 - INFO - Classifying emotions for reviews in task: dc958a5-a3a-4a29-baac-f32f8a19f0fe
(base) nldh@choudhary@nldh-ls-kadbook-Pro nldh:~$
```

The screenshot shows the Docker Desktop interface with the 'Containers' tab active. The container 'k8s_proj-worker-proj-worker-55c7cd4b48-48n29_default_62747253-4373-4047-a37b-36ac202cf59f_0' is selected. The 'Debug mode' tab is active, showing the container's logs. The terminal window shows the following output:

```
2024-12-07 10:37:51,315 - INFO - Performing topic modeling for task: 76c96dcf-e2ca-48dd-afba-02f60bb87842
2024-12-07 10:37:51,364 - INFO - Topic modeling results saved to MinIO: 76c96dcf-e2ca-48dd-afba-02f60bb87842/topics.csv
2024-12-07 10:37:51,365 - INFO - Extracted product name: Apple 2024 MacBook Air 13-inch Laptop with M3 chip. Built for Apple Intelligence
ce, 13.6-inch Liquid Retina Display, 16GB Unified Memory, 256GB SSD Storage, Backlit Keyboard, Touch ID; Midnight
2024-12-07 10:37:51,365 - INFO - Set callback for 76c96dcf-e2ca-48dd-afba-02f60bb87842: Apple 2024 MacBook Air 13-inch Laptop with M3 c
hip. Built for Apple Intelligence, 13.6-inch Liquid Retina Display, 16GB Unified Memory, 256GB SSD Storage, Backlit Keyboard, Touch ID;
Midnight
2024-12-07 10:37:51,366 - INFO - Set callback for 76c96dcf-e2ca-48dd-afba-02f60bb87842: Apple 2024 MacBook Air 13-inch Laptop with M3 c
hip. Built for Apple Intelligence, 13.6-inch Liquid Retina Display, 16GB Unified Memory, 256GB SSD Storage, Backlit Keyboard, Touch ID;
Midnight, summary, and sentiments
2024-12-07 10:37:51,366 - INFO - Task completed for reviewhash: 76c96dcf-e2ca-48dd-afba-02f60bb87842
(base) nldh@choudhary@nldh-ls-kadbook-Pro nldh:~$
```

Working System - UI



Amazon Link Processor

Enter Amazon Link:

Working System - Results

Amazon Link Processor

Enter Amazon Link:

Submit

Status: completed

Callback Details

Product Name: Caseative for iPhone 14 Pro Case, Solid Color Curly Wave Frame Soft Compatible with iPhone Case (Green Blue,iPhone 14 Pro)

Summary: Love it but with this particular color it gets dirty quick! Would definitely buy again just in another color. Got so many compliments so that was a plus! Love how it protected my phone, wasn't bulky, & was ascetically pleasing! Also, you can't beat the price either!

Sentiments:

- Positive: 43.67010703897662%
- Neutral: 16.63010647495578%
- Negative: 39.6997864860676%

Working System - Storage (Processed Data)

Object Browser

Start typing to filter objects in the bucket

reviews

Created on: Thu, Dec 05 2024 23:30:27 (MST) Access: PRIVATE 174.9 KiB - 10 Objects

Rewind

Refresh
















Upload

< reviews

Create new path




	Name	Last Modified	Size
<input type="checkbox"/>	04dc93af-d957-4529-839f-024e31397e0f		-
<input type="checkbox"/>	34811a88-1ea1-435a-a343-a84764d3bc55		-
<input type="checkbox"/>	9a8c6480-ef2c-4b57-8906-bf4b6b16c75d		-
<input type="checkbox"/>	a646d8c6-a7b5-488f-8a6b-f5de276c65e7		-
<input type="checkbox"/>	b31ed16b-36ce-43b2-b8d8-df3a1a11662c		-
<input type="checkbox"/>	b4a40b2a-7fae-42a9-8f5a-8ef5f91aa6a2		-
<input type="checkbox"/>	c2983716-ce87-4757-b0f0-ffe8e520ec43		-
<input type="checkbox"/>	cc20d737-90f2-4aaf-a74c-c0c1bb65b0dd		-
<input type="checkbox"/>	df33ab69-ed68-47e4-b7ce-e74d496d4f28		-
<input type="checkbox"/>	e09002f3-4e83-45ae-9a79-e570650ec4ff		-


Working System - Storage (Processed Data)



← Object Browser


🔍 Start typing to filter objects in the bucket




**reviews**

Created on: **Thu, Dec 05 2024 23:30:27 (MST)** Access: **PRIVATE** 174.9 KiB - 10 Objects

Rewind ↶ Refresh ↺ Upload 📤

< reviews / 04dc93af-d957-4529-839f-024e31397e0f 

Create new path 📁

<input type="checkbox"/>	▲ Name	Last Modified	Size
<input type="checkbox"/>	 reviews.csv	Thu, Dec 05 2024 23:30 (MST)	17.9 KiB

Workload Handling and Bottlenecks

Workload Handling:

The system can currently handle light workloads, processing a few hundred simple operations per second. This can be scaled as necessary.

Bottlenecks:

1. **Rate Limiting from Amazon:** Restrictions on the frequency and volume of requests to prevent overloading their servers. Currently, we extract reviews only from the first 5 pages to stay within these limits
Future Work: Implementing request throttling and rotating proxy servers to bypass rate limits.
2. **Fake Reviews:** Finding and removing unreliable or fake reviews.
Future Work: Using advanced machine learning models to improve the detection and removal of fake reviews.
3. **Models for Analysis:** Ensuring accurate and efficient text processing.
OpenAI API: Limited by usage caps, requiring a paid version for continued analysis.
Future Work: Improve performance with fine-tuning and faster models for real-time processing.

DEMO

References

- <https://www.geeksforgeeks.org/web-scraping-amazon-customer-reviews/>
- <https://arxiv.org/abs/1810.04805>
- <https://arxiv.org/abs/1910.13461>
- <https://huggingface.co/docs/transformers/index>
- <https://towardsdatascience.com/end-to-end-topic-modeling-in-python-latent-dirichlet-allocation-lda-35ce4ed6b3e0>
- <https://towardsdatascience.com/latent-dirichlet-allocation-lda-9d1cd064ffa2>
- <https://realpython.com/python-redis/>
- <https://min.io/docs/kes/>
- Class notes, slides, assignments