Product Sentiment Analyzer and Review Dashboard

Author: Nithishwar T
Date: September 5, 2025
GitHub Repository: https://github.com/nithishwar17/Project-Sentiment

1.Abstract:

This project focuses on developing a full-stack web application for analyzing product reviews from e-commerce platforms using Natural Language Processing (NLP) techniques. The system automatically scrapes product reviews, applies sentiment analysis models (specifically TextBlob, with the architecture supporting others like VADER), and stores the structured results in a cloud-native database, MongoDB Atlas. A Flask-based backend API manages the data scraping and analysis pipeline, while a React.js frontend presents the insights through a clean, interactive dashboard. This solution provides a powerful tool for businesses and customers to understand market sentiment effectively and make data-driven decisions.

2. Objectives:

- To develop an end-to-end pipeline for real-time product review sentiment analysis.
- To integrate dynamic web scraping (using Selenium for Amazon) for live data collection.
- To apply NLP sentiment analysis techniques to classify customer reviews as positive, negative, or neutral.
- To design and build an interactive dashboard using React.js for visualizing product sentiment data.
- To store and retrieve scraped reviews and analysis results using a scalable cloud database (MongoDB Atlas).

3. Software Requirements:

- **Python:** Version 3.12+
- **Backend Framework:** Flask
- **Database:** MongoDB Atlas (Cloud-hosted)
- **Frontend Framework:** React.js
- **Runtime Environment:** Node.js & npm
- **Python Libraries:** Selenium, BeautifulSoup, TextBlob, PyMongo, python-dotenv
- **Web Browser:** Google Chrome

4. System Architecture:

The application follows a client-server architecture where the frontend (client) communicates with the backend (server) via a REST API.

Workflow:

1. The user enters a product URL into the React.js web interface.
2. The frontend sends the URL to the Flask backend via an API request.
3. The Flask backend triggers the Selenium-based scraper to collect reviews from the specified URL.
4. The backend then analyzes the sentiment of each review using the TextBlob library.
5. The analysis results (reviews, sentiment scores, and summary) are stored as a document in the MongoDB Atlas database.
6. The backend sends the results back to the React.js frontend.
7. The frontend dynamically updates the dashboard to display pie charts and detailed review statistics.

---

5. Implementation:

5.1 Backend (Flask API)

A RESTful API was created with a primary endpoint:

- `POST /api/analyze`: Receives a product URL, orchestrates the scraping and analysis process, saves the data to MongoDB, and returns the final JSON response.

5.2 Sentiment Analysis

- **TextBlob:** Utilized for its straightforward polarity scoring system. Each review's text is processed to generate a polarity score between -1.0 (negative) and +1.0 (positive), which is then used for classification.

5.3 Database (MongoDB Atlas)

A NoSQL database was used to store unstructured data effectively.

- **`analysis_results` collection:** Each document in this collection stores the product URL, a summary of sentiment counts, a detailed list of individual reviews with their sentiment labels, and a timestamp of the analysis.

5.4 Frontend (React.js Dashboard)

- **Search Bar:** A controlled component captures the user-submitted URL.
- **Review List:** A scrollable list displays individual reviews with styled sentiment badges (positive, negative, neutral).

- **Pie Chart:** A react-chartjs-2 component provides an immediate visual summary of the sentiment distribution.
- **State Management:** The main App.js component manages the application's state, including loading indicators and error messages, providing a smooth user experience during the long-running scraping process.

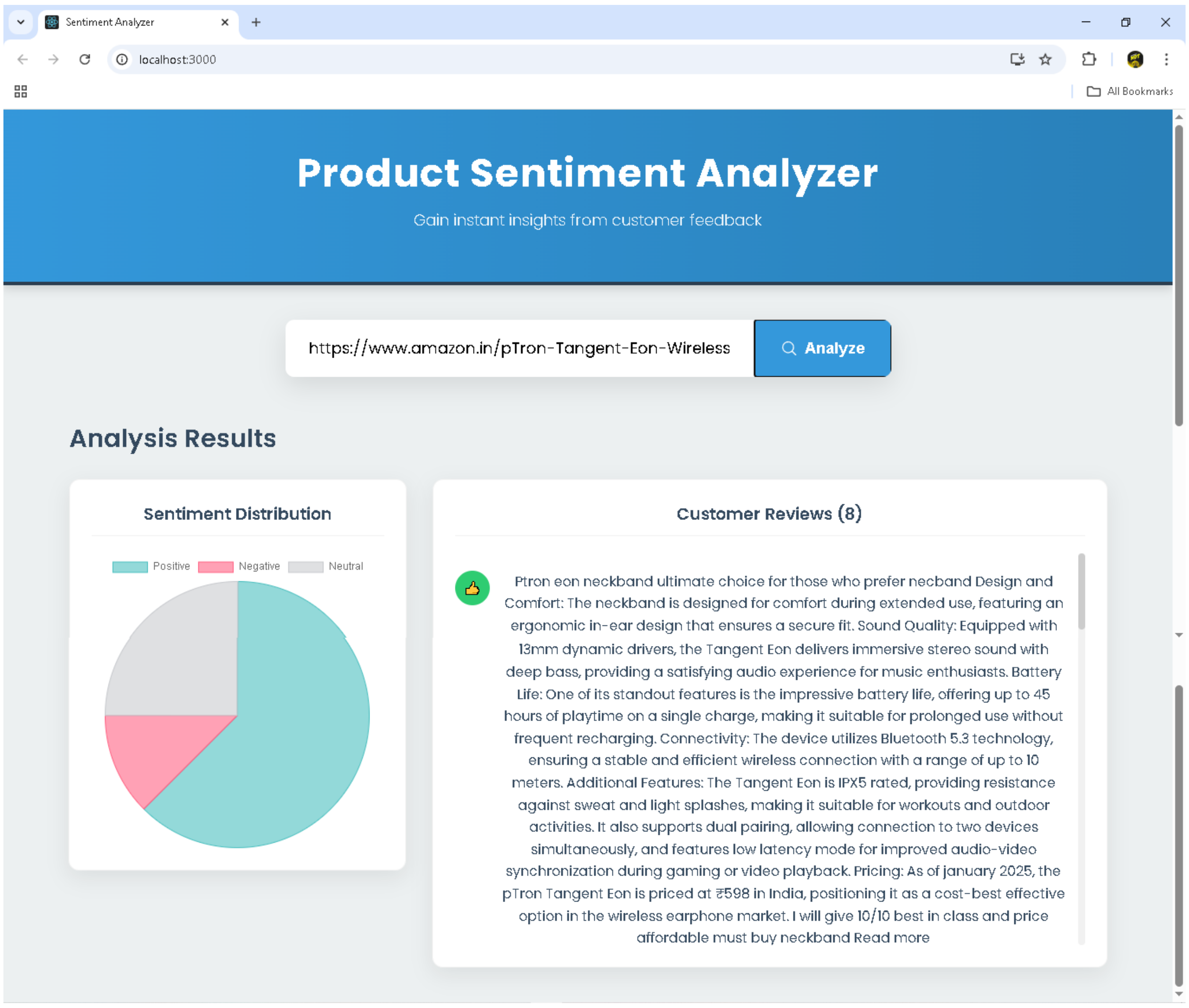
5.5 Web Scraping

- **Amazon (Selenium):** Selenium was used to control a Chrome browser, which was essential for handling dynamic, JavaScript-heavy websites and managing the interactive login process required to access reviews. WebDriverWait was implemented for robustly waiting for elements to load.

---

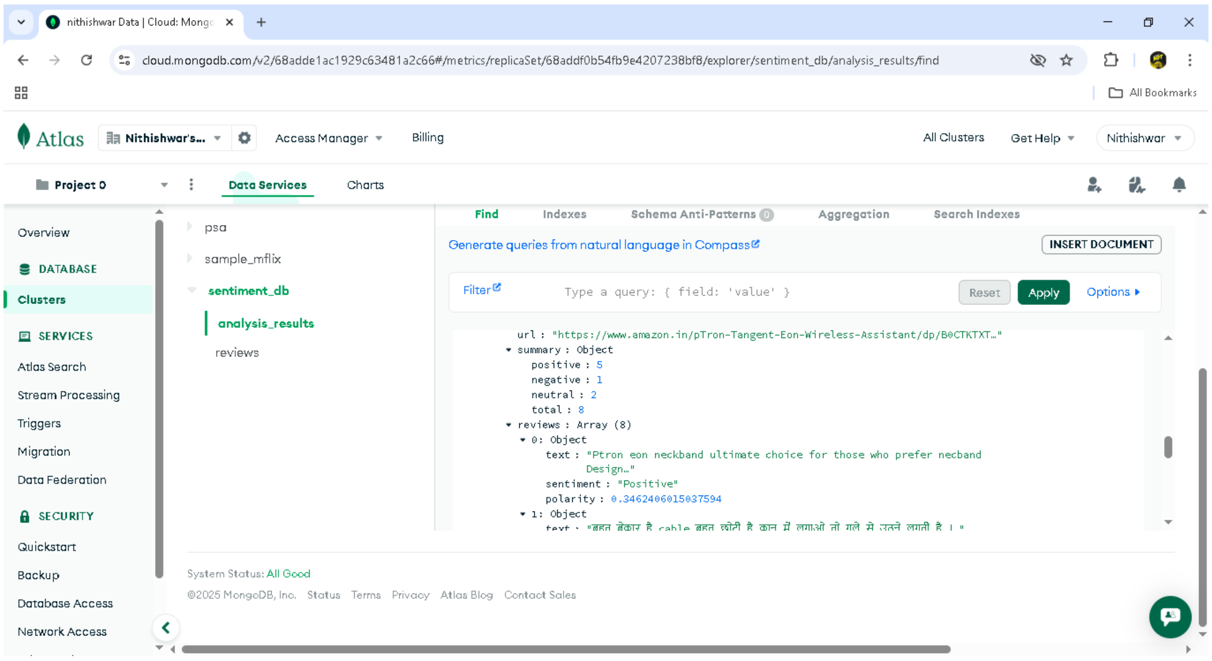
6. Results & Screenshots:

- **Screenshot of the React.js dashboard showing the pie chart and review list:**



Ptron eon neckband ultimate choice for those who prefer necband Design and Comfort: The neckband is designed for comfort during extended use, featuring an ergonomic in-ear design that ensures a secure fit. Sound Quality: Equipped with 13mm dynamic drivers, the Tangent Eon delivers immersive stereo sound with deep bass, providing a satisfying audio experience for music enthusiasts. Battery Life: One of its standout features is the impressive battery life, offering up to 45 hours of playtime on a single charge, making it suitable for prolonged use without frequent recharging. Connectivity: The device utilizes Bluetooth 5.3 technology, ensuring a stable and efficient wireless connection with a range of up to 10 meters. Additional Features: The Tangent Eon is IPX5 rated, providing resistance against sweat and light splashes, making it suitable for workouts and outdoor activities. It also supports dual pairing, allowing connection to two devices simultaneously, and features low latency mode for improved audio-video synchronization during gaming or video playback. Pricing: As of january 2025, the pTron Tangent Eon is priced at ₹598 in India, positioning it as a cost-best effective option in the wireless earphone market. I will give 10/10 best in class and price affordable must buy neckband Read more

- **Screenshot of MongoDB Atlas showing the stored review data:**



```
      url : "https://www.amazon.in/pTron-Tangent-Eon-Wireless-Assistant/dp/B0CTKTXT…"
    ▾ summary : Object
        positive : 5
        negative : 1
        neutral : 2
        total : 8
    ▾ reviews : Array (8)
      ▾ 0: Object
          text : "Ptron eon neckband ultimate choice for those who prefer necband
                 Design…"
          sentiment : "Positive"
          polarity : 0.3462406015037594
      ▾ 1: Object
          text : "बहुत बेकार है cable बहुत छोटी है कान में लगाओ तो गले से उतरने लगती है । "
```

7. Advantages:

- **Real-time Analysis:** Provides up-to-date sentiment insights directly from live e-commerce pages.
- **Automated Data Collection:** Eliminates the need for manual review reading and data entry.
- **Cloud Storage:** MongoDB Atlas offers a scalable, reliable, and accessible database solution.
- **Interactive Dashboard:** The visual interface allows for quick and effective decision-making.

8. Limitations:

- **Scraping Fragility:** Changes in the e-commerce website's HTML structure can break the scraper, requiring code updates.
- **Rate Limiting:** E-commerce sites may temporarily block the server's IP if too many requests are made in a short period.
- **Model Accuracy:** The accuracy of the sentiment analysis is dependent on the pre-trained TextBlob model, which may not understand all nuances or context.
- **Internet Dependency:** The application requires a constant internet connection for both scraping and database access.

9. Future Enhancements:

- **Custom Model Training:** Train a custom deep learning model (e.g., using BERT or RoBERTa) on a product review dataset for significantly improved sentiment accuracy.
- **Multi-Language Support:** Add support for analyzing reviews in other languages.
- **Cloud Deployment:** Deploy the Flask backend and React frontend on cloud platforms (e.g., Render, Vercel) for global public access.
- **Product Comparison:** Enable users to analyze and compare the sentiment of multiple products side-by-side.

---

10. Conclusion:

This project demonstrates the use of **NLP, Flask, MongoDB, and React.js** to build a full-stack **Product Sentiment Analyzer**. It provides insights into customer opinions and can be extended for business intelligence, market research, and e-commerce analytics.

---

11. References

- Flask Documentation
- MongoDB Atlas
- Selenium Docs
- TextBlob
- VADER Sentiment
