DES646 - AI/ML for Designers

Crowdsourced product feedback analyzer using NLP

Documentation

Team Members:
Akshat Kasarwal (240085)
Vanashree Hirpurkar (241131)

Instructor:
Amar Kumar Behera

Abstract

The Customer Feedback Analyzer is an AI-augmented system developed to transform unstructured customer reviews into actionable design insights. Built on a hybrid architecture combining ***FastAPI*** *(backend) and* ***Next.js*** *(frontend), the system integrates multiple natural language processing (NLP) modules for* ***sentiment analysis, emotion detection****, and* ***intent recognition****, along with a* ***Retrieval-Augmented Generation (RAG)*** *model for contextual summaries and question answering. Using the Women's E-Commerce Clothing Reviews dataset as a base, the system precomputes review analytics, predicts customer satisfaction levels through* ***Net Promoter Score (NPS)****, and presents department-level insights via an interactive dashboard. Additionally, a* ***Chrome extension*** *enables live review analysis directly on e-commerce pages. The project demonstrates how design teams can use AI-driven feedback intelligence to identify user pain points, optimize product features, and enhance customer experience.*

Contents

1 Introduction

In today's digital marketplace, customers freely express their opinions about products through online reviews and ratings. These reviews form a valuable yet **unstructured and noisy data source** that reflects real user experiences, preferences, and frustrations. For designers and product teams, interpreting this large volume of feedback is essential for improving product quality and user satisfaction yet **manual analysis is time-consuming, inconsistent, and often lacks actionable insights.**
The Feedback Analyzer project was conceived to bridge this gap between **raw customer feedback** and **design intelligence**. It leverages artificial intelligence and natural language processing (NLP) to **automatically extract emotions, sentiments, and intents** from textual reviews. By integrating a **Retrieval-Augmented Generation (RAG)** module, the system can also generate contextual summaries and explanations, enabling designers to ask natural-language questions such as *"What are the main complaints about sizing in the dresses category?"* and receive precise, data-driven answers.

Objectives:
The Feedback Analyzer project aims to overcome these limitations by building an AI-powered system that can:

1. **Transform unstructured text feedback** into structured, interpretable design insights.
2. **Perform multi-level sentiment and emotion analysis** across departments.
3. **Compute Net Promoter Scores (NPS)** and predict user satisfaction trends from textual reviews.
4. **Enable natural-language querying and summarization** through a RAG (Retrieval-Augmented Generation) interface.
5. **Provide designers with actionable visual analytics and AI-driven explanations,** enabling faster and more informed design decisions.

2 About Dataset

The Women's E-Commerce Clothing Reviews dataset, sourced from Kaggle, is a comprehensive collection of customer feedback on women's apparel sold through an online retail platform. It contains **23,486 records** and **10 primary attributes**, capturing both textual and structured information about customer experiences, product categories, and satisfaction levels.

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 23486 entries, 0 to 23485
Data columns (total 11 columns):
 #   Column                   Non-Null Count  Dtype
---  ------                   --------------  -----
 0   Unnamed: 0               23486 non-null  int64
 1   Clothing ID              23486 non-null  int64
 2   Age                      23486 non-null  int64
 3   Title                    19676 non-null  object
 4   Review Text              22641 non-null  object
 5   Rating                   23486 non-null  int64
 6   Recommended IND          23486 non-null  int64
 7   Positive Feedback Count  23486 non-null  int64
 8   Division Name            23472 non-null  object
 9   Department Name          23472 non-null  object
 10  Class Name               23472 non-null  object
dtypes: int64(6), object(5)
memory usage: 2.0+ MB
```

For this project, an extensive **Exploratory Data Analysis (EDA)** was conducted on the dataset to understand its structure, distribution, and key behavioral patterns across departments and classes. The analysis revealed that most reviews were concentrated in categories like Tops, Dresses, and Knits, with the General and Dresses departments contributing the highest number of reviews. Rating distributions indicated a positive bias, with the majority of users providing ratings of 4 or 5, aligning with overall favorable sentiment trends. Following EDA, the dataset was cleaned and filtered to retain only essential variables such as Review Text, Rating, Department Name, and Class Name, forming the input for downstream NLP models. The text data then underwent preprocessing including tokenization, lowercasing, and punctuation removal before being passed through modules for sentiment analysis, emotion clustering, and intent detection.

3 TECHNICAL IMPLEMENTATION

The **Feedback Analyzer System** integrates multiple AI-driven components into a cohesive full-stack application. The implementation is organized around the **FastAPI backend**, **Next.js frontend**, and a modular **NLP pipeline** coordinated by an **Orchestrator**. This section describes the technical functioning of each layer, key endpoints, and workflow logic.

3.1 Data Flow Summary

1. Raw review data is preprocessed and stored locally.
2. During startup, the backend loads cached analytics (`dashboard_reviews.jsonl`, `dashboard_summary.json`) for fast retrieval.
3. The frontend and Chrome extension communicate with the FastAPI backend for analysis and visualization.
4. The **RAG (Retrieval-Augmented Generation)** module retrieves relevant text embeddings via **FAISS** and generates contextual explanations using **Gemini API**.

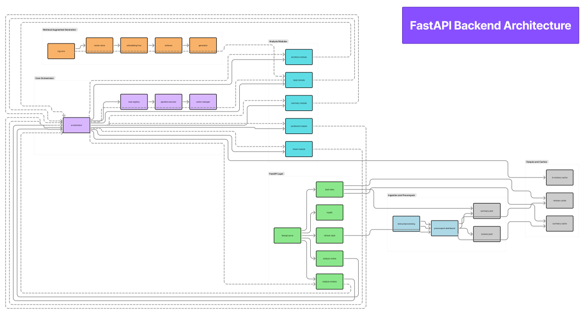
3.2 Backend Implementation

The backend, implemented using **FastAPI**, serves as the main analytical engine for data ingestion, processing, and AI inference. It exposes REST endpoints that connect the frontend dashboard and the Chrome extension to various computational modules.

Key Endpoints

endpoint	method	Functionality
/dashboard_data	GET	Returns precomputed sentiment, emotion, and NPS aggregates for all or selected departments.
/refresh_dashboard	POST	Rebuilds sentiment and emotion caches using precompute_dashboard.py without restarting the server.
/analyze_review	POST	Performs full pipeline analysis for a single review (sentiment, emotion, intent, NPS).
/analyze_reviews	POST	Batch analysis for multiple reviews (used by Chrome extension).
/query	POST	Handles RAG-based question answering using FAISS + Gemini LLM.

Backend Architecture



3.3 Orchestrator Logic

The **Orchestrator** (`orchestrator.py`) acts as the core inference pipeline that integrates results from multiple specialized NLP modules. When a review or batch of reviews is received through the `/analyze_review` or `/analyze_reviews` endpoints, the orchestrator performs the following sequence of operations:

1. **Sentiment Analysis:** Invokes `sentiment.py` (VADER) to compute polarity scores and assign sentiment labels such as *Positive*, *Negative*, or *Neutral*.

2. **Emotion Detection:** Uses `emotions.py` for embedding-based clustering and maps results to emotion categories such as *Joy*, *Anger*, *Sadness*, *Surprise*, and *Fear*.

3. **Intent Recognition:** Applies `intent.py` (clustering-based) to categorize user intent into classes such as *Complaint*, *Suggestion*, or *Praise*.

4. **NPS Calculation:** If a numerical rating is present, the Orchestrator computes the **Net Promoter Score (NPS)** using the formula:

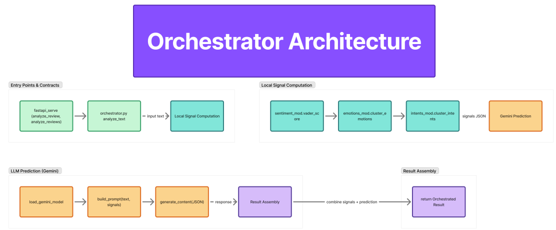
$$\text{NPS} = \left(\frac{\text{Promoters} - \text{Detractors}}{\text{Total}} \right) \times 100$$

   If no explicit rating is available, it predicts the NPS using a trained regression model (`nps_regressor.joblib`).

5. **Response Packaging:** Aggregates all outputs - **sentiment**, **emotion**, **intent**, and **NPS** into a unified JSON response, which is returned to the API layer.

This modular pipeline ensures scalability, allowing additional detectors (e.g., sarcasm detection or multilingual sentiment analysis) to be added easily without altering existing logic.

Orchestrator Architecture



3.4 RAGbot (Retrieval-Augmented Generation)

The **RAG module** (`rag.py`) enhances the system's interpretability by enabling contextual question answering. It operates through the following main steps:

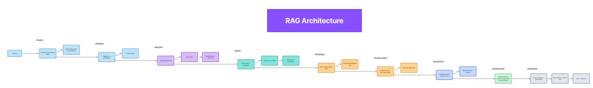
1. **Vector Search (Retrieval):** Uses **FAISS** to search for semantically similar reviews or summaries from the precomputed embeddings stored in `faiss_index/`.

2. **Context Composition:** Selects the top-k relevant text chunks and concatenates them into a structured prompt.

3. **Generation (Augmentation):** Sends the contextualized query to the **Gemini LLM**, which synthesizes a factual, design-oriented response.

This setup allows designers to query insights such as:
What sizing issues do users frequently mention in dresses?
and receive concise, AI-generated summaries grounded in actual customer data.

RAG Architecture



3.5 Frontend Integration

The frontend is implemented using **Next.js** with **TypeScript** and serves as the system's visualization and interaction layer. It communicates with the FastAPI backend through an API proxy (`/api/dashboard` → `/dashboard_data`) to fetch and render analytics.

Key Features

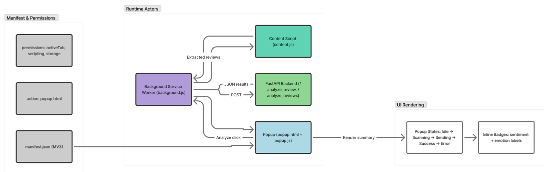
- **Department and Class Filters:** Allows scoped analysis of reviews by department or product class for targeted insights.
- **Interactive Visualizations:** Displays sentiment and emotion trends using **Bar Charts** and **Pie Charts** for intuitive understanding.
- **AI Insights Section:** Connects to the `/query` endpoint, enabling contextual question answering powered by the RAG module and Gemini LLM.
- **Responsive Design:** Optimized for both desktop dashboards and Chrome extension popup views, ensuring seamless user experience across interfaces.

3.6 Chrome Extension Integration

The Chrome extension enables users to analyze live customer reviews directly from e-commerce platforms(currently on local host products page). It provides real-time access to sentiment and emotion analytics without needing to manually upload datasets.

- **Live Review Analysis:** When a user highlights or clicks on reviews, the extension sends them to the backend's `/analyze_reviews` endpoint for processing.
- **Interactive Results Display:** The analyzed results including sentiment breakdown, emotion distribution, and average NPS are shown in an on-page popup overlay.
- **Batch Review Handling:** For multiple reviews, the extension aggregates average ratings and overall sentiment trends to provide an at-a-glance summary.
- **Real-world Applicability:** This integration demonstrates the practical use of the system beyond static datasets, bringing AI-powered insights directly to designers and researchers in situ.

Chrome-Extension Architecture



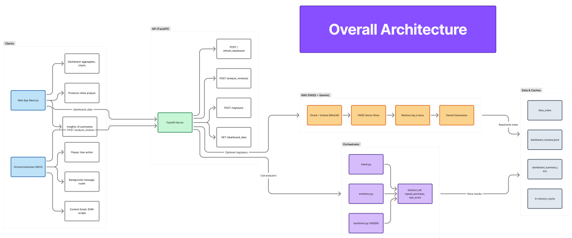
3.7 Performance Optimization Techniques

To ensure scalability and low-latency performance, the system employs several optimization strategies across both backend and frontend layers.

- **Precomputation:** Reduces real-time workload by caching analytics and reusing computed summaries for faster dashboard loading.
- **Lazy Initialization:** Loads heavy models such as Gemini and FAISS only when required, minimizing startup time and memory usage.
- **Sampling:** Utilizes representative subsets of data to render the dashboard quickly without compromising analytical accuracy.
- **Asynchronous API Calls:** Leverages FastAPI's asynchronous capabilities to efficiently handle multiple simultaneous user requests.

- **DataFrame Persistence:** Maintains processed data in memory, reducing repetitive disk I/O and enabling rapid response generation.

3.8 Overall Architecture



4 Results & Visualizations

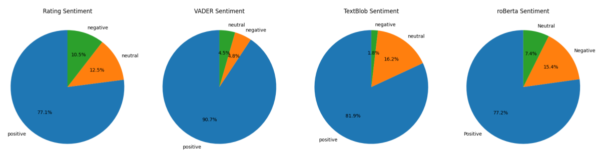
The **Feedback Analyzer** system transforms unstructured customer reviews into measurable and interpretable insights for designers. Through its combination of analytical models and visual storytelling, the system provides both quantitative summaries and qualitative narratives, helping identify user satisfaction patterns, emotional tones, and design improvement opportunities.

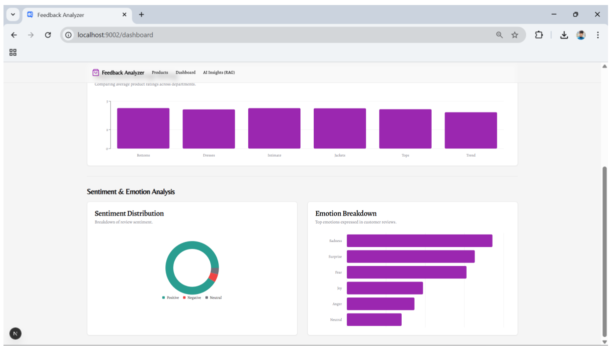
4.1 Sentiment Analysis Visualization

The sentiment distribution provides a macro-level understanding of how customers perceive various product departments. From the exploratory data analysis (EDA), the following observations were made:

- Most reviews exhibit **positive polarity** (ratings 4-5), suggesting high overall product satisfaction.
- VADER sentiment analysis indicates that approximately **90.7%** of reviews are positive, **4.5%** neutral, and **4.8%** negative.
- **Dresses** and **Tops** departments show the highest positive sentiment counts.
- **Bottoms** and **Intimates** display more mixed opinions due to sizing inconsistencies.

Visualizations:

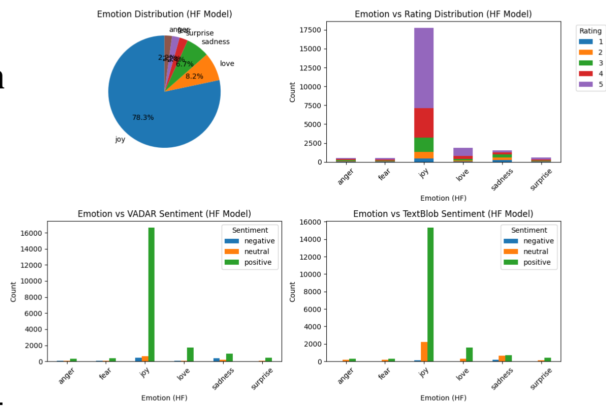




4.2 Emotion Analysis Visualization

The emotion clustering model identifies dominant emotional tones across product reviews.

Visualizations:



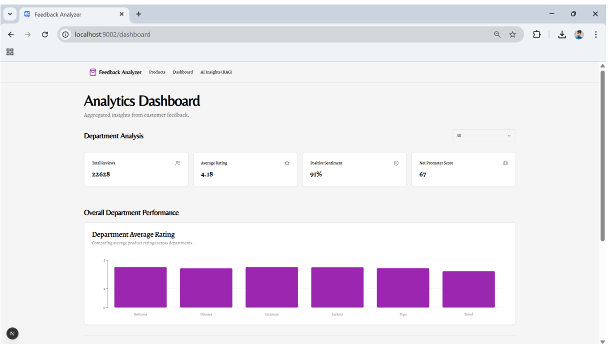
Example Insight:
"Products in the 'Dresses' department evoke more joy and appreciation, while 'Bottoms' show higher anger and frustration emotion scores."

4.3 Net Promoter Score (NPS) Trends

Using review ratings, the system automatically computes the **Net Promoter Score (NPS)** to quantify overall brand loyalty and customer advocacy. Each review is categorized into the following groups:

- **Promoters:** Rating ≥ 4
- **Passives:** Rating = 3
- **Detractors:** Rating ≤ 2

The final computed NPS across all departments was approximately **+67**, indicating strong customer satisfaction but suggesting potential improvements in areas such as sizing consistency.
Visualizations:

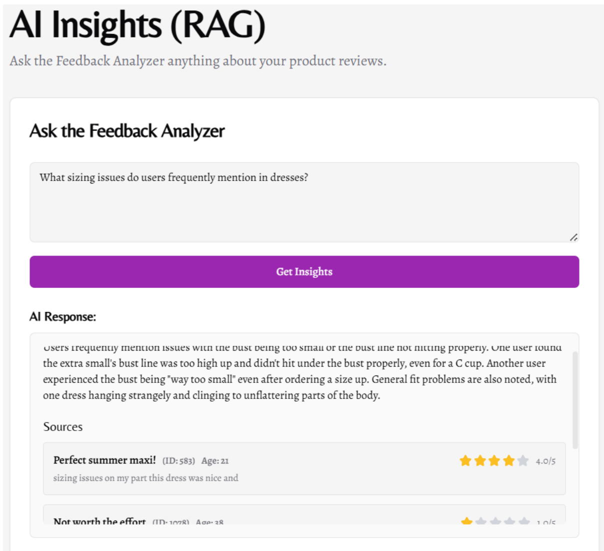


4.4 AI Insights (RAG)

The **AI Insights** section of the dashboard enables users to ask design-relevant questions directly in natural language.
Examples of RAG Queries:

- "What sizing issues do users frequently mention in dresses?"
- "Which departments have the highest complaint rates?"
- "Summarize positive feedback trends for tops."

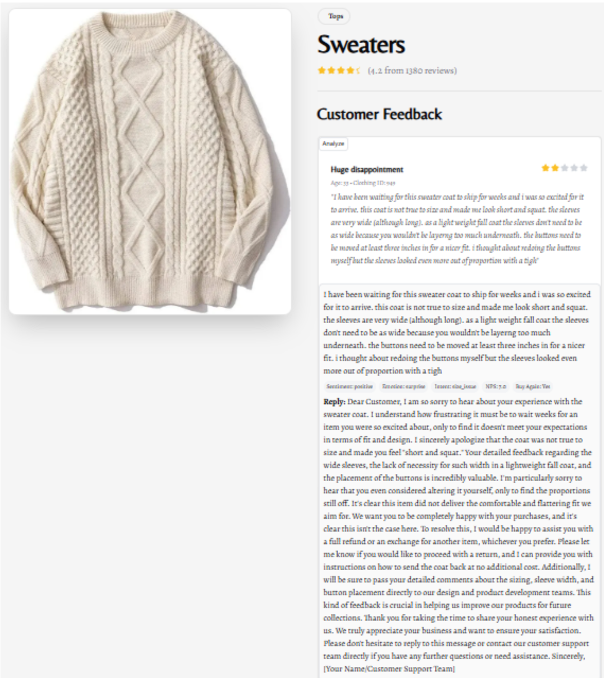
Example Output:
"Users frequently mention issues with the bust being too small or the bust line not hitting properly. One reviewer noted the extra small's bust line was too high, even for a C cup, and hoped sizing up would resolve it. Another user found the bust too small, even after ordering and receiving the next size up. General fit problems are also cited, with one dress hanging strangely and clinging to unflattering parts despite seeming large. Additionally, length can be a concern, as one petite user found a maxi dress to be two inches above her feet."
Visualizations:



4.5 Reply Generator

The integrated **Reply Generator** (`reply.py`) crafts AI-based, tone-appropriate responses for customer service or brand communication. It dynamically adapts tone and phrasing based on the detected sentiment of each review.
Tone Adaptation Examples:

- **Positive Review:** "We're thrilled you loved your purchase! Thank you for your feedback."
- **Negative Review:** "We're sorry for your experience. We'll ensure better fit and quality in future designs."



This feature demonstrates how AI can support **empathetic, consistent, and human-like brand communication**, enhancing both responsiveness and user trust.

4.6 Performance Metrics

The system's efficiency was evaluated across multiple dimensions, including data loading, inference latency, and frontend responsiveness. The results indicate a strong balance between computational performance and user experience.

Metric	Result	Commentary
Cache Load Time	~2.1 seconds	From JSONL to in-memory DataFrame
Review Analysis Latency	~0.7 seconds / review	Orchestrator + Emotion + Sentiment
RAG Query Time	4-5 seconds	Includes FAISS retrieval + Gemini API
Frontend FPS	~60 fps	Smooth rendering on Chrome dashboard

These results demonstrate that the system achieves an optimal balance between **speed, interpretability, and analytical accuracy**, making it suitable for both academic research and real-world deployment.

5 Design Reflection

The **Feedback Analyzer** project represents the intersection of artificial intelligence, data visualization, and human-centered design thinking. Beyond its analytical capabilities, the system embodies a design philosophy: making complex AI insights interpretable, actionable, and empathetic for non-technical users such as designers and brand strategists.

5.1 Design Goals and Learnings

At the outset, the design challenge was clear how can we transform thousands of unstructured customer reviews into insights that designers can trust and act upon? Through iterative development and user testing, several key learnings emerged:

- **Interpretability over Complexity:** Instead of relying on opaque black-box models, the system emphasizes explainable outputs - clear sentiment labels, emotion breakdowns, and transparent NPS computations. This helps designers build confidence in AI results.

- **From Data to Story:** Visual dashboards and AI summaries convert statistical outputs into narrative insights, allowing designers to "read the story" behind user sentiment rather than interpret raw numbers.
- **Human–AI Collaboration:** The RAG-bot module and Reply Generator are not replacements for designers but collaborative tools extending their capabilities to understand feedback faster and respond empathetically.

5.2 User-Centered AI Integration

The system was designed around **designers as end-users**, not data scientists. Key design principles that guided the interface and interaction model include:

- **Clarity:** Visual charts communicate sentiment and emotion at a glance, helping users quickly grasp the overall tone of customer feedback.
- **Focus:** Filters such as *Department* and *Class* enable users to narrow analysis to relevant product categories, maintaining analytical precision without overwhelming detail.
- **Empathy:** The tone-adaptive *Reply Generator* ensures that AI-generated responses remain aligned with the brand voice and customer emotion, promoting human-like communication.
- **Feedback Loop:** Designers can refine their understanding of customer sentiment through interactive exploration and RAG-based queries, turning passive data into an active design resource.

This approach transforms feedback analysis from a technical task into a creative, decision-support process that bridges data analytics and design thinking.

5.3 Ethical and Responsible AI Considerations

In developing the system, particular attention was given to **data ethics** and **bias awareness**. The following measures were implemented to ensure fairness, transparency, and accountability throughout the AI pipeline:

- **Bias in Data:** Since the dataset represents a specific product category (*women's fashion*), models may reflect inherent demographic or linguistic biases. These were mitigated by neutral preprocessing techniques and balanced sampling during emotion clustering.
- **Transparency:** The inclusion of explainable intermediate outputs such as sentiment scores and emotion probabilities ensures that results remain traceable and interpretable.
- **AI Responsibility:** The system intentionally avoids making prescriptive or automated decisions. Instead, it provides assistive recommendations that keep the human designer in control of interpretation and action.

By foregrounding **transparency**, **empathy**, and **interpretability**, the *Feedback Analyzer* demonstrates a responsible model for integrating AI into design workflows.

5.4 Reflection on Design Process

From early exploratory data analysis (EDA) and prototype sketches to the final interactive dashboard, the project evolved through multiple **iterative design cycles**. Each phase contributed to refining both functionality and user experience:

- **Exploration:** Understanding the dataset structure, identifying user pain points, and framing the design problem from a user-centered perspective.
- **Synthesis:** Translating user needs into concrete visual and interactive AI tools that bridge analytical rigor with design intuition.
- **Refinement:** Enhancing system performance, interpretability, and aesthetic coherence through feedback, testing, and iteration.

Each stage reinforced the importance of **collaboration between data analysis and design sensibility** - an embodiment of the *DES646* course objective to make AI both accessible and meaningful for creative decision-making.

5.5 Key Takeaways

The project highlights several overarching lessons at the intersection of artificial intelligence and design practice:

- **AI as an Amplifier:** AI can amplify - not replace - human understanding in the design feedback loop, serving as a collaborator rather than a substitute.
- **Human-Centered Priorities:** Designers value *clarity*, *empathy*, and *context* over sheer analytical power, reaffirming the need for interpretability in AI systems.
- **Visual Storytelling:** Data visualization and narrative framing remain central to making AI outputs meaningful and actionable for creative users.
- **Future Directions:** Next-generation tools should continue to emphasize *explainability*, *inclusivity*, and *trust* in human-AI collaboration.

6 Future Scope

The Feedback Analyzer system establishes a foundation for AI-driven design intelligence, yet its potential extends far beyond the current prototype. As the intersection of AI and design continues to evolve, the following extensions can further enhance the system's capabilities, scalability, and real-world impact.

6.1 Technical Enhancements

To extend the capabilities and scalability of the *Feedback Analyzer* system, several technical improvements can be pursued in future iterations:

- **Multilingual Review Analysis:** Integrating a multilingual NLP pipeline would enable the system to analyze feedback in languages beyond English, using models such as the Google Translate API or multilingual BERT. This enhancement would expand the tool's applicability to global e-commerce platforms.
- **Image-Aided Sentiment Detection:** Many reviews include images that implicitly express user satisfaction. Incorporating visual sentiment analysis through CNN or CLIP-based models would allow a more holistic understanding of user experiences.
- **Real-Time Stream Processing:** Deploying the backend on cloud platforms such as Firebase or AWS Lambda could enable real-time analysis of customer reviews as they are posted online, enhancing responsiveness for live product feedback monitoring.

- **Enhanced RAG Summarization:** The current RAG pipeline can be extended with hybrid retrieval methods (FAISS + BM25) or fine-tuned summarization models, improving the accuracy and contextual relevance of generated insights.

6.2 UX and Frontend Improvements

Beyond backend and model optimizations, user experience (UX) and interface design play a crucial role in shaping how effectively designers engage with AI-generated insights. Future iterations of the *Feedback Analyzer* could include the following enhancements:

- **Dynamic Visualization Dashboards:** Incorporating interactive, drill-down analytics would allow designers to explore feedback hierarchically from department $\rightarrow$ class $\rightarrow$ product $\rightarrow$ individual review. Such dynamic visualizations would make data exploration more intuitive and context-rich.
- **Custom Query Interface:** Expanding the RAG-based question-answering module into a chat-style assistant could facilitate natural, conversational exploration of design insights, enabling users to query data in their own words.
- **Cross-Platform Support:** Deploying the Chrome extension as a universal browser plugin (compatible with Edge, Firefox, and Safari) would enhance accessibility and usability across diverse research and retail environments.

6.3 Predictive and Prescriptive Extensions

Beyond descriptive analysis, the next evolution of the *Feedback Analyzer* could integrate predictive and prescriptive intelligence - moving from understanding user sentiment to anticipating and addressing user needs.

- **Customer Retention Prediction:** Leveraging historical review data, the system could predict the likelihood of repeat purchases or customer churn. These insights would help design and marketing teams identify at-risk segments and tailor engagement strategies accordingly.
- **Design Suggestion Generator:** A future AI module could cluster recurring feedback patterns and automatically recommend design improvements for instance, “reduce sleeve length” or “enhance fabric stretch.” Such prescriptive guidance would bridge the gap between user sentiment analysis and actionable product innovation.

6.4 Academic and Commercial Potential

The *Feedback Analyzer* demonstrates both pedagogical and industry-level relevance, bridging academic exploration with practical application.

- **Educational Use:** The system can be adapted as a learning platform for design students to understand how artificial intelligence interprets emotions, intent, and sentiment in textual feedback. It provides a tangible interface for exploring human–AI interaction and the ethics of automated interpretation.
- **Commercial Deployment:** A packaged version of the *Feedback Analyzer* could be marketed to retail brands as a *Feedback Intelligence Dashboard*, enabling real-time monitoring and analysis of customer experience data. This would allow design and marketing teams to make faster, insight-driven decisions.
- **Patent Readiness:** The integrated use of RAG-driven insights, predictive NPS computation, and real-time sentiment visualization represents a novel approach

to design analytics. This innovation positions the system as IPDF-ready for potential patent disclosure and commercial evaluation.

6.5 Vision

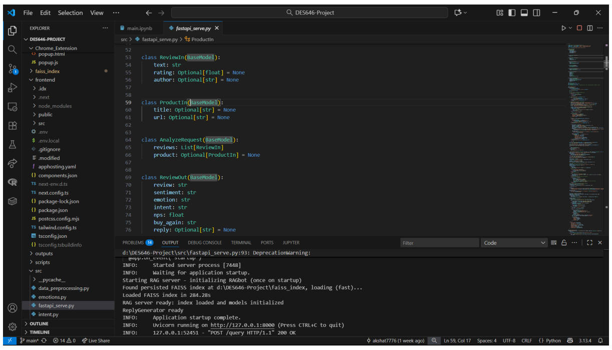
Ultimately, the long-term vision of the *Feedback Analyzer* is to evolve into a **Design Decision Support System (DDSS)** , a platform where artificial intelligence continuously learns from human feedback, identifies design flaws, and co-creates with designers.
By merging quantitative AI precision with qualitative human empathy, the system aspires to redefine the boundaries of creative collaboration paving the way for a new paradigm of *AI-augmented creative intelligence* in design.

7 Conclusion

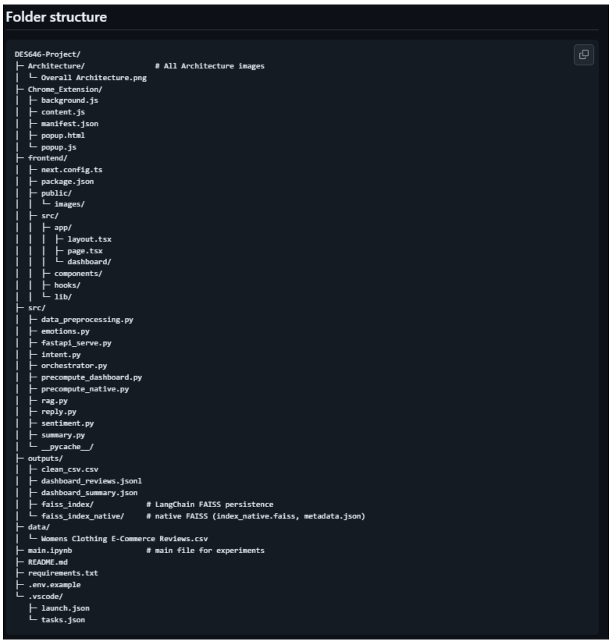
The Feedback Analyzer project demonstrates how artificial intelligence can meaningfully augment design decision-making by transforming raw, unstructured customer reviews into structured, interpretable insights.
Through its integration of **sentiment analysis, emotion detection, intent recognition,** and **RAG-based contextual summarization**, the system bridges the gap between quantitative analytics and qualitative human understanding.
By combining **FastAPI** (for intelligent backend processing), **Next.js** (for interactive data visualization), and a **Chrome extension** (for real-time application), the project offers a full-stack ecosystem that empowers designers to explore, interpret, and respond to customer feedback efficiently.
The inclusion of **precomputed caching, FAISS-based retrieval, and LLM-driven synthesis** ensures both technical performance and analytical depth, while the **Reply Generator** introduces empathy into automated feedback handling.
From a design perspective, the system emphasizes **transparency, interpretability, and collaboration** - showing that AI tools can act as creative partners rather than black-box evaluators.
The project aligns with the objectives of **DES646 - AI/ML for Designers**, showcasing how design students can use AI not merely as a computational tool but as a medium to **understand people, context, and emotion** through data.
Ultimately, Feedback Analyzer represents a prototype of a future design intelligence system - one where **AI augments human creativity**, supports **data-driven empathy**, and turns customer voices into actionable insights that shape better, more human-centered products.

8 Appendix

8.1 Code snippets



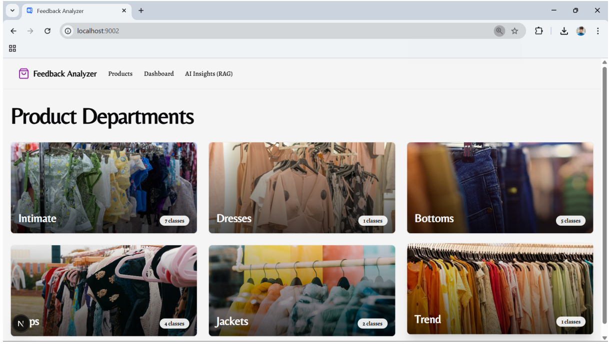
8.2 Folder structure diagram

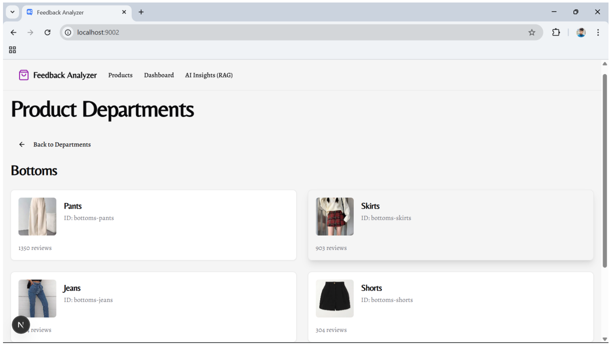


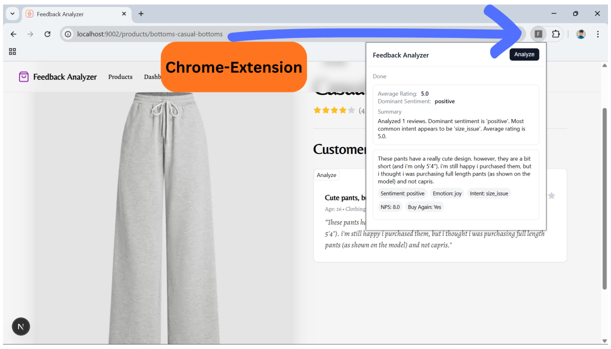
Folder structure

```
DES646-Project/
├─ Architecture/                # All Architecture images
│  └─ Overall Architecture.png
├─ Chrome_Extension/
│  ├─ background.js
│  ├─ content.js
│  ├─ manifest.json
│  ├─ popup.html
│  └─ popup.js
├─ frontend/
│  ├─ next.config.ts
│  ├─ package.json
│  ├─ public/
│  │  └─ images/
│  ├─ src/
│  │  ├─ app/
│  │  │  ├─ layout.tsx
│  │  │  ├─ page.tsx
│  │  │  └─ dashboard/
│  │  ├─ components/
│  │  ├─ hooks/
│  │  └─ lib/
├─ src/
│  ├─ data_preprocessing.py
│  ├─ emotions.py
│  ├─ fastapi_serve.py
│  ├─ intent.py
│  ├─ orchestrator.py
│  ├─ precompute_dashboard.py
│  ├─ precompute_native.py
│  ├─ rag.py
│  ├─ reply.py
│  ├─ sentiment.py
│  ├─ summary.py
│  └─ __pycache__/
├─ outputs/
│  ├─ clean_csv.csv
│  ├─ dashboard_reviews.jsonl
│  ├─ dashboard_summary.json
│  ├─ faiss_index/            # LangChain FAISS persistence
│  └─ faiss_index_native/     # native FAISS (index_native.faiss, metadata.json)
├─ data/
│  └─ Womens Clothing E-Commerce Reviews.csv
├─ main.ipynb                 # main file for experiments
├─ README.md
├─ requirements.txt
├─ .env.example
└─ .vscode/
   ├─ launch.json
   └─ tasks.json
```

8.3 Snapshot of Web-Dashboard







9 Citations

- Hota, N. (2018). Women's E-Commerce Clothing Reviews Dataset.
- Rajaraman, A., Ullman, J. D. (2011). Mining of Massive Datasets. Cambridge University Press.
- Hutto, C., Gilbert, E. (2014). VADER: A Parsimonious Rule-based Model for Sentiment Analysis of Social Media Text. Proceedings of the Eighth International Conference on Weblogs and Social Media (ICWSM).
- Google. (2024). Gemini API Documentation. Google AI Developer Portal.
- Facebook AI Research. (2017). FAISS: A Library for Efficient Similarity Search and Clustering of Dense Vectors.
- FastAPI Documentation. (2024). FastAPI Web Framework.
- Next.js Documentation. (2024). Next.js 14 – React Framework for Production. Vercel.
- Pedregosa, F., et al. (2011). Scikit-learn: Machine Learning in Python. Journal of Machine Learning Research
- Hugging Face. (2023). Transformers and Sentence Embeddings.
- Plotly. (2024). Plotly Graphing Library for JavaScript and Python
- OpenAI. (2024). Responsible AI and Explainability Principles. OpenAI Research Blog.