Product Recommendation System : Task Report

Author: Sivasankaran K
Date: June 2025

Page 1: Introduction

With the increasing number of mobile devices and a flood of user-generated reviews on e-commerce platforms like Flipkart, users often face difficulty choosing the right smartphone. This project aims to build a smart mobile recommender system that combines collaborative filtering, sentiment analysis, and Retrieval-Augmented Generation (RAG) using local large language models (LLMs). It processes over 5,000 real Flipkart reviews to assist buyers in making better purchasing decisions.

Page 2: Data Collection & Preprocessing

- **Data Source**: Flipkart mobile product pages
- **Tools Used**: Selenium(scraping); Pandas (processing)
- **Reviews Collected**: 5,000+ reviews across 15 smartphones

Cleaning Steps: - Removed duplicates and nulls - Normalized column types (e.g., product ID, user ID) - Stored final cleaned dataset as `cleaned_final_project_data.csv`

Page 3: Sentiment Analysis

- **Library**: TextBlob
- **Task**: Classify each review as `positive`, `negative`, or `neutral`

Process: - Polarity scores calculated from each review - Sentiment labels assigned based on thresholds: - `> 0.1`: Positive - `< -0.1`: Negative - `else`: Neutral

Saved as `final_processed_reviews.csv`

Page 4: Collaborative & Hybrid Filtering

- **Method**: Truncated SVD on User-Product matrix
- **Matrix**: Users × Product Ratings

Collaborative Filtering: - SVD decomposed the matrix into latent features - Cosine similarity computed between users

Hybrid Filtering: - Combined SVD rating with average sentiment score per product - Weighted formula: `0.7 * rating + 0.3 * sentiment` - Output: `recommend_hybrid(user_id)` returns top 5 personalized recommendations

---

Page 5: Vector Store (RAG Setup)

- **Embedding Model**: `sentence-transformers/all-MiniLM-L6-v2`
- **Vector Store**: FAISS

Steps: - Grouped reviews by product - Used LangChain's RecursiveCharacterTextSplitter to chunk data - Embedded using HuggingFaceEmbeddings - Saved vector store as `models/vector_store/`

Purpose: Enable fast semantic search and grounding for question-answering via RAG.

Page 6: Local LLM + RAG Integration

- **LLM Used**: Mistral (via Ollama) fallback to Gemma 2B
- **Framework**: LangChain

QA Setup: - Built a RetrievalQA chain using FAISS retriever and Ollama LLM - Passed filtered product IDs + user question to prompt template - Returned grounded, review-based answers

Example: *Q: Is the camera good?* A: Based on reviews for Pixel 7a and OPPO F27, most users highlighted strong low-light performance and sharp detail quality.

Page 7: Streamlit Frontend (app.py)

- **Frontend Tool**: Streamlit
- **UI Features**:
    - Sidebar filter for feature & sentiment
    - Display of filtered product list
    - Text input for user question
    - Button to trigger RAG + LLM answer

UX Notes: - Answer only shown once product filter applied - Response shown along with product list used

---

Page 8: Evaluation Metrics

Metric	Result
RMSE (Hybrid)	1.45
Precision@3	0.67
Sentiment Accuracy	1.00
Sentiment F1 Score	1.00
RAG QA Human Score	4.2/5

Evaluation Scripts: - `evaluation/evaluate.py` - Used manually labeled `sentiment_test_labeled.csv`

Page 9: What We Tried & What Failed

Tried: - OpenAI GPT-3.5 for RAG → Hit quota error (429) - `mistralai/Mistral-7B-Instruct-v0.1` on HuggingFace → Gated access, failed to load - Using `textblob` inside app → Too slow, moved to preprocessing - Full LLM answer inside `streamlit` button → Caused latency, fixed by caching - Mistral on low-RAM device → Replaced with Gemma 2B

Learnings: - Lightweight local models like `gemma:2b` are better for <4 GB RAM - LangChain `.run()` fails with multiple outputs → must use `.invoke()` - Keeping vector store outside UI logic speeds up Streamlit

---

Page 10: Conclusion

This Flipkart Smart Mobile Recommender achieves all task goals: - Personalized recommendation with hybrid logic - RAG-powered product Q&A grounded in real user reviews - Works entirely offline (Mistral/Gemma + FAISS) - Evaluated using both metrics and human feedback

This project is ready for demonstration and submission.

Next Improvements: - Add review summarization - Include product metadata (price, specs) - Integrate model comparison toggle.

Task Completed by: **Sivasankaran K**