**🧠 Intelligent Complaint Analysis for Financial Services**

**Final Submission – Report**

**📌 Introduction**

Financial institutions receive thousands of customer complaints every month. Manually analyzing these complaints for trends, recurring issues, or product-specific pain points is time-consuming and error-prone.

This project builds an AI-powered **Retrieval-Augmented Generation (RAG)** system to help financial service teams retrieve and summarize relevant customer feedback automatically. Using real-world data from the **Consumer Financial Protection Bureau (CFPB)**, we designed a pipeline that embeds, indexes, and semantically retrieves complaint narratives to answer questions with relevant, trustworthy context.

**⚙️ Methodology**

We approached the project in four structured tasks:

**Task 1: Exploratory Data Analysis (EDA) & Preprocessing**

* Loaded the full CFPB dataset.
* Filtered complaints to focus on 5 key financial products: Credit Card, Personal Loan, BNPL, Savings Account, and Money Transfers.
* Removed entries with empty narratives and cleaned text (lowercased, removed boilerplate).
* Saved the cleaned dataset (filtered\_complaints.csv) for downstream processing.

**Task 2: Text Chunking, Embedding, and Vector Store Indexing**

* Long narratives were split using RecursiveCharacterTextSplitter with chunk\_size=300 and chunk\_overlap=50.
* Used sentence-transformers/all-MiniLM-L6-v2 for embedding due to its speed and semantic accuracy.
* Stored embeddings and metadata in **ChromaDB**, allowing fast vector search.

**Task 3: Retrieval and Generation (RAG Core Logic)**

* User questions are embedded and matched to top-k relevant chunks.
* A prompt template provides context to the LLM to answer questions using retrieved chunks.
* Hugging Face LLM (distilGPT2) used for generating contextual answers.
* Conducted qualitative evaluations on 5 real-world queries to test output relevance and accuracy.

**Task 4: Interactive Chat Interface**

* Built a user-friendly **Streamlit** app.
* Allows inputting questions, receiving AI-generated answers, and viewing source complaint excerpts.
* Enhances transparency and trust in the system with source display.

**🚧 Challenges & Solutions**

**Challenge 1:** CFPB narratives were unstructured, repetitive, and lengthy.  
✅ *Solution:* Text cleaning and chunking helped standardize the inputs, while embedding model minimized semantic noise.

**Challenge 2:** Vector DB crashed with large data during indexing.  
✅ *Solution:* Reduced batch size and excluded unnecessary large files using .gitignore.

**Challenge 3:** Initial GitHub repo failed to push due to excessive file size.  
✅ *Solution:* Reinitialized a fresh repo and versioned only essential code/artifacts.

**Challenge 4:** Streamlit warnings due to improper execution.  
✅ *Solution:* Used streamlit run app.py instead of plain python app.py to launch the app properly.

**✅ Evaluation of Outcomes and Recommendations**

| **Question** | **Quality (1–5)** | **Analysis** |
| --- | --- | --- |
| What are the most common complaints about personal loans? | 5 | Identified repeated delay themes and interest issues clearly. |
| Do users face problems with money transfers? | 4 | Answer accurate, though fewer complaint sources available. |
| Are savings accounts often closed without notice? | 5 | Excellent alignment with retrieved evidence. |
| How are credit card disputes handled? | 4 | Answer informative, could improve source variety. |
| Is there a delay in BNPL refunds? | 5 | Clear answer grounded in retrieved complaints. |

**Recommendations for Improvement:**

* Integrate feedback loops (user thumbs-up/down) for retraining the model.
* Add streaming LLMs (like Mistral or LLaMA) for faster, low-latency generation.
* Expand dataset to cover more financial products for broader use.

**🧩 Alignment with Business Needs & Technical Execution**

This solution aligns with financial institutions’ need to reduce support workload, identify product pain points, and respond rapidly to regulatory concerns.  
From a technical standpoint, the use of sentence-transformers, semantic chunking, and a vector DB + LLM combination ensures fast, accurate, and explainable outputs.

The report and pipeline maintain transparency by showing **retrieved evidence**, reinforcing trust with stakeholders.

**🎯 Conclusion**

This project successfully delivers a working **AI-powered complaint analyzer** for the financial services sector.  
It combines cutting-edge **RAG architecture**, intuitive **UI**, and real-world **CFPB data** to generate grounded, useful answers from thousands of customer complaints.

With minimal compute, modular design, and open-source tooling, the system is scalable and ready for production or further research.