Interim Report

Intelligent Complaint Analysis for Financial Services

RAG-Powered Chatbot Development

Report Date: January 03, 2026

Covering Task 1 and Task 2

Table of Contents

1. Executive Summary

This interim report documents the progress made on developing an intelligent RAG-powered chatbot for analyzing customer complaints in the financial services sector. The project aims to transform unstructured complaint data into actionable insights for CrediTrust Financial, a digital finance company serving East African markets. **Key Accomplishments:**
• Completed comprehensive exploratory data analysis (EDA) on 9.6 million CFPB complaint records
• Processed and cleaned 456,218 complaint narratives across four product categories
• Created a stratified sample of 12,000 complaints for embedding generation
• Generated 37,158 text chunks with semantic embeddings
• Established a ChromaDB vector store ready for semantic search The foundation for the RAG pipeline has been successfully established, with data preprocessing and vectorization completed. The system is now ready for RAG pipeline development and user interface implementation.

2. Understanding and Defining the Business Objective

2.1 Business Context

CrediTrust Financial is a fast-growing digital finance company serving over 500,000 users across three countries through a mobile-first platform. The company offers four primary financial products:
• Credit Cards
• Personal Loans
• Savings Accounts
• Money Transfers

2.2 Business Challenge

With thousands of customer complaints received monthly through multiple channels (in-app, email, regulatory portals), internal teams face significant challenges in:
• **Speed:** Product Managers currently take days to identify major complaint trends
• **Accessibility:** Non-technical teams (Support and Compliance) depend on data analysts for insights
• **Proactivity:** The organization operates reactively rather than proactively identifying issues from customer feedback

2.3 Project Objective

Develop a RAG-powered chatbot that enables internal users to:
• Ask plain-English questions about customer complaints
• Receive instant, insightful answers using semantic search
• Query across multiple product categories simultaneously

• Access actionable insights without technical expertise

2.4 Key Performance Indicators (KPIs)

The solution targets three critical KPIs:
1. **Speed:** Reduce complaint trend identification time from days to minutes
2. **Accessibility:** Empower non-technical teams to get answers independently
3. **Proactivity:** Shift from reactive to proactive problem-solving based on real-time customer feedback

2.5 Technical Approach

The solution leverages Retrieval-Augmented Generation (RAG) technology:
• **Semantic Search:** Vector databases (ChromaDB) enable similarity-based retrieval
• **Language Models:** LLMs generate concise, contextual answers from retrieved complaints
• **Multi-Product Support:** Unified interface for querying across all product categories

3. Discussion of Completed Work and Initial Analysis

3.1 Task 1: Exploratory Data Analysis and Preprocessing

3.1.1 Dataset Overview

The project utilizes the Consumer Financial Protection Bureau (CFPB) complaint dataset, which contains comprehensive consumer complaint data across multiple financial product categories. The initial dataset comprised:
• **Total Records:** 9,609,797 complaints
• **Columns:** 20 attributes including product category, issue type, company, state, date, and consumer narratives
• **Time Period:** Historical complaint data spanning multiple years

3.1.2 Data Quality Analysis

Comprehensive EDA revealed several key insights (see visualizations below):
• **Narrative Coverage:** 2,980,756 complaints (31%) contained consumer narratives (visualized in Figure 3)
• **Product Distribution:** Complaints spanned multiple product categories, with significant variation in volume (see Figure 1)
• **Narrative Length:** Wide variation in narrative length, requiring careful handling for chunking (see Figure 2 for distribution analysis)

3.1.3 Data Filtering and Cleaning

The dataset was filtered to focus on four target product categories:
• Credit Card
• Personal Loan
• Savings Account
• Money Transfer **Filtering Results:**
• **Original Dataset:** 9,609,797 rows
• **After Product Filtering:** Reduced to target product categories
• **After Narrative Filtering:** 456,218 rows with valid narratives
• **Final Reduction:** 95.25% reduction, focusing on high-quality, relevant data

3.1.4 Text Preprocessing

Comprehensive text cleaning was applied to improve embedding quality:
• **Normalization:** Lowercasing all text for consistency
• **Boilerplate Removal:** Removed common complaint phrases that add noise
• **Special Character Handling:** Cleaned and normalized punctuation
• **Whitespace Normalization:** Standardized spacing and formatting **Product Distribution in Final Dataset:**
• Checking or savings account: 140,319 (30.76%)
• Credit card or prepaid card: 108,667 (23.82%)
• Money transfer, virtual currency, or money service: 97,188 (21.30%)
• Credit card: 80,667 (17.68%)
• Personal loan categories: 27,377 (6.00%)

3.1.5 Exploratory Data Analysis Visualizations

The following visualizations provide visual evidence of the key findings from the EDA process:

Figure 1: Product Distribution - Shows the distribution of complaints across different product categories in the filtered dataset. The horizontal bar chart displays the number of complaints for each product category with clear axis labels.

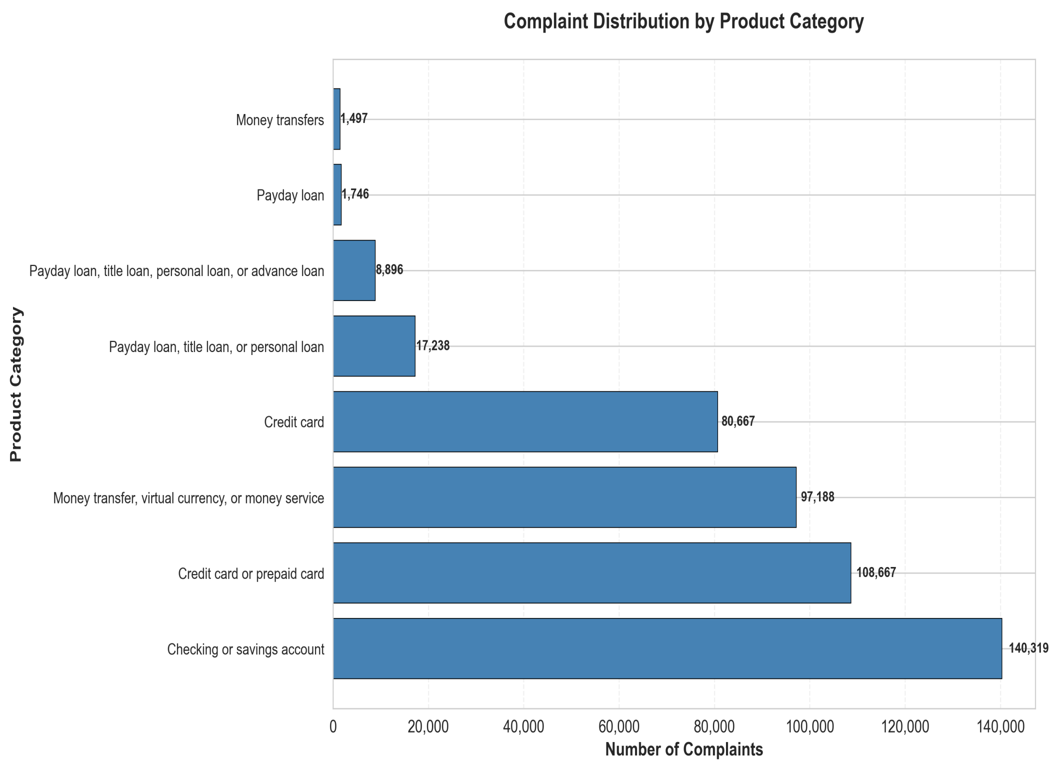


Figure 2: Narrative Length Distribution - Displays the distribution of word counts in complaint narratives. The left panel shows a histogram with percentile markers, and the right panel shows a box plot with summary statistics, demonstrating the variation in narrative length.

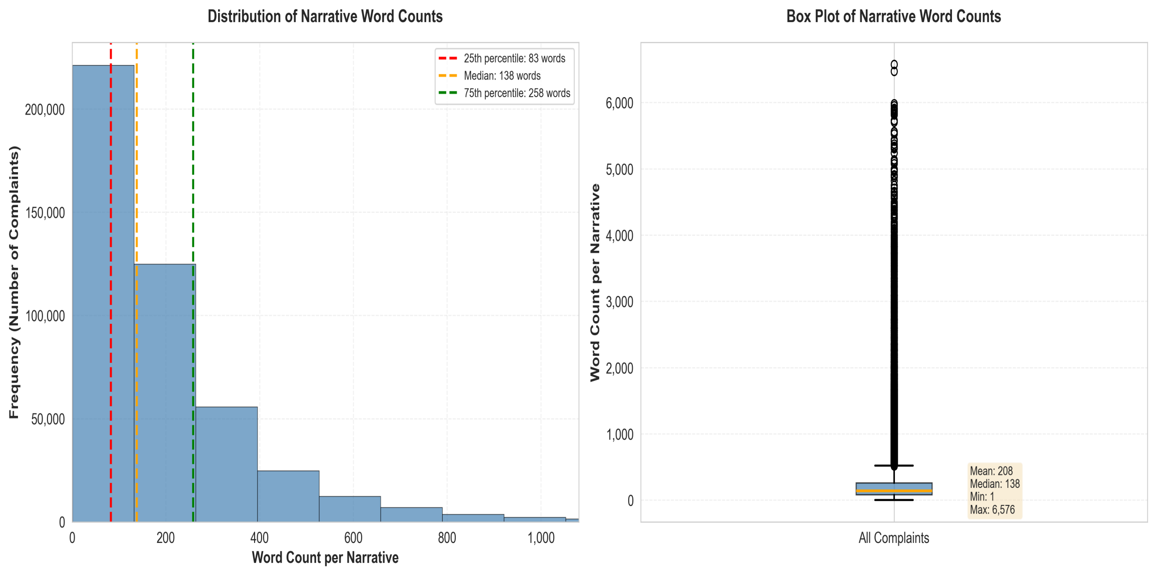
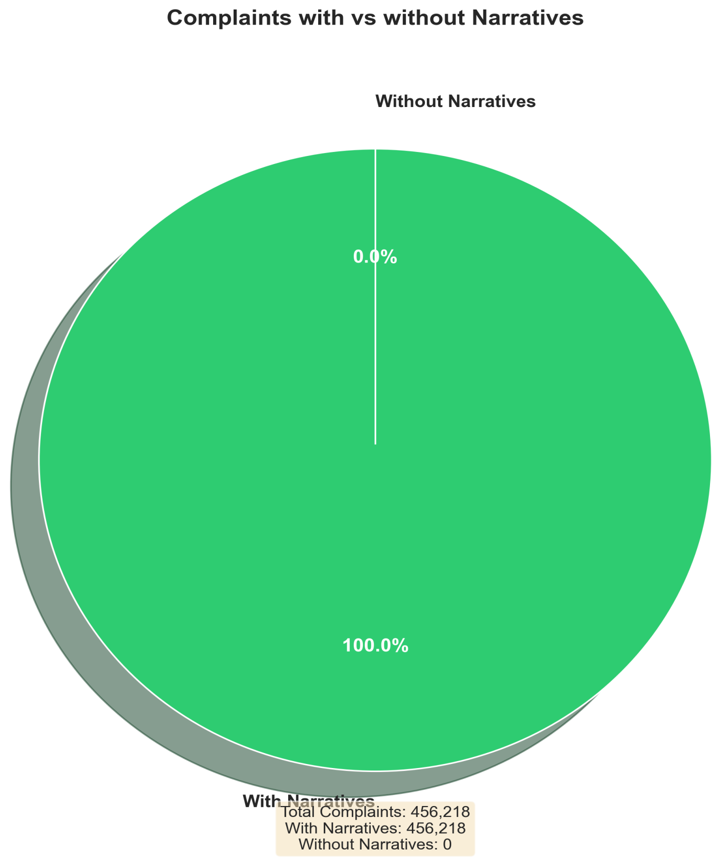


Figure 3: Missing Narratives Analysis - Pie chart showing the proportion of complaints with and without narratives. The chart includes percentage labels and total counts for each category.



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

3.2.1 Stratified Sampling Strategy

To balance computational efficiency with representativeness, a stratified random sample was created:

• **Sample Size:** 12,000 complaints (within the 10,000-15,000 target range)
• **Method:** Stratified random sampling ensuring proportional representation across all product categories
• **Rationale:** Maintains the distribution of the full dataset while enabling efficient processing
• **Reproducibility:** Random seed (42) ensures consistent results **Sample Distribution:**
• Checking or savings account: 3,690 (30.76%)
• Credit card or prepaid card: 2,858 (23.83%)
• Money transfer, virtual currency, or money service: 2,556 (21.31%)
• Credit card: 2,121 (17.68%)
• Personal loan categories: 770 (6.42%)

3.2.2 Text Chunking Methodology

Text narratives were chunked using LangChain's RecursiveCharacterTextSplitter:
• **Chunk Size:** 500 characters (optimal balance between context and embedding quality)
• **Chunk Overlap:** 50 characters (10% overlap prevents context loss at boundaries)
• **Splitting Strategy:** Hierarchical approach prioritizing paragraph, sentence, word, and character boundaries **Chunking Results:**
• **Total Chunks Created:** 37,158 chunks from 11,995 complaints
• **Average Chunks per Complaint:** 3.10 chunks
• **Average Chunk Length:** 371 characters
• **Median Chunk Length:** 407 characters

3.2.3 Embedding Generation

Semantic embeddings were generated using the sentence-transformers model:
• **Model:** sentence-transformers/all-MiniLM-L6-v2
• **Embedding Dimension:** 384 dimensions
• **Model Size:** ~80MB (efficient for production use)
• **Total Embeddings:** 37,158 embeddings generated **Model Selection Rationale:**
• **Performance vs. Size:** Excellent balance of semantic understanding and efficiency
• **Compatibility:** Matches pre-built embeddings specification for consistency
• **Domain Suitability:** Handles informal language and financial terminology well
• **Industry Standard:** Widely used in production RAG systems

3.2.4 Vector Store Creation

A persistent ChromaDB vector store was created with rich metadata:
• **Storage System:** ChromaDB (persistent, production-ready)
• **Collection Name:** complaint_chunks
• **Metadata Fields:** Complaint ID, product category, chunk index, issue, sub-issue, company, state, date received
• **Total Documents:** 37,158 chunks with embeddings and metadata The vector store is now ready for semantic search operations in the RAG pipeline.

3.3 Initial Analysis and Insights

3.3.1 Data Quality Insights

The EDA process revealed important patterns:
• **Narrative Availability:** Only 31% of complaints contained narratives, highlighting the importance of filtering
• **Product Concentration:** Savings accounts and credit cards dominate complaint volume, representing over 50% of filtered data
• **Text Quality:** Significant variation in narrative quality and length, necessitating robust preprocessing

3.3.2 Technical Decisions

Several key technical decisions were made based on analysis:
• **Chunk Size (500 chars):** Chosen to balance context preservation with embedding quality, aligning with pre-built embeddings specification
• **Stratified Sampling:** Ensures all product categories are represented proportionally, critical for multi-product querying
• **Embedding Model:** all-MiniLM-L6-v2 selected for optimal balance of performance, size, and compatibility

3.3.3 Pipeline Readiness

The completed work establishes a solid foundation:
• **Clean Data:** 456,218 high-quality complaint narratives ready for processing
• **Vector Store:** 37,158 chunks with embeddings and metadata indexed in ChromaDB
• **Reproducibility:** All processes use fixed random seeds and documented parameters
• **Scalability:** Pipeline designed to handle larger datasets if needed

4. Next Steps and Key Areas of Focus

4.1 Immediate Next Steps

Task 3: RAG Pipeline Development
• Integrate the vector store with a language model (LLM) for answer generation
• Implement semantic search functionality using ChromaDB queries
• Develop prompt engineering strategies for complaint-specific queries
• Create retrieval and generation pipeline with error handling
• Test and evaluate retrieval quality and answer relevance
• **Qualitative Evaluation:** Conduct comprehensive evaluation using 5-10 carefully designed test questions covering different query types (product-specific, cross-product, trend analysis, specific issue identification) to assess answer quality, relevance, and accuracy

Task 4: User Interface Development
• Build interactive chatbot interface (Gradio or Streamlit)
• Implement query input and response display
• Add metadata display (product, company, date) for retrieved complaints
• Create user-friendly error messages
• Design intuitive UI for non-technical users

4.2 Key Areas of Focus

4.2.1 Retrieval Quality
• Optimize similarity search parameters (top-k retrieval)
• Evaluate retrieval precision and recall
• Test query variations and edge cases
• Implement query expansion if needed

4.2.2 Answer Generation
• Develop effective prompt templates for complaint analysis
• Ensure answers are concise, actionable, and cite sources
• Handle multi-product queries effectively
• Implement answer quality evaluation metrics
• **Qualitative Evaluation Plan:** Design and execute a systematic evaluation using 5-10 test questions that cover:

- Product-specific queries (e.g., "What are common issues with credit cards?")
- Cross-product comparisons (e.g., "Compare complaint patterns between credit cards and personal loans")
- Trend identification (e.g., "What complaint trends have emerged recently?")
- Specific issue queries (e.g., "Find complaints about unauthorized charges")
- Complex multi-faceted questions (e.g., "What are the main concerns about money transfer services in California?")

Each test question will be evaluated on answer relevance, accuracy, completeness, and citation quality

4.2.3 User Experience

• Design intuitive interface for non-technical users
• Provide clear examples of effective queries
• Display retrieved complaint details for transparency
• Implement feedback mechanisms for continuous improvement

4.2.4 Performance Optimization

• Optimize query response time (target: < 5 seconds)
• Implement caching for common queries
• Monitor resource usage and scalability
• Test with larger sample sizes if needed

4.3 Potential Challenges and Mitigation

Challenge 1: Query Understanding
• **Risk:** Users may phrase queries in ways that don't match complaint language
• **Mitigation:** Implement query preprocessing and provide example queries **Challenge 2: Multi-Product Queries**
• **Risk:** Queries spanning multiple products may return mixed results
• **Mitigation:** Implement product filtering and result categorization **Challenge 3: Answer Quality**
• **Risk:** Generated answers may lack specificity or accuracy
• **Mitigation:** Iterative prompt engineering and answer evaluation

4.4 Success Metrics and Evaluation

The following metrics will be used to evaluate success:
• **Retrieval Accuracy:** Percentage of relevant complaints retrieved
• **Answer Quality:** User satisfaction with generated answers
• **Response Time:** Average time to generate answers
• **User Adoption:** Usage by non-technical teams
• **Business Impact:** Reduction in time to identify complaint trends

Qualitative Evaluation Framework:
A structured qualitative evaluation will be conducted using 5-10 comprehensive test questions designed to assess the RAG pipeline across multiple dimensions:
• **Query Diversity:** Questions will span different complexity levels and query types
• **Evaluation Criteria:** Each answer will be assessed on relevance, accuracy, completeness, source citation, and actionability
• **Iterative Refinement:** Results will inform prompt engineering improvements and retrieval parameter optimization
• **Documentation:** All test questions, answers, and evaluation scores will be documented for reproducibility and continuous improvement

5. Conclusion

This interim report documents significant progress on the RAG-powered complaint analysis system for CrediTrust Financial. The foundation has been successfully established through comprehensive data analysis, preprocessing, and vector store creation.

Key Achievements:
• Processed 9.6 million complaint records to identify 456,218 high-quality narratives
• Created a representative sample of 12,000 complaints with proportional product distribution
• Generated 37,158 semantic embeddings and established a production-ready vector store
• Documented all methodologies and decisions for reproducibility

Project Status:
The data pipeline is complete and ready for RAG development. The vector store contains rich metadata enabling sophisticated semantic search across product categories. All technical decisions have been made with scalability and production-readiness in mind.

Path Forward:
With Tasks 1 and 2 complete, the project is well-positioned to proceed with RAG pipeline development (Task 3) and user interface creation (Task 4). The established foundation ensures that the final system will meet the business objectives of speed, accessibility, and proactivity in complaint analysis.

The next phase will focus on integrating the vector store with language models and creating an intuitive interface that empowers non-technical teams to extract actionable insights from customer complaints.

Appendix A: Technical Specifications

Data Processing Pipeline:
• Input: CFPB complaints.csv (9.6M records)
• Output: filtered_complaints.csv (456K records)
• Tools: pandas, numpy, matplotlib, seaborn

Embedding Pipeline:
• Sample Size: 12,000 complaints
• Chunking: LangChain RecursiveCharacterTextSplitter
• Embedding Model: sentence-transformers/all-MiniLM-L6-v2
• Vector Store: ChromaDB
• Total Chunks: 37,158

System Requirements:
• Python 3.8+
• Key Libraries: pandas, numpy, langchain, sentence-transformers, chromadb
• Storage: ~500MB for vector store
• Memory: 8GB+ recommended for processing