Architecture Document: Home Loan Prediction and Recommendation System

1. Introduction

This document describes the architecture of the Home Loan Prediction and Recommendation System, whi
machine learning (ML) and a retrieval-augmented generation (RAG) approach for personalized financial re

2. System Overview

The system is a Flask-based web service that predicts the likelihood of a user applying for a home loan ba
their financial profile and provides personalized financial recommendations using a language model (LLM)
with a vector database.

3. Technology Stack

- Backend Framework: Flask
- Machine Learning: Scikit-learn, Joblib
- Vector Database: ChromaDB
- Language Model (LLM): Hugging Face and Groq's LLaMA
- Data Processing: Pandas, NumPy
- Configuration Management: dotenv

4. Key Components

4.1 Data Processing and Feature Engineering

- Extracts customer financial attributes such as income, account balance, employment status, etc.
- Categorical attributes are label-encoded.
- Numerical attributes are scaled using a pre-trained scaler.

4.2 Machine Learning Model

- A pre-trained ML model (home_loan_model.pkl) is used to predict the likelihood of applying for a home lo
- Predictions are binary: "Likely to apply" or "Unlikely to apply".

4.3 Retrieval-Augmented Generation (RAG) for Personalized Recommendations

- Uses ChromaDB for document retrieval based on similarity search.
- If the user is likely to apply for a home loan, relevant home loan products are recommended.
- Otherwise, investment product recommendations are provided.
- The retrieved context is fed into an LLM (LLaMA) to generate personalized suggestions.

5. API Endpoints

Endpoint	Method	Description
/api/predict_home_loan	POST	Predicts home loan likelihood and provides personalized recommenda

6. Deployment Considerations

- Model Hosting: Ensure that the ML model and scaler files are stored securely.
- Environment Variables: Store API keys securely using dotenv.
- Scalability: Use a WSGI server like Gunicorn for production deployments.
- Security: Validate and sanitize API inputs to prevent attacks.

7. Conclusion

This architecture efficiently integrates ML-based prediction with RAG-based recommendations, providing a
data-driven financial advisory system. Future enhancements can include real-time data integration and mu
capabilities.

8. System Architecture Diagram