AI-Driven Social Support Eligibility and Economic Enablement System

1 Introduction

Social support systems play a critical role in ensuring economic stability, social equity, and long-term empowerment of individuals and families. However, traditional eligibility assessment processes are often manual, time consuming, opaque, and difficult to scale, particularly when applications contain heterogeneous data such as free-text descriptions, scanned documents, and financial records.

This project presents an **AI driven, agentic decision support system** designed to assist in the evaluation of applications for **economic social support** and **economic enablement programs**. The proposed solution integrates classical machine learning, generative AI, and modular agent orchestration to deliver transparent, auditable, and policy aligned decisions.

The system is designed with the following principles in mind:

- Transparency and explainability
- Privacy preserving local execution
- Scalability and maintainability
- Clear separation between decision making and language understanding

2 High-Level Architecture

2.1 Architectural Overview

The solution follows a pipeline based, agentic architecture, where each component (agent) is responsible for a well defined task. Data flows sequentially through the system, enabling traceability, auditability, and modular extensibility.

2.2 Architecture Diagram

Figure 1 illustrates the high-level system architecture and data flow.

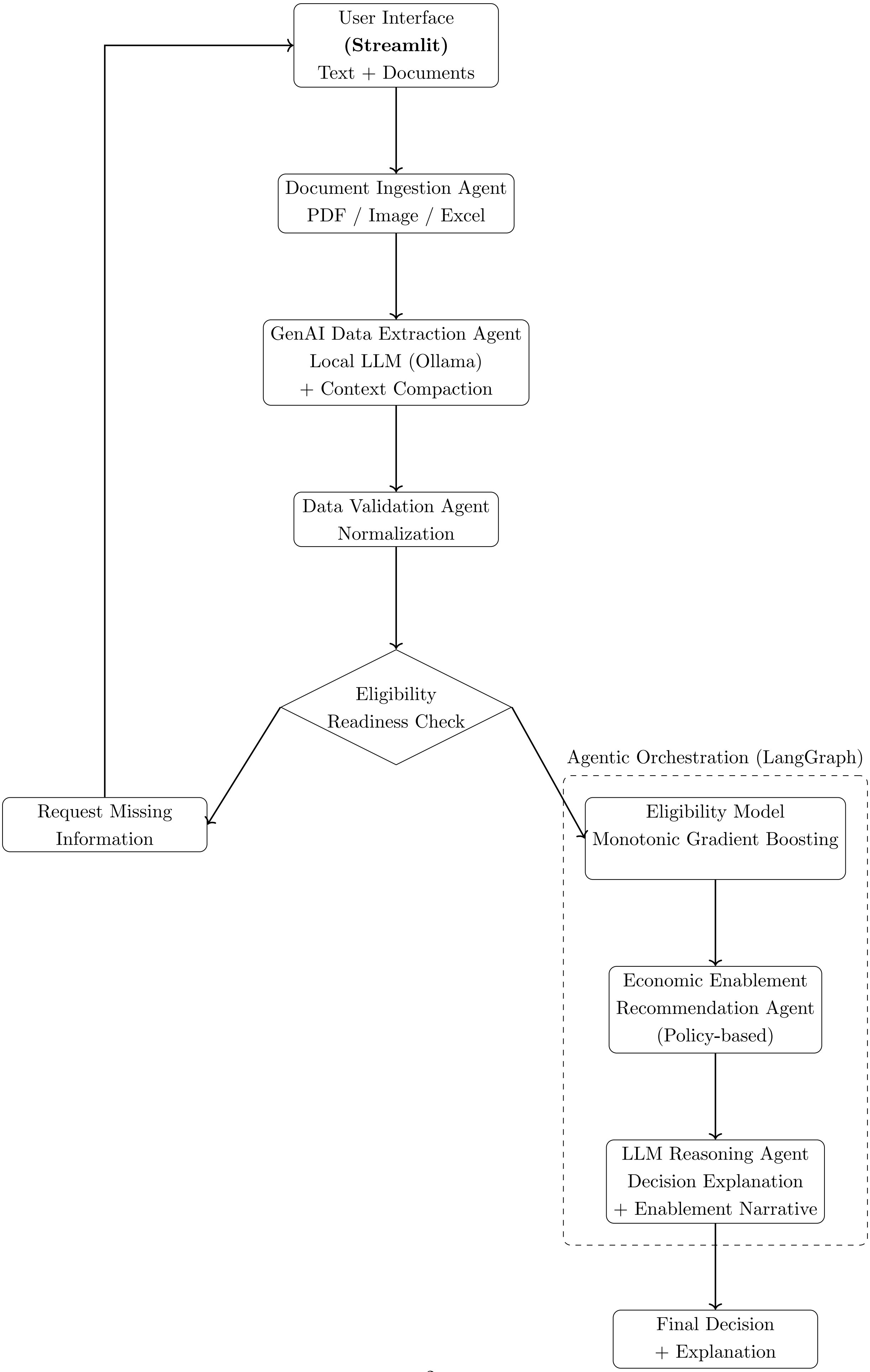

Figure 1: Architecture and data flow of the AI-driven social support eligibility and economic

Note: The diagram shows user interaction, document ingestion, GenAI extraction, validation, eligibility assessment, enablement recommendation, and explanation generation.

2.3 Data Flow Characteristics

- Unidirectional and deterministic data flow
- Clear agent boundaries and responsibilities
- Structured intermediate representations
- Generative AI is not used for final eligibility decisions

3 Tool and Technology Justification

This section justifies the selection of tools and technologies used in the system, considering suitability, scalability, maintainability, performance, security, and alignment with public-sector deployment constraints.

3.1 Programming Language: Python

Python was selected as the primary implementation language due to its strong ecosystem for machine learning, document processing, and natural language processing.

Justification:

- Extensive support for classical ML (scikit-learn) and explainability tools
- Mature libraries for PDF, image, and tabular data processing
- Rapid prototyping and readability, enabling maintainable codebases
- Widely adopted in government and enterprise analytics systems

3.2 User Interface: Streamlit

Streamlit was used to build the interactive application interface.

Suitability:

- Designed for data-driven and decision-support applications
- Native support for file uploads (PDF, images, Excel)
- Natural conversational interaction via chat-style UI

Scalability and Maintainability:

- Stateless execution model with explicit session management
- Clear separation between UI logic and backend agents
- Easy migration to containerized or internal web environments

Security Considerations:

- Runs locally without external network calls
- Suitable for deployment within secure government networks

3.3 Document Processing Tools

Handling heterogeneous document formats is critical, as real-world applications frequently include scanned IDs, bank statements, and spreadsheets.

Tool	Justification
PyMuPDF	Efficient and accurate extraction of text from digitally generated PDFs. Selected over alternatives (e.g., pdfminer) due to superior performance, layout handling, and robustness for multipage documents.
pytesseract (OCR)	Enables optical character recognition for scanned documents and photographs. Necessary for handling real-world cases where PDFs are image-based rather than text-based.
Pillow (PIL)	Used for image preprocessing prior to OCR (resizing, normalization). Improves OCR accuracy and robustness across image qualities.
openpyxl / pandas	Enables structured parsing of Excel files containing financial data. Chosen for reliability, schema preservation, and ease of integration with downstream validation logic.

Table 1: Justification of document processing tools.

Rationale for Multi-Tool Approach: No single extraction method is sufficient for all document types. The system explicitly routes documents to the appropriate extraction pipeline based on format, ensuring accuracy and reducing inference errors.

3.4 Generative AI: Local LLM via Ollama

A locally hosted Large Language Model (LLM) is used exclusively for language understanding tasks, not decision making.

Roles of the LLM:

- Schema constrained extraction of structured fields from unstructured text
- Identification of missing or incomplete information
- Generation of natural language explanations for decisions

Justification for Local Deployment:

- No external API calls or data leakage
- No dependency on cloud services or API keys
- Suitable for sensitive personal and financial data

3.5 Machine Learning Model: Random Forest Classifier

The eligibility decision is performed using a Random Forest classifier trained on policy aligned synthetic data.

Why Random Forest:

- Handles heterogeneous numerical and categorical features
- Captures non-linear relationships without heavy feature engineering
- Robust to noise and partial missing data
- Provides feature importance metrics for explainability

Why Not Deep Learning:

- Deep models are harder to audit and explain
- Higher computational cost with limited benefit for tabular data
- Reduced transparency for public-sector decision-making

Why Not Rule-Based Only:

- Hard-coded rules lack adaptability
- Difficult to maintain as policies evolve
- Poor handling of interacting variables

The Random Forest model provides a balance between interpretability, performance, and policy flexibility.

3.6 Agentic Orchestration Design

The system is implemented as a set of modular agents rather than a monolithic pipeline.

Benefits of Agent-Based Design:

- Clear separation of responsibilities
- Easier testing and debugging
- Independent evolution of components
- Improved traceability for audits

Each agent operates on structured state, ensuring deterministic behavior and reproducibility.

4 Design Rationale: Separation of Extraction, Decision, and Explanation

The proposed system follows a deliberate separation of responsibilities between information extraction, eligibility decision-making, and user-facing explanation and guidance. This design ensures automation efficiency while maintaining fairness, transparency, and regulatory compliance.

4.1 Information Extraction and Preprocessing

The initial stages of the pipeline focus exclusively on transforming unstructured applicant inputs into structured, machine-readable information. User-provided text and documents are ingested and processed by a GenAI-based extraction agent operating under a strict, schema-constrained prompt. The role of the LLM at this stage is limited to factual extraction only; it does not perform inference, reasoning, or decision-making.

Deterministic preprocessing and validation steps normalize numeric values, standardize categorical attributes, and preserve missing information as explicit `null` values. This ensures that downstream models receive clean and auditable inputs while avoiding implicit assumptions or silent imputations.

4.2 Eligibility Decision via Deterministic Machine Learning

Eligibility assessment is performed by a policy-aligned Monotonic Gradient Boosting model. The model consumes validated features such as income, income per capita, employment history, family size, and net worth. Monotonic constraints enforce intuitive and fair relationships between features and eligibility outcomes, improving interpretability and compliance with policy expectations.

The eligibility model produces a discrete outcome—`APPROVE`, `SOFT_DECLINE`, or `REJECT`—along with a confidence score and explicit decision signals. Importantly, this stage is fully deterministic at inference time and does not involve any generative components.

4.3 LLM-Based Explanation and Enablement Guidance

Following the eligibility decision, a separate LLM reasoning agent is used solely for communication and guidance. This agent receives the final eligibility outcome, transparent model signals, and a compacted representation of applicant-provided context. It generates a human-readable explanation that reflects the model's logic without re-evaluating or modifying the decision.

When applicable, the reasoning agent also provides economic enablement recommendations such as job matching, training, or career counseling. These recommendations are grounded in validated applicant attributes and contextual information and are strictly decoupled from the eligibility decision itself.

4.4 Benefits of the Layered Design

This layered approach offers several advantages:

- **Safety and Fairness:** Eligibility decisions are made by a deterministic, policy-aligned ML model rather than a generative system.
- **Auditability:** Each stage produces explicit, inspectable outputs, enabling end-to-end traceability.
- **Explainability:** LLM-generated explanations are grounded in model signals, preventing hallucinated justifications.
- **Scalability:** Modular agents allow individual components to evolve independently without compromising system integrity.

Overall, the design leverages the strengths of both traditional machine learning and large language models while mitigating their respective risks, making it suitable for high-stakes, government-operated social support systems.

5 Alignment with Problem Requirements

The proposed solution satisfies all stated requirements:

- Assessment based on income, employment history, family size, wealth, and demographics
- Approval, soft decline, or rejection outcomes
- Economic enablement recommendations
- Modular GenAI and ML agents

6 Security, Privacy, and Compliance

The system is designed for high security environments:

- Fully local execution
- No external APIs
- No environment variables or secrets
- Auditable decision pipeline

7 Future Improvements

While the proposed system demonstrates effective automation and explainability for social support assessment, several extensions can further improve its robustness, scalability, and decision quality.

7.1 Technical Enhancements

Future work will focus on strengthening the decision-making core through additional safeguards and learning capabilities:

- **Fairness and Bias Auditing:** Incorporating systematic bias audits across demographic subgroups to detect and mitigate unintended disparities in eligibility outcomes.
- **Hybrid Policy–ML Decisioning:** Integrating explicit policy rule engines alongside the ML model to handle exceptional cases, regulatory constraints, and hard eligibility boundaries.
- **Continuous Learning Pipelines:** Enabling periodic retraining of the eligibility model using anonymized historical decisions and outcomes, subject to governance and approval workflows.

7.2 Feature Engineering from User Inputs

The current eligibility model relies on a compact set of policy-aligned features. Future iterations can significantly enhance decision quality by deriving richer features from user-provided text and documents:

- Extraction of skill profiles, domain expertise, and employment gaps from resumes and work history.
- Temporal features such as employment continuity, income stability trends, and duration of unemployment.
- Household-level indicators derived from multi-document consistency (e.g., recurring expenses, dependents across records).

These features can be incorporated into the ML model in a controlled manner to improve predictive performance while maintaining interpretability and fairness constraints.

7.3 Explainability and Reasoning Efficiency

The explainability layer can be further improved in terms of both efficiency and depth:

- **Context-Aware Summarization:** Enhancing context compaction techniques to retain salient applicant information while minimizing token usage for LLM reasoning.
- **Cached and Template-Guided Explanations:** Reusing explanation templates for common decision patterns to reduce latency and improve consistency.
- **Structured Explanation Outputs:** Generating explanations in partially structured formats to facilitate auditing, translation, and downstream analytics.

7.4 System Integration and Deployment

To support real-world adoption, the system can be extended with deeper institutional integration:

- RESTful APIs for application submission, decision retrieval, and explanation delivery.
- Secure integration with government registries, employment databases, and social welfare systems to reduce manual document submission.
- Role-based access controls and audit dashboards for case officers and supervisors.

7.5 Scalable Data Pipelines

At scale, additional data infrastructure enhancements can be introduced:

- Batch processing modes for large-scale social programs and retrospective eligibility reviews.
- Event-driven architectures to support real-time updates, document uploads, and user interactions.
- Monitoring pipelines for model drift, data quality degradation, and explanation consistency.

Overall, these enhancements aim to evolve the system into a continuously improving, policy-compliant decision platform while preserving transparency, efficiency, and public trust.

8 Conclusion

This project demonstrates a practical, explainable, and scalable AI system for social support eligibility assessment. By separating decision-making from language understanding and structuring the solution around modular agents, the system achieves transparency, policy alignment, and extensibility suitable for real-world deployment.