

AI-Powered Loan Eligibility Advisor System

Overview

This project aims to develop an intelligent advisory system that predicts loan eligibility using machine learning, explains decisions using SHAP, interacts with users via a chatbot, and generates detailed PDF reports. The system is designed to empower users with transparency, clarity, and actionable insights.

This project focuses on creating an intelligent advisory system that predicts loan eligibility through machine learning. It provides clear, actionable insights and transparency to users by explaining decisions using SHAP, interacting via a chatbot, and generating comprehensive PDF reports.

Objectives

- Predict loan eligibility based on applicant data
 - Provide explainable AI outputs using SHAP
 - Enable user interaction through a context-aware chatbot
 - Generate downloadable PDF reports summarizing predictions
 - Deploy a user-friendly web interface using Streamlit
-

Technology Stack

Component	Tools & Libraries
Programming Language	Python
ML Models	Logistic Regression, Random Forest, XGBoost
Explainability	SHAP
Chatbot	Rasa or Langchain
UI Framework	Streamlit
PDF Generation	FPDF or ReportLab

Project Structure

```
loan_advisor/
├── data/           # Raw and processed datasets
├── models/         # Trained ML models
├── preprocessing/  # Data cleaning and transformation scripts
├── explainability/ # SHAP integration and plots
├── chatbot/        # Intent recognition and dialogue management
├── reports/        # PDF generation scripts
├── app/            # Streamlit UI components
├── utils/          # Helper functions and config files
└── README.md       # Project overview and setup instructions
```

Development Roadmap

1. Requirements Definition

- Identify user personas and use cases
- Finalize feature set and success metrics

2. Data Preparation

- Source dataset (e.g., Kaggle Loan Prediction)
- Handle missing values, encode categorical features
- Normalize and split data into train/test sets

3. Model Development

- Train multiple models: Logistic Regression, Random Forest, XGBoost
- Evaluate using accuracy, precision, recall, ROC-AUC
- Select optimal model for deployment

4. Explainability Integration

- Use SHAP to visualize feature importance
- Generate individual prediction explanations
- Embed SHAP plots in UI and reports

5. Chatbot Implementation

- Design intents: eligibility check, FAQs, document guidance
- Train and deploy chatbot using Rasa or Langchain
- Connect chatbot to prediction backend

6. PDF Report Generation

- Create structured reports with user inputs, prediction results, and SHAP insights
- Enable download via Streamlit interface

7. Web App Development

- Build interactive UI with Streamlit
- Include input forms, chatbot window, prediction output, SHAP plots, and download button
- Ensure responsive design and error handling

8. Testing & Validation

- Perform unit and integration testing
- Simulate user scenarios for feedback
- Refine chatbot responses and UI layout

9. Documentation & Portfolio Integration

- Write README with setup guide, screenshots, and demo links
- Document model performance and explainability
- Add project to GitHub with modular code and clear comments

10. Final Deployment

- Host application on Streamlit Cloud or Render
- Share demo video and walkthrough
- Collect feedback for future improvements

Deliverables

- Trained ML model with SHAP explainability
- Functional chatbot integrated with prediction engine
- Streamlit-based web application
- Downloadable PDF reports
- Complete GitHub repository with documentation