1. Project Overview

- **Objective**: The main objective of this project is to develop a web application that predicts whether a loan application will be approved or rejected based on various applicant attributes like income, loan amount, credit history, and gender.
- **Tools Used**:
    - **Python**: The programming language used to develop the application.
    - **Streamlit**: A library for creating web applications with Python.
    - **scikit-learn**: A machine learning library used to train the prediction model.
    - **pandas**: A library for data manipulation and analysis.

2. Data Collection and Preprocessing

- **Dataset**: The project uses a dataset named train.csv, which includes relevant features for predicting loan eligibility.
- **Data Features**:
    - ApplicantIncome: The income of the applicant.
    - LoanAmount: The amount of loan requested by the applicant.
    - Credit_History: Indicates the credit history of the applicant (1 for good and 0 for bad).
    - Gender: Gender of the applicant (Male or Female).
    - Loan_Status: Indicates if the loan was approved (Y for Yes, N for No).
- **Data Preprocessing Steps**:
    - Load the dataset using pandas.
    - Select relevant columns for prediction.
    - Convert categorical variables (like Gender and Loan_Status) to numeric values for model training.
    - Drop any missing values to ensure clean data for training.

3. Model Training

- **Model Selection**: The Random Forest classifier is chosen for this project due to its effectiveness in classification tasks and ability to handle non-linear data.
- **Train-Test Split**: The dataset is split into training and testing sets (80% for training and 20% for testing) using train_test_split.
- **Training the Model**: The Random Forest model is trained using the training data (features and target labels).

4. User Interface Development

- **Streamlit Setup**: The user interface is built using Streamlit, which provides an easy way to create interactive web applications.
- **Input Fields**: The UI allows users to input:
    - Applicant Income
    - Loan Amount
    - Credit History (Good or Bad)
    - Gender
- **Prediction Button**: Once the user inputs the necessary data, they can click a "Predict" button to get the loan eligibility prediction.

5. Prediction Process

- **Making Predictions**: When the user clicks the "Predict" button, the application:
    - Calls the trained model with the user inputs.
    - Returns a prediction (Loan Approved or Loan Rejected).
- **Display Results**: The prediction result is displayed on the web app, providing clear feedback to the user.

6. Running the Application

- **Setup Instructions**:
    1. Clone the repository containing the project code.
    2. Navigate to the project directory.
    3. Install the required Python packages using a requirements.txt file.
    4. Run the Streamlit application using the command:

bash

Copy code

streamlit run app.py

7. Future Improvements

- Some potential future enhancements can be made, such as:
    - Improving the model with more features or using different algorithms.
    - Adding data visualization components to help users understand the prediction logic.
    - Collecting user feedback to enhance the application's usability.

8. Conclusion

In conclusion, our Loan Eligibility Prediction project demonstrates a practical application of machine learning in the financial sector. By automating the loan approval process, we can enhance efficiency,

reduce human error, and provide quick responses to applicants, ultimately leading to a better user experience.

Github repository: https://github.com/deepak4194/LoanPrediction/

App Link: http://Loanprediction12.streamlit.app