METASCIFOR TECHNOLOGIES

DATA SCIENCE TRAINING

MAJOR PROJECT REPORT

LOAN PREDICTION SYSTEM

- S. POORNIMA GAYATHRI

1. Introduction

The loan prediction system helps financial institutions decide whether to approve or reject a loan application based on applicant data. This system utilizes two machine learning models: Logistic Regression and Decision Tree Classifier. The system has been deployed as a web application using Flask. The solution aims to automate the decision-making process, improving both accuracy and efficiency while minimizing manual review efforts.

2. Problem Statement

Banks and financial institutions need to assess a borrower's credibility based on certain features. This manual process can be prone to biases and inefficiencies. The loan prediction system automates this process using machine learning, allowing for consistent and rapid loan approval predictions.

3. Dataset Overview

The dataset includes various features that help predict loan approval. Key attributes include:

- **loan_ID**: Unique Loan ID (not used in the model).
- **No of dependents**: number of dependents on the applicant.
- **Education**: Graduate or Not Graduate.
- **Self_Employed**: Whether the applicant is self-employed or not
- **Income_annum**: Annual income of the applicant.
- **Loan_amount**: Total loan amount.
- **Loan_Term**: Term of loan repayment (in years).
- **Cibil_score**: cibil score of the applicant.
- **Residential_assets_value**: Residential asset value of applicant.
- **Commercial_assets_value**: Commercial asset value of applicant
- **Luxury_assets_value**: luxury asset value of applicant
- **bank_asset_value**: bank asset value of applicant
- **loan_Status**: Target variable, indicating loan approved or rejected.

4. Data Cleaning and Preparation

The raw dataset required several preprocessing steps before being used for model training:

- **Handling Missing Values and Duplicates**: Missing data in columns are checked and there are no missing data and duplicated records.

- **Encoding Categorical Features**: Categorical variables (e.g., Education, Self_Employed) are converted into numerical representations using
    - Label Encoding.
- **Feature Scaling**: Since the range of numeric features like Applicant's annual Income and Loan Amount varied significantly, standardization or normalization was applied to bring the features to a comparable scale.

5. Exploratory Data Analysis (EDA)

Distribution of various features and relation between them were explored. For example:

- Annual Income distribution and presence of any outliers
- Cibil Score vs Education and Self-employed
- Applicants with a higher cibil score had higher chances of loan approval.
- Cibil score had higher correlation with loan approval status

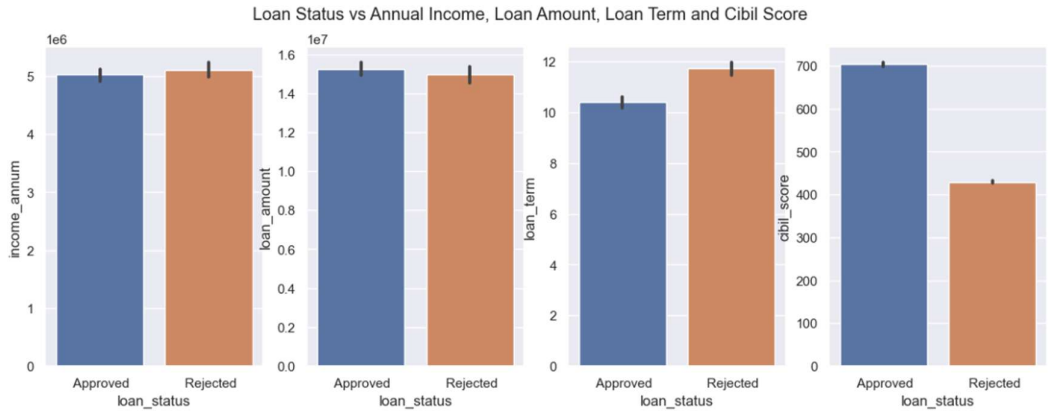

Fig. Loan Status vs various features

6. Model Development

Two machine learning models were trained on the pre-processed data: Logistic Regression and Decision Tree Classifier.

Logistic Regression

- **Objective**: Predict the probability that an applicant's loan will be approved.
- **Implementation**: The logistic regression model calculates the probability of the binary outcome using the sigmoid function.

Decision Tree Classifier

- **Objective**: Classify whether a loan will be approved or rejected by learning decision rules inferred from the data.
- **Implementation**: A tree structure was created where internal nodes represent conditions on features, and leaf nodes represent the predicted outcome (approved/rejected).

7. Model Training and Evaluation

- **Train-Test Split**: The dataset was split into 80% training and 20% test sets to evaluate model performance.
- **Metrics**: Both models were evaluated using the following metrics:
    - **Accuracy**: The percentage of correct predictions.
    - **Precision**: The proportion of positive identifications that were correct.
    - **Recall**: The proportion of actual positives that were identified correctly.
    - **F1-Score**: The harmonic mean of precision and recall, useful for imbalanced datasets.

Classifier	Accuracy	Precision	Recall	F1 score
Logistic Regression	92%	92%	92%	0.92
Decision Tree Classifier	98%	98%	98%	98%

The Decision Tree Classifier outperformed the Logistic Regression model in terms of both accuracy and generalization on the test set, making it the preferred model for this project.

8. Model Deployment with Flask

The Flask web application was developed to allow users to interact with the loan prediction system through a user-friendly interface.

Steps to Deploy

1. **Flask Setup**: Flask was used as the backend framework to host the model and handle user requests.
2. **Input Form**: An HTML form was created to collect input from users, such as income, cibil score and other details.
3. **Prediction Endpoint**: Flask receives user inputs, processes them, and passes them to the trained model for prediction.

4. **Output**: The prediction (loan approved or rejected) is displayed to the user in a readable format.

9. User Interface and Interaction

- **Front-End**: Simple and intuitive web interface using HTML, allowing users to input their data.
- **User Flow**:
    - Users enter relevant loan details such as income, loan amount, and credit history.
    - The system processes the input and displays a prediction result. Loan is viable. Can be approved/loan is not viable. Can be rejected.

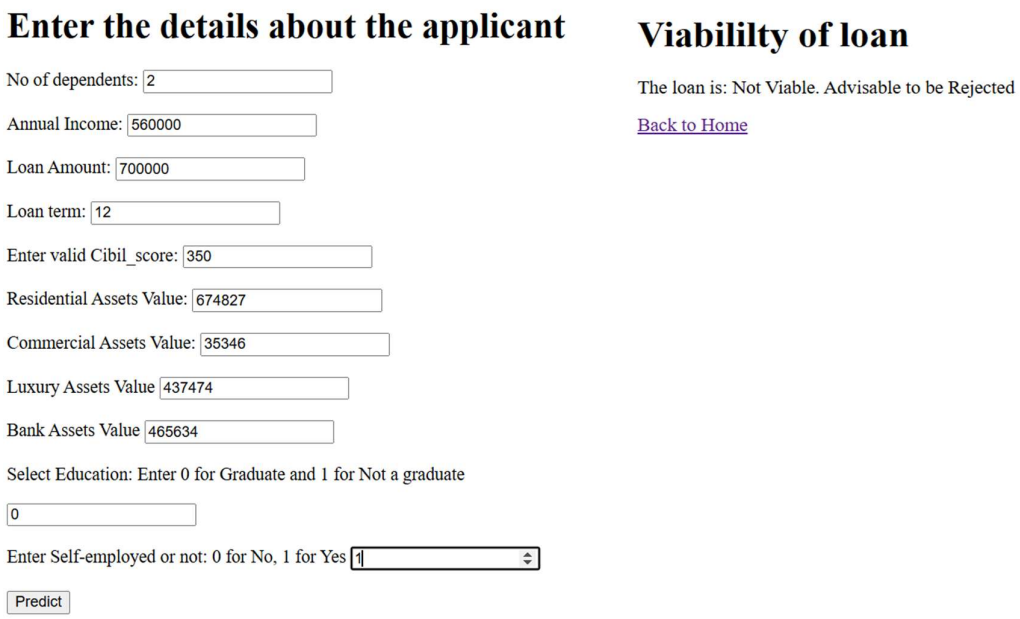

Fig. User Interface

10. Challenges and Future Work

- **Imbalanced Dataset**: The target variable (loan approval) was slightly imbalanced, as more loans were approved than rejected. Handling this imbalance effectively could improve model performance.
- **Feature Engineering**: Additional domain-specific features (e.g., employment stability, detailed credit score) could be added to improve the prediction accuracy.

- **Model Optimization**: Techniques like ensemble methods can be explored to boost the model's performance further.

11. Conclusion

The loan prediction system provides an efficient, automated solution for financial institutions to evaluate loan applications. Decision Tree Classifier proved to be the more reliable model based on evaluation metrics, and the system was successfully deployed using Flask for real-time prediction. The system can be further enhanced by incorporating more features, refining the models, and improving deployment strategies.