Project Title Loan Approval Prediction Using Machine Learning

Chapter 1 Project Overview

This project aims to build a predictive model to automate the loan approval process. By analyzing historical data of loan applicants, the model can learn the factors that influence whether a loan is approved or rejected. Implementing this system can help financial institutions:

- Make faster and more consistent decisions
- Reduce human errors
- Streamline operational processes

The project serves as a proof-of-concept to demonstrate how machine learning can assist in financial decision-making.

Objectives

The main objectives of this project are:

Understand the characteristics and patterns in the dataset.

Handle missing values and encode categorical variables to make the data suitable for machine learning.

Train multiple machine learning models for loan approval classification and evaluate their performance.

Identify the model that performs best based on key performance metrics.

Develop a simple, user-friendly interface using Streamlit to demonstrate the model in action.

Scope

Scope of this project are import and explore the dataset, Clean data, handle missing values, and encode categorical features, Analyze the dataset to identify patterns and trends, train different machine learning models such as Logistic Regression and Random Forest Classifier, Compare models using performance metrics to select the best one and deploy the best model through a simple Streamlit application.

Chapter 2 Dataset Description

The dataset used in this project is obtained from an Excel file .It contains 614 entries and 13 columns, providing detailed information about loan applicants.

The dataset includes the following features:

Feature	Description
Loan_ID	Unique identifier for each loan application.
Gender	Gender of the applicant (Male/Female).
Married	Marital status of the applicant (Yes/No).
Dependents	Number of dependents.
Education	Education level of the applicant (Graduate/Not Graduate).
Self_Employed	Self-employment status (Yes/No).
ApplicantIncome	Applicant’s monthly income.
CoapplicantIncome	Co-applicant’s monthly income.
LoanAmount	Loan amount requested (in thousands).
Loan_Amount_Term	Term/duration of the loan (in months).
Credit_History	Whether credit history meets guidelines (1.0 = yes, 0.0 = no).
Property_Area	Area where the property is located (Urban/Rural/Semiurban)
Loan_Status	Outcome of the loan application: ‘Y’ = approved, ‘N’ = rejected.

Initial Data Inspection

- Size: 614 rows × 13 columns.
- Missing Values: Some columns have missing values, including Gender, Married, Dependents, Self_Employed, LoanAmount, Loan_Amount_Term, and Credit_History. These missing values will be addressed during preprocessing.
- Data Types: The dataset contains a mix of numerical (int64, float64) and categorical (object) features.

Chapter 3 Data Preprocessing

Raw data contains inconsistencies, missing values, and features that are not directly usable by machine learning algorithms. Proper preprocessing ensures that the data is clean, consistent, and optimized for model training, which improves accuracy and stability. In this project, the following preprocessing steps were carried out:

3.1 Handling Missing Values

Missing data is common in real-world datasets. If not addressed properly, it can introduce bias or reduce model performance. In this dataset, missing values were found in both numerical and categorical columns:

Numerical Columns are LoanAmount, Loan_Amount_Term, Credit_History.LoanAmount and Loan_Amount_Term were imputed using the median value because these features are continuous and may contain outliers. Median imputation is robust to outliers.

Credit_History was imputed with the mode (most frequent value), as it is a binary indicator (1.0 or 0.0).

Categorical Columns are Gender, Married, Dependents, Self_Employed. Proper handling of missing values ensures that no important patterns are lost and the model does not produce biased predictions due to missing information.

3.2 Feature Engineering

Feature engineering is the process of creating new features or transforming existing ones to improve model performance. In this project:

Total Income: Created by combining ApplicantIncome and CoapplicantIncome.

Formula: TotalIncome = ApplicantIncome + CoapplicantIncome

Log Transformation: Applied to LoanAmount and TotalIncome. 3.3 Encoding Categorical Variables

Machine learning algorithms require numerical inputs. Therefore, categorical variables were converted into numerical format:

Encoding column are Gender, Married, Education, Self_Employed, property Area, Dependents. Label Encoding assigns an integer to each category (e.g., Male = 0, Female = 1).

One-Hot Encoding Creates separate binary columns for each category in features like Property_Area (Urban, Semiurban, Rural).Loan_Status was encoded as 1 = Approved (Y) and 0= Rejected (N).

3.4 Scaling Numerical Features

Scaling ensures that features with larger numeric ranges do not dominate the model training process.

Selected column are ApplicantIncome, CoapplicantIncome, LoanAmount, Loan_Amount_Term, TotalIncome

StandardScaler is used to transforms features to have a mean of 0 and standard deviation of 1.

Chapter 4 Exploratory Data Analysis (EDA)

Exploratory Data Analysis (EDA) is the process of visualizing and summarizing data to identify hidden patterns, and detect potential issues before model building. It helps in understanding relationships between features and the target variable (Loan_Status), which guides feature selection and preprocessing.

In this project, EDA was performed in three major stages Distribution Analysis, Categorical Analysis, and Relationship with Target Variable.

4.1 Distribution Analysis of Numerical Features

Numerical features such as ApplicantIncome, CoapplicantIncome, LoanAmount, and Loan_Amount_Term were analyzed using histograms, density plots, and boxplots**.**

ApplicantIncome is right-skewed distribution. Most applicants fall within a low-to-medium income range, but some applicants earn significantly higher (outliers).A logarithmic transformation was applied to reduce skewness.

CoapplicantIncome are clustered around the lower range. A significant number of applicants reported zero co-applicant income.

LoanAmount is right-skewed. Most loans are small-to-moderate, but a few large loan requests exist, creating long tails. A logarithmic transformation was applied to normalize distribution.

In Loan_Amount_Term most loans are for 360 months (30 years)**,** showing a standard preference among applicants. Smaller terms (like 120 months or 180 months) are less common.

Normalization of skewed features (ApplicantIncome, LoanAmount, TotalIncome) ensures that models like Logistic Regression perform better.

4.2 Categorical Feature Analysis

Categorical features such as Gender, Married, Dependents, Education, Self_Employed, Property_Area, and Credit_History were analyzed using bar plots and count plots**.**

In Gender majority of applicants are male**,** but a notable number of females also apply.

In Married most applicants are married, reflecting real-world demographics where families jointly apply for loans.

In Dependents many applicants have zero dependents**,** while fewer have 1–3 dependents.

In Education majority of applicants are graduates**,** but non-graduates are also present in significant numbers.

In Self_Employed most applicants are not self-employed (working in salaried jobs).

In Property_Area applicants are distributed across Urban, Semiurban, and Rural areas, with Semiurban having the highest share.

In Credit_History majority of applicants have a credit history of 1.0 (meeting financial guidelines).

Credit history, marital status, and property area are likely to play a key role in loan approval decisions.

4.3 Relationship between Features and Loan Status

The relationship of independent features with the target variable (Loan_Status) was analyzed to understand which factors strongly influence loan approval.

Gender vs Loan_Status

Approval rates are slightly higher for male applicants, but females also receive a significant proportion of approvals.

Marital Status vs Loan_Status

Married applicants have a higher approval rate compared to unmarried ones, possibly due to combined household income.

Dependents vs Loan_Status

Applicants with fewer dependents tend to have higher approval chances. More dependents often mean higher financial burden, lowering approval likelihood.

Education vs Loan_Status

Graduates generally show higher approval rates compared to non-graduates. Education may be correlated with stable income and repayment ability.

Self_Employed vs Loan_Status

Salaried applicants have higher approval chances compared to self-employed applicants. This may be due to the perceived stability of salaried income.

Credit_History vs Loan_Status

The strongest predictor of loan approval. Applicants with a credit history of 1.0 (good credit) have a very high approval rate, while those with 0.0 have much lower chances.

Property_Area vs Loan_Status

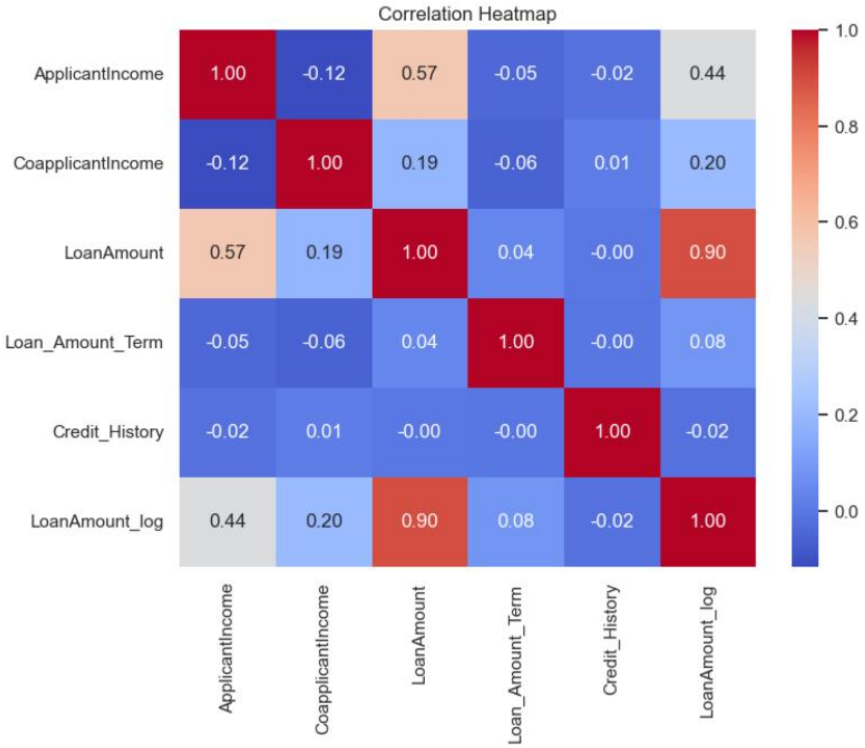
Applicants in Semiurban areas have the highest approval rate, followed by Urban, with Rural areas having the lowest.

Income vs Loan_Status

Higher income applicants generally get loans approved, but extremely high income does not guarantee approval. Coapplicant income has less influence compared to applicant income.

4.4 Correlation Analysis

Correlation analysis measures the strength and direction of a linear relationship between two numerical variables, commonly quantified using the Pearson Correlation Coefficient, which ranges from -1 to +1. A value of +1 indicates a perfect positive correlation, meaning that as one variable increases, the other also increases proportionally, while -1 indicates a perfect negative correlation, where one variable increases as the other decreases. A value near 0 suggests no linear relationship, though non-linear associations may still exist. In datasets with multiple numerical features, a correlation matrix can be computed and visualized using a heatmap**,** where bright colors indicate strong positive correlations, dark colors indicate strong negative correlations, and neutral colors indicate weak or no correlation. In the context of the loan approval project, correlation analysis helps identify relationships such as the expected positive correlation between ApplicantIncome and LoanAmount, or between ApplicantIncome and CoapplicantIncome. It also aids in detecting potential multicollinearity, which can affect models like Logistic Regression, though tree-based models like Random Forest are generally robust. Furthermore, after encoding categorical variables (e.g., Credit_History, Property_Area, Loan_Status), correlations with numerical features or the target variable can be calculated; for instance, a strong positive correlation between Credit_History and Loan_Status numerically confirms that applicants with a good credit history are more likely to have their loans approved.



	ApplicantIncome	CoapplicantIncome	LoanAmount	Loan_Amount_Term	Credit_History	LoanAmount_log
ApplicantIncome	1.00	-0.12	0.57	-0.05	-0.02	0.44
CoapplicantIncome	-0.12	1.00	0.19	-0.06	0.01	0.20
LoanAmount	0.57	0.19	1.00	0.04	-0.00	0.90
Loan_Amount_Term	-0.05	-0.06	0.04	1.00	-0.00	0.08
Credit_History	-0.02	0.01	-0.00	-0.00	1.00	-0.02
LoanAmount_log	0.44	0.20	0.90	0.08	-0.02	1.00

Chapter 5 Model Building

The main goal of this step was to train different machine learning algorithms to classify whether a loan application should be approved or rejected.

For this project, multiple models were selected to ensure both variety and reliability in performance. The chosen algorithms included:

Decision Tree Classifier (DT):
Selected for its simplicity and interpretability. Decision Trees split data based on feature values, making them easy to understand and visualize. They also act as a baseline model for comparison with more advanced techniques.

K-Nearest Neighbors (KNN):
Chosen because of its ability to handle non-linear relationships between features and the target variable. KNN classifies instances based on the similarity to their nearest neighbors in the dataset.

Support Vector Machine (SVM):
Included due to its strong theoretical foundation and ability to work well in complex classification tasks by finding an optimal separating hyperplane between classes.

Random Forest Classifier (RF):
Used as an advanced ensemble learning method that combines multiple decision trees. It was expected to perform better than a single decision tree by reducing overfitting and improving generalization.

Logistic Regression (LR):
Selected as a simple yet powerful linear model, widely used in binary classification problems. Logistic Regression was expected to serve as a strong baseline due to its efficiency and interpretability, which is important for financial applications.

Training Process

The dataset was split into training and testing sets to evaluate model performance fairly on unseen data. A Pipeline was implemented to combine preprocessing steps (such as imputation, encoding, and scaling) with the training of models, ensuring that the workflow was clean, consistent, and reproducible.

For some models, hyperparameters were fine-tuned to improve accuracy. In particular, the Random Forest Classifier was optimized using GridSearchCV, which systematically tested different parameter combinations to find the best configuration.

By training these diverse algorithms, the project ensured that the final model selection was based on both performance and practical suitability for deployment.

Chapter 6 Model Evaluation

Once the models were trained, it was necessary to evaluate their performance on the test dataset. The main purpose of evaluation was to ensure that the models were not just memorizing the training data but could also generalize well to unseen data. For this project, different classifiers such as Logistic Regression, Random Forest, Decision Tree, KNN, and SVM were compared. Among them, Logistic Regression and Random Forest showed the strongest results, so their evaluation was emphasized.

6.1 Evaluation Approach

The models were evaluated using multiple metrics to gain a complete understanding of their predictive performance. A Confusion Matrix and a Classification Report were generated to analyze the classification results in detail. Additionally, the ROC-AUC score was used to measure the discriminative power of the models across various thresholds

6.2 Performance Metrics

Accuracy**:** The proportion of correctly predicted instances out of the total instances.

Precision: The proportion of correctly predicted positive cases (loan approvals) among all predicted positives.

Recall (Sensitivity)**:** The proportion of actual positive cases that were correctly identified by the model. High recall ensures that very few eligible loans are wrongly rejected.

F1-Score**:** The harmonic mean of precision and recall. Balances the trade-off between precision and recall, which is useful when both metrics are important.

Confusion Matrix: Summarizes classification results by showing:

True Positives (TP): Correctly predicted approved loans.

True Negatives (TN): Correctly predicted rejected loans.

False Positives (FP): Incorrectly approved rejected loans.

False Negatives (FN): Incorrectly rejected approved loans.

Provides a detailed picture of how errors are distributed.

ROC-AUC (Receiver Operating Characteristic – Area Under Curve)

Measures the model's ability to distinguish between approved and not approved loans across different thresholds.

Higher ROC-AUC indicates stronger discriminative power, which is crucial in binary classification tasks.

6.3 Model Comparison and Results

Decision Tree**:** Accuracy of 68.29%**.** Simple and interpretable but prone to overfitting, leading to weaker performance on unseen data.

K-Nearest Neighbors (KNN)**:** Accuracy of 73.17%**.** Performed better than Decision Tree, but sensitive to noisy data and less effective for scaling.

Support Vector Machine (SVM)**:** Accuracy of 65.04%**.** Lowest among all tested models, struggled to handle categorical variables and dataset characteristics.

Random Forest: Accuracy of 77.24**%**. Strong performance due to ensemble learning, reduced overfitting compared to a single Decision Tree.

Logistic Regression**:** Accuracy of 78.86%**.** Outperformed all other models. Despite being simple, it captured important patterns effectively and provided interpretable results.

6.4 Final Model Selection

Although Random Forest was a strong candidate, Logistic Regression was selected as the final model for deployment. The main reasons are:

It achieved the highest accuracy (78.86%) among all tested models.

It provided interpretable coefficients, making it easier to explain the influence of features like income, credit history, and loan amount.

Logistic Regression is computationally efficient and simple to implement in applications such as loan approval systems.

It balances performance and interpretability, which is critical in financial decision-making where transparency is required.

Thus, Logistic Regression was chosen as the most suitable model for this project and deployed in the Streamlit application for loan approval prediction.

Chapter 7 Deployment

The trained Logistic Regression model was deployed using Streamlit, creating a simple and interactive web application. Users can input applicant details such as income, credit history, education, marital status, and loan amount, and the app predicts whether a loan will be approved or rejected**.** The application provides real-time results and visual feedback, making it practical for non-technical users. Streamlit allows easy deployment on localhost and can be extended for cloud-based use in the future.

Conclusion

This project successfully built a loan approval prediction system using Logistic Regression. After data preprocessing, EDA, and model evaluation, Logistic Regression achieved the highest accuracy of 78.86% and provided interpretable results. The model was deployed using Streamlit, offering a user-friendly interface for real-time predictions. Overall, the system demonstrates how machine learning can streamline loan approvals, reduce human bias, and provide actionable insights. Future improvements could include more data, advanced models, and cloud deployment for broader accessibility.