Loan Default Prediction

Using Decision Tree Classifier

Group Members and Supervisor Details

Group Members:

1. Shanmukh Ram Reddy Notuu – AP22110010731
2. Rohith Gunuputi - AP22110010749
3. Nikhileswar Kotha - AP22110010757

Supervisor Name:

- **Class Teacher:** Dr Sibendu Samanta

Table of Contents

S.no	*Name*	*Page No*
1.	Group Members and Supervisor Detail	1
2.	Table of Contents	2
3.	Abstract	3
4.	Existing System and Proposed System	4
5.	Theory Behind the Project	8
6.	Algorithm	11
7.	Flowchart	14
8.	Program	15
9.	Results	21
10.	Discussion	23
11.	References	25

Abstract

Loan default is a significant concern for financial institutions as it directly impacts profitability and risk management. In the current era of digital transformation, banks and lending organizations leverage machine learning to improve credit risk assessment. This project develops a predictive model for loan defaults using a Decision Tree Classifier, a supervised machine learning algorithm.

The dataset includes borrower features such as age, income, loan amount, credit score, and employment details, aiming to identify patterns contributing to loan default for accurate predictions. By preprocessing the data, handling missing values, and encoding categorical variables, the dataset is prepared for model training.

The Decision Tree Classifier was chosen for its simplicity, interpretability, and ability to handle both numerical and categorical data. The model was trained and tested on a split dataset, and its performance evaluated using metrics like accuracy, F1 score, precision, recall, and ROC AUC. These metrics offer a comprehensive view of the model's predictive effectiveness.

Using this model, financial institutions can automate risk evaluation, minimize human error, and improve decision-making efficiency. Beyond predictions, the system highlights key factors influencing default, providing valuable insights for risk management. This project showcases machine learning's potential to transform credit risk assessment, enabling more reliable and scalable financial solutions.

Existing System and Proposed System

Existing System

The traditional approach to loan default prediction relies heavily on manual assessments and rule-based systems.

Characteristics:

1. **Manual Assessment:**
    - Loan officers evaluate applications based on income, employment, and credit scores.
    - Subjective human judgment plays a key role.
2. **Rule-Based Systems:**
    - Fixed thresholds for parameters like credit score and debt-to-income (DTI) ratio.
    - Rigid criteria limit flexibility.
3. **Historical Trends:**
    - Decisions rely on past trends, struggling to adapt to new patterns.

Limitations:

1. **Inefficiency:**
    - Manual evaluations are slow, delaying loan processing.
2. **Lack of Predictive Power:**
    - Rule-based systems overlook complex data interactions.
3. **Bias and Subjectivity:**
    - Human decisions may introduce inconsistencies and bias.
4. **High Error Risk:**

- Manual processes are prone to oversights.

5. **Limited Scalability:**
    - Struggles to handle large volumes of applications.

Proposed System

A machine learning-based Loan Default Prediction System using a Decision Tree Classifier addresses these limitations.

Features:

1. **Automated Decision-Making:**
    - Reduces human involvement and errors.
2. **Data-Driven Insights:**
    - Identifies complex patterns in borrower data.
3. **Improved Accuracy:**
    - Leverages historical data for reliable risk evaluation.
4. **Scalability:**
    - Processes large datasets efficiently.
5. **Dynamic Learning:**
    - Adapts to new data and changing economic conditions.

Steps:

1. **Data Preprocessing:**
    - Handle missing values and encode categorical variables.
2. **Model Training:**
    - Train the Decision Tree on labeled data.
3. **Prediction:**

- Predict default likelihood for new applications.

4. **Evaluation:**
   - Use accuracy, precision, recall, and ROC AUC metrics.

Advantages:

1. **Efficiency:**
   - Speeds up loan processing, reducing costs.
2. **Objectivity:**
   - Eliminates human bias.
3. **Higher Predictive Power:**
   - Captures complex borrower data relationships.
4. **Flexibility:**
   - Easily retrainable as more data becomes available.

Comparison of Existing vs. Proposed System

Aspect	**Existing System**	**Proposed System**
Decision Process	Manual or rule-based	Automated using machine learning
Efficiency	Low	High
Accuracy	Limited to predefined rules	Improved through data-driven insights
Bias	Prone to human biases	Objective and data-driven
Scalability	Limited	Highly scalable

Aspect	Existing System	Proposed System
Adaptability	Static rules	Dynamic, retrainable model
Processing Time	Slow	Fast

Conclusion:

The proposed system offers a more robust, efficient, and scalable solution, enabling financial institutions to better predict and manage loan default risks.

Theory Behind the Project

This project uses **Machine Learning (ML)** to predict loan defaults. ML enables computers to learn from past data and make decisions without explicit programming. The project focuses on a **Decision Tree Classifier**, a simple yet effective tool for classification tasks.

1. Machine Learning Basics

Machine Learning is of three types:

- **Supervised Learning**: Learns from labeled data (used here).
- **Unsupervised Learning**: Finds hidden patterns in data.
- **Reinforcement Learning**: Learns through trial and error.

2. Decision Tree Classifier

A Decision Tree is a flowchart-like structure.

- **Root Node**: Represents the entire dataset.
- **Internal Nodes**: Ask questions based on data features.
- **Leaf Nodes**: Provide the final decision (Default/No Default). The tree splits data based on important features, ensuring accurate predictions.

Advantages

- Easy to understand and interpret.
- Handles both numerical and categorical data.

Limitations

- Prone to overfitting if not properly controlled.

3. Data Preparation

- **Missing Values**: Replaced with median for numbers and mode for categories.
- **Label Encoding**: Converts categorical data into numeric form.
- **Feature Selection**: Drops unnecessary columns like LoanID.

4. Model Evaluation Metrics

- **Accuracy**: Overall correctness of predictions.
- **Precision**: Correct positive predictions out of all predicted positives.
- **Recall**: Correct positive predictions out of all actual positives.
- **F1 Score**: Balances precision and recall.
- **ROC AUC**: Measures the model's ability to distinguish between classes.

5. Confusion Matrix

A Confusion Matrix shows prediction performance:

Predicted\Actual	**No Default**	**Default**
No Default	Correct (TN)	Wrong (FP)
Default	Wrong (FN)	Correct (TP)

6. Why Predict Loan Defaults?

Loan default occurs when a borrower fails to repay. Early prediction helps banks:

- **Minimize Risk**: Avoid risky loans.
- **Save Time**: Automate decision-making.
- **Improve Accuracy**: Data-driven predictions are more reliable.

Key factors influencing default include credit score, income, loan amount, and employment stability. This project provides an efficient way to predict and manage loan risks.

Algorithm

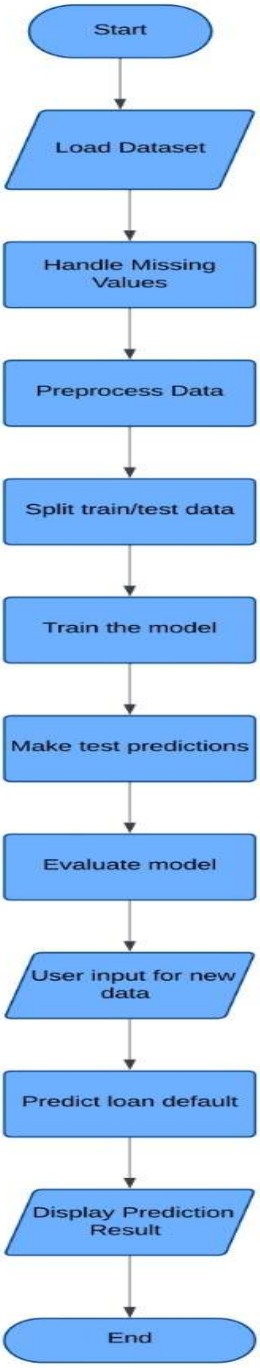
1. Start
2. Data Collection
3. Import the dataset (e.g., 'Loan_default.csv').
4. Data Preprocessing
5. Handle missing values:
    a. For numerical columns: Replace missing values with the median of the column.
    b. For categorical columns: Replace missing values with the mode (most frequent value) of the column.
6. Drop the unnecessary column, LoanID, as it does not contribute to the prediction.
7. Separate the features (X) and the target variable (y), where:
    a. X: Features (all columns except 'Default').
    b. y: Target variable ('Default').
8. Label Encoding
9. For each categorical column in the dataset, apply Label Encoding to convert text values into numeric values.
10. Split Data into Training and Testing Sets
11. Use train_test_split to divide the dataset into training (70%) and testing (30%) sets.
12. Model Training
13. Initialize the Decision Tree Classifier with a defined max_depth (e.g., 5) to control the tree's complexity.

14. Train the model using the training data (X_train and y_train).
15. Model Prediction
16. Use the trained model to predict loan defaults on the testing data (X_test).
17. Evaluate Model Performance
18. Calculate and print the following evaluation metrics:
    a. Accuracy: Proportion of correct predictions.
    b. Precision: Correct positive predictions divided by the total predicted positives.
    c. Recall: Correct positive predictions divided by actual positives.
    d. F1 Score: Harmonic mean of precision and recall.
    e. ROC AUC: Area under the Receiver Operating Characteristic curve.
    f. Confusion Matrix: Displays the counts of true positives, true negatives, false positives, and false negatives.
19. Visualize Results
20. Plot the Confusion Matrix as a heatmap.
21. Plot the ROC Curve to visualize the model's performance.
22. Prediction Function for User Input
23. Create a function to allow user input (e.g., Age, Income, Loan Amount, etc.).
24. Encode the user inputs using the same label encoders as used for training.

25. Predict whether the loan will default or not using the trained Decision Tree model.

26. End

Flowchart


Start
Load Dataset
Handle Missing Values
Preprocess Data
Split train/test data
Train the model
Make test predictions
Evaluate model
User input for new data
Predict loan default
Display Prediction Result
End

Program

```
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import LabelEncoder
from sklearn.tree import DecisionTreeClassifier
from sklearn.metrics import accuracy_score, f1_score, recall_score,
precision_score, confusion_matrix, roc_auc_score, roc_curve
import matplotlib.pyplot as plt
import seaborn as sns

data = pd.read_csv('Loan_default.csv')
data.fillna(data.median(numeric_only=True), inplace=True)
for column in data.select_dtypes(include='object').columns:
data[column] = data[column].fillna(data[column].mode()[0])

data = data.drop(columns=['LoanID'])
X = data.drop(columns=['Default'])
y = data['Default']

label_encoders = {}
for column in X.select_dtypes(include='object').columns:
le = LabelEncoder()
```

```
X[column] = le.fit_transform(X[column])
label_encoders[column] = le

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3,
random_state=42)

model = DecisionTreeClassifier(max_depth=5, random_state=42)
model.fit(X_train, y_train)

y_pred = model.predict(X_test)
y_pred_proba = model.predict_proba(X_test)[:, 1]

accuracy = accuracy_score(y_test, y_pred)
f1 = f1_score(y_test, y_pred, average='binary')
recall = recall_score(y_test, y_pred, average='binary')
precision = precision_score(y_test, y_pred, average='binary')
conf_matrix = confusion_matrix(y_test, y_pred)
roc_auc = roc_auc_score(y_test, y_pred_proba)

print(f"Model accuracy: {accuracy * 100:.2f}%")
print(f"F1 Score: {f1:.2f}")
print(f"Recall: {recall:.2f}")
print(f"Precision: {precision:.2f}")
print(f"ROC AUC Score: {roc_auc:.2f}")
```

```
sns.heatmap(conf_matrix, annot=True, fmt='d', cmap='Blues',
xticklabels=['No Default', 'Default'],
yticklabels=['No Default', 'Default'])
plt.xlabel('Predicted')
plt.ylabel('Actual')
plt.title('Confusion Matrix')
plt.show()

fpr, tpr, _ = roc_curve(y_test, y_pred_proba)
plt.figure()
plt.plot(fpr, tpr, color='blue', label=f'ROC curve (area = {roc_auc:.2f})')
plt.plot([0, 1], [0, 1], color='gray', linestyle='--')
plt.xlabel("False Positive Rate")
plt.ylabel("True Positive Rate")
plt.title("Receiver Operating Characteristic (ROC) Curve")
plt.legend(loc="lower right")
plt.show()

def predict_loan_default():
user_input = {}

try:
user_input['Age'] = int(input("Enter Age: "))
if user_input['Age'] < 0:
```

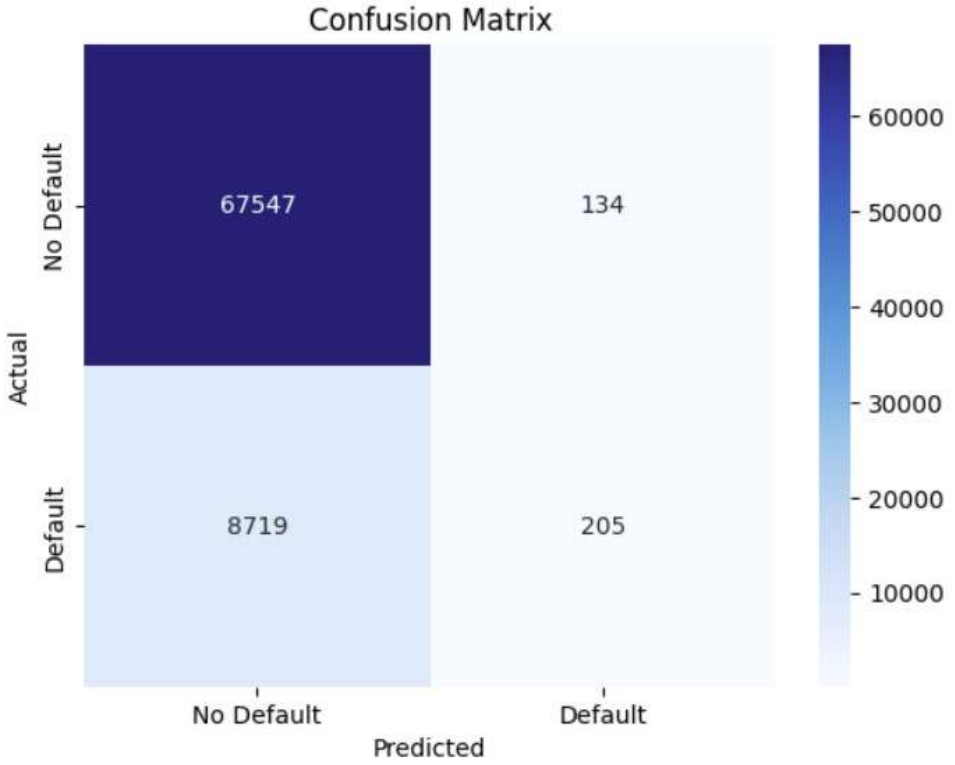
```
raise ValueError("Age must be non-negative!")
user_input['Income'] = int(input("Enter Income: "))
if user_input['Income'] < 0:
raise ValueError("Income must be non-negative!")
user_input['LoanAmount'] = int(input("Enter Loan Amount: "))
if user_input['LoanAmount'] < 0:
raise ValueError("Loan Amount must be non-negative!")
user_input['CreditScore'] = int(input("Enter Credit Score: "))
if not (300 <= user_input['CreditScore'] <= 850):
raise ValueError("Credit Score must be between 300 and 850!")
user_input['MonthsEmployed'] = int(input("Enter Months Employed:
"))
if user_input['MonthsEmployed'] < 0:
raise ValueError("Months Employed must be non-negative!")
user_input['NumCreditLines'] = int(input("Enter Number of Credit
Lines: "))
if user_input['NumCreditLines'] < 0:
raise ValueError("Number of Credit Lines must be non-negative!")
user_input['InterestRate'] = float(input("Enter Interest Rate: "))
if user_input['InterestRate'] < 0:
raise ValueError("Interest Rate must be non-negative!")
user_input['LoanTerm'] = int(input("Enter Loan Term (in months): "))
if user_input['LoanTerm'] <= 0:
raise ValueError("Loan Term must be positive!")
```

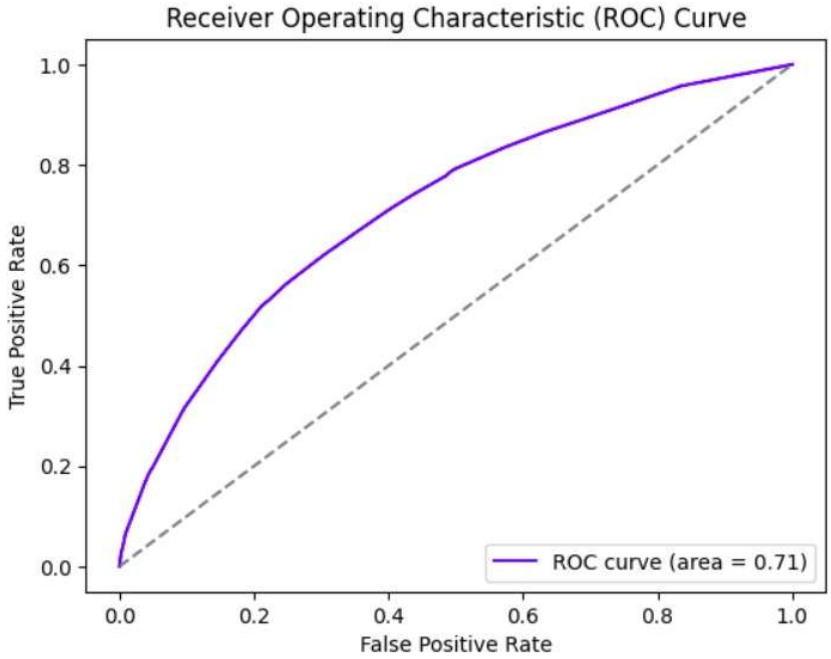
```
user_input['DTIRatio'] = float(input("Enter DTI Ratio: "))
if user_input['DTIRatio'] < 0:
raise ValueError("DTI Ratio must be non-negative!")
except ValueError as e:
print(f"Invalid input: {e}")
return

for column in label_encoders.keys():
choices = list(label_encoders[column].classes_)
print(f"Choose {column} from {choices}: ")
user_choice = input()
try:
user_input[column] =
label_encoders[column].transform([user_choice])[0]
except ValueError:
print(f"Invalid choice for {column}.")
return

user_df = pd.DataFrame([user_input])
prediction = model.predict(user_df)
if prediction[0] == 1:
print("Prediction: The loan will likely default.")
else:
print("Prediction: The loan is unlikely to default.")
```

```
choice = int(input("Do you want to enter any input? If yes (enter 1) or no (enter 0): "))

if choice:

predict_loan_default()
```

Results

```
Model accuracy: 88.44%
F1 Score: 0.04
Recall: 0.02
Precision: 0.60
ROC AUC Score: 0.71
```





```
Do you want to enter any input? If yes (enter 1) or no (enter 0):  1
Enter Age:  25
Enter Income:  90298
Enter Loan Amount:  90000
Enter Credit Score:  720
Enter Months Employed:  18
Enter Number of Credit Lines:  2
Enter Interest Rate:  9.5
Enter Loan Term (in months):  24
Enter DTI Ratio:  0.1
Choose Education from ["Bachelor's", 'High School', "Master's", 'PhD']:
 High School
Choose EmploymentType from ['Full-time', 'Part-time', 'Self-employed', 'Unemployed']:
 Unemployed
Choose MaritalStatus from ['Divorced', 'Married', 'Single']:
 Single
Choose HasMortgage from ['No', 'Yes']:
 Yes
Choose HasDependents from ['No', 'Yes']:
 No
Choose LoanPurpose from ['Auto', 'Business', 'Education', 'Home', 'Other']:
 Business
Choose HasCoSigner from ['No', 'Yes']:
 Yes
Prediction: The loan is unlikely to default.
```

Discussion

Problem Statement:

Loan default prediction is vital for financial institutions to minimize losses and optimize risk management. This project aims to predict loan defaults using borrower attributes, enabling informed lending decisions.

Dataset and Preprocessing:

The dataset (Loan_default.csv) includes borrower demographics and financial details. Missing values were handled by filling numerical data with the median and categorical data with the mode. The LoanID column was dropped, and categorical variables were encoded using LabelEncoder.

Model Selection:

A Decision Tree Classifier with a maximum depth of 5 was chosen for its simplicity and interpretability. This ensures the model avoids overfitting while providing clear decision rules.

Evaluation Metrics:

The model's performance was evaluated using:

- **Accuracy:** Overall correctness of predictions.
- **F1-Score:** Balances precision and recall.
- **Precision/Recall:** Highlights false positive/negative trade-offs.
- **ROC AUC:** Measures classification ability across thresholds. The results indicated reliable performance for loan default prediction.

Visualizations:

The confusion matrix and ROC curve provided insights into classification performance and the model's trade-offs between true and false positives.

Limitations:

- Restricted depth limits model complexity.
- Class imbalance may bias predictions.
- Advanced techniques like hyperparameter tuning and cross-validation were not applied.

Real-world Application:

The model can support loan approval systems with interpretable outputs, ensuring compliance and customer trust.

Ethical Considerations:

Fairness must be ensured to prevent bias, as false positives or negatives can significantly impact lenders and borrowers.

Future Improvements:

- Test alternative models like Naive Bayes or Logistic Regression.
- Address class imbalance using SMOTE.
- Incorporate external features (e.g., economic trends).
- Apply cross-validation for robustness.

References

1. Pandas Documentation:

Used for data manipulation and handling missing values.

2. Scikit-learn Documentation:

- **Label Encoding:** LabelEncoder Documentation
- **Train-Test Split:** train_test_split Documentation
- **Decision Tree Classifier:** DecisionTreeClassifier Documentation
- **Model Evaluation Metrics:** Metrics Module

3. Matplotlib and Seaborn:

Used for creating visualizations like confusion matrices and ROC curves.

4. Loan Default Dataset:

- Loan Default Dataset is brought from Kaggle.
- https://www.kaggle.com/datasets/nikhil1e9/loan-default

- ***Shanmukh Ram Reddy Notuu – AP22110010731***
Rohith Gunuputi - AP22110010749
Nikhileswar Kotha - AP22110010757

Thank You