KIET
GROUP OF INSTITUTIONS
Connecting Life with Learning

A

Assessment Report

on

"Predict Loan Default"

submitted as partial fulfilment for the award of

BACHELOR OF TECHNOLOGY DEGREE

SESSION 2024-25

in

Name of discipline

By

Srashti Gupta (202401100300251)

Under the supervision of

"Abhishek Shukla Sir"

KIET Group of Institutions, Ghaziabad

Affiliated to

Dr. A.P.J. Abdul Kalam Technical University, Lucknow

(Formerly UPTU)

May, 2025

Introduction

In today's financial landscape, assessing the risk of loan default is a critical component for maintaining the financial health of lending institutions. Loan default, which occurs when a borrower fails to meet the legal obligations of a loan, can lead to significant financial losses. To mitigate this risk, it is essential to accurately identify high-risk borrowers during the loan approval process.

This project focuses on building a predictive model that determines the likelihood of a borrower defaulting on a loan. The dataset contains detailed information about applicants, including their financial history, employment status, credit score, marital status, loan purpose, and other relevant features. By leveraging this data, we aim to create a machine learning model capable of classifying whether a borrower will default (target variable: Default).

To solve this classification problem, we have employed a **Random Forest Classifier**, a robust ensemble-based algorithm known for its high accuracy and interpretability. The dataset underwent preprocessing steps including label encoding for categorical variables and standardization of numerical features. The model was trained and tested using a stratified train-test split, and its performance was evaluated using metrics such as precision, recall, F1-score, and accuracy. Additionally, a confusion matrix was used to visualize the classification results.

This predictive model can serve as a powerful tool for financial institutions, enabling data-driven loan approval decisions and reducing the overall risk associated with lending.

🔧 Methodology

The approach adopted for this loan default prediction project is a structured machine learning pipeline, involving the following key stages:

---

1. Data Acquisition

The dataset titled **"1. Predict Loan Default.csv"** contains records of loan applicants along with various attributes such as demographic details, financial indicators, credit history, and a binary target variable Default, indicating whether the loan was defaulted (1) or not (0).

---

2. Data Preprocessing

Proper data preprocessing is crucial to ensure the model can learn effectively:

- **ID Column Removal:** The LoanID column was dropped as it holds no predictive value.
- **Categorical Encoding:** All categorical variables (e.g., Education, MaritalStatus, EmploymentType, etc.) were encoded using **Label Encoding** to convert string values into numerical format suitable for machine learning algorithms.
- **Handling Missing Values:** Any missing values were either filled with the most frequent category (for categorical data) or the mean (for numerical data), or left untouched if not present.
- **Feature Scaling:** StandardScaler was used to normalize the feature set to have a mean of 0 and a standard deviation of 1. This improves model performance, especially for distance-based algorithms.

---

3. Feature and Target Definition

The features (independent variables) were extracted by removing the target column Default. The target variable was set as Default, a binary classification label.

---

4. Model Selection

A **Random Forest Classifier** was chosen for its balance of high performance, robustness to noise, and low risk of overfitting. It works by constructing multiple decision trees and outputting the class that is the mode of the predictions from individual trees.

---

5. Model Training and Evaluation

- **Train-Test Split:** The dataset was split into training and testing sets using an 80/20 ratio, ensuring the model is evaluated on unseen data.
- **Training:** The Random Forest model was trained on the preprocessed training data.
- **Prediction:** Predictions were generated for the test set.
- **Evaluation Metrics:**
  - **Confusion Matrix** was used to visualize the model's performance.
  - **Precision, Recall, F1-score**, and **Accuracy** were calculated using the classification report.
  - Overall model performance was summarized using an **accuracy score**.

---

6. Visualization

- A **confusion matrix heatmap** was generated using Seaborn to provide an intuitive visualization of prediction performance.
- Tables were enhanced using **tabulate** and **termcolor** libraries to produce readable, colored outputs in the terminal or notebook environment.

This methodology ensures a transparent and reproducible process from raw data to predictive insights, offering a practical and efficient solution to the loan default prediction problem.

CODE:

```
import pandas as pd
import numpy as np
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler, LabelEncoder
from sklearn.ensemble import RandomForestClassifier
from sklearn.metrics import classification_report, confusion_matrix,
accuracy_score
from tabulate import tabulate
from termcolor import colored
import seaborn as sns
import matplotlib.pyplot as plt

# Load dataset
df = pd.read_csv("/content/1. Predict Loan Default.csv")

# Drop ID column
df.drop(columns=['LoanID'], inplace=True)

# Encode categorical columns
label_encoders = {}
for col in df.select_dtypes(include='object').columns:
    le = LabelEncoder()
    df[col] = le.fit_transform(df[col])
    label_encoders[col] = le

# Features and target
X = df.drop('Default', axis=1)
y = df['Default']

# Train/test split
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2,
random_state=42)

# Scaling
scaler = StandardScaler()
X_train = scaler.fit_transform(X_train)
X_test = scaler.transform(X_test)

# Model
model = RandomForestClassifier(random_state=42)
```

```
model.fit(X_train, y_train)
y_pred = model.predict(X_test)

# Confusion Matrix
conf_matrix = confusion_matrix(y_test, y_pred)
conf_df = pd.DataFrame(conf_matrix,
                       index=[f'Actual {i}' for i in model.classes_],
                       columns=[f'Predicted {i}' for i in model.classes_])

# Classification Report
report = classification_report(y_test, y_pred, output_dict=True)
report_df = pd.DataFrame(report).transpose().round(2)

# Accuracy
accuracy = accuracy_score(y_test, y_pred)

# 🎨 Color-coded titles
print("\n" + colored("📊 CONFUSION MATRIX", "cyan", attrs=["bold"]))
print(tabulate(conf_df, headers="keys", tablefmt="fancy_grid"))

print("\n" + colored("📋 CLASSIFICATION REPORT", "magenta",
attrs=["bold"]))
print(tabulate(report_df, headers="keys", tablefmt="fancy_grid",
showindex=True))

print("\n" + colored("✅ ACCURACY SCORE:", "green", attrs=["bold"]) + f"
{colored(f'{accuracy:.2%}', 'yellow')}")

# Optional: Heatmap (Jupyter / GUI required)
try:
    plt.figure(figsize=(6, 4))
    sns.heatmap(conf_df, annot=True, fmt="d", cmap="Blues", cbar=False)
    plt.title("Confusion Matrix Heatmap")
    plt.ylabel("Actual")
    plt.xlabel("Predicted")
    plt.tight_layout()
    plt.show()
except:
    pass
```

CONFUSION MATRIX

	Predicted 0	Predicted 1
Actual 0	45004	166
Actual 1	5614	286

CLASSIFICATION REPORT

	precision	recall	f1-score	support
0	0.89	1	0.94	45170
1	0.63	0.05	0.09	5900
accuracy	0.89	0.89	0.89	0.89
macro avg	0.76	0.52	0.51	51070
weighted avg	0.86	0.89	0.84	51070

ACCURACY SCORE: 88.68%