**Assessment Report**

on

# "Loan Default Prediction"

SESSION 2024-25

in

## CSE(AIML)

By

Jitesh Kumar (202401100400103)

**Under the supervision of**

"Mr. Abhishek Shukla"

## KIET Group of Institutions, Ghaziabad

**April, 2025**

# Loan Default Prediction Report

## 1. Introduction

The goal of this project is to build a machine learning model to predict whether a borrower will default on a loan based on their financial history and credit data. Accurate prediction of loan default helps financial institutions minimize risk and make informed lending decisions.

## 2. Dataset Overview

The dataset used is titled '1. Predict Loan Default.csv'. It contains borrower financial and credit data. The target variable is 'Default' where 0 indicates no default and 1 indicates default.

Preprocessing steps included:

- - Dropping the 'LoanID' column
- - Removing rows with missing values
- - Encoding categorical variables using LabelEncoder

## 3. Handling Class Imbalance

Loan default data is often imbalanced, meaning there are fewer defaulters than non-defaulters. To address this, SMOTE (Synthetic Minority Oversampling Technique) was used to balance the dataset by generating synthetic examples of the minority class.

## 4. Model Building

The model used is a RandomForestClassifier from Scikit-learn. The 'class_weight' parameter was set to 'balanced'.

The dataset was split into 80% training and 20% testing sets.

## 5. Evaluation Metrics

After training, the model was evaluated on the test set with the following results:

- Confusion Matrix:

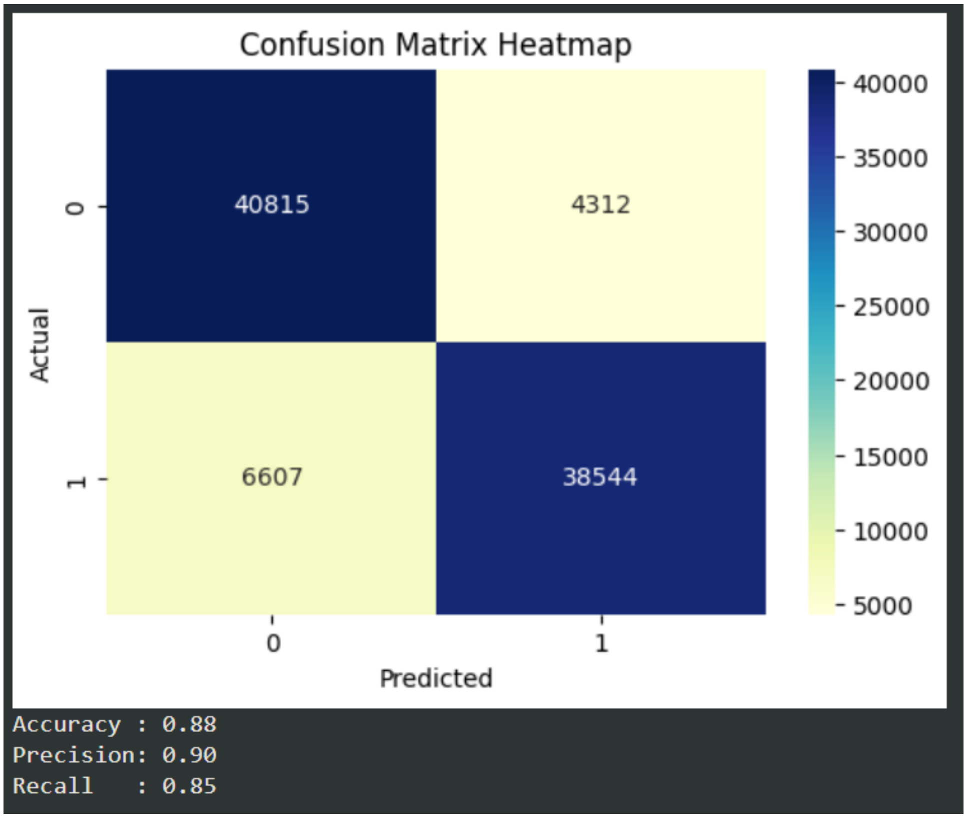
| | Predicted: No | Predicted: Yes |
|---|---|---|
| Actual: No | 40,815 | 4,312 |
| Actual: Yes | 6,607 | 38,544 |

- Performance Metrics:
- - Accuracy: 0.88

- - Precision: 0.90
- - Recall: 0.85

## 6. Visualization

The confusion matrix heatmap below provides a visual representation of the model's performance.




## 7. Conclusion

The Random Forest model combined with SMOTE performed well, achieving a good balance between precision and recall. This model can be a valuable tool for banks and lending institutions to evaluate loan applications more reliably.

```
# Install imbalanced-learn for SMOTE
!pip install -q imbalanced-learn

# Import necessary libraries
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns

from sklearn.model_selection import train_test_split
from sklearn.ensemble import RandomForestClassifier
from sklearn.preprocessing import LabelEncoder
from sklearn.metrics import confusion_matrix, accuracy_score, precision_score, recall_score
from imblearn.over_sampling import SMOTE

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

# Drop LoanID and handle missing values
df.drop(columns=['LoanID'], inplace=True)
df.dropna(inplace=True)

# Encode categorical variables
label_encoder = LabelEncoder()
for col in df.columns:
    if df[col].dtype == 'object':
        df[col] = label_encoder.fit_transform(df[col])

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

```
# Apply SMOTE to handle class imbalance
smote = SMOTE(random_state=42)
X_resampled, y_resampled = smote.fit_resample(X, y)

# Train-test split
X_train, X_test, y_train, y_test = train_test_split(X_resampled, y_resampled, test_size=0.2, random_state=42)

# Train model with class weighting
model = RandomForestClassifier(class_weight='balanced', random_state=42)
model.fit(X_train, y_train)

# Predictions
y_pred = model.predict(X_test)

# Confusion Matrix
cm = confusion_matrix(y_test, y_pred)

plt.figure(figsize=(6,4))
sns.heatmap(cm, annot=True, fmt='d', cmap='YlGnBu')
plt.xlabel("Predicted")
plt.ylabel("Actual")
plt.title("Confusion Matrix Heatmap")
plt.show()

# Evaluation metrics
acc = accuracy_score(y_test, y_pred)
prec = precision_score(y_test, y_pred)
rec = recall_score(y_test, y_pred)

print(f"Accuracy : {acc:.2f}")
print(f"Precision: {prec:.2f}")
print(f"Recall   : {rec:.2f}")
```