****

**A Assesment Report**

On

**“Problem Statement-Predict Loan Default”**

submitted as partial fulfillment for the award of

**BACHELOR OF TECHNOLOGY**

**DEGREE**

SESSION 2024-25

in

**CSE(AIML)**

By

Urvi Gupta (202401100400201)

**Under the supervision of**

“Abhishek Shukla”

**KIET Group of Institutions, Ghaziabad**

Affiliated to

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

(Formerly UPTU)

**May, 2025**

**INDEX**

**S.no Title Page no.**

1. **Title Page 1**
2. **Introduction 2**
3. **Methodology 3**
4. **CODE 4-5**
5. **Output/Result 6**
6. **References 7**

**Title Page**

**Problem Statement**:

Predict Loan Default: Classify whether a borrower will default on a loan using financial history and credit scores.

**Personal Details**:

* **Name**: Urvi Gupta
* **Roll Number**: 202401100400201
* **Discipline**: BTech CSE (AIML)
* **Course**: Artificial Intelligence
* **Assessment**: AI MSE-2

**Introduction**

This project aims to predict whether a borrower will default on a loan based on financial and credit score data.  
We use machine learning techniques for classification. The goal is to build a model that can learn from past data and make accurate predictions on new applicants.  
The problem is solved using **Random Forest Classifier**, a robust ensemble learning algorithm.

**Methodology**

1. **Data Upload**: Uploaded the .csv file using Google Colab.
2. **Data Preprocessing**:
   * Dropped non-relevant columns like LoanID.
   * Removed missing values.
   * Encoded categorical variables using LabelEncoder.
   * Scaled the features using StandardScaler.
3. **Model Training**:
   * Split data into training and test sets (80-20 split).
   * Used RandomForestClassifier for classification.
4. **Evaluation**:
   * Used metrics: **accuracy**, **precision**, **recall**, and **confusion matrix**.

**CODE**

# 📌 Step 1: Import necessary libraries

from google.colab import files

uploaded = files.upload()

import pandas as pd

import numpy as np

import seaborn as sns

import matplotlib.pyplot as plt

from sklearn.model\_selection import train\_test\_split

from sklearn.preprocessing import LabelEncoder, StandardScaler

from sklearn.ensemble import RandomForestClassifier

from sklearn.metrics import confusion\_matrix, classification\_report, accuracy\_score, precision\_score, recall\_score

# ✅ Load the uploaded file (use the exact file name)

df = pd.read\_csv('1. Predict Loan Default.csv')

# Drop 'LoanID' column (if exists)

if 'LoanID' in df.columns:

    df = df.drop(columns=['LoanID'])

# Drop missing values

df = df.dropna()

# 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

# Split features and target

X = df.drop('Default', axis=1)

y = df['Default']

# Scale features

scaler = StandardScaler()

X\_scaled = scaler.fit\_transform(X)

# Split into training and testing sets

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X\_scaled, y, test\_size=0.2, random\_state=42)

# Train model

model = RandomForestClassifier(random\_state=42)

model.fit(X\_train, y\_train)

# Predict

y\_pred = model.predict(X\_test)

# Confusion Matrix Heatmap

cm = confusion\_matrix(y\_test, y\_pred)

plt.figure(figsize=(6, 4))

sns.heatmap(cm, annot=True, fmt='d', cmap='Blues')

plt.title('Confusion Matrix')

plt.xlabel('Predicted')

plt.ylabel('Actual')

plt.show()

# Evaluation Metrics

accuracy = accuracy\_score(y\_test, y\_pred)

precision = precision\_score(y\_test, y\_pred)

recall = recall\_score(y\_test, y\_pred)

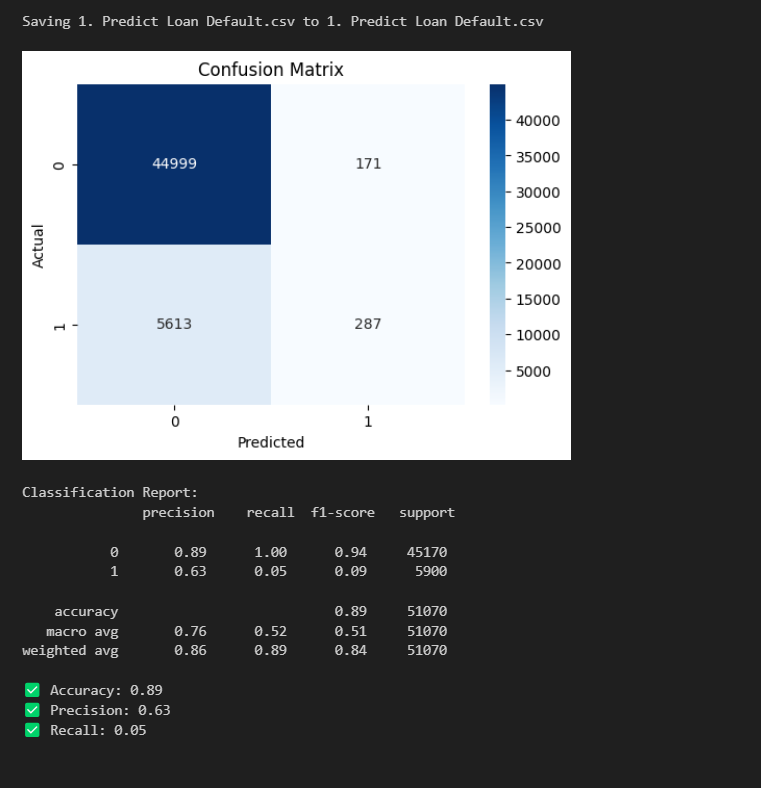
print("Classification Report:\n", classification\_report(y\_test, y\_pred))

print(f"✅ Accuracy: {accuracy:.2f}")

print(f"✅ Precision: {precision:.2f}")

print(f"✅ Recall: {recall:.2f}")

**Output/Result**

**–**

**References**

* Dataset Source: Kaggle
* Libraries used: scikit-learn, pandas, seaborn, matplotlib
* Code executed on: Google Colab
* Special thanks to: Faculty and AI department