An Assessment Report

on

“Problem Statement”

submitted as partial fulfillment for the award of

BACHELOR OF TECHNOLOGY DEGREE

SESSION 2024-25

in

CSEAI

By

Tanmay Kesharwani (202401100300261)

Under the supervision of

“MR. ABHISHEK SHUKLA”

KIET Group of Institutions, Ghaziabad

May, 2025

Introduction

In the modern financial ecosystem, lending institutions face significant risks when providing loans to individuals. One of the critical challenges is predicting whether a borrower is likely to default on a loan. Accurately identifying such risks not only safeguards financial institutions but also contributes to the health of the economy.

This project focuses on building a machine learning model that predicts loan default using various borrower-related features like income, credit score, employment status, loan purpose, and more.

Methodology

Our approach followed these major steps:

1. Data Exploration
   - Dataset includes borrower data such as CreditScore, Income, LoanAmount, etc.
   - The target variable is Default (1 for default, 0 for no default).
2. Data Preprocessing
   - Missing Values: Dropped for simplicity.
   - Encoding: Label Encoding was applied to categorical variables.
   - Splitting: Dataset split into 80% training and 20% testing sets.
3. Model Selection
   - Chose Random Forest Classifier due to its efficiency on tabular datasets and ability to handle both linear and non-linear relationships.

4. Evaluation Metrics

- Accuracy, Precision, and Recall were used to evaluate performance.

5. Confusion Matrix

- Used a heatmap to visualize true positives, false positives, true negatives, and false negatives.

Code

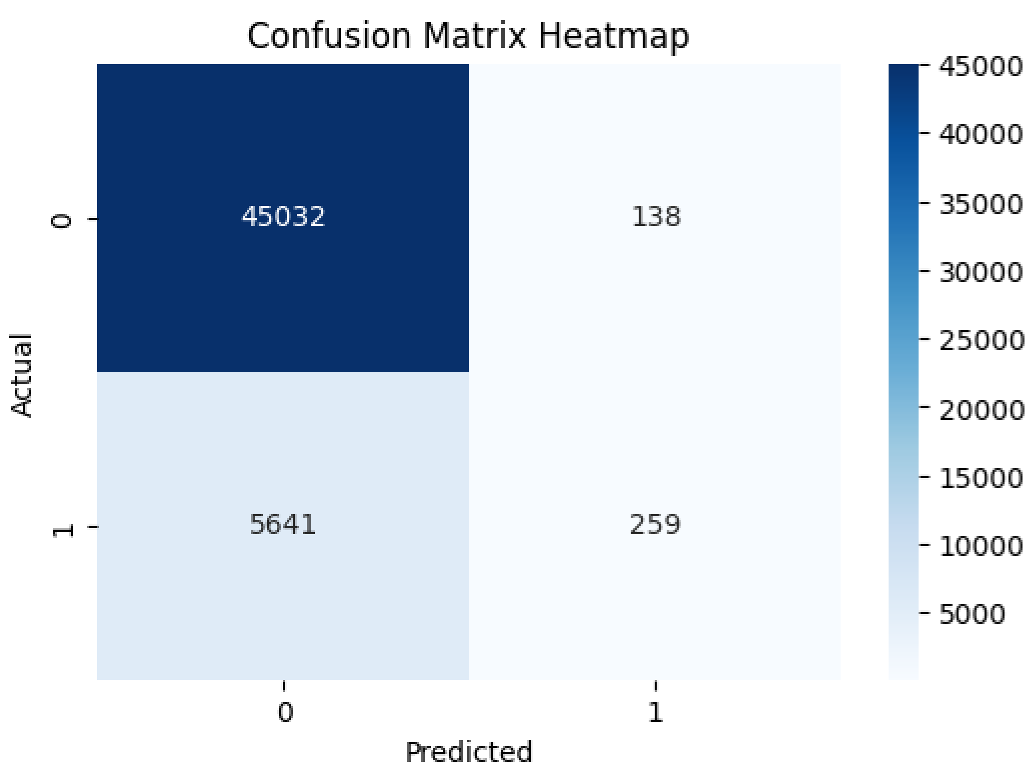
```
from google.colab import files
uploaded = files.upload()
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import LabelEncoder
from sklearn.ensemble import RandomForestClassifier
from sklearn.metrics import confusion_matrix, accuracy_score, precision_score, recall_score

# 1. Load dataset
df = pd.read_csv("Predict Loan Default.csv")
print("First 5 rows:\n", df.head())

# 2. Data Info
print("\nBasic Info:")
print(df.info())

# 3. Handle missing values (drop or fill)
df = df.dropna()  # for simplicity

# 4. Encode categorical variables (if any)
le = LabelEncoder()
for col in df.select_dtypes(include=['object']).columns:
    df[col] = le.fit_transform(df[col])
# 5. Separate features and target
```

```
X = df.drop('Default', axis=1)  # replace 'Loan_Default' with actual column name if different
y = df['Default']

# 6. Split data into train and test
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

# 7. Train a classifier
model = RandomForestClassifier(random_state=42)
model.fit(X_train, y_train)

# 8. Predict and Evaluate
y_pred = model.predict(X_test)
acc = accuracy_score(y_test, y_pred)
prec = precision_score(y_test, y_pred)
rec = recall_score(y_test, y_pred)
print(f"\nAccuracy: {acc:.2f}")
print(f"Precision: {prec:.2f}")
print(f"Recall: {rec:.2f}")

# 9. Confusion Matrix
cm = confusion_matrix(y_test, y_pred)

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

Output

```
First 5 rows:
       LoanID  Age  Income  LoanAmount  CreditScore  MonthsEmployed  \
0  I38PQUQS96   56   85994       50587          520              80   
1  HPSK72WA7R   69   50432      124440          458              15   
2  C1OZ6DPJ8Y   46   84208      129188          451              26   
3  V2KKSFM3UN   32   31713       44799          743               0   
4  EY08JDHTZP   60   20437        9139          633               8   

   NumCreditLines  InterestRate  LoanTerm  DTIRatio    Education  \
0               4         15.23        36      0.44   Bachelor's   
1               1          4.81        60      0.68     Master's   
2               3         21.17        24      0.31     Master's   
3               3          7.07        24      0.23  High School   
4               4          6.51        48      0.73   Bachelor's   

  EmploymentType MaritalStatus HasMortgage HasDependents LoanPurpose  \
0      Full-time      Divorced         Yes           Yes       Other   
1      Full-time       Married          No            No       Other   
2     Unemployed      Divorced         Yes           Yes        Auto   
3      Full-time       Married          No            No    Business   
4     Unemployed      Divorced          No           Yes        Auto   

  HasCoSigner  Default  
0         Yes        0  
1         Yes        0  
2          No        1  
3          No        0  
4          No        0  
```

```
Basic Info:
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 255347 entries, 0 to 255346
Data columns (total 18 columns):
 #   Column          Non-Null Count   Dtype  
---  ------          --------------   -----  
 0   LoanID          255347 non-null  object 
 1   Age             255347 non-null  int64  
 2   Income          255347 non-null  int64  
 3   LoanAmount      255347 non-null  int64  
 4   CreditScore     255347 non-null  int64  
 5   MonthsEmployed  255347 non-null  int64  
 6   NumCreditLines  255347 non-null  int64  
 7   InterestRate    255347 non-null  float64
 8   LoanTerm        255347 non-null  int64  
 9   DTIRatio        255347 non-null  float64
 10  Education       255347 non-null  object 
 11  EmploymentType  255347 non-null  object 
 12  MaritalStatus   255347 non-null  object 
 13  HasMortgage     255347 non-null  object 
 14  HasDependents   255347 non-null  object 
 15  LoanPurpose     255347 non-null  object 
 16  HasCoSigner     255347 non-null  object 
 17  Default         255347 non-null  int64  
dtypes: float64(2), int64(8), object(8)
memory usage: 35.1+ MB
None

Accuracy: 0.89
Precision: 0.65
Recall: 0.04
```



References / Credits

- Dataset Source: Provided for academic use (uploaded by project author).
- Libraries Used: pandas, numpy, matplotlib, seaborn, sklearn