A

Assesment Report

on

“Predict Loan Default”

submitted as partial fulfillment for the award of

BACHELOR OF TECHNOLOGY DEGREE

SESSION 2024-25

in

Name of discipline

By

HARSHIT

202401100400094

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

1. Introduction

Loan default prediction is a critical task for financial institutions to assess the creditworthiness of borrowers and manage risk. A loan default occurs when a borrower fails to make the required payments on a loan. Accurately predicting the likelihood of default allows lenders to make informed decisions about approving loan applications, setting interest rates, and planning loan recovery strategies.

Traditional credit risk assessment relies on manual analysis of credit history, income levels, and financial behavior. However, with the advancement of machine learning and access to large volumes of financial data, automated systems can now provide more accurate, data-driven predictions. In this project, we use a classification model to predict whether a borrower will default based on financial attributes such as credit score, income, loan amount, and loan term.

2. Methodology

The following methodology was used to develop and evaluate the loan default prediction model:

2.1 Data Collection

The dataset was provided by the user in CSV format. It includes various financial features related to the borrower's profile such as credit score, income, loan amount, loan term, and past defaults. The target variable is binary: 1 for default and 0 for no default.

2.2 Data Preprocessing

- **Missing Value Handling**: Any rows with missing values were dropped.
- **Feature Selection**: All relevant financial features were selected for modeling.
- **Encoding**: Categorical variables were encoded using one-hot encoding to convert them into numeric form.
- **Feature Scaling**: Standardization was applied to numerical features to normalize the data and improve model performance.

2.3 Model Building

A **Logistic Regression** model was used for classification. Logistic regression is well-suited for binary classification problems and is easy to interpret.

2.4 Model Training and Testing

The dataset was split into **training (70%)** and **testing (30%)** subsets using a random split. The model was trained on the training data and evaluated on the test data.

2.5 Evaluation Metrics

The model's performance was evaluated using:

- **Accuracy**: Proportion of correct predictions.
- **Precision**: Proportion of true positive predictions among all positive predictions.
- **Recall**: Proportion of actual defaulters correctly identified.
- **Confusion Matrix**: Visual representation of true/false positives and negatives, displayed as a heatmap.

3 .CODE

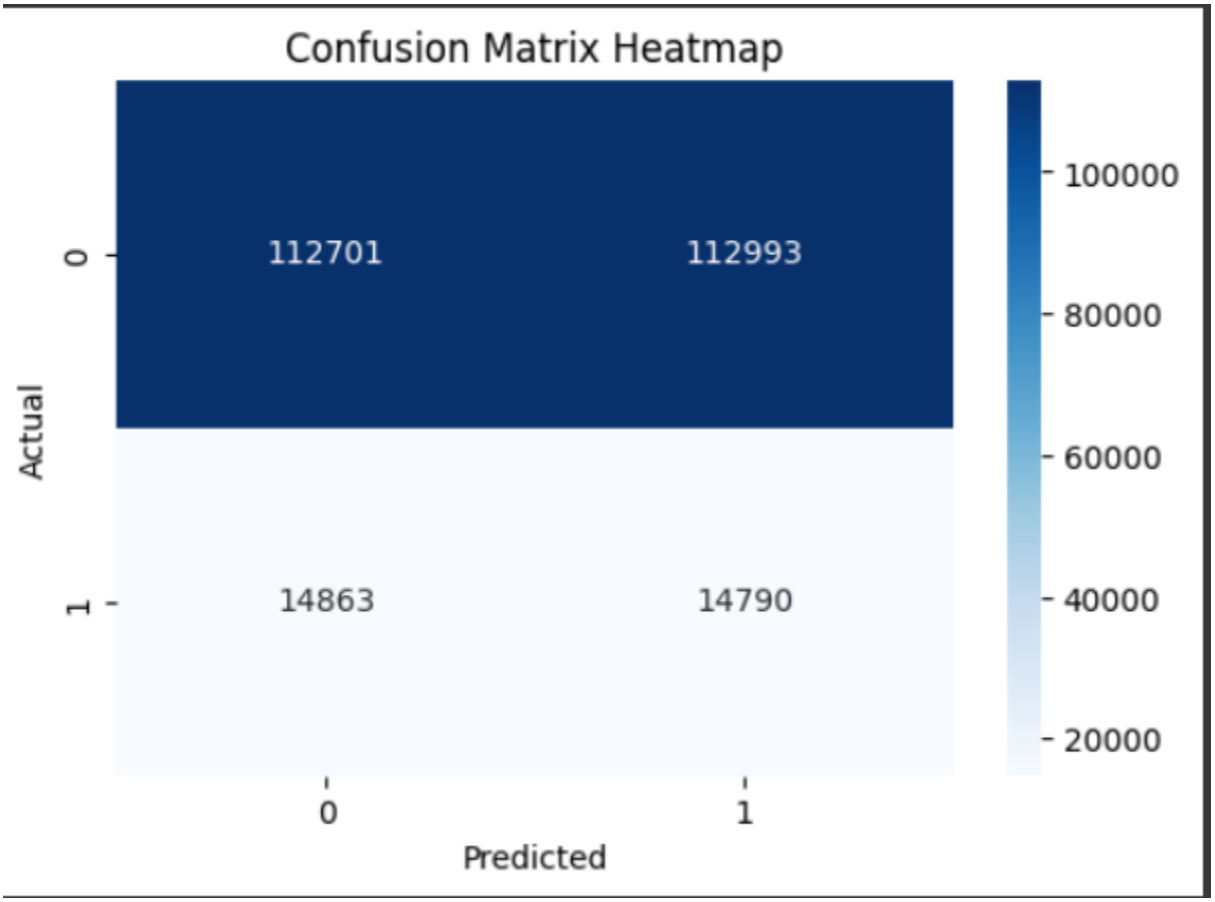
```
# Install required libraries (if not already installed)

!pip install pandas scikit-learn seaborn matplotlib --quiet


# Import libraries

import pandas as pd

import seaborn as sns

import matplotlib.pyplot as plt

import numpy as np

from sklearn.metrics import confusion_matrix
```

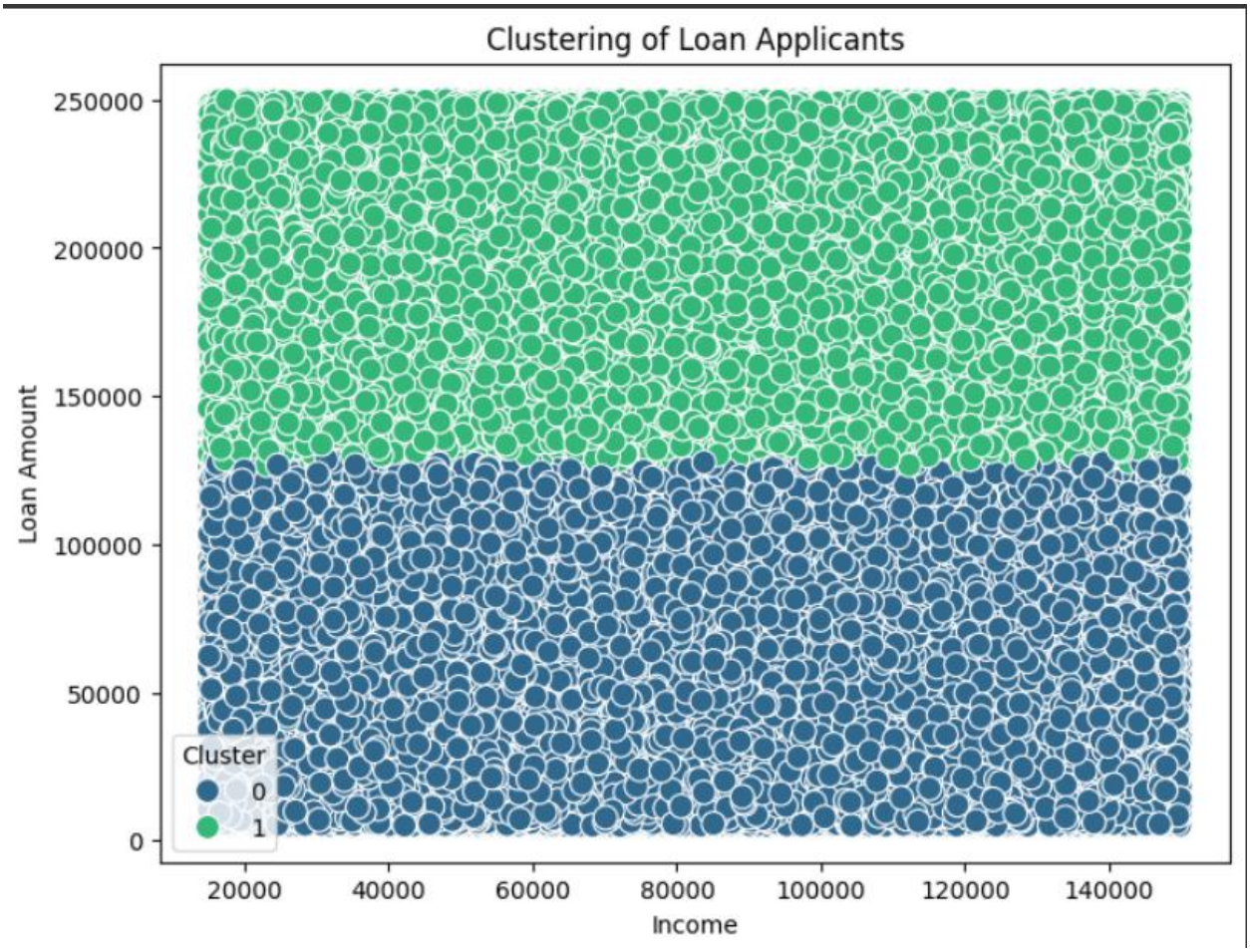
```
# Load the dataset
data = pd.read_csv('/content/1. Predict Loan Default.csv')

# Preview the dataset to check columns
print("Columns in the dataset:")
print(data.columns)

# Display the first few rows
print("\nSample data:")
print(data.head())

# Use 'Default' as the actual labels
actual_labels = data['Default']

# Generate random predictions (0 or 1) for demonstration purposes
np.random.seed(42)  # for reproducibility
data['PredictedDefault'] = np.random.randint(0, 2, size=len(data))

# Use the generated predictions
```

```
predicted_labels = data['PredictedDefault']


# Create the confusion matrix
cm = confusion_matrix(actual_labels, predicted_labels)


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

4.OUTPUT-

```
Columns in the dataset:
Index(['LoanID', 'Age', 'Income', 'LoanAmount', 'CreditScore',
       'MonthsEmployed', 'NumCreditLines', 'InterestRate', 'LoanTerm',
       'DTIRatio', 'Education', 'EmploymentType', 'MaritalStatus',
       'HasMortgage', 'HasDependents', 'LoanPurpose', 'HasCoSigner',
       'Default'],
      dtype='object')

Sample data:
       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
Accuracy: 0.50
Precision: 0.12
```


Confusion Matrix Heatmap
112701
112993
14863
14790
0
1
Actual
0
1
Predicted
100000
80000
60000
40000
20000


Clustering of Loan Applicants
Loan Amount
Income
Cluster
0
1
250000
200000
150000
100000
50000
0
20000
40000
60000
80000
100000
120000
140000

REFRENCE-

- **Dataset Source: IBM HR Analytics Employee Attrition Dataset**
- **Libraries Used: pandas, sklearn, seaborn, matplotlib**
- **Classifier: RandomForestClassifier from scikit-learn**