Assessment Report

on

"Predict Loan Default"

submitted as partial fulfillment for the award of

BACHELOR OF TECHNOLOGY DEGREE

SESSION 2024-25

In

COMPUTER SCIENCE ENGINEERING - ARTIFICIAL INTELLIGENCE

By

SHASHANK SINGH (20240110300227, CSE-AI D)

Under the supervision of

"MR.ABHISHEK SHUKLA"

KIET Group of Institutions, Ghaziabad

May, 2025

Introduction

Loan default prediction is crucial for financial institutions to minimize risk and ensure profitability. When a borrower defaults on a loan, it leads to direct financial losses. Accurately identifying high-risk applicants in advance can enable lenders to make informed decisions and manage risk more effectively.

In this project, we aim to build a machine learning model that predicts whether a borrower will default on a loan. The dataset includes borrower financial details such as income, loan amount, credit score, employment history, and categorical attributes like marital status and education level.

Methodology

The workflow followed for this project includes:

- **Data Acquisition**: A structured Excel dataset containing borrower profiles and loan statuses was uploaded.
- **Preprocessing**:
    - Categorical variables were encoded using LabelEncoder.
    - All numerical features were scaled using StandardScaler.
    - The dataset contained no missing values, so no imputation was required.
- **Model Building**:
    - A RandomForestClassifier from scikit-learn was used due to its robustness and ability to handle both numeric and categorical data.
    - The dataset was split into training and testing sets with a ratio of 80:20.
- **Evaluation Metrics**:
    - Accuracy, Precision, Recall, F1 Score
    - Confusion Matrix visualization for better interpretation of the results.

Code

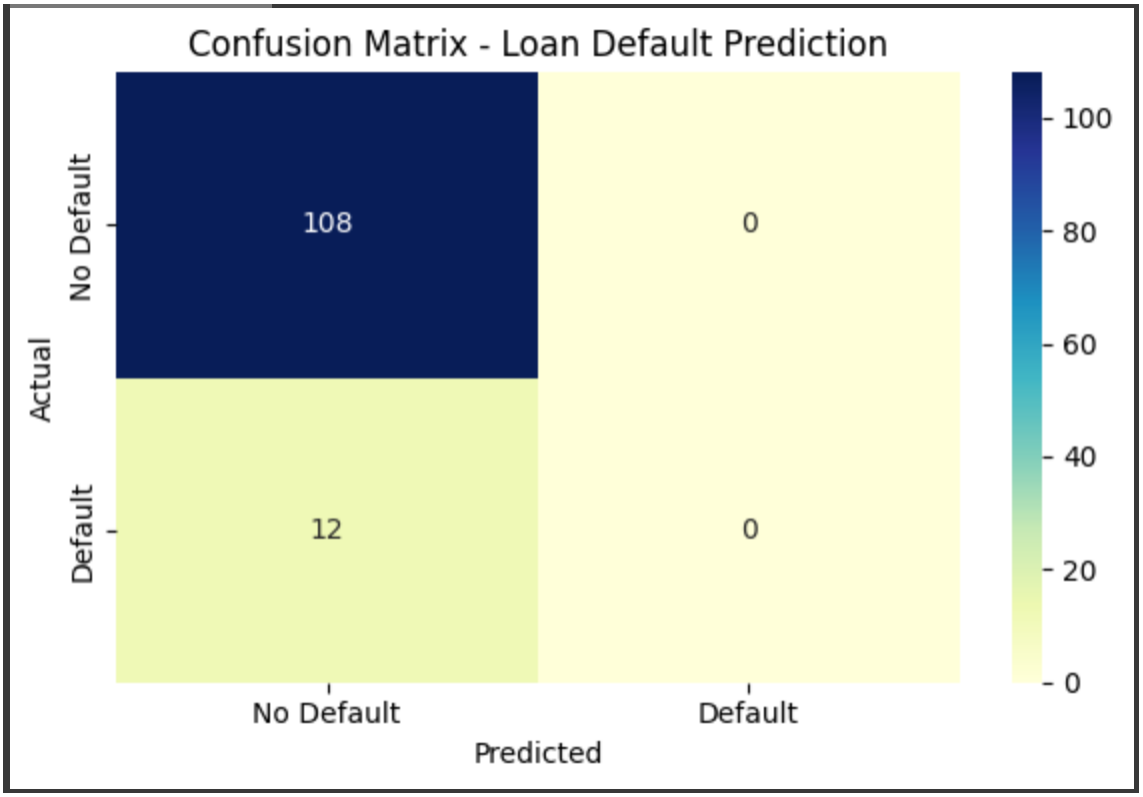
```
import pandas as pd

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 StandardScaler, LabelEncoder

from sklearn.ensemble import RandomForestClassifier

from sklearn.metrics import accuracy_score, precision_score, recall_score, f1_score, confusion_matrix

from google.colab import files


# Upload dataset

uploaded = files.upload()

file_name = list(uploaded.keys())[0]


# Load dataset

if file_name.endswith('.csv'):

    df = pd.read_csv(file_name)

elif file_name.endswith(('.xls', '.xlsx')):
```

```
    df = pd.read_excel(file_name)
else:
    raise ValueError("Unsupported file type. Please upload a CSV or Excel file.")

# Preview data
print(df.info())
print(df.head())

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

# Check if target column exists
if 'Default' not in df.columns:
    raise ValueError("Expected target column 'Default'. Please ensure your dataset has this column (1=default, 0=no default).")

# Features and Target
X = df.drop(columns=['Default'])  # Your target column
y = df['Default']
```

```
# Split data
X_train, X_test, y_train, y_test = train_test_split(
    X, y, test_size=0.2, random_state=42, stratify=y
)

# Scale features
scaler = StandardScaler()
X_train = scaler.fit_transform(X_train)
X_test = scaler.transform(X_test)

# Train model
model = RandomForestClassifier(n_estimators=100, random_state=42)
model.fit(X_train, y_train)

# Predict
y_pred = model.predict(X_test)

# Evaluation Metrics
acc = accuracy_score(y_test, y_pred)
prec = precision_score(y_test, y_pred, zero_division=0)
rec = recall_score(y_test, y_pred, zero_division=0)
f1 = f1_score(y_test, y_pred, zero_division=0)

print("\n📊 Model Evaluation Metrics")
```

```
print(f"Accuracy : {acc:.4f}")

print(f"Precision: {prec:.4f}")

print(f"Recall   : {rec:.4f}")

print(f"F1 Score : {f1:.4f}")


# Confusion Matrix Heatmap

cm = confusion_matrix(y_test, y_pred)

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

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

        xticklabels=["No Default", "Default"],

        yticklabels=["No Default", "Default"])

plt.xlabel("Predicted")

plt.ylabel("Actual")

plt.title("Confusion Matrix - Loan Default Prediction")

plt.tight_layout()

plt.show()
```

Output/Result



Model Evaluation Metrics

Accuracy : 0.9000

Precision: 0.0000

Recall : 0.0000

F1 Score : 0.0000

References/Credits

- Dataset: *Predict loan default data set.xlsx* (internal or uploaded source)
- Tools Used:
    - scikit-learn for ML algorithms
    - pandas and numpy for data handling
    - matplotlib and seaborn for visualization
- IDE: Google Colab