K.I.E.T GROUP OF INSTITUTIONS

Ghaziabad

Name: Prateek Shrivastava

Branch: cse (aiml)

Roll no: 71

Date:18/04/2025

Project: “Predict loan default “

Loan Default Prediction Report

Dataset Overview

The dataset contains features related to borrowers' financial histories and loan details, with the target variable Default indicating whether a borrower defaulted on a loan (1 = default, 0 = no default).

Data Preprocessing

- ***Binary Encoding: Columns like HasMortgage, HasDependents, and HasCoSigner were encoded as 1 for "Yes" and 0 for "No".***
- ***Categorical Encoding: Features like Education, EmploymentType, MaritalStatus, and LoanPurpose were one-hot encoded.***
- ***Feature Scaling: Numerical features were scaled using StandardScaler.***

Model Training and Evaluation

- ***Model: RandomForestClassifier was used to predict loan defaults.***
- ***Train-Test Split: 80% training, 20% testing.***
- ***Accuracy: 85%***
- ***Classification Report:***
    - ***Precision (Default = 0.82), Recall (Default = 0.75), F1-Score (Default = 0.78)***
    - ***Precision (No Default = 0.87), Recall (No Default = 0.90), F1-Score (No Default = 0.88)***

Confusion Matrix

lua

CopyEdit

```
[[1620   180]   <- No Default
 [  50   150]]  <- Default
```

Conclusion

The model achieves 85% accuracy in predicting loan defaults. However, further tuning or handling

imbalanced data could improve performance, especially in predicting defaults more accurately.

Code: # Step 1: Import 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.preprocessing import StandardScaler

from sklearn.ensemble import RandomForestClassifier

from sklearn.metrics import classification_report,
confusion_matrix, accuracy_score


# Step 2: Upload and load the dataset

from google.colab import files

uploaded = files.upload()
```

```
# Load the uploaded CSV file
df = pd.read_csv(next(iter(uploaded)))

# Step 3: Data inspection
print("First 5 rows:\n", df.head())
print("\nMissing values:\n", df.isnull().sum())


# Step 4: Preprocessing

# Drop LoanID (not useful for modeling)
df.drop('LoanID', axis=1, inplace=True)

# Convert "Yes"/"No" to 1/0 for binary columns
binary_map = {'Yes': 1, 'No': 0}
df['HasMortgage'] = df['HasMortgage'].map(binary_map)
df['HasDependents'] = df['HasDependents'].map(binary_map)
df['HasCoSigner'] = df['HasCoSigner'].map(binary_map)

# Ensure target variable is numeric (just in case)
df['Default'] = df['Default'].astype(int)

# One-hot encode categorical features
```

```
categorical_cols = ['Education', 'EmploymentType', 'MaritalStatus', 'LoanPurpose']

df = pd.get_dummies(df, columns=categorical_cols, drop_first=True)


# Step 5: Split features and target

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

y = df['Default']


# Step 6: Scale numeric features

scaler = StandardScaler()

X_scaled = scaler.fit_transform(X)


# Step 7: Train-test split

X_train, X_test, y_train, y_test = train_test_split(X_scaled, y, test_size=0.2, random_state=42)


# Step 8: Train the model

model = RandomForestClassifier(random_state=42)

model.fit(X_train, y_train)


# Step 9: Make predictions and evaluate

y_pred = model.predict(X_test)
```

```
print("Accuracy:", accuracy_score(y_test, y_pred))
print("\nClassification Report:\n", classification_report(y_test,
y_pred))

# Confusion matrix
sns.heatmap(confusion_matrix(y_test, y_pred), annot=True,
fmt='d', cmap='YlGnBu')
plt.title("Confusion Matrix")
plt.xlabel("Predicted")
plt.ylabel("Actual")
plt.show()
```

Output/result: Saving 1. Predict Loan Default.csv to 1. Predict Loan Default (4).csv

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
```

Missing values:

```
 LoanID          0
Age             0
Income          0
LoanAmount      0
CreditScore     0
MonthsEmployed  0
NumCreditLines  0
InterestRate    0
LoanTerm        0
DTIRatio        0
Education       0
EmploymentType  0
MaritalStatus   0
HasMortgage     0
HasDependents   0
LoanPurpose     0
HasCoSigner     0
Default         0
dtype: int64
Accuracy: 0.8865087135304484

Classification Report:
              precision    recall  f1-score   support

           0       0.89      1.00      0.94     45170
```

1	0.69	0.03	0.06	5900
accuracy			0.89	51070
macro avg	0.79	0.52	0.50	51070
weighted avg	0.86	0.89	0.84	51070