Assesment Report

on

“Predict Loan Default”

submitted as partial fulfillment for the award of

BACHELOR OF TECHNOLOGY

DEGREE

SESSION 2024-25

in

CSE-AIML

By

Name (202401100400134)

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

Introduction

Problem Statement

Banks and financial institutions face a big risk when lending money — **some people fail to repay their loans**, which leads to losses. This project aims to help banks **predict whether a loan applicant is likely to default (not repay the loan)** based on their personal and financial information.

When a bank or financial institution gives out a loan (like a home loan, car loan, or personal loan), it expects the borrower to **repay it over time** with interest. However, **not all borrowers repay their loans** — and those who fail to repay are said to have **defaulted**.

Loan defaults result in **financial losses for banks**, so they want to minimize risk by **predicting in advance** whether a loan applicant is likely to default.

Methodology

The approach to solving the loan default prediction problem involves building a supervised machine learning model that classifies whether a loan applicant is likely to default or not. We begin by loading and inspecting the dataset to understand its structure and identify any missing or irrelevant data. Preprocessing steps such as dropping unnecessary columns, handling missing values, encoding categorical variables, and scaling numerical features ensure the data is clean and model-ready. We perform exploratory data analysis to visualize relationships between features and detect any class imbalances. The dataset is then split into training and testing sets to evaluate the model's generalization capability.

A Random Forest Classifier is chosen due to its robustness and ability to handle mixed data types, and it is trained on the prepared dataset. The model’s performance is assessed using accuracy, precision, recall, F1-score, and confusion matrix. Finally, feature importance is analyzed to interpret which variables have the most impact on the prediction, enabling a transparent and data-driven approach to loan risk assessment.

Code

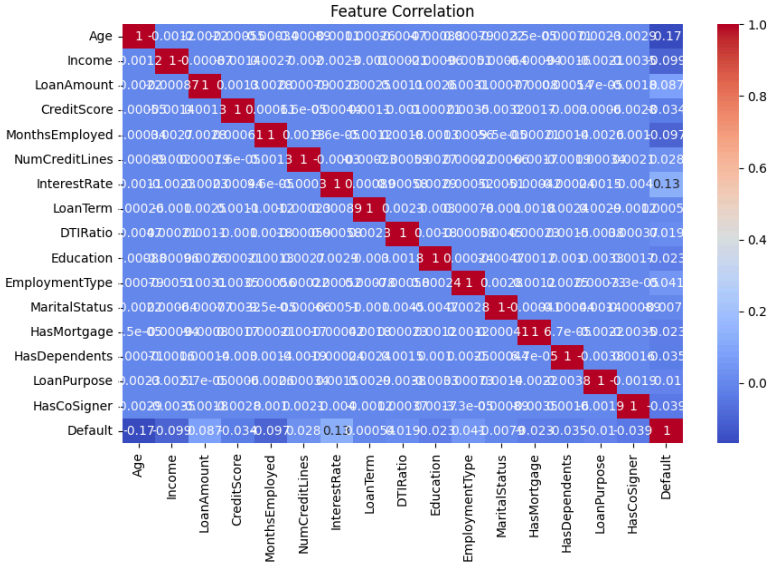
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import LabelEncoder, StandardScaler
from sklearn.ensemble import RandomForestClassifier
from sklearn.metrics import classification_report, confusion_matrix
df = pd.read_csv("1. Predict Loan Default.csv")
print("Dataset loaded successfully!")
df.head()
print("Shape:", df.shape)
print("Missing values:\n", df.isnull().sum())
df.drop("LoanID", axis=1, inplace=True)
cat_cols = df.select_dtypes(include='object').columns
le = LabelEncoder()
for col in cat_cols:
    df[col] = le.fit_transform(df[col])
df.head()
plt.figure(figsize=(10, 6))
sns.heatmap(df.corr(), annot=True, cmap="coolwarm")
plt.title("Feature Correlation")
plt.show()
sns.countplot(x="Default", data=df)
plt.title("Class Distribution")
plt.show()
X = df.drop("Default", axis=1)
y = df["Default"]
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2,
random_state=42)
scaler = StandardScaler()
X_train_scaled = scaler.fit_transform(X_train)
X_test_scaled = scaler.transform(X_test)
model = RandomForestClassifier(n_estimators=100, random_state=42)
model.fit(X_train_scaled, y_train)
y_pred = model.predict(X_test_scaled)
print("Classification Report:\n", classification_report(y_test, y_pred))
print("Confusion Matrix:\n", confusion_matrix(y_test, y_pred))
```

```
importances = model.feature_importances_
feat_names = X.columns
plt.figure(figsize=(10, 6))
sns.barplot(x=importances, y=feat_names)
plt.title("Feature Importance")
plt.xlabel("Importance Score")
plt.ylabel("Features")
plt.show()
def predict_new_user(data_dict):
    """
    Predict loan default for a new user data dictionary.
    Example:
    data_dict = {
        'Gender': 'Male',
        'Age': 35,
        'Income': 50000,
        ...
    }
    """
    new_data = pd.DataFrame([data_dict])
       for col in cat_cols:
        if col in new_data.columns:
            new_data[col] = le.transform(new_data[col])
    new_data = new_data[X.columns]
    new_data_scaled = scaler.transform(new_data)
    prediction = model.predict(new_data_scaled)[0]
    result = "❌ Likely to Default" if prediction == 1 else "✅ No Default
Risk"
    print(f"Prediction: {result}")
sample_user = {
    'Gender': 'Male',
    'Age': 35,
    'Income': 55000,
    'LoanAmount': 20000,
    'CreditScore': 700,
    'LoanTerm': 12,
    'DTIRatio': 0.4,
    'Education': 'Graduate',
    'Married': 'Yes',
    'Self_Employed': 'No',
```

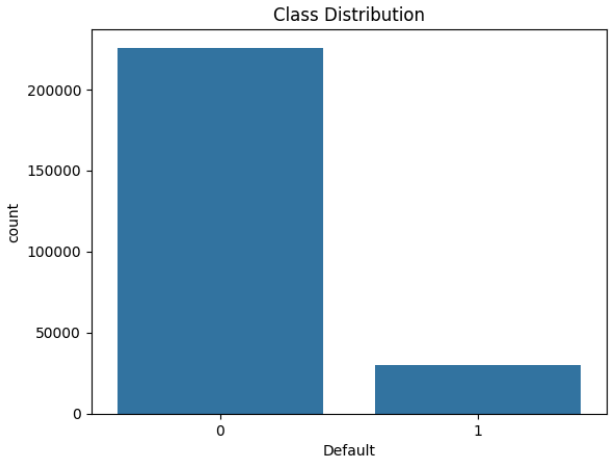
```
    'Property_Area': 'Urban',
    'Dependents': '0'
}
predict_new_user(sample_user)
```

Output/Result

1. Predict Loan Default.csv(text/csv) - 24834870 bytes, last modified: 4/18/2025 - 100% done
Saving 1. Predict Loan Default.csv to 1. Predict Loan Default (4).csv
Dataset loaded successfully!
Shape: (255347, 18)
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


Class Distribution
200000
150000
100000
50000
0
count
0
1
Default