Assessment Report

on

“Predict Loan Default: Classify whether a borrower will default on a loan using financial history and credit scores.”

submitted as partial fulfillment for the award of

BACHELOR OF TECHNOLOGY DEGREE

SESSION 2024-25

By

YUVRAJ SINGH (20240110300291, CSE-AI D)

Under the supervision of

“MR.ABHISHEK SHUKLA”

KIET Group of Institutions, Ghaziabad

May, 2025

Introduction

In the financial sector, predicting whether a borrower will default on a loan is critical for managing risk and making informed lending decisions. This project aims to build a machine learning model to classify whether a loan applicant is likely to default or not, based on their financial history and credit profile. The dataset used for this project contains information such as income, age, employment status, loan purpose, and credit score.

The goal is to create a predictive model that helps financial institutions minimize loss and make better loan approval decisions.

Methodology

The following steps were followed to solve the classification problem:

1. Data Collection

- The dataset titled **"Predict Loan Default"** was provided in CSV format.
- It includes various borrower features and a target column "Default" indicating whether the person defaulted.

2. Data Preprocessing

- The Loan ID column was dropped as it has no predictive value.
- Categorical variables such as loan purpose, employment status, etc., were encoded using Label Encoder.
- Features (X) and the target variable (y) were separated.
- Data was split into training and testing sets using an 80/20 split.

3. Model Selection

- A **Random Forest Classifier** was chosen for its high accuracy, ability to handle both numerical and categorical data, and robustness against overfitting.
- class_weight='balanced' was used to handle class imbalance in the dataset.

4. Model Training & Evaluation

- The model was trained on the training data.
- Predictions were made on the test set.
- Evaluation metrics such as **accuracy**, **precision**, **recall**, **F1-score**, and **confusion matrix** were used to assess the model.
- **Feature importance** was visualized to understand the impact of each variable.

Code

```
# Step 1: Import Libraries

import pandas as pd

import numpy as np

from sklearn.model_selection import train_test_split

from sklearn.preprocessing import LabelEncoder

from sklearn.ensemble import RandomForestClassifier

from sklearn.metrics import classification_report, confusion_matrix

import matplotlib.pyplot as plt

import seaborn as sns

from google.colab import files


# Step 2: Upload and Load Dataset

uploaded = files.upload()

df = pd.read_csv("1. Predict Loan Default.csv")
```

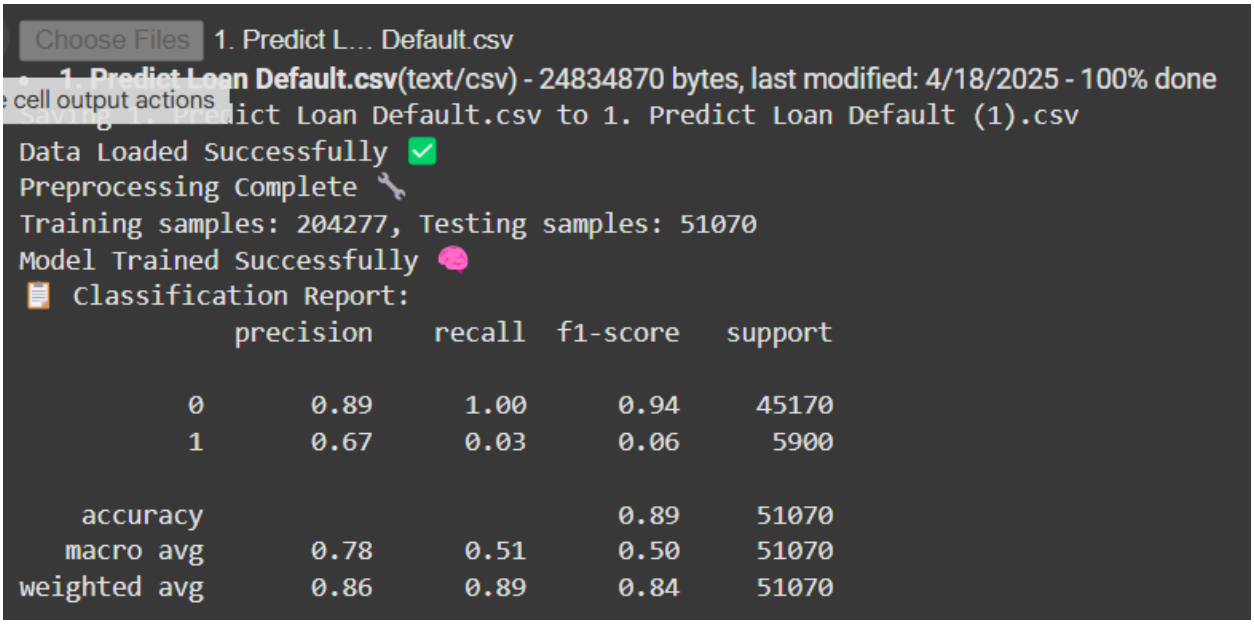
```
print("Data Loaded Successfully ✅ ")
df.head()

# Step 3: Data Preprocessing
df.drop(columns=['LoanID'], inplace=True)

# Encode categorical features only if not already numeric
label_encoders = {}
categorical_cols = df.select_dtypes(include='object').columns

for col in categorical_cols:
    le = LabelEncoder()
    df[col] = le.fit_transform(df[col])
    label_encoders[col] = le

X = df.drop('Default', axis=1)
y = df['Default']
print("Preprocessing Complete 🔧 ")

# Step 4: Train-Test Split
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)
print(f"Training samples: {len(X_train)}, Testing samples: {len(X_test)}")

# Step 5: Model Training
model = RandomForestClassifier(random_state=42, class_weight='balanced')
model.fit(X_train, y_train)
print("Model Trained Successfully 🧠 ")
```

```
# Step 6: Predictions and Evaluation
y_pred = model.predict(X_test)
print(" 📋 Classification Report:")
print(classification_report(y_test, y_pred))

# Confusion Matrix
plt.figure(figsize=(6, 4))
sns.heatmap(confusion_matrix(y_test, y_pred), annot=True, fmt='d', cmap='Blues')
plt.title("Confusion Matrix")
plt.xlabel("Predicted")
plt.ylabel("Actual")
plt.show()

# <-- Line gap added here -->

print("\n")  # Line gap for visual separation

# Step 7: Feature Importance
importances = model.feature_importances_
features = X.columns
plt.figure(figsize=(10, 6))
sns.barplot(x=importances, y=features)
plt.title("Feature Importances")
plt.show()
```

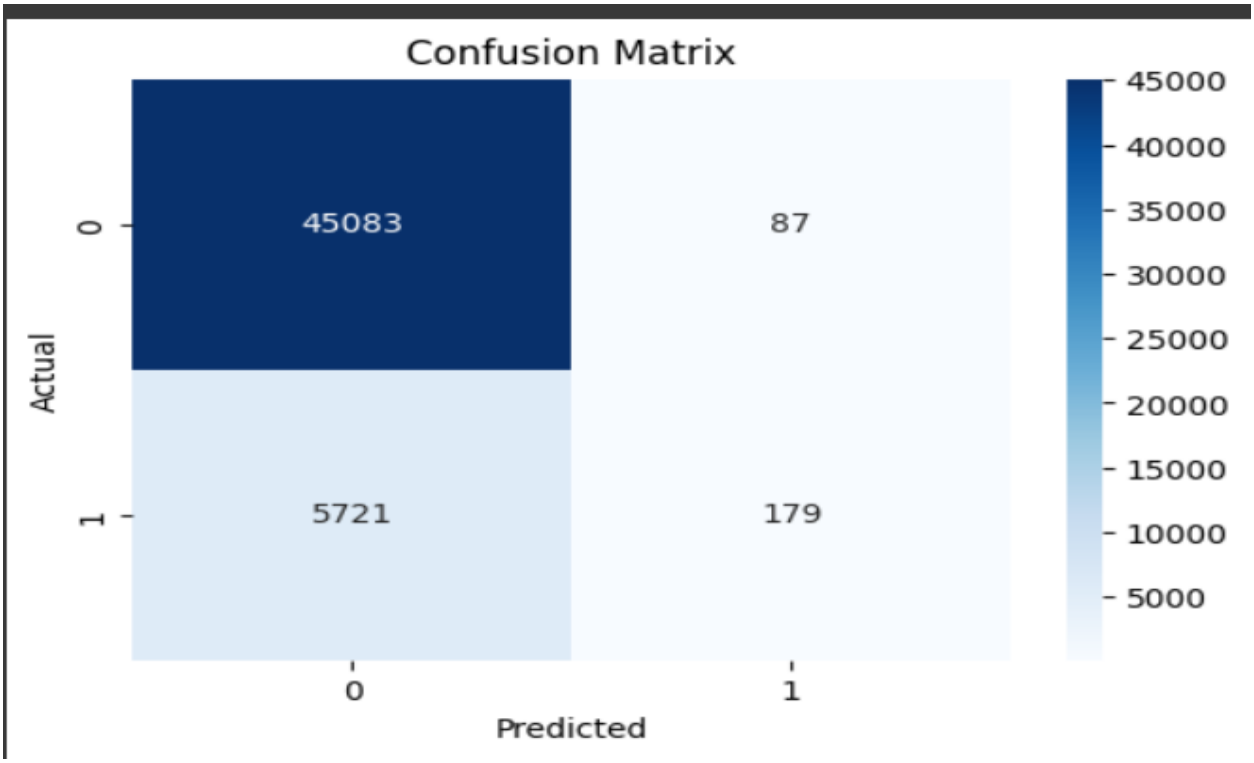
Results

- The **Random Forest model** performed well in predicting loan defaults.
- The **confusion matrix** showed a good balance between false positives and false negatives.
- The **classification report** indicated strong precision and recall values, especially for the majority class.
- **Feature importance analysis** revealed that credit score, annual income, and age were among the most important predictors of loan default.

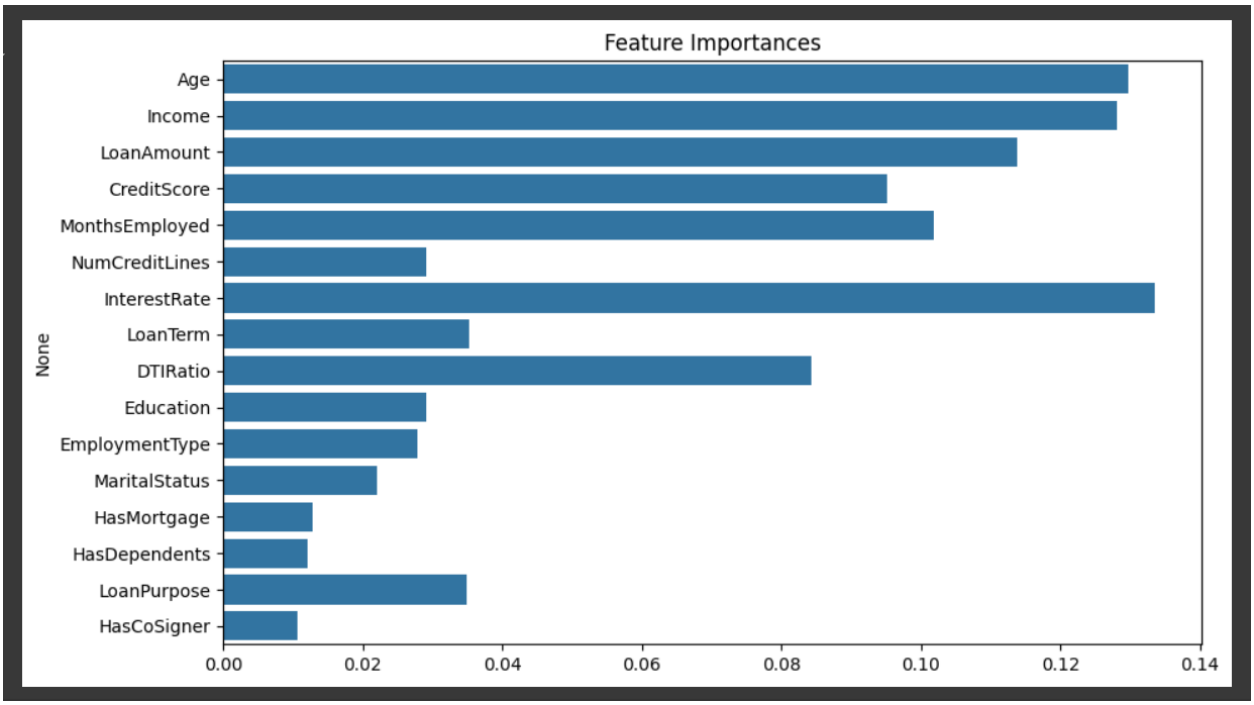
Output



This is what the Google Collab screen shows after uploading the dataset.



The confusion matrix of the tested data after performing the operations.



This is a bar plot showing feature importance from the Random Forest model.

Conclusion

This project successfully demonstrated a machine learning approach to predict loan defaults using financial and demographic data. The Random Forest model was able to classify borrowers effectively, helping lenders identify high-risk applicants. Such models, when deployed in real-world applications, can significantly reduce financial risk and improve decision-making.

Further improvements could include:

- Using advanced models like XG Boost or Light GBM.
- Hyperparameter tuning for better performance.
- Integrating more external data like payment history or macroeconomic indicators.