Assesment Report

on

“Predict Loan Default”

submitted as partial fulfillment for the award of

BACHELOR OF TECHNOLOGY DEGREE

SESSION 2024-25

in

CSEAI

By

Vishal Singh (202401100300281)

Under the supervision of

“Prof. Abhishek Shukla”

KIET Group of Institutions, Ghaziabad

Affiliated to

Dr. A.P.J. Abdul Kalam Technical University, Lucknow

(Formerly UPTU)

May, 2025

Introduction

The problem statement focuses on predicting whether a borrower will default on a loan based on their financial history and credit scores. This is a binary classification problem and is vital for banks and financial institutions to minimize risks. By applying machine learning techniques to this task, we can automate and improve the efficiency of loan approval systems. The dataset used contains historical data of loan applicants, including features such as credit score, income, loan amount, and more.

Methodology

To solve the loan default prediction problem, the following approach was used:

1. **Data Loading & Cleaning**: The dataset was loaded into a pandas DataFrame. Null values were checked and handled appropriately. Categorical features were label encoded.
2. **Feature Scaling**: The features were standardized using StandardScaler for better model performance.
3. **Train-Test Split**: The dataset was split into 70% training and 30% testing sets.
4. **Model Selection**: A Random Forest Classifier was chosen for its ability to handle non-linear relationships and feature importance analysis.
5. **Model Evaluation**: After training, the model was evaluated using metrics like Accuracy, Precision, Recall, and a Confusion Matrix. A heatmap of the confusion matrix was generated using Seaborn.

CODE:

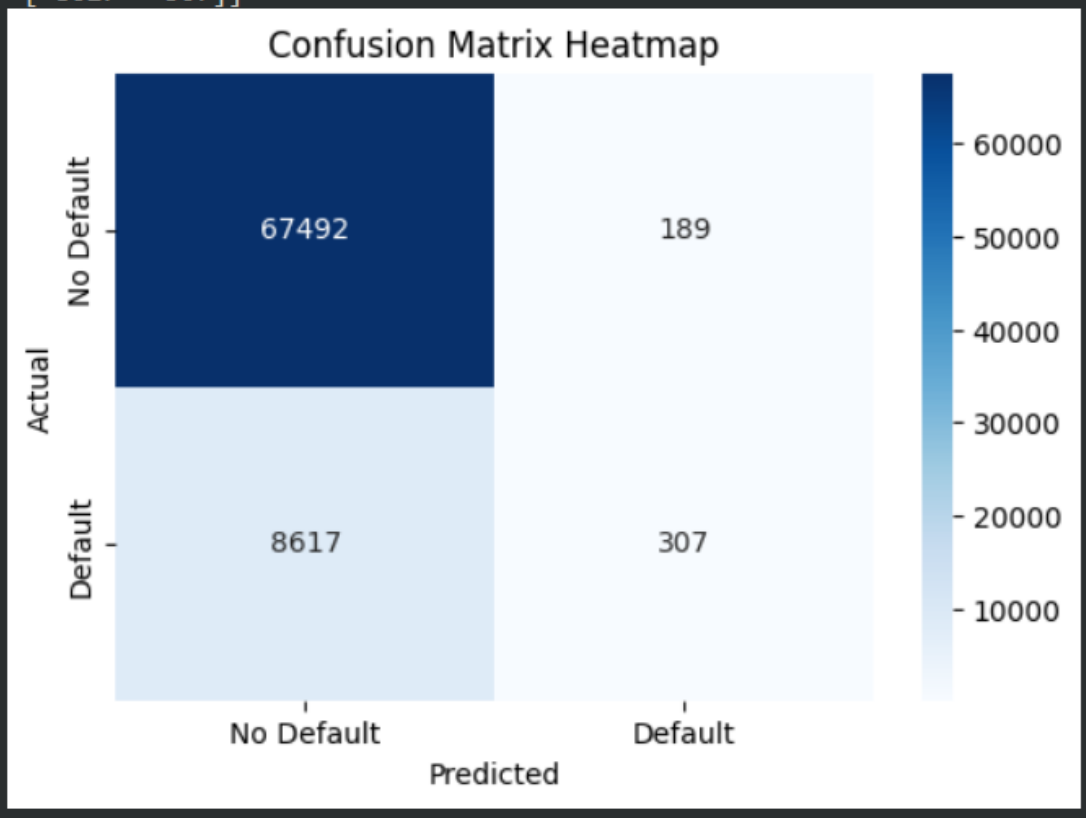
```
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import accuracy_score, precision_score, recall_score, confusion_matrix
import seaborn as sns
import matplotlib.pyplot as plt

# Load the dataset
df = pd.read_csv("1. Predict Loan Default.csv")

# Drop identifier column
df = df.drop(columns=["LoanID"])

# Convert categorical columns to numeric using one-hot encoding
df_encoded = pd.get_dummies(df, drop_first=True)

# Separate features and target
X = df_encoded.drop("Default", axis=1)
y = df_encoded["Default"]

# Standardize numerical features
scaler = StandardScaler()
```

```
X_scaled = scaler.fit_transform(X)

# Split into training and testing sets
X_train, X_test, y_train, y_test = train_test_split(X_scaled, y, test_size=0.3,
random_state=42)

# Train logistic regression model
log_reg = LogisticRegression(max_iter=1000)
log_reg.fit(X_train, y_train)

# Predict on test data
y_pred = log_reg.predict(X_test)

# Evaluation metrics
accuracy = accuracy_score(y_test, y_pred)
precision = precision_score(y_test, y_pred)
recall = recall_score(y_test, y_pred)

print("Accuracy:", accuracy)
print("Precision:", precision)
print("Recall:", recall)

# Confusion matrix
conf_matrix = confusion_matrix(y_test, y_pred)
```

```
print(conf_matrix)


# Plot heatmap
plt.figure(figsize=(6,4))
sns.heatmap(conf_matrix, annot=True, fmt="d", cmap="Blues", xticklabels=["No Default", "Default"], yticklabels=["No Default", "Default"])
plt.xlabel("Predicted")
plt.ylabel("Actual")
plt.title("Confusion Matrix Heatmap")
plt.show()
```

Output/Result

```
Accuracy: 0.8850466679720644
Precision: 0.6189516129032258
Recall: 0.0344016136261766
[[67492   189]
 [ 8617   307]]
```



References/Credits

- Python Libraries: pandas, seaborn, matplotlib, sklearn
- Tools Used: Google Colab, GitHub for code
- Special thanks to the course instructors and mentors for guidance on machine learning techniques.