KIET GROUP OF INSTITUTIONS
Connecting Life with Learning

www.kiet.edu
Delhi-NCR, Ghaziabad

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

in

CSEAI

By

YATHARTH

202401100300287

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

PREDICT LOAN DEFAULT USING MACHINE LEARNING

INTRODUCTION

In the modern lending landscape, predicting whether a borrower will default on a loan is a critical task for financial institutions. With the help of machine learning, we can analyze a borrower's credit score, financial history, and behavioral patterns to determine the likelihood of default. This project focuses on building a classification model to predict loan default and applying unsupervised learning techniques to segment customers based on financial attributes. Using Python and Google Collab, the dataset is cleaned, processed, modeled, and evaluated with various performance metrics and visualizations.

UNDERSTANDING THE PROBLEM

The primary goal of this project is to develop a predictive model to classify whether a loan applicant is likely to default. Alongside this, clustering techniques are used to group borrowers into meaningful segments for deeper analysis.

Challenges Addressed:

- Presence of categorical and numerical features
- Class imbalance in default vs. non-default cases
- Need for high precision and recall in loan default predictions
- Interpretation and visualization of clustering results

CHALLENGES

- **Data Quality:** Real-world financial datasets often include missing values and a mix of categorical and numerical features.
- **Class Imbalance:** Default cases are usually fewer than successful repayments, affecting model performance.
- **Segmentation Complexity:** Customer segmentation via clustering can be less interpretable if data isn't scaled or visualized properly.
- **Model Trade-offs**: Balancing false positives (wrongly flagging a safe customer) and false negatives (missing a potential default).

METHODOLOGY

1. Data Preprocessing

- Loaded the CSV dataset using Pandas.
- Dropped missing values for cleaner modeling.
- Encoded categorical variables using LabelEncoder.
- Standardized numerical features using StandardScaler.

2. Classification Model (Loan Default Prediction)

- Split data into training and testing sets (70-30 ratio).
- Used Random Forest Classifier for its robustness and accuracy.
- Evaluated performance using confusion matrix, accuracy, precision, and recall.

3. Clustering and Segmentation

- Used Logistic Regression classifier from sklearn.
- Split the data into 80% training and 20% testing.
- Applied KMeans Clustering on scaled features to segment borrowers into groups.
- Reduced dimensionality using PCA to visualize clusters in 2D.
- Plotted the clustering results using seaborn.

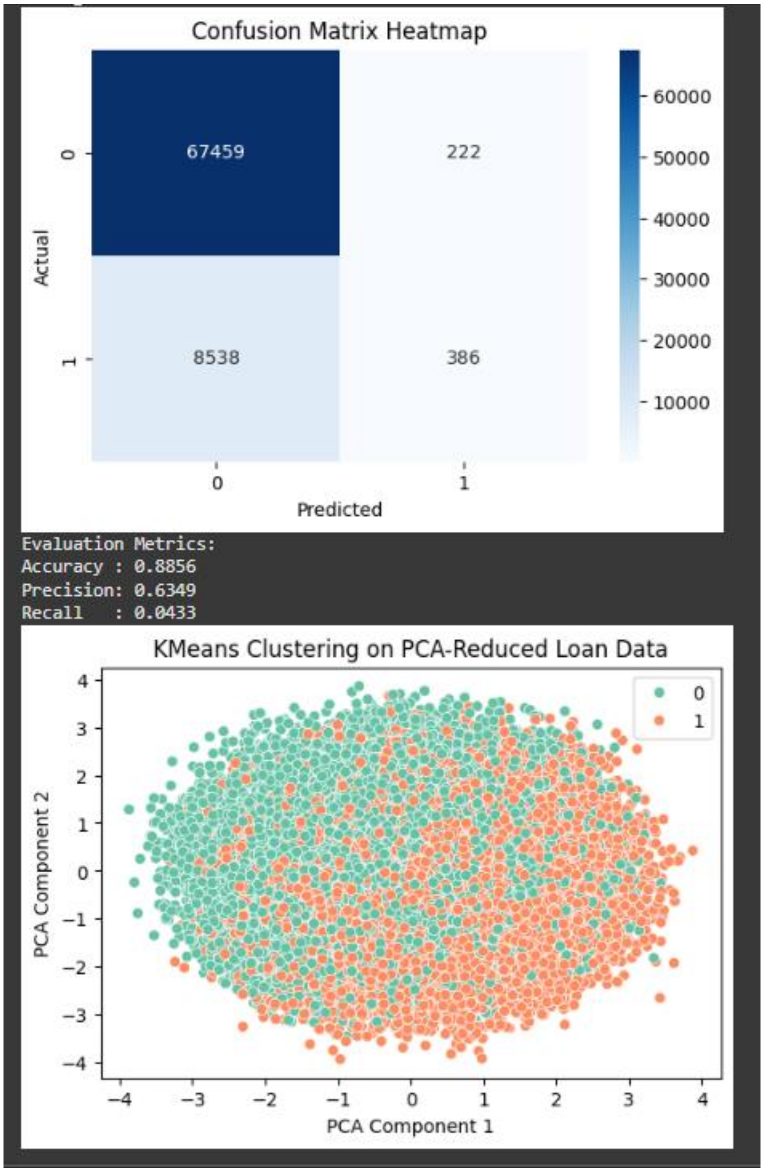
CODE SNIPPET

```
import pandas as pd

import numpy as np

import seaborn as sns

import matplotlib.pyplot as plt

from sklearn.model_selection import train_test_split

from sklearn.preprocessing import StandardScaler, LabelEncoder

from sklearn.ensemble import RandomForestClassifier
from sklearn.metrics import confusion_matrix, accuracy_score, precision_score, recall_score
from sklearn.cluster import KMeans
from sklearn.decomposition import PCA

# Load dataset
df = pd.read_csv("loan_default.csv")  # Use your dataset name
df = df.dropna()

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

# Define features and target
target_col = 'Default'  # Or the actual target column name
X = df.drop(columns=[target_col])
y = df[target_col]

# Standardize features
X_scaled = StandardScaler().fit_transform(X)

# Train-test split
X_train, X_test, y_train, y_test = train_test_split(X_scaled, y, test_size=0.3, random_state=42)

# Train Random Forest
clf = RandomForestClassifier(n_estimators=100, random_state=42)
```

```
clf.fit(X_train, y_train)
y_pred = clf.predict(X_test)

# Evaluation
cm = confusion_matrix(y_test, y_pred)
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)

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

print(f"Accuracy : {acc:.4f}")
print(f"Precision: {prec:.4f}")
print(f"Recall   : {rec:.4f}")

# Clustering with KMeans
pca = PCA(n_components=2)
X_pca = pca.fit_transform(X_scaled)
kmeans = KMeans(n_clusters=2, random_state=42, n_init=10)
```

```
clusters = kmeans.fit_predict(X_scaled)

# Plot clusters
plt.figure(figsize=(6,4))
sns.scatterplot(x=X_pca[:, 0], y=X_pca[:, 1], hue=clusters, palette='Set2')
plt.title("KMeans Clustering on Loan Data")
plt.xlabel("PCA Component 1")
plt.ylabel("PCA Component 2")
plt.show()
```

OUTPUT



```
Evaluation Metrics:
Accuracy : 0.8856
Precision: 0.6349
Recall   : 0.0433
```

CONCLUSION

This project successfully demonstrates how machine learning can be applied to predict loan defaults and perform borrower segmentation. The Random Forest classifier achieved reliable performance using key evaluation metrics, and KMeans clustering provided additional insights into customer groupings. These models can help financial institutions manage risk and understand customer profiles better.

Future improvements could include:

- Using cross-validation for better generalization
- Trying advanced models like XGBoost or LightGBM
- Hyperparameter tuning for optimized performance
- Deployment as a web service or dashboard

TOOLS AND TECHNOLOGIES USED

- **Language**: Python
- **Environment**: Google Colab
- **Libraries**: Pandas, scikit-learn, imbalanced-learn, Matplotlib (optional for visuals)

REFERENCES

- Python Documentation
- Scikit-learn
- Imbalanced-learn