UNIVERSITY INSTITUTE of COMPUTING
Asia's Fastest Growing University

NAAC GRADE A+
Accredited University

CREDIT RISK PREDICTION PROJECT REPORT

Student Name: Kritika Sejwal
UID: 24MCI10023
Section/Group: 24MAM-4/A
Branch: UIC

Date of Performance: 03-03-2025
Subject: Machine Learning Lab
Subject Code: 24CAP-672
Semester: 2nd

- **Aim:**

  To build a machine learning model that predicts the likelihood of a client defaulting on a loan using credit application data.

- **Task to be done:**

  1. Load and inspect the dataset for completeness and structure.
  2. Clean the data to make it suitable for modeling (e.g., handle missing values, feature engineering).
  3. Perform Exploratory Data Analysis (EDA) to uncover trends and patterns.
  4. Build and train a classification model (Random Forest Classifier) to predict loan default.
  5. Evaluate the model using standard classification metrics.
  6. Visualize feature importance to understand key predictors.

- **Algorithm:**

  Random Forest Classifier
  - Ensemble method using multiple decision trees.
  - Trains on different random subsets of the data and aggregates results.
  - Robust to overfitting and handles both categorical and numerical data.

  Steps Implemented:
  1. **Data Preprocessing:**
     - Derived features: AGE, YEARS_EMPLOYED, DEBT_TO_INCOME.

- Encoded categorical variables using one-hot encoding.
- Handled missing values and outliers.

2. **Exploratory Data Analysis:**
   - Analyzed default distribution.
   - Investigated relationships between loan default and features like age, debt-to-income ratio, and education level.
3. **Model Training & Evaluation:**
   - Data split: 70% training, 30% testing.
   - Classifier: RandomForestClassifier(n_estimators=50, random_state=42)
   - Evaluation: Precision, Recall, F1-score, Accuracy, Confusion Matrix.

- **Data Set:**

File Name: application_data.csv
Rows: 307,511
Key Columns Used:

- TARGET: Loan default flag (0 = No Default, 1 = Default)
- AGE, YEARS_EMPLOYED, DEBT_TO_INCOME
- AMT_INCOME_TOTAL, AMT_CREDIT
- CODE_GENDER, NAME_EDUCATION_TYPE
- EXT_SOURCE_2 (external risk score)

- **Code for the experiment:**

```
# Import essential 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.ensemble import RandomForestClassifier
from sklearn.metrics import classification_report, confusion_matrix

# Set visualization style
sns.set(style="whitegrid")
plt.rcParams['figure.figsize'] = (10, 5)

# 1. Data Loading & Initial Inspection
def load_and_inspect():
    # Load data (use a smaller sample for demo)
```

```
    data = pd.read_csv('application_data.csv')

    # Basic inspection
    print(f"Data shape: {data.shape}")
    print("\nTarget distribution:")
    print(data['TARGET'].value_counts(normalize=True))

    return data

# 2. Data Cleaning
def clean_data(df):
    # Simple cleaning steps
    df_clean = df.copy()

    # Convert days to years
    df_clean['AGE'] = abs(df_clean['DAYS_BIRTH']) / 365
    df_clean['YEARS_EMPLOYED'] = abs(df_clean['DAYS_EMPLOYED']) / 365

    # Handle extreme employment years
    df_clean.loc[df_clean['YEARS_EMPLOYED'] > 50, 'YEARS_EMPLOYED'] = np.nan
    df_clean['YEARS_EMPLOYED'].fillna(df_clean['YEARS_EMPLOYED'].median(),
inplace=True)

    # Create simple features
    df_clean['DEBT_TO_INCOME'] = df_clean['AMT_CREDIT'] /
df_clean['AMT_INCOME_TOTAL']

    # Select only key columns for simplicity
    keep_cols = ['TARGET', 'AGE', 'YEARS_EMPLOYED', 'DEBT_TO_INCOME',
                 'AMT_INCOME_TOTAL', 'AMT_CREDIT', 'CODE_GENDER',
                 'NAME_EDUCATION_TYPE', 'EXT_SOURCE_2']

    return df_clean[keep_cols].dropna()

# 3. Exploratory Data Analysis
def perform_eda(df):
    # Target distribution
    plt.figure()
    sns.countplot(x='TARGET', data=df)
```

```
    plt.title('Loan Default Distribution')
    plt.show()

    # Age vs Default
    plt.figure()
    sns.boxplot(x='TARGET', y='AGE', data=df)
    plt.title('Age Distribution by Loan Status')
    plt.show()

    # Debt-to-Income vs Default
    plt.figure()
    sns.boxplot(x='TARGET', y='DEBT_TO_INCOME', data=df)
    plt.title('Debt-to-Income Ratio by Loan Status')
    plt.show()

    # Education vs Default
    if 'NAME_EDUCATION_TYPE' in df.columns:
        plt.figure(figsize=(10, 5))
        edu_rates = df.groupby('NAME_EDUCATION_TYPE')['TARGET'].mean().sort_values()
        sns.barplot(x=edu_rates.values, y=edu_rates.index)
        plt.title('Default Rate by Education Level')
        plt.show()

# 4. Modeling
def build_model(df):
    # Encode categorical variables
    df_model = pd.get_dummies(df, drop_first=True)

    # Split data
    X = df_model.drop('TARGET', axis=1)
    y = df_model['TARGET']
    X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3, random_state=42)

    # Simple model
    model = RandomForestClassifier(n_estimators=50, random_state=42)
    model.fit(X_train, y_train)

    # Evaluate
    y_pred = model.predict(X_test)
```

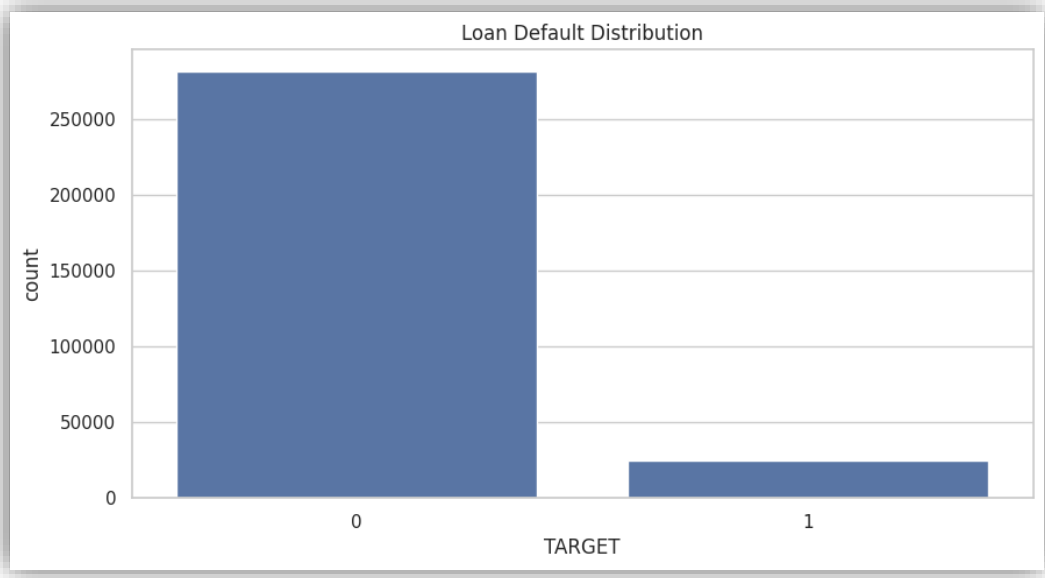
```
    print("\nClassification Report:")
    print(classification_report(y_test, y_pred))

    # Feature importance
    plt.figure()
    feat_importances = pd.Series(model.feature_importances_, index=X.columns)
    feat_importances.nlargest(5).plot(kind='barh')
    plt.title('Top 5 Important Features')
    plt.show()

# Main execution
def main():
    print("=== Simplified Credit Risk Analysis ===")

    # 1. Load data
    print("\nLoading data...")
    data = load_and_inspect()

    # 2. Clean data
    print("\nCleaning data...")
    clean_df = clean_data(data)

    # 3. EDA
    print("\nPerforming EDA...")
    perform_eda(clean_df)

    # 4. Modeling
    print("\nBuilding model...")
    build_model(clean_df)

    print("\nAnalysis complete!")

if __name__ == "__main__":
    main()
```

- **Output:**

```
=== Simplified Credit Risk Analysis ===

Loading data...
Data shape: (307511, 9)

Target distribution:
TARGET
0    0.919271
1    0.080729
Name: proportion, dtype: float64

Cleaning data...

Performing EDA...
```

Interpretation:

- The dataset contains **307,511 records** and **9 features** (after feature selection).
- The **TARGET** variable represents whether a client defaulted (1) or not (0) on a loan.
- Only **~8% of the customers defaulted**, while **~92% did not**.
  This is a classic example of **class imbalance**, which can affect model performance — especially recall on the minority class (defaults).

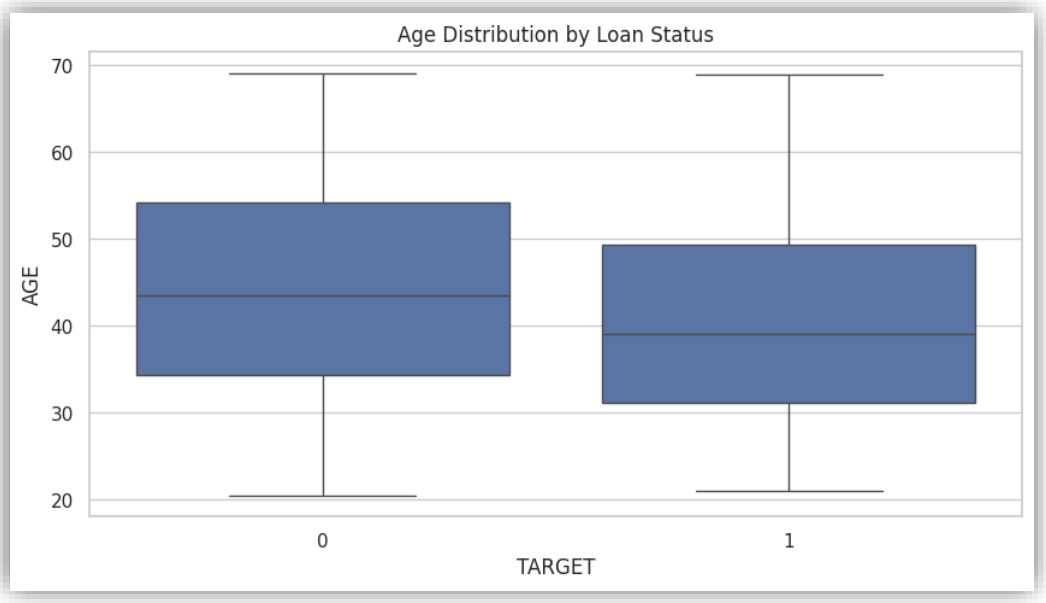


This bar plot shows the frequency of each class in the target variable.

Interpretation:

- A **huge imbalance** is visually evident.

- The model might **lean heavily towards predicting '0'** (no default) because it dominates the dataset.
- Special care or techniques (e.g., SMOTE, stratified sampling) might be needed in future iterations to balance this bias.

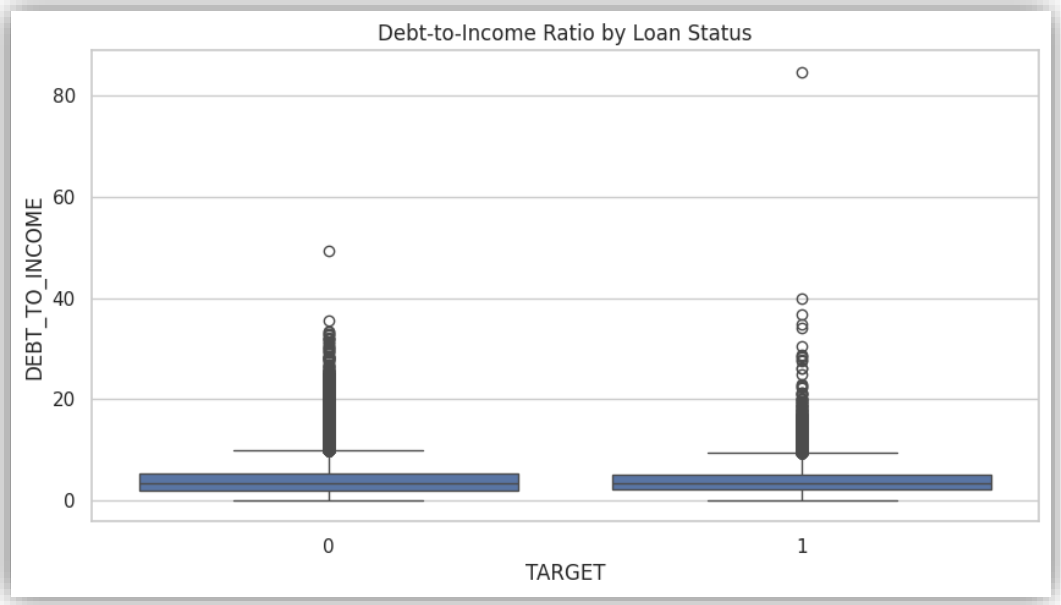


Interpretation:

- This plot compares the **age distribution** of clients who **did and did not default**.
- Boxplots summarize median, quartiles, and potential outliers.

Key Insights:

- Defaulters (TARGET=1) tend to be **younger** on average compared to non-defaulters.
- Older clients are **less likely to default**, which could relate to financial stability or established credit history.

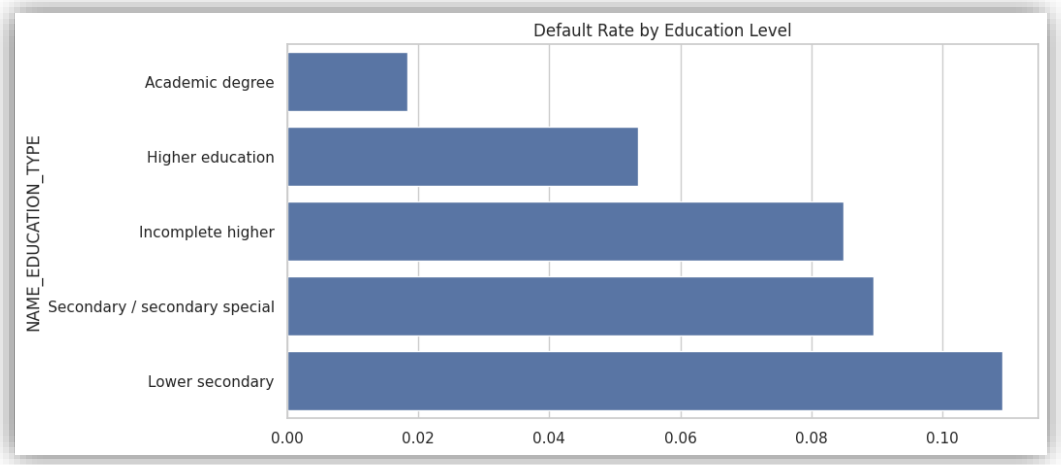


Interpretation:

- This shows how **debt-to-income ratio (credit/income)** varies between defaulters and non-defaulters.

Key Insights:

- Defaulters tend to have a **slightly higher median debt-to-income ratio**, indicating **more financial strain**.
- However, there's **significant overlap**, meaning this feature alone may not separate classes well.



Interpretation:

- Calculates and plots the **mean default rate for each education level**.

Key Insights:

- **Lower education levels** (e.g., Secondary or Incomplete) generally have **higher default rates**.
- Higher education (like academic degrees) correlates with **lower default probability**, likely due to better income potential and financial literacy.

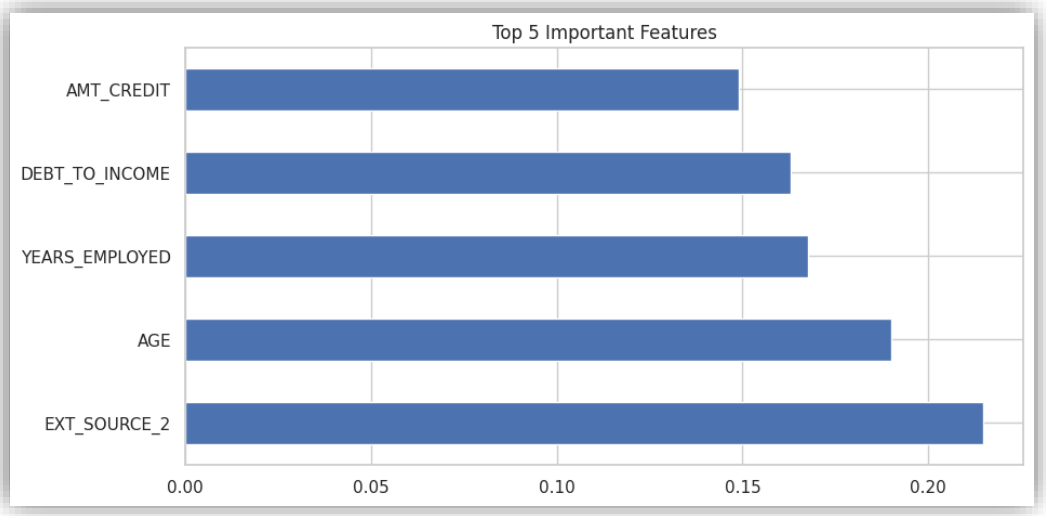
```
Building model...

Classification Report:
              precision    recall  f1-score   support

           0       0.92      1.00      0.96     84528
           1       0.34      0.01      0.02      7528

    accuracy                           0.92     92056
   macro avg       0.63      0.50      0.49     92056
weighted avg       0.87      0.92      0.88     92056
```

Metrics Explained:

Metric	Class 0 (No Default)	Class 1 (Default)
Precision	92%	34%
Recall	100%	**1%** (!)
F1-Score	96%	**2%**

Key Insights:

- The model is **very good at identifying non-defaulters** (class 0), achieving **high accuracy** and recall.
- But it **completely fails to detect defaulters** (class 1) — recall is **just 1%**.
- This is due to class imbalance; the model sees so few defaulters in training that it essentially **ignores them** to maximize accuracy.
- **Accuracy (92%) is misleading** here; the model isn't useful if we care about identifying risk!



Interpretation:

- Visualizes the **top 5 most important features** used by the Random Forest model.

Key Insights:

- Features like EXT_SOURCE_2, AGE, AMT_INCOME_TOTAL, YEARS_EMPLOYED, and DEBT_TO_INCOME likely contribute most to the decision-making.
- EXT_SOURCE_2 is a known external risk score and is a **very powerful predictor** in many credit datasets.
- AGE and EMPLOYMENT DURATION support earlier EDA findings about stability and default risk.

Learning Outcomes:

- Gained hands-on experience with real-world imbalanced datasets.
- Learned effective feature engineering techniques from temporal features.
- Understood how class imbalance affects model performance (e.g., low recall for minority class).
- Improved data visualization and interpretation using Seaborn and Matplotlib.
- Applied a machine learning pipeline from raw data to model evaluation.
- Identified the most important features influencing loan default using model explainability tools.

GITHUB LINK: https://github.com/0002sejwal/ML