**Credit Default Prediction Using Logistic Regression**

**Project Report**

**Submitted By:**

**Madhavan G**

**Date:**

**August 25, 2024**

**Institution:**

***Vels institute of science and technology and advanced studies***

**Project: Credit Default Prediction Using Logistic Regression**

**1. Project Overview**

**Objective**

The objective of this project is to build a logistic regression model to predict whetherr a person will default on their credit, based on features such as income, age, loan amount, and the loan-to-income ratio.

**Dataset**

We are using a dataset from [GitHub](https://github.com/ybifoundation/Dataset/raw/main/Credit%20Default.csv), which contains information about 2000 individuals, including their income, age, loan amount, loan-to-income ratio, and whether they defaulted on their credit.

**2. Extract**

**Data Overview**

* **Income**: The annual income of the individual.
* **Age**: The age of the individual.
* **Loan**: The loan amount taken by the individual.
* **Loan to Income**: The ratio of the loan amount to the individual's income.
* **Default**: A binary variable indicating whether the individual defaulted on their credit (1) or not (0).

The dataset consists of 2000 records with no missing values.

**3. Explanation**

**Methodology**

1. **Data Preprocessing**:
   * The data is loaded and explored to understand its structure and summary statistics.
   * The target variable (Default) is separated from the features (Income, Age, Loan, Loan to Income).
2. **Model Selection**:
   * A logistic regression model is chosen due to its effectiveness in binary classification problems like this one.
3. **Model Training**:
   * The data is split into a training set (70%) and a testing set (30%).
   * The logistic regression model is trained on the training data.
4. **Model Evaluation**:
   * The model's predictions on the test data are evaluated using a confusion matrix, accuracy score, and classification report.

**4. Coding**

python

Copy code

# Step 1: Import necessary libraries

import pandas as pd

from sklearn.model\_selection import train\_test\_split

from sklearn.linear\_model import LogisticRegression

from sklearn.metrics import confusion\_matrix, accuracy\_score, classification\_report

# Step 2: Import the data

url = 'https://github.com/ybifoundation/Dataset/raw/main/Credit%20Default.csv'

default = pd.read\_csv(url)

# Display the first few rows of the dataset

print(default.head())

# Check the structure and summary statistics of the dataset

print(default.info())

print(default.describe())

# Step 3: Define target variable (y) and features (X)

y = default['Default']

X = default.drop(columns=['Default'])

# Step 4: Split the data into training and testing sets

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, train\_size=0.7, random\_state=2529)

# Step 5: Select and train the logistic regression model

model = LogisticRegression()

model.fit(X\_train, y\_train)

# Step 6: Make predictions on the test set

y\_pred = model.predict(X\_test)

# Step 7: Evaluate the model

# Confusion Matrix

conf\_matrix = confusion\_matrix(y\_test, y\_pred)

print("Confusion Matrix:")

print(conf\_matrix)

# Accuracy Score

accuracy = accuracy\_score(y\_test, y\_pred)

print(f"Accuracy: {accuracy:.2f}")

# Classification Report

class\_report = classification\_report(y\_test, y\_pred)

print("Classification Report:")

print(class\_report)

**5. Output**

**Sample Output from the Code**

* **First Few Rows of the Dataset**:

yaml

Copy code

Income Age Loan Loan to Income Default

0 66155.92510 59.017015 8106.532131 0.122537 0

1 34415.15397 48.117153 6564.745018 0.190752 0

2 57317.17006 63.108049 8020.953296 0.139940 0

3 42709.53420 45.751972 6103.642260 0.142911 0

4 66952.68885 18.584336 8770.099235 0.130990 1

* **Confusion Matrix**:

lua

Copy code

Confusion Matrix:

[[506 13]

[ 17 64]]

* **Accuracy**:

makefile

Copy code

Accuracy: 0.95

* **Classification Report**:

markdown

Copy code

precision recall f1-score support

0 0.97 0.97 0.97 519

1 0.83 0.79 0.81 81

accuracy 0.95 600

macro avg 0.90 0.88 0.89 600

weighted avg 0.95 0.95 0.95 600

**6. Conclusion**

The logistic regression model performed well on the test set, achieving an accuracy of 95%. The model was particularly effective at predicting non-default cases (precision and recall of 0.97). However, the model's performance was slightly lower for default cases, with a precision of 0.83 and a recall of 0.79. This slight imbalance in precision and recall for the default class suggests that while the model is reliable overall, it may occasionally miss predicting defaults correctly.

**Recommendations**

* **Address Class Imbalance**: Consider techniques like SMOTE or adjusting class weights to improve the model's performance on the minority class (default).
* **Feature Engineering**: Explore adding or transforming features to improve model accuracy and interpretability.
* **Model Comparison**: Evaluate other models, such as Random Forests or Support Vector Machines, to compare performance.

This project provides a solid foundation in using logistic regression for binary classification, with clear steps from data extraction to model evaluation and interpretation.