CREDIT SCORE PREDICTION

Harshita Sharma

Roll No.- 46

CSE(AI)-B

Uni Roll no.-202403001100120

KIET Group Of Institutions

**2. Introduction**

**What is Credit Score Prediction?**

Credit score prediction is the process of using historical financial data to determine whether a person has a good or bad credit score. A credit score is a numerical representation of a person’s creditworthiness, which helps banks and financial institutions make loan decisions.

**Why is Credit Score Prediction Important?**

* Helps banks decide loan approvals
* Reduces financial risk
* Provides insights into creditworthiness
* Prevents fraudulent activities

**Factors Affecting Credit Score:**

1. **Payment History** – Whether payments were made on time
2. **Credit Utilization** – Percentage of credit used
3. **Length of Credit History** – Duration of credit account usage
4. **Types of Credit** – Loans, mortgages, credit card

**3. Methodology**

**Step 1: Data Collection**

* The dataset consists of age, income, loan amount, credit history, and credit score.
* The data is loaded using Pandas.

**Step 2: Data Preprocessing**

* Handling missing values
* Converting categorical data into numerical form
* Normalizing the dataset using StandardScaler

**Step 3: Model Training**

* Logistic Regression is used for classification.
* The dataset is split into training and testing sets.

**Step 4: Model Evaluation**

* Accuracy score is calculated.
* Confusion matrix is used to visualize predictions.

**Step 5: Prediction**

* The trained model is used to predict new credit scores.

TYPED CODE:

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.preprocessing import StandardScaler

from sklearn.linear\_model import LogisticRegression

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

# 1️⃣ Load the dataset (Replace with your actual dataset)

df = pd.read\_csv("credit\_data.csv")

# 2️⃣ Select Features and Target

X = df[['CustomerID','Age', 'Income', 'LoanAmount', 'CreditScore']]  # Features

y = df['CreditScore']  # Target (1 = Good, 0 = Bad)

# 3️⃣ Visualize Credit Score Distribution

plt.figure(figsize=(6, 4))

sns.countplot(x=y, palette="pastel")

plt.title("Credit Score Distribution")

plt.xlabel("CreditScore (0 = Bad, 1 = Good)")

plt.ylabel("Count")

plt.show()

# 4️⃣ Visualize Income Distribution

plt.figure(figsize=(8, 5))

sns.histplot(df['Income'], bins=30, kde=True, color='blue')

plt.title("Income Distribution")

plt.xlabel("Income")

plt.ylabel("Frequency")

plt.show()

# 5️⃣ Split Data into Training & Testing Sets

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size=0.2, random\_state=42)

# 6️⃣ Normalize Data

scaler = StandardScaler()

X\_train = scaler.fit\_transform(X\_train)

X\_test = scaler.transform(X\_test)

# 7️⃣ Train Model

model = LogisticRegression()

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

# 8️⃣ Make Predictions

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

# 9️⃣ Confusion Matrix Visualization

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

plt.figure(figsize=(6, 4))

sns.heatmap(conf\_matrix, annot=True, fmt="d", cmap="Blues", xticklabels=['Bad', 'Good'], yticklabels=['Bad', 'Good'])

plt.title("Confusion Matrix")

plt.xlabel("Predicted")

plt.ylabel("Actual")

plt.show()

# 🔟 Show Model Accuracy

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

print("Model Accuracy:", accuracy)

# 📊 Bar Chart for Model Accuracy

plt.figure(figsize=(5, 4))

plt.bar(["Accuracy"], [accuracy], color="green")

plt.ylim(0, 1)

plt.ylabel("Score")

plt.title("Model Accuracy")

plt.show()





