Problem Statement:-

Credit Score Prediction

NAME:- Yash Mishra

University Roll No:- 202401100300285

Class Roll No:- 65

Branch:- CSE_AI

Section:- D

Introduction

Credit score prediction is crucial for banks and financial institutions to assess a person's creditworthiness. The goal of this project is to predict an individual's credit score based on their **age, income, and loan amount**. This helps in determining loan approvals, credit limits, and interest rates. The problem is formulated as a **regression problem**, where the model predicts a numerical credit score.

Methodology

1. **Data Preprocessing:** Removed unnecessary columns, handled missing values, and standardized features.
2. **Model Selection:** Used **Linear Regression**, suitable for small datasets.
3. **Training & Testing:** Split data into **80% training** and **20% testing**.
4. **Evaluation:** Used **Mean Absolute Error (MAE), Mean Squared Error (MSE), and R^2 Score** to assess performance.
5. **Prediction:** The model predicts credit scores for new customers based on given inputs.

Code

```
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler
from sklearn.linear_model import LinearRegression
from sklearn.metrics import mean_absolute_error, mean_squared_error, r2_score

# Load the dataset
df = pd.read_csv("credit_data.csv")

# Drop unnecessary columns
df = df.drop(columns=["CustomerID"])

# Define Features (X) and Target Variable (y)
X = df.drop(columns=["CreditScore"])
y = df["CreditScore"]

# 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)

# Scale features
```

```
scaler = StandardScaler()
X_train_scaled = scaler.fit_transform(X_train)
X_test_scaled = scaler.transform(X_test)

# Train the model
model = LinearRegression()
model.fit(X_train_scaled, y_train)

# Make predictions
y_pred = model.predict(X_test_scaled)

# Evaluate the model
mae = mean_absolute_error(y_test, y_pred)
mse = mean_squared_error(y_test, y_pred)
r2 = r2_score(y_test, y_pred)

print("Mean Absolute Error (MAE):", mae)
print("Mean Squared Error (MSE):", mse)
print("R² Score:", r2)

# Predict for a new customer
new_customer = pd.DataFrame([[30, 50000, 20000]],
columns=X.columns)
new_customer_scaled = scaler.transform(new_customer)
```

```
predicted_score = model.predict(new_customer_scaled)
print("Predicted Credit Score:", predicted_score[0])
```

Output/Result

Mean Absolute Error (MAE): 265.03404961567776

Mean Squared Error (MSE): 77039.28428102133

R^2 Score: -9.446447294345873

Predicted Credit Score: 651.0666706564242

Mean Absolute Error (MAE): 265.03404961567776

Mean Squared Error (MSE): 77039.28428102133

R^2 Score: -9.446447294345873

Predicted Credit Score: 651.0666706564242

References/Credits

- Dataset: Provided dataset "credit_data.csv"
- Libraries Used: Pandas, Scikit-learn
- Website used: ChatGPT