

# SUBJECT:INTRODUCTION TO AI.

## Report Title

CREDIT SCORE PREDICTOR: CLEAN  
AND TRANSFORM FINANCIAL DATA  
TO IMPROVE CREDIT RISK  
ASSESSMENT MODELS.

Name: [kushagra rastogi]

Roll No.: 202401100400113]

Institution: [KIET group of institution]

CLASS ROLL NO.=40

# Credit Score Data Preprocessing and Analysis Report.

## Introduction

Credit scoring is a crucial aspect of financial decision-making, used by banks and lenders to evaluate an individual's creditworthiness. The objective of this project is to preprocess and analyze a credit score dataset to ensure it is clean, normalized, and ready for machine learning models. This involves handling missing values, normalizing numerical features, encoding categorical variables, and visualizing key trends in the data.

## **Methodology**

### **Step 1: Data Upload and Inspection**

- The dataset was uploaded and read into a Pandas DataFrame.
- Basic information about the dataset, such as column names, data types, and missing values, was displayed.

### **Step 2: Handling Missing Values**

- Missing values in numeric columns were filled with the **median** value to prevent data distortion.
- Missing categorical values were filled using the **mode** (most frequent value).

### **Step 3: Feature Scaling and Normalization**

- Numerical features were scaled using **MinMaxScaler**, transforming all values into the range [0,1] to ensure uniformity.

### **Step 4: Encoding Categorical Variables**

- Categorical columns (e.g., Credit History, Credit Score) were converted into numerical form using **Label Encoding**.

## Step 5: Data Visualization

- **Credit Score Distribution:** A bar plot was used to visualize the distribution of credit scores.
- **Correlation Heatmap:** A heatmap was generated to show correlations between features.
- **Loan Amount vs. Income:** A scatter plot was created to analyze the relationship between **Loan Amount** and **Income** based on credit score categories.

## Step 6: Data Preparation for Machine Learning

- The dataset was cleaned and transformed, making it ready for model training.
- The 'ID' column was dropped as it does not contribute to predictions.
- Features (X) and target variable (y) were separated.

### Code

```
# Step 1: Install and Import Required Libraries
import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
from sklearn.preprocessing import MinMaxScaler,
LabelEncoder
from google.colab import files

# Step 2: Ask User to Upload CSV File
print("📁 Please upload the 'credit_score_data.csv' file.")
```

```
uploaded = files.upload()

# Step 3: Read the uploaded CSV file
file_name = list(uploaded.keys())[0] # Get the uploaded file name
df = pd.read_csv(file_name)

# Step 4: Display basic info about the dataset
print("\n🔍 First 5 rows of the dataset:")
print(df.head())

print("\n📊 Dataset Info:")
print(df.info())

print("\n☑️ Checking for Missing Values:")
print(df.isnull().sum())

# Step 5: Handle Missing Values (if any)
df.fillna(df.median(numeric_only=True), inplace=True) # Fill missing values with median

# Step 6: Normalize Numeric Features (Scaling)
scaler = MinMaxScaler()
num_cols = ["Age", "Income", "Loan_Amount",
"Loan_Duration", "Debt_to_Income", "Number_of_Loans",
"Late_Payments"]
df[num_cols] = scaler.fit_transform(df[num_cols])

# Step 7: Encode Categorical Features
```

```
df["Credit_History"] =  
df["Credit_History"].astype("category").cat.codes  
df["Credit_Score"] =  
df["Credit_Score"].astype("category").cat.codes # Target  
variable
```

## # Step 8: Data Visualization

```
# ◇ Credit Score Distribution  
plt.figure(figsize=(6,4))  
sns.countplot(x=df["Credit_Score"], palette="viridis")  
plt.title("Credit Score Distribution")  
plt.xlabel("Credit Score (0 = Bad, 1 = Average, 2 = Good)")  
plt.ylabel("Count")  
plt.show()
```

```
# ◇ Correlation Heatmap  
plt.figure(figsize=(10,6))  
sns.heatmap(df.corr(), annot=True, cmap="coolwarm",  
linewdiths=0.5)  
plt.title("Feature Correlation Heatmap")  
plt.show()
```

```
# ◇ Loan Amount vs. Income (Scatter Plot)  
plt.figure(figsize=(8,5))  
sns.scatterplot(x=df["Income"], y=df["Loan_Amount"],  
hue=df["Credit_Score"], palette="coolwarm")  
plt.title("Loan Amount vs. Income")  
plt.xlabel("Income (Normalized)")
```

```
plt.ylabel("Loan Amount (Normalized)")  
plt.show()  
  
# Step 9: Prepare Data for Model Training  
X = df.drop(["ID", "Credit_Score"], axis=1) # Features  
y = df["Credit_Score"] # Target  
  
print("\n☑ Data is cleaned, transformed, and ready for  
machine learning models!")
```

```
# Optional: Save cleaned data  
df.to_csv("cleaned_credit_score_data.csv", index=False)  
print("\n📁 Cleaned dataset saved as  
'cleaned_credit_score_data.csv'.")
```

## Output/Result

```

📁 Please upload the 'credit_score_data.csv' file.
Choose Files credit_score_data.csv
• credit_score_data.csv(text/csv) - 36620 bytes, last modified: 3/10/2025 - 100% done
Saving credit_score_data.csv to credit_score_data (1).csv

🔍 First 5 rows of the dataset:
   ID  Age  Income  Loan_Amount  Loan_Duration  Credit_History \
0   1    18     125724        9392           54              1
1   2    54     104563       33843           30              0
2   3    60     102528       41954           14              0
3   4    28     30879        16072           10              1
4   5    52     88309        9267            25              1

   Debt_to_Income  Number_of_Loans  Late_Payments  Credit_Score
0          45.97             10                 4              1
1          25.05              0                19              2
2          27.39              0                 5              0
3          44.37              2                19              1
4          15.14              2                 5              0

📊 Dataset Info:
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1000 entries, 0 to 999
Data columns (total 10 columns):
 #   Column           Non-Null Count  Dtype  
--- 
 0   ID               1000 non-null    int64  
 1   Age              1000 non-null    int64  
 2   Income            1000 non-null    int64  
 3   Loan_Amount       1000 non-null    int64  
 4   Loan_Duration     1000 non-null    int64  
 5   Credit_History    1000 non-null    int64  
 6   Debt_to_Income    1000 non-null    float64 
 7   Number_of_Loans   1000 non-null    int64  
 8   Late_Payments     1000 non-null    int64  
 9   Credit_Score      1000 non-null    int64  
dtypes: float64(1), int64(9)
memory usage: 78.3 KB
None

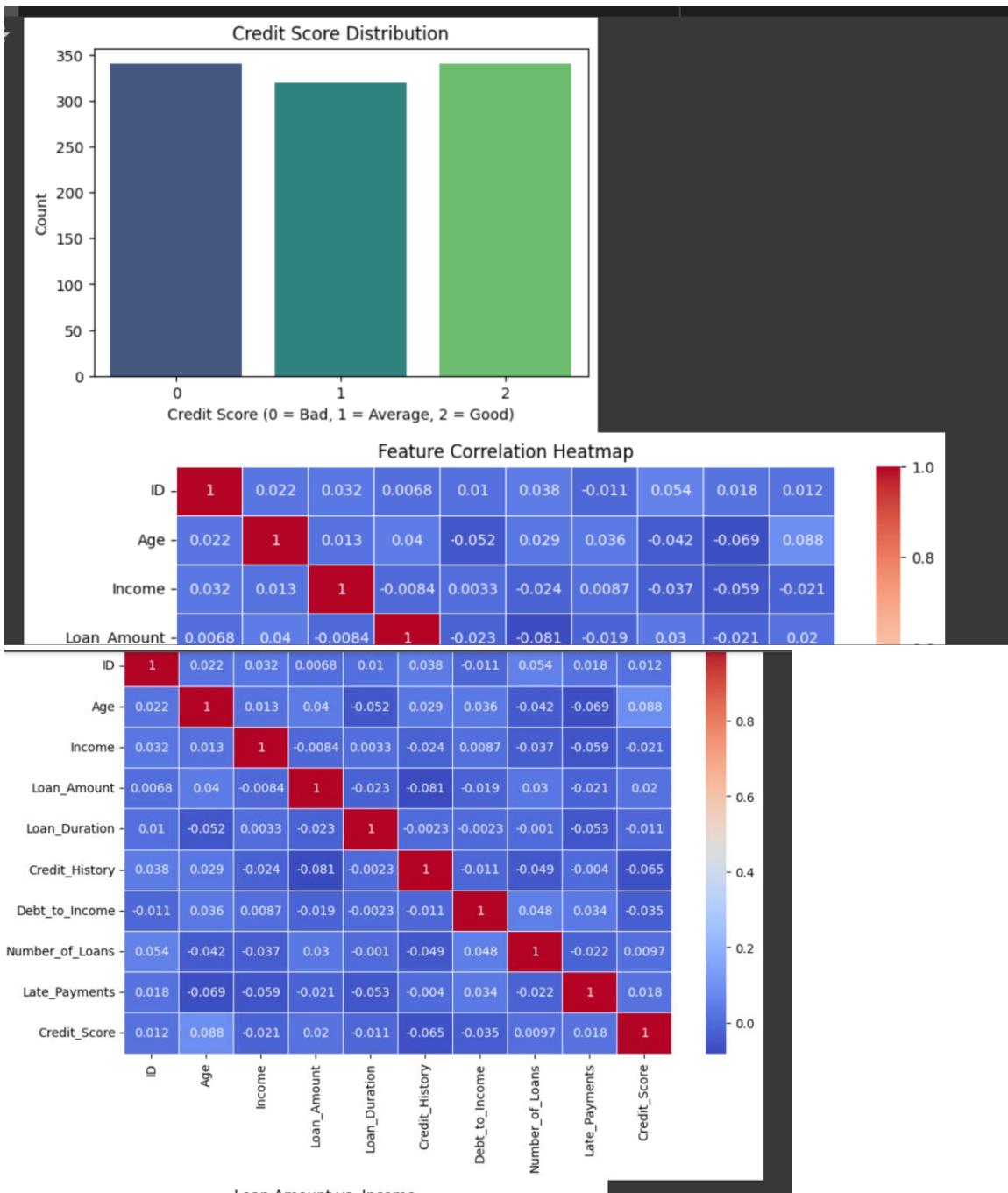
☑️ Checking for Missing Values:
ID          0
Age         0
Income       0
Loan_Amount  0
Loan_Duration 0
Credit_History 0
Debt_to_Income 0
Number_of_Loans 0
Late_Payments 0
Credit_Score  0
dtype: int64
<ipython-input-12-fe3d700e0459>:43: FutureWarning:

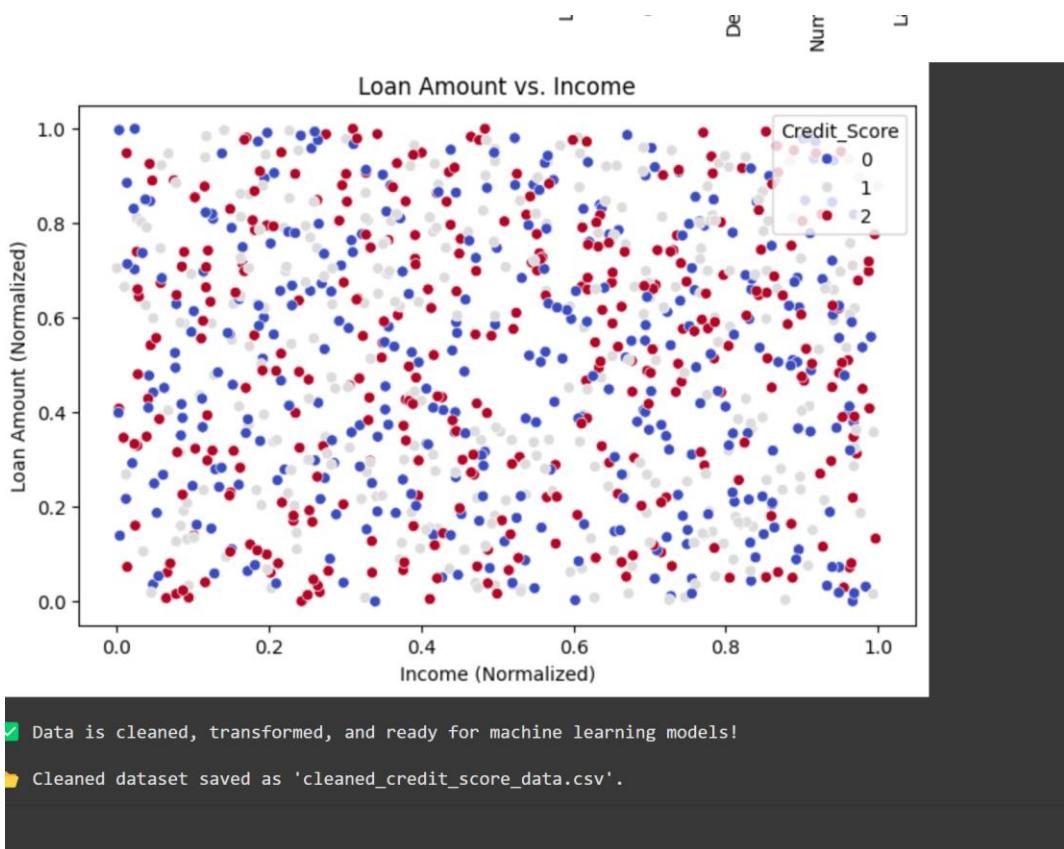
Passing `palette` without assigning `hue` is deprecated and will be removed in v0.14.0. Assign the `x` variable to `hue` and set `legend=F
sns.countplot(x=df["Credit_Score"], palette="viridis")

```

The chart displays the frequency of different credit scores. The x-axis categories are 3, 4, and 5. The y-axis shows the count of observations, with major ticks at 250, 300, and 350. The distribution is roughly equal across the three categories.

Credit Score	Count
3	~320
4	~310
5	~320





## References/Credits

- Dataset Source: CHAT GPT
- Libraries Used: Pandas, NumPy, Matplotlib, Seaborn, Scikit-learn
- Images and Graphs generated using Python visualization libraries.