# Finance Data Cleaning and Visualization Project

## Introduction

This project demonstrates how to clean, transform, normalize, and encode a finance dataset using Python libraries such as Pandas and NumPy. It also includes data visualization using Matplotlib for a clear understanding of the dataset.

## Raw Finance Dataset

A sample of the generated raw finance dataset with missing values and duplicates is shown below:

TransactionID Customer Amount Category PaymentMethod \  
0 1 Customer\_4 3485.0 Groceries Credit Card   
1 2 Customer\_5 6496.0 Groceries Cash   
2 3 Customer\_3 8766.0 Clothing UPI   
3 4 Customer\_5 9374.0 Electronics UPI   
4 5 Customer\_5 2658.0 Entertainment Debit Card   
5 6 Customer\_2 NaN Groceries Credit Card   
6 7 Customer\_3 2147.0 Entertainment Cash   
7 8 Customer\_3 2847.0 Electronics Debit Card   
8 9 Customer\_3 9267.0 Electronics Cash   
9 10 Customer\_5 289.0 Electronics Cash   
  
 TransactionDate   
0 2024-03-01   
1 2024-03-02   
2 2024-03-03   
3 2024-03-04   
4 2024-03-05   
5 2024-03-06   
6 2024-03-07   
7 2024-03-08   
8 2024-03-09   
9 2024-03-10

Download the raw dataset for reference:

[Download Raw Data](sandbox:/mnt/data/raw\_finance\_data.csv)

## Step 1: Data Cleaning

In this step, we will handle missing values, remove duplicates, and ensure the dataset is clean.  
We'll perform the following actions using Pandas:  
1. Identify and drop duplicate records.  
2. Handle missing values using appropriate techniques.  
3. Validate the data types.

### Python Code for Data Cleaning:

# Import necessary libraries  
import pandas as pd  
  
# Load dataset  
df = pd.read\_csv('raw\_finance\_data.csv')  
  
# Check for duplicates  
print("Duplicate Records:")  
print(df[df.duplicated()])  
  
# Remove duplicates  
df = df.drop\_duplicates()  
print("Duplicates removed.")  
  
# Check for missing values  
print("Missing Values:")  
print(df.isnull().sum())  
  
# Handle missing Amount by imputing with mean  
df['Amount'].fillna(df['Amount'].mean(), inplace=True)  
  
# Handle missing Category by filling with mode  
df['Category'].fillna(df['Category'].mode()[0], inplace=True)  
  
# Validate data types  
print("Data Types:")  
print(df.dtypes)

## Step 2: Data Transformation

Data transformation involves converting data into a usable format. In this step, we'll:  
1. Convert categorical data into numeric using Label Encoding.  
2. Extract additional insights from TransactionDate.

### Python Code for Data Transformation:

from sklearn.preprocessing import LabelEncoder  
  
# Label Encoding for categorical columns  
label\_encoder = LabelEncoder()  
df['Customer'] = label\_encoder.fit\_transform(df['Customer'])  
df['Category'] = label\_encoder.fit\_transform(df['Category'])  
df['PaymentMethod'] = label\_encoder.fit\_transform(df['PaymentMethod'])  
  
# Extract day, month, and year from TransactionDate  
df['TransactionDate'] = pd.to\_datetime(df['TransactionDate'])  
df['Day'] = df['TransactionDate'].dt.day  
df['Month'] = df['TransactionDate'].dt.month  
df['Year'] = df['TransactionDate'].dt.year  
  
print("Data after Transformation:")  
print(df.head())

## Step 3: Data Normalization

Normalization is the process of scaling numerical data to a standard range, typically between 0 and 1.  
This helps machine learning models converge faster and improves accuracy.  
We'll apply Min-Max Normalization on the Amount column.

### Python Code for Data Normalization:

from sklearn.preprocessing import MinMaxScaler  
  
scaler = MinMaxScaler()  
df['Amount'] = scaler.fit\_transform(df[['Amount']])  
  
print("Data after Normalization:")  
print(df.head())

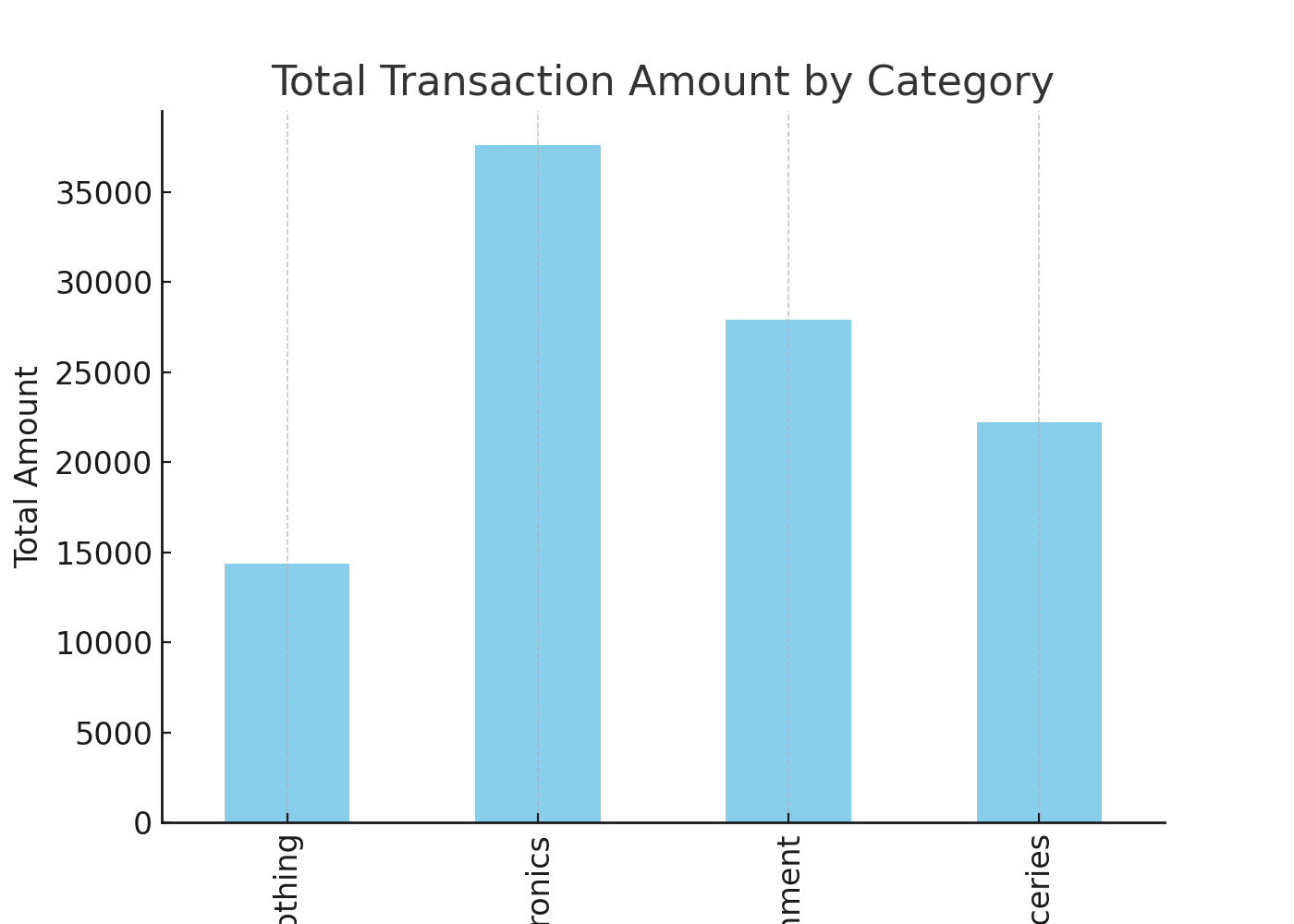
## Step 4: Encoding

Encoding involves converting categorical variables into numerical values. In this step, we have already applied Label Encoding.  
This is particularly useful when working with machine learning algorithms.

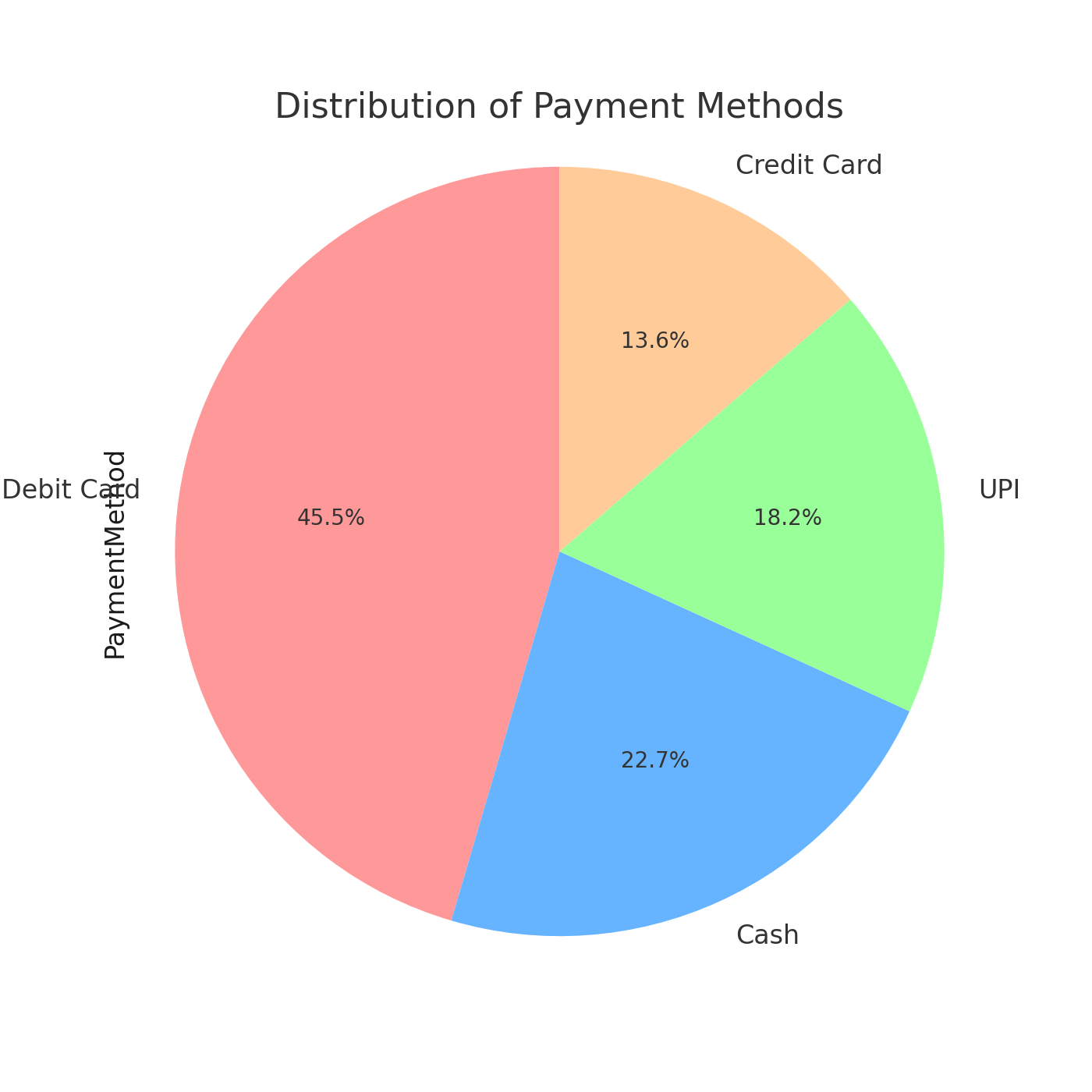
## Step 5: Data Visualization

We'll create two simple visualizations using Matplotlib to gain insights from the dataset.  
1. A \*\*Bar Plot\*\* showing the total transaction amount by category.  
2. A \*\*Pie Chart\*\* representing the distribution of payment methods.

### Visualization 1: Total Transaction Amount by Category



### Visualization 2: Distribution of Payment Methods



## Conclusion

This project demonstrated the step-by-step process of cleaning, transforming, normalizing, and encoding a financial dataset.  
We also visualized important insights using bar plots and pie charts.  
These steps are essential for data preprocessing before applying machine learning algorithms.