

Batch: D-2

K. J. Somaiya College of Engineering, Mumbai-77 (A Constituent College of Somaiya Vidyavihar University) **Department of Computer Engineering**

Roll No.: 16010122151



Title: Implement data pre-processing using python on real world dataset
Course Outcome:
CO1 Understand basic concepts of data analytics to solve real-world problems
Books/ Journals/ Websites referred:
Geeksforgeeks.com
Resources used:
Google colab for writing python scripts
Theory (About Data PreprData preprocessing is essential for preparing raw data for analysis and modeling. It involves:
1. Data Cleaning:
- Handling Missing Values: Using techniques like deletion or imputation to address gaps in data.
- Removing Duplicates: Ensuring no repeated data points.

2. Data Transformation:

-Normalization/Standardization: Scaling data to ensure consistency, especially for algorithms sensitive to feature magnitudes.

- Outlier Detection: Identifying and managing unusually extreme values.





- Encoding Categorical Data: Converting categorical variables into numerical formats, like one-hot encoding.
- Feature Engineering: Creating new features or modifying existing ones to improve model performance.

3. Data Reduction:

- Dimensionality Reduction: Reducing the number of features using methods like PCA to eliminate redundant data.
- Feature Selection: Selecting the most relevant features to enhance model efficiency and accuracy.

Preprocessing ensures that the data is clean, consistent, and ready for effective analysis or modeling.ocessing):

Program:

```
import pandas as pd
import numpy as np
# Sample data

data = {
    'name': ['Alice', 'Bob', 'Charlie', 'Dave', 'Eve'],
    'age': [25, np.nan, 30, 22, 35],
    'gender': ['F', 'M', 'M', 'M', 'F'],
    'income': [50000, 60000, 75000, np.nan, 80000]
}

df = pd.DataFrame(data)
```





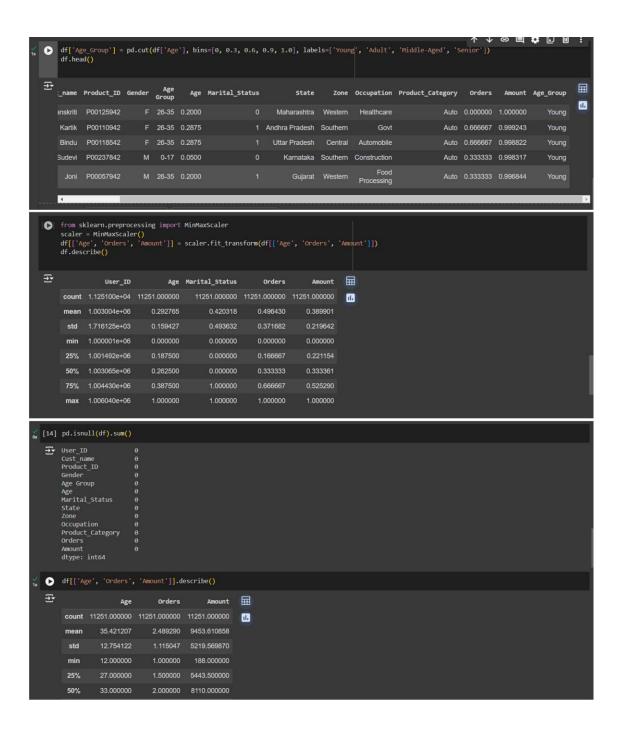
```
# Display the original data
print("Original DataFrame:")
print(df)
# User-defined function for discretization
def discretize_age(age):
  if age < 30:
    return 'Young'
  elif age \geq 30 and age < 40:
    return 'Middle-aged'
  else:
    return 'Old'
# Handling missing values (NaN)
# Fill missing values in 'age' with the mean age
mean_age = df['age'].mean()
df['age'].fillna(mean_age, inplace=True)
# Apply discretization function to 'age' column
df['age_category'] = df['age'].apply(discretize_age)
# Drop rows with missing values in any column
df.dropna(inplace=True)
# Convert categorical variables (gender) to numerical
df['gender'] = df['gender'].map(\{'F': 0, 'M': 1\})
# Data normalization Min -Max
# Normalize 'income' column to range [0, 1]
min_income = df['income'].min()
max_income = df['income'].max()
```





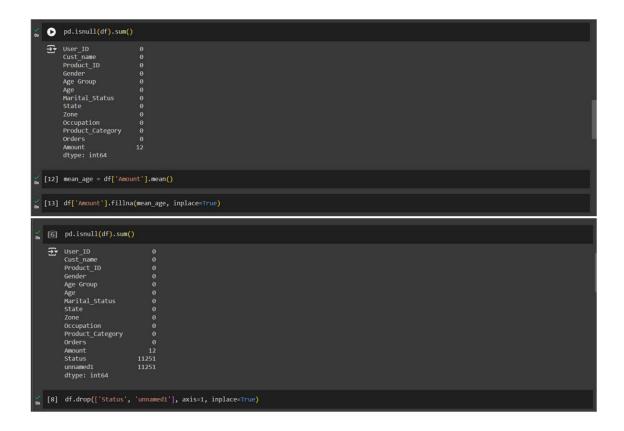






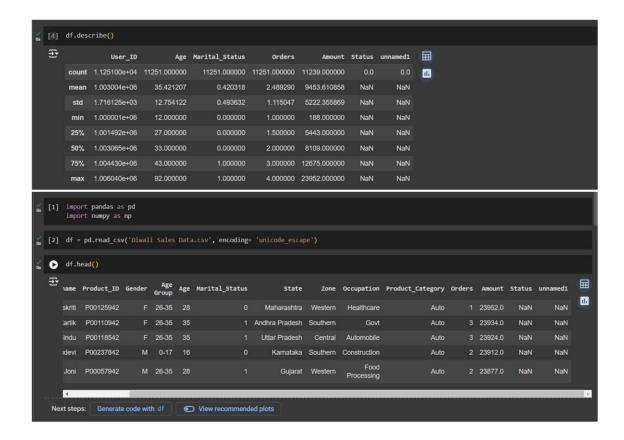












Conclusion (Students should write in their own words):

Through this experiment we get to know about the different techniques of Data Analysis including Normalization and Kiscíctizatio→.

Post lab questions:

Q.1 What are some common challenges encountered during data cleaning? How did you handle missing values in the provided dataset?

Common Challenges:





- Missing Data: Incomplete data entries are common and can arise from data entry errors or system issues.
- Outliers: Extreme values that don't conform to expected patterns can skew analysis.
- Inconsistent Data: Variations in data formats, units, or values that should be standardized.
- Duplicate Entries: Repeated data points that need to be identified and removed.
- Irrelevant Data: Presence of unnecessary or redundant features that don't contribute to the analysis.

Handling Missing Values:

- Deletion: If missing values are minimal, rows or columns can be removed.
- Imputation: Filling in missing data using methods like mean, median, or mode for numerical data, or using the most frequent value for categorical data.
- Advanced Techniques: Using algorithms like KNN (K-Nearest Neighbors)
 imputation, which predicts missing values based on the similarity with other
 data points.

Q.2 Explain the importance of data normalization in the context of machine learning models. How does normalizing benefit the analysis?

Importance of Data Normalization:

- Ensures Consistency: Many machine learning algorithms, such as gradient descent-based methods, are sensitive to the scale of input data. Features with larger ranges can disproportionately influence the model.
- Improves Model Performance: Normalizing data ensures that each feature contributes equally, leading to faster convergence during training and better overall model accuracy.
- Enhances Interpretability: Normalized data allows for easier interpretation and comparison of feature importance.

Benefits to Analysis:





- Avoiding Bias: Prevents any single feature from dominating the model due to its scale.
- Facilitating Faster Learning: Helps algorithms converge more quickly, improving efficiency.
- Boosting Accuracy: Leads to more accurate and reliable model predictions.

Q.3 Discuss why it's essential to convert categorical variables like 'gender' into numerical representations.

Essentiality of Conversion:

- Compatibility with Algorithms: Most machine learning algorithms require numerical input, so categorical variables must be converted to be processed correctly.
- Improved Model Performance: Converting categorical variables into numerical representations (e.g., one-hot encoding) allows models to treat these variables appropriately, enhancing prediction accuracy.
- Capturing Relationships: Numerical representations help in capturing relationships between different categories, especially in distance-based models like KNN or clustering algorithms.

Example:

• Gender Encoding: The categorical variable 'gender' can be converted into numerical values (e.g., 0 for 'male' and 1 for 'female') or one-hot encoded (e.g., [1,0] for 'male' and [0,1] for 'female') to ensure the model understands and utilizes this information effectively.