Name: Swarnika Singh Class: D15B Roll no: 65

EXPERIMENT-3

AIM: Experiment to preprocess dataset using different preprocessing techniques.

Theory*:*

Data preprocessing is a vital step in the data analysis and machine learning pipeline. It involves transforming raw data into a clean, formatted, and structured form suitable for analysis or model training. Here are some common tasks and techniques involved in data preprocessing:

- Data Cleaning: Data cleaning is the process of identifying and correcting errors, inconsistencies, and missing values in datasets. This step ensures that the data is accurate and reliable for analysis or modeling.
- Handling Missing Values: Identify and handle missing data, either by imputing values (e.g., using the mean or median) or removing instances with missing data.
- Outlier Detection and Treatment: Identify and handle outliers that can significantly impact analysis or model training.
- Data Transformation: Data transformation involves converting raw data into a format that is more suitable and informative for analysis or modeling. The goal is to enhance data quality, make it compatible with specific algorithms, and reveal underlying patterns or relationships.
- Normalization/Scaling: Standardize numerical features to bring them to a common scale. This is important for algorithms sensitive to feature magnitude.
- Encoding Categorical Variables: Convert categorical variables into a numerical format that can be used by machine learning algorithms, using techniques like one-hot encoding or label encoding.
- Feature Engineering: Feature engineering involves creating new features or modifying existing features to improve the performance of machine learning models. This can include creating interaction terms, polynomial features, or extracting features from text or images.
- Dimensionality Reduction: Dimensionality reduction techniques are used to reduce the number of features in a dataset while preserving important information. This can help reduce overfitting and improve model performance.
- Data Integration: Data integration involves combining data from multiple sources to create a unified view. This can help enrich the dataset and provide more comprehensive insights.
- Data Discretization: Data discretization is the process of converting continuous data into discrete intervals. This can simplify the data and make it easier to analyze.
- Data Normalization: Data normalization is the process of rescaling the data to have a mean of 0 and a standard deviation of 1. This can improve the performance of some machine learning algorithms.
- Data Sampling: Data sampling involves selecting a subset of data points from a larger dataset. This can be useful for creating smaller, more manageable datasets for analysis or model training.

Name: Swarnika Singh Class: D15B Roll no: 65

- Overall, data preprocessing is a critical step in the data analysis and machine learning process, as it helps ensure that the data is clean, accurate, and suitable for the intended analysis or modeling task

Code-

1.Removing and replacing missing values-

```
[8] import pandas as pd
    df = pd.read_csv('/content/library_dataset.csv')
```

```
df.describe()
```

	Rating	Copies	Pages	Price
count	500.000000	500.000000	500.000000	500.000000
mean	2.964000	50.176000	529.964000	54.792000
std	1.382217	28.610735	265.496681	26.194523
min	1.000000	1.000000	51.000000	10.000000
25%	2.000000	27.000000	305.750000	33.000000
50%	3.000000	48.000000	519.500000	54.000000
75%	4.000000	75.000000	764.500000	75.000000
max	5.000000	101.000000	997.000000	101.000000

Name: Swarnika Singh Class: D15B Roll no: 65

```
+ Code + Text

import pandas as pd

# Load your dataset
df = pd.read_csv('/content/library_dataset.csv')

# Remove rows with missing values
df_no_missing = df.dropna()

# Fill missing values with the mean of each column
df_filled = df.fillna(df.mean())

# Display the DataFrame with missing values removed
print("DataFrame with missing values removed:")
print(df_no_missing)

# Display the DataFrame with missing values filled with mean
print("\nDataFrame with missing values filled with mean:")
print(df_filled)
```

```
DataFrame with missing values removed:
              Author                                          Book  Rating  \
0        Jeffery Gray               Managed encompassing structure       5
1       Tracey Torres             Stand-alone transitional support       2
2         John Thomas       Cross-platform fresh-thinking analyzer       1
3         Alex Hansen      Profit-focused encompassing contingency       1
4    Isaiah Armstrong  Self-enabling dedicated budgetary management       1
..                ...                                          ...     ...
495    Eduardo Brooks               Multi-lateral composite model       4
496     Brenda Ortega  Persistent non-volatile process improvement       5
497     Ariana Arnold      Horizontal next generation service-desk       5
498   Mark Richardson             Streamlined maximized moratorium       3
499     Sylvia Haynes       Decentralized client-driven task-force       5

     Copies  Pages  Price     Genre   Language
0        27    222     27      than   Estonian
1        85    224     13   require  Aragonese
2        46    554     46  everyone       Manx
3        83    722     59      same   Chamorro
4        91    699     47    likely  Inuktitut
..      ...    ...    ...       ...        ...
495       8    150    100      live    Lingala
```

2. removing noisy values(Binning technique)-

Name: Swarnika Singh Class: D15B Roll no: 65

```python
import pandas as pd

# Load your dataset
df = pd.read_csv('/content/library_dataset.csv')

# Define the bins
bins = [0, 20, 40, 60, 80, 100] # Define your own bins here

# Create a new column to store bin labels
df['Pages'] = pd.cut(df['Pages'], bins=bins, labels=False)

# Calculate the mean or median of each bin
bin_means = df.groupby('Pages')['Pages'].mean() # You can also use median() instead of mean()

# Replace noisy values with the mean or median of each bin
df['Pages'] = df.groupby('Pages')['Pages'].apply(lambda x: x.fillna(x.mean())) # Use median() if preferred

# Drop the temporary bin column
df.drop(columns=['Pages'], inplace=True)

# Display the DataFrame with noisy values replaced
print(df)
```

Output-

```
              Author                                    Book  Rating  \
0       Jeffery Gray            Managed encompassing structure       5
1      Tracey Torres           Stand-alone transitional support      2
2        John Thomas       Cross-platform fresh-thinking analyzer      1
3        Alex Hansen      Profit-focused encompassing contingency      1
4    Isaiah Armstrong  Self-enabling dedicated budgetary management      1
..                ...                                       ...     ...
495    Eduardo Brooks              Multi-lateral composite model      4
496     Brenda Ortega  Persistent non-volatile process improvement      5
497     Ariana Arnold      Horizontal next generation service-desk      5
498   Mark Richardson            Streamlined maximized moratorium      3
499     Sylvia Haynes       Decentralized client-driven task-force      5

     Copies  Price     Genre   Language
0        27     27      than   Estonian
1        85     13   require  Aragonese
2        46     46  everyone       Manx
3        83     59      same   Chamorro
4        91     47    likely  Inuktitut
..      ...    ...       ...        ...
495       8    100      live    Lingala
496      79     99      onto     Yoruba
497      47     92    spring    Serbian
498       8     54     event      Latin
499      91     54  recently     Aymara

[500 rows x 7 columns]
<ipython-input-14-26274734e582>:16: FutureWarning: Not prepending group keys to the result index of transform-like app
To preserve the previous behavior, use

        >>> .groupby(..., group_keys=False)

To adopt the future behavior and silence this warning, use

        >>> .groupby(..., group_keys=True)
  df['Pages'] = df.groupby('Pages')['Pages'].apply(lambda x: x.fillna(x.mean())) # Use median() if preferred
```

✓ 0s completed at 9:45AM

3.removing outliers-Interquartile Range Method

Name: Swarnika Singh Class: D15B Roll no: 65

```python
import pandas as pd

# Load your dataset
df = pd.read_csv('/content/library_dataset.csv')

# Verify the column names
print(df.columns)

# Define the column for which you want to remove outliers
column_name = 'num_lower'

# Check if the column exists
if column_name not in df.columns:
    print(f"Column '{column_name}' does not exist in the DataFrame.")
else:
    # Calculate the first quartile (Q1) and third quartile (Q3)
    Q1 = df[column_name].quantile(0.25)
    Q3 = df[column_name].quantile(0.75)

    # Calculate the interquartile range (IQR)
    IQR = Q3 - Q1

    # Define the lower and upper bounds for outliers removal
    lower_bound = Q1 - 1.5 * IQR
    upper_bound = Q3 + 1.5 * IQR

    # Filter the DataFrame to remove outliers
    df_no_outliers = df[(df[column_name] >= lower_bound) & (df[column_name] <= upper_bound)]

    # Display the DataFrame with outliers removed
    print(df_no_outliers)
```

Output:

```
Index(['Author', 'Book', 'Rating', 'Copies', 'Pages', 'Price', 'Genre',
       'Language'],
      dtype='object')
Column 'num_lower' does not exist in the DataFrame.
```

4.boxplot-

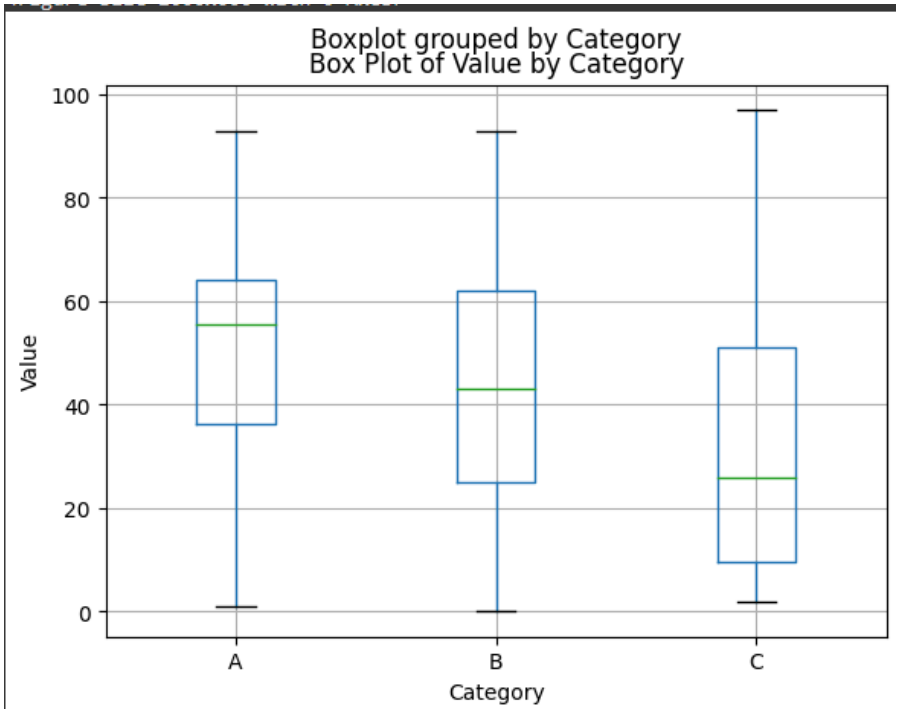
Name: Swarnika Singh Class: D15B Roll no: 65

```python
import matplotlib.pyplot as plt
import pandas as pd
import numpy as np

# Create a sample DataFrame
data = {
    'Category': np.random.choice(['A', 'B', 'C'], 100),
    'Value': np.random.randint(0, 100, 100)
}
df = pd.DataFrame(data)

# Create a box plot
plt.figure(figsize=(10, 6))
df.boxplot(by='Category', column='Value')
plt.title('Box Plot of Value by Category')
plt.xlabel('Category')
plt.ylabel('Value')
plt.grid(True)
plt.show()
```

Output:



5.Converting numerical attributes into categorical/One hot encoding.

Name: Swarnika Singh Class: D15B Roll no: 65

```python
import pandas as pd

# Load your dataset
df = pd.read_csv('/content/library_dataset.csv')

# Specify the numerical attribute(s) you want to convert
numerical_attributes = ['Pages', 'Price']

# Convert numerical attributes to categorical using pd.cut()
for column in numerical_attributes:
    df[column + '_category'] = pd.cut(df[column], bins=3, labels=['low', 'medium', 'high']) # Adjust b

# Perform one-hot encoding using pd.get_dummies()
df_encoded = pd.get_dummies(df, columns=[column + '_category' for column in numerical_attributes])

# Display the encoded DataFrame
print(df_encoded.head())
```

Output:

```
              Author                                             Book  Rating  \
0      Jeffery Gray                Managed encompassing structure       5
1     Tracey Torres              Stand-alone transitional support       2
2       John Thomas        Cross-platform fresh-thinking analyzer       1
3       Alex Hansen       Profit-focused encompassing contingency       1
4  Isaiah Armstrong  Self-enabling dedicated budgetary management       1

   Copies  Pages  Price     Genre   Language  Pages_category_low  \
0      27    222     27      than   Estonian                   1
1      85    224     13   require  Aragonese                   1
2      46    554     46  everyone       Manx                   0
3      83    722     59      same   Chamorro                   0
4      91    699     47    likely  Inuktitut                   0

   Pages_category_medium  Pages_category_high  Price_category_low  \
0                      0                    0                   1
1                      0                    0                   1
2                      1                    0                   0
3                      0                    1                   0
4                      0                    1                   0

   Price_category_medium  Price_category_high
0                      0                    0
1                      0                    0
2                      1                    0
3                      1                    0
4                      1                    0
```

Name: Swarnika Singh Class: D15B Roll no: 65

6. Z-Score Normalization-

```
import pandas as pd
from sklearn.preprocessing import StandardScaler

# Load your dataset
df = pd.read_csv('/content/library_dataset.csv') # Replace 'your_dataset.csv' with the path to your dataset

# Select numerical columns for normalization
numeric_columns = df.select_dtypes(include=["number"]).columns

# Perform Z-score normalization
scaler = StandardScaler()
df[numeric_columns] = scaler.fit_transform(df[numeric_columns])

# Display the normalized DataFrame
print(df.head())
```

Output:

```
              Author                                              Book    Rating  \
0      Jeffery Gray                Managed encompassing structure  1.474471
1     Tracey Torres              Stand-alone transitional support -0.698129
2       John Thomas        Cross-platform fresh-thinking analyzer -1.422328
3       Alex Hansen       Profit-focused encompassing contingency -1.422328
4  Isaiah Armstrong  Self-enabling dedicated budgetary management -1.422328

     Copies     Pages     Price     Genre   Language
0 -0.810857 -1.161116 -1.062048      than   Estonian
1  1.218384 -1.153575 -1.597046   require  Aragonese
2 -0.146105  0.090623 -0.335979  everyone       Manx
3  1.148411  0.724033  0.160805      same   Chamorro
4  1.428306  0.637316 -0.297765    likely  Inuktitut
```

```
[ ]
```

7.Data Reduction-

Name: Swarnika Singh Class: D15B Roll no: 65

```python
import pandas as pd

# Load your dataset
df = pd.read_csv('/content/library_dataset.csv') # Replace 'your_dataset.csv' with the path to your dataset

# Set the desired reduction factor (percentage of instances to keep)
reduction_factor = 0.5 # Adjust as needed (e.g., 0.5 means keeping 50% of instances)

# Randomly select a subset of instances
reduced_df = df.sample(frac=reduction_factor, random_state=42) # Set random_state for reproducibility

# Display the reduced DataFrame
print(reduced_df)
```

Output:

```
                  Author                                          Book  Rating  \
361         James Sherman          Front-line eco-centric flexibility       4
73        Jenna Valentine     Quality-focused zero tolerance analyzer       2
374           Derek James         Configurable intermediate help-desk       4
155    Terrence Dixon DDS      Fully-configurable analyzing help-desk       3
104       Elizabeth Green                      Focused modular toolset       1
..                    ...                                         ...     ...
103          Melissa West  Future-proofed zero administration support       3
81        Regina Whitaker              Sharable dedicated contingency       4
38          Bonnie Reeves  Business-focused mission-critical solution       5
314  Matthew Francis DVM          Function-based composite emulation       3
167      Rebecca Perez MD               Switchable optimal capability       1

     Copies  Pages  Price    Genre         Language
361      82    945     21  nothing  Western Frisian
73       41    436     25   arrive       Portuguese
374      77    686     25     hold     Luba-Katanga
155       3    246     48     fish         Galician
104      41    303     32    sound          Amharic
..      ...    ...    ...      ...              ...
103      47    931     44     best         Kuanyama
81       97    707     60     born          Chinese
38       70    318     36   animal         Corsican
314      77    386     94    their           Nepali
167      56    731     68    where           Kikuyu

[250 rows x 8 columns]
```

CONCLUSION- We successfully understood and implemented data preprocessing using various preprocessing techniques.