Practical 3

Data Cleaning and Preprocessing

Aim: To clean and preprocess a dataset by handling missing values, removing duplicates, and normalizing data.

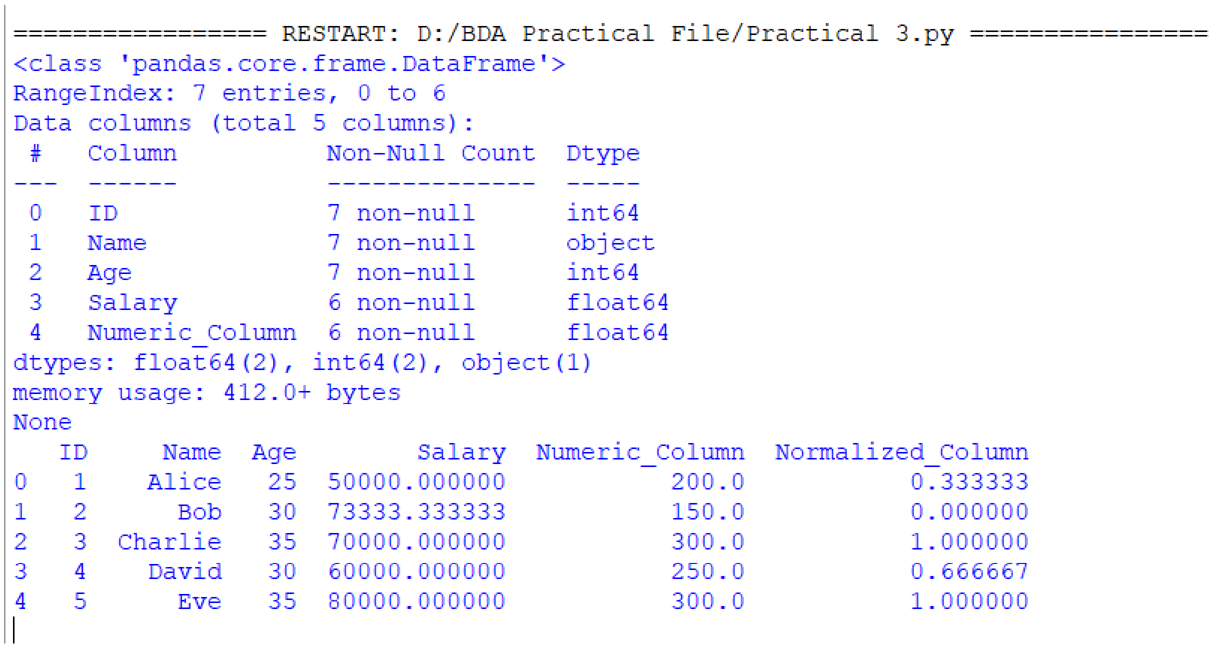
Steps:

1. **Import a dataset (e.g., a CSV file).**
2. **Inspect the dataset for missing values and duplicates.**
3. **Handle missing values by filling them with mean/median or dropping rows.**
4. **Normalize a numerical column to a scale of 0 to 1.**
5. **Display the cleaned dataset.**

Code:

```
import pandas as pd
from sklearn.preprocessing import MinMaxScaler
# Step 1: Load dataset
df = pd.read_csv("sample_dataset.csv")
# Step 2: Inspect dataset
print(df.info())
# Step 3: Handle missing values in numeric columns
numeric_cols = df.select_dtypes(include=['number']).columns  # Get numeric columns
df[numeric_cols] = df[numeric_cols].fillna(df[numeric_cols].mean())  # Fill NaN with column mean
# Step 4: Remove duplicates
df.drop_duplicates(inplace=True)
# Step 5: Normalize a specific numerical column
scaler = MinMaxScaler()
df['Normalized_Column'] = scaler.fit_transform(df[['Numeric_Column']])
# Display cleaned dataset
print(df.head())
```

Output:



```
================= RESTART: D:/BDA Practical File/Practical 3.py ================
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 7 entries, 0 to 6
Data columns (total 5 columns):
 #   Column          Non-Null Count  Dtype
---  ------          --------------  -----
 0   ID              7 non-null      int64
 1   Name            7 non-null      object
 2   Age             7 non-null      int64
 3   Salary          6 non-null      float64
 4   Numeric_Column  6 non-null      float64
dtypes: float64(2), int64(2), object(1)
memory usage: 412.0+ bytes
None
   ID     Name  Age        Salary  Numeric_Column  Normalized_Column
0   1    Alice   25  50000.000000           200.0           0.333333
1   2      Bob   30  73333.333333           150.0           0.000000
2   3  Charlie   35  70000.000000           300.0           1.000000
3   4    David   30  60000.000000           250.0           0.666667
4   5      Eve   35  80000.000000           300.0           1.000000
```