EXP NO: 3	EDA-DATA CLEANING

AIM

To clean data by handling missing values, duplicates, data types, and normalization.

PROBLEM STATEMENT

Clean a dataset by removing nulls, duplicates, and normalizing numeric fields.

ALGORITHM

1. Load dataset.
2. Detect missing values (isnull).
3. Fill or drop missing values.
4. Remove duplicates.
5. Convert data types.
6. Normalize numeric columns.

SAMPLE CODE

```
import pandas as pd
from sklearn.preprocessing import StandardScaler, MinMaxScaler
import matplotlib.pyplot as plt


# Step 1: Load dataset
df = pd.read_csv('StudentsPerformance.csv')
df.head()
```

	gender	race/ethnicity	parental level of education	lunch	test preparation course	math score	reading score	writing score
0	female	group B	bachelor's degree	standard	none	72	72	74
1	female	group C	some college	standard	completed	69	90	88
2	female	group B	master's degree	standard	none	90	95	93
3	male	group A	associate's degree	free/reduced	none	47	57	44
4	male	group C	some college	standard	none	76	78	75

```
df.shape
```

```
(1005, 8)
```

```
# Step 2: Handle Missing Values
# Detect
missing_info = df.isnull().sum()
print("Missing values:\n", missing_info)
```

```
Missing values:
 gender                        0
race/ethnicity                 0
parental level of education    7
lunch                          0
test preparation course        0
math score                     0
reading score                  0
writing score                  0
dtype: int64
```

```
# Fill or Drop (based on context)
df.fillna({
   'parental level of education': df['parental level of education'].mode()[0],
   'lunch': df['lunch'].mode()[0]
}, inplace=True)
missing_info = df.isnull().sum()
missing_info
```

```
gender                        0
race/ethnicity                0
parental level of education    0
lunch                          0
test preparation course        0
math score                     0
reading score                  0
writing score                  0
dtype: int64
```

```
duplicates = df[df.duplicated()]
duplicates
```

	gender	race/ethnicity	parental level of education	lunch	test preparation course	math score	reading score	writing score
1000	male	group D	some college	standard	none	76	64	66
1001	male	group C	associate's degree	standard	none	46	43	42
1002	female	group B	bachelor's degree	standard	none	67	86	83
1003	male	group E	some high school	standard	none	92	87	78
1004	male	group C	bachelor's degree	standard	completed	83	82	84

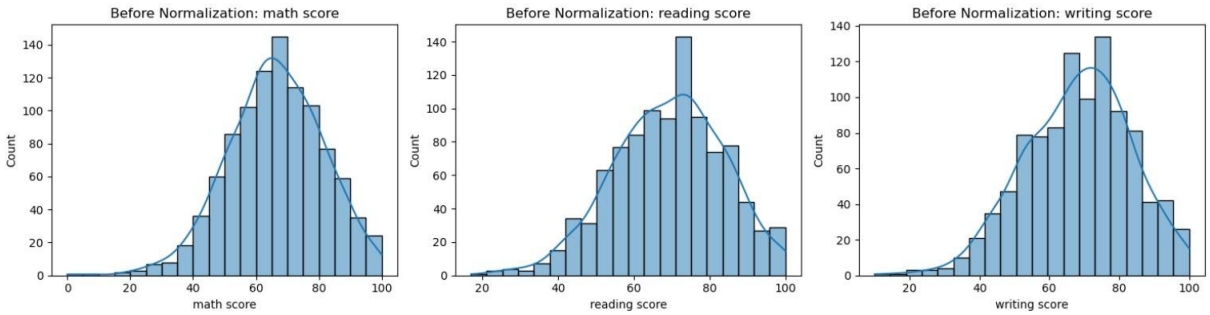
```
duplicates.shape
```

(5, 8)

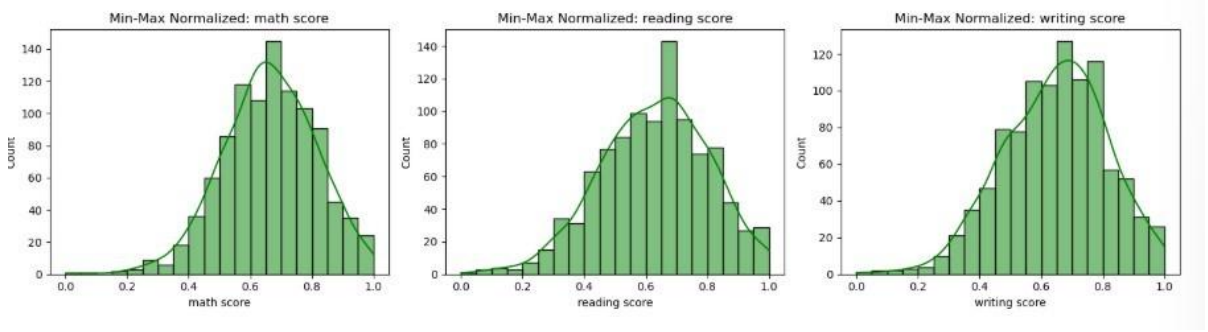
```
# Drop duplicates
df.drop_duplicates(inplace=True)
df.shape
# Step 4: Convert Data Types (if needed)
# For consistency, make sure string columns are lowercase
categorical_cols = ['gender', 'race/ethnicity', 'parental level of education', 'lunch', 'test preparation course']
for col in categorical_cols:
    df[col] = df[col].astype(str).str.lower().str.strip()
categorical_cols
['gender',
 'race/ethnicity',
 'parental level of education',
 'lunch',
 'test preparation course']

numeric_cols = ['math score', 'reading score', 'writing score']
numeric_cols
```

['math score', 'reading score', 'writing score']

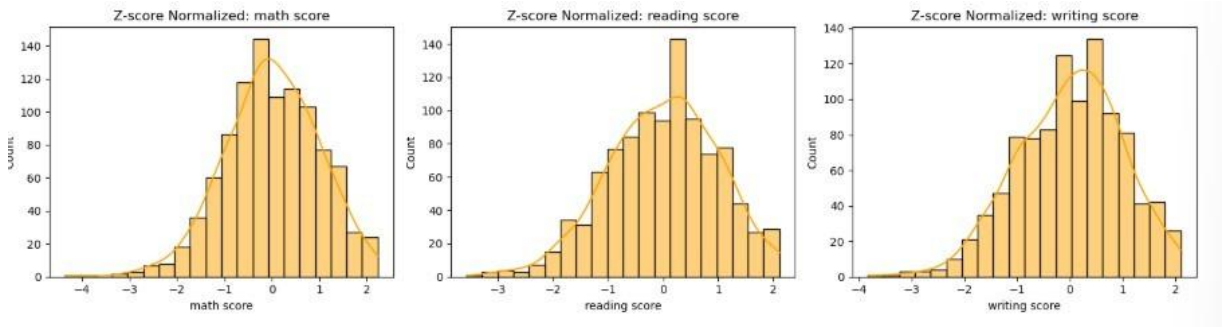
```
plt.figure(figsize=(15, 4))
for i, col in enumerate(numeric_cols):
    plt.subplot(1, 3, i+1)
    sns.histplot(df[col], kde=True, bins=20)
    plt.title(f'Before Normalization: {col}')
plt.tight_layout()
plt.show()
```




```
minmax_scaler = MinMaxScaler()
df_minmax = df.copy()
df_minmax[numeric_cols] = minmax_scaler.fit_transform(df[numeric_cols])
plt.figure(figsize=(15, 4))
for i, col in enumerate(numeric_cols):
    plt.subplot(1, 3, i+1)
    sns.histplot(df_minmax[col], kde=True, bins=20, color='green')
    plt.title(f'Min-Max Normalized: {col}')
plt.tight_layout()
plt.show()
```




```
# Standard Scaling (Z-score)
zscore_scaler = StandardScaler()
df_zscore = df.copy()
df_zscore[numeric_cols] = zscore_scaler.fit_transform(df[numeric_cols])


plt.figure(figsize=(15, 4))
for i, col in enumerate(numeric_cols):
    plt.subplot(1, 3, i+1)
    sns.histplot(df_zscore[col], kde=True, bins=20, color='orange')
    plt.title(f'Z-score Normalized: {col}')
plt.tight_layout()
plt.show()
```



RESULT:

Thus, the program successfully created a Jupyter Notebook showcasing Python code handling missing values, removing duplicates and unnecessary data, Data type conversion and normalizing data.