

NAME : Nevetha S K

EMAIL ID : nevethasenthilkumar@gmail.com

# PYTHON CODING ASSESSMENT

---

## DATA CLEANING IN PYTHON USING PANDAS :

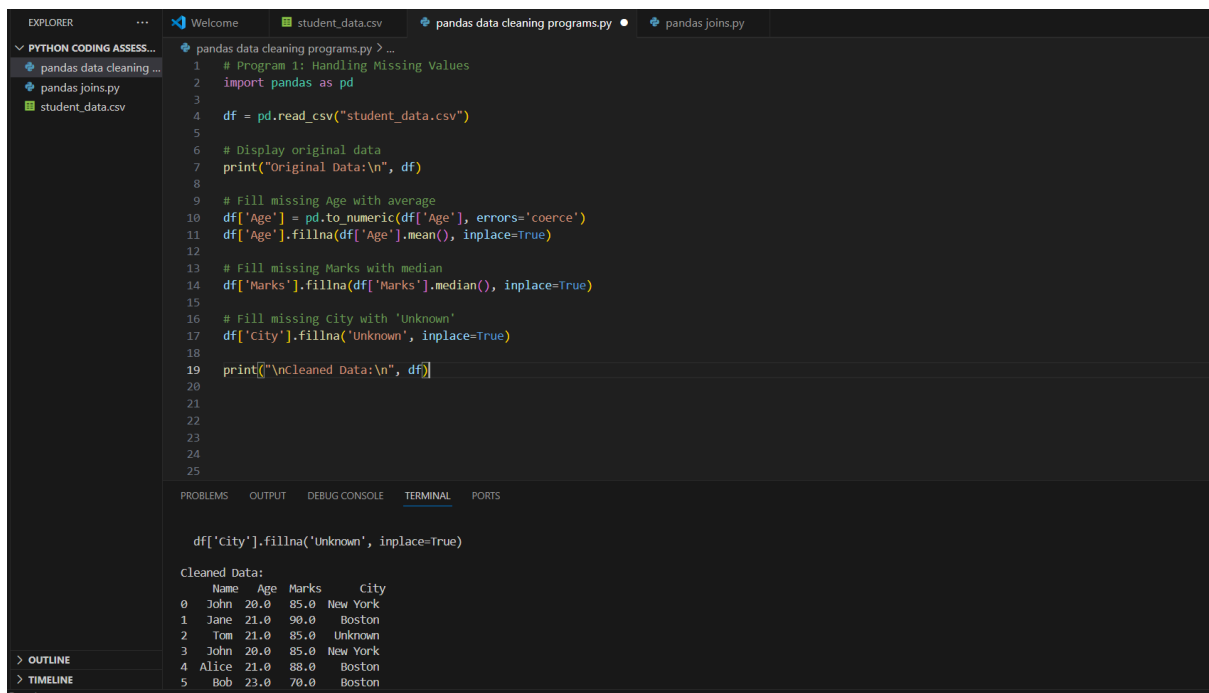
Data cleaning is a critical step in the data preprocessing phase. It involves handling:

- Missing values
- Duplicates
- Incorrect data types
- Outliers
- Inconsistent formatting

**Example Dataset :** student\_data.csv

```
student_data.csv
1  Name, Age, Marks, City
2  John, 20, 85, New York
3  Jane, , 90, Boston
4  Tom, 21, ,
5  John, 20, 85, New York
6  Alice, twenty-two, 88, Boston
7  Bob, 23, 70, Boston
8
```

# Data Cleaning Programs :



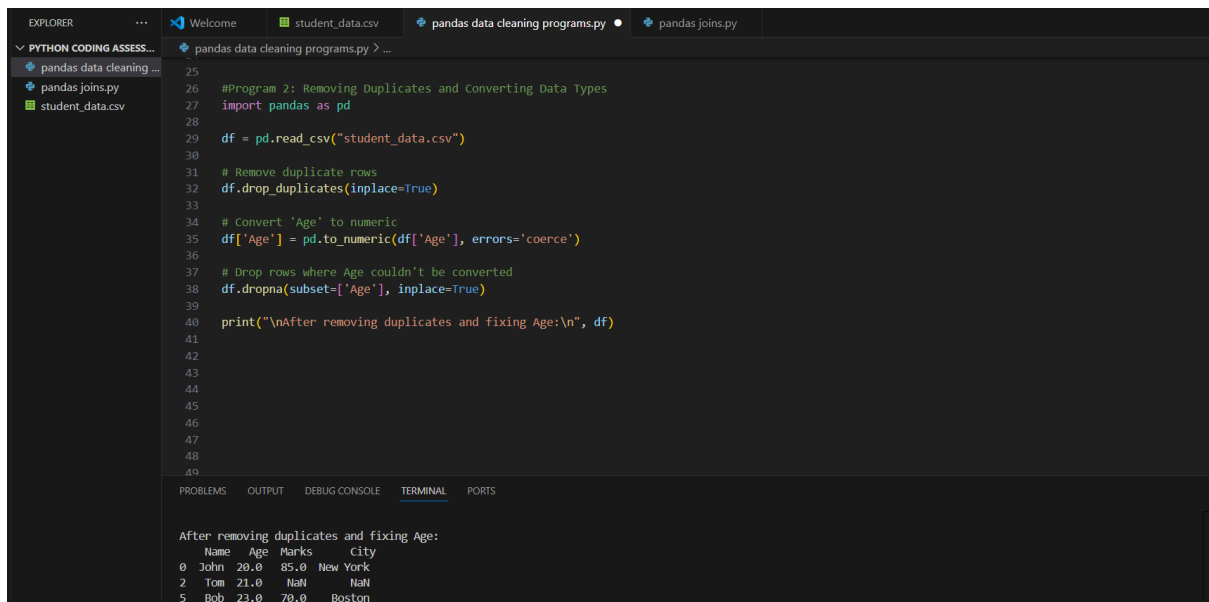
```
1 # Program 1: Handling Missing Values
2 import pandas as pd
3
4 df = pd.read_csv("student_data.csv")
5
6 # Display original data
7 print("Original Data:\n", df)
8
9 # Fill missing Age with average
10 df['Age'] = pd.to_numeric(df['Age'], errors='coerce')
11 df['Age'].fillna(df['Age'].mean(), inplace=True)
12
13 # Fill missing Marks with median
14 df['Marks'].fillna(df['Marks'].median(), inplace=True)
15
16 # Fill missing City with 'Unknown'
17 df['City'].fillna('Unknown', inplace=True)
18
19 print("\nCleaned Data:\n", df)
```

PROBLEMS OUTPUT DEBUG CONSOLE **TERMINAL** PORTS

```
df['City'].fillna('Unknown', inplace=True)
```

Cleaned Data:

	Name	Age	Marks	City
0	John	20.0	85.0	New York
1	Jane	21.0	90.0	Boston
2	Tom	21.0	85.0	Unknown
3	John	20.0	85.0	New York
4	Alice	21.0	88.0	Boston
5	Bob	23.0	70.0	Boston



```
25
26 #Program 2: Removing Duplicates and Converting Data Types
27 import pandas as pd
28
29 df = pd.read_csv("student_data.csv")
30
31 # Remove duplicate rows
32 df.drop_duplicates(inplace=True)
33
34 # Convert 'Age' to numeric
35 df['Age'] = pd.to_numeric(df['Age'], errors='coerce')
36
37 # Drop rows where Age couldn't be converted
38 df.dropna(subset=['Age'], inplace=True)
39
40 print("\nAfter removing duplicates and fixing Age:\n", df)
```

PROBLEMS OUTPUT DEBUG CONSOLE **TERMINAL** PORTS

After removing duplicates and fixing Age:

	Name	Age	Marks	City
0	John	20.0	85.0	New York
2	Tom	21.0	NaN	NaN
5	Bob	23.0	70.0	Boston

The screenshot shows a VS Code editor with a Python file named `pandas data cleaning programs.py`. The script performs the following operations:

- Imports pandas as `pd`.
- Reads `student_data.csv` into a DataFrame `df`.
- Standardizes city names to lowercase and strips whitespace: `df['City'] = df['City'].str.lower().str.strip()`.
- Replaces NaN and fixes data types: `df['Age'] = pd.to_numeric(df['Age'], errors='coerce')`.
- Fills missing values in the 'City' column with 'unknown': `df['City'].fillna('unknown', inplace=True)`.
- Prints the DataFrame: `print("\nCity Standardization:\n", df)`.

The terminal output shows the result of the `fillna` operation and the final DataFrame:

```
df['City'].fillna('unknown', inplace=True)
```

City Standardization:

	Name	Age	Marks	City
0	John	20.0	85.0	new york
1	Jane	NaN	90.0	boston
2	Tom	21.0	NaN	unknown
3	John	20.0	85.0	new york
4	Alice	NaN	88.0	boston
5	Bob	23.0	70.0	boston

## PANDAS JOINS IN PYTHON :

Pandas supports the following joins similar to SQL:

- Inner Join
- Left Join
- Right Join
- Outer Join

EXPLORER

WELCOME

student\_data.csv

pandas data cleaning programs.py

pandas joins.py

PYTHON CODING ASSESS...

pandas data cleaning ...

pandas joins.py

student\_data.csv

```
1 import pandas as pd
2
3 # Load the original dataset
4 df = pd.read_csv("student_data.csv")
5
6 # Clean the 'Age' column to ensure numeric data
7 df['Age'] = pd.to_numeric(df['Age'], errors='coerce')
8
9 # Create two smaller DataFrames:
10 # One with Name, Age, and City
11 student_info = df[['Name', 'Age', 'City']].drop_duplicates()
12
13 # Another with Name and Marks
14 student_marks = df[['Name', 'Marks']].drop_duplicates()
15
16 # INNER JOIN on 'Name'
17 inner = pd.merge(student_info, student_marks, on='Name', how='inner')
18 print("\n=== INNER JOIN ===\n", inner)
19
20 # LEFT JOIN on 'Name'
21 left = pd.merge(student_info, student_marks, on='Name', how='left')
22 print("\n=== LEFT JOIN ===\n", left)
23
24 # OUTER JOIN on 'Name'
25 outer = pd.merge(student_info, student_marks, on='Name', how='outer')
26 print("\n=== OUTER JOIN ===\n", outer)
27
```

PROBLEMS

OUTPUT

DEBUG CONSOLE

TERMINAL

PORTS

```
=== INNER JOIN ===
   Name  Age  City  Marks
0  John  20.0  New York  85.0
1  Jane  NaN  Boston  90.0
2   Tom  21.0   NaN   NaN
3  Alice  NaN  Boston  88.0
4   Bob  23.0  Boston  70.0

=== LEFT JOIN ===
   Name  Age  City  Marks
0  John  20.0  New York  85.0
1  Jane  NaN  Boston  90.0
2   Tom  21.0   NaN   NaN
3  Alice  NaN  Boston  88.0
4   Bob  23.0  Boston  70.0

=== OUTER JOIN ===
   Name  Age  City  Marks
0  Alice  NaN  Boston  88.0
1   Bob  23.0  Boston  70.0
2  Jane  NaN  Boston  90.0
3  John  20.0  New York  85.0
4   Tom  21.0   NaN   NaN
```