Course Name:	**Data Analysis Laboratory (216H03L501)**	**Semester:**	**V**
Date of Performance:	**21/ 07 / 2025**	**DIV/ Batch No:**	**HDA2**
Student Name:	**Aaryan Sharma**	**Roll No:**	**16010123012**

Experiment No: 2

Title: Implement data pre-processing using python on real world dataset

Objectives of the Experiment:

COs to be achieved:

CO1: Understand basic concepts of data analytics to solve real-world problems

Books/ Journals/ Websites referred:

1. https://pandas.pydata.org/docs/getting_started/index.html#getting-started

Theory:

Data preprocessing is the process of cleaning and transforming raw data to make it suitable for analysis and machine learning. Using Pandas, we can handle missing values (dropna(), fillna()), encode categorical data (Label/One-Hot Encoding), normalize numerical features (scaling values to a common range), and discretize continuous data (grouping into categories). It also includes removing duplicates and correcting inconsistent data. These steps improve data quality, consistency, and model performance.

Problem statement/ Tasks

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 >= 30 and age < 40:
        return 'Middle-aged'
    else:
        return 'Old'
 # To view result
df['age_group'] = df['age'].apply(discretize_age)
# 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})

# find employee with maximum salary

df.loc[df['income'].idxmax(), 'name']

#find youngest employee

df.loc[df['age'].idxmax(), 'name']

# Data normalization Min -Max

# Normalize 'income' column to range [0, 1]

min_income = df['income'].min()

max_income = df['income'].max()

df['income_normalized'] = (df['income'] - min_income) / (max_income - min_income)


# Display cleaned, preprocessed, and discretized data

print("\nCleaned, Preprocessed, and Discretized DataFrame:")

print(df)
```

Task: Download the real time data set and implement data preprocessing techniques on the real time data se

Source of the dataset (URL):

Platform used by the student:

Following points should be written by students

Different steps in Data Preprocessing:

- **Finding missing, null values**
- **Replacing missing, null values with statistical parameters**
- **Encoding categorical data if needed (Write user defined function)**
- **Normalization (Write user defined function)**
- **Discretization (Write user defined function)**

Code :

```
def discretize_CGPA(CGPA):
    if CGPA < 4.0:
        return 'Low'
    elif CGPA >= 4.0 and CGPA < 8.0:
        return 'Medium'
    else:
        return 'High'

csp['cgpa_group'] = csp['CGPA'].apply(discretize_CGPA)
csp
```

	College_ID	IQ	Prev_Sem_Result	CGPA	Academic_Performance	Internship_Experience	Extra_Curricular_Score	Communication_Skills	Projects_Completed	Placement	cgpa_group
0	CLG0030	107	6.61	6.28	8	No	8	8	4	No	Medium
1	CLG0061	97	5.52	5.37	8	No	7	8	0	No	Medium
2	CLG0036	109	5.36	5.83	9	No	3	1	1	No	Medium
3	CLG0055	122	5.47	5.75	6	Yes	1	6	1	No	Medium
4	CLG0004	96	7.91	7.69	7	No	8	10	2	No	Medium
...	...	...	...	...	...	...	...	...	...	...	...
9995	CLG0021	119	8.41	8.29	4	No	1	8	0	Yes	High
9996	CLG0098	70	9.25	9.34	7	No	0	7	2	No	High
9997	CLG0066	89	6.08	6.25	3	Yes	3	9	5	No	Medium
9998	CLG0045	107	8.77	8.92	3	No	7	5	1	No	High
9999	CLG0060	109	9.41	9.77	8	No	3	5	5	No	High

10000 rows × 11 columns

```
[15] print("Missing values:\n", csp.isnull().sum())
```

```
Missing values:
 College_ID                0
IQ                         0
Prev_Sem_Result            0
CGPA                       0
Academic_Performance       0
Internship_Experience      0
Extra_Curricular_Score     0
Communication_Skills       0
Projects_Completed         0
Placement                  0
cgpa_group                 0
dtype: int64
```

```
csp = pd.read_csv(os.path.join(path, "college_student_placement_dataset.csv"))
csp['Internship_Experience'] = csp['Internship_Experience'].map({'No': 0, 'Yes': 1})
csp
```

	College_ID	IQ	Prev_Sem_Result	CGPA	Academic_Performance	Internship_Experience	Extra_Curricular_Score	Communication_Skills	Projects_Completed	Placement
0	CLG0030	107	6.61	6.28	8	0	8	8	4	No
1	CLG0061	97	5.52	5.37	8	0	7	8	0	No
2	CLG0036	109	5.36	5.83	9	0	3	1	1	No
3	CLG0055	122	5.47	5.75	6	1	1	6	1	No
4	CLG0004	96	7.91	7.69	7	0	8	10	2	No
...	...	...	...	...	...	...	...	...	...	...
9995	CLG0021	119	8.41	8.29	4	0	1	8	0	Yes
9996	CLG0098	70	9.25	9.34	7	0	0	7	2	No
9997	CLG0066	89	6.08	6.25	3	1	3	9	5	No
9998	CLG0045	107	8.77	8.92	3	0	7	5	1	No
9999	CLG0060	109	9.41	9.77	8	0	3	5	5	No

10000 rows × 10 columns

```
min_Communication_Skills = csp['Communication_Skills'].min()
max_Communication_Skills = csp['Communication_Skills'].max()
csp['Communication_Skills_normalised'] = (csp['Communication_Skills'] - min_Communication_Skills) / (max_Communication_Skills - min_Communication_Skills)

print("\nCleaned, Preprocessed, and Discretized DataFrame:")
print(csp)
```

```
Cleaned, Preprocessed, and Discretized DataFrame:
     College_ID   IQ  Prev_Sem_Result  CGPA  Academic_Performance  \
0       CLG0030  107             6.61  6.28                     8
1       CLG0061   97             5.52  5.37                     8
2       CLG0036  109             5.36  5.83                     9
3       CLG0055  122             5.47  5.75                     6
4       CLG0004   96             7.91  7.69                     7
...         ...  ...              ...   ...                   ...
9995    CLG0021  119             8.41  8.29                     4
9996    CLG0098   70             9.25  9.34                     7
9997    CLG0066   89             6.08  6.25                     3
9998    CLG0045  107             8.77  8.92                     3
9999    CLG0060  109             9.41  9.77                     8

      Internship_Experience  Extra_Curricular_Score  Communication_Skills  \
0                         0                       8                     8
1                         0                       7                     8
2                         0                       3                     1
3                         1                       1                     6
4                         0                       8                    10
...                     ...                     ...                   ...
9995                      0                       1                     8
9996                      0                       0                     7
9997                      1                       3                     9
9998                      0                       7                     5
9999                      0                       3                     5

      Projects_Completed Placement  Communication_Skills_normalised
0                      4        No                         0.777778
1                      0        No                         0.777778
2                      1        No                         0.000000
3                      1        No                         0.555556
4                      2        No                         1.000000
...                  ...       ...                              ...
9995                   0       Yes                         0.777778
9996                   2        No                         0.666667
9997                   5        No                         0.888889
9998                   1        No                         0.444444
9999                   5        No                         0.444444

[10000 rows x 11 columns]
```

Post Lab Subjective/Objective type Questions:

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

Data cleaning is a crucial step in data analysis, and some common challenges include:

Missing Values, Duplicate Records, Inconsistent Data Types, Outliers, Inconsistent Formatting

In the College Student Placement Factors dataset, missing values were handled using Pandas functions:

Used df.isnull().sum() to identify columns with missing values.

For columns with very few missing values, rows were dropped using df.dropna().

Replaced numerical missing values with the mean or median.

Replaced categorical missing values with the most frequent value (mode).

This ensured that the dataset was complete and ready for further analysis without introducing bias.

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

Data normalization scales numerical features to a common range (e.g., 0–1) so all features contribute equally to a machine learning model. It prevents features with large ranges from dominating others, ensuring balanced learning. Normalization improves model accuracy, speeds up convergence, reduces bias, and enhances distance-based calculations, making training more stable and effective.

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

Machine learning models require numerical data because mathematical operations can't be applied to text-based categories. Converting variables like gender into numbers ensures model compatibility, faster processing, and accurate pattern learning. Encoding methods like Label Encoding (assigns numbers) and One-Hot Encoding (creates binary columns) make categorical data usable for analysis and improve model performance.

Conclusion:

I have successfully completed this experiment on data preprocessing using Pandas. I learned how to handle missing values, encode categorical variables, normalize numerical data, and discretize continuous data. This experiment helped me understand the importance of preprocessing to ensure data quality, improve model performance, and derive meaningful insights.