```
Data Profiling:
    df.info()
    df.describe()
    df.isnull().sum()
    df.nunique()
    df.duplicated()
    df.duplicated('Column_name')
    df.drop_duplicates(subset=[Column_name'])


Data profiling is the process of examining, analyzing, and creating
useful summaries of data.
```

Data Profiling: Pandas

In [1]:

```
import pandas as pd
```

In [2]:

```
df=pd.read_csv('/Users/pragatigupta/Documents/AI And ML/Linkedin Post/
```

In [3]:

```
#df.head()
# to see the full dataset '=
pd.set_option("display.max_rows",None)
df
```

Out[3]:

	ID	Student_ID	Gender	AGE	Score	CLASS
0	1.0	17975	F	15	6.7	y
1	2.0	34221	M	16	6.5	y
2	3.0	47975	F	17	5.5	y
3	4.0	87656	F	14	6.8	y
4	5.0	34223	M	15	7.1	y
5	6.0	34224	F	16	2.3	N
6	7.0	34225	F	17	2.0	n
7	8.0	34227	M	15	4.7	N
8	9.0	34229	M	16	2.6	N
9	10.0	34230	F	17	6.7	y
10	11.0	34231	F	14	6.5	Y
11	NaN	87656	F	14	6.8	y
12	2.0	34221	M	16	6.5	y
13	14.0	34224	F	16	2.3	N

14	15.0	34235	F	14	3.5	N
15	16.0	34236	M	15	5.5	y
16	17.0	34237	F	16	5.9	y
17	18.0	87654	F	17	6.7	y
18	19.0	34238	F	15	6.5	y
19	20.0	34239	F	16	5.5	Y
20	21.0	Null	F	17	6.8	Y
21	22.0	12744	F	14	7.1	y
22	23.0	34302	F	15	6.5	y
23	24.0	NaN	M	16	5.5	Y
24	25.0	34242	F	17	6.8	y
25	26.0	46675	F	15	6.7	y
26	27.0	45566	M	16	6.5	y
27	28.0	34309	M	17	5.5	y
28	29.0	87664	M	14	6.8	Y
29	30.0	34245	F	15	7.1	y

# DATA TYPES

#Data Types and Formats ID is float i.w we need to change it to int

```
In [4]: info=df.info()
        info
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 30 entries, 0 to 29
Data columns (total 6 columns):
 #   Column      Non-Null Count  Dtype  
---  ------      --------------  -----  
 0   ID          29 non-null     float64
 1   Student_ID  29 non-null     object 
 2   Gender      30 non-null     object 
 3   AGE         30 non-null     int64  
 4   Score       30 non-null     float64
 5   CLASS       30 non-null     object 
dtypes: float64(2), int64(1), object(3)
memory usage: 1.5+ KB
```

In [9]:

```
# Convert the float ID column to int
#df['ID'] = df['ID'].astype(int) >>>>>>>>>> wll give error bcz we havnt



# Replace NaN values with a specific integer (e.g., 0)
df['ID'] = df['ID'].fillna(0).astype(int)
info=df.info()
info
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 30 entries, 0 to 29
Data columns (total 6 columns):
 #   Column      Non-Null Count  Dtype  
---  ------      --------------  -----  
 0   ID          30 non-null     int64  
 1   Student_ID  29 non-null     object 
 2   Gender      30 non-null     object 
 3   AGE         30 non-null     int64  
 4   Score       30 non-null     float64
 5   CLASS       30 non-null     object 
dtypes: float64(1), int64(2), object(3)
memory usage: 1.5+ KB
```

#ID is float i.w we need to change it to int

In [51]:

```
# Get basic statistics
summary = df.describe()
summary
```

Out[51]:

	ID	AGE	Score
count	29.000000	30.000000	30.000000
mean	15.241379	15.566667	5.730000
std	9.276141	1.072648	1.578334
min	1.000000	14.000000	2.000000
25%	7.000000	15.000000	5.500000
50%	16.000000	16.000000	6.500000
75%	23.000000	16.000000	6.775000
max	30.000000	17.000000	7.100000

Finidngs : Age_Group is between 14 -17 , Scores Avg 5.7 (Max Pass) , ID count 29 (one Id is must be missing or Null)

In [52]:

```
# Count unique values for each column
unique_counts = df.nunique()
unique_counts
```

Out[52]:

```
ID            28
Student_ID    26
Gender         2
AGE            4
Score         11
CLASS          8
dtype: int64
```

In [53]:

```
# Check for missing values
missing_values = df.isnull().sum()
missing_values
```

Out[53]:

```
ID            1
Student_ID    1
Gender        0
AGE           0
Score         0
CLASS         0
dtype: int64
```

In [54]:

```
# Check for duplicate rows
duplicates = df[df.duplicated()]
duplicates
```

Out[54]:

	ID	Student_ID	Gender	AGE	Score	CLASS
12	2.0	34221	M	16	6.5	y

In [56]:

```
# Check for duplicate rows
duplicates = df[df['Student_ID'].duplicated()]
duplicates
```

Out[56]:

	ID	Student_ID	Gender	AGE	Score	CLASS
11	NaN	87656	F	14	6.8	y
12	2.0	34221	M	16	6.5	y
13	14.0	34224	F	16	2.3	N

In [58]:

```
# Drop duplicates based on the ''Student ID'' column
df_no_duplicates_Student_ID = df.drop_duplicates(subset=['Student_ID']
print(df_no_duplicates_Student_ID)
```

```
      ID Student_ID Gender  AGE  Score CLASS
0    1.0      17975      F   15    6.7    y
1    2.0      34221      M   16    6.5    y
2    3.0      47975      F   17    5.5    y
3    4.0      87656      F   14    6.8    y
4    5.0      34223      M   15    7.1    y
5    6.0      34224      F   16    2.3     N
6    7.0      34225      F   17    2.0     n
7    8.0      34227      M   15    4.7     N
8    9.0      34229      M   16    2.6    N
9   10.0      34230      F   17    6.7    y
10  11.0      34231      F   14    6.5    Y
14  15.0      34235      F   14    3.5     N
15  16.0      34236      M   15    5.5    y
16  17.0      34237      F   16    5.9    y
17  18.0      87654      F   17    6.7    y
18  19.0      34238      F   15    6.5    y
19  20.0      34239      F   16    5.5     Y
20  21.0       Null      F   17    6.8    Y
21  22.0      12744      F   14    7.1     y
22  23.0      34302      F   15    6.5    y
23  24.0        NaN      M   16    5.5     Y
24  25.0      34242      F   17    6.8    y
25  26.0      46675      F   15    6.7    y
26  27.0      45566      M   16    6.5    y
27  28.0      34309      M   17    5.5    y
28  29.0      87664      M   14    6.8     Y
29  30.0      34245      F   15    7.1    y
```

In [ ]:

In [ ]: