In [1]:
```
import pandas as pd
import numpy as np
import sklearn as sk
import warnings
warnings.filterwarnings(action='ignore')
```

In [5]:
```
df=pd.read_csv(r"C:\Users\Shree\Downloads\A2.csv")
df
```

Out[5]:

	Roll No	Name	Subject 1	Subject 2	Subject 3	Subject 4	Attendance
0	1	Student_1	100	62	73.0	92.0	96
1	2	Student_2	72	97	82.0	NaN	78
2	3	Student_3	100	88	71.0	99.0	-94
3	4	Student_4	72	99	NaN	84.0	86
4	5	Student_5	97	70	84.0	70.0	86
...	...	...	...	...	...	...	...
996	997	Student_997	88	68	84.0	66.0	98
997	998	Student_998	61	96	62.0	84.0	83
998	999	Student_999	72	76	90.0	72.0	90
999	1000	Student_1000	68	87	100.0	76.0	79
1000	1	Student_1	100	62	73.0	92.0	96

1001 rows × 7 columns

In [7]:
```
df.shape
```

Out[7]:
```
(1001, 7)
```

In [9]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1001 entries, 0 to 1000
Data columns (total 7 columns):
 #   Column      Non-Null Count  Dtype  
---  ------      --------------  -----  
 0   Roll No     1001 non-null   int64  
 1   Name        1001 non-null   object 
 2   Subject 1   1001 non-null   int64  
 3   Subject 2   1001 non-null   int64  
 4   Subject 3   1000 non-null   float64
 5   Subject 4   1000 non-null   float64
 6   Attendance  1001 non-null   int64  
dtypes: float64(2), int64(4), object(1)
memory usage: 54.9+ KB
```

In [11]:
```
df.describe()
```

Out[11]:

	Roll No	Subject 1	Subject 2	Subject 3	Subject 4	Attendance
count	1001.000000	1001.000000	1001.000000	1000.000000	1000.000000	1001.000000
mean	500.000999	79.233766	80.064935	79.890000	80.545000	85.416583
std	289.106384	12.085913	11.904318	11.539457	11.793688	10.277319
min	1.000000	60.000000	60.000000	50.000000	40.000000	-94.000000
25%	250.000000	69.000000	70.000000	70.000000	71.000000	79.000000
50%	500.000000	79.000000	80.000000	80.000000	81.000000	86.000000
75%	750.000000	90.000000	91.000000	90.000000	91.000000	93.000000
max	1000.000000	100.000000	100.000000	100.000000	100.000000	100.000000

In [13]:
```
df.head()
```

Out[13]:

	Roll No	Name	Subject 1	Subject 2	Subject 3	Subject 4	Attendance
0	1	Student_1	100	62	73.0	92.0	96
1	2	Student_2	72	97	82.0	NaN	78
2	3	Student_3	100	88	71.0	99.0	-94
3	4	Student_4	72	99	NaN	84.0	86
4	5	Student_5	97	70	84.0	70.0	86

In [15]:
```
df.isnull()
```

Out[15]:

	Roll No	Name	Subject 1	Subject 2	Subject 3	Subject 4	Attendance
0	False	False	False	False	False	False	False
1	False	False	False	False	False	True	False
2	False	False	False	False	False	False	False
3	False	False	False	False	True	False	False
4	False	False	False	False	False	False	False
...	...	...	...	...	...	...	...
996	False	False	False	False	False	False	False
997	False	False	False	False	False	False	False
998	False	False	False	False	False	False	False
999	False	False	False	False	False	False	False
1000	False	False	False	False	False	False	False

1001 rows × 7 columns

In [17]:
```
df.isnull().sum()
```

Out[17]:
```
Roll No       0
Name          0
Subject 1     0
Subject 2     0
Subject 3     1
Subject 4     1
Attendance    0
dtype: int64
```

In [19]:
```
df.isnull().sum().sum()
```

Out[19]:
```
2
```

In [21]:
```
df1=df.fillna(value=0)
df1
```

Out[21]:

	Roll No	Name	Subject 1	Subject 2	Subject 3	Subject 4	Attendance
0	1	Student_1	100	62	73.0	92.0	96
1	2	Student_2	72	97	82.0	0.0	78
2	3	Student_3	100	88	71.0	99.0	-94
3	4	Student_4	72	99	0.0	84.0	86
4	5	Student_5	97	70	84.0	70.0	86
...	...	...	...	...	...	...	...
996	997	Student_997	88	68	84.0	66.0	98
997	998	Student_998	61	96	62.0	84.0	83
998	999	Student_999	72	76	90.0	72.0	90
999	1000	Student_1000	68	87	100.0	76.0	79
1000	1	Student_1	100	62	73.0	92.0	96

1001 rows × 7 columns

In [23]:
```
df2=df.fillna(method='pad')#pervious
df2
```

Out[23]:

	Roll No	Name	Subject 1	Subject 2	Subject 3	Subject 4	Attendance
0	1	Student_1	100	62	73.0	92.0	96
1	2	Student_2	72	97	82.0	92.0	78
2	3	Student_3	100	88	71.0	99.0	-94
3	4	Student_4	72	99	71.0	84.0	86
4	5	Student_5	97	70	84.0	70.0	86
...	...	...	...	...	...	...	...
996	997	Student_997	88	68	84.0	66.0	98
997	998	Student_998	61	96	62.0	84.0	83
998	999	Student_999	72	76	90.0	72.0	90
999	1000	Student_1000	68	87	100.0	76.0	79
1000	1	Student_1	100	62	73.0	92.0	96

1001 rows × 7 columns

In [25]:
```
df2=df.fillna(method='bfill')#backtrack
df2
```

Out[25]:

	Roll No	Name	Subject 1	Subject 2	Subject 3	Subject 4	Attendance
0	1	Student_1	100	62	73.0	92.0	96
1	2	Student_2	72	97	82.0	99.0	78
2	3	Student_3	100	88	71.0	99.0	-94
3	4	Student_4	72	99	84.0	84.0	86
4	5	Student_5	97	70	84.0	70.0	86
...	...	...	...	...	...	...	...
996	997	Student_997	88	68	84.0	66.0	98
997	998	Student_998	61	96	62.0	84.0	83
998	999	Student_999	72	76	90.0	72.0	90
999	1000	Student_1000	68	87	100.0	76.0	79
1000	1	Student_1	100	62	73.0	92.0	96

1001 rows × 7 columns

In [27]:
```
df3=df.fillna(method='bfill',axis=1)#back(axis 1 =x axis)d
df3
```

Out[27]:

	Roll No	Name	Subject 1	Subject 2	Subject 3	Subject 4	Attendance
0	1	Student_1	100	62	73.0	92.0	96
1	2	Student_2	72	97	82.0	78	78
2	3	Student_3	100	88	71.0	99.0	-94
3	4	Student_4	72	99	84.0	84.0	86
4	5	Student_5	97	70	84.0	70.0	86
...	...	...	...	...	...	...	...
996	997	Student_997	88	68	84.0	66.0	98
997	998	Student_998	61	96	62.0	84.0	83
998	999	Student_999	72	76	90.0	72.0	90
999	1000	Student_1000	68	87	100.0	76.0	79
1000	1	Student_1	100	62	73.0	92.0	96

1001 rows × 7 columns

In [29]:
```
df4=df.fillna(method='ffill',axis=1)#previous
df4
```

Out[29]:

	Roll No	Name	Subject 1	Subject 2	Subject 3	Subject 4	Attendance
0	1	Student_1	100	62	73.0	92.0	96
1	2	Student_2	72	97	82.0	82.0	78
2	3	Student_3	100	88	71.0	99.0	-94
3	4	Student_4	72	99	99	84.0	86
4	5	Student_5	97	70	84.0	70.0	86
...	...	...	...	...	...	...	...
996	997	Student_997	88	68	84.0	66.0	98
997	998	Student_998	61	96	62.0	84.0	83
998	999	Student_999	72	76	90.0	72.0	90
999	1000	Student_1000	68	87	100.0	76.0	79
1000	1	Student_1	100	62	73.0	92.0	96

1001 rows × 7 columns

In [31]:
```
df5=df.fillna(method='pad',axis=1)#same as ffill
df5
```

Out[31]:

	Roll No	Name	Subject 1	Subject 2	Subject 3	Subject 4	Attendance
0	1	Student_1	100	62	73.0	92.0	96
1	2	Student_2	72	97	82.0	82.0	78
2	3	Student_3	100	88	71.0	99.0	-94
3	4	Student_4	72	99	99	84.0	86
4	5	Student_5	97	70	84.0	70.0	86
...	...	...	...	...	...	...	...
996	997	Student_997	88	68	84.0	66.0	98
997	998	Student_998	61	96	62.0	84.0	83
998	999	Student_999	72	76	90.0	72.0	90
999	1000	Student_1000	68	87	100.0	76.0	79
1000	1	Student_1	100	62	73.0	92.0	96

1001 rows × 7 columns

In [37]:
```
df6=df.fillna({'Subject 4':'abcd','Subject 3':'dcba'})
df6
```

Out[37]:

	Roll No	Name	Subject 1	Subject 2	Subject 3	Subject 4	Attendance
0	1	Student_1	100	62	73.0	92.0	96
1	2	Student_2	72	97	82.0	abcd	78
2	3	Student_3	100	88	71.0	99.0	-94
3	4	Student_4	72	99	dcba	84.0	86
4	5	Student_5	97	70	84.0	70.0	86
...	...	...	...	...	...	...	...
996	997	Student_997	88	68	84.0	66.0	98
997	998	Student_998	61	96	62.0	84.0	83
998	999	Student_999	72	76	90.0	72.0	90
999	1000	Student_1000	68	87	100.0	76.0	79
1000	1	Student_1	100	62	73.0	92.0	96

1001 rows × 7 columns

In [39]:
```
df.describe(include='all')#statistic summary
```

Out[39]:

	Roll No	Name	Subject 1	Subject 2	Subject 3	Subject 4	At
count	1001.000000	1001	1001.000000	1001.000000	1000.000000	1000.000000	10
unique	NaN	1000	NaN	NaN	NaN	NaN	
top	NaN	Student_1	NaN	NaN	NaN	NaN	
freq	NaN	2	NaN	NaN	NaN	NaN	
mean	500.000999	NaN	79.233766	80.064935	79.890000	80.545000	
std	289.106384	NaN	12.085913	11.904318	11.539457	11.793688	
min	1.000000	NaN	60.000000	60.000000	50.000000	40.000000	-'
25%	250.000000	NaN	69.000000	70.000000	70.000000	71.000000	
50%	500.000000	NaN	79.000000	80.000000	80.000000	81.000000	
75%	750.000000	NaN	90.000000	91.000000	90.000000	91.000000	
max	1000.000000	NaN	100.000000	100.000000	100.000000	100.000000	1(

In [41]:
```
df.dtypes
```

Out[41]:
```
Roll No         int64
Name           object
Subject 1       int64
Subject 2       int64
Subject 3     float64
Subject 4     float64
Attendance      int64
dtype: object
```

In [43]:
```
df7=df.fillna(method='pad',axis=0)#y axis
df7
```

Out[43]:

	Roll No	Name	Subject 1	Subject 2	Subject 3	Subject 4	Attendance
0	1	Student_1	100	62	73.0	92.0	96
1	2	Student_2	72	97	82.0	92.0	78
2	3	Student_3	100	88	71.0	99.0	-94
3	4	Student_4	72	99	71.0	84.0	86
4	5	Student_5	97	70	84.0	70.0	86
...	...	...	...	...	...	...	...
996	997	Student_997	88	68	84.0	66.0	98
997	998	Student_998	61	96	62.0	84.0	83
998	999	Student_999	72	76	90.0	72.0	90
999	1000	Student_1000	68	87	100.0	76.0	79
1000	1	Student_1	100	62	73.0	92.0	96

1001 rows × 7 columns

In [49]:
```
df8=df.fillna(value=df['Subject 3'].mean())#nan replace by mean of subject 3
df8
```

Out[49]:

	Roll No	Name	Subject 1	Subject 2	Subject 3	Subject 4	Attendance
0	1	Student_1	100	62	73.00	92.00	96
1	2	Student_2	72	97	82.00	79.89	78
2	3	Student_3	100	88	71.00	99.00	-94
3	4	Student_4	72	99	79.89	84.00	86
4	5	Student_5	97	70	84.00	70.00	86
...	...	...	...	...	...	...	...
996	997	Student_997	88	68	84.00	66.00	98
997	998	Student_998	61	96	62.00	84.00	83
998	999	Student_999	72	76	90.00	72.00	90
999	1000	Student_1000	68	87	100.00	76.00	79
1000	1	Student_1	100	62	73.00	92.00	96

1001 rows × 7 columns

In [51]:
```
df9=df.dropna()#remove row where nan is present
df9
```

Out[51]:

	Roll No	Name	Subject 1	Subject 2	Subject 3	Subject 4	Attendance
0	1	Student_1	100	62	73.0	92.0	96
2	3	Student_3	100	88	71.0	99.0	-94
4	5	Student_5	97	70	84.0	70.0	86
5	6	Student_6	98	76	89.0	92.0	82
6	7	Student_7	61	64	97.0	98.0	83
...	...	...	...	...	...	...	...
996	997	Student_997	88	68	84.0	66.0	98
997	998	Student_998	61	96	62.0	84.0	83
998	999	Student_999	72	76	90.0	72.0	90
999	1000	Student_1000	68	87	100.0	76.0	79
1000	1	Student_1	100	62	73.0	92.0	96

999 rows × 7 columns

In [53]:
```
df10=df.dropna(how='any')
df10
```

Out[53]:

	Roll No	Name	Subject 1	Subject 2	Subject 3	Subject 4	Attendance
0	1	Student_1	100	62	73.0	92.0	96
2	3	Student_3	100	88	71.0	99.0	-94
4	5	Student_5	97	70	84.0	70.0	86
5	6	Student_6	98	76	89.0	92.0	82
6	7	Student_7	61	64	97.0	98.0	83
...	...	...	...	...	...	...	...
996	997	Student_997	88	68	84.0	66.0	98
997	998	Student_998	61	96	62.0	84.0	83
998	999	Student_999	72	76	90.0	72.0	90
999	1000	Student_1000	68	87	100.0	76.0	79
1000	1	Student_1	100	62	73.0	92.0	96

999 rows × 7 columns

In [55]:
```
df11=df.dropna(how='all')
df11
```

Out[55]:

	Roll No	Name	Subject 1	Subject 2	Subject 3	Subject 4	Attendance
0	1	Student_1	100	62	73.0	92.0	96
1	2	Student_2	72	97	82.0	NaN	78
2	3	Student_3	100	88	71.0	99.0	-94
3	4	Student_4	72	99	NaN	84.0	86
4	5	Student_5	97	70	84.0	70.0	86
...	...	...	...	...	...	...	...
996	997	Student_997	88	68	84.0	66.0	98
997	998	Student_998	61	96	62.0	84.0	83
998	999	Student_999	72	76	90.0	72.0	90
999	1000	Student_1000	68	87	100.0	76.0	79
1000	1	Student_1	100	62	73.0	92.0	96

1001 rows × 7 columns

In [57]:
```
df.shape
```

Out[57]:
```
(1001, 7)
```

In [67]:
```
df9.shape
```

Out[67]:
```
(999, 7)
```

In [69]:
```
df6.shape
```

Out[69]:
```
(1001, 7)
```

In [71]:
```
df1.dtypes
```

Out[71]:
```
Roll No         int64
Name           object
Subject 1       int64
Subject 2       int64
Subject 3     float64
Subject 4     float64
Attendance      int64
dtype: object
```

In [79]:
```
numeric_cols=['Subject 1','Subject 2','Subject 3','Subject 4','Attendance']
df[numeric_cols]=df[numeric_cols].fillna(method='ffill')
df[numeric_cols]=(df[numeric_cols]-df[numeric_cols].min())/(df[numeric_cols].max
print(df)
```

```
      Roll No          Name  Subject 1  Subject 2  Subject 3  Subject 4  \
0           1     Student_1      1.000      0.050       0.46   0.866667
1           2     Student_2      0.300      0.925       0.64   0.866667
2           3     Student_3      1.000      0.700       0.42   0.983333
3           4     Student_4      0.300      0.975       0.42   0.733333
4           5     Student_5      0.925      0.250       0.68   0.500000
...       ...           ...        ...        ...        ...        ...
996       997   Student_997      0.700      0.200       0.68   0.433333
997       998   Student_998      0.025      0.900       0.24   0.733333
998       999   Student_999      0.300      0.400       0.80   0.533333
999      1000  Student_1000      0.200      0.675       1.00   0.600000
1000        1     Student_1      1.000      0.050       0.46   0.866667

      Attendance
0       0.979381
1       0.886598
2       0.000000
3       0.927835
4       0.927835
...          ...
996     0.989691
997     0.912371
998     0.948454
999     0.891753
1000    0.979381

[1001 rows x 7 columns]
```

In [81]:
```
print(df[numeric_cols].min())
```

```
Subject 1     0.0
Subject 2     0.0
Subject 3     0.0
Subject 4     0.0
Attendance    0.0
dtype: float64
```

In [83]:
```
print(df[numeric_cols].max())
```

```
Subject 1     1.0
Subject 2     1.0
Subject 3     1.0
Subject 4     1.0
Attendance    1.0
dtype: float64
```

In [97]:
```
from sklearn.preprocessing import LabelEncoder
le=LabelEncoder()
df['Name_encoded']=le.fit_transform(df['Name'])
df
```

Out[97]:

	Roll No	Name	Subject 1	Subject 2	Subject 3	Subject 4	Attendance	Name_enco
0	1	Student_1	1.000	0.050	0.46	0.866667	0.979381	
1	2	Student_2	0.300	0.925	0.64	0.866667	0.886598	
2	3	Student_3	1.000	0.700	0.42	0.983333	0.000000	
3	4	Student_4	0.300	0.975	0.42	0.733333	0.927835	
4	5	Student_5	0.925	0.250	0.68	0.500000	0.927835	
...	...	...	...	...	...	...	...	
996	997	Student_997	0.700	0.200	0.68	0.433333	0.989691	
997	998	Student_998	0.025	0.900	0.24	0.733333	0.912371	
998	999	Student_999	0.300	0.400	0.80	0.533333	0.948454	
999	1000	Student_1000	0.200	0.675	1.00	0.600000	0.891753	
1000	1	Student_1	1.000	0.050	0.46	0.866667	0.979381	

1001 rows × 8 columns

In [101…

```
from sklearn.preprocessing import LabelEncoder
lr=LabelEncoder()
df['Name_encoded']=le.fit_transform(df['Name'])
df
```

Out[101…

	Roll No	Name	Subject 1	Subject 2	Subject 3	Subject 4	Attendance	Name_enco
0	1	Student_1	1.000	0.050	0.46	0.866667	0.979381	
1	2	Student_2	0.300	0.925	0.64	0.866667	0.886598	
2	3	Student_3	1.000	0.700	0.42	0.983333	0.000000	
3	4	Student_4	0.300	0.975	0.42	0.733333	0.927835	
4	5	Student_5	0.925	0.250	0.68	0.500000	0.927835	
...	...	...	...	...	...	...	...	
996	997	Student_997	0.700	0.200	0.68	0.433333	0.989691	
997	998	Student_998	0.025	0.900	0.24	0.733333	0.912371	
998	999	Student_999	0.300	0.400	0.80	0.533333	0.948454	
999	1000	Student_1000	0.200	0.675	1.00	0.600000	0.891753	
1000	1	Student_1	1.000	0.050	0.46	0.866667	0.979381	

1001 rows × 8 columns

In [ ]: