In [3]:
```
#Write a python program to clean and manipulate unstructured text da
#Steps to be performed:
#1. Webscrape the customer review of a product from any e commerce w
#2. Load the dataset containing unstructured textual data
#3. Clean the data by removing special characters, numbers and unnec
#4. Extract key phrases like product name and ratings using regex an
#5. Convert the extracted information into structured dataset with m
#Write a python program, to handle missing data using multiple techn

import pandas as pd
import numpy as np
import re
from sklearn.preprocessing import MinMaxScaler, StandardScaler
data = {
    'review': [
        'Product Name: A Rating: 5 Great product! Highly recommend i
        'Product Name: B Rating: 3 Average quality. Could be better.
        'Product Name: C Rating: 4 Good value for the price.',
        'Product Name: D Rating: 2 Not what I expected.',
        'Product Name: E Rating: 4 Excellent! Will buy again.',
        'Product Name: F Rating: 1 Terrible experience! Do not buy!'
        'Product Name: G Rating: None Worst product ever!']
}
df = pd.DataFrame(data)
def clean_text(text):
    text = re.sub(r'[^a-zA-Z\s:]', '', text)
    text = re.sub(r'\s+', ' ', text).strip()
    return text
df['cleaned_review'] = df['review'].apply(clean_text)
def extract_product_and_rating(text):
    product_name = re.search(r'Product Name:\s*(.*?)\s*Rating:', tex
    rating = re.search(r'Rating:\s*(\d+|None)', text)
    return {
        'product_name': product_name.group(1) if product_name else N
        'rating': None if rating is None else int(rating.group(1)) i
    }
df[['product_name', 'rating']] = df['review'].apply(extract_product_
df.loc[5, 'rating'] = np.nan
forward_filled_df = df.fillna(method='ffill')
backward_filled_df = df.fillna(method='bfill')
min_max_scaler = MinMaxScaler()
structured_df = forward_filled_df[['product_name', 'rating', 'cleane
structured_df['rating_min_max'] = min_max_scaler.fit_transform(struc
standard_scaler = StandardScaler()
structured_df['rating_z_score'] = standard_scaler.fit_transform(stru
```

```
print(structured_df[['product_name', 'rating', 'rating_min_max', 'ra
```

	product_name	rating	rating_min_max	rating_z_score
0	A	5.0	1.000000	1.459993
1	B	3.0	0.333333	-0.811107
2	C	4.0	0.666667	0.324443
3	D	2.0	0.000000	-1.946657
4	E	4.0	0.666667	0.324443
5	F	4.0	0.666667	0.324443
6	G	4.0	0.666667	0.324443

In [4]:
```
#1BM22AI035
#LAB 3
#handling missing data transformation write a python program to hand
#multiple techniques such as mean,median,mode,forward and backward f
#apply data transformation methods such as min-max scaling and z sco
#analyze how different techniques affect the data set
import numpy as np
import pandas as pd
from sklearn.preprocessing import MinMaxScaler, StandardScaler
data={
    'A':[1,2,np.nan,4,5,np.nan,7,8,9,10,11,np.nan,13,14,15],
    'B':[np.nan,1,2,3,np.nan,5,6,7,8,np.nan,10,11,12,np.nan,14],
    'C':[1,2,3,4,5,6,np.nan,8,9,10,np.nan,12,13,14,15],
    'D':[1,2,3,4,5,6,7,8,np.nan,np.nan,np.nan,np.nan,np.nan,np.nan,n
    'G':[np.nan,np.nan,np.nan,np.nan,np.nan,np.nan,np.nan,np.nan,np.
}
df=pd.DataFrame(data)
mean_filled=df.fillna(df.mean())
median_filled=df.fillna(df.median())
mode_filled=df.fillna(df.mode().iloc[0])
forward_filled=df.fillna(method='ffill')
backward_filled=df.fillna(method='bfill')
scaler_minmax=MinMaxScaler()
scaler_zscore=StandardScaler()
mean_filled_scaled=pd.DataFrame(scaler_minmax.fit_transform(mean_fil
median_filled_scaled=pd.DataFrame(scaler_minmax.fit_transform(median
mode_filled_scaled=pd.DataFrame(scaler_minmax.fit_transform(mode_fil
forward_filled_scaled=pd.DataFrame(scaler_minmax.fit_transform(forwa
backward_filled_scaled=pd.DataFrame(scaler_minmax.fit_transform(back
mean_filled_zscore=pd.DataFrame(scaler_zscore.fit_transform(mean_fil
median_filled_zscore=pd.DataFrame(scaler_zscore.fit_transform(median
mode_filled_zscore=pd.DataFrame(scaler_zscore.fit_transform(mode_fil
forward_filled_zscore=pd.DataFrame(scaler_zscore.fit_transform(forwa
backward_filled_zscore=pd.DataFrame(scaler_zscore.fit_transform(back
results={
    'Mean Filled':mean_filled_scaled,
    'Median Filled':median_filled_scaled,
    'Mode Filled':mode_filled_scaled,
    'Forward Filled':forward_filled_scaled,
    'Backward Filled':backward_filled_scaled,
    'Mean Z-Score':mean_filled_zscore,
    'Median Z-Score':median_filled_zscore,
    'Mode Z-Score':mode_filled_zscore,
    'Forward Z-Score':forward_filled_zscore,
    'Backward Z-Score':backward_filled_zscore
}
for name,result in results.items():
```

```
    print(f"{name}:\n{result}\n")
```

```
Mean Filled:
           A         B         C         D   G
0   0.000000  0.475524  0.000000  0.000000 NaN
1   0.071429  0.000000  0.071429  0.142857 NaN
2   0.517857  0.076923  0.142857  0.285714 NaN
3   0.214286  0.153846  0.214286  0.428571 NaN
4   0.285714  0.475524  0.285714  0.571429 NaN
5   0.517857  0.307692  0.357143  0.714286 NaN
6   0.428571  0.384615  0.489011  0.857143 NaN
7   0.500000  0.461538  0.500000  1.000000 NaN
8   0.571429  0.538462  0.571429  0.500000 NaN
9   0.642857  0.475524  0.642857  0.500000 NaN
10  0.714286  0.692308  0.489011  0.500000 NaN
11  0.517857  0.769231  0.785714  0.500000 NaN
12  0.857143  0.846154  0.857143  0.500000 NaN
13  0.928571  0.475524  0.928571  0.500000 NaN
14  1.000000  1.000000  1.000000  0.500000 NaN

Median Filled:
```

In [ ]: