In [1]:

```
#air quality
#1)data cleaning
#2)data integration
#3)data transformation
#4)error correcting
```

In [3]:

```
import pandas as pd
```

In [33]:

```
df=pd.read_csv(r"C:\Users\sagar\Desktop\air.csv",sep=",")
```

```
C:\Users\sagar\AppData\Local\Temp\ipykernel_17584\4160904067.py:1: DtypeWa
rning: Columns (0) have mixed types. Specify dtype option on import or set
low_memory=False.
  df=pd.read_csv(r"C:\Users\sagar\Desktop\air.csv",sep=",")
```

In [34]:

```
df.head()
```

Out[34]:

	stn_code	sampling_date	state	location	agency	type	so2	no2	rspm	spm
0	150.0	February - M021990	Andhra Pradesh	Hyderabad	NaN	Residential, Rural and other Areas	4.8	17.4	NaN	NaN
1	151.0	February - M021990	Andhra Pradesh	Hyderabad	NaN	Industrial Area	3.1	7.0	NaN	NaN
2	152.0	February - M021990	Andhra Pradesh	Hyderabad	NaN	Residential, Rural and other Areas	6.2	28.5	NaN	NaN
3	150.0	March - M031990	Andhra Pradesh	Hyderabad	NaN	Residential, Rural and other Areas	6.3	14.7	NaN	NaN
4	151.0	March - M031990	Andhra Pradesh	Hyderabad	NaN	Industrial Area	4.7	7.5	NaN	NaN

In [35]:

```
#Data Cleaning
df.isnull()
```

Out[35]:

	stn_code	sampling_date	state	location	agency	type	so2	no2	rspm	spm	l
0	False	False	False	False	True	False	False	False	True	True	
1	False	False	False	False	True	False	False	False	True	True	
2	False	False	False	False	True	False	False	False	True	True	
3	False	False	False	False	True	False	False	False	True	True	
4	False	False	False	False	True	False	False	False	True	True	
...	...	...	...	...	...	...	...	...	...	...	
435737	False	False	False	False	False	False	False	False	False	True	
435738	False	False	False	False	False	False	False	False	False	True	
435739	True	True	False	True	True	True	True	True	True	True	
435740	True	True	False	True	True	True	True	True	True	True	
435741	True	True	False	True	True	True	True	True	True	True	

435742 rows × 13 columns

In [37]:

```
df.dropna(subset=['date'])
```

Out[37]:

	stn_code	sampling_date	state	location	agency	type	so2	no2	rspn
0	150.0	February - M021990	Andhra Pradesh	Hyderabad	NaN	Residential, Rural and other Areas	4.8	17.4	NaN
1	151.0	February - M021990	Andhra Pradesh	Hyderabad	NaN	Industrial Area	3.1	7.0	NaN
2	152.0	February - M021990	Andhra Pradesh	Hyderabad	NaN	Residential, Rural and other Areas	6.2	28.5	NaN
3	150.0	March - M031990	Andhra Pradesh	Hyderabad	NaN	Residential, Rural and other Areas	6.3	14.7	NaN
4	151.0	March - M031990	Andhra Pradesh	Hyderabad	NaN	Industrial Area	4.7	7.5	NaN
...	...	...	...	...	...	...	...	...	...
435734	SAMP	15-12-15	West Bengal	ULUBERIA	West Bengal State Pollution Control Board	RIRUO	20.0	44.0	148.(
435735	SAMP	18-12-15	West Bengal	ULUBERIA	West Bengal State Pollution Control Board	RIRUO	17.0	44.0	131.(
435736	SAMP	21-12-15	West Bengal	ULUBERIA	West Bengal State Pollution Control Board	RIRUO	18.0	45.0	140.(
435737	SAMP	24-12-15	West Bengal	ULUBERIA	West Bengal State Pollution Control Board	RIRUO	22.0	50.0	143.(
435738	SAMP	29-12-15	West Bengal	ULUBERIA	West Bengal State Pollution Control Board	RIRUO	20.0	46.0	171.(

435735 rows × 13 columns

In [38]:

```
df.columns
```

Out[38]:

```
Index(['stn_code', 'sampling_date', 'state', 'location', 'agency', 'type',
       'so2', 'no2', 'rspm', 'spm', 'location_monitoring_station', 'pm2_
5',
       'date'],
      dtype='object')
```

In [39]:

```
df.drop(['so2', 'no2', 'rspm', 'spm'],axis=1)
```

Out[39]:

	stn_code	sampling_date	state	location	agency	type	location_mo
0	150.0	February - M021990	Andhra Pradesh	Hyderabad	NaN	Residential, Rural and other Areas	
1	151.0	February - M021990	Andhra Pradesh	Hyderabad	NaN	Industrial Area	
2	152.0	February - M021990	Andhra Pradesh	Hyderabad	NaN	Residential, Rural and other Areas	
3	150.0	March - M031990	Andhra Pradesh	Hyderabad	NaN	Residential, Rural and other Areas	
4	151.0	March - M031990	Andhra Pradesh	Hyderabad	NaN	Industrial Area	
...	...	...	...	...	...	...	
435737	SAMP	24-12-15	West Bengal	ULUBERIA	West Bengal State Pollution Control Board	RIRUO	Indu
435738	SAMP	29-12-15	West Bengal	ULUBERIA	West Bengal State Pollution Control Board	RIRUO	Indu
435739	NaN	NaN	andaman-and-nicobar-islands	NaN	NaN	NaN	
435740	NaN	NaN	Lakshadweep	NaN	NaN	NaN	
435741	NaN	NaN	Tripura	NaN	NaN	NaN	

435742 rows × 9 columns

In [41]:

```
df.fillna(0)
```

Out[41]:

	stn_code	sampling_date	state	location	agency	type	so2	no2
0	150.0	February - M021990	Andhra Pradesh	Hyderabad	0	Residential, Rural and other Areas	4.8	17.4
1	151.0	February - M021990	Andhra Pradesh	Hyderabad	0	Industrial Area	3.1	7.0
2	152.0	February - M021990	Andhra Pradesh	Hyderabad	0	Residential, Rural and other Areas	6.2	28.5
3	150.0	March - M031990	Andhra Pradesh	Hyderabad	0	Residential, Rural and other Areas	6.3	14.7
4	151.0	March - M031990	Andhra Pradesh	Hyderabad	0	Industrial Area	4.7	7.5
...	...	...	...	...	...	...	...	...
435737	SAMP	24-12-15	West Bengal	ULUBERIA	West Bengal State Pollution Control Board	RIRUO	22.0	50.0
435738	SAMP	29-12-15	West Bengal	ULUBERIA	West Bengal State Pollution Control Board	RIRUO	20.0	46.0
435739	0	0	andaman-and-nicobar-islands	0	0	0	0.0	0.0
435740	0	0	Lakshadweep	0	0	0	0.0	0.0
435741	0	0	Tripura	0	0	0	0.0	0.0

435742 rows × 13 columns

In [42]:

```
#Data Integration
from sklearn.impute import SimpleImputer
import numpy as np
```

In [44]:

```
df.columns
```

Out[44]:

```
Index(['stn_code', 'sampling_date', 'state', 'location', 'agency', 'type',
       'so2', 'no2', 'rspm', 'spm', 'location_monitoring_station', 'pm2_
5',
       'date'],
      dtype='object')
```

In [45]:

```
columns = ['so2', 'no2', 'rspm', 'spm']
```

In [46]:

```
imp = SimpleImputer(missing_values=np.NaN, strategy='mean')
```

In [49]:

```
df[columns] = imp.fit_transform(df[columns])
```

In [51]:

```
df[columns]
```

Out[51]:

	so2	no2	rspm	spm
0	4.800000	17.400000	108.832784	220.78348
1	3.100000	7.000000	108.832784	220.78348
2	6.200000	28.500000	108.832784	220.78348
3	6.300000	14.700000	108.832784	220.78348
4	4.700000	7.500000	108.832784	220.78348
...	...	...	...	...
435737	22.000000	50.000000	143.000000	220.78348
435738	20.000000	46.000000	171.000000	220.78348
435739	10.829414	25.809623	108.832784	220.78348
435740	10.829414	25.809623	108.832784	220.78348
435741	10.829414	25.809623	108.832784	220.78348

435742 rows × 4 columns

In [53]:

```
df[columns].isnull()
```

Out[53]:

	so2	no2	rspm	spm
0	False	False	False	False
1	False	False	False	False
2	False	False	False	False
3	False	False	False	False
4	False	False	False	False
...	...	...	...	...
435737	False	False	False	False
435738	False	False	False	False
435739	False	False	False	False
435740	False	False	False	False
435741	False	False	False	False

435742 rows × 4 columns

In [55]:

```
#Data Transformation
df.columns
```

Out[55]:

```
Index(['stn_code', 'sampling_date', 'state', 'location', 'agency', 'type',
       'so2', 'no2', 'rspm', 'spm', 'location_monitoring_station', 'pm2_
5',
       'date'],
      dtype='object')
```

In [56]:

```
df['type']
```

Out[56]:

```
0         Residential, Rural and other Areas
1                           Industrial Area
2         Residential, Rural and other Areas
3         Residential, Rural and other Areas
4                           Industrial Area
                        ...
435737                                RIRUO
435738                                RIRUO
435739                                  NaN
435740                                  NaN
435741                                  NaN
Name: type, Length: 435742, dtype: object
```

In [57]:

```
df.head()
```

Out[57]:

	stn_code	sampling_date	state	location	agency	type	so2	no2	rspm
0	150.0	February - M021990	Andhra Pradesh	Hyderabad	NaN	Residential, Rural and other Areas	4.8	17.4	108.832784
1	151.0	February - M021990	Andhra Pradesh	Hyderabad	NaN	Industrial Area	3.1	7.0	108.832784
2	152.0	February - M021990	Andhra Pradesh	Hyderabad	NaN	Residential, Rural and other Areas	6.2	28.5	108.832784
3	150.0	March - M031990	Andhra Pradesh	Hyderabad	NaN	Residential, Rural and other Areas	6.3	14.7	108.832784
4	151.0	March - M031990	Andhra Pradesh	Hyderabad	NaN	Industrial Area	4.7	7.5	108.832784

In [58]:

```
#categorical to categorical
df['type'].replace({'Residential, Rural and other Areas':'RRA','Industrial Area':'IA','R
```

In [59]:

```
#categorical to numerical
df['type'].replace({'RRA':1,'IA':2,'RIR':3})
```

Out[59]:

```
0           1
1           2
2           1
3           1
4           2
         ...
435737      3
435738      3
435739    NaN
435740    NaN
435741    NaN
Name: type, Length: 435742, dtype: object
```

In [60]:

```
#Label Encoding
from sklearn.preprocessing import LabelEncoder
```

In [61]:

```
label = LabelEncoder()
```

In [63]:

```
df['state'] = label.fit_transform(df['state'])
```

In [64]:

```
df['state']
```

Out[64]:

```
0          0
1          0
2          0
3          0
4          0
          ..
435737    35
435738    35
435739    36
435740    17
435741    31
Name: state, Length: 435742, dtype: int32
```

In [65]:

```
############## Error Correcting
df.nunique()
```

Out[65]:

```
stn_code                        803
sampling_date                  5485
state                            37
location                        304
agency                           64
type                             10
so2                            4198
no2                            6865
rspm                           6066
spm                            6669
location_monitoring_station     991
pm2_5                           433
date                           5067
dtype: int64
```

In [67]:

```
df['state'].unique()
```

Out[67]:

```
array([ 0,  1,  2,  3,  4,  5,  6,  7,  8,  9, 10, 11, 12, 13, 14, 15, 16,
       18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 32, 33, 34, 35,
       36, 17, 31])
```

In [68]:

```
import numpy as np
```

In [70]:

```
p=df.loc[df['state']==4, 'state']=np.NaN
```

In [71]:

```
p
```

Out[71]:

```
nan
```

In [ ]:

```

```