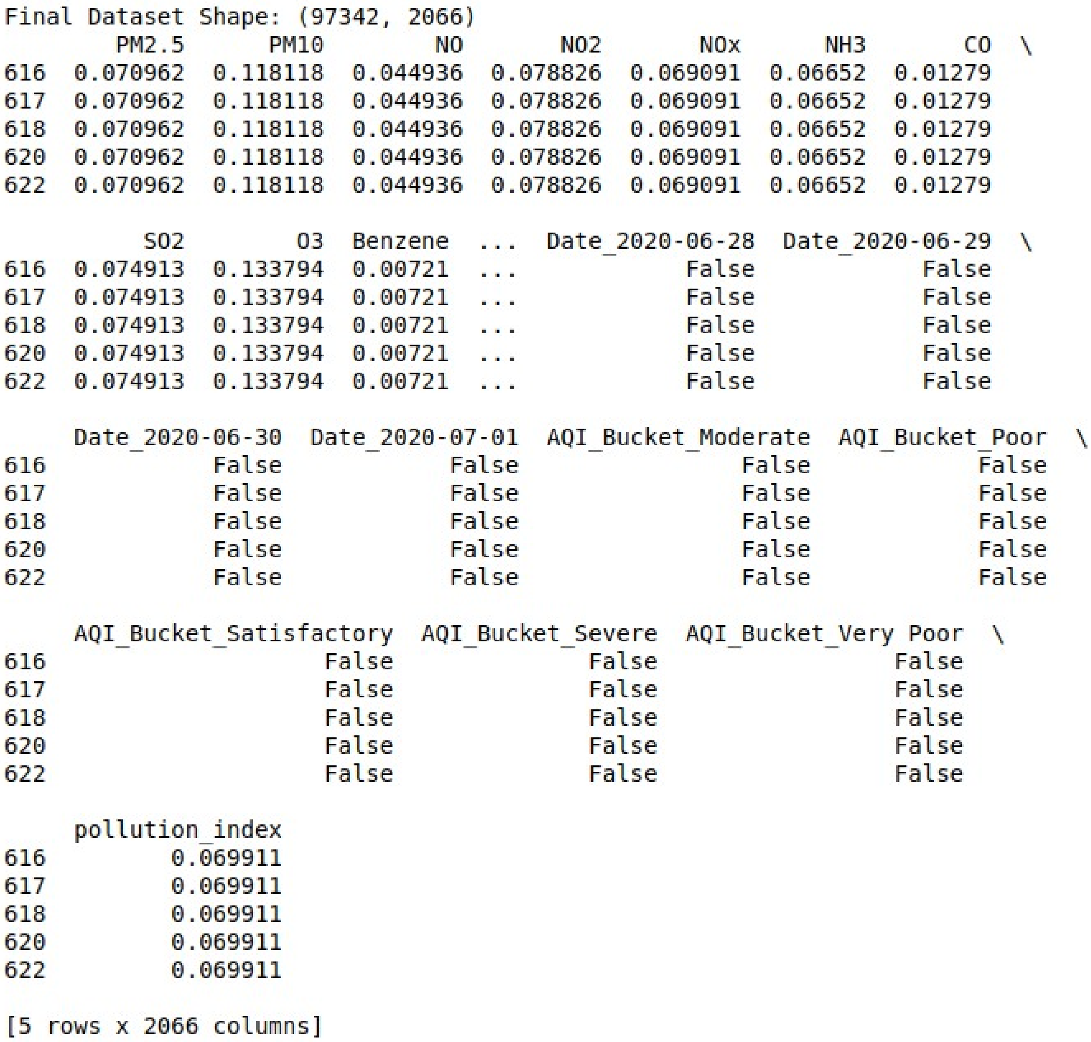
```
import pandas as pd
import numpy as np
from sklearn.preprocessing import MinMaxScaler

# -----------------------------
# Load datasets
# -----------------------------
air = pd.read_csv("air_quality.csv")
heart = pd.read_csv("heart.xls")

# -----------------------------
# (a) Data Cleaning
# -----------------------------

# Remove duplicates
air.drop_duplicates(inplace=True)
heart.drop_duplicates(inplace=True)

# Handle missing values (numeric → mean)
air.fillna(air.mean(numeric_only=True), inplace=True)
heart.fillna(heart.mean(numeric_only=True), inplace=True)

# -----------------------------
# (b) Data Integration
# -----------------------------
# Merge datasets using common column (e.g., 'city')
air["city"] = air["city"].astype(str)
heart["city"] = heart["city"].astype(str)
heart["city"] = np.random.choice(air["city"].unique(), len(heart))
data = pd.merge(air, heart, on="city", how="left")

# -----------------------------
# (c) Data Transformation
# -----------------------------

# Normalize numeric columns
scaler = MinMaxScaler()
num_cols = data.select_dtypes(include=np.number).columns
data[num_cols] = scaler.fit_transform(data[num_cols])

# Encode categorical columns
data = pd.get_dummies(data, drop_first=True)

# Create a new feature (pollution index)
if {"PM10", "NO2", "CO"}.issubset(data.columns):
    data["pollution_index"] = (data["PM10"] + data["NO2"] + data["CO"]) / 3

# -----------------------------
# (d) Error Correcting
# -----------------------------

# Remove invalid ages
```

```
if "age" in data.columns:
    data = data[data["age"] > 0]

# Fix cholesterol out-of-range values
if "cholesterol" in data.columns:
    data["cholesterol"] = data["cholesterol"].clip(0.25, 0.75)

# Remove outliers using IQR
Q1 = data[num_cols].quantile(0.25)
Q3 = data[num_cols].quantile(0.75)
IQR = Q3 - Q1

data = data[~((data[num_cols] < (Q1 - 1.5 * IQR)) |
              (data[num_cols] > (Q3 + 1.5 * IQR))).any(axis=1)]

# -----------------------------
# Final Output
# -----------------------------
print("Final Dataset Shape:", data.shape)
print(data.head())
```

OUTPUT:-



```
Final Dataset Shape: (97342, 2066)
        PM2.5      PM10        NO       NO2       NOx      NH3       CO  \
616  0.070962  0.118118  0.044936  0.078826  0.069091  0.06652  0.01279
617  0.070962  0.118118  0.044936  0.078826  0.069091  0.06652  0.01279
618  0.070962  0.118118  0.044936  0.078826  0.069091  0.06652  0.01279
620  0.070962  0.118118  0.044936  0.078826  0.069091  0.06652  0.01279
622  0.070962  0.118118  0.044936  0.078826  0.069091  0.06652  0.01279

          SO2        O3  Benzene  ...  Date_2020-06-28  Date_2020-06-29  \
616  0.074913  0.133794  0.00721  ...            False            False
617  0.074913  0.133794  0.00721  ...            False            False
618  0.074913  0.133794  0.00721  ...            False            False
620  0.074913  0.133794  0.00721  ...            False            False
622  0.074913  0.133794  0.00721  ...            False            False

     Date_2020-06-30  Date_2020-07-01  AQI_Bucket_Moderate  AQI_Bucket_Poor  \
616            False            False                False            False
617            False            False                False            False
618            False            False                False            False
620            False            False                False            False
622            False            False                False            False

     AQI_Bucket_Satisfactory  AQI_Bucket_Severe  AQI_Bucket_Very Poor  \
616                    False              False                 False
617                    False              False                 False
618                    False              False                 False
620                    False              False                 False
622                    False              False                 False

     pollution_index
616         0.069911
617         0.069911
618         0.069911
620         0.069911
622         0.069911

[5 rows x 2066 columns]
```