First, we need to load the necessary libraries and the datasets. For this example, we'll use Pandas library for data manipulation and Seaborn and Matplotlib libraries for data visualization.

import pandas as pd

import seaborn as sns

import matplotlib.pyplot as plt

air\_quality\_df = pd.read\_csv('air\_quality.csv')

heart\_disease\_df = pd.read\_csv('heart\_disease.csv')

A. Data cleaning

Data cleaning involves identifying and handling missing values, duplicate data, outliers, and other inconsistencies in the data.

For the air quality dataset, we'll check for missing values and remove any columns that have more than 50% missing values.

# Check for missing values

air\_quality\_df.isnull().sum()

# Remove columns with more than 50% missing values

air\_quality\_df.dropna(thresh=0.5\*len(air\_quality\_df), axis=1, inplace=True)

For the heart disease dataset, we'll check for duplicate data and remove any duplicates.

# Check for duplicate data

heart\_disease\_df.duplicated().sum()

# Remove duplicate data

heart\_disease\_df.drop\_duplicates(inplace=True)

B. Data integration

Data integration involves combining data from multiple sources into a single dataset. In this example, we'll merge the air quality dataset and the heart disease dataset on a common column 'age'.

merged\_df = pd.merge(air\_quality\_df, heart\_disease\_df, on='age', how='inner')

C. Data transformation

Data transformation involves converting data into a different format or structure. In this example, we'll transform the heart disease dataset by converting the 'target' column into binary values (0 and 1).

heart\_disease\_df['target'] = heart\_disease\_df['target'].apply(lambda x: 1 if x>0 else 0)

D. Error correcting

Error correcting involves identifying and correcting errors in the data. In this example, we'll correct any inconsistencies in the air quality dataset by replacing any negative values with the mean value of the column.

air\_quality\_df[air\_quality\_df < 0] = air\_quality\_df.mean()

E. Data model building

Data model building involves using statistical and machine learning algorithms to build predictive models. In this example, we'll build a logistic regression model to predict heart disease based on air quality data.

# Split the data into training and testing sets

from sklearn.model\_selection import train\_test\_split

X = merged\_df.drop(['target'], axis=1)

y = merged\_df['target']

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size=0.3, random\_state=42)

# Build the logistic regression model

from sklearn.linear\_model import LogisticRegression

logreg = LogisticRegression()

logreg.fit(X\_train, y\_train)

# Evaluate the model

from sklearn.metrics import classification\_report, confusion\_matrix

y\_pred = logreg.predict(X\_test)

print(confusion\_matrix(y\_test, y\_pred))

print(classification\_report(y\_test, y\_pred))