**PHASE – 3 PROJECT FOR DATA ANALYSTICS**

**TITLE: Loading and preprocessing the given set of data**

Building a dataset typically involves the following steps: data collection, data loading, and data preprocessing. Here's a high-level guide on how to build a dataset:

1. Define Your Dataset: First, you need to determine the purpose and scope of your dataset. What kind of data are you interested in (text, images, numerical data, etc.)? What are you trying to achieve with this dataset?

2. Data Collection: Collect the raw data that you want to include in your dataset. This can involve web scraping, downloading public datasets, or gathering data through sensors or surveys.

3. Data Loading:

a. Choose a Storage Format: Decide on a storage format for your dataset. Common formats include CSV, JSON, XML, or specific formats for images (e.g., JPEG, PNG) or text (e.g., plain text, HTML).

b. Load Data: Use a programming language like Python and relevant libraries (e.g., Pandas for tabular data, NumPy for numerical data, Pillow for images) to load the data into memory. You might also consider using database systems if dealing with large datasets.

c. Data Validation: Check the loaded data for any missing or erroneous values. Data validation is essential to ensure the dataset's integrity.

4. Data Preprocessing:

a. Cleaning Data: Handle missing values, outliers, and inconsistent data. This may involve imputing missing values, removing duplicates, or correcting errors.

b. Feature Extraction: If your dataset contains unstructured data (e.g., text or images), you might need to extract features or convert the data into a format suitable for your analysis or machine learning model.

c. Normalization/Scaling: Scale numerical features if necessary to ensure that they are on a consistent scale.

d. Encoding: Convert categorical variables into numerical form, using techniques like one-hot encoding or label encoding.

e. Splitting Data: Divide your dataset into training, validation, and test sets to evaluate the performance of your machine learning models.

5. Save the Processed Dataset: After preprocessing, save the dataset in a format suitable for your analysis or machine learning tasks. This could be another CSV, HDF5, or any other format that suits your needs.

6. Documentation: Maintain documentation that describes the dataset, its features, and any preprocessing steps you've applied. This is important for sharing your dataset with others.

**Here the source code for loading the dataset**

Import pandas as pd

data = pd. read­\_csv(“C:\Users\91904\Downloads\datafile.csv”)#load a dataset

import pandas as pd

# Specify the path to your Excel file

excel\_file = 'C:\Users\prithivi raj\Desktop\data file.csv'

# Read the Excel file into a DataFrame

df = pd.read\_csv(‘datafile’)

# Display the first few rows of the DataFrame to get an overview of the data

print(df.head())

# Display general information about the DataFrame

print(df.info())

# Display basic statistics about numeric columns

print(df.describe())

# Display the column names

print("Column names:")

print(df.columns)

# Display the number of rows and columns in the DataFrame

print("Shape of the DataFrame:")

print(df.shape)

output:

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Stn Code | | Sampling Date | | State | | City/Town/Village/Area | | | Location of Monitoring Station | Agency | Type of Location | SO2 | NO2 | RSPM/PM10 | PM 2.5 | |
| ######## | | Tamil Nadu | | Chennai | | | Kathivakkam, Municipal Kalyana Mandapam, Chennai | Tamilnadu State Pollution Control Board | | | | | | | | Industrial Area | | 11 | 17 | 55 | NA |
| ######## | | Tamil Nadu | | Chennai | | | Govt. High School, Manali, Chennai. | Tamilnadu State Pollution Control Board | | | | | | | | Industrial Area | | 12 | 14 | 68 | NA |
| ######## | | Tamil Nadu | | Chennai | | | Thiruvottiyur, Chennai | Tamilnadu State Pollution Control Board | | | | | | | | Industrial Area | | 12 | 13 | 49 | NA |
| ######## | | Tamil Nadu | | Chennai | | | Thiyagaraya Nagar, Chennai | Tamilnadu State Pollution Control Board | | | | | | | | Residential, Rural and other Areas | | 16 | 23 | 142 | NA |
| ######## | | Tamil Nadu | | Chennai | | | Anna Nagar, Chennai | Tamilnadu State Pollution Control Board | | | | | | | | Residential, Rural and other Areas | | 17 | 23 | 73 | NA |
| ######## | | Tamil Nadu | | Chennai | | | Adyar, Chennai | Tamilnadu State Pollution Control Board | | | | | | | | Residential, Rural and other Areas | | 14 | 20 | 42 | NA |
| ######## | | Tamil Nadu | | Chennai | | | Kilpauk, Chennai | Tamilnadu State Pollution Control Board | | | | | | | | Residential, Rural and other Areas | | 18 | 23 | 83 | NA |
| ######## | | Tamil Nadu | | Chennai | | | Madras Medical College, Chennai | National Environmental Engineering Research Institute | | | | | | | | Residential, Rural and other Areas | | 15 | 38 | 20 | NA |
| ######## | | Tamil Nadu | | Chennai | | | Thiruvottiyur Municipal Office, Chennai | National Environmental Engineering Research Institute | | | | | | | | Industrial Area | | 14 | 42 | 31 | NA |
| ######## | | Tamil Nadu | | Coimbatore | | | Poniarajapuram, On the top of DEL, Coimbatore | Tamilnadu  State Pollution  Control Board | | | | | | | | Residential, Rural and other Areas | | 4 | 23 | 67 | NA |
| ######## | | Tamil Nadu | | Coimbatore | | | SIDCO Office, Coimbatore | Tamilnadu State Pollution Control Board | | | | | | | | Industrial Area | | 4 | 26 | 75 | NA |
|  |  | |  | |

**Here the source code for preprocessing the dataset**

import pandas as pd

from sklearn.preprocessing import StandardScaler

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

# Load the dataset

data = pd.read\_csv("C:\Users\91904\Downloads\datafile.csv ")

# Data cleaning and handling missing values (if needed)

data = data.drop()

# Split the dataset into features (X) and target (y)

X = data.drop(columns=["city"])

y = data["RSPM/PM10"]

# Split the dataset into training and testing sets

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size=0.2, random\_state=42)

# Feature scaling or normalization (if needed)

scaler = StandardScaler()

X\_train = scaler.fit\_transform(X\_train)

X\_test = scaler.transform(X\_test)

# Your other preprocessing steps here

# Now you have X\_train, X\_test, y\_train, and y\_test ready for your machine learning model