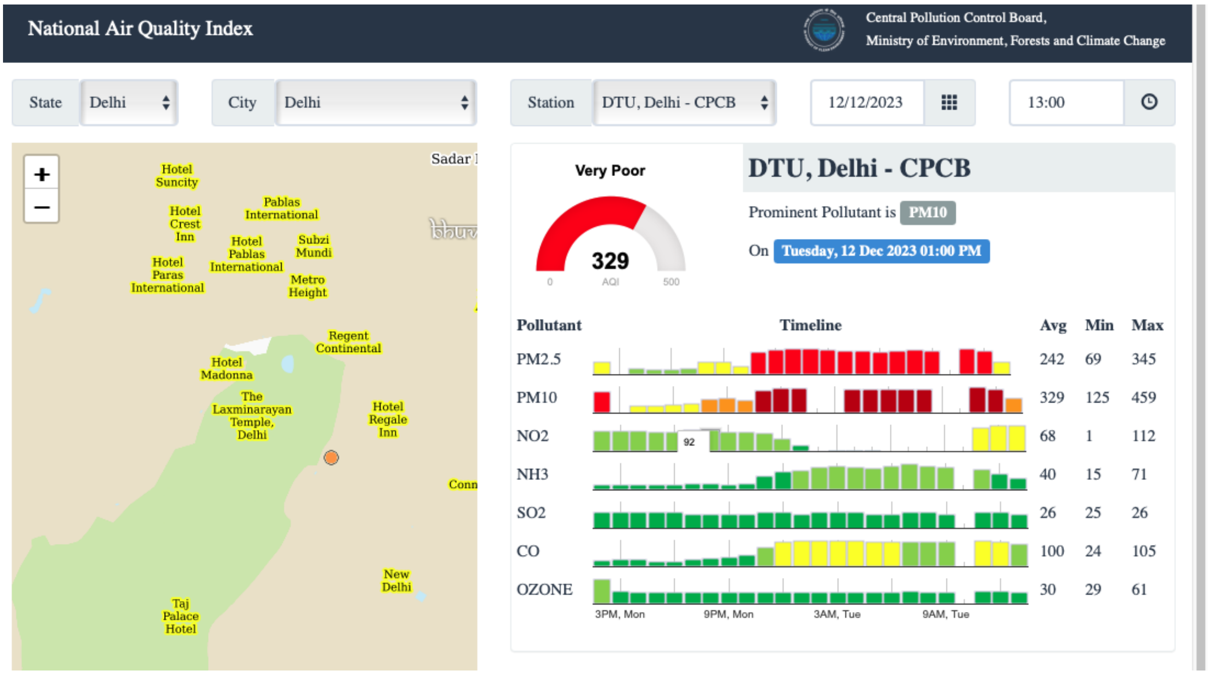
Database & Analytics Programming Documentation of Project Work Done

Thapelo Khantsi
School of Computing
National College of Ireland
Dublin, Ireland
x23131535@student.ncirl.ie

The following document entails details of the steps done in the group project.

1. Data Selection:

The original data from Central Pollution Control (CPCB) which is an organisation for air and water quality monitoring in India. Since the objective was to work with semi-structured data, Kaggle was used to get data that was either XML or JSON. And the data source had both, therefore the JSON file was selected instead. The data acquired was India City Air Quality that had different elements present in air. The goal of the project was to programmatically process and analyse the data to predict or determine pollutants that are most significant to poor air quality in India Cities.



National Air Quality Index

Central Pollution Control Board,
Ministry of Environment, Forests and Climate Change

State: Delhi
City: Delhi
Station: DTU, Delhi - CPCB
12/12/2023
13:00

Very Poor
329
AQI

DTU, Delhi - CPCB
Prominent Pollutant is PM10
On Tuesday, 12 Dec 2023 01:00 PM

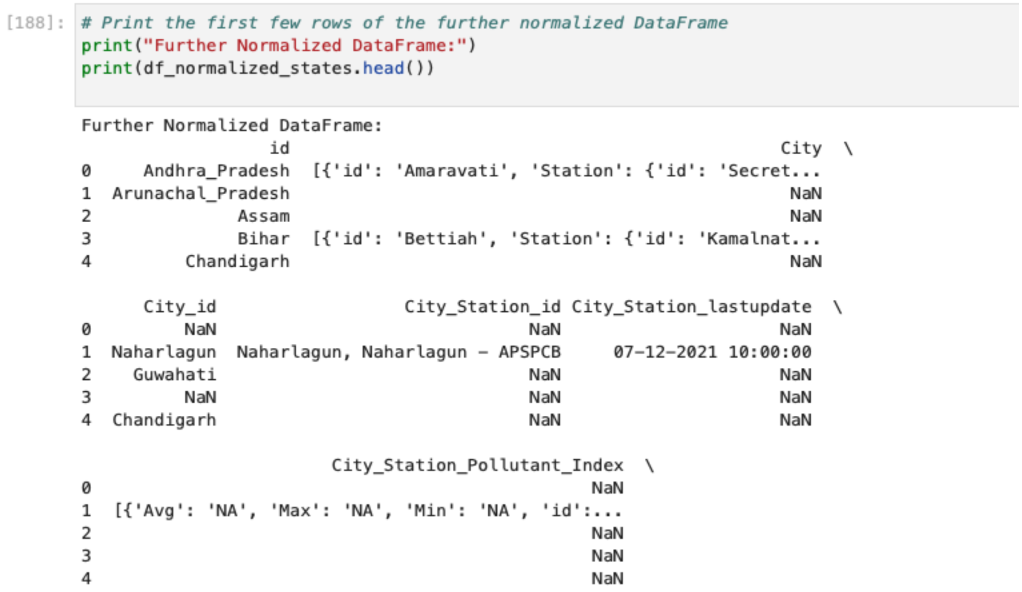
Pollutant	Timeline	Avg	Min	Max
PM2.5		242	69	345
PM10		329	125	459
NO2		68	1	112
NH3		40	15	71
SO2		26	25	26
CO		100	24	105
OZONE		30	29	61

2. MongoDB and PostgreSQL Set-up

These two databases were set up using Docker, MongodDB was mainly to take the JSON file dataset and put it into MongodDB, then perform ETL processes in order to analyse the data. PostgreSQL was to store structured dataset from MongoDB.

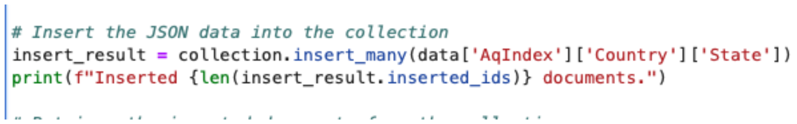
3. Connection to MongoDB

Python was used to achieve the connection to MongoDB. With this connection; a database and collection were created. Then the JSON file was loaded and inserted into the collection. The challenge here was that first the data was inserted not considering the structure of data as a python dictionary. This later created problems in data processing and transformation as the data normalization efforts did not provide accurate structure for a data frame that could be used to do analysis. As we can see from the screenshot below. The first step had created just a single document which required rigorous steps in normalization of data.



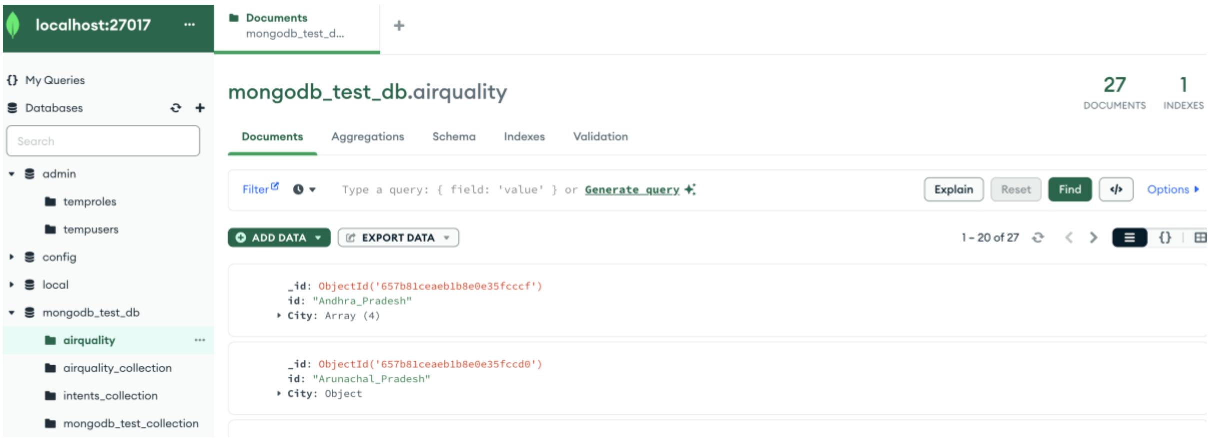
```
[188]: # Print the first few rows of the further normalized DataFrame
       print("Further Normalized DataFrame:")
       print(df_normalized_states.head())

       Further Normalized DataFrame:
                         id                                              City  \
       0     Andhra_Pradesh  [{'id': 'Amaravati', 'Station': {'id': 'Secret...
       1  Arunachal_Pradesh                                               NaN
       2              Assam                                               NaN
       3              Bihar  [{'id': 'Bettiah', 'Station': {'id': 'Kamalnat...
       4         Chandigarh                                               NaN

              City_id                City_Station_id City_Station_lastupdate  \
       0          NaN                            NaN                     NaN
       1  Naharlagun  Naharlagun, Naharlagun - APSPCB     07-12-2021 10:00:00
       2    Guwahati                            NaN                     NaN
       3          NaN                            NaN                     NaN
       4  Chandigarh                            NaN                     NaN

                               City_Station_Pollutant_Index  \
       0                                                NaN
       1  [{'Avg': 'NA', 'Max': 'NA', 'Min': 'NA', 'id':...
       2                                                NaN
       3                                                NaN
       4                                                NaN
```

To manage this, JSON file was inserted into mongoDB with an idea of data being a python dictionary and indexing it to get a specific structure within the data.



```
# Insert the JSON data into the collection
insert_result = collection.insert_many(data['AqIndex']['Country']['State'])
print(f"Inserted {len(insert_result.inserted_ids)} documents.")
```

This solution resulted to 27 documents being created which made is more manageable to do further transformation and create a data frame that can be used to analyse the data.



```
_id: ObjectId('657b81ceaeb1b8e0e35fcccf')
id: "Andhra_Pradesh"
▸ City: Array (4)

_id: ObjectId('657b81ceaeb1b8e0e35fccd0')
id: "Arunachal_Pradesh"
▸ City: Object
```

4. Data Retrieval From MondoDB and Data Cleaning

Python was also used to retrieve data from mongoDB in order to create a data frame that will be used for analysis. The first data frame had a shape of (27,4) . And this data frame was not suitable enough to work with. The data was further transformed into a second data frame that had a shape of 829 rows and 8 columns, which was much better to use for analysis.

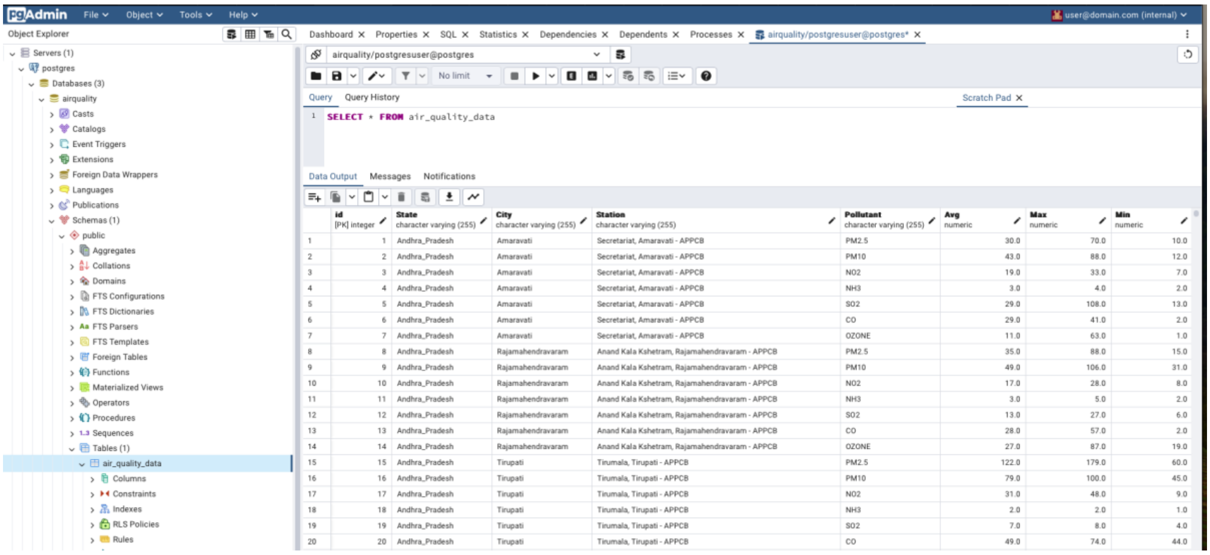
5. Data Visualisation

Some data visualisation were done to understand the data further and get insides.

6. Storing Data frame created in MongoDB into PostgreSQL database

In this process:

- Connection to SQL server was performed
- Database for air quality was created as well which would be used to process the data.
- Provided PostgreSQL details in order to create a table in PostgreSQL that will store the data based on the MongoDB data frame.
- Lastly the data from PostgreSQL was read into a data frame to use for further visualizations.



airquality/postgresuser@postgres

```
SELECT * FROM air_quality_data
```

	id [PK] integer	State character varying (255)	City character varying (255)	Station character varying (255)	Pollutant character varying (255)	Avg numeric	Max numeric	Min numeric
1	1	Andhra_Pradesh	Amaravati	Secretariat, Amaravati - APPCB	PM2.5	30.0	70.0	10.0
2	2	Andhra_Pradesh	Amaravati	Secretariat, Amaravati - APPCB	PM10	43.0	88.0	12.0
3	3	Andhra_Pradesh	Amaravati	Secretariat, Amaravati - APPCB	NO2	19.0	33.0	7.0
4	4	Andhra_Pradesh	Amaravati	Secretariat, Amaravati - APPCB	NH3	3.0	4.0	2.0
5	5	Andhra_Pradesh	Amaravati	Secretariat, Amaravati - APPCB	SO2	29.0	108.0	13.0
6	6	Andhra_Pradesh	Amaravati	Secretariat, Amaravati - APPCB	CO	29.0	41.0	2.0
7	7	Andhra_Pradesh	Amaravati	Secretariat, Amaravati - APPCB	OZONE	11.0	63.0	1.0
8	8	Andhra_Pradesh	Rajamahendravaram	Anand Kala Kshetram, Rajamahendravaram - APPCB	PM2.5	35.0	88.0	15.0
9	9	Andhra_Pradesh	Rajamahendravaram	Anand Kala Kshetram, Rajamahendravaram - APPCB	PM10	49.0	106.0	31.0
10	10	Andhra_Pradesh	Rajamahendravaram	Anand Kala Kshetram, Rajamahendravaram - APPCB	NO2	17.0	28.0	8.0
11	11	Andhra_Pradesh	Rajamahendravaram	Anand Kala Kshetram, Rajamahendravaram - APPCB	NH3	3.0	5.0	2.0
12	12	Andhra_Pradesh	Rajamahendravaram	Anand Kala Kshetram, Rajamahendravaram - APPCB	SO2	13.0	27.0	6.0
13	13	Andhra_Pradesh	Rajamahendravaram	Anand Kala Kshetram, Rajamahendravaram - APPCB	CO	28.0	57.0	2.0
14	14	Andhra_Pradesh	Rajamahendravaram	Anand Kala Kshetram, Rajamahendravaram - APPCB	OZONE	27.0	87.0	19.0
15	15	Andhra_Pradesh	Tirupati	Tirumala, Tirupati - APPCB	PM2.5	122.0	179.0	60.0
16	16	Andhra_Pradesh	Tirupati	Tirumala, Tirupati - APPCB	PM10	79.0	100.0	45.0
17	17	Andhra_Pradesh	Tirupati	Tirumala, Tirupati - APPCB	NO2	31.0	48.0	9.0
18	18	Andhra_Pradesh	Tirupati	Tirumala, Tirupati - APPCB	NH3	2.0	2.0	1.0
19	19	Andhra_Pradesh	Tirupati	Tirumala, Tirupati - APPCB	SO2	7.0	8.0	4.0
20	20	Andhra_Pradesh	Tirupati	Tirumala, Tirupati - APPCB	CO	49.0	74.0	44.0

7. Further Predictive Analysis Using Machine Learning Models.

Linear Regression and Random Forest Regression models were used to support the analysis done from data visualisations about the factors that influence poor air quality in India.

8. GitHub Code Storage

Code was lastly committed to GitHub.

9. Reporting

We collectively written a report for the work done. Dividing each section into Dataset 1, Dataset 2, and Dataset 3; so that every member can document work done for their dataset.

10. Video Presentation

This was the last requirement in the project and we used teams to create a video presentation.