Project Selection

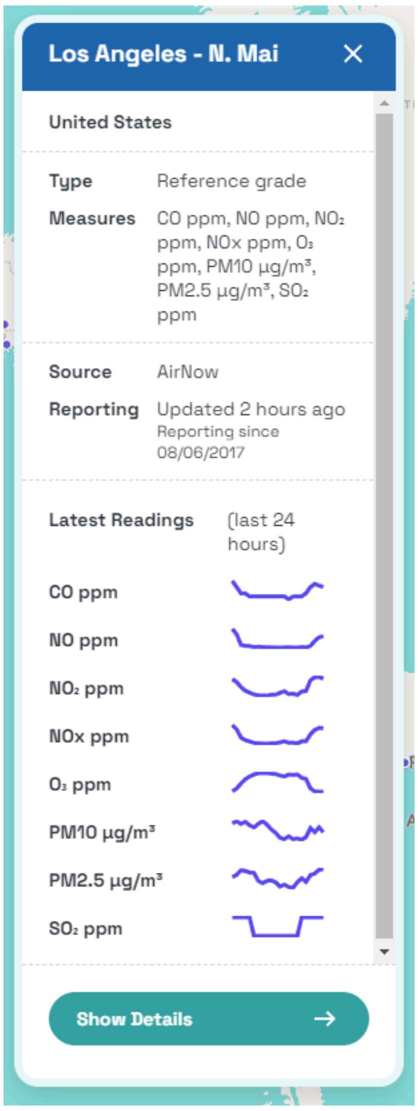
Objective

- Identify which sensor can be eliminated to optimally reduce cost.

Hypothesis

Eliminating the `no` sensor (Nitric Oxide) in N. Mai, Los Angeles California (CA), will have a minimal impact on overall air quality monitoring. This is based on the strong correlation, interdependence, or redundancy of `no` with other related pollutants, such as `no2` and `nox`. By leveraging data from these sensors, it can effectively infer `no` levels, thereby optimally reducing project expenses while maintaining the integrity of air quality data.

Reference on Chosen Location ID: https://explore.openaq.org/locations/7936



Los Angeles - N. Mai

United States

Type Reference grade

Measures CO ppm, NO ppm, NO_2 ppm, NOx ppm, O_3 ppm, PM10 µg/m^3, PM2.5 µg/m^3, SO_2 ppm

Source AirNow

Reporting Updated 2 hours ago
Reporting since 08/06/2017

Latest Readings (last 24 hours)

CO ppm

NO ppm

NO_2 ppm

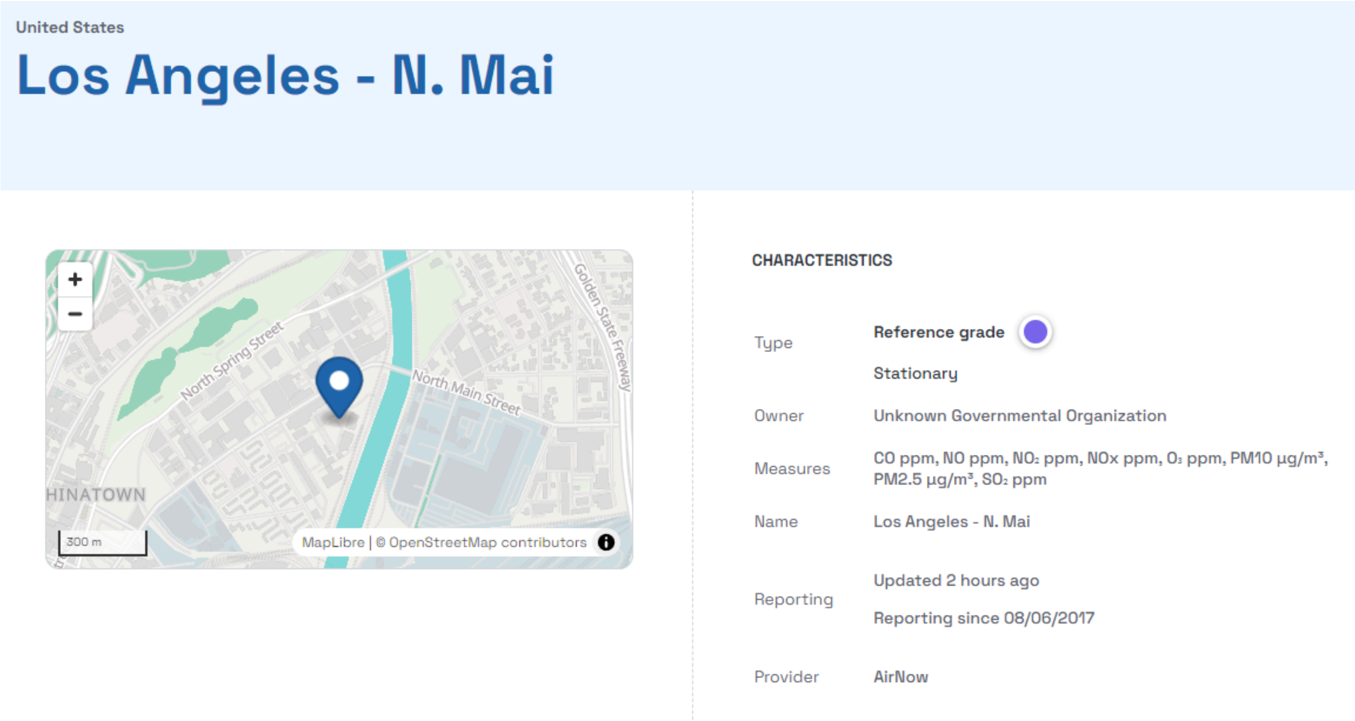
NOx ppm

O_3 ppm

PM10 µg/m^3

PM2.5 µg/m^3

SO_2 ppm



United States

Los Angeles - N. Mai

CHARACTERISTICS

Type Reference grade
Stationary

Owner Unknown Governmental Organization

Measures CO ppm, NO ppm, NO_2 ppm, NOx ppm, O_3 ppm, PM10 µg/m^3, PM2.5 µg/m^3, SO_2 ppm

Name Los Angeles - N. Mai

Reporting Updated 2 hours ago
Reporting since 08/06/2017

Provider AirNow

Latest Readings

NO ppm | Last 24 hours | Linear | Update

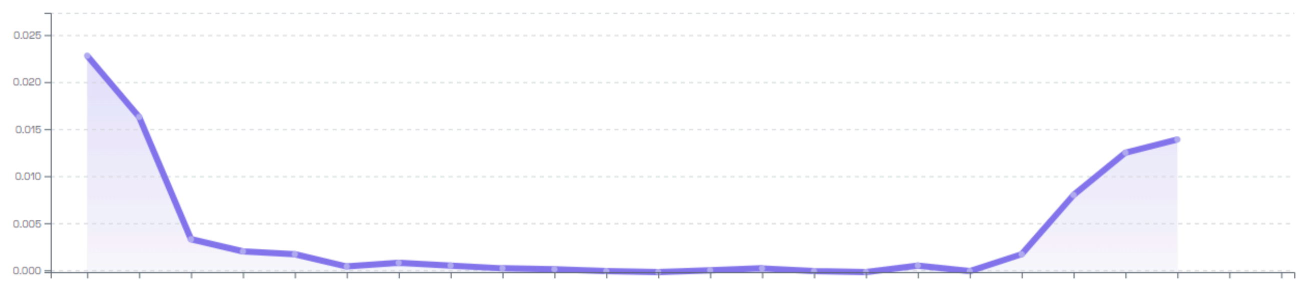


Chart shows local times (America/Los_Angeles)

Patterns

NO ppm | 2024 | Update

Hour of day

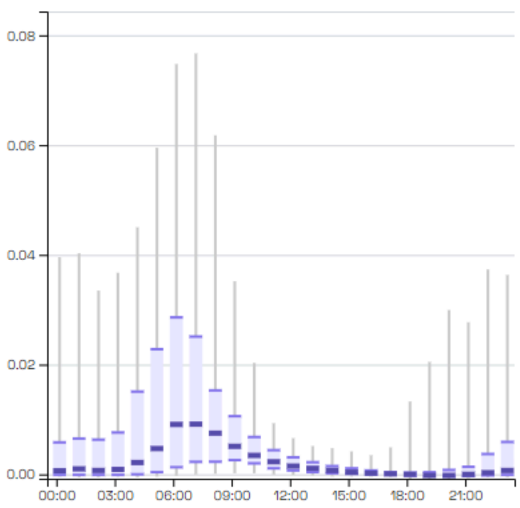


Chart shows local times (America/Los_Angeles)

Procedure

```
# Installed the Spark libraries
!pip install pyspark
!pip install findspark
```

```
Requirement already satisfied: pyspark in /usr/local/lib/python3.10/dist-packages (3.5.3)
Requirement already satisfied: py4j==0.10.9.7 in /usr/local/lib/python3.10/dist-packages (from pyspark) (0.10.9.7)
Collecting findspark
  Downloading findspark-2.0.1-py2.py3-none-any.whl.metadata (352 bytes)
Downloading findspark-2.0.1-py2.py3-none-any.whl (4.4 kB)
Installing collected packages: findspark
Successfully installed findspark-2.0.1
```

```
# Installed the awscli or the AWS S3 for pulling the data from chosen location ID
!pip install awscli
```

```
Collecting awscli
  Downloading awscli-1.36.4-py3-none-any.whl.metadata (11 kB)
Collecting botocore==1.35.63 (from awscli)
  Downloading botocore-1.35.63-py3-none-any.whl.metadata (5.7 kB)
Collecting docutils<0.17,>=0.10 (from awscli)
  Downloading docutils-0.16-py2.py3-none-any.whl.metadata (2.7 kB)
Collecting s3transfer<0.11.0,>=0.10.0 (from awscli)
  Downloading s3transfer-0.10.3-py3-none-any.whl.metadata (1.7 kB)
Requirement already satisfied: PyYAML<6.1,>=3.10 in /usr/local/lib/python3.10/dist-packages (from awscli) (6.0.2)
Collecting colorama<0.4.7,>=0.2.5 (from awscli)
  Downloading colorama-0.4.6-py2.py3-none-any.whl.metadata (17 kB)
Collecting rsa<4.8,>=3.1.2 (from awscli)
  Downloading rsa-4.7.2-py3-none-any.whl.metadata (3.6 kB)
Collecting jmespath<2.0.0,>=0.7.1 (from botocore==1.35.63->awscli)
  Downloading jmespath-1.0.1-py3-none-any.whl.metadata (7.6 kB)
Requirement already satisfied: python-dateutil<3.0.0,>=2.1 in /usr/local/lib/python3.10/dist-packages (from botocore==1.35.63->awscli) (
Requirement already satisfied: urllib3!=2.2.0,<3,>=1.25.4 in /usr/local/lib/python3.10/dist-packages (from botocore==1.35.63->awscli) (2
Requirement already satisfied: pyasn1>=0.1.3 in /usr/local/lib/python3.10/dist-packages (from rsa<4.8,>=3.1.2->awscli) (0.6.1)
Requirement already satisfied: six>=1.5 in /usr/local/lib/python3.10/dist-packages (from python-dateutil<3.0.0,>=2.1->botocore==1.35.63-
Downloading awscli-1.36.4-py3-none-any.whl (4.5 MB)
   ──────────────────────────────────────── 4.5/4.5 MB 36.6 MB/s eta 0:00:00
Downloading botocore-1.35.63-py3-none-any.whl (12.8 MB)
   ──────────────────────────────────────── 12.8/12.8 MB 62.9 MB/s eta 0:00:00
Downloading colorama-0.4.6-py2.py3-none-any.whl (25 kB)
Downloading docutils-0.16-py2.py3-none-any.whl (548 kB)
```

```
    ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 548.2/548.2 kB 29.6 MB/s eta 0:00:00
Downloading rsa-4.7.2-py3-none-any.whl (34 kB)
Downloading s3transfer-0.10.3-py3-none-any.whl (82 kB)
   ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 82.6/82.6 kB 5.3 MB/s eta 0:00:00
Downloading jmespath-1.0.1-py3-none-any.whl (20 kB)
Installing collected packages: rsa, jmespath, docutils, colorama, botocore, s3transfer, awscli
  Attempting uninstall: rsa
    Found existing installation: rsa 4.9
    Uninstalling rsa-4.9:
      Successfully uninstalled rsa-4.9
  Attempting uninstall: docutils
    Found existing installation: docutils 0.21.2
    Uninstalling docutils-0.21.2:
      Successfully uninstalled docutils-0.21.2
ERROR: pip's dependency resolver does not currently take into account all the packages that are installed. This behaviour is the source
sphinx 8.1.3 requires docutils<0.22,>=0.20, but you have docutils 0.16 which is incompatible.
Successfully installed awscli-1.36.4 botocore-1.35.63 colorama-0.4.6 docutils-0.16 jmespath-1.0.1 rsa-4.7.2 s3transfer-0.10.3
```

```
# Imported the matplotlib
import matplotlib.pyplot as plt
import numpy as np
```

```
# Imported the findspark
import findspark
findspark.init()
from pyspark.sql import SparkSession

spark = SparkSession.builder \
        .master('local[*]') \
        .appName('ProjectSelection') \
        .getOrCreate()

print(spark.version)
```

```
3.5.3
```

```
# Created a directory for the data
!mkdir raw_7936
!ls
```

```
raw_7936  sample_data
```

```
# Extracted data from AWS S3 openaq-data-archive similar with previous coding exercises.
!aws s3 cp --recursive --no-sign-request s3://openaq-data-archive/records/csv.gz/locationid=7936/ raw_7936
```

```
download: s3://openaq-data-archive/records/csv.gz/locationid=7936/year=2017/month=06/location-7936-20170610.csv.gz to raw_7936/year=2
download: s3://openaq-data-archive/records/csv.gz/locationid=7936/year=2017/month=06/location-7936-20170608.csv.gz to raw_7936/year=2
download: s3://openaq-data-archive/records/csv.gz/locationid=7936/year=2017/month=06/location-7936-20170616.csv.gz to raw_7936/year=2
download: s3://openaq-data-archive/records/csv.gz/locationid=7936/year=2017/month=06/location-7936-20170622.csv.gz to raw_7936/year=2
download: s3://openaq-data-archive/records/csv.gz/locationid=7936/year=2017/month=06/location-7936-20170611.csv.gz to raw_7936/year=2
download: s3://openaq-data-archive/records/csv.gz/locationid=7936/year=2017/month=06/location-7936-20170621.csv.gz to raw_7936/year=2
download: s3://openaq-data-archive/records/csv.gz/locationid=7936/year=2017/month=06/location-7936-20170614.csv.gz to raw_7936/year=2
download: s3://openaq-data-archive/records/csv.gz/locationid=7936/year=2017/month=06/location-7936-20170612.csv.gz to raw_7936/year=2
download: s3://openaq-data-archive/records/csv.gz/locationid=7936/year=2017/month=06/location-7936-20170613.csv.gz to raw_7936/year=2
download: s3://openaq-data-archive/records/csv.gz/locationid=7936/year=2017/month=06/location-7936-20170619.csv.gz to raw_7936/year=2
download: s3://openaq-data-archive/records/csv.gz/locationid=7936/year=2017/month=06/location-7936-20170609.csv.gz to raw_7936/year=2
download: s3://openaq-data-archive/records/csv.gz/locationid=7936/year=2017/month=06/location-7936-20170627.csv.gz to raw_7936/year=2
download: s3://openaq-data-archive/records/csv.gz/locationid=7936/year=2017/month=06/location-7936-20170624.csv.gz to raw_7936/year=2
download: s3://openaq-data-archive/records/csv.gz/locationid=7936/year=2017/month=06/location-7936-20170620.csv.gz to raw_7936/year=2
download: s3://openaq-data-archive/records/csv.gz/locationid=7936/year=2017/month=06/location-7936-20170629.csv.gz to raw_7936/year=2
download: s3://openaq-data-archive/records/csv.gz/locationid=7936/year=2017/month=06/location-7936-20170607.csv.gz to raw_7936/year=2
download: s3://openaq-data-archive/records/csv.gz/locationid=7936/year=2017/month=06/location-7936-20170617.csv.gz to raw_7936/year=2
download: s3://openaq-data-archive/records/csv.gz/locationid=7936/year=2017/month=06/location-7936-20170626.csv.gz to raw_7936/year=2
download: s3://openaq-data-archive/records/csv.gz/locationid=7936/year=2017/month=07/location-7936-20170701.csv.gz to raw_7936/year=2
download: s3://openaq-data-archive/records/csv.gz/locationid=7936/year=2017/month=06/location-7936-20170630.csv.gz to raw_7936/year=2
download: s3://openaq-data-archive/records/csv.gz/locationid=7936/year=2017/month=06/location-7936-20170623.csv.gz to raw_7936/year=2
download: s3://openaq-data-archive/records/csv.gz/locationid=7936/year=2017/month=07/location-7936-20170703.csv.gz to raw_7936/year=2
download: s3://openaq-data-archive/records/csv.gz/locationid=7936/year=2017/month=07/location-7936-20170702.csv.gz to raw_7936/year=2
download: s3://openaq-data-archive/records/csv.gz/locationid=7936/year=2017/month=07/location-7936-20170708.csv.gz to raw_7936/year=2
download: s3://openaq-data-archive/records/csv.gz/locationid=7936/year=2017/month=07/location-7936-20170712.csv.gz to raw_7936/year=2
download: s3://openaq-data-archive/records/csv.gz/locationid=7936/year=2017/month=07/location-7936-20170706.csv.gz to raw_7936/year=2
download: s3://openaq-data-archive/records/csv.gz/locationid=7936/year=2017/month=07/location-7936-20170711.csv.gz to raw_7936/year=2
download: s3://openaq-data-archive/records/csv.gz/locationid=7936/year=2017/month=06/location-7936-20170618.csv.gz to raw_7936/year=2
download: s3://openaq-data-archive/records/csv.gz/locationid=7936/year=2017/month=06/location-7936-20170615.csv.gz to raw_7936/year=2
download: s3://openaq-data-archive/records/csv.gz/locationid=7936/year=2017/month=07/location-7936-20170714.csv.gz to raw_7936/year=2
download: s3://openaq-data-archive/records/csv.gz/locationid=7936/year=2017/month=07/location-7936-20170713.csv.gz to raw_7936/year=2
download: s3://openaq-data-archive/records/csv.gz/locationid=7936/year=2017/month=07/location-7936-20170705.csv.gz to raw_7936/year=2
download: s3://openaq-data-archive/records/csv.gz/locationid=7936/year=2017/month=07/location-7936-20170709.csv.gz to raw_7936/year=2
download: s3://openaq-data-archive/records/csv.gz/locationid=7936/year=2017/month=08/location-7936-20170811.csv.gz to raw_7936/year=2
download: s3://openaq-data-archive/records/csv.gz/locationid=7936/year=2017/month=07/location-7936-20170717.csv.gz to raw_7936/year=2
download: s3://openaq-data-archive/records/csv.gz/locationid=7936/year=2017/month=07/location-7936-20170715.csv.gz to raw_7936/year=2
download: s3://openaq-data-archive/records/csv.gz/locationid=7936/year=2017/month=07/location-7936-20170719.csv.gz to raw_7936/year=2
download: s3://openaq-data-archive/records/csv.gz/locationid=7936/year=2017/month=07/location-7936-20170718.csv.gz to raw_7936/year=2
download: s3://openaq-data-archive/records/csv.gz/locationid=7936/year=2017/month=07/location-7936-20170710.csv.gz to raw_7936/year=2
download: s3://openaq-data-archive/records/csv.gz/locationid=7936/year=2017/month=08/location-7936-20170812.csv.gz to raw_7936/year=2
download: s3://openaq-data-archive/records/csv.gz/locationid=7936/year=2017/month=08/location-7936-20170813.csv.gz to raw_7936/year=2
download: s3://openaq-data-archive/records/csv.gz/locationid=7936/year=2017/month=07/location-7936-20170720.csv.gz to raw_7936/year=2
download: s3://openaq-data-archive/records/csv.gz/locationid=7936/year=2017/month=06/location-7936-20170628.csv.gz to raw_7936/year=2
download: s3://openaq-data-archive/records/csv.gz/locationid=7936/year=2017/month=08/location-7936-20170816.csv.gz to raw_7936/year=2
download: s3://openaq-data-archive/records/csv.gz/locationid=7936/year=2017/month=08/location-7936-20170814.csv.gz to raw_7936/year=2
download: s3://openaq-data-archive/records/csv.gz/locationid=7936/year=2017/month=08/location-7936-20170817.csv.gz to raw_7936/year=2
download: s3://openaq-data-archive/records/csv.gz/locationid=7936/year=2017/month=07/location-7936-20170716.csv.gz to raw_7936/year=2
download: s3://openaq-data-archive/records/csv.gz/locationid=7936/year=2017/month=08/location-7936-20170819.csv.gz to raw_7936/year=2
download: s3://openaq-data-archive/records/csv.gz/locationid=7936/year=2017/month=07/location-7936-20170704.csv.gz to raw_7936/year=2
download: s3://openaq-data-archive/records/csv.gz/locationid=7936/year=2017/month=08/location-7936-20170822.csv.gz to raw_7936/year=2
download: s3://openaq-data-archive/records/csv.gz/locationid=7936/year=2017/month=08/location-7936-20170820.csv.gz to raw_7936/year=2
download: s3://openaq-data-archive/records/csv.gz/locationid=7936/year=2017/month=08/location-7936-20170825.csv.gz to raw_7936/year=2
download: s3://openaq-data-archive/records/csv.gz/locationid=7936/year=2017/month=08/location-7936-20170827.csv.gz to raw_7936/year=2
download: s3://openaq-data-archive/records/csv.gz/locationid=7936/year=2017/month=08/location-7936-20170823.csv.gz to raw_7936/year=2
download: s3://openaq-data-archive/records/csv.gz/locationid=7936/year=2017/month=08/location-7936-20170815.csv.gz to raw_7936/year=2
download: s3://openaq-data-archive/records/csv.gz/locationid=7936/year=2017/month=08/location-7936-20170829.csv.gz to raw_7936/year=2
download: s3://openaq-data-archive/records/csv.gz/locationid=7936/year=2017/month=06/location-7936-20170625.csv.gz to raw_7936/year=2
```

```
# Defined the 7938 and displayed top 5 rows from the dataset
df_7936 = spark.read.csv('/content/raw_7936/*/*/', inferSchema=True, header=True)
df_7936.show(5)
```

```
+-----------+----------+--------------------+--------------------+---------+-------------------+---------+-----+-----+
|location_id|sensors_id|            location|            datetime|      lat|                lon|parameter|units|value|
+-----------+----------+--------------------+--------------------+---------+-------------------+---------+-----+-----+
|       7936|     25195|Los Angeles - N. ...|2024-10-25T01:00:...|34.066429|-118.22675500000001|     pm10|µg/m³| 30.0|
|       7936|     25195|Los Angeles - N. ...|2024-10-25T02:00:...|34.066429|-118.22675500000001|     pm10|µg/m³| 31.0|
|       7936|     25195|Los Angeles - N. ...|2024-10-25T03:00:...|34.066429|-118.22675500000001|     pm10|µg/m³| 22.0|
|       7936|     25195|Los Angeles - N. ...|2024-10-25T04:00:...|34.066429|-118.22675500000001|     pm10|µg/m³| 28.0|
|       7936|     25195|Los Angeles - N. ...|2024-10-25T05:00:...|34.066429|-118.22675500000001|     pm10|µg/m³| 27.0|
+-----------+----------+--------------------+--------------------+---------+-------------------+---------+-----+-----+
only showing top 5 rows
```

```
# Displayed the number of rows from the dataset
df_7936.count()
```

```
298933
```

Observation and Analysis

Based on my observations, the dataset contains nearly 300,000 rows `(298,933)` of air quality data collected from the location I chose which is from "**Los Angeles - N. Mai**". Each record represents a measurement from a specific sensor, taken **hourly**. The data includes various air pollutants such as `pm10`, `pm2.5`, `so2`, `co`, `nox`, `no2`, and `o3`. Furthermore, these pollutants are critical for monitoring air quality, and the goal is to determine if the Nitric Oxide (`no`) sensor can be removed or eliminated without significantly affecting the quality of air monitoring. Since `no` is closely related to other pollutants like `no2` and `nox`, there's a possibility that its levels can be accurately estimated using data from those sensors. This could help reduce project costs while maintaining reliable air quality data. The next steps involve analyzing the relationships between these pollutants and testing if I can predict `no` levels accurately based on other sensors in the methdology part.

Conclusion

In summary, the dataset provides **hourly** air quality measurements, which is the optimal duration for this analysis since it ensures a high-resolution view of pollutant levels and their relationships over time. This granularity will help me accurately assess the correlation between Nitric Oxide (`no`) and other pollutants such as `no2` and `nox` since the challenge is doing the methdology as granular as possible.

I also think the hypothesis—that the `no` sensor can be eliminated while maintaining data quality—can be addressed using supervised learning techniques on this large dataset. By leveraging predictive models, I aim to estimate `no` levels based on data from other sensors, validating whether `no` measurements can be reliably inferred.

The main challenge in answering the hypothesis lies in processing and analyzing the large volume of data effectively. Ensuring the data is clean and correctly formatted for machine learning models is important. Additionally, selecting the best model to achieve high prediction accuracy could have a technical challenge for me.

The current difficulty in studying this dataset is managing its size and complexity, particularly ensuring that computational resources and data transformations align well with the requirements for effective machine learning analysis. Further, interpreting the results in a way that aligns with the project's cost-reduction goal is another challenge to be tackled.