**SAT5165 – Big Data Analytics**

**Large Project Report**

Project Title: **Large-Scale Analysis of Air Quality and Respiratory Health Risks Using PySpark**

**Team Members:** Mary Nnipaa Meteku, Dennis Owusu, Uttam Kumar Bellamkonda, Sucharitha Reddy Dammareddygari, Fredrick Damptey

GitHub Repository:

<https://github.com/ukbellam/Large-Scale-Analysis-of-Air-Quality-and-Respiratory-Health-Risk>

**1. Introduction**

This project explores large-scale air quality data using Apache Spark for distributed processing. It examines pollutant trends and their impact on air quality, employing Spark MLlib for statistical and predictive analysis. The dataset used was the U.S. Pollution Data (2000–2016), with over 2.5 million records of PM2.5, Ozone, SO₂, and NO₂ levels. The goal was to clean, process, and analyze the data efficiently while testing Spark’s performance on multiple virtual machines.

**2. Environment Configuration**

Each member configured a Spark cluster on two virtual machines. The IP configurations are listed below:

|  |  |  |
| --- | --- | --- |
| **Team** | **VM1 (Master)** | **VM2 (Worker)** |
| Mary Nnipaa Meteku | 192.168.13.146 | 192.168.13.147 |
| Dennis Owusu | 192.168.13.179 | 192.168.13.180 |
| Uttam Kumar Bellamkonda | 192.168.13.107 | 192.168.13.108 |
| Sucharitha Reddy Dammareddygari | 192.168.13.113 | 192.168.13.114 |
| Fredrick Damptey | 192.168.13.116 | 192.168.13.117 |

**3. Explanation of Codes and Method**

The PySpark project begins by loading the U.S. Pollution dataset with read.csv(), using inferred headers and schema to create a Spark DataFrame. Column names are cleaned to remove spaces and special characters for easier referencing. Missing numeric values are replaced with column means using pyspark.sql.functions.mean(), while missing text entries (State, County, City) are filled with “Unknown.” A loop checks for remaining nulls, and printSchema() confirms correct data types. The cleaned data is saved with write.csv() in overwrite mode. Descriptive statistics are produced with describe(), and groupBy() with avg() computes average pollutant levels by state. Correlation analysis is performed using MLlib’s VectorAssembler and Correlation.corr() to calculate Pearson correlations between pollutants and the Air Quality Index (AQI). Finally, Spark cluster commands (start-master.sh, start-worker.sh) manage master–worker connections, and the Spark Web UI monitors cluster activity and job progress.

**4.1 Mary Nnipaa Meteku – Data Cleaning and Missing Value Handling**

I focused on cleaning and preparing the dataset for analysis. I used PySpark DataFrames to read, rename columns, impute missing values, check schema consistency, and export the cleaned data. To start, I set up and verified the Spark master and worker nodes across his virtual machines (VMs: 192.168.13.146 and 192.168.13.147), ensuring that distributed computation was operational for scalable data analysis.

A screenshot of a computer

AI-generated content may be incorrect.

Figure 1.1: Mary’s Spark Master Node (192.168.13.146) started successfully.

A screenshot of a computer

AI-generated content may be incorrect.

Figure 1.2: Mary’s Worker Node (192.168.13.147) connected to Spark Master.

A screenshot of a computer

AI-generated content may be incorrect.

Figure 1.3: Mary’s Spark Web UI showing connected worker node.

A black and white screen with white text

AI-generated content may be incorrect.

Figure 1.4: Initial data loading and preview of the dataset.



Figure 1.5: Column renaming and missing value handling.

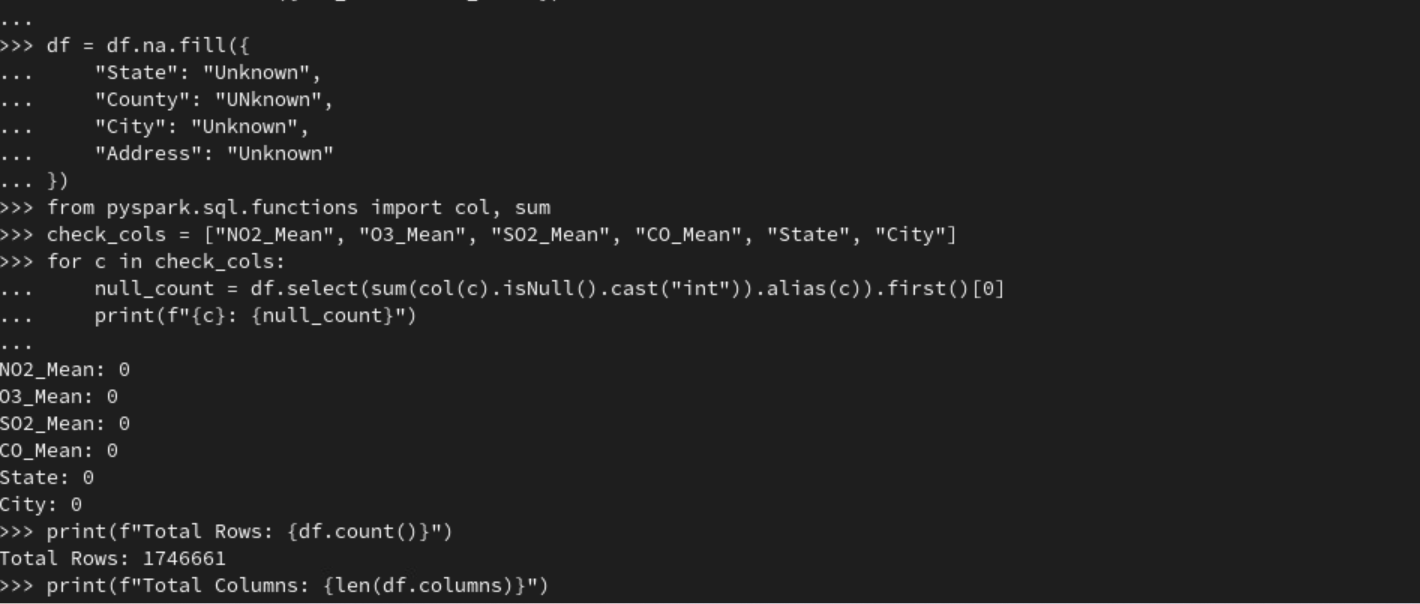


Figure 1.6: Verification of null value counts after imputation.

A computer screen shot of a code

AI-generated content may be incorrect.

Figure 1.7: Schema inspection after cleaning.

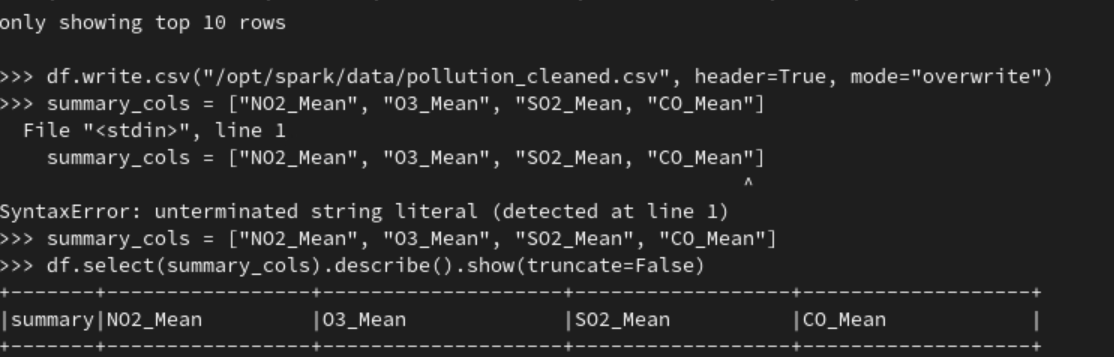


Figure 1.8: Cleaned dataset successfully saved in Spark directory.

**3.2 Dennis Owusu – Correlation Analysis**

My section involved performing correlation analysis using Spark MLlib. I calculated Pearson correlations between NO₂, SO₂, O₃, CO, and AQI, then displayed correlation matrices and schema outputs. To start, I set up and verified the Spark master and worker nodes across his virtual machines (VMs: 192.168.13.179 and 192.168.13.180), ensuring that distributed computation was operational for scalable data analysis.

A screen shot of a computer

AI-generated content may be incorrect.

Figure 2.1: Dennis’s Spark Master Node (192.168.13.179) started successfully.

A screen shot of a computer

AI-generated content may be incorrect.

Figure 2.2: Dennis’s Worker Node (192.168.13.180) connected to Master.

A screenshot of a computer

AI-generated content may be incorrect.

Figure 2.3: Spark Master Web UI (Dennis) confirming one active worker node.

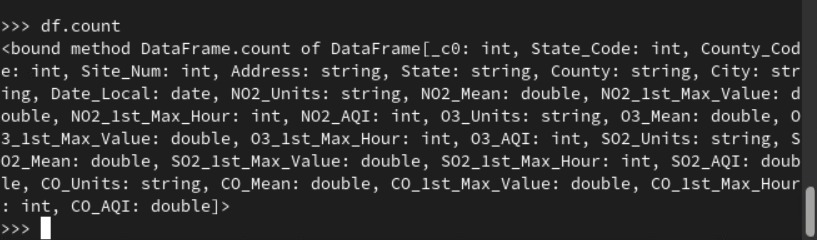


Figure 2.4: DataFrame count verification before correlation analysis.

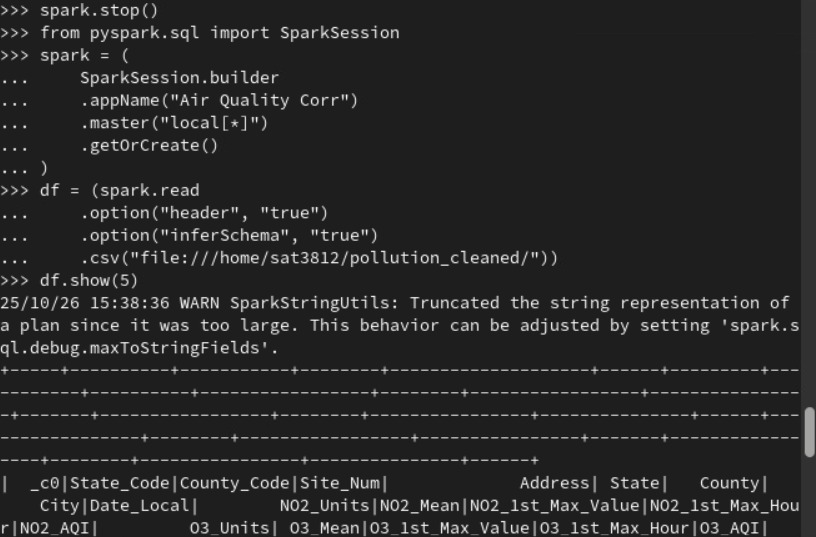


Figure 2.5: Cleaned dataset loaded for correlation analysis.

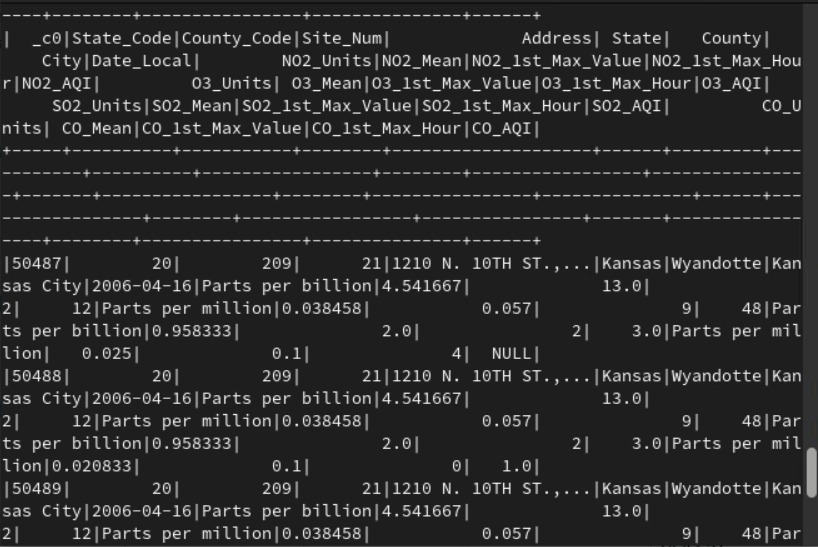


Figure 2.6: Sample data displayed before correlation computation.

A computer screen shot of a code

AI-generated content may be incorrect.

Figure 2.7: Schema validation confirming dataset readiness for correlation.

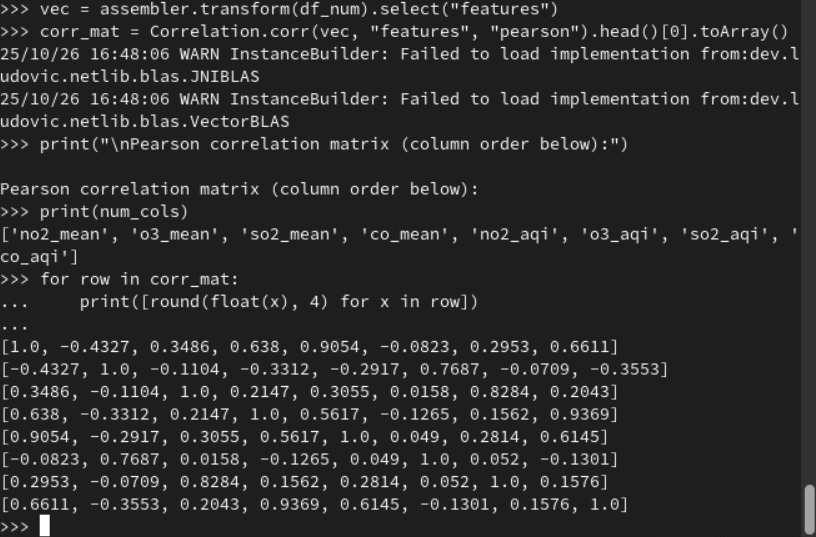


Figure 2.8: Pearson correlation matrix generated using Spark MLlib.

**3.3 Uttam Kumar Bellamkonda – Regression Analysis**

The goal of the regression analysis is to predict the Air Quality Index (AQI) based on pollutant concentration variables—NO₂, O₃, SO₂, CO, PM2.5, and PM10, using a linear regression model in PySpark. To start, I set up and verified the Spark master and worker nodes across his virtual machines (VMs: 192.168.13.107 and 192.168.13.108), ensuring that distributed computation was operational for scalable data analysis.

A screenshot of a computer

AI-generated content may be incorrect.

Figure 3.1: Uttam’s Spark Master Node (192.168.13.107) started successfully.

A computer screen with white text

AI-generated content may be incorrect.

Figure 3.2: Uttam’s Worker Node (192.168.13.108) connected to Spark Master.

A screenshot of a computer

AI-generated content may be incorrect.

Figure 3.3: Spark Master Web UI (Uttam) confirming one active worker node

The dataset is loaded using the spark.read.csv() function with headers and schema inference enabled. The code dynamically identifies pollutant columns (NO₂\_AQI, O₃\_AQI, SO₂\_AQI, CO\_AQI, PM2.5\_AQI, PM10\_AQI) and creates a new column 'AQI\_label' representing the maximum AQI among these pollutants. This label serves as the dependent variable for regression analysis.

A screenshot of a computer program

AI-generated content may be incorrect.

Figure 3.4: Feature assembly, scaling, and model training.

A screenshot of a computer screen

AI-generated content may be incorrect.

Figure 3.5: Predicted vs. actual AQI values (sample output).

**3.4 Fredrick Damptey – Chi-Square Test for Feature Selection**

I conducted a Chi-Square test of independence to evaluate the relationship between pollutant concentration levels and the frequency of unhealthy air days. Using PySpark’s ChiSquareTest module, the analysis tested whether categorical pollutant categories (low, moderate, high) had a statistically significant association with unhealthy air conditions. To start, I set up and verified the Spark master and worker nodes across his virtual machines (VMs: 192.168.13.116 and 192.168.13.117), ensuring that distributed computation was operational for scalable data analysis.

A screenshot of a computer

AI-generated content may be incorrect.

Figure 4.1: Fredrick’s Spark Master Node (192.168.13.116) started successfully

A screenshot of a computer

AI-generated content may be incorrect.

Figure 4.2: Fredrick’s Worker Node (192.168.13.117) connected to Spark Master.

A screenshot of a computer

AI-generated content may be incorrect.

Figure 4.3: Spark Master Web UI (Fredrick) confirming one active worker node

A computer screen with white text

AI-generated content may be incorrect.

Figure 4.4: Air Quality Category Classification and Chi-Square Calculation

A screenshot of a computer

AI-generated content may be incorrect.

Figure 4.5: Chi-Square Test Results

**3.5 Sucharitha reddy Dammareddygari – Dimensionality Reduction (PCA)**

I used **Principal Component Analysis (PCA)** to reduce the dimensionality of the air quality dataset while preserving most of the important information. The dataset included many related pollutant measures, such as PM2.5, PM10, NO₂, SO₂, CO, and O₃. To start, I set up and verified the Spark master and worker nodes across his virtual machines (VMs: 192.168.13.116 and 192.168.13.117), ensuring that distributed computation was operational for scalable data analysis. Before applying PCA, all the data were standardized so that every feature had equal importance. Then, using **Spark MLlib**, the PCA model found the main components that explain most of the data variation. The results showed that the **first few components** captured most of the total information, which means we could work with fewer features without losing accuracy. This made our analysis **faster, simpler, and easier to understand**. Reduced data can also be used to train **predictive models** to classify air quality days as “healthy” or “unhealthy.”

A screen shot of a computer

AI-generated content may be incorrect.

Figure 5.1: Sucharitha’s Spark Master Node (192.168.13.113) started successfully

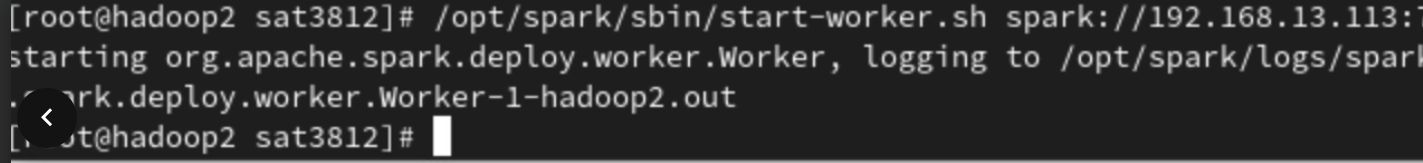


Figure 5.2: Sucharitha’s Worker Node (192.168.13.114) connected to Spark Master.

A screenshot of a computer

AI-generated content may be incorrect.

Figure 5.3: Spark Master Web UI (Sucharitha) confirming one active worker node

A screenshot of a computer

AI-generated content may be incorrect.

Figure 5.4: Dimensionality Reduction

**5. Conclusion**

This project showed how PySpark can be used to handle and analyze very large air quality datasets quickly and efficiently. Using Spark on multiple virtual machines helped the team process more than two million pollution records faster than traditional tools.

Each team member worked on a specific part that built the project step by step. Mary cleaned the data to make sure it was accurate and complete. Dennis found how pollutants like NO₂, SO₂, CO, and O₃ are related to air quality. Uttam built regression models to predict the Air Quality Index from pollutant levels. Fredrick used the Chi-Square test to find which pollutants have the strongest effect on unhealthy air days. Sucharitha reduced the dataset size using PCA to make it easier and faster to analyze while keeping most of the important information.

The final results showed that NO₂ and PM2.5 were the main pollutants causing poor air quality. The project proved that Spark is very powerful for large scale environmental analysis and can be used for future air quality monitoring, prediction, and decision-making to protect public health.