DATA INGESTION PIPELINE

TRANSFORMING AIR QUALITY DATA

AUTHOR: CARL BEBLI

PRODUCT OBJECTIVE

Objective

To build an end-to-end data ingestion pipeline that fetches, processes, and stores Air quality data for further analysis and reporting

Focus

Automating the data collection from Air Quality API, processing it, and storing it in a PostgreSQL database using Docker and Airflow.

ROLE

Project Planning

Defined the project scope, objectives and timeline

Devel*opment*

Implemented the data ingestion pipeline, developed Airflow DAGs, and configured Docker containers

Documentation

Created detailed documentation and a comprehensive README file

Testing and Debugging

Ensured data accuracy, reliability, and handled any technical issues.

PROJECT OVERVIEW

airflow : Contains Airflow DAGs and configuration files

data_fetch: Scripts for fetching air quality data from an API

database: Contains SQL scripts for database schema creation

visualize_air_quality_data.py: Python script for data visualization.

Various visualizations: Box plot, histogram, line plot, pie chart, and seaborn visualization

README.md: Documentation of the project.

TECHNICAL CHALLENGES

DOCKER CONFIGURATION → *Setting up and configuring Docker containers for Airflow and PostgreSQL*

ERROR HANDLING → *Implementing robust error handling and logging within the data pipeline.*

DATA TRANSFORMATION → *Implementing robust error handling and logging within the data pipeline.*

SOLUTION APPROACH

API

DATA INGESTION

Used Python to fetch weather data from Air Quality API

ORCHESTRATION

Employed Apache Airflow to schedule and manage the ETL pipeline.

CONTAINERIZATION

Utilized Docker to containerize the application, ensuring consistent environments.

DATA STORAGE

PostgreSQL

Stored processed data in a PostgreSQL database, ensuring it's readily accessible for analysis.

RESULT AND IMPACT

✓ AUTOMATED PIPELINE

Successfully automated the data ingestion and processing pipeline

✓ *SCALABILITY*

Built a scalable solution that can handle varying data loads

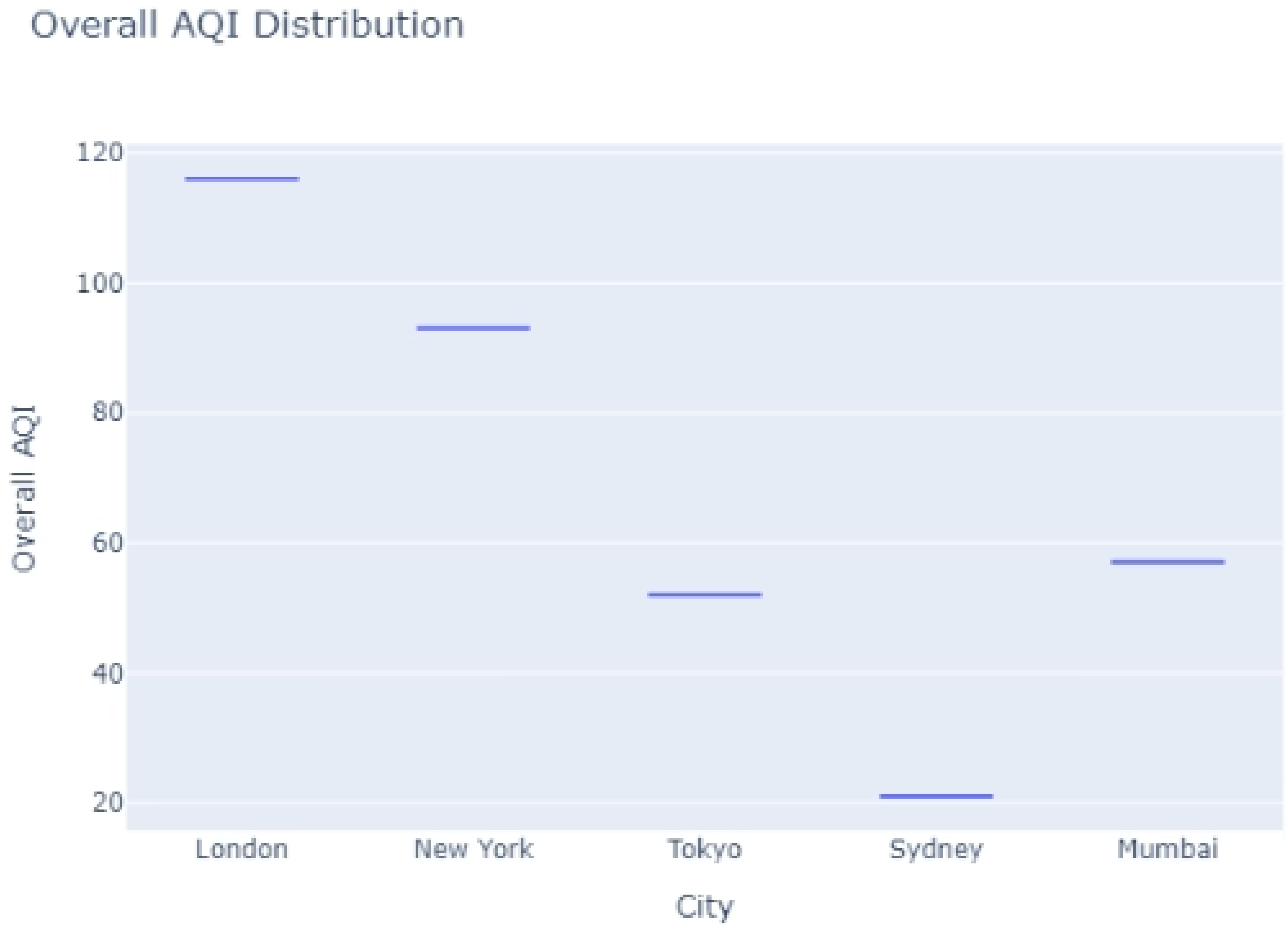
✓ *DATA ACCESSIBILITY*

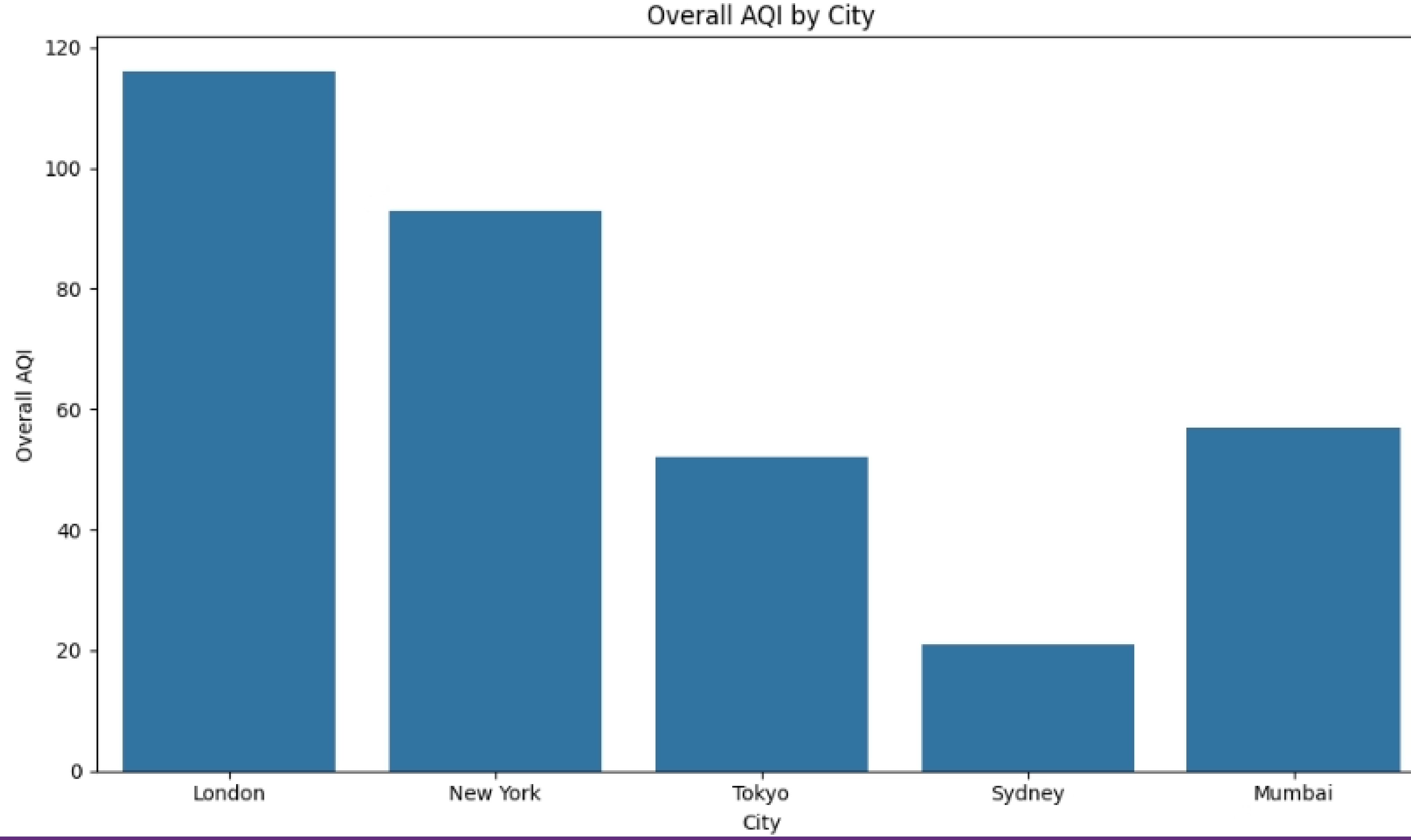
Provided easy access to processed weather data for analytics and reporting

✓ *EFFICIENCY*

Reduced manual intervention, allowing for continuous data updates and improved data accuracy

GRAPHS


Overall AQI Distribution
120
100
80
60
40
20
Overall AQI
London
New York
Tokyo
Sydney
Mumbai
City


Overall AQI by City
120
100
80
60
40
20
0
Overall AQI
London
New York
Tokyo
Sydney
Mumbai
City

LESSONS LEARNED

- ***Docker Mastery:*** *Gained indepth understanding of Docker and container orchestration.*
- *API Integration*: ***Learned best practices for integrating and handling third-party APIs.***
- ***Error Handling:*** *Developed skills in implementing robust error handling and logging mechanisms.*
- **Workflow Management:** *Enhanced my ability to design and manage data workflows using Airflow.*

ADAPTABILITY

1. Designed the pipeline to accommodate additional data sources with minimal changes

2. Built a solution that can scale horizontally to handle increased data volumes and complexity

TECHNICAL TOOLS

PYTHON	*Used for data fetching and processing scripts*
AIRFLOW	*Managed and scheduled ETL workflows*
DOCKER	*Containerized the application for consistency and scalability*
POSTGRESQL	*Stored processed data for analysis and reporting.*
AIR QUALITY API	*Source of data*

SUMMARY

- *Demonstrated strong abilities in handling complex data engineering challenges.*

- *Applied innovative solutions to create a robust and scalable data ingestion pipeline.*

- *Equipped to handle evolving data sources and project requirements.*