**Contribution Report**

**Group 15**

**Members**: Tushar Gharpure; x23289902  
Pinto Baghel; x23287501  
Utkarsh Satpute; x23135760

**Dataset Name**: [Global Greenhouse Gas Emissions Dataset]

**1. Summary of Contribution**

I have complete the important part of this project and my role is to develop and execute the ETL pipeline for this project. My responsibilities includes extracting data from APIs, transforming data and loading it into MongoDb database for initial stage and after that transforming the Json data into structured data in the form of csv and storing it into local. Furthermore, this data is loaded into postgres and then tableu for further visualization. For ETL pipeline I have used Dagster to create pipeline flow. Dagster is a orchestration tool which is used to automate the pipeline flow which will do data integration seamlessly from various sources.

**2. Tasks and Responsibilities**

**2.1 Dataset Selection**

**Dataset used**: For this project our group has selected three dataset i.e. Global Greenhouse Gas Emissions Dataset (fetched by using API Json format), Global Air Pollution Dataset(CSV) and Annual Emission Dataset(CSV). This API dataset that is Greenhouse Gas Emission dataset has all the information regarding pollutant levels in detail, which is necessary for further analysis with respect to annual emissions dataset and air quality dataset.

* **Source of the dataset**: Data was fetched by using APIs.
* **Rationale for selecting this dataset**: The dataset provides detailed pollutant-level information, which is crucial for correlating with annual emissions and air quality data, meeting the project’s objectives.
  1. **Data Preprocessing and Transformation**
* **Key Steps Taken**: The next step I did for Greenhouse Gas Emissions dataset is preprocessing and data transformation which is essential for preparation of datasets. I made a call by using python to fetch the data by using API and store it into MongoDB, before storing it to mongodb I have done some transformation that is further useful and necessary for alignment with other two datasets. This alignment is done to get mark of our project objectives. I removed the unnecessary columns such as world\_region, filtered the data to include only years between 2000 and 2021, and standardized country codes using a custom function. If there’s a country column, it adds a new column called country\_code, which changes country names into their standardized three-letter codes. In the end, it returns the cleaned and organized data, ready for saving, analysis, or storing in a database. The both csv datasets are preprocessed and clean by other two group member.
* **Tools/Technologies Used**: Python, Pandas, MongoDB, PostgreSQL, and Dagster.

**2.3 Database Integration**

* **Databases Used**: MongoDB and PostgreSQL.
* **Process**: For database integration I have used mongodb database by simply creating a database name called as tushardb. The transformed data that is stored in mongodb is exported to local. While exporting into local the data is exported in the form of structured data that is in CSV. The cleaned global gas emission data were uploaded to PostgresSQL for further analysis and visualization. This process resulted in a unified database containing all cleaned and preprocessed data, ready for analysis.
* **Outcome**: Unified and well-structured datasets ready for Tableau analysis.

**2.4 Analysis and Visualization**

* **Techniques Applied**:
  + Performed ETL automation using Dagster to ensure efficient workflow.

**2.5 Documentation and Reporting**

* **Sections Authored**: Data Preprocessing and ETL Methodology for the Greenhouse Emissions Dataset.
* **Contribution to Final Report**: Checking all the sections on ETL and integration, ensuring accuracy and completeness in workflow.

**3. Key Challenges and Solutions**

The nested JSON data returned from the API was one of the major challenges, as it required extra effort in transformation and cleaning. To solve this, I created a custom transformation function that was in charge of flattening and cleaning the data efficiently. Another challenge was making the datasets consistent, as they were in different formats with varying levels of detail. This was resolved by standardization of column names, data types, and formats during preprocessing. Other challenges were the automation of data flow across databases, this I overcame using Dagster to create a ETL pipeline. Moreover, data from MongoDB could not be directly loaded into PostgreSQL and Tableau due to port issues. I introduced a solution where the transformed data was converted to CSV format and stored locally. The locally stored CSVs were then successfully loaded into Tableau for analysis and visualization.

**4. Reflection and Lessons Learned**

This project taught me a great deal of experience in end-to-end data workflow management. I enhanced my technical skills in API integrations, data transformation, and database management by mastering ETL workflows using Dagster. I also learned more about working with MongoDB and PostgreSQL for multi-database integration. On the soft skills front, I improved my collaboration and communication by working with this team and refined my time management and problem-solving when handling big datasets. The most important problem was the fact that data could not directly be loaded into PostgreSQL and Tableau from MongoDB because of an issue at the port level. I designed a solution to transform the data into CSV format and export it, locally saving the information to then load into PostgreSQL and Tableau. This workaround made sure all datasets were integrated together nicely for analysis and visualization, despite the technical constraints.