***Final Semester Project***

***Assessing Comorbidity and Demographic Influence on covid-19 Patients Outcome from Mexican Government covid-19 Dataset from kaggle through Automated ETL, Data Analysis and Visualization.***

**Name: Odhiambo Nancy Anyango**

**Major: Data Science and Analytics**

**Course: Data Engineering, CIS 660**

**Professor: Sorio Boit**

**Semester: Winter 2025**

**Project overview**

**Introduction  
This project involved creating a complete and automated ETL workflow, data analysis and visualization for the covid-19 data set from the Mexican government from kaggle. The extraction of the data set was directly automated from kaggle site using the kagglehub api key, transformation executed in Kestra platform using python scripts to check various data characteristics and fit them to meet the required features for the project and finally the loading to PostgreSQL. Insights on the patient status outcome i.e Dead or Alive varying from different commorbidities and various age groups was analyzed and visualized using VS code as both .ipynb file for visualizations and analysis and .py for the streamlit dashboard by creating visualizations of different data merits and dashboards. The main aim of this dashboard is to improve health care and public health sector management of the pandemic in the future and any other related outbreaks by implementing the same strategies.**

1. ***Software set up***

***The softwares used in this project were;***

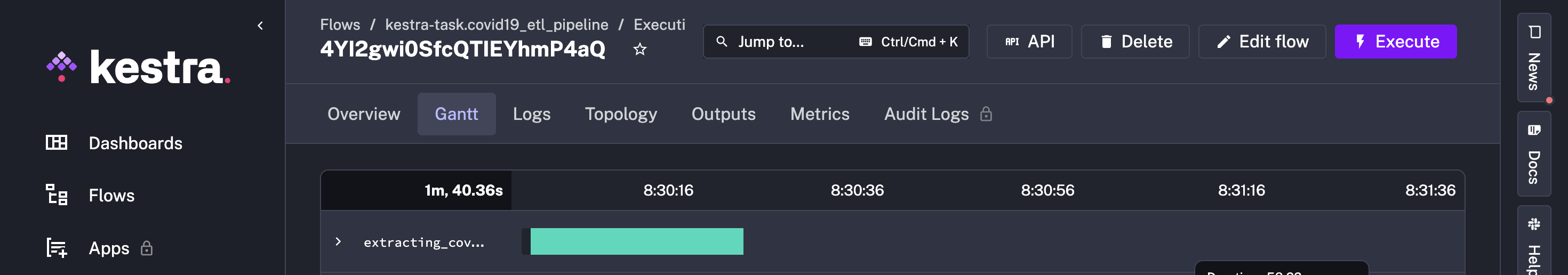
* ***VS code, Docker, Kestra, Postgres***

***VS code was used in the initial stages of the project to run a demo of the whole ETL process but in a python environment with a .py script. This was necessitated by the need verify various transformational needs and the characteristics of the data that need to be transformed. After demo project the docker-compose yaml was run in vs code for containerization of the postgres-db and Kestra. The containerization was a success and the result showed that postgres-db was running on port 5432 and was healthy while Kestra was running on port 8080.***

|  |  |
| --- | --- |
|  |  |

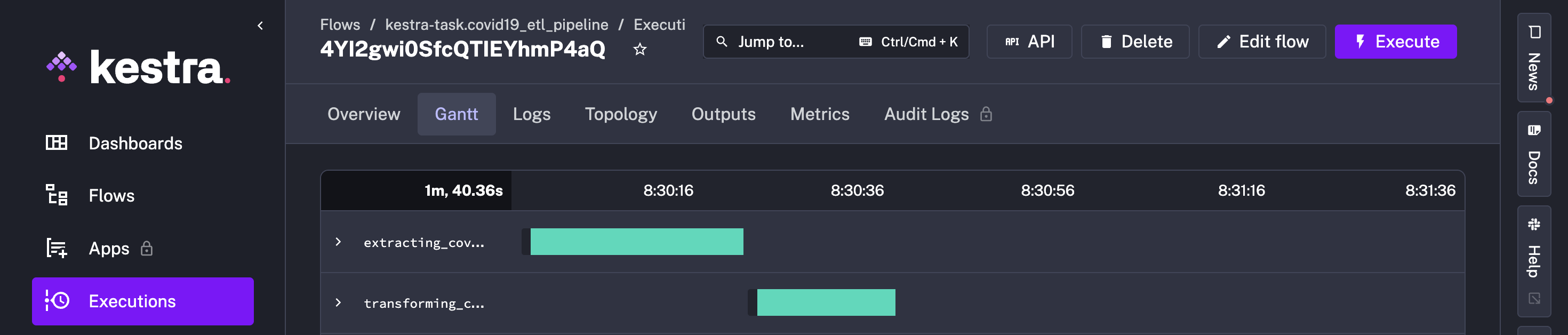
1. ***Data extraction***

***The ETL pipeline named* covid19\_etl\_pipeline was designed to automate the extraction of t*he Mexican covid-19 data set from kaggle using the authenticated API request into Kestra software in a task named extracting\_covid\_data.* The process began by setting up a Python virtual environment using the venv module followed by activating the environment and importing pandas, numpy, and requests for data handling and API communication. The code was configured to utilize Kaggle API credentials through the variables KAGGLE\_USERNAME and KAGGLE\_KEY which allowed secure access to Kaggle’s dataset repository. The COVID-19 dataset was downloaded directly in zip format directly from Kaggle using an HTTP GET request where authorization was passed via the API key in the request headers. The downloaded content being a zip file was handled using Python’s ZipFile class and extracted in-memory using BytesIO to avoid saving the zip file locally. "Covid Data.csv" file within the archive was read into a pandas DataFrame while specifying the encoding format as 'windows-1252' to handle potential character issues. Finally the extracted data was saved locally as a CSV file named covid\_data.csv completing the extraction phase and preparing the data for subsequent transformation and loading stages within the pipeline.**

******

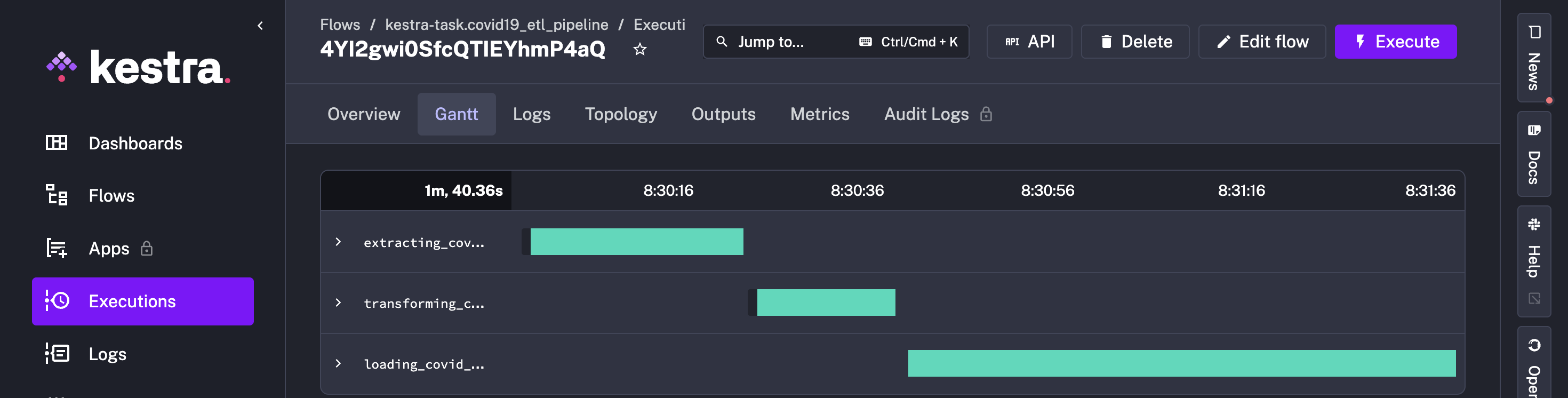
1. ***Data Transformation***

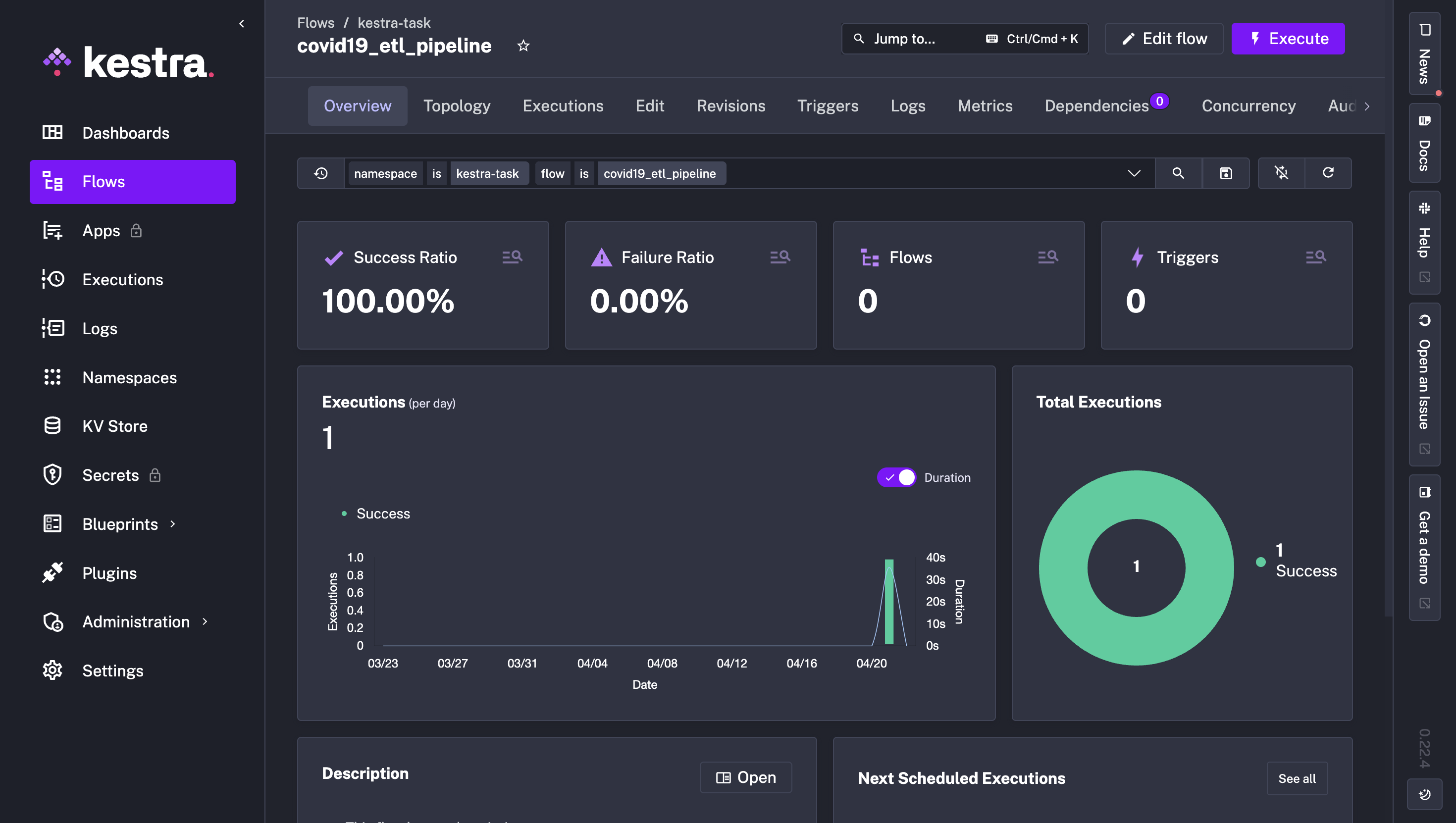
In the **transformation phase of the ETL pipeline the COVID-19 dataset was cleaned and preprocessed to prepare it for analysis and visualization. Python virtual environment and the required libraries were installed into a pandas DataFrame. Column names were standardized by converting them to snake case and lowercase for consistency. Several columns were renamed to more meaningful names such as converting "hipertension" to "hypertension", "patient\_type" to "admission\_status." The ICU and intubation statuses were adjusted by filling them with 'No' encoded as 2 where patients were not admitted since they had 97, 98 and 99 values which represented missing values. The dataset was subsetted to retain only relevant columns of interest from the whole data frame from 21 variables to 15 variables. The classification of infection severity was filtered for rows with either mild, moderate, or severe to only capture those that were confirmed covid-19 patients and remove the rest of the patients in the database. Custom mapping functions were applied to transform numeric-coded values into human-readable categorical labels, such as converting sex codes to "Male" or "Female" and status codes to "Alive" or "Dead." Binary fields like diabetes or hypertension were also converted from numeric codes to "Yes" or "No." Any rows containing unknown values or placeholder codes like 97, 98, or 99 in the age column were removed to enhance data reliability. Additionally outlier removal was performed on the age variable using the interquartile range (IQR) method. A copy of the cleaned data was then created for encoding purposes where binary categorical variables were mapped to 1s and 0s and one-hot encoding was applied to the classification field for correlation analysis. The script also engineered an age\_group feature to categorize patients into age brackets and calculated the proportion of critical health outcomes (ICU admission, intubation and death) within each age group. Finally the transformed dataset was saved as covid\_data.csv ready for loading into the database.**

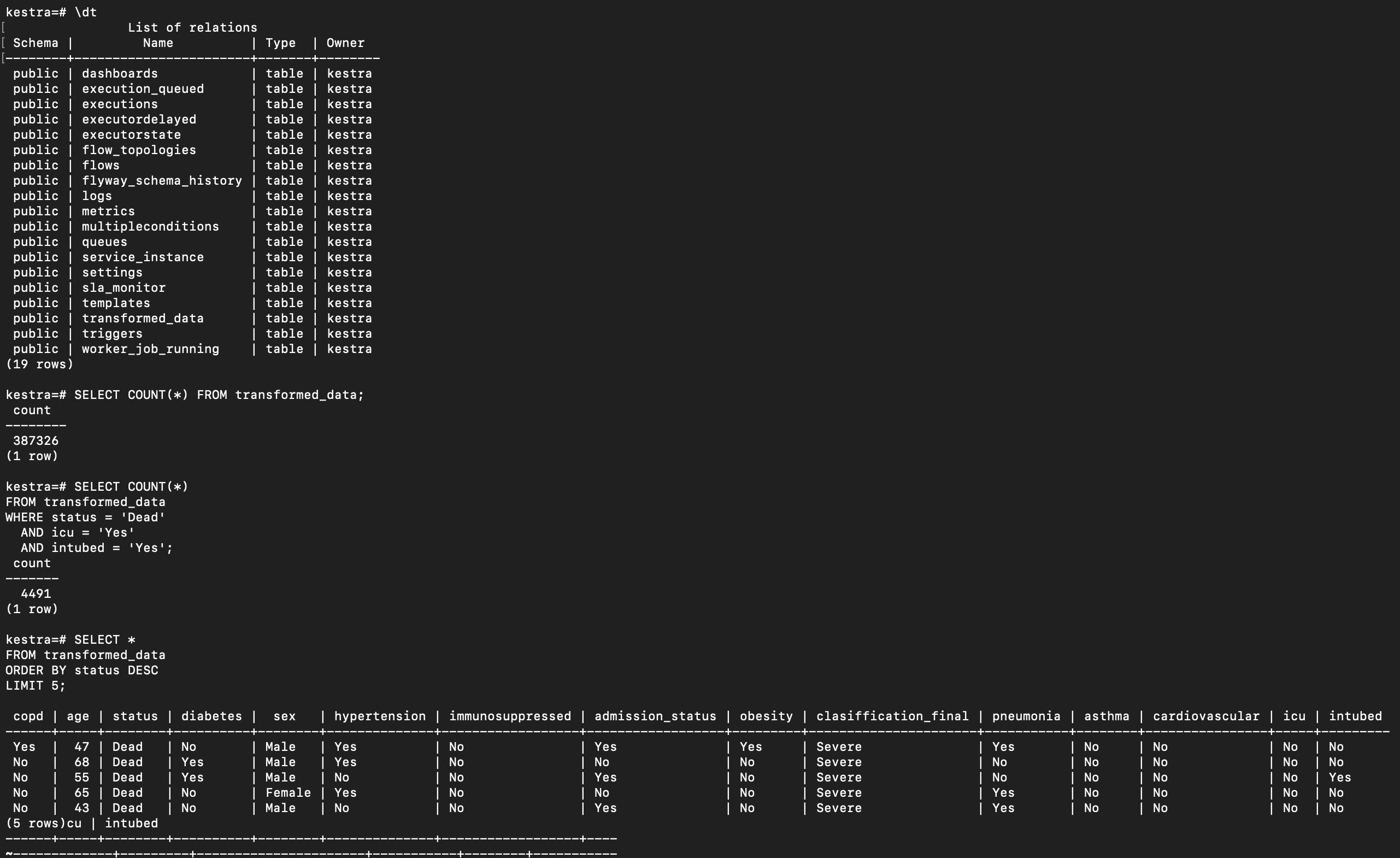
******

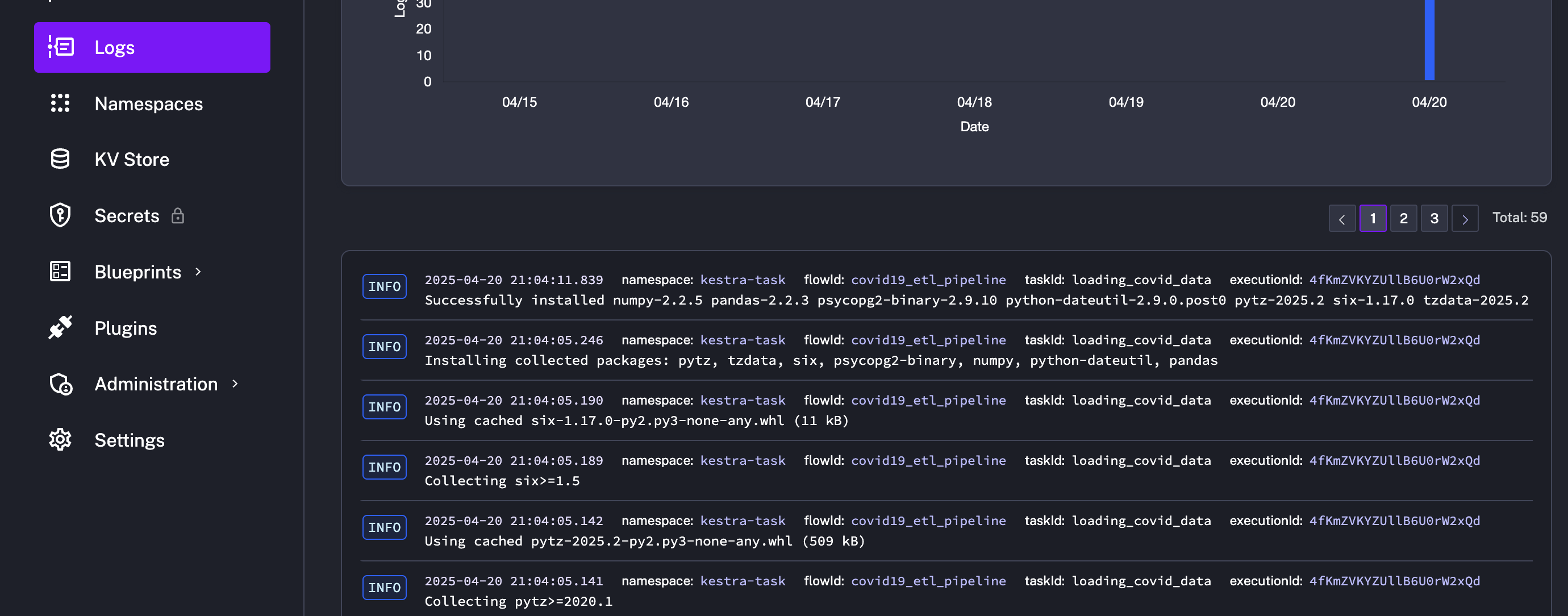
1. ***Data loading postgres***

In **the loading phase of the ETL pipeline the transformed data was loaded into postgres database named kestra in docker.A Python virtual environment was initialized and the required libraries installed. It then read the output CSV file generated from the transformation step into a pandas DataFrame. A connection was established to the PostgreSQL database running within a Docker container using the psycopg2 library providing necessary connection parameters such as the host, database name, username, and password. Once the connection was established a SQL query to create a table named transformed\_data if it did not already exist was executed. This table was structured to include columns that align with the transformed data fields specifying appropriate data types for each. After ensuring the table was in place my code iterated over each row in the DataFrame and inserted the corresponding data into the PostgreSQL table using prepared SQL statements to avoid SQL injection and ensure data integrity. The changes were committed and finally both the cursor and the database connection closed successfully completing the data loading process. This step ensured that the transformed COVID-19 data was securely stored in a structured relational database ready for querying, analysis and reporting. Th e transformed data has 387,326 rows from** 1,048,576 **and 15 columns/ features from 21 features.**

******

******

******

******

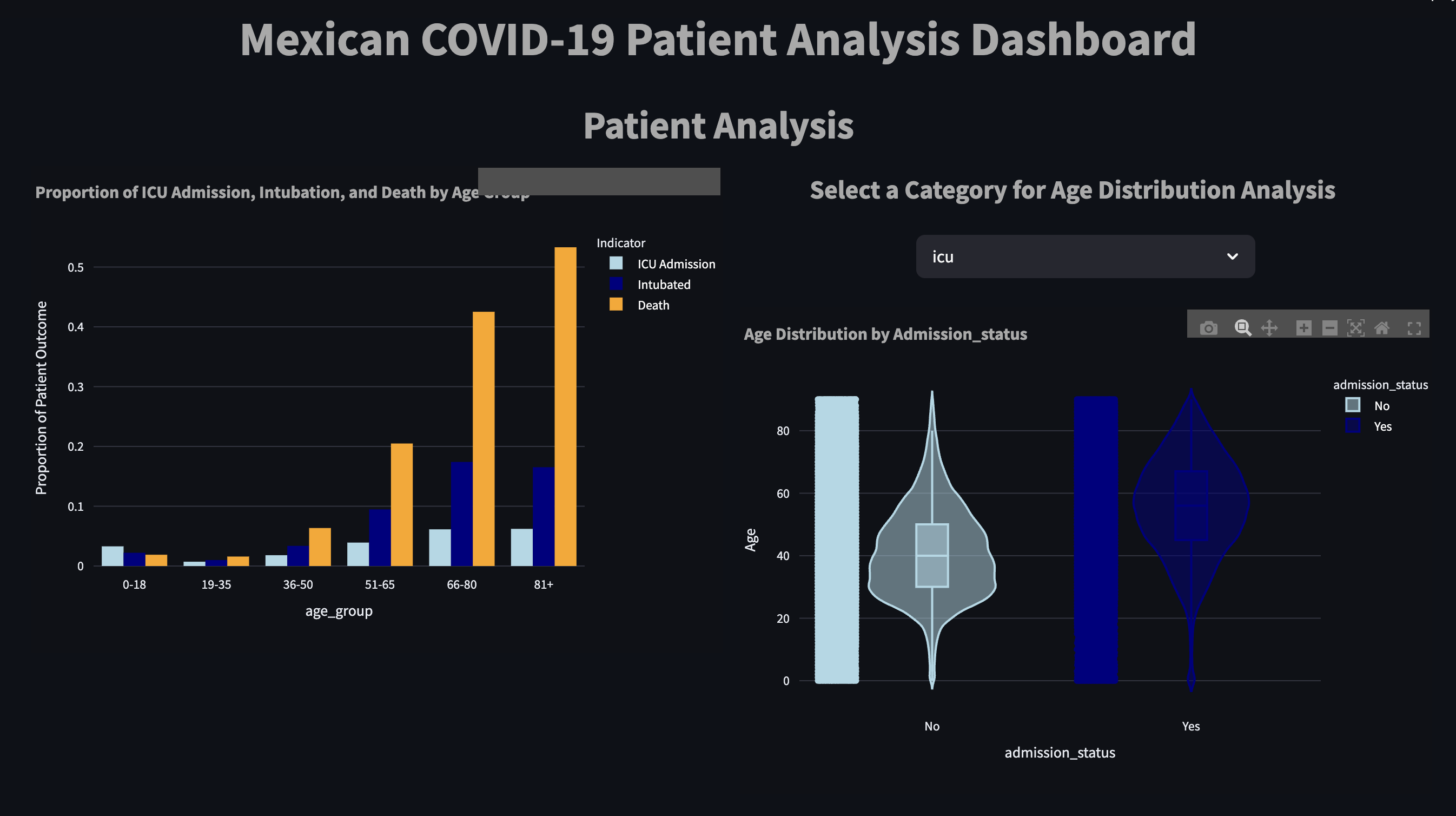
1. ***Data loading to VS code for visualization and analysis***

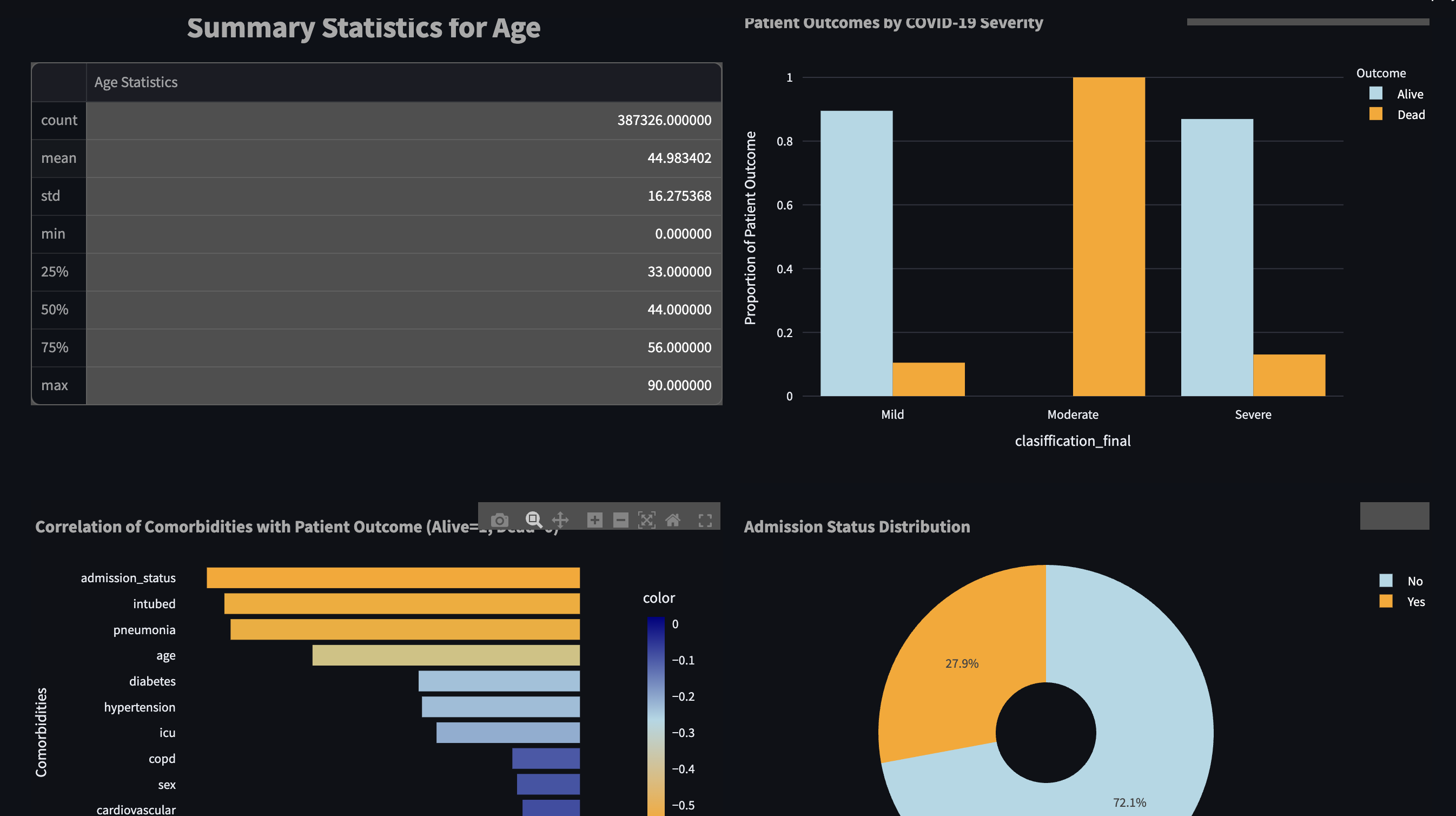
***From my project overview my aim was to analyze how different commobidities affect covid-19 patients and their connection with patient outcomes and the interventions required for patient care in times of pandemic.* This analysis uses SQLAlchemy, Pandas, Seaborn and Matplotlib to explore COVID-19 patient data stored in a PostgreSQL database called kestra containerized in docker. The transformed data is read into a DataFrame to generate key visualizations including severity-based outcomes, ICU and intubation rates by age group and age distributions across patient status categories. It further examines comorbidity co-occurrence among deceased patients and computes correlations between features and mortality. The variables in this dat set are; copd, sex, age, admission\_status, status, clasiffication\_final, obesity, pneumonia, diabetes, intubed, icu, asthma, cardiovascular, immunosuppressed.**

|  |  |  |
| --- | --- | --- |
|  |  |  |
|  |  |  |

1. ***Streamlit Dashboard creation.***

Created The **Mexican COVID-19 Patient Analysis Dashboard as an interactive web application built with Streamlit, leveraging PostgreSQL for data storage and Plotly for visualizations. It analyzes patient outcomes, age distribution, ICU admissions, intubation rates and comorbidity correlations with mortality. The dashboard features optimized performance through caching mechanisms. Key insights include outcome differences by severity level, age-related risks, and the impact of comorbidities on patient survival providing a clear data-driven overview of COVID-19 trends in Mexico during the covid-19 pandemic period. This dashboard as well allows a user to use the drop down menu to pick the type of outcome required to be visualized by age to check for density of the distribution of each that is the icu, intubation and death status outcomes.**

******

******

1. ***Insights from data analysis, visualization and dashboard.***

***From the outcomes visualizations in the dashboard it is clear to infer that older age group were mostly affected by covid-19. There is increase in intubation, icu admissions and death cases gradually as age increases among the covid-19 patients. The violin plots show density increase in age for the outcomes as well with most higher than the projected mean of 44yrs.***

***Something intriguing from the severity plot with proportion of patient status outcome, those with moderate severity were reported to have died the most unlike the thinking would be that those with severe outcome would be the worst hit. I suppose this means that there might have been less intervention accorded to the patients that showed moderate severity which led to their demise in bigger numbers. Those with mild were as well not affected much. The severe group received a good care package which saw them through the treatment of covid-19 infection.***

***The correlation plot showed how each and every variable correlated to status i.e. Death, survival which showed different scales such as admission\_status (admitted Yes/ No) showing great correlation with death meaning the likelihood to dying is highly related to being admitted to the hospital than not being admitted, followed by intubation, pneumonia, increase in age, diabetes, hypertension, icu, copd, sex, cardiovascular, obesity, immunosuppressed and lastly asthma which shows a positive correlation meaning there is likely higher chances of survival with asthma condition.***

***And the last visualization, pie chart shows that all patients who were tested for covid admission status distribution. Shows that 27.9% of the patients were admitted while 72.1% were not submitted and might have been taking home based care at that time.***

1. ***Conclusion & Future recommendations***

***From the analysis I would conclude that the there is need for enhanced care for the elderly in terms of pandemic since they are mostly affected either because of immune system being weak or other underlying factors, home based care is necessary for those who show mild symptoms and those with moderate symptoms require keen observation and enhanced care from the health providers to ensure that all the risk factors are arrested in time within the time that they are admitted in hospital.***

***Need for ready infrastructure for intubation as it shows that this is a highly required support for the patients during the pandemic.***

***Future improvement for ETL process***

* ***Focus more on improving ETL skills and google cloud interaction with postgres.***
* ***Review of big query and nginx***
* ***Build on the project and come up with a finalized working project with key insights, modeling & evaluation and future recommendations in the dashboard.***