San Francisco Crime Pattern Analysis with Snowflake, Airflow, dbt, and PowerBI

Pranav Reddy Gaddam[#1], Anvay Bhanap[*2]
Yashwanth Reddy Katipally[#3], Thejes Raj Gangadhar[*4]

[#]*Department of Applied Data Science, San Jose State University*
1 Washington Sq, San Jose, CA 95192, USA
[1]pranavreddy.gaddam@sjsu.edu
[2]anvay.bhanap@sjsu.edu
[3]yashwanthreddy.katipally@sjsu.edu
[4]thejesraj.gangadhar@sjsu.edu

***Abstract* – This project aims to investigate the crime patterns present in San Francisco by analyzing incident reports from 2018 to 2024. Utilizing both historical and real-time data information, this study explores temporal trends, geographic crime distributions, and resolution effectiveness. The process integrates Snowflake for warehousing, Airflow for ETL workflows, dbt for ELT transformations, and PowerBI for dashboarding and visualizations. Findings reveal noticeable trends, including peak incidents during summer months, spatial clustering in areas with high population density, and an overall doubling of crime post-COVID. The proposed solution highlights how actionable insights can guide effective resource allocation and strategic policymaking, enabling the city to enhance its crime management efforts.**

***Keywords* - ETL, ELT, COVID, dbt, DAG, Socrata**

I. Problem Statement

San Francisco's crime rates are an everlasting concern for residents, tourists, and policymakers, with recent trends indicating further challenges. As of this year, the city has experienced a complex crime landscape characterized by persistent property crimes, increased drug-related offences, and significant incidents of theft and violence. Understanding the yearly and locational patterns of crime is essential for preventive measures and resource distribution to take place. Despite available data, deriving insights and conducting effective actions based on the data alone remains challenging. The objective of this report is to present a solution that can analyze the data over the past 6 years and glean patterns by time, area, and incident category to provide government officials and law enforcement with an evidence-based understanding of what measures need to be taken to make strides in crime prevention and policy development.

II. Introduction

As crime trends evolve in urban areas, analyzing incident reports is crucial for proactive safety measures. San Francisco's social environment in 2024 reflects a complex outlook, from the staggering rise in property crimes and organized retail theft. Recent data also suggests a persistent rise in car break-ins, with over 6000 incidents reported in the first quarter of 2024 alone, and a notable surge of drug peddling in open air markets, especially in neighborhoods like the Tenderloin and South of Market. Violent crimes, though not increasing as drastically, remain a critical concern, with assault and robbery rates presenting a continuous challenge for law enforcement. San Francisco's unique socio-economic state, marked by its high-density localities and financial and cultural diversity, makes it a particular challenge to address issues in crime management. This project adopts an ELT approach to process crime incident data, utilizing Airflow for scheduling tasks and dbt for refining data transformations. By integrating these datasets into PowerBI dashboards, hidden trends can be brought to light, such as crime hotspots and seasonal spikes, supporting efforts to improve community safety.

III. Research

San Francisco's crime landscape presents a complex challenge for residents, tourists, and policymakers alike. Persistent property crimes, increased drug-related offenses, and rising incidents of theft and violence call for a more detailed analysis of crime patterns across different times, locations, and categories. Several studies have contributed to understanding these dynamics, applying advanced methodologies such as machine learning, spatial analysis, and complex network analysis to uncover crime trends and provide actionable insights. Machine learning has been instrumental in crime prediction and analysis. For example, Rasool et al. [1] used machine learning models to forecast crime trends in San Francisco, highlighting how such techniques can aid law enforcement in anticipating crime occurrences and optimizing resource allocation. Similarly, Pradhan [2] conducted exploratory data analysis (EDA) on crime data, identifying key temporal and spatial patterns that could inform crime prevention strategies. The ability to predict crime occurrences using machine learning models helps policymakers address criminal activity proactively, rather than reactively. In addition to temporal analysis, spatial patterns play a significant role in understanding crime. Chen et al. [3] explored how changes in human activities, especially during the COVID-19 pandemic, influenced crime patterns in San Francisco. They found that stay-at-home mandates led to notable shifts in crime distribution, indicating the importance of integrating human behavior dynamics when analyzing crime data. This spatio-temporal understanding of crime is vital for resource allocation and planning effective prevention measures, as it helps target high-crime areas during specific times.

Complex network analysis offers another powerful tool for uncovering hidden patterns in crime data. Spadon et al. [4] applied this methodology to study the interconnections between crime events in urban areas. By treating crime data as a network, they were able to identify crime hotspots and areas with high criminal activity. This approach provides a more granular understanding of crime occurrences, which is essential for law enforcement agencies seeking to prioritize intervention in specific neighborhoods. Moreover, advanced statistical models like the Gaussian Cox process have been used to model crime events over both time and space. Shirota and Gelfand [5] applied this approach to analyze crime in San Francisco, focusing on how crime events unfold in a given location over time. Their research contributes to developing predictive models for identifying crime hotspots, helping law enforcement agencies adjust their strategies accordingly. These studies collectively demonstrate that analyzing crime through machine learning, spatial analysis, complex networks, and statistical models provides significant insights into criminal behavior in San Francisco. By applying these methodologies, policymakers and law enforcement agencies can craft evidence-based strategies for crime prevention, better allocate resources, and address the evolving challenges in urban crime dynamics.

IV. Methodology

Analyzing crime trends in San Francisco needs a strong pipeline to integrate historical and live data, facilitating workable insights for city officials. With the city facing increasing crime rates, understanding the dynamics that cause such results is crucial for strategic financial allocation and law enforcement planning. The methodology detailed in the following sections provides a comprehensive overview of how these crime hotspots and yearly trends are found by integrating historical datasets with recent updates, securing an accurate and insightful crime management solution. Figure 1 shows how the ETL pipeline was constructed using Airflow, dbt, and PowerBI, implementing a straightforward process from data extraction to visualization. Initially, crime data was extracted from the San Francisco government data portal and loaded into Snowflake. Airflow automated the extraction and stages the processes through DAG to maintain the scheduled updates. This ensured that historical data and live updates could coexist in a structured and accessible environment, prepared for the next phase. During transformation, dbt refined and aggregated the data to produce a structured dataset that highlighted crime patterns by district and category. This dbt framework optimized transformations directly into Snowflake, which maintained a high level of efficiency by avoiding unnecessary data transfers. Incremental updates were implemented using the datetime library to make sure that only the new records were processed, which significantly reduced resource consumption and improved system performance. Then, PowerBI dashboard visualized the collected data, and displayed interactive views of crime patterns, hotspots, and incident categories. Policymakers can explore the time-based trends to empower city officials to work towards solutions and preventative measures. These visualizations can aid strategic decision-making, confirming that resources are granted intentionally to areas with higher crime incidence, thereby optimizing public safety efforts and fostering stronger community engagement. Further implementation of ongoing updates was achieved through an ELT approach. Live data was integrated using Socrata API with raw entries loaded directly into Snowflake. Previously described aggregation methods were handled downstream in dbt. The model optimized data integration by using datetime based incremental updates, which minimized computational resources compared to ETL's traditional row-by-row merging. This dual implementation facilitated comprehensive data processing, which accommodated both historical records and real-time information while maintaining analytical scalability.

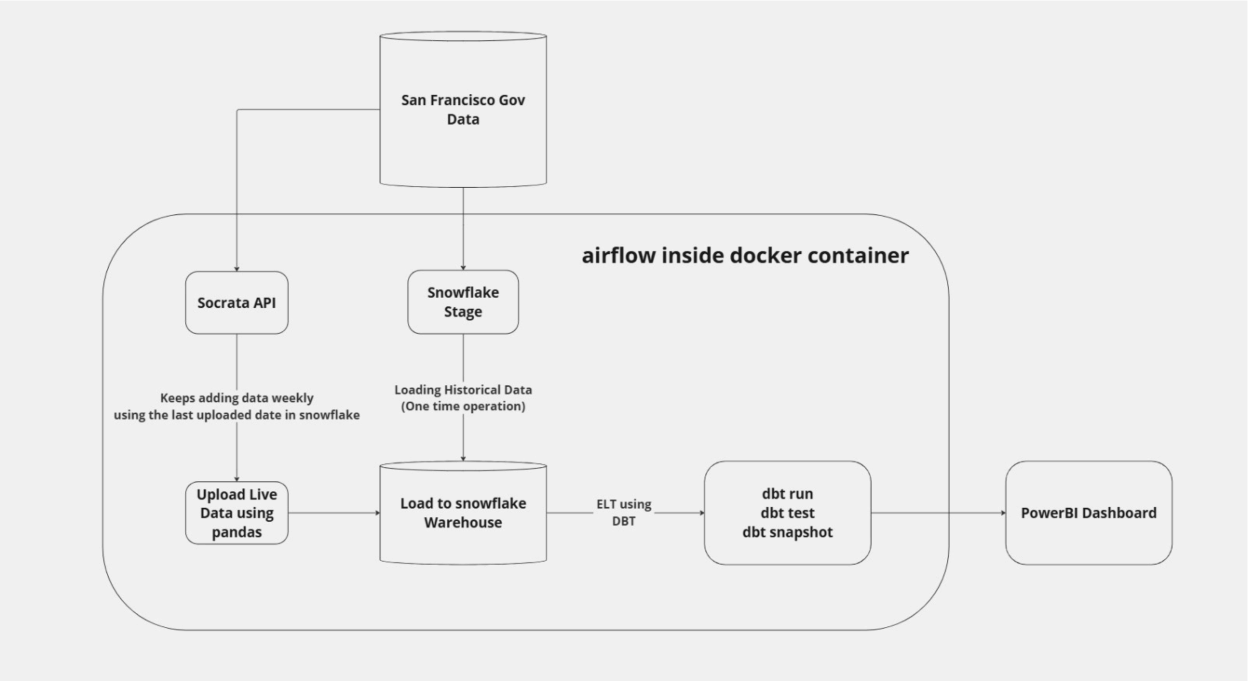

San Francisco Gov Data
airflow inside docker container
Socrata API
Snowflake Stage
Keeps adding data weekly using the last uploaded date in snowflake
Loading Historical Data (One time operation)
Upload Live Data using pandas
Load to snowflake Warehouse
ELT using DBT
dbt run
dbt test
dbt snapshot
PowerBI Dashboard

Fig. 1. System diagram of data pipeline for crime trend analysis using SFODP, Airflow, dbt, and Snowflake

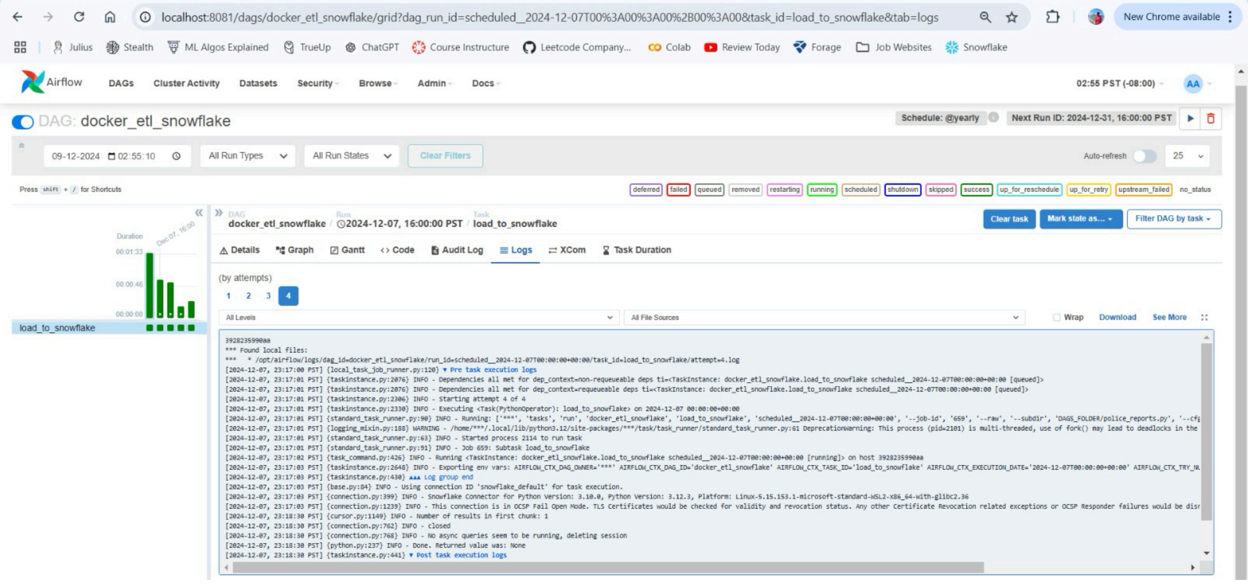
A. Extracting Historical Data from San Francisco Open Data Portal (see Python code)

In order to establish the foundation for crime trend analysis, the project was initiated by sourcing historical data of the past 6 years from the San Francisco Open Data Portal (SFODP) as a CSV file. The file consisted of a variety of fields referencing the day and time of the event – Incident Datetime, Report Datetime, Incident Day of Week –, the type of crime that occurred –

Incident Category, Subcategory, Report Type, Resolution –, and the area in which it happened – latitude, longitude, neighborhood – as well as the nearest police district. All together, there were 862,000 rows of information gathered from all San Francisco neighborhoods. The preprocessing began with cleaning the data to remove missing values and standardizing the datetime format. Additionally, irrelevant columns, like Dispatch Number and Supervisor District, were removed from the working dataset as they were irrelevant to the overarching story being told through the data. By the end of the cleaning process, the 34 columns were reduced to a usable 13, as shown in Table 1. The resulting dataset was uploaded into a Snowflake data warehouse using its staging feature, ensuring secure storage. Though the data extraction process was manual, it was automated for consistency using a SnowSQL script integrated with Airflow. A Directed Acyclic Graph (DAG) was developed to regulate this upload procedure (Fig. 2), allowing for seamless integration into Snowflake. While this DAG was designed as a one-time execution, because of the static nature of a historical data CSV, it showcased the project's scalable approach to handling similar datasets in the future.

TABLE I
DATA TYPES FOR INCIDENTS TABLE

Column Name	**Data Type**	**Constraints**
Incident ID	VARCHAR	PRIMARY KEY (datetime, ID)
Incident Datetime	TIMESTAMP	
Incident Category	VARCHAR	
Incident Day of Week	VARCHAR	
Analysis Neighborhood	VARCHAR	
Incident Subcategory	VARCHAR	
Intersection	VARCHAR	
Latitude	VARCHAR	
Longitude	VARCHAR	
Police District	VARCHAR	
Report Datetime	TIMESTAMP	
Row ID	VARCHAR	
Resolution	VARCHAR	

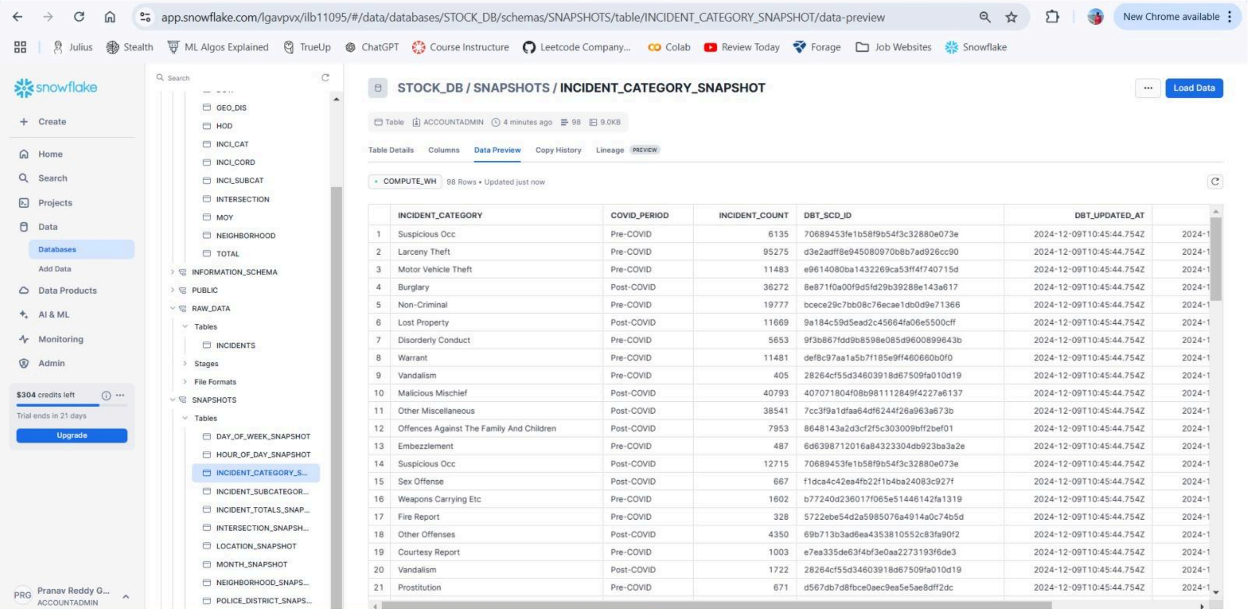


```
3928235990aa
*** Found local files:
***   * /opt/airflow/logs/dag_id=docker_etl_snowflake/run_id=scheduled__2024-12-07T00:00:00+00:00/task_id=load_to_snowflake/attempt=4.log
[2024-12-07, 23:17:00 PST] {local_task_job_runner.py:120} ▼ Pre task execution logs
[2024-12-07, 23:17:01 PST] {taskinstance.py:2076} INFO - Dependencies all met for dep_context=non-requeueable deps ti=<TaskInstance: docker_etl_snowflake.load_to_snowflake scheduled__2024-12-07T00:00:00+00:00 [queued]>
[2024-12-07, 23:17:01 PST] {taskinstance.py:2076} INFO - Dependencies all met for dep_context=requeueable deps ti=<TaskInstance: docker_etl_snowflake.load_to_snowflake scheduled__2024-12-07T00:00:00+00:00 [queued]>
[2024-12-07, 23:17:01 PST] {taskinstance.py:2306} INFO - Starting attempt 4 of 4
[2024-12-07, 23:17:01 PST] {taskinstance.py:2330} INFO - Executing <Task(PythonOperator): load_to_snowflake> on 2024-12-07 00:00:00+00:00
[2024-12-07, 23:17:01 PST] {standard_task_runner.py:90} INFO - Running: ['***', 'tasks', 'run', 'docker_etl_snowflake', 'load_to_snowflake', 'scheduled__2024-12-07T00:00:00+00:00', '--job-id', '659', '--raw', '--subdir', 'DAGS_FOLDER/police_reports.py', '--cfg
[2024-12-07, 23:17:01 PST] {logging_mixin.py:188} WARNING - /home/***/.local/lib/python3.12/site-packages/***/task/task_runner/standard_task_runner.py:61 DeprecationWarning: This process (pid=2101) is multi-threaded, use of fork() may lead to deadlocks in the
[2024-12-07, 23:17:01 PST] {standard_task_runner.py:63} INFO - Started process 2114 to run task
[2024-12-07, 23:17:01 PST] {standard_task_runner.py:91} INFO - Job 659: Subtask load_to_snowflake
[2024-12-07, 23:17:02 PST] {task_command.py:426} INFO - Running <TaskInstance: docker_etl_snowflake.load_to_snowflake scheduled__2024-12-07T00:00:00+00:00 [running]> on host 3928235990aa
[2024-12-07, 23:17:03 PST] {taskinstance.py:2648} INFO - Exporting env vars: AIRFLOW_CTX_DAG_OWNER='***' AIRFLOW_CTX_DAG_ID='docker_etl_snowflake' AIRFLOW_CTX_TASK_ID='load_to_snowflake' AIRFLOW_CTX_EXECUTION_DATE='2024-12-07T00:00:00+00:00' AIRFLOW_CTX_TRY_NU
[2024-12-07, 23:17:03 PST] {taskinstance.py:430} ▲▲▲ Log group end
[2024-12-07, 23:17:03 PST] {base.py:84} INFO - Using connection ID 'snowflake_default' for task execution.
[2024-12-07, 23:17:03 PST] {connection.py:399} INFO - Snowflake Connector for Python Version: 3.10.0, Python Version: 3.12.3, Platform: Linux-5.15.153.1-microsoft-standard-WSL2-x86_64-with-glibc2.36
[2024-12-07, 23:17:03 PST] {connection.py:1239} INFO - This connection is in OCSP Fail Open Mode. TLS Certificates would be checked for validity and revocation status. Any other Certificate Revocation related exceptions or OCSP Responder failures would be disr
[2024-12-07, 23:18:30 PST] {cursor.py:1149} INFO - Number of results in first chunk: 1
[2024-12-07, 23:18:30 PST] {connection.py:762} INFO - closed
[2024-12-07, 23:18:30 PST] {connection.py:768} INFO - No async queries seem to be running, deleting session
[2024-12-07, 23:18:30 PST] {python.py:237} INFO - Done. Returned value was: None
[2024-12-07, 23:18:30 PST] {taskinstance.py:441} ▼ Post task execution logs
```

Fig. 2. Airflow logs of loading historical data to Snowflake

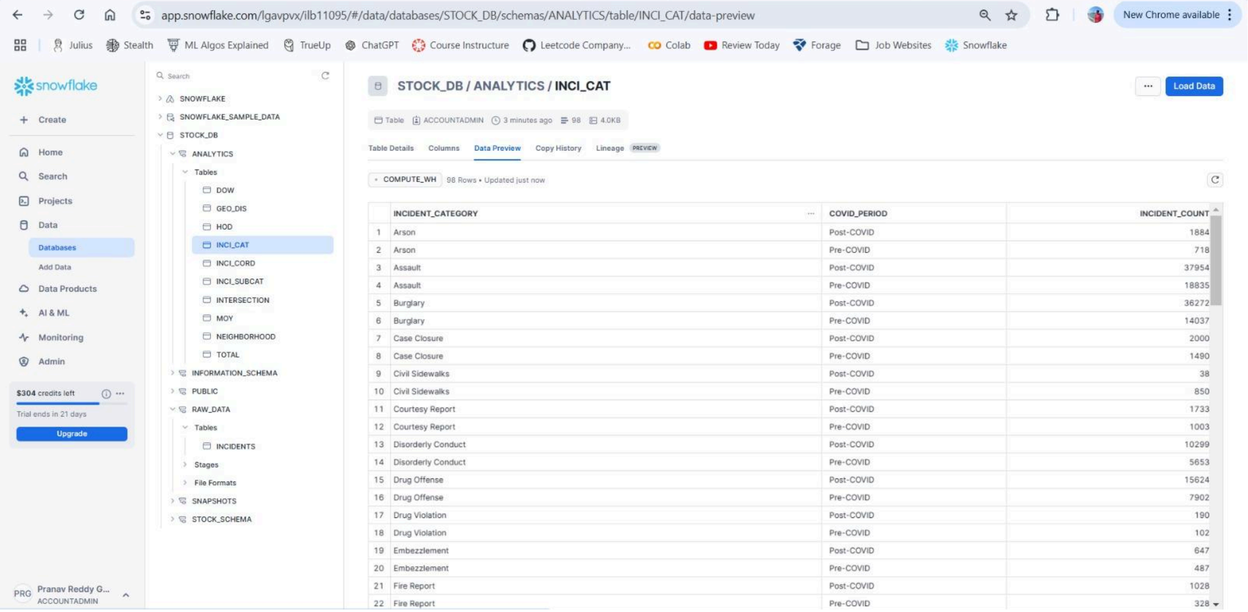
B. *Implementing ETL Processes with dbt (see SQL code)*

Once the historical data was ingested, dbt (Data Build Tool) was implemented to perform the essential ETL processes to refine and transform the raw data. The transformations were centered on aggregating crimes by category and district (Fig. 3, Fig. 4) allowing the data to present associated patterns for any upcoming decision-making. The modular SQL-driven framework provided by dbt enabled all the creation of reusable models to dynamically adjust updates or new data sources. Furthermore, dbt facilitated data quality checks (Fig. 5, Fig, 6, Fig. 6) to ensure that accuracy and integrity were maintained prior to the analysis phase. By performing these transformations inside Snowflake, it was possible to keep the data transfer overhead to a minimum and focus on improving efficiency. The result generated a structured dataset ready for visualization and deeper exploration.



	INCIDENT_CATEGORY	COVID_PERIOD	INCIDENT_COUNT	DBT_SCD_ID	DBT_UPDATED_AT	
1	Suspicious Occ	Pre-COVID	6135	70689453fe1b58f9b54f3c32880e073e	2024-12-09T10:45:44.754Z	2024-1
2	Larceny Theft	Pre-COVID	95275	d3e2adff8e945080970b8b7ad926cc90	2024-12-09T10:45:44.754Z	2024-1
3	Motor Vehicle Theft	Pre-COVID	11483	e9614080ba1432269ca53ff4f740715d	2024-12-09T10:45:44.754Z	2024-1
4	Burglary	Post-COVID	36272	8e871f0a00f9d5fd29b39288e143a617	2024-12-09T10:45:44.754Z	2024-1
5	Non-Criminal	Pre-COVID	19777	bcece29c7bb08c76ecae1db0d9e71366	2024-12-09T10:45:44.754Z	2024-1
6	Lost Property	Post-COVID	11669	9a184c59d5ead2c45664fa06e5500cff	2024-12-09T10:45:44.754Z	2024-1
7	Disorderly Conduct	Pre-COVID	5653	9f3b867fdd9b8598e085d9600899643b	2024-12-09T10:45:44.754Z	2024-1
8	Warrant	Pre-COVID	11481	def8c97aa1a5b7f185e9ff460660b0f0	2024-12-09T10:45:44.754Z	2024-1
9	Vandalism	Pre-COVID	405	28264cf55d34603918d67509fa010d19	2024-12-09T10:45:44.754Z	2024-1
10	Malicious Mischief	Post-COVID	40793	407071804f08b981112849f4227a6137	2024-12-09T10:45:44.754Z	2024-1
11	Other Miscellaneous	Post-COVID	38541	7cc3f9a1dfaa64df6244f26a963a673b	2024-12-09T10:45:44.754Z	2024-1
12	Offences Against The Family And Children	Post-COVID	7953	8648143a2d3cf2f5c303009bff2bef01	2024-12-09T10:45:44.754Z	2024-1
13	Embezzlement	Pre-COVID	487	6d6398712016a84323304db923ba3a2e	2024-12-09T10:45:44.754Z	2024-1
14	Suspicious Occ	Post-COVID	12715	70689453fe1b58f9b54f3c32880e073e	2024-12-09T10:45:44.754Z	2024-1
15	Sex Offense	Post-COVID	667	f1dca4c42ea4fb22f1b4ba24083c927f	2024-12-09T10:45:44.754Z	2024-1
16	Weapons Carrying Etc	Pre-COVID	1602	b77240d236017f065e51446142fa1319	2024-12-09T10:45:44.754Z	2024-1
17	Fire Report	Pre-COVID	328	5722ebe54d2a5985076a4914a0c74b5d	2024-12-09T10:45:44.754Z	2024-1
18	Other Offenses	Post-COVID	4350	69b713b3ad6ea4353810552c83fa90f2	2024-12-09T10:45:44.754Z	2024-1
19	Courtesy Report	Pre-COVID	1003	e7ea335de63f4bf3e0aa2273193f6de3	2024-12-09T10:45:44.754Z	2024-1
20	Vandalism	Post-COVID	1722	28264cf55d34603918d67509fa010d19	2024-12-09T10:45:44.754Z	2024-1
21	Prostitution	Pre-COVID	671	d567db7d8fbce0aec9ea5e5ae8dff2dc	2024-12-09T10:45:44.754Z	2024-1

Fig. 3. dbt snapshot of incident categories in Snowflake



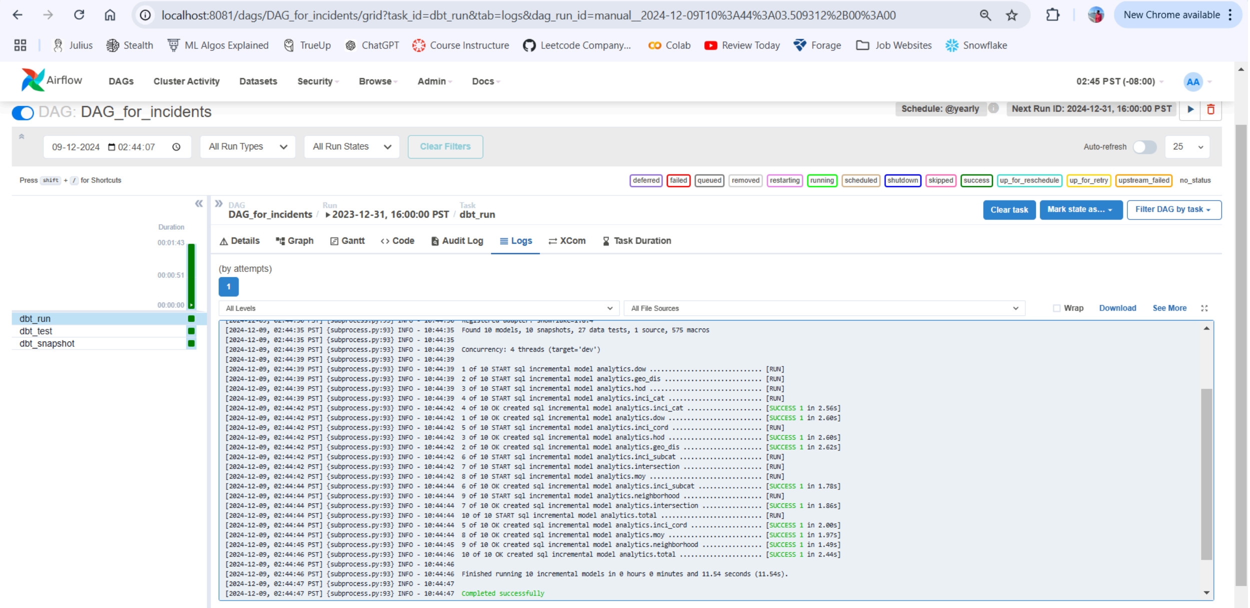
STOCK_DB / ANALYTICS / INCI_CAT

Table · ACCOUNTADMIN · 3 minutes ago · 98 · 4.0KB

COMPUTE_WH · 98 Rows • Updated just now

	INCIDENT_CATEGORY	COVID_PERIOD	INCIDENT_COUNT
1	Arson	Post-COVID	1884
2	Arson	Pre-COVID	718
3	Assault	Post-COVID	37954
4	Assault	Pre-COVID	18835
5	Burglary	Post-COVID	36272
6	Burglary	Pre-COVID	14037
7	Case Closure	Post-COVID	2000
8	Case Closure	Pre-COVID	1490
9	Civil Sidewalks	Post-COVID	38
10	Civil Sidewalks	Pre-COVID	850
11	Courtesy Report	Post-COVID	1733
12	Courtesy Report	Pre-COVID	1003
13	Disorderly Conduct	Post-COVID	10299
14	Disorderly Conduct	Pre-COVID	5653
15	Drug Offense	Post-COVID	15624
16	Drug Offense	Pre-COVID	7902
17	Drug Violation	Post-COVID	190
18	Drug Violation	Pre-COVID	102
19	Embezzlement	Post-COVID	647
20	Embezzlement	Pre-COVID	487
21	Fire Report	Post-COVID	1028
22	Fire Report	Pre-COVID	328

Fig. 4. dbt output table descriptions in Snowflake

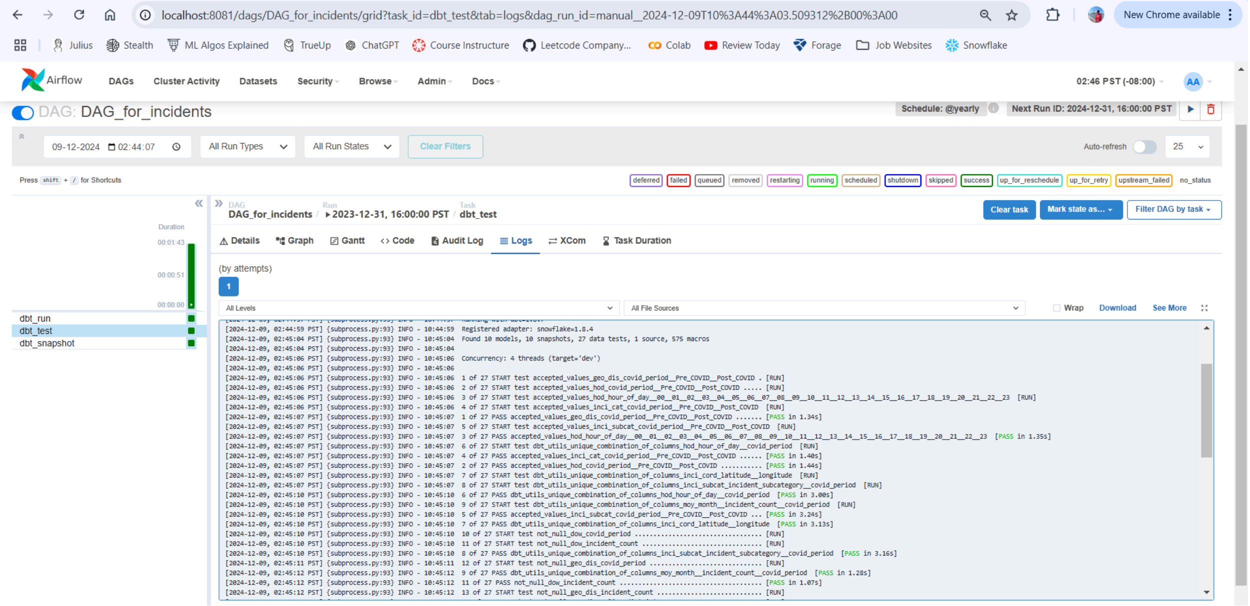


DAG_for_incidents / ▸2023-12-31, 16:00:00 PST / dbt_run

(by attempts)

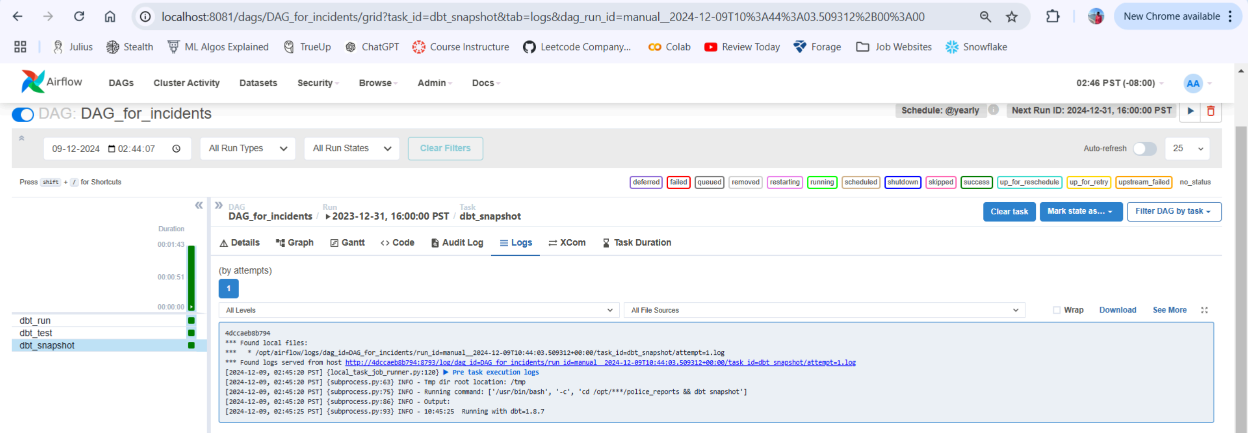
```
[2024-12-09, 02:44:35 PST] {subprocess.py:93} INFO - 10:44:35  Found 10 models, 10 snapshots, 27 data tests, 1 source, 575 macros
[2024-12-09, 02:44:35 PST] {subprocess.py:93} INFO - 10:44:35
[2024-12-09, 02:44:39 PST] {subprocess.py:93} INFO - 10:44:39  Concurrency: 4 threads (target='dev')
[2024-12-09, 02:44:39 PST] {subprocess.py:93} INFO - 10:44:39
[2024-12-09, 02:44:39 PST] {subprocess.py:93} INFO - 10:44:39  1 of 10 START sql incremental model analytics.dow ............................ [RUN]
[2024-12-09, 02:44:39 PST] {subprocess.py:93} INFO - 10:44:39  2 of 10 START sql incremental model analytics.geo_dis ........................ [RUN]
[2024-12-09, 02:44:39 PST] {subprocess.py:93} INFO - 10:44:39  3 of 10 START sql incremental model analytics.hod ............................ [RUN]
[2024-12-09, 02:44:39 PST] {subprocess.py:93} INFO - 10:44:39  4 of 10 START sql incremental model analytics.inci_cat ....................... [RUN]
[2024-12-09, 02:44:42 PST] {subprocess.py:93} INFO - 10:44:42  4 of 10 OK created sql incremental model analytics.inci_cat .................. [SUCCESS 1 in 2.56s]
[2024-12-09, 02:44:42 PST] {subprocess.py:93} INFO - 10:44:42  1 of 10 OK created sql incremental model analytics.dow ....................... [SUCCESS 1 in 2.60s]
[2024-12-09, 02:44:42 PST] {subprocess.py:93} INFO - 10:44:42  5 of 10 START sql incremental model analytics.inci_cord ...................... [RUN]
[2024-12-09, 02:44:42 PST] {subprocess.py:93} INFO - 10:44:42  3 of 10 OK created sql incremental model analytics.hod ....................... [SUCCESS 1 in 2.60s]
[2024-12-09, 02:44:42 PST] {subprocess.py:93} INFO - 10:44:42  2 of 10 OK created sql incremental model analytics.geo_dis ................... [SUCCESS 1 in 2.62s]
[2024-12-09, 02:44:42 PST] {subprocess.py:93} INFO - 10:44:42  6 of 10 START sql incremental model analytics.inci_subcat .................... [RUN]
[2024-12-09, 02:44:42 PST] {subprocess.py:93} INFO - 10:44:42  7 of 10 START sql incremental model analytics.intersection ................... [RUN]
[2024-12-09, 02:44:42 PST] {subprocess.py:93} INFO - 10:44:42  8 of 10 START sql incremental model analytics.moy ............................ [RUN]
[2024-12-09, 02:44:44 PST] {subprocess.py:93} INFO - 10:44:44  6 of 10 OK created sql incremental model analytics.inci_subcat ............... [SUCCESS 1 in 1.78s]
[2024-12-09, 02:44:44 PST] {subprocess.py:93} INFO - 10:44:44  9 of 10 START sql incremental model analytics.neighborhood ................... [RUN]
[2024-12-09, 02:44:44 PST] {subprocess.py:93} INFO - 10:44:44  7 of 10 OK created sql incremental model analytics.intersection .............. [SUCCESS 1 in 1.86s]
[2024-12-09, 02:44:44 PST] {subprocess.py:93} INFO - 10:44:44  10 of 10 START sql incremental model analytics.total ......................... [RUN]
[2024-12-09, 02:44:44 PST] {subprocess.py:93} INFO - 10:44:44  5 of 10 OK created sql incremental model analytics.inci_cord ................. [SUCCESS 1 in 2.00s]
[2024-12-09, 02:44:44 PST] {subprocess.py:93} INFO - 10:44:44  8 of 10 OK created sql incremental model analytics.moy ....................... [SUCCESS 1 in 1.97s]
[2024-12-09, 02:44:45 PST] {subprocess.py:93} INFO - 10:44:45  9 of 10 OK created sql incremental model analytics.neighborhood .............. [SUCCESS 1 in 1.49s]
[2024-12-09, 02:44:46 PST] {subprocess.py:93} INFO - 10:44:46  10 of 10 OK created sql incremental model analytics.total .................... [SUCCESS 1 in 2.44s]
[2024-12-09, 02:44:46 PST] {subprocess.py:93} INFO - 10:44:46
[2024-12-09, 02:44:46 PST] {subprocess.py:93} INFO - 10:44:46  Finished running 10 incremental models in 0 hours 0 minutes and 11.54 seconds (11.54s).
[2024-12-09, 02:44:47 PST] {subprocess.py:93} INFO - 10:44:47
[2024-12-09, 02:44:47 PST] {subprocess.py:93} INFO - 10:44:47  Completed successfully
```

Fig. 5. Airflow logs of running all dbt models



```
[2024-12-09, 02:44:59 PST] {subprocess.py:93} INFO - 10:44:59  Registered adapter: snowflake=1.8.4
[2024-12-09, 02:45:04 PST] {subprocess.py:93} INFO - 10:45:04  Found 10 models, 10 snapshots, 27 data tests, 1 source, 575 macros
[2024-12-09, 02:45:04 PST] {subprocess.py:93} INFO - 10:45:04
[2024-12-09, 02:45:06 PST] {subprocess.py:93} INFO - 10:45:06  Concurrency: 4 threads (target='dev')
[2024-12-09, 02:45:06 PST] {subprocess.py:93} INFO - 10:45:06
[2024-12-09, 02:45:06 PST] {subprocess.py:93} INFO - 10:45:06  1 of 27 START test accepted_values_geo_dis_covid_period__Pre_COVID__Post_COVID . [RUN]
[2024-12-09, 02:45:06 PST] {subprocess.py:93} INFO - 10:45:06  2 of 27 START test accepted_values_hod_covid_period__Pre_COVID__Post_COVID ..... [RUN]
[2024-12-09, 02:45:06 PST] {subprocess.py:93} INFO - 10:45:06  3 of 27 START test accepted_values_hod_hour_of_day__00__01__02__03__04__05__06__07__08__09__10__11__12__13__14__15__16__17__18__19__20__21__22__23  [RUN]
[2024-12-09, 02:45:06 PST] {subprocess.py:93} INFO - 10:45:06  4 of 27 START test accepted_values_inci_cat_covid_period__Pre_COVID__Post_COVID  [RUN]
[2024-12-09, 02:45:07 PST] {subprocess.py:93} INFO - 10:45:07  1 of 27 PASS accepted_values_geo_dis_covid_period__Pre_COVID__Post_COVID ....... [PASS in 1.34s]
[2024-12-09, 02:45:07 PST] {subprocess.py:93} INFO - 10:45:07  5 of 27 START test accepted_values_inci_subcat_covid_period__Pre_COVID__Post_COVID  [RUN]
[2024-12-09, 02:45:07 PST] {subprocess.py:93} INFO - 10:45:07  3 of 27 PASS accepted_values_hod_hour_of_day__00__01__02__03__04__05__06__07__08__09__10__11__12__13__14__15__16__17__18__19__20__21__22__23  [PASS in 1.35s]
[2024-12-09, 02:45:07 PST] {subprocess.py:93} INFO - 10:45:07  6 of 27 START test dbt_utils_unique_combination_of_columns_hod_hour_of_day__covid_period  [RUN]
[2024-12-09, 02:45:07 PST] {subprocess.py:93} INFO - 10:45:07  4 of 27 PASS accepted_values_inci_cat_covid_period__Pre_COVID__Post_COVID ...... [PASS in 1.40s]
[2024-12-09, 02:45:07 PST] {subprocess.py:93} INFO - 10:45:07  2 of 27 PASS accepted_values_hod_covid_period__Pre_COVID__Post_COVID ........... [PASS in 1.44s]
[2024-12-09, 02:45:07 PST] {subprocess.py:93} INFO - 10:45:07  7 of 27 START test dbt_utils_unique_combination_of_columns_inci_cord_latitude__longitude  [RUN]
[2024-12-09, 02:45:07 PST] {subprocess.py:93} INFO - 10:45:07  8 of 27 START test dbt_utils_unique_combination_of_columns_inci_subcat_incident_subcategory__covid_period  [RUN]
[2024-12-09, 02:45:10 PST] {subprocess.py:93} INFO - 10:45:10  6 of 27 PASS dbt_utils_unique_combination_of_columns_hod_hour_of_day__covid_period  [PASS in 3.00s]
[2024-12-09, 02:45:10 PST] {subprocess.py:93} INFO - 10:45:10  9 of 27 START test dbt_utils_unique_combination_of_columns_moy_month__incident_count__covid_period  [RUN]
[2024-12-09, 02:45:10 PST] {subprocess.py:93} INFO - 10:45:10  5 of 27 PASS accepted_values_inci_subcat_covid_period__Pre_COVID__Post_COVID ... [PASS in 3.24s]
[2024-12-09, 02:45:10 PST] {subprocess.py:93} INFO - 10:45:10  7 of 27 PASS dbt_utils_unique_combination_of_columns_inci_cord_latitude__longitude  [PASS in 3.13s]
[2024-12-09, 02:45:10 PST] {subprocess.py:93} INFO - 10:45:10  10 of 27 START test not_null_dow_covid_period ................................. [RUN]
[2024-12-09, 02:45:10 PST] {subprocess.py:93} INFO - 10:45:10  11 of 27 START test not_null_dow_incident_count ................................ [RUN]
[2024-12-09, 02:45:10 PST] {subprocess.py:93} INFO - 10:45:10  8 of 27 PASS dbt_utils_unique_combination_of_columns_inci_subcat_incident_subcategory__covid_period  [PASS in 3.16s]
[2024-12-09, 02:45:11 PST] {subprocess.py:93} INFO - 10:45:11  12 of 27 START test not_null_geo_dis_covid_period ............................. [RUN]
[2024-12-09, 02:45:12 PST] {subprocess.py:93} INFO - 10:45:12  9 of 27 PASS dbt_utils_unique_combination_of_columns_moy_month__incident_count__covid_period  [PASS in 1.28s]
[2024-12-09, 02:45:12 PST] {subprocess.py:93} INFO - 10:45:12  11 of 27 PASS not_null_dow_incident_count ...................................... [PASS in 1.07s]
[2024-12-09, 02:45:12 PST] {subprocess.py:93} INFO - 10:45:12  13 of 27 START test not_null_geo_dis_incident_count ............................ [RUN]
```

Fig. 6. Airflow logs of running all dbt tests

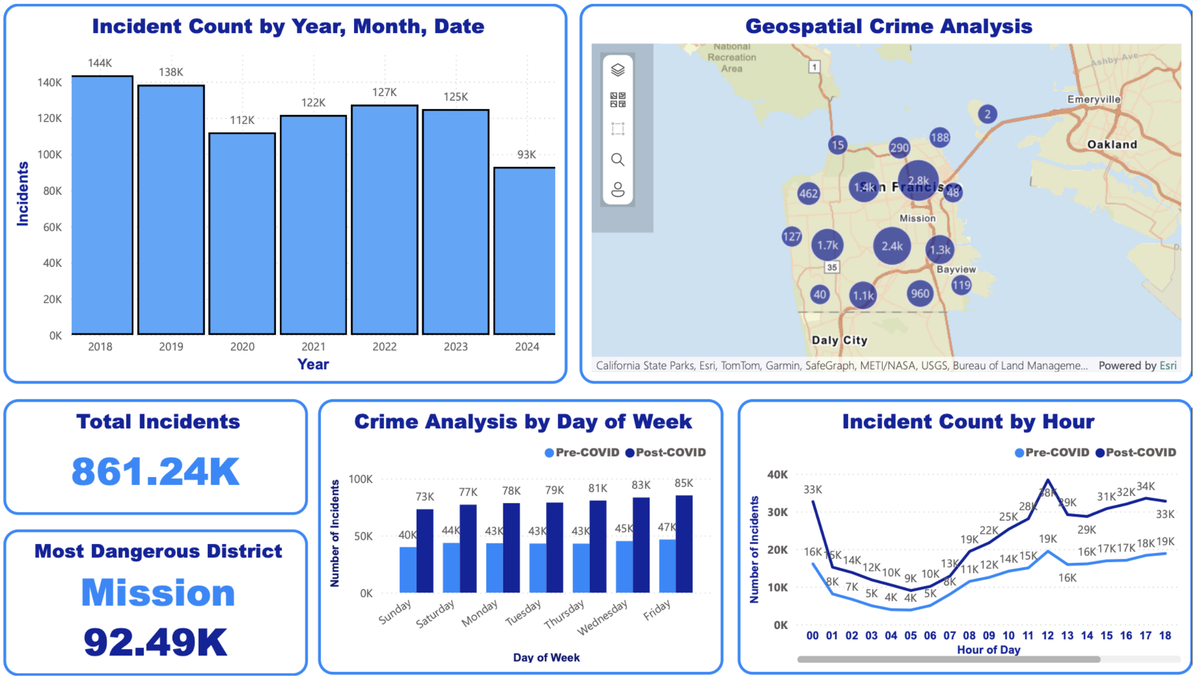


```
4dccaeb8b794
*** Found local files:
***   * /opt/airflow/logs/dag_id=DAG_for_incidents/run_id=manual__2024-12-09T10:44:03.509312+00:00/task_id=dbt_snapshot/attempt=1.log
*** Found logs served from host http://4dccaeb8b794:8793/log/dag_id=DAG_for_incidents/run_id=manual__2024-12-09T10:44:03.509312+00:00/task_id=dbt_snapshot/attempt=1.log
[2024-12-09, 02:45:20 PST] {local_task_job_runner.py:120} ▶ Pre task execution logs
[2024-12-09, 02:45:20 PST] {subprocess.py:63} INFO - Tmp dir root location: /tmp
[2024-12-09, 02:45:20 PST] {subprocess.py:75} INFO - Running command: ['/usr/bin/bash', '-c', 'cd /opt/***/police_reports && dbt snapshot']
[2024-12-09, 02:45:20 PST] {subprocess.py:86} INFO - Output:
[2024-12-09, 02:45:25 PST] {subprocess.py:93} INFO - 10:45:25  Running with dbt=1.8.7
```

Fig. 7. Airflow logs of running all dbt snapshots

C. *Creating Visualizations in PowerBI*

The transformed data was loaded into PowerBI to design an engaging and interactive dashboard (Fig. 8). The display visualized key crime statistics, like incidence amounts based on time in the year, localities, and category types. By implementing machine learning techniques to advance the visualization, geographic clustered heat maps were created, along with time series graphs that compared the difference in incidence pre- and post-COVID. Breaking down the dashboard, Figure 9 shows a bar chart presenting a trend of reported incidents based on years, though the scale can be adjusted to show daily and monthly trends, as well as those based on day of the week. The map presented in Figure 10 illustrates the spatial distribution of incidents, referencing areas with varying levels of activity with total incident counts in accordance with the temporal measurement in the previous bar chart. The two metrics in the bottom left corner of the dashboard present the total number of incidents since 2018 and the most dangerous San Francisco district, followed by the number of incidents that took place there. The grouped bar chart in Figure 11. shows the comparison of incident counts across the days of the week, along with how the amount of cases have changed after 2020. Finally, the line chart in the bottom right corner (Fig. 12) also compares the difference in incident counts before and after COVID, but does so by hour of day. These insights serve to empower city officials in prioritizing resource allocation, identify areas in need of intervention, and formulate strategies for long-term crime reduction.



Total Incidents

861.24K

Most Dangerous District

Mission

92.49K

Fig. 8. PowerBI dashboard depicting San Francisco Crime Analysis

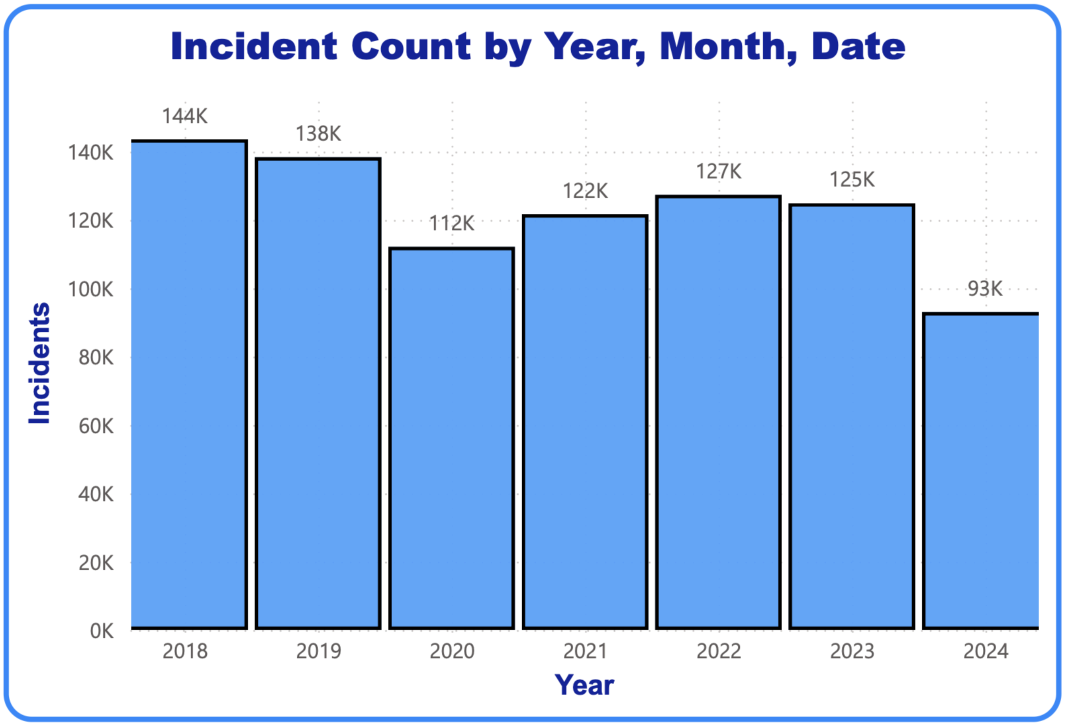


Fig. 9. Incident Count by Year, Month, Date

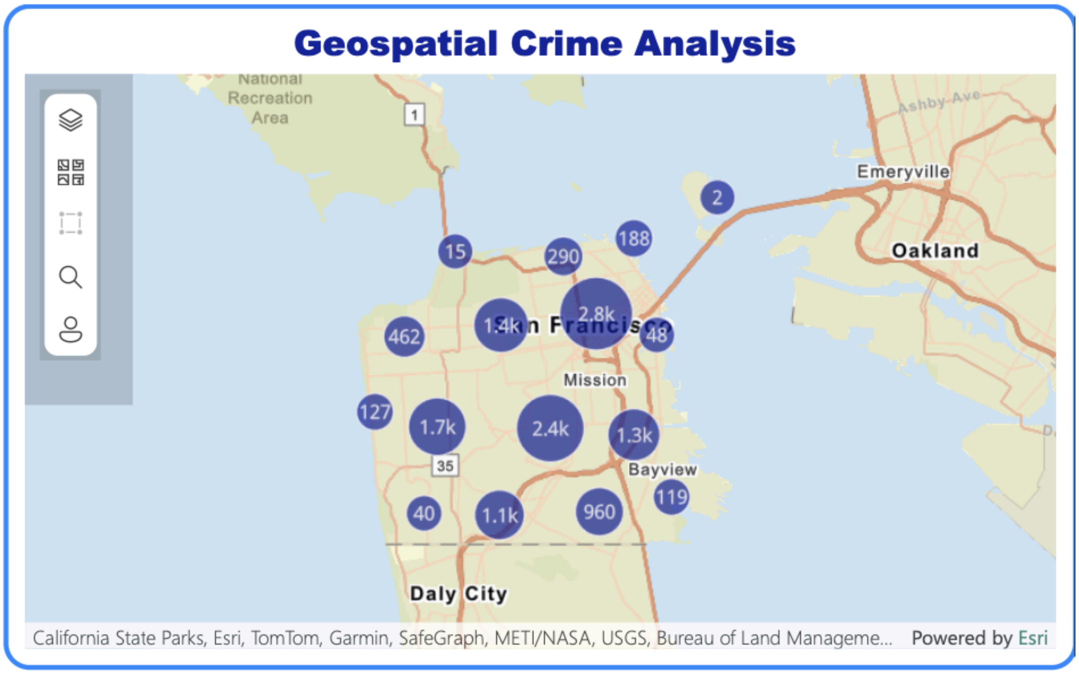


Fig. 10. Geospatial Crime Analysis showing crime hotspots

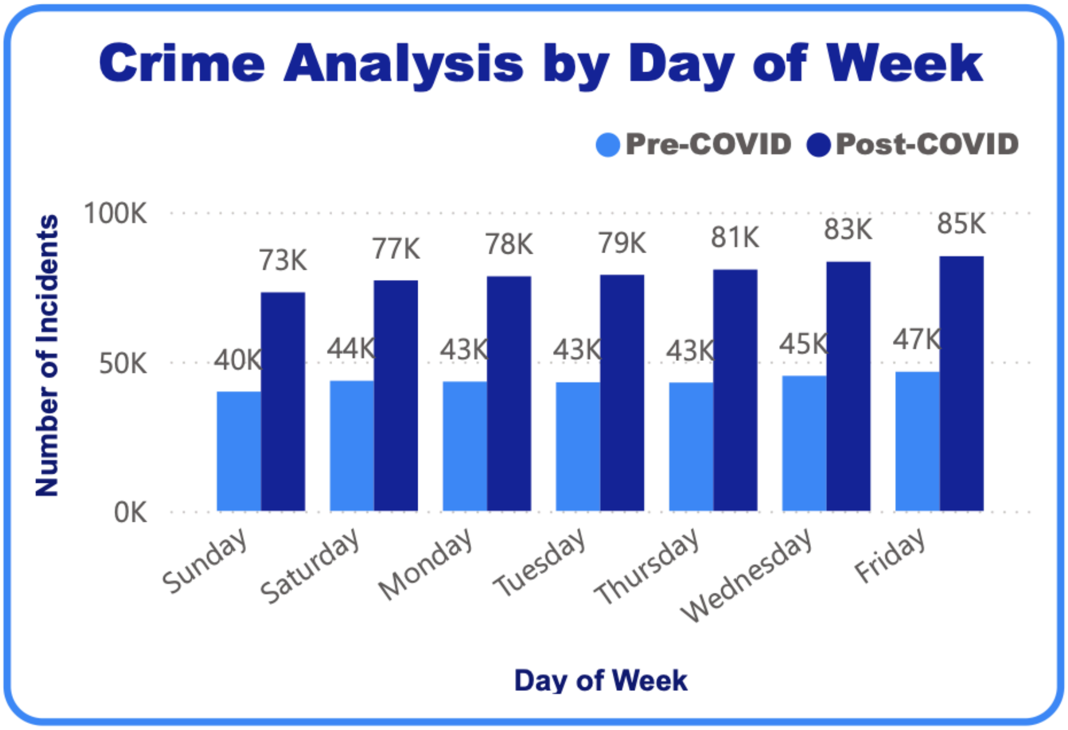


Fig. 11. Crime Analysis by Day of Week (pre- and post-COVID)

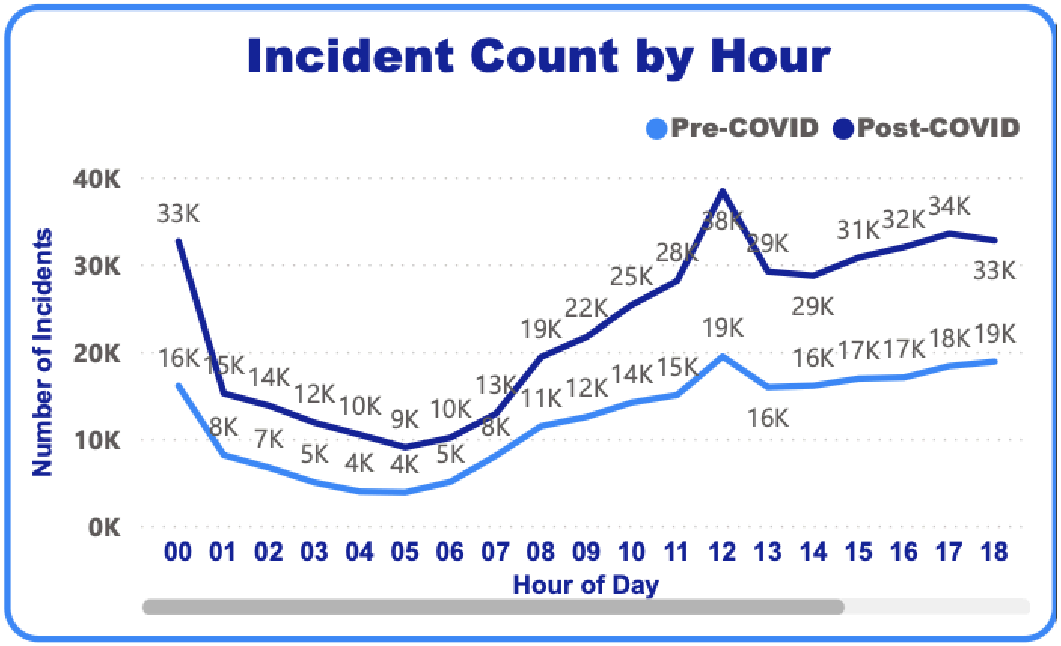
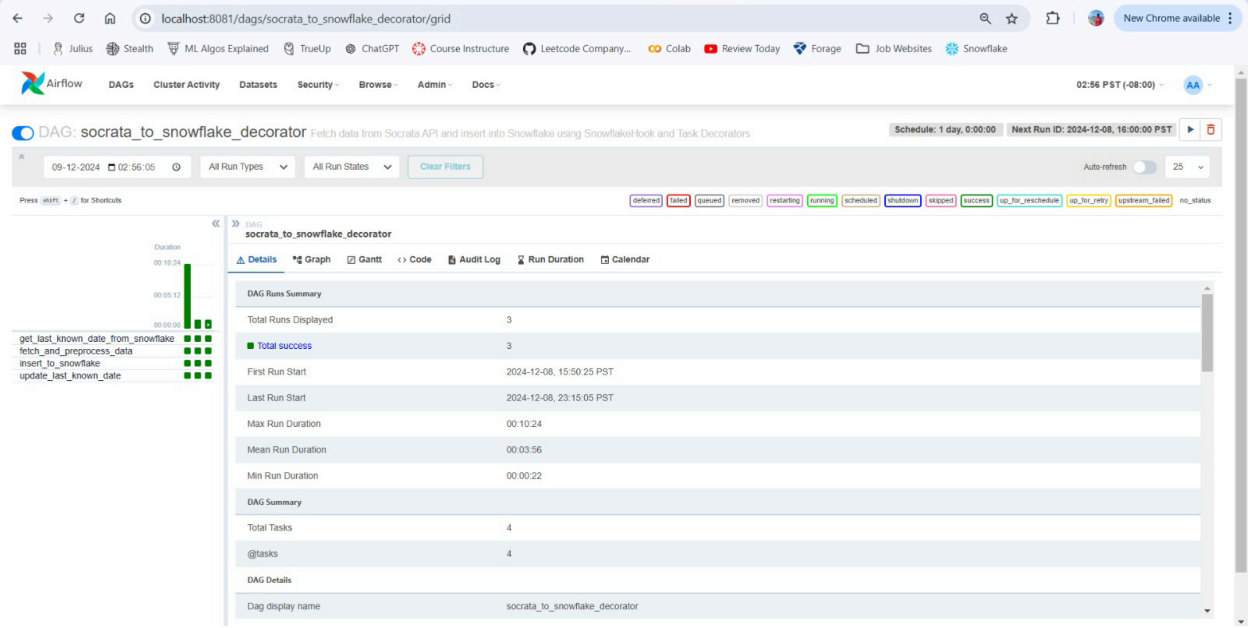


Fig. 12. Incident Count by Hour (pre- and post-COVID)

D. Integration of Live Governmental Data from Socrata (see Python code)

To complement the collection and representation of the historical dataset, new API integration was added from Socrata, an open data platform that allows users access to government data. This enhancement allowed the Snowflake table to maintain an updated repository of crime statistics. Figure 13 depicts how the Airflow DAG was modified to schedule and run weekly, ensuring that the system maintains current incident reports. Unlike the one-and-done execution for historical data, this DAG implements dynamic scheduling to automate the retrieval and ingestion process continuously.



DAG: socrata_to_snowflake_decorator Fetch data from Socrata API and insert into Snowflake using SnowflakeHook and Task Decorators

Schedule: 1 day, 0:00:00 Next Run ID: 2024-12-08, 16:00:00 PST

09-12-2024 02:56:05

get_last_known_date_from_snowflake
fetch_and_preprocess_data
insert_to_snowflake
update_last_known_date

DAG
socrata_to_snowflake_decorator

DAG Runs Summary	
Total Runs Displayed	3
🟩 Total success	3
First Run Start	2024-12-08, 15:50:25 PST
Last Run Start	2024-12-08, 23:15:05 PST
Max Run Duration	00:10:24
Mean Run Duration	00:03:56
Min Run Duration	00:00:22
DAG Summary	
Total Tasks	4
@tasks	4
DAG Details	
Dag display name	socrata_to_snowflake_decorator

Fig. 13. Airflow logs of loading API data to Snowflake

E. *Optimizing Incremental Updates*

Instead of using the traditional MERGE statement for handling duplicates, a more effective strategy was used for incremental updates. The MERGE function is reliable in most cases but here it consumes a significant amount of computing resources as it individually processes each row of the dataset. To avoid this limitation, the datetime library was employed to track the maximum timestamp in the current table. Python's timedelta() function allowed the DAG to calculate the next starting point for ingestion by adding one minute to the maximum timestamp. Each DAG execution updated the last_known_time variable, ensuring that only the new records were fetched and integrated. This approach significantly reduced resource usage and improved the pipeline's performance. To ensure that the DAG checks entries properly, error handling has been implemented using the try catch method where if no new records are found, task implementation will automatically complete and this has been ensured for all the following tasks as well.

F. *Automating Data Consistency and Accuracy*

The updated DAG ensured consistency and data accuracy through thorough testing before deployment. Incremental updates were validated against the raw data to confirm the accuracy of the last_known_time mechanism. The utility of Airfow's task dependency management further confirmed that updates occurred in the correct sequence, restricting any duplication or data loss. This automated solution enabled the pipeline to maintain a balance between efficient and reliable transformation while seamlessly handling large and dynamic datasets.

V. Results

The dashboard offers a multifaceted presentation of crime data through a variety of useful visualizations. The Incident Count by Year, Month, Date bar chart illustrates a yearly trend in reported incidents, peaking in 2019 at 144K and declining to an expected average of 125K in 2024 (recognizing that the 93K does not include the total for December). The Geospatial Crime Analysis clustered heatmap highlights the geographic distribution of crime, with hotspots appearing in Financial District, Mission, and the Tenderloin., visualized with proportional circles and yearly crime amounts of 2.8K and 2.4K. The overall metrics in the bottom left corner summarize the number of incidents over the past six years, totalling at 861.24K, while the most dangerous district identified Mission with the highest number of incidents: 92.49K. The Crime Analysis by Day of Week bar chart compares the incident counts across different days, distinguishing between pre-COVID and post-COVID patterns, with the highest activity observed towards the end of the work week (Wednesday through Friday). Lastly, the Incident Counts by Hour line chart examines hourly trends, showing the bulk of incidents during the afternoon, peaking at 12 PM, following a gradual rise in the evening, from 4 – 6 PM, with another peak around midnight.

VI. Analysis

The development of the San Francisco Crime Analysis project presented several challenges that required innovative solutions to ensure efficiency and scalability. One significant hurdle was the API limitations, as the Socrata API only allowed retrieval of 50,000 records per query. To overcome this, the complete historical dataset was downloaded in CSV format for bulk processing, while the API was reserved for live incremental updates, adhering to its record limits. Another challenge involved the inefficiency of row-by-row `INSERT INTO` statements for large datasets, which resulted in slow processing times. This was resolved by adopting a staging approach, where historical data was temporarily staged in Snowflake before being ingested in bulk using `COPY INTO` commands, significantly reducing runtime. For live data updates, the initial use of `MERGE` statements proved overly complex and computationally expensive. Instead, a more streamlined last-known-date approach was implemented, fetching only new records by querying the API based on the latest `incident_datetime` in Snowflake. Additionally, data preprocessing revealed critical fields with null or missing values, which were systematically handled by dropping invalid rows and validating the dataset before ingestion. Another challenge was ensuring historical data integrity while enabling time-series analysis. This was addressed through dbt snapshots, which captured and preserved changes in historical data. These strategies collectively ensured the pipeline's robustness, enabling seamless integration of both historical and live data while maintaining accuracy and efficiency. The project highlights the importance of balancing performance optimization with effective data handling for real-world analytical pipelines.

VII. Conclusion

In conclusion, this project was able to demonstrate the potential of integrated data warehousing and analytics in addressing urban safety challenges. Data-driven approaches have already shown promising results in the city's crime management. For example, the SFPD used predictive policing technologies and targeted analytics to direct a 15% reduction in property crimes between 2021 and 2023, especially in the high-risk neighborhoods of the Tenderloin and Mission District. Through the implementation of data-informed strategies, law enforcement have been able to deploy resources more purposefully, learning to proactive crime prevention. By combining historical and real-time datasets, the project uncovered genuine insights into the common incidental patterns. Future work can focus on predictive modeling to anticipate high-risk periods and integrating socio-economic data for a hologist understanding of crime drivers. The results will inevitably inform policymakers in crafting helpful and effective solutions based on defined data to build a safer San Francisco.

VIII. References

[1] AzharGhafoor, "Forecasting the Trends and Patterns of Crime in San Francisco using Machine Learning Model," ResearchGate, https://github.com/AzharGhafoor/PDF_XSS_PAYLOADS (accessed Dec. 9, 2024).

[2] I. Pradhan, "Exploratory Data Analysis and crime prediction in San ...," Exploratory Data Analysis And Crime Prediction In San Francisco, https://scholarworks.sjsu.edu/cgi/viewcontent.cgi?referer=&httpsredir=1&article=1633&context=etd_projects (accessed Dec. 9, 2024).

[3] T. Chen, "Spatio-temporal stratified associations between urban human activities and crime patterns: a case study in San Francisco around the COVID-19 stay-at-home mandate," ResearchGate, https://www.researchgate.net/publication/361117742_Spatio-temporal_stratified_associations_between_urban_human_activities_and_crime_patterns_a_case_study_in_San_Francisco_around_the_COVID-19_stay-at-home_mandate (accessed Dec. 9, 2024)..

[4] G. Spadon et al., "Complex network tools to understand the behavior of criminality in urban areas," arXiv.org, https://arxiv.org/abs/1612.06115 (accessed Dec. 9, 2024).

[5] S. Shirota and A. E. Gelfand, "Space and circular time log Gaussian Cox processes with application to crime event data," arXiv.org, https://arxiv.org/abs/1611.08719 (accessed Dec. 9, 2024).