Course project - Big Data Concepts

Dataset: Chicago crimes dataset 2001 to present
https://data.cityofchicago.org/Public-Safety/Crimes-2001-to-Present/ijzp-q8t2

Introduction

The motive for this hands-on project is to utilize the tools and concepts learned from the coursework. I have implemented a data pipeline that will start with downloading the data from its source. Once downloaded, the data has to be transformed in order to create a standardized format that can be used to visualize it. This transformation stage can include removing unnecessary columns and converting the date-time format. At the end of this process, we are left with a dataset ready for visualization. The environment I have used for implementing this project is the Google cloud platform. In this project, I utilized the following resources: establishing the data pipeline, developing a storage model using GCP cloud storage buckets, and utilizing a parallel programming framework to modify or clean the data using Dataproc. The transformed data is stored in BigQuery to get insights from the dataset and visualize the data with Tableau.

This dataset includes reported crimes (with the exception of murders, which provide data for each victim) that happened in the City of Chicago from 2001 to the present The information is derived from the Chicago Police Department's CLEAR (Citizen Law Enforcement Analysis and Reporting) system. This dataset is of size 2 GB and consists of 7 million data points.

BACKGROUND

The problem that is addressed here is to investigate and offer a user an efficient and scalable data exploration based on a very big dataset (Chicago crimes). To address this problem I have created a pipeline using the Google Cloud Platform. I have used multiple concepts that I learned during this course including Data pipeline and lifecycle, GCP, data visualization, Ingestion and storage, Data modeling, and Distributed Computing and File Systems. The first choice was to employ Hadoop, a Cloud Platform, and a serverless data warehouse all of these provide scalability and efficiency. Google Cloud Platform was the finest option for me. The next challenge was to find a panda-free substitute. PySpark was utilized to tackle this problem of cleaning a sparse large dataset like mine since it can make use of Hadoop's parallel computing capabilities by utilizing the distributed file system. Then Using BigQuery to draw the insights from the transformed dataset and visualize them using Tableau.

The dataset I have used here is the Chicago crimes dataset and I find this dataset interesting because Chicago is the financial capital of the United States and also it is one of the most crimes reported states. To address this problem I thought to draw insights into how the crime trend varies over the years. How are different category crimes vary? What locations are the hotspots for the crimes? Etc. By deriving these insights we can know which places are safe and at what times. The outcome of this project is an ETL pipeline with learning outcomes like GCP, Dataproc, BigQuery, and Tableau.

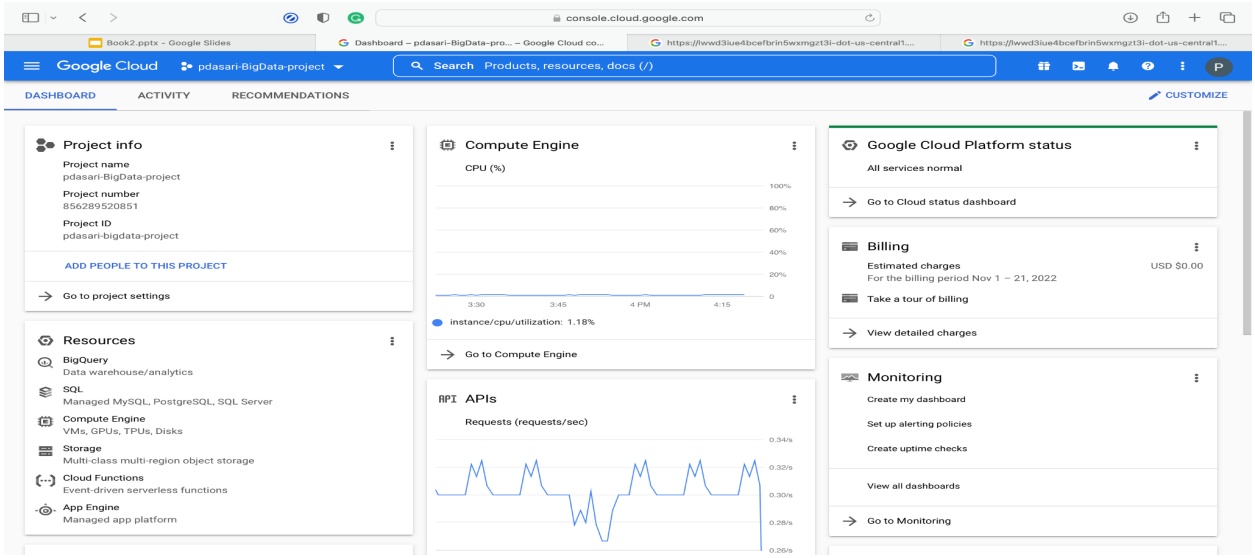
Methodologies

The technological setup I have used in this project is:

- Created the GCP storage bucket
- Dataproc for running an instance
- Jupyter Notebooks and PySpark to transform the data
- Creating the BigQuery Dataset

Steps:

1. Firstly I have created a project pdasari-BigData-project

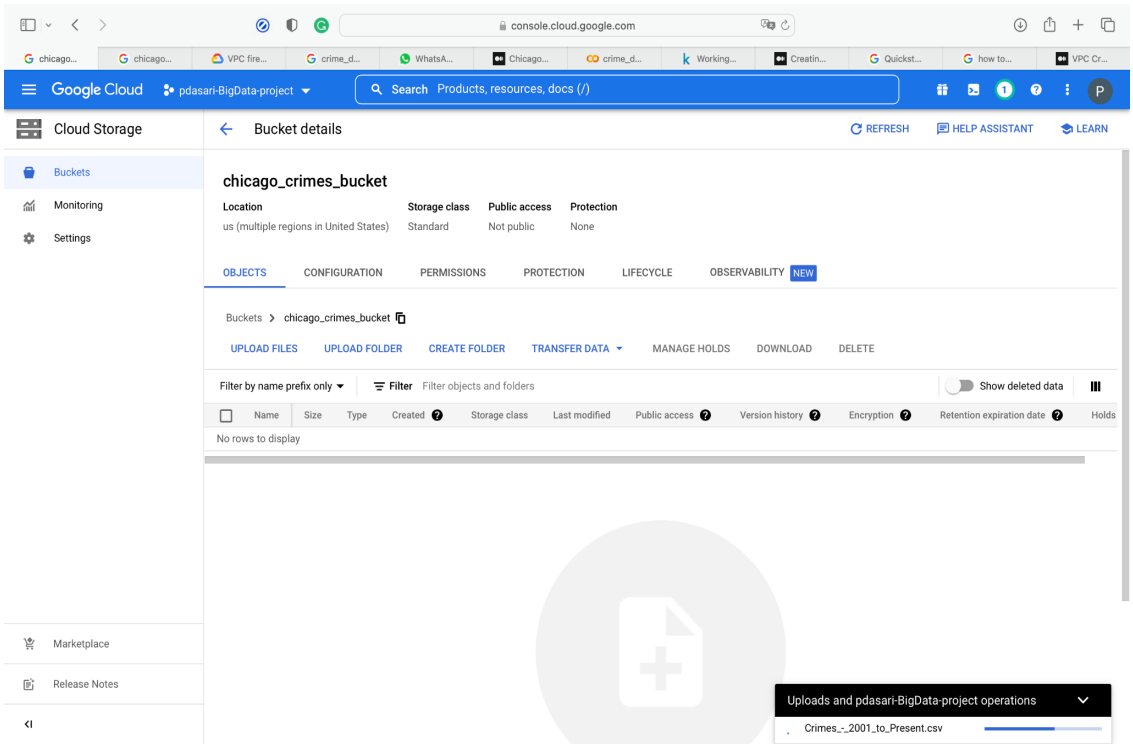


2. Then I created a cloud storage bucket to store the downloaded dataset and for Hadoop-distributed file storage, Jupyter notebooks.

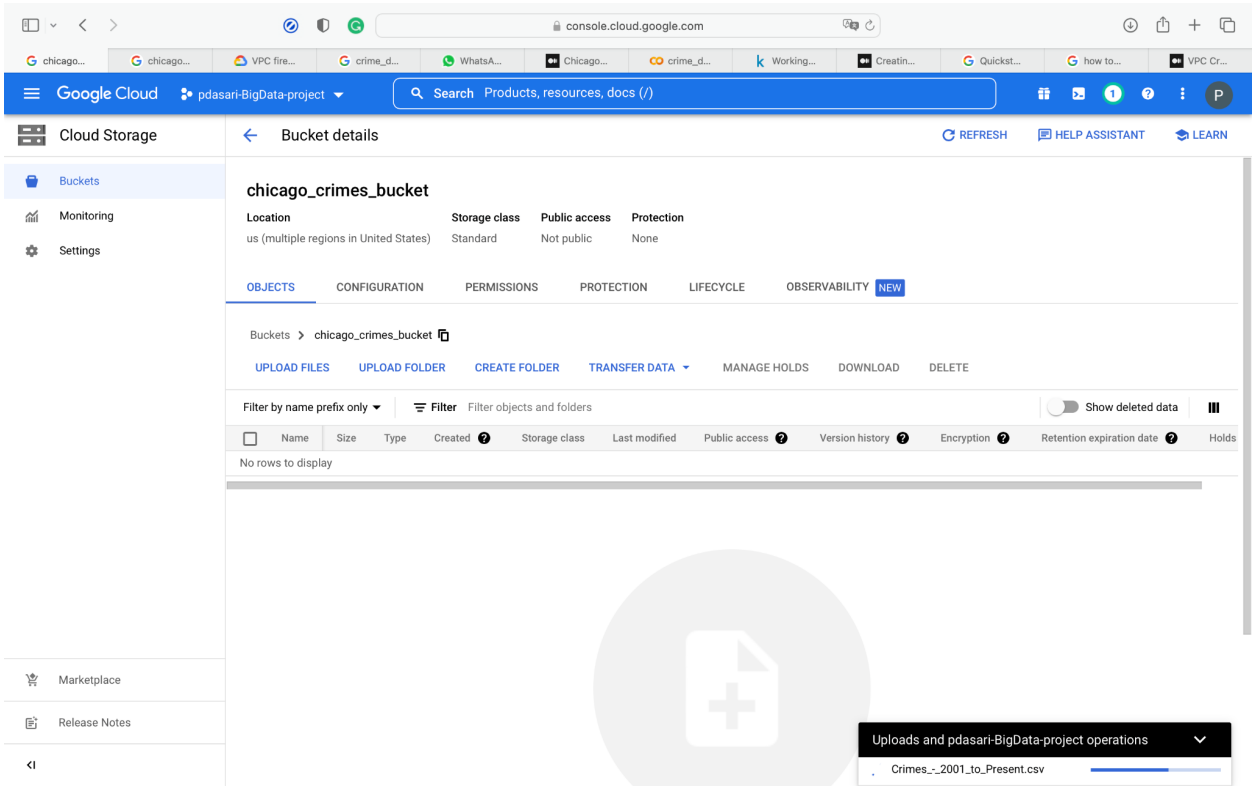
   By following these steps

   Cloud storage → Browser → Create Bucket

   Then I uploaded my data file into the bucket.



3. Then I set up a VPC (virtual private cloud) connection to provide the networking functionality to the compute engines created by the Dataproc

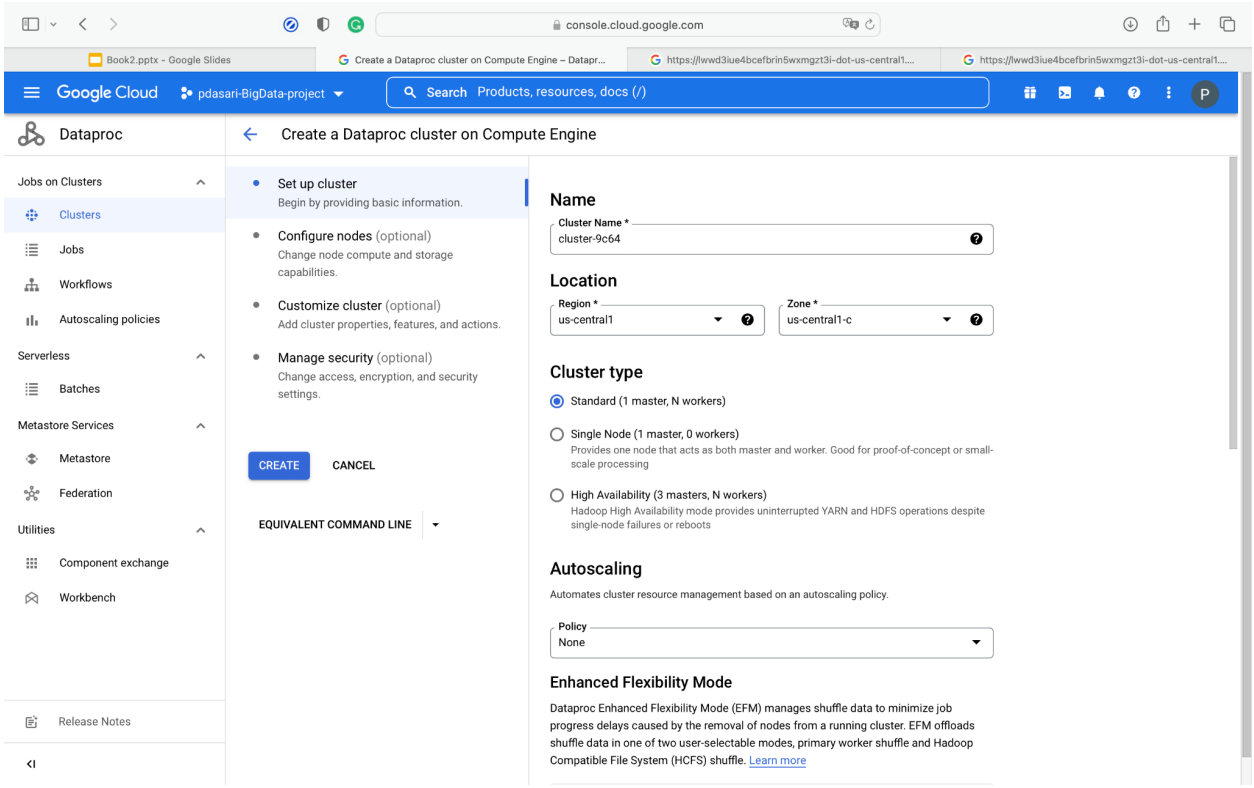


4. Using the Google cloud platform's Big data capability to create clusters

   Data proc: Clusters -> Dataproc Make a cluster

   Setting up the cluster:

   - Cluster name: pyspark-cluster
   - Cluster Type: Standard (1 master, N workers)
   - Image type: 1.5 (Debian 10, Hadoop 2.10, Spark 2.4)
   - Enable component gateway
   - Enable Anaconda and Jupyter notebook in optional components



Configuring nodes:

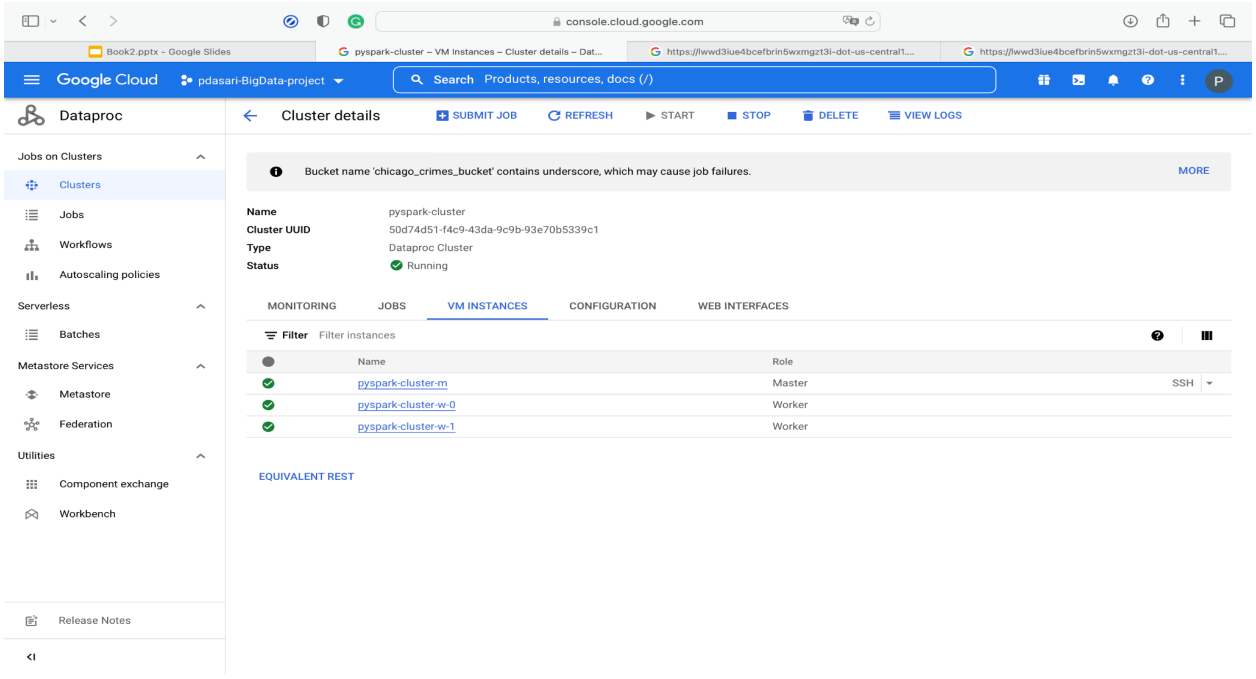
Except for the Primary disk size, leave the default settings alone. Change the size of the primary disk to suit your needs. I upgraded to 50 GB. Master Node: 4vCPU, 15 GB RAM, and 50 GB storage. 4vCPU 15 GB RAM 2 worker nodes with 50GB storage

Customize cluster:

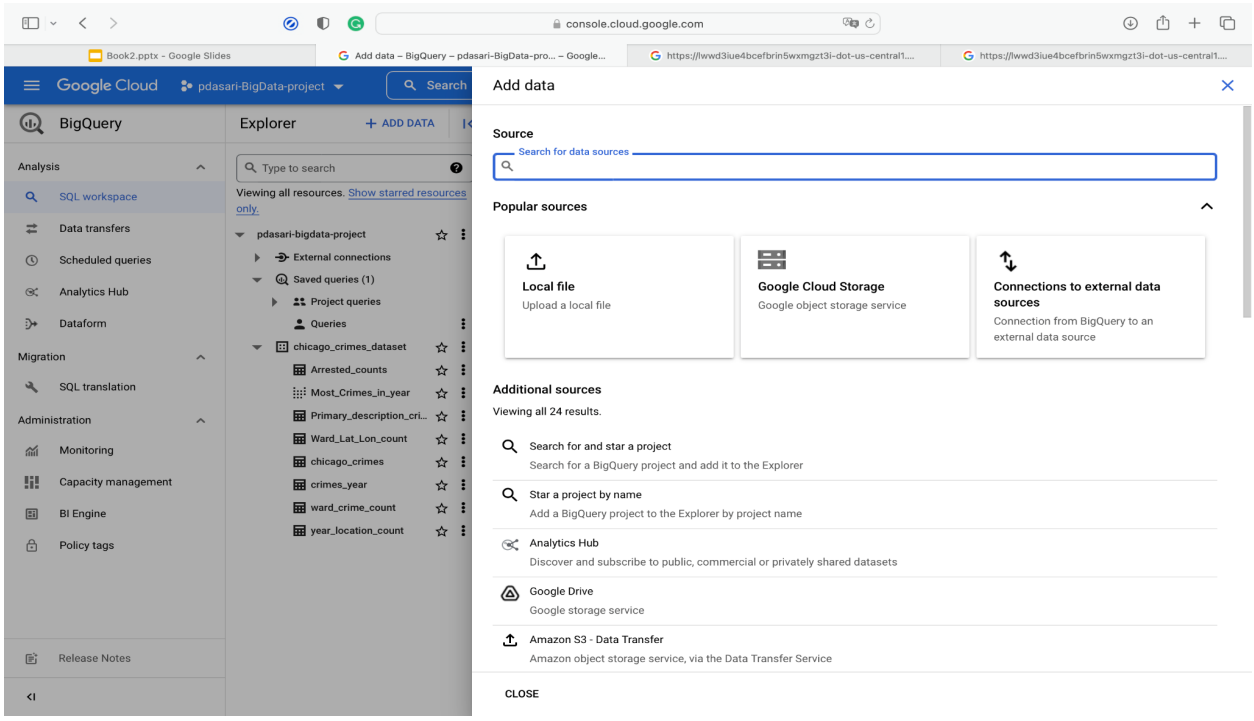
Continue using the default settings. Choose the bucket that was first established as your Cloud storage staging bucket. This bucket can hold code (for example, pyspark code

Manage Security:

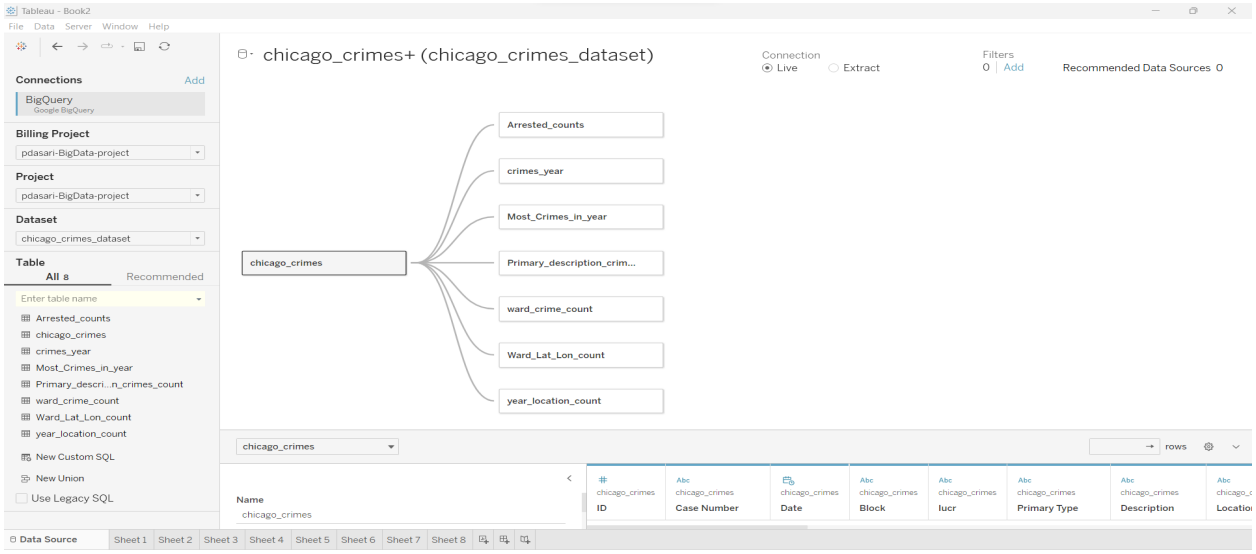
Allow API access to all Google cloud services within the same project in



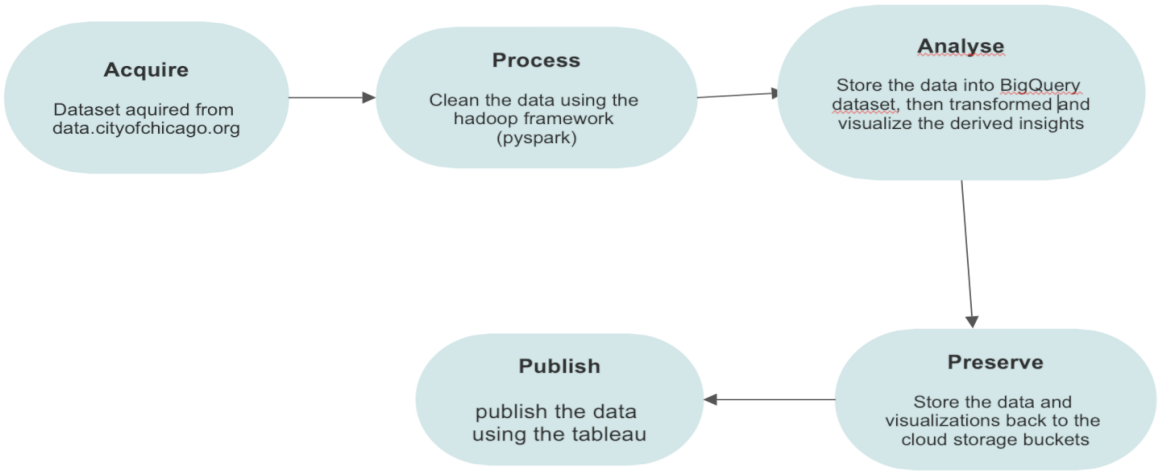
5. Adding the dataset stored in the cloud storage to the BigQuery dataset.



6. Connecting the big query data to Tableau to visualize the data. This will help to stream the results and visualize them.

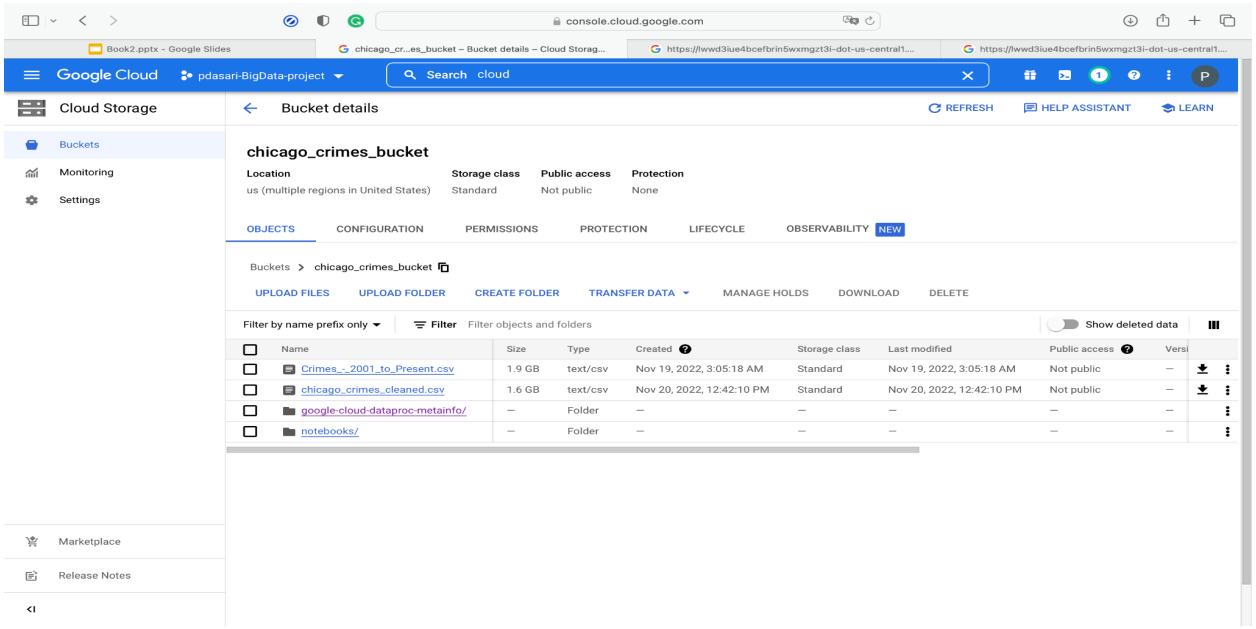


The pipeline I have used in this project is referenced from the USGS data lifecycle model (https://www.usgs.gov/data-management/data-lifecycle)

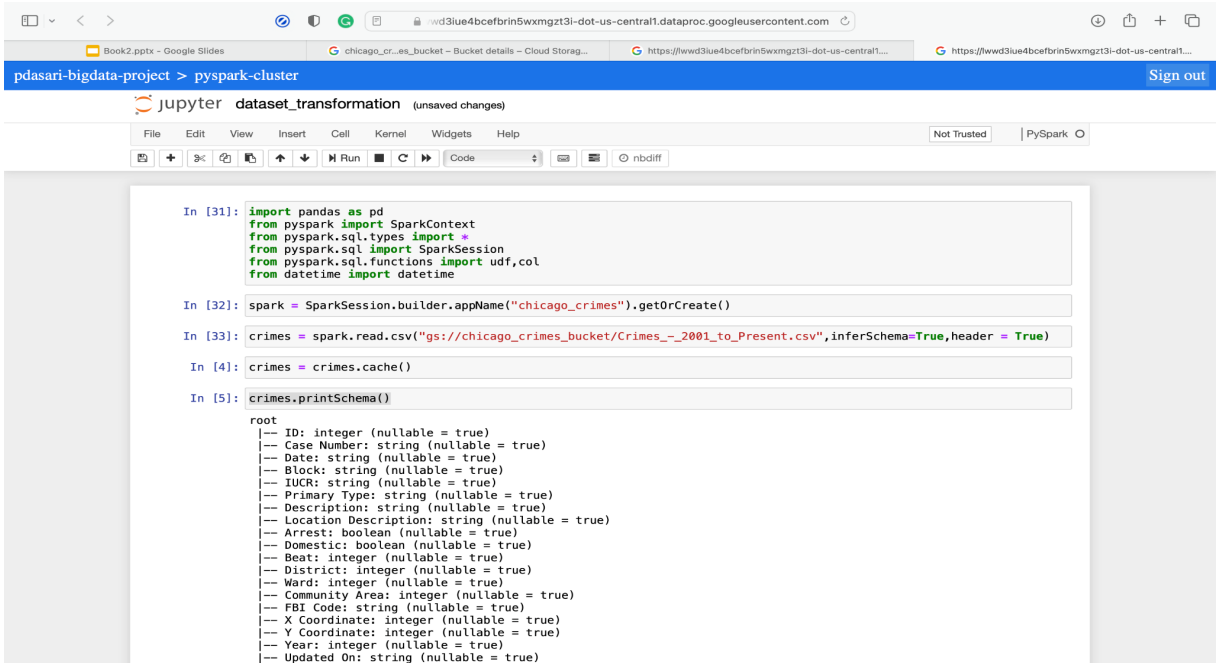


Results

Uploaded the downloaded dataset to the Google cloud storage bucket.

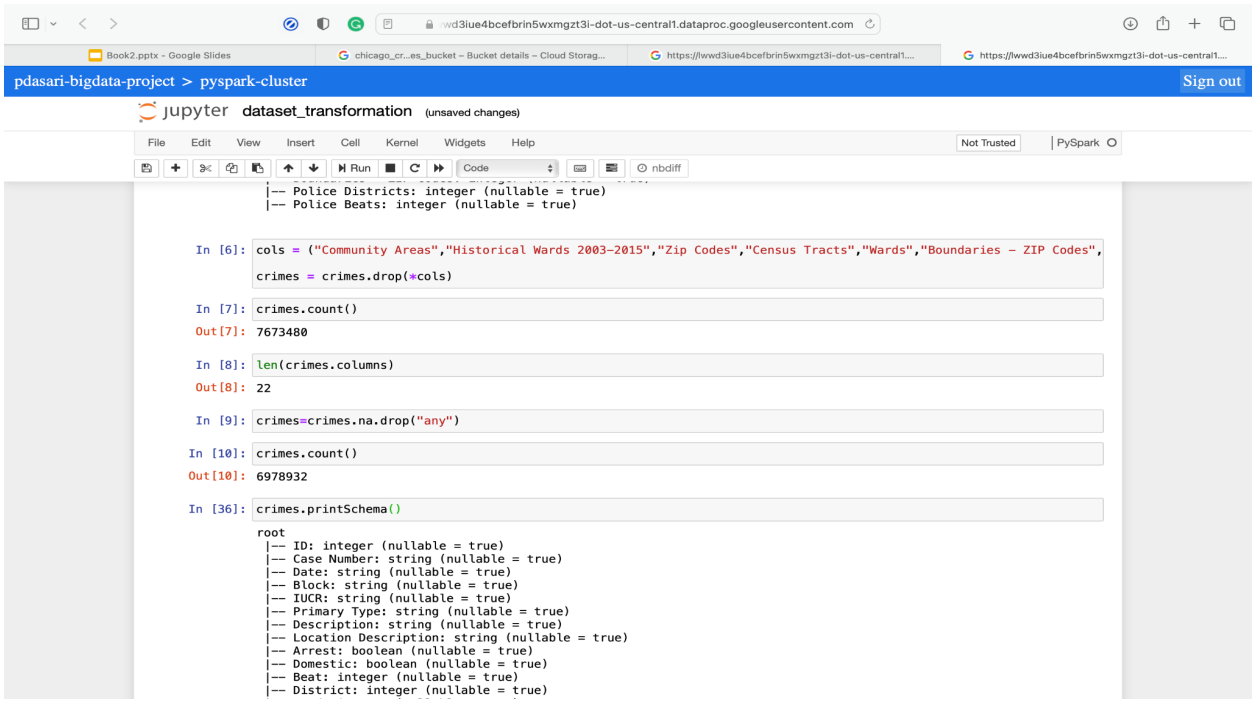


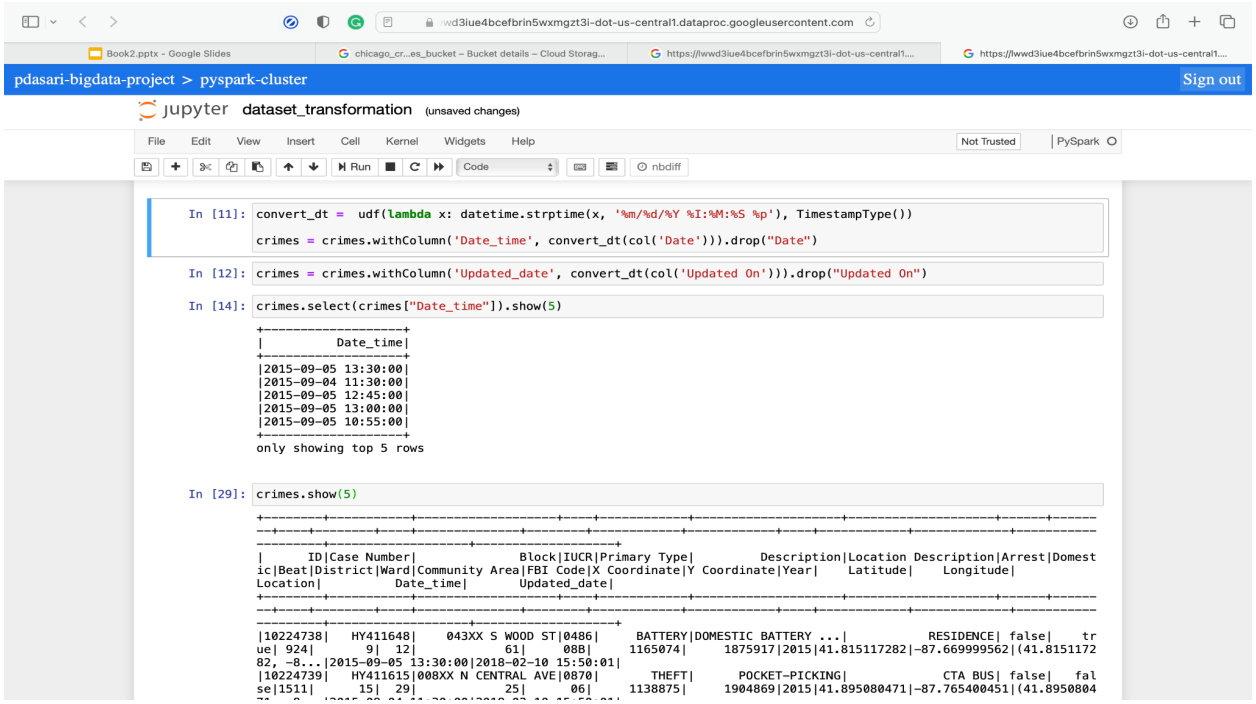
Then I cleaned the data, in which I utilized pyspark since it supports parallel computation. This dataset is required since it is large, and using pandas will make the procedure sluggish. I stored the generated dataset back into the storage bucket after cleaning and processing it.



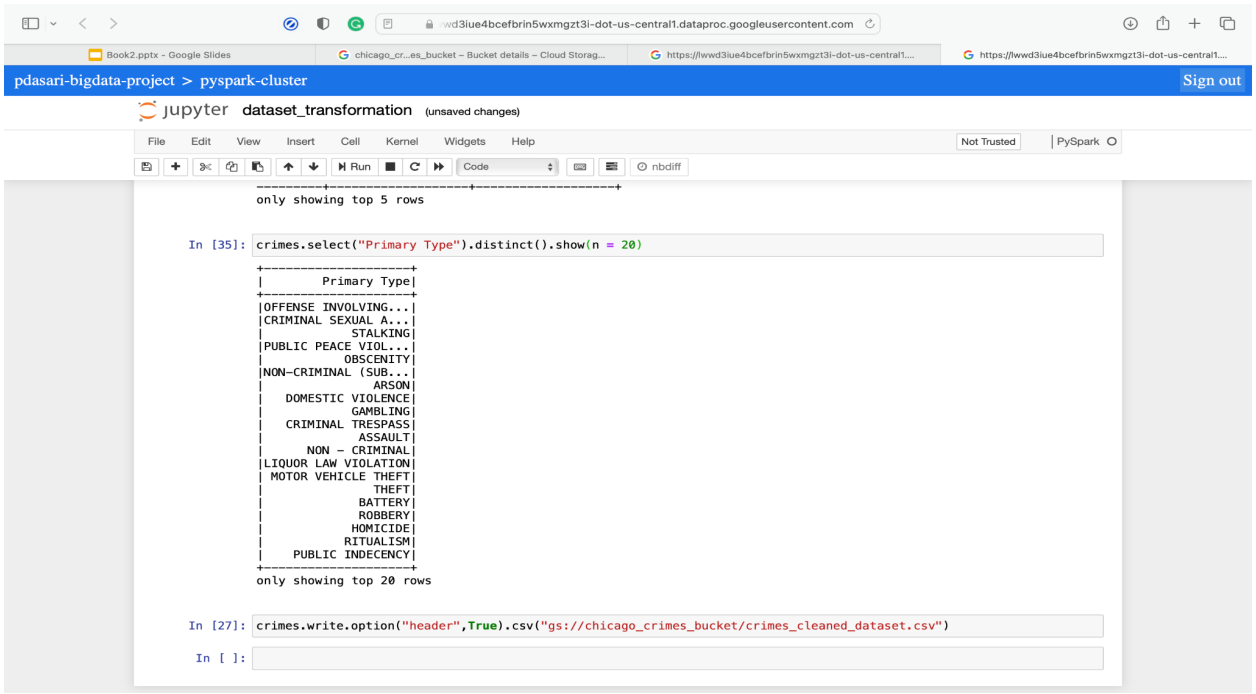
Created a spark session to create a spark data frame table and execute the SQL on the data frame. The spark will infer the schema from the CSV file.

Then I checked for the shape of the data frame and dropped all the columns which more null values that aren't useful for my analysis. And then dropped the rows that contain any null values.



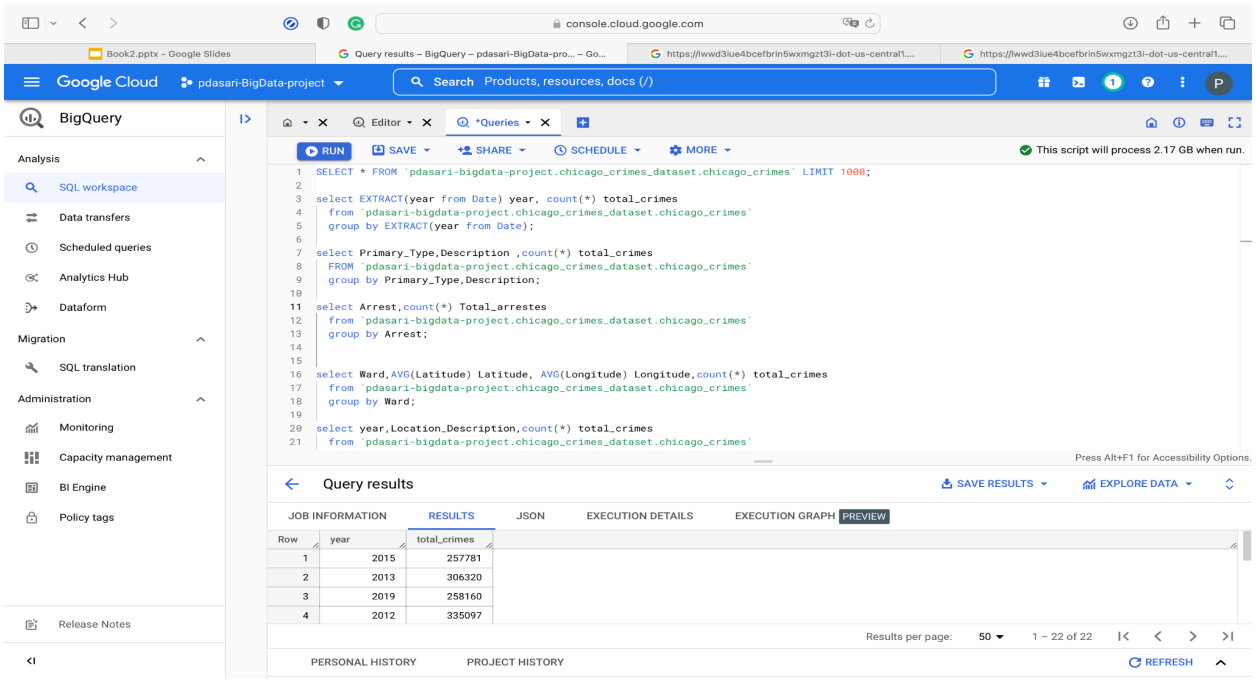


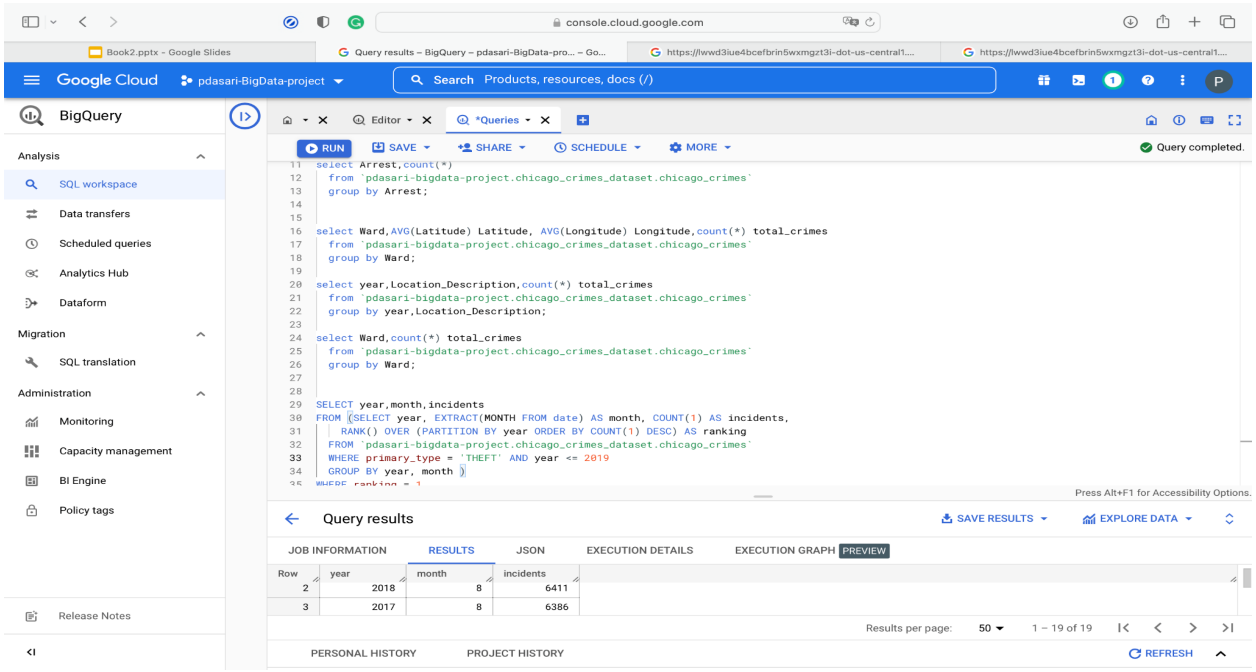
The Date column is formatted as a string. I utilized the user-defined functions to convert it to timestamp format (UDF). With withcolumn function, I added a new column, and with a drop, I deleted one or more columns.



Then I saved the spark data frame back to the cloud storage in CSV format.

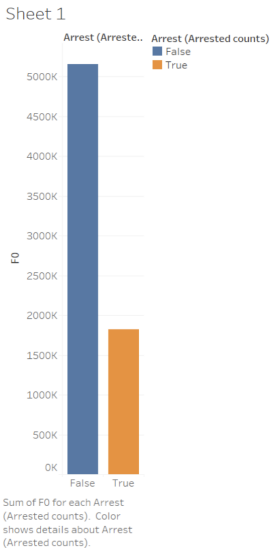
Now I have loaded the clean dataset to BigQuery to transform the dataset and prepare a table with the required data and attributes. I wrote some SQL queries to perform this step.

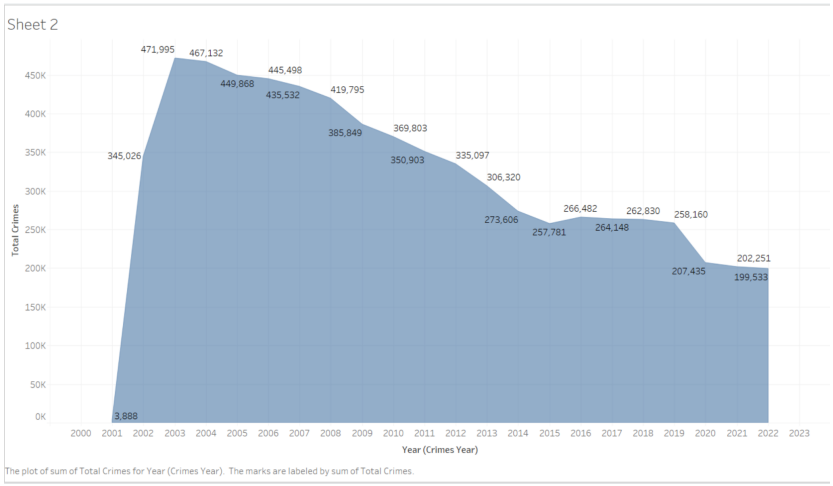


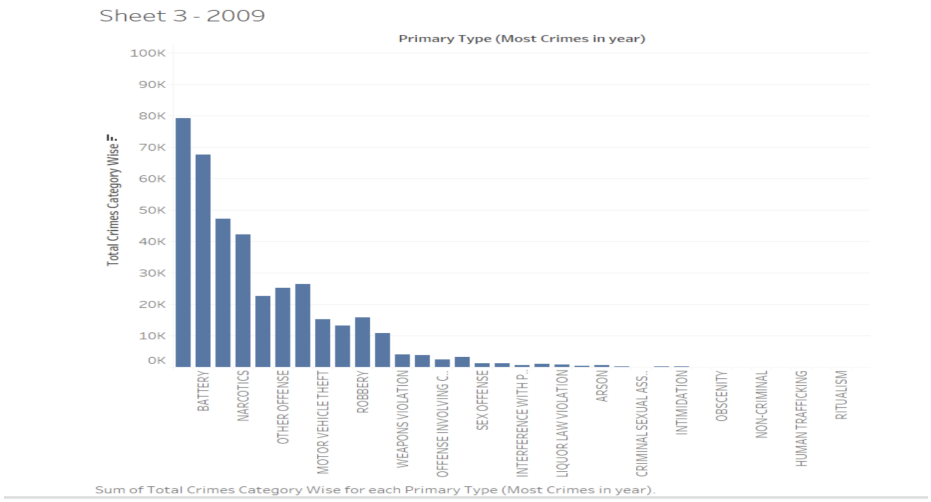


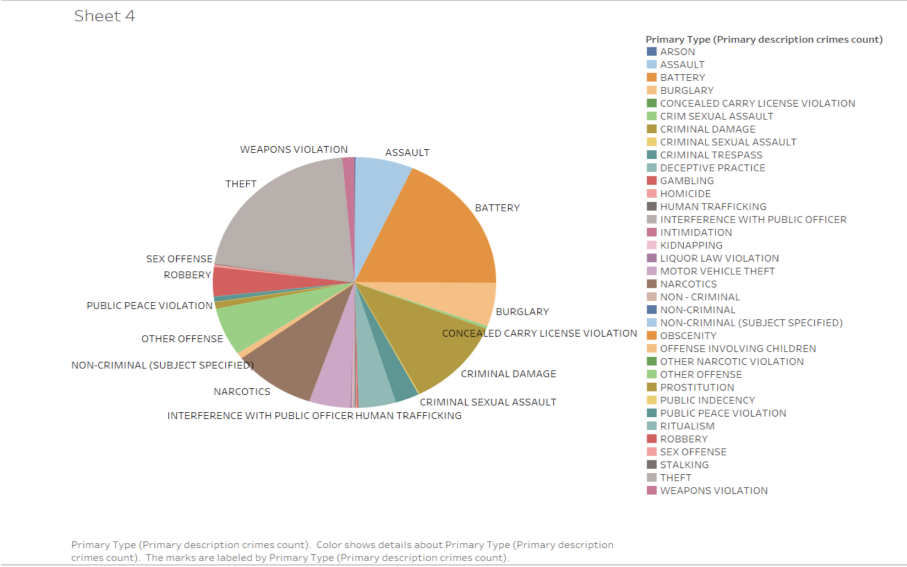
Next, I created a dataset table to store the query results. And visualized them using the tableau as it is insightful and informative to view data. Below are the

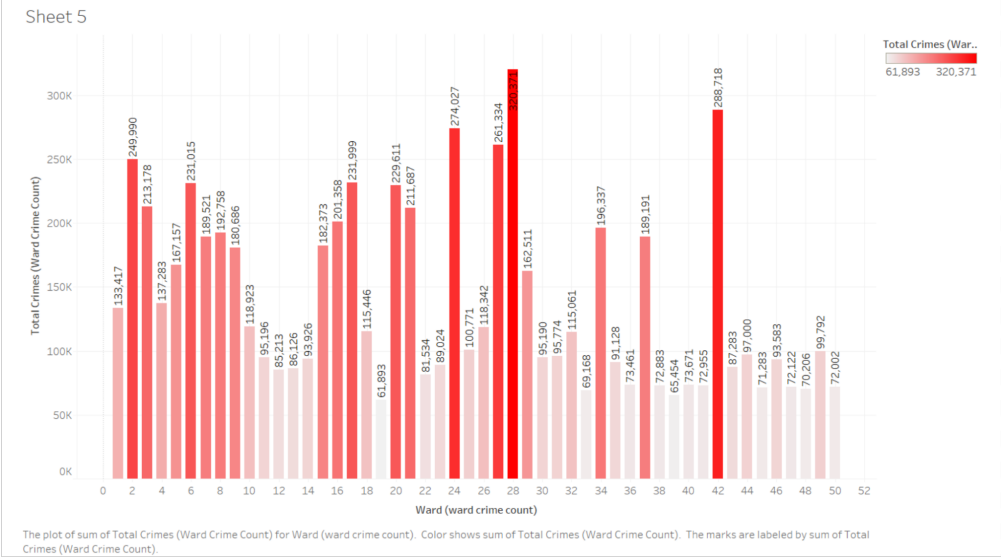
different kinds of visuals I have created as per data requirements.











Then I published the data into the tableau server and made it available as embedded excel.

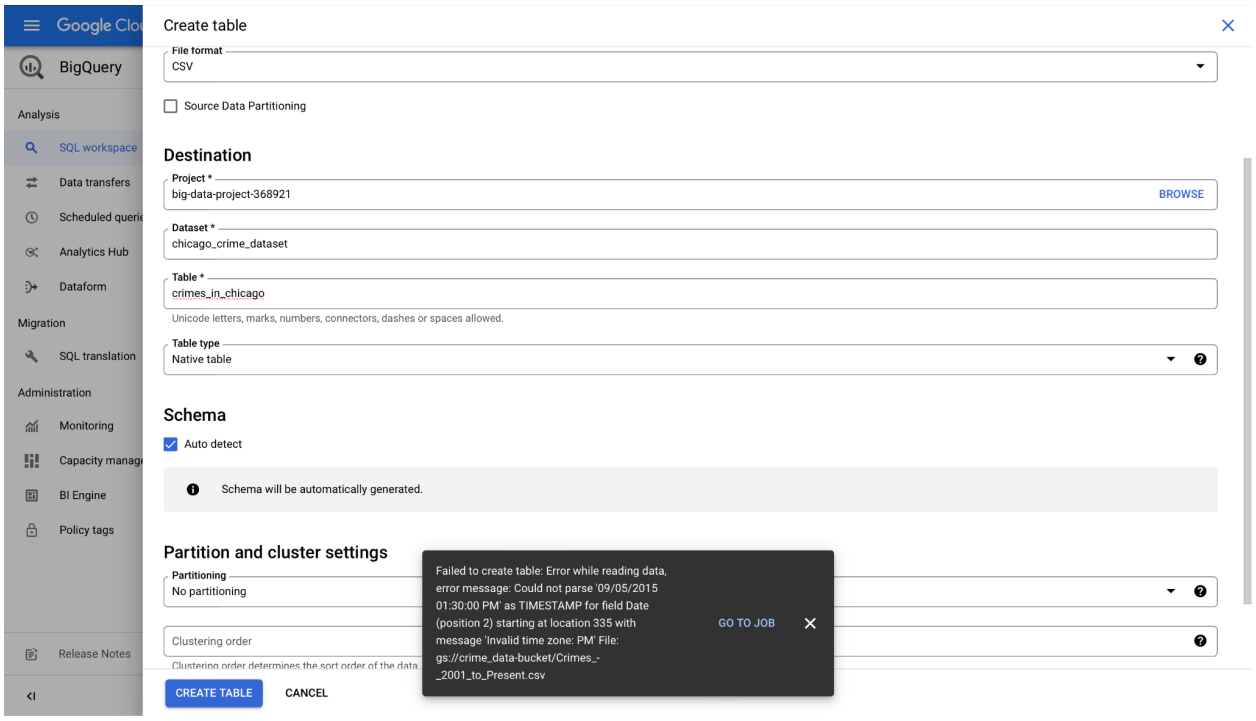
Discussion

We can interpret from the visuals clearly how crime cases vary over the years. From the first visualization we can see that the total number of arrests is less over the years this might be the case that the offenders didn't get caught. In Sheet 2 we can clearly see that the total number of crimes is decreasing from 2002 - 2022 year on year this might be the case that Chicago policing has improved greatly over time. In sheet 3 I have visualized the top 20 crime categories that are happening in Chicago city. Of all the categories battery and narcotics are the highest occurred crimes in 2009. In sheet 4 I have visualized the pie chart that depicts the percentage of each crime that has occurred. Of all of them, theft has the highest percentage. In the last visualization, I plotted the sum of total crimes for each ward the color shows the sum of total crimes and we can clearly see that ward 28 is an unsafe place with the most number of crimes and ward 19 has the least number of crimes registered.

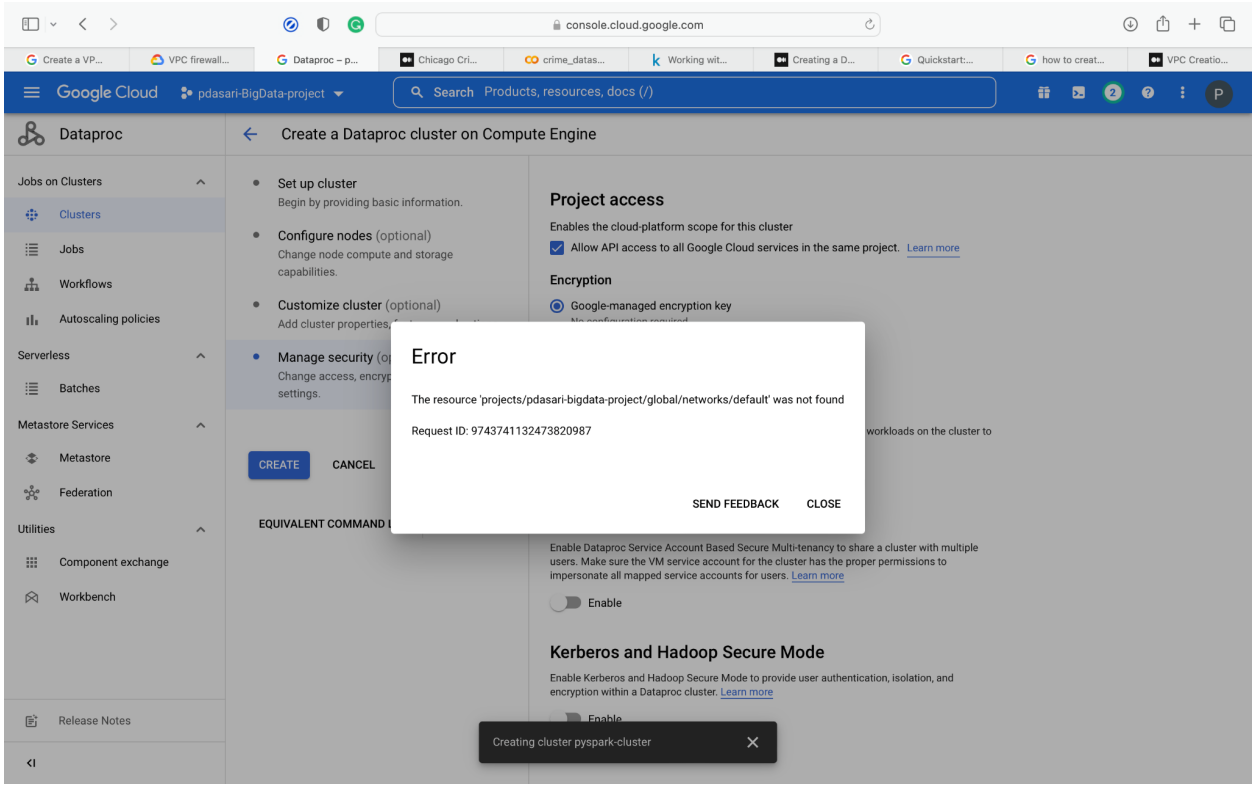
I built the ETL pipeline using the technologies and techniques covered in this course. I utilized the GCP platform, a public dataset from data.cityofchicago.org as the data model, distributed computing pyspark to clean the data, transformations in BigQuery, and a connection between the BigQuery dataset and table and the external visualization tool PowerBI.

Barriers of failure

The issue I have faced here is that in my data Date column is formatted as a string. So BigQuery failed to create the table.



Another issue I have faced here is while creating the cluster in Dataproc that is the VPC connection is not set up usually there should be a default VPC connection that will be present but in my case, it isn't there I have to create the VPC connection.



While creating the cluster initially I selected the image type as 2.0 (Debian 10, Hadoop 3.2, Spark 3.1) and anaconda doesn't work with image version 2.0.

Conclusion

Here, I've decided to use the Google Cloud Platform to create the ETL process. I built up an ETL pipeline using GCP, ingested and extracted data, cleaned the dataset using the Python Spark API, a distributed computing framework, conducted transformations with BigQuery, linked BigQuery to an external visualization tool called Tableau, and then visualized the outcomes. I was able to construct a pipeline effectively, improving my knowledge of how to do so, which I subsequently applied to a whole project. I've also learned how to create a virtual machine interface and use it to run notebooks, which is arguably the most important. These abilities will be of utmost use in both the working world and the setting in which I will be employed.

I have published my pyspark code to clean the data into GitHub here is the link.

https://github.com/pranay-98/I-535_project/blob/39fedd1c2b753b76e90a3ae4316d7e4764fd2595/dataset_transformation.ipynb

References

- https://data.cityofchicago.org/Public-Safety/Crimes-2001-to-Present/ijzp-q8t2
- https://www.usgs.gov/data-management/data-lifecycle
- https://cloud.google.com/bigquery/public-data/
- https://cloud.google.com/bigquery/docs/reference/standard-sql/conversion_functions
- https://docs.databricks.com/getting-started/dataframes-python.html
- https://supermetrics.com/blog/connect-tableau-to-google-bigquery?utm_source=google&utm_medium=cpc&utm_campaign=supermetrics-googledatastudio-phrase&utm_adgroup=datastudio-dsa&utm_category=search-nonbrand&utm_term=&location=&gclid=Cj0KCQiA4OybBhCzARIsAIcfn9mqjpaVmhu1BW09UNJ_R6_tpaFqqVCdsD_JSiKy5bOARq2BEe1pyl4aAuxpEALw_wcB