CLOUD ANALYTICS AND DATA WAREHOUSE IMPLEMENTATION

INTRODUCTION:

Understanding past trends and current transactions is crucial for organizations, to make informed decisions in the quickly changing world. This type of information originates from The New York Times, which is a well-grounded newspaper specialized in reporting news, opinions and matters of culture.

An abundant pool of real time transactional data and historical archives is accessible courtesy of NYT's APIs, which are application programming interfaces. It is due to this special combination that it becomes possible to have a profound understanding of past events, societal shifts, and current developments in various areas such as politics, economics, culture, and technology.

DATA SELECTION:

The New York Times dataset is a well-known and established source that is widely regarded for its accuracy, integrity, and journalistic standards. This dataset offers dependable and verifiable information that researchers and analysts may rely on for their analysis.

The dataset provides real-time transactional data as well as historical data, collecting the most recent news stories, activities. Because of this real-time feature, researchers can keep up with the latest trends and advancements as they happen.

Content strategy decisions can be informed by insights obtained from the examination of historical trends and real-time data, which can help discover popular themes and upcoming trends.

DATA SET:

This subsection explores what we're looking at NYT the dataset . This collection of news articles serves as the bedrock for our exploration of New York Times data. As a result, here we will examine closely the structure of the dataset with special attention paid to the attributes which are used to describe each individual article. Only by getting a good grasp of the information will we be able to analyze it effectively, extracting meaningful conclusions.

The table called "Article" holds data on various articles related to news. Article data is about "web_url" (web address), "snippet" (short summary), "print_page" and "print_section" by other attributes. The information on where an article was taken from is shown by the "source" entry. "multimedia" array describes multimedia elements like images and videos more fully.

The other fields of this array are: "type" (image or video etc.), "url", "height", "width" and "caption". In addition, the table for storing article metadata includes attributes such as "headline", "main" (which can be nested in the "headline") and "kicker", "byline" (Author names), among others. Each article also has specific identifiers—"uri" and "_id" that are saved in this table.

DATABASE CONFIGURATION

A virtual private cloud is created to enhance security, to isolate the network, hybrid cloud networking and many other facilities.

Enhanced Security:

Resources can be isolated from other networks and the public internet by being deployed inside a private virtual private cloud (VPC). By reducing the attack surface, this isolation aids in preventing unwanted access to your resources.

Network Isolation:

Using a private virtual private cloud (VPC), you can establish a logically isolated network environment in which you can deploy and administer your resources without external parties or other tenants interfering.

Controlled Access:

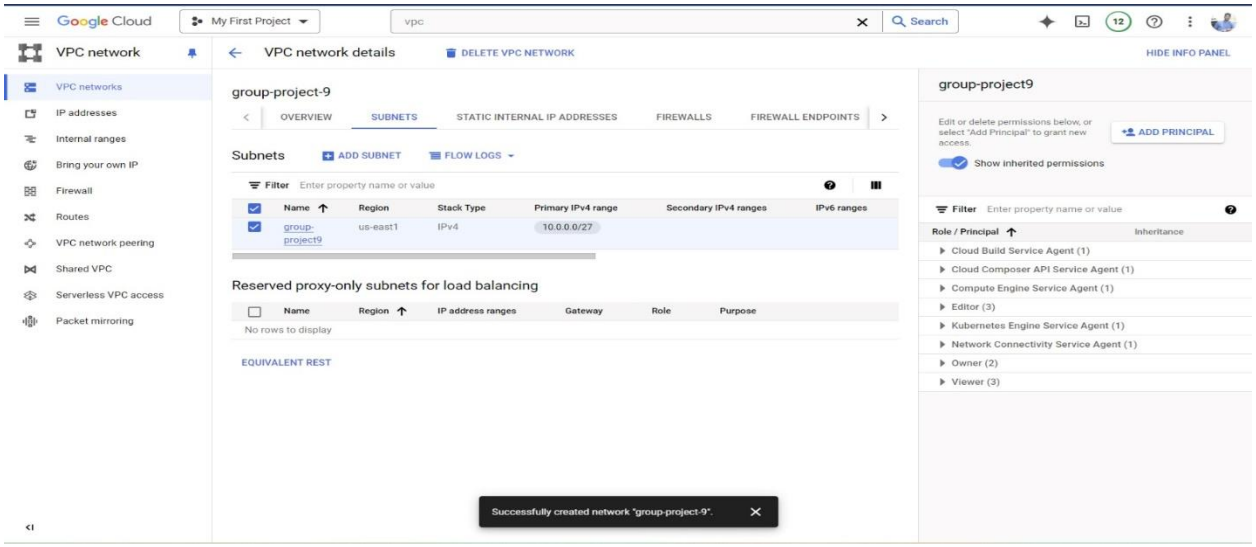
By establishing firewall rules, routing guidelines, and network ACLs (Access Control Lists) inside the VPC, you have fine-grained control over network access. This lets you apply security guidelines and limit resource-to-resource contact in accordance with your company's needs.

Hybrid Cloud Connectivity:

Private VPCs make it possible for you to create safe links between your cloud resources and on-premises infrastructure. Workload migration and integration between environments are made easier as a result.

Figure 1

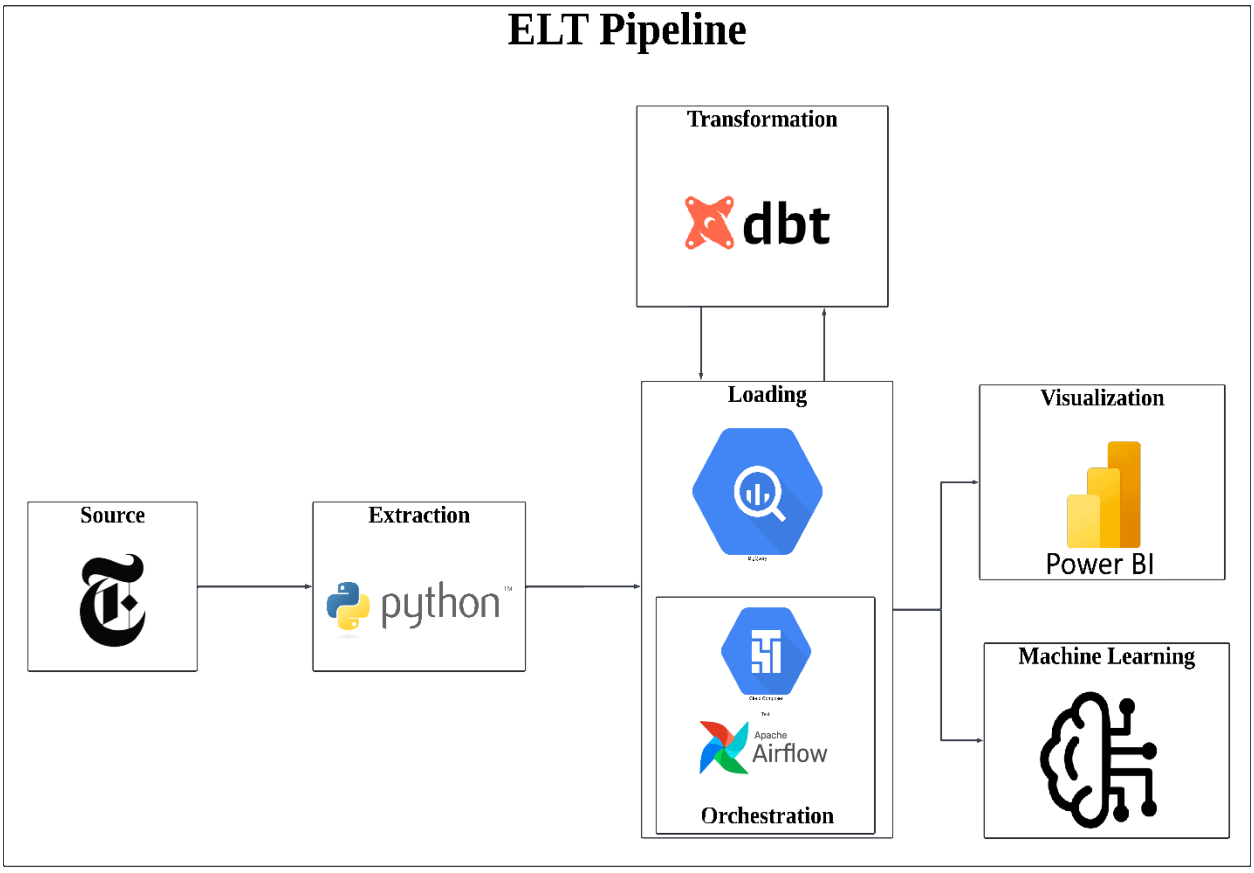
Snapshot of VPC created on GCP.



OBJECTIVES :

- Establish a robust data pipeline that seamlessly integrates with NYT's APIs and automate the extraction of both historical data and real-time transactional data.
- Implementing workflows to automatically update the dataset so that the analysis is always updated with the most recent data from NYT.
- To build a centralized data warehouse to consolidate and organize the NYT dataset, providing a unified platform for storage, retrieval, and analysis.
- Push the extracted data to cloud storage solutions, facilitating easy accessibility, scalability, and collaboration across distributed teams.
- Use data transformation techniques to clean up, standardize, and add value to the dataset.
- Create interactive dashboards with BI tools that show important metrics, patterns, and conclusions drawn from the NYT dataset.
- Machine learning algorithms are applied to data to predict and identify trends.

METHODOLOGY:



ELT process is used to build the entire pipeline. ELT is chosen because in case there is any modification needed to done to the data, one can simply run the model from transformation rather than running the entire pipeline.

DATA ACQUISITION – EXTRACTION :

In the process of exploring data from December 2022 to the present date, it became apparent that the dataset available on Kaggle contained significantly fewer records compared to those accessible via the API.

To retrieve data from API, one needs to overcome constraints like pagination and rate limit. Pagination is a constraint of limiting retrieval of records count. Rate limit is a constraint of limiting the requests over a period. This constraints changes for different types of APIs.

Archival Data :

Historic data of NYT is retrieved from "Archive API" **.** This API has only a rate limit of 5 requests per minute where 1 request retrieves one month's data.

Real-Time Data:

Streaming data of NYT is retrieved from "Article search API". This API has both rate limit and pagination constraints. Rate limit constraints, allowing only 5 requests per minute and 500 requests per day. Pagination presented an additional challenge, with each API call yielding a single page containing 10 records and a maximum of 100 pages, limiting us to access 1010 records of transactional data for a day.

To address these issues, a recursive function was defined, regulating API requests to adhere to the rate limit. This function incorporated a "*time.sleep"* function in Python, set to 12 seconds intervals, ensuring that no more than 5 requests were made per minute where each request retrieves only 10 records. This Python script effectively managed API requests within these constraints, facilitating a robust and compliant data retrieval process.

Automation:

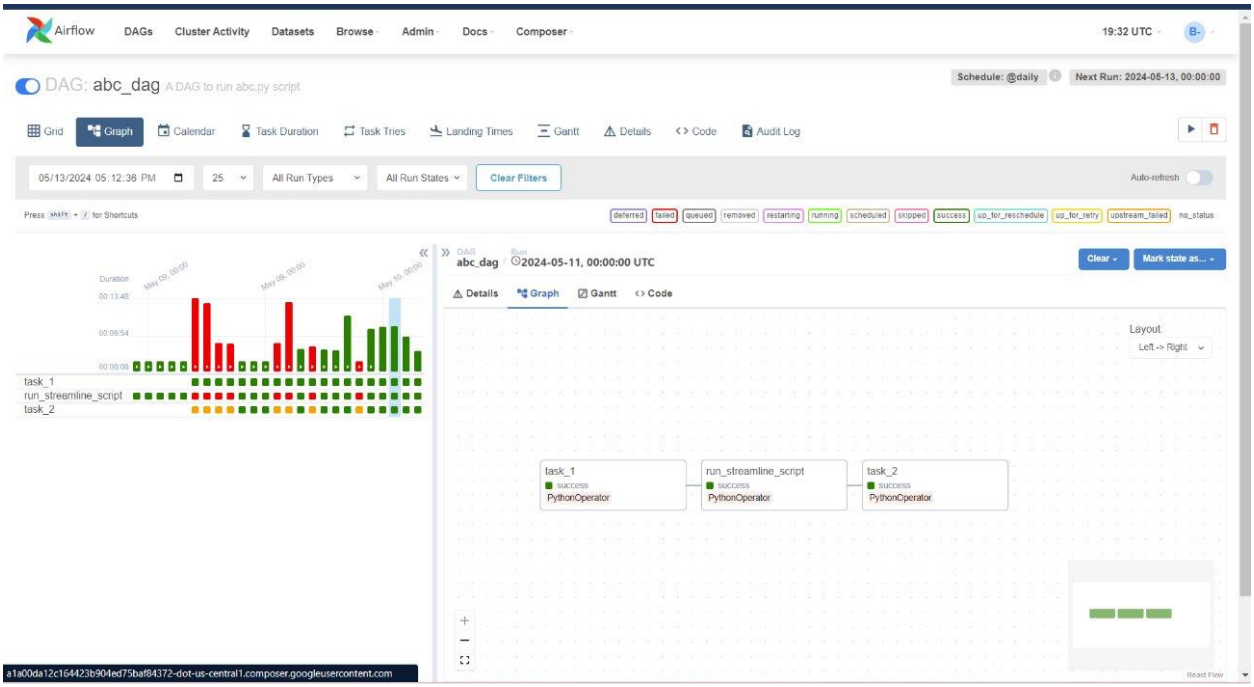
A DAG (Directed Acyclic Graph) was established in Big Query to ensure the seamless integration of newly fetched data from the API into our warehouse. This DAG orchestrates the data pipeline by performing several key steps.

The DAG orchestrates the data pipeline in several steps. Initially, it extracts data from both yesterday and the day before yesterday from the API. Subsequently, it compares the API data with the data from the day before yesterday in the warehouse and pushes the data into the data warehouse.

The directed acyclic graph is a python script with airflow supporting libraries which automate the live data extraction. Air flow is connected to Big Query using "Composer". The DAG file is uploaded to Google composer which acts as a catalyst to ensure the orchestration and automation.

Figure 2

Airflow DAG snapshot



A DAG script is created to extract NYT live data everyday at 5PM PST. This script also ensures that there are no archival data updates, if there is an update in archival record it will delete the older record and append it into log table after updating the older record.

DATA WAREHOUSE - LOADING:

The historical data was complex and included a variety of data columns in JSON format, including lists of dictionaries, arrays, structs, and arrays of structs. Big Query turned out to be the best answer for project needs, as it provides a perfect framework for investigation by allowing both relational and NoSQL formats to be accommodated while maintaining the flexibility of SQL queries. Big Query architecture allows us to store vast amounts of data in cloud environments. Also, it provides a direct connection to business tools like PowerBI. This platform is cost effective when compared to other cloud platforms.

Two folders were created in Big Query bucket to load and store raw and transformed data separately. The data that has been extracted using python was loaded in Big Query bucket by establishing a connection between python and Big Query. Google Big Query credentials were used and required libraries were imported to establish a python connection. Python is used to create table schema in Big Query. And the same table schema that was mentioned in API documentation was followed.

In the process of loading live data into the warehouse, considerations like logging were addressed. For example, if the DAG runs today, it updates yesterday's data into the warehouse and

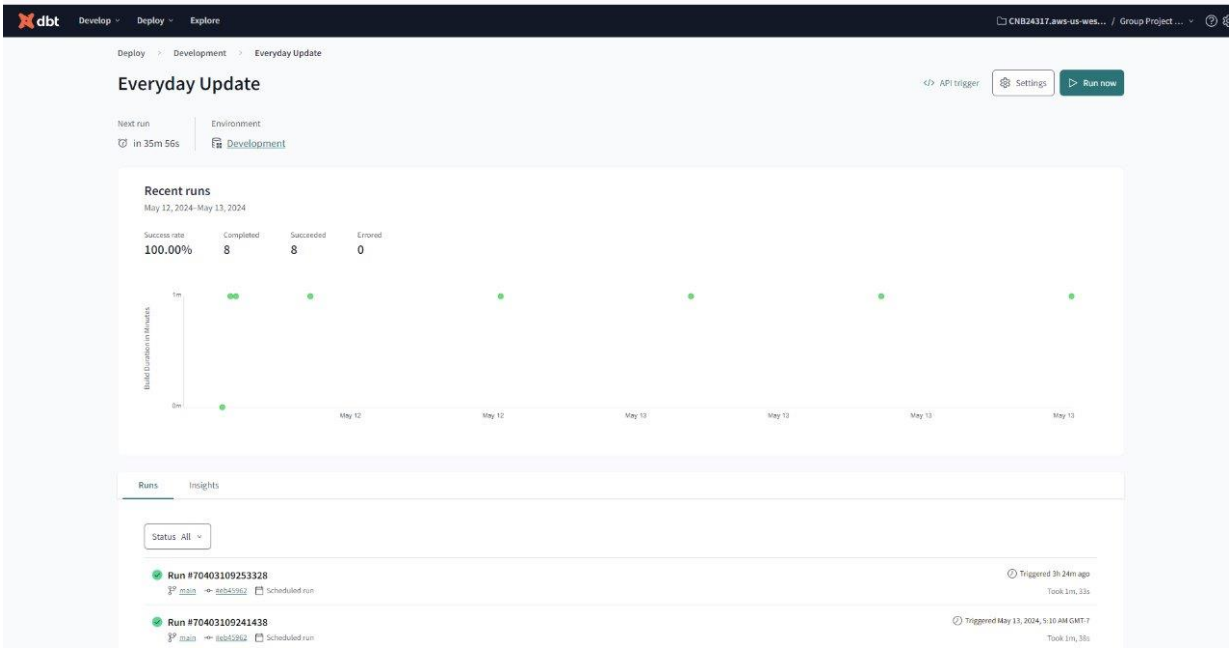
simultaneously compares the day before yesterday's data to identify any changes or updates. Upon detecting modifications, the DAG pushes the updated record from the warehouse to the log table along with a timestamp, and subsequently deletes that record from the warehouse. Then, the DAG proceeds to push both yesterday's data and the updated data (from the day before yesterday) to the Big Query cloud, ensuring that our warehouse remains current and reflective of the latest data trends.

DATA WAREHOUSE IMPLEMENTATION – TRANSFORMATION

DBT is used to transform data that was uploaded in the cloud (Big Query bucket). Though Big Query allows us to transform the data, DBT is preferred because of its flexibility to define macros and packages which reduce redundancy and improve maintainability of the data transformation logic. Version control systems like Git are easily integrated with DBT, enabling reliable versioning of data models and transformations. DBT has a facility to schedule tasks. This scheduling helps us to automate and run all the DBT transformations every time the DAG completes the task. This makes it easy for team members to work together and to track changes over time.

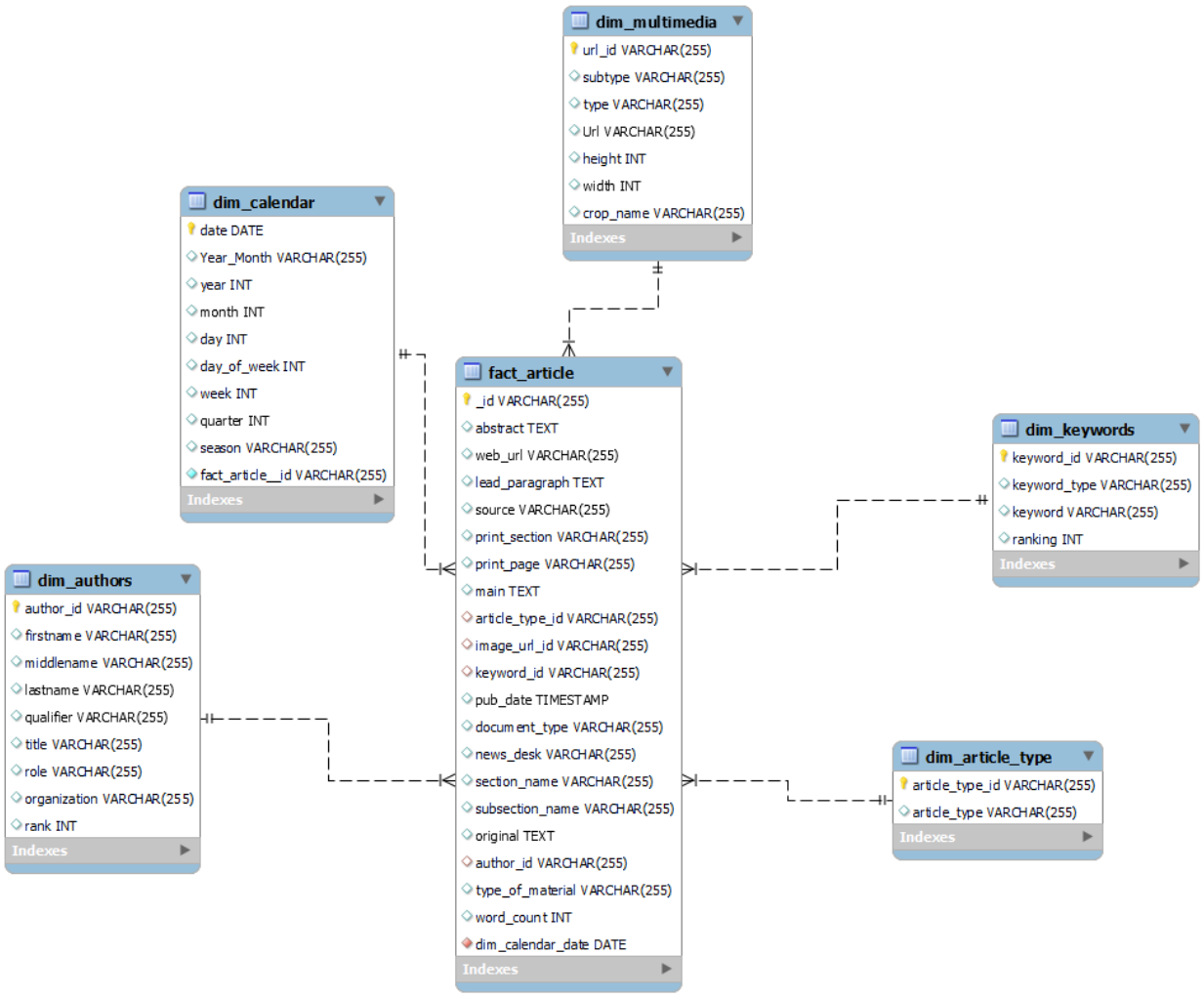
Figure 3

Snapshot of DBT scheduling



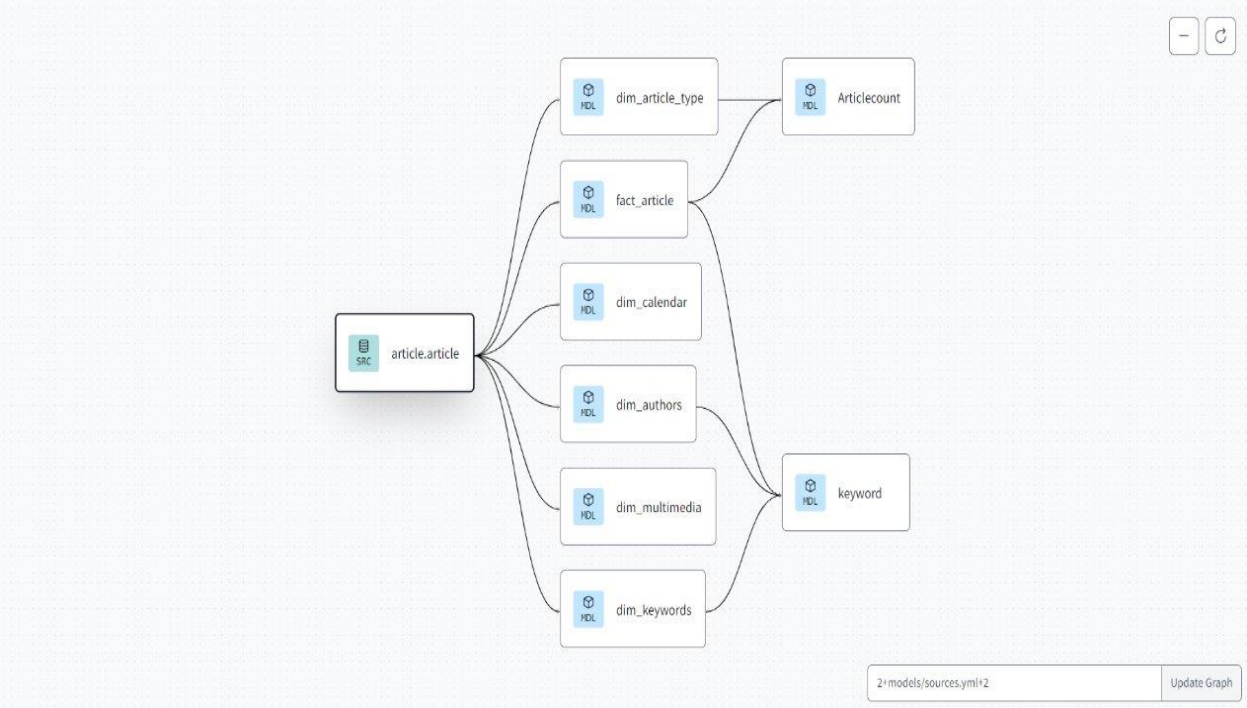
Star schema was employed to transform the data over snowflake schema because snowflake schema has more normalized dimensions tables leading to complex query structures with more joins resulting in poor performance.

Figure 4
Snapshot of ER diagram



By segregating the data into distinct facts tables containing measurable metrics and dimensions tables offering contextual and descriptive attributes, we achieved a streamlined architecture conducive to efficient data analysis. This clear separation facilitates easier navigation through the dataset, reduces redundancy, and promotes scalability.

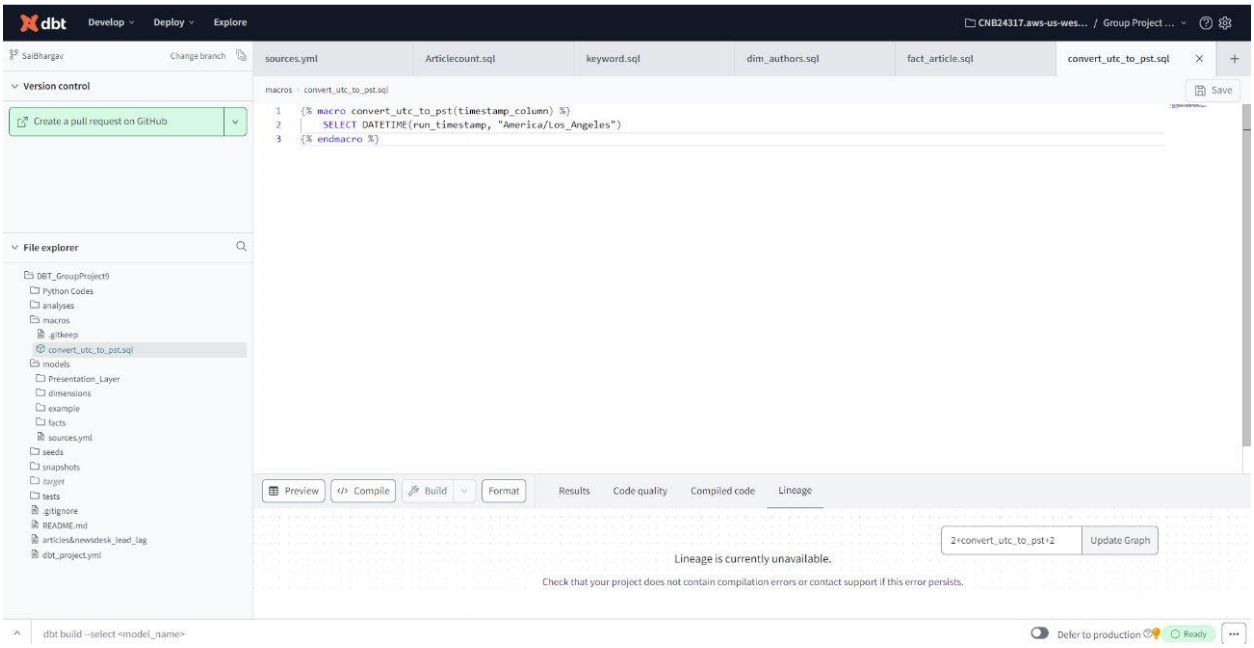
Figure 5
Data Lineage (STAR Schema)



For the data transformation process, we employed the star schema, partitioning the dataset into facts and dimensions tables. This approach offers several advantages over other schemas, such as snowflake. Firstly, it enhances query performance by minimizing the number of join statements needed to retrieve data. This streamlined querying process leads to faster response times and improved efficiency. Additionally, the star schema simplifies data querying and enhances analytical capabilities by providing a clear and intuitive structure.

In DBT, a macro is implemented for addressing specific data transformation requirements. The data retrieved from the API contains timestamps in UTC format, which may not align with the project desired time zone for analysis. To overcome this, a macro was created that converts UTC timestamps to PST (Pacific Standard Time) timestamps using DateTime function applied on run_timestamp column, ensuring consistency and accuracy in data analysis process.

Figure 6
Snapshot of DBT macros



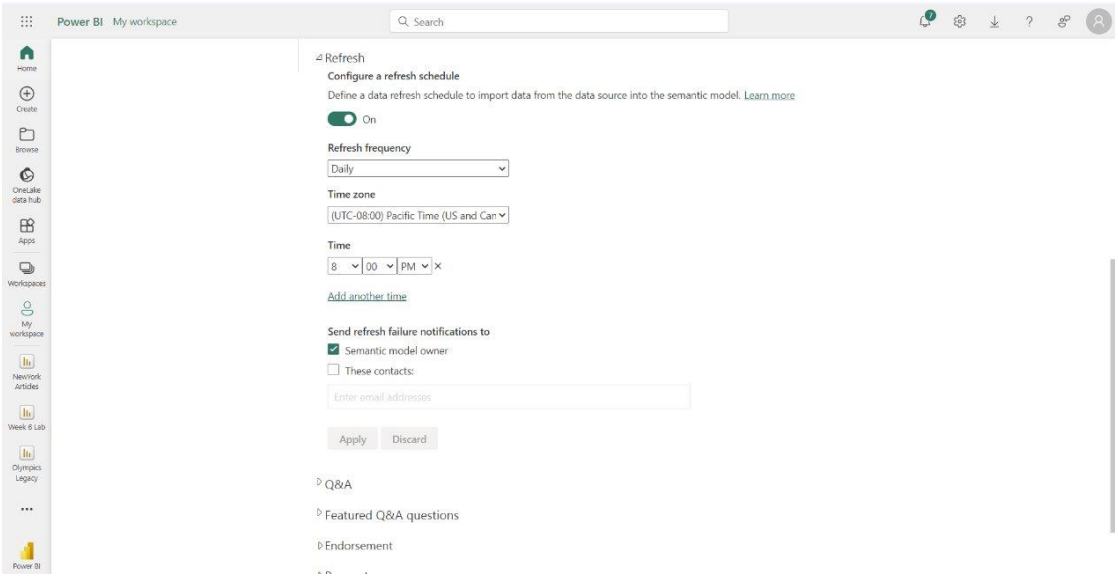
Data has been finally transformed into facts and dimensions using DBT and pushed to cloud. Raw data and transformed data are stored separately to reduce the risk of modifying or corrupting the original data during data transformation. The transformed data had SQL operations done on it for analyzing the information from it, draw insights and report it in a dashboard visualization format to assist in making informed decisions.

BUSINESS INTELLIGENCE – POWERBI

The business intelligence tool Power BI was selected to visualize various trends and highlights derived from the transformed dataset. Its user-friendly interface and robust features for establishing relationships between tables made it an ideal choice for our visualization needs. We connected Power BI and BigQuery so that easy access to the dimensions and data stored in BigQuery could be provided. Through this integration, we were able to take advantage of the advantages offered by each platform: BigQuery's strength in data storage and retrieval was combined with Power BI's ability to provide user-friendly, interactive data presentation. We can refresh Power BI to get new data into visuals.

Figure 7

Snapshot of PowerBI scheduled refresh.



To obtain insights from the data that support well-informed strategic decision-making processes and offer insightful direction for upcoming actions and projects, Power BI visualizations were important in this process. All things considered, BigQuery and Power BI together proved to be a potent toolkit for turning data into useful insights and achieving business objectives. Two dashboards were made for the NYT dataset to show different trends.

Figure 8

Dashboard-1

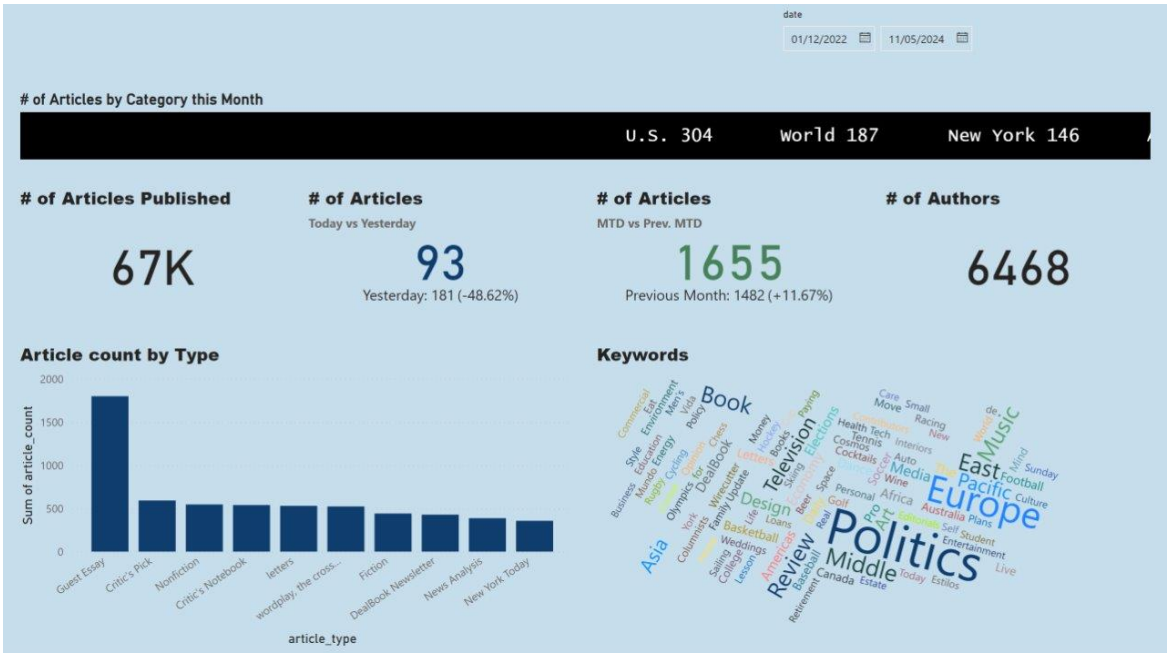
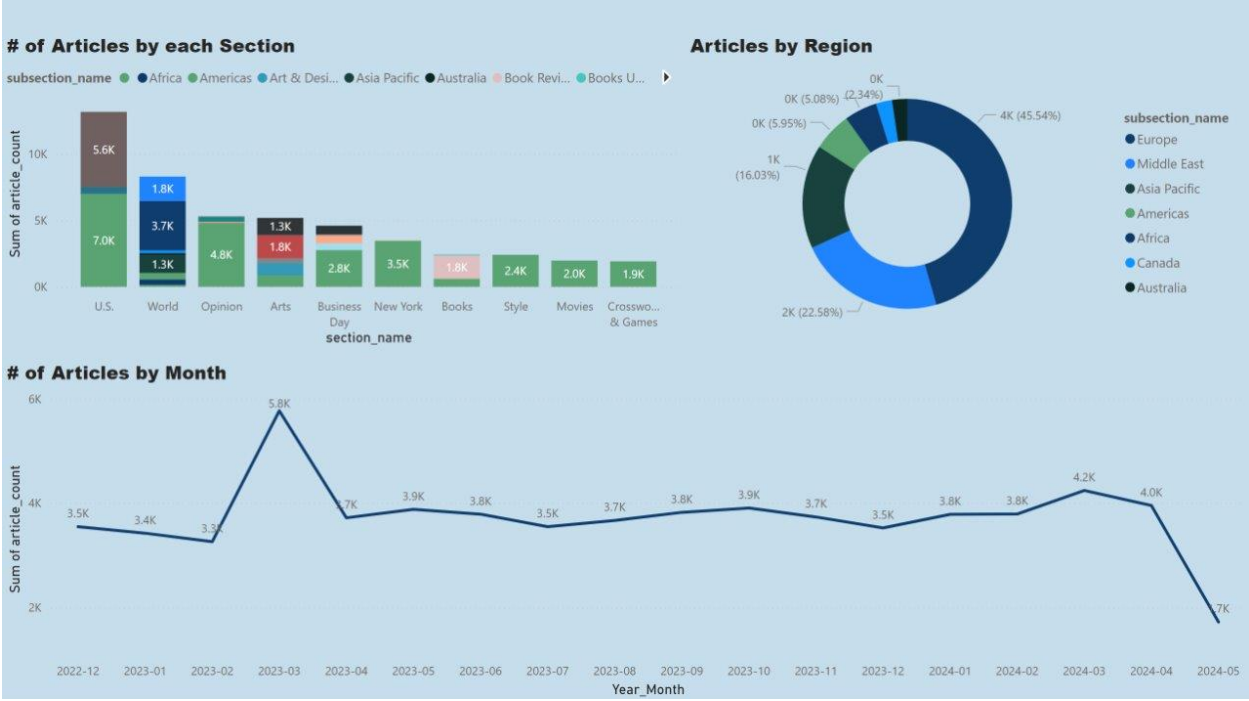


Figure 9
Dashboard-2



Insights from Dashboard:

KPI

KPI Charts created in the dashboards help to monitor and evaluate the performance of the process or data. In this New York Times data visualization four KPI are created to provide the details of article counts.

1. Articles published KPI cards represents the total count of the articles published. This helps to have an idea about the number of articles published from 2023 to date.
2. Today Articles published vs Yesterday represents the comparison between yesterday and today's published articles count.
3. Mtd vs previous Mtd KPI represents the comparison between articles published from current calendar month up to today's date with the previous month data. The current month published articles count is 11.7% more than the previous month.
4. Authors KPI provides information about the total number of authors who published articles in the New York Times.

Word Cloud

Word cloud is used to display the most frequently used keywords in articles. From the visualization, many articles are related to US politics and Elections and Business-related topics.

Bar charts

Bar charts are created in the dashboard to display the sum of article count based on two categories i.e. article type and article section. In the article type bar chart, Essay type articles are published more compared to other types. The remaining article types are approximately published in equal number. In article section and sub-section bar chart, Sum of article count is estimated based on the news category section.

Pie chart

Pie chart is used to represent the count of articles published in each region. The highest number of articles are published in Europe and Africa and least is Australia and Asia Pacific.

Line charts

Line charts show the increasing pattern of articles published over time, with steeper rise in few months like March.

This dashboard helps to understand the brief details about the articles published over the time in the New York times dataset. This helps the business in taking strategic and informed decisions about publishing articles in the NYT.

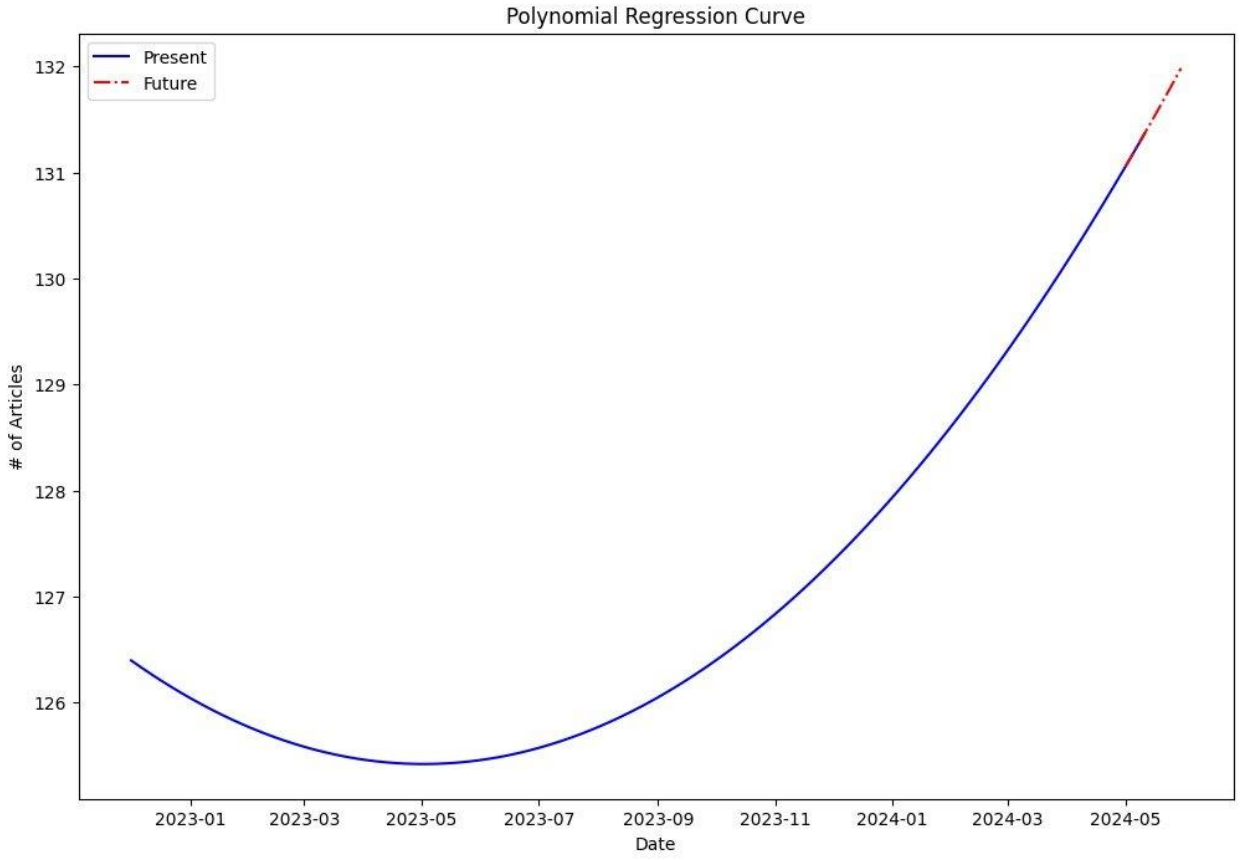
MACHINE LEARNING – POLYNOMIAL REGRESSION MODEL

Polynomial regression is a supervised learning algorithm used in predictive analysis. Polynomial regression allows you to capture non-linear relationships between input and output variables by introducing polynomial terms of the input variables.

In the New York Times dataset, article published data was chosen as the independent variable, and the number of articles published per day was taken as the target variable. By employing polynomial regression with a degree of 2, trends in the data can be captured. This allows one to make more accurate predictions on the number of articles that might be published each day. This machine learning model provides valuable insights for further decision-making and strategic planning. The plotted graph demonstrates the prediction of future trends.

Figure 10

Polynomial Regression Plot



Recommendations:

Based on article count predictions, The New York Times can make informed decisions to optimize their content strategy and resource allocation. Here are some recommendations:

Content Planning: Use the predictions to plan and schedule content creation, ensuring a consistent flow of articles to meet predicted demand.

Resource Allocation: Allocate resources effectively based on predicted article counts to ensure that the right number of staff and resources are available to meet publishing needs.

Trend Analysis: Analyze trends in article counts over time to identify patterns and adjust content strategy accordingly to capitalize on popular topics or address any potential dips in readership.

Audience Engagement: Use predictions to tailor content to audience preferences.

CONCLUSION

Essentially, this project serves as an example of how data-driven strategies may revolutionize organizational performance and creativity. Through the seamless integration of cloud deployment, BI, ML, and ETL automation, we have created a strong framework that facilitates the extraction of actionable insights from data, drives value generation, and cultivates an organizational culture of data-driven decision-making. We can seize even more chances for expansion, improvement, and influence in the dynamic field of data analytics and intelligence as we carry on iterating and improving our procedures.

GIT LINK - https://github.com/SaiBhargav3110/DBT_GroupProject9.git