**Weather Data collection and insight generation with Big Data technologies.**

Nagaraju, Jayanna, Goudar.  
Department of Information, medien und technik *SRH HOCHSCHULE*HEIDELBERG, GERMANY.

*Abstract* — This paper presents a comprehensive approach to collecting, preprocessing, and analyzing weather data using the Google Cloud Platform (GCP). Leveraging the Bright Sky API, our team gathered extensive weather data, which was subsequently processed and stored in GCP. Key aspects of the project include data collection using python and it’s associated libraries, preprocessing using Dataproc, storage in BigQuery, and visualization using Looker Studio. The project addressed significant challenges in data pipeline setup and management, offering valuable insights into efficient data handling and analysis.

Keywords—GCP, SCHEDULER, CLOUD STORAGE, BIG QUERY, LOOKER STUDIO.

# **Introduction**

# Weather data is critical for various applications some of the prominent applications include supply chain management and disaster management. This project is aimed to collect and analyze weather data to uncover patterns and correlations. Our approach utilized GCP and it’s in house tools for data storage and processing, ensuring robust and scalable data handling.

1. ***Methodology***

***A. Data Collection****The data collection phase involved the use of the Bright Sky API, a free and open-source weather data API provided by the DWD (Deutscher Wetterdienst, Germany's national meteorological service). This API was selected due to its comprehensive data coverage and ease of access.*

*A diagram of a project

Description automatically generated*

1. ***API Details****: The Bright Sky API offers various endpoints for retrieving current weather, historical weather data, and forecasts. For this project, we focused on historical weather data to analyze long-term patterns and trends.*
2. ***Data Acquisition****: We initially collected data consisting of 15,000 records and 20 features (columns). The data included parameters such as temperature, humidity, wind speed, precipitation levels, and sunshine duration. The API responses were in JSON format, which were then converted to CSV for easier processing.This was acccomplished using a cloud function where we ran a python script to fetch data from the API.*

A screenshot of a computer

Description automatically generated

III. **Data Processing**

A. Data preprocessing is a crucial step to ensure the quality and usability of the data for analysis. Naturally the data fetched from the the API would not be in the right form for usage and hence some pre-processing was employed from our end. Our pre-processing pipeline was implemented using Google Cloud Dataproc, a fully managed Apache Spark and Apache Hadoop service inside the GCP itself.

1. **Storing Raw Data**: Raw data from the API was stored in Google Cloud Storage (GCS) buckets. GCS provides scalable and durable storage, making it an ideal choice for handling large volumes of data.
2. **Data Preprocessing Steps**:

* **Removing Duplicates**: Duplicate records were identified and removed to avoid redundancy.
* **Removing Unwanted Columns**: Columns that were not necessary for our analysis, such as metadata or redundant identifiers, were removed.
* **Handling Missing Values**: Rows with null values were carefully examined. If the missing data was substantial, the entire row was removed to maintain data integrity.

1. **Transformation and Enrichment**: Additional transformations and enrichments were performed using Google Cloud Data-proc. This tool provides an intuitive interface for cleaning, structuring, and enriching raw data.

A screenshot of a computer

Description automatically generated

**C. Data Storage and Querying**The processed data was stored in **Google Big Query**, a fully-managed, serverless data warehouse that allows for fast SQL queries using the processing power of Google's infrastructure.

1. **Integration with Big Query:** The pre-processed data was pushed to Big Query which houses an in house database of sorts where it was stored as tables. Big Query's architecture supports complex queries and large-scale data analysis, making it suitable for our needs**.**
2. **Data Pipeline:** The entire data pipeline, from API data retrieval to preprocessing and storage in Big-Query, was automated using Python scripts. These scripts were executed in a Google Cloud Data-proc environment, ensuring seamless integration and efficient data handling.

# Data Visualization and User Stories

* ***A. Visualization Tools****Data visualization is essential for interpreting the results of data analysis. For this project, we used Looker Studio, an in-house visualization tool from Google.*
* ***Tool Selection****: Looker Studio was chosen due to its powerful visualization capabilities and seamless integration with BigQuery. It provides a user-friendly interface for creating interactive and insightful visualizations.*
* ***Connecting Data Sources****: Establishing a connection between Looker Studio and our BigQuery tables was straightforward. This integration allowed us to create dynamic dashboards and visualizations based on real-time data.*

**B. User Story 1: Analysis of Temperature Trends Over Time** As a meteorologist, I want to analyse temperature trends over different periods (seasonally) to identify patterns in the temperature.

1. **Objective**: The goal was to study how temperature changes over different seasons and identify any recurring patterns.
2. **Visualization**: We used line plots to display temperature trends over time during various seasons over the peiodd of a year.
3. **Insights**: The line plots revealed seasonal temperature variations, helping meteorologists understand long-term trends and predict future changes.

A graph with blue dots

Description automatically generated

**C. User Story 2: Studying Correlation Between Sunshine and Precipitation Levels**

As a climate researcher, I want to study the correlation between sunshine and precipitation levels to understand their relationship and how they affect each other.

1. **Objective**: This analysis aimed to explore the relationship between the amount of sunshine and precipitation levels. Understanding this relationship can provide insights into weather patterns and climate behaviour.
2. **Visualization**: Scatter plots were used to visualize the correlation between sunshine duration and precipitation levels. Correlation coefficients were calculated to quantify the relationship.
3. **Insights**: The relationship between precipitation and sunshine is inversely proportional. The higher the sunshine, lesser the precipitation and so on.

A graph of blue and green bars

Description automatically generated

**IV. Challenges and Solutions**

**A. Challenges**

* **Setup and Implementation of Airflow**: Initially we had tried to implement Apache Airflow, during which one of the major challenges was setting up and implementing Apache Airflow for orchestrating the data pipeline. Airflow required careful configuration and testing to ensure reliable operation. Even though DAG’s were created, connecting them to databases was not an easy task and often involved errors and error log lookups which was not at all easy to interpret.
* **Connecting Pipelines to Databases**: Establishing seamless connections between various components of the data pipeline (e.g., from GCS to Data-proc to Big Query) posed significant challenges. Ensuring data consistency and integrity across these transitions was crucial.

B. Solutions

* **Troubleshooting Airflow**: Extensive troubleshooting and testing were conducted to configure Airflow correctly. Documentation and community support played a crucial role in resolving issues.
* **Pipeline Optimization**: We optimized the data pipeline by implementing robust error handling and data validation mechanisms. This ensured smooth data flow and minimized disruptions.

#### V. Conclusion

The project demonstrated the effective use of GCP for weather data collection and analysis. Key takeaways include the importance of efficient data preprocessing, the power of Big Query for data analysis, and the utility of Looker Studio for visualization. The experience gained in handling APIs, building data pipelines, and integrating various GCP services provided valuable insights into modern data engineering practices.

#### Acknowledgments

We thank the DWD for providing the Bright Sky API and our mentors for their guidance throughout the project. Most importantly we give sincere thanks to professors Mohammed Abufouda and Dr. Frank Schulz.

#### References

[1] DWD: <https://www.dwd.de/DE/Home/home_node.html>

[2] Bright Sky: [https://brightsky.dev](https://brightsky.dev/)

[3] Google Cloud Platform Documentation available:

<https://cloud.google.com/docs>

#### Author Information

* **Gurudarshan Bangalore Nagaraju –** Weather website, project topic brainstorming, User story and Data Visualisation, Presentation PPT and IEEE Report.
* **Rakesh Nagaratta Jayanna –** GCP connections.
* **Amogh Shrinivas Goudar –** API.