Environmental Sensor Data Analysis

Group 4:

Sanket Janolkar

Gautam Nair

Apoorva Dharadhar

Izhaar Khan

Table of Contents

Description of Project

Introduction

In an era where environmental changes are becoming increasingly unpredictable, there is a critical need to better understand and forecast these changes to mitigate their impacts effectively. This project aims to tackle this challenge by constructing a data pipeline that processes environmental sensor data to enable advanced analytics and predictive modelling. By analysing this data, we can identify trends and patterns that help in predicting environmental conditions, contributing to more informed decision-making in environmental policy and management.

Problem Statement

Analysing environmental data that can provide actionable insights into changing environmental conditions. Predicting the weather for future dates based on the trained Machine Learning Models.

Motivation for the Project

The decision to focus on this problem stems from an increasing urgency to adapt to and mitigate the effects of environmental changes. With the growing availability of environmental data from sensors and the advancement in cloud computing and machine learning, there is a significant opportunity to harness these technologies to foster a better understanding of environmental dynamics.

Objectives

The primary goal is to demonstrate a robust data handling and analysis framework capable of:

- Efficiently extracting and storing vast amounts of sensor data.
- Transforming the raw data into a form suitable for advanced analysis and machine learning.
- Providing both basic analytics and complex predictive insights through visualizations and model deployment.

Data Engineering Goals

The data engineering component of this project is structured around several key goals:

- Efficient Data Ingestion and Storage: To extract data from diverse sources such as GitHub repository and API and store it reliably in Azure Data Lake.
- Automated Data Workflow Management: Utilizing Apache Airflow to orchestrate data processing, ensuring data integrity and timeliness.

- Robust Data Processing and Transformation: Data processing tasks, including cleaning, normalization, and feature engineering to prepare data for analysis.
- Scalable Data Analytics and Machine Learning Deployment: To analyse the processed data for insights and predict future environmental trends using machine learning models.

Downstream Consumers

The downstream consumers of the data and insights from this pipeline would include:

- Environmental Agencies and Policymakers: Who require detailed reports and forecasts to make informed decisions regarding environmental management and legislation.
- Research Institutions: That could use the data for academic research and the development of more advanced environmental prediction models.
- Public and Community-Based Organizations: Interested in understanding local environmental trends to advocate for community-specific actions.

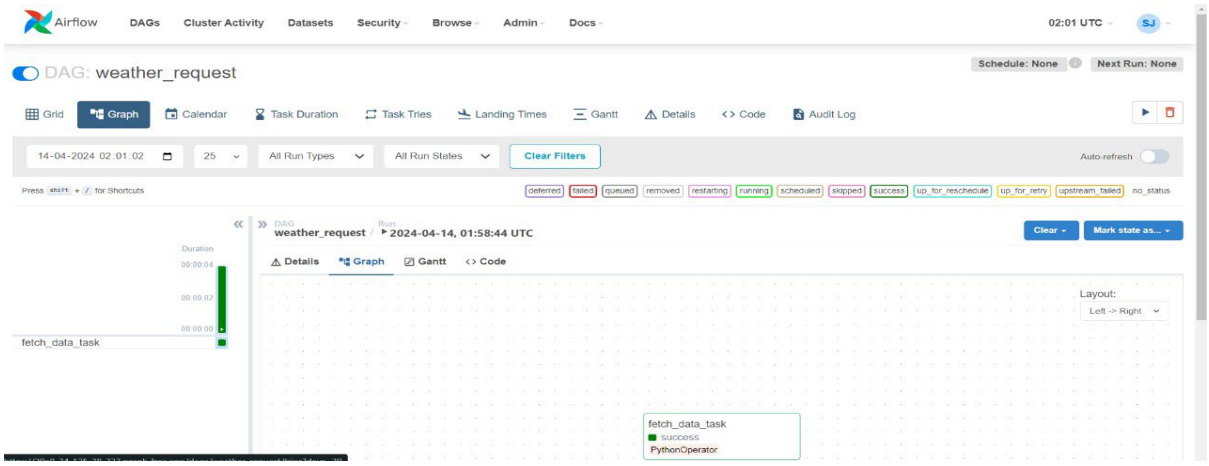
Technologies and Methodology

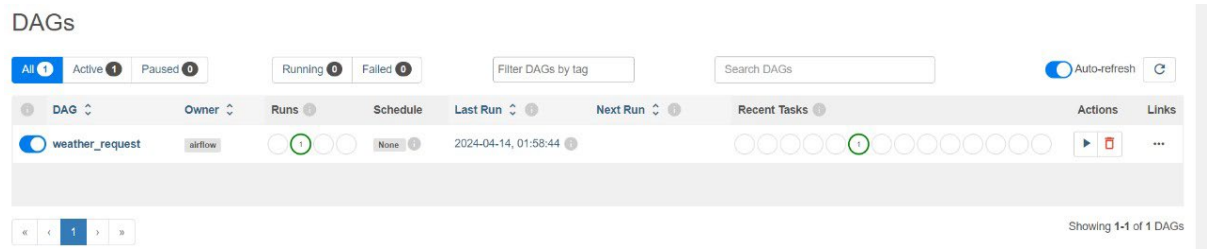
The technologies selected for this project include:

- **Google Colab:** This is where data ingestion and transformation take place, here we have extracted the data from the GitHub repository that has the flat files (historical weather data from 2018 to 2024). After ingestion we have done the ETL process in the same google colab environment.

The above screenshot suggests that the data is fetched from GitHub and unzipped for further processing.

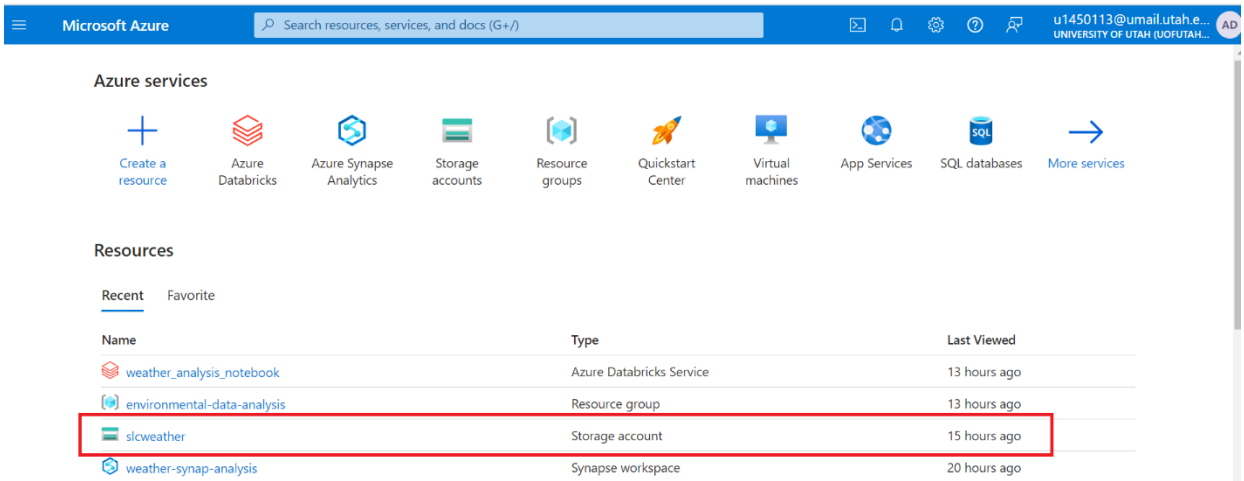
- **Apache Airflow:** Is used to manage the workflow, specifically to trigger the API call, the data from which is used for weather forecasting for the future dates.



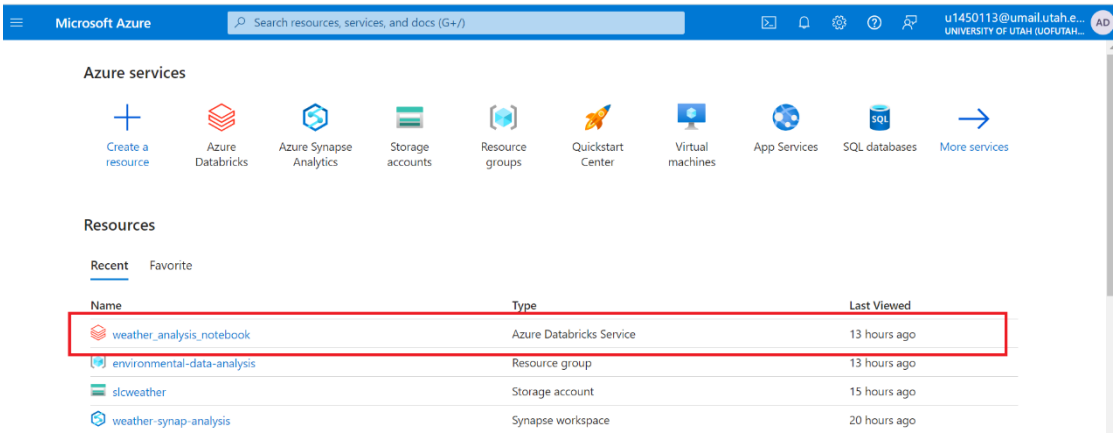


The above snippet suggests that the task 'fetch_data_task' is completed in the Airflow and the DAG weather_request is completed.

- **Azure Data Lake:** The transformed data is stored in the Data Lake. In Azure, we established a resource group named 'environmental_data_analysis' to facilitate our project endeavours. Utilizing Azure's suite of services, we employed Data Lake, Azure Synapse Analytics, and Databricks to develop machine learning models for predictive analytics.



- **Databricks:** Databricks provides tools that help you connect your data sources to one platform to analyse datasets.



- Visual crossing weather API**:** To test our machine learning model and predict the future weather.

Data Source

The primary source of data is the API from **Visual crossing weather** and the dataset pulled from GitHub. The dataset contains the data from 2018 to 2024.

The GitHub link from which the data is pulled into Google Colab:

https://github.com/sanket101-git/Local-Climatological-Data/raw/main/Sensor_Data.zip

The API from which the data is used for testing:

https://weather.visualcrossing.com/VisualCrossingWebServices/rest/services/timeline/84102/2024-04-24?key=APQ2ETC2EEYUG77LYQELXXHLY&include=days

Here, in the above API, we are providing the zip code and the date for which you want the weather forecast for.

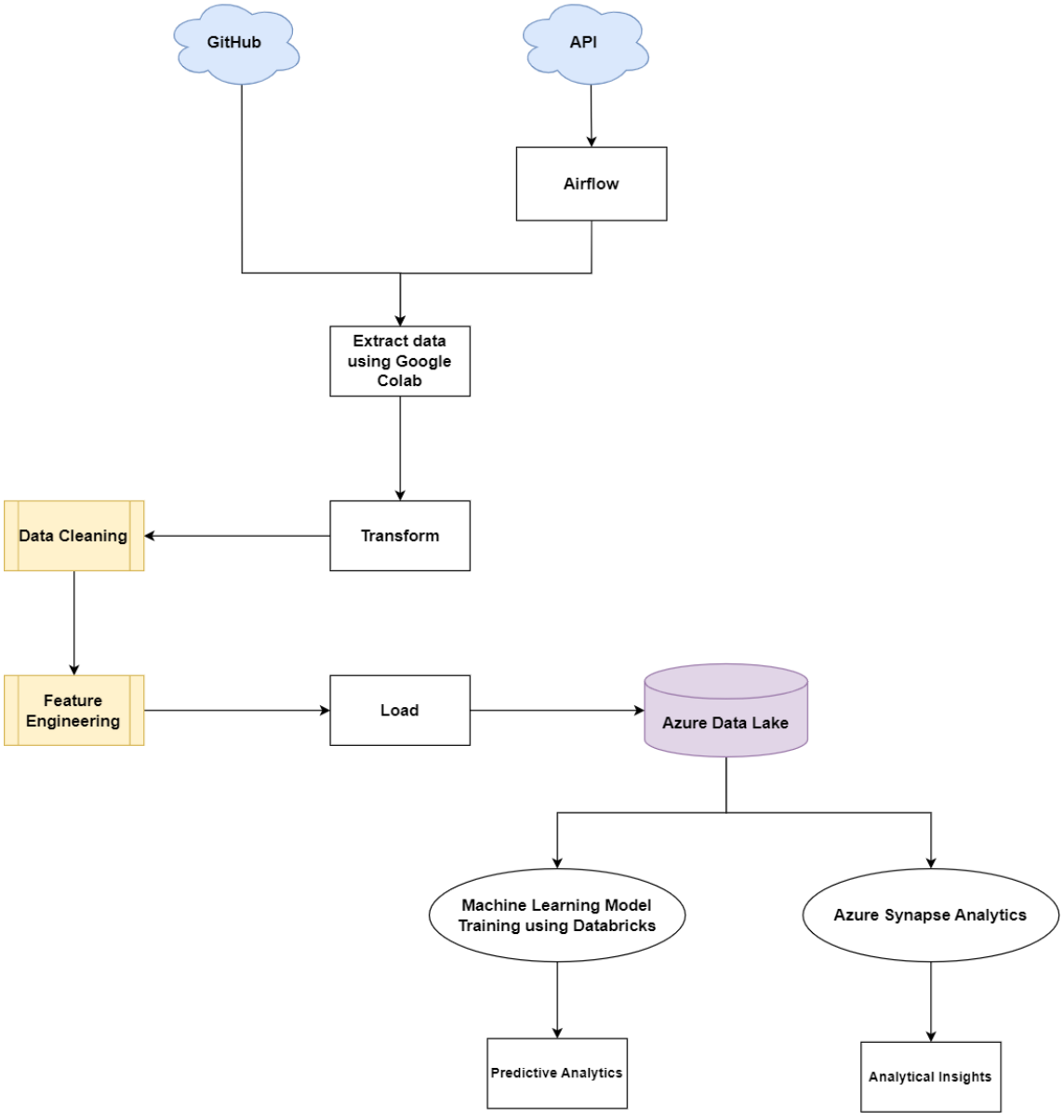
Workflow and Architecture

The workflow consists of four main stages:

- **Extract:** Data is fetched from Visual Crossing Weather website's API and dataset. Dataset contains the weather information from 2018 to 2024. This dataset is uploaded to GitHub from which it is later extracted into Google Colab and unzipped for further processing. The data which is obtained from both these data sources is stored in the Azure Data Lake. API is used for testing the trained Machine Learning model for weather forecasting of the future dates. The Airflow is used for calling the API at the time of weather forecasting for the future dates.
- **Load**: The data pulled from GitHub is ingested to google colab for transformation and then the cleaned data is loaded into Azure Data Lake.

- **Transform:** Data cleaning, feature engineering, and normalization processes are applied to prepare the data for analysis.
- **Serve:** The processed data is utilized to generate analytics and drive a machine learning model that predicts future environmental conditions.

Workflow Diagram:



The following were the changes that were made in the project proposal:

- We have extracted the data from GitHub.
- The extracted data is cleaned, and feature engineering is performed on the data. After transforming the data, the data is loaded into the Azure Data Lake.
- We have used the data fetched from the API to perform testing on our machine learning model and predict the future weather.
- The machine learning models are trained, and predictive analytics is performed on Databricks.

Task 2: The Data Engineering Lifecycle

Please explain how you went about working on your project.

In order to choose a topic that would guarantee each team member would be able to contribute meaningfully, we first conducted extensive study and analysis. We chose an environmental sensor data analysis project because it provided a wide variety of activities that matched the skills of each team member, from data engineering to machine learning. Because Google Colab is known and convenient from class assignments, we used it for both data ingestion and manipulation. We had originally intended to utilize Google Cloud, but we changed our minds and decided to use Azure instead because of its better resources, which included Databricks for creating and evaluating machine learning models, Azure Synapse Analytics for processing data, and Azure Data Lake for storage. This technology improved our project's capacity to accurately predict environmental trends and enabled an efficient workflow management system. Throughout the project, we emphasized collaboration and maintained regular communication through Zoom meetings, text messaging, and in-person meetings every week to ensure a cohesive and productive team environment. Additionally, we kept a shared Word file to document all work done, ensuring transparency and continuity in our project documentation.

Please lay out which team members tacked various stages of the data engineering lifecycle.

Every team member contributed significantly to the effective completion of our environmental sensor data analysis project in a way that was specific to their areas of experience and talent, fostering a cooperative and productive work atmosphere. The first step of the data gathering process was led by Gautam, who skilfully oversaw the extraction of historical and current meteorological data from a GitHub repository and through APIs. He used Apache Airflow to automate and oversee this process in Google Colab. He had to adjust assure uninterrupted data flow due to issues with API rate restrictions and data uniformity.

The next round of data transformation was handled by Apoorva, who concentrated on feature engineering and data cleansing to prepare the dataset for analysis. She ran into problems with non-uniform formats and missing data, which made it necessary to create strong cleaning methods to normalize the dataset across various data sources.

During the data loading stage, Sanket played a key role in effectively managing the storage of the converted data in Azure Data Lake and making use of Azure Synapse Analytics to enable additional data modification and analysis. He had trouble streamlining the data storage and retrieval operations, so he had to adjust Azure's settings to improve scalability and performance.

Izhaar concentrated on using the organized and cleansed data to develop and implement machine learning models for predictive analytics, experimenting with different methods to improve the precision of our forecasts. To provide resilience and reliability in predictions, he experimented with several model validation strategies due to the problem of model overfitting, especially with the Decision Tree algorithm.

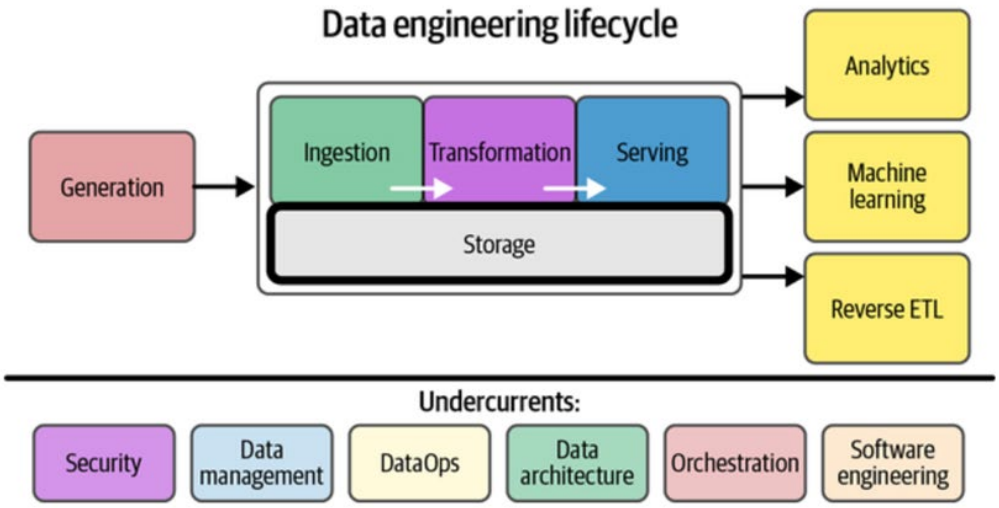
How did you collaborate with team members and divide project work?

We ensured that each team member could give their best work in their area of expertise by taking a rigorous and strategic approach to teamwork.

We created open lines of communication early on, holding frequent meetings and using common digital spaces, which made it possible for us to keep input and information flowing continuously throughout the project. The complexity of the responsibilities involved meant that this approach was necessary to ensure that everyone was on the same page and to coordinate our efforts. Using each person's unique qualities and technical abilities, the work was divided.

Every team member participated in the creation of the final report and presentation in addition to their assigned responsibilities. We were able to combine insights and discoveries cohesively through this collaborative process of document and presentation production, making sure that our collective knowledge was reflected in the final deliverables.

Explain the technical work needed to accomplish each stage of the lifecycle. This where the bulk of your writing should be. Think of this as documenting your project thoroughly. If someone came upon your system with no prior knowledge of it, how would you explain every detail to ensure they are experts by the end of this section?



Generation: We have considered two data sources, API, and sensor dataset. This dataset was taken from the Visual Crossing Weather website and was hosted on GitHub repository.

Ingestion: The data is pulled from the GitHub repository, unzipped, and utilized for further processes along with the data from the API. After cleaning and transforming the data, the data is loaded into the Azure Data Lake.

Transformation: Some unnecessary columns like sunrise timing, sunset columns, wind gust, description, station name (data source from which data is fetched, as it was not relevant for further predictions), were not adding value to the data. The columns having more than 60% data values as null were eliminated. Converting the datatype of 'datetime' from string to datetime to extract the year and month separately for data visualization.

Serving: The data will be served for training machine learning models to make accurate weather predictions and for visualizing the weather data, to analyse the patterns in weather changes over the coming months. Here, the focus is on extracting value from the dataset. Data's value lies in its practical use, making it crucial to apply advanced techniques. ML engineers play a vital role in this stage, leveraging cutting-edge methodologies. Extracting value from data is essential for informed decision-making. Unused data remains inert, highlighting the importance of utilization. Our objective is to unlock insights and drive actionable outcomes. This phase marks the culmination of our data engineering lifecycle. We aim to maximize the value of our project's dataset through effective utilization.

Storage: The data after transformation is stored in the data lake.

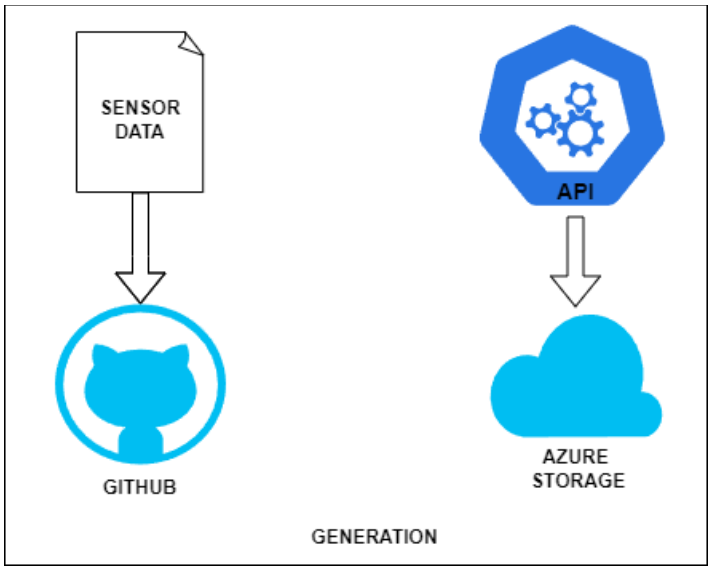
Analytics: For deriving analytical insights, we used Azure Synapse Analytics.

Machine Learning: Trained machine learning models such as **KNN**, Decision Tree, Naïve Bayes. The accuracy of KNN and Naïve Bayes was 84% and Decision Tree is 100% accurate as the data as the model is overfitting, with more data Decision Tree would make more accurate predictions.

Project Visualization

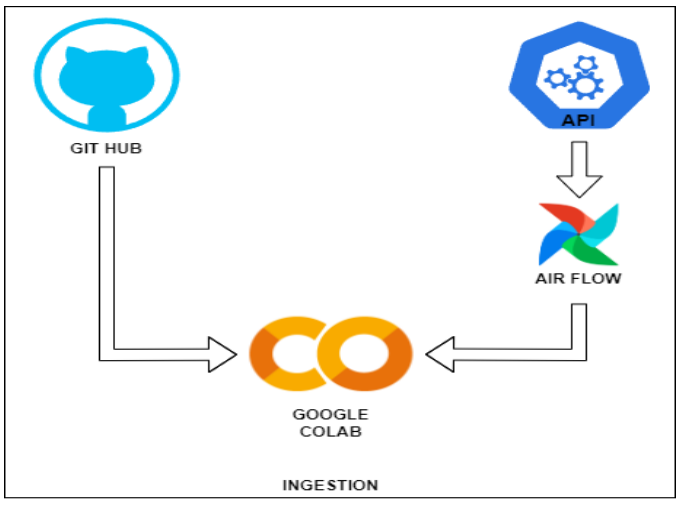
Please create various graphs to illustrate each component of the lifecycle to make it easy for a reader to see how you organized your data pipelines from source systems to destination systems.

GENERATION:



The data for this project is sourced from the Visual Crossing Weather Data website and is comprised of two distinct types: historical and API data. The historical weather data, encompassing six years of records collected from sensors, is stored in flat file format. These files are subsequently uploaded and maintained on GitHub for ease of access and version control. Conversely, API data is stored in the Azure Data Cloud, ensuring immediate availability and scalability for processing needs. This dual-source approach facilitates a comprehensive analysis of weather patterns by combining long-term historical insights with up-to-the-minute data observations.

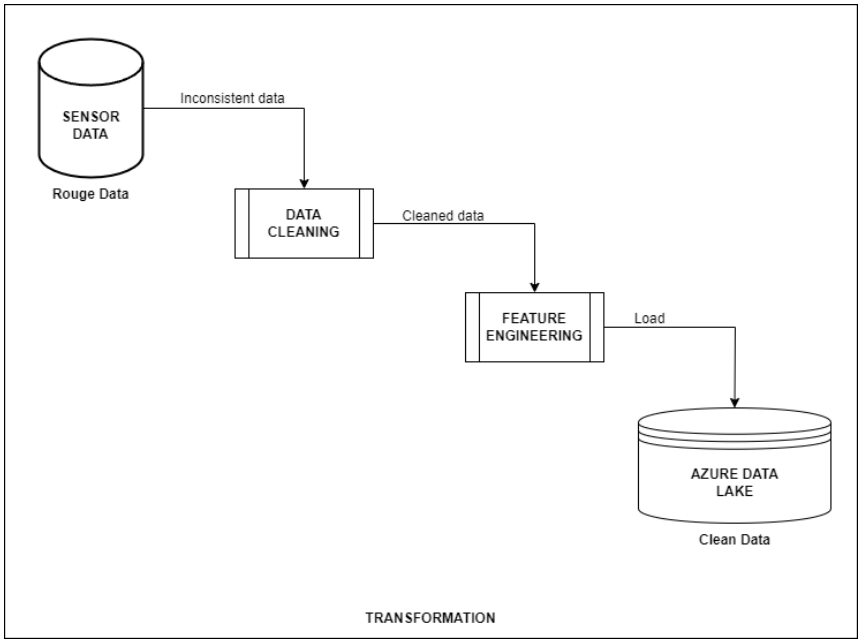
INGESTION:



In the Google Colab environment, data is retrieved from the GitHub repository to undergo a thorough inspection. This inspection involves examining how the data is organized, formatted, and stored, focusing on aspects such as tables, columns, rows, cells, and individual values. This process ensures that the data's structure meets the requirements for further analysis and processing.

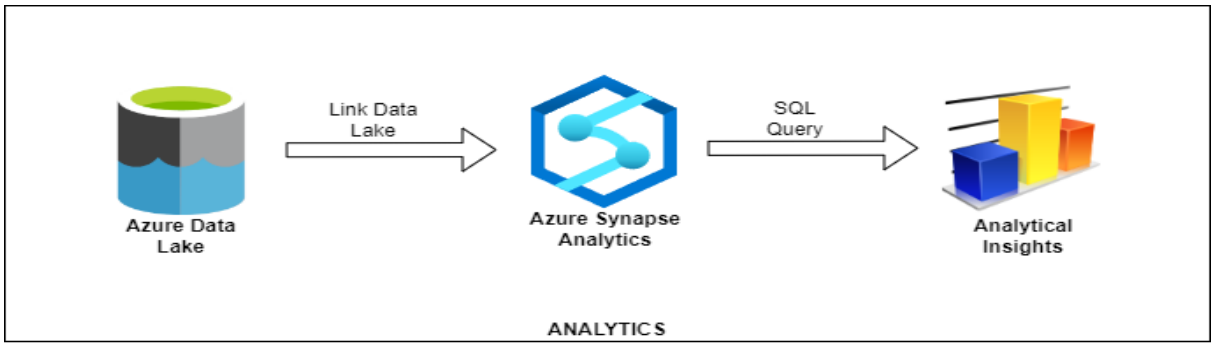
Additionally, Apache Airflow is utilized to manage the retrieval of real-time data via an API. This data is then stored as JSON files, a format conducive to both accessibility and further manipulation. This setup allows for efficient data flow management and automation of the data retrieval process, enhancing the pipeline's overall functionality and responsiveness.

TRANSFORMATION:



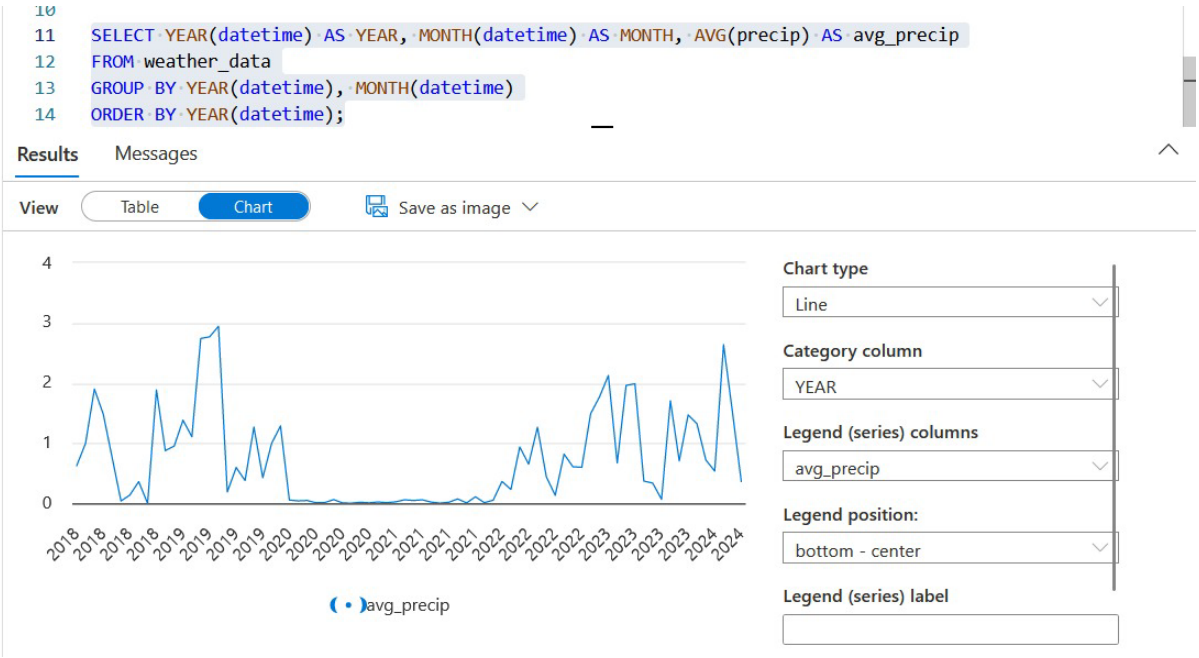
Within the Google Colab environment, the data underwent a meticulous transformation process to optimize its usability for further analysis. During the data cleaning phase, several modifications were made to streamline the dataset. Firstly, unnecessary columns such as sunrise and sunset timings, wind gust measurements, descriptions, and station names—which identified the sources of the data and were deemed irrelevant for predictive modelling—were removed. This step was crucial in decluttering the dataset and focusing on the variables that add value to the analysis. Secondly, any columns that had more than 60% of their data values missing were eliminated. This measure ensured the integrity and reliability of the analysis by maintaining a dataset robust enough for meaningful insights. Additionally, the ‘datetime’ column, initially formatted as a string, was converted into the datetime data type. This conversion was essential as it allowed for the separate extraction of year and month components, facilitating more detailed data visualization, and aiding in trend analysis. These transformations were vital for enhancing the dataset's accuracy and effectiveness in predictive analytics, ensuring the data was optimally prepared for the analytical objectives of the project.

ANALYTICS:



The data was pulled from the Azure Data Lake and then it underwent visualization processes within Synapse Analytics, where SQL queries were employed to extract insights such as year-wise average temperature and precipitation.

As depicted in the accompanying diagrams, these findings offer valuable insights into environmental trends and patterns, contributing to our project's overarching objectives.



```
SELECT YEAR(datetime) AS YEAR, MONTH(datetime) AS MONTH, AVG(tempmax) AS avg_tempmax
FROM weather_data
GROUP BY YEAR(datetime), MONTH(datetime)
ORDER BY YEAR(datetime);
```

Results Messages

View Table Chart Save as image

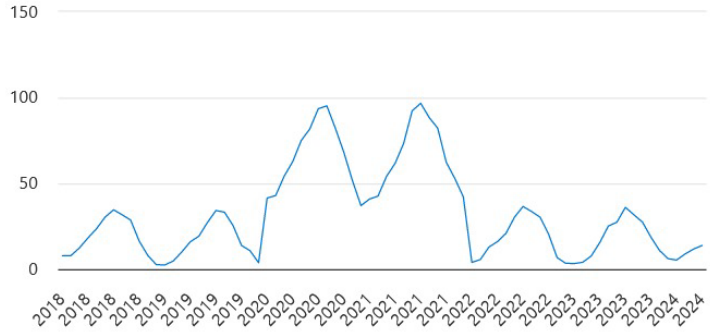


Chart type: Line

Category column: YEAR

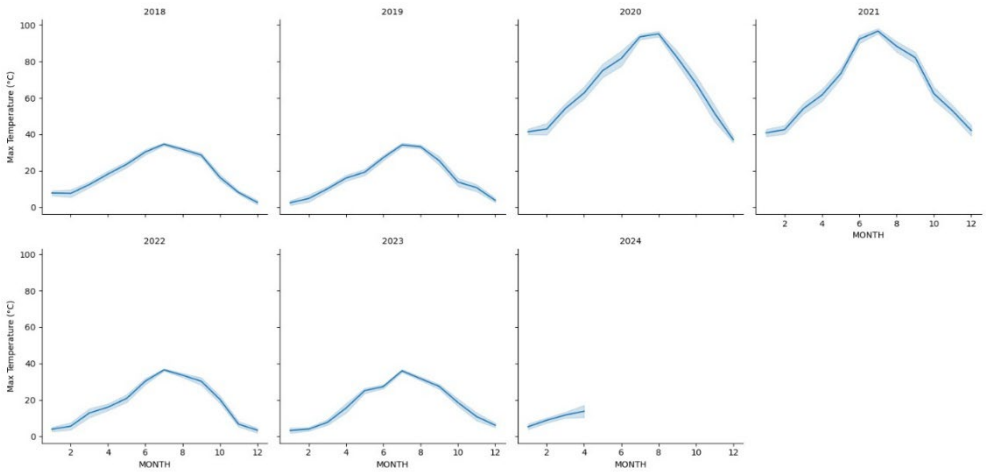
Legend (series) columns: avg_tempmax

Legend position: bottom - center

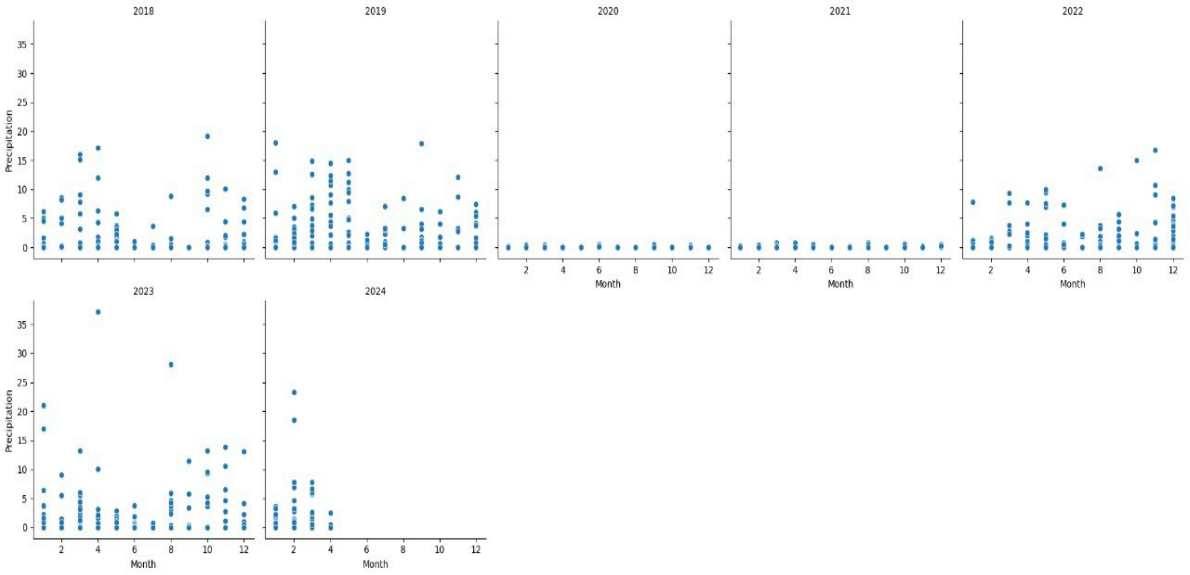
Legend (series) label

Within the Data Lake, we leveraged its functionalities to manage our data effectively. Specifically, we organized our data within a 'weather_data' container to streamline accessibility. Data was seamlessly loaded from Colab through a designated pipeline and deposited into the Azure Data Lake container.

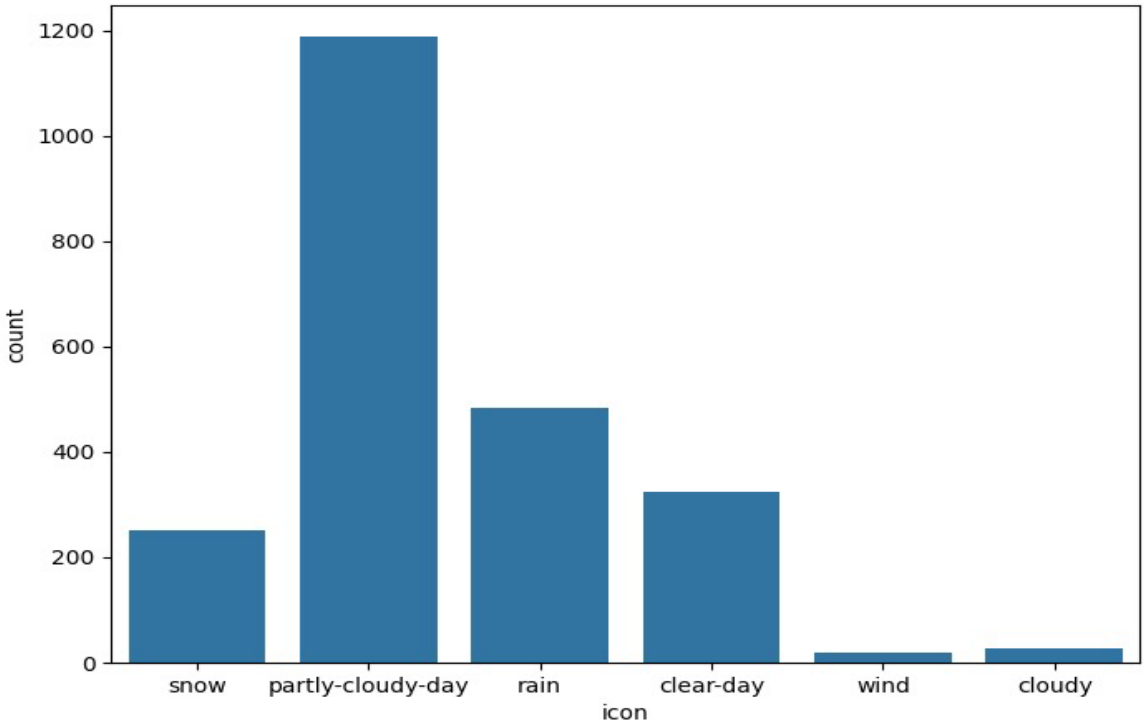
ANALYTICAL INSIGHTS



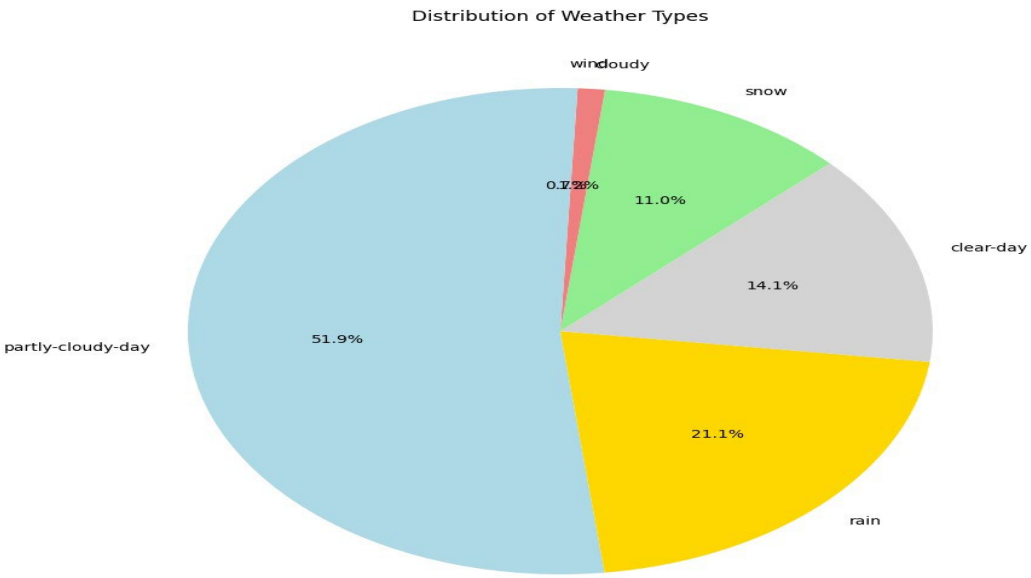
The above line graph showing the weather trends (maximum temperature) over the years starting from 2018 to 2024(till the month of April). As can be inferred, the maximum temperature is almost 100 degrees Fahrenheit in the years 2020 and 2021.



The above Scatter plot indicates the amount of precipitation across the years from 2018 to 2024 for all the months. Supporting the claim that the temperature was high in the years 2020 and 2021, from the weather trends shown earlier, the scatter plots also indicate that the amount of precipitation was less in the years 2020 and 2021.

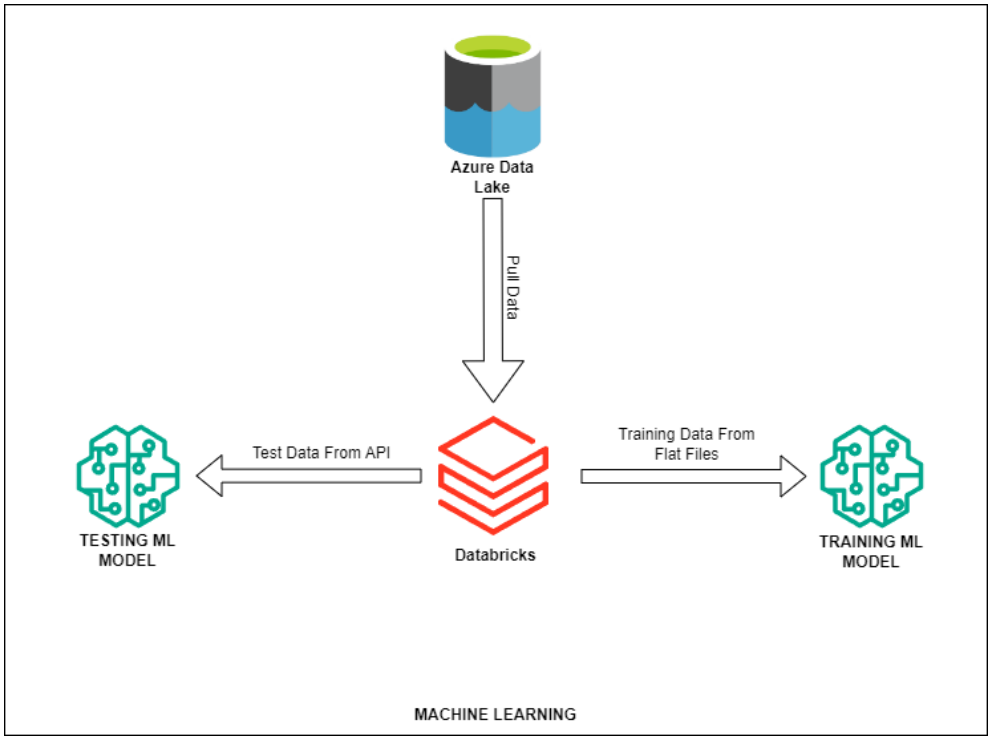


The above is a histogram for the different weather types.



This is a pie chart which indicates the distribution of weather types.

MACHINE LEARNING:



We handle our data analysis in the Databricks environment by retrieving data from an API and historical data saved in flat files from Azure storage. This technique aids in the efficient training of our machine learning models. Specifically, we train our models using historical data, and then we test them on API data to determine their performance. Three distinct machine learning algorithms—Naive Bayes, K-Nearest Neighbors (KNN), and Decision Tree—were

chosen to be tested for our project. We discovered that the Naive Bayes and KNN models both produced results with a strong accuracy of 86% after conducting our testing. The Decision Tree model stood out, though, with perfect accuracy of 100%. It suggests that there may be an issue with overfitting, where the model performs poorly on new data because to its excessive adherence to the training set. This means that, if we want to use the Decision Tree approach effectively, we will need to train it on a wider and more diverse set of data.

Project Retrospective Part I

How did your intended pipeline differ from the actual implementation?

We have retrieved the data from GitHub. Following extraction, the data undergoes a cleaning process and feature engineering. Once transformed, it is uploaded to the Azure Data Lake. We have employed the API-fetched data to test our machine-learning model and forecast future weather conditions. Using Databricks, the machine learning models are trained, and predictive analytics are conducted.

What did you learn along the way that you wish you would have known before?

Knowing a bit about Azure cloud services like Synapse Analytics and Databricks beforehand would have been handy. It would have helped us plan better and design our project to fit these platforms more smoothly. Understanding how to work with data in these tools from the start would have made things easier and improved our performance. Plus, knowing how to manage costs and resources effectively would have saved us some free credits in our Azure account. And staying updated on the latest features and tips would have kept us on top of our game. Overall, having some prior knowledge of Azure Synapse Analytics and Databricks would have made our project journey much smoother.

What were your biggest hurdles in the project?

We encountered several hurdles during the project. Firstly, the dataset obtained from the website contained numerous null values, making it challenging to determine which columns were essential for accurate predictions. Secondly, the data received from the API had limitations on historical data access. The free version of the API restricted both the number of calls allowed and the range of dates for fetching data without cost. These limitations increased the complexity of our data collection process and required us to plan to use our resources efficiently and carefully.

Would a streaming solution have been a better choice?

Implementing a streaming solution for your project means analysing environmental data as it comes in instead of waiting and processing it all at once. This real-time approach helps you quickly spot changes in ecological conditions and respond faster. It is like having a constant flow of information that lets you make decisions promptly. Streaming solutions are flexible

and can handle large amounts of data without slowing down, which is crucial for processing all the information from environmental sensors. Overall, using a streaming setup makes the project more agile, and better equipped to deal with the ever-changing nature of ecological data.

Project Retrospective Part II:

Finally, if you were taking an advanced data engineering course What would you learn more to help you accomplish the project more effectively next time?

In addition, gaining hands-on experience with these cloud platforms would have provided us with practical skills in setting up data storage, processing pipelines, and deploying applications, which are essential for modern data engineering projects. Furthermore, understanding the fundamentals of Machine Learning Algorithms would have empowered us to explore advanced analytics techniques, such as predictive modelling and classification, to derive deeper insights from our data. Overall, a solid foundation in cloud services and machine learning would have significantly streamlined our project development process and enhanced the quality of our deliverables.

What were your most important and critical takeaways from the project?

The project dived deep into using Azure's cloud platform to store data after changing it. We needed to know how to handle data from various places like APIs and flat files. A big part of the project was using tools like Airflow to organize tasks and Azure to store data. This helped us manage data well and make sure everything went Smoothly. In the end, we used Azure effectively to handle data and meet our project goals.

Pretend for a moment that you are in front of a review committee who is ready to give you more funding to hire new team members or technical architecture solutions. What would you ask for to be more successful in a future project? Be humble and honest with the team and tools you would ask for.

I request funding for team expansion and technical solutions to enhance our project's success. Hiring specialized team members in data engineering and analysis will augment our capabilities. Secondly, investing in DOMO and Alteryx software will streamline our ETL processes. DOMO offers intuitive data integration and visualization tools, fostering collaboration and informed decision-making. Meanwhile, Alteryx simplifies data preparation and offers advanced analytics, empowering predictive modeling and spatial analysis. These tools and team enhancements will boost efficiency, improve data quality, and accelerate our project outcomes. With your support, we can ensure our projects have a more prosperous and impactful future.

Conclusion

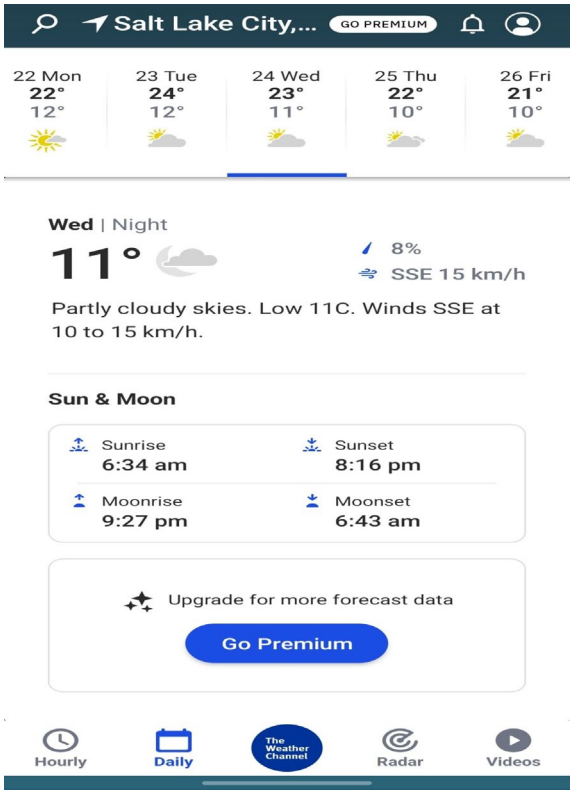
```
if prediction[0] in unique_values_df['icon_encoded']:
  print(unique_values_df[unique_values_df['icon_encoded']==prediction[0]]['icon'].iloc[0])
```

```
partly-cloudy-day
```

```
Just now (<1s)                                                    17                    Python

print(date)
```

```
2024-04-24
```



As can be seen, the weather predicted by our machine learning model aligns with the prediction from The Weather Channel App. Both show ‘partly cloudy’ weather forecasts.