# Data 226 Project Report
# Weather API Real-time & Historical Dashboards

Group 6: Lam Tran, Khac Minh Dai Vo, Matthew Leffler, Shao-Yu Huang, Yilin Sun
Department of Applied Data Science, San Jose State University

GitHub Link: https://github.com/matthewleffler1/DATA226_PROJECT

## I. Abstract

This project presents an end-to-end weather analytics pipeline that transforms raw weather API data into meaningful insights through data engineering, visualization, and simple predictions. Historical and current weather data—including temperature, wind speed, humidity, UV index, rainfall, and cloud coverage, etc.—were programmatically collected using a weather API and stored in Snowflake. After data ingestion (ETL), transformation processes (ELT) were applied to compute relevant averages and metrics. The processed data was then visualized in Tableau and Power BI, resulting in a multi-layered dashboard that enables dynamic metric selection, location-based comparisons, and trend analysis through both interactive maps and historical line charts. In the final stage, machine learning models were implemented to predict key weather indicators based on temporal and spatial patterns. This pipeline demonstrates a scalable, modular approach to building weather intelligence systems for monitoring, reporting, and forecasting.

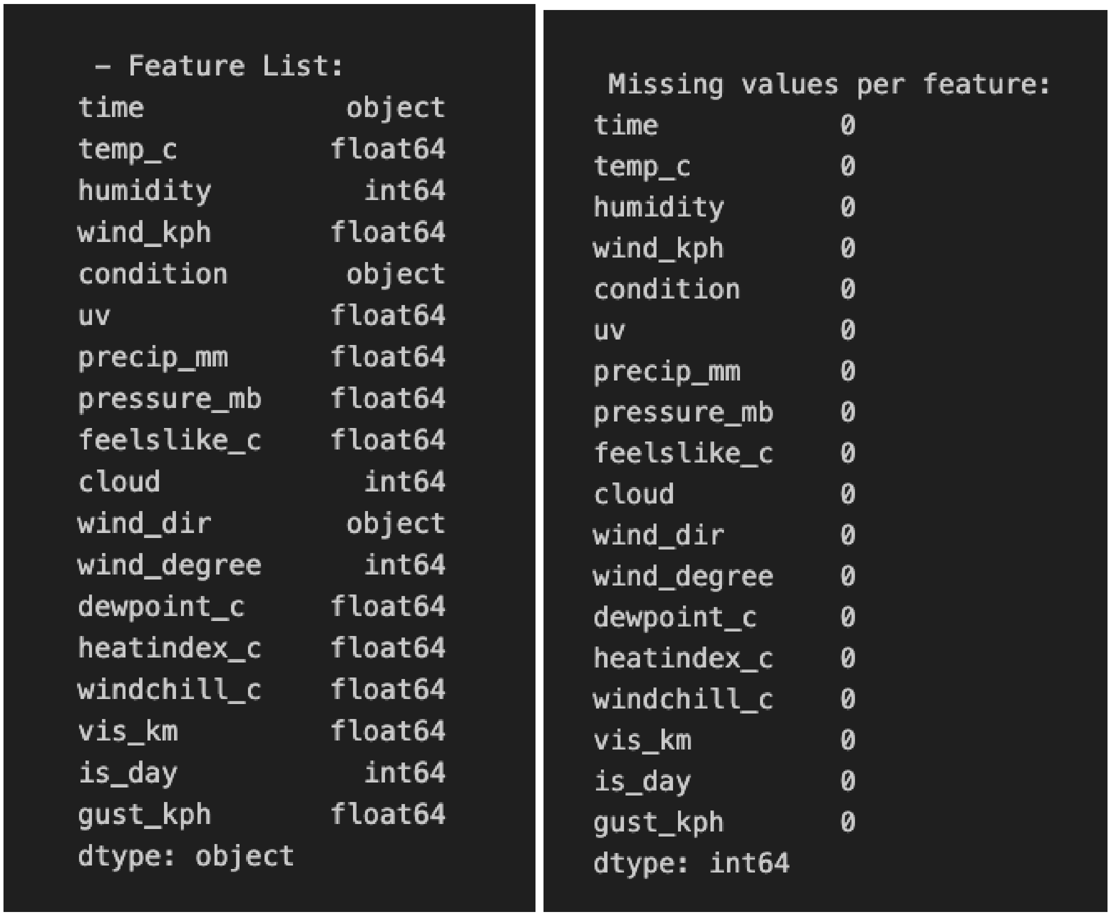
## II. Introduction

Accurate and accessible weather data plays a critical role in sectors ranging from agriculture and transportation to public safety and energy management. However, raw weather data, especially when collected frequently and at a high resolution, can be difficult to process and interpret without a structured data pipeline and intuitive visual interfaces. This project addresses that challenge by building a full-stack weather analytics system that begins with raw data extraction and culminates in actionable insights.

The pipeline starts with automated extraction of weather metrics from a public weather API, which provides hourly measurements for multiple cities in California. This data is ingested into Snowflake, where transformations are performed to compute rolling averages to summarize trends across time. Visualizations were built in Tableau and Power BI to allow end users to explore these trends interactively via map dashboards and time-series charts. A parameterized interface enables users to toggle between metrics and analyze location-specific patterns. Lastly, predictive modeling techniques were employed to forecast future weather metrics using historical data, offering a data-driven foundation for anticipating shifts in environmental conditions. This project demonstrates how cloud-based warehousing, modern BI tools, and machine learning can be combined to derive value from real-time environmental data.

## III. Dataset Description

The dataset contains 18 weather-related features with non-missing values, and some of them show strong relationships while others are more independent. For example, temperature (temp_c) is very strongly correlated with feels-like temperature, heat index, and wind chill, since all of them describe how hot or cold it feels. Similarly, UV index tends to increase with temperature and daylight, but decreases when humidity or cloud cover is high. On the other hand, features like precipitation (precip_mm) and atmospheric pressure (pressure_mb) show weak or no correlation with most other features, meaning they vary more independently. This mix of correlated and uncorrelated variables makes the dataset useful for both understanding weather patterns and building predictive models.



```
 - Feature List:
time              object
temp_c           float64
humidity           int64
wind_kph         float64
condition         object
uv               float64
precip_mm        float64
pressure_mb      float64
feelslike_c      float64
cloud              int64
wind_dir          object
wind_degree        int64
dewpoint_c       float64
heatindex_c      float64
windchill_c      float64
vis_km           float64
is_day             int64
gust_kph         float64
dtype: object
```

```
 Missing values per feature:
time           0
temp_c         0
humidity       0
wind_kph       0
condition      0
uv             0
precip_mm      0
pressure_mb    0
feelslike_c    0
cloud          0
wind_dir       0
wind_degree    0
dewpoint_c     0
heatindex_c    0
windchill_c    0
vis_km         0
is_day         0
gust_kph       0
dtype: int64
```

*Fig. 1. Features Available in Dataset*

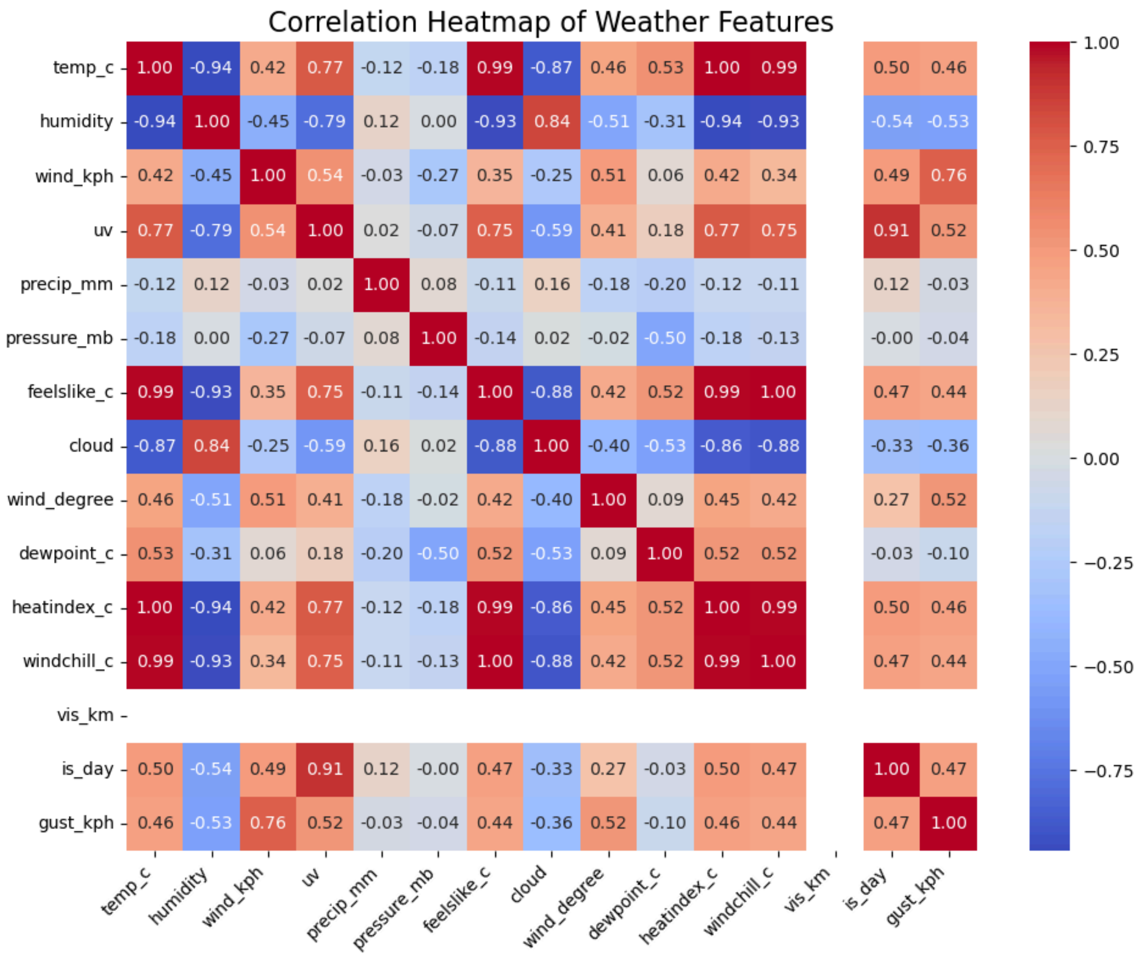


*Fig. 2. Correlation Heatmap of Dataset Features*

## IV. System Architecture

While each component of the system (ETL, ELT, and BI/Visualizations) use the same underlying dataset to extract the appropriate information, each step uses its own distinct tools to complete the pipeline's task.

### *A. ETL*

For the ETL step, we will be extracting raw weather data using Weather API in conjunction with the Geolocation in order to extract both weather data as well as geographic data about inputted cities. In order to store the information extracted from the API, we will be using Snowflake as the data warehouse to store the raw data for later abstractions to be made. While this process can be done manually, inconsistency in running the pipeline can be an issue for manual inputs, so we will be using Apache Airflow to automatically run the task on an hourly basis in order to extract the most recent data available from the API

### *B. ELT*

While the Weather API offers many niche features such as tidal information and moon-phase information which may impact weather conditions, we are only interested in creating a weather dashboard to compare conditions from city to city, so we will be using dbt to not only run SQL queries against our raw data to filter and create abstractions of the data, but it will also be used to ensure data quality checks to ensure BI tools don't run into data quality issues downstream. To do so, the ELT process will gather the raw data from Snowflake, run data quality tests and SQL queries against the data via dbt, and will save the results back into Snowflake. This process will be automated by once again having Apache Airflow run the ELT dag, which runs the dbt commands for us users to automate the dbt tasks.

### *C. BI/Visualizations*

Now that the data is filtered and cleaned, the data will be loaded into Tableau and Power BI in order to create visualizations to both allow for quick understanding of the data, as well as bridge the gap between the technical (data team) side and the casual viewer (business side). Unfortunately, due to licensing changes, Tableau's free version no longer offers the Snowflake connector, so the data is manually loaded from Snowflake into Tableau, and since this process is required for Tableau, **it can manually be loaded into Power BI as well.** In this case, we will be using Tableau Public in order to plot up-to-date geographical data offered from the weather API (using the latitude and longitude features), and a real-time dashboard will be created to summarize the weather conditions using Power BI.

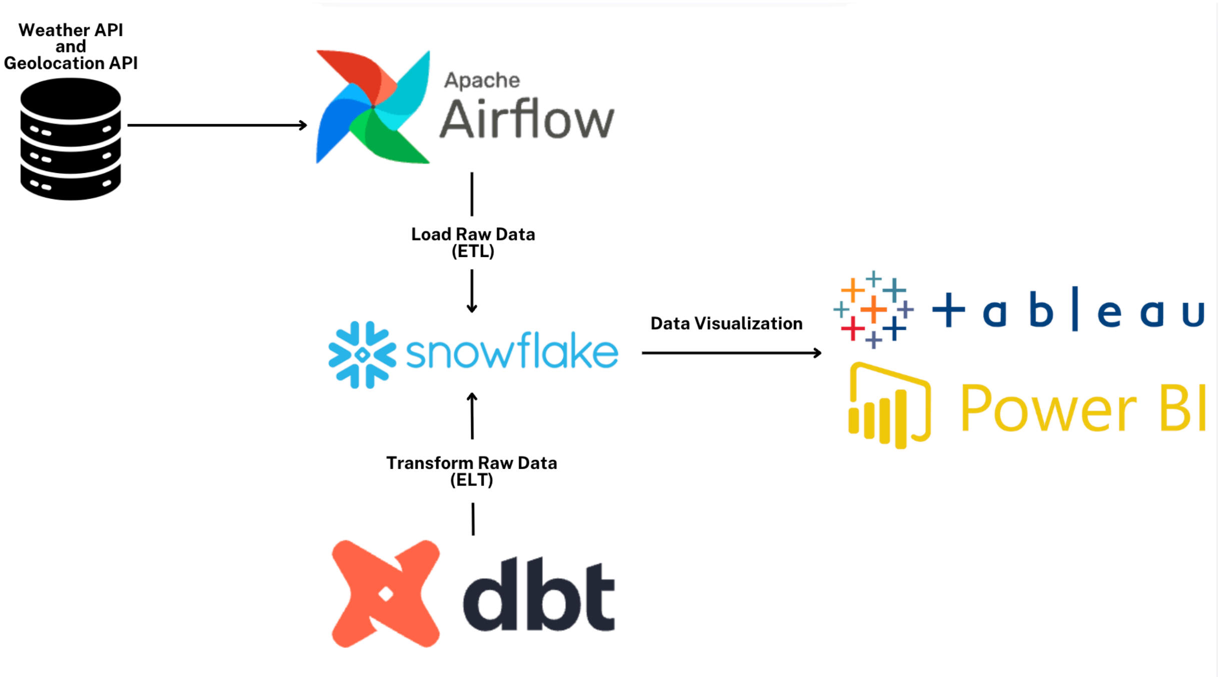


*Fig. 3. System Design Architecture*

## V. DB Schema

In terms of establishing a schema for the tables, we will first be performing light transformations within the ETL process, then will do heavier filtering and quality checks using dbt. This process allows for quick debugging, because by checking the data quality and tasks using Airflow's Web UI, we can easily pinpoint the tasks which cause the pipeline to go awry. As seen in Figure 4, we will have five DAGs running in conjunction with one another. The first DAG will perform ETL in order to gather raw data relating to San Jose only for dashboarding purposes, the second will gather information about various zip codes which allow a more general sense of weather information around all of California, and the final DAG will be performing the ELT, which checks for data acting in an expected fashion.

### *A. San Jose Weather Data*

The first DAG we created will be responsible for extracting insights about San Jose weather specifically.

*1) ETL:* Unlike the DAG which gathers information about various different cities around California in real-time, this DAG operates in a fashion such that it summarizes data throughout the day, and offers historical tracking to see trends in San Jose's weather. In this case, since a single city is being examined, the geographical information offered by the dataset is not as important as the other dag, so we will simply filter out all the features excluding localtime, cloudiness, humidity, precipitation in inches, temperature in Fahrenheit, UV index, and the wind speed in MPH. We extract this statistical data for downstream operations, which create prediction models and manual labeling along with simple prediction models to determine whether the weather is viable for sports or not.

*2) ELT:* In the case of preparing data for dashboarding, we create abstractions on the data in order to calculate rolling averages to see weather changes over the duration of the dataset. In addition to creating averages based on the statistical information, we will keep the raw data for additional use, so we can perform normal time-series analysis on the timestamp without grouping columns together. In this case, since we are seeing weather changes at different times throughout the day, once the data is loaded into the dataframe, we don't want that value to be duplicated and inputted again. Therefore, we will ensure dbt ensures the local time column is unique.

### *B. Zip Code Weather Data*

*1) ETL:* As for the ETL process related to the weather based on imputed zip codes, we extract a larger amount of information from the Weather API data than the San Jose Specific data. This is because in this case, the geographical distance between the measurements poses a much more important factor in skewing the statistical information of the weather. In this case, not only do we extract the same information as the San Jose weather data, but also the wind direction, region, and geographical information for geographical plotting of the data.

*2) ELT:* When preparing the data for the BI/Visualizations team, it is important for non-null data points to be present in order to ensure graphs don't contain gaps within the time series data. Therefore, dbt will be used to ensure both non-null elements only are inputted into the Analytics schema, as well as that the values fall within correct ranges (i.e. non-negative wind speeds, precipitation amount, etc.)

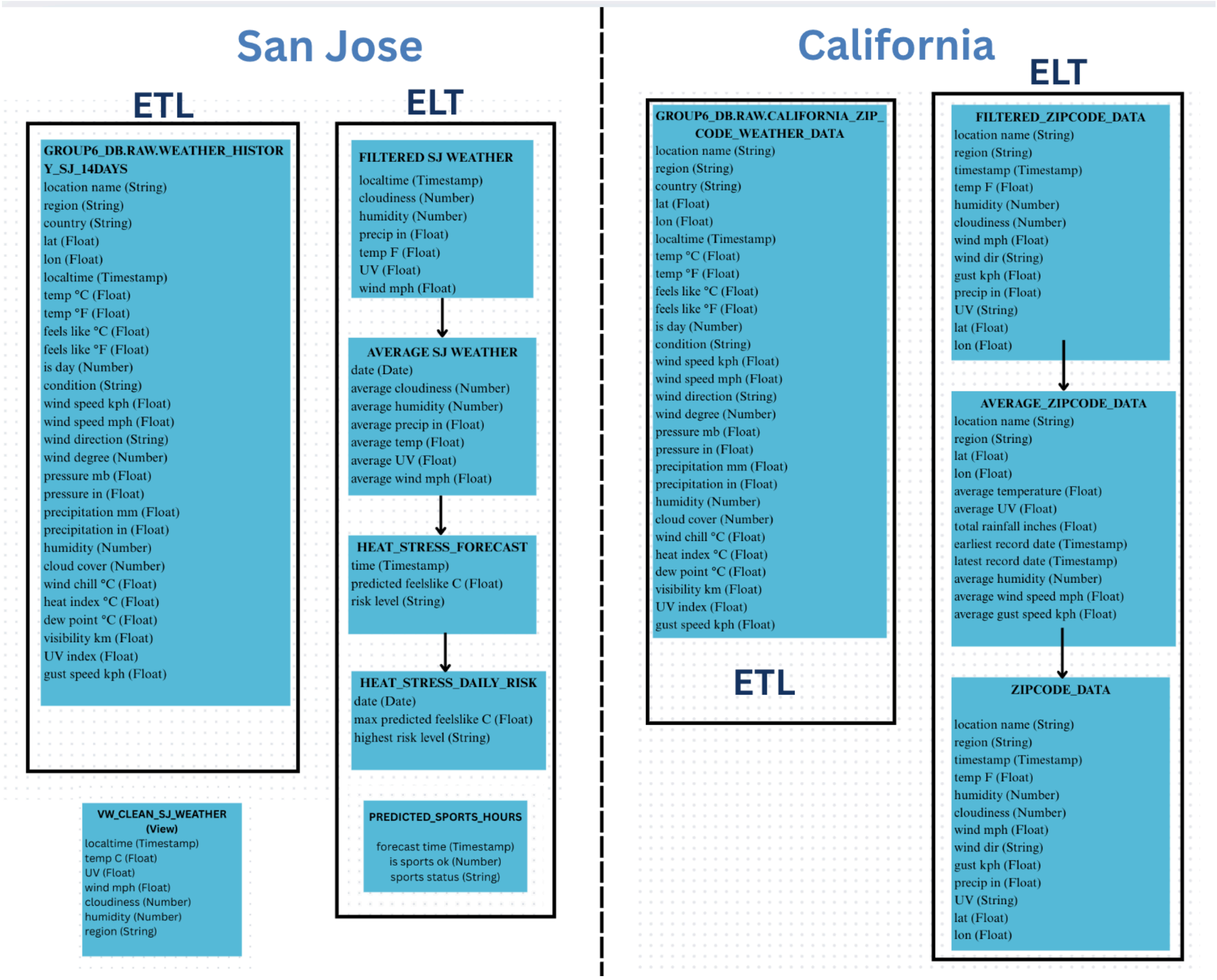


*Fig. 4. System Design Architecture*

For the ETL step, we will be extracting raw weather data using Weather API in conjunction with the Geolocation in order to extract both weather data as well as geographic data about inputted cities. In order to store the information extracted from the API, we will be using Snowflake as the data warehouse to store the raw data for later abstractions to be made. While this process can be done manually, inconsistency in running the pipeline can be an issue for manual inputs, so we will be using Apache Airflow to automatically run the task on an hourly basis in order to extract the most recent data available from the API.

## VI. Data Pipelines

### *A. ETL Process*

This section outlines the Extract, Transform, and Load (ETL) processes implemented for collecting, processing, and storing weather data using Apache Airflow. Two Airflow Directed Acyclic Graphs (DAGs) were developed to handle different types of weather data: one for collecting current weather conditions from various cities across California, and another for retrieving historical hourly weather data specifically for San Jose. The processed data is loaded into Snowflake for further analysis and visualization.

*1) Current Weather Data for California Zip Codes:* The first ETL pipeline, defined in the weather_ca_top10 DAG, is designed to fetch and store real-time weather data from multiple popular zip code locations across California. The pipeline is scheduled to run hourly and targets a Snowflake table named GROUP6_DB.RAW.CALIFORNIA_ZIP_CODE_WEATHER_DATA.

a) *Extraction:* The extraction phase involves calling the WeatherAPI's /current.json endpoint using a list of predefined ZIP codes that represent a geographically diverse set of popular locations in California. These ZIP codes are hardcoded into the script to cover coastal, inland, northern, and southern regions. For each ZIP code, an HTTP request is sent to the API, and the JSON response is collected.

*b) Transformation:* The raw data is parsed to extract essential weather attributes such as temperature, humidity, pressure, wind speed and direction, UV index, visibility, and local time. In addition to these standard features, optional metrics like wind chill, dew point, and heat index are also included if available. The data is flattened into a list of dictionaries, making it suitable for relational database storage.

c) *Loading*: In the loading phase, the pipeline first ensures the target table exists in Snowflake. It uses CREATE TABLE IF NOT EXISTS logic to define the schema. To maintain data integrity and idempotency, any existing records for the same location and timestamp (localtime) are deleted before inserting the new values. This prevents duplicate entries and ensures that the latest observation is always stored.

*2) Historical Hourly Weather Data for San Jose:* The second ETL pipeline is the running DAG of the weather historical data of San Jose in the past 14 days. The pipeline is scheduled to run daily and automatically populates GROUP6_DB.RAWEATHER_HISTORY_SJ_14DAYS table inside Snowflake.

a) *Extraction:* The extraction task queries the WeatherAPI's /history.json endpoint in a loop that goes back 14 days from the current date. Each API call fetches a single day’s worth of hourly data. The task handles errors gracefully, logging any issues encountered during data retrieval.

b) *Transformation*: Each day’s response is parsed to extract hourly weather measurements. These include timestamped weather conditions such as temperature, humidity, wind attributes, precipitation, and other atmospheric indicators. All hourly records are structured into a flat format, similar to the real-time pipeline, ensuring consistency across datasets.

*c) Loading:* The transformed hourly records are loaded into Snowflake using a similar approach to the current weather ETL. The target table is created if it does not exist. For each hourly entry, any existing record with the same location_name and localtime is first deleted before inserting the new one. This design guarantees that data updates do not result in duplicate entries and ensures consistent historical tracking.

*3) Conclusion:* Both DAGs implement a clear and efficient ETL workflow that supports high-quality, structured weather data ingestion. By integrating external API data with Snowflake and orchestrating the process via Airflow, this setup ensures scalability, reliability, and ease of maintenance. These pipelines enable robust downstream analytics and visualization capabilities for both real-time and historical weather trends in California.

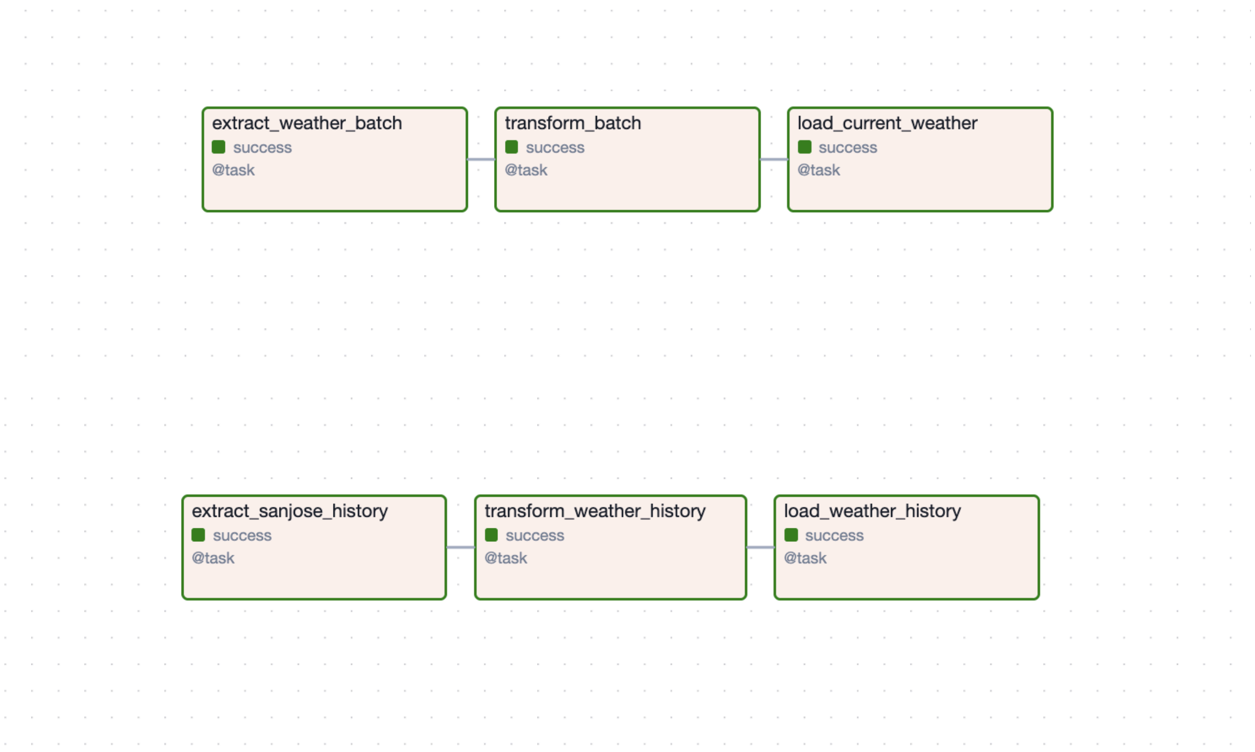


*Fig. 5. ETL Dag Graphs*

### *B. ELT Process*

This section outlines the Extract, Load, Transform (ELT) process using dbt to perform data quality checks (ensuring non-null values, and values fall into proper ranges), create snapshots for data retrieval in case the pipeline breaks, and run SQL against the raw data to create abstractions in an automated fashion using Apache Airflow.Within the ELT dag, we will be running the “dbt build” command, which will enable us to use a single dbt command to run dbt run, dbt test, and dbt snapshot all in one command (in that order).

*1) Extract:* In the first step of the ELT pipeline, we will be extracting the raw data which was stored within Snowflake during the ETL process. When running dbt build, the first dbt models to run are located in the input folder, which run SQL commands on the data to perform both filters, as well as abstractions using group by’s and averages of the weather statistics. Second, it will test established conditions for the data, such as checking for nulls, non-negatives, etc, and lastly, the command will create a snapshot of the processed data.

a) *Run Input Models:* Two input models run SQL against the raw data. The first is the raw_sj_weather_data.sql, and the second is the raw_zipcode_weather_data.sql file. As for the San Jose weather model, the only thing the model performs is filtering for only the columns used downstream. The model extracts the localtime, TEMP_F, WIND_MPH, PRECIP_IN, HUMIDITY, renames CLOUD to CLOUDINESS (since this is a percentage of cloud coverage in the area), and the UV index. In addition, the zip code data only performs filtering in this step as well, extracting all of the same rows, including the LOCATION_NAME (city name), LAT, LON, CONDITION, and GUST_KPH columns. The reasoning for these additional columns being added is that San Jose is a single location, while the cities were carefully selected to span a wider area of California. Therefore, we can perform analysis on temperature differences depending on the region, in contrast to only gathering data points about San Jose, where the weather is often similar across the city.

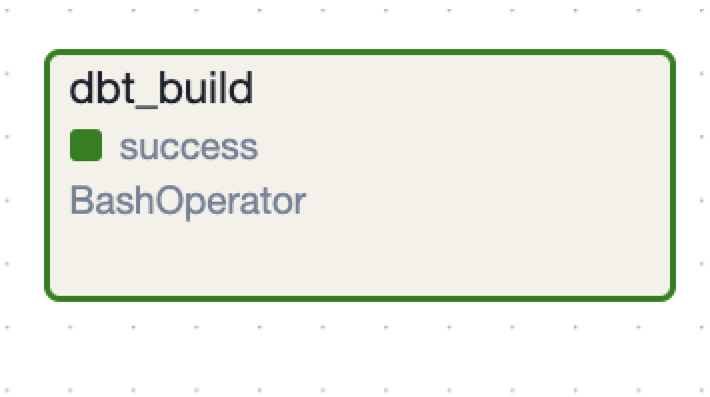
b) *Test for Data Integrity*: While testing the data integrity is done after all the models are run, the tests are still performed on the data produced by the input model. In this case, we will leverage dbt in order to perform a series of tests, including checking for all the columns to ensure non-null values. For the filtered San Jose weather data, we ensure the localtime column is both non-null, as well as unique so it can function as a primary key. We want to ensure that each timestamp has its own individual datapoint to prevent overlapping data points in the visualizations. Additionally, we ensure measurements such as wind speed, precipitation in inches, humidity, cloudiness, and UV index have a minimum value of 0, because an amount lower than that is impossible, suggesting data integrity was compromised upstream. Lastly, we ensure cloudiness is in a range between 0 and 100, as the measurement is a percentage of clouds in the sky bounded by that range. In terms of the zip code data, we perform more-or-less the same data integrity checks (non-null, and minimum value of 0 for measurements alongside a range for cloudiness).

*2) Transform:* During the transformation phase of the ELT pipeline, we will create abstractions based on the results of the input models which calculate day-to-day averages alongside passing simple filtered values along to the output for normal time-series analysis.

a) *Run Output Models:* While only two models existed to filter the results of the raw data, four output models exist to perform further abstractions and transformations on the filtered data. In regards to the San Jose weather data, the first file, average_sj_weather.sql casts the timestamp as a date, then calculates an ongoing average of all the data stored related to San Jose weather within Snowflake. This allows us to compare average weather statistics on a day-by-day basis to see weather trends over a larger time frame without scoping in too far. The second file relating to San Jose's weather simply returns all of the results of the input, as this allows for comparing time-series data within an individual day without narrowing the scope. As for the zip code data, the models theoretically work nearly identically to the San Jose models; however, we add a new column to the dataframe. For further analysis of weather trends throughout California, we wanted to test whether or not the latitude (north or south) of the city's location had an impact on whether weather conditions made an impact on the results. Therefore, we calculated the northernmost and southernmost latitudes of California, divided the state into thirds, and split each city into categories of Northern California, Southern California, and Central California based on the results. Additionally, for scalability purposes, we saved the earliest and latest weather recording date within the table. This is to ensure skewed results don't appear, especially with columns such as the sum of precipitation in inches being in the dataframe, as a city keeping track of that data longer will likely accumulate rainfall overtime more rapidly than a newly added point.

b) *Test for Data Integrity*: While it may seem like overkill to run the same non-null and value tests mentioned in the previous section on the abstracted data, the process helps in determining where along the pipeline errors are occurring for fast debugging. Therefore, we perform the same data quality checks as above, with the addition of testing valid values for the REGION column containing a value of either "Northern California," "Central California," or "Southern California."

*3) Load:* After dbt build runs the code and tests for data quality, we load two different tables into the database. The first is new, abstracted tables which get loaded with the same names as their SQL files (which was set by the dbt project folder), and the second is the one created by dbt snapshot, which keeps a running track of edits made to the tables, and dates in the data are valid. This allows for both tracking when data has gone bad, as well as offering a bootleg version of Snowflake's Time Travel feature, allowing the old data to replace old records in the database before it was changed to a noisy value.



dbt_build

🟩 success

BashOperator

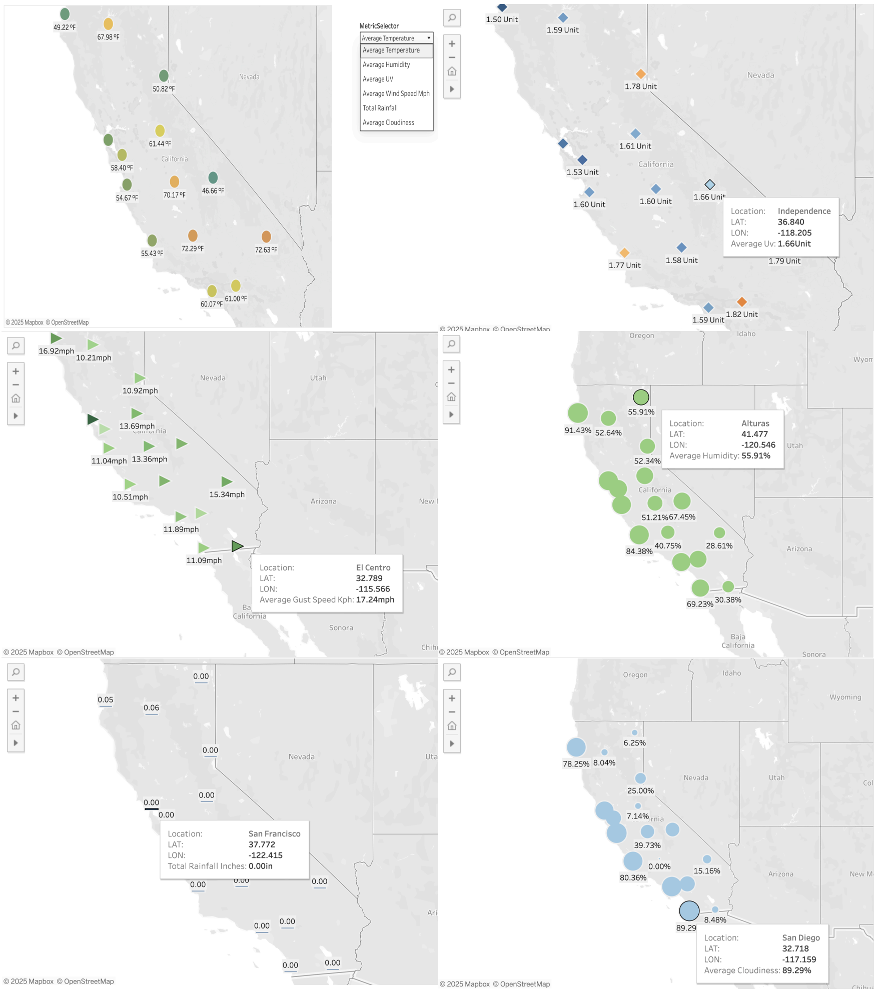
*Fig. 5. ELT DAG Graph*

## VII. Business Intelligence/Visualizations

### *A. Real-Time Map Dashboard*

We developed a map visualization of six key weather metrics, temperature, humidity, UV Index, wind speed, rainfall, and cloudiness from California's top 14 cities. Built using Tableau and powered by weather data retrieved from the table AVERAGE_ZIPCODE_DATA, the dashboard allows users to explore spatial patterns in weather conditions by selecting different metrics via a unified parameter dropdown. Each metric is visualized using a chart style best suited to its nature: color-coded circles for temperature and humidity, directional wind icons for gust speed, density shading for cloud coverage, and vertical Gantt-style bars for rainfall accumulation.

The dashboard is designed with clarity and usability in mind. Only the currently selected metric is shown, while others remain hidden. Hovering over each location reveals detailed tooltips with values rounded to appropriate decimal places and labeled with units (e.g., °F, %, mph), along with contextual information such as city names and coordinates. This tool is especially useful for comparative regional analysis or monitoring localized weather trends, making it an effective interface for both technical users and general audiences interested in understanding environmental patterns.



Location: Independence
LAT: 36.840
LON: -118.205
Average Uv: 1.66Unit

Location: El Centro
LAT: 32.789
LON: -115.566
Average Gust Speed Kph: 17.24mph

Location: Alturas
LAT: 41.477
LON: -120.546
Average Humidity: 55.91%

Location: San Francisco
LAT: 37.772
LON: -122.415
Total Rainfall Inches: 0.00in

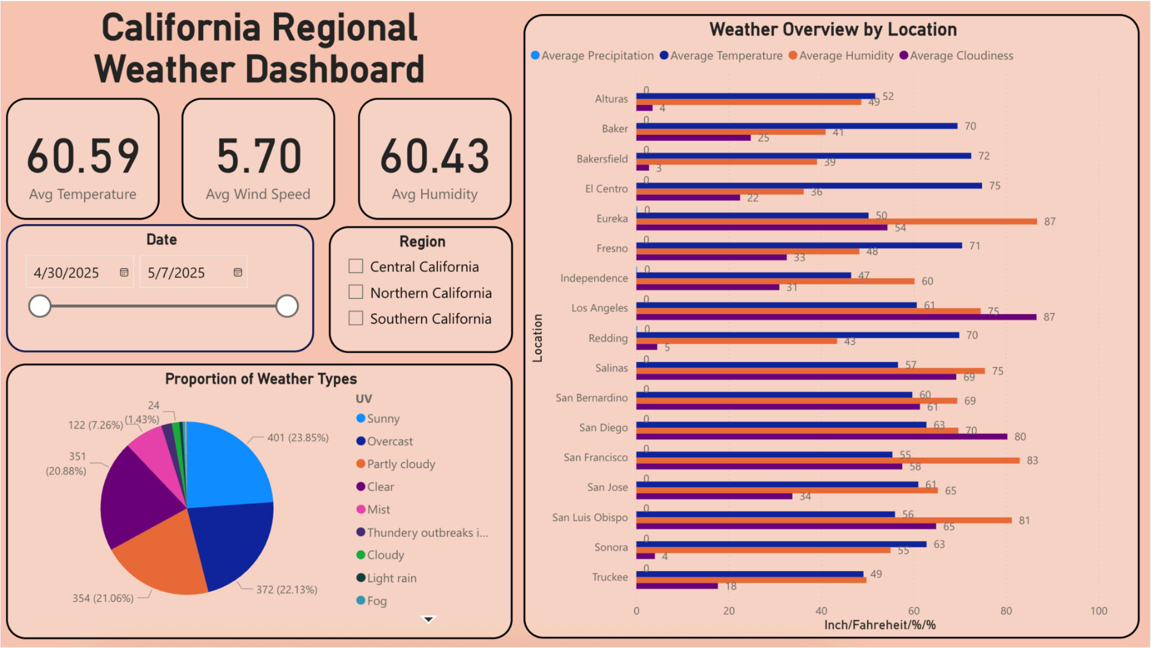
Location: San Diego
LAT: 32.718
LON: -117.159
Average Cloudiness: 89.29%

*Fig. 6. California Weather Statistics Visualizations*

## *B. Historical Dashboards*

We developed a comprehensive set of interactive dashboards using Power BI to provide actionable insights from integrated weather data across California. These dashboards combine historical trends, forecast analysis, and sports suitability prediction to support informed decision-making for outdoor activities, public health, and environmental planning. The dashboard suite consists of three interconnected components:

1) *California Regional Weather Dashboard:* This component enables region-wise comparisons across Central, Northern, and Southern California. Users can filter by region and date range to explore localized differences. A clustered bar chart visualizes average temperature, humidity, cloudiness, and precipitation across major California cities, providing insight into regional microclimate diversity. A pie chart presents the UV index distribution by weather condition, allowing users to assess solar exposure risks based on cloud cover and UV trends in different zones.



California Regional Weather Dashboard

60.59 Avg Temperature

5.70 Avg Wind Speed

60.43 Avg Humidity

Date

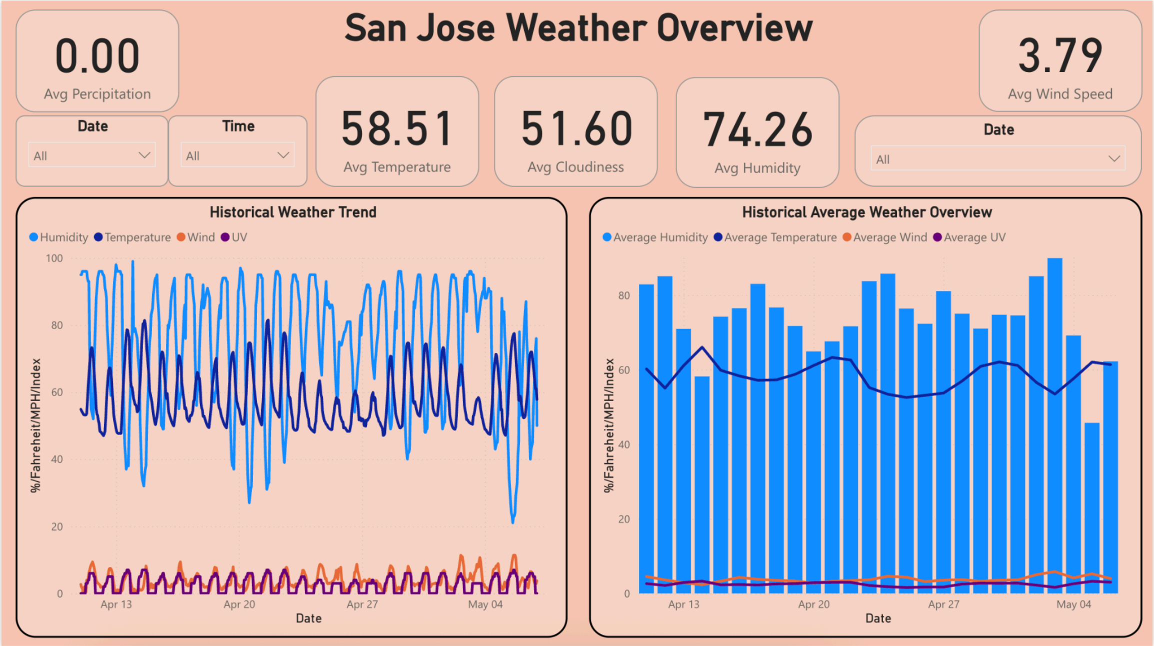
4/30/2025 – 5/7/2025

Region

□ Central California
□ Northern California
□ Southern California

*Fig. 7. California Cities Dashboard*

2) *14 Day San Jose Weather Overview:* This dashboard focuses on historical and near-real-time weather conditions in San Jose. Key metrics such as average temperature (°F), humidity (%), cloudiness (%), wind speed (MPH), and UV (Index) are summarized via KPI cards. A multi-line chart presents a 14-day weather trend, tracking temperature, humidity, wind speed, and cloudiness fluctuations. A bar and line combo chart also provides an average daily weather overview, including temperature, humidity, and UV index, helping users evaluate short-term weather consistency and health-related risk factors.


San Jose Weather Overview

0.00 Avg Percipitation

3.79 Avg Wind Speed

58.51 Avg Temperature

51.60 Avg Cloudiness

74.26 Avg Humidity

Date: All

Time: All

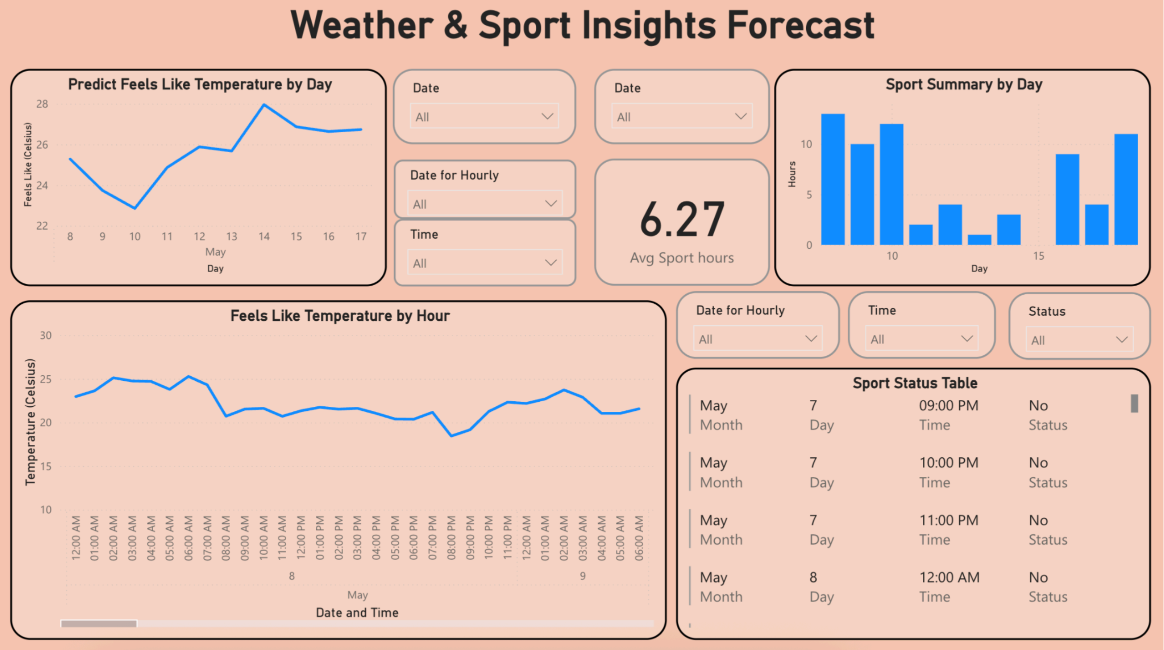
Date: All

*Fig. 8. San Jose Weather Dashboard*

3) *14 Day Sorts Suitability Forecast Dashboard:* The final dashboard emphasizes weather-driven support for physical activity planning. Forecasted "feels like" temperature data provides both daily and hourly trends, enabling users to identify the most suitable time windows for outdoor sports. Key features include:

- A line chart of predicted daily feels-like temperature
- A detailed hourly temperature trend
- A bar chart summarizing the number of suitable hours per day (based on a weather-derived sports suitability flag)
- A KPI card calculating the average daily sport-OK hours
- A data table listing hourly sports suitability status

These visualizations are supplemented with dynamic date, hour, and status slicers, allowing for targeted exploration and personalized planning.



Weather & Sport Insights Forecast

Date: All

Date: All

Date for Hourly: All

Time: All

6.27
Avg Sport hours

Date for Hourly: All

Time: All

Status: All

Sport Status Table

| | | | |
|---|---|---|---|
| May<br>Month | 7<br>Day | 09:00 PM<br>Time | No<br>Status |
| May<br>Month | 7<br>Day | 10:00 PM<br>Time | No<br>Status |
| May<br>Month | 7<br>Day | 11:00 PM<br>Time | No<br>Status |
| May<br>Month | 8<br>Day | 12:00 AM<br>Time | No<br>Status |

*Fig. 9. Weather and Sport Insights Forecast Dashboard*

## VIII. Analysis

### *A. 14-Day Weather Trends Across California*

Our visualized weather data analysis reveals distinct climate characteristics across various California regions. From the 14-day trend data, we observed consistent temperature and UV patterns in San Jose, with minor humidity and wind speed fluctuations, indicating stable local weather conditions. The average weather overview chart further confirmed that UV exposure generally correlated with sunnier days, while higher cloudiness and humidity aligned with lower UV readings. Regionally, the data highlighted notable differences: Central California tends to experience higher temperatures and lower humidity, while Northern regions showed more precipitation and cloud cover. Southern California, in contrast, displayed relatively balanced weather indicators but slightly elevated UV levels. These insights suggest potential applications in public safety (e.g., UV advisories), infrastructure planning (e.g., anticipating weather impacts), and even tourism (e.g., identifying more favorable regions for travel). Integrating real-time ZIP code weather data also allows for near-instant assessment of conditions, supporting reactive decision-making. Our analysis shows that combining historical and real-time data yields a rich, actionable understanding of regional climate dynamics.

### *B. Heat Stress Risk Forecast Prediction*

This section analyzes the predicted heat stress risk in San Jose over a 14-day period, based on hourly weather forecasts. The purpose is to determine when outdoor conditions might feel uncomfortably hot or pose a potential health risk, even if raw temperatures appear normal. To make this assessment, a machine learning model was used to predict the "feels like" temperature, a more accurate measure of human-perceived heat that accounts for environmental factors like humidity and wind.

The model used in this system is a Random Forest Regressor, stored in a file named heat_model.pkl. This model was selected because it handles non-linear relationships effectively, which is important when predicting feels-like temperature (target feature). It is influenced by multiple interacting weather variables such as temperature, humidity, UV index, cloud cover, wind speed, and gusts (other features - inputs). Random Forest is also robust to noise and outliers, as it works by averaging predictions from many decision trees, reducing the likelihood of overfitting. This makes it a suitable choice for real-world data, which can be messy or incomplete. Additionally, because the training dataset was synthetic and based on a limited two-week period, Random Forest is ideal for producing stable and reliable predictions without needing a large amount of historical data.

After predicting the hourly “feels like” temperature, each value was classified into a heat stress risk category (ECMWF, 2024):

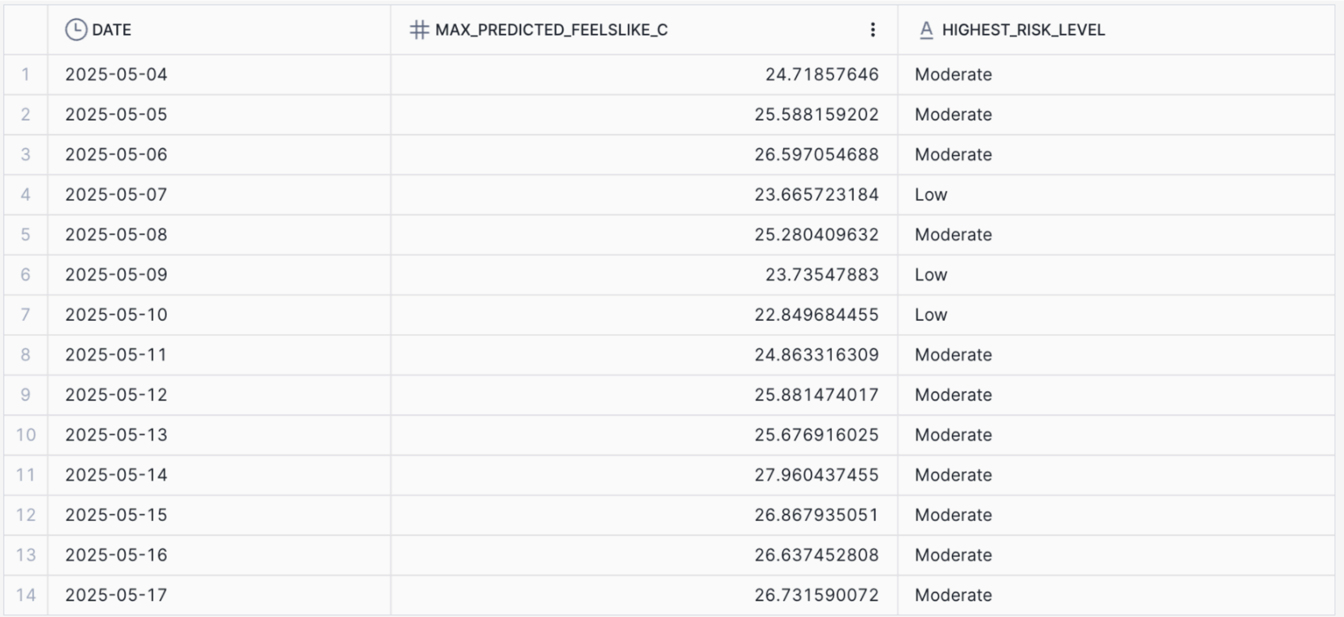
1) **Low**: Less than 24°C
2) **Moderate**: Between 24°C and 32°
3) **High**: 32°C or more

During the forecasted period, the result tells that no hours were classified as High risk, suggesting generally mild weather for early May. The predictions were stored in two Snowflake tables: one containing the hourly Heat Stress Forecast table, and another summarizing the maximum daily heat risk table.

The data shows that most hours were categorized as Low risk. However, several days experienced Moderate risk periods, typically around midday to late afternoon. These include May 6, May 14, and May 17, which saw peak “feels like” temperatures above 26°C. Conditions during these times were often marked by high UV exposure and lower wind speeds, both of which can intensify heat perception so it is not recommended to do outside activities or stay outside without protection for a long time.

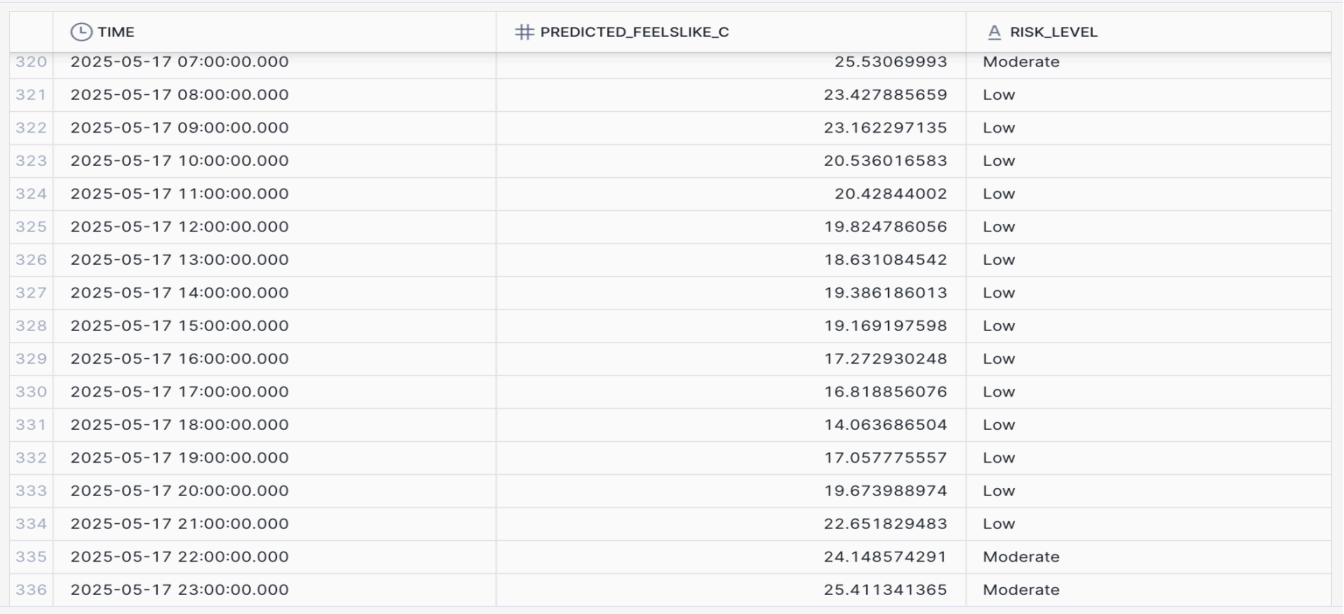
Out of the 14 forecasted days, 10 showed Moderate as the highest daily risk level, while only 4 days (May 7, 9, 10, and 15) remained consistently in the Low range. This highlights the importance of analyzing hourly variations, as daily averages can miss short but significant heat stress periods.

This forecasting system can potentially support a variety of practical applications. Public health organizations can issue heat advisories in advance, employers can adjust outdoor work schedules, and local agencies can plan cooling resources for high-risk hours. By integrating machine learning with real-time weather forecasts, this approach provides a more accurate and actionable way to monitor and manage heat-related risks



| | DATE | MAX_PREDICTED_FEELSLIKE_C | HIGHEST_RISK_LEVEL |
|---|---|---|---|
| 1 | 2025-05-04 | 24.71857646 | Moderate |
| 2 | 2025-05-05 | 25.588159202 | Moderate |
| 3 | 2025-05-06 | 26.597054688 | Moderate |
| 4 | 2025-05-07 | 23.665723184 | Low |
| 5 | 2025-05-08 | 25.280409632 | Moderate |
| 6 | 2025-05-09 | 23.73547883 | Low |
| 7 | 2025-05-10 | 22.849684455 | Low |
| 8 | 2025-05-11 | 24.863316309 | Moderate |
| 9 | 2025-05-12 | 25.881474017 | Moderate |
| 10 | 2025-05-13 | 25.676916025 | Moderate |
| 11 | 2025-05-14 | 27.960437455 | Moderate |
| 12 | 2025-05-15 | 26.867935051 | Moderate |
| 13 | 2025-05-16 | 26.637452808 | Moderate |
| 14 | 2025-05-17 | 26.731590072 | Moderate |

*Fig. 10. Daily Heat Stress Risk Level*



|  | TIME | PREDICTED_FEELSLIKE_C | RISK_LEVEL |
|---|---|---|---|
| 320 | 2025-05-17 07:00:00.000 | 25.53069993 | Moderate |
| 321 | 2025-05-17 08:00:00.000 | 23.427885659 | Low |
| 322 | 2025-05-17 09:00:00.000 | 23.162297135 | Low |
| 323 | 2025-05-17 10:00:00.000 | 20.536016583 | Low |
| 324 | 2025-05-17 11:00:00.000 | 20.42844002 | Low |
| 325 | 2025-05-17 12:00:00.000 | 19.824786056 | Low |
| 326 | 2025-05-17 13:00:00.000 | 18.631084542 | Low |
| 327 | 2025-05-17 14:00:00.000 | 19.386186013 | Low |
| 328 | 2025-05-17 15:00:00.000 | 19.169197598 | Low |
| 329 | 2025-05-17 16:00:00.000 | 17.272930248 | Low |
| 330 | 2025-05-17 17:00:00.000 | 16.818856076 | Low |
| 331 | 2025-05-17 18:00:00.000 | 14.063686504 | Low |
| 332 | 2025-05-17 19:00:00.000 | 17.057775557 | Low |
| 333 | 2025-05-17 20:00:00.000 | 19.673988974 | Low |
| 334 | 2025-05-17 21:00:00.000 | 22.651829483 | Low |
| 335 | 2025-05-17 22:00:00.000 | 24.148574291 | Moderate |
| 336 | 2025-05-17 23:00:00.000 | 25.411341365 | Moderate |

*Fig. 11. Daily Heat Stress Forecast*

### *C. Sports Suitability Forecast Analysis*

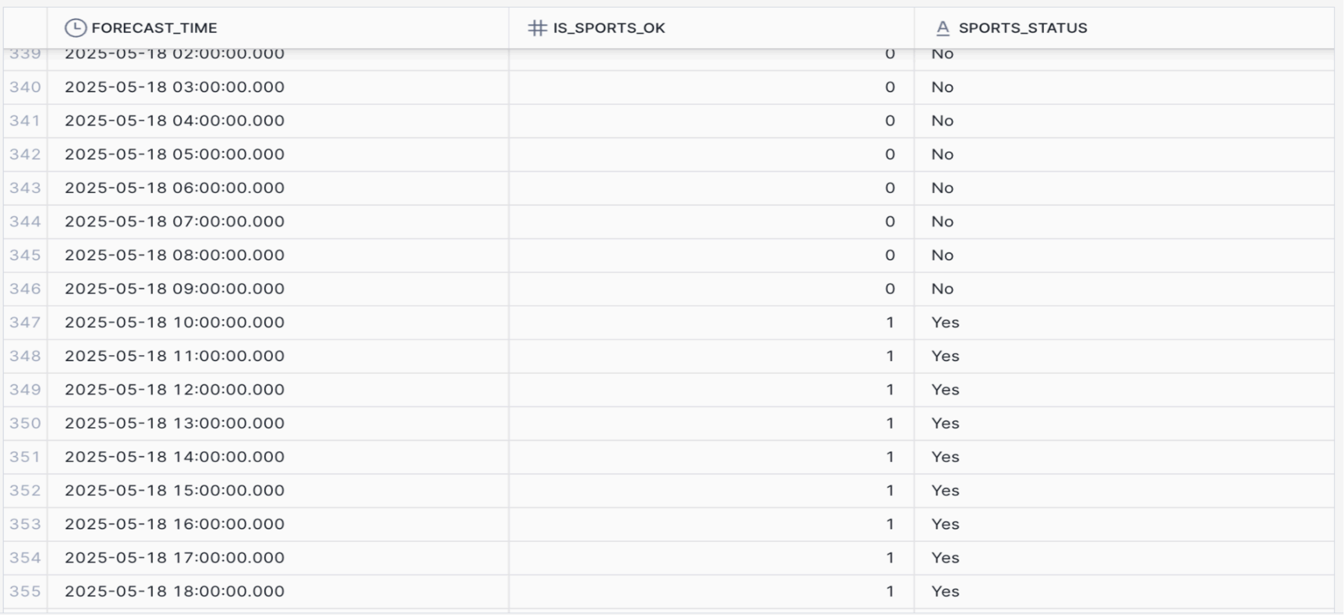
This section presents an analysis of weather-based predictions for sports suitability in San Jose over a 14-day forecast period. The objective is to identify hours during which outdoor sports and physical activities are expected to be safe and comfortable, based on current weather conditions. The predictions were generated through a rule-based system implemented in an Airflow DAG Sports Hour Prediction python code and the data was fetched from forecast JSON file that was provided by the WeatherAPI. This ensures the reliability of this analysis. The system also applies a set of environmental criteria based on scientific information to hourly weather forecast data, classifying each hour as either suitable or unsuitable for sports outside activities.

The classification is determined by several factors that directly influence physical comfort and safety. An hour is marked as suitable for sports if the temperature falls between 15°C and 28°C (CDC, 2023; Athletics Canada, 2023), the humidity level is below 80% (CDC, 2023), wind speed does not exceed 15 kilometers per hour (NWS, 2023), cloud cover is under 70% (Windy App, 2023), the UV index remains below 8 (EPA, 2023), visibility is at least 8 kilometers (Athletics Canada, 2023), and the hour occurs during daylight (Athletics Canada, 2023). These thresholds were selected to represent moderate and safe weather conditions that support physical activity without significant risk of overheating, sunburn, or reduced visibility.

Once the forecast data is classified, the results are stored in the Snowflake table PREDICTED_SPORTS_HOURS. Each record includes the hour of the day, whether it meets the suitability criteria, and a corresponding status label ("Yes" or "No"). To provide a higher-level overview, the system also aggregates this data daily and stores the results in SPORTS_DAILY_SUMMARY. This summary records the total number of hours per day that are deemed suitable for sports and identifies the best day within the forecast period, basically the day with the highest count of suitable hours.

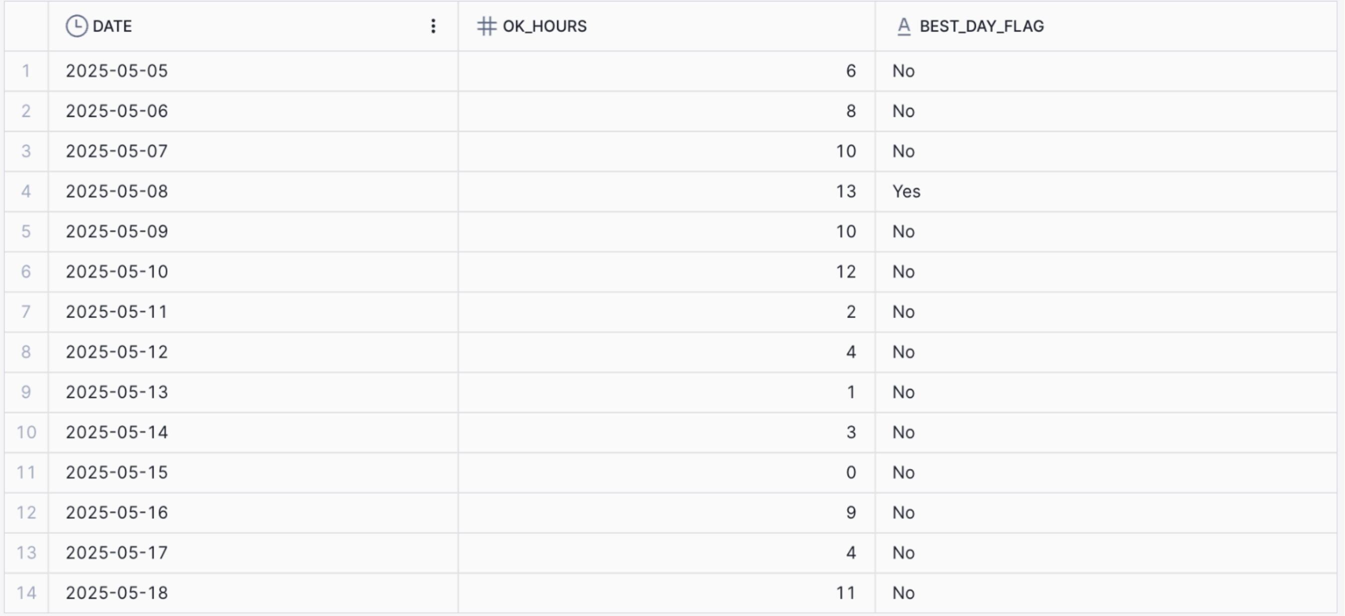
The evaluation of the results shows distinct daily patterns. On most days, suitable conditions for sports begin to appear in the late morning and persist until early evening which makes sense because it is consistent with natural heating patterns and daylight availability. Early morning and nighttime hours are generally not suitable due factors such lower temperatures, limited light, and often higher humidity. The data for May 8 illustrates the most favorable conditions during the period, with a total of 13 suitable hours. This day stands out as the most optimal for outdoor sports, being listed as the “best day” in the summary table. Other days with extended favorable windows include May 10 and May 18, which recorded 12 and 11 suitable hours, respectively. May 7 and May 9 also performed well, each showing 10 hours marked as safe for sports activities. In contrast, May 15 exhibited no suitable hours, indicating consistently poor weather conditions throughout the entire day. This variation highlights the sensitivity of outdoor suitability to even minor fluctuations in temperature, wind, cloud cover, and UV exposure.

The system provides a useful tool for guiding outdoor sports planning. By delivering hour-by-hour forecasts and aggregating daily suitability metrics, the prediction pipeline supports informed scheduling decisions for athletes, coaches, recreation planners, and the general public. Although the approach is based on hard-coded environmental thresholds rather than machine learning, it offers a transparent and responsive method for identifying optimal weather windows. The inclusion of multiple weather variables support the accuracy of the predictions, ensuring that recommendations reflect more than just raw temperature readings.



| | FORECAST_TIME | IS_SPORTS_OK | SPORTS_STATUS |
|---|---|---|---|
| 339 | 2025-05-18 02:00:00.000 | 0 | No |
| 340 | 2025-05-18 03:00:00.000 | 0 | No |
| 341 | 2025-05-18 04:00:00.000 | 0 | No |
| 342 | 2025-05-18 05:00:00.000 | 0 | No |
| 343 | 2025-05-18 06:00:00.000 | 0 | No |
| 344 | 2025-05-18 07:00:00.000 | 0 | No |
| 345 | 2025-05-18 08:00:00.000 | 0 | No |
| 346 | 2025-05-18 09:00:00.000 | 0 | No |
| 347 | 2025-05-18 10:00:00.000 | 1 | Yes |
| 348 | 2025-05-18 11:00:00.000 | 1 | Yes |
| 349 | 2025-05-18 12:00:00.000 | 1 | Yes |
| 350 | 2025-05-18 13:00:00.000 | 1 | Yes |
| 351 | 2025-05-18 14:00:00.000 | 1 | Yes |
| 352 | 2025-05-18 15:00:00.000 | 1 | Yes |
| 353 | 2025-05-18 16:00:00.000 | 1 | Yes |
| 354 | 2025-05-18 17:00:00.000 | 1 | Yes |
| 355 | 2025-05-18 18:00:00.000 | 1 | Yes |

*Fig. 12. Sports Viability Prediction*



|  | DATE | OK_HOURS | BEST_DAY_FLAG |
|---|---|---|---|
| 1 | 2025-05-05 | 6 | No |
| 2 | 2025-05-06 | 8 | No |
| 3 | 2025-05-07 | 10 | No |
| 4 | 2025-05-08 | 13 | Yes |
| 5 | 2025-05-09 | 10 | No |
| 6 | 2025-05-10 | 12 | No |
| 7 | 2025-05-11 | 2 | No |
| 8 | 2025-05-12 | 4 | No |
| 9 | 2025-05-13 | 1 | No |
| 10 | 2025-05-14 | 3 | No |
| 11 | 2025-05-15 | 0 | No |
| 12 | 2025-05-16 | 9 | No |
| 13 | 2025-05-17 | 4 | No |
| 14 | 2025-05-18 | 11 | No |

*Fig. 13. Sporty Viability Summary*

## IX. Conclusion

While it is time-consuming gathering weather information about more niche topics in such a timely manner, we showed that a simple ETL + ELT pipeline can not only gather the necessary data in a timely manner, but keep a real-time account of the weather conditions and predictions as well. Through this simple pipeline, we were able to gain insight into how the geography of different regions play a role in affecting local weather within California, and successfully implemented ML models to predict if the conditions are fine for outside activities.

### *A. Limitations*

While meaningful analysis was obtainable through the use of such tools and data engineering techniques, several issues are within the realm of possibility to approach with available resources.

*1) Accessing More Historical Data:* As per the limitations of the free-tier offered by weather API, users only have access to a year's worth of free data for the first 14 days, then, historical data is limited to 7 days. This leads to a large limitation on weather predictions, because such few data points are taken into consideration, and especially when predicting information about forecasts for sports, this is not enough time to take information about season trends into account, limiting accuracy of models. Access to more data can lead to accessing better model predictions.

*2) Cross-Referencing Weather APIs:* In some situations, it appears the weather API rates cloudiness on a bizarre scale, and seems to sometimes assign a boolean value to cloudiness instead of a percentage to cloudiness. Additionally, when cross referencing weather reports with precipitation and humidity amounts, there appears to be a conflict in results. Therefore, assembling multiple APIs may be deemed better for analytical purposes, but this brings up another data accessibility problem if those results are not publicly available.

*3) Snowflake Storage:* While Snowflake gives $400 worth of credit in their free-tier, we quickly noticed this would be a problem with running these pipelines as either a long-term project, or when trying to run the DAGS on an interval quicker than hourly. As of now, we have been running the pipeline for about a week, and $200 credits were already used. Therefore, a more accessible or premium storage warehouse would improve the longevity of the project.

### *B. What's Next*

In addition to fixing the above limitations of the project, there are several other implementations that can both improve the prediction model, as well as give more insightful information about weather trends around California.

1) *Implementing Other APIs:* While this project was quite self-contained (only using the Weather API alongside the Geographical API), the project can be expanded to perform analysis on information such as rain-out for sports, or flight delays due to weather conditions. With access to sports forecasts or an API offering flight information, we can potentially detect the weather conditions that cause flights and games to be delayed or cancelled due to weather conditions being deemed unsafe.

2) *Clustering Based on Similar Weather Patterns:* In order to gain a further understanding of how geography plays a role in dictating the weather, we can perform analysis on the terrain around the cities, and group cities into clusters based on either the landscape, elevation, or similar weather conditions. An example of how the landscape changes weather conditions was evident in the geographical chart, specifically the region in southern California. San Diego, a coastal city, constantly had drastically different weather conditions than El Centro, which although they are close distance-wise, El Centro is more towards the desert.

3) *Expanding Beyond California:* Naturally, now that the infrastructure of the pipeline is working and producing accurate results, the next step would be to scale out, and either span more of California's area, or expand beyond the borders of the state. With this expansion, different geographical considerations need to be taken into account, and further analysis into latitude and longitude may play a wider role when expanding to different regions.

## References

[1] “Heat stress: what is it and how is it measured? | Copernicus,” *climate.copernicus.eu*, May 22, 2024. https://climate.copernicus.eu/heat-stress-what-it-and-how-it-measured

[2] CDC, “About Heat and Your Health,” *Heat Health*, Jun. 25, 2024. https://www.cdc.gov/heat-health/about/index.html?CDC_AA_refVal=https%3A%2F%2Fwww.cdc.gov%2Fextreme-heat%2Fprevention%2Findex.html

[3] US EPA,OAR, “UV Index Scale | US EPA,” *US EPA*, May 24, 2019. https://www.epa.gov/sunsafety/uv-index-scale-0

[4] N. US Department of Commerce, “Estimating Wind,” *www.weather.gov*. https://www.weather.gov/pqr/wind

[5] Rayn, “What Wind Speed is Too Much for Pickleball? Discover the Limits!,” *Pickleball Fact*, Jan. 03, 2024. https://pickleballfact.com/what-wind-speed-is-too-much-for-pickleball/ (accessed May 07, 2025).

[6] “What is cloud cover and how else do we measure clouds,” *WINDY.APP*. https://windy.app/blog/what-is-cloud-cover.html

[7] “Extreme Weather Guidelines.” Available: https://athletics.ca/wp-content/uploads/2023/09/AC-Extreme-Weather-Guidelines_as-of-September-2023.pdf