Team Information:

Team Name: Data Wizards

Team Members: K. Thrishank: API Integration, Backend Logic

K. Surya Teja: Documentation & Testing

R.V. Deekshitha: AI model training

M. Gagan Sai: Data Collection, Hyperparameter Tuning

S. Charith: UI/UX & Frontend Development

Contact Email: A22126552018 KARRITHRISHANK

1. Selected API & Data Collection

(a) API Name: Weather API

(b) API Endpoint Used: https://api.open-meteo.com/v1/forecast

(c) Link to API Documentation: https://open-meteo.com/en/docs/historical-weather-api

(d) Type of Data Retrieved (JSON, CSV, XML, etc.): JSON

(e) How frequently is data fetched? (Real-time, Batch, On-Demand, etc.): Hourly Batch Processing

Challenges faced in retrieving API data and how they were handled:

- As the number of months increases the missing data keeps on increasing
- Identification of countries through its longitude and latitude and inserting the country name on to the data file

2. Data Exploration & Understanding

(a) Overview of the retrieved data (e.g., number of records, structure, key attributes):

It contains the following columns:

Index(['date', 'temperature_2m', 'relative_humidity_2m', 'dew_point_2m', 'apparent_temperature', 'precipitation_probability', 'precipitation', 'rain', 'showers', 'snowfall', 'snow_depth', 'weather_code', 'pressure_msl', 'surface_pressure', 'cloud_cover', 'cloud_cover_low', 'cloud_cover_mid', 'cloud_cover_high', 'visibility', 'evapotranspiration', 'et0_fao_evapotranspiration', 'vapour_pressure_deficit', 'wind_speed_10m', 'wind_speed_80m', 'wind_speed_120m', 'wind_speed_180m', 'wind_direction_10m', 'wind_direction_80m', 'wind_direction_120m', 'wind_direction_180m', 'wind_gusts_10m', 'temperature_80m', 'temperature_120m', 'temperature_180m', 'soil_temperature_0cm', 'soil_temperature_6cm', 'soil_temperature_18cm', 'soil_temperature_54cm', 'soil_moisture_0_to_1cm', 'soil_moisture_1_to_3cm', 'soil_moisture_3_to_9cm', 'soil_moisture_9_to_27cm', 'soil_moisture_27_to_81cm'], dtype='object'

Structure: 54,144 rows, 44 columns

(b) Summarize key insights from the raw data (e.g., distributions, trends, missing values):

1. Distributions:
    - Several columns have high counts of missing or zero values, which indicates that data distribution might be sparse for some features.
    - Columns such as precipitation_probability and evapotranspiration have no missing values, implying complete data for these variables.
    - Columns like soil_temperature_0cm, soil_temperature_6cm, and soil_moisture_0_to_1cm have a large number of missing values (3785), suggesting potential gaps in soil-related metrics.
2. Trends:
    - Weather-related columns like temperature_2m, relative_humidity_2m, and wind_speed_10m have relatively fewer missing values, which could make them reliable for trend analysis over time.
    - Data related to higher altitudes (temperature_180m, wind_speed_180m) and soil metrics have significant missing entries, possibly indicating measurement challenges at these levels.
3. Missing Values:
    - A high count of missing data is observed in:
        - soil_temperature and soil_moisture columns.
        - Columns like rain, snowfall, weather_code, and snow_depth also have substantial missing values (1380, 1080, and 3785 respectively).
    - Zero missing values are seen in columns like date, precipitation_probability, and Country, indicating complete records.
4. Potential Issues:
    - The high missing data in specific features like wind_speed_180m (3785) and soil_temperature_54cm (3785) might lead to bias or inaccuracies in modeling if not addressed appropriately.
    - Variability in missingness across features suggests the need for a tailored imputation or data exclusion strategy.
5. Focus for Analysis:

- Reliable features with fewer missing values (temperature_2m, pressure_msl, relative_humidity_2m) can be prioritized for initial analysis.
- Variables with high missing values may need preprocessing techniques, such as imputation or exclusion, depending on their importance.

(c) Any API Rate Limits or Constraints Faced During Data Retrieval?

- No rate limits or constraints were encountered during data retrieval.
- The data was successfully fetched for the period 2024-12-01 to 2025-03-01 without any interruptions or issues.
- The API provided seamless access to historical weather data, ensuring smooth and efficient data collection for all 24 cities.

3. Data Cleaning & Preprocessing

(a) Handling Missing or Incomplete Data

- Missing values in the dataset were addressed using backward filling and interpolation:
    - Backward Filling: Missing values were filled by propagating the last observed value backward in time.
    - Interpolation: Missing values were estimated by interpolating between known data points, ensuring smooth transitions and reducing gaps.

(b) Data Type Transformation

- No data type transformations were performed on the dataset. All columns retained their original data types as provided in the raw data.

(c) Feature Engineering

- New Column Added:
    - A new column, "City", was introduced based on existing columns to enrich the data and provide context for location-based analysis.
- New Features Generated:
    - Travel Comfort Index: This index was computed using a combination of weather-related features (e.g., temperature, humidity, and wind speed) to assess overall comfort for travel conditions.
    - Temperature-Humidity Interaction: This feature represents the interaction effect of temperature and humidity to capture how these factors together influence environmental conditions.

- Updated Column Count:

- After adding the new column and two new features, the total number of columns increased to 46.

4. Data Storage & Pipeline

(a) Data Storage Location

- The primary storage for the API data was in a Pandas DataFrame, enabling efficient manipulation and analysis during the preprocessing stage.
- After preprocessing:
    - The cleaned and prepared data was exported to CSV files to serve as training datasets for machine learning models.

(b) Data Pipeline Structure

- An ETL (Extract, Transform, Load) Pipeline was implemented to manage the data workflow:
    - Extract:
        - API data was fetched manually for 24 tourist cities worldwide at scheduled intervals.
    - Transform:
        - Data cleaning processes like handling missing values (backward filling and interpolation), feature extraction (e.g., Travel Comfort Index, Temperature-Humidity Interaction), and normalization were applied to ensure quality and usability.
    - Load:
        - The transformed data was stored in CSV files for machine learning tasks.
        - Processed data was also uploaded to MongoDB for long-term archival and easy retrieval.

(c) Data Refresh/Update

- Data refreshes and updates were performed manually at specific intervals by re-fetching API data and reapplying the ETL pipeline steps. This approach ensures the most recent data is included while maintaining consistency in preprocessing.

5. Data Integrity & Quality Checks

(a) Quality Checks to Ensure Data Correctness

- The retrieved API data was cross-verified with actual climatic conditions by conducting independent web searches on Google to confirm the accuracy of temperature, humidity, wind speed, and precipitation data for the 24 cities.
- Additional checks included:
    - Ensuring data completeness by verifying that all mandatory fields were populated.
    - Comparing calculated values (e.g., dew point, apparent temperature) with standard formula outputs.
    - Checking for time consistency by validating timestamps for each batch of data.

(b) Outlier Detection and Handling

- Outliers in the dataset were detected using statistical methods, such as identifying data points lying beyond 1.5 times the interquartile range (IQR).
- Detected outliers were normalized using the Min-Max Scaling technique to bring them within the range of 0–1, ensuring uniformity without discarding potentially valuable data.

(c) Measures to Prevent Duplicate or Inconsistent Data

- Since the data was manually fetched in batches for 24 cities, no duplicate rows were observed, ensuring consistency in loading.
- Preventive measures included:
    - Assigning unique identifiers for each city and timestamp combination.
    - Validating data formats and ensuring all rows adhered to the expected structure during preprocessing.

6. Preprocessed Data Structure & Readiness for Modeling

(a) Overview of the Final Dataset After Preprocessing

- The final dataset contains 57,144 instances after preprocessing.
- Two additional columns were introduced through feature extraction (e.g., Travel Comfort Index and Temperature-Humidity Interaction), expanding the dataset beyond the original columns retrieved from the API.
- Unnecessary columns were removed during preprocessing, ensuring only relevant features were retained for modeling purposes.

(b) Structure for Training/Testing Models

- The dataset is well-structured for training and testing models:

- Numerical features have been normalized (e.g., Min-Max Scaling) for uniformity and compatibility with machine learning algorithms.
- Categorical columns (if any) are encoded using suitable techniques, such as one-hot encoding or label encoding, enabling seamless integration into ML pipelines.
- The dataset is clean, with no duplicates or missing values, making it readily usable for model training.

- Further splitting into training, validation, and test sets ensures robustness during evaluation.

(c) Data Augmentation Techniques

- No data augmentation was applied in this process, as it might not be directly relevant to the dataset's structure (climatic data is typically not augmented).

7. Challenges & Solutions

(a) Biggest Challenges Faced in Handling and Preprocessing API Data

1. Handling Missing Values:
   - The dataset contained a large number of missing values, making imputation challenging.
   - There was uncertainty about whether the imputed values would accurately represent the actual data, potentially impacting the model's performance.
2. Feature Engineering Difficulties:
   - Calculating the Travel Comfort Index and Temperature-Humidity Interaction required assigning appropriate weights to various contributing factors.
   - Determining the ideal weights to ensure accurate results proved to be a complex task, as it involved balancing multiple variables with different influences.

(b) Solutions to Overcome the Challenges

1. Imputation and Preprocessing:
   - Multiple imputation and preprocessing techniques were tested through a trial-and-error approach to address the missing values.
   - Backward filling and interpolation were ultimately chosen as they provided a balance between simplicity and accuracy.
   - Regular validation and comparison of the results ensured that the imputed data was reasonable and usable for analysis.

2. Weight Assignment for Feature Engineering:
    - Weights for the Travel Comfort Index and Temperature-Humidity Interaction were assigned iteratively using a trial-and-error method.
    - Frequent adjustments and conversions were made, experimenting with different weight combinations until the outputs were logical and aligned with expected trends.
    - Feedback from domain knowledge and validation against real-world conditions helped refine the weights for better results.

8. Supporting Code & References

(a) Attach or provide links to code snippets showcasing data handling and preprocessing.

GitHub Link for code snippets related to data handling and data preprocessing: Smart-Travel-Tracker

(b) References Used in Data Preprocessing

1. API Documentation:
    - The primary reference for data fetching and API usage was the official API documentation.
    - Link to the API: Open-Meteo
2. Other References:
    - Standard methods and best practices for handling missing values, normalization, and feature engineering were referenced from online resources like Stack Overflow and AI Tools like ChatGPT and Blackbox AI.
    - Domain knowledge for weather-related calculations (e.g., Travel Comfort Index) was cross-verified with travel ratings and user experiences on the websites of a few airline services used for visits to the cities, as well as other resources available online.