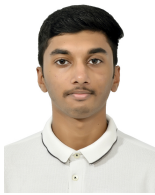


Exploratory Data Analysis on

METEOROLOGY

by

Group 7



Ayush Popshetwar
ID: 202201412
Course: BTech(ICT)



Parjanya Rajput
ID: 202201115
Course: BTech(ICT)



Vats Shah
ID: 202201417
Course: BTech(ICT)

Course Code: IT 462
Semester: Autumn 2024

Under the guidance of

Dr. Gopinath Panda



Dhirubhai Ambani Institute of Information and Communication Technology

December 2, 2024

ACKNOWLEDGMENT

I would like to express my heartfelt gratitude for the guidance and support you provided throughout the duration of my project titled "Meteorology". Your expertise, encouragement, and constructive feedback have been instrumental in overcoming challenges and enhancing the quality of this work.

I also extend my sincere appreciation to DAICT for providing the resources and fostering a collaborative and innovative environment. The facilities and support made available were crucial for conducting thorough research and analysis.

I am thankful to my peers and colleagues for their camaraderie and valuable insights, which have significantly contributed to the progress and completion of this project. This endeavor has been an immensely rewarding journey, offering numerous learning opportunities. The skills and knowledge gained will serve as a strong foundation for future pursuits. Once again, thank you for your unwavering guidance and support. Your mentorship and the environment provided by DAICT have been key factors in the successful completion of this project.

Sincerely,
Ayush Popshetwar, 202201412
Parjanya Rajput, 202201115
Vats Shah, 202201417

DECLARATION

We, [Ayush 202201412, Parjanya 202201115, Vats 202201417] hereby declare that the EDA project work presented in this report is our original work and has not been submitted for any other academic degree. All the sources cited in this report have been appropriately referenced.

We acknowledge that the data used in this project is obtained from the data.gov site. We also declare that we have adhered to the terms and conditions mentioned in the website for using the dataset. We confirm that the dataset used in this project is true and accurate to the best of our knowledge.

We acknowledge that we have received no external help or assistance in conducting this project, except for the guidance provided by our mentor Prof. Gopinath Panda. We declare that there is no conflict of interest in conducting this EDA project.

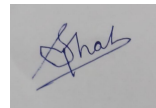
We hereby sign the declaration statement and confirm the submission of this report on 2nd December, 2024.



Ayush Popshetwar
ID: 202201412
Course: BTech(ICT)



Parjanya Rajput
ID: 202201115
Course: BTech(ICT)



Vats Shah
ID: 202201417
Course: BTech(ICT)

CERTIFICATE

This is to certify that Group 7 comprising Ayush Popshetwar, Parjanya Rajput and Vats Shah has successfully completed an exploratory data analysis (EDA) project on Meteorology, which was obtained from data.gov.in.

The EDA project presented by Group 7 is their original work and has been completed under the guidance of the course instructor, Prof. Gopinath Panda, who has provided support and guidance throughout the project. The project is based on a thorough analysis of the Energy and carbon dioxide fluxes, meteorology and soil physics observed at INCOMPASS land surface stations in India, 2016 to 2017 dataset, and the results presented in the report are based on the data obtained from the dataset.

This certificate is issued to recognize the successful completion of the EDA project on Meteorology, which demonstrates the analytical skills and knowledge of the students of Group # in the field of data analysis.

Signed,
Dr. Gopinath Panda,
IT 462 Course Instructor
Dhirubhai Ambani Institute of Information and Communication Technology
Gandhinagar, Gujarat, INDIA.

December 2, 2024

Contents

List of Figures	5
1 Introduction	1
1.1 Project idea	1
1.1.1 Goals And Objectives	1
1.2 Data Collection	2
1.3 Dataset Description	2
1.4 Packages Required	5
2 Data Cleaning	7
2.1 Missing Data Analysis	7
2.1.1 Missing Data Bar Plot	9
2.1.2 Missing Data Matrix	10
2.1.3 Missing Data Dendrogram	11
2.2 Imputation	12
2.2.1 Backward Fill	12
2.2.2 Forward Fill	12
2.3 Data Formatting	13
2.4 Outlier Detection and Treatment	13
2.4.1 Outlier Detection	13
2.4.2 Visualization of Outliers	14
2.4.3 Outlier Treatment	16
3 Visualization	17
3.1 Univariate analysis	17
3.1.1 Histograms	17
3.1.2 Time Series Plots	20
3.2 Multivariate Analysis	23
3.2.1 Heat Map / Correlation Matrix	23
3.2.2 Scatter Plots	25
4 Feature Engineering	28
4.1 Feature Selection Based on Univariate and Multivariate Analysis	28
4.2 Feature Creation	30
4.2.1 Wind Power as a Feature	30

4.2.2	Net Shortwave and Longwave Radiation as a feature	32
4.2.3	Violin Plots of Air Temperature by Day/Night and Season	34
4.3	Feature extraction	35
4.3.1	Relevance of Dimensionality Reduction	35
4.3.2	How PCA Reduces Dimension	35
4.3.3	PCA Table and Analysis	36
5	Model fitting	37
5.1	ML algorithms	37
5.1.1	Description of Models Used	37
5.1.2	Best Model for Transformed Data	38
6	Conclusion & Future scope	39
6.1	Conclusions (Findings And Observations)	39
6.2	Challenges	40
6.3	Future Scope	40

List of Figures

1.1	Description of the figure.	5
2.1	Summary statistics of the dataset before identifying missing values.	8
2.2	Bar plot visualization of missing data for each column.	9
2.3	Matrix visualization of missing data patterns.	10
2.4	Dendrogram visualization of missing data correlations.	11
2.5	Summary statistics of the dataset after imputation.	12
2.6	Box Plot for Temperature (Ta)	14
2.7	Box Plot for H(Sensible Heat Flux)	15
2.8	Box Plot for G1(Soil Heat Flux)	15
3.1	Histogram for G2 (Soil Heat Flux at 0.03 m, Location 2)	18
3.2	Histogram for Pa (Barometric Pressure)	19
3.3	Histogram for RH (Relative Humidity)	20
3.4	Time Series Plot of Ta (Air Temperature in °C)	21
3.5	Time Series Plot of LE (Latent Heat Flux Density in W/m ²)	21
3.6	Heat Map showing the correlation between various variables. Darker colors represent stronger correlations (positive or negative).	23
3.7	Scatter plot of Air Temperature (T_a) vs Relative Humidity (RH).	25
3.8	Scatter plot of Air Temperature (T_a) vs Barometric Pressure (Pa).	26
3.9	Scatter plot of Air Temperature (T_a) vs Soil Temperature at 0.05m (Pa).	27
4.1	Correlation between Air Temperature (Ta) and Wind Speed (WS_EC)	30
4.2	Correlation between Air Temperature (Ta) and Wind Direction (WD_EC)	30
4.3	Air Temperature (Ta) vs Wind Power ($WS_EC^2 * WD_EC$)	30
4.4	Incoming Shortwave Radiation (SWin)	32
4.5	Outgoing Shortwave Radiation (SWout)	32
4.6	Net Shortwave Radiation (Net_SW)	32
4.7	Incoming Longwave Radiation (LWin)	33
4.8	Outgoing Longwave Radiation (LWout)	33
4.9	Net Longwave Radiation (Net_LW)	33
4.10	Air Temperature (Ta) by Day/Night	34
4.11	Air Temperature (Ta) by Season	34
4.12	PCA Transformation Table	36
5.1	Model Performance Comparison	38

Abstract

This project analyzes meteorological and soil data from the INCOMPASS project with the goal of predicting air temperature (T_a). Meteorology plays a crucial role in understanding climate patterns, weather forecasting, and environmental monitoring. By examining the relationship between weather conditions like radiation, humidity, wind speed, and soil moisture, we aim to develop accurate predictive models for temperature forecasting. The data is preprocessed to handle missing values and outliers, with dimensionality reduction performed using PCA. Various machine learning models, including Linear Regression and Random Forest, are applied to predict air temperature. Visualization techniques such as scatter plots and correlation matrices are used to explore key relationships, helping improve the understanding of temperature dynamics and contributing to more accurate climate modeling.

Chapter 1. Introduction

The Earth's atmosphere and surface interact dynamically, influencing weather patterns, climate variability, and environmental conditions. This project leverages data collected under the INCOMPASS project, which documented surface-atmosphere exchanges in India from 2016 to 2017. The dataset contains high-resolution observations of energy and carbon dioxide fluxes, meteorological variables, soil physics, and atmospheric turbulence. This rich dataset provides an opportunity to explore and model key meteorological parameters, gaining insights into their variability and interdependencies.

The study focuses on analyzing air temperature (T_a), relative humidity (RH), and other meteorological factors to understand their interactions with soil and atmospheric properties. The data-driven approach includes preprocessing (handling missing values, capping outliers), exploratory data analysis (EDA), and predictive modeling using machine learning. EDA techniques, such as correlation matrices, regression plots, and violin plots, reveal patterns across temporal and categorical dimensions, while machine learning models predict key meteorological metrics.

By combining advanced statistical and computational techniques, this project aims to enhance the understanding of meteorological dynamics, contributing to improved forecasting and climate analysis methods.

1.1 Project idea

This project explores the dynamic interactions between Earth's surface and atmosphere using data from the INCOMPASS project, which recorded energy and carbon dioxide fluxes, meteorological observations, and soil physics metrics in India. By analyzing this comprehensive dataset, the project aims to understand the relationships between air temperature, humidity, and other variables under varying environmental conditions. Employing data preprocessing, exploratory data analysis (EDA), and machine learning, the study seeks to reveal key insights into meteorological variability and predict critical parameters effectively.

1.1.1 Goals And Objectives

The primary goal of this project is to analyze and model meteorological data to understand the interplay between air temperature, relative humidity, soil physics, and energy fluxes using the INCOMPASS dataset. The project involves preprocessing the data to handle missing values, outliers, and inconsistencies, followed by exploratory data analysis (EDA) to uncover patterns and relationships using techniques like correlation matrices, scatter plots, and regression lines. Key objectives include applying principal component analysis (PCA) for dimensionality reduction, training predictive models such

as Linear Regression, Random Forest, and Gradient Boosting, and evaluating their performance using metrics like MAE, MSE, and R^2 . Additionally, the project examines seasonal and diurnal variations, providing insights into meteorological variability and contributing to advancements in weather forecasting and environmental research.

1.2 Data Collection

Initially, we explored data.gov.in for meteorological datasets but could not find datasets with sufficient observations and features for our analysis. To overcome this, we researched alternative methods to locate effective datasets and discovered the Google Dataset Search Engine. Using this tool, we identified a comprehensive dataset on data.gov, titled Energy and Carbon Dioxide Fluxes, Meteorology, and Soil Physics Observed at INCOMPASS Land Surface Stations in India (2016–2017).

This dataset includes high-resolution measurements taken at 30-minute intervals from three land surface stations located in Kanpur, Dharwad, and Berambadi. With a wealth of observations and features, this dataset was ideally suited for our project, providing the necessary depth and granularity for meaningful analysis.

1.3 Dataset Description

The dataset includes ancillary weather and soil physics observations, as well as variables describing atmospheric turbulence and the quality of the turbulent flux observations. Meteorological observations include: the net radiation and its incoming and outgoing short-wave and long-wave components, air temperature, barometric pressure, relative humidity, wind speed and direction, and rainfall. Soil physics observations include: soil heat fluxes, soil temperatures, and soil volumetric water content. Observations were collected under the Interaction of Convective Organization and Monsoon Precipitation, Atmosphere, Surface and Sea (INCOMPASS) Project.

The dataset includes approximately 3500 rows and 44 features for each region (Kanpur, Dharwad, and Berambadi). The data columns are as follows:

- **DateTime:** Timestamp of the observation (object type). This feature indicates the exact time at which the measurement was recorded.
- **H:** Sensible heat flux (W/m^2). This represents the heat flux due to the temperature difference between the surface and the atmosphere, driving convective heat transfer.
- **H_unc:** Uncertainty in sensible heat flux (W/m^2). This represents the error margin associated with the measurement of sensible heat flux.
- **LE:** Latent heat flux (W/m^2). This represents the heat transferred due to water evaporation or transpiration from the soil or vegetation.
- **LE_unc:** Uncertainty in latent heat flux (W/m^2). This feature quantifies the uncertainty in the measurement of latent heat flux.

- **Tau:** Friction velocity (m/s). This parameter measures the velocity at which air is mixed due to surface roughness and turbulence, impacting heat and moisture exchange.
- **Tau_unc:** Uncertainty in friction velocity (m/s). This feature quantifies the uncertainty in the friction velocity measurement.
- **NEE:** Net ecosystem exchange of CO₂ ($\mu\text{mol CO}_2/\text{m}^2/\text{s}$). This represents the difference between CO₂ absorbed by the ecosystem and that emitted by the soil and plants.
- **NEE_unc:** Uncertainty in NEE ($\mu\text{mol CO}_2/\text{m}^2/\text{s}$). This feature measures the uncertainty in the net ecosystem exchange of CO₂.
- **CO2_strg:** CO₂ storage ($\mu\text{mol CO}_2/\text{m}^2/\text{s}$). This indicates the amount of CO₂ stored in the soil or ecosystem, contributing to carbon sequestration.
- **Pa:** Atmospheric pressure (Pa). This measures the pressure exerted by the atmosphere at the location of observation, influencing weather patterns.
- **Ta:** Air temperature (°C). The temperature of the air at the surface, influencing evaporation, convection, and overall weather dynamics.
- **RH:** Relative humidity (
- **SWin:** Incoming shortwave radiation (W/m^2). This represents solar radiation received by the Earth's surface, influencing energy balance and weather patterns.
- **SWout:** Outgoing shortwave radiation (W/m^2). This represents the amount of solar radiation reflected or scattered away from the Earth's surface.
- **LWin:** Incoming longwave radiation (W/m^2). This measures infrared radiation received by the Earth's surface, contributing to the energy balance.
- **LWout:** Outgoing longwave radiation (W/m^2). This represents the infrared radiation emitted from the Earth's surface back into space.
- **Rnet:** Net radiation (W/m^2). This is the balance of incoming and outgoing shortwave and long-wave radiation, indicating the overall energy available at the surface.
- **G1:** Soil heat flux at 5 cm depth (W/m^2). This measures the heat flow through the soil at a depth of 5 cm, important for understanding soil energy dynamics.
- **G2:** Soil heat flux at 10 cm depth (W/m^2). This measures the heat flow through the soil at a depth of 10 cm, providing insight into deeper soil dynamics.
- **Tsoil_0.05_1:** Soil temperature at 0-5 cm depth (°C) for sensor 1. This represents the temperature of the soil at the shallowest depth (0-5 cm) measured by the first sensor.
- **Tsoil_0.15_1:** Soil temperature at 5-15 cm depth (°C) for sensor 1. This measures the temperature of the soil at a depth of 5-15 cm, providing data for deeper soil layers.
- **Tsoil_0.05_2:** Soil temperature at 0-5 cm depth (°C) for sensor 2. This represents the temperature at the shallow soil depth (0-5 cm) measured by the second sensor.

- **Tsoil_0.15_2**: Soil temperature at 5-15 cm depth ($^{\circ}\text{C}$) for sensor 2. This measures the temperature of the soil at a depth of 5-15 cm, similar to the previous feature but for the second sensor.
- **VWC_0.05_1**: Volumetric water content at 0-5 cm depth (m^3/m^3) for sensor 1. This represents the amount of water present in the soil at the 0-5 cm depth measured by the first sensor.
- **VWC_0.15_1**: Volumetric water content at 5-15 cm depth (m^3/m^3) for sensor 1. This measures the water content in the soil at the 5-15 cm depth for the first sensor.
- **VWC_0.05_2**: Volumetric water content at 0-5 cm depth (m^3/m^3) for sensor 2. This represents the water content at the 0-5 cm depth as measured by the second sensor.
- **VWC_0.15_2**: Volumetric water content at 5-15 cm depth (m^3/m^3) for sensor 2. This measures the water content in the soil at a 5-15 cm depth for the second sensor.
- **Precipitation**: Rainfall (mm). This feature measures the amount of rainfall at the observation site in millimeters.
- **WS_EC**: Wind speed at the eddy covariance site (m/s). This measures the wind speed at the site where the flux measurements were taken.
- **WD_EC**: Wind direction at the eddy covariance site (degrees). This indicates the direction of the wind at the eddy covariance measurement site.
- **WS_10m**: Wind speed at 10 meters above ground (m/s). This represents the wind speed measured at a height of 10 meters above the surface.
- **WD_10m**: Wind direction at 10 meters above ground (degrees). This measures the direction of the wind at a height of 10 meters above the ground.
- **Ustar**: Friction velocity at the eddy covariance site (m/s). This quantifies the turbulence of the air above the ground, important for estimating heat and moisture fluxes.
- **TKE**: Turbulent kinetic energy (m^2/s^2). This is a measure of the turbulence in the atmosphere, related to the intensity of mixing.
- **Tstar**: Temperature for turbulence flux measurements ($^{\circ}\text{C}$). This feature captures the temperature used in calculations for turbulent flux measurements.
- **L**: Monin-Obukhov length (m). This is a parameter used in boundary layer meteorology to describe the relative importance of buoyancy and shear in the atmosphere.
- **zL**: Roughness length (m). This describes the height at which the wind speed theoretically becomes zero due to surface roughness.
- **x_peak**: Peak of spectral density (Hz). This represents the frequency at which the highest energy in the turbulence spectrum is observed.
- **x_10**: Frequency threshold for temperature fluctuations at 10 Hz (unitless). This indicates the threshold frequency for capturing temperature fluctuations.

- **x_30**: Frequency threshold for temperature fluctuations at 30 Hz (unitless). This feature captures the frequency threshold for temperature variations at 30 Hz.
- **x_50**: Frequency threshold for temperature fluctuations at 50 Hz (unitless). This represents the frequency at which the temperature fluctuations are observed at 50 Hz.
- **x_70**: Frequency threshold for temperature fluctuations at 70 Hz (unitless). This indicates the threshold frequency at 70 Hz for capturing temperature variations.
- **x_90**: Frequency threshold for temperature fluctuations at 90 Hz (unitless). This represents the frequency threshold for temperature fluctuations at 90 Hz.

1.4 Packages Required

```
# Importing Libraries
import pandas as pd #
import requests
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
import warnings
import missingno as msno
from sklearn.decomposition import PCA
from sklearn.preprocessing import StandardScaler
from sklearn.preprocessing import OneHotEncoder
from sklearn.compose import ColumnTransformer
from sklearn.pipeline import Pipeline
from sklearn.impute import SimpleImputer
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression
from sklearn.ensemble import RandomForestRegressor
from sklearn.ensemble import GradientBoostingRegressor
from sklearn.svm import SVR
from sklearn.metrics import mean_absolute_error, mean_squared_error, r2_score
warnings.filterwarnings('ignore')
```

Figure 1.1: Description of the figure.

The following Python packages were used in this project:

1. **pandas**: For storing data in a DataFrame and performing data manipulation and analysis.
2. **requests**: For making HTTP requests to external websites or APIs to fetch data.
3. **numpy**: For performing numerical operations on large arrays and matrices.
4. **matplotlib.pyplot**: For creating static, animated, and interactive visualizations.

5. **seaborn**: For enhanced data visualization with statistical plots such as heatmaps and regression plots.
6. **warnings**: For handling and suppressing warning messages to keep the output clean.
7. **missingno**: For visualizing missing data in datasets.
8. **sklearn.decomposition.PCA**: For performing Principal Component Analysis (PCA) to reduce data dimensions.
9. **sklearn.preprocessing.StandardScaler**: For standardizing the dataset by scaling features to have zero mean and unit variance.
10. **sklearn.preprocessing.OneHotEncoder**: For encoding categorical variables into a format that can be used by machine learning models.
11. **sklearn.compose.ColumnTransformer**: For applying transformations to selected columns of the dataset.
12. **sklearn.pipeline.Pipeline**: For creating a pipeline of data transformations and model fitting steps.
13. **sklearn.impute.SimpleImputer**: For handling missing data by imputing values based on statistical methods.
14. **sklearn.model_selection.train_test_split**: For splitting the dataset into training and testing subsets.
15. **sklearn.linear_model.LinearRegression**: For building a linear regression model to predict continuous values.
16. **sklearn.ensemble.RandomForestRegressor**: For building a Random Forest Regressor model to predict continuous values.
17. **sklearn.ensemble.GradientBoostingRegressor**: For building a Gradient Boosting Regressor model to predict continuous values.
18. **sklearn.svm.SVR**: For building a Support Vector Regression (SVR) model for predicting continuous values.
19. **sklearn.metrics.mean_absolute_error**: For calculating the mean absolute error (MAE) to evaluate model performance.
20. **sklearn.metrics.mean_squared_error**: For calculating the mean squared error (MSE) to evaluate model performance.
21. **sklearn.metrics.r2_score**: For calculating the R-squared score to evaluate the model's performance in predicting the target variable.

Chapter 2. Data Cleaning

Data cleaning is an essential step in the Exploratory Data Analysis (EDA) process. It involves identifying and rectifying errors, inconsistencies, and missing values in the dataset, ensuring that the data is accurate, complete, and ready for further analysis. The quality of data directly impacts the results and performance of any machine learning model. Without proper cleaning, the data can introduce noise, bias, and inaccuracies, leading to misleading insights and poor model predictions. In our project, we focused on two primary aspects of data cleaning: Missing Data Analysis and Imputation. By performing missing data analysis and imputation, we were able to prepare a clean dataset that could be effectively used for further analysis, visualization, and machine learning modeling.

2.1 Missing Data Analysis

In the initial stage of our data cleaning process, we first used the `.isnull().sum()` method to check for missing values in the dataset. This method calculates the total number of missing values in each column, and based on the results, we initially thought that there were no missing values in the dataset. However, when we used the `.describe()` method to calculate summary statistics, we noticed that the mean values were highly inaccurate, which indicated the presence of an issue. Upon further inspection, we realized that the missing values were not represented as NaN but as a placeholder value of `-9999.00`. This value was likely used to indicate missing or unavailable data in the dataset.

To address this, we replaced all instances of `-9999.00` in the dataset with NaN (Not a Number), which is a more appropriate representation for missing data in pandas. This step allowed us to correctly identify and handle the missing values, ensuring that they were properly imputed or managed in subsequent steps. The replacement of `-9999.00` with NaN allowed us to conduct a more accurate analysis and proceed with the data cleaning process effectively.

After replacing the placeholder values, we calculated that the total percentage of missing data in the dataset was 15.69% using `totalmissingpercentage = df.isnull().sum().sum() * 100 / df.size`.

	DateTime	H	H_unc	LE	LE_unc	Tau	Tau_unc	NEE	NEE_unc	CO2_strg	...
count	35088	35088.000000	35088.000000	35088.000000	35088.000000	35088.000000	35088.000000	35088.000000	35088.000000	35088.000000	...
mean	2016-12-31 12:15:00	-2274.467463	-1051.165419	-4579.673662	-2212.388262	-1054.049990	-1054.095192	-4693.835270	-4690.827987	-4724.854619	...
min	2016-01-01 00:30:00	-9999.000000	-9999.000000	-9999.000000	-9999.000000	-9999.000000	-9999.000000	-9999.000000	-9999.000000	-9999.000000	...
25%	2016-07-01 18:22:30	-37.551850	0.381200	-9999.000000	0.167250	0.001600	0.000300	-9999.000000	-9999.000000	-9999.000000	...
50%	2016-12-31 12:15:00	-3.356050	1.124250	-1.166200	1.865600	0.018000	0.002900	-31.255250	0.190850	-1.645700	...
75%	2017-07-02 06:07:30	18.999175	3.517950	50.775325	9.251275	0.072700	0.009600	0.552675	0.895725	-0.019000	...
max	2018-01-01 00:00:00	384.834600	352.249800	794.017600	464.591700	2.783200	0.236900	20.953000	33.230000	35.811700	...
std	NaN	4216.648596	3071.694830	5037.148685	4159.690308	3070.701013	3070.685495	4987.836464	4990.654808	4991.956765	...

8 rows × 44 columns

Figure 2.1: Summary statistics of the dataset before identifying missing values.

2.1.1 Missing Data Bar Plot

The bar plot provides a clear visualization of the missing data count for each column in the dataset. It helps identify which columns have significant missing values and their overall distribution.

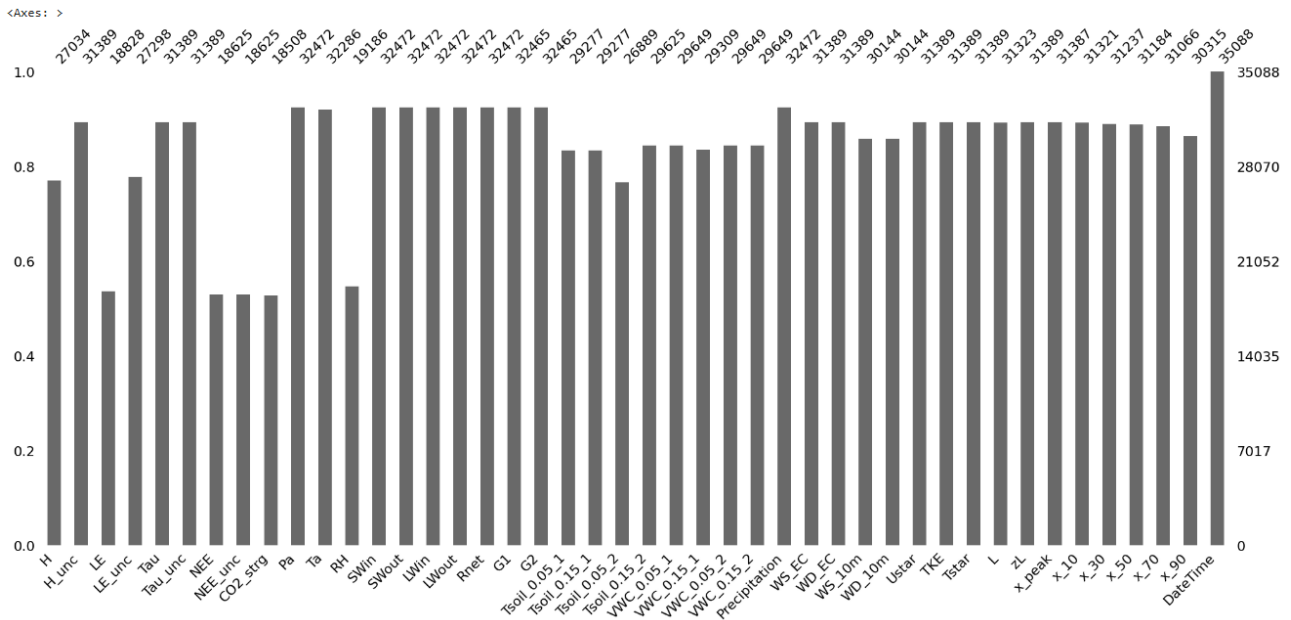


Figure 2.2: Bar plot visualization of missing data for each column.

2.1.2 Missing Data Matrix

The matrix plot offers a detailed view of missing data patterns, showing which rows and columns contain missing values. It is particularly useful for detecting patterns in data absence.

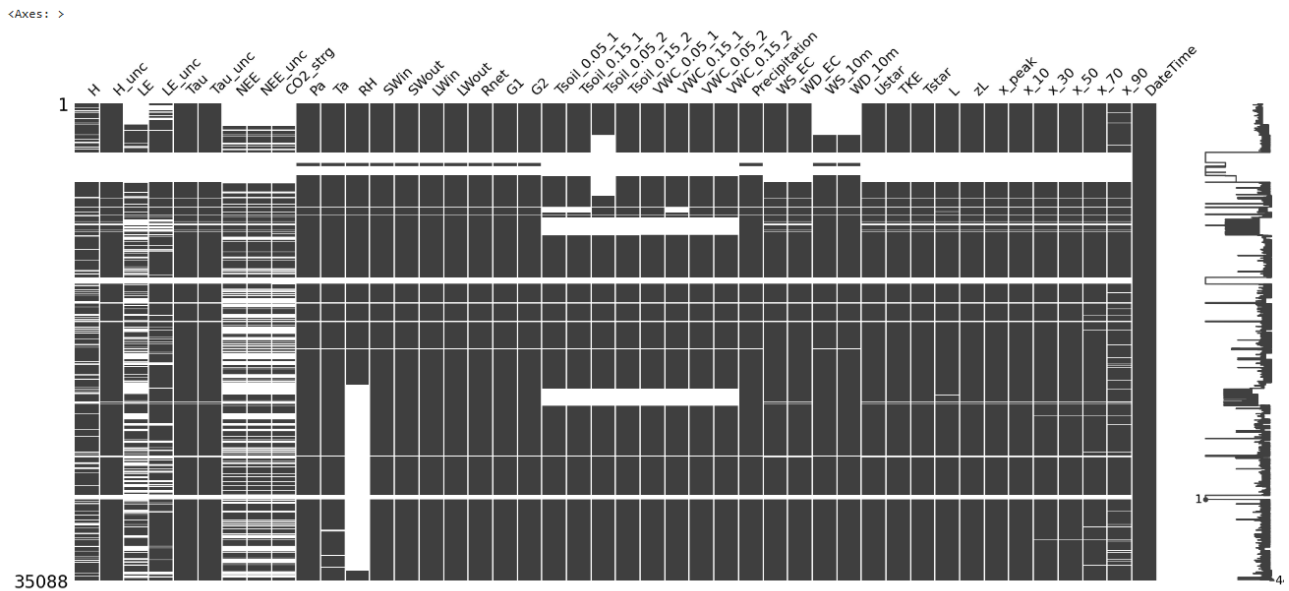


Figure 2.3: Matrix visualization of missing data patterns.

2.1.3 Missing Data Dendrogram

The dendrogram plot clusters the columns based on the similarity of their missing value patterns. This hierarchical representation provides insights into how missing data is correlated across different columns.

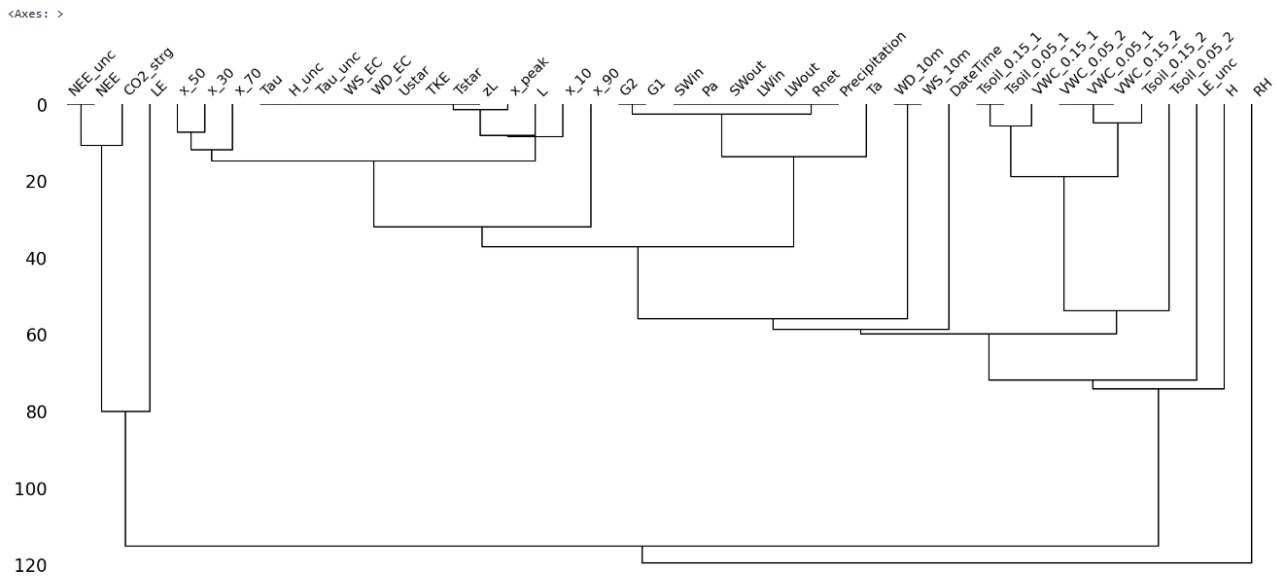


Figure 2.4: Dendrogram visualization of missing data correlations.

2.2 Imputation

After identifying and representing the missing data appropriately, we proceeded to handle the missing values through imputation. Given that the dataset records data at 30-minute intervals, we opted for time-series-specific imputation techniques to preserve the temporal relationships in the data.

2.2.1 Backward Fill

Initially, we applied the backward fill method for imputation. This method replaces missing values with the next available value, which is ideal for time-series data as it ensures that missing values are replaced by the data points closest in time. This approach is more appropriate for our dataset compared to using mean or median imputation, which could distort the temporal consistency of the data.

After applying backward fill, the total percentage of missing data in the dataset was reduced to **0.000712%**. However, some missing values still persisted, particularly at the end of certain columns where no subsequent values existed for replacement.

2.2.2 Forward Fill

To address the remaining missing values, we employed the forward fill method. This method replaces missing values with the most recent available value, further ensuring that temporal relationships in the data are maintained.

After applying forward fill, the total percentage of missing data in the dataset was successfully reduced to **0.000000%**. The imputed dataset was then re-evaluated using the `.describe()` method, and the summary statistics now appeared more accurate and consistent, reflecting the effectiveness of the imputation process.

	H	H_unc	LE	LE_unc	Tau	Tau_unc	NEE	NEE_unc	CO2_strg	Pa	...
count	35088.000000	35088.000000	35088.000000	35088.000000	35088.000000	35088.000000	35088.000000	35088.000000	35088.000000	35088.000000	...
mean	18.167776	3.154424	87.585952	11.649759	0.061638	0.007102	-2.402739	1.159530	-0.220842	99.339105	...
min	-139.084300	0.002000	-88.019000	0.001800	0.000000	0.000000	-72.650700	0.002100	-39.907800	98.000000	...
25%	-10.107300	0.582875	5.697600	1.428300	0.004100	0.000800	-5.101250	0.448300	-0.340700	98.780000	...
50%	-0.993050	1.549150	44.814950	5.293200	0.031600	0.004900	0.835100	0.792950	-0.030300	99.203300	...
75%	22.590975	3.549950	140.420500	12.549700	0.101600	0.012300	3.432900	1.320100	0.245900	99.908600	...
max	384.834600	352.249800	794.017600	464.591700	2.783200	0.236900	20.953000	33.230000	35.811700	100.920000	...
std	59.914014	4.751439	111.126810	23.848383	0.080077	0.007682	10.830005	1.559649	2.720159	0.673760	...

8 rows × 44 columns

Figure 2.5: Summary statistics of the dataset after imputation.

2.3 Data Formatting

Data formatting is a crucial step in data preprocessing to ensure that all features are in a consistent and usable format. Properly formatted data facilitates smoother analysis and reduces the likelihood of errors during subsequent processes like visualization and modeling.

In our dataset, the `DateTime` feature was initially of `datetime` object, which is not optimal for performing time-series analysis. To address this, we converted the `DateTime` feature into a `datetime` format using the Python `pandas` library:

```
# Transform 'DateTime' column to timestamp
data['DateTime'] = pd.to_datetime(data['DateTime'])
```

This transformation allowed us to leverage the full capabilities of the `datetime` datatype, such as easy extraction of temporal information (Eg: year, month, day, hour) and efficient time-series operations.

Upon inspection, all other features in the dataset were of `float` datatype and already in the correct format. Hence, no additional formatting was required for those features.

2.4 Outlier Detection and Treatment

Outlier detection and treatment are essential steps in Exploratory Data Analysis (EDA) as outliers can significantly influence the results of statistical analyses and machine learning models. In this project, outliers were identified and handled using the Interquartile Range (IQR) method.

2.4.1 Outlier Detection

We analyzed outliers in all numerical features using the IQR method. This approach calculates the lower and upper bounds based on the 25th percentile ($Q1$) and 75th percentile ($Q3$) as follows:

$$\text{IQR} = Q3 - Q1 \\ \text{Lower Bound} = Q1 - 1.5 \times \text{IQR} \\ \text{Upper Bound} = Q3 + 1.5 \times \text{IQR}$$

Values falling outside these bounds are considered outliers. Using this method, the number of outliers for each feature was determined. Some features with the highest number of outliers included:

- `CO2_strg`: 7793
- `H`: 5081
- `L`: 5646
- `zL`: 7700
- `G1`: 2901

2.4.2 Visualization of Outliers

Box plots were used to visualize the distribution of data and the presence of outliers. Below are the box plots for some of the significant features: **Temperature (Ta)**, **H(Sensible Heat Flux)**, and **G1(Soil Heat Flux)**.

Number of outliers in Ta: 0

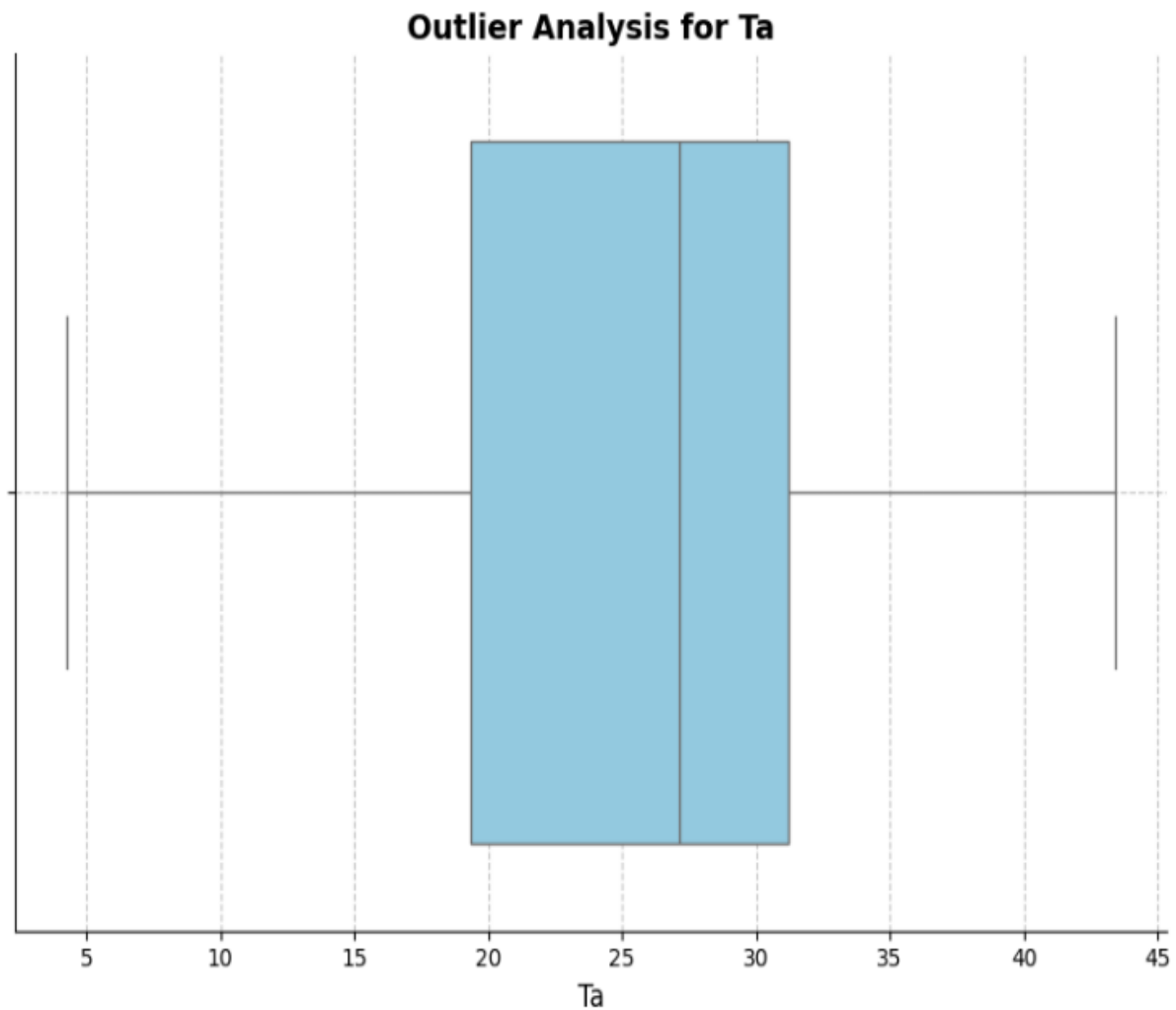


Figure 2.6: Box Plot for **Temperature (Ta)**

Number of outliers in H: 5081

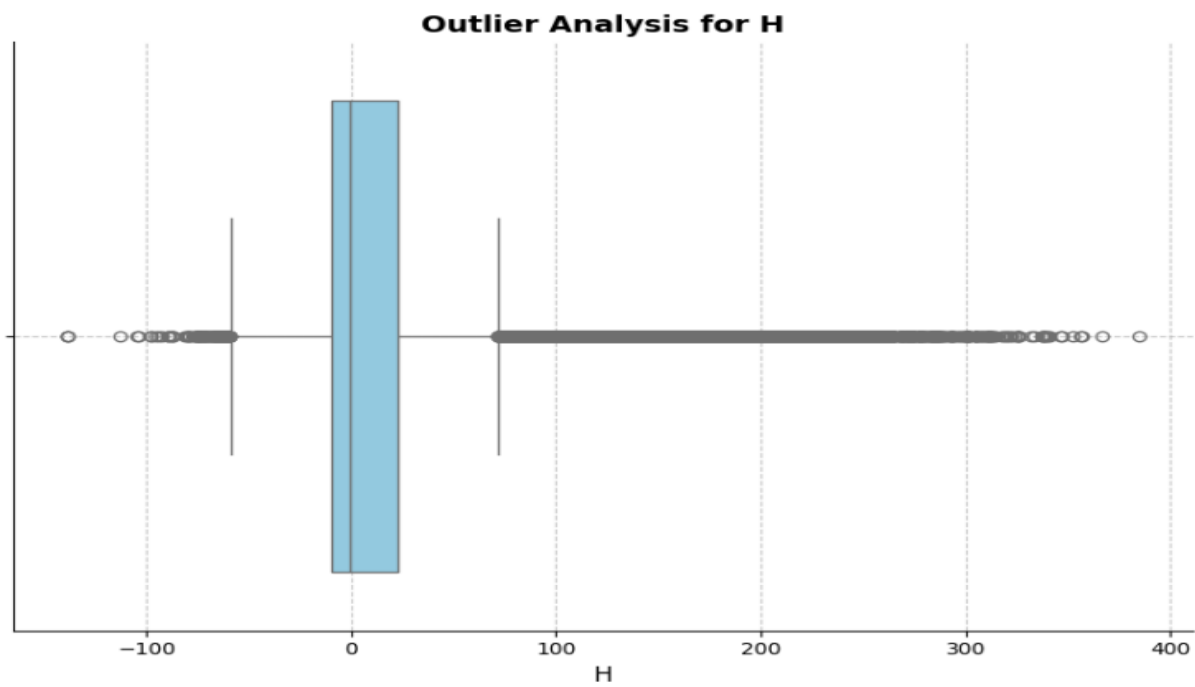


Figure 2.7: Box Plot for H(Sensible Heat Flux)

Number of outliers in G1: 2901

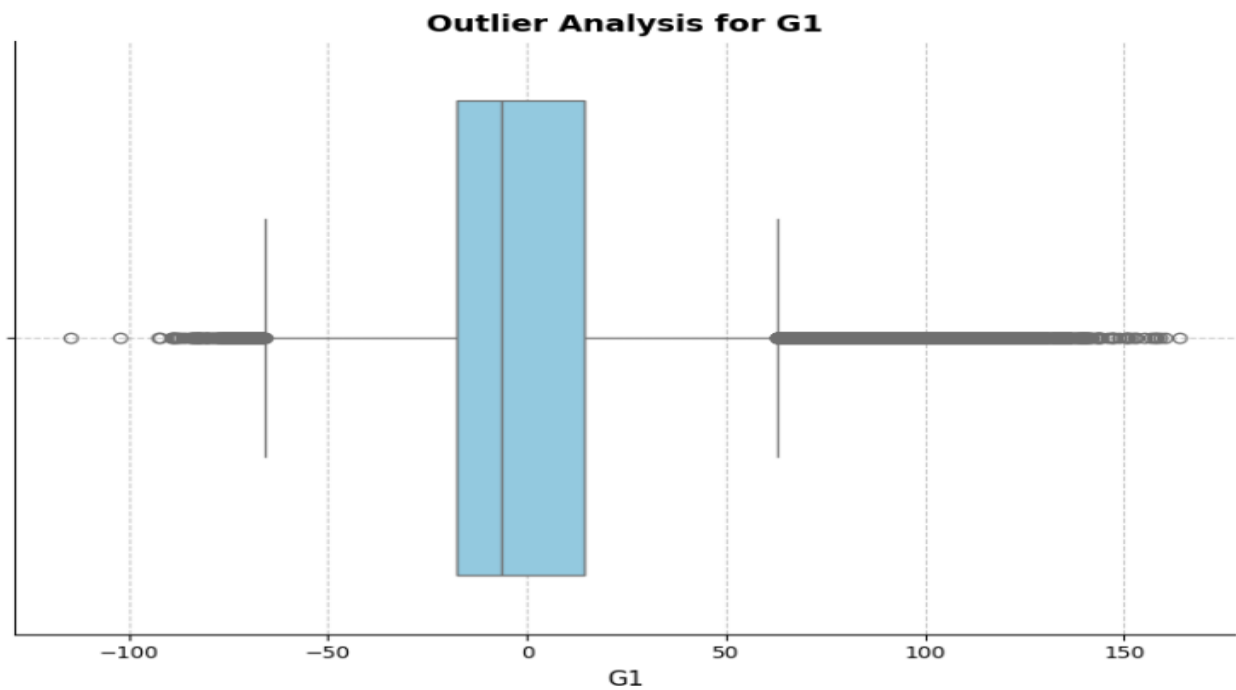


Figure 2.8: Box Plot for G1(Soil Heat Flux)

2.4.3 Outlier Treatment

Instead of removing the outliers completely, we opted for **capping**, where outliers were adjusted to the lower and upper bounds of the IQR. This approach ensured that no data was lost while mitigating the influence of extreme values. The capping process for each numerical feature was implemented as follows:

```
# Cap outliers using IQR for each numerical column
for col in numerical_cols:
    Q1 = df[col].quantile(0.25)
    Q3 = df[col].quantile(0.75)
    IQR = Q3 - Q1
    lower_bound = Q1 - 1.5 * IQR
    upper_bound = Q3 + 1.5 * IQR

    # Cap outliers
    df[col] = np.clip(df[col], lower_bound, upper_bound)
```

After capping, the data distribution was re-evaluated, and the influence of extreme values was reduced significantly. This approach preserved the dataset's integrity and ensured that subsequent analyses and model training steps were not adversely impacted by extreme values. The total percentage of outliers detected across the dataset was calculated to now be **00.00%**.

Chapter 3. Visualization

In this chapter, we present various visualizations to explore the dataset and uncover underlying patterns, relationships, and distributions. The following types of plots have been used:

- **Correlation Matrix:** A heatmap representation of the correlation matrix is used to explore relationships between different variables. It visually identifies which variables are strongly or weakly correlated, which can be useful for feature selection in modeling.
- **Scatter Plot:** Scatter plots are used to examine the relationship between two continuous variables. They provide insight into trends, patterns, and the degree of correlation between variables.
- **Histogram:** Histograms are used to visualize the distribution of a single continuous variable. It provides a way to understand the frequency distribution and skewness of the data, highlighting where most of the data points lie.
- **Line Plot:** Line plots are used to visualize trends over time or any ordered data. They help in understanding how a variable changes with respect to another variable, especially in time series analysis.
- **Violin Plot:** Violin plots are a combination of box plots and kernel density plots. They are used to show the distribution of a continuous variable across different categories. This plot helps in comparing the distributions and detecting outliers.

The visualizations provided valuable insights into the data distribution and temporal patterns, which will help inform the modeling approach.

3.1 Univariate analysis

For univariate analysis, we performed visualizations using histograms and time series plots for each feature in the dataset.

3.1.1 Histograms

Histograms were plotted for all numerical features to analyze the distribution of each variable. The histograms include kernel density estimation (KDE) curves to visualize the probability density of the data. These plots allow for identifying skewness, kurtosis, and the spread of data. The histograms provide insight into the spread and shape of the data for each numerical feature, such as air temperature

(Ta), relative humidity (RH), and others. Some of the histograms are shown below, along with their inferences and observations.

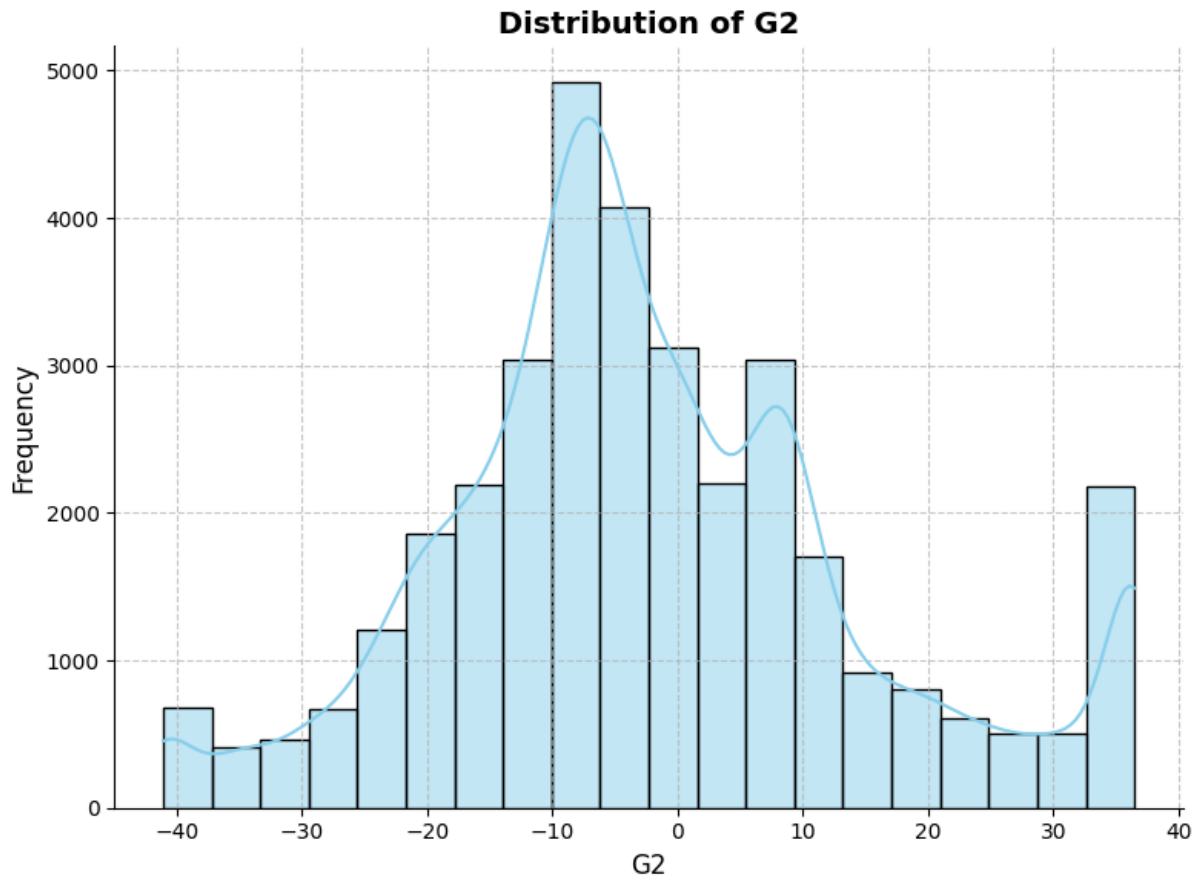


Figure 3.1: Histogram for G2 (Soil Heat Flux at 0.03 m, Location 2)

Inference and Observations: The given distribution of soil heat flux (G2) at 0.03m depth, location 2 shows:

- **Skewed distribution:** More values on the higher end.
- **Centered around 0:** Most values are close to zero.
- **Wide spread:** Significant variation in values.
- **Outliers:** Presence of extreme values.

This suggests that the soil at this location experiences varying heat fluxes, influenced by factors like seasonality, soil properties, and weather conditions. Further analysis with statistical measures and time-series analysis can provide deeper insights.

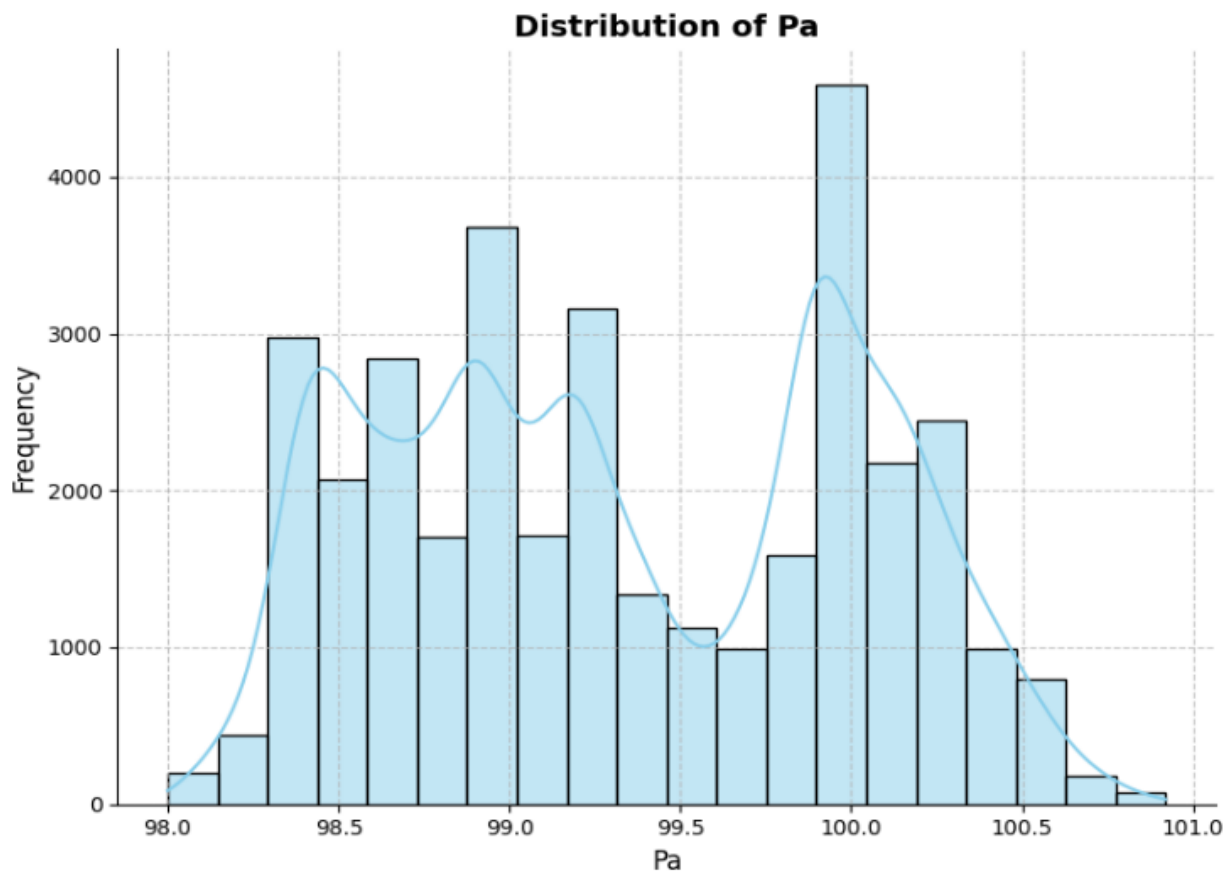


Figure 3.2: Histogram for Pa (Barometric Pressure)

Inference and Observations: The distribution of barometric pressure appears to be approximately normal, indicating relatively stable atmospheric conditions during the measurement period. The peak around 100 Pa suggests that the majority of measurements were centered around this value, with minimal fluctuations.

However, the presence of a few potential outliers on the lower end of the distribution might indicate occasional disturbances or measurement errors. These deviations from the normal pattern could be caused by factors such as sudden weather changes, instrument calibration issues, or other external influences..

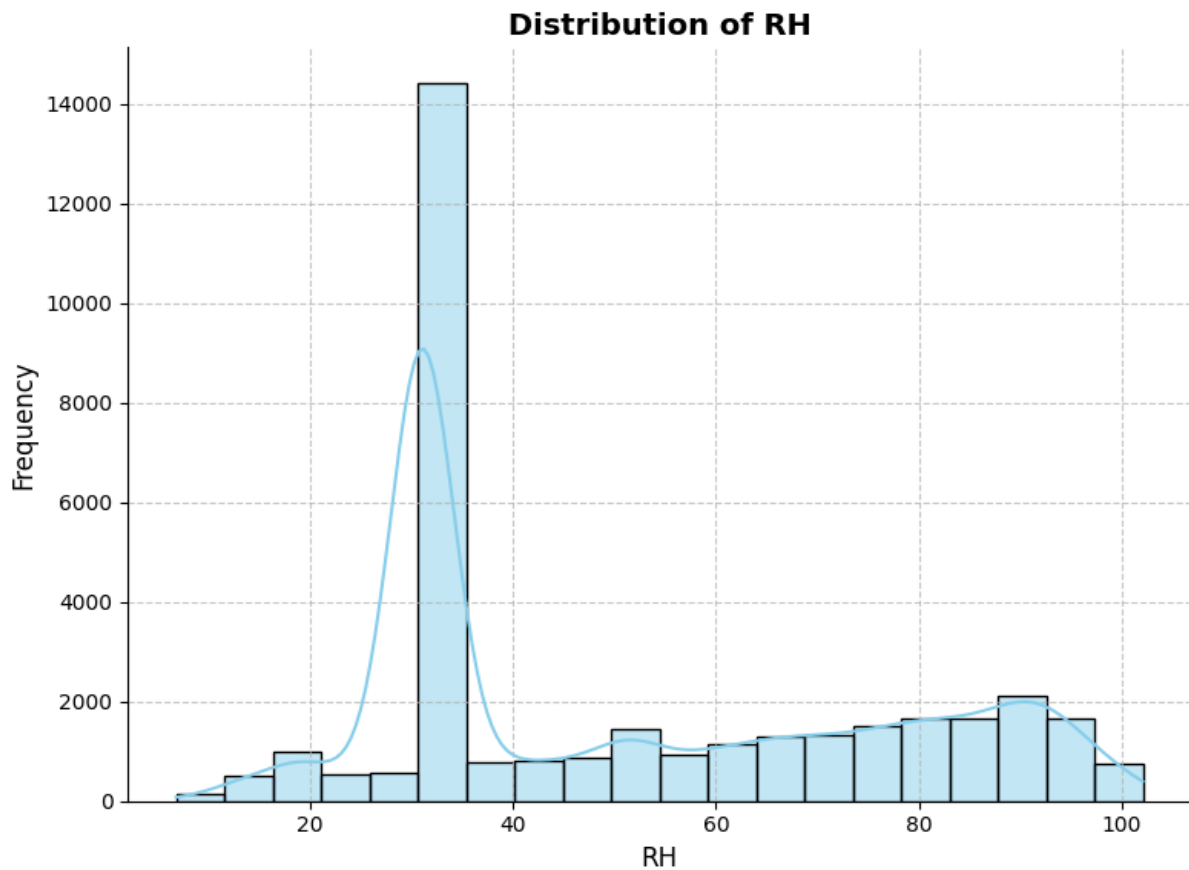


Figure 3.3: Histogram for RH (Relative Humidity)

Inference and Observations: The distribution of Relative Humidity (RH) is skewed to the right, indicating a predominantly dry environment. The peak around 30-40 percent RH suggests that most measurements fall within this range, indicating that the environment is typically dry. However, the long tail towards higher RH values indicates occasional periods of higher humidity, possibly due to specific weather conditions like rainfall, fog, or indoor humidity control systems.

3.1.2 Time Series Plots

In addition to histograms, time series plots were created for all numerical features with respect to the timestamp (**DateTime**). These plots allow us to observe trends, periodicity, and potential anomalies over time. These plots were used to analyze the behavior of features such as air temperature, soil temperature, wind speed, and other environmental factors across the timeline of the data. Some of the time series plots are shown below, along with their inferences and observations.

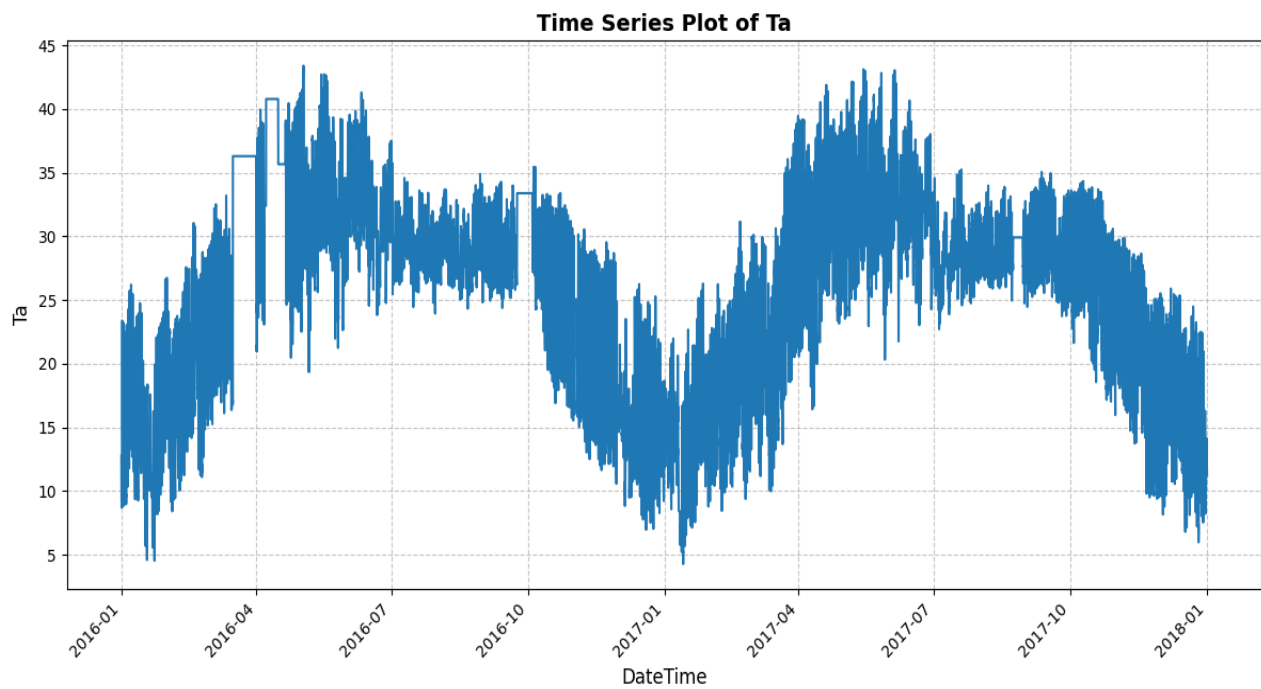


Figure 3.4: Time Series Plot of Ta (Air Temperature in °C)

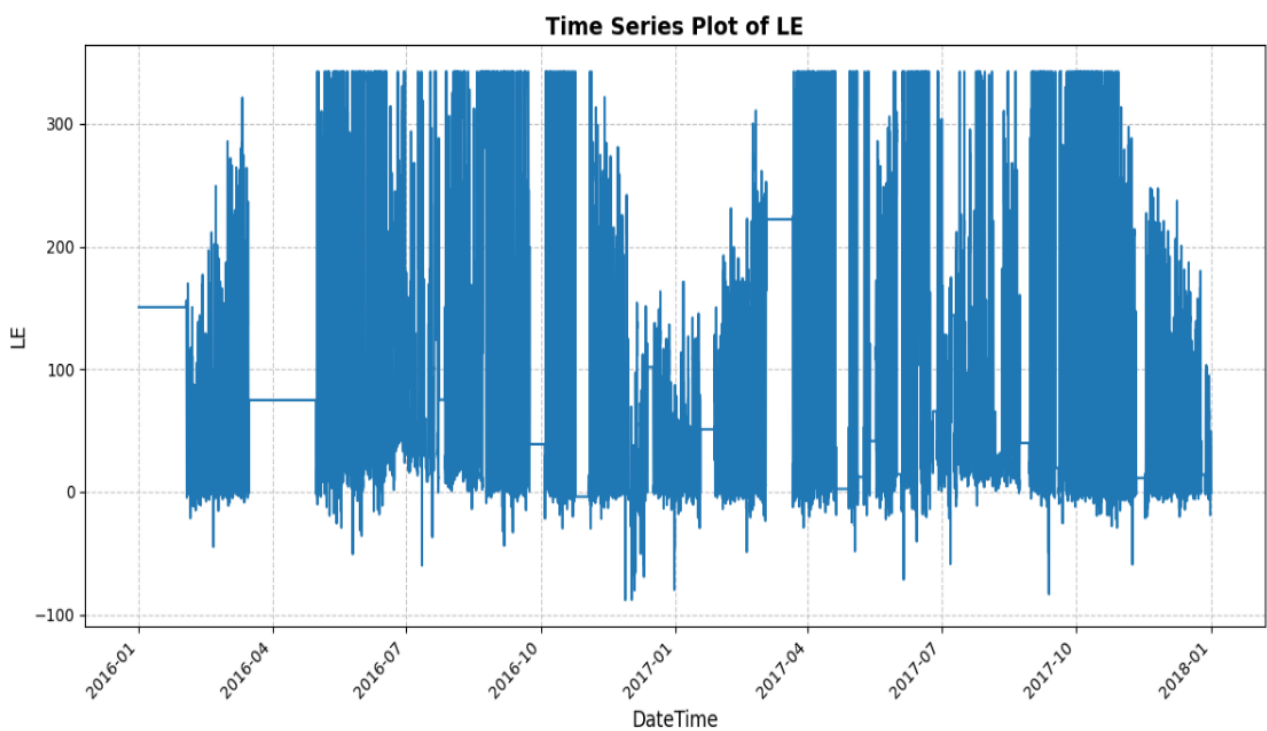


Figure 3.5: Time Series Plot of LE (Latent Heat Flux Density in W/m²)

Inference and Observations:

- **Ta (Air Temperature):** The time series plot exhibits clear seasonal variations, with higher temperatures during summer months and lower temperatures in winter months. You may also observe an overall increasing or decreasing trend over time depending on your data. Within each season, there are significant fluctuations in the values of Ta. These fluctuations can be attributed to short-term weather events such as heatwaves, cold spells, or storms. Additionally, there appears to be a slight upward trend over the entire time period, which might suggest a long-term increase in the average value of Ta.
- **LE (Latent Heat Flux Density):** The time series plot of LE exhibits significant variability and frequent zero values, suggesting potential issues with data quality or measurement errors. The lack of a discernible trend or seasonal pattern indicates that the underlying process generating the LE values is likely irregular and influenced by various factors that are not captured by time-related variables.

3.2 Multivariate Analysis

3.2.1 Heat Map / Correlation Matrix

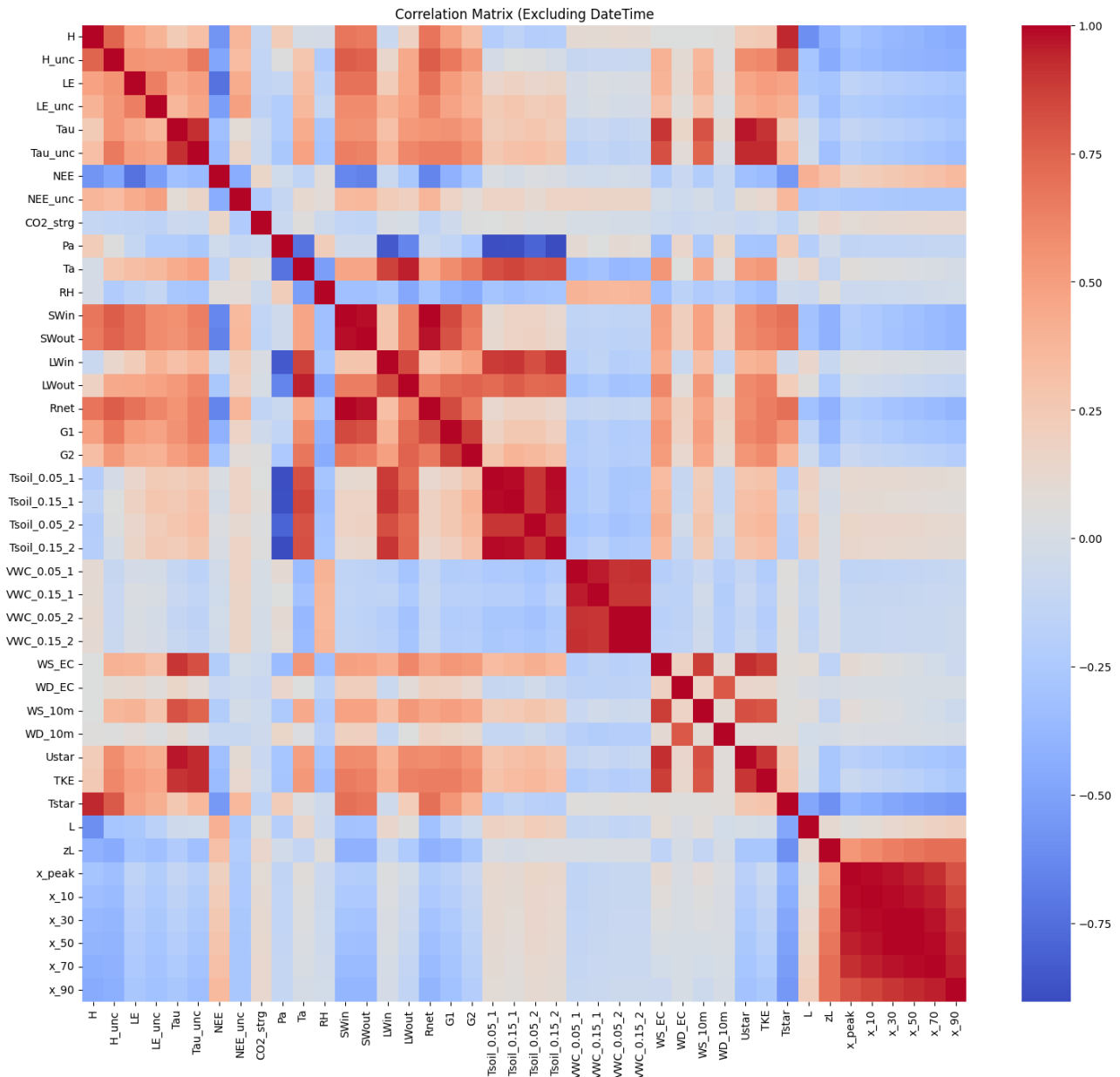


Figure 3.6: Heat Map showing the correlation between various variables. Darker colors represent stronger correlations (positive or negative).

Observations:

- Strong positive correlations are observed between:
 - **Ta** (Air Temperature) and **SWin** (Incoming Shortwave Radiation), **Tsoil_0.05_1**, **Tsoil_0.15_1**,

- Tsoil_0.05_2**, and **Tsoil_0.15_2** (Soil Temperatures). This indicates that higher solar radiation leads to an increase in both air and soil temperatures.
- **SWin** and **Rnet** (Net Radiation), showing that incoming shortwave radiation contributes significantly to net radiation.
 - **LWin** (Incoming Longwave Radiation) and **LWout** (Outgoing Longwave Radiation), suggesting that surface-emitted longwave radiation is influenced by atmospheric incoming radiation.
 - **H** (Sensible Heat Flux) and **LE** (Latent Heat Flux), indicating concurrent processes of convection and evaporation.
 - Soil temperatures at different depths (**Tsoil_0.05_1** with **Tsoil_0.15_1** and **Tsoil_0.05_2** with **Tsoil_0.15_2**), due to heat transfer through the soil profile.
- Strong negative correlations include:
 - **Ta** (Air Temperature) and **RH** (Relative Humidity), showing an inverse relationship where warmer air holds more moisture, leading to lower relative humidity.
 - **SWout** (Outgoing Shortwave Radiation) and **VWC_0.05_1** (Soil Volumetric Water Content), suggesting that wetter soils absorb more shortwave radiation and reflect less.
 - Moderate correlations:
 - Positive correlation between wind speed (**WS_EC**, **WS_10m**) and **H** (Sensible Heat Flux), where higher wind speeds enhance convective heat transfer.
 - Negative correlations between **Pa** (Barometric Pressure) and both **Ta** (Air Temperature) and **RH** (Relative Humidity).
 - Weak or no correlations:
 - Some variables, such as **NEE** (Net Ecosystem CO₂ Exchange), show weak correlations with most other variables, indicating minimal linear relationships.

3.2.2 Scatter Plots

Air Temperature (T_a) vs Relative Humidity (RH)

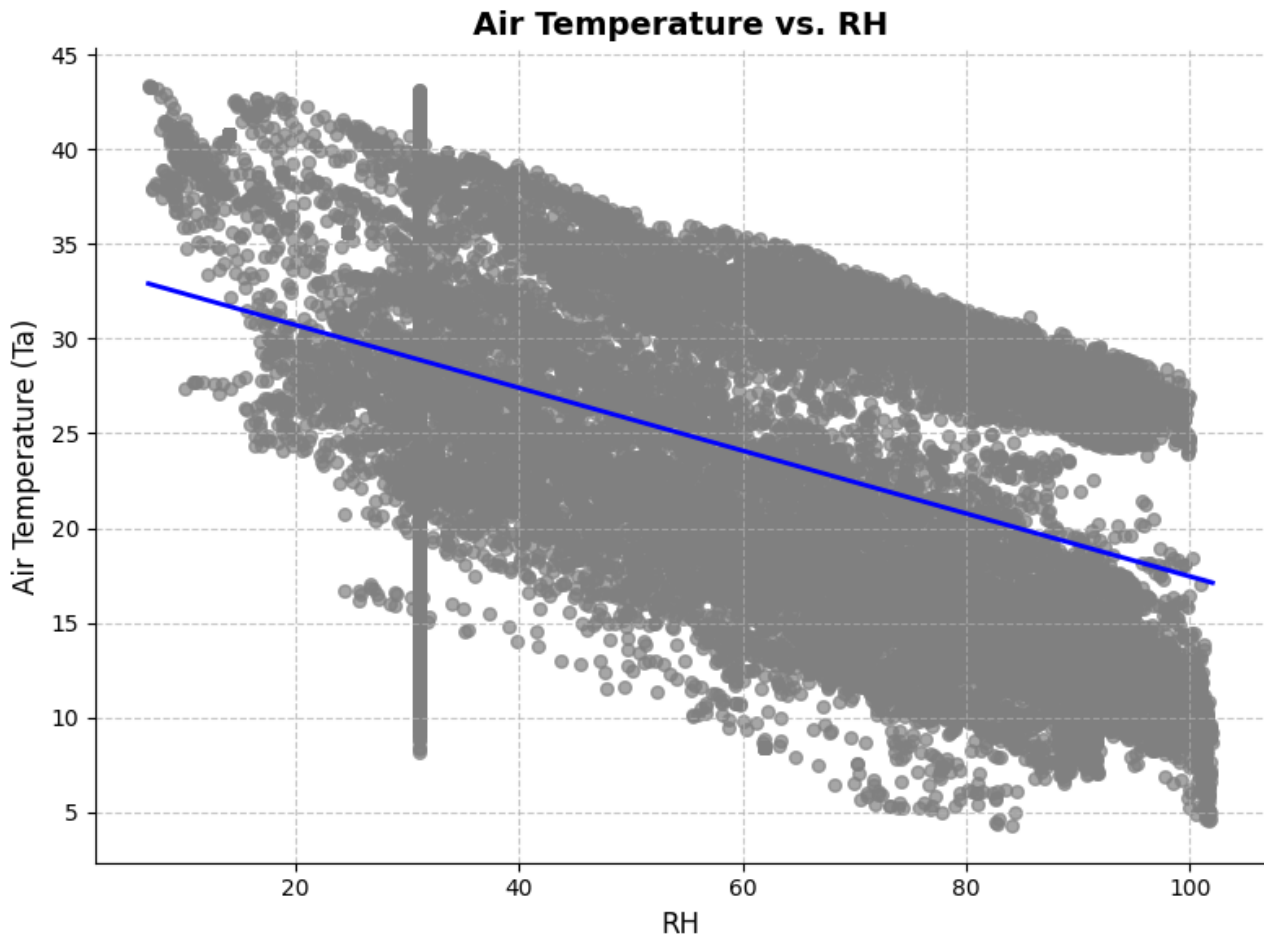


Figure 3.7: Scatter plot of Air Temperature (T_a) vs Relative Humidity (RH).

Observations:

- **General Trend:** The scatter plot shows a clear negative correlation between Air Temperature (T_a) and Relative Humidity (RH). As air temperature increases, relative humidity tends to decrease.
- **Scatter:** There is a noticeable scatter of data points around the regression line. This indicates that the relationship between T_a and RH is not perfectly linear and is influenced by other factors.
- **Distribution:** The data points are more concentrated at lower temperatures and higher relative humidity, and they become more dispersed as temperature increases and relative humidity decreases.

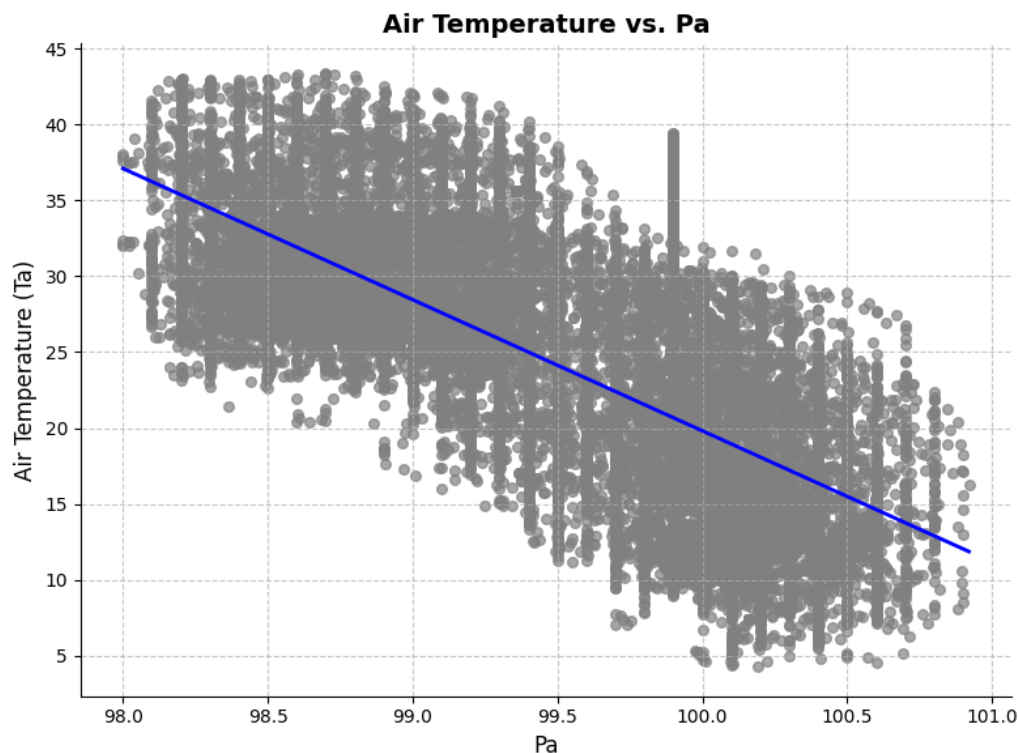
Air Temperature (T_a) vs Barometric Pressure (P_a)

Figure 3.8: Scatter plot of Air Temperature (T_a) vs Barometric Pressure (P_a).

Observations:

- **General Trend:** The scatter plot likely reveals a weak to moderate negative correlation between Air Temperature (T_a) and Barometric Pressure (P_a). This implies that as air temperature increases, barometric pressure tends to decrease, although the relationship might not be very strong.
- **Scatter:** There is considerable scatter of data points around the regression line, indicating that the relationship between T_a and P_a is not perfectly linear and is influenced by other factors.
- **Distribution:** The distribution of data points might vary depending on the specific dataset and location. In general, a slightly higher density of points is observed at lower temperatures and higher pressures, with a more dispersed distribution at higher temperatures and lower pressures.

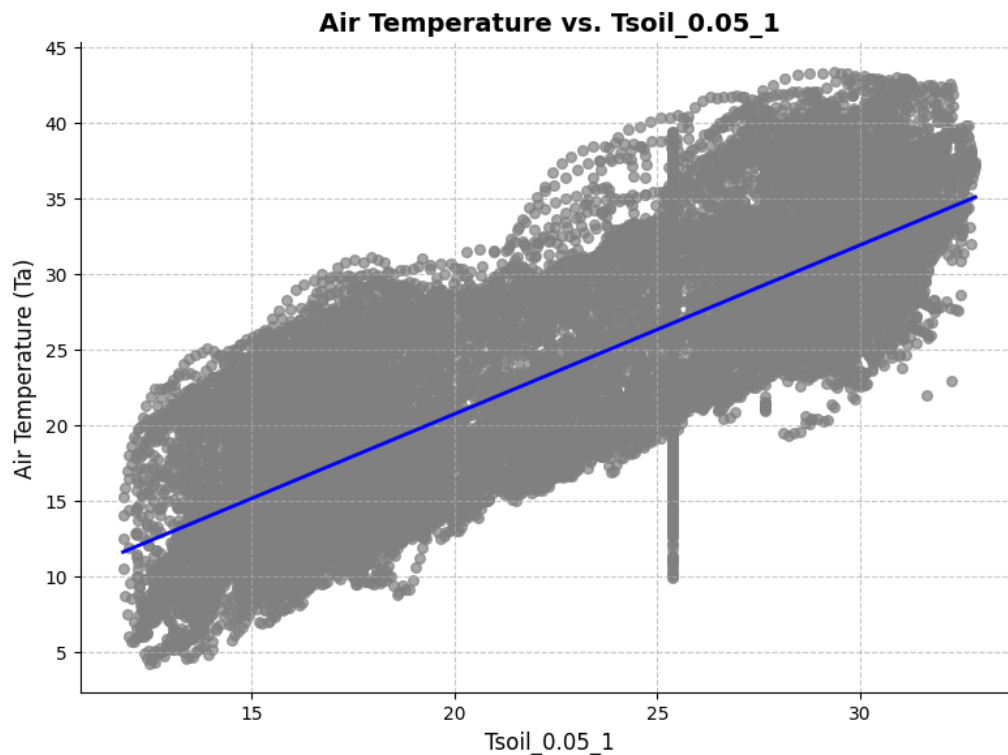


Figure 3.9: Scatter plot of Air Temperature (T_a) vs Soil Temperature at 0.05m (P_a).

Observations:

- **General Trend:** The scatter plot will likely reveal a strong positive correlation between Air Temperature (T_a) and Soil Temperature at 0.05m depth ($T_{soil.0.05.1}$). This indicates that as air temperature increases, soil temperature at that depth also tends to increase.
- **Scatter:** There might be some scatter of data points around the regression line, but the scatter is expected to be less compared to the scatter plots of T_a vs. RH or T_a vs. P_a . This suggests a relatively strong and consistent relationship between T_a and $T_{soil.0.05.1}$.
- **Distribution:** The data points are likely to be clustered along the regression line, indicating a relatively linear relationship. The distribution might be denser at moderate temperatures and become more dispersed at extreme temperatures.

Chapter 4. Feature Engineering

Feature engineering is a crucial step in the data preprocessing pipeline that transforms raw data into a format suitable for model training. It involves selecting, creating, and modifying features to enhance their relevance and predictive power for the selected analytical models. By focusing on both the individual characteristics of features and the relationships between them, feature engineering ensures that the dataset is informative and free from noise or redundancy.

This chapter explores various techniques for identifying important features, generating new features using domain knowledge, and applying transformations that optimize model performance. Each stage of feature engineering plays an integral role in improving the quality of the dataset and, consequently, the success of the predictive modeling process. Through effective feature engineering, the dataset becomes better aligned with the model's requirements, leading to more accurate and reliable predictions.

4.1 Feature Selection Based on Univariate and Multivariate Analysis

The following features were selected based on their relevance and significance derived from both univariate and multivariate analysis, including the correlation heat map, scatter plots, histograms, and time series analysis:

- **SWin (Incoming Shortwave Radiation):** Strongly correlated with air temperature and net radiation, highlighting its influence on energy balance.
- **SWout (Outgoing Shortwave Radiation):** Indicates reflective properties of the surface and its relation to soil water content.
- **LWin (Incoming Longwave Radiation):** Represents atmospheric radiation and its impact on surface temperatures.
- **LWout (Outgoing Longwave Radiation):** Depicts energy loss from the surface, closely linked with LWin.
- **G1, G2 (Soil Heat Flux at 0.03 m at locations 1 and 2):** Key variables to understand subsurface heat dynamics.
- **Tsoil_0.05_1, Tsoil_0.05_2 (Soil Temperature at 0.05 m at locations 1 and 2):** Represent temperature variations across soil depths and locations.

- **Pa (Barometric Pressure):** Demonstrates moderate correlations with air temperature and humidity, influencing weather patterns.
- **WS_EC (Mean Wind Speed):** Captures the impact of wind on heat flux and energy exchange processes.
- **WD_EC (Wind Direction from EC):** Provides directional insights into wind patterns affecting fluxes.
- **VWC_0.05_1 (Soil Volumetric Water Content at 0.05 m):** Indicates soil moisture and its role in radiation and energy absorption.
- **Ustar (Friction Velocity):** A crucial parameter for understanding momentum transfer and turbulence.
- **Tstar (Temperature Scaling Parameter):** Highlights temperature fluctuations in relation to stability and turbulence.
- **LE (Latent Heat Flux Density):** Measures the heat used for evaporation, critical for energy and water cycle studies.

Rationale for Feature Selection:

- These features were selected based on their strong correlations with target variables from univariate analysis, as well as their interactions with other features observed through multivariate analysis, including the correlation heat map.
- Scatter plots, histograms, and time series analysis were also employed to examine the distribution, trends, and temporal patterns of these features, further justifying their selection.
- The selection ensures that both individual (univariate) significance and inter-variable (multivariate) relationships are considered, capturing the most critical environmental interactions that influence energy, radiation, and flux dynamics.

4.2 Feature Creation

4.2.1 Wind Power as a Feature

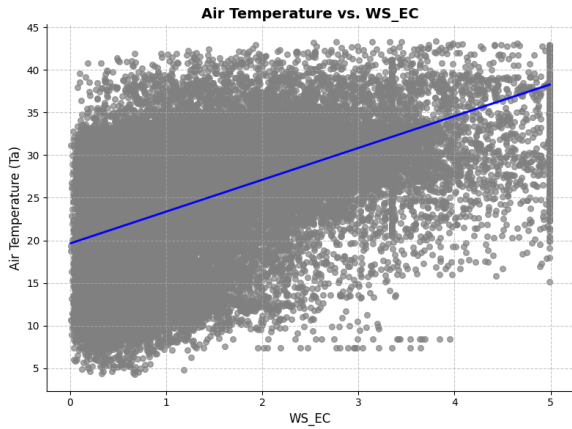


Figure 4.1: Correlation between Air Temperature (Ta) and Wind Speed (WS_EC)

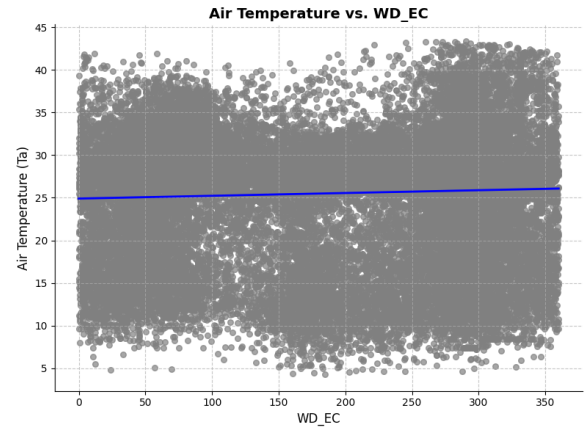


Figure 4.2: Correlation between Air Temperature (Ta) and Wind Direction (WD_EC)

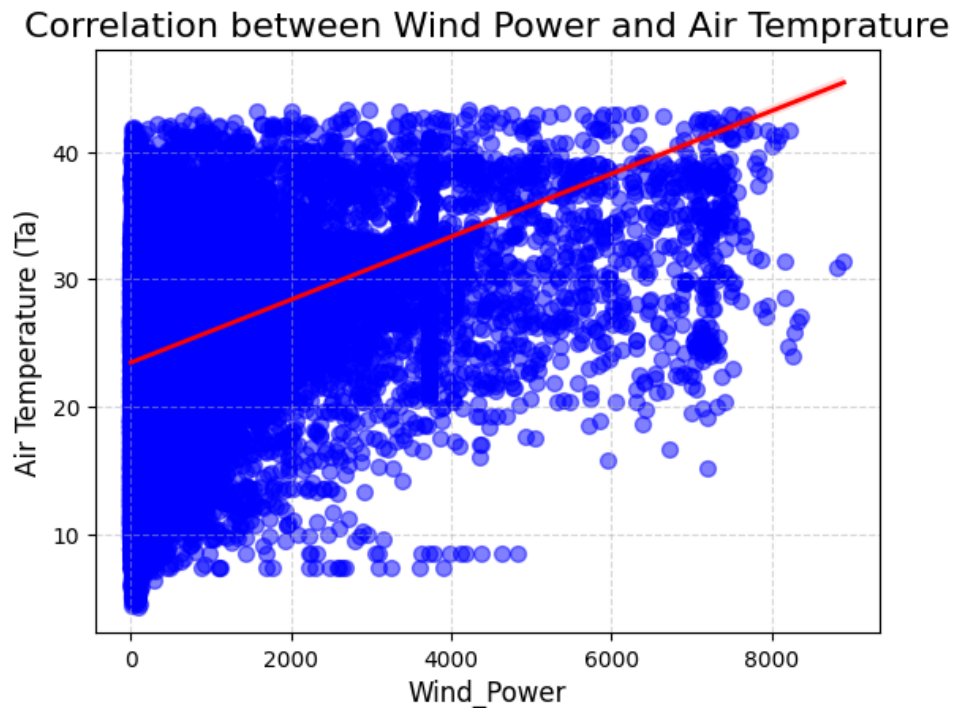


Figure 4.3: Air Temperature (Ta) vs Wind Power ($WS_EC^2 \times WD_EC$)

Rationale for Wind Power as a Feature:

- **Wind Power as a Combined Indicator:** Wind_Power, calculated as $WS_EC^2 \times WD_EC$, captures both wind speed and direction. Squaring wind speed emphasizes its impact, making

Wind_Power a useful metric for understanding the combined influence of these factors on temperature.

- **Statistical Correlation:** The correlation between Wind_Power and air temperature can reveal valuable relationships in the data. By including Wind_Power, we incorporate a combined feature that could improve model performance by accounting for variability in both wind speed and direction.
- **Modeling Considerations:** Including Wind_Power may improve the model's explanatory power by capturing a non-linear relationship with temperature. Feature importance analysis can help assess its relevance in comparison to other features.
- **Prediction Enhancement:** Using Wind_Power as a feature in predictive models could enhance their accuracy. It integrates information about wind dynamics that might not be fully captured by wind speed or direction alone, potentially leading to better temperature predictions.
- **Data-driven Approach:** The inclusion of Wind_Power is guided by its statistical relationship with the target variable. This approach leverages data analysis techniques, such as correlation analysis and feature selection, to identify and include features that improve model performance.

4.2.2 Net Shortwave and Longwave Radiation as a feature

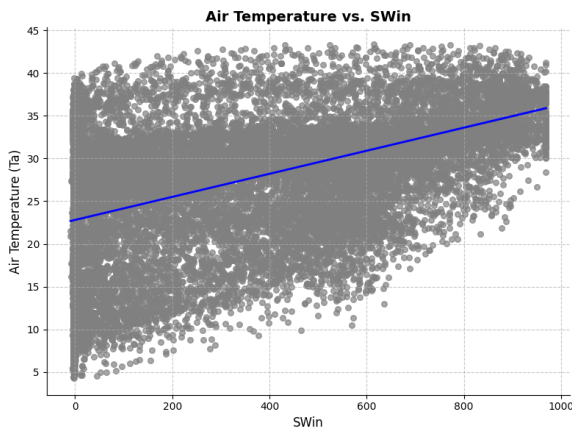


Figure 4.4: Incoming Shortwave Radiation (SWin)

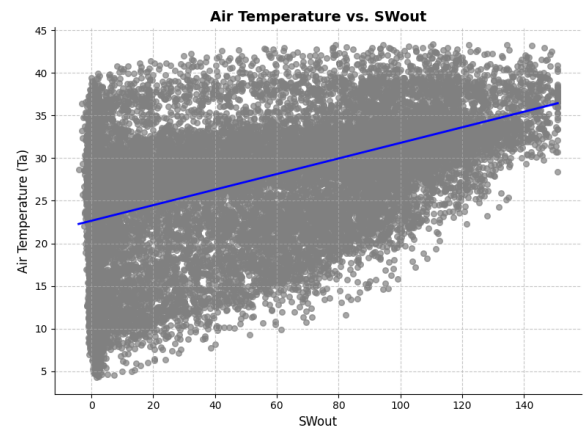


Figure 4.5: Outgoing Shortwave Radiation (SWout)

Correlation between Net Short Wave and Air Temperature

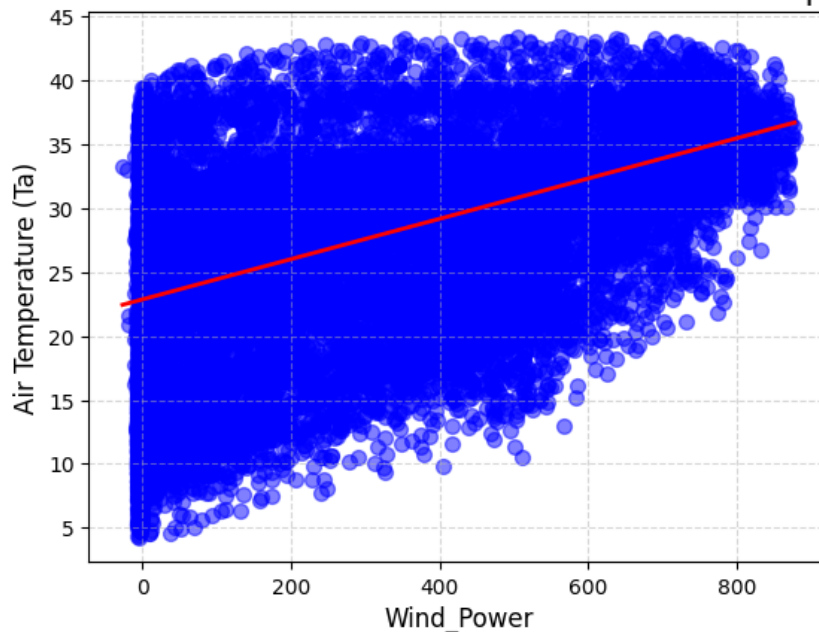


Figure 4.6: Net Shortwave Radiation (Net_SW)

Rationale for Net Shortwave as Feature:

- **Net Shortwave Radiation (Net_SW):** Net_SW represents the difference between incoming short-wave radiation from the sun (SWin) and outgoing shortwave radiation reflected by the Earth's surface (SWout). It essentially quantifies the amount of solar energy absorbed by the surface. This absorbed energy contributes to heating the surface and the surrounding air.

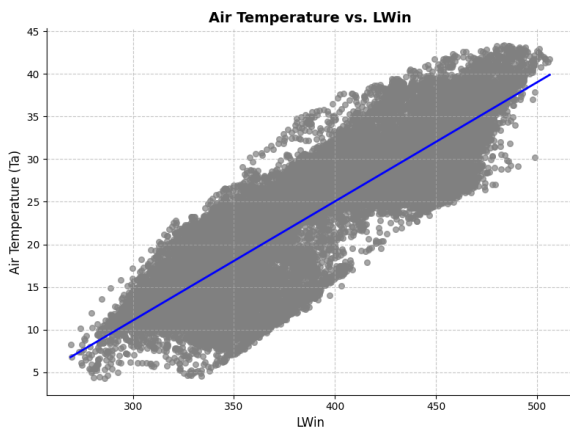


Figure 4.7: Incoming Longwave Radiation (LWin)

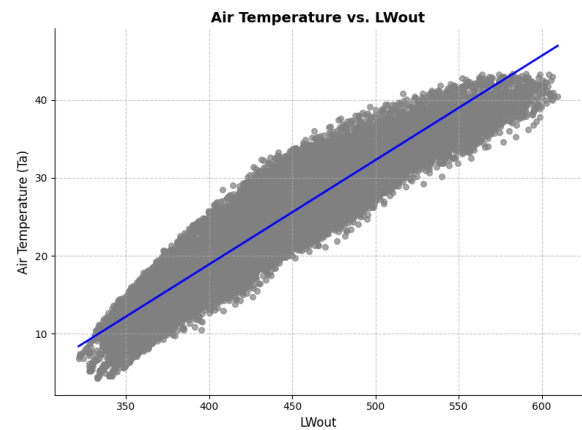


Figure 4.8: Outgoing Longwave Radiation (LWout)

Correlation between Net Long Wave and Air Temperature

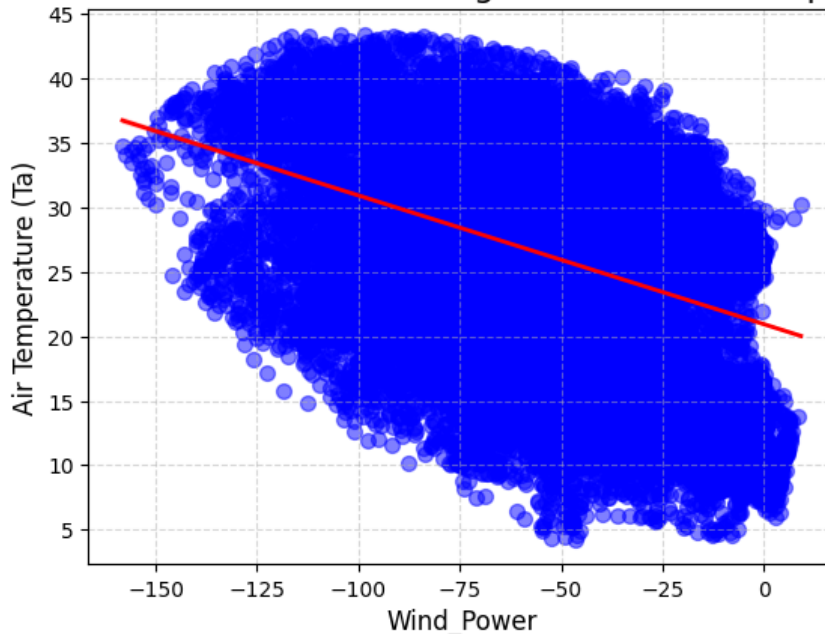


Figure 4.9: Net Longwave Radiation (Net_LW)

Rationale for Net Longwave Radiation as a Feature:

- **Net Longwave Radiation (Net_LW):** Net_LW represents the difference between incoming longwave radiation from the atmosphere (LWin) and outgoing longwave radiation emitted by the Earth's surface (LWout). It quantifies the net exchange of thermal radiation between the surface and the atmosphere. This exchange is influenced by factors such as air temperature, humidity, and cloud cover.

Conclusion for Net Shortwave and Longwave

- **Radiation Balance:** The Earth's surface and atmosphere constantly exchange energy in the form of radiation. The balance between incoming and outgoing radiation determines the net radiation, which is a crucial factor in driving temperature changes.

- **Relationship with Air Temperature:** Both Net_SW and Net_LW have a strong influence on air temperature. Positive Net_SW values indicate a net gain of solar energy, leading to surface and air warming. Positive Net_LW values indicate a net gain of thermal radiation from the atmosphere, further contributing to warming. Conversely, negative values represent a net loss of energy, leading to cooling.

4.2.3 Violin Plots of Air Temperature by Day/Night and Season

In this section, we present the violin plots for Air Temperature (Ta) by the categorical columns "Day/Night" and "Season." These plots provide valuable insights into how temperature distributions vary across different temporal periods.

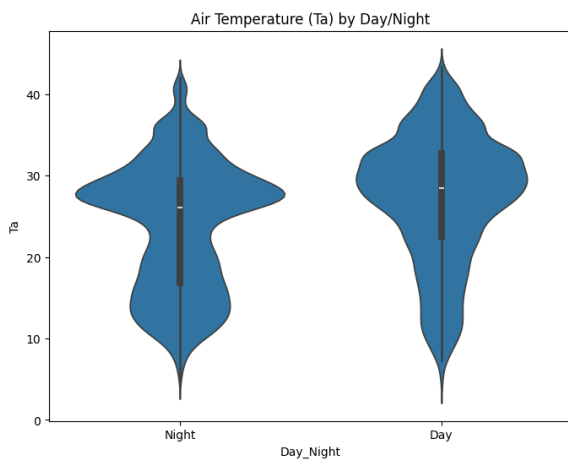


Figure 4.10: Air Temperature (Ta) by Day/Night

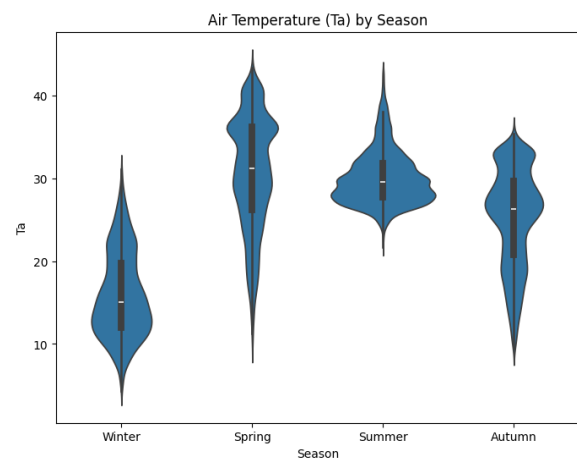


Figure 4.11: Air Temperature (Ta) by Season

Observations from Violin Plots:

- **1. Air Temperature (Ta) by Day/Night:**
 - **Difference in Distributions:** The violin plot for Ta by Day/Night will likely show distinct distributions for the two categories. The "Day" category will generally have a wider distribution with higher median and mean temperatures compared to the "Night" category.
 - **Diurnal Variation:** This observation confirms the expected diurnal variation in air temperature, where temperatures are typically higher during the day due to solar radiation and lower at night due to radiative cooling.
 - **Skewness:** The distribution for the "Day" category might be slightly skewed towards higher temperatures, indicating the occurrence of warmer daytime temperatures. The "Night" category might have a more symmetrical distribution or a slight skew towards lower temperatures.
- **2. Air Temperature (Ta) by Season:**
 - **Seasonal Differences:** The violin plot for Ta by Season will likely show clear differences in distributions across the four seasons (Winter, Spring, Summer, Autumn).

- **Temperature Range:** The "Summer" category will generally have the highest median and mean temperatures, followed by "Spring" and "Autumn," with "Winter" having the lowest temperatures. This reflects the seasonal variations in solar radiation and the length of daylight hours.
- **Distribution Shape:** The distributions for different seasons might have varying shapes. For example, the "Summer" distribution might be wider and more skewed towards higher temperatures, while the "Winter" distribution might be narrower and more symmetrical.

Relevance of Categorical Columns:

- **Capturing Temporal Variations:** The violin plots clearly demonstrate the relevance of the categorical columns, `Day_Night` and `Season`, in capturing temporal variations in air temperature. These categories represent distinct time periods with different temperature patterns, allowing us to better understand and model temperature dynamics.

4.3 Feature extraction

4.3.1 Relevance of Dimensionality Reduction

Curse of Dimensionality: In machine learning, as the number of features increases, the data becomes sparse, making it difficult for models to find meaningful patterns and generalize effectively.

Reducing Complexity: Dimensionality reduction techniques like PCA reduce the number of features while preserving essential information, simplifying the data and enhancing model performance.

Removing Redundancy: PCA identifies and eliminates highly correlated features that can lead to overfitting, ensuring that only unique, informative features are retained.

Visualization: By reducing dimensionality, we can visualize data more easily, uncovering patterns and relationships that are not apparent in high-dimensional spaces.

4.3.2 How PCA Reduces Dimension

PCA is a linear technique that identifies principal components—orthogonal combinations of original features that capture the maximum variance.

1. **Eigenvalue Decomposition:** PCA involves decomposing the covariance matrix to find eigenvectors (directions of principal components) and eigenvalues (variance explained by each component).
2. **Component Selection:** Components are ranked by eigenvalues, with those capturing the most variance selected as new features.
3. **Projection:** The original data is projected onto the lower-dimensional space defined by these principal components, resulting in a dataset with fewer features.

Summary: In summary, dimensionality reduction techniques like PCA are essential for improving model performance and data visualization while addressing challenges posed by high-dimensional datasets.

4.3.3 PCA Table and Analysis

	PC1	PC2	PC3	PC4	PC5	PC6	PC7
0	-3.279658	-2.118623	0.915993	0.908147	-0.830622	-0.222978	-1.054357
1	-3.287485	-2.213112	0.993583	0.802352	-0.912301	-0.216196	-0.907788
2	-3.260849	-2.219052	0.941224	1.007417	-0.910446	-0.226061	-0.898955
3	-3.324172	-2.260553	0.985739	0.875959	-0.918831	-0.186172	-0.928877
4	-3.293317	-2.170063	0.922810	1.126876	-0.873063	-0.201875	-1.002305

Figure 4.12: PCA Transformation Table

PCA Table Analysis

- **PC1, PC2, PC3, ...:** These are the labels for the principal components. They represent the new axes in the transformed feature space after applying PCA.
- **Values:** The values in each column represent the coordinates of the data points in the new feature space defined by the principal components. Each data point is now represented by a set of 7 values, one for each principal component.
- **What the Numbers Show:**
 - **Reduced Dimensionality:** The original dataset with many features has been transformed into a new dataset with only 7 features (principal components). This results in a reduced dimensionality, making the data easier to analyze.
 - **Variance Explained:** The first principal component (PC1) captures the most variance in the data, followed by PC2, PC3, and so on. The amount of variance explained by each component is proportional to its corresponding eigenvalue.
 - **Data Representation:** The values in the table represent how each data point is positioned in the new feature space. Data points with similar values for a particular principal component are considered to be similar in that dimension of the data. This allows for a more meaningful analysis by clustering similar data points based on principal components rather than the original features.

Chapter 5. Model fitting

Model fitting is the process of training a machine learning model on a given dataset to uncover the underlying relationships between the features and the target variable. The objective is to identify the model parameters that minimize the discrepancy between the predicted and actual values of the target variable. This process typically involves three key steps:

Choosing a Model: The first step is to select an appropriate model based on the characteristics of the data and the specific prediction task at hand. This involves considering various model types such as linear or non-linear regression, decision trees, or ensemble methods, each with their strengths and limitations.

Training the Model: Once a model is chosen, the next step is to train it by feeding it the training data. During this phase, the model learns by adjusting its internal parameters to reduce the error between the predicted values and the true values, typically by using optimization algorithms like gradient descent.

Evaluating the Model: After the model has been trained, its performance is assessed using unseen data (test set). This evaluation is crucial to ensure the model generalizes well to new data and is not overfitting to the training data. Performance metrics such as Mean Absolute Error (MAE), Mean Squared Error (MSE), R-squared (R^2), and Root Mean Squared Error (RMSE) are commonly used to quantify the model's predictive accuracy and reliability.

By completing these steps, the model fitting process aims to create a robust and reliable predictive model that can effectively forecast future values of the target variable.

5.1 ML algorithms

5.1.1 Description of Models Used

In this analysis, four different regression models were used to predict the target variable. Each model has its unique characteristics and approach to solving the regression problem.

- **Linear Regression:** A simple linear model that assumes a linear relationship between the features and the target variable. It finds the best-fitting line by minimizing the sum of squared errors between the predictions and actual values. This model is easy to interpret and works well when the relationship between features and target is linear.
- **Random Forest:** An ensemble learning method that combines multiple decision trees to make predictions. It builds a forest of trees, each trained on a random subset of the data, and averages

their predictions to improve accuracy and robustness. Random Forest is effective for capturing non-linear relationships and handling large datasets with multiple features.

- **Gradient Boosting:** Another ensemble learning method that builds a sequence of decision trees. Each tree corrects the errors of the previous tree. Gradient Boosting uses a gradient descent algorithm to minimize the loss function and improve the model's predictions, often resulting in high predictive accuracy, especially for complex datasets.
- **Support Vector Regression (SVR):** A powerful model that uses support vectors to define a hyperplane that best separates the data points. SVR is effective for handling non-linear relationships and high-dimensional data. It works by finding a boundary that maximizes the margin between the support vectors while maintaining the error within a specified threshold.

5.1.2 Best Model for Transformed Data

Based on the evaluation results (assuming you have evaluated these models on the transformed data), the best model is determined by comparing the performance metrics, such as Mean Absolute Error (MAE), Mean Squared Error (MSE), Root Mean Squared Error (RMSE), and R-squared (R2) score. The model with the lowest MAE, MSE, RMSE, and the highest R2 score is considered the best for predicting the transformed data.

	MAE	MSE	R2	RMSE
Linear Regression	1.923738	5.686760	0.908785	2.384693
Random Forest	0.805801	1.412486	0.977344	1.188480
Gradient Boosting	1.347791	3.310910	0.946893	1.819591
Support Vector Regression	0.992213	1.946833	0.968773	1.395290

Figure 5.1: Model Performance Comparison

Chapter 6. Conclusion & Future scope

6.1 Conclusions (Findings And Observations)

The project effectively explored the potential of meteorological and soil physics data in predicting air temperature (T_a), offering insights into the interplay between various environmental variables. Key findings include:

- **Understanding Relationships:** The analysis successfully highlighted key relationships, such as the influence of relative humidity (RH) and radiation on air temperature, essential for accurate forecasting.
- **Dimensionality Reduction:** Principal Component Analysis (PCA) streamlined the dataset by reducing redundant features while preserving critical information. This optimization enhanced model efficiency without compromising prediction accuracy.
- **Model Performance:**
 - Among the models tested, *Random Forest Regressor* achieved the best results, with the lowest MAE , MSE , $RMSE$, and the highest R^2 (0.977344), indicating its robustness in handling complex, non-linear relationships in meteorological data.
 - *Support Vector Regression (SUR)* also performed admirably, showcasing its suitability for non-linear datasets.
- **Visualization Insights:** Exploratory techniques like scatter plots and violin plots provided a deeper understanding of feature distributions and interactions, informing effective feature selection and model design.
- **Significance of Meteorology:** Accurate temperature predictions are critical for climate studies, agriculture, and resource management. This study underscores the value of combining meteorological data with advanced analytics for reliable forecasting.

6.2 Challenges

Finding a Good Dataset: One of the key challenges was sourcing a high-quality, relevant dataset that provided comprehensive meteorological and soil physics data. The dataset had to cover various environmental factors, including air temperature, relative humidity, and radiation, to support accurate prediction modeling. Additionally, the dataset needed to be large and detailed enough to ensure reliable model performance.

Data Quality and Preprocessing: The data quality posed significant challenges, including missing values, outliers, and inconsistencies. Proper data cleaning and preprocessing were crucial steps to ensure that the dataset was suitable for analysis. We had to handle missing values through imputation and outlier detection techniques to preserve the integrity of the data.

Feature Selection and Extraction from 45 Features: The dataset contained a large number of features, and selecting the most relevant ones for model training was challenging. Identifying key variables that had a significant impact on predicting air temperature while discarding irrelevant or redundant features was crucial for improving model performance and efficiency.

Dimensionality Reduction using PCA: To address the high dimensionality of the data, we applied Principal Component Analysis (PCA) to reduce the number of features while retaining essential information. However, this posed a challenge in terms of balancing the loss of information with the gain in model efficiency. It required careful tuning and evaluation to ensure that the reduced features still captured the critical patterns in the data.

Overfitting and Underfitting: Another challenge was ensuring that the models did not overfit or underfit the data. Overfitting led to models that performed well on the training data but failed to generalize to new data, while underfitting resulted in poor performance across all datasets. Careful tuning of model hyperparameters and validation techniques were essential to strike the right balance.

6.3 Future Scope

- **Enhanced Model Generalization:** Experimenting with advanced algorithms such as deep learning, ensemble methods, or transformer-based models can improve accuracy for complex datasets. Transfer learning techniques leveraging pre-trained models on meteorological data also hold potential for further refinement.
- **Real-Time Prediction:** Developing frameworks for live meteorological data processing will enable real-time temperature predictions, benefiting applications like weather forecasting, agriculture, and urban planning. Incorporating Internet of Things (IoT) devices will support continuous data collection and updates.
- **Geographic Diversification:** Extending the model to datasets from diverse regions and climates

can test its scalability and robustness, while comparative studies between regions will help understand geographical influences on predictive accuracy.

- **Climate Change Analysis:** Utilizing the model for long-term climate trend predictions can support policy-making and environmental conservation. The model can also help assess the impact of global warming on temperature patterns and variability.
- **Educational and Collaborative Use:** Sharing the project as an open-source toolkit will encourage research and learning. Collaborating with meteorological agencies will refine the model using expert insights and data resources.

In conclusion, these future directions highlight the versatility of the project in advancing predictive modeling in meteorology. By adopting advanced technologies, broadening the dataset scope, and fostering interdisciplinary collaborations, this work paves the way for significant contributions to climate prediction and resource management, addressing pressing global environmental challenges.

Group Contribution

Ayush Popshetwar, 202201412

Project Report, ipynb file

Parjanya Rajput, 202201115

ipynb file, Project Report

Vats Shah, 202201417

Presentation Slides, Project Report

Short Bio

1. **Parjanya Rajput** is a passionate engineering undergrad student at renowned DA-IICT institute. Interested in building Mobile applications. Academically he is interested in Machine Learning and Data related courses and that's why he have taken EDA(IT-462) and Machine Learning(IE-406) as electives for 5th sem. Moreover he is interested in hardware and electronics too.

In addition to his technical skills, Parjanya is also involved in co-curricular activities and currently holds the Convener Position at Microsoft Student Technical Club and Dy. Convener Position at Electronics Hobby Club at DA-IICT.

Outside of work, Parjanya enjoys watching anime and playing table tennis

2. **Ayush Popshetwar** is pursuing a Bachelor's degree in B.Tech Information and Communication Technology at DA-IICT. He is actively involved in various extracurricular activities, including being the Deputy Convenor of Film Club and Quiz Club of DA-IICT. In addition his academic pursuits, he is passionate about web development and is currently learning more about it. He enjoys programming and has a strong interest in database man-

agement systems.

He is also keen on exploring other academic subjects related to technology and computer science, including machine learning, data science, and software engineering. He is constantly looking to expand their knowledge by working on projects and collaborating with peers. Outside of academics, He enjoys playing sports such as athletics, badminton and football, and playing the drums.

3. **Vats Shah** is a passionate software engineering undergrad student. He is pursuing a Bachelor's degree in Computer Science from Dhirubhai Ambani Institute of Information and Communication Technology. He is proficient in programming languages such as Python, C++. In addition to academic pursuits, he is interested in learning about web development and is currently learning more about it. He is constantly looking to expand his knowledge by working on projects and collaborating with peers.

Outside of academics, Vats Shah enjoys playing the casino, watching tv shows and volunteering.

References

- [1] Energy and Carbon Dioxide Fluxes, Meteorology, and Soil Physics Observed at IN-COMPASS Land Surface Stations in India (2016-2017). Retrieved from <https://www.data.gov.uk/dataset/91e0c1e4-0e4d-42fc-b37e-70a8d1704e28/energy-and-carbon-dioxide-fluxes-meteorology-and-soil-physics-observed-at-incompass>
- [2] Dataset Supporting Document. Retrieved from <https://docs.google.com/document/d/13uJ5NBTA6vZm6BIQevCSyMcUdDv0lZT/edit?usp=sharing&oid=117142123167681849616&rtpof=true&sd=true>
- [3] Comparison between J48 Decision Tree, SVM, and MLP in Weather Forecasting. *ResearchGate*. Retrieved from https://www.researchgate.net/publication/344868685_Comparison_between_J48_Decision_Tree_SVM_and_MLP_in_Weather_Forecasting
- [4] YouTube Video Reference: *Weather Forecasting using Machine Learning*. Retrieved from <https://www.youtube.com/watch?v=FgakZw6K1QQ&t=1125s>
- [5] YouTube Video Reference: *Introduction to Weather Forecasting*. Retrieved from <https://www.youtube.com/watch?v=baqxB04PhI8&t=424s>