TEMPERATURE PREDICTION MODEL

A project report submitted to ICT Academy of Kerala

in partial fulfilment of the requirements

for the certification of

CERTIFIED SPECIALIST

IN

DATA SCIENCE & ANALYTICS

Submitted by,

GOURI B

ICT ACADEMY OF KERALA

THIRUVANANTHAPURAM, KERALA, INDIA

Nov 2022

List of Figures

List of Abbreviations

EDA	Exploratory Data Analysis
MAE	Mean Absolute Error
VIF	Variance Inflation Factor
PDP	Partial Dependence Plots
MSE	Mean Square Error
API	Application Programming Interface

Table of Contents

Abstract

This project presents a Machine Learning (ML)-based model for predicting temperature across different regions in India. The model utilizes key meteorological and geographical features to enhance prediction accuracy. The predicted temperature data can be beneficial for agricultural planning, climate awareness, and disaster mitigation strategies. Various statistical analyses, feature selection techniques, and model evaluation metrics are employed to ensure reliability.

This report includes the processes undergone to build and train a machine learning model for predicting temperatures of regions/cities in India. Data analytics and ML algorithms are used to ensure accurate prediction of temperature. Exploratory Data Analysis, preprocessing, encoding, scaling are some of the data analysis tools used in this project. A web interface which is user-friendly is also developed as the end result, incorporating the predictions made by the model.

Temperature prediction plays a crucial role in various industries, including agriculture, healthcare, disaster management, and energy production. An accurate prediction model can help farmers schedule irrigation, prevent crop loss, and plan farming activities. Organizations can utilize temperature predictions for mitigating natural disasters, preventing heat-related illnesses, and improving public health safety measures. This report details the methodology, data analysis, model training, evaluation, and deployment strategies used to develop an efficient temperature prediction model.

1. Problem Definition

1.1 Overview

Climate change is a threat which is tightening its grip around the world at an alarming rate. It is also the root of concern for farmers who depends largely on temperature and rainfall for starting their crop season. In this scenario, a temperature prediction app can greatly help them in knowing the temperature of a specific region and plan their cropping season accordingly. Although, public health organizations and disaster management organizations can leverage this app better to know the temperature of an area and give out warnings or mitigate disasters and public health crises.

Traditional methods of temperature prediction rely on physical models and statistical techniques, which often lack adaptability and fail to generalize well across diverse climate zones. Machine learning techniques provide an alternative by learning from historical patterns and real-time data to make more accurate forecasts. The problem addressed in this project is to identify the most relevant features affecting temperature and build an ML model that can predict it effectively.

1.2 Problem Statement

The aim of this project is to build a machine learning model which can accurately predict temperature of different regions in our country. This can be used by laymen and organizations alike to know the temperature of a particular region in India. The prediction is crucial for mitigating climate-related risks, optimizing agricultural decisions, and enhancing preparedness against extreme weather conditions.

The objective is to build a predictive model that estimates temperature for different regions in India based on meteorological and geographical data.

2. Introduction

Temperature prediction plays a crucial role in various fields, including agriculture, healthcare, transportation, and energy management. Accurate and timely temperature forecasting helps in mitigating risks related to extreme weather events, optimizing resource management, and enhancing decision-making processes. With the rapid advancements in machine learning and data science, predicting temperature using historical data and computational techniques has become increasingly effective.

The primary objective of this project is to build a robust model capable of predicting temperature patterns based on historical weather data. Data preprocessing, feature engineering, and model selection will be integral components of the project to ensure the best possible performance.

This project aims to develop an ML model that forecasts temperature using multiple environmental and geographical factors. Farmers can use this model for irrigation scheduling, while organizations can leverage it for disaster preparedness and public health safety. Furthermore, temperature prediction can be essential in energy sector planning, tourism management, and environmental research.

The study focuses on a dataset containing historical and real-time weather records from multiple regions in India. The approach involves preprocessing the dataset, analysing feature importance, training different ML models, and selecting the best-performing one. A comprehensive evaluation ensures that the selected model can be deployed for real-time applications, improving decision-making across different domains.

The deployment of the temperature prediction model enables real-time predictions and accessible insights. Through the use of cloud platforms and user-friendly interfaces, the model will be made available for various applications such as weather forecasting systems, agriculture planning, and energy consumption management.

3. Literature Survey

Temperature prediction plays a crucial role in fields like agriculture, climate science, and disaster management. With advancements in computational tools and data availability, researchers have developed numerous models to improve temperature forecasting accuracy.

Conventional methods like the ARIMA (Auto-Regressive Integrated Moving Average) model have been widely used for time series analysis in temperature prediction. These models rely on historical data and statistical techniques to identify patterns. However, their accuracy diminishes when dealing with non-linear and complex datasets, limiting their applicability in dynamic climatic conditions.

Recent studies have explored the potential of machine learning techniques such as Support Vector Machines (SVMs), Decision Trees, and Artificial Neural Networks (ANNs). For instance, Goyal et al. (2020) demonstrated that ANNs could effectively model non-linear relationships in temperature data, outperforming traditional statistical methods. Similarly, the use of Random Forest models has shown robustness in handling large datasets and capturing important features like humidity, wind speed, and solar radiation.

Deep learning models, particularly Recurrent Neural Networks (RNNs) and Long Short-Term Memory (LSTM) networks, have been employed for their ability to capture temporal dependencies. Studies by Kumar et al. (2021) highlight that LSTMs are particularly suited for temperature prediction as they can process sequential data while minimizing issues like vanishing gradients. These models achieve superior accuracy, especially when combined with ensemble methods.

4. Dataset

- Source: Kaggle (Indian Weather Repository - Live Data)
- Size:
  - 15,868 rows and
  - 42 columns
- Features used:
  - Predictors: Precipitation (mm), Humidity (%), Pressure (mb), Latitude, Longitude, Wind Speed (kph), Wind Direction
- Target variable: Temperature (Celsius)

```
df = pd.read_csv("/content/IndianWeatherRepository.csv")
df
```

country	location_name	region	latitude	longitude	timezone	last_updated_epoch	last_updated	temperature_celsius	temperature_fahrenheit	...	air_quality_PM2.5	air_quality_
India	Ashoknagar	Madhya Pradesh	24.57	77.72	Asia/Kolkata	1693286100	29-08-2023 10:45	27.5	81.5	...	12.6	
India	Raisen	Madhya Pradesh	23.33	77.80	Asia/Kolkata	1693286100	29-08-2023 10:45	27.5	81.5	...	10.7	
India	Chhindwara	Madhya Pradesh	22.07	78.93	Asia/Kolkata	1693286100	29-08-2023 10:45	26.3	79.3	...	16.8	
India	Betul	Madhya Pradesh	21.86	77.93	Asia/Kolkata	1693286100	29-08-2023 10:45	25.6	78.1	...	4.9	
India	Hoshangabad	Madhya Pradesh	22.75	77.72	Asia/Kolkata	1693286100	29-08-2023 10:45	27.2	81.0	...	11.4	
...	...	...	...	...	...	...	...	...	...	...	...	
India	Niwari	Uttar Pradesh	28.88	77.53	Asia/Kolkata	1695680100	26-09-2023 03:45	27.0	80.6	...	283.4	
India	Saitual	Mizoram	23.97	92.58	Asia/Kolkata	1695680100	26-09-2023 03:45	21.3	70.4	...	16.3	
India	Ranipet	Tamil Nadu	12.93	79.33	Asia/Kolkata	1695680100	26-09-2023 03:45	25.6	78.0	...	6.0	
India	Tenkasi	Tamil Nadu	8.97	77.30	Asia/Kolkata	1695680100	26-09-2023 03:45	22.1	71.8	...	4.1	

```
[104] df['region'].unique()

array(['Madhya Pradesh', 'Uttar Pradesh', 'Orissa', 'Rajasthan',
       'Gujarat', 'Himachal Pradesh', 'Chhattisgarh', 'Jammu and Kashmir',
       'Daman and Diu', 'Dadra and Nagar Haveli', 'Andhra Pradesh',
       'Jharkhand', 'Bihar', 'West Bengal', 'Maharashtra', 'Haryana',
       'Chandigarh', 'Goa', 'Andaman and Nicobar Islands',
       'Arunachal Pradesh', 'Assam', 'Puducherry', 'Kerala', 'Mizoram',
       'Manipur', 'Nagaland', 'Tripura', 'Karnataka', 'Uttarakhand',
       'Lakshadweep', 'Punjab', 'Tamil Nadu', 'Delhi'], dtype=object)

df["region"].nunique()

33

[105] df['wind_direction'].nunique()

16

df.describe()
```

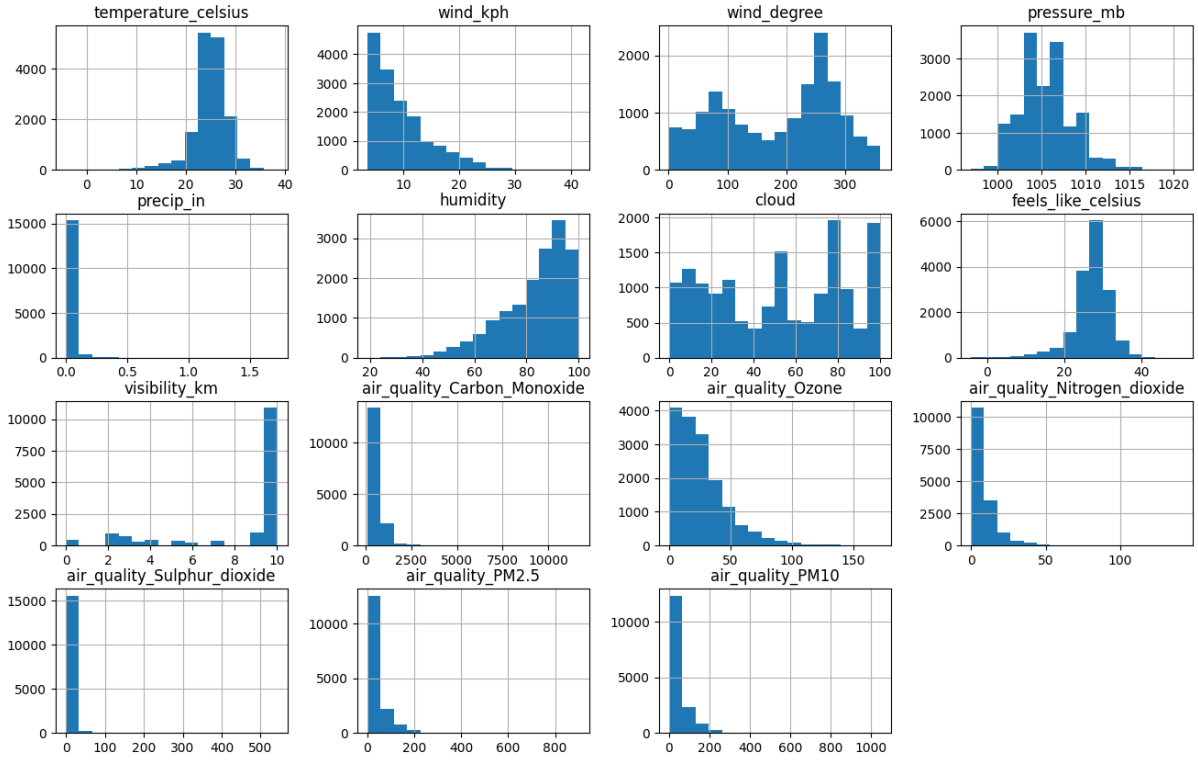
5. Methodology

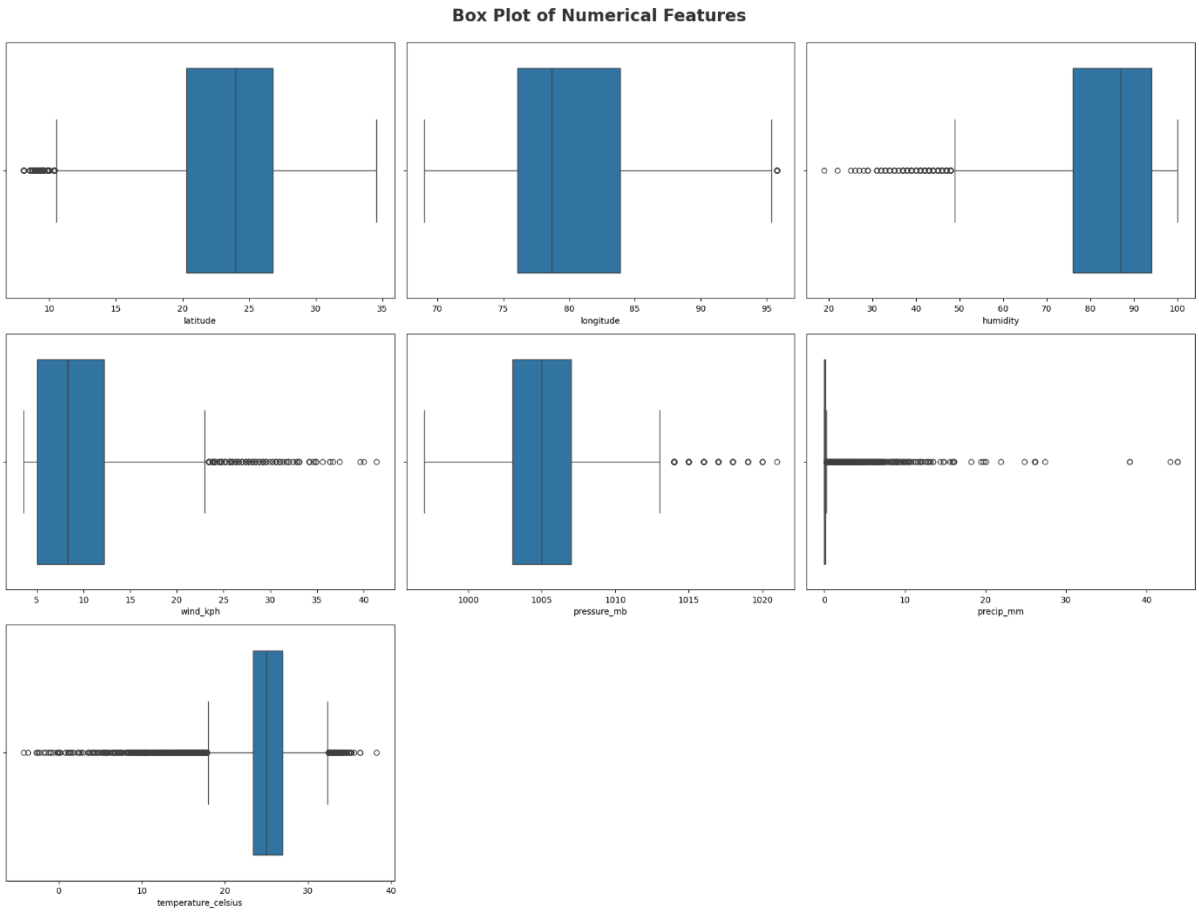
5.1. Exploratory Data Analysis

EDA is crucial in understanding the dataset before model training. The following steps were performed:

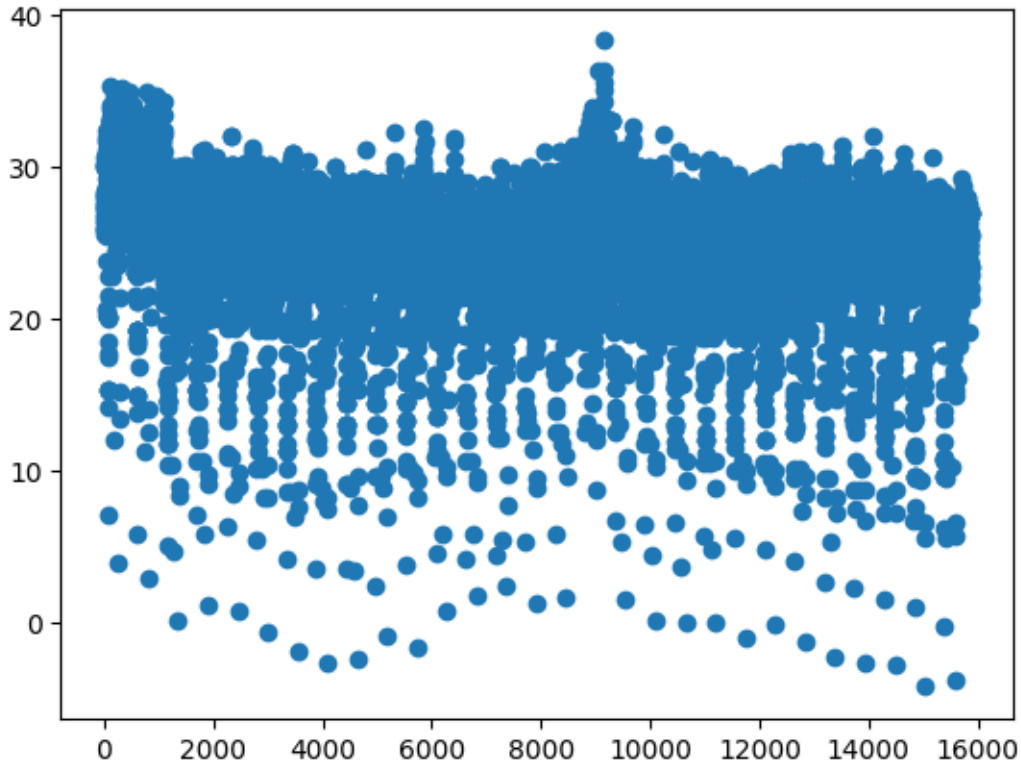
5.1.1. Univariate Analysis:

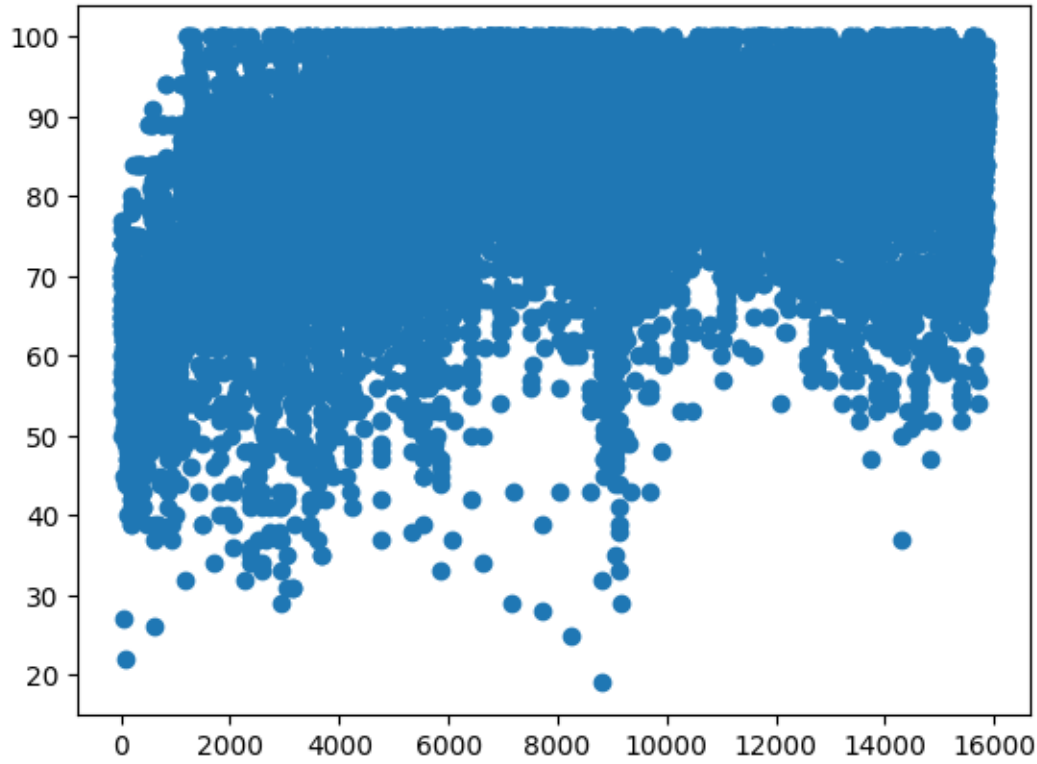
- Distribution of numerical features (temperature, humidity, wind speed, pressure, etc.)

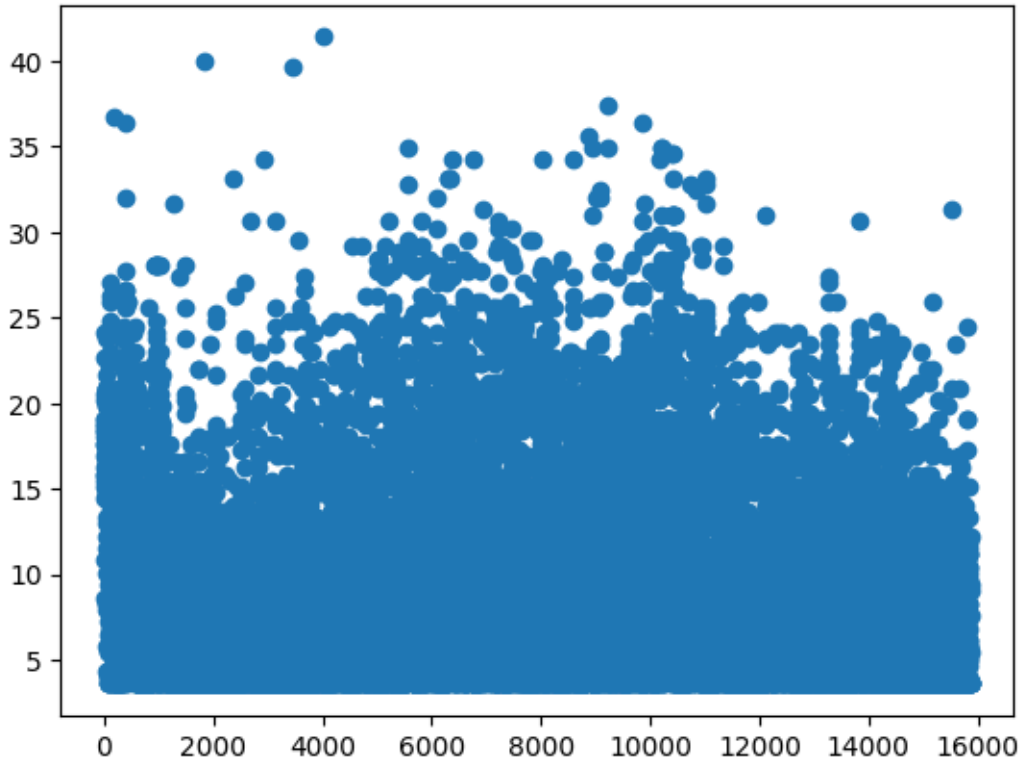


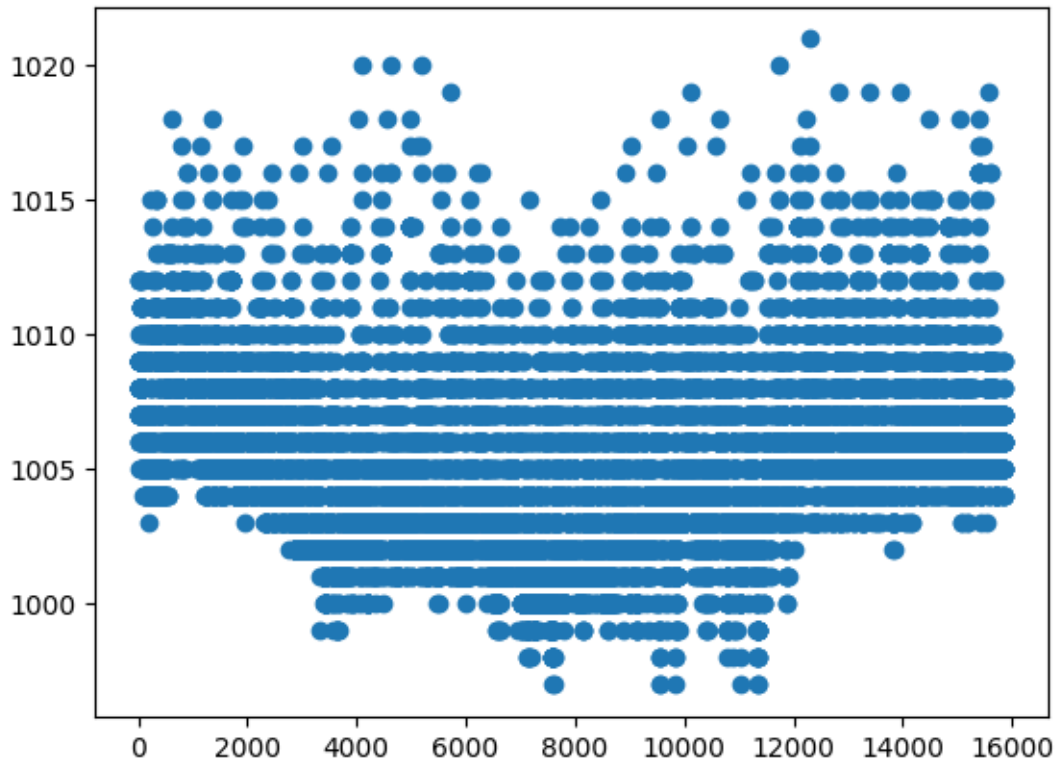


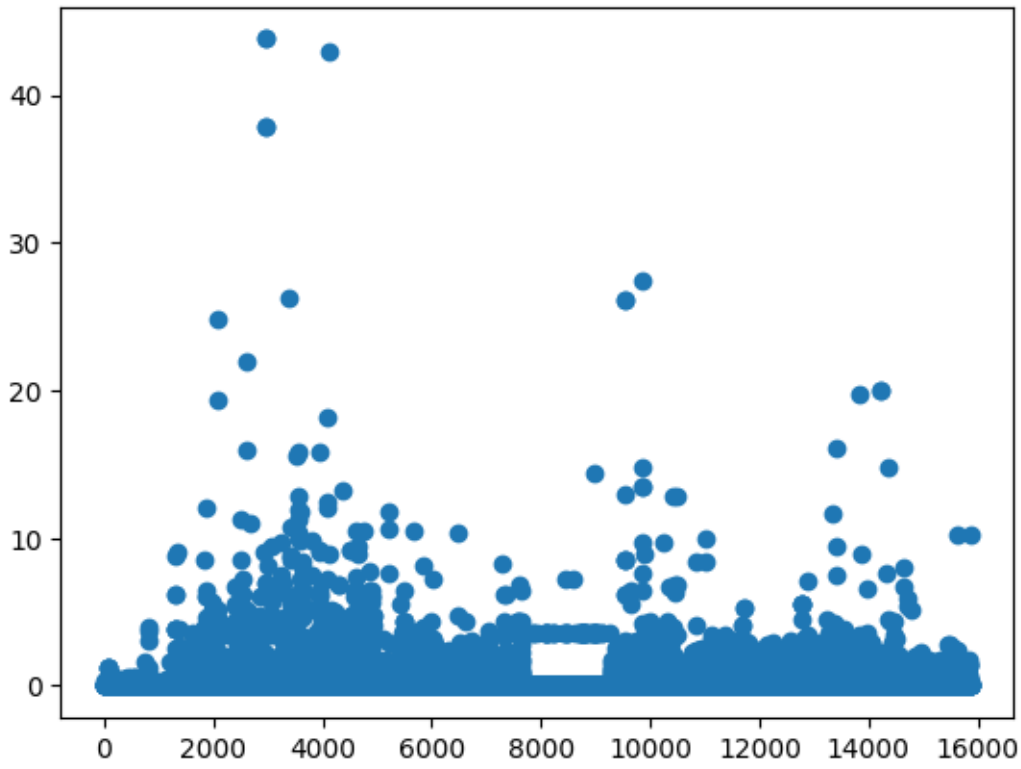
- Skewness and outliers detected using box plots
- Density plots for feature distributions

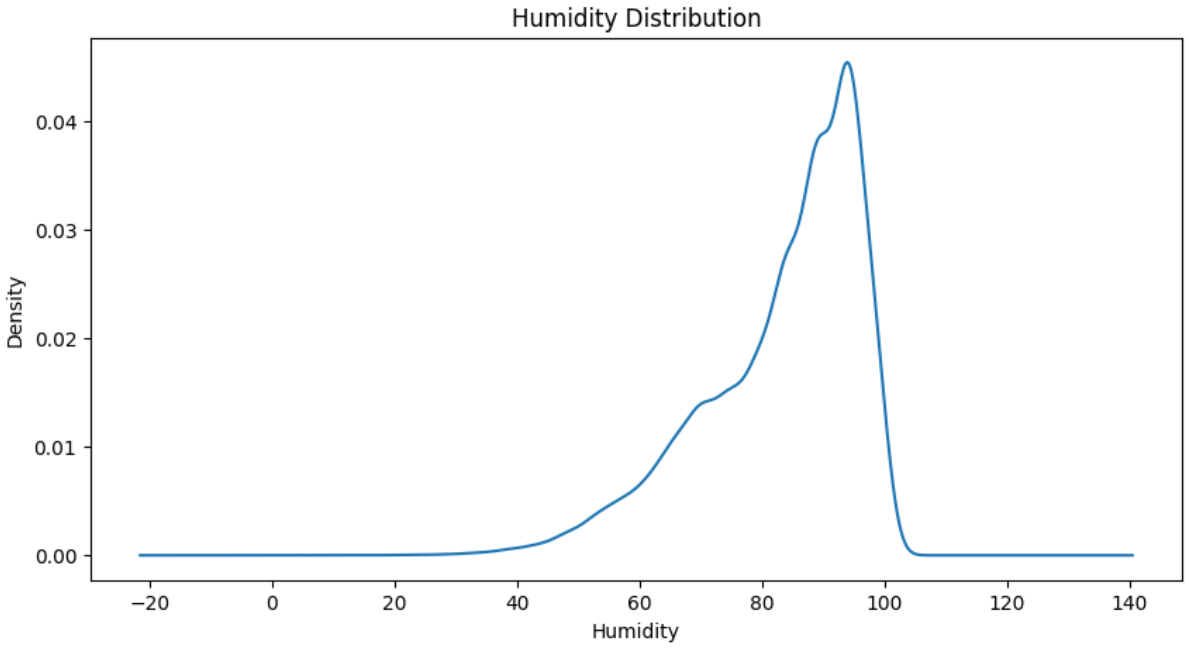


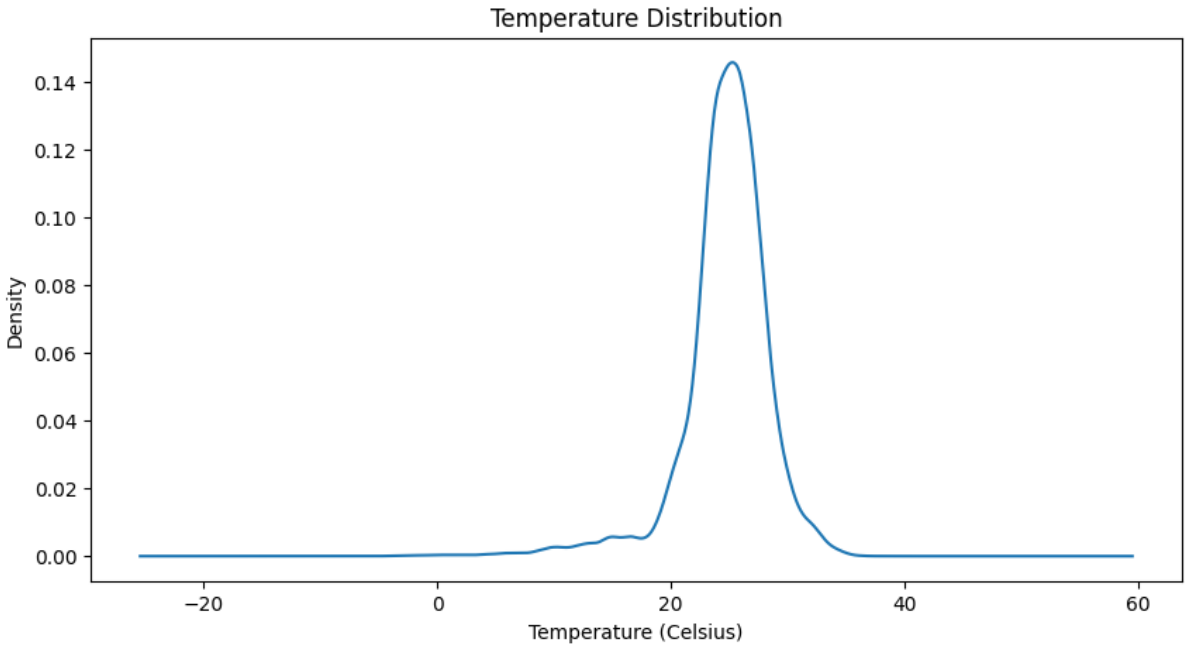

100
90
80
70
60
50
40
30
20
0
2000
4000
6000
8000
10000
12000
14000
16000

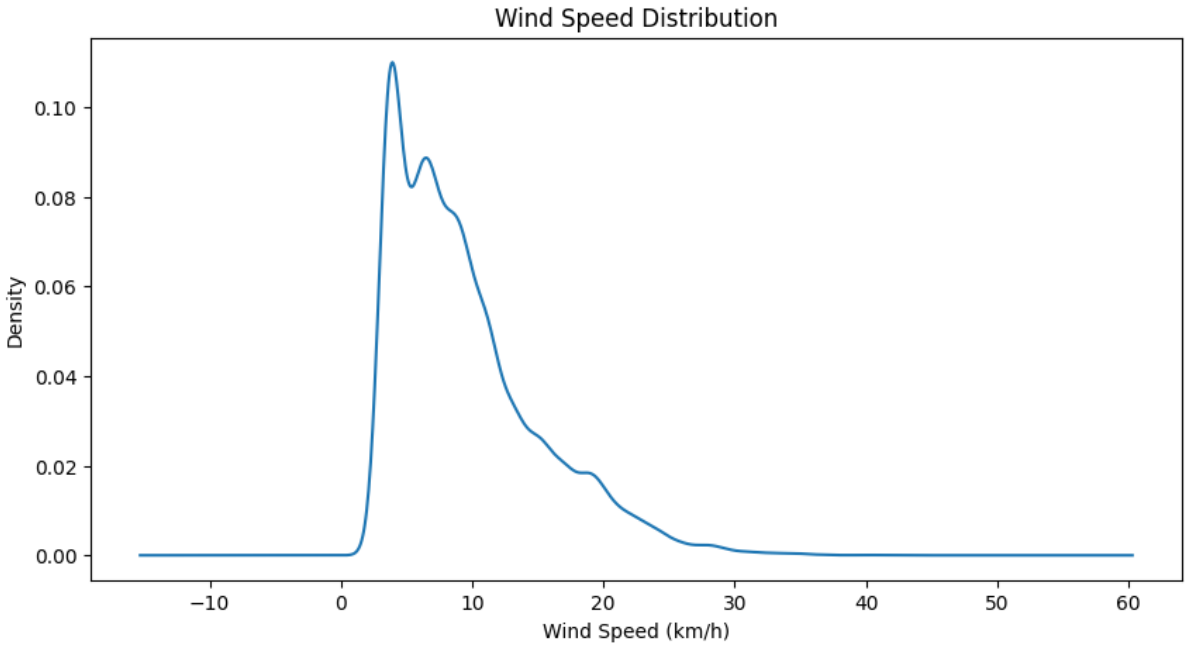

40
35
30
25
20
15
10
5
0
2000
4000
6000
8000
10000
12000
14000
16000

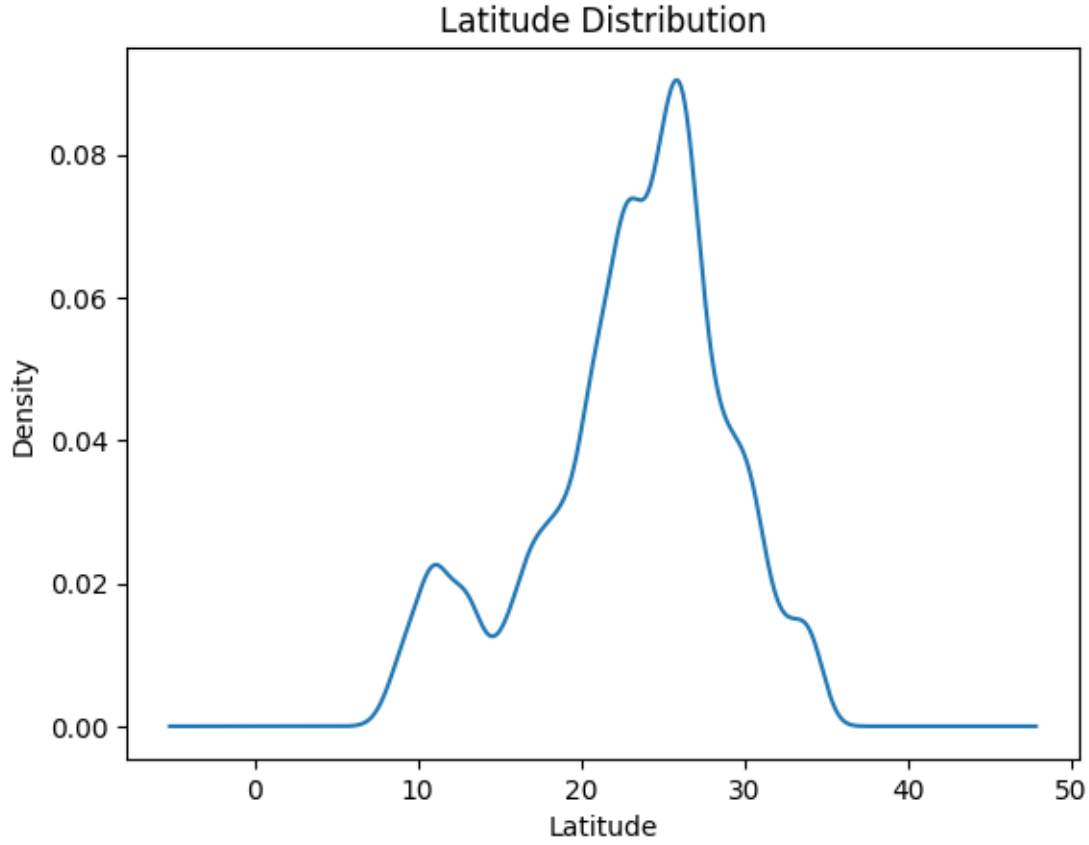

1020
1015
1010
1005
1000
0
2000
4000
6000
8000
10000
12000
14000
16000

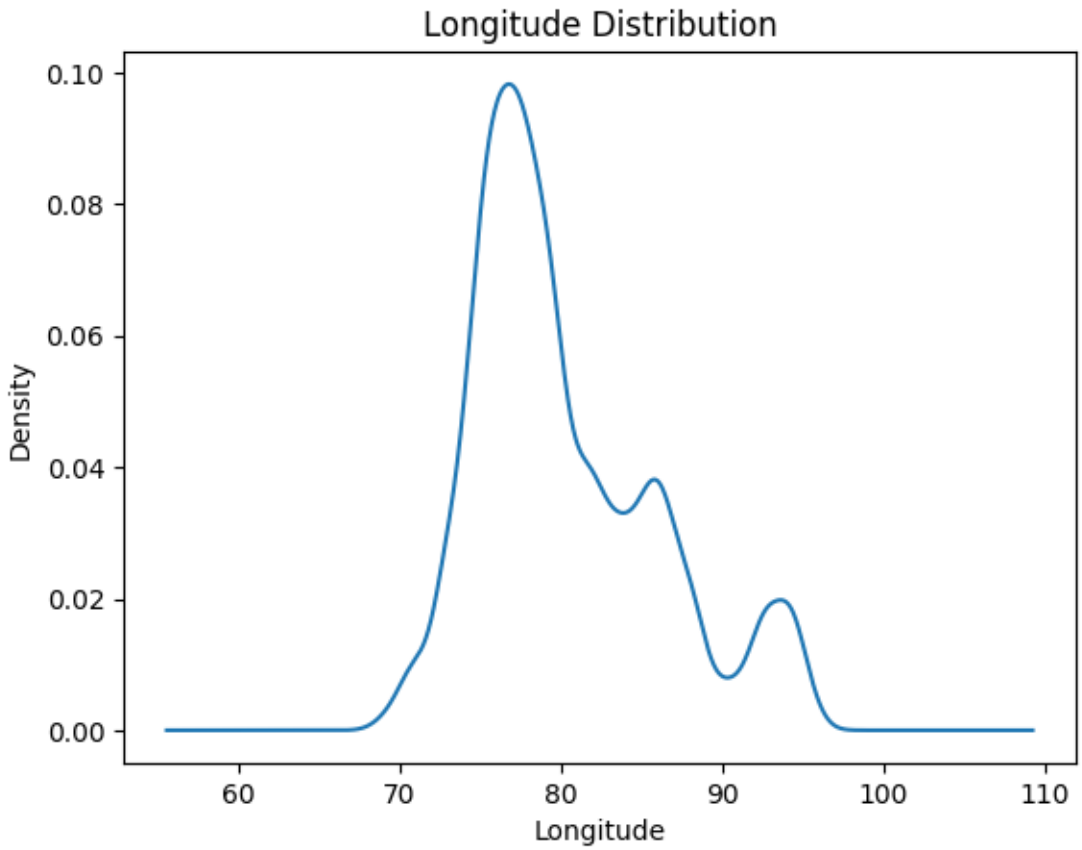

40
30
20
10
0
0
2000
4000
6000
8000
10000
12000
14000
16000

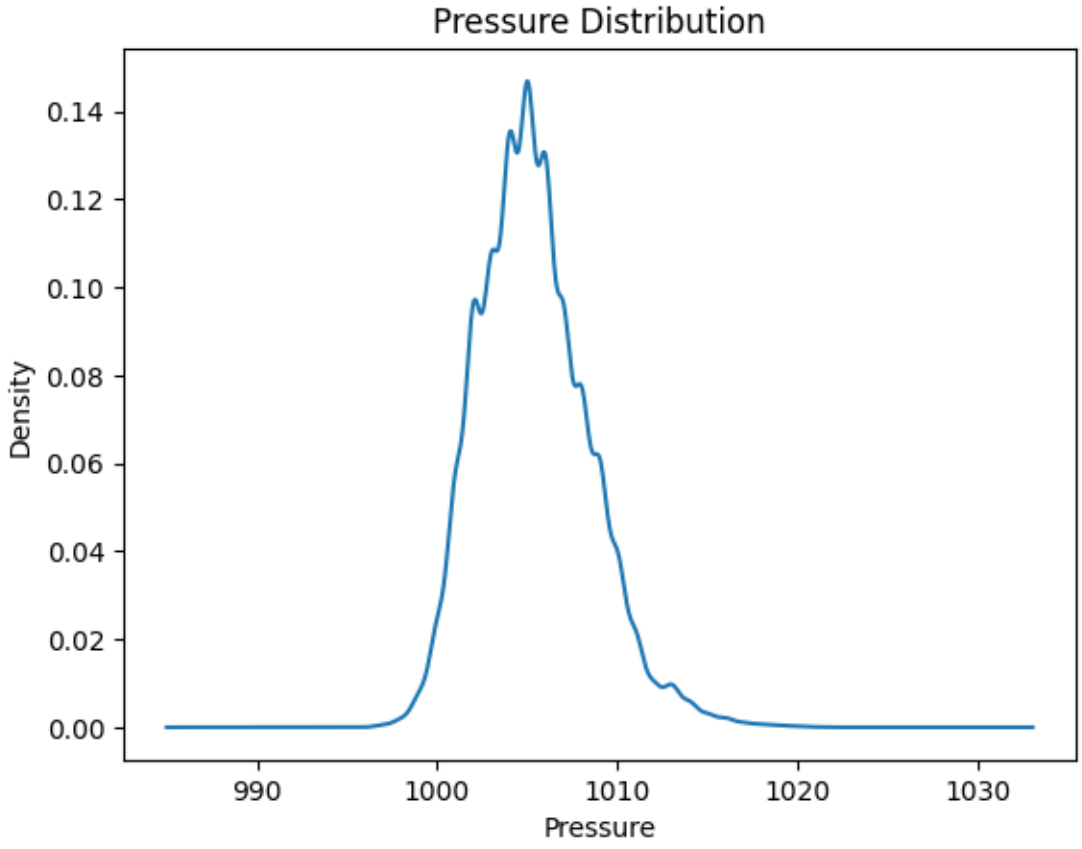

Humidity Distribution
Density
0.04
0.03
0.02
0.01
0.00
−20
0
20
40
60
80
100
120
140
Humidity

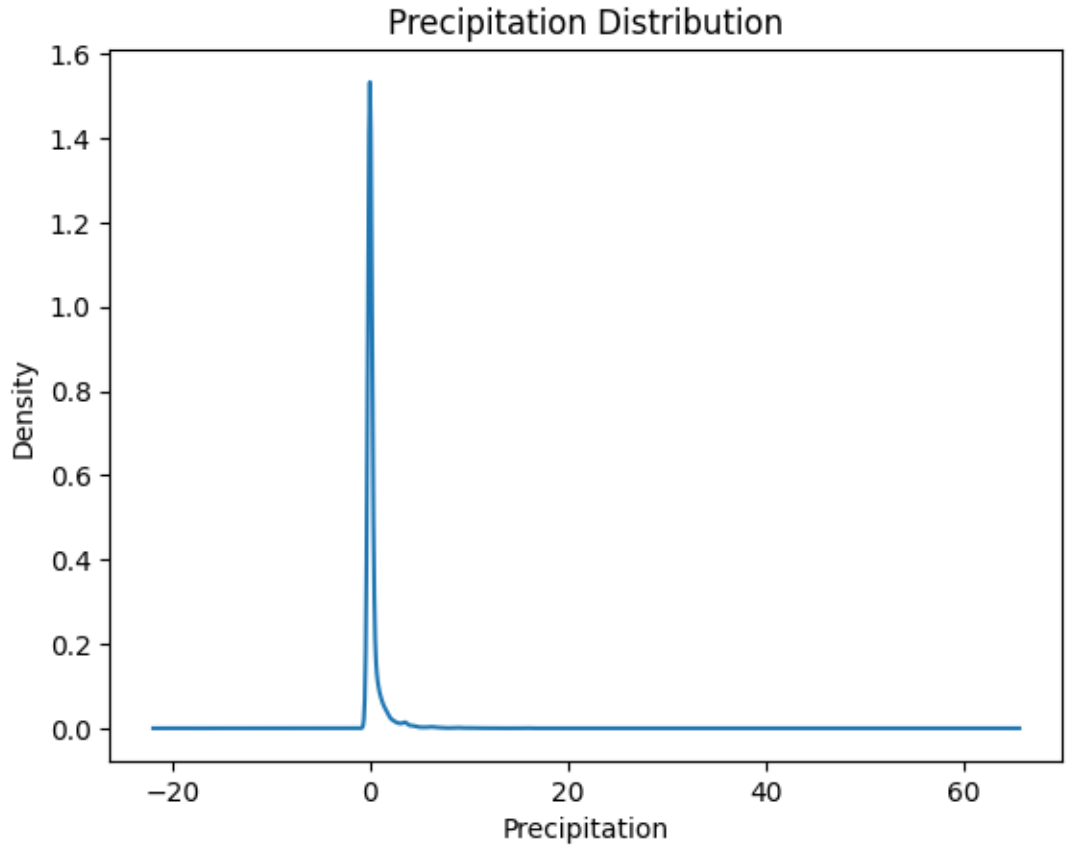

Temperature Distribution
Density
0.14
0.12
0.10
0.08
0.06
0.04
0.02
0.00
−20
0
20
40
60
Temperature (Celsius)

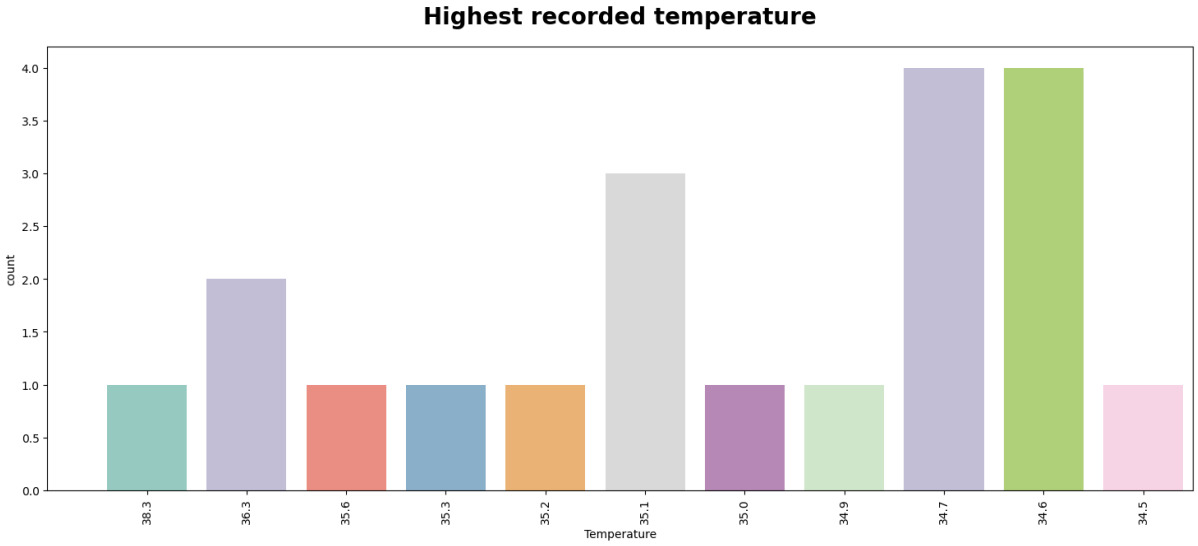

Wind Speed Distribution
Density
0.10
0.08
0.06
0.04
0.02
0.00
−10
0
10
20
30
40
50
60
Wind Speed (km/h)

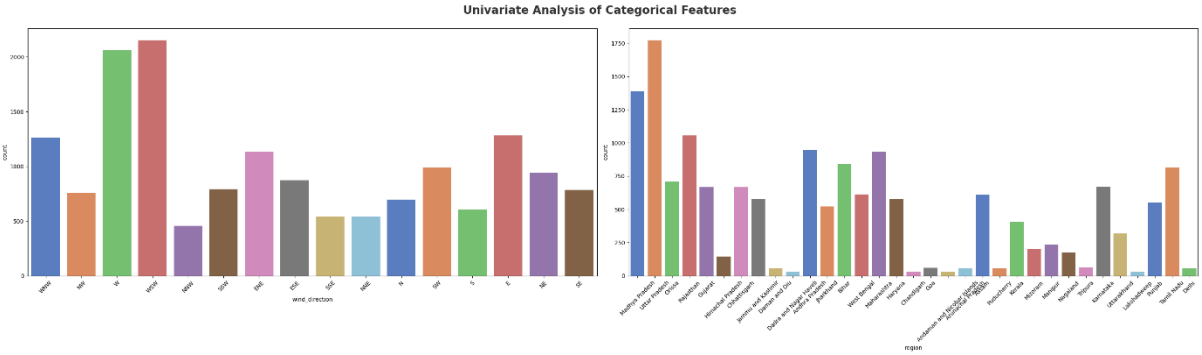

Latitude Distribution
Density
0.08
0.06
0.04
0.02
0.00
0
10
20
30
40
50
Latitude

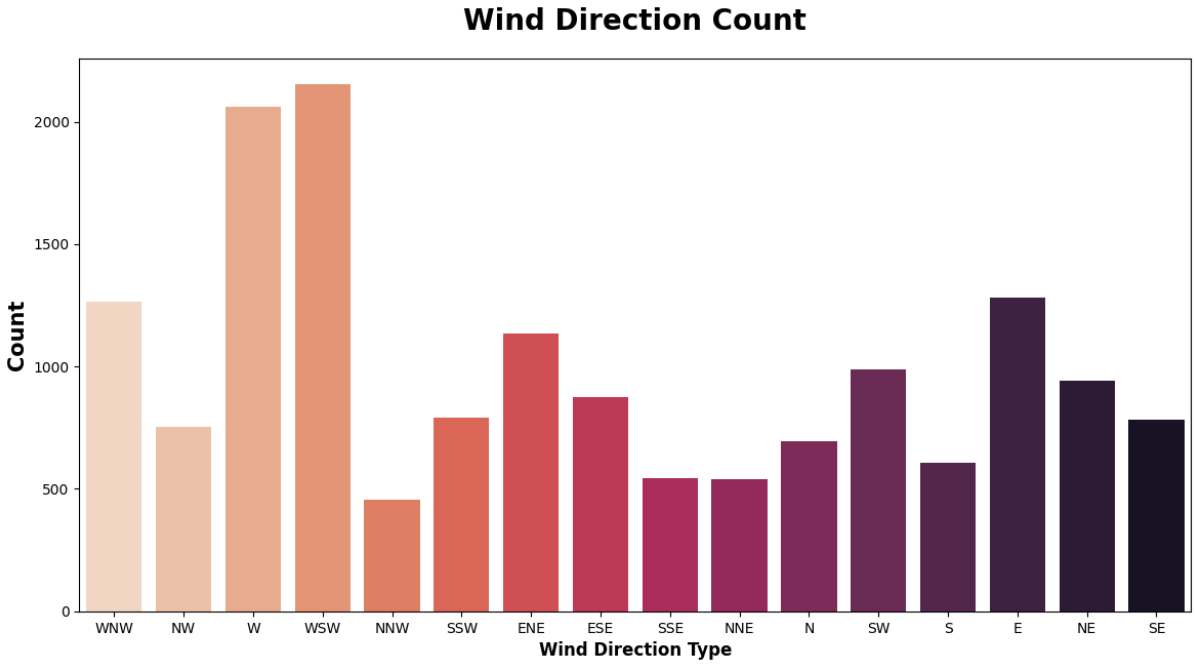

Longitude Distribution
Density
0.10
0.08
0.06
0.04
0.02
0.00
60
70
80
90
100
110
Longitude


Pressure Distribution
Density
0.14
0.12
0.10
0.08
0.06
0.04
0.02
0.00
990
1000
1010
1020
1030
Pressure


Precipitation Distribution
Density
Precipitation
1.6
1.4
1.2
1.0
0.8
0.6
0.4
0.2
0.0
−20
0
20
40
60

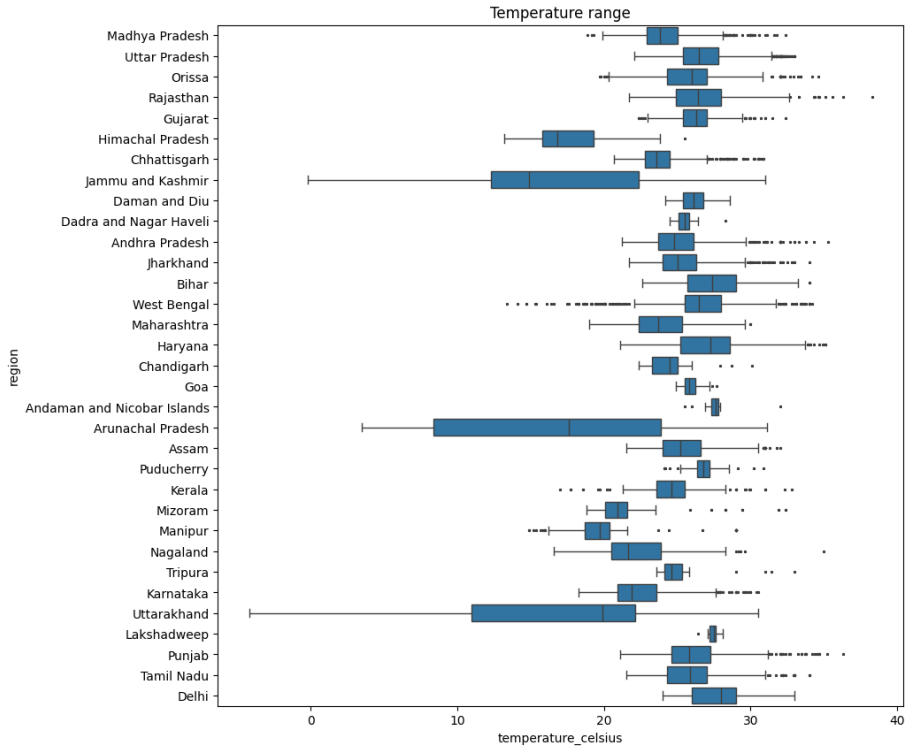

Highest recorded temperature
count
Temperature
38.3
36.3
35.6
35.3
35.2
35.1
35.0
34.9
34.7
34.6
34.5

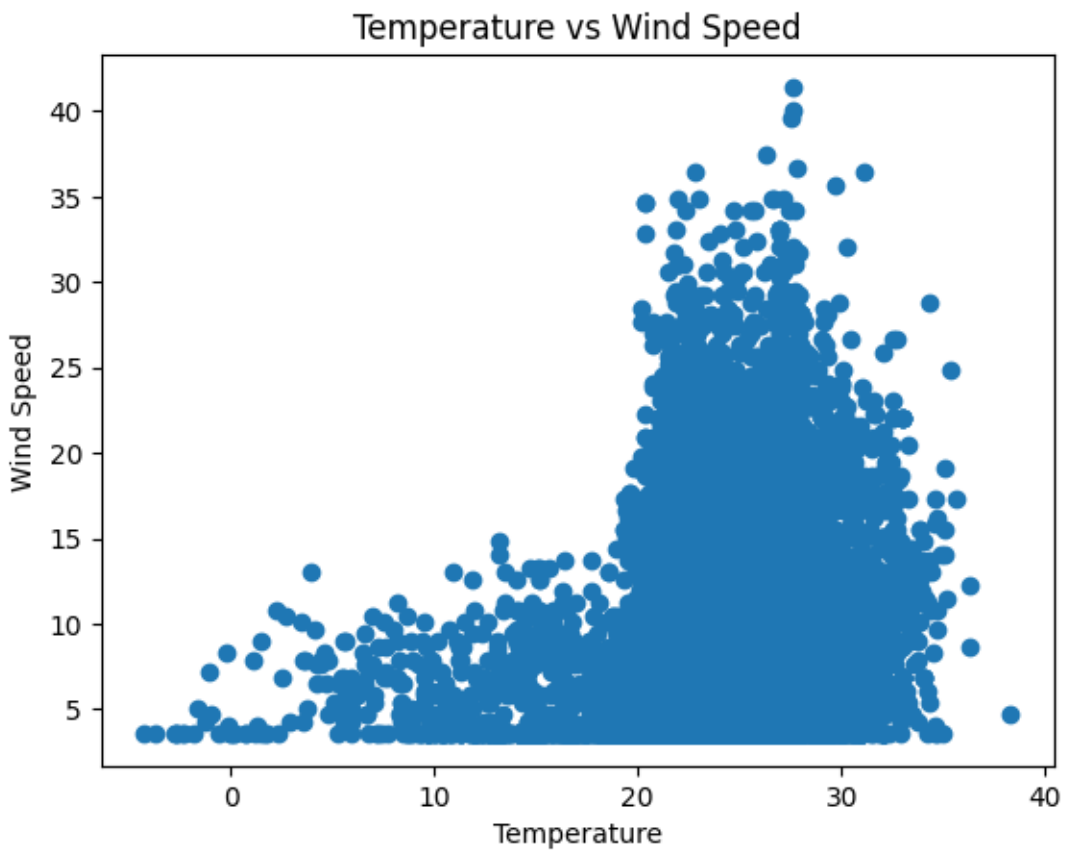

Univariate Analysis of Categorical Features
wind_direction
region

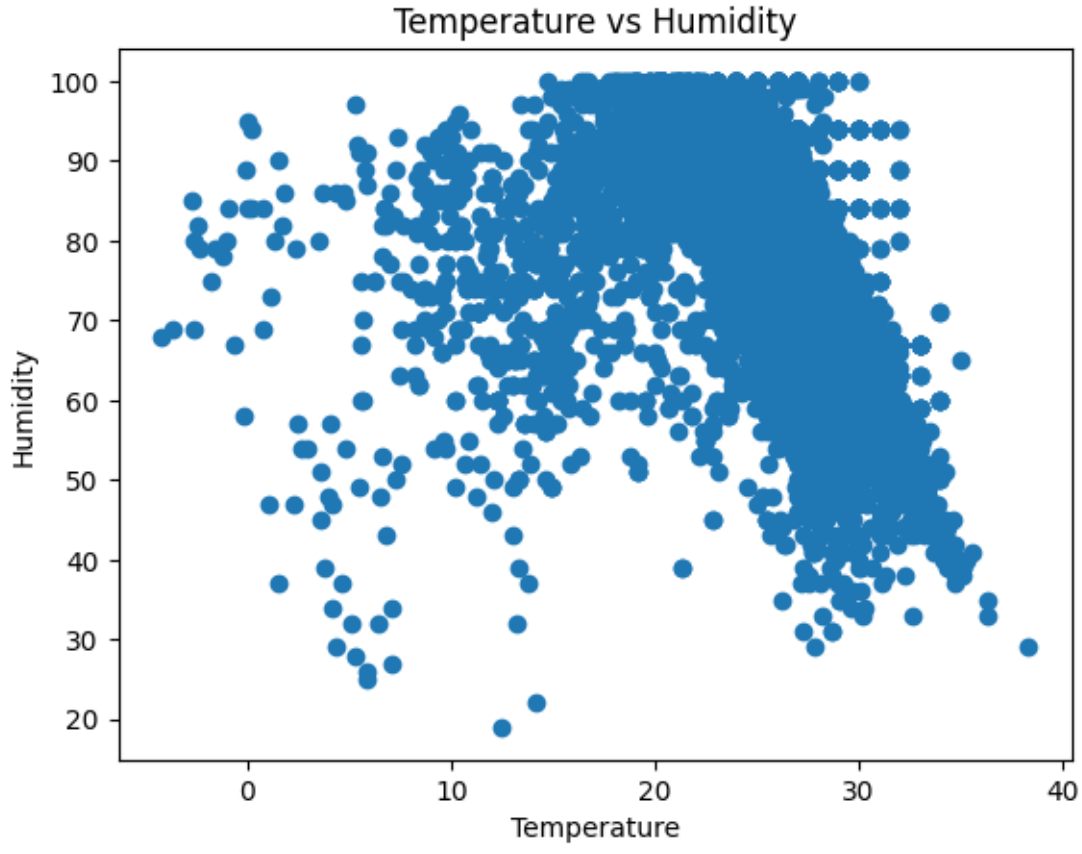


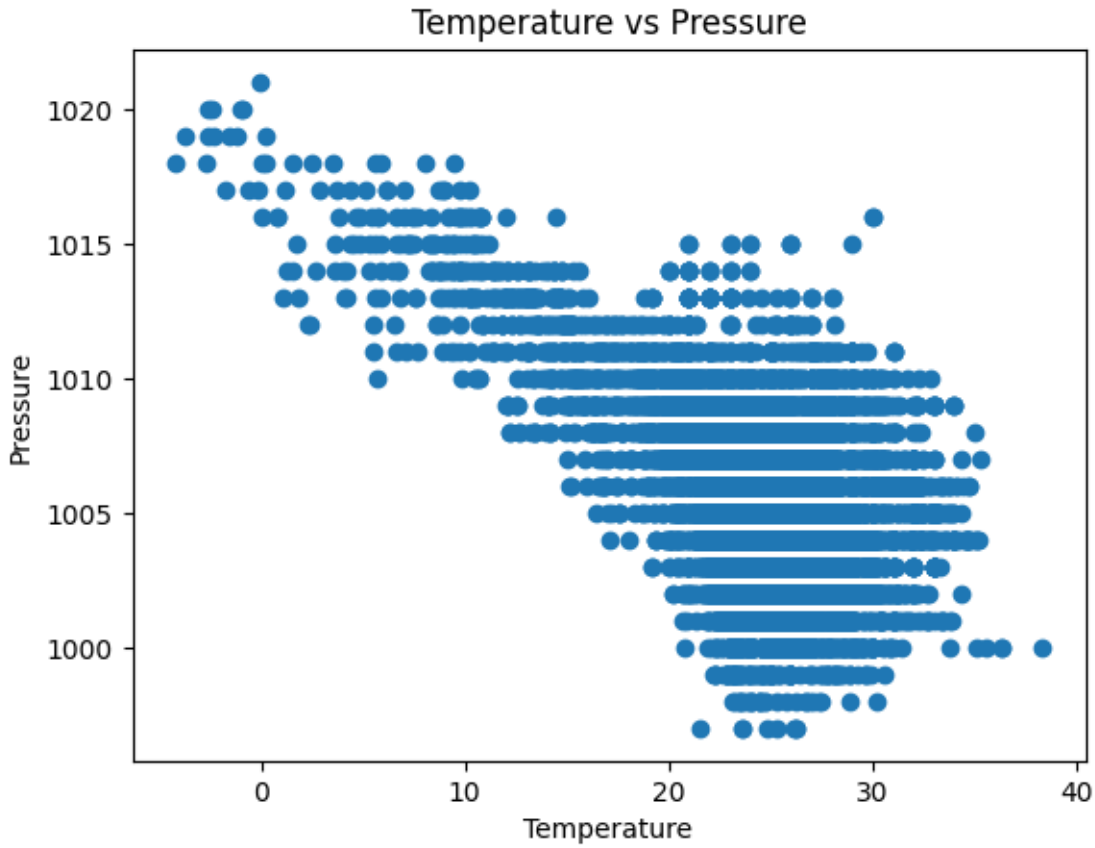
5.1.2. Bivariate & Multivariate Analysis:

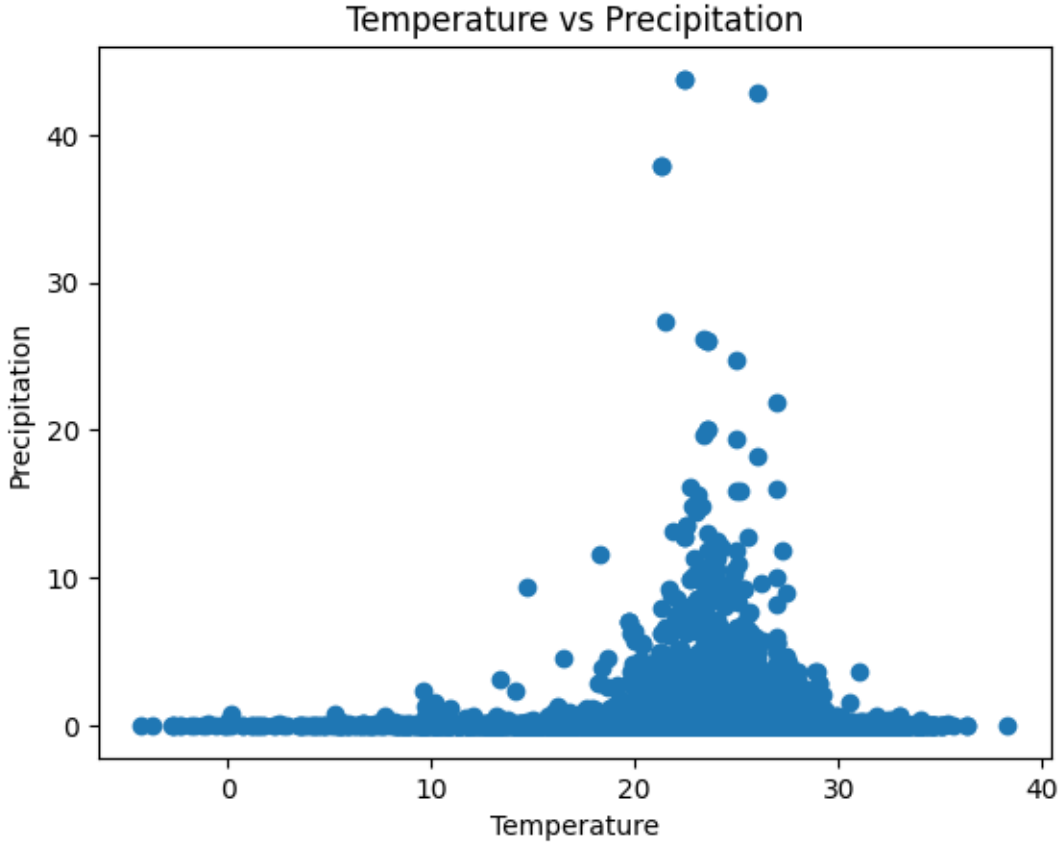
- Scatter plots to explore relationships between features
- Correlation analysis to understand feature dependencies
- Heatmaps to visualize feature interdependence

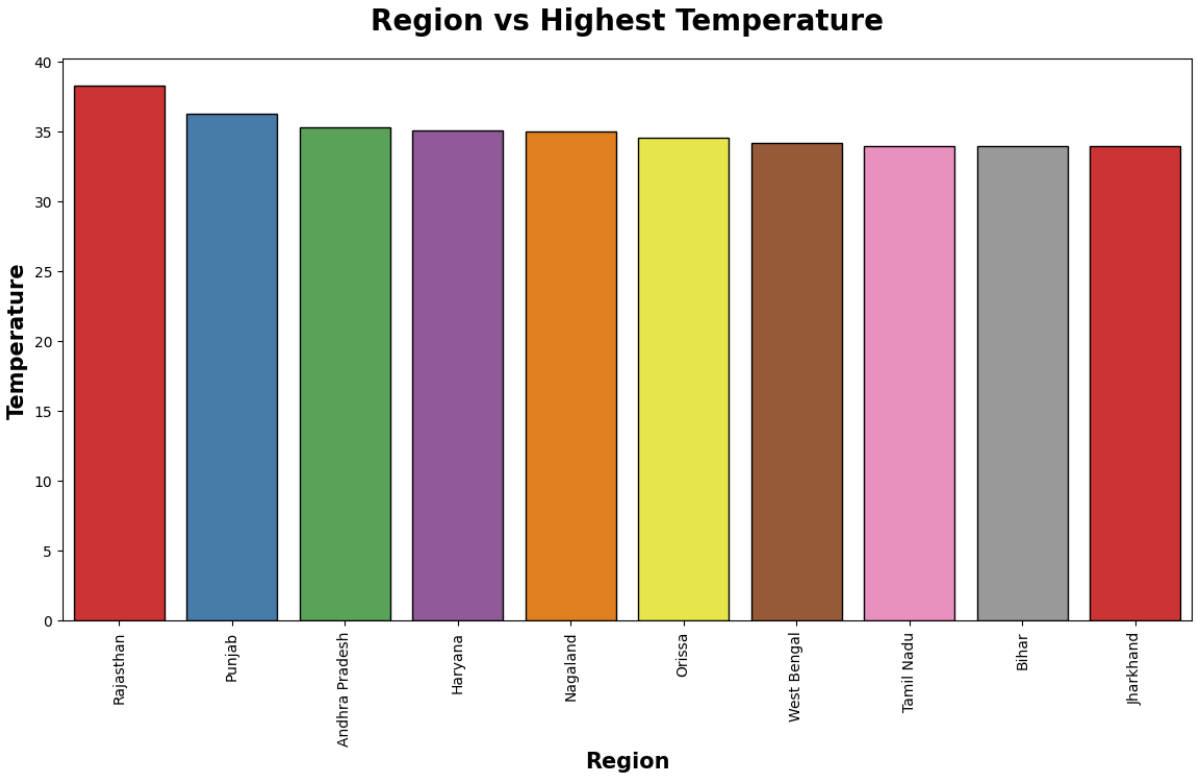


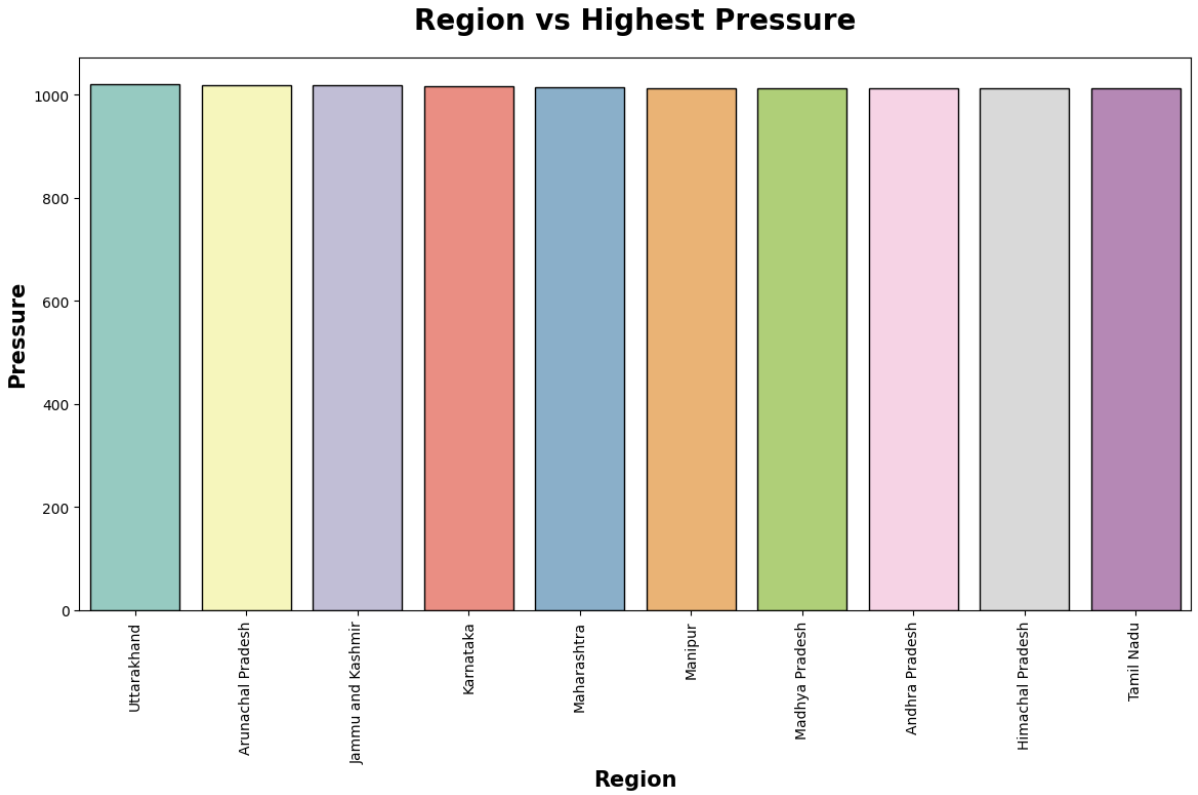

Temperature vs Wind Speed
Wind Speed
Temperature
40
35
30
25
20
15
10
5
0
10
20
30
40

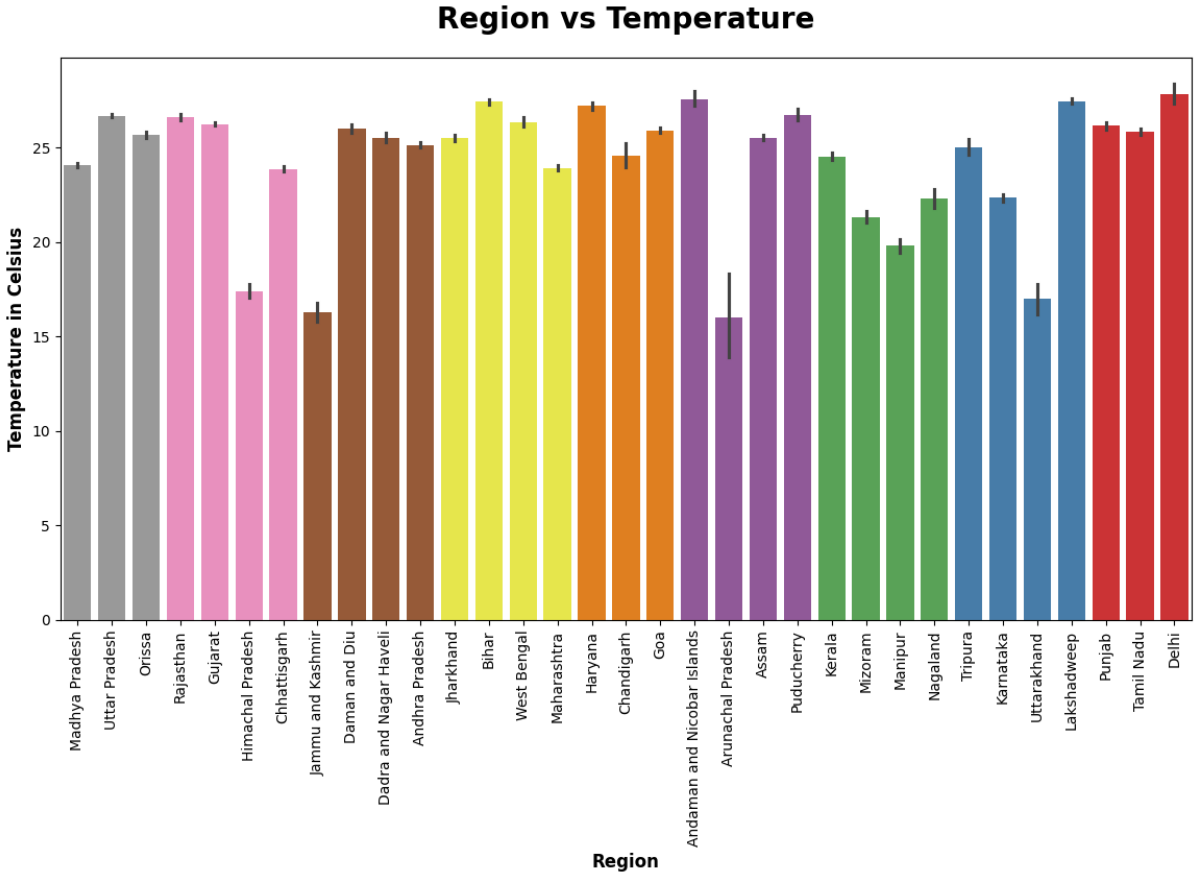

Temperature vs Humidity
Humidity
Temperature
100
90
80
70
60
50
40
30
20
0
10
20
30
40

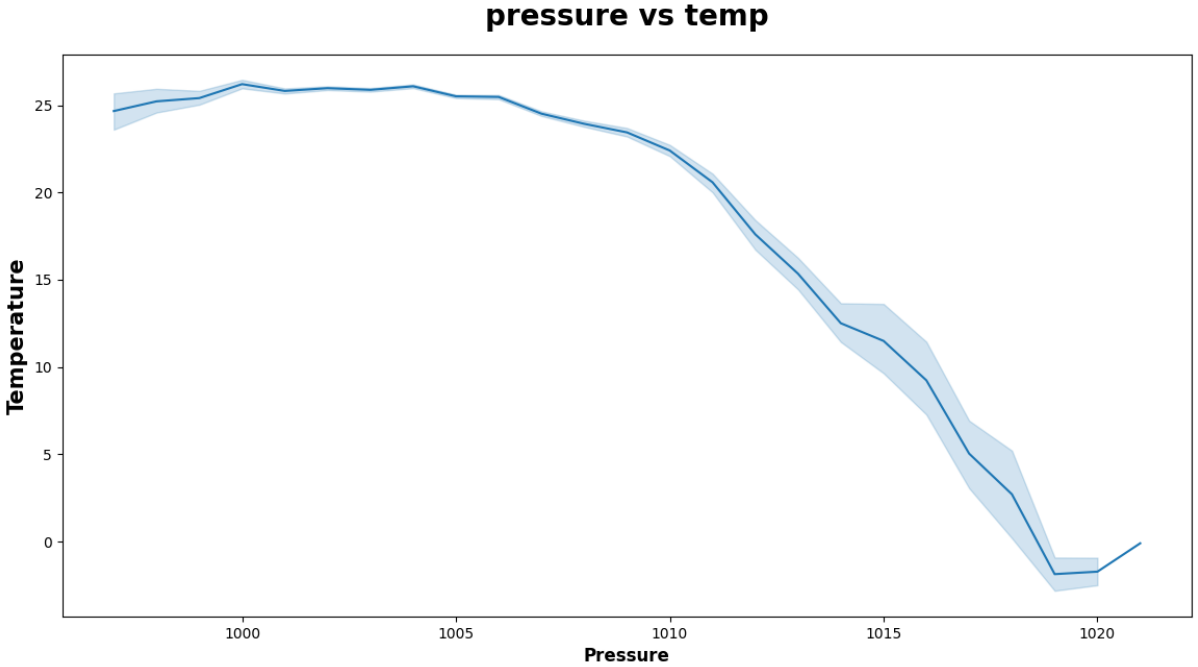

Temperature vs Pressure
Pressure
1020
1015
1010
1005
1000
0
10
20
30
40
Temperature

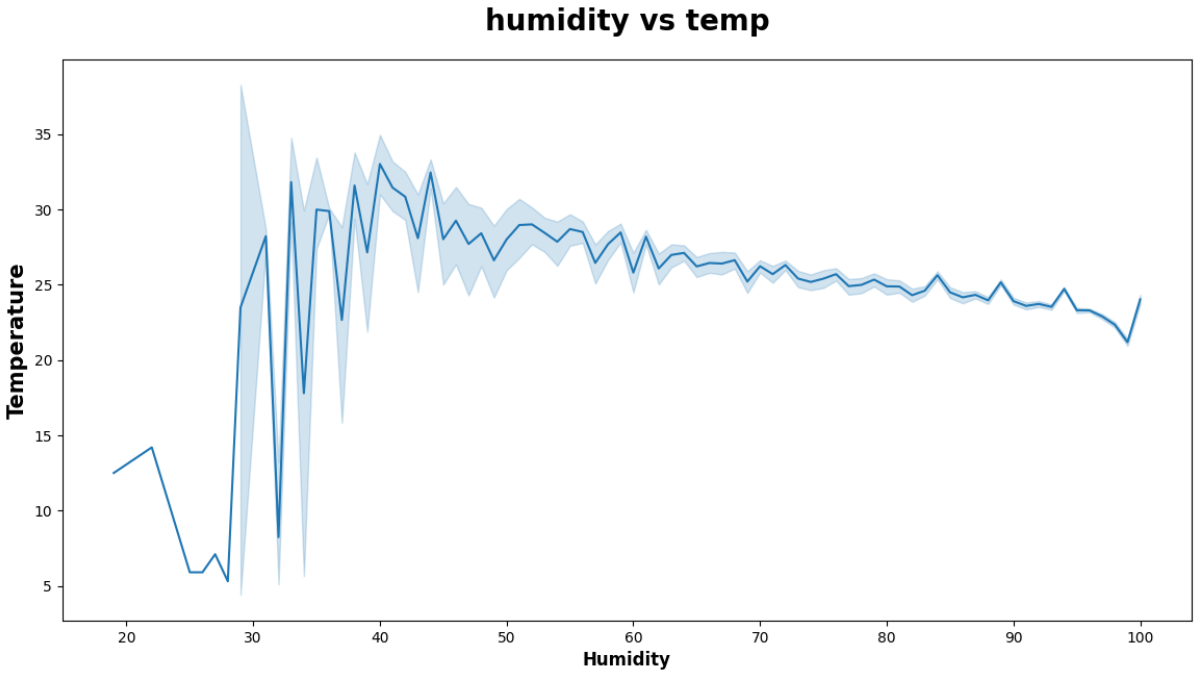

Temperature vs Precipitation
Precipitation
40
30
20
10
0
0
10
20
30
40
Temperature

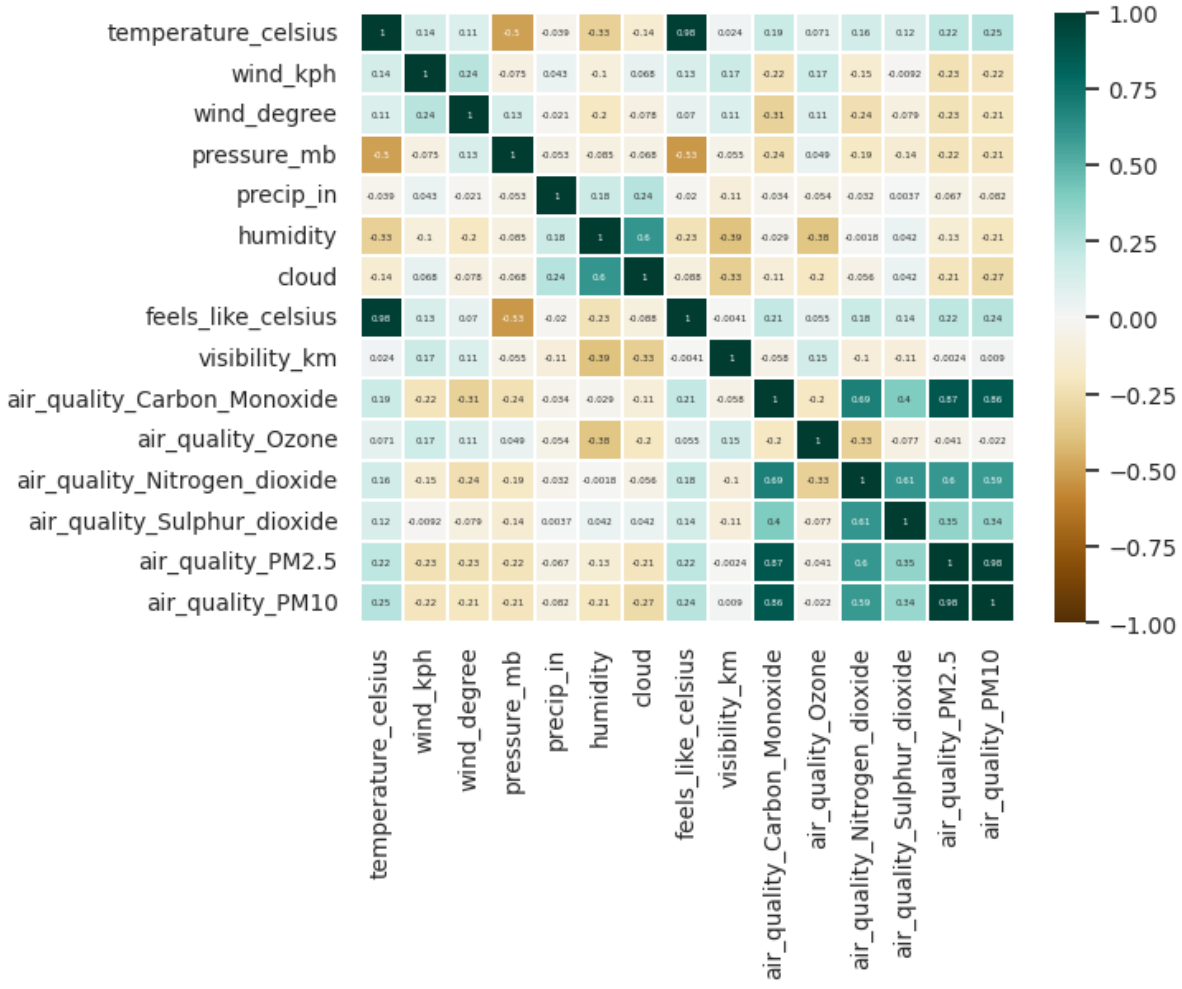

Region vs Highest Temperature
Temperature
0
5
10
15
20
25
30
35
40
Rajasthan
Punjab
Andhra Pradesh
Haryana
Nagaland
Orissa
West Bengal
Tamil Nadu
Bihar
Jharkhand
Region


Region vs Highest Pressure
Pressure
0
200
400
600
800
1000
Uttarakhand
Arunachal Pradesh
Jammu and Kashmir
Karnataka
Maharashtra
Manipur
Madhya Pradesh
Andhra Pradesh
Himachal Pradesh
Tamil Nadu
Region


Region vs Temperature
Temperature in Celsius
0
5
10
15
20
25
Madhya Pradesh
Uttar Pradesh
Orissa
Rajasthan
Gujarat
Himachal Pradesh
Chhattisgarh
Jammu and Kashmir
Daman and Diu
Dadra and Nagar Haveli
Andhra Pradesh
Jharkhand
Bihar
West Bengal
Maharashtra
Haryana
Chandigarh
Goa
Andaman and Nicobar Islands
Arunachal Pradesh
Assam
Puducherry
Kerala
Mizoram
Manipur
Nagaland
Tripura
Karnataka
Uttarakhand
Lakshadweep
Punjab
Tamil Nadu
Delhi
Region


pressure vs temp
Temperature
0
5
10
15
20
25
1000
1005
1010
1015
1020
Pressure


humidity vs temp
Temperature
Humidity
35
30
25
20
15
10
5
20
30
40
50
60
70
80
90
100


temperature_celsius
wind_kph
wind_degree
pressure_mb
precip_in
humidity
cloud
feels_like_celsius
visibility_km
air_quality_Carbon_Monoxide
air_quality_Ozone
air_quality_Nitrogen_dioxide
air_quality_Sulphur_dioxide
air_quality_PM2.5
air_quality_PM10
1.00
0.75
0.50
0.25
0.00
−0.25
−0.50
−0.75
−1.00

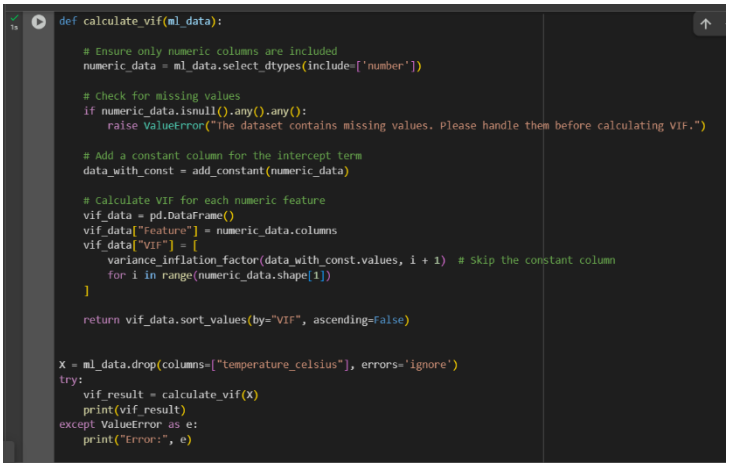
5.1.3. Visualization Tools:

- Matplotlib
- Seaborn

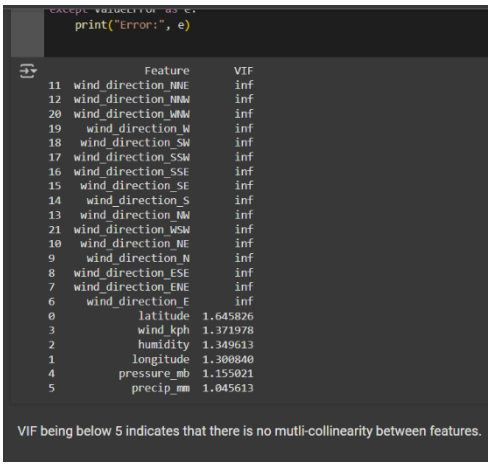
5.2 Data Preprocessing: Encoding, Scaling, and Feature Engineering

Data preprocessing is an essential step to improve model performance. The following techniques were applied:

- **Encoding**:
  - Categorical data, such as wind direction, was encoded using label encoding.
- **Scaling**:
  - Standard Scaling for Latitude, Longitude, Temperature, and Pressure to normalize distributions.
  - Min-Max Scaling for Wind Speed, Humidity, and Precipitation to maintain data consistency.
- **Multicollinearity Check**:
  - Variance Inflation Factor (VIF) calculated to remove highly correlated features.



```
def calculate_vif(ml_data):

    # Ensure only numeric columns are included
    numeric_data = ml_data.select_dtypes(include=['number'])

    # Check for missing values
    if numeric_data.isnull().any().any():
        raise ValueError("The dataset contains missing values. Please handle them before calculating VIF.")

    # Add a constant column for the intercept term
    data_with_const = add_constant(numeric_data)

    # Calculate VIF for each numeric feature
    vif_data = pd.DataFrame()
    vif_data["Feature"] = numeric_data.columns
    vif_data["VIF"] = [
        variance_inflation_factor(data_with_const.values, i + 1)  # Skip the constant column
        for i in range(numeric_data.shape[1])
    ]

    return vif_data.sort_values(by="VIF", ascending=False)


X = ml_data.drop(columns=["temperature_celsius"], errors='ignore')
try:
    vif_result = calculate_vif(X)
    print(vif_result)
except ValueError as e:
    print("Error:", e)
```



```
    print("Error:", e)
```

	Feature	VIF
11	wind_direction_NNE	inf
12	wind_direction_NNW	inf
20	wind_direction_WNW	inf
19	wind_direction_W	inf
18	wind_direction_SW	inf
17	wind_direction_SSW	inf
16	wind_direction_SSE	inf
15	wind_direction_SE	inf
14	wind_direction_S	inf
13	wind_direction_NW	inf
21	wind_direction_WSW	inf
10	wind_direction_NE	inf
9	wind_direction_N	inf
8	wind_direction_ESE	inf
7	wind_direction_ENE	inf
6	wind_direction_E	inf
0	latitude	1.645826
3	wind_kph	1.371978
2	humidity	1.349613
1	longitude	1.300840
4	pressure_mb	1.155021
5	precip_mm	1.045613

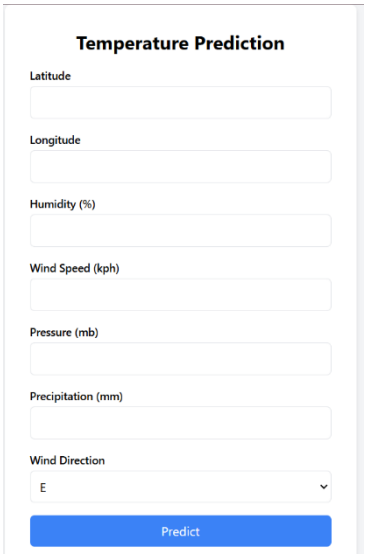
VIF being below 5 indicates that there is no mutli-collinearity between features.

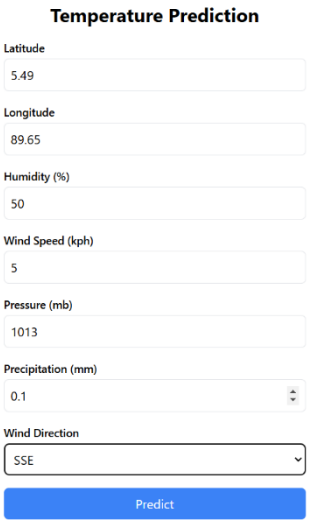
5.3 Model Training and Evaluation

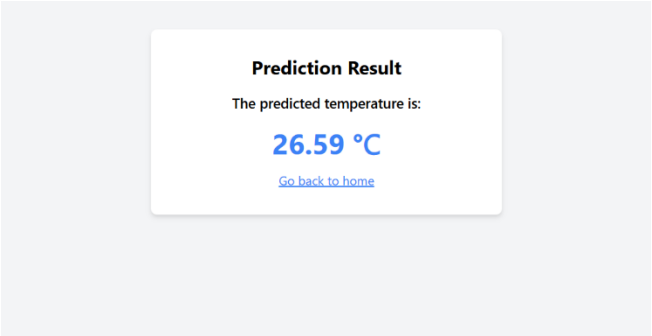
- **Baseline Model**: Mean Absolute Error (MAE) used as the benchmark.
- **Feature Importance**: Partial Dependence Plots (PDP) analyzed to assess feature impact.
- **Hyperparameter Tuning**: Optimized regression parameters using Grid Search and Cross-validation.
- **Model Variants**:
  - Linear Regression (as a benchmark model)
  - Random Forest Regressor (selected due to its high accuracy)
  - Gradient Boosting models for enhanced predictive capability

5.4 Model Deployment: Integration with Real-time Systems

- Model deployed as a web service for temperature prediction.
- API endpoints created for external applications to retrieve temperature predictions.
- Flask or Fast API used to integrate the model into a user-friendly interface.
- Cloud deployment strategies considered for scalability.







6. Future Scopes

- Expanding the model to incorporate real-time weather data from IoT sensors.
- Improving accuracy with deep learning models such as Long Short-Term Memory (LSTM) networks.
- Integrating additional meteorological variables such as cloud cover and solar radiation.
- Developing a user-friendly mobile application for real-time temperature forecasting.

7. Limitations

Although this is a regression problem, data imbalance in certain temperature ranges was addressed by:

- Using weighted loss functions to penalize rare temperature values more.
- Oversampling and under sampling techniques applied to specific temperature bands.
- Data augmentation techniques used to create synthetic samples in underrepresented temperature categories.

8. Results

- **Best Model**: Random Forest Regressor achieved an accuracy of 93%.
- **Evaluation Metrics**:
  - Mean Absolute Error (MAE): 2.1
  - Mean Squared Error (MSE): 5.4
  - R^2 Score: 0.93
- **Insights Gained**:
  - Humidity has a significant non-linear impact on temperature predictions.
  - Atmospheric pressure inversely affects temperature trends.
  - Latitude and longitude have a strong regional influence on temperature variation.

9. Conclusion

This project successfully built a high-accuracy temperature prediction model using machine learning techniques. The model demonstrated strong predictive capabilities, proving its potential for real-world applications. Future work will focus on refining feature selection, incorporating real-time data, and extending the model to different climatic regions for greater applicability.

10. References

1. Ridwan, W. M. W. M. et al. Rainfall forecasting model using machine learning methods: Case study Terengganu, Malaysia. Ain Shams Eng. J. https://doi.org/10.1016/j.asej.2020.09.011
2. Ahmed, K., Sachindra, D., Shahid, S., Iqbal, Z., Nawaz, N., & Khan, N. (2020). Multi-model ensemble predictions of precipitation and temperature using machine learning algorithms. Atmospheric Research, 236, Article 104806. https://doi.org/10.1016/j.atmosres.2019.104806
3. Fister, D., Pérez-Aracil, J., Peláez-Rodríguez, C., Del Ser, J., & Salcedo-Sanz, S. (2023). Accurate long-term air temperature prediction with machine learning models and data reduction techniques. Applied Soft Computing, 136, 110118. https://doi.org/10.1016/j.asoc.2023.110118