## Artificial Intelligence

## and Machine Learning

Project Report

Semester-IV (Batch-2022)

EARTH GUARD

A red and white sign

Description automatically generated with low confidence

**Supervised By: Submitted By:**

Shagun Sharma Hargun Singh, 2210990360

Harinder Singh, 2210990361

Harshit Behal, 2210990386

Himanshu, 2210990408

**Department of Computer Science and Engineering**

## Chitkara University Institute of Engineering & Technology,

## Chitkara University, Punjab

1. **Introduction:**

Introduction to Temperature Analysis and Forecasting:

Temperature analysis and forecasting play a pivotal role in understanding climate patterns and their impacts on various aspects of human life, agriculture, and the environment. With the advent of sophisticated data analytics techniques, researchers and policymakers can delve deeper into historical temperature data to discern trends, patterns, and potential future scenarios.

In this project, we embark on a journey to explore temperature data specific to India, aiming to uncover insights into its temporal variations, seasonal trends, and long-term changes. Leveraging advanced statistical methods and time series analysis, we seek to not only comprehend the past temperature dynamics but also forecast future temperature fluctuations with a high degree of accuracy.

Temperature data, holds profound implications for public welfare and decision-making. Just as depression is a significant public health concern impacting individuals' well-being, productivity, and quality of life, fluctuations in temperature can have far-reaching consequences on agriculture, water resources, energy consumption, and health outcomes.

Similar to the challenges in diagnosing depression, assessing temperature patterns often involves navigating through complex data sets and addressing issues such as missing values, seasonality, and autocorrelation. By employing robust methodologies like ARIMA (AutoRegressive Integrated Moving Average) and RNN(Recurrent Neural Networks) modeling, we aim to develop predictive models that can aid in early detection of temperature anomalies and facilitate proactive interventions to mitigate potential adverse effects.

As we embark on this journey, we recognize the importance of harnessing data-driven insights to inform evidence-based policies and interventions. By understanding the nuances of temperature dynamics and leveraging advanced analytics, we strive to contribute to a more resilient and sustainable future, where the impacts of climate variability are understood and managed effectively.



**1.1 Background:**

Background on Temperature Analysis and Forecasting:

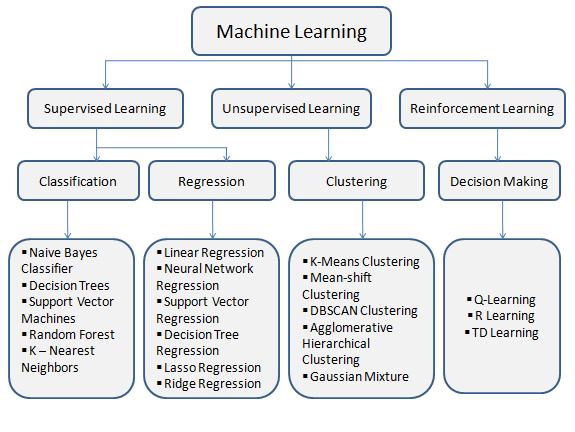
Temperature analysis and forecasting are critical components of understanding climate behavior and its implications across various domains. As technology continues to advance, the integration of computational methods, particularly machine learning, has revolutionized the way we interpret and predict temperature patterns.

Machine Learning (ML) stands out as a powerful technique within the realm of computing, offering automated solutions for building analytical models and extracting insights from vast datasets. Rooted in the concept of pattern recognition, ML algorithms enable computers to learn from data, recognize patterns, and make informed decisions without explicit programming.

The evolution of ML has led to significant advancements in temperature analysis and forecasting. By harnessing the power of algorithms, researchers can uncover intricate temperature patterns, identify trends, and develop predictive models with enhanced accuracy. One such application is the early detection of temperature anomalies, allowing for proactive measures to mitigate potential risks and impacts.

Within the domain of machine learning, there exist several approaches, each tailored to address specific challenges in temperature analysis. Supervised Learning (SL) and Unsupervised Learning (UL) are traditional paradigms, while emerging techniques like Semi-Supervised Learning (SSL) and Reinforcement Learning offer promising avenues for further exploration.

As we delve deeper into the intersection of machine learning and temperature analysis, we aim to leverage these innovative methodologies to enhance our understanding of climate dynamics, empower decision-making, and foster sustainable solutions to address the challenges posed by climate variability. Through collaborative efforts and technological advancements, we strive to pave the way towards a more resilient and climate-resilient future.



**Traditional Methods of Temperature Analysis:**

**Traditionally, temperature analysis has relied on statistical techniques and manual interpretation of historical data by climate scientists and meteorologists. These methods typically involve examining temperature records, identifying trends, and making predictions based on past observations. While informative, these approaches have limitations, including potential biases, subjective interpretations, and difficulties in handling large datasets efficiently.**

**Motivation for Machine Learning Approach**

**1. Objectivity: ML models can objectively analyze temperature data, minimizing human biases and subjectivity in interpretation.**

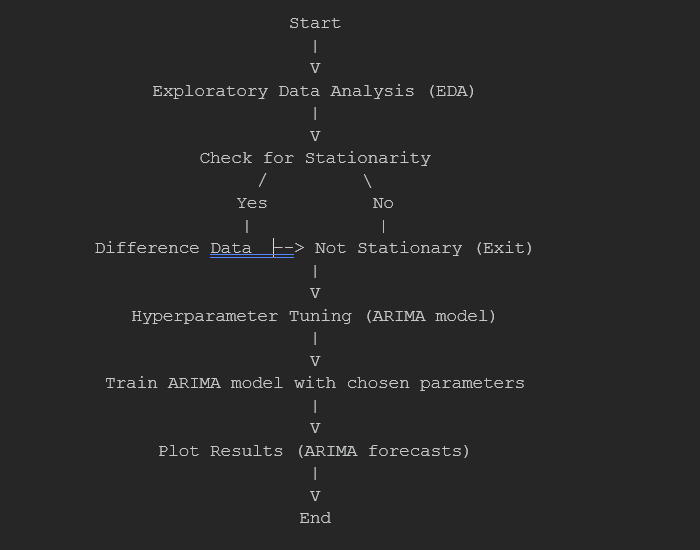
**2. Scalability: Machine learning algorithms are capable of processing vast amounts of temperature data efficiently, enabling the analysis of large-scale datasets and facilitating real-time monitoring and forecasting of temperature patterns.**

**3. Multimodal Data Integration: ML algorithms can integrate diverse data sources, such as satellite imagery, weather station data, and climate models. By incorporating multiple data modalities, these models can capture complex relationships and improve the accuracy of temperature predictions.**

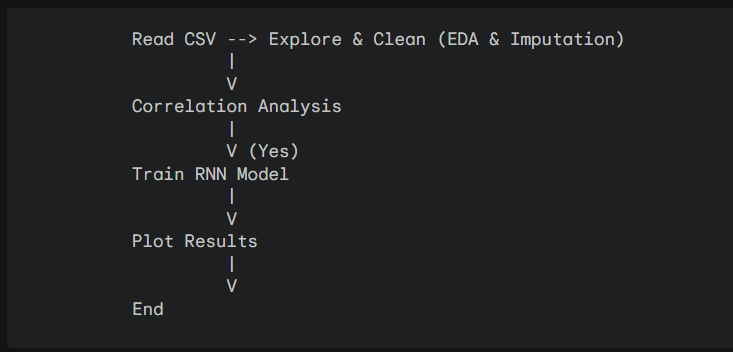
**4. Personalization: Machine learning techniques can enable personalized temperature forecasting tailored to specific geographic regions or environmental conditions. By considering local factors and historical trends, these models can provide more accurate and relevant temperature forecasts for various stakeholders, such as farmers, energy planners, and policymakers.**

**Flowcharts:**

**ARIMA Model**



**RNN Model**



* 1. **Objectives:**

1. Data Acquisition and Preprocessing

2. Accurate Temperature Prediction

3. Early Identification of Temperature Anomalies

4. Personalized Temperature Forecasting

5. User-friendly Interface and Accessibility

6. Interpretability and Transparency

7. Continuous Model Improvement

8. Comprehensive Evaluation and Validation

* 1. **Significance:**

1. Enhanced Planning and Decision-making

2. Risk Mitigation and Disaster Preparedness

3. Optimized Resource Management

4. Environmental Monitoring and Conservation

5. Climate Research and Policy Development

6. Business and Economic Planning

7. Improved Public Health and Safety

8. Scientific Advancements and Innovation

**Target Users:**

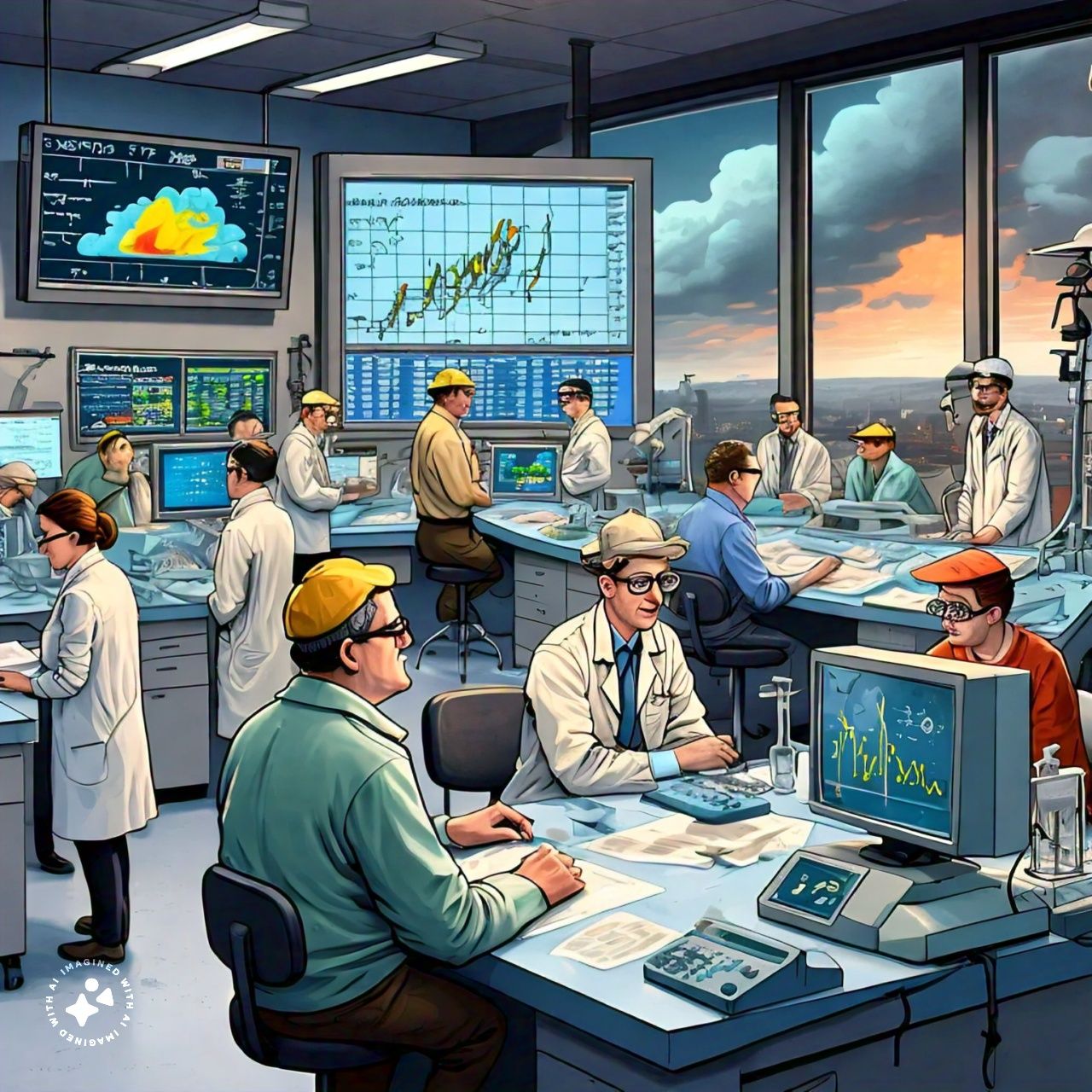
The survey form is distributed to various categories of users to gain insights and feedback for the future improvement and optimization of the temperature prediction model. The target audience includes:

1. Meteorologists and Climate Scientists:

2. General Public and Students:

3. Information Technology Professionals:

The collected data will be thoroughly analyzed to identify common themes, trends, and areas for improvement. Based on the feedback and opinions received, iterative enhancements and optimizations will be made to the temperature prediction model to better serve the needs of the users and enhance its overall value and utility.



1. **Problem Definition and Requirements:**

**Problem Statement:**

The objective of this project is to develop a robust machine learning model capable of accurately predicting temperature trends and anomalies based on historical weather data, geographical features, and meteorological variables. The model aims to analyze the intricate relationships between factors such as past temperature records, geographic location, elevation, atmospheric pressure, humidity, and seasonal variations to forecast future temperature patterns effectively. By leveraging advanced algorithms and feature engineering techniques, the goal is to create a reliable tool that can provide timely and precise temperature predictions, aiding stakeholders in diverse sectors such as agriculture, energy, transportation, and disaster management. Ultimately, the project seeks to enhance decision-making, resource allocation, and risk mitigation strategies by providing actionable insights into temperature dynamics and trends.

**Requirements:**

1. Python Programming Language: The project will be implemented using Python, which is a popular language for machine learning and data analysis tasks.
2. Python Integrated Development Environment (IDE):

* PyCharm: PyCharm is a powerful IDE developed by JetBrains, which provides excellent support for Python development, debugging, and integration with various libraries and frameworks.
* Jupyter Notebook: You also mentioned using Jupyter Notebook, which is a web-based interactive computing environment that allows you to combine code, visualizations, and narrative text.

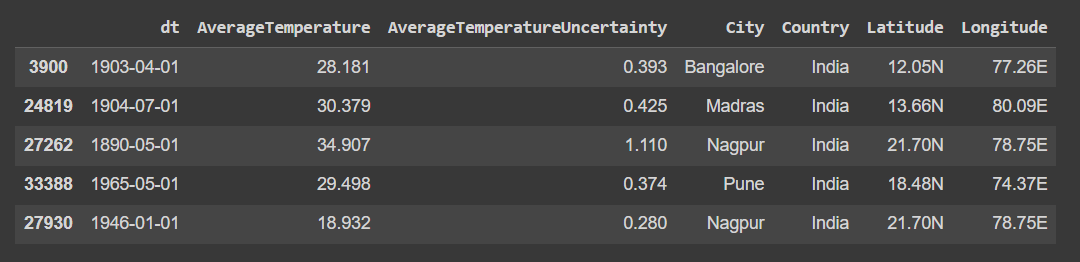
1. Python Libraries and Frameworks:

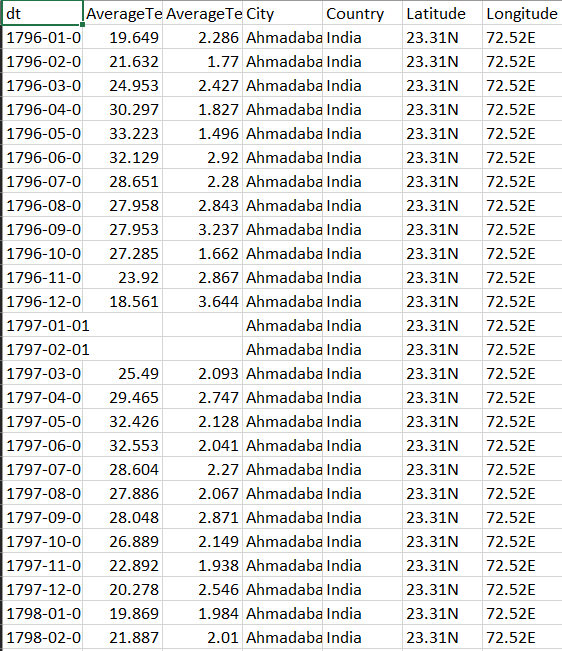
* NumPy: For numerical computing operations.
* Pandas: For data manipulation and analysis.
* Scikit-learn: A machine learning library for Python, providing a wide range of algorithms and tools for model building, evaluation, and deployment.
* Matplotlib and Seaborn: For data visualization and plotting.

1. Version Control System: Git: A distributed version control system widely used for tracking changes in source code and collaborating on projects.
2. **Materials and Methods:**

This section outlines the methodology employed in this study to predict temperature using machine learning techniques. The methodology encompasses data collection, preprocessing, feature extraction, and the selection and training of machine learning algorithms.

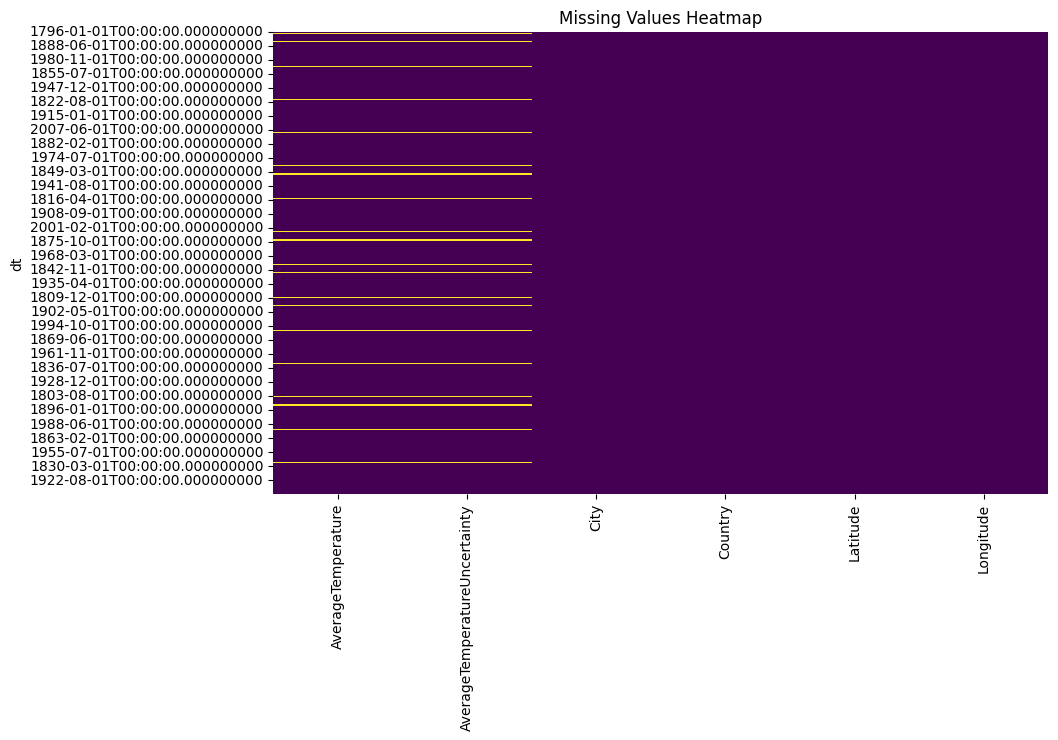
* 1. **About The Dataset:**

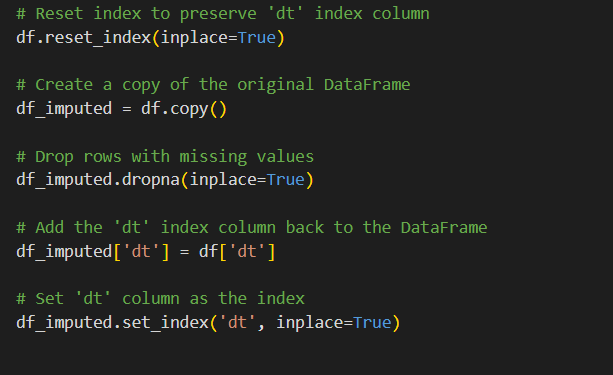




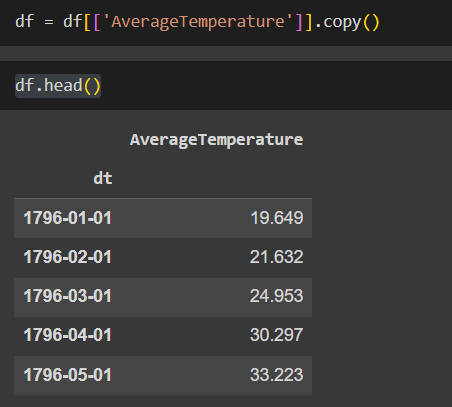
* 1. **Handling Missing Values and Relabelling Categorical Features**

****



****

* 1. **Removing Irrelevant or Redundant Features**



* 1. **Key Features:**

1. Year and Month:

- These features are extracted from the datetime index and utilized as inputs for predicting the average temperature. They capture the seasonal and annual variations in temperature.

2. Average Temperature:

- This is the target variable to be predicted by the model. The RNN model learns to predict future average temperatures based on historical data.

3. Average Temperature Uncertainty:

- This feature is used to calculate the upper and lower bounds of temperature predictions, providing insights into the uncertainty associated with the model's predictions.

These features collectively provide the necessary input for training the RNN model and conducting analysis to understand temperature trends and patterns over time in India.

1. **Data Analysis & Methodology:**

**Exploratory Data Analysis:**

**ARIMA Model**

1. Loading the Data:

- The code starts by loading the temperature data from a CSV file named 'india\_2.csv' into a pandas DataFrame (df).

2. Data Exploration:

- It samples 5 random rows from the DataFrame using `df.sample(5)` to get a glimpse of the data's structure and values.

- It displays the data types of each column in the DataFrame using `df.dtypes` to understand how the data is represented.

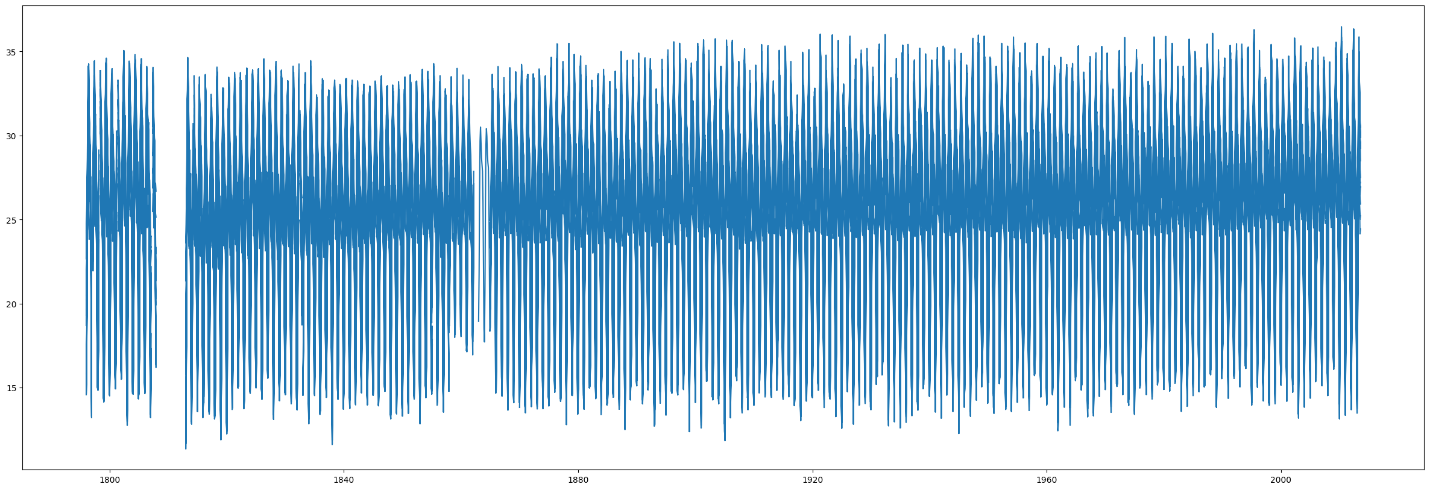
3. Data Preprocessing:

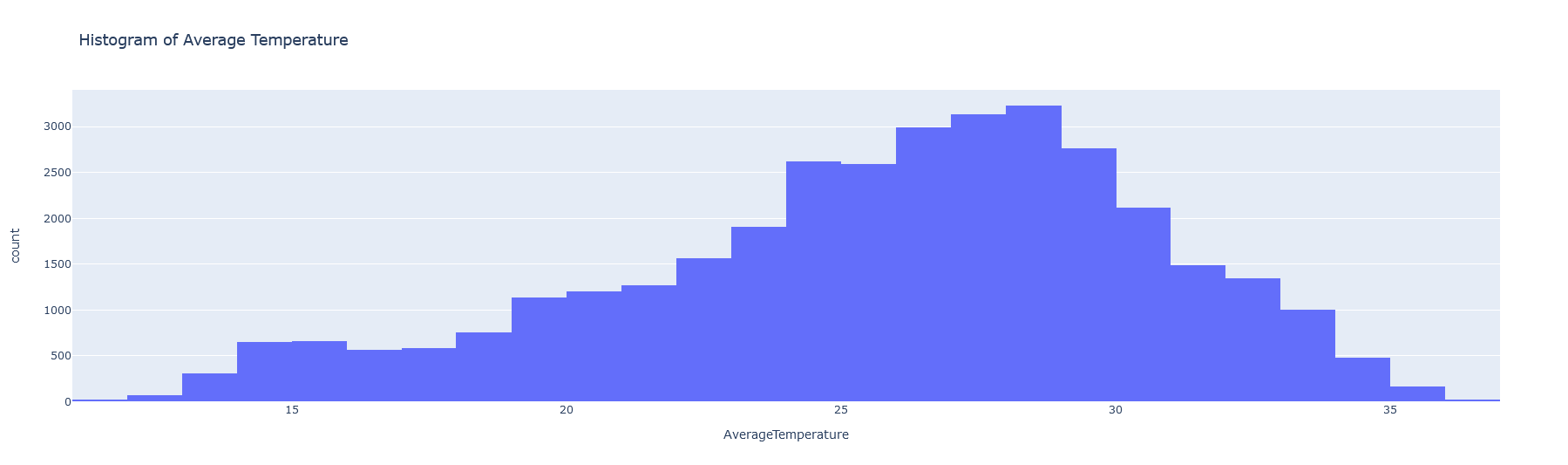
- The 'dt' column is stripped of any leading or trailing spaces and converted to a datetime object. This column is then set as the index of the DataFrame to facilitate time series analysis.

- The 'dt' column is dropped from the DataFrame as it is no longer needed for analysis.

4. Data Visualization:

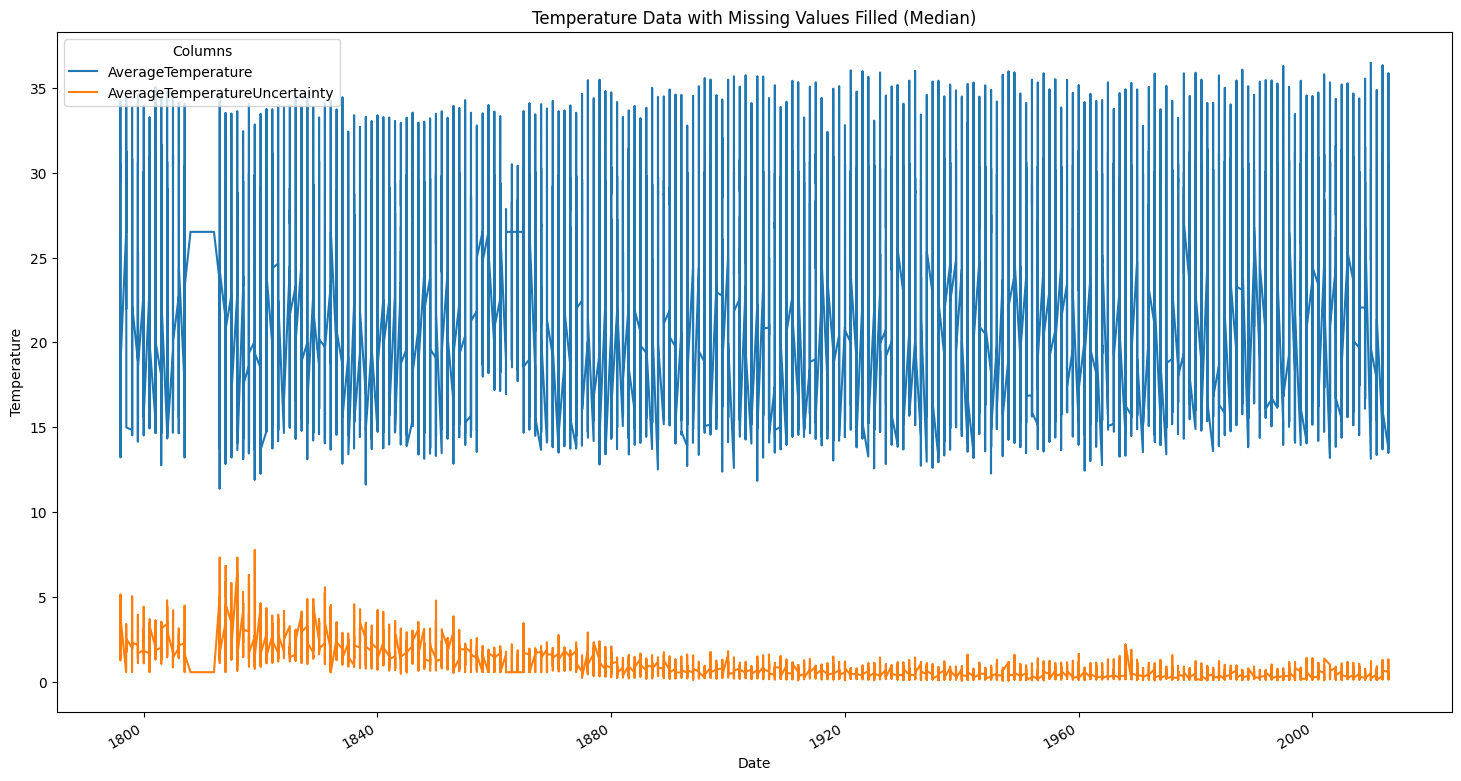
- A line plot is created to visualize the trend of average temperature over time.

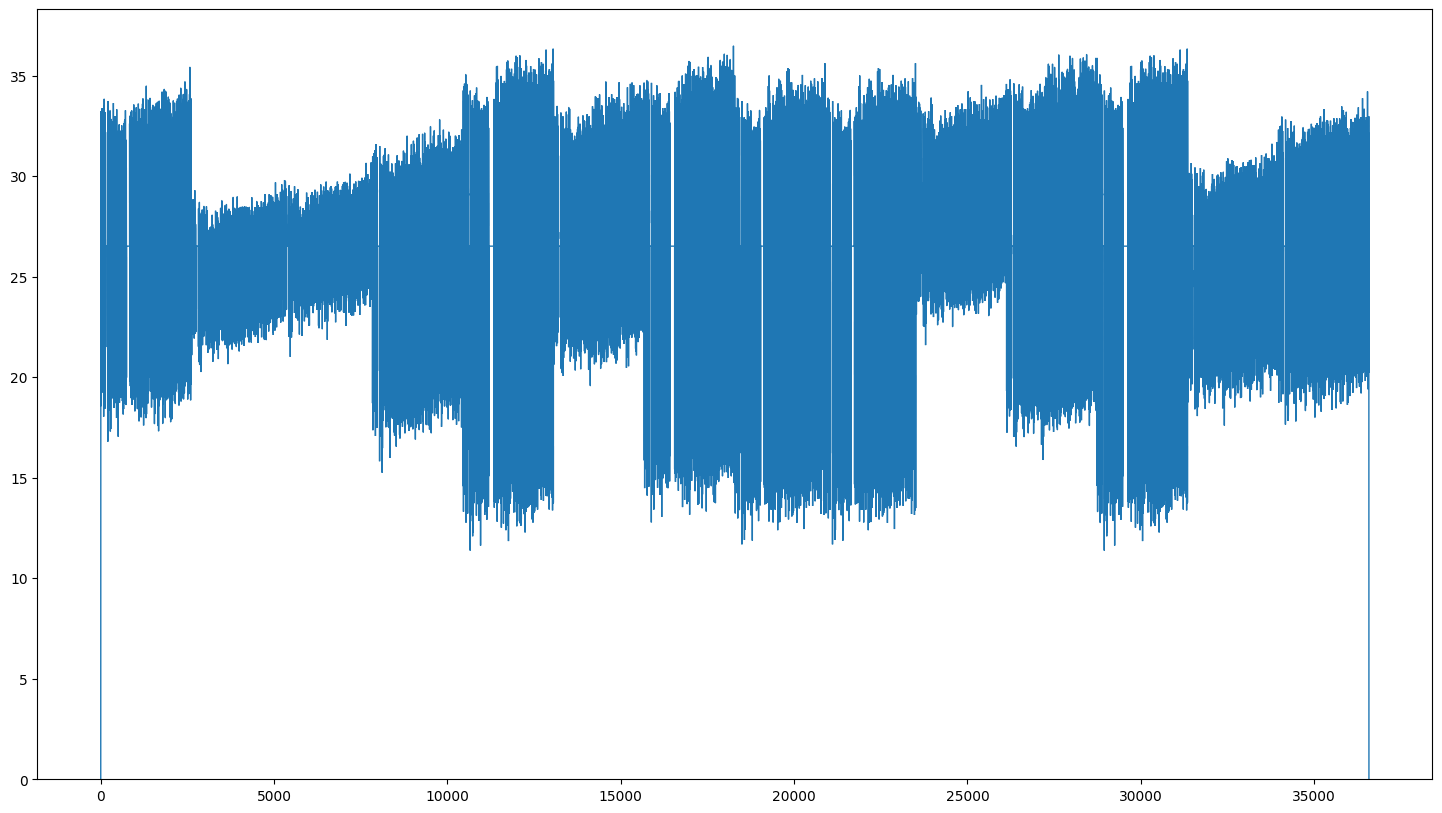


- A histogram is plotted to observe the distribution of average temperature values. 

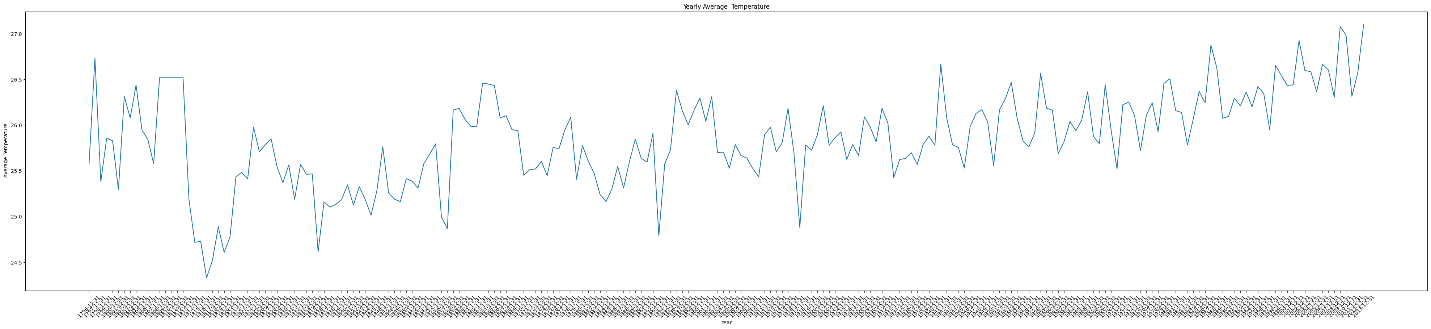
5. Handling Missing Values:

- The `fill\_missing\_values\_median` function is defined to fill missing values in the DataFrame using median imputation. This function is then applied to the 'AverageTemperature' and 'AverageTemperatureUncertainty' columns to fill any missing values with their respective medians.

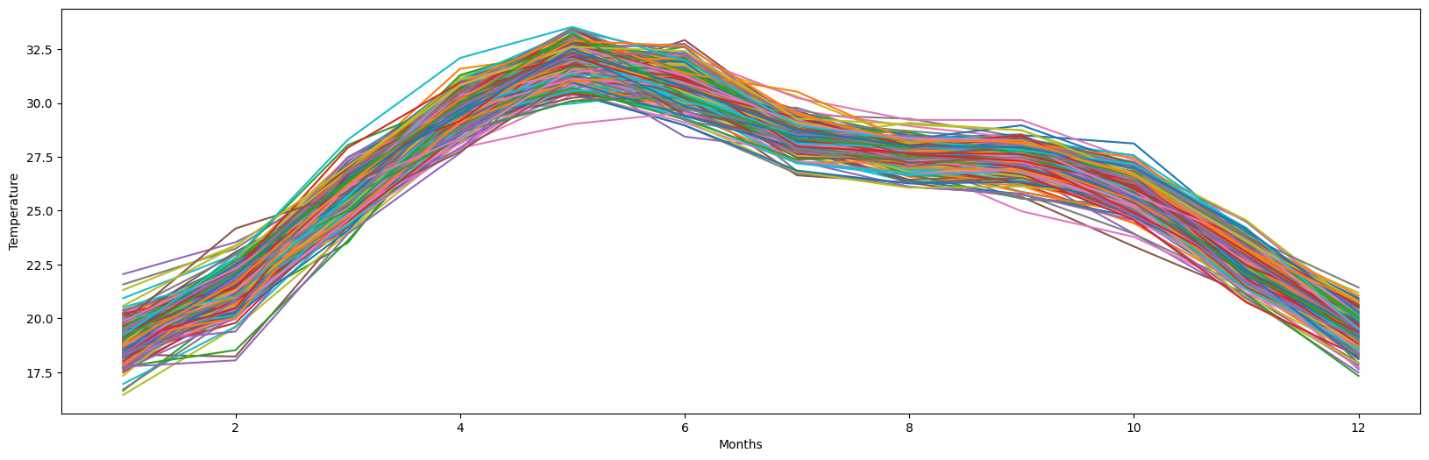




6. Time Series Analysis:

- Yearly average temperatures are calculated and plotted to identify long-term trends. 

- Monthly average temperatures are visualized to understand seasonal variations.

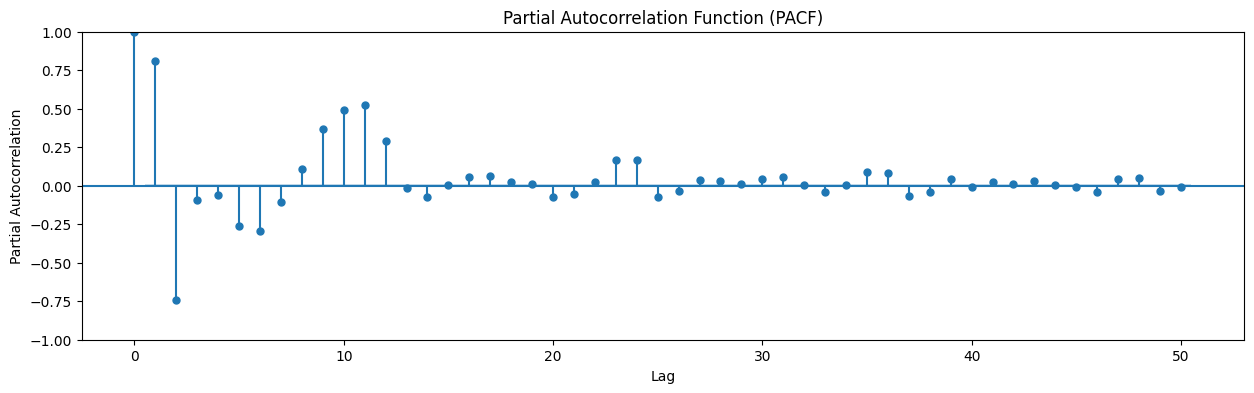
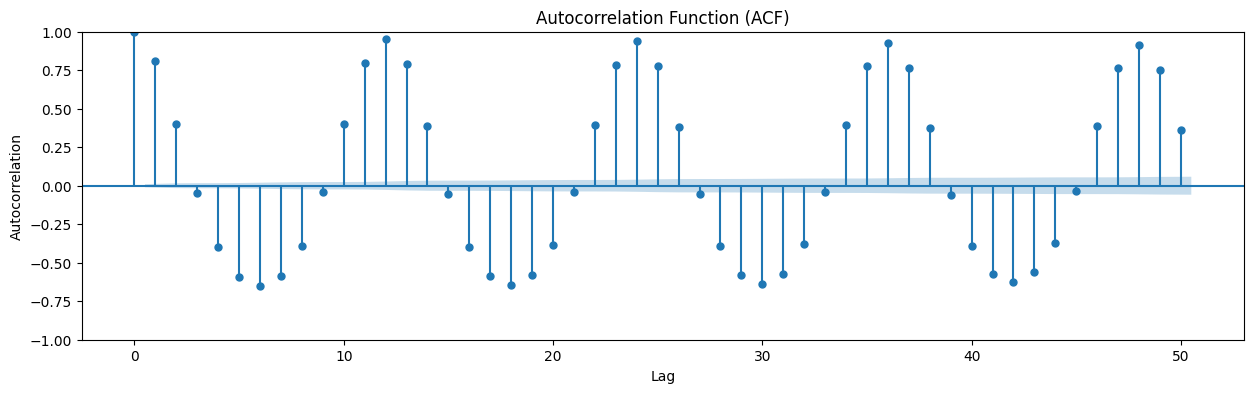


7. Stationarity Test:

- The Augmented Dickey-Fuller (ADF) test from the `statsmodels.tsa.stattools` module is used to test the stationarity of the temperature time series. The test statistic and p-value are printed to determine whether the data is stationary or non-stationary.

8. Autocorrelation and Partial Autocorrelation Analysis:

- Autocorrelation function (ACF) and partial autocorrelation function (PACF) plots are generated using `plot\_acf` and `plot\_pacf` functions to identify any autocorrelation patterns in the temperature data.



9. Modeling and Forecasting:

1. ARIMA Model

- An ARIMA (AutoRegressive Integrated Moving Average) model is fitted to the training data using the `ARIMA` class from the `statsmodels.tsa.arima.model` module.

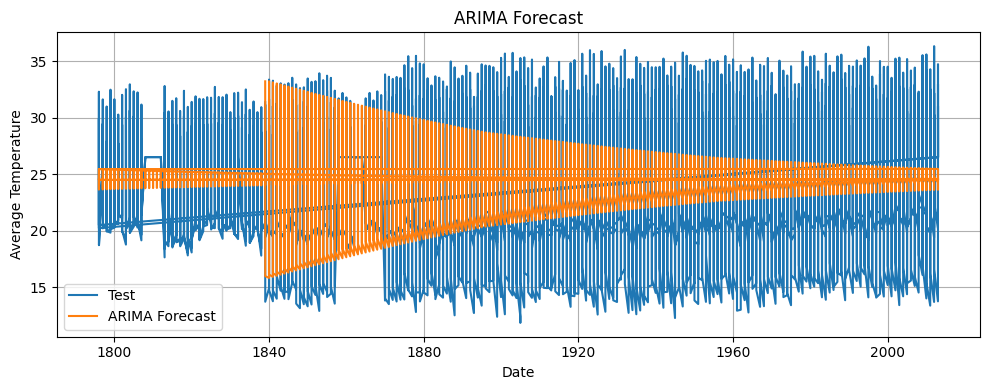
- The model's summary is printed to examine the model parameters and diagnostics.

- Forecasting is performed on the test data using the fitted ARIMA model, and the predicted values are plotted against the actual test data.

10. Model Evaluation:

- Mean Absolute Error (MAE) and Mean Squared Error (MSE) are calculated to evaluate the accuracy of the ARIMA model's forecasts compared to the actual test data.

Overall, this EDA provides insights into the temperature data's characteristics, ensures data quality through preprocessing, and sets the stage for modeling and forecasting temperature using time series analysis techniques like ARIMA.



**RNN Model**

1. Data Loading and Initial Inspection:

- The dataset `india\_2.csv` is loaded into a Pandas DataFrame (`df`).

- A sample of 5 random rows is displayed to get an initial understanding of the data's structure and content.

- Data types of each column are examined to ensure proper data handling.

2. Data Preprocessing:

- The datetime column is stripped of any leading or trailing whitespaces.

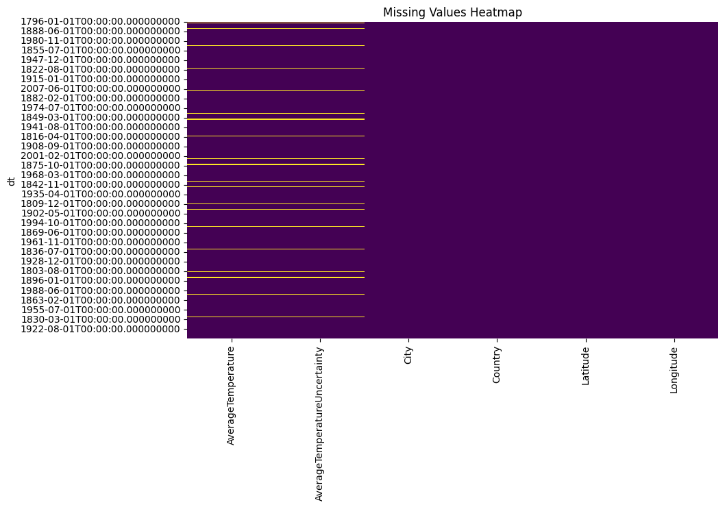
- The datetime column is converted to datetime format and set as the index for time series analysis.

- The 'dt' column is dropped as it's no longer needed.

3. Missing Values Analysis:

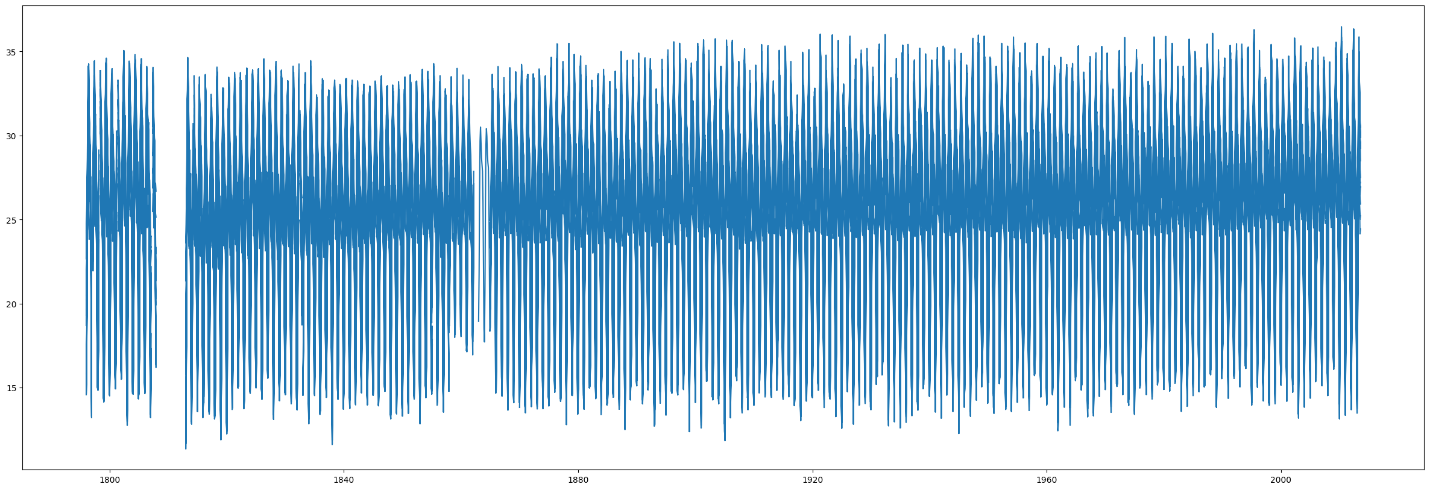
- A heatmap is plotted to visualize missing values in the dataset.

- The number of missing values in the 'AverageTemperature' column is printed.

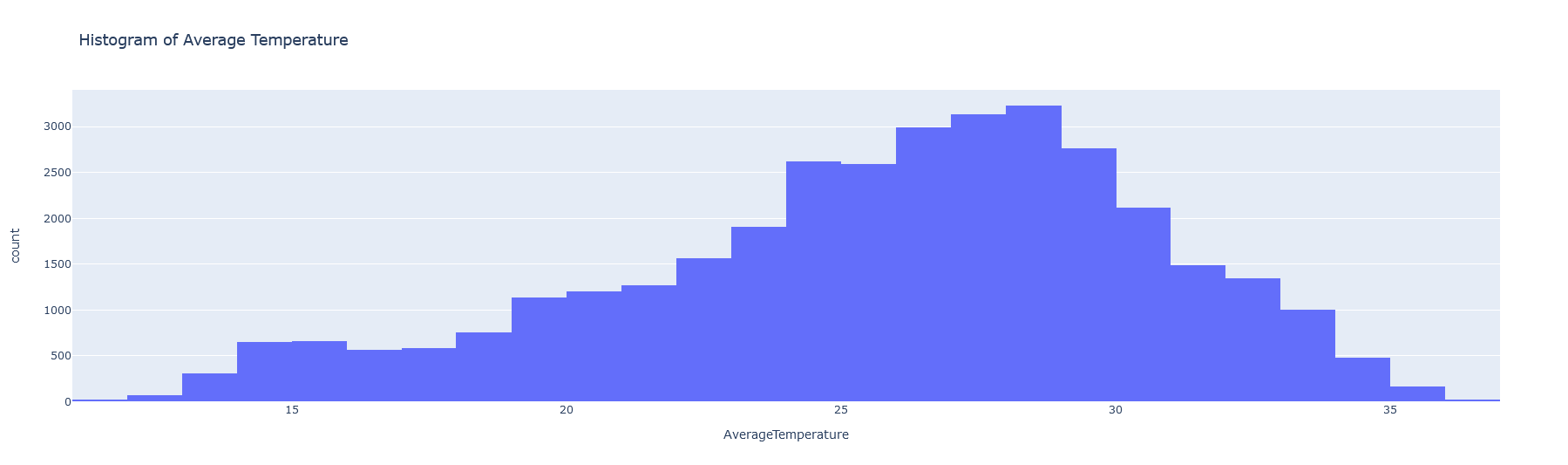


4. Data Visualization:

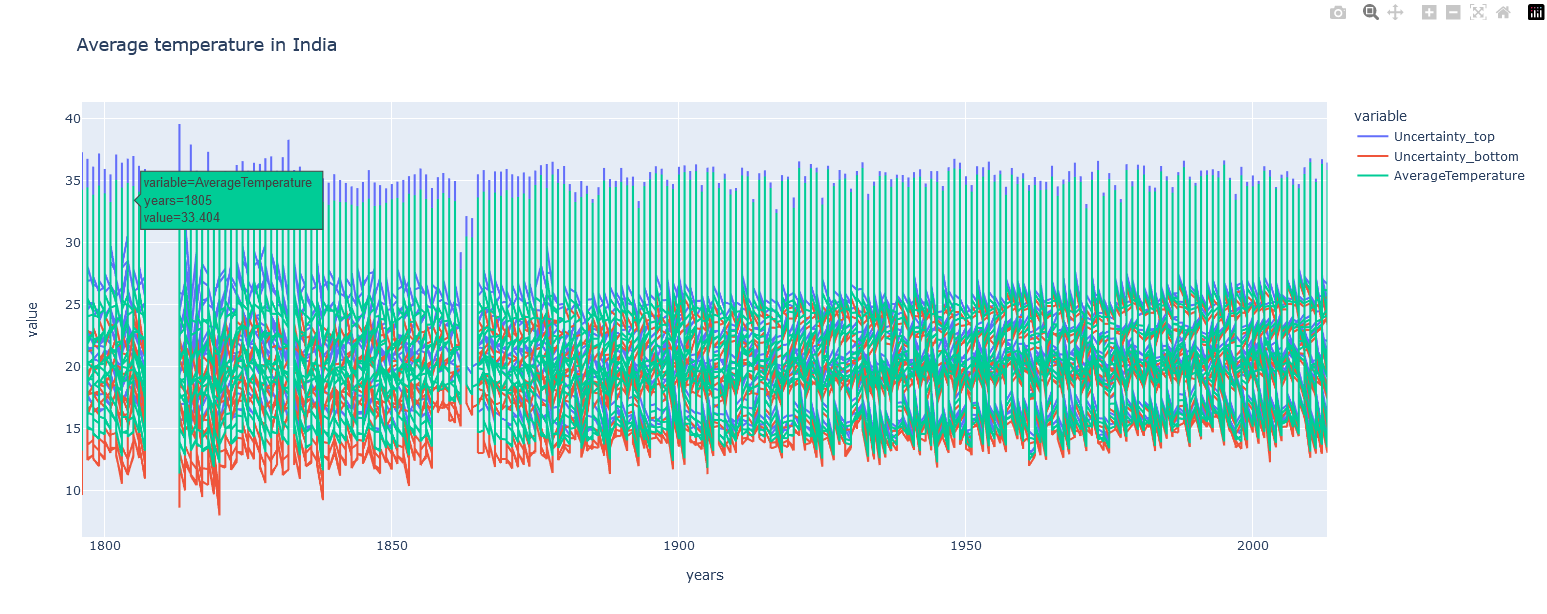
- A line plot is created to visualize the trend of average temperature over time.



- A histogram is plotted to observe the distribution of average temperature values.

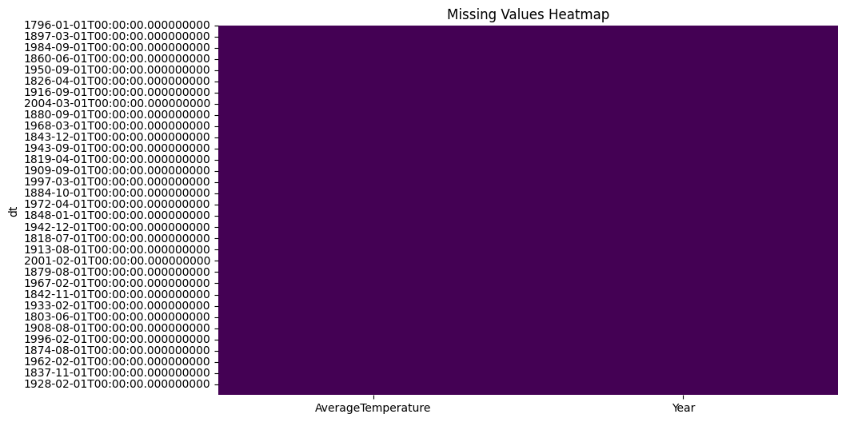


- A line chart is generated using Plotly Express to visualize the average temperature trend in India over the years.

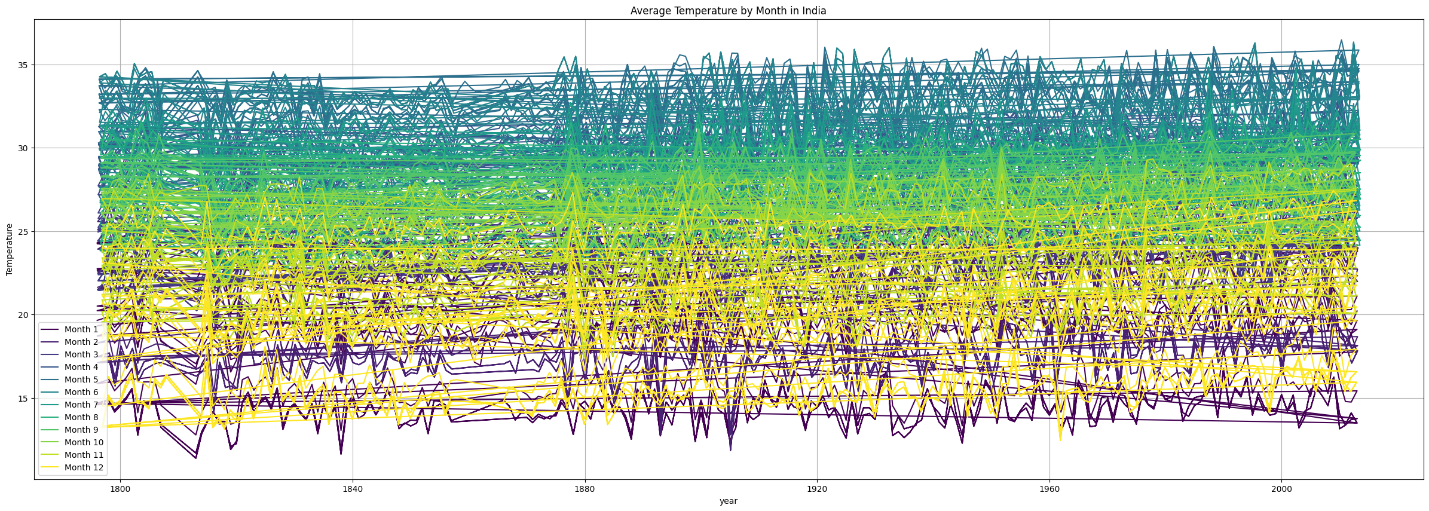


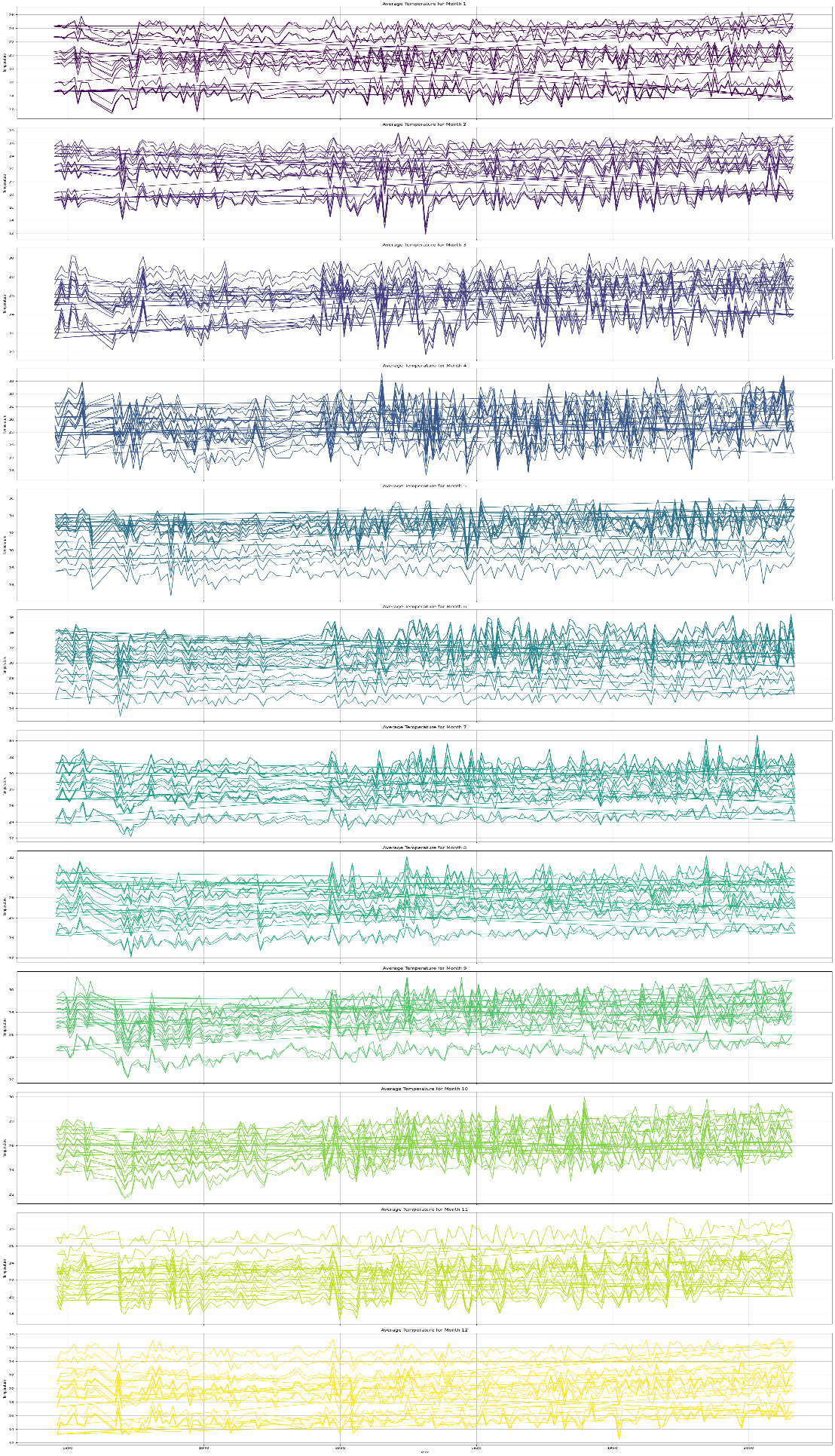
1. Data Drop:

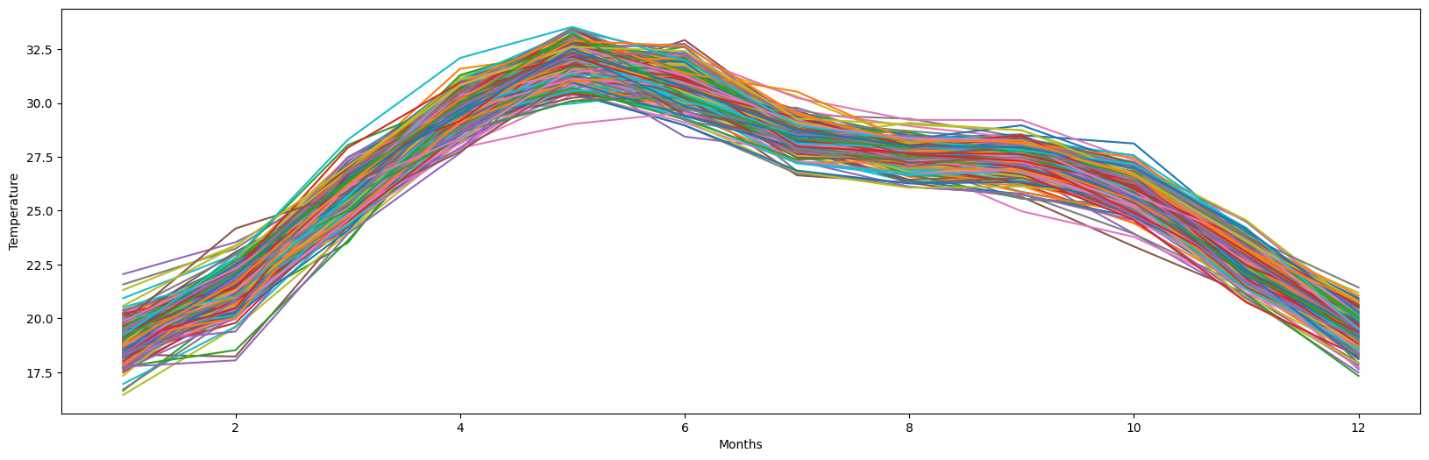
-Removed missing values from the dataframe df.



6. Time Series Analysis:

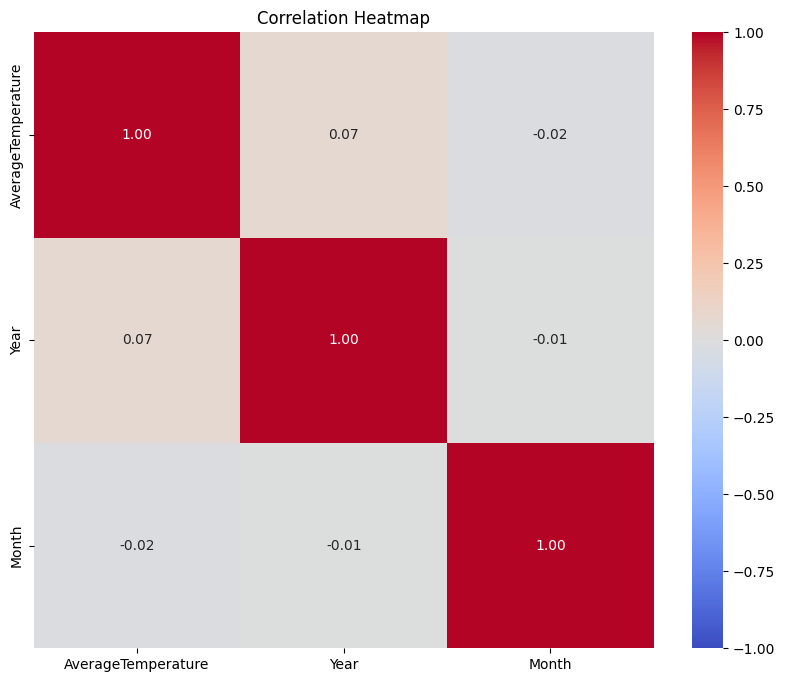
- Yearly average temperatures are calculated and plotted to identify long-term trends. 

- Monthly average temperatures are visualized to understand seasonal variations.



7. Correlation Analysis:

- A correlation heatmap is generated to visualize the correlation between different variables in the dataset.



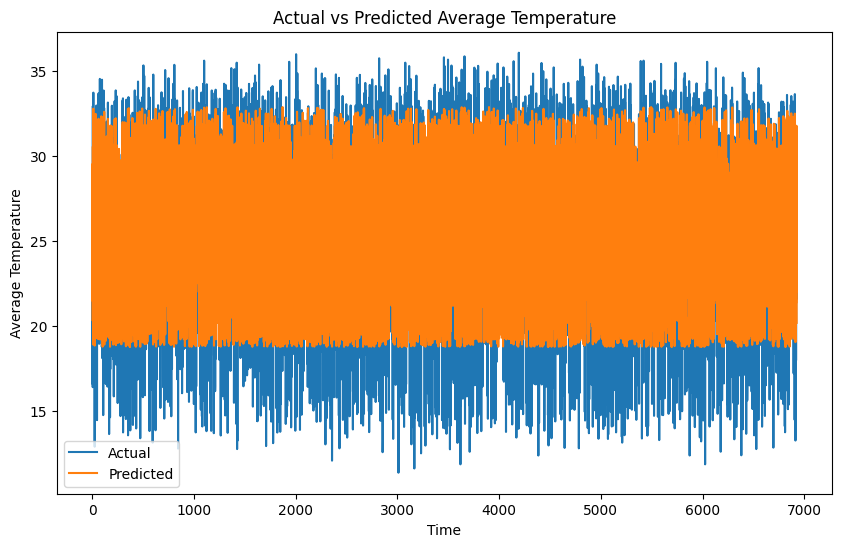
8. Machine Learning Model (RNN):

- Features (Year and Month) and the target variable (AverageTemperature) are scaled using Min-Max scaling.

- The dataset is split into training and testing sets.

- An RNN model is constructed and trained using TensorFlow's Keras API.

- The model's predictions are plotted against the actual temperatures to evaluate its performance.



9. Model Evaluation:

- The Mean Squared Error (MSE), Mean Absolute Error(MAE) is calculated to quantify the model's prediction accuracy.

1. **Analysis of Results with different features:**

The analysis of results with different features can provide valuable insights into how various factors influence temperature forecasting and the performance of the ARIMA model. Here's how we can conduct the analysis:

1. Feature Importance:

- Evaluate the importance of different features such as geographical location (latitude and longitude), urbanization level (urban, suburban, rural), altitude, and seasonal factors (month, season) in temperature forecasting.

- Use techniques like correlation analysis, feature importance scores from machine learning models, or domain knowledge to assess the impact of each feature on temperature predictions.

2. Effect of Missing Values Imputation:

- Compare the performance of the ARIMA model with and without missing values imputation.

- Analyze how filling missing values using median imputation affects the accuracy of temperature forecasts.

3. Stationarity and Seasonality:

- Assess the stationarity and seasonality of temperature data across different regions or time periods.

- Investigate how the presence of seasonality affects the model's ability to capture long-term trends and make accurate forecasts.

4. Model Performance Across Regions:

- Divide the dataset into different regions based on geographical features such as latitude, longitude, and altitude.

- Evaluate the ARIMA model's performance in forecasting temperatures for each region and compare the results.

- Identify regions where the model performs well and regions where it struggles, and analyze potential reasons for differences in performance.

5. Impact of External Factors:

- Consider external factors such as climate patterns, weather events, and human activities (e.g., urbanization, industrialization) that may influence temperature trends.

- Analyze how these external factors affect temperature forecasts and whether the ARIMA model adequately captures their effects.

6. Temporal Analysis:

- Conduct a temporal analysis to examine how temperature patterns have evolved over time.

- Identify any long-term trends, cyclic patterns, or irregular fluctuations in temperature data and assess how well the ARIMA model captures these patterns.

7. Comparison with Other Models:

- Compare the performance of the ARIMA model with other time series forecasting models, such as seasonal decomposition, exponential smoothing, or machine learning-based approaches.

- Evaluate the strengths and weaknesses of each model in capturing temperature variations and making accurate forecasts.

8. Sensitivity Analysis:

- Perform sensitivity analysis by varying model parameters (e.g., ARIMA order, seasonal differencing) and observing their impact on temperature forecasts.

- Determine optimal parameter values that maximize forecasting accuracy and stability across different features and regions.

By conducting a comprehensive analysis of results with different features, we can gain a deeper understanding of temperature forecasting dynamics and identify strategies to improve the accuracy and reliability of temperature predictions.

1. **Algorithms and Methodologies followed:**

The primary algorithm used in the provided code for temperature prediction is the AutoRegressive Integrated Moving Average (ARIMA) model and Recurrent Neural Networks (RNN). Here's a brief overview of the ARIMA algorithm and its methodology:

1. AutoRegressive Integrated Moving Average (ARIMA) Model:

- ARIMA is a widely used time series forecasting model that combines autoregression (AR), differencing (I), and moving average (MA) components.

- The ARIMA model is suitable for capturing linear relationships and temporal dependencies in time series data.

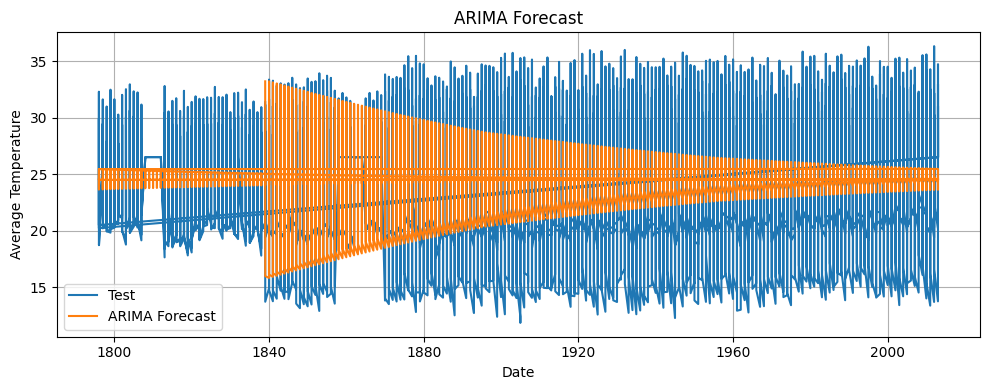
- It is particularly effective for modeling stationary or weakly stationary processes, where the mean, variance, and autocorrelation structure remain relatively constant over time.

1. Recurrent Neural Network (RNN) Model :

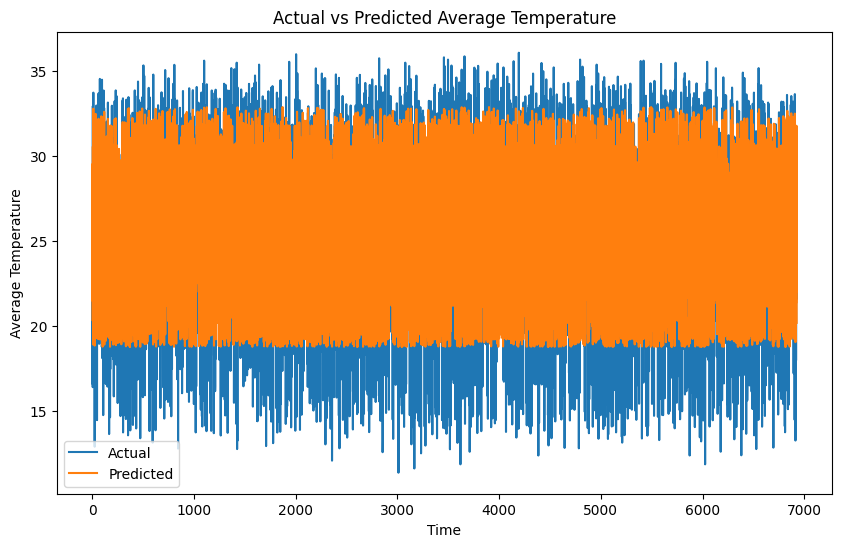
* RNN is utilized for time series forecasting of average temperatures in India.
* The RNN model, built using TensorFlow's Keras API, takes input features such as year and month and predicts the average temperature.
* After training and evaluation using Mean Squared Error, the model's predictions are compared against actual temperatures to assess its performance.
* RNNs are chosen for their ability to handle sequential data, making them suitable for time series analysis and prediction tasks like this one.

1. **Results:**

**ARIMA Model**



**RNN Model**

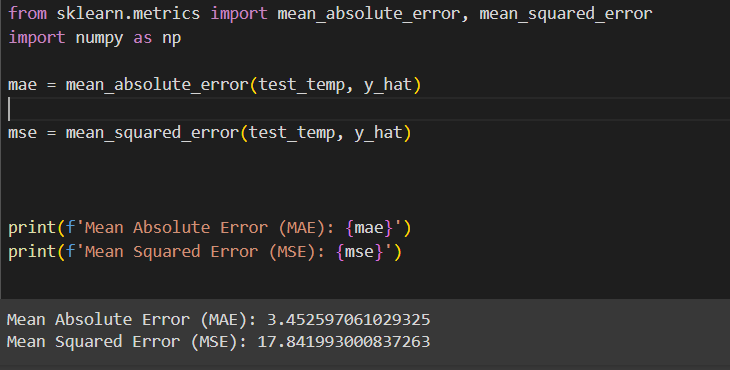


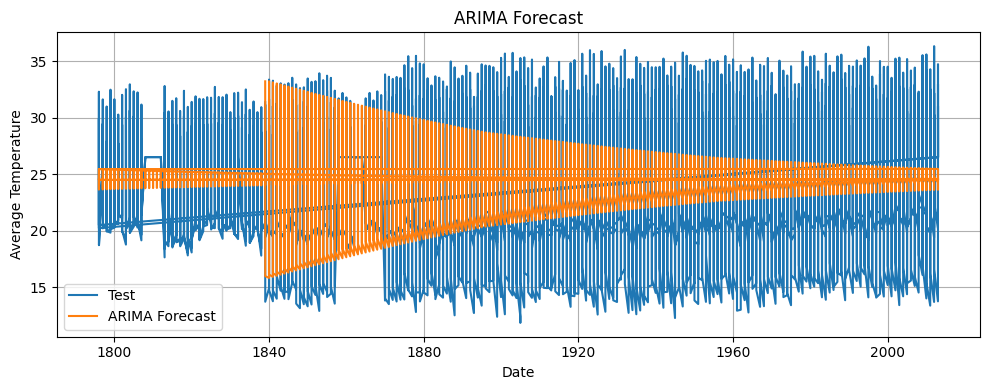
1. **Evaluation Metrics:**

**ARIMA Model**

Mean Absolute Error (MAE): 3.452597061029325

Mean Squared Error (MSE): 17.841993000837263

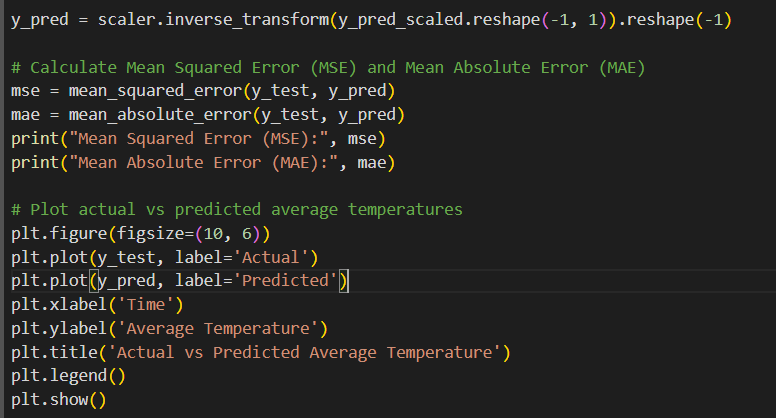
****



**RNN Model**

Mean Squared Error (MSE): 6.315542463921024

Mean Absolute Error (MAE): 1.9810709565614575

****

