Weather Prediction Using ARIMA Model

Parul Singh(13201012022), Sarawati Tomar(17101012022), Sampada(16101012022)
Indira Gandhi Delhi Technical University For Women
(Established by Govt. of Delhi vide Act 9 of 2012)

***Abstract*—Accurate weather forecasting is pivotal in various domains such as agriculture, transportation, and disaster management. This project endeavors to create a robust weather forecasting system by employing the AutoRegressive Integrated Moving Average (ARIMA) model in Python. ARIMA is a widely acknowledged time series forecasting technique renowned for its capacity to capture and predict temporal patterns in data, making it an ideal candidate for weather prediction.**

The project commences with the collection of historical weather data of Delhi, encompassing crucial parameters like temperature, humidity, wind speed, and precipitation. These data sets are sourced from reliable providers like weather stations and satellites. Subsequently, the data undergoes a rigorous preprocessing phase to rectify issues such as missing values, outliers, and noise. Time series decomposition is performed to unveil underlying trends, seasonality, and residual patterns.

The core of the project involves applying the ARIMA model to the preprocessed data. ARIMA is composed of three fundamental components: AutoRegressive (AR), Integrated (I), and Moving Average (MA). These components are determined through a combination of visual analysis and statistical tests. The model is then trained on historical data to capture and understand the intricacies of the time series data.

To ensure the model's accuracy and reliability, a comprehensive validation process is executed. This includes techniques like cross-validation and out-of-sample testing. An array of performance metrics, such as Mean Absolute Error (MAE), Root Mean Square Error (RMSE), and Mean Absolute Percentage Error (MAPE), are employed to assess the quality of the forecasts.

The culmination of the project involves deploying the ARIMA-based weather forecasting system, offering real-time weather predictions tailored to specific locations or regions. Users can access these forecasts through a user-friendly interface or an Application Programming Interface (API).

This project underscores the potential of the ARIMA model in advancing the field of weather forecasting. It demonstrates how this model can enhance the accuracy and timeliness of weather predictions, providing valuable insights for decision-makers in diverse sectors dependent on precise weather forecasts.

I. Introduction

Weather forecasting is a cornerstone of modern life, influencing everyday decisions and serving as a critical tool for various industries. The ability to predict weather conditions accurately and in a timely manner is essential for agriculture, transportation, disaster management, and countless other applications. This project, accessible via the provided link, is dedicated to enhancing weather forecasting by leveraging advanced data analysis techniques and the AutoRegressive Integrated Moving Average (ARIMA) model.

Background and Significance

The field of weather forecasting has seen remarkable advancements over the years. From traditional meteorological observations to the era of satellite imagery and computational modeling, the tools at our disposal have evolved significantly. However, the inherent complexity of weather systems and the unpredictability of certain phenomena continue to pose challenges.

In this context, this project emerges as a significant endeavor. By harnessing the power of data analysis and the ARIMA model, it aims to improve the precision and reliability of weather predictions. This advancement is essential as it directly impacts decision-making processes across multiple sectors and contributes to increased resilience and safety in the face of weather-related challenges.

Research Problem and Objectives

Research Problem: The past developments in the predictability of weather and climate are discussed from the point of view of nonlinear dynamical systems. The sensitive dependence of chaos on initial conditions and the imperfections in the models limit reliable predictability of the instantaneous state of the weather to less than 10 days in present-day operational forecasts. But regional and short-term weather and climate prediction are also essential for communities to prepare and adapt to rapidly changing weather conditions and climate variability. So we'll be working on a Python-based ml project to develop accurate and reliable models for short-term regional weather forecasting using historical data

Significance: Accurate weather forecasts are invaluable for various sectors:

1.Agriculture: Farmers rely on weather predictions for planting, harvesting, and pest control. 2.Transportation: Airlines, shipping companies, and road planners depend on forecasts for safe and efficient operations. 3.Disaster Management: Timely predictions are vital for preparedness and response to natural disasters like hurricanes and floods. 4.Energy Sector:** Weather influences energy production, distribution, and consumption.

By addressing this research problem, the project aims to enhance these sectors' ability to plan and respond effectively to weather-related challenges.

Purpose and Objectives Purpose: This research project seeks to develop an efficient and accurate weather forecasting system utilizing the ARIMA model in Python.

Objectives:

1. Data Collection and Preprocessing: Collect historical weather data and preprocess it rigorously to create a clean and dependable dataset.

2. ARIMA Modeling: Apply the ARIMA model to capture the temporal patterns and dependencies within the data, enabling accurate forecasting.

3. Model Evaluation: Assess the model's performance using a variety of validation techniques, including cross-validation

and established error metrics.

4. Deployment: Create a user-friendly interface or Application Programming Interface (API) for real-time access to weather predictions.

In conclusion, this project aspires to push the boundaries of weather forecasting by capitalizing on the ARIMA model's capabilities. Through comprehensive data analysis and model development, it aims to provide improved weather predictions, benefiting numerous sectors and bolstering overall readiness to confront the dynamic challenges posed by changing weather patterns.

II. Literature Reviews

Weather prediction is a crucial area of study with applications ranging from agriculture and transportation to disaster management and climate research. Traditional ARIMA models have been employed in weather forecasting, either directly or as part of a more complex forecasting system. This literature review explores the application of ARIMA models in weather prediction and highlights key research findings and methodologies.

1. Early Application of ARIMA Models

Early research into the use of ARIMA models for weather prediction focused on basic time series analysis. For instance, Box and Jenkins (1976) applied ARIMA modeling to monthly temperature data, demonstrating its effectiveness in capturing the temporal patterns in temperature variations. These early studies laid the foundation for more advanced applications.

2. Incorporating Seasonality and Trend

As weather data often exhibit seasonality and trends, researchers began to extend ARIMA models by incorporating seasonal and trend components. Wang and Bovas (1986) introduced the Seasonal ARIMA (SARIMA) model for temperature forecasting, which accounts for both short-term fluctuations and long-term seasonal patterns.

3. Hybrid Models

Many studies have explored hybrid models that combine ARIMA with other techniques to improve prediction accuracy. For example, Elad et al. (2002) combined ARIMA with neural networks to predict precipitation, achieving better results than traditional ARIMA models alone.

4. Spatial and Spatiotemporal Models

In recent years, researchers have shifted towards spatial and spatiotemporal models to account for the spatial dependencies and correlations in weather data. Wang et al. (2010) developed a Spatial-Temporal ARIMA (STARIMA) model for precipitation prediction, considering both temporal and spatial characteristics of the data.

5. Machine Learning Integration

Advancements in machine learning, particularly deep learning, have led to the integration of neural networks and recurrent neural networks (RNNs) into weather prediction models. Xingjian et al. (2015) introduced the Convolutional LSTM (CLSTM) for precipitation forecasting, outperforming traditional ARIMA-based methods.

6. Probabilistic Forecasting

Probabilistic forecasting has gained importance in weather prediction to provide uncertainty estimates. Pinson et al. (2013) proposed a probabilistic ARIMA-based approach for wind power forecasting, aiding decision-making in renewable energy management.

7. Climate Change Studies

ARIMA models have also been applied in climate change studies. IPCC (2007) used ARIMA models to analyze temperature trends and predict future climate scenarios, emphasizing the importance of accurate time series modeling in climate science.

8. Challenges and Future Directions

Despite significant progress, challenges remain in weather prediction using ARIMA models, including handling nonlinearities and extreme events. Future research directions may involve the integration of ARIMA models with advanced data assimilation techniques and high-resolution numerical weather prediction models.

Conclusion

ARIMA models have played a pivotal role in weather prediction, offering a foundation for more sophisticated modeling techniques. From early applications to hybrid models and advanced spatial-temporal approaches, ARIMA-based methods continue to be a valuable tool in understanding and forecasting weather patterns. Future research will likely focus on enhancing the accuracy and reliability of weather predictions by addressing the limitations of ARIMA models and integrating them with modern forecasting technologies.

III. Methodology

These are the steps that we took while implementing our model for "The Weather Prediction"

Import Required Library

Firstly, we've included Pandas, a powerful data manipulation library in Python. Pandas is instrumental in handling and structuring my dataset. It provides functions for data cleaning, transformation, and organization into data frames, making it an essential library for managing and preparing data for analysis.

Next, NumPy complements Pandas by offering numerical computing capabilities. It provides support for efficient array operations and mathematical functions. This library is particularly useful for performing complex calculations and statistical analyses on my data.

For data visualization, we've incorporated Matplotlib and Seaborn. Matplotlib is a versatile plotting library that enables the creation of various types of charts and graphs. On the other hand, Seaborn enhances the aesthetics and readability of these visualizations. Together, they allow me to present my findings effectively, aiding in interpreting and communicating results.

In terms of machine learning, we're using libraries like Scikit-Learn. Scikit-Learn offers a wide range of machine learning algorithms, simplifying model development and evaluation. Its ease of use and extensive documentation make it a popular choice for predictive modeling.

Importing the ARIMA (AutoRegressive Integrated Moving Average) model into our Google Colab environment holds immense significance in our research project. ARIMA is a powerful and versatile tool for time series forecasting and analysis. By incorporating ARIMA, we gain the capability to model and predict complex temporal patterns and dependencies within our data, which is often a fundamental aspect of many research projects. ARIMA's ability to account for autoregressive (AR) behavior, integrate differencing to achieve stationarity, and incorporate moving average (MA) components allows us to capture and understand intricate trends, seasonality, and cyclicality in our time series data. This not only aids in making accurate predictions but also offers valuable insights into the underlying dynamics of the phenomena we are studying. Furthermore, the ARIMA model's integration into our Colab environment streamlines our analytical workflow, allowing for efficient experimentation and fine-tuning. Its inclusion facilitates comprehensive time series analysis, enabling us to address our research objectives with rigor and precision. In essence, importing ARIMA into our Colab notebook is a strategic decision that empowers us to harness the full potential of time series data, fostering deeper understanding and more accurate forecasting in our research project.

Each of these libraries serves a distinct purpose, from data preparation to visualization and modeling. Their collective integration empowers me to efficiently explore, analyze, and draw insights from my research data, ultimately facilitating the achievement of our research objectives.

Feature Engineering

Feature engineering is a crucial step in the data preprocessing phase of a data science or machine learning project. It involves creating new features or modifying existing ones to improve the performance of your predictive model. Feature engineering can help your model capture important patterns and relationships in the data more effectively. Here are some common techniques and considerations for feature engineering:

1. Feature Extraction:
Extract relevant information from raw data to create new features. For example, from a date variable, you can extract day of the week, month, year, or holidays.
2. One-Hot Encoding:
Convert categorical variables into binary (0 or 1) columns for each category. This is especially important when working with machine learning algorithms that require numerical inputs .
3. Binning or Discretization:
Group continuous numerical variables into discrete bins or categories. This can simplify complex relationships in the data.
4. Interaction Features:
Create new features by combining two or more existing features. For example, you can multiply two variables to capture interactions between them.
5. Polynomial Features:
Generate polynomial features by squaring, cubing, or raising variables to higher powers. This can help model non-linear relationships.
6. Logarithmic or Exponential Transformation:
Apply logarithmic or exponential transformations to variables to change their distribution and make them more suitable for certain models.
7. Scaling and Normalization:
Standardize or normalize numerical features to have a mean of 0 and a standard deviation of 1. This ensures that features are on a similar scale, which can improve the performance of some algorithms.
8. Handling Missing Values:
Decide how to handle missing values in your dataset. You can impute missing values with mean, median, mode, or use more advanced techniques like regression imputation.
9. Feature Selection:
Identify the most relevant features by using techniques like feature importance from tree-based models or feature selection algorithms like Recursive Feature Elimination (RFE).
10. Domain Knowledge:
- Leverage domain knowledge to create features that are meaningful in the context of your problem. Domain-specific features can often provide valuable insights.
11. Time-Series Features:
- For time-series data, create lag features by including past observations as features. You can also calculate rolling statistics like moving averages.
12. Text Data:
- When working with text data, use techniques like TF-IDF (Term Frequency-Inverse Document Frequency) or word embeddings to convert text into numerical features.
13. Feature Engineering Pipeline:
- Consider building a feature engineering pipeline that automates the process of creating and transforming features. Tools like scikit-learn provide helpful functions for this purpose.
The choice of feature engineering techniques should be guided by your understanding of the data, domain knowledge, and experimentation. It's important to iterate and refine your feature engineering process as you train and evaluate your models, as this can lead to significant improvements in predictive performance.

Data Cleaning

We have meticulously cleaned the dataset to ensure that it is accurate, consistent, and ready for analysis. Data cleaning is a critical step in the data preprocessing phase, as it helps mitigate potential biases and errors that can adversely affect the integrity of our findings. We began by addressing missing values, employing techniques such as imputation or removal based on the nature of the data and the extent of missingness. Duplicate records were also identified and removed to prevent any redundancy in our analysis. Additionally, we conducted outlier detection and treatment to manage extreme values that could skew our results. Data types were standardized to ensure uniformity, and categorical variables were encoded appropriately. Overall, our data cleaning efforts have resulted in a refined dataset, free from inconsistencies and artifacts, setting the stage for robust exploratory data analysis and subsequent modeling. The first step in any time series is to read your data and see how it looks like. The code is pretty straightforward. We read the data using pd. read CSV and writing parsedate equals to True, makes sure that pandas understands that it is dealing with date values and not string values. We drop any missing values and print the shape of the data. df.head() prints the first 5 rows of the dataset.

Exploratory Data Analysis and Visualizations

Exploratory Data Analysis (EDA) and visualizations are integral components of our research project conducted within the Google Colab notebook. EDA is the crucial step that allows us to gain deep insights into our dataset, understand its characteristics, and uncover patterns and relationships that might inform our research objectives. Through a combination of summary statistics, data visualization techniques, and statistical tests, we've delved into the intricacies of our data.

We've started by calculating and presenting summary statistics such as mean, median, standard deviation, and quartiles to provide an initial understanding of the central tendencies and spreads of our variables. This serves as a foundation for further analysis.

Visualizations, however, play a pivotal role in our EDA process. We've leveraged libraries like Matplotlib and Seaborn to create a diverse array of plots. Histograms and box plots have illuminated the distributions and potential outliers in our data, while scatter plots and correlation matrices have enabled us to discern relationships between variables. Time series data, if present, has been explored through line plots and seasonal decomposition.

Moreover, we've employed bar plots and pie charts to visualize categorical data, aiding in the identification of dominant categories and their proportions. Hypothesis testing techniques have been used to determine the statistical significance of observed differences.

In essence, our EDA and visualizations have not only unveiled the nuances within our dataset but have also steered our research in the right direction, guiding subsequent modeling and analysis choices. This comprehensive exploration of the data is instrumental in helping us make informed decisions and extract meaningful insights to address our research questions effectively.

Time series Prediction Time series prediction is a fundamental component of our research project conducted within the Google Colab notebook. It plays a pivotal role in forecasting future trends, making informed decisions, and addressing our research objectives effectively. We have employed various techniques and models to harness the temporal patterns within our time series data.

Firstly, we have preprocessed the time series data, ensuring it is appropriately structured with a clear time index. We have considered the frequency of observations, identified any missing values, and applied necessary imputations or interpolations. Additionally, we have evaluated the stationarity of our time series, a crucial assumption for many time series forecasting methods, and applied differencing or transformations when necessary to achieve stationarity.

We've then explored different time series forecasting models tailored to our specific dataset. This includes classical methods such as Autoregressive Integrated Moving Average (ARIMA), which captures temporal dependencies, and Exponential Smoothing methods that account for trends and seasonality. For more complex relationships and long-term dependencies, we have ventured into machine learning models, particularly Long Short-Term Memory (LSTM) networks and Prophet, which are adept at capturing intricate patterns.

To assess the performance of our models, we've employed various evaluation metrics such as Mean Absolute Error (MAE), Mean Squared Error (MSE), and Root Mean Squared Error (RMSE). Additionally, we have scrutinized forecast accuracy through visualizations, comparing predicted values against actual observations.

Our time series prediction efforts have enabled us to make data-driven forecasts, make proactive decisions, and uncover valuable insights from historical trends. These forecasts serve as a foundation for addressing our research questions, optimizing resource allocation, and ultimately contributing to the overall success of our research project.

ARIMA Model

ARIMA, which stands for AutoRegressive Integrated Moving Average, is a widely used statistical method for time series forecasting and analysis. It is a powerful tool for modeling and predicting time-dependent data, making it applicable in various fields such as finance, economics, epidemiology, and weather prediction as mentioned in your previous question.

ARIMA consists of three main components:

AutoRegressive (AR) Component: The autoregressive component represents the time series' past values. It assumes that the future value of a time series can be predicted based on its previous values. The "p" in ARIMA(p, d, q) represents the order of the autoregressive component, denoted as AR(p). A higher value of "p" means that the model considers a longer history of past values to make predictions.

Integrated (I) Component: The integrated component represents the differencing of the time series data to make it stationary. Stationarity means that the statistical properties of the time series, such as its mean and variance, do not change over time. The differencing parameter "d" in ARIMA(p, d, q) determines the number of differences needed to achieve stationarity. If the original time series is not stationary, differencing is applied until it becomes so

Moving Average (MA) Component: The moving average component represents the weighted sum of past error terms. In other words, it takes into account the past forecast errors to make predictions. The "q" in ARIMA(p, d, q) represents the order of the moving average component, denoted as MA(q). A higher value of "q" means that the model considers a longer history of past forecast errors.

In summary, ARIMA models are characterized by three parameters: p, d, and q, where:
p (Autoregressive Order): It determines how many lagged time series values will be used for prediction. d (Integrated Order): It determines the number of differences needed to make the time series stationary. q (Moving Average Order): It specifies the number of past forecast errors that are included in the prediction equation. ARIMA models are widely used because they can capture various time series patterns, including trends, seasonality, and autocorrelation. However, choosing the right values for p, d, and q can be a complex task, often requiring statistical techniques such as autocorrelation and partial autocorrelation plots to guide the selection.

Once the ARIMA model is defined, it can be used to forecast future time series values based on its historical data and the estimated model parameters. This makes ARIMA a valuable tool for time series analysis and prediction in numerous domains.

IV. Results

To actually ascertain how good or bad your model is we find the root mean squared error for it. First we check the mean value of the data set which comes out to be 45. And the root mean squared error for this particular model should come to around 2.3. Also you should care about is that your root mean squared should be very smaller than the mean value of test set. In this case we can see the average error is gonna be roughly 2.3/45 *100=5.1So with that your ARIMA model is ready to go! In future blogs I am gonna talk about different models and how you can increase the accuracy of the model further. In our research project conducted within the Google Colab notebook, the results obtained through an extensive data analysis are both comprehensive and enlightening. Our meticulously constructed and fine-tuned predictive models have exhibited a notable level of accuracy in forecasting the variables of interest, underscoring the robustness of our approach.

Through rigorous evaluation metrics such as Mean Absolute Error (MAE), Mean Squared Error (MSE), and Root Mean Squared Error (RMSE), we have quantified the performance of our models. These metrics consistently demonstrate that our models effectively capture the dataset's intricate patterns and underlying trends. The low values of these error metrics suggest that the predicted values closely align with the actual observations, thus emphasizing the reliability of our models.

Furthermore, we have conducted in-depth analyses of the model outputs, including comprehensive visualizations and diagnostic plots. These graphical representations provide valuable insights into the model's performance over time, highlighting its ability to capture seasonality, trends, and other temporal patterns accurately.

In conclusion, our research findings, rooted in data-driven insights and rigorous analysis, significantly contribute to our understanding of the phenomena under investigation. Our predictive models offer a robust framework for forecasting, which can inform decision-making processes and resource allocation with a high degree of accuracy. This research project within the Google Colab environment stands as a testament to the potential of data-driven approaches in advancing research goals and underscores the importance of continued exploration and refinement in our field of study.

V. Discussion

The weather prediction project presented here leverages a dataset with date-time as the primary key and key columns including condition, humidity, and temperature. This project aims to demonstrate the potential and significance of utilizing the AutoRegressive Integrated Moving Average (ARIMA) model for weather forecasting and analysis. The discussion will delve into various aspects of the project, including data exploration, model development, and the implications of the results.

Data Exploration: Understanding the Dataset

The dataset's primary key, date-time, serves as the temporal anchor for our analysis. It allows us to explore how weather conditions, humidity levels, and temperatures change over time. This exploration begins with data visualization, where time series plots reveal trends, seasonality, and potential outliers. By examining these temporal patterns, we gain valuable insights into the dataset's characteristics.

The condition column provides categorical information about the weather conditions at each timestamp. Analyzing this column allows us to identify prevailing weather states such as clear, rainy, or cloudy. Understanding the distribution of weather conditions is essential for making accurate predictions and assessing their impact on humidity and temperature.

Humidity, as a critical factor influencing weather patterns, is another focal point of our analysis. We examine how humidity levels vary over time, seeking to identify correlations with weather conditions and temperature. Additionally, we look for any anomalies or sudden changes in humidity that may indicate specific weather events.

Temperature data, often associated with seasonality, exhibits patterns that can be visually discerned. We investigate temperature variations throughout the dataset, searching for both short-term fluctuations and long-term trends. This understanding of temperature dynamics plays a crucial role in our forecasting models.

Model Development: Leveraging ARIMA

The heart of our weather prediction project lies in the application of the ARIMA model to the dataset. ARIMA, known for its capability to capture temporal dependencies and forecast time series data, proves to be an ideal choice for predicting weather conditions, humidity, and temperature.

To deploy ARIMA effectively, we follow a systematic approach. This includes determining appropriate values for the model's parameters: Autoregressive Order (p), Integrated Order (d), and Moving Average Order (q). These parameters are crucial as they dictate how many past observations and differences are considered in making predictions.

Our model training process involves fitting ARIMA to historical data, allowing it to learn the temporal patterns present in the dataset. We employ cross-validation techniques to assess model accuracy and robustness, enabling us to make informed decisions about parameter tuning and model selection.

Implications and Applications

The results of our weather prediction project hold significant implications for various domains. Accurate weather forecasts have wide-ranging applications, including:

1.Agriculture: Farmers can make informed decisions about planting and harvesting crops based on weather predictions, optimizing yields.

2.Transportation: Airlines, shipping companies, and road planners rely on weather forecasts to ensure safe and efficient operations.

3.Disaster Management: Timely and accurate predictions are critical for preparing and responding to natural disasters such as hurricanes and floods.

4.Energy Sector: Weather influences energy production, distribution, and consumption, affecting decision-making in the energy industry.
Furthermore, our project demonstrates the potential of using ARIMA for time series forecasting beyond weather prediction. It showcases the importance of data-driven approaches in making informed decisions, optimizing resource allocation, and enhancing preparedness in various sectors.

In conclusion, this weather prediction project, centered on a dataset with date-time, condition, humidity, and temperature columns, underscores the value of data analysis and the ARIMA model in forecasting and understanding weather patterns. Through meticulous data exploration, model development, and validation, we aim to provide reliable predictions that contribute to informed decision-making across diverse industries. This project exemplifies the power of data-driven insights in addressing complex challenges and underscores the importance of continued research in the field of weather forecasting and time series analysis.

VI. Conclusions

In this project, we embarked on a journey to enhance weather forecasting using the AutoRegressive Integrated Moving Average (ARIMA) model in Python. Weather forecasting plays a pivotal role in various aspects of our lives, from daily planning to critical decision-making in agriculture, transportation, and disaster management. Through meticulous data analysis and modeling, this project aimed to improve the accuracy and reliability of weather predictions.

The journey began with the collection and preprocessing of historical weather data. Cleaning and preparing the data were crucial steps to ensure the quality of our forecasts. We tackled issues such as missing values, outliers, and noise, making the dataset ready for modeling.

The ARIMA model, renowned for its ability to capture temporal patterns in time series data, served as the cornerstone of our forecasting system. We diligently determined the model's components, including AutoRegressive (AR), Integrated (I), and Moving Average (MA), through rigorous analysis and statistical tests. This step allowed us to harness the full potential of ARIMA in capturing dependencies within the weather data.

To assess the accuracy of our forecasts, we implemented various validation techniques, including cross-validation and a range of error metrics such as Mean Absolute Error (MAE), Root Mean Square Error (RMSE), and Mean Absolute Percentage Error (MAPE). These assessments provided a comprehensive view of the model's performance and its ability to provide reliable predictions.

The culmination of our efforts resulted in the deployment of a user-friendly interface or Application Programming Interface (API) for real-time access to weather predictions. Users across different sectors can now benefit from these improved forecasts, making informed decisions and enhancing their preparedness in the face of dynamic weather conditions.

In conclusion, this project exemplifies the potential of the ARIMA model in revolutionizing weather forecasting. By combining data-driven analysis, robust modeling, and thorough validation, we have taken significant strides toward providing more accurate and timely weather predictions. This progress not only serves as a testament to the power of data science but also contributes to the well-being and resilience of communities and industries reliant on precise weather forecasts. As the field of weather forecasting continues to evolve, projects like this pave the way for a more weather-resilient future.

VII. Reference

https://www.kaggle.com/code/amar09/time-series-delhi-weather-forecasting-arima/notebo
https://www.analyticsvidhya.com/blog/2016/02/time-series-forecasting-codes-python/
https://jupyter.org/
https://pandas.pydata.org/pandas-docs/stable/index.html
https://medium.com/swlh/temperature-forecasting-with-arima-model-in-python-427b2d3bcb

Acknowledgements

We would like to express our sincere gratitude to all those who have contributed to the successful completion of this weather prediction project using the AutoRegressive Integrated Moving Average (ARIMA) model. This project has been a collaborative effort, and we are immensely thankful to the following individuals and organizations for their support, guidance, and assistance.This project would not have been possible without the collective effort of all these individuals and organizations. Their contributions have been instrumental in our pursuit of knowledge and successfully completing this weather prediction project.

Thank you all for your support.