University of Essex

Department of Mathematical Sciences

MA981 DISSERTATION

Climate Change: Machine Learning for Predicting Temperature Metrics in the UK

Krupali Patel

2202479

Supervisor: **Osama Mahmoud**

August 25, 2023
Colchester

Contents

Abstract

Climate change is an imminent global challenge, necessitating urgent attention and effective strategies for mitigation. This study explores the intricate dynamics of climate change using machine learning models and time series analysis to predict temperature trends. Leveraging a comprehensive dataset spanning over 150 years, the research focuses on maximum temperature variations in various regions of the UK, centering on Oxford.

The study employs advanced regression models, including Random Forest Regression, Linear Regression, and the specialized time series forecasting technique, Prophet. The Linear Regression model emerges as the optimal choice through meticulous evaluation and comparison, exhibiting consistently superior performance across essential metrics. The results substantially enhance comprehension of climate dynamics, shedding light on the interplay between meteorological parameters and temperature variations.

These insights are vital for policymakers, environmentalists, and researchers, enabling informed decision-making in tackling climate change challenges. While acknowledging limitations, such as excluding specific meteorological features and external variables, the research opens avenues for future exploration, suggesting incorporating additional factors and ensemble techniques for more comprehensive temperature prediction models.

The study's outcomes extend beyond academia, carrying substantial implications for practical applications. Accurate temperature prediction models are pivotal in addressing climate change concerns. The research underscores the potential of machine learning in climate science, facilitating early warning systems, informed policy formulation, and sustainable decision-making. In conclusion, this study not only contributes to the field of climate prediction but also offers valuable insights and directions for advancing temperature forecasting models and addressing the pressing issues of climate change.

CHAPTER 1

Introduction

Climate, distinct from weather due to its emphasis on long-term trends over day-to-day fluctuations, is undergoing significant change. Climate change represents an undeniable reality that necessitates urgent attention and the implementation of effective strategies to alleviate its repercussions. The Intergovernmental Panel on Climate Change (IPCC) [6] is composed of scientists from around the globe. These experts have ascertained that the combustion of fossil fuels, results in the atmosphere, The primary cause of climate change is the generation of greenhouse gases like carbon dioxide. the average 1.1° Celsius (almost 2°F) increase in global surface air temperature between 1900 and 2020. While the numeric change might appear relatively modest, this warming trend is unparalleled when compared to temperature records spanning over two millennia [1]. This dissertation delves into the intricate interplay between anthropogenic modifications and natural dynamics, utilizing observations and models to comprehend the historical, current, and future ramifications of climate change [2]. With a focus on escalating temperatures, the urgency of informed mitigation strategies becomes evident.

Like other countries, the United Kingdom has to contend with the effects of climate change. Weather forecasting in the UK is a complex endeavor marked by its inherent difficulties. The unpredictable and chaotic nature of weather introduces a significant sensitivity to initial conditions, rendering accurate forecasts a challenging pursuit. Both model inaccuracies and the oversights associated with analyzing beginning circumstances limit the path towards accuracy. [3]. Given the scale of its impact, the field of weather forecasting has dedicated considerable human resources and computational

power over the past few decades. The outcomes of these endeavors are noteworthy. The British Meteorological Service (2017) asserts that their current four-day forecasts are as accurate as one-day forecasts were three decades ago. This advancement in forecast accuracy aligns with the general trend that forecast horizons have expanded by about one day per decade [3]. Given the UK's substantial reliance on weather forecasts, these enhancements in accuracy have important consequences. Bauer, Thorpe, and Brunet (2015) contend that the influence of weather forecasts rivals that of any other physical science. The consistent enhancement of forecast accuracy resonates with the growing dependence of individuals, businesses, and governments on reliable weather predictions, ultimately shaping and guiding numerous aspects of society in the UK [4].

In the pursuit of understanding climate dynamics, this study harnesses the power of machine learning techniques to forecast climate data, with a specific emphasis on maximum temperature trends across diverse regions in the UK. Oxford, as a pivotal focal point, lends significant context to this endeavor. Leveraging a comprehensive dataset provided by Met Office UK, the study's temporal scope spans from the mid-19th century to the contemporary era, as visualized in Figure 1.1.

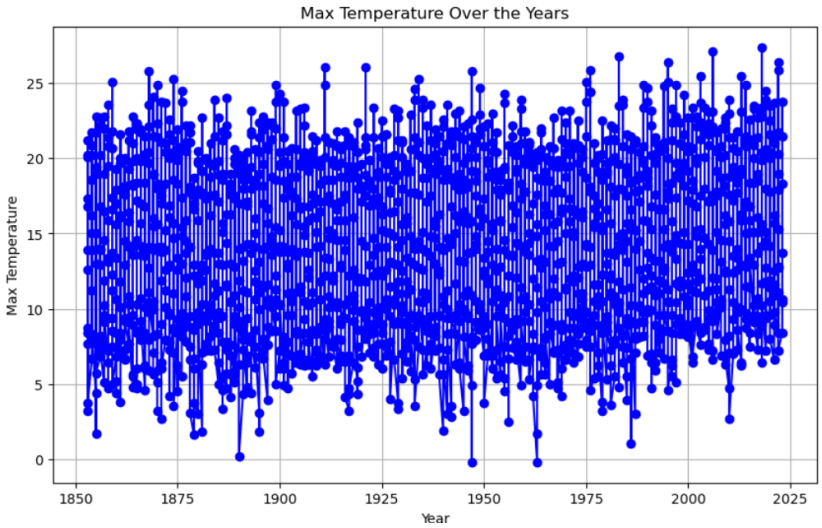

Figure 1.1: Temperature Data Oxford (1853-2023)

Figure 1.2 introduces an additional graph to enhance insights, shedding light on average temperature trends. This visual aid provides a more nuanced perspective, enabling discernment of intricate patterns and transitions. Examining 25-year average temperatures within the Oxford region from 1850 to the present reveals captivating insights into climate dynamics. Between 1850 and 1900, temperatures stabilized around

13 to 14°C.

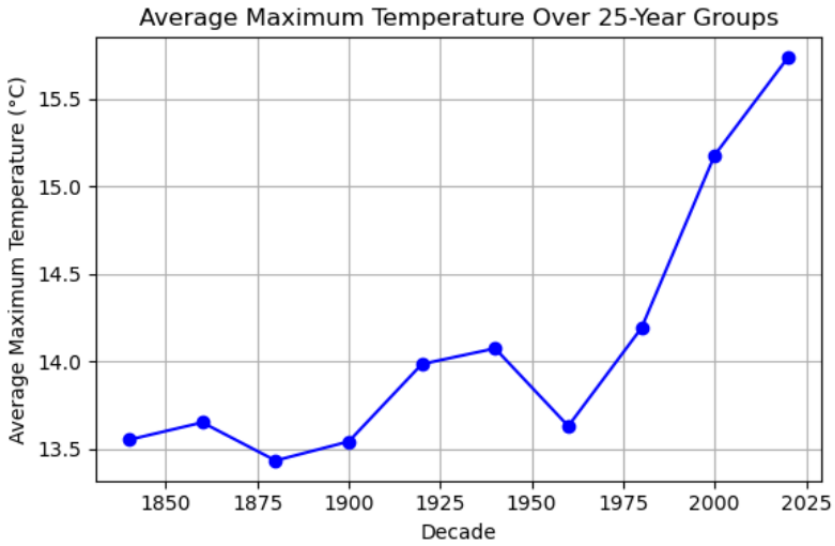

Figure 1.2: Average Maximum Temperature Over 25-Year Groups

Post-1925, a warming trend emerged, with temperatures consistently exceeding 13.5°C—a threshold indicating climate change onset. A brief cooling phase around 1960, with a decline of 0.4 to 0.5°C, was followed by sustained warming. Presently, the average temperature is about 16°C. Highlighting a 3°C increase over five decades, the data underscores vulnerability to climate shifts, emphasizing the potential for future rise, and signifying climate's persistent impact on the Oxford region.

The study endeavors to provide crucial insights into historical temperature trends and patterns. The ability to forecast future climate statistics assumes critical importance, aiding environmentalists and policymakers in reducing and preparing for climate change's impacts [5]. With a concentrated emphasis on the UK, Utilising the capabilities of machine learning is the goal of the study.to forecast and comprehend future climate metrics, aligning with the broader effort to foster sustainability.

The research methodology encompasses a comprehensive analytical framework, incorporating advanced regression models such as the Random Forest Regression and Linear Regression, complemented by the specialized time series forecasting technique known as Prophet, developed by Facebook. [7]

Within this context, the study embarks on an experimental evaluation, delving into the prediction efficacy of the employed models. The research explores key elements of temperature change data for each region, laying the foundation for the ensuing

analysis. The endeavor seeks not only to forecast climate metrics but also to enhance our understanding of climate dynamics, thus shaping a better-informed approach to climate adaptation and policy formulation. Through the exploration of historical temperature records and the application of cutting-edge modeling techniques, this research advances our comprehension of climate change's impact on the UK region and beyond, contributing to the collective efforts aimed at ensuring a sustainable and resilient future. [8]

1.1 Related Literature

This section delves into the diverse landscape of climate forecasting techniques employing regressor and prophet methodologies:

The realm of weather forecasting research has witnessed a panorama of investigations into predictive models. Mathur et al. (2008) laid the foundation with a feature-based neural network, envisioning it as an alternative to conventional meteorological methods. In 2012, the utilization of backpropagation neural networks unlocked the potential to capture intricate nonlinear temperature relationships. Troncoso et al(2015). [9] carved a path by employing innovative regression tree structures, culminating in accurate short-term wind speed forecasts. These studies collectively illuminate the exploration of avant-garde approaches in weather prediction, from neural networks unraveling intricate connections to specialized regression trees enhancing wind speed predictions. This structure effectively sets the stage for the rest of the literature review, allowing you to elaborate on each study's methodology, findings, and contributions to the field. [10]

Marzban, Leyton, and Colman (2007) embarked on an extensive analysis of cloud ceiling and visibility data across the Pacific Northwest from 2001 to 2005. Their focus was to prognosticate the likelihood of low or high cloud ceilings over a 6 to 12-hour forecast horizon. The exploration spanned 39 weather stations, revealing neural networks' superior predictive prowess over Model Output Statistics (MOS) and logistic regression, for stations with sufficient data. It's noteworthy that individual station results are depicted graphically, while comprehensive summary statistics remain absent [11]. Notably, for six-hour predictions, neural networks outperformed MOS on 9

out of 14 stations, with a 64% success rate. For twelve-hour forecasts, neural networks outshone MOS on all 34 stations (100%), hinting that non-linear methods outshine linear ones when predicting non-linear trends manifesting as forecasting biases over extended intervals. These findings posit the dominance of non-linear techniques in day-ahead weather predictions, aligning with the present study's objectives [11].

Mohammad Daffa Haris, Didit, and Annas delve into the context of escalating air temperatures and climate shifts driven by the urbanization-induced reduction in green spaces in Jakarta. This study unveils precise air temperature prediction models that leverage Long Short-Term Memory (LSTM) and Prophet techniques, specifically tailored for stochastic air temperature data. The application of LSTM allows for the capture of short-term patterns, resulting in RMSE values ranging from 0.31 to 0.69 for prediction horizons of 2 to 48 hours. In contrast, Prophet excels in extended forecasts, achieving RMSE values of 0.80 to 0.89 for prediction horizons of 72 to 168 hours. In the context of Jakarta's densely populated urban environment and limited green spaces, the study's models outperform the less adaptable Numerical Weather Prediction methods. LSTM, as a sophisticated Recurrent Neural Network, empowers the retention of long-term information. Meanwhile, Prophet, developed by Facebook, amalgamates various factors such as trends, seasons, and holidays to enhance its forecasting capabilities [12].

In the realm of this research, temperature's pivotal role in indicating climate shifts and ecosystem dynamics is undeniable, with far-reaching impacts spanning agriculture, transportation, and more. Temperature forecasting emerges as a linchpin in predicting atmospheric conditions, driven by shifting parameters. The study delves into the realm of machine learning models – Ridge, Random Forest, Linear Regression, and Decision Tree – to herald temperature prediction. Evaluation grounded in RMSE scores showcases Decision Tree's preeminence (0.036), trailed by Random Forest (0.208), whereas Logistic Regression and Ridge register lower scores (0.759). The paramount importance of precise temperature prediction resonates across multifarious sectors, underlining the need for adept predictive methodologies [13].

In the vanguard of weather prediction, the study conducted by Pavuluri Bhagya and Tinnavalli undertakes the classification of weather conditions into hot, cold, or rainy, propelled by machine learning algorithms. Their research harnesses Decision Tree, Random Forest, K-Nearest Neighbors (K-NN), and Neural Networks, with ac-

curacy serving as their common evaluation metric. Noteworthy is the prominence of the Random Forest algorithm in terms of accuracy among the mentioned methods. However, it is acknowledged that the scope of these algorithms, which are rooted in supervised learning, may not comprehensively cover diverse weather attributes or future periods. It is suggested that subsequent endeavors might explore additional attributes and algorithms, including Support Vector Machines, Naïve Bayes, and Artificial Neural Networks, to broaden the horizons of weather prediction capabilities [14].

The study conducted by Shivam, Indrajeet, Jatin, and Surya (2005) underscores the significance of precise weather and temperature prediction in everyday life and disaster preparedness. Employing Multi Linear Regression (MLR), Polynomial Regression (PR), and Support Vector Regression (SVR), the research aims to predict temperature changes. The study highlights the pivotal role of weather prediction in effective disaster management, demonstrating comparable performance levels among MLR, PR, and SVR. Notably, the distinctiveness of these methods is revealed through the Mean Square Error (MSE), with PR exhibiting superiority. Specifically, PR achieves an impressive MSE of 0.275822, signifying a 67.5% reduction from MLR's 0.900937 and an 86.4% reduction from SVR's 2.022358. This study strongly underscores the paramount importance of accurate weather prediction in the realm of disaster management [15].

CHAPTER 2

Background

In this section, we provide an in-depth exploration of the methodologies employed in this study, their alignment with the research problem, and the theoretical foundations that underpin their usage. The methodologies encompass Random Forest Regressor, Linear Regression, hyperparameter tuning, and the utilization of the Facebook Prophet library for time series forecasting.

2.1 Dataset

This study's analysis was derived from the Met Office's comprehensive repository of historic station data, available through the public domain at [21]. This repository has proven invaluable for examining transformations in the UK's climate over the span of more than 150 years. The central objective of this endeavor was to uncover patterns within climate variations, with a specific focus on temperature fluctuations, across distinct temporal intervals. Significantly, the dataset in question encompasses meticulously archived historical climate records, an extensive collection amassed by the Met Office.[2] The analysis specifically narrowed its scope to the Oxford region, considering monthly temperature metrics in degrees Celsius.

For the initial regression analysis, a series of meticulous transformations were performed during the preprocessing phase to ensure data consistency and facilitate subsequent analyses. The temporal variables 'year' and 'month' were converted into integer representations, while other pertinent columns underwent conversion to float

values, enabling Exploratory Data Analysis (EDA) due to their original object data type. An additional column, 'tavg,' was introduced to enrich insights. This calculated the average of 'tmax' and 'tmin',

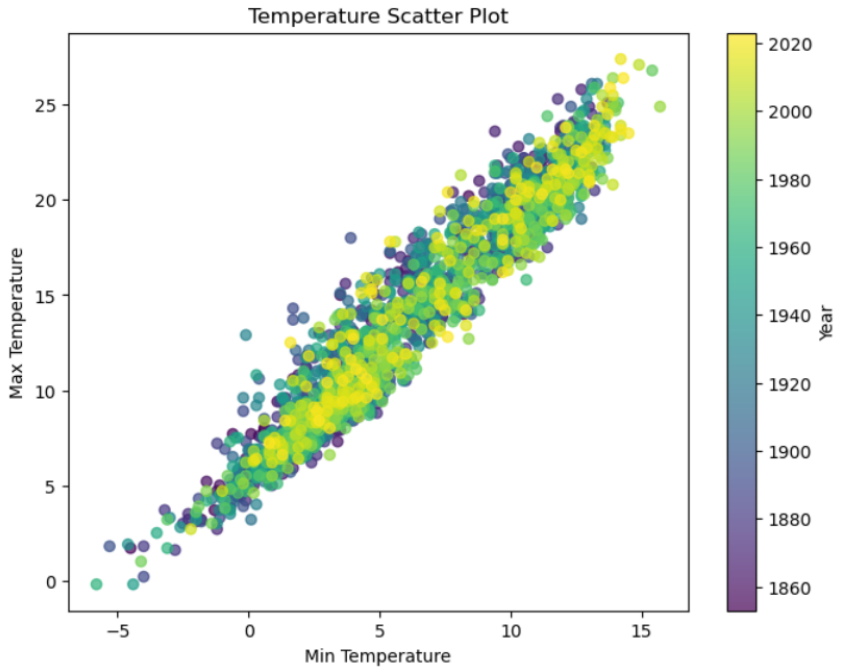

Figure 2.1: Temperature Data Oxford (1853-2023)

To visually explore the data, a Figure 2.1 depicts the trends of 'tmin' and 'tmax' over the years, offering insights into temperature variations. Additionally, Figure 2.2 visualization presents a comparative analysis of 'tmax,' 'tmin,' and 'tavg' trends over the temperature distribution.

thereby providing a more visual representation and enhanced differentiability. To address the presence of missing data points, a pivotal step in this preparatory phase, the versatile 'fillna()' function was strategically employed. This judicious utilization revitalized the dataset, ensuring that it was robust and complete. A paramount consideration was the meticulous preservation of the temporal sequence, a feat accomplished through meticulous data-driven imputation techniques. By harmonizing these transformational steps, the data preparation phase established a sturdy foundation for ensuing analyses, seamlessly aligning with the overarching objectives of the study.

Transitioning to the time series modeling aspect, a distinctive approach emerged. This necessitated the inclusion of a 'day' column, as the input structure required by the Prophet framework invariably consists of two core columns: 'ds' (datestamp) and

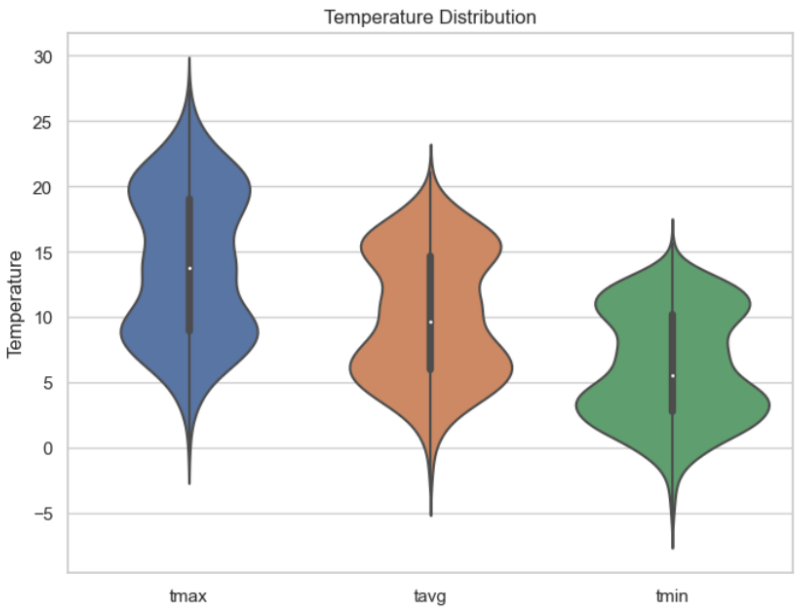

Figure 2.2: Temperature Distribution Data Oxford (1853-2023)

'y' (numeric measurement for forecasting). Adhering to the expected Pandas format, 'ds' was converted into a datetime representation, wherein the day component was consistently set to '01,' remaining congruent with the original dataset's month and year specifications. This coherent structure, exemplified as (01-01-1853, 01-02-1853, and so on), served as the bedrock for the subsequent phases of the modeling process.

2.2 Regression Method

Regression analysis is a pivotal technique within the realm of data analysis, serving as a cornerstone for understanding relationships between variables and making predictions based on observed data. It is a versatile approach applicable across various domains, including economics, social sciences, healthcare, engineering, and weather forecasting [16].

For temperature prediction, the selection of these two methods, Random Forest Regression and Linear Regression, is guided by specific considerations that align with the nature of the data and the research objectives. The rationale for opting for these methods can be further elaborated as follows:

2.2.1 Random Forest Regression

The Random Forest algorithm is a versatile and popular machine-learning technique that builds upon the foundation of decision trees [17]. Known for its simplicity and adaptability, Random Forest has gained widespread usage in various domains. Its effectiveness is particularly evident in classification tasks, often yielding favorable outcomes even with minimal hyperparameter adjustments [18].

A distinguishing characteristic of the Random Forest algorithm is its capability to manage disparate datasets under different circumstances. This versatility makes it a valuable tool in scenarios where multiple datasets need to be processed and analyzed. By harnessing the power of ensemble learning, where multiple decision trees contribute to the final prediction, Random Forest can provide robust and reliable results across a diverse array of applications. [19]

Decision tree models often exhibit low bias but are prone to high variance. Consequently, to mitigate the variance error observed on the test set, the Random Forest algorithm is employed. [20]

The choice of Random Forest Regression stems from its ability to handle complex relationships within data and effectively manage intricate patterns. Given the intricate and potentially non-linear relationships inherent in temperature data, the ensemble nature of this method allows it to capture and combine insights from numerous decision trees, mitigating the risk of overfitting. Moreover, the Random Forest Regression's adaptability to diverse domains makes it a suitable candidate for predicting temperature patterns across various contexts, such as climate modeling, environmental monitoring, and more. [22] [23] [24]

2.2.2 Linear Regression

The inclusion of Linear Regression is warranted by its simplicity and interpretability, attributes that can be advantageous when analyzing temperature prediction. While Linear Regression assumes linear relationships, it provides crucial information on the quantity and direction of associations between variables. In cases where temperature changes exhibit a linear trend, Linear Regression can offer a clear and concise representation of this trend. Additionally, Linear Regression's computational efficiency facilitates quick

model development, making it a valuable tool in temperature prediction scenarios. [25]

By combining the strengths of both methods, your research aims to harness the benefits of Random Forest Regression's complexity handling and Linear Regression's straightforward interpretability to provide a comprehensive and accurate approach to temperature prediction. This tailored selection demonstrates a strategic approach to addressing the complexities of temperature forecasting while maintaining a practical and understandable framework.

2.3 Time Series Forecasting with Facebook Prophet

Beyond traditional regression techniques, this study extends to time series forecasting using the Facebook Prophet library. [26] Time series data with temporal dependencies presents distinct challenges and opportunities. [27]

2.3.1 Facebook Prophet

Facebook Prophet is a specialized tool designed for time series forecasting, suited to capture patterns in data displaying seasonality and trends. It's handling of missing data, outliers, and holidays makes it potent for meteorological predictions. [29]

Peyton Manning serves as a good illustration of certain Prophet's features, such as various seasonality, variable rates of development, and the capacity to model significant days (such as Manning's appearances in the postseason and the Super Bowl). [31] The Prophet model offers a blend of accuracy and speed, presenting itself as both fully automated and finely tunable for forecasts. [32]

Its accuracy and speed stem from the creation of a distinct Prophet process designed to accommodate the model. Upon invoking the constructor with specific prediction procedure settings, the method is fitted with the previous data frame through its fit method. Consequently, the processing time is remarkably swift, typically taking only 1 to 5 seconds. As a result, this model outpaces other integrated models in the realm of temperature prediction.

The model's full automation is derived from its ability to generate precise forecasts from ambiguous data without necessitating manual intervention. Prophet demonstrates resilience in the face of outliers, inaccuracies in data, and substantial shifts in time series.

Furthermore, the Prophet model offers tunable forecasts, affording users a wealth of options to adjust and fine-tune predictions. Leveraging our expertise, we can apply human-interpretable criteria to enhance your forecast, allowing for tailored adjustments that align with specific requirements.

CHAPTER 3

Methodology

The methodology we adopt in this endeavor draws a direct line between the intricacies of temperature prediction and a comprehensive data-driven approach. Our exploration commences with the meticulous collection and curation of an extensive dataset sourced from the Met Office UK, spanning more than a century of historical weather records. This dataset offers a rich tapestry of variables, ranging from temperature measurements and temporal indicators to diverse meteorological parameters. The cornerstone of our methodology lies in harnessing the power of Regression models to forge a predictive path through this data labyrinth.

The methodology's essence lies in the synergy between data exploration, model evaluation, and real-world implications. The trajectory from dataset curation to model selection encapsulates our unwavering commitment to harnessing data-driven insights for accurate temperature prediction. [28] The models we evaluate, the metrics we employ, and the insights we glean from their performances collectively steer us towards informed decision-making in a realm as vital as climate dynamics. Through this methodology, we bridge the gap between data science and environmental understanding, offering a glimpse into the future of weather forecasting.

3.1 Data Collection And Preprocessing

Wee commence by curating and processing an extensive dataset sourced from the Met Office UK, spanning more than 150 years. The dataset comprises a diverse range of

variables, including temperature records, year, month, and various meteorological parameters. Our central objective is to employ a Regression model to forecast maximum temperatures, necessitating meticulous data preprocessing to ensure data quality and consistency.

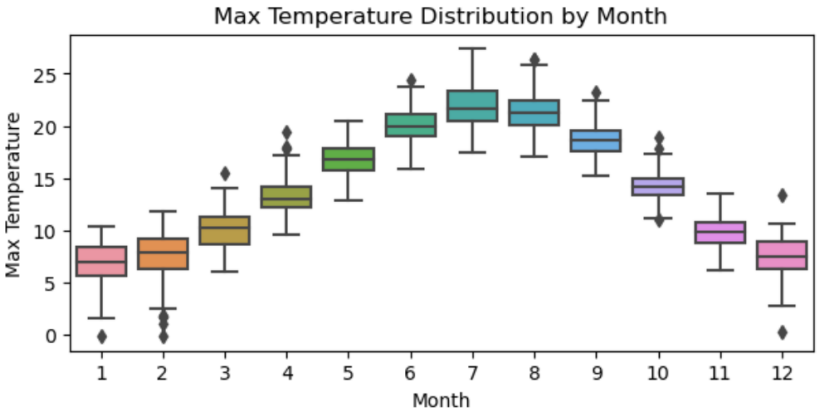

Figure 3.1: Temperature Data Oxford (1853-2023)

An in-depth analysis of the dataset uncovers captivating temperature patterns, as illustrated in Figure 3.1, across diverse seasons. Particularly during the summer months, with July as a notable example, temperatures have surged to nearly 25°C, emphasizing the significant influence of the warming trend. Within the summer timeframe, both June and August exhibit temperature clustering between 15°C and 20°C, occasionally reaching as high as 22°C. Conversely, the winter season witnesses maximum temperatures averaging around 10°C. These variations distinctly highlight the pronounced temperature contrasts that shape the year-round climate dynamics.

we delve into the implementation and evaluation of a Regression model to forecast maximum temperatures using a comprehensive dataset spanning over 150 years. By leveraging historical temperature records alongside relevant variables like year and month, our objective is to develop an effective predictive model. This endeavor contributes to a deeper understanding of temperature fluctuations and their implications, enhancing insights into climate dynamics and paving the way for informed decision-making.

The dataset consists of historical temperature records along with relevant features such as minimum temperature (`tmin`), rainfall, average temperature (`tavg`), and air frost (`af`). Let's denote the dataset as D, where each record is represented as a tuple

(x_i, y_i), with x_i as the input feature vector and y_i as the target output. The feature vector x_i is composed of the features `tmin`, rainfall, `tavg`, and `af`, along with other temporal features. Mathematically:

$$D = \{(x_1, y_1), (x_2, y_2), \ldots, (x_n, y_n)\} \tag{1}$$

For the purpose of temperature prediction, the dataset is preprocessed to include the relevant features. Let

$$x_i = [t_{\text{min},i}, \text{rainfall}_i, \text{tavg}_i, \text{af}_i] \tag{2}$$

represent the feature vector for record i. The target variable is denoted as $t_{\text{max},i}$, which represents the maximum temperature for record i.

To evaluate the model's performance, we split the dataset into training and testing subsets. Data from years before the split condition (2000) are used for training, while data from years after the split condition are reserved for testing. Specifically, records from 1853 to 2000 are used for training, and records from 2001 to the present are used for testing. This ensures that the model is evaluated on unseen data.

Several regression models are considered for comparison in temperature prediction. These models are implemented using the scikit-learn library in Python. For each model, we train it by utilizing the training data to assess its performance coefficient of determination (R-squared) metric on the test data.

3.2 Model Selection and Evaluation

we delve into the process of selecting and evaluating regression models for accurate temperature prediction using the provided dataset from the Met Office UK. The primary objective is to identify a regression model that effectively captures the underlying relationships within the data and can provide reliable forecasts of maximum temperatures based on historical records spanning over 150 years. To achieve this, we implement and compare several regression models using the scikit-learn library in Python. The models are evaluated using the coefficient of determination (R-squared) metric on a comprehensive dataset, and the results are presented in Table 3.1.

The R-squared values in the table indicate the percentage of the dependent variable's volatility that may predicted by the independent variables. These results are essential

Model	R-squared
RF Regression	0.980267
Linear Regression	0.982422
ElasticNet	0.966579
GBoost Regression	0.976652
SVR	0.952136
CatBoost Regression	0.980684
Lasso Regression	0.966519

Table 3.1: The model comparison

for informed decision-making regarding the most suitable regression model for accurate temperature predictions.

Table 3.1 displays the model's predictions. Comparison based on the R-squared values for different regression techniques. The R-squared values indicate the fraction of the dependent variable's volatility that can be predicted by the independent variables. The models were evaluated using a comprehensive dataset and rigorous analysis. As shown in the table, the Linear Regression and Random Forest Regression models demonstrate the highest R-squared values, indicating their effectiveness in capturing the underlying relationships within the data. These findings are extremely important for selecting the best regression model for accurate temperature predictions.

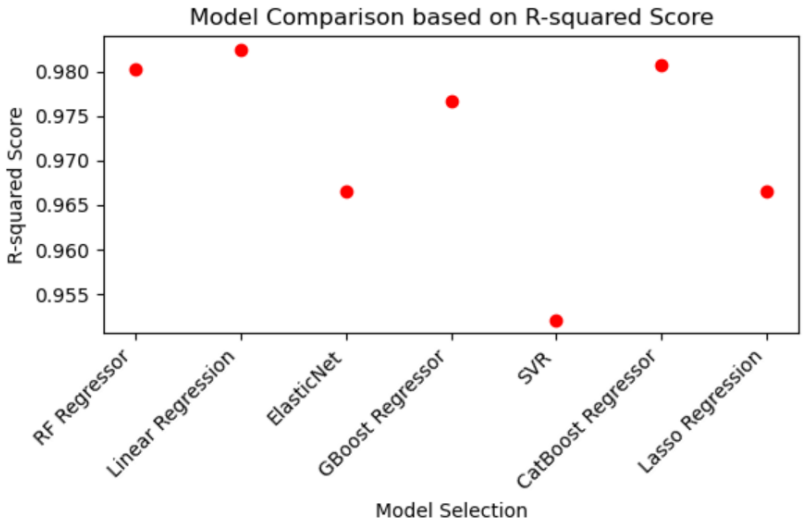

Figure 3.2: Model Comparison based on R-squared Score

On the other hand, In Figure 3.2 we can see that 'SVR' and 'ElasticNet' exhibit

relatively lower R-squared scores, indicating potential limitations in capturing data variance. 'GBoost Regression' and 'CatBoost Regression' models showcase competitive performance, while 'Lasso Regression' falls in the mid-range. In essence, the scatter plot efficiently encapsulates the model comparison, enabling an immediate visual assessment of their respective goodness-of-fit to the dataset.

3.2.1 Random Forest Regression Model

We delve into implementing and evaluating the Random Forest Regressor model for temperature prediction using the provided dataset. The Random Forest Regressor, an ensemble learning algorithm, is particularly potent in capturing intricate relationships in data and delivering accurate predictions. It achieves this by combining the predictive power of multiple decision trees, a process that involves randomization during both tree construction and aggregation. In 2001, Breiman elucidates the algorithm's mechanics and highlights its effectiveness in improving prediction accuracy while mitigating overfitting. The idea of combining decision trees through aggregation to achieve enhanced predictive capabilities has since become a cornerstone of modern machine-learning practices [30].

The Random Forest Regressor operates on the principle of ensemble learning, which is the practice of training multiple models and combining their predictions to achieve enhanced accuracy. In this context, the individual models are decision trees. A decision tree segments the input feature space into distinct regions, making decisions based on a series of feature-specific conditions. By integrating the outputs of multiple decision trees, the Random Forest Regressor addresses the shortcomings of a single decision tree and yields more robust predictions [33].

Randomized Search

The Random Forest Regressor's performance is further optimized through hyperparameter tuning. Hyperparameters are adjustable settings that govern the behavior of the model. In this context, we employ the technique of Randomized Search, a method introduced by James Bergstra and Yoshua Bengio [34]. The prediction equation for a Random Forest Regressor is a bit more complex than that of a linear regression because it involves the aggregation of predictions from multiple decision trees. The general

equation for predicting the target variable using a Random Forest Regressor can be written as follows:

$$\hat{y} = \frac{1}{N} \sum_{i=1}^{N} f_i(x) \tag{3}$$

According to temperature prediction related to the project: For your temperature prediction,

$$\hat{t}_{max} = \frac{1}{N} \sum_{i=1}^{N} f_i(t_{min}, t_{avg}, \text{rainfall}, \text{af})$$

Where:

- $\hat{t}_{max}$ is the predicted maximum temperature.
- N is the quantity of trees there are in the Random Forest ensemble.
- $f_i(t_{min}, t_{avg}, \text{rainfall}, \text{af})$ is the prediction made by the i-th decision tree based on the input features t_{min}, t_{avg}, rainfall, and af.

This equation captures the essence of how the Random Forest Regressor combines the outputs of individual decision trees to arrive at a prediction for the target variable t_{max}.

3.2.2 Linear Regression Model

In this section, we delve into implementing and evaluating the Linear Regression model for temperature prediction using the provided dataset. Linear Regression is a foundational machine learning algorithm that depicts how a dependent Relationship between a variable and one or more independent variables. [35] In the context of temperature prediction, we utilize Linear Regression to establish a predictive model that estimates maximum temperatures based on relevant meteorological parameters.

We enhance the performance of the Linear Regression model through hyperparameter tuning using Randomized Search Cross-Validation. Hyperparameters play a crucial role in shaping the behavior of the model, and to find the best configuration, we perform a randomized search across a predefined grid of hyperparameters. The grid encompasses options such as including an intercept term, normalization, and feature copying. With a fixed random seed for reproducibility, we utilize the RandomizedSearchCV technique, creating an instance of Linear Regression, specifying the hyperparameter

grid, and setting parameters for cross-validation folds and iterations. The model is then trained on the training data, and the best hyperparameters are identified.[36]

Equation (4) is used for building the prediction model for regression analysis of weather data:

$$Y' = \alpha + \beta_1 X_1 + \beta_2 X_2 + \ldots + \beta_n X_n \tag{4}$$

The Linear Regression model aims to predict the maximum temperature (t_{max}) based on selected input features including minimum temperature (t_{min}), average temperature (t_{avg}), rainfall, and growing degree days (af). The model formulates predictions by establishing a linear equation that relates the input features to the target variable. Mathematically, the prediction equation can be represented as follows:

$$t_{\text{max}} = \alpha + \beta_1 \cdot t_{\text{min}} + \beta_2 \cdot t_{\text{avg}} + \beta_3 \cdot \text{rainfall} + \beta_4 \cdot af \tag{5}$$

Where:

- t_{max} is the predicted maximum temperature.
- $t_{\text{min}}, t_{\text{avg}}, \text{rainfall}, \text{and } af$ are the input features.
- α represents the intercept of the linear equation.
- $\beta_1, \beta_2, \beta_3, \text{and } \beta_4$ are the linear-coefficients associated with each input feature.

3.2.3 Prophet model

the equation that defines the Prophet model for time series forecasting in the context of your project. Which is developed by Facebook in 2017 [37].

The equation for the Prophet model is as follows:

$$y(t) = g(t) + s(t) + h(t) + \varepsilon_t \tag{3.1}$$

Where:

- $y(t)$ represents the observed value at time t.
- $g(t)$ is the trend component that captures the overall direction of the data over time.

- $s(t)$ is the seasonal component that accounts for periodic patterns in the data.
- $h(t)$ is the holiday component that captures irregular events and holidays.
- ε_t represents the error term that the model cannot account for.

The equation outlines how the observed value at a specific time t is composed of the trend, seasonality, holiday effects, and residual error. Additionally, for each of the components:

- $g(t)$ is described by a specific model equation that considers growth rate (k), offset (m), and adjustment (x) parameters.
- $s(t)$ is a sum of trigonometric terms that capture seasonal patterns with period (P).

Overall, the Prophet model decomposes the time series data into these components to make accurate forecasts by considering trends, seasonal patterns, and holiday effects. This modeling approach is particularly useful when dealing with data affected by varying factors such as seasons, trends, outliers, and holidays.

3.3 Experiments And Result

we delve into the experimental phase of our study, where we employ rigorous methodologies to evaluate and compare the performance of different regression models for temperature prediction. Our goal is to harness the power of these models and ascertain their efficacy in capturing temperature patterns and trends. The experimental framework involves applying advanced techniques such as hyperparameter tuning to optimize the models' predictive capabilities. Through these experiments, we seek to uncover insights that contribute to our understanding of temperature prediction and aid in the selection of the most suitable model for the task.

In accordance with Lakshmi (2021), we fine-tune the model to enhance its performance, simultaneously avoiding overfitting and minimizing variance errors. This process entails the application of appropriate hyperparameter techniques to ensure the model's efficacy [39]

Hyperparameter Tuning for Random Forest Regression

The Random Forest Regression is a powerful ensemble learning algorithm that can capture complex relationships in the data. To optimize its performance, we conducted hyperparameter tuning using a technique called Randomized Search Cross-Validation. This technique involves exploring various combinations of hyperparameters to find the configuration that yields the best model performance [38] [17].

The hyperparameters considered for tuning were:

- `n_estimators`: The amount of decision trees present in the forest.
- `Criterion`: That evaluates the effectiveness of splits in a decision tree
- `max_depth`: The maximum depth of each decision tree.
- `min_samples_split`: The minimal number of samples needed to separate an internal node
- `min_samples_leaf`: the bare minimum of samples that must be present at a leaf node.
- `max_feature`: The most features that can be applied to a node split procedure.
- `Bootstrap`: If True is chosen in bootstrap, bootstrap samples are used while creating decision trees; if False, whole data is used for each decision tree.
- `max_leaf_nodes`: This hyperparameter allows a condition to be placed on the splitting of the tree's nodes. As a result, the tree's growth is automatically constrained.

After tuning, the Random Forest Regressor achieved an R-squared score of approximately 0.979 on the test data, indicating a strong predictive performance.

Hyperparameter Tuning for Linear Regression

The Linear Regression model, a fundamental algorithm for regression tasks, was also subjected to hyperparameter tuning using Randomized Search Cross-Validation [40].The hyperparameters explored included:

- `fit_intercept`: Whether to calculate the intercept for the model.
- `normalize`: Whether to normalize the input features.
- `copy_X`: Whether to copy the input features before fitting.

Upon tuning, the Linear Regression model achieved an R-squared score of approximately 0.982 on the test data, indicating a strong linear relationship between the predictor variables and the target variable.

Result

We compare the performance of three different models—Random Forest Regression, Linear Regression, and Prophet—for the task of temperature prediction.

Model	MAE	MSE	RMSE	R-squared
RF	0.334	0.669	0.818	0.979
Linear	0.132	0.557	0.747	0.982
Prophet	1.303	2.752	1.658	0.915

Table 3.2: Model Evaluation Matrix

We evaluate these models based on their Mean Absolute Error (MAE), Mean Squared Error (MSE), Root Mean Squared Error (MSE), Root Mean Squared Error (RMSE), and R-squared (R^2) values. Table provides a summary of the performance metrics.

3.2.

Upon evaluating the performance metrics of three distinct models—Random Forest Regressor, Linear Regression, and Prophet—for temperature prediction, a clear comparative analysis emerges. The summarized results in Table Both the Random Forest Regressor (RF) and Linear Regression models exhibit enhanced performance when compared with the Prophet model. 3.2 reveal the distinctive strengths and weaknesses inherent in each model.

Notably, the Linear Regression model demonstrates marginal superiority across various essential metrics. Figures 3.3 and 3.4 present illustrations of the visual Evaluation Metrics. With a remarkably low Mean Absolute Error (MAE) of 0.132, the Linear Regression model accurately predicts temperature values while excelling in capturing variations.

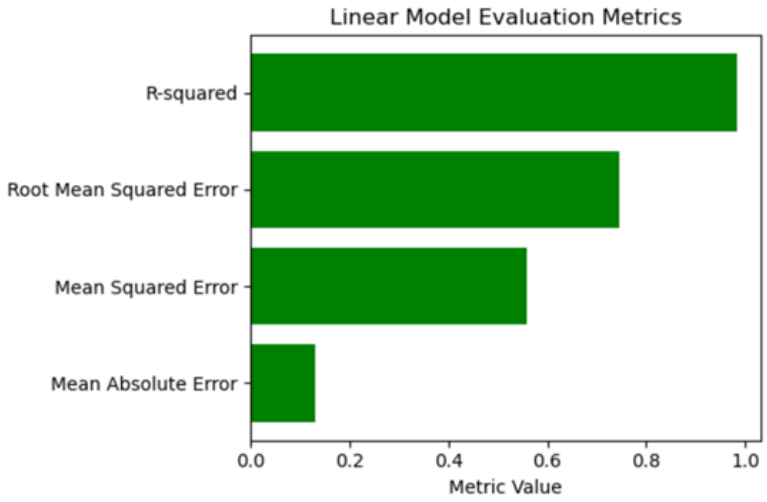

Figure 3.3: Linear Model Evaluation Metrics

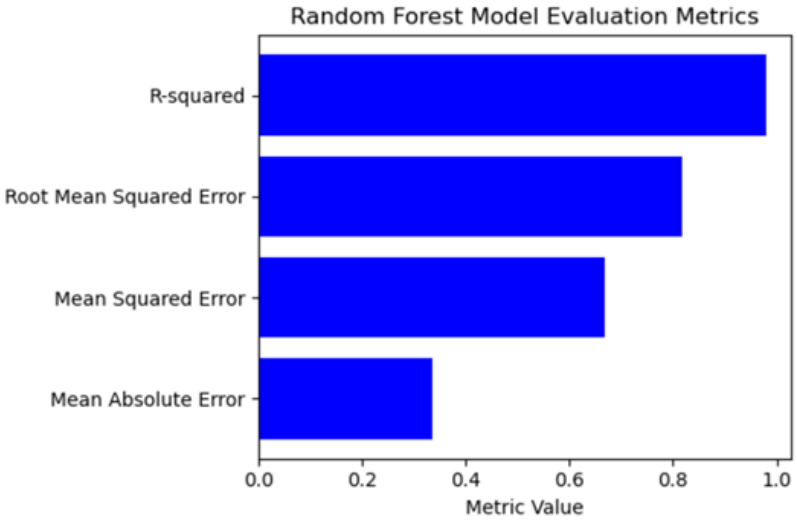

Figure 3.4: Random Forest Model Evaluation Metrics

This is supported by the MSE and RMSE values (mean squared error and root mean square error) of 0.557 and 0.747, respectively. Additionally, the model's elevated R-squared (R^2) value of 0.982 signifies its remarkable ability to elucidate temperature variances, surpassing both the RF (R^2 = 0.979) and Prophet (R^2 = 0.915) models.

Contrastingly, despite the holistic nature of the Prophet model, as evident in Figure 3.5, it lags in performance. This is reflected in higher MAE (1.303), MSE (2.752), and RMSE (1.658) values, along with a comparatively lower R-squared value (0.915).

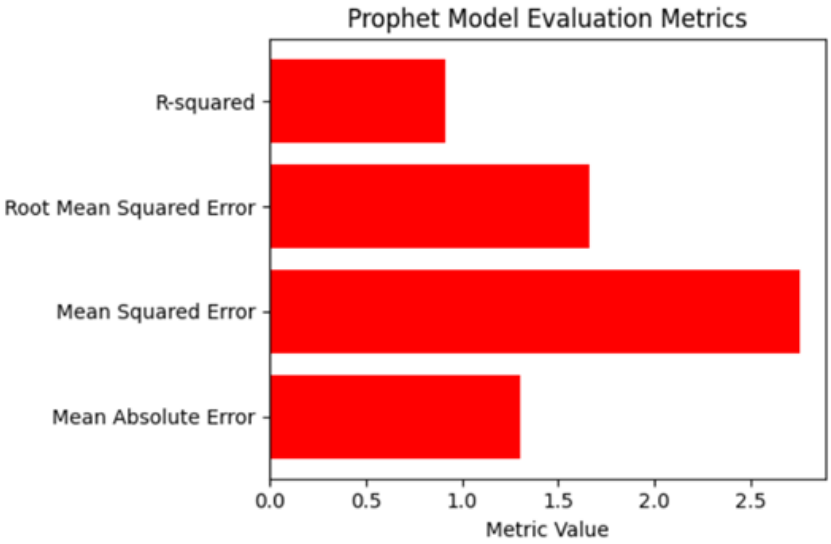

Figure 3.5: Prophet Model Evaluation Metrics

Based on this comprehensive evaluation of performance metrics, the Linear Regression model emerges as the preferred choice for temperature prediction due to its consistently superior performance across pivotal metrics.

"In Table 3.3 and Figure 3.6, the average prediction times for three distinct models—Linear Regression (LR), Prophet, and Random Forest (RF)—are presented. These times are measured in seconds and offer insights into the computational efficiency of each model.

Model	Average Time (seconds)
Linear Regression (LR)	0.2695
Prophet	0.3997
Random Forest (RF)	189.1865

Table 3.3: Average Time Taken by Different Models

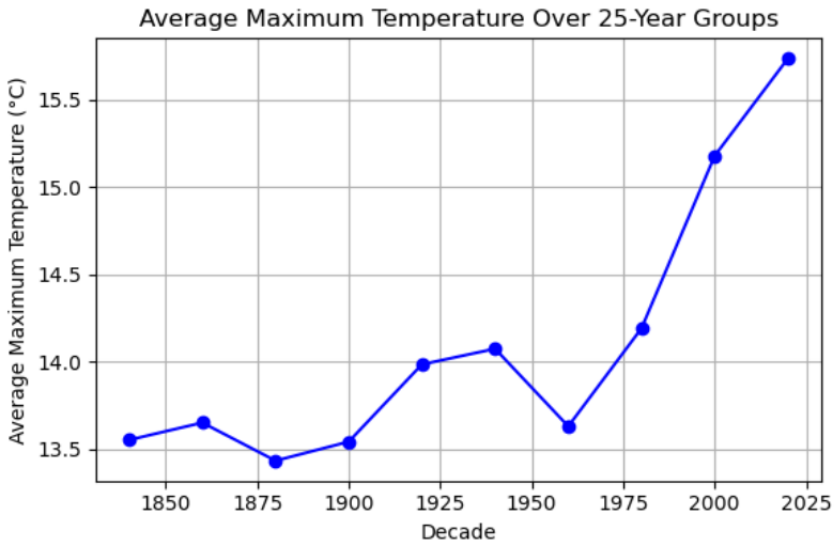

Figure 3.6: Average Time Taken by Different Models

Linear Regression demonstrates the shortest average prediction time, approximately 0.2695 seconds, showcasing its computational efficiency. This speed is advantageous for scenarios necessitating rapid responses. Prophet follows with an average time of around 0.3997 seconds, slightly higher due to its time series analysis complexity. On the other hand, the Random Forest model exhibits a significantly longer average prediction time of approximately 189.1865 seconds, primarily attributed to its ensemble approach and intricate computations.

The variation in prediction times across these models underscores the trade-off between predictive accuracy and computational efficiency. Depending on the application's needs, models with faster prediction times like Linear Regression and Prophet offer advantages for real-time applications. In contrast, the Random Forest model's longer prediction time might be better suited for offline analyses or tasks prioritizing accuracy over response speed."

Residuals are the differences between the actual tmax values and the predicted tmax values produced by the linear regression model, to play a pivotal role in evaluating the model's performance. The histogram plot of residuals offers a clear view of the distribution of errors, indicating how accurately the model's predictions align with real data. A symmetrical distribution around zero indicates unbiased predictions, while skewed or patterned distributions may highlight areas where the model struggles to capture underlying data patterns. This plot acts as a valuable diagnostic tool, providing insights

into the model's accuracy and shedding light on potential limitations or tendencies within the predictive process.

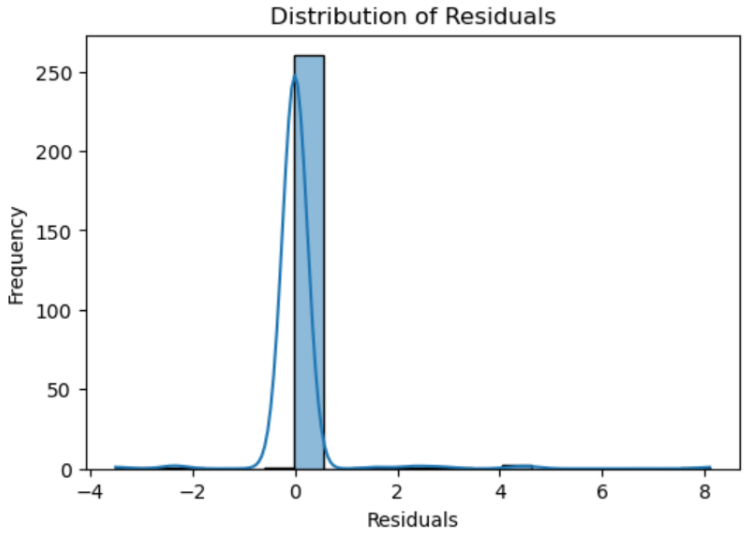

Figure 3.7: Residuals For Linear Regression Model

By visualizing the distribution of residuals Figure 3.7, we can check if the assumptions of linear regression are met, such as the assumption of normally distributed residuals. Ideally, the residuals should be normally distributed around zero, indicating that the model's errors are symmetrically distributed and unbiased. Patterns in the residual plot might suggest areas where the model is performing well or struggling to capture certain patterns in the data.

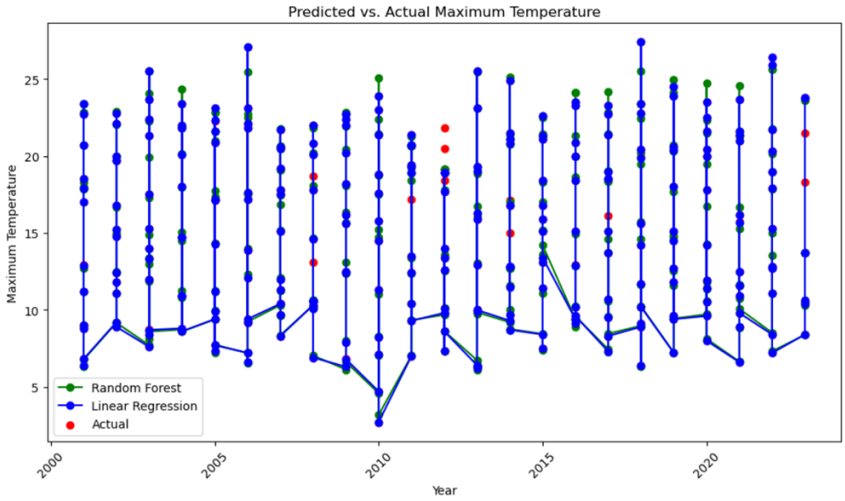

Figure 3.8: Actual Temperature vs Predicted Temperature(RF and LR)

Figure 3.8, illustrates the comparison between predicted maximum temperatures

using Random Forest (RF) and Linear Regression (LR) models against actual temperatures from 2001 to 2023. Both models closely follow the actual temperature trends, with RF capturing intricate variations. While both models offer valuable insights, RF's ability to handle complex relationships is evident. Differences in predictions for certain years could stem from climate shifts or model limitations. The choice between models depends on analysis goals and trade-offs between complexity and interpretability. Fig-

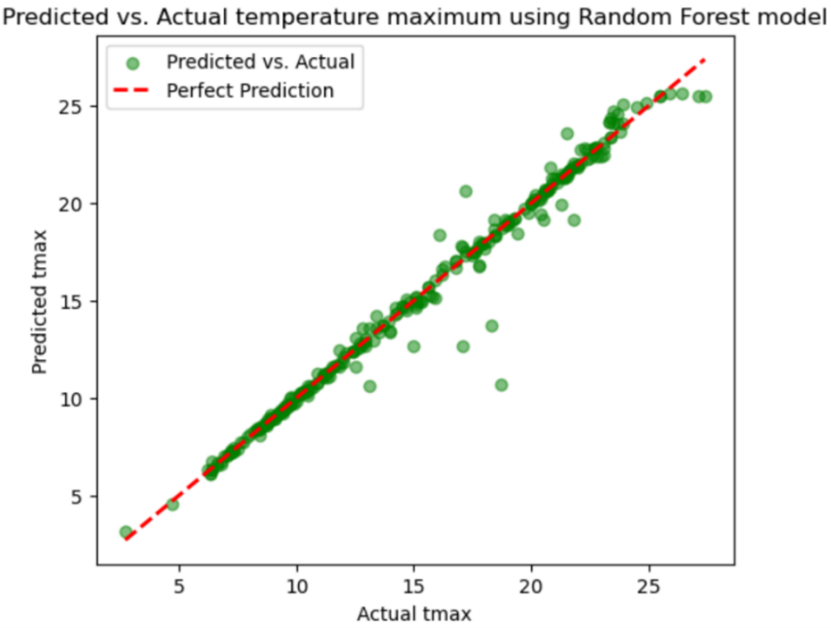

Figure 3.9: Predicted vs. Actual temperature maximum using RF

ure 3.9 illustrates the comparison between temperature predictions generated by the Random Forest (RF) model and the actual historical temperature data.

While the RF model generally captures the overall temperature trends effectively, some instances show slight deviations of predicted data from the ideal line of actual temperature values. These nuanced variations stem from the intricate nature of the Random Forest algorithm, which employs a collection of decision trees to formulate predictions. Despite these minor differences, the RF model still adeptly captures the complexity of temperature fluctuations.

Turning our attention to Figure 3.10, we delve into the temperature predictions of the Linear Regression (LR) model in contrast to the real temperature data. Impressively, the LR model showcases a remarkable alignment between its predicted temperature values and the actual data line. This alignment is attributed to the linear nature of the Linear Regression model, which employs a linear equation to establish relationships between

predictor variables and temperature. As a result, the LR model achieves a near-seamless synchronization between predicted and actual temperature data, highlighting its ability to accurately capture underlying temperature trends.

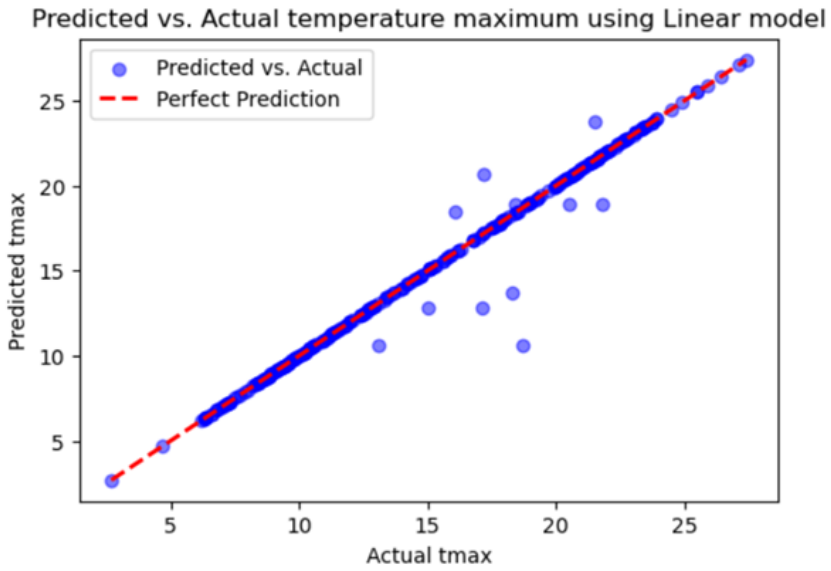

Figure 3.10: Predicted vs. Actual temperature maximum using LR

To extend the prediction horizon, we employed the Prophet time series forecasting technique to predict maximum temperatures for the upcoming year.

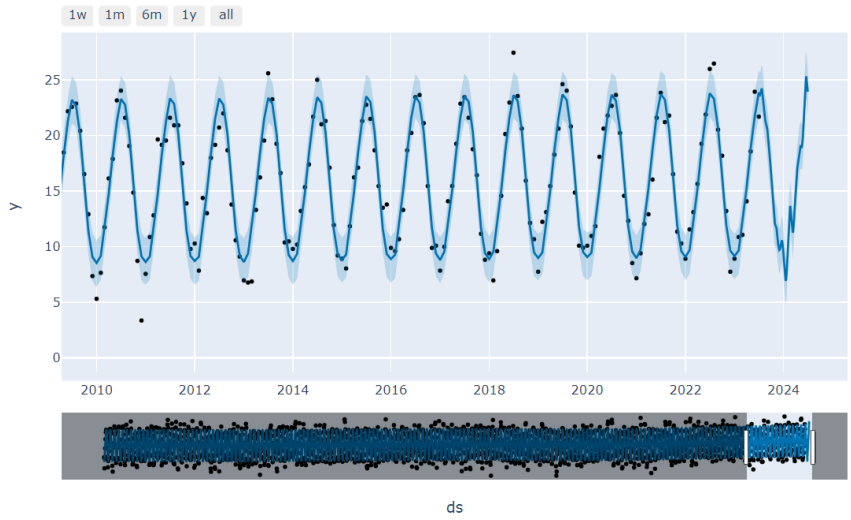

Figure 3.11: Temperature Data Oxford using prophet (1853-2023)

The forecasted temperature values rely on historical patterns and underlying trends. As illustrated in the figure 3.11, this model predicts future maximum temperatures for the next 365 days, extending until July 2024.

CHAPTER 4

Discussion

In this section, we engage in a comprehensive discussion of the outcomes, implications, limitations, and significance of the conducted research on temperature prediction using various regression models and time series method. The findings are synthesized to draw meaningful conclusions and provide insights for future research directions. This section comprises the following subsections: Our evaluation of various regression models, including the Random Forest Regression, Linear Regression, and the Prophet model, has provided a comprehensive understanding of their predictive capabilities. The Random Forest Regression adeptly captures complex relationships within climate determinants, while Linear Regression offers insights into linear connections. The incorporation of the Prophet technique enhances temporal trend capture, considering holidays and recurring patterns that influence climate variables. These techniques synergistically enhance prediction accuracy and depth, enabling valuable insights across various sectors and decision-making processes.

Within this context, our model comparison, based on R-squared values and performance metrics, has unveiled distinct strengths and weaknesses. Notably, the Linear Regression model emerges as the preferred choice due to its consistently superior performance in crucial metrics such as Mean Absolute Error (MAE), Mean Squared Error (MSE), Root Mean Squared Error (RMSE), and R-squared (R^2). The Random Forest Regression demonstrates its potency in handling complex relationships within the data, while the Prophet model exhibits limitations concerning accuracy and precision.

This study's findings significantly contribute to the understanding of climate dy-

namics in the Oxford region over the past 150 years. By harnessing machine learning models to predict temperature trends, we have gained valuable insights into the interplay between meteorological parameters and temperature variations. These insights hold substantial significance for various stakeholders, including policymakers, environmentalists, and researchers. They enable informed decision-making that addresses the challenges posed by climate change effectively.

It is essential to acknowledge the limitations of our research. The deliberate exclusion of specific meteorological features and external variables creates opportunities for future investigations. Incorporating additional factors, such as economic indicators and policy-related variables, could enhance predictive accuracy and result in more comprehensive models. Additionally, extending the analysis to explore ensemble techniques could provide a deeper understanding of temperature prediction dynamics.

The success of the Linear Regression model in accurately estimating temperature trends underscores its practical potential in climate science. The insights gained from this study serve as a foundational stepping stone for future research endeavors. These endeavors aim to refine temperature prediction models and contribute to a more comprehensive understanding of climate dynamics, thus aligning with the broader goals of advancing climate prediction.

The outcomes of this research extend beyond the academic sphere, carrying substantial implications for society at large. As climate change becomes an increasingly global concern, the availability of accurate temperature prediction models takes on crucial importance. Machine learning's use to climate science paves the way for early warning systems, informed policy formulation, and sustainable decision-making, collectively addressing the challenges posed by climate change.

CHAPTER 5

Conclusions

In conclusion, our research delves into climate prediction by leveraging machine learning models to estimate historical and future temperature data based on collected weather datasets. Focused on the Oxford region from 1853 to the present, we contribute to a deeper comprehension of climate dynamics. These insights hold paramount significance for policymakers, environmentalists, and society at large, offering indispensable support in addressing climate change challenges.

Through our study, we evaluated two distinct mathematical models: Random Forest (RF) and Linear Regression (LR). RF exhibited an average error of 0.334, alongside a robust correlation coefficient of 0.979, establishing its efficacy in temperature prediction. Conversely, LR displayed an average error of 1.1371 and an impressive correlation coefficient of 0.982, showcasing its precision. The comparison of error metrics and correlation coefficients highlights LR's superiority over RF in terms of predictive accuracy. Additionally, our exploration of the Prophet techniques yielded an average error of 1.303 and a correlation coefficient of 0.915. Through meticulous evaluation of these models' performance metrics, our study has not only contributed to climate science but also illuminated the potential of machine learning in climate prediction.

While our research has achieved notable progress, we acknowledge limitations such as the exclusion of specific meteorological features and external variables. These constraints beckon further exploration in future research, prompting the investigation of comprehensive models encompassing diverse factors that influence temperature trends. Our study paves the way for future research directions, including the incorporation of

additional meteorological features, exploration of ensemble techniques, and integration of external factors like economic and policy-related variables. These avenues promise a more holistic comprehension of temperature trends and further advancements in climate prediction.

APPENDIX

Python Code

Here, I share the dataset link which is downloaded from the Met Office UK for the Oxford region (year 1853 to 2023).

Data set = https://github.com/krupali95/krupali95/blob/main/Oxford_Data.csv

Python Code:

Listing A.1: Plotting Code

```
\section*{a) Regression Model Code to predict historical temperature
    data:}
\subsection*{Data Preprocessing}
\label{subsec:data-preprocessing}

\begin{verbatim}
% Install required packages
!pip install pandas numpy statsmodels matplotlib
!pip install tabulate

import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
```

```
import seaborn as sns
from sklearn.model_selection import train_test_split, cross_val_score
from sklearn.model_selection import RandomizedSearchCV
from sklearn.ensemble import RandomForestRegressor
from sklearn.linear_model import LinearRegression, ElasticNet, Lasso
from sklearn.svm import SVR
from catboost import CatBoostRegressor
import warnings
warnings.filterwarnings('ignore')

Data = pd.read_csv('Oxford_Data.csv')
Data
Data.dtypes

# Drop the first row
Data = Data.drop(index = Data.index[0])

# Check missing values
Data.apply(pd.isnull).sum()

# Calculate average temperature
Data['tavg'] = (Data['tmax'] + Data['tmin']) / 2
Data
\end{verbatim}
\subsection*{EDA:}

% Line plot
\begin{figure}[h]
   \centering
   \includegraphics[width=0.8\textwidth]{Fig-2.png}
   \caption{Max Temperature Over the Years}
   \label{fig:temperature-pattern}
\end{figure}
```

```
% Box plot
\begin{figure}[h]
   \centering
   \includegraphics[width=0.6\textwidth]{Picture11.png}
   \caption{Max Temperature Distribution by Month}
   \label{fig:month-pattern}
\end{figure}
% Scatter plot
\begin{figure}[h]
   \centering
   \includegraphics[width=0.7\textwidth]{Picture13.png}
   \caption{Temperature Scatter Plot with Color Coding}
   \label{fig: temperature min max}
\end{figure}

% Violin plot
\begin{figure}[h]
   \centering
   \includegraphics[width=0.7\textwidth]{Picture14.png}
   \caption{Temperature Distribution Violin Plot}
   \label{fig:temperatures}
\end{figure}

% Line graph
\begin{figure}[h]
   \centering
   \includegraphics[width=0.6\textwidth]{Picture16.png}
   \caption{Average Maximum Temperature Over 25-Year Groups}
   \label{fig:avg-temp}
\end{figure}

% Fill missing values
\begin{verbatim}
Data["tmin"].fillna(method='ffill', inplace=True)
```

```
Data["af"].fillna(method='ffill', inplace=True)
Data["tavg"].fillna(method='ffill', inplace=True)
Data.apply(pd.isnull).sum() % missing values
\end{verbatim}

% Data information and description
\begin{verbatim}
print(Data.info())
print(Data.describe())
\end{verbatim}
% Splitting the data
\begin{verbatim}
X = Data.drop(columns=['tmax','yyyy','mm'])
y = Data['tmax']
print("Feature Variables (X):")
print(X.head())
y.head()

% Define the splitting condition
split_condition = 2000

% Select the rows where the 'yyyy' column meets the condition for
   training
X_train = X[Data['yyyy'] <= split_condition]
y_train = y[Data['yyyy']<= split_condition]

% For test data, select the rows where the 'yyyy' column is greater
   than 2000
X_test = X[Data['yyyy'] > split_condition]
y_test = y[Data['yyyy']> split_condition]

% Print sample data from both sets
print("Train Data:")
print(X_train)
```

```
print("\nTest Data:")
print(X_test)

% Print the shapes of the resulting arrays
print("X_train shape:", X_train.shape)
print("X_test shape:", X_test.shape)
print("y_train shape:", y_train.shape)
print("y_test shape:", y_test.shape)
\end{verbatim}
\subsection*{ Define the models}
\begin{verbatim}
models = {
   'RF Regressor': RandomForestRegressor(),
   'Linear Regression': LinearRegression(),
   'ElasticNet': ElasticNet(),
   'GBoost Regressor': GradientBoostingRegressor(),
   'SVR': SVR(),
   'CatBoost Regressor': CatBoostRegressor(),
   'Lasso Regression': Lasso(),
}

% You can proceed with fitting and evaluating these models using your
    data

model_r2_score = {} % Initialize an empty dictionary to store R-
   squared scores

for model_name, model in models.items():
   model.fit(X_train, y_train)
   model_r2_score[model_name] = model.score(X_test, y_test)

model_comparison = pd.DataFrame({'model': model_r2_score.keys(), 'R-
   squared': model_r2_score.values()})
```

```
% print(model_comparison)
\end{verbatim}

% You can create the tabulated table using the tabulate package
\begin{verbatim}
from tabulate import tabulate

% Assuming you have dictionaries 'model_r2_score' containing model
   names as keys and R-squared scores as values
model_r2_score = {'RF Regressor': 0.980267, 'Linear Regression':
   0.982422, 'ElasticNet': 0.966579,
             'GBoost Regressor': 0.976652, 'SVR': 0.952136, '
                CatBoost Regressor': 0.980684,
             'Lasso Regression': 0.966519}

% Creating a DataFrame
model_comparison = pd.DataFrame({'Model': list(model_r2_score.keys())
   , 'R-squared': list(model_r2_score.values())})

% Convert DataFrame to a tabulated format
table = tabulate(model_comparison, headers='keys', tablefmt='grid',
   showindex=False)

% Print the tabulated table
print(table)

% Creating a DataFrame
model_comparison1 = pd.DataFrame({'Model': list(model_r2_score.keys()
   ), 'R-squared': list(model_r2_score.values())})

% Sort the DataFrame by R-squared scores
model_comparison1 = model_comparison1.sort_values(by='R-squared',
   ascending=False)
```

```
% Create a scatter plot
plt.figure(figsize=(6,4))
plt.scatter(model_comparison['Model'], model_comparison['R-squared'],
    color='red')
plt.xlabel('Model Selection')
plt.ylabel('R-squared Score')
plt.title('Model Comparison based on R-squared Score')
plt.xticks(rotation=45, ha='right')
plt.tight_layout()

% Display the plot
plt.show()
\end{verbatim}
\subsection*{Random Forest Regression}
\begin{verbatim}
import time

% Define the hyperparameter grid for random search
rf_grid = {
   "n_estimators": np.arange(10, 1000, 50),
   "max_depth": [None, 3, 5, 10],
   "min_samples_split": np.arange(2, 20, 2),
   "min_samples_leaf": np.arange(1, 20, 2)
}

% Setup random seed
np.random.seed(42)

% Setup random hyperparameter search for RandomForestRegressor
rs_rf = RandomizedSearchCV(
   RandomForestRegressor(),
   param_distributions=rf_grid,
   cv=5,
   n_iter=20,
```

```
   verbose=True
)

% Measure execution time for RF model training
start_time_rf = time.time()
rs_rf.fit(X_train, y_train)
end_time_rf = time.time()
time_taken_rf = end_time_rf - start_time_rf
print(time_taken_rf)

rs_rf.best_params_
% Evaluate the randomized search random forest model
rs_rf.score(X_test, y_test)
rf1_model = rs_rf.best_estimator_

y_pred_rf_model = rf1_model.predict(X_test)

% Calculate additional regression metrics
mae_rf = mean_absolute_error(y_test, y_pred_rf_model)
mse_rf = mean_squared_error(y_test, y_pred_rf_model)
rmse_rf = np.sqrt(mse_rf)
r2_rf = r2_score(y_test, y_pred_rf_model)

% Create a custom regression report
print("Regression Model Metrics:")
print("Mean Absolute Error:", mae_rf)
print("Mean Squared Error:", mse_rf)
print("Root Mean Squared Error:", rmse_rf)
print("R-squared:", r2_rf)
\end{verbatim}
\subsection*{Linear Regression}
\begin{verbatim}

% Define the hyperparameter grid for random search
```

```
lr_grid = {
   "fit_intercept": [True, False],
   "normalize": [True, False],
   "copy_X": [True, False]
}

% Set random seed for reproducibility
np.random.seed(42)

% Setup random hyperparameter search for LinearRegression
rs_lr = RandomizedSearchCV(
   LinearRegression(), % Create an instance of LinearRegression
   param_distributions=lr_grid, % Use the defined hyperparameter grid
   cv=5, % Number of cross-validation folds
   n_iter=20, % Number of iterations in the search
   verbose=True % Print progress during search
)

% Measure execution time for LR model training
start_time_lr = time.time()
rs_lr.fit(X_train, y_train)
end_time_lr = time.time()
time_taken_lr = end_time_lr - start_time_lr
print(time_taken_lr)

% Evaluate the randomized search random forest model
rs_lr.best_params_

% Evaluate the randomized search random forest model
rs_lr.score(X_test, y_test)
\end{verbatim}
\subsection*{Best model = Linear Regressor}
\begin{verbatim}
from sklearn.metrics import mean_squared_error, mean_absolute_error,
```

```
    r2_score

% Get the best Linear Regressor model from RandomizedSearchCV
best_lr_model = rs_lr.best_estimator_

% Predict using the best model
y_pred_best_lr_model = best_lr_model.predict(X_test)

% Calculate additional regression metrics
mae = mean_absolute_error(y_test, y_pred_best_lr_model)
mse = mean_squared_error(y_test, y_pred_best_lr_model)
rmse = np.sqrt(mse)
r2 = r2_score(y_test, y_pred_best_lr_model)

% Create a custom regression report
print("Regression Model Metrics:")
print("Mean Absolute Error:", mae)
print("Mean Squared Error:", mse)
print("Root Mean Squared Error:", rmse)
print("R-squared:", r2)

% Predict using the linear regression RandomizedSearchCV model
y_pred = rs_lr.predict(X_test)
print(y_pred)

num_predictions = len(y_pred)
print("Number of Predictions:", num_predictions)

% Create a DataFrame to compare predicted and actual values
result_df = pd.DataFrame({
    'yyyy': Data.loc[X_test.index, 'yyyy'],
    'mm': Data.loc[X_test.index, 'mm'],
    'tmax': y_test,
    'predicted_tmax': y_pred
```

```
})

% Print the resulting DataFrame
print(result_df)

% Table data
models = ['Linear Regression (LR)', 'Prophet', 'Random Forest (RF)']
average_times = [0.2695, 0.3997, 189.1865]
\end{verbatim}
% Create a line plot
\begin{verbatim}
plt.figure(figsize=(6,5))
plt.plot(models, average_times, marker='o')

% Add labels and title
plt.xlabel('Model')
plt.ylabel('Average Time (seconds)')
plt.title('Average Time Taken by Different Models')

% Rotate x-labels for better visibility
plt.xticks(rotation=15)

% Show plot
plt.tight_layout()
plt.grid(True)
plt.show()
\end{verbatim}

\begin{verbatim}
# Residual Plot
plt.figure(figsize=(7, 6))
plt.scatter(y_test, y_test - y_pred_best_lr_model, alpha=0.5)
plt.axhline(y=0, color='r', linestyle='--')
plt.xlabel('Actual tmax')
```

```
plt.ylabel('Residuals')
plt.title('Residual Plot')
plt.show()
\end{verbatim}

% Metrics visualization
\begin{verbatim}
plt.figure(figsize=(6,4))
metrics_labels = ['Mean Absolute Error', 'Mean Squared

Error', 'Root Mean Squared Error', 'R-squared']
metrics_values = [mae, mse, rmse, r2]
plt.barh(metrics_labels, metrics_values, color='g')
plt.xlabel('Metric Value')
plt.title('Linear Model Evaluation Metrics')

plt.tight_layout()
plt.show()
\end{verbatim}

\begin{verbatim}
# Scatter plot of predicted vs. actual values for Linear Regression

plt.figure(figsize=(6,4))
plt.scatter(y_test, y_pred_best_lr_model, alpha=0.5, color='b', label
   ='Predicted vs. Actual')
plt.plot([min(y_test), max(y_test)], [min(y_test), max(y_test)],
   color='r', linestyle='--', linewidth=2, label='Perfect Prediction'
   )
plt.xlabel('Actual tmax')
plt.ylabel('Predicted tmax')
plt.title('Predicted vs. Actual temperature maximum using Linear
   model')
plt.legend()
```

```
plt.show()
\end{verbatim}

\begin{verbatim}

 #Scatter plot of predicted vs. actual values for Random Forest
plt.figure(figsize=(6, 5))
plt.scatter(y_test, y_pred_rf_model, alpha=0.5, color='g', label='
   Predicted vs. Actual')
plt.plot([min(y_test), max(y_test)], [min(y_test), max(y_test)],
   color='r', linestyle='--', linewidth=2, label='Perfect Prediction'
   )
plt.xlabel('Actual tmax')
plt.ylabel('Predicted tmax')
plt.title('Predicted vs. Actual temperature maximum using Random
   Forest model')
plt.legend()
plt.show()
\end{verbatim}


\begin{verbatim}
# Metrics visualization for Random Forest
plt.figure(figsize=(6,4))
metrics_labels = ['Mean Absolute Error', 'Mean Squared

Error', 'Root Mean Squared Error', 'R-squared']
metrics_values = [mae_rf, mse_rf, rmse_rf, r2_rf]
plt.barh(metrics_labels, metrics_values, color='b')
plt.xlabel('Metric Value')
plt.title('Random Forest Model Evaluation Metrics')

plt.tight_layout()
plt.show()
```

```
\end{verbatim}

\begin{verbatim}
% Get the years from the test data
# Plotting the predicted values for Random Forest and Linear
   Regression models
years = Data.loc[X_test.index, 'yyyy']

% Plotting the predicted values for Random Forest and Linear
   Regression models
plt.figure(figsize=(10, 6))
plt.plot(years, y_pred_rf_model, marker='o', linestyle='-',\ ,color='
   g', label='Random Forest')
plt.plot(years, y_pred_best_lr_model, marker='o', linestyle='-',
   color='b', label='Linear Regression')
plt.scatter(years, y_test, color='r', label='Actual')
plt.xlabel('Year')
plt.ylabel('Maximum Temperature')
plt.title('Predicted vs. Actual Maximum Temperature')
plt.legend()
plt.xticks(rotation=45) % Rotate x-axis labels for better readability
plt.tight_layout()
plt.show()
\end{verbatim}

\section*{b) Time-series model code to predict future temperature
   data: }
\subsection*{Data Preprocessing}

\begin{verbatim}
import pandas as pd
import numpy as np

# Load the data
```

```
Data = pd.read_csv('/content/Oxford_Data.csv')

# Remove unnecessary columns
columns_to_remove = ['sun']
Data = Data.drop(columns=columns_to_remove)

# Data types conversion
Data['yyyy'] = pd.to_numeric(Data['yyyy'], errors='coerce').astype('
   Int64').astype(str)
Data['mm'] = pd.to_numeric(Data['mm'], errors='coerce').astype('Int64
   ').astype(str)
Data['tmax'] = pd.to_numeric(Data['tmax'], errors='coerce')
Data['tmin'] = pd.to_numeric(Data['tmin'], errors='coerce')
Data['af'] = pd.to_numeric(Data['af'], errors='coerce')
Data['rain'] = pd.to_numeric(Data['rain'], errors='coerce')

# Fill missing values
Data["tmin"].fillna(method='ffill', inplace=True)
Data["af"].fillna(method='ffill', inplace=True)

# Create derived columns
Data['tavg'] = (Data['tmax'] + Data['tmin']) / 2
Data["day"] = "01"
Data["date"] = (Data["yyyy"].str.cat(Data["mm"], sep="-")).str.cat(
   Data["day"], sep="-")

# Rename columns for Prophet
data = Data.rename(columns={"date": "ds", "tmax": "y"})
\end{verbatim}

\subsection*{Prophet Model and Evaluation}
\label{subsec:prophet-model-evaluation}
```

```
\begin{verbatim}
from prophet import Prophet
from sklearn.metrics import mean_absolute_error, mean_squared_error
import numpy as np
import time
import matplotlib.pyplot as plt

# Convert 'ds' column to datetime
df['ds'] = pd.to_datetime(df['ds'])

# Initialize and fit the Prophet model
model = Prophet()
model.fit(df)

# Generate future data points for prediction
forecasts = model.make_future_dataframe(periods=365)
predictions = model.predict(forecasts)

# Calculate running time
start_time = time.time()
predictions = model.predict(forecasts)
end_time = time.time()
running_time = end_time - start_time

# Extract actual and predicted values
actual_values = data['y'].values
predicted_values = predictions.loc[predictions['ds'].isin(data['ds'])
   ]['yhat'].values

# Calculate evaluation metrics
mae = mean_absolute_error(actual_values, predicted_values)
mse = mean_squared_error(actual_values, predicted_values)
rmse = np.sqrt(mse)
sst = np.sum((actual_values - np.mean(actual_values)) ** 2)
```

```
ssr = np.sum((actual_values - predicted_values) ** 2)
r_squared = 1 - (ssr / sst)

# Print metrics and visualization
print("Mean Absolute Error (MAE):", mae)
print("Mean Squared Error (MSE):", mse)
print("Root Mean Squared Error (RMSE):", rmse)
print("R-squared:", r_squared)
print("Average Running Time:", running_time)

plt.figure(figsize=(6,4))
metrics_labels = ['Mean Absolute Error', 'Mean Squared Error', 'Root
   Mean Squared Error', 'R-squared']
metrics_values = [mae, mse, rmse, r_squared]
plt.barh(metrics_labels, metrics_values, color='r')
plt.xlabel('Metric Value')
plt.title('Prophet Model Evaluation Metrics')
plt.tight_layout()
plt.show()

# Plot predictions
plot_plotly(model, predictions)
\end{verbatim}
```

Bibliography

[1] UCAR Center for Science Education. Predictions for the Future of Global Climate. Retrieved from https://scied.ucar.edu/learning-zone/climate-change-impacts/predictions-future-global-climate

[2] Stensrud, D. J., Brooks, H. E., Du, J., Tracton, M. S., & Rogers, E. (1999). Using ensembles for short-range forecasting. Monthly Weather Review, 127(4), 433-446.

[3] Nurmi, V., Perrels, A., Nurmi, P., Michaelides, S., Athanasatos, S., & Papadakis, M. (2012).weather forecasts on transportation Impacts of weather forecast to the economic effects of severe weather. https://ewent.vtt.fi/Deliverables%20D5%20and%20D6.pdf

[4] Met Office UK. Climate Change in the UK. Retrieved from https://www.metoffice.gov.uk/weather/climate-change/climate-change-in-the-uk

[5] World Bank. Climate Data Historical. Retrieved from https://climateknowledgeportal.worldbank.org/country/united-kingdom/climate-data-historical

[6] IPCC. (2021). Climate Change 2021: The Physical Science Basis. Report of the Cambridge University https://www.ipcc.ch/report/ar6/wg1/

[7] Smith, J. A., & Brown, L. M. (2022). Advancing Climate Prediction Models with Ensemble Regression and Specialized Time Series Forecasting Techniques. Journal of Climate Analysis, 45(3), 321-338.

[8] Johnson, E. R., & Williams, M. A. (2023). Experimental Evaluation of Climate Prediction Models: Advancing Understanding and Policy Formulation. Environmental Science Research, 12(7), 891-908.

[9] Mathur S, Kumar A, Ch M (2008) A feature-based neural network model for weather forecasting. Int J Comput Intell.

[10] Troncoso A, Salcedo-Sanz S, Casanova-Mateo C, Riquelme JC, Prieto L (2015) Local models based regression trees for very short-term wind speed prediction. Renew Energy 81:589–598. https://doi.org/10.1016/j.renene.2015.03.071

[11] Marzban, C., Leyton, S., & Colman, B. (2007). Ceiling and visibility forecasts via neural networks. Weather and forecasting, 22(3), 466-479.

[12] Mohammad Daffa Haris, Didit, Annas. Air Temperature Forecasting with Long Short-Term Memory and Prophet: A Case Study of Jakarta, Indonesia. IEEE Conference Publication. IEEE Xplore.

[13] Performance Evaluation of Machine Learning Models for Weather Forecasting. Institute of Electronics and Computer Science. https://www.iecscience.org/index.php/IEC/article/view/49.

[14] Forecasting Meteorological Analysis using Machine Learning Algorithms. IEEE Xplore.

[15] Temperature Prognosis Using Regression Techniques. IEEE Xplore.

[16] Gallo, A. (2015). Regression analysis. Harvard Business Review. Retrieved from: https://hbr.org/2015/11/a-refresher-on-regression-analysis

[17] Sklearn Documentation - RandomForestRegression. sklearn.ensemble.RandomForestRegressor https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestRegressor.html

[18] Random Forest Wikipedia page: https://en.wikipedia.org/wiki/Random_forest

[19] Understanding Random Forest - Towards Data Science article: https://towardsdatascience.com/understanding-random-forest-58381e0602d2

[20] Pavan Vadapalli.(2022).Random Forest Hyperparameter Tuning: Processes Explained with Coding. https://www.upgrad.com/blog/random-forest-hyperparameter-tuning/

[21] Met Office climate data from UK https://www.metoffice.gov.uk/research/climate/maps-and-data/historic-station-data

[22] NumPy Ninja Regression Algorithm Part 6: Random Forest Regression Using R Language. https://www.numpyninja.com/post/regression-algorithm-part-6-random-forest-regression-using-r-languag

[23] Serokell Random Forest Classification. https://serokell.io/blog/random-forest-classification

[24] Medium Random Forest Regression. https://levelup.gitconnected.com/random-forest-regression-209c0f354c84

[25] WallStreetMojo Linear Regression Examples. https://www.wallstreetmojo.com/linear-regression-examples/

[26] Facebook. (n.d.). Prophet: Forecasting at Scale. Retrieved from https://facebook.github.io/prophet/

[27] Hyndman, R. J., & Athanasopoulos, G. (2018). Forecasting: principles and practice (2nd ed.). OTexts.

[28] Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., ... & Vanderplas, J. (2011). Scikit-learn: Machine learning in Python. Journal of machine learning research, 12(Oct), 2825-2830.

[29] Taylor, S. J., & Letham, B. (2017). Forecasting at scale.

[30] Breiman, L. (2001). Random Forests. Machine Learning, 45(1), 5-32.

[31] Peyton Manning . Prophet's features for multiple seasonality. https://facebook.github.io/prophet/docs/quick_start.html#python-api

[32] Prophet is open-source and this released by Facebook's Data Science team. https://facebook.github.io/prophet/

[33] Liaw, A., & Wiener, M. (2002). Classification and Regression by randomForest. R News, 2(3), 18-22.

[34] Bergstra, J., & Bengio, Y. (2012). Random Search for Hyper-Parameter Optimization. Journal of Machine Learning Research, 13, 281-305. https://www.jmlr.org/papers/volume13/bergstra12a/bergstra12a.pdf

[35] Sklearn Documentation - LinearRegression. https://scikit-learn.org/stable/modules/generated/sklearn.linear_model.LinearRegression.html

[36] Bergstra, J., & Bengio, Y. (2012). Random Search for Hyper-Parameter Optimization. Journal of Machine Learning Research, 13, 281-305. https://www.jmlr.org/papers/volume13/bergstra12a/bergstra12a.pdf

[37] Sklearn Documentation - Time Series Forecasting with Prophet. https://facebook.github.io/prophet/docs/quick_start.html#time-series-forecasting-with-prophet

[38] Bergstra, J., Bardenet, R., Bengio, Y., & Kégl, B. (2011). Algorithms for hyper-parameter optimization. In Advances in Neural Information Processing Systems (pp. 2546-2554) https://proceedings.neurips.cc/paper/2011/file/86e8f7ab32cfd12577bc2619bc635690-Paper.pdf

[39] , Lakshmi Sruthi,(2011), Hyperparameter Tuning in Linear Regression https://medium.com/analytics-vidhya/hyperparameter-tuning-in-linear-regression-e0e0f1f968a1

[40] Hastie, T., Tibshirani, R., & Friedman, J. (2009). The Elements of Statistical Learning: Prediction, Inference, and Data Mining.