Predicting Carbon Dioxide Emission Levels

A Project Report
Presented to
Dr. Vidhyacharan Bhaskar
San Jose State University
In Partial Fulfillment
Of the Requirements for
CMPE 272
Enterprise Software Platforms

By
Moxank Patel
Nguyen Bui
12/2023

Table of Contents

Predicting Carbon Dioxide Emission Levels

Moxank Patel
Computer Engineering
San José State University
San Jose, California, USA
moxankprakashbhai.patel@sjsu.edu

Nguyen Bui
Software Engineering
San José State University
San Jose, California, USA
nguyen.bui@sjsu.edu

ABSTRACT - In response to the critical challenge posed by global warming, this study proposes an innovative approach to monitor atmospheric carbon dioxide (CO2) levels. Leveraging advanced machine learning techniques, we utilize a comprehensive dataset derived from the Sentinel-5P satellite, accessed via Kaggle. This methodology circumvents the limitations inherent in traditional Non-Dispersive Infrared (NDIR) sensor-based measurements, such as the need for physical sensor deployment and associated high costs. Our approach employs various predictive models, including Linear Regression, Lasso Regression, and Ensemble Regression Algorithms, with the ensemble model demonstrating superior performance, evidenced by a Root Mean Square Error of 10. This research signifies a significant advancement in environmental monitoring, offering a scalable, cost-effective, and precise method for tracking global CO2 emissions, which is vital for informed decision-making in climate change mitigation efforts.

Index Terms - Sentinel-5P, NDIR sensor, Linear Regression, Ensemble Learning, Root Mean Square Error.

I. INTRODUCTION

In the contemporary global context, the profound challenge of global warming has emerged as a pressing concern, necessitating a comprehensive strategy to tackle its root cause—the escalating levels of carbon dioxide emissions [1]. An effective and pragmatic approach to curb emissions involves deploying sophisticated monitoring instruments to scrutinize carbon dioxide concentrations. The state-of-the-art methodology uses on-ground sensors known as Non-Dispersive Infrared Sensors (NDIR) [2]. Despite their commendable attributes of high accuracy and reliability, a significant impediment surfaces due to the impracticality of maintaining a physical presence at every location for continuous monitoring.

To surmount this logistical challenge, the present research advocates for the strategic integration of machine learning algorithms applied to historical datasets of carbon dioxide emission levels. The dynamic capabilities of machine learning algorithms in discerning intricate patterns within data and subsequently leveraging these patterns to inform future decisions have been well-documented [3]. In the specific context of predicting carbon dioxide emission levels, regression algorithms stand out as instrumental tools. These algorithms are designed to establish a mapping between given parameters and the target variable, employing either linear or nonlinear functions. Examples of such algorithms include linear regression, polynomial regression, XGB regression, Lasso Regression, and Ridge Regression, among others, all of which have exhibited effectiveness across diverse datasets. This research aims to harness the rich insights encapsulated in the Kaggle Dataset [4] to predict carbon emission levels with enhanced accuracy and reliability.

Before delving into the predictive modeling of carbon emissions, a critical prerequisite is the meticulous data preprocessing to ensure cleanliness, thereby facilitating the algorithmic identification of the target variable. Data cleaning procedures encompass a spectrum of operations, including replacing null values, data centering, and the feature selection of the most pivotal attributes for training purposes [5]. Following a rigorous analysis, the identified optimal model for this study is XGB Regression, showcasing a commendable performance with a Root Mean Squared Error of 10.15.

The subsequent sections of this paper will unfold comprehensively, encompassing a thorough review of related works, a robust justification for the chosen project, an elucidation of the system architecture employed, a detailed

presentation of results, a conclusive summary, and, importantly, an exploration of potential avenues for future research directions.

II. RELATED WORK AND BACKGROUND

In addressing global warming, a critical concern is the increasing carbon dioxide (CO2) emissions, identified as a primary driver of climate change [1]. Therefore, the urgency to have countermeasures against global is an absolute necessity[2]. Traditionally, ground-level CO2 measurements have been conducted using sensors, with Non-Dispersive Infrared (NDIR) sensors being particularly noted for their high accuracy in CO2 detection, utilizing advanced infrared technology [3]. However, some researchers use predictive algorithms for this task and have shown significant progress [4].

Various research endeavors have contributed valuable insights and methodologies in the evolving landscape of predictive modeling for CO2 emissions. Notably, the work by Qin et al. explored the use of Support Vector Machines (SVM) in predicting CO2 levels, demonstrating the substantial capabilities of machine learning algorithms in environmental monitoring. This study underscores the potential for machine learning techniques to be pivotal in addressing environmental challenges [2]. A comparative study by Yang Meng and Hossain also delved into the efficacy of four distinct models for forecasting CO2 levels [3]. This research provided a comprehensive analysis of the strengths and weaknesses of various predictive models by comparing different regression algorithms and providing a user-friendly interface.

Furthermore, the specific issue of traffic-related emissions has been the focus of a study conducted in Seoul. This research utilized machine learning algorithms to estimate CO2 emissions from traffic, incorporating diverse factors such as traffic volume and wind speed. This approach exemplifies the application of predictive modeling [4] in understanding and mitigating sector-specific emission sources.

Collaborative efforts in data collection have been instrumental in advancing research in CO2 emission modeling. Darius Moruri's team, in collaboration with platforms such as Zindi and Kaggle, has been pivotal in gathering extensive datasets. These datasets encompass key indicators like traffic patterns and atmospheric conditions, which are crucial for developing and refining predictive models [5-7].

This body of work collectively contributes to the growing field of predictive modeling for CO2 emissions, offering diverse perspectives and methodologies. These studies not only enhance our understanding of emission patterns but also pave the way for developing innovative strategies to mitigate the impacts of global warming.

III. PROJECT JUSTIFICATION

The escalating crisis of global warming necessitates the implementation of preventative measures, particularly the control of rising carbon dioxide emissions. A critical component in addressing this challenge is to monitor effectively and, where possible, forecast carbon dioxide levels across diverse locations. This research project seeks to harness the capabilities of the Sentinel-5P satellite to predict carbon emission levels in remote areas. This approach provides a viable alternative to Non-Dispersive Infrared (NDIR) sensors.

The Sentinel-5P satellite measures various atmospheric parameters, including UV aerosol indices, carbon levels, sulfur dioxide concentrations, and other relevant chemical components. The availability of such comprehensive satellite data presents a unique opportunity to employ machine learning algorithms for predictive analysis. This study applies various regression algorithms to determine the most effective model for the dataset.

The goal is to identify an algorithm that accurately predicts current carbon emissions and forecasts future levels. Establishing a predictive framework makes it possible to implement countermeasures proactively ahead of potential exponential rises in carbon emissions.

IV. PROPOSED PROCEDURE FOR PREDICTING CARBON EMISSION LEVELS

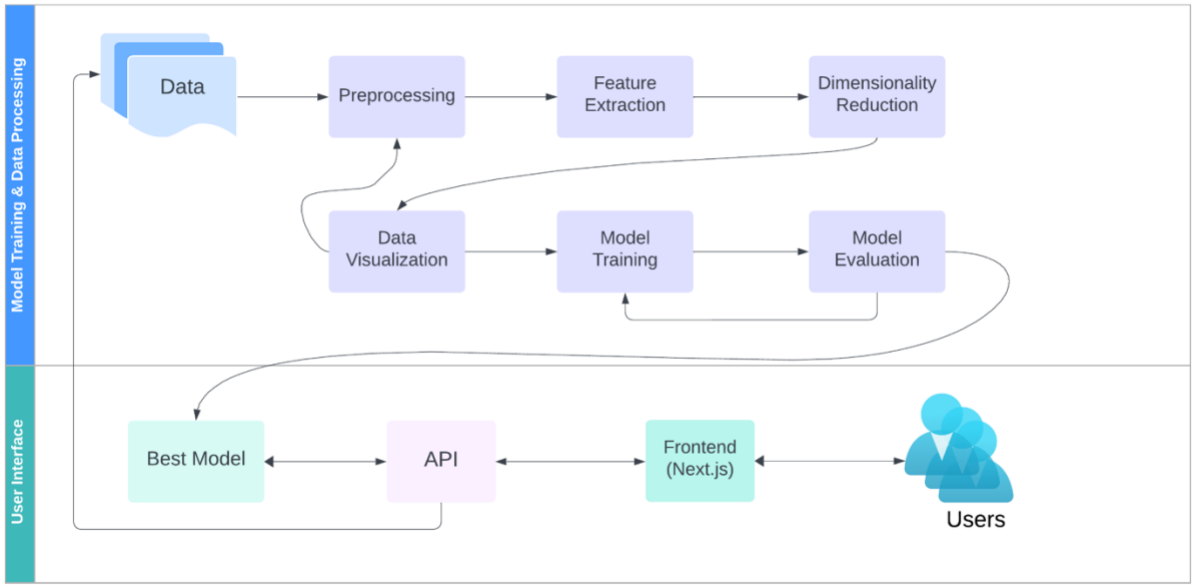


Fig. 1 Architecture of the System.

Figure 1 illustrates the comprehensive architecture of the proposed system, which is systematically divided into two distinct yet interconnected phases. The initial phase is dedicated to data processing and model training, a crucial step in the system's overall functionality. This phase encompasses the meticulous collection, cleaning, and preprocessing of data, followed by the strategic selection of appropriate machine learning models for training. The process involves utilizing advanced algorithms and techniques to ensure that the model is robust, accurate, and capable of handling the complexity of the analyzed environmental data.

The system architecture's second phase focuses on applying the trained model to display results in a user-friendly interface effectively. This phase is essential in translating the complex data predictions into an accessible and understandable format for end-users. The interface is designed to be intuitive, allowing users to interact with the system, view predictions, and understand the implications of the data in real-time.

A. Preprocessing

A comprehensive data preprocessing phase was undertaken before inputting the data for model processing. This phase was critical to ensure the integrity and accuracy of the data fed into the model. A key step in this process involved addressing the issue of null values within the dataset. The chosen strategy for dealing with these null values was to replace them with the mean value of the respective attribute, a method commonly adopted to maintain statistical consistency. Following this, the data underwent a centering process, specifically mean centering. This technique is instrumental in normalizing the data, thereby enhancing the efficiency and accuracy of the predictions made by the model. Such meticulous preprocessing steps are essential in preparing the dataset for effective and reliable analysis in machine learning applications.

B. Feature Extraction & Dimensionality Reduction

As every feature is crucial in prediction, no features were eliminated because their correlation was very low. The data's dimensionality was reduced using Principal Component Analysis, and the fixate a, which is used in prediction, has a variance of 90% in comparison to the o original dataset.

C. Data Visualization

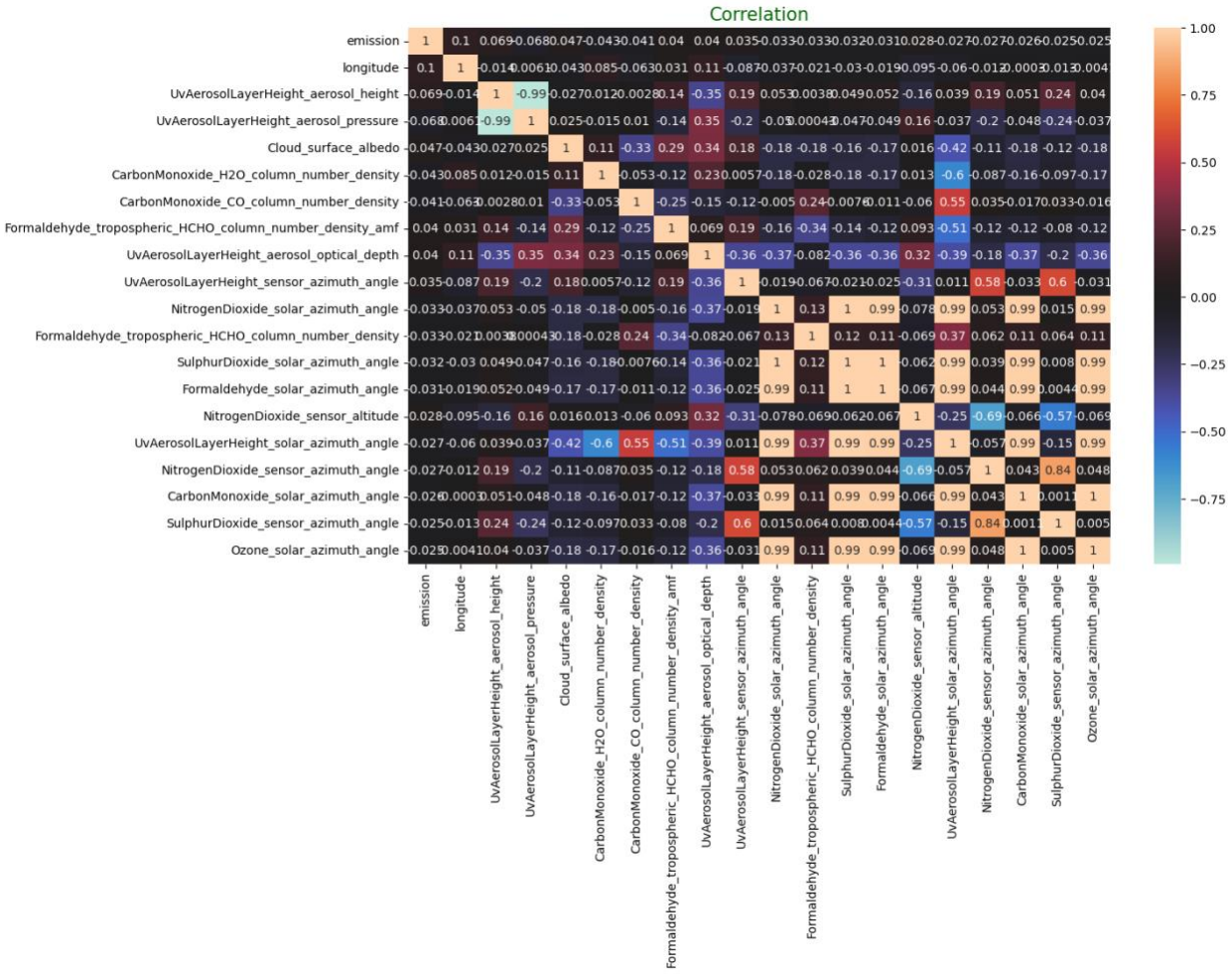


Figure 2. Correlation Between Emission and Other Variables.

Upon completion of the dataset preprocessing, a correlation heatmap was generated to examine the relationship between various attributes and the emission levels, which served as the target variable. This analysis is depicted in "Correlation Between Emission and Other Variables."

The heatmap provided a visual representation of the correlation coefficients, enabling an easy assessment of the strength and direction of relationships between the emissions and each attribute. Notably, the highest correlation observed was a coefficient of 0.11 with latitude. However, this relatively low value indicates a weak correlation between the emissions and the analyzed variables. The implication of these findings is significant; it suggests that the attributes under consideration do not substantially influence emission levels. This insight is crucial for understanding the dynamics of emission predictors and guides further analysis in identifying more impactful variables or reconsidering the model's approach to capturing the complexity of emission determinants.

TABLE I
STATISTICAL ANALYSIS OF THE DATASET

	latitude	longitude	year	week_no	emission
count	79023.000000	79023.000000	79023.000000	79023.000000	79023.000000
mean	-1.891072	29.880155	2020.000000	26.000000	81.940552
std	0.694522	0.810375	0.816502	15.297155	144.299648
min	-3.299000	28.228000	2019.000000	0.000000	0.000000
25%	-2.451000	29.262000	2019.000000	13.000000	9.797995
50%	-1.882000	29.883000	2020.000000	26.000000	45.593445
75%	-1.303000	30.471000	2021.000000	39.000000	109.549595
max	-0.510000	31.532000	2021.000000	52.000000	3167.768000

Table I presents a detailed statistical dataset analysis, encompassing key attributes such as latitude, longitude, year, week number, and emissions. This analysis provides a comprehensive overview of these variables, offering insights into their distribution and range within the dataset.

The table enumerates the count of observations for each attribute, all consistent at 79,023 entries, indicating a complete dataset without missing values for these variables. The mean values for latitude and longitude are -1.891072 and 29.880155, respectively, reflecting the central tendency of these geographical coordinates. The year attribute predominantly centers around 2020, with the mean and median (50th percentile) being 2020. The average week number is 26, indicating a mid-year distribution.

Standard deviation (std) values, which measure dispersion or variability in the data, are also listed. A standard deviation of 0.694522 for latitude and 0.810375 for longitude suggests moderate variability in these geographic coordinates. The emission values have a notably higher standard deviation of 144.299648, indicating a wide range of emission levels within the dataset.

The subsequent figure explores the relationship between these variables and emissions. The mappings revealed are nonlinear, indicating that a simple linear model may not accurately model this dataset. This insight is critical for selecting more appropriate, possibly nonlinear, modeling approaches for further analysis and prediction.

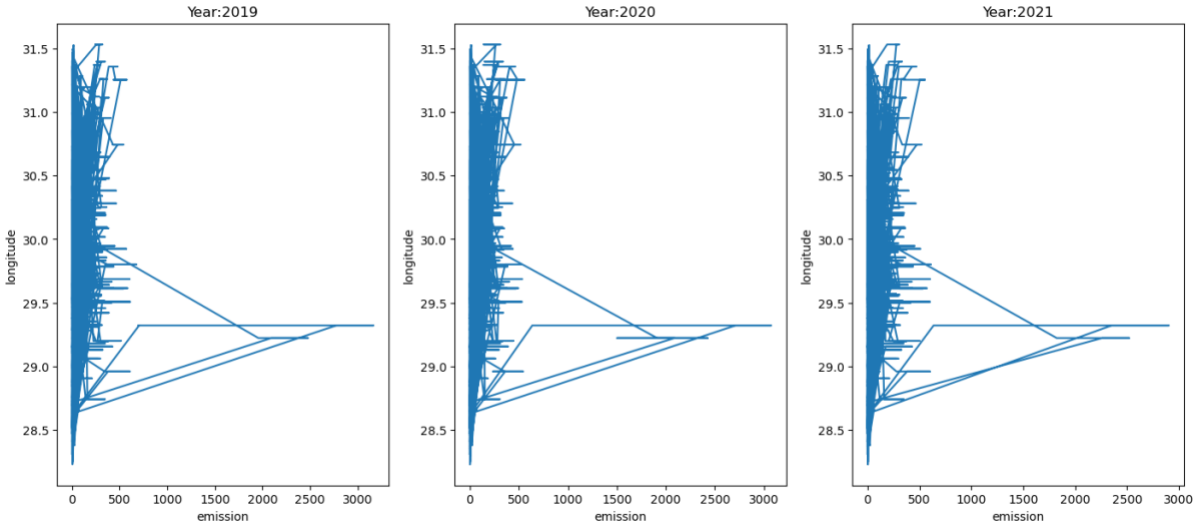


Figure 3.1. Line plot for latitude vs emission

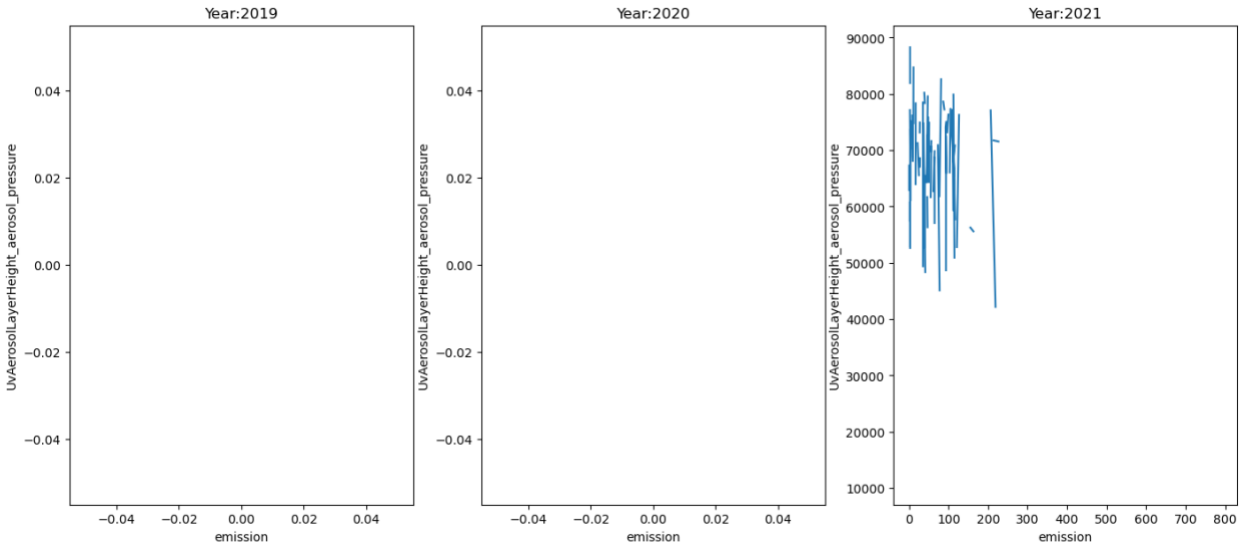


Figure 3.2. Line plot for different UV Aersol vs. emission

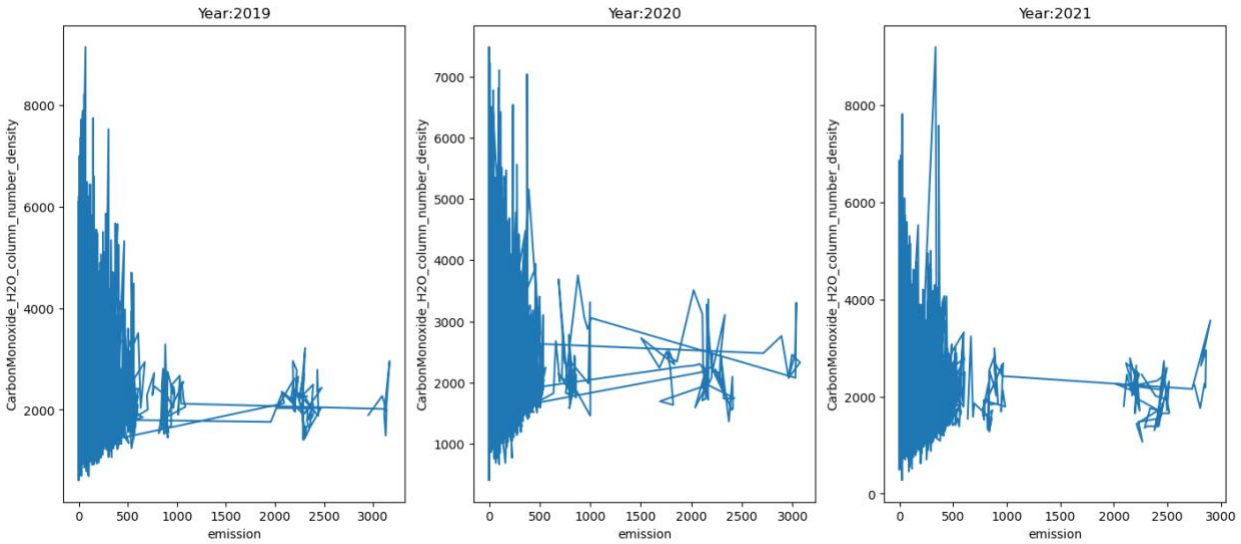


Figure 3.3. Line plot for different Carbon Monoxide vs emission.

D. Interface architecture

The interface architecture of the system plays a pivotal role in bridging the gap between the predictive model and the end-user experience.

- *User Interface (UI) Design and Functionality*

The UI is meticulously crafted to ensure user-friendliness while maintaining high functionality. It incorporates intuitive navigation tools and a clear layout, allowing users to input relevant parameters effortlessly. The design philosophy centers on simplicity and efficiency, ensuring users can easily navigate and utilize the system regardless of their technical expertise.

- *Data Input and Parameter Selection*

The data input and parameter selection module are at the UI's core. This module is structured to guide users in entering necessary information, such as geographical location, time parameters, and other relevant

environmental variables. The interface is equipped to handle various data formats and ensures seamless integration of user inputs into the predictive model.

- *Results Display and Visualization*

Once the predictive model processes the input data, the results are displayed comprehensibly and visually appealing. The UI includes features for dynamic data visualization, such as graphs, charts, and heatmaps, providing users with a clear understanding of the predictive outcomes. This visualization presents the estimated CO2 emission levels and offers insights into the underlying trends and patterns.

- *Interactive Features and User Engagement*

The UI incorporates interactive features such as adjustable parameters, real-time data updates, and scenario analysis tools to enhance user engagement. These features allow users to explore different scenarios, understand the impact of various variables on CO2 emissions, and gain a deeper insight into the environmental implications of their data.

In summary, the interface architecture is an integral component of the system, designed to connect the predictive model with the users seamlessly. It provides an intuitive, interactive, and informative platform, enhancing the overall user experience and the practical application of the system in environmental analysis and decision-making.

V. EXPERIMENT VALIDATIONS

This section will evaluate the performance of the proposed model. The section will also provide details regarding the environment setup and evaluation metrics used to evaluate the model performance. Further, the section will also highlight the limitations of the proposed system.

A. Environment Setup And Datasets

The system is divided into components: the first is React Frontend, and the second is Django API with Model. The system can be deployed on any platform after installing the dependencies. The installation instructions are available in the readme file

B. Evaluation Metrics

The evaluation metric for model performance is Root Mean Squared Error. It is the average squared difference between the predicted and original values. Mathematical it is:

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}$$

- Where n = number of observation
- y_i = Actual Value, $y_i^{\wedge}$ = Predicted Value

C. Results

The analysis evaluated various machine learning models for their effectiveness in predicting carbon dioxide emissions, as indicated by their respective Root Mean Square Error (RMSE) values, compiled in Table II. The Ensemble model employing the XGB Regressor demonstrated superior performance with the lowest RMSE of 10.15, indicating high predictive accuracy. In contrast, traditional models such as Linear Regression, Lasso Regression, and MLP Regressor (Neural Networks) exhibited significantly higher RMSE values, 135.52 and 140.57 138.97, respectively, suggesting lower predictive accuracy. Notably, models like the Decision Tree Regressor,

Radius Neighbors Regressor, K Neighbors Regressor, and another Ensemble model using the Random Forest Regressor achieved moderately low RMSEs, ranging from 15.38 to 22.40, highlighting their potential effectiveness. This comparative analysis underscores the varying degrees of efficacy among different regression models in the context of environmental data prediction.

TABLE II
MODEL'S SCORE

#	Model	Root Mean Square Error
1	LinearRegression	135.52
2	Lasso Regression	140.57
3	Ridge Regression	135.81
4	Decision Tree Regressor	22.40
5	Radius Neighbors Regressor	22.40
6	K Neighbors Regressor	22.40
7	Ensemble (Random Forest Regressor)	15.38
8	MLP Regressor (Neural Networks)	138.97
9	Ensemble (XGBRegressor)	10.15

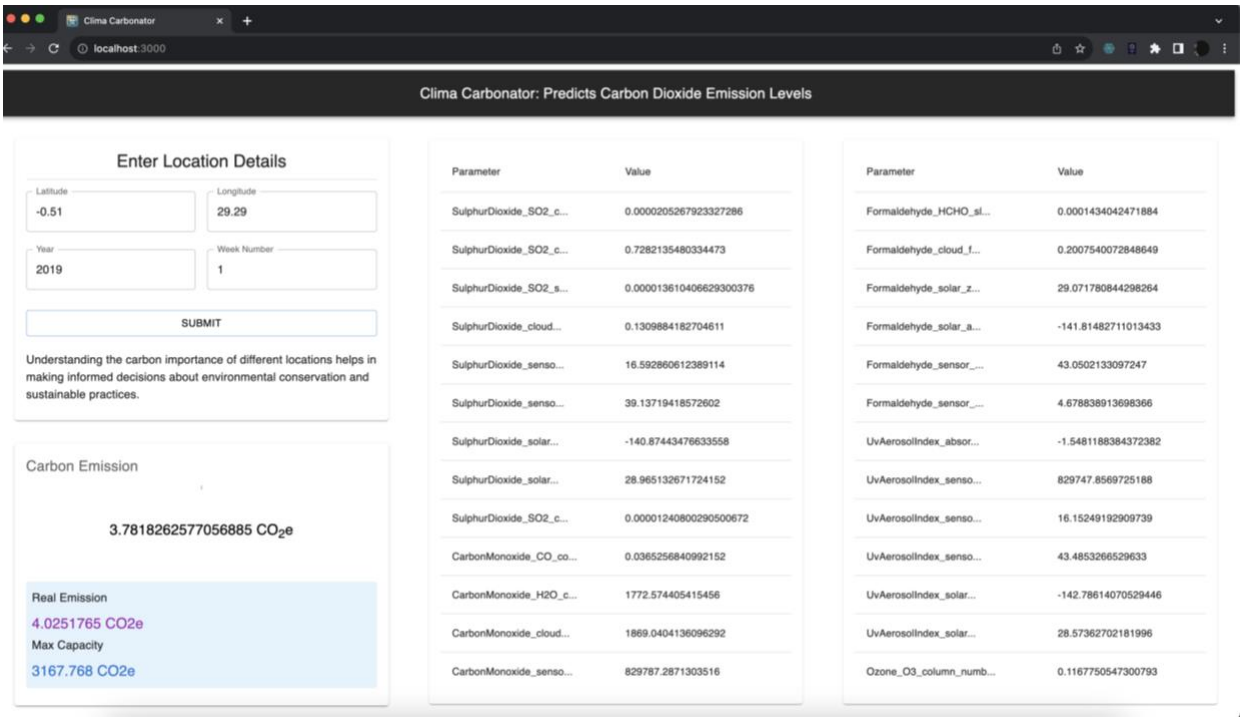


Figure 4. Snapshot the User Interface.

Figure 4 presents a user interface (UI) for the "Clima Carbonator," a tool designed to predict carbon dioxide emission levels. Here is a breakdown of the UI elements and functionality:

1. Location Input Section: The UI features a form where users can enter specific location details, including latitude, longitude, year, and week number. Once the details are entered, users can submit the data through a "SUBMIT" button for processing.

2. Emission Results Display: The results are presented clearly and concisely, displaying the calculated carbon emission for the entered location details. The display includes the "Carbon Emission," "Real Emission," and "Max Capacity" metrics, all denoted in CO2e (carbon dioxide equivalent), which is a standard unit for measuring carbon footprints.

3. Parameters and Values: To the right, two columns are filled with various parameters and their corresponding values. These represent the input data to the model in calculating the emissions. These parameters include substances like sulfur dioxide (SO2) and carbon monoxide (CO), along with their specific conditions like clouds, sensors, and solar influences.

VI. CONCLUSION AND FUTURE WORKS

This research outlines an effective way to predict carbon dioxide emissions with a Root Mean Square Error (RMSE) of 10.15. The algorithm mostly uses geolocation data and temporal variables like the year and time for its predictive analysis. This level of accuracy is noteworthy, but it is important to note that the model does not consider some potential parameters during the prediction process due to significant noise generated by their synthetic nature. However, these parameters could be useful in enhancing the accuracy of the predictions if measured in real-time.

The study suggests integrating a map API within the user interface to improve the user experience and provide a more interactive way of understanding emission patterns. This feature would allow the dynamic representation of real-time emissions data as a heatmap overlay on geographical maps. Additionally, the platform's interface could be developed further to include functionalities for emission forecasting. This would extend the platform's utility, offering users valuable insights into future emission trends and facilitating informed decision-making in environmental management and policy formulation.

This research is a significant advance in using satellite data and machine learning algorithms for environmental monitoring and prediction. By integrating real-time data visualization and forecasting capabilities, this tool can help address the challenges posed by global carbon emissions and climate change.

VII. REFERENCES

[1] I. C. Ward, "Will global warming reduce the carbon emissions of the Yorkshire Humber Region's domestic building stock—A scoping study," Energy and Buildings, vol. 40, no. 6, pp. 998–1003, 2008. [Online]. Available: https://doi.org/10.1016/j.enbuild.2007.08.007

[2] U. Olausson, "Global warming—global responsibility? Media frames of collective action and scientific certainty," Public Understanding of Science (Bristol, England), vol. 18, no. 4, pp. 421–436, 2009. [Online]. Available: https://doi.org/10.1177/0963662507081242

[3] L. Zhou, Y. He, Q. Zhang, and L. Zhang, "Carbon Dioxide Sensor Module Based on NDIR Technology," Micromachines, vol. 12, no. 7, Art. no. 845, 2021. [Online]. Available: https://doi.org/10.3390/mi12070845

[4]. D. Moruri, A. Bray, W. Reade, and A. Chow, "Predict CO2 Emissions in Rwanda," Kaggle, 2023. [Online]. Available: https://kaggle.com/competitions/playground-series-s3e20

[5] X. Qin et al., "China's carbon dioxide emission forecast based on improved marine predator algorithm and multi-kernel support vector regression," Environmental Science and Pollution Research International, vol. 30, no. 3, pp. 5730–5748, 2023. [Online]. Available: https://doi.org/10.1007/s11356-022-22302-7

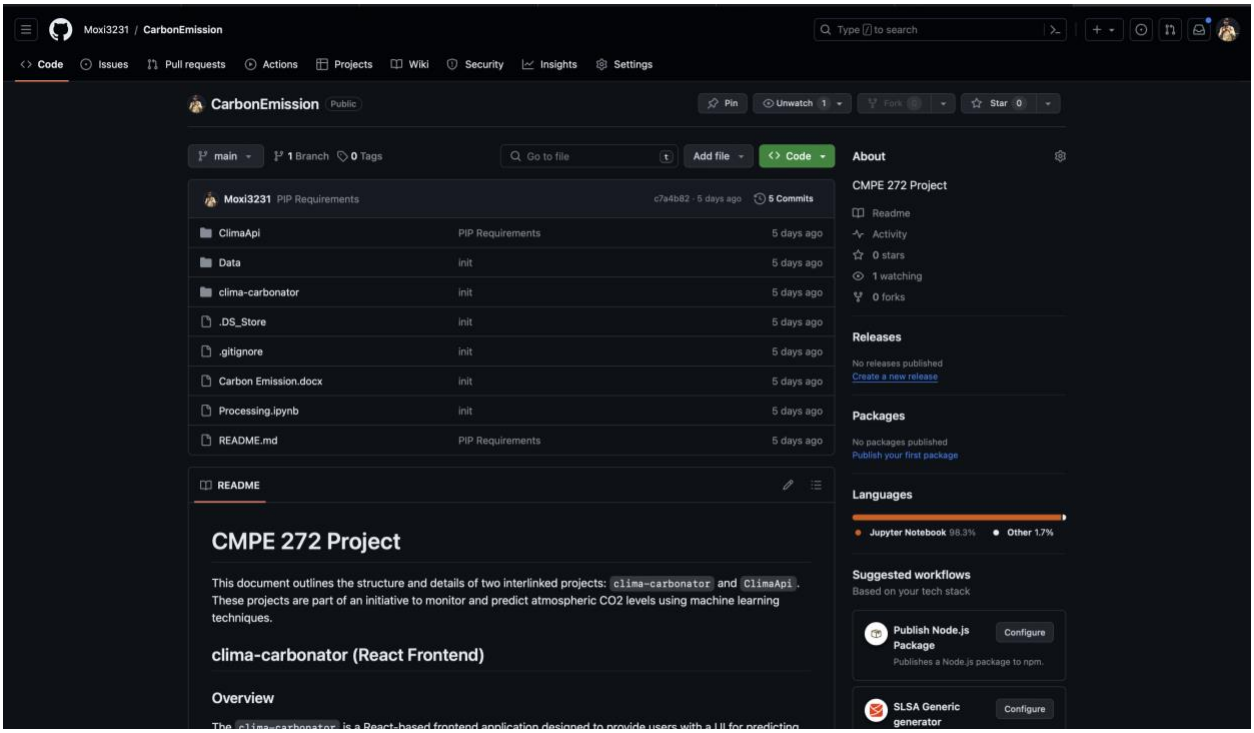
[6] Y. Meng and H. Noman, "Predicting CO2 Emission Footprint Using AI through Machine Learning," Atmosphere, vol. 13, no. 11, Art. no. 1871, 2022. [Online]. Available: https://doi.org/10.3390/atmos13111871

[7] C. Park et al., "Machine Learning Based Estimation of Urban On-road CO2 Concentration in Seoul," Environmental Research, vol. 231, p. 116256, 2023. [Online]. Available: https://doi.org/10.1016/j.envres.2023.116256

Appendix A: Source Code and Setup

GitHub Repo: https://github.com/Moxi3231/CarbonEmission

The above repository contains the source code for the interface, data processing, data modeling, model training, and model selection.



The repository is organized into several distinct sections:

1. ClimaApi: This section hosts a Django-based API, a crucial bridge facilitating interactions between the user interface and the underlying model. It is designed to handle requests and responses, ensuring seamless communication and data flow.

2. Data: This directory is dedicated to housing the training data. It is a critical repository containing the datasets used for training the machine learning models. The data stored here is likely diverse and voluminous, serving as the foundation for the model's learning and accuracy.

3. Clima-carbonator: This part of the repository features an interface built using Next.js, a React framework. It is designed to be intuitive, providing a smooth user experience while accessing the model's functionalities.

4. Processing.ipynb: This Jupyter notebook contains comprehensive code for various stages of the model's development. It includes data preprocessing, the initial step in preparing the data for training. Feature selection, a process that involves choosing the most relevant data attributes for the model, is also covered. Following this, the notebook details the methods used for model training, where the machine learning

algorithm learns from the training data. Finally, it includes selecting the best model based on performance metrics.

Running the project: The run-through and setup are provided in the Readme.md file.