Kathmandu University

Department of Computer Science and Engineering

Dhulikhel, Kavre

A Project Report

on

"Analyzing Nepal's Weather, Urbanization Trends, and Their Impacts on Agriculture Using Machine Learning Approach"

[Code No: COMP 311]

(For partial fulfillment of 3rd Year 1st Semester in Computer Science)

Submitted by

Sujan Ghimire (15)
Bigyan Kumar Piya (41)
Manasvi Sharma (52)
Sewan Uprety (60)

Submitted to

Mr. Nabin Ghimire

Computer Science Coordinator

Department of Computer Science and Engineering

Submission Date: 18th June 2024

Bona fide Certificate

This project work on

“Analyzing Nepal's Weather, Urbanization Trends, and Their Impacts on Agriculture Using Machine Learning Approach”

is the bona fide work of

Sujan Ghimire

Bigyan Kumar Piya

Manasvi Sharma

Sewan Uprety

who carried out the project work under my supervision.

Project Supervisor

_____

Dr. Sudan Jha

Professor

Department of Computer Science and Engineering

Date:

Abstract

This project endeavors to predict crop yields in Nepal using machine learning and time series forecasting techniques based on comprehensive historical data. As an agricultural nation, Nepal faces significant challenges in optimizing crop productivity amidst varying climatic conditions across its diverse districts. The dataset includes crucial parameters like district, crop type, year, yield, and diverse weather variables, providing a rich foundation for analysis. The primary goal is to harness this dataset to develop robust predictive models capable of forecasting future crop yields. By integrating advanced machine learning algorithms such as Gradient Boosting, Random Forest, SVR, and Decision Trees, alongside time series analysis using ARIMA models, the project aims to capture and utilize intricate temporal and spatial patterns in crop yield data

Keywords: Nepal, weather patterns, agriculture, data analysis, machine learning, Arima, Time series forecast

Table of Contents

List of Figures

List of Tables

Acronyms/Abbreviations:

ANN	Artificial Neural Network
AR	Autoregression
ARIMA	Auto Regressive Integrated Moving Average
CNN	Convolutional Neural Network
CPU	Central Processing Unit
EDA	Exploratory Data Analysis
GPU	Graphics Processing Unit
MA	Moving Average
MAE	Mean Absolute Error
MLP	Multi-Layer Perceptron
MSE	Mean Squared Error
RMSE	Root Mean Squared Error
SVR	Support Vector Regression
LSTM	Long Short-Term Memory
SVM	Support Vector Machine
DNN	Deep Neural Network

Chapter 1 Introduction

1.1 Background

Since Nepal is an agricultural country, this stream of occupation remains the primary source of livelihood for a significant portion of Nepal's population. Agriculture contributes to approximately 28% of the national GDP and employs over 60% of the workforce. However, the yield from agricultural activities is highly variable, influenced by a myriad of factors ranging from climatic conditions and soil health to farming practices and market access.

In order to address this issue, we have integrated data from multiple sources into our project and employed machine learning algorithms to predict the yield of crops in different regions of Nepal. In order to create diversification, we have used datasets of 10 districts of Nepal and predicted the yield of those regions by training them on 4 machine learning algorithms and a neural network. In order to make it even more reasonable, we have used only 5 crops that are majorly grown in these regions. Different influencing factors such as temperature, rainfall, pesticide use, humidity, and others fall under parameters in our dataset. We have used the datasets of different decades to analyze the crop yield through different time periods for these districts.

1.2 Objectives

- Create a predictive model for crop yields of major crops in different districts using diverse datasets.
- Analyze and identify the key variables that significantly impact crop yields in the selected districts.
- Provide actionable insights to aid farmers and policymakers in making informed decisions about planting and resource allocation.
- Offer district-specific recommendations and highlight regional disparities to support local agricultural growth.

1.3 Motivation and Significance

The motivation behind predicting crop yield in Nepal using machine learning stems from the need to enhance agricultural productivity and ensure food security in a country where agriculture is the backbone of the economy. Given the significant impact of climate variability, resource limitations, and diverse topographical challenges, there is an urgent need for precise, data-driven insights to support farmers and policymakers. Based on the yield of different crops in different regions of Nepal, we can determine the type of crop that is the most viable for production within that region, and extend the framework to broader applications across Nepal.

Chapter 2 Related Works

2.1 An Artificial Neural Network for Predicting Crops Yield in Nepal (2014) by Tirtha Ranjit and Leisa Armstrong

In the project "An Artificial Neural Network for Predicting Crops Yield in Nepal" by Tirtha Ranjeet and Leisa Armstrong (Edith Cowan University), published in 2014, the authors explore the application of Artificial Neural Networks (ANNs) for predicting crop yields in Nepal's Siraha district (eastern Terai region). Their study focuses on integrating extensive datasets encompassing climatic variables, soil characteristics, and agricultural practices sourced from governmental and local databases. Their ANN model, trained on historical paddy field data including climate (rainfall, temperature) and fertilizer use, achieved low prediction errors. This suggests ANNs could be a valuable tool for Nepal's agriculture, though the research focused on a specific crop and region, limiting its direct application everywhere in Nepal. Results indicate that the ANN model achieves high accuracy in predicting crop yields, outperforming traditional statistical methods by effectively capturing the nuanced interactions between environmental factors and agricultural outcomes. The study underscores the ANN's scalability and adaptability, suggesting its potential for broader applications in improving agricultural decision-making and resource management in Nepal's diverse farming landscapes.

2.2 Crop Yield Prediction using Machine Learning and Deep Learning Techniques (2023) by Mathur, Jain and Nijhawan

The research paper "Crop Yield Prediction using Machine Learning and Deep Learning Techniques" by Kavita Jhajharia, Pratistha Mathur, Sanchit Jain, and Sukriti Nijhawan provides an in-depth analysis of different ML and DL algorithms for crop yield prediction. Among the algorithms tested, Random Forest (RF) performed the best,

offering superior accuracy compared to Support Vector Machines (SVM) and Long Short-Term Memory (LSTM) networks. RF's effectiveness is attributed to its ability to handle high-dimensional data and capture complex interactions between features. The study utilized high-resolution sensor and satellite imagery data, highlighting the importance of quality data in achieving accurate predictions. Additionally, the research underscores the potential of these technologies to revolutionize agricultural practices by enabling precise yield forecasts, thereby optimizing resource use and decision-making processes. Future research directions include enhancing model algorithms, incorporating more diverse data sources, and addressing challenges related to data quality and computational requirements. This comprehensive approach suggests a promising future for AI-driven advancements in agriculture.

2.3 Crop Yield Prediction using Deep Neural Networks (2019) by Saeed Khaki and Lizhi Wang

This research conducted by Saeed Khaki and Lizhi Wang explores the effectiveness of Deep Neural Networks (DNNs) for predicting crop yield. Utilizing extensive datasets from various sources, including weather conditions, soil properties, and historical yield data, the study employs DNN models to analyze and predict crop yields. The authors trained two DNN models: one for yield and another for a "check yield" (reference point). The difference between the predicted yields from these models was used as the final yield prediction. This approach proved more effective than using a single DNN for yield difference directly. The model captured the influence of genotype and environmental factors on yield and check yield. The model outperformed other methods like Lasso regression, shallow neural networks, and regression trees. The results suggested environmental factors have a stronger influence on crop yield compared to genetics in this study. The results demonstrate that DNNs significantly outperform traditional machine learning methods in prediction accuracy, offering a

robust tool for agricultural planning and decision-making, ultimately contributing to more efficient and sustainable farming practices.

Chapter 3 Design and Implementation

3.1 Workflow of Program

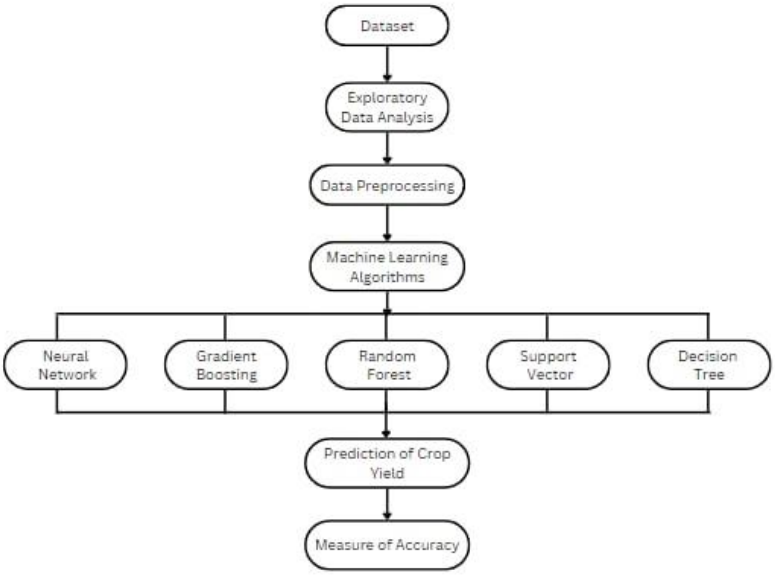

Figure 3.1.1 Workflow of program

3.2 Overall architecture

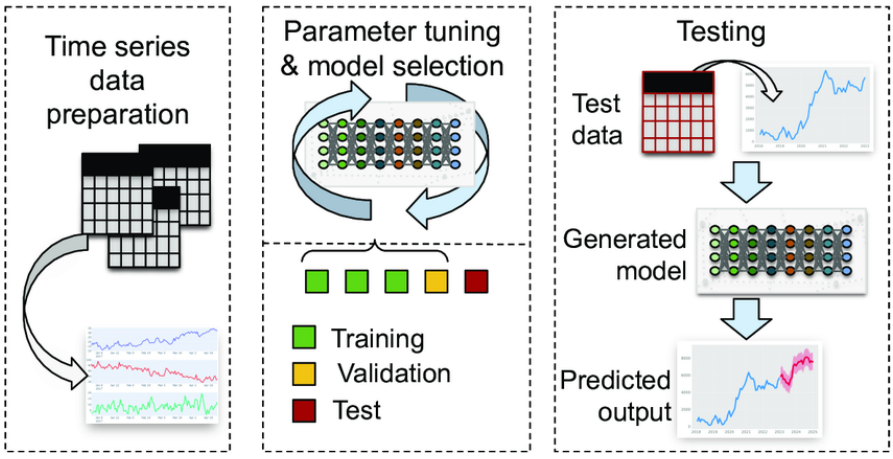

Figure 3.2.1 Methodology of project

3.3 Data Collection

Firstly, we gathered datasets related to weather patterns and agriculture statistics, indicators from various sources such as the Ministry of Agriculture and livestock development, NASA POWER project, and relevant research publications.

Also, we ensured data quality and consistency by conducting thorough checks for missing values, outliers, and inconsistencies. Ministry of Agriculture and livestock development in its annual report provided comprehensive agricultural data including district-wise crop production, crop types, and yields. NASA POWER project offered detailed weather data such as temperature, precipitation, humidity, and solar radiation.

3.4 Data Preprocessing

It is the process of cleaning the datasets by addressing missing values, outliers, and inconsistencies using appropriate techniques such as imputation, removal, or interpolation.

Transforming the data as necessary through techniques such as normalization, standardization, or feature engineering to ensure compatibility with the chosen analytical methods.

Figure 3.43.4.1 Preprocessing of data

3.5 Exploratory Data Analysis (EDA)

When we were done collecting and preprocessing the data, we performed EDA on the data to gain insights on the data we were working on. It focuses on understanding of

the pattern, relationships or distributions of the data set and also identifies the irregularities in the sets that might require future study

	District	Crop	Year	Area	Production	Yield	RH2M	GWETTOP	T2M_MAX	T2M_MIN	GWETROOT	WS2M_MAX	WS2M_MIN	PRECTOTCORR
0	Jhapa	Paddy	197980	87070	148020.0	1700	65.38	0.61	38.30	8.25	0.62	7.17	0.02	5.27
1	Jhapa	Paddy	198081	87000	152250.0	1750	62.19	0.59	40.09	7.88	0.60	6.55	0.02	5.27
2	Jhapa	Paddy	198182	91380	137070.0	1500	55.19	0.52	40.82	6.88	0.55	9.45	0.02	5.27
3	Jhapa	Paddy	198283	84350	126570.0	1501	59.56	0.56	41.83	6.31	0.59	8.33	0.05	5.27
4	Jhapa	Paddy	198384	90860	160820.0	1770	59.25	0.55	40.69	6.08	0.59	7.30	0.02	5.27
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
2095	Kapilbastu	Wheat	201718	25667	102456.0	3988	53.81	0.49	42.40	4.72	0.52	8.21	0.05	5.27
2096	Kapilbastu	Wheat	201819	25667	103935.0	4050	58.50	0.53	45.02	5.39	0.55	9.67	0.04	5.27
2097	Kapilbastu	Wheat	201920	26118	103167.0	3950	62.00	0.56	43.95	5.13	0.57	7.41	0.03	5.27
2098	Kapilbastu	Wheat	202021	26725	99085.0	3710	61.69	0.60	42.00	6.59	0.63	8.12	0.01	6.31
2099	Kapilbastu	Wheat	202122	26292	98184.0	3730	63.69	0.55	42.98	5.76	0.56	8.02	0.03	3.82

2100 rows × 14 columns

Figure 3.5.1 Data table 2100x14

As we can see, there are 2100 rows and 14 columns in our dataset. After that, we dug out the statistical entities of the data like mean, median, mode, correlation and skewness. These statistical entities helped us to get an overview of frequency distribution and biasness of the data and also how much the features are related with one another. We also explored whether the data has any duplicate, null features or not.

Since we were mainly concerned with the yield of these 5 crops in different regions of Nepal, we created a bar graph to demonstrate their respective productions.

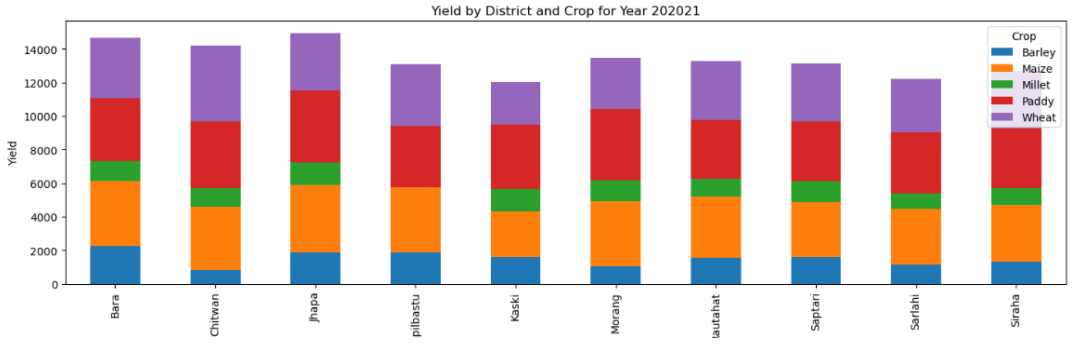

Figure 3.5.2 Crops average production per area in different districts for year 2020-2021

We were keen on learning how much the area used for cultivation influenced the yield, so we even created a scatter plot for it.

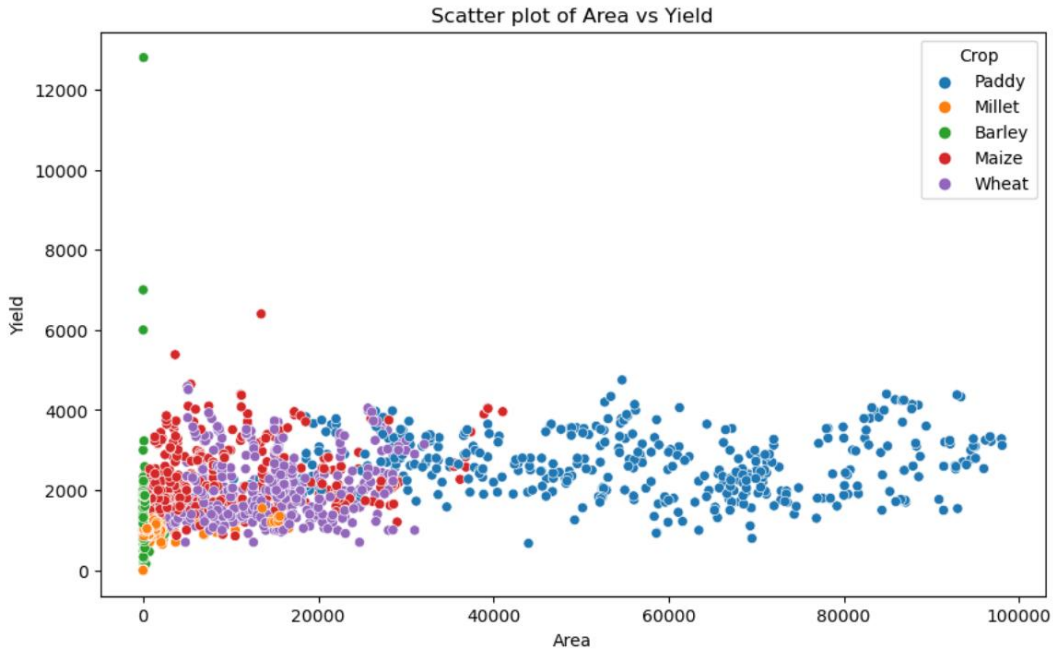

Figure 3.5.3 Scatter plot for yield vs area

To show the correlation between the parameters, we created a heatmap for the correlation matrix. By analyzing the correlation matrix, we made informed decisions about feature selection and engineering, ultimately leading to a more accurate and efficient machine learning model for predicting crop yield.

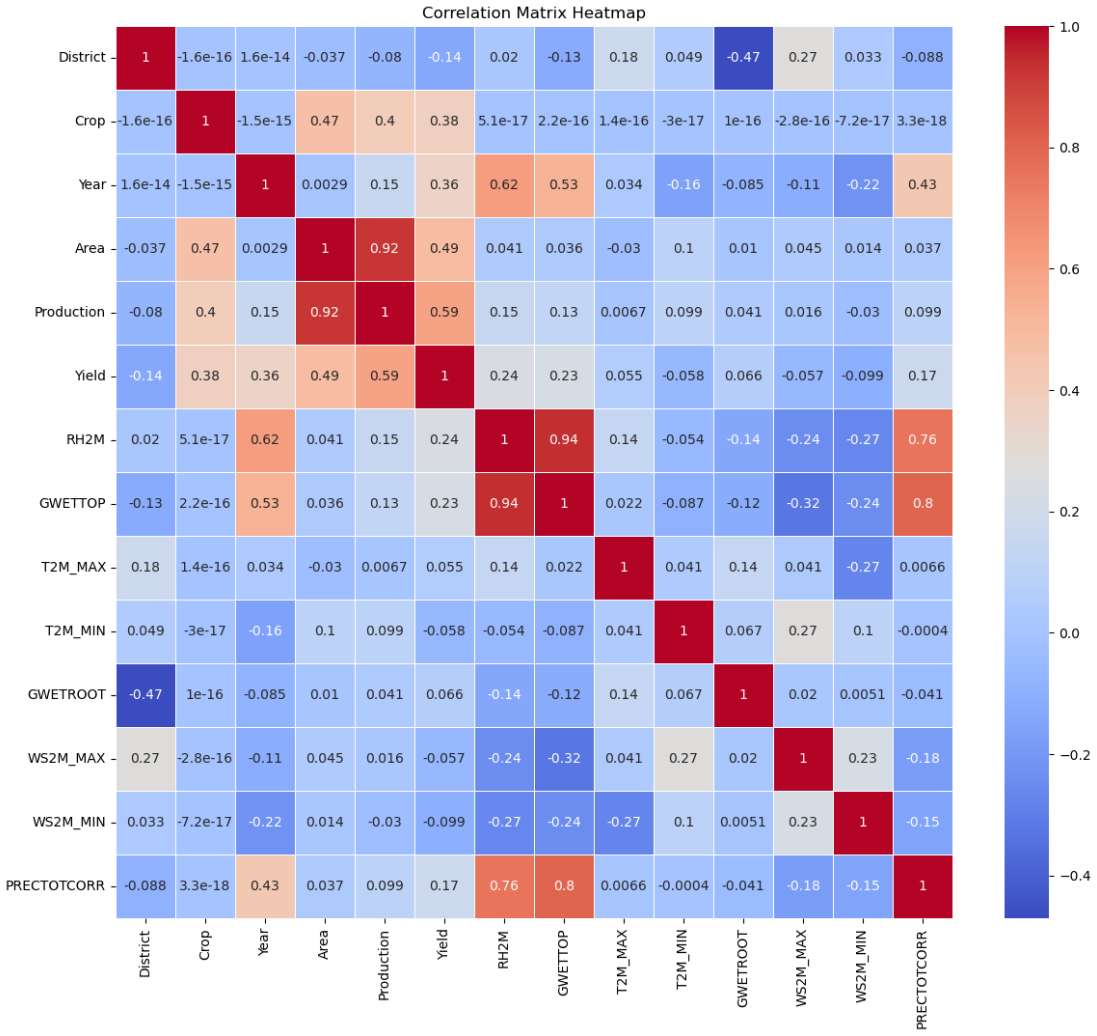

Figure 3.5.4 Heatmap to show correlation

Some parameters along with their description is given below:

Abbreviations	Feature Description
RH2M	Humidity
GWETTOP	Surface Soil moisture

T2M_MAX	Maximum temperature at 2 meters
T2M_MIN	Minimum temperature at 2 meters
GWETROOT	Root zone soil wetness
WS2M_MAX	Wind Speed at 2 Meters Maximum
WS2M_MIN	Wind Speed at 2 Meters Minimum
PRECTOTCORR	Bias corrected total precipitation
Area	Area of land for particular crop in particular district
Crop	Name of crop
District	Name of district
Year	Year in which crop is grown

Table 3.5.1 Some Parameters along with thier description

3.6 One hot encoding and normalization

Since we are dealing with categorical data, we performed one hot encoding to convert our categorical data into a numerical format that machine learning algorithms can work with. Likewise, normalization ensures that numerical features are on a similar scale.

3.7 Training and Testing Split

We then split the dataset in the ratio 80:20 of the existing data for training and the remaining for the test with random state=42 which keeps the training set and test set constant.

Figure 3.7.1 Train-test split

3.8 Model Selection and Training

Selecting appropriate statistical and machine learning models based on the nature of the data and research objectives. Here we used 6 ML models including one deep learning model to train our dataset and compare their performance. By finalizing our ml models, we split our dataset into test train segments and trained our ML models on the training dataset.

3.8.1 Arima Model:

An autoregressive integrated moving average, or ARIMA, is a statistical analysis model that uses time series data to either better understand the data set or to predict future trends. A statistical model is autoregressive if it predicts future values based on past values. For example, an ARIMA model might seek to predict a stock's future prices based on its past performance or forecast a company's earnings based on past periods. An ARIMA model can be understood by outlining each of its components as follows:

- Autoregression (AR): refers to a model that shows a changing variable that regresses on its own lagged, or prior, values.
- Integrated (I): represents the differencing of raw observations to allow the time series to become stationary (i.e., data values are replaced by the difference between the data values and the previous values).

- Moving average (MA): incorporates the dependency between an observation and a residual error from a moving average model applied to lagged observations.

ARIMA Parameters: Each component in ARIMA functions as a parameter with a standard notation. For ARIMA models, a standard notation would be ARIMA with p, d, and q, where integer values substitute for the parameters to indicate the type of ARIMA model used. The parameters can be defined as:1

- *p*: the number of lag observations in the model, also known as the lag order.
- *d*: the number of times the raw observations are differenced; also known as the degree of differencing.
- *q*: the size of the moving average window, also known as the order of the moving average.

ARIMA model has produced the following results, which illustrate the time series forecasting capabilities of the model. This output visually represents the model's ability to capture patterns and trends in the data, showcasing its effectiveness in predicting future values based on past observations. By analyzing these results, one can gain insights into the accuracy and reliability of the ARIMA model in handling

time series data.

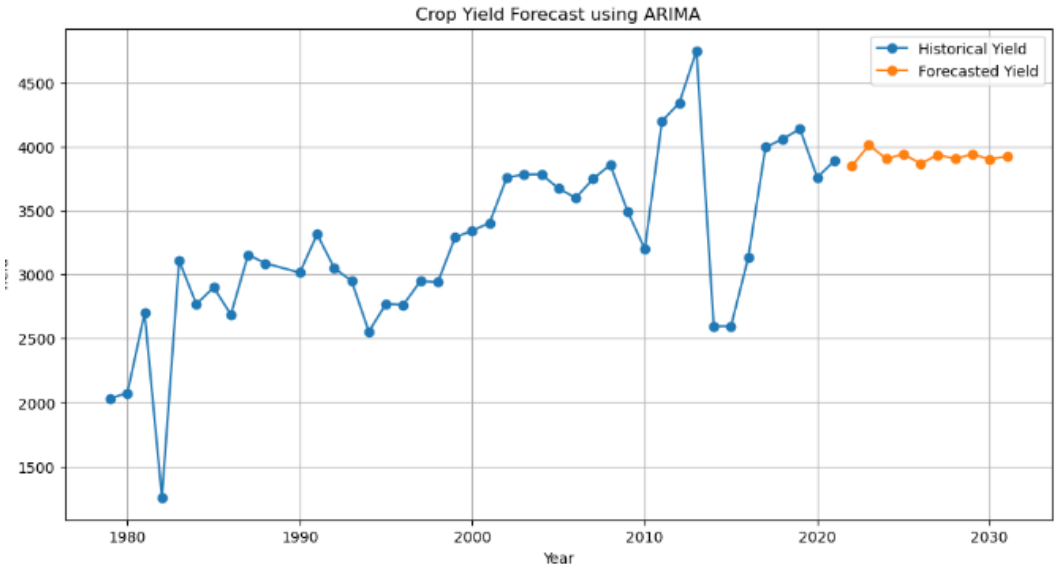

Figure 3.8.1 Arima model implementation

3.8.2 Gradient boosting

Gradient Boosting is a powerful boosting algorithm that combines several weak learners into strong learners, in which each new model is trained to minimize the loss function such as mean squared error or cross-entropy of the previous model using gradient descent. In each iteration, the algorithm computes the gradient of the loss function with respect to the predictions of the current ensemble and then trains a new weak model to minimize this gradient. The predictions of the new model are then added to the ensemble, and the process is repeated until a stopping criterion is met.

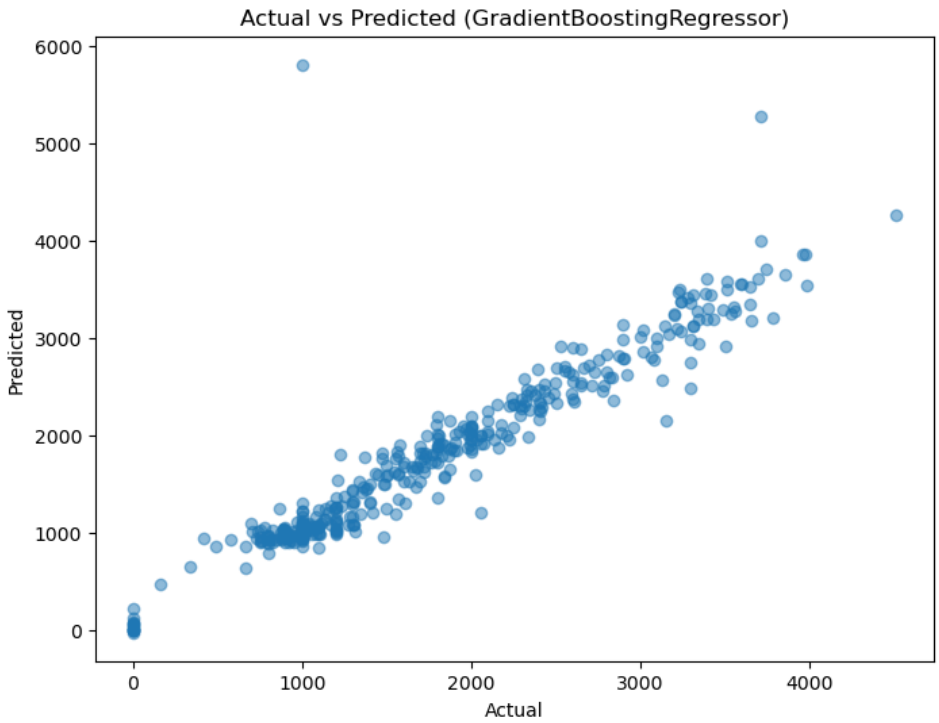

Figure 3.8.2 Performance of gradient boosting

3.8.3 Random Forest Regression

Random Forest Regression in machine learning is an ensemble technique capable of performing both regression and classification tasks with the use of multiple decision trees and a technique called Bootstrap and Aggregation, commonly known as bagging. The basic idea behind this is to combine multiple decision trees in determining the final output rather than relying on individual decision trees. Random Forest has multiple decision trees as base learning models. We randomly perform row sampling and feature sampling from the dataset forming sample from the dataset forming sample datasets for every model.

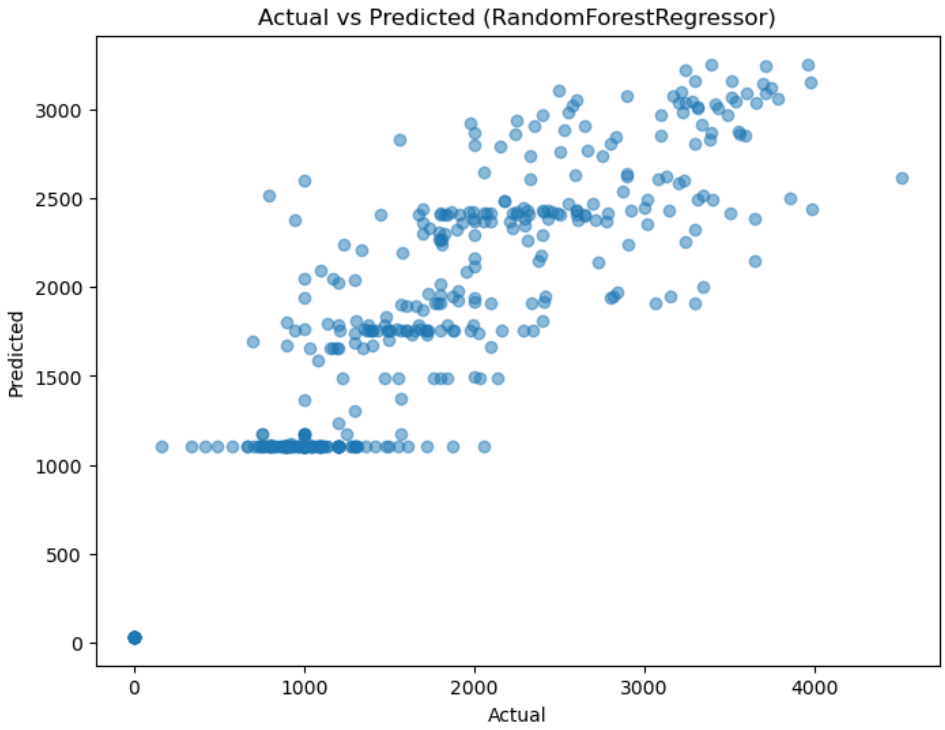

Figure 3.8.3 Performance of random forest regressor

3.8.4 Support Vector Regression:

Support vector regression (SVR) is a type of support vector machine (SVM) that is used for regression tasks. It tries to find a function that best predicts the continuous output value for a given input value. SVR can use both linear and non-linear kernels. A linear kernel is a simple dot product between two input vectors, while a non-linear kernel is a more complex function that can capture more intricate patterns in the data. The choice of kernel depends on the data's characteristics and the task's complexity.

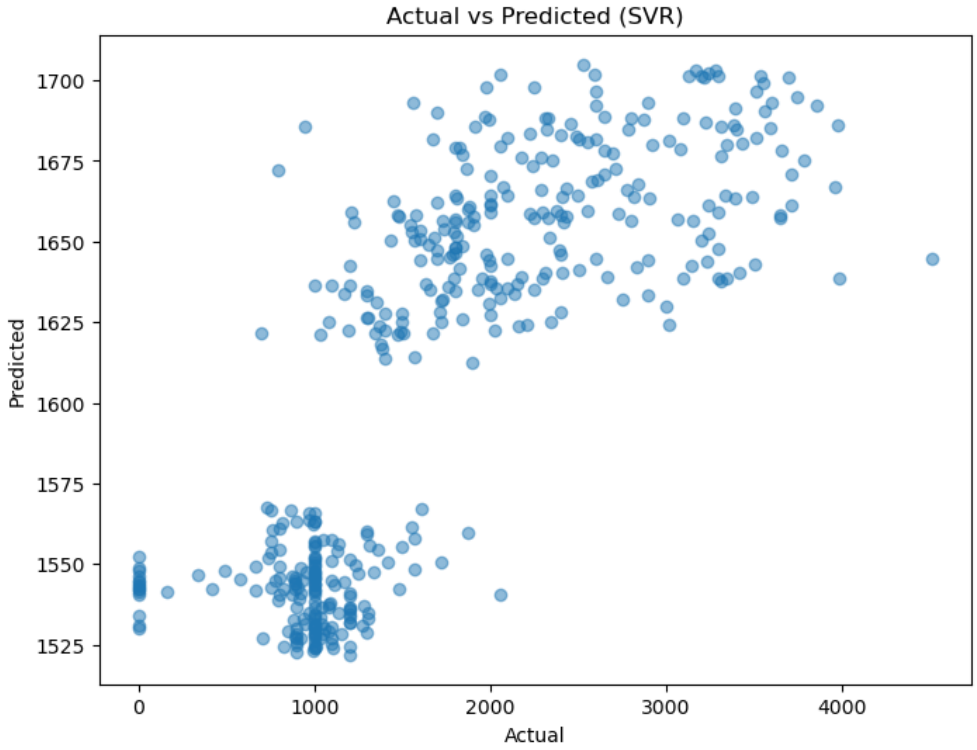

Figure 3.8.4 Performance of SVR

3.8.5 Decision Tree regression

A Decision Tree Regressor is a machine learning algorithm used for predicting continuous values by learning decision rules derived from the data features. It works by recursively splitting the dataset into subsets based on feature values, creating a tree-like model of decisions. Each internal node represents a test on a feature, each branch represents the outcome of the test, and each leaf node represents a predicted value. The goal is to minimize the variance within each subset, making the predictions as accurate as possible. Decision Tree Regressors are easy to interpret and visualize, handle both numerical and categorical data, and do not require feature scaling. However, they can be prone to overfitting, especially with deep trees, and might not generalize well to unseen data. Regularization

techniques such as pruning, setting a maximum depth, or a minimum number of samples per leaf can help mitigate overfitting.

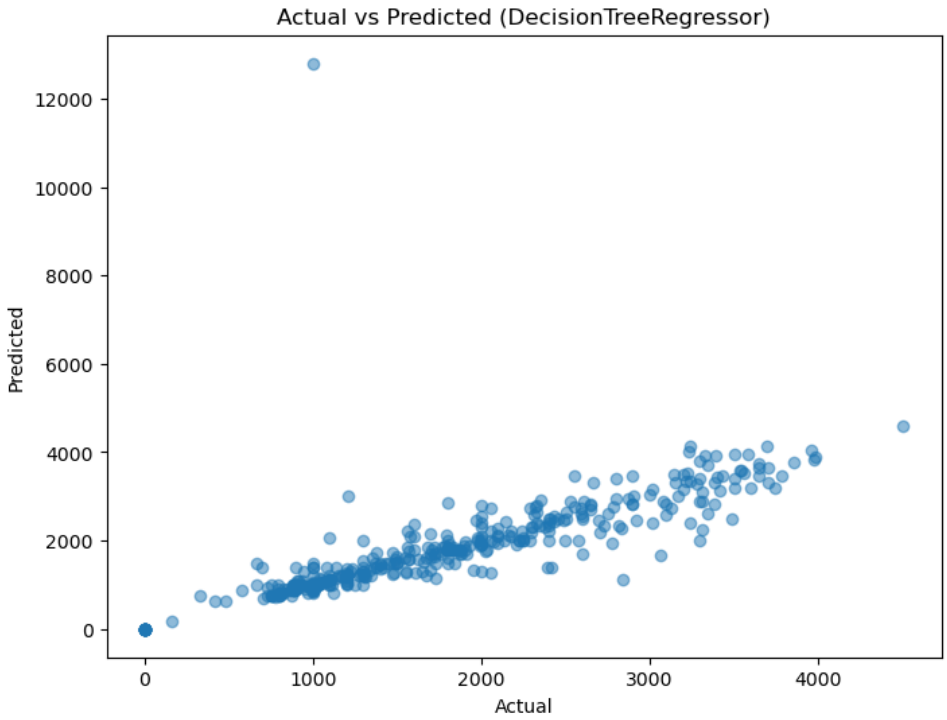

Figure 3.8.5 Performance of Decision Tree Regressor

3.8.6 Neural Network

This neural network is a feedforward neural network built using the Keras Sequential API. It consists of several layers stacked on top of each other, and it is designed for a regression task (predicting a continuous value).

Input Layer

- input_shape=(train_data.shape[1],): This defines the shape of the input data. train_data.shape[1] indicates the number of features in the training data. Each sample in the training data will have this number of features.

First Hidden Layer

keras.layers.Dense(128, activation='relu'):

Dense Layer: A fully connected layer with 128 neurons.

Activation Function: relu (Rectified Linear Unit), which introduces non-linearity to help the model learn more complex patterns.

Dropout Layer

keras.layers.Dropout(0.2):

Dropout: A regularization technique where 20% (0.2) of the neurons in the previous layer are randomly set to zero during training. This helps prevent overfitting by making the network more robust.

Second Hidden Layer

keras.layers.Dense(64, activation='relu'):

Dense Layer: Another fully connected layer, but with 64 neurons.

Activation Function: Again, relu.

Dropout Layer

keras.layers.Dropout(0.2):

Dropout: Again, 20% of the neurons are randomly set to zero during training to prevent overfitting.

Output Layer

keras.layers.Dense(1):

Dense Layer: The output layer with a single neuron.

No Activation Function: Since this is a regression task, the output is a continuous value. Therefore, no activation function is applied to the output layer.

Model Compilation

After defining the architecture, the model is compiled with specific settings:

Optimizer: adam

The Adam optimizer is an adaptive learning rate optimization algorithm that is computationally efficient and well-suited for large datasets and parameters.

Loss Function: mean_squared_error

Mean Squared Error (MSE) is used as the loss function, which is common for regression tasks. It measures the average of the squares of the errors—that is, the difference between the predicted and actual values.

Metrics: mae

Mean Absolute Error (MAE) is used as an additional metric to evaluate the model's performance. MAE measures the average absolute difference between the predicted and actual values, providing another perspective on the model's accuracy.

3.9 Model Evaluation

Model evaluation is a critical phase where the performance of the trained models is assessed using various metrics. For this project, metrics such as Mean Absolute Error (MAE), Mean Squared Error (MSE), and Root Mean Squared Error (RMSE) were employed. These metrics provide a quantitative measure of the models' accuracy and predictive power.

MAE measures the average magnitude of errors in the predictions, without considering their direction, making it a straightforward metric to interpret. MSE, on the other hand, squares the errors before averaging them, penalizing larger errors more significantly, which helps in identifying models that make large prediction errors. RMSE further builds on MSE by taking the square root of the average squared errors, providing a measure of the average prediction error in the same units as the target variable. The

comparative analysis of these metrics enabled the identification of the most reliable model, which was then selected for generating predictions.

3.10 Insights Generation and Interpretation

We analyze the model outputs to derive insights into the relationships between weather, urbanization, agriculture, and health outcomes in Nepal. Then interpret the findings in the context of sustainable development and public health, identifying key drivers and potential interventions.

3.11 System Requirement Specifications

3.12 Software Specifications

3.12.1 Programming Languages and Libraries

We used Python programming language for data analysis, machine learning modeling, and visualization. Key libraries include pandas and numpy for data manipulation, matplotlib and seaborn for data visualization, scikit-learn for machine learning.

3.12.2 Statistical and Machine Learning Tools

We implemented ARIMA (Autoregressive Integrated Moving Average) models for time series analysis and forecasting.

3.12.3 Data Visualization Tools

We used these libraries for creating visualizations such as line plots, scatter plots, heatmaps, and interactive charts to explore and communicate the findings.

3.12.4 IDE (Integrated Development Environment)

Jupyter Notebook provides an interactive environment for data analysis, visualization, and model development, enabling iterative and exploratory analysis.

3.13 Hardware Specifications

3.13.1 Computing Resources

CPU: A multi-core processor with sufficient processing power (e.g., Intel Core i5 or higher) for handling computational tasks involved in data analysis and modeling.

RAM: Adequate memory capacity (e.g., 8GB or higher) to accommodate large datasets and model training processes.

Storage: Sufficient storage space (e.g., SSD or HDD) for storing datasets, code files, and project-related documents.

3.13.2 Graphics Processing Unit (GPU)

GPU: Dedicated GPU (NVIDIA GeForce or AMD Radeon) for accelerating computation-intensive tasks, particularly for deep learning models and large-scale data processing.

CUDA or OpenCL support: Ensure compatibility with GPU-accelerated libraries and frameworks for machine learning tasks.

Chapter 4 Discussion on the Achievements

4.1 Dataset

- We were able to create a proper CSV dataset using various report of ministry of agriculture and livestock and NASA power project.
- Our dataset contains 10 district of Nepal and 5 crop annually from 1980-2022.

4.2 EDA

- We were able to create immersive EDA where user can input different features.
- Result from EDA were self-explanatory.

4.3 Performance Analysis

In this project, various machine learning algorithms like Gradient Boosting Regressor, Random Forest Regressor, Support Vector Regressor, Decision Tree Regressor and statistical model like ARIMA also deep learning algorithm like neural network are used to predict yield of crops. The accuracy for individual algorithms has to measure and whichever algorithm is giving the best accuracy, that is considered for the crop yield prediction.

For evaluating the experiment, evaluation metrics like a r2 score is considered.

4.3.1 R2-Score

The R-squared (R^2) score, also known as the coefficient of determination, is a statistical measure used in regression analysis to assess the goodness of fit of a model. It indicates the proportion of the variance in the dependent variable that is predictable from the independent variable(s). Here's a brief overview:

Definition: The R^2 score is defined as the ratio of the explained variance to the total variance. It ranges from 0 to 1, where:

- **0**: The model explains none of the variability of the response data around its mean.
- **1**: The model explains all the variability of the response data around its mean.

Formula: The R^2 score is calculated as:

$$\text{R2 score} = 1 - \frac{\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}{\sum_{i=1}^{n}(y_i - \bar{y})^2} \qquad \text{R2 score} = 1 - \frac{SS_R}{SS_M}$$

where:

- SSres is the sum of squares of residuals (the difference between the observed and predicted values).
- SStot is the total sum of squares (the difference between the observed values and the mean of observed values).

Interpretation:

- **High R^2**: Indicates a good fit, meaning the model explains a large portion of the variance in the outcome variable.
- **Low R^2**: Indicates a poor fit, meaning the model does not explain much of the variance.

S.N	Model Name	Accuracy
1.	Gradient Boosting Regressor	0.88
2.	Random Forest Regressor	0.74

3.	Support Vector Regressor	0.077
4.	Decision Tree Regressor	0.50
5.	Neural Network	0.76

Table 4.3.1 R2 Score

4.4 Comparative Analysis

Project Name	Models used	Performance metrics
An Artificial Neural Network for Predicting Crops Yield in Nepal	Artificial Neural Network (ANN)	Sum of Squares Error: 1.471
		Relative error: 0.302
Crop Yield Prediction using Machine Learning and Deep Learning Techniques	Random Forest	R2 score: 0.968
		RMSE: 0.0356
		MAE: 0.025
	SVM	R2 score: 0.898
		RMSE: 0.059
		MAE: 0.047
	Lasso regression	R2 score: 0.814

		RMSE: 0.0806
		MAE: 0.0588
	Gradient Descent	R2 score: 0.737
		RMSE:0.096
		MAE:0.0790
	LSTM	R2 score: 0.757
		RMSE: 0.501
		MAE: 0.416
Crop Yield Prediction Using Deep Neural Networks (2019)	DNN	RMSE: 0.12
Our project	Gradient Boosting Regressor	R2 score: 0.88
	Random Forest Regressor	R2 score: 0.74
	Support Vector Regression	R2 score: 0.077
	Decision Tree Regression	R2 score: 0.50
	Neural Network	R2 score: 0.76

Table 4.4.1 Comparative Analysis

Based on these scores, we can compare the performance of different models against other similar projects.

Chapter 5 Conclusion and Recommendation

5.1 Limitations

Data Quality and Coverage:

- Incomplete or inconsistent historical data may affect model accuracy.
- Focus on Nepal limits the generalizability of the model to other regions.

Feature Set:

- Excludes economic and social factors that can impact crop yield.
- Weather data may lack granularity to capture microclimatic variations within districts.

Model Limitations:

- ARIMA model might not capture complex patterns as effectively as advanced models like LSTM.
- Machine learning models used are less interpretable compared to simpler statistical models.

5.2 Future Enhancements

Data Enrichment:

- Incorporate satellite imagery for improved spatial resolution.
- Integrate economic and social data to provide a more comprehensive analysis.

Advanced Modeling Techniques:

- Implement LSTM and hybrid models for better long-term predictions.
- Use ensemble methods to enhance accuracy and robustness.

Interactive Visualization and UI:

- Develop interactive dashboards using tools like Dash or Streamlit.
- Enable real-time data updates for continuous prediction refinement.

References

- Rimal, B., Sloan, S., Keshtkar, H., Sharma, R., Rijal, S., & Shrestha, U. B. (2020). Patterns of Historical and Future Urban Expansion in Nepal. *Remote Sensing*, 12(4), 628. https://www.mdpi.com/2072-4292/12/4/628
- Ishtiaque, A., Shrestha, M., & Chhetri, N. (2017). Rapid Urban Growth in the Kathmandu Valley, Nepal: Monitoring Land Use Land Cover Dynamics of a Himalayan City with Landsat Imageries. *Environments*, 4(4), 72. https://www.mdpi.com/2076-3298/4/4/72
- GeeksforGeeks. (n.d.). Python ARIMA Model for Time Series Forecasting. https://www.geeksforgeeks.org/python-arima-model-for-time-series-forecasting/
- Abdullah, S. M. (2021). To Predict Air Pollution using Machine Learning and Arima Model. *International Journal of Engineering Research & Technology*. https://www.ijert.org/to-predict-air-pollution-using-machine-learning-and-arima-model
- ArubaCloud. (n.d.). Time Series Forecasting in Python – Part 1. https://www.arubacloud.com/tutorials/time-series-forecasting-in-python-%28part1%29.aspx
- Towards Data Science. (n.d.). What is an ARIMA Model Taking a quick peek into ARIMA modeling. https://towardsdatascience.com/what-is-an-arima-model-9e200f06f9eb
- ResearchGate. (2021). Time Series Analysis Using ARIMA Model for Air Pollution Prediction in Hyderabad. https://www.researchgate.net/publication/351770527_Time_Series_Analysis_Using_ARIMA_Model_for_Air_Pollution_Prediction_in_Hyderabad_City_of_India
- Khaki, S., & Wang, L. (2019). Crop Yield Prediction Using Deep Neural Networks. *Frontiers in Plant Science*. https://www.frontiersin.org/articles/10.3389/fpls.2019.00621/full
- Jeong, J. H., Resop, J. P., Mueller, N. D., Fleisher, D. H., Yun, K., Butler, E. E., ... & Kim, S. H. (2016). Crop Yield Prediction using Machine Learning and Deep Learning Techniques. *Environmental Modelling & Software*.
- https://www.sciencedirect.com/science/article/pii/S1364815219304407

- Aryal, J. P., Sapkota, T. B., Khurana, R., & Rahut, D. B. (2020). An Artificial Neural Network for Predicting Crops Yield in Nepal. *International Journal of Agricultural Sustainability*,19-33. http://ro.ecu.edu.au/cgi/viewcontent.cgi?article=1002&context=theses_hons
- Jha, K., Doshi, A., Patel, P., & Shah, M. (2019). Crop yield prediction using machine learning: A systematic literature review. *Computers and Electronics in Agriculture*,163,https://www.researchgate.net/publication/332514891_Crop_yield_prediction_using_machine_learning_A_systematic_literature_review
- Ministry of Agriculture and Livestock Development. (n.d.). Agriculture statistics. Government of Nepal. Retrieved June 18, 2024, from https://moald.gov.np/publication-types/agriculture-statistics/
- NASA Langley Research Center. (n.d.). POWER Data Access Viewer. NASA. Retrieved June 18, 2024, from https://power.larc.nasa.gov/data-access-viewer/