Predicting Crop Yield Using Climate Variables with a Machine Learning Approach

by
Yongchan Lee

Presented in Partial Fulfillment of the
Requirements of Senior Independent Study

Advised by
Drew Pasteur
Statistical & Data Science

Fall 2025

Abstract

Ensuring food security is a critical challenge as the global population and climate instability continue to rise. This study focuses on predicting the yields of corn, soybean, and wheat under these unstable conditions using a machine learning approach. The model utilizes six years of county-level data for five climate variables: temperature, precipitation, solar radiation, CO_2, and soil moisture. An Extreme Gradient Boosting (XGBoost) model was trained for each crop, and the SHAP (SHapley Additive exPlanations) framework was employed to ensure model interpretability.

Separate models were built for each crop, with the corn and soybean models demonstrating high predictive accuracy, achieving R^2 values of 0.76 and 0.81, respectively. In contrast, the wheat model yielded a lower R^2 value of approximately 0.5. A SHAP analysis of the high-performing corn and soybean models revealed that temperature and precipitation during the summer growing season were the most influential predictors, a finding that aligns with established agronomic principles.

The ability to predict the yield of major food crops is a critical step toward ensuring food security, especially under unstable climate conditions. The findings from this study can empower governments to formulate data-driven agricultural policies for the future, while also providing farmers with new, data-backed insights to consider when creating their planting plans.

Dedication

This thesis was made possible by my parents, who have supported me in every way and raised me with love and care for the past 25 years. Although they could not offer academic help in a field different from their own, they served as my emotional pillar. I am grateful that many of my accomplishments over the last four years were possible because I inherited their character.

Acknowledgements

I would like to express my sincere gratitude to my Independent Study (IS) advisor, Dr. Drew Pasteur, for his invaluable guidance and mentorship, which kept me on the right path throughout this research. I am also truly grateful to Dr. Matthew Mariola for providing the essential resources necessary for this study. Finally, I would like to thank The College of Wooster for granting me the wonderful opportunity to undertake this Senior IS project.

Table of Contents

List of Figures

List of Tables

1. INTRODUCTION

The human population on Earth has been steadily increasing over the past centuries. In 1800, the global population was approximately 1 billion[1]. However, as of 2022, this number has reached 8 billion, and according to a United Nations report, this increasing trend is expected to continue. Projections estimate that the population will grow to 9.7 billion by 2050 and 10.4 billion by 2080[2]. As the population continues to expand, ensuring a sustainable and secure food supply becomes an urgent necessity.

However, climate change has exacerbated global food insecurity, making it increasingly difficult to provide enough food for the growing population. One study found that between 2014 and 2019, food insecurity increased from 19.3% to 30.7% worldwide, reflecting an 11.4% increase over just five years[3]. Food insecurity threatens not only the health and well-being of millions but also the stability of economies and governments. Among various food sources, crops play a fundamental role in global food production. The crops provide sustenance for both humans and livestock. However, crops are highly sensitive to climate conditions, and climate change has led to more extreme and unpredictable weather patterns[4] [5]. As a result, predicting crop yield has become more challenging than ever before.

Despite these challenges, accurate crop yield prediction is crucial for ensuring sustainability and resilience in global food systems. From a social perspective, forecasting crop yields allows governments and organizations to anticipate and prevent famines before they occur. It ensures food distribution efforts can be effectively planned. From an economic perspective, reliable crop predictions can significantly aid policy development and agricultural decision-making, allowing farmers, investors, and policymakers to prepare for potential shortages or surpluses in advance[6].

Given the increasing unpredictability of climate conditions, reliable crop yield prediction models has become a priority. Traditional statistical or empirical models have been widely used to forecast crop yields, but they often struggle to capture the complex, nonlinear relationships between climate variables and crop productivity. Recent advancements in machine learning (ML) have introduced new approaches that leverage large-scale climate data to improve predictive accuracy. However, different studies have employed varying crops, climate variables, and modeling techniques.

1.1. Literature Review

This literature review explores existing research on crop yield prediction, focusing on three key areas: (1) the crops and climate variables used in previous studies, (2) machine learning models employed in crop yield prediction, and (3) limitations and challenges in existing studies.

1.1.1. Crops and Climate Variables Used in Previous Studies

Crop yield is influenced by various climate factors, including temperature, precipitation, CO_2 concentration, soil moisture, and extreme weather events. Different studies have examined how these variables impact specific crop types across different regions, highlighting the importance of localized climate conditions in yield prediction.

Several studies have focused on wheat and maize. Ruß et al. (2008) applied neural networks for wheat yield prediction, incorporating an amount of nitrogen fertilizer applied, vegetation index, and electrical conductivity as primary climate variables[7]. Similarly, Jeong et al. (2016) compared random forests and multiple linear regression for wheat, maize, and potato yield prediction, integrating climate, soil, photoperiod, water, and fertilization parameters[8].

Matsumura et al. (2015) explored maize yield prediction using linear regression and artificial neural network, demonstrating that temperature, precipitation, and fertilizer consumption could serve as predictive factors for wheat productivity[9].

Rice yield prediction has also been a significant area of research. Gandhi et al. (2016) studied rice yield prediction in India, applying support vector machines while incorporating precipitation, temperature, reference crop evapotranspiration, and area[10]. In a separate study, Su et al. (2017) also trained the Support Vector Machine-Based Open Crop Model (SBOCM) to predict rice production in China, integrating station information, soil information, and daily information, such as air pressure, temperature, and humidity[11].

Sugarcane, another essential crop, has been analyzed using various climate variables. Everingham et al. (2016) employed random forests for regional sugarcane yield prediction, incorporating The Agricultural Production Systems Simulator, rainfall, radiation, temperature, and Southern Oscillation Index[12]. Fernandes et al. (2017) applied neural networks ensemble to forecast sugarcane yield in Brazil, integrating normalized difference vegetation index (NDVI) as key predictor[13].

In addition to staple crops, several studies have investigated fruit yield prediction. Črtomir et al. (2012) used neural networks and image visualization techniques for early apple yield prediction, using images of apples taken from the distance of 2.0m[14]. Torgbor et al. (2023) predicted mango yield using six differenent machine learning approaches, demonstrating the influence of plant indexes, rainfall, temperature, evapotranspiration, solar radiation, and vapor pressure deficit[15].

These studies collectively highlight the importance of climate variables in crop yield prediction, reinforcing the need for high-resolution climate data to improve forecasting accuracy.

The variability in crop response to climate factors suggests that region-specific models must be developed to enhance prediction reliability.

1.1.2. Machine Learning Models Employed in Crop Yield Prediction

As climate conditions become increasingly unpredictable, researchers have turned to machine learning (ML) techniques to improve crop yield forecasting. Various ML models have been applied, ranging from traditional regression techniques to deep learning architectures, demonstrating their ability to capture complex relationships between climate variables and crop productivity.

Neural networks have been widely used for crop yield prediction due to their capacity to model nonlinear interactions between climate factors. Ruß et al. (2008), Baral et al. (2011)[16], and Cakir et al. (2014)[17] applied artificial neural networks (ANNs) to predict wheat and rice yields. You et al. (2017)[18] introduced a deep Gaussian process model for crop yield prediction, demonstrating that the deep learning technique outperform competing techniqueswhen handling large-scale remote sensing data.

Regression-based models have also been explored in several studies. Matsumura et al. (2015)[9] compared multiple linear regression (MLR) and neural networks for maize yield forecasting. Similarly, Shastry et al. (2017) used regression model to predict maize, wheat, and cotton yield, demonstrating that regression models are a suitable method for predicting yield production[19].

Tree-based models have shown promising results in crop yield prediction. Gonzalez-Sanchez et al. (2014) applied M5-prime regression trees, k-nearest neighbors (KNN), and support vector machines (SVM) for large-scale crop yield prediction[20]. Jeong et al. (2016)[8] and Everingham et al. (2016)[12] found that random forests outperformed traditional models in global

and regional crop yield forecasting, demonstrating their ability to handle high-dimensional climate data efficiently.

Ensemble models have been increasingly used to enhance prediction accuracy. Umamaheswari & Madhumathi (2024) combined multiple regression, k-nearest neighbour (KNN), support vetor machine (SVM), and random forest regressor (RF) as base models, and the study used LASSO regression the meta model to predict crop yield[21]. Shahhosseini et al. (2020) [22] utilized various machine learning models including linear regression, LASSO regression, Extreme Gradient Boosting (XGBoost), LightGBM, and random forest. Random forest features high variance and low bias, while the gradient boosting technique iteratively improves the model based on weak learners. Additionally, several two-level stacking ensemble models and an optimized weighted ensemble model were employed to enhance prediction performance. These studies suggest that ML-based models consistently outperform traditional statistical methods by capturing complex interactions between climate variables. However, model selection depends on data availability, computational resources, and regional climate variability.

1.1.3. Limitations and Challenges in Existing Studies

Several researchers have highlighted key limitations and challenges in using machine learning for crop yield prediction, particularly in relation to climate variables. Torgbor et al. (2023)[15] emphasize the issue of data quality and availability and notes that inconsistencies in grower-reported yield data and inaccuracies in remote sensing measurements create significant uncertainty. Similarly, Aldhyani et al. (2022)[6] stress that a lack of high-resolution climate and soil data limits the effectiveness of artificial intelligence models in forecasting yield, particularly in regions where comprehensive environmental datasets are unavailable.

The complexity of climate-yield relationships also presents a significant challenge. Su et al. (2017)[11] highlight that traditional statistical models, such as multiple linear regression (MLR), fail to capture the nonlinear interactions between climate variables and crop productivity. While artificial neural networks (ANNs) and support vector machines (SVMs) have shown improved accuracy, these models require large datasets and extensive computational resources for effective training. Shahhosseini et al. (2020)[22] discuss the issue of overfitting in deep learning models, which reduces their generalizability to new environments. Additionally, they note that the "black box" nature of many ML models makes them difficult to interpret.

Another challenge involves the integration of climate data into predictive models. Shahhosseini et al. (2020)[22] found that while incorporating climate variables such as precipitation, temperature, and solar radiation can improve accuracy, some studies suggest that remote sensing data alone can achieve similar results. Moreover, Su et al. (2017)[11] indicate that long-term climate projections may not align well with short-term crop yield fluctuations, reducing their effectiveness in operational decision-making. Addressing these challenges will require improved data collection strategies, more robust feature selection methods, and hybrid modeling approaches that integrate both machine learning and mechanistic crop models for enhanced accuracy and interpretability.

1.2. Study Overview

The paper aims to investigate the relationship between the yield of corn, soybean, and wheat and key climate change-related variables, including precipitation, solar radiation, CO_2 levels, soil moisture, and temperature. It seeks to predict crop yield using the machine learning method called *Extreme Gradient Boosting* (XGBoost) which have succeeded in recent studies[23]

[24] in predicting crop yield. Other than the XGBoost model, several machine learning models have been utilized, and their performance is evaluated based on R^2 and RMSE values.

This study differs from previous research in five ways:

I. **Utilizing remote sensing data instead of relying solely on ground-based observations**
II. **Applying XGBoost, a tree-based boosting model known for its effectiveness in handling structured data**
III. **Focusing on county-level data which can provide a more localized and detailed analysis**
IV. **Employing *SHapley Additive exPlanations* (SHAP) to make the machine learning model interpretable**
V. **Developing and comparing models for the three major crops primarily produced in the United States.**

I hope this study provides a solution in predicting crop yield even under the unstable climate patterns of today. The predicted values can contribute to tackling future food insecurity challenges and aid in more informed decision-making for agricultural planning and policy development.

2. METHOD/THEORY

2.1. Crop and State Selection

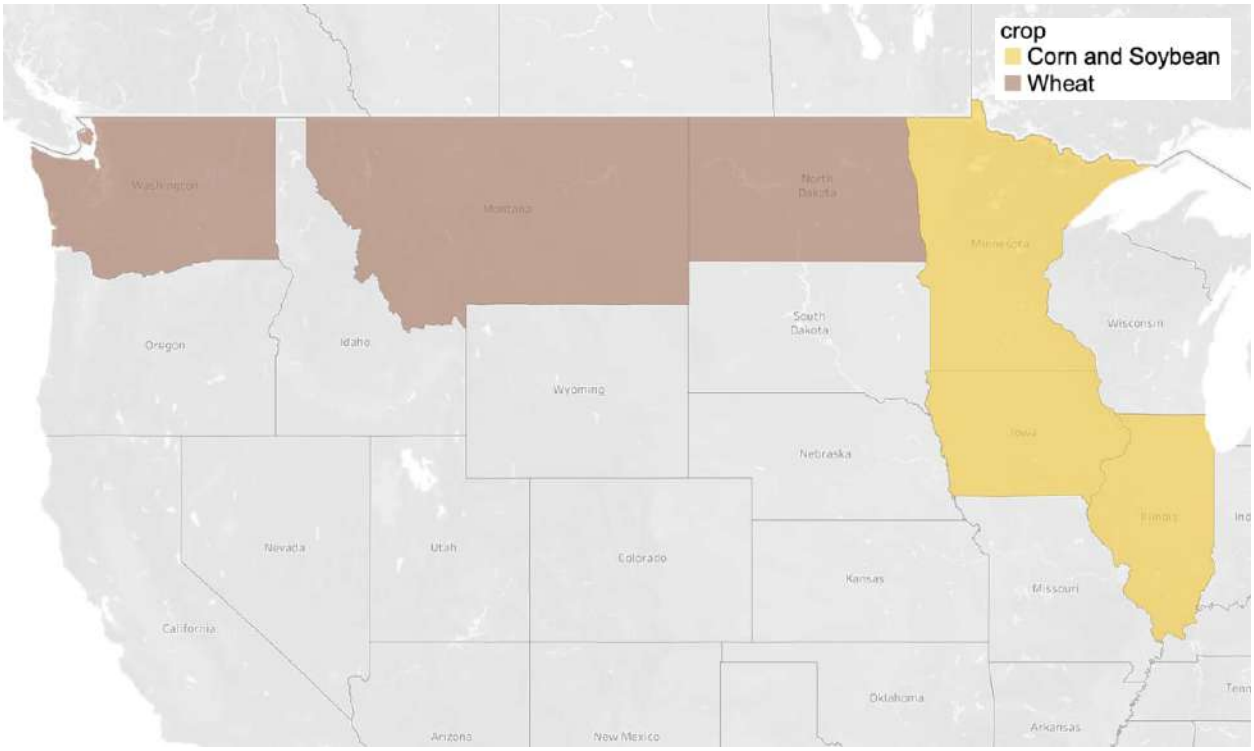


Figure 2-1. Selected States and Crops

To predict crop yield, the first step is to determine which crops will be analyzed. As mentioned in the Introduction, climate change significantly contributes to food insecurity, so the selection of crops was based on those that are primarily consumed by humans.

According to the 2023 USDA Crop Production Report, the four largest crops by production area in 2023 were corn (93 million acres), soybeans (82 million acres), hay (53 million acres), and wheat (37 million acres)[25]. Among these, hay is primarily produced for animal feed rather than human consumption. In contrast, corn, soybeans, and wheat are major staple crops directly consumed by humans or used in food production. Therefore, this study

focuses on corn, soybeans, and wheat, as they are the most widely cultivated food crops in the United States and play a critical role in addressing food security concerns.

Similar to crop selection, the selection of states was based on identifying the major producers of each crop. This process was also guided by data from the USDA Crop Production Report. Since the leading corn and soybean-producing states largely overlap, the same states were chosen for both crops: Illinois, Iowa, and Minnesota. These states are in the Midwestern United States, which serves as the primary production region for corn and soybeans.

In contrast, the top wheat-producing states are in the northwestern region of the country. Based on production data, the selected states for wheat are North Dakota, Montana, and Washington. Initially, Kansas was considered for wheat, as it is a major producer. However, due to the unavailability of crop yield data for Kansas, Washington was chosen as a replacement.

Figure 2-1 above displays the three states assigned to each crop. The states shown in yellow—Minnesota, Iowa, and Illinois—represent the regions for corn and soybean. Washington, Montana, and North Dakota, which represent the region for wheat, are shown in brown.

2.2. Feature Selection

The goal of this study is to determine whether crop yield can be predicted using climate variables. In crop yield prediction, the selection of climate variables is a critical factor. To ensure a rigorous selection process, this study references a systematic literature[26] review that analyzed many research papers related to machine learning-based crop yield prediction. This review ranked features based on how frequently they were used in different studies. However, since the review focused solely on crop yield prediction, many of the independent variables listed were not directly related to climate factors.

From the climate-relevant variables, features temperature, precipitation, and solar radiation were frequently used. In the list, these are the variables significantly affected by climate change. Additionally, two other critical climate variables, CO_2 levels and soil moisture, were included, even though they were not among the features in the literature review.

Temperature was the most frequently used variable, appearing 24 times in the literature review paper. It is a fundamental component of climate change, as global temperatures are rising[27]. In plant growth, optimal temperature ranges (0°C - 40°C) are crucial[28], as extremely high or low temperatures can inhibit plant development.

Precipitation was used nine times across studies. Climate change has introduced various precipitation patterns, including: 1) increased atmospheric moisture and intensified rainfall events and 2) dry regions becoming drier, and wet regions becoming wetter[29]. Since water is essential for plant growth, these irregular precipitation patterns can have both positive and negative effects which makes precipitation a complex but important factor[30].

Solar radiation was referenced ten times in the literature review. It is evident that there is a relationship between climate change and solar radiation. As higher temperatures trap moisture in the atmosphere, it leads to increased cloud formation, and this formation can obstruct solar radiation[31]. However, solar radiation is essential for plant growth, as it directly drives photosynthesis.

CO_2 levels were included because CO_2 is a fundamental factor for plant growth. Climate change is increasing atmospheric CO_2 levels globally[32], and CO_2 plays an essential role in photosynthesis with solar radiation as mentioned. Due to this connection, CO_2 was selected as a key climate variable.

Soil moisture represents the water content stored in the soil, which plant roots rely on for growth. Climate change has been linked to more frequent and prolonged droughts, but deep soil

layers remain relatively unaffected by short-term dryness[33]. Since plants with deep root systems interact with climate change differently, my study includes soil moisture from depths of 5 cm to 50 cm to better analyze this relationship. A more detailed explanation of the soil moisture dataset and processing methods will be provided in the Data/Process section.

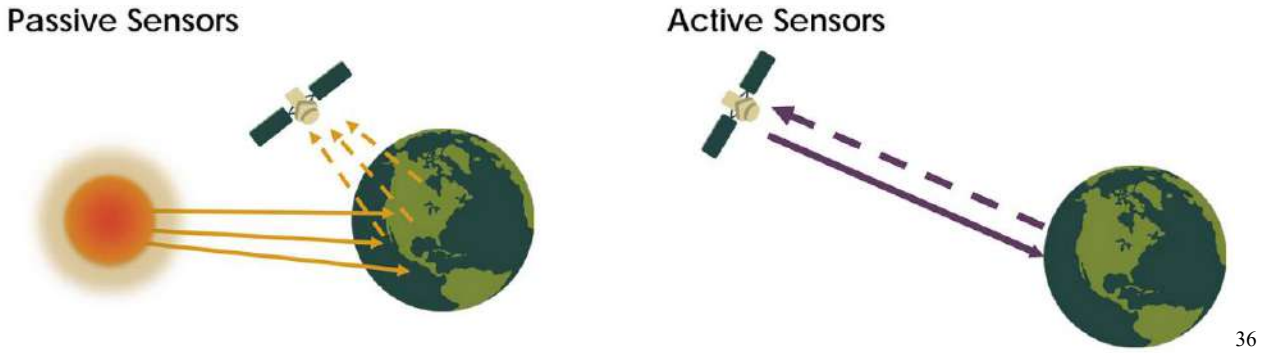
2.3. Remote Sensing

Remote sensing is a technique that acquires information about the Earth's surface without direct contact, typically using instruments like satellites. The type of data obtained through remote sensing varies depending on factors such as the type of satellite, the sensors onboard, and the satellite's orbit. This study utilizes climate data obtained from three different remote sensing methods.

Satellite orbits are categorized into three types[34]. Low Earth orbit ranges from 160 to 2,000 kilometers above the Earth's surface. An example includes polar-orbiting platforms, which rotate at a 90-degree angle to the equatorial plane. Medium Earth orbit ranges from 2,000 to 35,500 kilometers above Earth, completing an orbit in approximately 12 hours and scanning the same spot twice daily. High Earth orbit, positioned 35,500 kilometers above Earth, maintains a geosynchronous orbit, allowing it to continuously scan a specific area.

There are two primary methods for satellites to collect surface data without direct contact: passive and active[35]. Passive instruments rely on natural energy from the sun, using radiometers and spectrometers to measure reflected electromagnetic radiation from the Earth's surface, such as land or ocean. A limitation of passive instruments is their inability to penetrate cloud cover. Active instruments, in contrast, emit their own energy via radar and collect the reflected signals. Active instruments, equipped with radio detection, radar instruments,

altimeters, and scatterometers, primarily use microwaves, enabling them to penetrate cloud cover, a capability lacking in passive instruments.



[36]

Figure 2-2. Passive and Active Methods. Reproduced from [36]

The collected data is classified into levels ranging from Level 0 to Level 4, indicating the degree of processing from raw data. Level 0 data is raw, reconstructed but unprocessed at full resolution. Level 1 data is reconstructed, unprocessed at full resolution, and includes time-referenced and ancillary information. Level 2 data consist of derived geophysical variables at the same resolution and location as the Level 1 source data. Level 3 data include variables mapped on uniform space-time grid scales, usually with some completeness and consistency. Finally, Level 4 data represent the most processed data, possibly including modeled output and measurements from multiple satellites over several days.

2.4. Spatial Interpolation (Inverse Distance Weighting)

Interpolation is a mathematical data prediction method that predicts new data based on existing data. This interpolation technique can also be applied to data with spatial characteristics. An example is the data collected by the Polar Orbital platform from remote sensing. When this platform collects data, the data is not scanned globally but appears in a band shape with missing parts. This creates empty data in the two-dimensional spatial data, and spatial interpolation is

used to fill in the empty data. In this study, inverse distance weighting was used as the interpolation method.

Inverse distance weighting (IDW) is a spatial interpolation technique first proposed in 1968 by Donald Shepard's paper "A two-dimensional interpolation function for irregularly-spaced data.[37]" IDW starts with the assumption that the value of surrounding measurement points is inversely proportional to the distance from the estimation point. For example, when IDW is applied to empty spaces in CO_2 data, the CO_2 values of nearby areas contribute greatly to predicting the empty values, while the CO_2 values far from the empty spaces contribute less. In general, IDW can be expressed as the following formula:

$$Z(x) = \frac{\sum_{i=1}^{N} w_i Z_i}{\sum_{i=1}^{N} w_i} = \frac{\sum_{i=1}^{N} \frac{Z_i}{d_i^P}}{\sum_{i=1}^{N} \frac{1}{d_i^P}}$$

where $Z(x)$ is an estimated value at the target location x, Z_i is an observed value at location i, and d_i is a distance between the estimation point x and observed point i. In the formula, p represent power parameter. As p increases, the weight for distant point decreases rapidly. N is number of observed points used for interpolation.

IDW Example

Station	X (Longitude)	Y (Latitude)	Temperature (°C)
A	2	3	25
B	5	7	30
C	8	6	28
D	4	5	**?**

Table 2-1. Inverse Distance Weighting

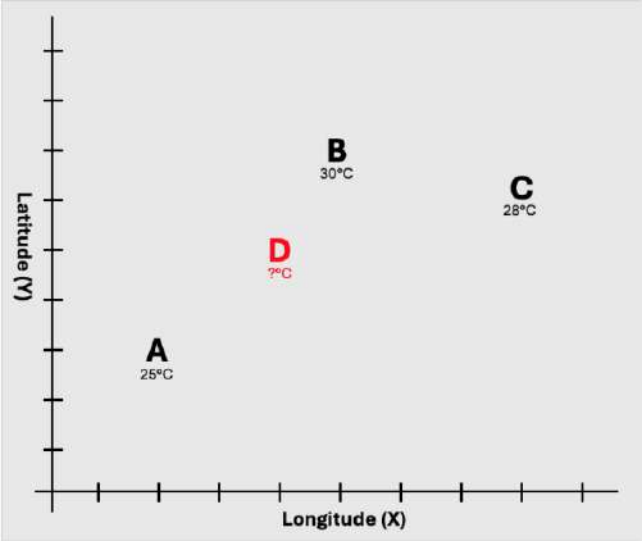


Figure 2-3. IDW Example in a Diagram

Here is a good example using IDW estimating spatial values. Assuming that there are three different stations having temperature records in °C. The task here is that finding temperature at the station D located at (4,5) using Inverse Distance Weighting (IDW). The example uses p=2. The choice of p=2 corresponds to an inverse-square relationship, meaning the influence of an observed point decreases quadratically with distance. For example, a point that is twice as far away from the target has only one-fourth the influence.

The calculation will be divided into four steps:

STEP #1: Compute the Distance of Each Station to D

Using the Euclidean distance formula d =$\sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$:

- $d_A = \sqrt{(4-2)^2 + (5-3)^2} = \sqrt{4+4} = 2.83$
- $d_B = \sqrt{(4-5)^2 + (5-7)^2} = \sqrt{1+4} = 2.24$
- $d_C = \sqrt{(4-8)^2 + (5-6)^2} = \sqrt{16+1} = 4.12$

STEP #2: Compute the Weight

Using IDW formula $w_i = 1/d_i^P$ with $P = 2$:

- $w_A = 1/2.83^2 = 1/8 = 0.125$
- $w_B = 1/2.24^2 = 1/5 = 0.2$
- $w_C = 1/4.12^2 = 1/17 = 0.0588$

STEP #3: Normalize the Weights

Total wight sum $1/\sum_{i=1}^{N} w_i$:

- $W = w_A + w_B + w_C = 0.125 + 0.2 + 0.0588 = 0.3838$

Normalized weights:

- $W_A = w_A/W = 0.125/0.2828 = 0.3257$
- $W_B = w_B/W = 0.2/0.2828 = 0.5212$
- $W_C = w_C/W = 0.0588/0.2828 = 0.1531$

STEP #4: Compute the Estimated Temperature at the Station D:

- $TD = (W_A \times T_A) + (W_B \times T_B) + (W_C \times T_C) = (0.3257 \times 25) + (0.5212 \times 30) + (0.1531 \times 28) = 8.14 + 15.64 + 4.29 = 28.07°C$

When using IDW, the parameter p and the number of observed data points N are key factors that significantly affect the accuracy of the estimated values[38]. In this study, similar to the example, IDW parameter value of $p = 2$ was used. A $p = 2$ assigns higher influence on closer points while giving less weight to farther points, ensuring that nearby observations contribute more to the estimated value. Additionally, unlike the example which used 3 number of data, $N = 5$ was used in the study. Instead of selecting five points randomly, five nearest observations were extracted based on the location of Z(x).

2.5. Machine Learning Method

Machine learning is "the scientific study of algorithms and statistical models that computer systems use to perform a specific task without being explicitly programmed.[39]" In other words, it is a technique that enables computers to learn and handle data more efficiently. Machine learning primarily consists of four types of algorithms: Supervised Learning, Unsupervised Learning, Semi-Supervised Learning, and Reinforcement Learning. Supervised Learning trains a machine to match new data with appropriate labels based on given data and labels. Unsupervised Learning provides only data without labels, allowing the machine to discover meaningful structures within the given data. Semi-Supervised Learning, a hybridization of supervised and unsupervised learning, operates on both labeled and unlabeled data. Reinforcement Learning introduces a reward system, allowing the machine to learn optimal behavior by receiving rewards for correct actions[40].

This study uses a supervised learning algorithm, where both data and labels are present, to train a machine to find appropriate labels for new data. Supervised learning is further divided into two categories based on whether the labels are continuous or discrete. If the labels are continuous, it is a regression problem; if they are discrete, it is a classification problem. This study uses a continuous label, crop yield, making it a regression problem.

There are various types of models that handle regression problems. This study uses Extreme Gradient Boosting, or XGBoost, which is based on decision trees. To fully explain XGBoost, it is necessary to understand what Ensemble Learning is, what Boosting is within Ensemble Learning, and what Gradient Boosting is.

2.5.1. Decision Tree and Ensemble Learning

Ensemble Learning can be used with any algorithm. Among them, the model used in this study is a tree-based model, XGBoost, so I will briefly explain decision trees and then explain tree-based ensemble learning.

A Decision Tree is designed to mimic the shape of a tree, and its main components include nodes and branches. Important steps include splitting, stopping, and pruning. Nodes have three types: Root nodes, decision nodes, and leaf nodes.

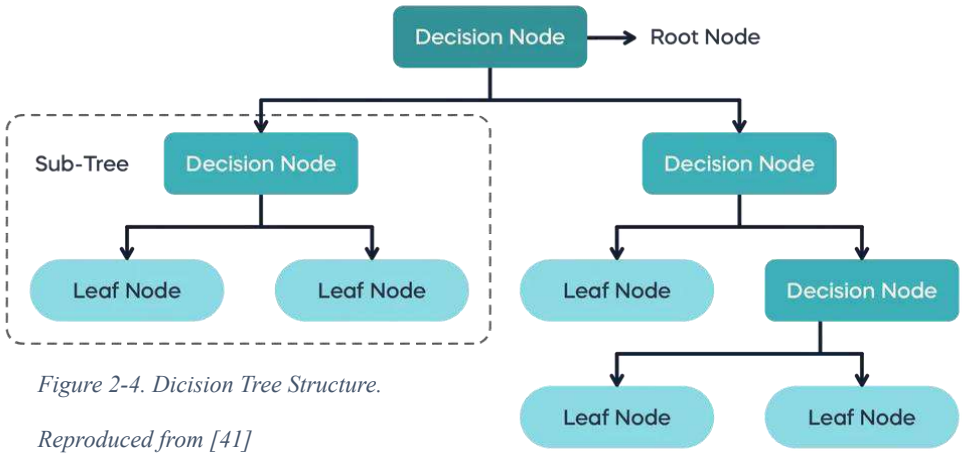


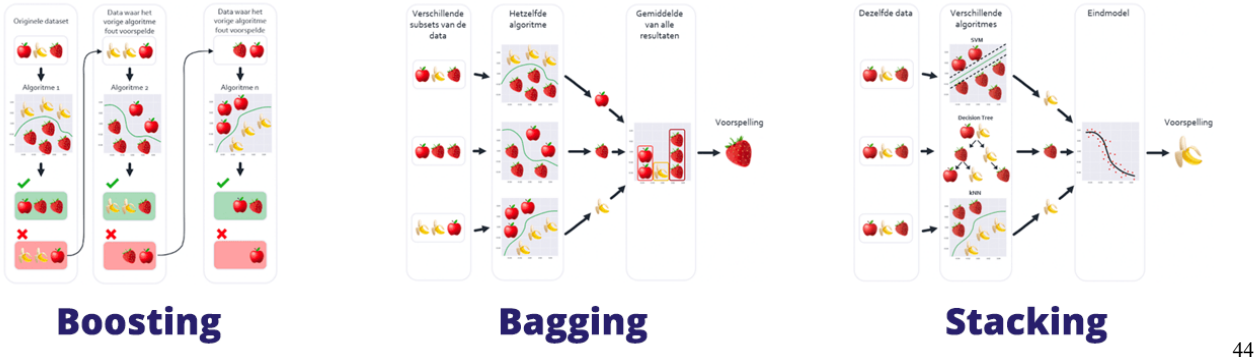
Figure 2-4. Dicision Tree Structure.

Reproduced from [41]

41

Branches connect the nodes. Splitting is an essential step to move from one node to another, and the variable most related to the labeled data, or Y, is used for splitting. Stopping can vary in amount depending on the tree's complexity and robustness. Proper stopping can prevent an overfitted tree. Pruning is an alternative to the stopping step, preventing an overly fitted tree by cutting branches from a tree created without stopping[42]. Trees can be used for classification or regression problems. In classification problems, the result is the most frequent discrete target in the leaf node, and in regression problems, the result is the average of the continuous target values in the leaf node. However, even with stopping and pruning, small dataset is prone to an overfitting problem[41]. This challenge led to the concept of ensemble learning.

Ensemble Learning is 'the combination of multiple inducers to solve a particular machine learning task.[43]' In the context of decision trees, the ensemble learning is a method of gathering and summarizing results from multiple trees rather than relying on a single tree's result. There are several methods of ensemble learning, including Bagging, Boosting, and Stacking.



[44]

Figure 2-5. Structure of Ensemble Methods (Boosting, Bagging, and Staking). Reproduced from [44]

Bagging, also known as Bootstrap aggregating, is a method that creates multiple trees, or predictors, and uses the combined results from these trees. One important aspect of bootstrap aggregating is that not all trees are trained on the same dataset. Instead, each tree is assigned a new dataset through bootstrapping, and data used in one tree can be used again when creating other trees. In classification problems, the result is the discrete target that appears most frequently among the trees' results. In regression problems, the result is the average of all the trees' results[45]. One example of bagging model is a Random Forest.

Unlike bagging, boosting does not create independent trees. Instead, trees are created sequentially. As trees are created sequentially, each subsequent tree reflects the weights of the previous trees and compensates for the mistakes made by those trees[46]. An example of a boosting model is XGBoost.

Stacking is a method that stacks multiple models rather than using a single model. The main concepts are base models and a meta-model. Base models are trained on the same data but

with different algorithms, and their results are then used as new inputs for the meta-model, which performs the final prediction[47]. For example, if random forest and XGBoost are used as base models, the prediction values from these two models are then used as inputs for a meta-model, such as linear regression, to create the final prediction.

2.5.2. Boosting

While briefly explained in section 2.5.1, I would like to elaborate on how the boosting algorithm works. Boosting is a concept first proposed by Schapire[48], and its key features are weak learners (base learners) and strong learners. In a single sentence, the boosting algorithm aims to create a strong learner through the combination of weak learners. To understand how a strong learner is created, it is necessary to understand the boosting algorithm.

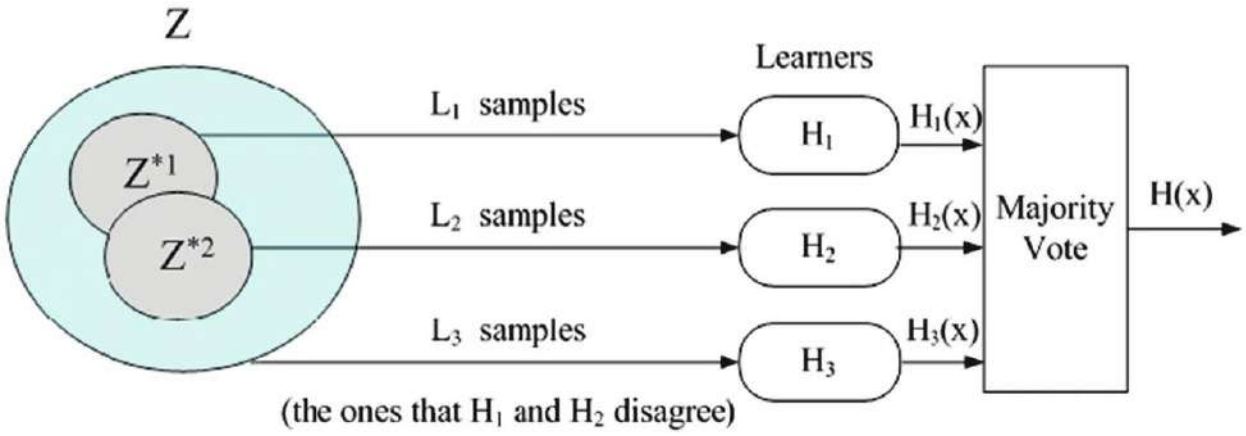


Figure 2-6. Basic Structure of Boosting Technique. Reproduced from [49]

Initially, the dataset is divided into three subsets: Z_1, Z_2, and Z_3, without replacement. A base learner H_1 is trained on Z_1, and predictions are made. Instances that are incorrectly predicted are included in Z_2. Another base learner H_2 is trained on Z_2. Predictions are made for H_1 and H_2 using all instances in Z_1 and Z_2, and only the instances where H_1 and H_2 disagree are included in Z_3. After training H_3 on Z_3, predictions are made for H_1, H_2, and H_3 on all data, and

the final prediction is made through a majority vote[49]. The figure above could be helpful to understand the algorithm.

This process is repeated, allowing the base learners to continue learning from incorrectly predicted instances, ultimately creating a strong learner. However, the boosting algorithm has a significant limitation that it is not efficient in small dataset due to the characteristic of no displacement[46]. To address this challenge, Adaptive Boosting, which increases weights for misclassified data points and allows the repeated use of datasets, was initially proposed. Furthermore, a concept called Gradient Boosting emerged, which focuses on minimizing residual errors sequentially instead of simply weighting on the critical points.

2.5.3. Gradient Boosting

This study addresses a regression problem, predicting continuous crop yield. However, as explained earlier, previous studies have shown how the Boosting algorithm works for classification problems. But when looking at the operational principles for regression, previous studies only describe Gradient Boosting, which uses residuals as weights. Therefore, the section 2.5.3 will aim to clarify how Gradient Boosting works for regression problems, a concept not fully covered in 2.5.2.

Unlike the normal Boosting technique that trains base learners on the disagreed instances, Gradient Boosting focuses on ensuring that each subsequent base learner has a smaller residual than the previous base learner, where the residual is the difference between actual target value and predicted target value.

Gradient Boosting[50] takes the following inputs and can be explained in five steps and:

- $(x_i, y_i)_{i=1}^{N}$, where x_i = input data and y_i = target value
- M = number of iterations

- Ψ(y, f): Loss Function (Gaussian L_2 Loss Function for Regression)
- h(x, θ): Base Learner (single simple tree)

STEP 1: Initialize f_0 as a constant

The loss function for the regression model is used as Gaussian L_2 Loss Function[51].

$$L(y, F(x)) = \frac{1}{2}\sum_{i=1}^{N}(y_i - F(x_i))^2,$$

where y_i = actual value and F = predicted value. When setting the initial model f_0, the optimal constant value that minimizes the loss function must be found, and the optimal constant value can be found by setting the first derivative of the Loss Function equals to zero. Solving this, we get the initial value f_0 is the mean of all target values y_i.

$$\frac{\partial L}{\partial F} = \frac{1}{2}\sum_{i=1}^{N}(y_i - F(x_i))^2 = 0 \qquad \rightarrow \qquad f_0 = \frac{1}{N}\sum_{i=1}^{N} y_i$$

STEP 2: Compute the Gradient

In this step, the gradient of the loss function is calculated. To find the gradient of the loss function, Loss Function must be differentiated.

$$g_t(x) = \frac{\partial \Psi\big(y, F(x)\big)}{\partial F(x)} = \frac{\partial}{\partial F(x)}\left(\frac{1}{2}\sum_{i=1}^{N}(y_i - F(x_i))^2\right) = -\sum_{i=1}^{N}(y_i - F(x_i))$$

The gradient value obtained by differentiation is negative of residual, which means gradient = -(y_{actual} - $y_{predicted}$).

STEP 3: Train a new base learner

The new base learner is trained in the direction of the loss function's gradient, residual.

$$h_t(x, \theta_t) \approx -g_t(x) = residual$$

STEP 4: Find the optimal step-size p_t, or learning rate

The optimal learning rate can be found using the following formula each time a new base learner is added. The process can lead to faster creation of a strong learner.

$$p_t = \arg min_p \sum_{i=1}^{N} \Psi\left[y_i, F_{t-1}(x_i) + p \cdot h(x_i, \theta_t)\right]$$

STEP 5: Update the model

The base learner trained in STEP 3 is added to the existing model to update it. The update formula is as follows:

$$F_t = F_{t-1} + p_t \cdot h(x, \theta_t)$$

Gradient Boosting Example

To further illustrate the concept of Gradient Boosting described above, I will explain it with a simple example for easier understanding.

Exercise Hour	Quality of Diet	Gender	Health Score
10	High	M	85
5	Low	F	60
8	Medium	F	72
12	High	M	90
6	Medium	M	75
3	Low	F	50

Table 2-2. Gradient Boosting Example

For the initial prediction, which is f_0 as a constant, the mean of target values is calculated: $f_0 = (85 + 60 + 72 + 90 + 75 + 50)/6 = 72$. Then, residuals are calculated as followed in the STEP2.

Exercise Hour	Quality of Diet	Gender	Health Score	f_0	Residual
10	High	M	85	72	13
5	Low	F	60	72	-12
8	Medium	F	72	72	0
12	High	M	90	72	18
6	Medium	M	75	72	3
3	Low	F	50	72	-22

Table 2-3. Gradient Boosting Example with Initial Prediction and Residual

A base learner is a simple model, such as a single simple tree, and this h(x) model is trained based on the residual, as explained in STEP 3. Let's assume that, in this example, the base learner is trained with two rules:

1. If Gender = F and if exercise hours < 6, then $h_1(x)$ = -17; if exercise hours >= 6, then h(x) = -2.
2. If Gender = M and Quality of Diet is not high, then $h_1(x)$= 3; if Quality of Diet is high, then $h_1(x)$= 15.

The values of $h_1(x)$ mentioned earlier are the average residual values of the corresponding samples. For example, the residuals of sample whose gender is F and exercise hour < 6 are -12 and -22. The average of -12 and -22 is -17 as shown in the calculation.

Exercise Hour	Quality of Diet	Gender	Health Score	Residual	$h_1(x)$
10	High	M	85	13	15
5	Low	F	60	-12	-17
8	Medium	F	72	0	-2
12	High	M	90	18	15
6	Medium	M	75	3	3
3	Low	F	50	-22	-17

Table 2-4. Gradient Boosting Exmaple with Residual and Trained Tree from the Residual

In STEP 4, the learning rate is typically optimized in each iteration using a specific formula. However, in this example, to explain the fundamental concept of Gradient Boosting, limitations of the number of iterations to 1 and the initial learning rate to 0.5 are applied. Skipping STEP 4 and proceeding directly to STEP 5, the model is updated, using the formula: $F_1 = F_0 + 0.5 \cdot h(x)$.

Exercise Hour	Quality of Diet	Gender	Health Score	Exercise Hour	f_0	f_1
10	High	M	85	10	72	79.5
5	Low	F	60	5	72	63.5
8	Medium	F	72	8	72	71.0
12	High	M	90	12	72	79.5
6	Medium	M	75	6	72	73.5
3	Low	F	50	3	72	63.5

Table 2-5. Gradient Boosting Example with the Prediction Values from the Updated Model

As the table shown, compared to the f_0 values, the $f_1$values got closer to the actual health scores. The reason for this significant improvement in just one iteration is that relatively high

learning rate was set. In practice, it is crucial to find an optimal learning rate that is neither too large nor too small to ensure effective model convergence.

2.5.4. Extreme Gradient Boosting (XGBoost)

Extreme Gradient Boosting (XGBoost)[52] has a similar algorithmic structure to the Gradient Boosting described in 2.5.3. After setting the initial prediction value, it calculates the difference between the current prediction and the actual value (residual or gradient). A new tree is then trained to reduce this difference, and this process is repeated to build a strong learner.

Despite sharing the same structure, XGBoost achieves better results than traditional Gradient Boosting due to key enhancements such as regularization and use of Hessian (second derivative of the loss function). Now, let's explain XGBoost using mathematical formulations.

STEP 1. Define the basic structure of XGBoost as an ensemble of trees.

$$\hat{y}_i = \phi(x_i) = \sum_{k=1}^{K} f_k(x_i)$$

Where $\hat{y}_i$ is a final predicted target values of the sample x_i, K is the number of base learners, and $f_k(x_i)$ is the prediction of k^{th} tree. This basic structure implies that XGBoost's final prediction is the combination of k numbers of trees.

STEP 2. Define the loss function used for optimization.

$$l(y, \hat{y}_i) = \frac{1}{2}\sum_{i=1}^{N}(y_i - \hat{y}_i)^2$$

$$1.\ \frac{\partial L}{\partial \hat{y}} = \frac{1}{2}\sum_{i=1}^{N}\left(y_i - \hat{y}_i\right)^2 = 0 \qquad \rightarrow \qquad f_0 = \frac{1}{N}\sum_{i=1}^{N} y_i$$

$$2.\ g_i = \left[\frac{\partial l(y_i, F(x_i))}{\partial F(x_i)}\right] \qquad 3.\ h_i = \left[\frac{\partial^2 l(y_i, F(x_i))}{\partial F(x_i)}\right]$$

As mentioned in 2.5.3, $l\left(y, \hat{y}_i\right)$ is the formula for the loss function. In Gradient Boosting, the loss function is used for two reasons: 1. Initializing model f_0 and 2. Calculating the gradient (first derivative of the function). However, in XGBoost, the loss function is used for three reasons: 1. Initializing model f_0, 2. Calculating the gradient g_i, and 3. Calculating the Hessian h_i (second derivative of the function).

STEP 3. Extend the loss function with a regularization term to prevent overfitting.

$$L(\phi) = \sum_i l\left(\hat{y}_i, y_i\right) + \sum_k \Omega\left(f_k\right), \qquad where\ \Omega(f_k) = \gamma T + \frac{1}{2}\lambda \|w\|^2$$

Where $L(\phi)$ is a new formula which has the loss function term $l\left(\hat{y}_i, y_i\right)$ and a regularization term $\Omega(f_k)$. In the regularization term, γ represents a penalty for the number of leafs, where T indicates the number of tree leaf. As T increases, the model becomes more complex, so γ is used to control tree growth at an appropriate level. λ represents a L2 Regularization, and w is a weight of the leaf. By adding the $\|w\|^2$ term, it prevents the leaf node weights from becoming excessively large. The, by multiplying by λ, it prevents the weights from becoming overly large, which helps prevent the model from overfitting.

STEP 4. Add New Tree $f_t(x_i)$ on the Loss Function with Regularization

Now, the loss function with regularization has been defined. The next step is to add a new tree $f_t(x_i)$ to the model to minimize the loss. The objective function at step t is:

$$L^{(t)} = \sum_{i=1}^{n} l\left(y_i, \hat{y}_i^{(t-1)} + f_t(x_i)\right) + \Omega(f_t)$$

Since directly optimizing this function is complex, second-order Taylor expansion is used. Taylor expansion is a mathematical method that approximates a function $f(x)$ near a specific point using derivatiaves. By using first and second derivatives of the loss function $l\left(y_i, \hat{y}^{(t-1)}\right)$, we can anticipate faster optimization and more accurate approximation.

$$L^{(t)} \simeq \sum_{i=1}^{n} \left[l\left(y_i, \hat{y}^{(t-1)}\right) + g_i\, f_t(x_i) + \frac{1}{2} h_i\, {f_t}^2(x_i) \right] + \Omega(f_t)$$

where g_i and h_i are the gradient (first derivative) and Hessian (second derivative) of the loss function as shown in the STEP 2. By removing the constant term l, simplified loss function is obtained:

$$\tilde{L}^{(t)} = \sum_{i=1}^{n} \left[g_i\, f_t(x_i) + \frac{1}{2} h_i\, {f_t}^2(x_i) \right] + \Omega(f_t)$$

Expanding the regularization term $\Omega(f_t)$:

$$\tilde{L}^{(t)} = \sum_{i=1}^{n} \left[g_i\, f_t(x_i) + \frac{1}{2} h_i\, {f_t}^2(x_i) \right] + \gamma T + \frac{1}{2}\lambda \sum_{j=1}^{T} {w_j}^2$$

$$= \sum_{j=1}^{T} \left[\left(\sum_{i \in I_j} g_i \right) w_j + \frac{1}{2} \left(\sum_{i \in I_j} h_i + \lambda \right) w_j^2 \right] + \gamma T$$

The optimal leaf weight for each node is derived as:

$$w_j^* = -\frac{\sum_{i \in I_j} g_i}{\sum_{i \in I_j} h_i + \lambda}$$

STEP 5. Compute the Optimal Tree Structure

After defining the new tree's structure and computing the leaf weights, determining how well the new tree reduces the loss is required. The optimized loss function is shown below. The equattion ensures that the tree structure is optimized to minimize the loss while preventing excessive complexity.

$$\tilde{L}^{(t)}(q) = -\frac{1}{2}\sum_{j=1}^{T}\frac{\left(\sum_{i\in I_j} g_i\right)^2}{\sum_{i\in I_j} h_i + \lambda} + \gamma T$$

STEP 6. Compute the Optimal Split

To improve the performance of the new tree, finding the best way to split the nodes is needed. The loss reduction (gain) caused by a split is computed as:

$$L_{split} = \frac{1}{2}\left[\frac{\left(\sum_{i\in I_L} g_i\right)^2}{\sum_{i\in I_L} h_i + \lambda} + \frac{\left(\sum_{i\in I_R} g_i\right)^2}{\sum_{i\in I_R} h_i + \lambda} - \frac{\left(\sum_{i\in I} g_i\right)^2}{\sum_{i\in I} h_i + \lambda}\right] - \gamma$$

Where I_L and I_R are the sets of data points assigned to the left and right child nodes after the split. The split that maximized the L_{split} will be selected.

STEP 7. Update the Model

Once the new tree has been trained and its split optimized, the model is updated iteratively. The new model function at step k is, where p is the learning rate:

$$f_k(x) = f_{k-1}(x) + p\, L^{(t)}$$

STEP 8: Repeat Until Convergence

At the end of the training, which means a predefined number of trees K is reached, and the loss function steps improving significantly, the final prediction for a sample x_i is calculated.

This equation shows how XGBoost is an ensemble boosting model that sums the outputs of multiple trees to make a final prediction.

$$\hat{y}_i = \phi(x_i) = \sum_{k=1}^{K} f_k(x_i)$$

2.6. Hyperparameter Optimization (Optuna)

In section 2.5, XGBoost mathematically explained, and various hyperparameters are observed. The easiest way for users to utilize this model is through code, and properly defining these hyperparameters is crucial. Hyperparameter tuning is essential when using machine learning models, as it significantly impacts performance. However, the challenge lies in finding the optimal set of hyperparameters. Manually trying out all possible combinations is nearly impossible. This is where hyperparameter optimization comes into play. Various optimization techniques exist, and this study employs Optuna, an automated hyperparameter optimization framework. Optuna differentiates itself from other optimization methods through three key design principles[53]: 1) Define-by-run API, 2) Efficient sampling and Pruning Mechanism, and 3) Scalable and Versatile System that is easy to setup.

Define-by-run API

Optuna adopts a Define-by-run approach, allowing for the dynamic construction of the hyperparameter search space. Traditional optimization methods follow a Define-and-run approach, where the search space must be predefined before execution. In contrast, Optuna's Define-by-run design enables the search space to be dynamically structured during runtime. This concept becomes clearer when viewed in python code[50].

For example, suppose we want to optimize the number of trees `n_estimators` in XGBoost. In traditional optimization methods, we would define a fixed parameter grid like:

```
param = {'n_estimators': [10, 50, 100, 200, 500]}
```

This setup restricts the search to only five predefined values, meaning the optimal number of trees is chosen from this limited set. However, with Optuna, the same task is defined:

```
param = {"n_estimator": trial.suggest_int('n_estimators', 10, 500)}
```

This allows dynamic exploration of the search space, where Optuna automatically optimizes and selects the best value for `n_estimators` from any integer between 10 and 500 across 20 trials.

Efficient sampling and Pruning Mechanism

There are two types of sampling methods: relational sampling, which utilizes correlations between parameters, and independent sampling, where parameters are treated independently. Optuna employs both sampling methods. It initially performs parameter exploration using Tree-structured Parzen Estimator (TPE), an independent sampling method based on Bayesian Optimization. Then, Optuna continues the search using Covariance Matrix Adaptation Evolution Strategy (CMA-ES), a relational sampling method that leverages parameter correlations.

Pruning is essential in decision trees, as it allows for more efficient resource utilization. The pruning process consists of two phases: 1) Periodically monitoring intermediate objective values and 2)Terminating trials that fail to meet predefined conditions. In Optuna, the ‘report API’ is responsible for monitoring, while the ‘should_prune API’ handles the early termination of unpromising trials. Additionally, Optuna provides the Asynchronous Successive Halving Algorithm (ASHA) for efficient pruning.

A useful analogy for ASHA is a talent competition where multiple contestants perform simultaneously, rather than one at a time. The advantage of this format is that judges can continuously evaluate performances, quickly eliminating weaker contestants and focusing on stronger ones, leading to a faster selection of the winner. Similarly, in the context of XGBoost, ASHA enables faster identification of the optimal tree structure in a distributed environment, improving the efficiency of hyperparameter optimization[50].

Scalable and Versatile System that is Easy to Setup

Optuna is a scalable and flexible optimization software designed to meet user requirements across various environments, including local and distributed systems. It sets the standard for next-generation optimization frameworks by offering customizable storage backends, compatibility with interactive analysis tools, real-time visualization, and easy installation, all while prioritizing user convenience[50].

2.7. Cross-Validation

Cross-validation is a widely used data resampling method for model selection and evaluation. It helps with hyperparameter tuning and prevents overfitting[54]. Before explaining cross-validation, it is important to first understand the basic concepts of training and testing in model development.When building a model using a dataset, the dataset is typically split into training and testing sets, often in an 8:2 or 7:3 ratio. The training set is used to train the model, while the testing set is used to evaluate its performance. The purpose of this split is to objectively assess the model's performance. However, simply dividing the dataset into training and testing sets is not always sufficient for building a good model.

Since the goal of modeling is to create a high-performing model, the model is usually fine-tuned based on the performance results from the testing set. This, however, creates an issue. Once the model's performance metrics are known, the focus shifts to improving those specific metrics. It makes the testing set no longer an objective measure of generalization.

To address this, the concept of validation is introduced. By incorporating a validation set into the dataset, the data is split into training, validation, and testing sets, often in a 7:2:1 or 6:2:2 ratio. In this setup, the model is first trained and then evaluated using the validation set. The performance on the validation set is used for tuning the model before the final performance is measured using the testing set. However, one issue with this approach is that the model may overfit the validation set, leading to misleading performance estimates.

To mitigate this problem, cross-validation is used. There are various types of cross-validation, including k-Fold Cross-Validation, Leave-One-Out Cross-Validation, Nested Cross-Validation, Single Hold-Out Validation, and Jackknife Resampling[51].

The study in question employs Repeated K-Fold Cross-Validation, which is a variant of the commonly used K-Fold Cross-Validation. In k-Fold Cross-Validation, the dataset is divided into k subsets. The model is trained and tested k times, with each subset serving as the testing set once while the remaining k-1 subsets serve as the training set. The final performance is calculated as the average of the k testing evaluations[51]. For example, if there are 100 data points and 5-Fold Cross-Validation is used, each subset will contain 20 data points, denoted as *s1, s2, s3, s4*, and *s5*. In the first iteration, *s1* is used as the testing set while *s2* to *s5* serve as the training set. This process is repeated five times, and the final model performance is determined by averaging the results from all five testing sets.

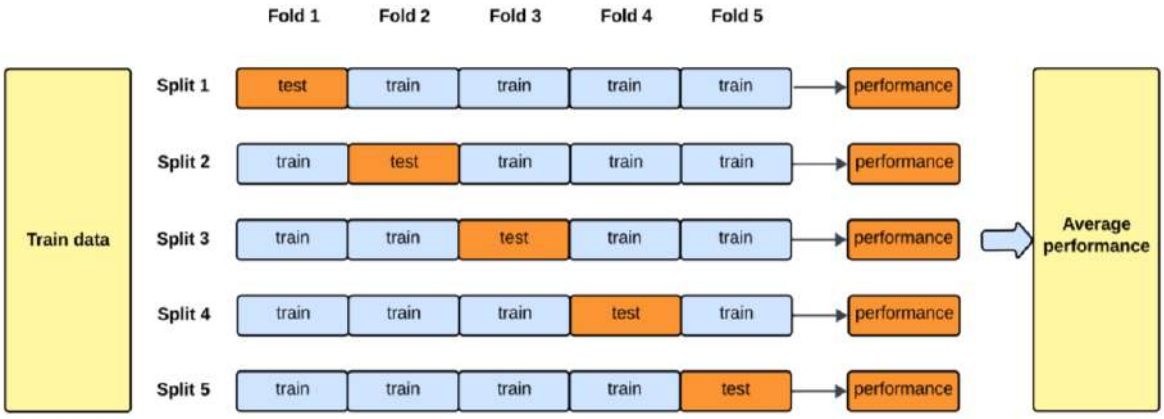


Figure 2-7. Visualization of Cross-Validation Structure. Reproduced from [56]

Repeated K-Fold Cross-Validation extends this concept by repeating the K-Fold Cross-Validation process multiple times. The final performance value is obtained by averaging the results across all repetitions, providing a more stable and reliable estimate of the model's performance [55] [56].

2.8. Model Interpretability

As mentioned earlier in 2.5. Machine Learning, we typically select a model suited to the nature of the problem - whether it be regression or classification - train it on the given data, and then compare its predictions with actual values. The choice of a model depends on the patterns we want to identify in the data.

If we are only interested in identifying simple linear relationships, a lightweight machine learning model, such as a linear regression, is sufficient. However, when we aim to capture nonlinear patterns for prediction, we often emply complex models. While these sophisticated and complex machine learning models can yield highly accurate results, they also introduce a significant challenge: the "black box problem[57]", where humans cannot easily understand how the model reaches its decisions.

In this study, I address this issue by utilizing SHapley Additive exPlanations (SHAP), which transforms complex models into explainable models, and it allows us to interpret their decision-making processes.

2.8.1. SHAP (SHapley Additive exPlanation)

All interpretable models must satisfy three key properties. The first property is Local Accuracy, which ensures that the explanation model must accurately reproduce the original model's predictions. The second property is Missingness, meaning that the contribution of any unused feature must be zero. The third property is Consistency, which states that if a model is modified in a way that increases or maintains the contribution of a certain feature, the attribution for that feature should not decrease. SHAP satisfies all three properties that an interpretable model must have[58].

An explanation model is defined as a method for explaining a model's predictions, and it takes the following form:

$$g(x') = \phi_0 + \sum_{i=1}^{M} \phi_i z_i'$$

where $g(x')$ is the explanation model, x' is a feature vector indicating whether a specific feature is used (outputting 0 or 1), ϕ_0 is a constant when inputs for all features are 0, and ϕ_i represents the contribution of a feature, which in this case corresponds to the SHAP value. M is the number of input features[54].

$$\phi_i = \sum_{S \subseteq N\{i\}} \frac{|S|!\,(|N| - |S| - 1)!}{|N|!} [f(S \cup \{i\}) - f(S)]$$

where N is the set of all input features, S is a subset of features excluding feature i, $S \subseteq N\{i\}$ represents all possible subsets that do not include feature i, $S \cup \{i\}$ represents a subset that

includes feature i, $f(S \cup \{i\})$ is the model's prediction when feature i is added, and $f(S)$ is the model's prediction when only subset S is used[54].

To simplify SHAP using a game analogy, imagine that five players participate in a game and win. SHAP can be understood as the measure of how much each player contributed to the victory. Since SHAP values can be both positive and negative. A negative SHAP value for feature i means that it contributed to decreasing the model's prediction. On the other hand, a positive SHAP value for feature i means that it contributed to increasing the model's prediction.

2.8.2. Tree SHAP

Tree SHAP is a newly proposed algorithm specifically designed for tree-based models. While tree-based models exhibit strong predictive performance, traditional feature importance measures—Gain, Split Count, and Permutation—fail to satisfy the third key property of an explanation model: Consistency. In tree-based models, feature importance is typically measured using three main criteria: 1) Gain: The reduction in loss when a feature is used for a split in the tree, 2) Split Count: The number of times a feature is used for a split in the tree, and 3) Permutation: The performance degradation of the model when the feature values are randomly shuffled[59].

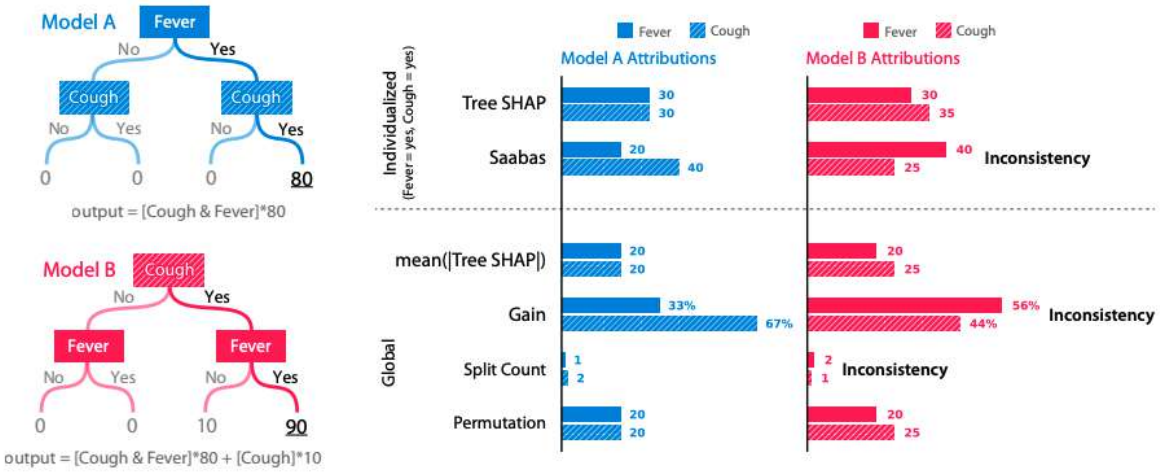


Figure 2-8. Inconsistent Feature Importance. Reproduced from [59]

As shown in the figure, even when models are trained on the same dataset, the calculated feature importance varies between models. This inconsistency makes the feature importance measures unreliable. To address this issue, Tree SHAP was proposed as an algorithm that leverages SHAP, the only method that satisfies all three key properties of an explanation model[59].

One of the most important characteristics of Tree SHAP is its fast computation speed. When compared to standard SHAP, the computational complexity of Tree SHAP was significantly reduced. While standard SHAP has exponential time complexity, Tree SHAP reduces it to polynomial time complexity, making it feasible for large-scale tree-based models[59].

2.9. Evaluation Metrics

To evaluate performances of the models, two main metrices are used: Root mean squared error (RMSE) and Pearson correlation coefficient (R^2). RMSE is a metric that represents the difference between actual values and predicted values. It is calculated by taking the square root of the mean of the squared differences between the actual and predicted values.

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

where $\hat{y}$ represents the prediction value, and y represents the actual value. RMSE measures how much the predicted values deviate from the actual values. By squaring the differences before averaging, RMSE is more sensitive to larger errors. While RMSE compares the model's predicted values with the actual values, R-squared indicates how well the trained model explains the variance in the data.

$$R^2(\hat{y}, y) = 1 - \frac{SSR}{SST} = 1 - \frac{\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}{\sum_{i=1}^{n}(y_i - \bar{y}_i)^2}$$

Similarly, $\hat{y}$ represents the prediction value, y represents the actual value, and $\bar{y}_i$ represents the mean of the actual values. R^2 measures how explanatory variables explains the variance of a response variable. The range of R^2 is between 0 and 1. Values closer to 1 indicate that the model explains the data well. However, the value could be negative. When the R^2 value is negative, it means that the model not only fails to explain the data at all but also performs worse than predicting with the mean value.

While RMSE provides a measure of the absolute error in the model's predictions, it can be difficult to judge the magnitude of the error without context. To address this, the Normalized Root Mean Squared Error (NRMSE) is also used to evaluate the model's performance on a relative scale. NRMSE is calculated by dividing the RMSE by the mean of the actual values ($\bar{y}$), as shown in the formula below:

$$NRMSE(\%) = \frac{RMSE(\hat{y}, y)}{\bar{y}} \cdot 100$$

This metric expresses the average prediction error as a percentage of the average yield. It provides a scale-independent measure of accuracy, which is particularly useful for comparing the performance of models across different crops that may have vastly different average yields.

3. DATA/PROCESS

3.1. Study Period

The dataset used in this study spans from January 2018 to December 2023, with monthly intervals. A six-year period was chosen because one or two years is too short to observe meaningful trends related to climate change. However, using monthly data to predict crop yield presents a specific challenge: the growing period. For instance, if planting begins in April, data from January to March of that year does not contribute to crop growth. Similarly, if harvesting occurs in November, the data from December is also irrelevant to that year's crop development. To address this, the study applied crop-specific planting and harvesting seasons to filter the data accordingly. Detailed planting and harvesting periods for each crop are described in *Section 3.3. Crop Variables*.

3.2. County Centroid

As explained in Section *2.Method/Theory*, this study utilizes satellite data. However, original satellite datasets typically cover global regions, so it was necessary to extract data relevant only to the regions of this study. Specifically, I needed to collect data at the county level for the six U.S. states identified in Section *2.1. Crop and State Selection*. One limitation I encountered was the complexity of extracting and averaging monthly satellite values across all pixels within each county. After discussions with my advisor, we decided to calculate the centroid coordinate of each county and use the satellite value at that point as the representative value for the entire county. *Figure 3-1* and *Figure 3-2* below illustrates the centroid coordinates for all counties within Montana and Iowa. Centroid coordinates were also generated for the

remaining four states: Washington, North Dakota, Illinois, and Minnesota. The figures below illustrate that the centroids were calculated effectively for counties with both regular and highly irregular shapes.

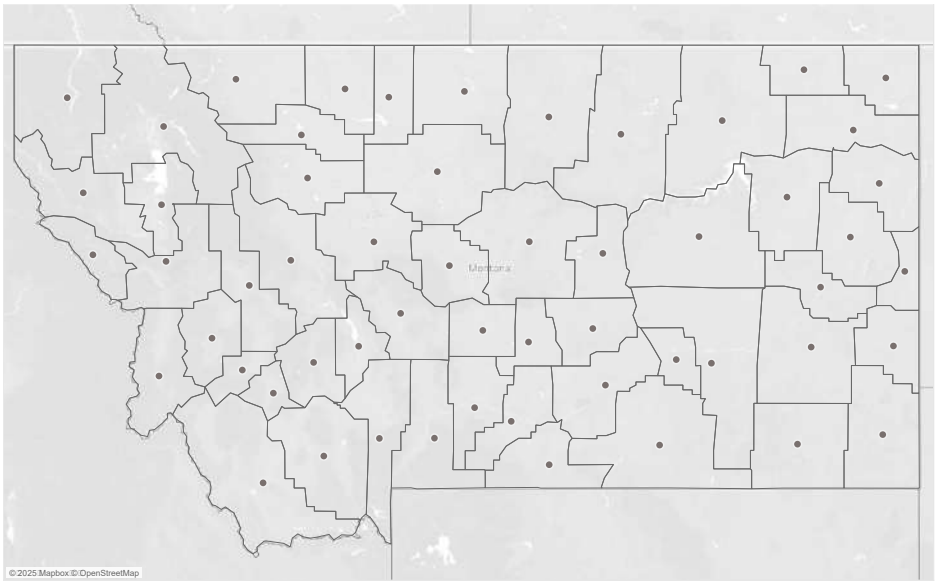


Figure 3-1. Counties Centroid Coordinates in Montana

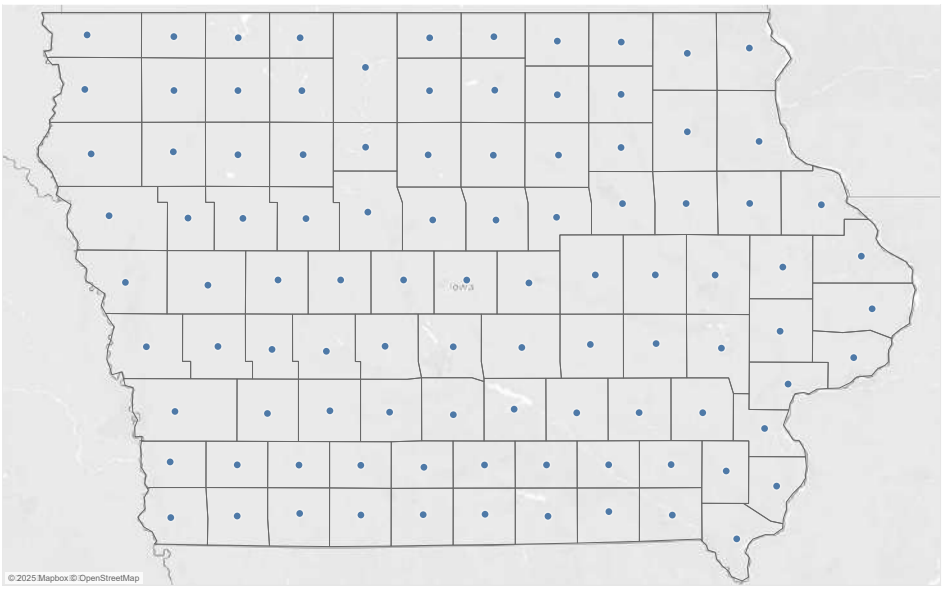


Figure 3-2. Counties Centroid Coordinates in Iowa

A shapefile, which is open to public data[60], containing county-level boundaries across the United States was used for this study. This shapefile includes a variable called STATEFP, which

assigns a unique numeric code to each U.S. state. For making a file that stores counties' centroids coordinates, I firstly extracted the counties in the following six states based on the STATEFP: Illinois (code 17), Iowa (code 19), Minnesota (code 27), Montana (code 30), North Dakota (code 38), and Washington (code 53). Each county in the shapefile is represented as a polygon, and the `geometry` variable contains the coordinates of the polygon's vertices that outline the county's boundary.

For the next, using the GeoPandas library, calculatation of the centroid of each county polygon based on its `geometry` has been done. Examples of the calculated centroids are shown in the *Figure 3-1* and *Figure 3-2*. Regardless of the shape of the polygon, the calculation in the library can automatically determine the centroid coordinates. These centroid values were saved as CSV files named 'county_centroid_corn.csv', 'county_centroid_soybean.csv', and 'county_centroid_wheat.csv'. Each CSV file contains five columns: COUNTY_NAME, STATEFP, STATE_NAME, LATITUDE, and LONGITUDE. As previously described, these files were used to extract spatially relevant values from the satellite datasets for each county.

3.3. Crop variable

As described in *2.1. Crop and State Selection*, this study focuses on three major crops that contribute significantly to crop production in the United States: corn, soybean, and wheat. The response variable used in this study is yield rather than total production. Before introducing the data, it is important to explain why yield was chosen over production.

Production represents the total output and is typically measured in tons or bushels. However, the counties used in this study vary greatly in size. Larger counties are likely to produce more simply due to their area, while smaller counties may produce less. Therefore, using

production would introduce a bias related to county size. To account for this, the study uses yield, which is normally measured in bushels per acre. Yield provides a size-independent measure of agricultural productivity, allowing for a more accurate analysis of the relationship between climate variables and crop performance.

The yield data was obtained from the United States Department of Agriculture (USDA)[61]. For all three crops—corn, soybean, and wheat—the data collection method used is the Survey. USDA data can be collected through two main methods. Firstly, CENSUS is conducted every five years, and this method involves sending surveys to all farms across the country to collect comprehensive data. Second, SURVEY is conducted annually, this method collects data from a representative sample of farms rather than the entire population[62].

The study adopted the Survey method. I determined that using a representative sample, rather than the entire population, would be sufficient for the study's goals. Additionally, using annual survey data allows us to analyze the relationship between yield and climate variables continuously over time.

3.3.1. Corn

The study period for corn yield spans from 2018 to 2023, and the study area includes all counties in Minnesota, Iowa, and Illinois. The unit of measurement for yield is bushels per acre (BU/ACRE), and the data is reported on an annual basis. As discussed in *3.1. Study Period*, the planting and harvesting seasons differ across crops. While the yield data is annual, the climate variables used in the study are provided monthly. For this study, the planting season applied to corn is from April to November.

Within the category of corn yield, there were several available options. Corn could be classified based on its intended use—such as corn for grain (used for human consumption) or

corn for silage (typically used for livestock feed). Additionally, yield data could be further categorized by farming practices, distinguishing between irrigated and non-irrigated corn.

In this study, I used corn for grain as the target yield variable and did not differentiate between irrigated and non-irrigated practices; instead, the yield data included both types of farming practices were used for the corn yield.

3.3.2. Soybean

The study period for soybean yield spans from 2018 to 2023, and the study area includes all counties in Minnesota, Iowa, and Illinois. The unit of measurement for yield is bushels per acre (BU/ACRE), and the data is reported on an annual basis. The planting season applied to soybean is from May to October.

Unlike the options available for corn yield, soybean yield did not have classifications based on end-use purpose. However, like corn, there were options to distinguish between irrigated and non-irrigated soybean yield. In this study, I used the total soybean yield, which includes both irrigated and non-irrigated yields.

3.3.3. Wheat

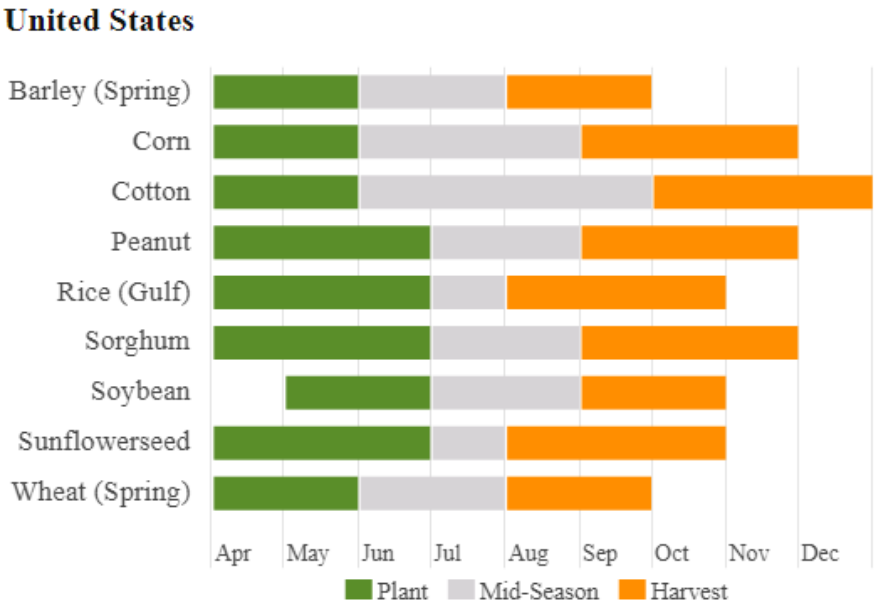
The study period for wheat yield spans from 2018 to 2023, and the study area includes all counties in North Dakota, Washington, and Montana. The unit of measurement for yield is bushels per acre (BU/ACRE), and the data is reported on an annual basis. The planting season applied to wheat is from April to September.

Unlike corn and soybean, wheat yield includes additional categories. Specifically, it can be classified into spring wheat and winter wheat, and there is also the option to include or

exclude a specific variety called durum wheat. In this study, I used irrigated and non-irrigated spring wheat yield, excluding durum.

The primary difference between spring and winter wheat lies in their harvesting periods—winter wheat is planted in the fall and harvested the following summer, requiring a dormancy (fermentation) period, whereas spring wheat is planted and harvested within the same year. For this reason, we chose to use spring wheat, as it aligns better with the study's annual climate variable framework.

Durum wheat was excluded because yield data that includes durum is not available at the county level for each state, making it incompatible with my analysis. By excluding durum, it was able to obtain county-level spring wheat yield data for Montana, Washington, and North Dakota.



63

Figure 3-3. Crop Planting and Harvesting Calendars for United States. Reproduced from [63]

3.4. Climate variable

To build a model that predicts the annual yield of corn, soybean, and wheat, the study used five independent variables, and all of them are closely related to climate change:

temperature, precipitation, solar radiation, CO_2, and soil moisture. Among these, temperature and precipitation data were readily available in numerical format from online sources. However, the remaining three variables - solar radiation, CO_2, and soil moisture - required processing of satellite data to convert them into usable, structured numerical datasets.

The final goal was to generate a CSV file that contains, for each county and for each year, the monthly values (January to December) for all five climate variables. In *3.4. Climate Variables*, I will describe each variable in detail and explain the steps taken to process the raw data into the final structured CSV format used in the study.

3.4.1. Temperature

For temperature, I used average temperature as the variable. The data was obtained from the National Centers for Environmental Information, under the National Oceanic and Atmospheric Administration (NOAA), specifically through the Climate at a Glance Mapping[64]. According to NOAA, the county-level data is sourced from the U.S. Climate Divisional Database. The unit of measurement is in degrees Fahrenheit.

On the website, users are required to manually select five options to retrieve the data. For the State option, I selected each of the six target states individually: Iowa, Minnesota, Illinois, Montana, North Dakota, and Washington. For the option 'Parameter', I chose 'Average Temperature'; for the option 'Year', I selected 2018 through 2023; for the option 'Month', I chose all months from January to December; and for the option 'Time Scale', I selected '1-Month'.

Since the website does not support downloading all desired data at once, I manually adjusted the State, Year, and Month options to download average temperature data for each county in CSV format. From the downloaded CSV files, I extracted the average temperature

values from the Temperature column and organized them by state. As a result, six final CSV files were created, named accordingly - for example, "Illinois_Average_Temperature.csv". Each final CSV file by the states contains the following columns: *County*, *State, Year, Jan, Feb*, ..., *Nov*, and *Dec*.

County	State	Year	Jan	Feb	Mar	Apr	May	Jun	July	Aug	Sep	Oct	Nov	Dec
ADAMS	ILLINOIS	2018	24.7	31.5	39.7	45.3	72.6	77.0	76.5	76.0	70.7	54.2	35.2	33.8
ALEXANDER	ILLINOIS	2018	29.1	41.6	46.6	51.2	74.0	78.9	79.8	77.0	74.1	60.2	41.8	41.3
BOND	ILLINOIS	2018	26.9	36.6	41.6	47.4	72.9	77.2	76.8	75.6	71.9	57.1	37.0	36.9
BOONE	ILLINOIS	2018	19.2	22.9	34.6	38.9	64.7	69.3	71.5	72.0	65.4	48.8	31.5	29.6
BROWN	ILLINOIS	2018	25.1	32.1	39.8	45.3	73.0	77.3	76.2	75.5	70.9	54.5	35.8	33.8
BUREAU	ILLINOIS	2018	20.4	25.5	36.4	41.1	68.0	72.2	72.7	73.1	67.5	50.2	32.8	31.4
CALHOUN	ILLINOIS	2018	26.6	35.1	41.6	47.6	72.9	77.6	77.2	76.3	72.1	56.6	36.4	36.0
CARROLL	ILLINOIS	2018	19.6	23.4	35.8	39.7	66.9	71.9	72.6	72.3	66.7	49.9	32.3	30.2
CASS	ILLINOIS	2018	24.8	31.9	39.2	45.3	72.6	76.4	75.4	75.1	70.7	54.4	35.1	33.2
CHAMPAIGN	ILLINOIS	2018	22.9	32.4	37.8	44.7	71.5	74.3	74.1	74.3	70.5	54.2	35.4	33.6
CHRISTIAN	ILLINOIS	2018	25.1	33.8	40.4	46.2	72.3	75.6	74.9	74.7	71.2	55.3	35.3	34.5
CLARK	ILLINOIS	2018	24.8	36.2	39.9	46.5	72.5	75.5	75.4	74.9	71.7	56.1	37.0	36.0
CLAY	ILLINOIS	2018	26.8	37.6	42.0	47.9	72.7	76.5	76.4	75.3	72.4	57.1	38.2	37.6
CLINTON	ILLINOIS	2018	27.0	37.8	42.7	48.4	73.4	78.1	77.9	76.2	72.8	57.9	38.3	38.1

Table 3-1. General Structure of the CSV files for Average Temperature

The table above represents the average temperature CSV file for Illinois, and the CSV files for the other states follow the same structure. The only difference among them is the number of rows, which varies depending on the number of counties in each state. Specifically, the number of rows for each state is as follows: 612 rows in Illinois, 594 rows in Iowa, 522 rows in Minnesota, 318 rows in North Dakota, 336 rows in Montana, and 234 rows in Washington.

It is important to clarify that each row represents a single county for a specific year. Since the study spans a six-year period, each county is represented by six distinct rows, which accounts for the high row counts per state. This "county-year" format serves as the target structure for all datasets developed in this study.

3.4.2. Precipitation

For precipitation, I used precipitation as the variable. Like the average temperature, the precipitation data was obtained from the Climate at a Glance[64] at NOAA. The unit of measurement is in inches. The precipitation value for a month is not an average precipitation. If the value is 3.1 inches in March, it means the total amount of precipitation accumulated over the entire month is 3.1 inches.

There were the same 5 options that users are required to select. All the processes are the Same as the average temperature, for the option 'Year', I selected 2018 through 2023; for the option 'Month', I chose all months from January to December; and for the option 'Time Scale', I selected '1-Month.' The only difference from the temperature is that, for the option 'Parameter', I chose 'Precipitation'.

I manually adjusted the options to download precipitation data for each county in CSV format. Then, I extracted the precipitation values and organized them by state. Six final CSV files were made, and the following figure is the basic structure of the precipitation CSV file.

County	State	Year	Jan	Feb	Mar	Apr	May	Jun	Jul	Aug	Sep	Oct	Nov	Dec
ADAMS	ILLINOIS	2018	1.23	3.37	3.89	0.97	2.39	2.79	3.43	6.25	5.59	5.94	2.51	2.17
ALEXANDER	ILLINOIS	2018	2.17	8.38	5.37	5.46	4.69	6.85	1.47	3.47	5.15	3.15	4.96	5.29
BOND	ILLINOIS	2018	1.28	6.02	6.66	3.70	2.97	6.10	3.98	4.57	3.68	1.68	3.16	3.64
BOONE	ILLINOIS	2018	2.13	2.94	0.91	1.69	4.71	8.42	2.70	5.01	7.01	5.63	2.10	1.87
BROWN	ILLINOIS	2018	1.40	3.39	4.42	1.59	3.06	2.74	4.24	6.68	4.68	5.16	3.09	1.87
BUREAU	ILLINOIS	2018	1.03	3.66	2.56	1.13	5.39	5.80	4.05	4.58	3.50	3.90	2.55	2.13
CALHOUN	ILLINOIS	2018	1.50	3.64	5.05	1.75	3.57	3.34	2.20	4.89	1.49	2.98	2.48	3.39
CARROLL	ILLINOIS	2018	0.89	3.24	2.19	1.30	4.39	8.62	3.57	7.06	7.75	5.97	2.04	2.29
CASS	ILLINOIS	2018	1.69	3.25	4.32	2.12	3.31	3.53	4.25	7.87	2.72	4.36	3.08	2.54
CHAMPAIGN	ILLINOIS	2018	0.97	5.36	2.98	2.44	3.80	7.06	4.69	4.19	4.67	3.09	3.37	3.07
CHRISTIAN	ILLINOIS	2018	0.97	4.52	4.99	2.59	2.82	7.97	5.95	5.25	2.55	2.02	2.67	3.37

Table 3-2. General Structure of the CSV files for Precipitation

Just like the average temperature data, the precipitation CSV file for each state follows the same structure, with the only difference being the number of rows, which corresponds to the number of counties in each state. Specifically, Illinois has 612 rows, Iowa has 594 rows, Minnesota has 522 rows, North Dakota has 318 rows, Montana has 336 rows, and Washington has 234 rows.

3.4.3. Solar Radiation

Photosynthesis is essential for plant growth, and solar radiation is a crucial factor for photosynthesis. In this study, the solar radiation data used comes from the Energy Balanced and Filled – Top of Atmosphere Edition 4.2.1 (EBAF-TOA Ed 4.2.1) - Level 3b[65]. The variable used from this dataset is "Solar Incoming." The definition of data levels, such as “Level 3b”, can be found in *2.3. Remote Sensing*.

This dataset is a Climate Data Record (CDR) produced by NASA’s Clouds and the Earth's Radiant Energy System (CERES) instrument. Like the previously used precipitation and temperature datasets, this data has a monthly average temporal resolution. However, it differs in terms of data storage format; while previous datasets were numerical and stored as CSV files, this one is provided as a NetCDF file. Network Common Data Form (NetCDF) is a file format storing multidimensional scientific data (variables)[66], so users need to extract the variables and convert into numerical form.

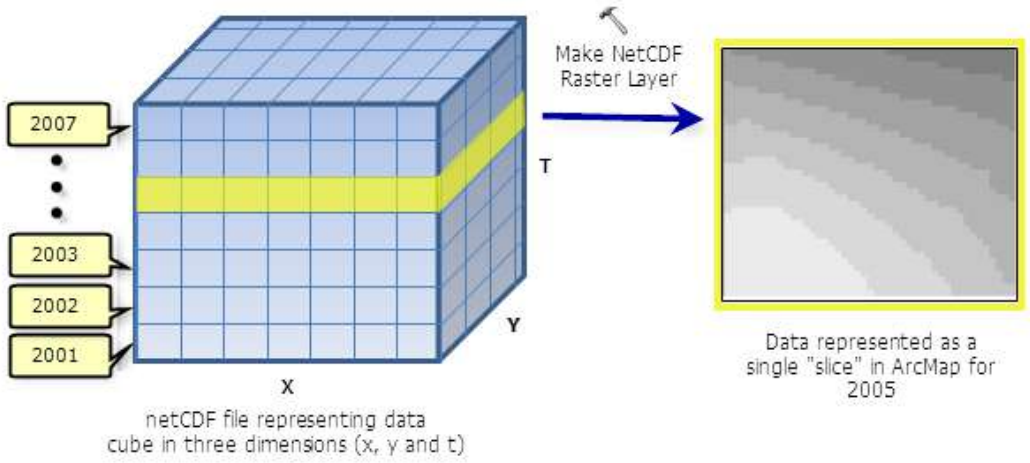


Figure 3-4. Structure of a NetCDF file. Reproduced from [66]

The Solar Incoming variable, or solar radiation, is measured in Watts per square Meter (W/m^2) and has a spatial resolution of $1° \times 1°$ on a global grid. This means each grid cell represents an area of approximately 12,000 km^2. The data includes values under both all-sky and clear-sky conditions.

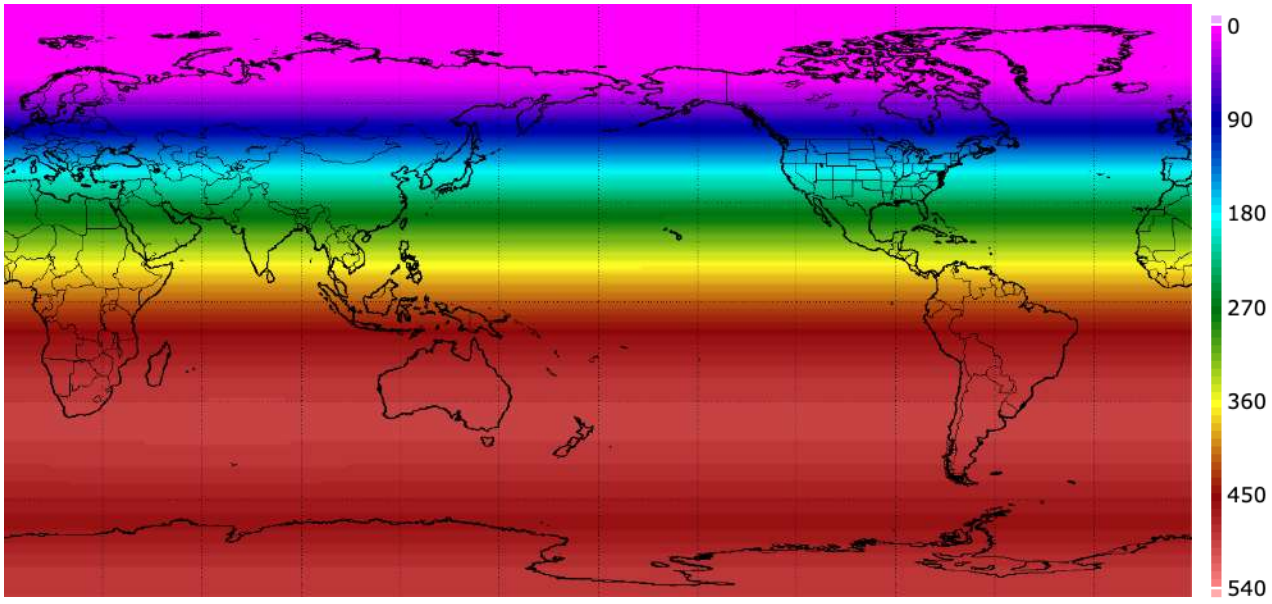


Figure 3-5. Visualization of the Solar Incoming in January 2018. Reproduced from [65]

The NetCDF file used in this study was downloaded with the study period set to January 2018 to December 2023. This file contains a 3-dimensional structure for the variable `solar_mon`, which holds monthly solar radiation data in units of (time, latitude, longitude). For each crop - corn, soybeans, and wheat - county centroid datasets created in *3.2.County Centroid* were loaded separately. Using the latitude and longitude coordinates from these centroid files, a processing code extracts the `solar_mon` value from the NetCDF file by selecting the nearest grid point. For the temporal dimension, year and month values were extracted using the time variable in the NetCDF dataset. Then, for each county, `solar_mon` values from 2018 to 2023 were organized into a data frame containing year, month, and `solar_mon`.

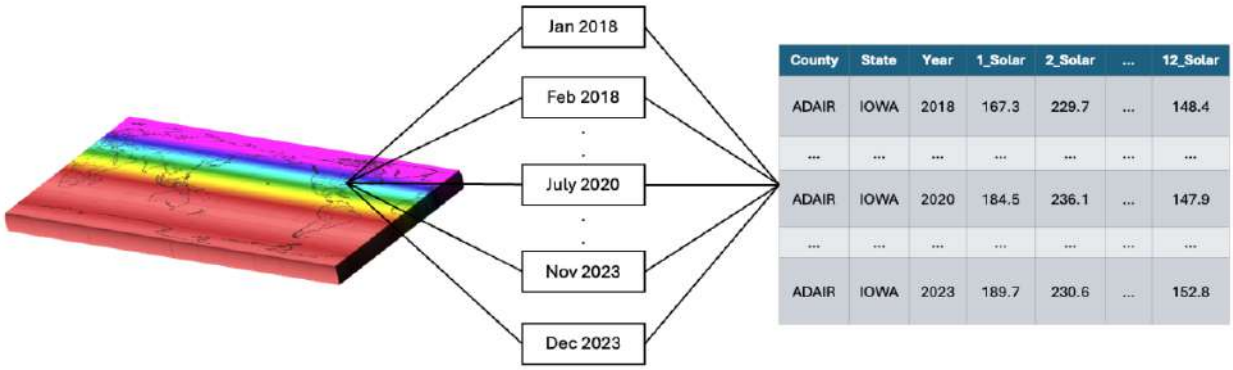


Figure 3-6. Solar Radiation Processing Visualization

All collected county-level data were combined into a single data frame and then reshaped into a pivot table where each column corresponds to a specific month's solar radiation. Finally, the processed data were saved as separate CSV files for each crop: solar_corn.csv, solar_soy.csv, and solar_wheat.csv. The structure of these final CSV files is consistent with those of the precipitation and temperature datasets used in this study.

County	State	Year	1_Solar	2_Solar	3_Solar	4_Solar	5_Solar	6_Solar	7_Solar	8_Solar	9_Solar	10_Solar	11_Solar	12_Solar
ADAIR	IOWA	2018	167.3	229.7	314.6	398.2	457.4	481.7	467.5	417.3	341.7	255.5	182.6	148.4
ADAIR	IOWA	2019	167.0	229.1	313.9	397.6	457.0	481.7	467.7	417.8	342.3	256.2	183.0	148.5
ADAIR	IOWA	2020	166.6	229.8	316.1	399.5	458.1	481.8	466.9	416.2	340.3	254.1	181.6	148.3
ADAIR	IOWA	2021	167.8	230.5	315.5	399.0	457.8	481.9	467.3	416.8	341.0	254.8	182.1	148.4
ADAIR	IOWA	2022	167.4	230.0	314.9	398.6	457.8	482.1	467.8	417.5	341.9	255.6	182.6	148.5
ADAIR	IOWA	2023	167.1	229.4	314.3	398.2	457.5	482.2	468.0	418.2	342.6	256.4	183.2	148.6
ADAMS	ILLINOIS	2018	181.5	242.8	324.9	404.3	459.2	481.4	468.2	421.7	350.5	267.7	196.4	162.7
ADAMS	ILLINOIS	2019	181.1	242.2	324.3	403.7	458.9	481.3	468.4	422.2	351.1	268.3	196.9	162.7
ADAMS	ILLINOIS	2020	180.8	243.0	326.4	405.4	459.9	481.4	467.7	420.7	349.1	266.3	195.5	162.6
ADAMS	ILLINOIS	2021	181.9	243.6	325.8	405.0	459.6	481.5	468.1	421.3	349.8	267.0	196.0	162.6
ADAMS	ILLINOIS	2022	181.6	243.1	325.3	404.7	459.7	481.8	468.5	422.0	350.7	267.8	196.5	162.7

Table 3-3. General Structure of the CSV files for Solar Radiation

While the precipitation and temperature CSV files are saved by states basis, the solar radiation data are saved by crops basis in CSV file. As a result, both the corn and soybean CSV files contain 1,728 rows each, which is the number of all counties in Iowa, Illinois, and Minnesota. The wheat file, on the other hand, contains 888 rows, which is the number of all counties in North Dakota, Washington, and Montana.

3.4.4. CO_2

Another climate variable used to predict crop yield in this study is CO_2 concentration. The CO_2 data was obtained from NASA's Orbiting Carbon Observatory-2 (OCO-2) mission and provided in a processed form as the OCO-2 Level 2 bias-corrected XCO_2 product (OCO2_L2_Lite_FP). This dataset contains daily measurements of column-averaged CO_2 concentrations, with the unit of measure in parts per million (ppm). Like the solar radiation data, it is delivered in netCDF format. The spatial resolution of the dataset is approximately 2.25 km x 1.29 km, and the temporal resolution is 16 days.

The distinction between daily data and 16-day temporal resolution can be confusing. Although the data is collected and provided on a daily basis, the temporal resolution of 16 days

refers to the time it takes for the satellite to revisit the same location on Earth. This means that for any specific location, new data is only available roughly every 16 days, even though global data is collected daily. The version of the dataset used in this study is 11.2r.

The data is available for download through NASA's GES DISC Earthdata Portal[67] for users with an authorized account. When a specific time range is selected – for example, from March 1 to March 10, 2025 - the portal returns a text file containing 11 download links, one for each day in that range. However, given the study's time span of six years (2018 - 2023), the total number of download links amounted to 2,136, making manual downloading impractical. To address this, I developed an automated data downloading pipeline, which uses the provided links to download the data and organize it into folders. A critical requirement when accessing the data through the pipeline is the use of an API token, which is assigned to each registered Earthdata account.

Unlike the solar radiation dataset, the CO_2 data processing involved multiple trials. The first approach attempted was to extract XCO_2 values at each county centroid from the daily netCDF files and compute monthly averages. However, this approach often resulted in missing values (NaNs) due to the absence of data at the exact centroid coordinates on many days.

The second approach involved using spatial interpolation (*Section 2.4*) via the Inverse Distance Weighting (IDW) method. If a value was not directly available at a centroid, the method predicted the value based on the five nearest points. However, this approach introduced another problem. As shown in the *Figure 3-6* below, OCO-2's daily measurements are often limited to striped swaths, and many centroids fall far outside these swaths. Predicting values from distant points using IDW proved unreliable in these cases.

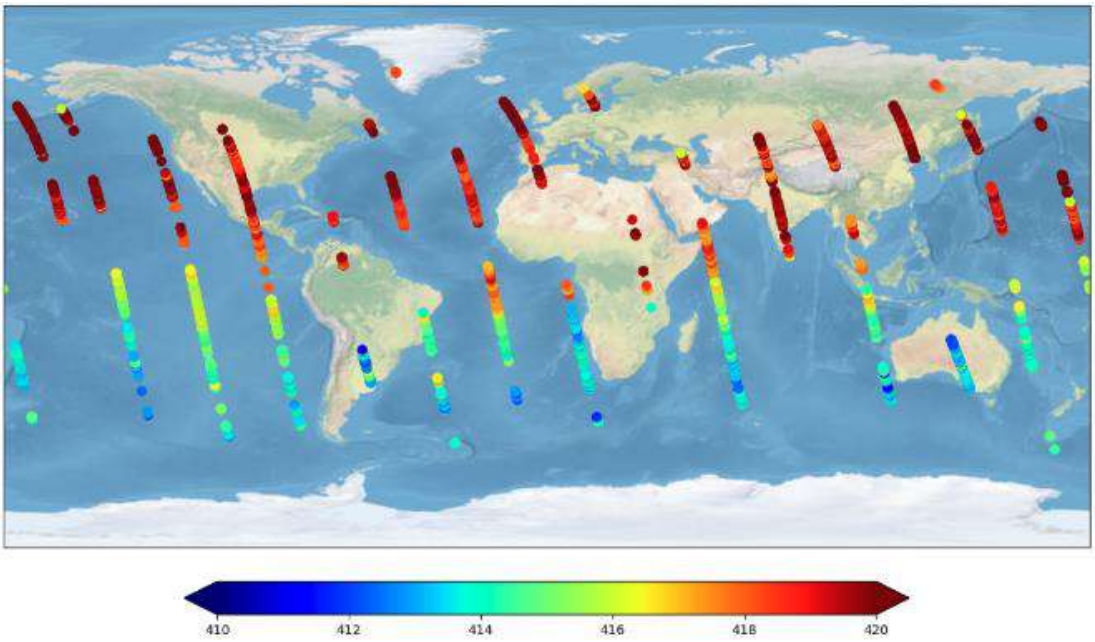


Figure 3-7. Visualization of Daily XCO2 Data. Reproduced from [67]

The final and most successful approach leveraged the satellite's temporal resolution. I combined 16 days' worth of daily netCDF files into one file. For each centroid, if a direct XCO_2 value was available in the combined file, it was used. If not, IDW was applied to estimate the value based on the five closest data points. After three iterations of testing, this approach enabled the successful extraction of monthly XCO_2 values for the centroids of each county growing each crop.

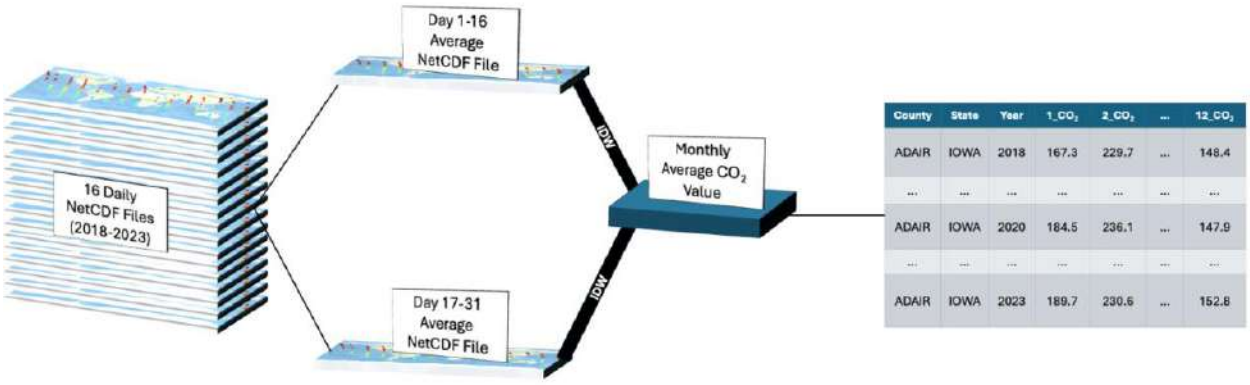


Figure 3-8. CO2 Processing Visualization

The resulting datasets were stored in three CSV files - “CO2_corn.csv”, “CO2_soy.csv”, and “CO2_wheat.csv” - which follow the same structure as the previously created CSV files for other climate variables. The CSV files are structured with the following columns: County, State, State_ID, Year, 1_CO2, 2_CO2, ..., 12_CO2. As with the previously created CSV files, the corn and soybean files each contain 1,728 rows, while the wheat file contains 888 rows.

County	State	State_ID	Year	1_CO2	2_CO2	3_CO2	4_CO2	5_CO2	6_CO2	7_CO2	8_CO2	9_CO2	10_CO2	11_CO2	12_CO2
ADAIR	IOWA	19	2018	412.921	408.016	410.725	410.923	412.258	407.323	402.534	400.589	406.341	408.672	406.596	408.493
ADAIR	IOWA	19	2019	412.265	410.639	409.586	411.547	413.474	408.895	403.408	404.833	408.565	408.896	408.973	412.776
ADAIR	IOWA	19	2020	412.212	413.324	414.884	420.360	415.251	411.498	413.505	409.242	410.841	413.418	413.037	415.688
ADAIR	IOWA	19	2021	416.104	416.934	416.424	417.587	419.113	414.508	413.308	411.409	410.928	413.822	418.327	417.265
ADAIR	IOWA	19	2022	418.867	420.540	419.537	419.203	420.336	418.384	412.093	411.700	414.163	416.944	419.851	421.208
ADAIR	IOWA	19	2023	421.770	420.577	418.119	425.220	422.914	419.962	412.957	414.724	419.393	416.828	422.262	419.374
ADAMS	ILLINOIS	17	2018	403.709	410.603	410.191	411.808	412.041	405.490	404.627	399.306	406.868	408.067	414.275	409.540
ADAMS	ILLINOIS	17	2019	409.993	412.324	412.934	412.625	413.437	414.318	409.624	407.525	407.562	410.238	411.228	412.151
ADAMS	ILLINOIS	17	2020	413.298	414.413	416.652	415.237	415.619	413.489	406.824	405.022	408.186	409.413	412.544	413.347
ADAMS	ILLINOIS	17	2021	406.850	416.840	418.683	417.439	417.355	415.314	411.348	411.650	412.370	417.004	416.915	412.856
ADAMS	ILLINOIS	17	2022	419.023	420.386	422.538	423.012	420.447	419.296	419.393	413.852	414.377	421.899	409.449	420.046
ADAMS	ILLINOIS	17	2023	419.542	421.625	422.357	424.568	420.565	417.005	414.704	415.439	416.044	418.867	424.147	420.404
ADAMS	IOWA	19	2018	408.525	407.226	413.948	411.353	411.419	409.438	405.486	405.707	409.719	406.770	406.662	399.372
ADAMS	IOWA	19	2019	411.449	411.099	411.926	410.979	414.487	408.530	407.880	405.578	409.430	408.773	410.875	412.636
ADAMS	IOWA	19	2020	412.846	414.715	414.515	419.697	415.157	412.523	411.163	408.943	410.755	412.956	412.899	414.917
ADAMS	IOWA	19	2021	416.676	417.139	416.357	417.810	416.794	415.077	412.898	410.810	411.064	414.278	418.406	417.075
ADAMS	IOWA	19	2022	418.739	420.359	419.755	422.524	420.866	419.685	411.345	412.966	412.910	415.217	420.224	420.786
ADAMS	IOWA	19	2023	422.174	420.530	417.562	424.315	422.473	419.178	414.317	415.007	419.186	417.970	419.855	423.575
AITKIN	MINNESOTA	27	2018	406.803	411.382	409.376	411.761	410.340	409.140	405.678	397.352	402.265	409.484	400.379	406.951

Table 3-4. General Structure of the CSV files for CO2

3.4.5. Soil Moisture

The final climate variable used to predict crop yield in this study is soil moisture. The soil moisture data was obtained from the AMSR2/GCOM-W1 surface soil moisture product (LPRM_AMSR2_DS_A_SOILM3), which is provided through NASA’s GES DISC [68] platform, similar to the CO_2 dataset.

As indicated by the data name, it is based on measurements collected by the Advanced Microwave Scanning Radiometer 2 (AMSR2) onboard the GCOM-W1 satellite. These measurements are acquired using the passive remote sensing method, as described in *2.3 Remote Sensing*. The received microwave signals are then processed through the Land Parameter

Retrieval Model (LPRM) to quantitatively estimate surface variables such as soil moisture, land surface temperature, and vegetation water content.

The data is provided in daily intervals and stored in netCDF format, with a spatial resolution of 10 km × 10 km and a temporal resolution of 1 day. Upon inspecting the netCDF files, aside from location variables like latitude and longitude, there are three main soil moisture-related variables: c1, c2, and x. c1 represents soil moisture from the surface down to 5 cm, c2 represents soil moisture from 5 cm to 50 cm, and x is a mixture of both c1 and c2.

The variable used in this study is c2. This variable was chosen because c1 is highly explained by precipitation, and c2 captures moisture at depths (5-50 cm) where crop roots are more likely to reside, making it more relevant for yield prediction.

The method for downloading and organizing this data followed the same automated pipeline used for CO_2 data. Using an API token linked to a NASA GES DISC account, a total of 2,136 daily files spanning six years were downloaded and stored in a structured folder system, organized by year and month.

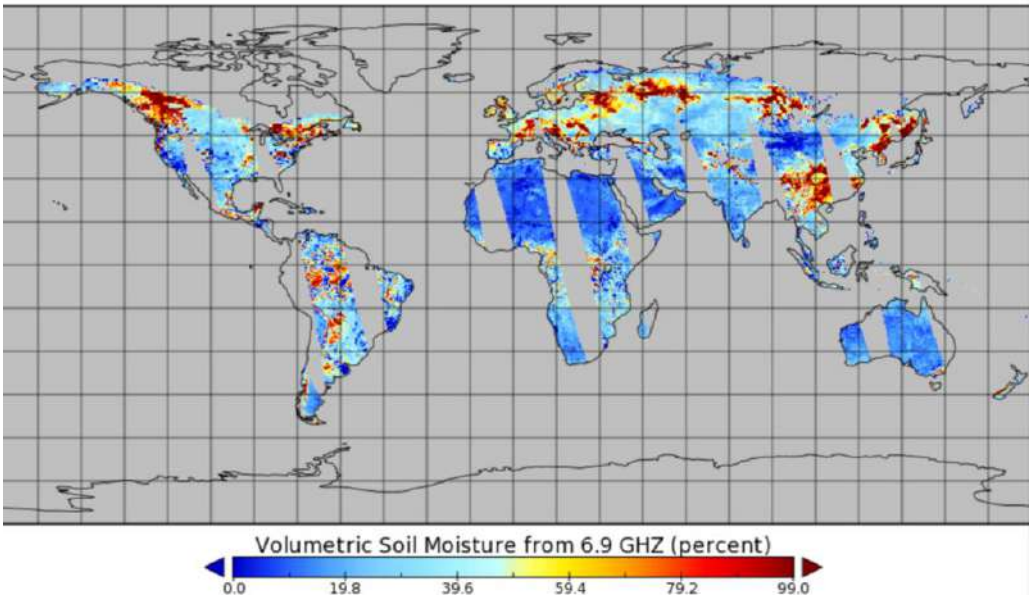


Figure 3-9. Visualization of Daily Soil Moisture Data. Reproduced from [68]

Several attempts were made to convert the soil moisture data into a fully structured CSV file containing numerical values. As seen in the *Figure 3-7* of the dataset, the satellite captures data in the paths, leaving significant gaps in spatial coverage, similar to the CO_2 data.

The initial approach involved extracting the soil moisture value at each county centroid from the daily netCDF file. If a value was not present at that location, I planned to use IDW (Inverse Distance Weighting) interpolation to estimate the missing value. However, as shown in the *Figure 3-7*, many areas were missing data on any given day, and using IDW in these cases led to unreliable or invalid estimates due to the large distance from the nearest available data points.

To address this, I attempted to combine multiple days of data into a single composite layer. The first trial combined three consecutive days, but gaps still remained. In the second trial, I used 7 days of data, but there is still gap. In the second trial, I used 15 days of data, which proved sufficient to provide data coverage for nearly all land areas. In cases where overlapping girds existed across days, I computed the average value for those grids.

From the 15-day composite, I extracted the soil moisture value at each county centroid. If a direct value was not available at the centroid, IDW interpolation was again used based on the nearest available data points within the 15-day window. These extracted values were then averaged monthly and stored in CSV files for all counties in the three states under study. The final output files were named: "corn_sm.csv", "soy_sm.csv", and "wheat_sm.csv".

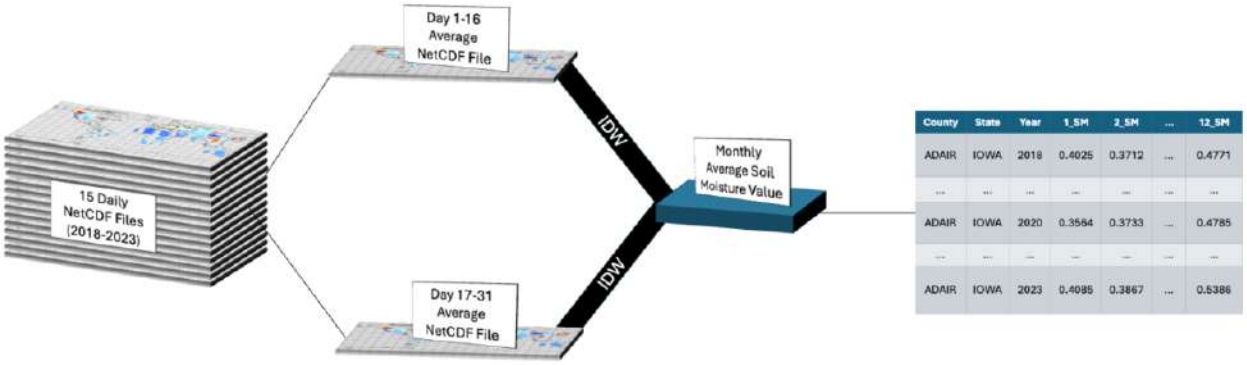


Figure 3-10. Soil Moisture Processing Visualization

During the data processing step, the original soil moisture values, which were expressed as percentages (ranging from 0 to 100), were normalized by dividing by 100 to obtain values ranging from 0 to 1. Since this is a simple normalization of percentage values, it is not expected to significantly affect the interpretation or analysis of the data. As with the previously created CSV files for other climate variables, the soil moisture data was also structured into tables with the following columns: County, State, Year, 1_SM, 2_SM, ..., 12_SM. The corn and soybean files each contain 1,728 rows, while the wheat file contains 888 rows.

County	State	Year	1_SM	2_SM	3_SM	4_SM	5_SM	6_SM	7_SM	8_SM	9_SM	10_SM	11_SM	12_SM
ADAIR	IOWA	2018	0.3889	0.3906	0.4284	0.3989	0.3096	0.2637	0.1873	0.2289	0.3795	0.4902	0.5197	0.4710
ADAIR	IOWA	2019	0.4185	0.3327	0.4112	0.4083	0.3730	0.2614	0.2060	0.2145	0.3509	0.4671	0.5186	0.4976
ADAIR	IOWA	2020	0.4020	0.4021	0.4557	0.3913	0.3258	0.2128	0.1833	0.1999	0.3689	0.4155	0.4970	0.4815
ADAIR	IOWA	2021	0.3596	0.3354	0.4349	0.3827	0.3203	0.2256	0.1852	0.1991	0.3124	0.4156	0.4785	0.4363
ADAIR	IOWA	2022	0.3784	0.4024	0.4351	0.4216	0.3455	0.2293	0.1889	0.1908	0.2930	0.3370	0.4527	0.4410
ADAIR	IOWA	2023	0.4291	0.4178	0.4231	0.3748	0.2925	0.2164	0.1775	0.2044	0.3117	0.4116	0.4569	0.5352
ADAMS	IOWA	2018	0.4453	0.4350	0.4756	0.4500	0.3530	0.2748	0.2101	0.2458	0.3955	0.5147	0.5759	0.5321
ADAMS	IOWA	2019	0.4682	0.3887	0.4551	0.4630	0.4114	0.2950	0.2256	0.2483	0.3725	0.4985	0.5804	0.5472
ADAMS	IOWA	2020	0.4743	0.4492	0.4983	0.4592	0.3732	0.2528	0.2156	0.2277	0.3894	0.4685	0.5651	0.5652
ADAMS	IOWA	2021	0.4323	0.4054	0.4884	0.4505	0.3691	0.2497	0.2191	0.2260	0.3340	0.4558	0.5426	0.5149
ADAMS	IOWA	2022	0.4157	0.4333	0.4881	0.4950	0.3854	0.2541	0.2245	0.2331	0.3213	0.3962	0.5388	0.5255
ADAMS	IOWA	2023	0.4933	0.4722	0.4805	0.4294	0.3294	0.2288	0.2075	0.2292	0.3329	0.4474	0.5168	0.5980

Table 3-5. General Structure of the CSV files for Soil Moisture

3.5. Data Merging

Here is a thorough description of process of how those csv files (crop yield + climate variables) are merged into a single dataset. When merging data, the most important factor is determining the key columns to use. By comparing all the previous tables, it is found that the common columns across all datasets were *County*, *State*, and *Year*. Each of the five climate variables would be merged into the crop yield data one by one using these key columns.

First, since the crop yield datasets already contained the County, State, and Year columns, I began by merging the precipitation data. Precipitation data was not organized by crop type but rather by state, so I first prepared three separate precipitation datasets for each crop. For corn, I combined precipitation data from Illinois, Iowa, and Minnesota and saved as "corn_precipitation.csv". For soybean, I also combined precipitation data from the same three states and saved as "soybean_precipitation.csv". For wheat, I combined precipitation data from North Dakota, Montana, and Washington and saved as "wheat_precipitation.csv".

Before merging with yield data, I renamed the monthly columns from Jan, Feb, Mar, ..., Nov, Dec to 1_Precip, 2_Precip, ..., 12_Precip to clearly indicate the month and variable type. Then, I merged the precipitation data into the crop yield data based on County, State, and Year. At this point, the resulting files contained both crop yield and precipitation data.

Next, I merged the temperature data into these tables. Temperature data was structured similarly to precipitation, stored by state, so I combined the appropriate states for each crop as before. The monthly columns were also renamed, from Jan, Feb, ..., Dec to 1_Temp, 2_Temp, ..., 12_Temp. Then, I merged the temperature data into the existing tables using the same key columns. At this stage, for each crop, I had a table containing crop yield, precipitation, and temperature.

The next step was to add solar radiation data. Unlike the precipitation and temperature, solar radiation and the remaining climate variables (CO_2 and soil moisture) were already organized by crop, not by state. Additionally, their monthly columns were already named according to the format #_Solar, #_CO2, and #_SM, rather than using month abbreviations. Therefore, for solar radiation, CO_2, and soil moisture, the merging process was straightforward: simply merging each variable into the existing table based on the key columns County, State, and Year. After merging the solar radiation data, I added the CO_2 data in the same way, followed by the soil moisture data.

As a result, I created three final CSV files containing both crop yield and climate variable information: “Merged_Corn.csv”, “Merged_Soybean.csv”, “Merged_Wheat.csv”. Each row in these final files represents the monthly climate conditions for a specific county and year. All three files share the same structure, and the following *Table 3-6* illustrates the basic format.

County	State	Year	Yield	1_Temp	...	12_Temp	1_Precip	...	12_Prcip	1_Solar	...	12_Solar	1_CO2	...	12_CO2	1_SM	...	12_SM

Table 3-6. General Structure of the Merged CSV files

3.6. Data Cleaning

Now, I would like to explain the data cleaning process used in this study. Before discussing the details, it is important to clearly define what data cleaning means. Data cleaning refers to the process of identifying and removing errors and inconsistencies in order to improve

the quality of the data[69]. In the context of this study, data cleaning involved refining the three crop-specific CSV files created in *Section 3.5. Data Merging* by eliminating errors and unnecessary information to ensure that the final datasets would better support accurate crop yield predictions.

The data cleaning process in my study consisted of two major steps: removing errors that appeared either during processing or was inherent in the original datasets, and removing irrelevant monthly climate variables based on each crop's planting and harvesting calendar.

First, during the exploration of the processed datasets, significant outliers were discovered in the solar radiation data. Typically, solar radiation values fall within the range of 200 to 400 watts per square foot. However, some values were found to exceed 9,000, which is physically unrealistic. Fortunately, these extreme outliers were relatively rare. Specifically, eight such records were found in the wheat dataset, three records in the corn dataset, and one record in the soybean dataset. To address this issue, I removed the entire rows containing these erroneous values. Then, I confirmed that no similar errors were present in the other climate variables except for solar radiation.

The second part of the data cleaning process focused on the removal of irrelevant data. As discussed previously in *Section 3.3. Crop Variables*, each crop has different planting and harvesting periods. Climate variables from months outside these growing periods are irrelevant to crop development. Therefore, for each crop, I removed monthly climate variables that fell outside of its active growing season.

For corn, the growing season spans from April to November, so the climate variables for January, February, March, and December were removed. For soybean, which grows from May to October, the climate variables for January to April and November to December were removed.

Similarly, for wheat, with a growing season from April to September, the climate variables for January to March and October to December were excluded from the dataset.

After these adjustments, each cleaned table contained only the relevant monthly climate variables aligned with the actual growth periods of the corresponding crop. The resulting cleaned tables provide a more accurate and focused dataset for predicting crop yield, and their basic structure is illustrated in the table shown below.

County	State	Year	Yield	4_Temp	...	11_Temp	4_Precip	...	11_Prcip	4_Solar	...	11_Solar	4_CO2	...	11_CO2	4_SM	...	11_SM
							Corn											

County	State	Year	Yield	5_Temp	...	10_Temp	5_Precip	...	10_Prcip	5_Solar	...	10_Solar	5_CO2	...	10_CO2	5_SM	...	10_SM
							Soybean											

County	State	Year	Yield	4_Temp	...	9_Temp	4_Precip	...	9_Prcip	4_Solar	...	9_Solar	4_CO2	...	9_CO2	4_SM	...	9_SM
							Wheat											

Table 3-7. General Structure of Cleaned CSV files for each Crop

3.7. Explanatory Data Analysis

This section presents the Exploratory Data Analysis (EDA) conducted on the cleaned dataset. The analysis is divided into two parts: first, an examination of the distributions and trends of the variables, and second, an exploration of the correlation matrices for each crop.

3.7.1. Distribution & Trend

Understanding the distribution and trends of the data prior to modeling is essential for several reasons. Visualizing the distribution helps identify biases, outliers, and the normality of the data, while trend patterns reveal how variables change over time and relate to one another. Since the target variable, crop yield, is available on an annual basis, I decided to visualize its distribution. In contrast, the explanatory variables, which are climate-related, are available monthly, so I chose to present their six-year average patterns using trend plots based on monthly averages. I believe these visualizations provide valuable insight into the dataset, especially for readers who do not have direct access to the data.

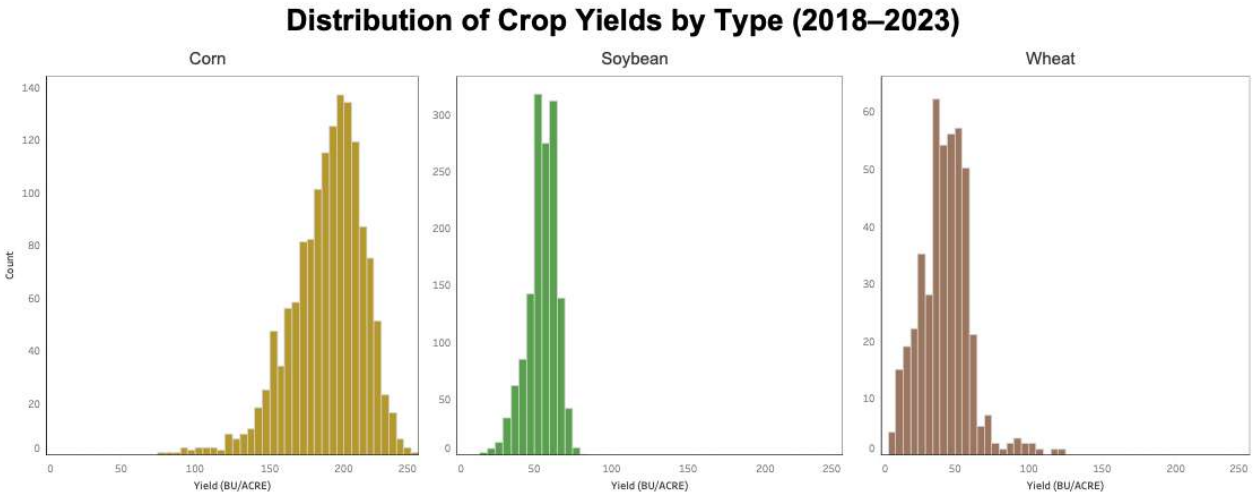


Figure 3-11. Distribution of Crop Yields by Type

Figure 3-11 illustrates the distribution of crop yields by crop type. Corn and soybean exhibit distributions that are slightly right-skewed but generally symmetric. On the other hand, the distribution of wheat is left-skewed and not symmetric. In terms of variability, wheat yields show greater variability compared to corn and soybean.

Figures 3-12 through 3-16 below present the average monthly climate values from 2018 to 2023 for all counties within each state. The blue line indicates Illinois, the orange line

indicates Iowa, the yellow line indicates Minnesota, the green line indicates Montana, the blue-gray line indicates North Dakota, and the purple line indicates Washington.

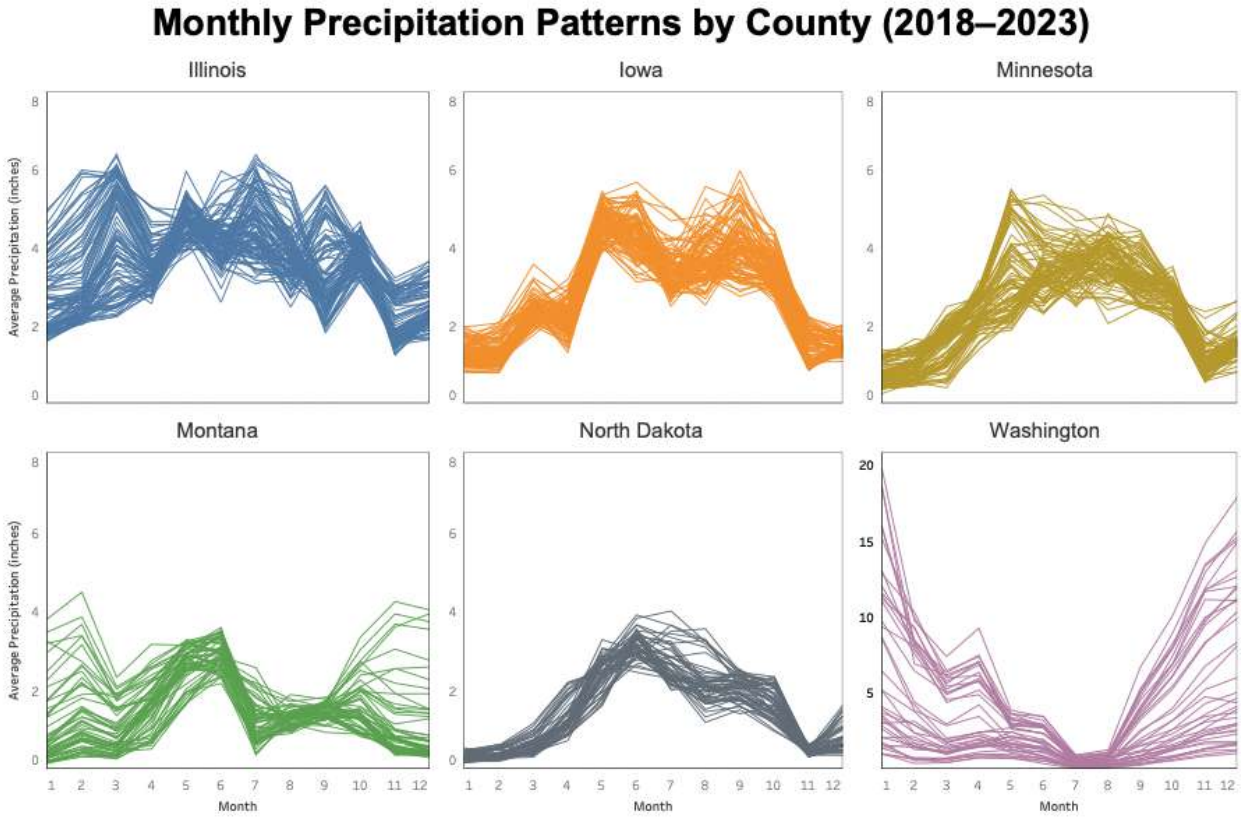


Figure 3-12. Monthtly Precitation Patterns

The graphs above illustrate the trends in precipitation. The x-axis of the graphs represents the months from January to December, while the y-axis indicates the average monthly precipitation in inches. For the counties in Illinois, Iowa, Minnesota, Montana, and North Dakota, the average precipitation over the six-year period generally ranges between 0 and 8 inches. In contrast, Washington exhibits a significantly different pattern, with a substantial number of its counties showing average precipitation levels between 10 and 20 inches, particularly at the beginning and end of the year. One of the most notable observations is that, unlike the other states, Washington exhibits low precipitation levels from May to September, while all the other states show higher precipitation during this period.

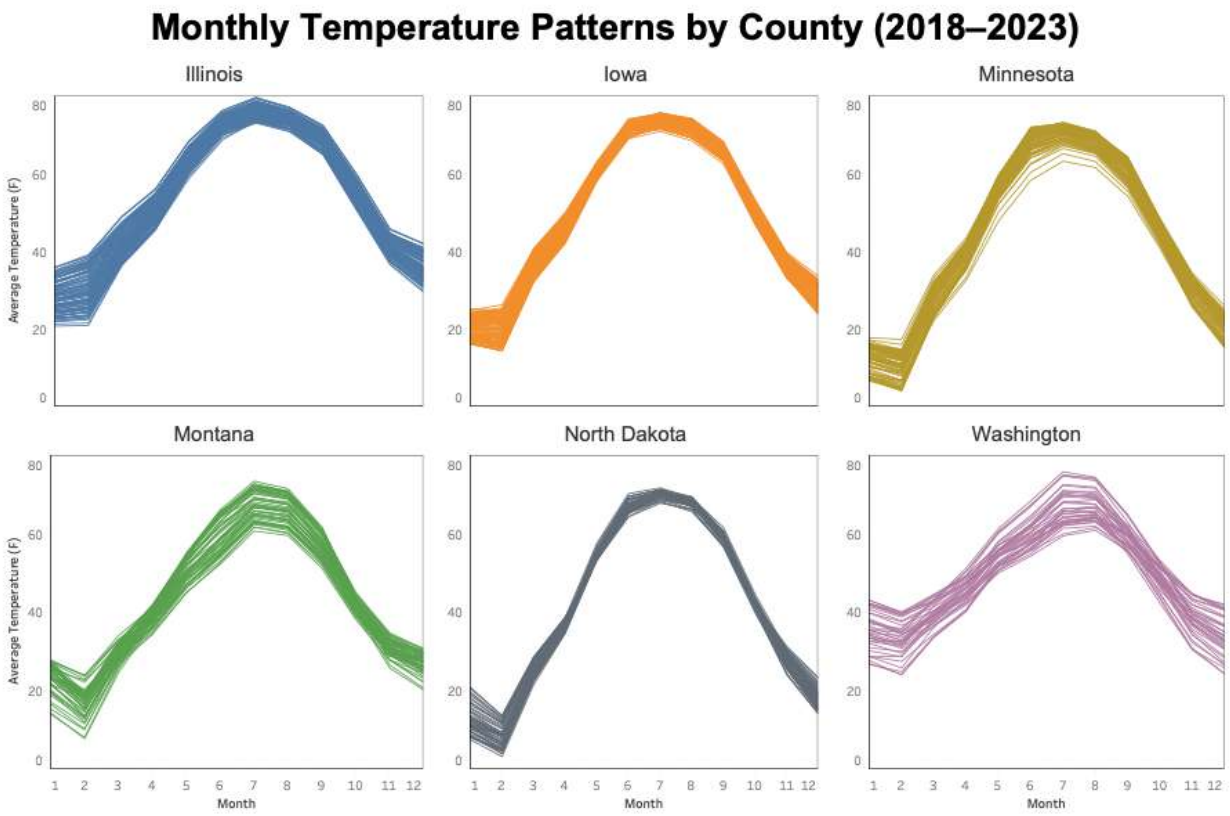


Figure 3-13. Monthtly Temperature Patterns

The graph above shows the temperature trend. The x-axis of the graphs represents the months from January to December, while the y-axis indicates the average temperature in Fahrenheit. Across all counties, the average temperature ranges between 0 and 80 degrees Fahrenheit. Overall, a predictable seasonal pattern is observable across all counties, regardless of the state. Temperatures are low at the beginning of the year, peak during the mid-summer months of July and August, and then gradually decrease toward the end of the year. One notable distinction is that the counties in Washington exhibit relatively warmer temperatures at the beginning and end of the year compared to the other states.

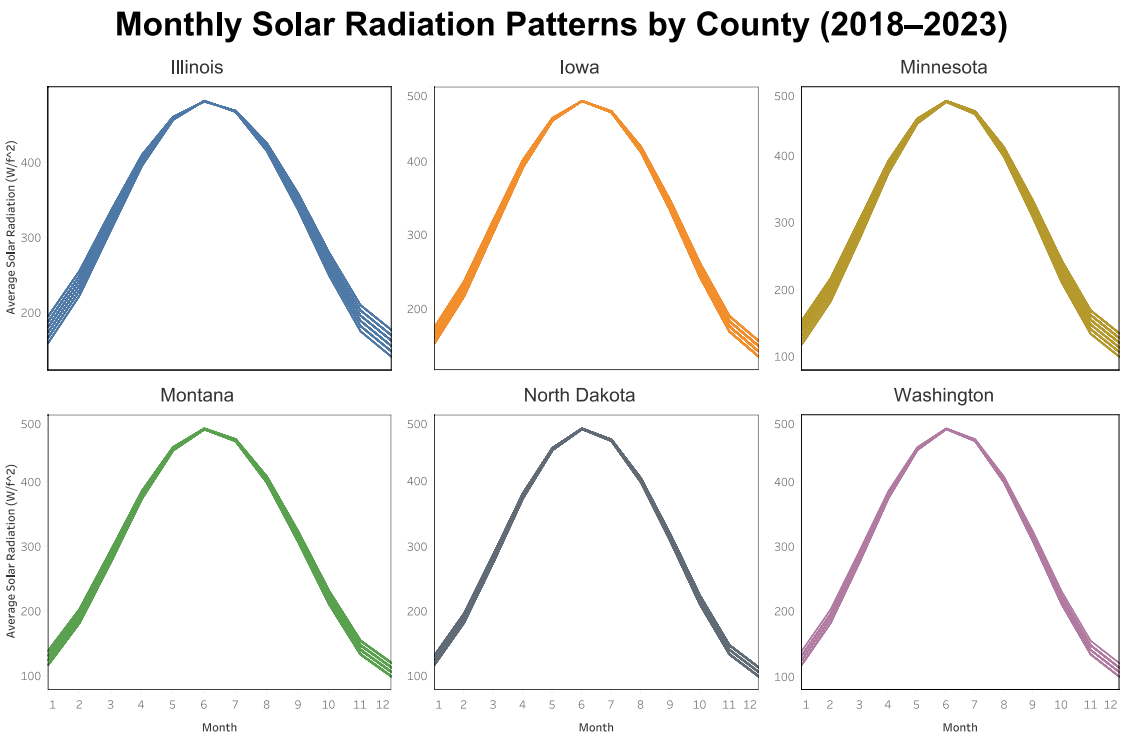


Figure 3-14. Monthtly Solar Radtiation Patterns

The graph above shows the trend of solar radiation. The x-axis of the graphs represents the months from January to December, and the y-axis indicates the average solar radiation in watts per square feet (W/f²), with a range up to 500. Similar to the temperature trends, all states exhibit a strong and predictable seasonal pattern: solar radiation is lowest during the winter months, peaks in the summer, and declines in the fall. One important point to note is that although all counties were included, only about 3 to 6 lines appear per state. This is due to the low resolution of the solar radiation data, which results in many counties sharing the same grid value.

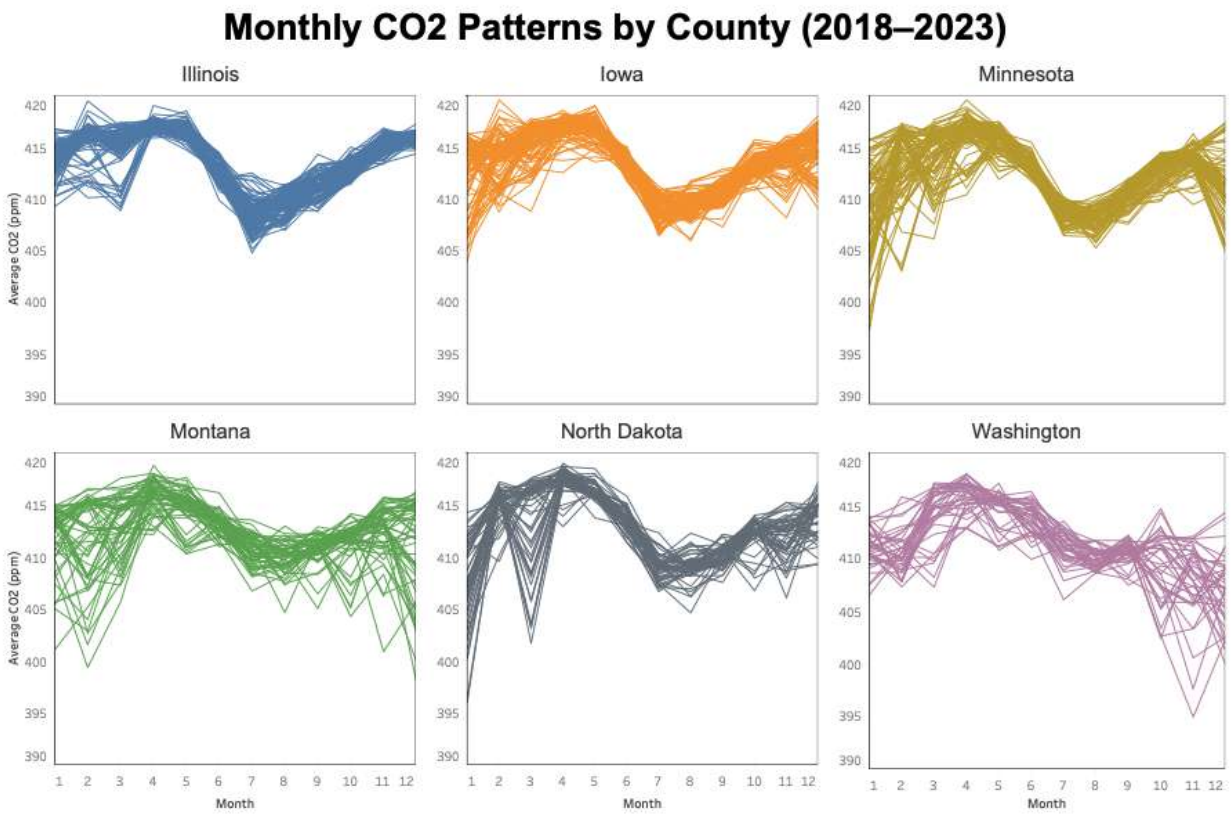


Figure 3-15. Monthtly CO2 Patterns

The graph above illustrates the CO_2 trends. The x-axis of the graphs represents the months from January to December, and the y-axis indicates the average CO2 concentration in parts per million (ppm). Counties in all states, with the exception of Washington, exhibit a similar seasonal trend. CO2 levels generally rise through March, begin to decline around April and May, and then gradually increase again toward the end of the year. Washington, however, displays a different pattern; while its CO2 levels also rise at the beginning of the year, they peak around April and May before beginning their decline.

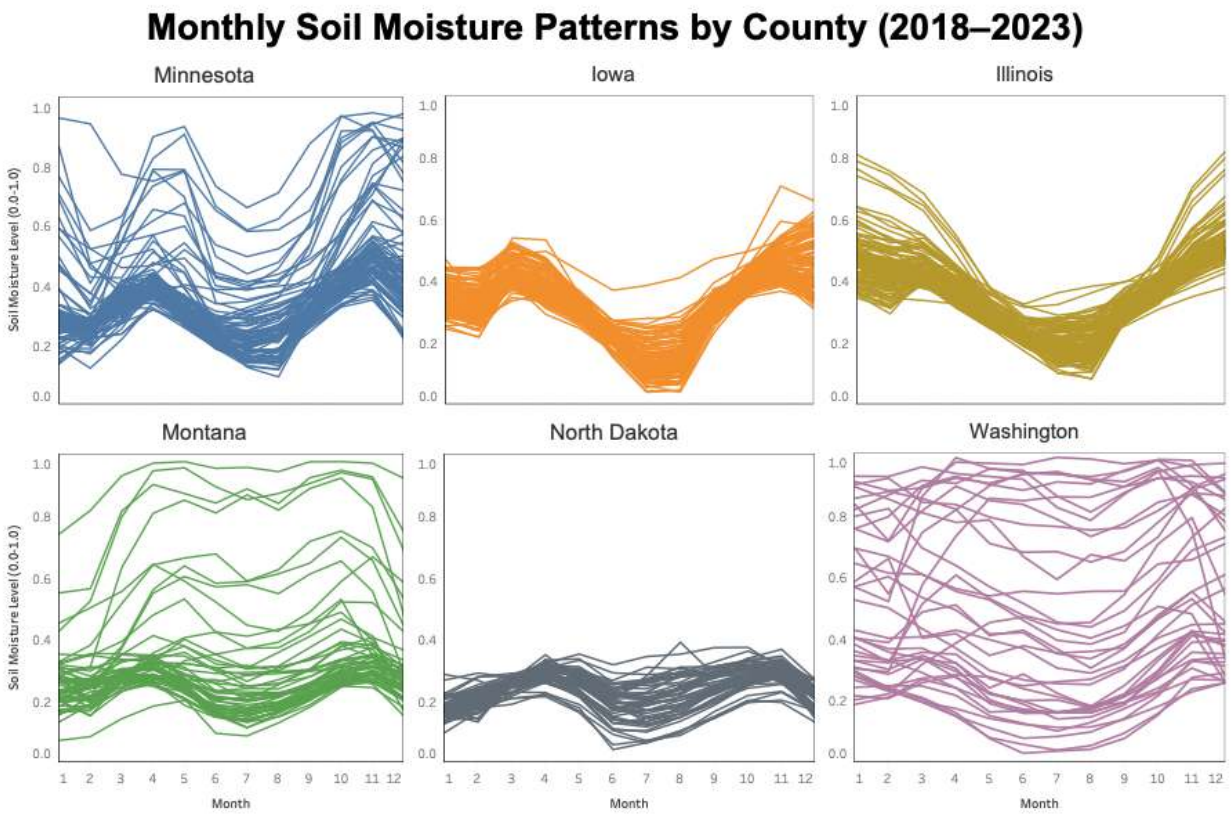


Figure 3-16. Monthtly Soil Moisture

The graph above shows the monthly trends of soil moisture across counties from 2018 to 2023. The x-axis of the graphs represents the months from January to December, and the y-axis indicates the soil moisture level, scaled as a value between 0.0 and 1.0. Counties in all states except for Washington exhibit a similar M-shaped seasonal pattern. Soil moisture is low in January and February, rises to a peak around March and April, drops to its lowest point in July and August, and then briefly rises again in the fall before declining into December. In contrast, the patterns for Washington's counties are highly variable and do not show a single, consistent trend, making them difficult to analyze as a group.

3.7.2. Correlation

The three figures below, Figure 3-17, 3-18, and 3-19, show the correlation matrix of variables for each crop. A correlation matrix helps identify relationships between variables, with

the strength of the relationship represented by the Pearson correlation coefficient (r^2 value). The values range from -1 to 1, where -1 indicates a strong negative correlation between two variables, 0 means there is no linear correlation, and 1 indicates a strong positive correlation between the variables.

There are two patterns that commonly appear across all three figures. First, the diagonal regions always show a correlation of 1, as each variable is perfectly correlated with itself. Second, variables such as CO_2, soil moisture, and solar radiation exhibit strong positive correlations with themselves across different months, regardless of the specific month.

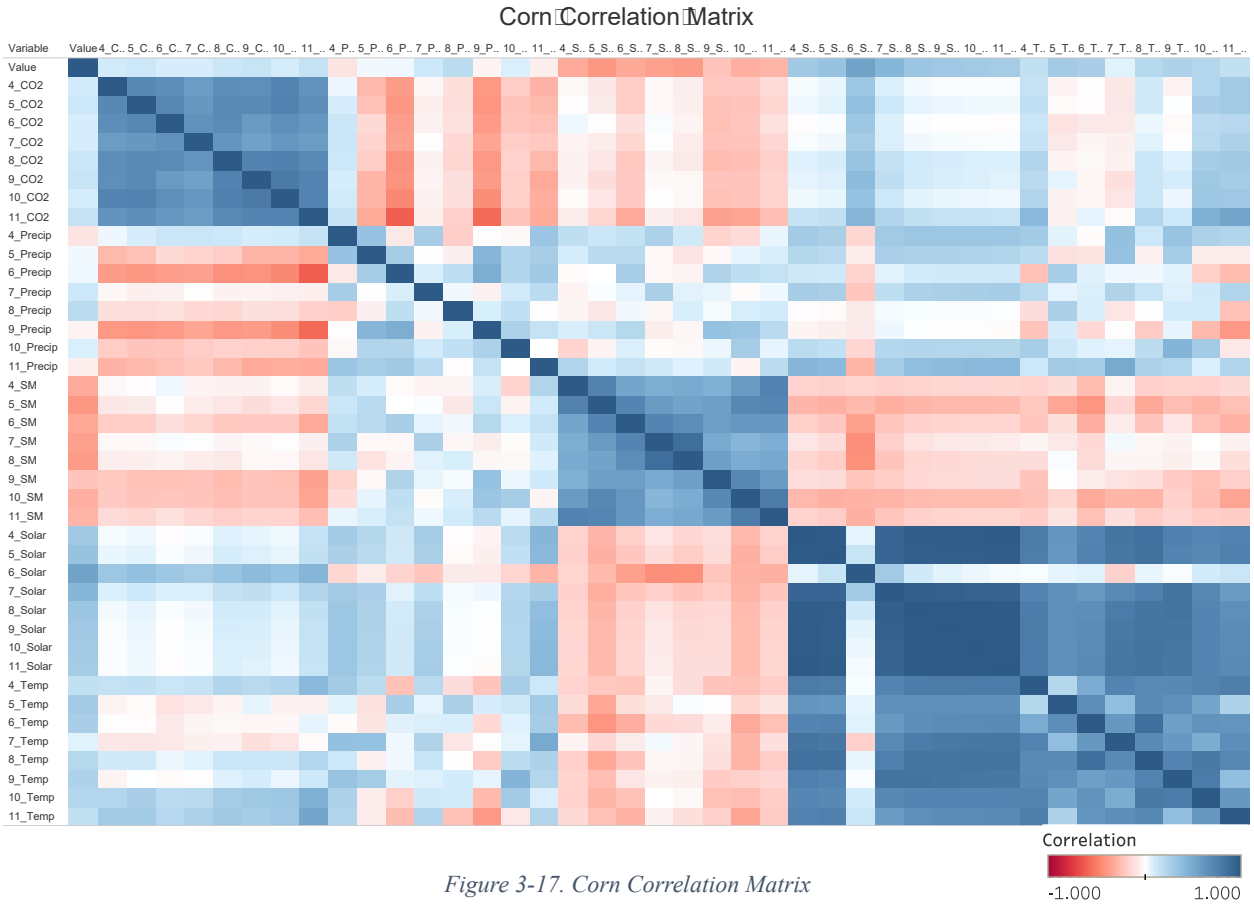


Figure 3-17. Corn Correlation Matrix

The two general patterns described above are also clearly visible in the corn variable correlation matrix. Beyond that, several additional insights can be observed. CO_2 and precipitation show an overall negative correlation, as do CO_2 and soil moisture. Similarly, solar

radiation and soil moisture display negative correlations across all months. Temperature and soil moisture also demonstrate a negative relationship.

When comparing yield with the other variables, we see that the correlation values generally fall between -0.5 and 0.5, indicating that none of the climate variables show a strong linear relationship with yield. One particularly interesting finding is the negative correlation between soil moisture and yield. While it's commonly believed that sufficient water is essential for plant growth, this suggests that higher soil moisture levels might actually have a slightly adverse effect on corn yield. Meanwhile, precipitation shows both weak positive and negative correlations with corn yield depending on the month, whereas CO_2, solar radiation, and temperature tend to be positively correlated with yield overall.

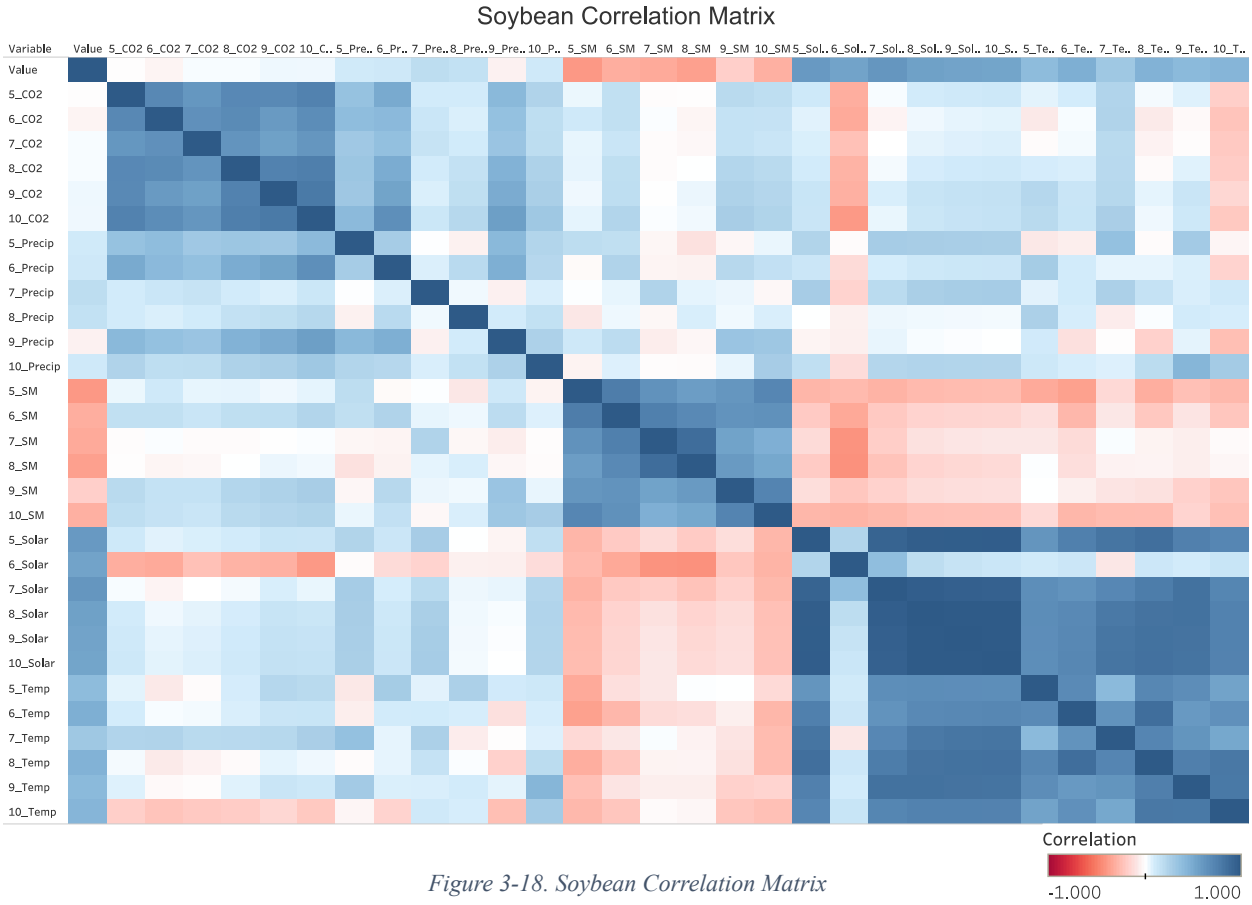


Figure 3-18. Soybean Correlation Matrix

Since the climate variables used for soybean are from the same regions as those used for corn, the relationships among the climate variables exhibit the same patterns previously discussed in the corn correlation matrix.

When examining the relationships between soybean yield and other variables, one key difference emerges: there are correlation values that fall outside the -0.5 to 0.5 range. Specifically, solar radiation shows a strong positive correlation with soybean yield, with an r^2 value greater than 0.5. In contrast, CO_2 has a correlation close to zero with yield, indicating little to no linear relationship. Soil moisture shows weak negative correlation with yield, while precipitation, solar radiation, and temperature all show weak but positive correlations with soybean yield.

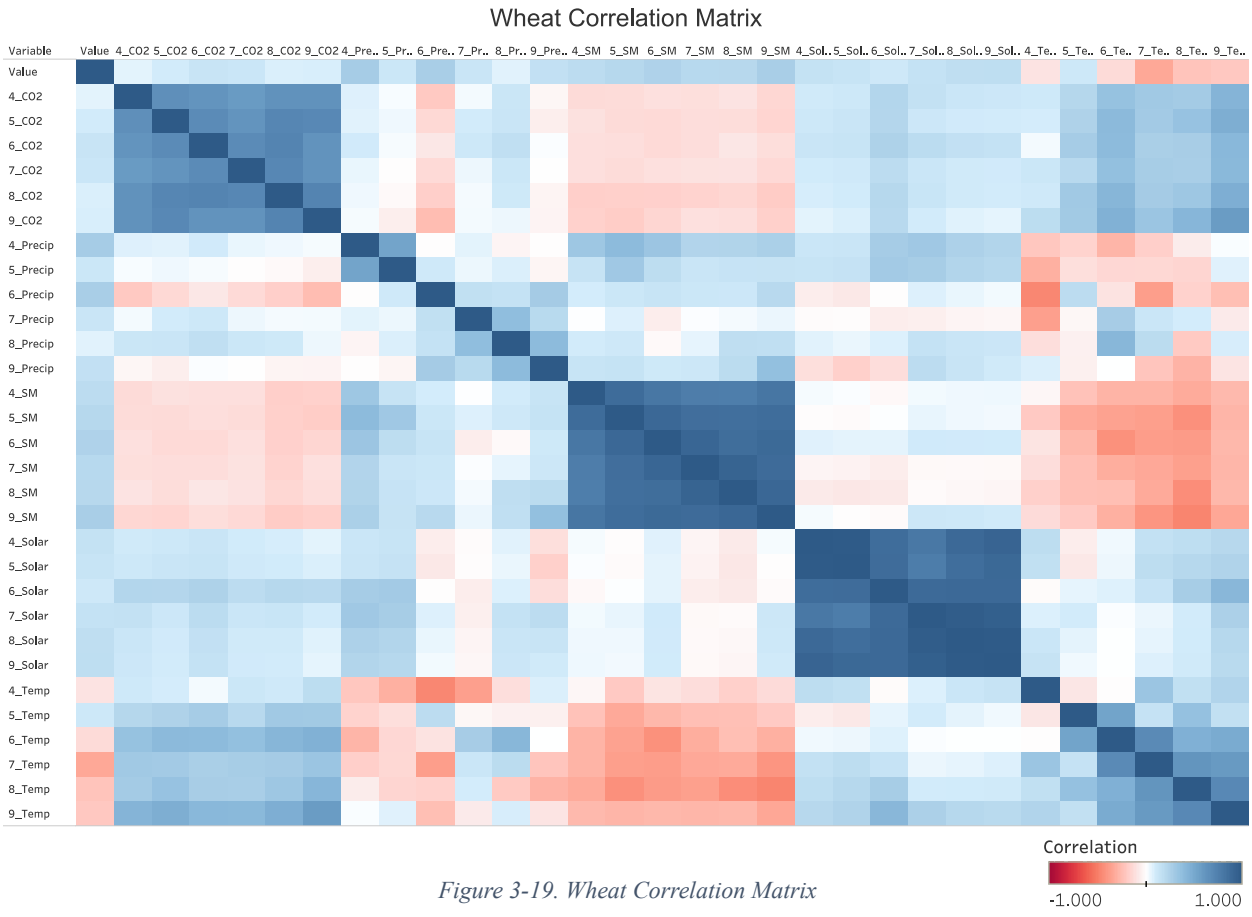


Figure 3-19. Wheat Correlation Matrix

When comparing only the climate variables, soil moisture and CO_2 generally show a weak negative correlation, while CO_2 has a positive correlation with both solar radiation and temperature. CO_2 and precipitation show mixed correlations - positive in some months and negative in others - while precipitation and temperature also display month-to-month variations in the direction of their correlation. Soil moisture and temperature generally show a negative correlation, and soil moisture and solar radiation appear to have almost no correlation, with r^2 values close to zero.

When comparing wheat yield to climate variables, one of the most notable findings is that April temperature, along with temperatures from June through September, shows a negative correlation with yield. Aside from these five variables, most of the remaining climate variables exhibit weak but positive correlations with wheat yield. Similar to the corn matrix, wheat yield shows weak linear correlations with all climate variables, with r-squared values falling between -0.5 and 0.5.

3.8. Modeling & Hyperparameter Tuning

All of the data processing steps and theoretical approaches discussed so far were ultimately aimed at building an effective prediction model. In this section, I explain how the data was modeled, how parameter tuning was performed, and how the model was tested with SHAP value.

3.8.1. Training

In order to accurately predict crop yield using XGBoost in this study, the model was trained on the prepared dataset using an appropriate learning strategy. To evaluate the

performance of the trained XGBoost model, the repeated cross-validation method described in Section 2.7 was used. As previously mentioned, separate models were trained for each crop, and the same training strategy was applied to all three models.

Specifically, 5-fold cross-validation was used, and instead of running it just once, the process was repeated 5 times, resulting in a total of 25 different train-test combinations. This approach provides a more general and reliable estimate of the model's performance

3.8.2. Hyperparameter Tuning

In this study, the Optuna optimization algorithm described in section 2.6, Hyperparameter Optimization (Optuna), was used to identify the optimal set of hyperparameters for each crop-specific model. XGBoost offers a large number of tunable hyperparameters, and if the user does not manually define them, default values are used instead. In this case, a total of eleven hyperparameters were tuned across all three crop models.

- **n_estimators** represents the number of boosting rounds. While a higher value increases the model’s capacity, it also raises the risk of overfitting.
- **learning_rate** controls the step size shrinkage during model updates and helps prevent overfitting. Smaller learning rates slow down training but result in more precise models.
- **max_depth** determines the maximum depth of each tree. Deeper trees can capture more complex patterns in the data.
- **min_child_weight** represents the minimum sum of instance weights (hessian) required in a child node.
- **subsample** refers to the ratio of the training dataset used to grow each tree, introducing randomness and reducing overfitting.

- **colsample_bytree** is the ratio of features randomly sampled for each tree, helping to reduce correlation between trees.
- **gamma** is the minimum loss reduction needed to make a further split in the tree, effectively enabling pruning.
- **reg_alpha** applies L1 regularization, which improves model robustness by shrinking feature weights.
- **reg_lambda** applies L2 regularization, which penalizes large weights to reduce model complexity.
- **scale_pos_weight** adjusts the balance between positive and negative class weights, particularly useful for imbalanced datasets.
- **max_leaves** defines the maximum number of leaves allowed in a tree, affecting its structural complexity.

The best combination was selected based on the lowest Root Mean Squared Error (RMSE) obtained during the tuning process. A total of 80 trials were conducted, meaning that the model tested 80 different hyperparameter conbinations, each evaluated using 5-fold cross-validation. The combination that yielded the lowest average RMSE was chosen for the final modeling.

Corn	Soybean	Wheat
n_estimators = 650, learning_rate= 0.027657231731611654, max_depth= 7, min_child_weight= 10, subsample= 0.6094565060007892, colsample_bytree= 0.4997046091735818, gamma= 2.5618475317290963e-05, reg_alpha= 0.004255342311513287, reg_lambda= 0.17775241538642822, scale_pos_weight= 1.4359596779139445, max_leaves= 169	n_estimators= 950, learning_rate= 0.027981775603789203, max_depth= 13, min_child_weight= 8, subsample= 0.7414864611604867, colsample_bytree= 0.4242319245356668, gamma= 0.00043657507051181755, reg_alpha= 0.17637268839976178, reg_lambda= 79.77511249980523, scale_pos_weight= 1.6283072685102191, max_leaves= 194	n_estimators= 1400, learning_rate= 0.035071328223594225, max_depth= 6, min_child_weight= 5, subsample= 0.5365140857555877, colsample_bytree= 0.5264773514843196, gamma= 6.582750730434762e-08, reg_alpha= 0.003032350088768076, reg_lambda= 29.537679335050647, scale_pos_weight= 0.9411166623933069, max_leaves= 146

Figure 3-20. Final Combinations of Hyperparameters for Three Different Models

3.8.3. Testing & Tree SHAP

As previously described, repeated cross-validation was used, employing 25 different training and testing set combinations. The evaluation metrics for the final models were RMSE (Root Mean Squared Error), NRMSE (Normalized Root Mean Squared Error). and R^2 (Coefficient of Determination). Detailed explanations of each metric can be found in Section 2.9, Evaluation Metrics.

Each hyperparameter combination produced two results - RMSE and R^2 - and the final reported values are the average of those 25 results. This approach enables a more generalized and robust evaluation of model performance.

To enhance model interpretability, Tree SHAP values were used. However, SHAP values are only calculated for data points included in the testing sets. Since every data point appears in the testing set at least once due to the repeated cross-validation setup, all data points have corresponding SHAP values.

4. RESULTS AND DISCUSSION

This chapter presents and discusses the results for each crop model, which were obtained using the processed data, the XGBoost algorithm, and the cross-validation method detailed in Chapter 3. First, the performance of the models is presented through three metrics: R^2, RMSE, and NRMSE. Following this, the models are interpreted using SHAP plots to understand their underlying predictive mechanisms. The implications of these results are then discussed. Finally, the chapter concludes by addressing the limitations faced during this study.

4.1. Model Performance

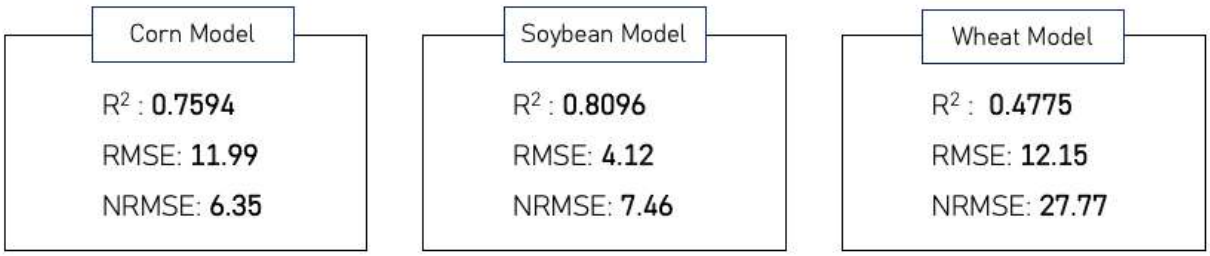


Figure 4-1. Model Performance by Crops

Figure 4-1 summarizes the evaluation metrics for each crop model. The corn model achieved an R^2 of 0.7594 and an RMSE of 11.99, recording the lowest NRMSE among the three models at 6.35%. The soybean model demonstrated the strongest performance, with the highest R^2 value of all models at 0.8096, an RMSE of 4.12, and an NRMSE of 7.46%. The wheat model yielded an R^2 of 0.4775, an RMSE of 12.15, and an NRMSE of 27.77%.

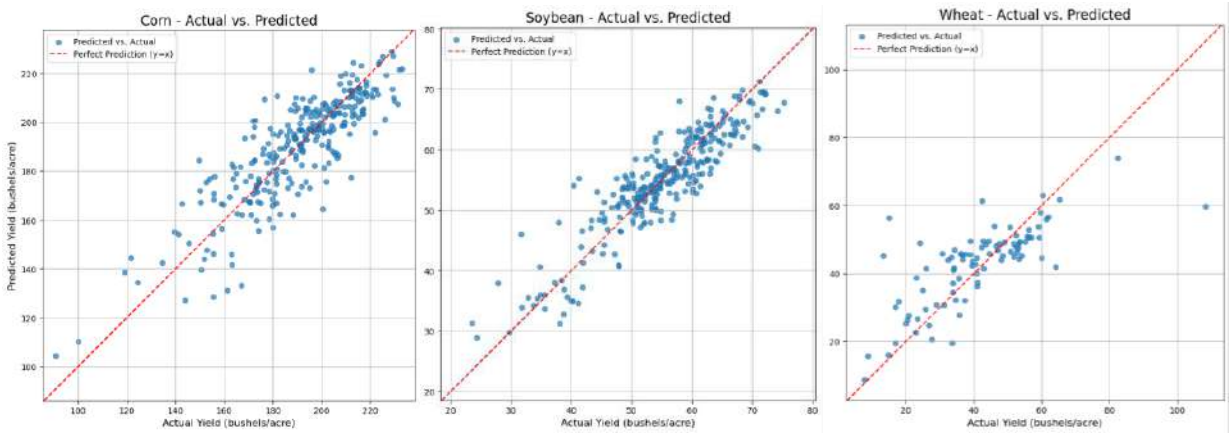


Figure 4-2. Scatterplot of actual versus predicted yield for model by crops

To supplement the numerical results, the model's performance is also presented visually. Figure 4-2 above displays scatterplots that compare the predicted values against the actual values for each crop model. The left, middle, and right plots correspond to the corn, soybean, and wheat models, respectively.

All three plots share a common structure: the x-axis represents the actual yield, and the y-axis represents the predicted yield. The diagonal line indicates a perfect prediction (y=x), and the proximity of the data points to this line serves as a visual indicator of model performance.

The quantitative results from Figure 4-1 are clearly reflected in these scatterplots. For the high-performing corn and soybean models, the points are tightly clustered around the red line. In contrast, the points for the lower-performing wheat model are more dispersed and deviate further from the line, visually confirming its weaker performance.

4.2. Feature Importance using SHAP

As discussed in the introduction, machine learning models are often considered 'black boxes' due to their lack of interpretability. However, this study utilizes the SHAP, especially the Tree SHAP, framework to analyze how each of the three models arrives at its predictions.

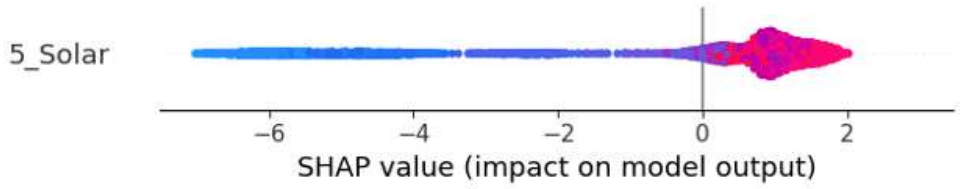


Figure 4-3. Example of the SHAP Summary Plot

SHAP values are calculated for every data point within the testing set. As this study employed cross-validation, each data point was part of a testing set at least once, allowing for the generation of SHAP values across the entire dataset. Before presenting the specific feature importance for each model, this section will first explain how to interpret the SHAP summary plot, which is used for visualizing these values.

Figure 4-3 above serves as an example of a summary plot. On the y-axis, features are ranked by their overall impact, with the most important feature at the top. The feature name, such as "5_Solar," indicates the variable (e.g., solar radiation in May). The x-axis represents the SHAP value. A positive SHAP value indicates a positive contribution to the prediction (i.e., it pushed the predicted yield higher), while a negative value indicates a negative contribution (i.e., it pushed the prediction lower). The color of each point on the plot corresponds to the feature's value for that specific data instance. Typically, red indicates a high feature value, and blue indicates a low feature value.

To interpret this example plot for corn yield prediction, one would observe the following: the blue points (representing low values of May solar radiation) are clustered on the negative side of the x-axis, suggesting a negative relationship with the predicted yield. Conversely, the red points (high values of May solar radiation) are on the positive side. Therefore, this plot can be interpreted to mean that low May solar radiation had a negative association with the predicted corn yield, while high May solar radiation had a positive association.

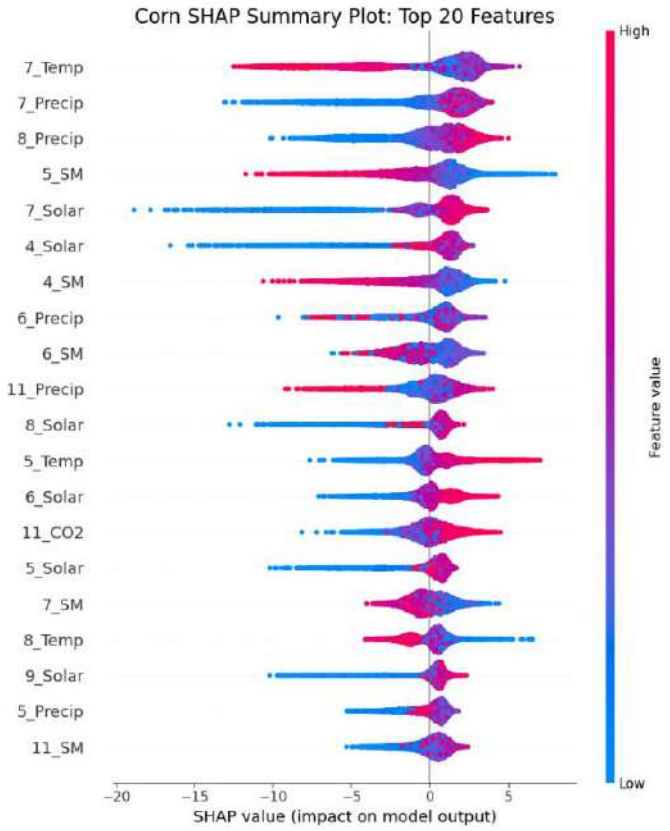


Figure 4-4. Corn SHAP Summary Plot

Figure 4-4 displays 20 features from SHAP summary plots for the corn model. The 20 features out of total 41 features are ranked in descending order of importance, which is determined by the mean absolute SHAP value. The five most important features for predicting corn yield were identified as: July temperature, July precipitation, August precipitation, May soil moisture, and July solar radiation.

A closer look at the plot reveals their specific impacts. High temperatures in July showed a negative relationship with the predicted yield. Low precipitation in both July and August also had a negative impact on the prediction. High soil moisture in May was associated with a negative impact. For July solar radiation, low values had a negative impact, while high values had a positive impact on the predicted yield.

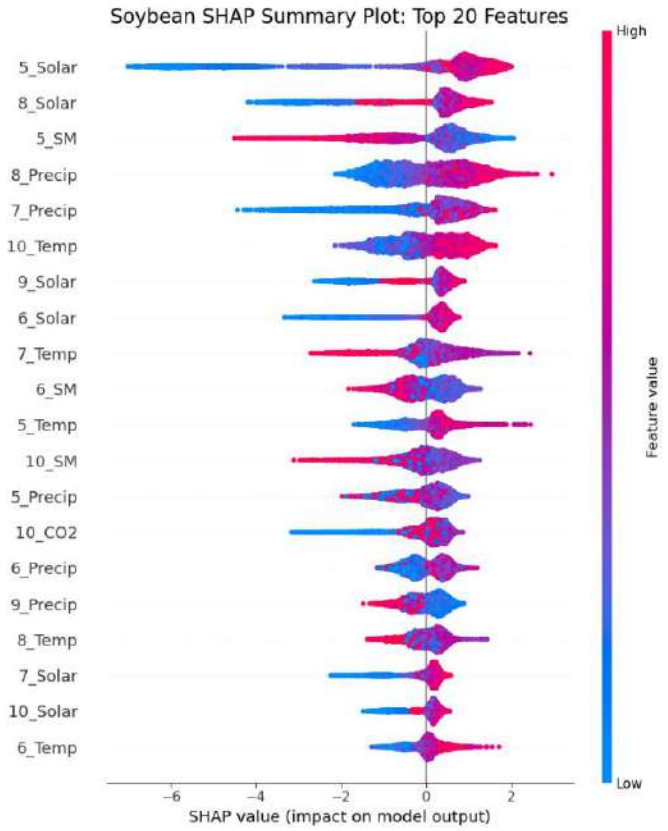


Figure 4-5. Soybean SHAP Summary Plot

Similar to the previous figure, Figure 4-5 presents the SHAP Summary Plot for the soybean model. Only top 20 features out of total 31 features are ranked by importance, and the five most critical features for predicting soybean yield were May solar radiation, August solar radiation, May soil moisture, August precipitation, and July precipitation, respectively.

An interpretation of these top features reveals the following findings. May solar radiation showed a clear positive relationship; low levels had a negative impact on the predicted yield, while high levels had a positive impact. August solar radiation showed a strong positive relationship with predicted yield. Low values of solar radiation consistently had a negative impact on the prediction of yield, while high values had a positive impact.

High levels of May soil moisture had a consistently negative impact on the prediction. August and July precipitation displayed a similar pattern to each other: low levels of rainfall had a negative impact, while very high levels also had a negative impact on the predicted yield.

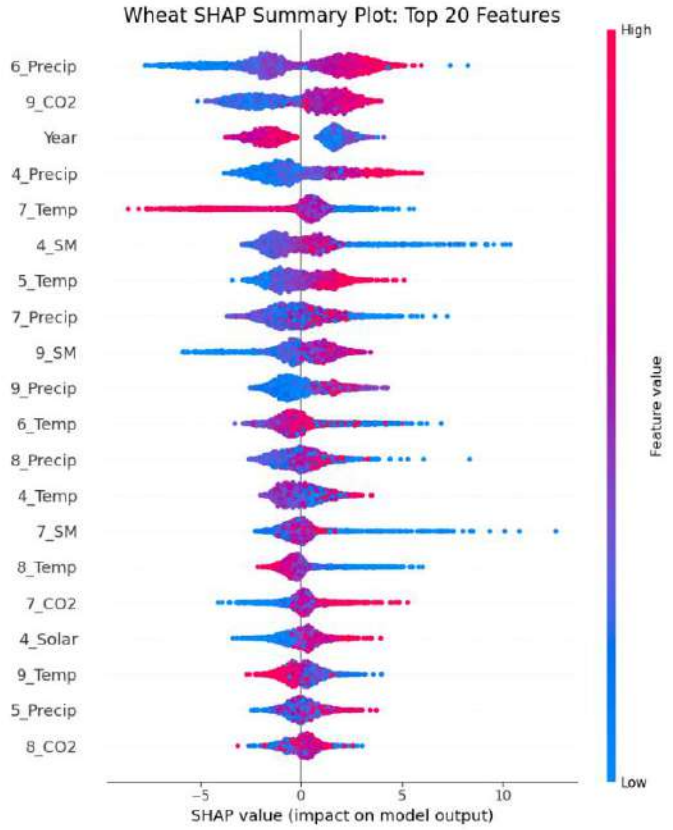


Figure 4-6. Wheat SHAP Summary Plot

Figure 4-6 presents the SHAP summary plot for the wheat model. Similarly, top 20 features out of total 31 features are ranked by their importance in predicting wheat yield. The five most influential variables were June precipitation, September CO_2 level, Year, April precipitation, and July temperature. It is important to note that because the wheat model exhibited lower performance compared to the other two models, the interpretations from this summary plot may include some inaccuracies.

Analyzing each feature reveals the followings. Low June precipitation had a negative impact on the predicted wheat yield, while high precipitation had a positive impact. Low

September CO_2 levels had a negative effect, whereas high CO_2 levels had a positive effect. The Year feature, which spans from 2018 to 2023, showed an interesting trend. Lower feature values (i.e., earlier years like 2018 and 2019) had a positive association with the prediction, while higher values (more recent years) had a negative association.

Similar to June, low April precipitation had a negative impact, and high precipitation had a positive impact. High July temperature was associated with a negative effect on the prediction, while low temperature was associated with a positive effect.

While it is true that all three crops have different harvesting periods and unique growth requirements, a comparison of their SHAP summary plots reveals a notable commonality: features from the middle of the year—specifically June, July, and August—consistently play a crucial role in the yield predictions.

4.3. Discussion

As detailed in Chapter 3, the models are compared using three primary metrics. A model is considered to have strong performance when its R^2 value approaches 1 and its RMSE and NRMSE values are low.

However, in this study, a direct comparison of absolute RMSE values between the models is not an appropriate method for evaluation. This is because, while the unit (bushel per acre) of yield is the same for all three crops, their average production volumes differ significantly. Consequently, a similar magnitude of error would naturally result in a higher RMSE for a crop with a higher average yield.

Therefore, evaluating the models based on R^2 and NRMSE, the results indicate that soybean is the crop best explained by the five climate variables (precipitation, soil moisture,

solar radiation, CO_2, and temperature), followed closely by corn. Conversely, wheat appears to be the most challenging crop to predict using this set of variables.

For the high-performing corn and soybean models, the discussion will compare the study's SHAP interpretations with the findings of existing studies to determine whether they align or diverge. Conversely, for the lower-performing wheat model, I will consult previous research to investigate why the five selected climate variables may have been insufficient for accurate prediction.

A common trend observed in the well-trained corn and soybean models is that higher precipitation in July and August led to higher yields, while higher temperatures resulted in lower yields. This tendency for both corn and soybean aligns with the findings of a study by Leng et al. (2016)[70], which focused on the relationship between crop production and the growing season's (June, July, August) temperature, precipitation, and radiation.

The study by Leng et al. concluded that high temperatures generally exerted a negative impact on the yields of both corn and soybean. The fact that this same trend is observable in my models suggests that they are capable of effectively explaining key aspects of corn and soybean growth.

Another feature warranting discussion is the relationship between soil moisture and the yields of corn and soybean. A common trend observed in both SHAP plots is that low soil moisture levels during the harvest period had a positive impact on the predicted yield, whereas high soil moisture had a negative impact.

The results from existing studies, such as Vennam (2023)[71], help to contextualize the general sensitivity of corn to soil moisture stress throughout its life cycle. Given that the interpretations for both the corn and soybean models align well with trends identified in previous research, it can be concluded that the explanations generated by these high-performing models

are supported by scientific evidence. The discussion will now shift to an analysis of the wheat model to investigate why it produced less accurate results.

Two hypotheses can be proposed to explain the weaker performance of the Wheat model compared to the other crop models. The first hypothesis is that the five selected climate variables are insufficient for accurately predicting wheat production. For instance, previous research by Stella et al. (2023)[72] has indicated that the presence of irrigation can have a more significant impact on wheat production than other environmental variables. Furthermore, another study by Yang et al. (2023)[73] suggests that while wheat is sensitive to climate factors, management practices are a particularly critical variable for this crop. In light of these studies, the fact that our study's wheat yield data includes both irrigated and non-irrigated production is considered a disadvantage for predictive accuracy.

Other research by Kaium et al. (2025)[74] also notes that wheat growth is heavily dependent on soil quality. While our study includes soil moisture as a variable, it does not account for other soil characteristics such as texture or organic matter content. For these reasons, the first hypothesis—that the five selected variables are insufficient for predicting wheat yield—appears to be valid.

The second hypothesis is regional heterogeneity. The geography of the three selected wheat-producing states is not uniform. While the corn and soybean states of Illinois and Iowa are relatively flat and climatically homogeneous, the wheat-producing state of Montana, for example, contains both mountains and valleys. This geographical diversity reduces the reliability of using a single centroid coordinate to represent each county. Due to significant intra-county variations, a centroid may not accurately represent the environmental variables for the entire county.

Based on these two hypotheses of missing variables and regional heterogeneity, it is evident that additional variables are needed to model wheat production accurately. Considering the geographical aspects, variables such as altitude may be necessary. Having explored the reasons for the Wheat model's weaker performance, I believe that a more robust model could be developed by incorporating these additional factors.

4.4. Application

Plant growth is governed by a myriad of interconnected relationships, and the result of the relationships is the fruits that we highly depend on for our food supply. Food security is of paramount importance, and any threat to the food is intrinsically linked to human well-being. However, anthropogenic activities have led to an increasingly unstable climate, creating unfavorable conditions for plant growth.

Through this study, we have arrived at three significant findings. Although the research focused on just three crops and yielded conclusive results primarily for corn and soybean, these findings are valuable, considering these grains are major staples of the global human diet.

First, this study demonstrates the potential to predict crop yields that were previously difficult to forecast due to unpredictable climate conditions. Second, it reveals that a reasonable prediction of crop growth can be achieved using a limited set of just five key variables, out of a countless number of factors involved in plant growth. Third, the analysis identifies which specific variables are most impactful during critical periods of each crop's development.

I hope these findings will be applied in various ways in the future. Two potential applications are food security management and decision support for farmers. Government agencies, such as the USDA, could utilize this model to generate precise, county-level yield

forecasts. This would enable them to proactively identify potential regional shortages or surpluses, aiding in strategic decisions regarding grain reserves and import/export policies. Furthermore, the relationships revealed through SHAP values could help farmers make more informed management decisions, such as determining optimal irrigation timing during critical growth stages.

4.5. Limitation

As with any research, this study is not without its limitations. Several challenges were encountered during the research process, which can be summarized into four main categories: the scope of variables, data resolution and representation, uncertainty from data processing, and the temporal scope.

The first limitation is the scope of variables. This study aimed to predict crop yield using five climate-related variables. These were selected based on existing studies and discussion with professors due to their close relationship with plant growth. However, it is certain that not all factors essential for plant growth were included. This limited scope of variables is considered a primary limitation.

The second limitation relates to data resolution and representation. The resolution of the solar radiation data was notably coarser than that of the other satellite data for CO_2 and soil moisture, which likely resulted in less precise solar radiation values at the county level. Furthermore, as discussed previously, a single centroid does not perfectly represent an entire county. While we proceeded with the assumption that it was a reasonable proxy, this represents a potential source of error.

The third limitation is the uncertainty from data processing. For the CO_2 and soil moisture data, the raw satellite data was processed using Inverse Distance Weighting (IDW) to handle gaps. While this processing step addresses shortcomings in the raw data, the resulting interpolated values are estimations, not direct measurements, and cannot convey the information for a given location with perfect accuracy. This introduces a layer of uncertainty into the model.

The final limitation is the temporal scope of the study. Initially, the 6-year period was deemed sufficient for model training, and as the results show, two of the models did indeed achieve strong performance. However, it is conceivable that a more robust model could have been developed if a longer time series had been available. For this reason, the study's timeframe is identified as the final limitation.

5. CONCLUSION

This study aimed to develop models for predicting the yields of corn, soybean, and wheat using five key climate-related variables: temperature, precipitation, solar radiation, CO_2, and soil moisture. Using an Extreme Gradient Boosting (XGBoost) algorithm, separate predictive models were created for each crop. The models for soybean and corn demonstrated strong performance, while the wheat model was less accurate in comparison. Through SHAP analysis, this research identified that climate conditions during the summer growing season, particularly temperature and precipitation, played the most influential roles in the predictions.

The primary contribution of this study lies in addressing the "black box" problem, a persistent issue in machine learning, through the application of SHAP values. While many existing studies focus solely on quantifying a model's predictive performance, this research went a step further. By utilizing SHAP, I was able to interpret the model's decision-making process, identifying not only which variables the model deemed important but also how each variable specifically impacted the predictions.

Despite the satisfactory results, this research faced several limitations. Future work could develop a more robust crop yield prediction model by incorporating a wider range of variables, improving data resolution, reducing the uncertainty from data processing, and extending the temporal scope beyond the current six-year period.

Finally, I hope the findings of this study can aid various sectors. While the forecasting model may seem trivial to some, it is crucial to remember that we rely on food as our fuel. In the current era of environmental instability and unpredictability, crop production has become increasingly volatile. If this research can help address the challenges of food security and provide valuable insights to the agricultural field, I believe it will have served a meaningful purpose.

6. WORK CITED

[1] Ritchie, Hannah, Lucas Rodés-Guirao, Edouard Mathieu, Marcel Gerber, Esteban Ortiz-Ospina, Joe Hasell and Max Roser (2023) - "Population Growth" Published online at OurWorldinData.org. https://ourworldindata.org/population-growth

[2] United Nation at un.org. https://www.un.org/en/global-issues/population

[3] Dasgupta, S., Robinson, E.J.Z. Attributing changes in food insecurity to a changing climate. Sci Rep 12, 4709 (2022). https://doi.org/10.1038/s41598-022-08696-x

[4] Walsh, M.K., et al. (2020). Climate indicators for agriculture (pdf) (29.1 MB). USDA Technical Bulletin 1953. Washington, DC, p. 1

[5] Ebi KL, Vanos J, Baldwin JW, Bell JE, Hondula DM, Errett NA, Hayes K, Reid CE, Saha S, Spector J, Berry P. Extreme Weather and Climate Change: Population Health and Health System Implications. Annu Rev Public Health. 2021 Apr 1;42:293-315. doi: 10.1146/annurev-publhealth-012420-105026. Epub 2021 Jan 6. PMID: 33406378; PMCID: PMC9013542.

[6] Al-Adhaileh MH, Aldhyani THH. Artificial intelligence framework for modeling and predicting crop yield to enhance food security in Saudi Arabia. PeerJ Comput Sci. 2022 Sep 30;8:e1104. doi: 10.7717/peerj-cs.1104. PMID: 36262130; PMCID: PMC9575863.

[7] Ruß, G., Kruse, R., Schneider, M., Wagner, P. (2008). Data Mining with Neural Networks for Wheat Yield Prediction. In: Perner, P. (eds) Advances in Data Mining. Medical Applications, E-Commerce, Marketing, and Theoretical Aspects. ICDM 2008. Lecture Notes in Computer Science(), vol 5077. Springer, Berlin, Heidelberg. https://doi.org/10.1007/978-3-540-70720-2_4

[8] Jeong JH, Resop JP, Mueller ND, Fleisher DH, Yun K, et al. (2016) Random Forests for Global and Regional Crop Yield Predictions. PLOS ONE 11(6): e0156571. https://doi.org/10.1371/journal.pone.0156571

[9] MATSUMURA K, GAITAN CF, SUGIMOTO K, CANNON AJ, HSIEH WW. Maize yield forecasting by linear regression and artificial neural networks in Jilin, China. The Journal of Agricultural Science. 2015;153(3):399-410. doi:10.1017/S0021859614000392

[10] Gandhi, N., et al. "Rice Crop Yield Prediction in India Using Support Vector Machines." *2016 13th International Joint Conference on Computer Science and Software Engineering (JCSSE)*, IEEE, 2016, pp. 1-5, doi:10.1109/JCSSE.2016.7748856.

[11] Su, Y.-X., et al. "Support Vector Machine-Based Open Crop Model (SBOCM): Case of Rice Production in China." *Saudi Journal of Biological Sciences*, vol. 24, no. 3, 2017, pp. 537-44, doi:10.1016/j.sjbs.2017.01.024.

[12] Everingham, Y., et al. "Accurate Prediction of Sugarcane Yield Using a Random Forest Algorithm." *Agronomy for Sustainable Development*, vol. 36, no. 27, 2016, doi:10.1007/s13593-016-0364-z.

[13] Fernandes, J. L., et al. "Sugarcane Yield Prediction in Brazil Using NDVI Time Series and Neural Networks Ensemble." *International Journal of Remote Sensing*, vol. 38, no. 16, 2017, pp. 4631-44, doi:10.1080/01431161.2017.1325531.

[14] Črtomir, R., et al. "Application of Neural Networks and Image Visualization for Early Forecast of Apple Yield." *Erwerbs-Obstbau*, vol. 54, 2012, pp. 69-76, doi:10.1007/s10341-012-0162-y.

[15] Torgbor, B. A., et al. "Integrating Remote Sensing and Weather Variables for Mango Yield Prediction Using a Machine Learning Approach." *Remote Sensing*, vol. 15, no. 12, 2023, p. 3075, doi:10.3390/rs15123075.

[16] Baral, S., et al. "Yield Prediction Using Artificial Neural Networks." *Communications in Computer and Information Science*, edited by V.V. Das, et al., vol. 142, Springer, 2011, pp. 1-6, doi:10.1007/978-3-642-19542-6_57.

[17] Çakir, Y., et al. "Yield Prediction of Wheat in South-East Region of Turkey by Using Artificial Neural Networks." *2014 The Third International Conference on Agro-Geoinformatics*, IEEE, 2014, pp. 1-4, doi:10.1109/Agro-Geoinformatics.2014.6910609.

[18] You, J., et al. "Deep Gaussian Process for Crop Yield Prediction Based on Remote Sensing Data." *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 31, no. 1, 2017, doi:10.1609/aaai.v31i1.11172.

[19] Shastry, Aditya, et al. "Prediction of Crop Yield Using Regression Techniques." *International Journal of Soft Computing*, vol. 12, no. 2, 2017, pp. 96-102.

[20] Gonzalez-Sanchez, A., et al. "Predictive Ability of Machine Learning Methods for Massive Crop Yield Prediction." *Spanish Journal of Agricultural Research*, vol. 12, no. 2, 2014, pp. 313-28, doi:10.5424/sjar/2014122-4439.

[21] U. K, and M. R. "Predicting Crop Yield Based on Stacking Ensemble Model in Machine Learning." *2024 8th International Conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud) (I-SMAC)*, IEEE, 2024, pp. 1831-36, doi:10.1109/I-SMAC61858.2024.10714785.

[22] Shahhosseini, M., et al. "Forecasting Corn Yield with Machine Learning Ensembles." *Frontiers in Plant Science*, vol. 11, 2020, doi:10.3389/fpls.2020.01120.

[23] Huber, F., et al. "Extreme Gradient Boosting for Yield Estimation Compared with Deep Learning Approaches." *Computers and Electronics in Agriculture*, vol. 198, 2022, p. 107346, doi:10.1016/j.compag.2022.107346.

[24] Li, Y., et al. "A County-Level Soybean Yield Prediction Framework Coupled with XGBoost and Multidimensional Feature Engineering." *International Journal of Applied Earth Observation and Geoinformation*, vol. 122, 2023, p. 103269, doi:10.1016/j.ijag.2023.103269.

[25] United States Department of Agriculture, National Agricultural Statistics Service. *Crop Production 2023 Summary*. Jan. 2024.

[26] Van Klompenburg, T., et al. "Crop Yield Prediction Using Machine Learning: A Systematic Literature Review." *Computers and Electronics in Agriculture*, vol. 177, 2020, p. 105709, doi:10.1016/j.compag.2020.105709.

[27] Lindsey, R., and Dahlman, L. "Climate Change: Global Temperature." *Climate.gov*, 16, 2020, pp. 1-5.

[28] Went, F. W. "The Effect of Temperature on Plant Growth." *Annual Review of Plant Physiology*, vol. 4, 1953, pp. 347-62, doi:10.1146/annurev.pp.04.060153.002023.

[29] Trenberth, Kevin E. "Changes in Precipitation with Climate Change." *Climate Research*, vol. 47, no. 1-2, 2011, pp. 123-38.

[30] Feldman, A. F., et al. "Plant Responses to Changing Rainfall Frequency and Intensity." *Nature Reviews Earth & Environment*, vol. 5, 2024, pp. 276-94, doi:10.1038/s43017-024-00534-0.

[31] Yin, J., et al. "Impacts of Solar Intermittency on Future Photovoltaic Reliability." *Nature Communications*, vol. 11, no. 4781, 2020, doi:10.1038/s41467-020-18602-6.

[32] NOAA Climate.gov. "Climate Change: Atmospheric Carbon Dioxide." *Climate.gov*, 14 Feb. 2024, www.climate.gov/news-features/understanding-climate/climate-change-atmospheric-carbon-dioxide. Accessed 27 Feb. 2025.

[33] Hoeppner, Susanne S., and Jeffrey S. Dukes. "Interactive Responses of Old-Field Plant Growth and Composition to Warming and Precipitation." *Global Change Biology*, vol. 18, no. 5, 2012, pp. 1754-68, doi:10.1111/j.1365-2486.2012.02674.x.

[34] Riebeck, Holli. "Catalog of Earth Satellite Orbits." *NASA Earth Observatory*, 4 Sept. 2009, earthobservatory.nasa.gov/features/OrbitsCatalog/page1.php.

[35] NASA Earthdata. "Remote Sensing." *NASA Earthdata*, www.earthdata.nasa.gov/learn/earth-observation-data-basics/remote-sensing.

[36] Ramadhi, Almi. "Satellite Data Sensor: Passive Sensors Vs Active Sensors." *Medium*, 9 Sept. 2021, medium.com/@almiramadhi/satellite-data-sensor-d0b0d0db7a8c. Accessed 4 Mar. 2025.

[37] Shepard, Donald. "A Two-Dimensional Interpolation Function for Irregularly-Spaced Data." *Proceedings of the 1968 23rd ACM National Conference*, 1968.

[38] Burrough, Peter A., et al. *Principles of Geographical Information Systems*. Oxford UP, 2015.

[39] Mahesh, Batta. "Machine Learning Algorithms-a Review." *International Journal of Science and Research (IJSR)*, vol. 9, no. 1, 2020, pp. 381-86.

[40] Sarker, Iqbal H. "Machine Learning: Algorithms, Real-World Applications and Research Directions." *SN Computer Science*, vol. 2, no. 3, 2021, p. 160.

[41] The 365 Team. "Introduction to Decision Trees: Why Should You Use Them?" *365 Data Science*, 15 May 2024, 365datascience.com/tutorials/machine-learning-tutorials/decision-trees/. Accessed 5 Mar. 2025.

[42] Song, Yan-Yan, and L. U. Ying. "Decision Tree Methods: Applications for Classification and Prediction." *Shanghai Archives of Psychiatry*, vol. 27, no. 2, 2015, p. 130.

[43] Sagi, Omer, and Lior Rokach. "Ensemble Learning: A Survey." *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, vol. 8, no. 4, 2018, p. e1249.

[44] Bhat, Harshini. "An Introduction to Ensemble Learning Techniques: Explained." *AlmaBetter Blog*, 1 Mar. 2024, en.wikipedia.org/wiki/Image_%28mathematics%29. Accessed 6 Mar. 2025.

[45] Breiman, L. "Bagging Predictors." *Machine Learning*, vol. 24, 1996, pp. 123-40, doi:10.1007/BF00058655.

[46] Che, Dongsheng, et al. "Decision Tree and Ensemble Learning Algorithms with Their Applications in Bioinformatics." *Software Tools and Algorithms for Biological Systems*, 2011, pp. 191-99.

[47] Zhang, H.W., et al. "Using Machine Learning to Develop a Stacking Ensemble Learning Model for the CT Radiomics Classification of Brain Metastases." *Scientific Reports*, vol. 14, 2024, doi:10.1038/s41598-024-80210-x.

[48] Schapire, R.E. "Theoretical Views of Boosting." *Computational Learning Theory*, edited by P. Fischer and H.U. Simon, Springer, 1999, pp. 1-10, doi:10.1007/3-540-49097-3_1.

[49] Ferreira, A.J., and M.A.T. Figueiredo. "Boosting Algorithms: A Review of Methods, Theory, and Applications." *Ensemble Machine Learning*, edited by Cha Zhang and Yunqian Ma, Springer, 2012, pp. 35-86, doi:10.1007/978-1-4419-9326-7_2.

[50] Friedman, Jerome, et al. "Additive Logistic Regression: A Statistical View of Boosting (with Discussion and a Rejoinder by the Authors)." *The Annals of Statistics*, vol. 28, no. 2, 2000, pp. 337-407.

[51] Natekin, Alexey, and Alois Knoll. "Gradient Boosting Machines, a Tutorial." *Frontiers in Neurorobotics*, vol. 7, 2013, p. 21.

[52] Chen, Tianqi, and Carlos Guestrin. "XGBoost: A Scalable Tree Boosting System." *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2016.

[53] Akiba, Takuya, et al. "Optuna: A Next-Generation Hyperparameter Optimization Framework." *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, 2019.

[54] Berrar, Daniel. "Cross-Validation." *arXiv preprint arXiv:1811.12808*, 2019, pp. 542-545.

[55] Zhong, Yi, et al. "Nested and Repeated Cross Validation for Classification Model with High-Dimensional Data." *Revista Colombiana de Estadistica*, vol. 43, no. 1, 2020, p. 103.

[56] Pelletier, H. "How-To: Cross Validation with Time Series Data." *Towards Data Science*, 29 Dec. 2023, towardsdatascience.com/how-to-cross-validation-with-time-series-data-49fdbb5e87ae.

[57] Rudin, Cynthia. "Stop Explaining Black Box Machine Learning Models for High Stakes Decisions and Use Interpretable Models Instead." *Nature Machine Intelligence*, vol. 1, no. 5, 2019, pp. 206-15.

[58] Lundberg, Scott M., and Su-In Lee. "A Unified Approach to Interpreting Model Predictions." *Advances in Neural Information Processing Systems* 30, 2017.

[59] Lundberg, Scott M., et al. "Consistent Individualized Feature Attribution for Tree Ensembles." *arXiv preprint arXiv:1802.03888*, 2018.

[60] U.S. Census Bureau. *Cartographic Boundary Files - Shapefile*. U.S. Department of Commerce, www.census.gov/geographies/mapping-files/time-series/geo/carto-boundary-file.html. Accessed 31 Mar. 2025.

[61] United States Department of Agriculture, National Agricultural Statistics Service. *Quick Stats*, quickstats.nass.usda.gov/. Accessed 31 Mar. 2025.

[62] Halvorson, J. "Census vs. Survey: What's the Difference?" *USDA Blog*, 1 Nov. 2022, www.usda.gov/about-usda/blog/2022/11/01/census-vs-survey-whats-difference.

[63] United States Department of Agriculture, Foreign Agricultural Service. "Crop Calendars for United States." *IPAD - International Production Assessment Division*, ipad.fas.usda.gov/rssiws/al/crop_calendar/us.aspx. Accessed 27 Mar. 2025.

[64] NOAA National Centers for Environmental Information. "Climate at a Glance: County Mapping." *NOAA*, Mar. 2025, www.ncei.noaa.gov/access/monitoring/climate-at-a-glance/county/mapping. Accessed 31 Mar. 2025.

[65] NASA CERES Team. "Energy Balanced and Filled (EBAF)." *NASA Langley Research Center*, ceres.larc.nasa.gov/data/. Accessed 2 Apr. 2025.

[66] Butler, Kevin. "Including netCDF Dimension Values in the Name of an Output Layer or Table." *ArcGIS Blog*, 27 Apr. 2012, www.esri.com/arcgis-blog/products/analytics/product-analytics/including-netcdf-dimension-values-in-the-name-of-an-output-layer-or-table/.

[67] NASA Goddard Earth Sciences Data and Information Services Center (GES DISC). (n.d.). OCO-2 Level 2 bias-corrected XCO2 and other select fields from the full-physics retrieval aggregated as daily files, Retrospective processing V11.2r (OCO2_L2_Lite_FP). NASA. Retrieved April 9, 2025, from https://disc.gsfc.nasa.gov/datasets/OCO2_L2_Lite_FP_11r/summary

[68] NASA Goddard Earth Sciences Data and Information Services Center (GES DISC). "OCO-2 Level 2 Bias-Corrected XCO2... V11.2r." *NASA*, disc.gsfc.nasa.gov/datasets/OCO2_L2_Lite_FP_11r/summary. Accessed 9 Apr. 2025.

[69] Rahm, E., and H. H. Do. "Data Cleaning: Problems and Current Approaches." *IEEE Data Engineering Bulletin*, vol. 23, no. 4, 2000, pp. 3-13.

[70] Leng, G., et al. "The Role of Climate Covariability on Crop Yields in the Conterminous United States." *Scientific Reports*, vol. 6, 2016, doi:10.1038/srep33160.

[71] Vennam, R. R. *Impact of Soil Moisture Stress at Different Phases of Corn Growth and Development*. 2023. Mississippi State U, Master's thesis. *Scholars Junction*, scholarsjunction.msstate.edu/td/5975.

[72] Stella, Tommaso, et al. "Wheat Crop Traits Conferring High Yield Potential May Also Improve Yield Stability Under Climate Change." *in silico Plants*, vol. 5, no. 2, 2023, doi:10.1093/insilicoplants/diad013.

[73] Yang, Y., et al. "Responses of Spring Wheat Growth to Climate Change in Different Climatic Regions of Northwest China." *Crop Science*, vol. 63, no. 2, 2023, pp. 899-911, doi:10.1002/csc2.20863.

[74] Kaium, M. A., et al. "Modeling Impacts of Climate-Induced Yield Variability and Adaptations on Wheat and Maize in a Sub-Tropical Monsoon Climate - Using Fuzzy Logic." *Scientific Reports*, vol. 15, no. 1, 16 July 2025, doi:10.1038/s41598-025-09820-3.