ABSTRACT

MCDOWELL, SHANA MONIQUE. Machine Learning Applications in Sweetpotato Production: A Comprehensive Approach to Yield Improvement. (Under the direction of Dr. Daniela Jones).

This work focuses on the development of machine learning models and pipelines designed to predict sweet potato growth and quality characteristics, while also identifying important features for in-season crop development. While prior studies have primarily examined sweet potato shape and size post-harvest, this research addresses the challenge of assessing underground crop growth during the growing season. This study involves developing both global and granular predictive models as well as providing guidelines for working with agricultural data and the establishment of flexible data and machine learning pipelines to expedite the model-building process. In Chapter 2, we developed machine learning (ML) models to identify key temporal and stationary environmental factors, including soil characteristics, weather variables, and field management practices, influencing sweetpotato shape and size during the growing season. These attributes were expressed as average weight and length-width ratio (LWR). Using data from a large sweetpotato grower in North Carolina managing over 5,000 acres, we collected information across multiple on-farm locations over three growing seasons. Sweetpotato root images were analyzed using computer vision algorithms to improve our understanding of developmental stages. Key predictors included days after planting (DAP), average temperature, minimum relative humidity, and specific management practices such as liquid fertilizer application timing. For weight prediction, slope and soil characteristics were also influential, while factors like wind speed and cation exchange capacity were critical for LWR. Random Forest models achieved the best predictive performance, with RMSE values of 5.57 oz (MAE=3.16) for weight (random split) and 0.62 (MAE=0.45) for LWR (field split). Nested K-fold cross-validation ensured robust model evaluation. Results confirmed the significant impact of both environmental and agronomic factors on sweetpotato traits, providing actionable insights for growers. Additionally, we presented a use-case scenario to demonstrate the practical application of these models. By leveraging ML predictions, farmers can implement targeted in-season management strategies to optimize harvest timing and sweetpotato quality, making the results both interpretable and actionable. This approach highlights the potential of ML in precision agriculture to enhance decision-making, improve crop profitability, and reduce post-harvest losses.

In Chapter 3, we present a method that includes a data and machine learning pipeline used to predict crop yield by combining in-season, agronomic, and environmental data, and incorporating key machine learning steps such as feature selection and hyperparameter tuning. The method was validated by comparing the predictive performance of four machine learning models using sweetpotato data collected from 13 counties in North Carolina. The results demonstrated that the methodology could predict sweetpotato attributes in-season and highlight important features for crop growth.

- The method requires only a unique location identifier and latitude and longitude coordinates as input.
- The machine learning pipeline streamlines key aspects of the machine learning workflow.

In Chapter 4, we build on the work from Chapter 2 by restructuring the data for use with Long-Short Term Memory recurrent neural networks to capture the temporal dependencies between photos captured at each location. Static models, random forest, linear regression, and extreme gradient boosting models were used for comparison. Results show that static models outperformed the dynamic model. When predicting average weight, the random forest model (RMSE:3.975, MAE:2.909, R^2:0.4683) using the random split achieved the highest performance, while linear regression (RMSE:0.4413, MAE:0.3465, R^2:0.4118) using the random split performed the best for predicting average length-width ratio. Important features for predicting average weight are recent development of average weight (avg_weight_oz_t2, avg_weight_oz1$\widehat{\ }$) and days after planting (DAP_t2, DAP_t1), and minimum relative humidity at the most recent timestep (min_rh_t2). Important features for predicting average length-width ratio are days after planting (DAP_t2, DAP_t1) from previous time steps, average temperature (avg_temp_t2) at the most recent timestep, average length-width ratio (avg_lwratio_t2) from the most recent timestep, and average wind speed (avg_ws_t1) from the first timestep.

Machine Learning Applications in Sweetpotato Production: A Comprehensive Approach to Yield Improvement

by
Shana Monique McDowell

A dissertation submitted to the Graduate Faculty of
North Carolina State University
in partial fulfillment of the
requirements for the degree of
Doctor of Philosophy

Biological Agricultural Engineering

Raleigh, North Carolina
2025

APPROVED BY:

Dr. Daniela Jones Chair of Advisory Committee	Dr. Michael Kudenov
Dr. Cranos Williams	Dr. Anderson de Quieroz

DEDICATION

Dedicated to my loving father, Charles McDowell, who supported me through every challenge of the PhD process and throughout my entire graduate studies. I am deeply grateful for all the college and workshop visits you drove me to before my PhD and for attending every graduation ceremony I have had. You have set the standard for success and encouraged me to aim higher each time.

BIOGRAPHY

Shana McDowell was born on March 16, 1988, in Charlotte, North Carolina. She received her Bachelor of Science in Pure Mathematics from Fayetteville State University in Fayetteville, NC, USA in December 2011. Later, she obtained her Master of Science degree in Mathematics with a concentration in Applied Mathematics from North Carolina Central University in Durham, NC, USA, in May 2019. In Fall 2021, she joined North Carolina State University, Raleigh, NC, USA, where she is a candidate for the Doctor of Philosophy degree in Biological and Agricultural Engineering. Her research interests include sweetpotato shape prediction, agricultural data analysis, and machine learning.

ACKNOWLEDGEMENTS

I would like to express my gratitude to my advisor, Dr. Daniela Jones, for being an exceptional mentor throughout my journey. Your guidance and support have been invaluable as I navigated this challenging process. You have helped me develop into a critical thinker and instilled confidence in my ability to become an independent researcher. I truly appreciate your honest and approachable mentoring style, along with your empathy towards the difficulties that can arise during this experience. Witnessing your growth as a professor while I have also grown has genuinely been inspirational. From you, I have gained confidence, resilience, and perseverance throughout my PhD journey.

I would also like to thank my committee members, Dr. Cranos Williams, Dr. Michael Kudenov, and Dr. Anderson de Queiroz, for their guidance and insight on my research studies. In addition, I would like to extend my gratitude to Scott Carpenter for all of his insight into sweetpotato growth and Scott Farms Inc. for their help and advice while collecting data. I would like to thank my fellow SWEET-APPS cohort, Dr. Enrique Pena Martinez, and Dr.Randi Butler for starting this process with me. WE MADE IT!! I would like to thank my friends and colleagues at the IDEALS lab, specifically Dr.Juliana Pin, Shelly Hunt, Dr.Tasmin Hossain, Leticia Santos, Annelise Intemann, and Laura Mora, for their encouragement and motivation and for making the lab so inviting.

Finally, I would like to thank my support system of friends, family, and mentors: my tribe. I am forever grateful for your support, prayers, and encouragement.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER

1

INTRODUCTION

1.1 Motivation

1.1.1 Sweetpotato Growth and Quality

Farmers and consumers of fresh produce often discard large amounts solely based on the aesthetics of marketability. Higher value is placed on fresh produce that adheres to an ideal shape and size (2; 3; 4; 5), while misshapen fruits and vegetables are generally rejected or sold at a reduced price even when suitable for consumption (6; 7). Harvest and production planning are essential for obtaining the desired physical attributes (8). Tomatoes, cucumbers, and sweetpotatoes are often discarded during harvest and packing due to an undesired shape and size. Johnson et al. (9) defined food loss as the unintentional result of agricultural processes with respect to storage infrastructure, technical limitations and packaging, and food waste as an effect of individual behavior that typically occurs within the retail, restaurant, and household levels of the supply chain. Approximately 20% of fruits, vegetables, roots, and tuberous crops grown on farms are lost, discarded, or culled during harvest (10). Because food loss and food waste negatively impact profit, a better understanding of the factors contributing to a crop's growth and physical attributes is

crucial. North Carolina (NC) is the leading sweetpotato producing state in the United States, contributing almost 50% of the nation's supply. Sweetpotatoes vary widely in shape and size and are classified according to USDA standards (11) and market requirements. Different markets require different shape and size specifications. For example, sweetpotatoes classified as "jumbo" are typically preferred in the processing industry for sweetpotato fries (12). Most sweetpotato packing facilities in NC sort the product before packing. Data that helps classify produce during packing, if collected, is not usually stored and hence not analyzed to trace produce and provenance. Sweetpotato shape attributes, such as weight and length-width ratio, are used to sort sweetpotatoes during packing. Thus, predicting these aspects of yield quality earlier in the growing season will equip growers with specific sweetpotato growth recommendations tailored to individual fields and the growing season, helping them plan for future seasons.

1.2 Background

1.2.1 Sweetpotato Shape and Size Prediction

Crop Modeling

Process- and machine-learning-based crop models have been used to better understand crop development. Process-based models, or traditional crop models, are mathematical representations of a system consisting of individual equations that describe various processes (13). These models are used to simulate system behavior and are typically based on scientific understanding. Machine Learning (ML), a subcategory of predictive analytics in which computers are used to analyze and recognize patterns in data to make future predictions, encompasses algorithms such as Random Forests, Gradient Boosting, and Neural Networks. ML models are designed to mimic human behavior and decision-making. Historically, traditional and ML crop models have been used to predict crop yield, quality, and development under stress (14; 15). ML models have gained popularity as process-based models require large amounts of detailed data about a system, as well as thorough research regarding the equations that represent a process. ML models can take many different data sources, can often be quicker to implement than process-based models, and can capture complex trends in complex datasets.

Crop development data can be collected in two ways: destructively or nondestructively. Destructive methods involve physically pulling the crop from the ground, disrupting any

further growth, and irreversibly altering the crop under test to perform a given measurement, while nondestructive methods leverage tools like remote sensing or ground penetrating radars to monitor crop development over time. In-season crop development data are often captured destructively and subsequently modeled using process-based methods. However, invasive techniques for monitoring crop quality are often labor-intensive and lack scalability (16). Many studies have presented nondestructive techniques to monitor crop development and quality during post-harvest operations using imaging and spectroscopy. Crop simulation models such as the Decision Support System for Agrotechnology Transfer (DSSAT) and the Agricultural Production System Simulator (APSIM) have been used to create simulations to assess crop development under a series of conditions. Although a DSSAT potato model exists, SUBSTOR-Potato, one has not been created specifically for sweetpotato development. This allows ML to be used to predict different aspects of sweetpotato development and yield.

Crop Yield and Quality Predictions

Remote sensing technologies have been used to nondestructively assess crop development and optimal harvest time using crop indices(17; 5). Tedesco et al. (17) presented a new nondestructive method to assess the phenology of sweetpotato and predict the yield of sweetpotato using remote sensing technology by establishing a connection between the growing degree days (GDD) and the vegetation indices further defining the development of the sweetpotato crop into three phenological stages, initial (<200 GDD), growth stage (200 - 500 GDD), and stabilization (>500 GDD). Carbajal-Carrasco et al. (5) used multitemporal remote sensing data to develop ML models to predict sweetpotato production in season at the county level and found that model performance was highest (RMSE = 3.53 $t.ha^{-}1$, R^2 = 0.44) when predicting sweetpotato production using a 16-day aggregation scheme and temporal data from the early-mid season.

Villordon et al. (18) presented a study of sweetpotato development using growing degree days to determine the optimal harvest time. The authors analyzed multiple GDD calculations and concluded that a GDD of 2600 is optimal to begin testing harvests in Louisiana. A later study by this same research team presented the development of a phenology-driven Bayesian belief network model (BxNET) to analyze the relationship between fresh market yield and agroclimatic variables influencing the initiation of root storage in sweetpotato cultivation, providing information for decision support systems and improved crop management practices (19). Pena Martinez et al. (4) optimized sweetpotato post-harvest opera-

tions by using two high-throughput phenotyping devices to quantify the distribution of sweetpotatoes during the post-harvest process. The results show that this technology can enhance packing efficiency and logistics and reduce post-harvest waste.

Common yield quality attributes include crop shape; length, width, curvature, sugar content, and flavor profile are often used to predict crop development. Direct measurement and digital image analysis are commonly used to predict the shape attributes of sweetpotatoes (20; 21; 6; 22; 2; 1; 3; 23). The yield quality attributes studied have been weight, specific gravity, surface area, volume, perimeter, curvature, roundness, elongation, major axis length, minor axis length, and length-width ratio. Most image analysis methods have analyzed shape features post-harvest rather than during the growing season. Wright et al.(20) studied and captured the physical attributes of sweetpotatoes post-harvest using image analysis techniques to measure the volume and surface area of 457 U.S. grade No.1 sweetpotatoes for several commercial varieties. Villordon et al.(21) evaluated the method of digital image analysis to assess the size and shape characteristics of sweetpotatoes during herbicide trials. The authors found that the length, width, and roundness of the storage root best described the variation in the sweetpotatoes using the principal component analysis. In a later study, Villordon et al. (22) presented a method to measure the surface area and volume of sweetpotatoes using a low-cost 3D scanner. Similarly, Boyette et al.(6), measured individual sweetpotato roots using a high-speed laser scanner to obtain shape characteristics. Haque et al. (2) used ML to classify sweetpotatoes based on their shape and size using images from a commercial optical sorter (24) located in a research station in NC. The author aimed to classify sweetpotatoes into Cull or USDA No. 1 grades and to further classify Cull-graded sweetpotatoes into curved, round, tailed, tapered, and other classes. Nguyen et al.(1) presented an instance segmentation algorithm, specifically a mask RCNN (Region-Based Convolutional Neural Network) algorithm trained using transfer learning, to classify individual sweetpotato regions in images and output shape metric estimates. Liu et al. (3) used ML to predict shape traits, curvature, and length-width ratio for various sweetpotato cultivars, using environmental and field management data from three research stations over four growing seasons (2018-2021). Gogineni et al. (23) developed an image-based system for sweetpotato yield and grade monitoring, employing multiple linear regression and neural networks to estimate sweetpotato weight. Key features included pixel area, polar moment of inertia, rectangular width and height, and major and minor axis lengths.

Although each of the above studies sheds light on the different aspects of the development of sweetpotatoes, currently, there are no other research studies exploring how data

splitting paired with ML can potentially highlight important agronomic and field management factors to determine the shape and size of sweetpotato earlier in the growing season at the field level. In this paper, we developed a predictive modeling framework that combines complex datasets such as environmental and field management in-season data to predict the average weight and the length-width ratio of sweetpotatoes during the growing season. Based on previous research, (18; 5; 3) we posit that this prediction will guide stakeholders in making data-informed in-season management decisions, optimizing harvest dates to produce the most favorable sweetpotato shapes and sizes for the market, thus increasing profit margins. We present our results in inches and ounces, as these are the units used in USDA standards (11) and the ones preferred by stakeholders in our industry. As expected, days after planting were found to be the most important in predicting the average weight and length-width ratio. In addition, the average temperature, days after liquid fertilizer application, minimum relative humidity, and slope were important to predict the average weight. Similarly, minimum relative humidity, average wind speed, average temperature, and cation exchange capacity are important to predict the average LWR. Random forest using random splitting (RMSE = 5.57 oz, MAE = 3.16) outperformed all other models in predicting average weight; random forest using field splitting (RMSE = 0.62 oz, MAE = 0.45) outperformed all other models in predicting average LWR.

1.2.2 Pipeline Development to support Sweetpotato Growth and Quality Predictions

Predicting crop quality early in the growing season supports timely and informed decision-making. Evaluating a crop's shape, size, and sugar content are common characteristics for estimating yield and quality (15). Because sweetpotatoes often grow irregularly, and their market value is closely tied to appearance, early predictions of shape and size are particularly valuable. Additionally, understanding when specific shapes and sizes will be available can help align harvest and marketing strategies with consumer preferences across different markets, ultimately maximizing profit margins. Researchers often use a combination of field management, weather, soil, and remote sensing data to predict crop yield or crop quality using machine learning (5; 3). While these datasets are accessible, they often require a diverse set of tools for data extraction and cleaning. Heterogeneous data integration is a common challenge in agricultural research, as it is crucial to ensure that observation points – whether structured or unstructured, or extracted from diverse sensors as tabular data or images- are properly aligned in both time and space. Depending

on the research question, agriculture datasets could be split by location, growing season, or in a sequential way to ensure a prediction model performs well on unseen data. Lastly, data volume presents another challenge, as collecting large amounts of agricultural data is often labor-intensive, and climate events can disrupt data collection. Therefore, developing methods to efficiently handle limited data is highly valuable. Many methodologies exist for predicting crop yield using agricultural datasets, but identifying which combination of methodologies best suits a particular problem and implementing can be labor-intensive. As the scale of agricultural research projects and the collected data grows, it will be valuable to have an automated but flexible framework that receives yield data or estimates, combines the environmental data obtained, and processes these data using machine learning models. This work presents a detailed data pipeline used to collect, compile, and prepare data for a machine learning workflow that robustly trains and tests machine learning models and compares the model performance on yield prediction. We provide code that outlines a streamlined methodology that could be used alone or combined with other pipelines and/or dashboards to accelerate the decision-making of growers and stakeholders by providing agronomic progression information on the crop earlier during the growing season. This method builds upon our previously presented pipeline (25) by combining both pipelines into one fully connected pipeline, improving accessibility and transforming this pipeline tool into a no-code experience to reduce the technical barrier.

1.2.3 Predicting Sweetpotato Growth and Quality Characteristics using Spatiotemporal Applications and Machine Learning

Numerous factors contribute to or control the development and final appearance of a crop at harvest (26). This final appearance can affect profit margins as consumers generally weigh whether to purchase of the appearance of the crop. For sweetpotato specifically, research has been conducted to classify shape (27), estimate grade (1), predict shape and the environmental and agronomic factors that influence shape, such as curvature, length-width ratio, volume, surface area, perimeter, length, and width (3).

Various methodologies, such as machine learning, deep learning, and computer vision, have been employed to analyze or predict sweetpotato shape considering different agronomic and abiotic factors. However, to date, none of these studies have utilized Long Short-Term Memory (LSTM) models for shape prediction or accounted for spatial dependence. Data sources typically used in these analyses include field management, weather, soil, topography, and remote sensing data. While including explanatory predictors has

proved successful for prediction, it is also crucial to consider the spatial relationship between data observations.

When predicting crop yield or quality attributes using collected data from multiple field locations, it can be assumed that field locations near one another will have similar soil, weather, or topography characteristics due to spatial dependence. Spatial autocorrelation (SA) is a method used to determine if observed patterns of a variable, such as yield, are spatially clustered, dispersed, or random. Precision agriculture is heavily dependent upon the ability to address spatial variation in crop yield or quality and the environmental factors that influence them (28; 29). Many machine learning models rely on the assumption of independence between observations, so if SA is present, this can cause some machine learning models to overfit (30). SA auto correlation has been used to identify management zones or areas that may need specialized attention to enhance yield, for corn and cotton yields. (28; 29).

To effectively address spatial dependence, clustering-based methods can be used to create groups of data with similar characteristics. K-Means clustering, a common unsupervised learning technique, generates groups or clusters based on the average position of the points within a group (31). The number of clusters is chosen before implementation. One limitation of this approach is that it is advisable to run the model multiple times to identify the best clustering results. K-means is considered a "greedy algorithm" because it relies solely on the information currently available when forming clusters, without taking into account how the selected centroids may impact future iterations.

1.3 Dissertation Outline

The primary goals of this research project are to (1) identify important agronomic and environmental factors that influence sweetpotato development, (2) develop data and machine learning pipelines that integrate information from various heterogeneous datasets and produce robust predictive models for use by researchers, extension agents, and growers, and (3) identify important agronomic and environmental factors that influence sweetpotato development by refining the methodology used in goal 1 to incorporate spatial and temporal information to make short-term predictions.

CHAPTER

2

MACHINE LEARNING FOR PREDICTING SWEETPOTATO GROWTH AND QUALITY: INTEGRATING IN-SEASON ROOT IMAGING WITH ENVIRONMENTAL AND AGRONOMIC INSIGHTS

2.1 Introduction

For many crops, consumer preferences and marketability are driven by the outward appearance of the crop. Sweet potatoes, in particular, develop in all shapes and sizes, which potentially affects the market value of the crop. In this Chapter, we develop machine learning models to predict sweet potato shape attributes that are used during sorting and packing and directly impact market value. We extract and combine various heterogeneous data sources into a larger dataset that is used to predict sweetpotato average weight and

length-width ratio, as well as identify influential factors that contribute to sweetpotato development.

2.2 Materials and Methods

2.2.1 Data Collection

On-Farm Samples

Covington sweetpotato storage roots, collected destructively, were imaged from three plants every 7-10 days over 57 field years over three growing seasons (21, 18, and 18 fields in years 2021, 2022, and 2023, respectively). On-farm samples were collected from fields in the following counties in North Carolina: Duplin, Durham, Edgecombe, Greene, Johnston, Jones, Lenoir, Nash, Pitt, Sampson, Wake, Wayne, and Wilson. A total of 581 images or samples were collected that illustrated 6,604 sweetpotato storage roots. Immediately after harvest, all of the below-ground storage roots were placed on a black background and imaged along a 4.72 cm (1.86 inch) blue tape for spatial/pixel calibration. Images were taken using a Samsung Galaxy S8 cell phone. To streamline image collection and processing, the cell phone was placed on a tripod stand at a known fixed distance above a stationary black background (1.19 cm or 47 inches high). We included a ring light and a polarized film over the ring light and the camera lens to reduce glare and reflections when imaging samples. Figure 2.1 illustrates the setup for collecting these images. These images were used to measure the area, width, and length of the sweetpotatoes in each sample. Sweetpotatoes' area was used to calculate weight, width, and length, all retrieved from the RCNN algorithm's output. The predicted variables for the ML models were average weight (WT) and average length-width ratio (LWR).

Figure 2.1: Tripod set up to capture sweetpotato root images

Field Management Features

Through the John Deere Operations Center database (32) of our industry partners, we collected the *xy coordinates* and the dates for each of the following field management operations carried out in each field studied: planting, harvest, application of liquid and dry fertilizers, the rate of application of liquid and dry fertilizers, and lay-by cultivation- the last point during the growing season of cultivation, when a grower applies herbicides and completes any in-row maintenance. In total, we included 6 field management features in our analysis.

Environmental Features

The environmental features included temporal and stationary information. Temporal features included daily weather data for the entire growing season of each field studied (days between the planting and harvest dates). Using these growing windows and each field's *xy-coordinates*, we obtained temporal features from NASA's POWER agroclimatology model

(33). These features included: total precipitation (total_precip, mm), average (avg), minimum (min) and maximum (max) solar radiation (rad, W/m^2), temperature (temp, °F), wind speed (ws, m/s), and relative humidity (rh, %). In total, we included 13 temporal environmental features in our analysis.

Stationary environmental features included soil characteristics obtained from ISRIC World Soil Information SoilGrids global gridded soil information using Google Earth engine (34). These characteristics included bulk density (bd, g/cm^3), cation exchange capacity (cec, meq/100g), clay content (%), silt content (%), sand content (%), nitrogen (n, g/kg), pH, and soil organic carbon (soc, g/kg). All soil variables are from depths of 0-5 and 5-15 cm at a resolution of 250 m. Other stationary environmental features, including soil texture (such as sand, loamy sand, sandy loam), and soil map unit symbols- used to identify soil types on a map (such as NoA, Po, WaB), were obtained from the Natural Resources Conservation Service (35). Lastly, topography features such as elevation (m), aspect (°), and slope (°) were added as stationary environmental features obtained using digital elevation models (DEM) from the Shuttle Radar Topographic Mission (SRTM), originally produced by NASA but later improved by the Consortium for Spatial Information (CGIAR CSI) (36). In total, we included 21 stationary environmental features in our analysis.

2.2.2 Data Processing

Image segmentation

The 2D images collected were input (Figure 2.2) to Mask RCNN used in this study and used to obtain the weight, length, and width of the sweetpotatoes. RCNN was based on Facebook's instance-based masking neural network (37), trained and annotated for sweetpotatoes as regions of interest (ROI) by Nguyen et al. (1). The output of RCNN included an image with generated masked ROIs and a CSV file that contained the following shape attributes extracted from the generated masks, area, length, and width in px^2 for each ROI or sweetpotato in an image. For example, the image input in Figure 2.4 was taken 83 days after planting and the output tabular file (.csv) included 8 observations for each identified ROI or identified sweetpotato region in the image.

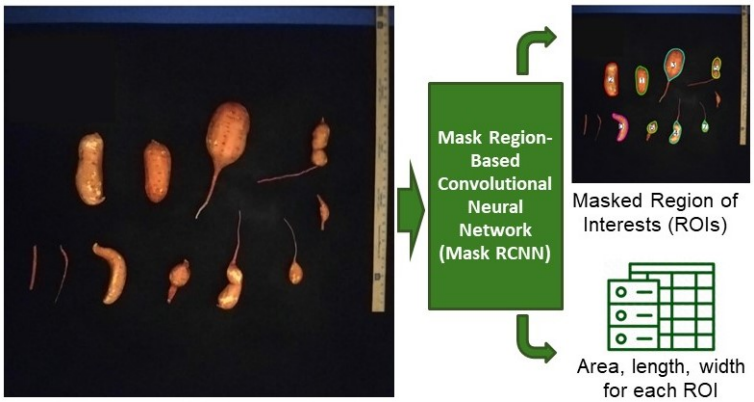

Figure 2.2: Process flow of inputs and outputs of the mask region-based convolutional neural network used in this study (1)

Area, length, and width

The ROI's area, length, and width in the RCNN mask output were translated to inches (in.) and $in.^2$ using the blue tape present in every image for calibration. We used equation 2.1 to convert the ROI's area from px^2 to in^2. Where, F is the adjustment factor, s is the width of the blue tape in $in.$, and t is the width of the blue tape in px. Based on the distance of our camera lens to the sweetpotatoes display platform, a fudge factor of 1.3 optimized our measurements.

$$A_{ROI_{in}} = F * (\frac{s}{t})^2 * A_{ROI_{px}} \tag{2.1}$$

Weight

Similarly to the non-linear model presented by Pena Martinez et al.(4), we used Equation 2.2 to calculate the predicted weight of each ROI in our images. A comparison of the observed weights (obtained with a scale) with the predicted sweetpotato weights (using the mask-RCNN and Equation 2.2) was plotted in Figure 2.3.

$$W_{RoI_{oz}} = 10^{(1.465 * log10(A_{ROI_{in}}) + 0.8749)} * 0.03527396 \tag{2.2}$$

Figure 2.3: Predicted vs. observed weight(oz) of sweetpotatoes (n=339)

Feature Engineering

To include information on sampling time, we calculated days after planting (DAP) by counting the number of days between the planting date and the sampling in the season or the imaged date. Similarly, field operation dates were translated into DAP to compare agricultural management data among the different fields studied. For example, if field A was planted on May 19 and liquid fertilizer (abbreviated as DAP_lf_app) was applied on June 28 in that field, then the days after planting for that operation (DAP_lf_app) would be 40 days.

All temporal environmental variables were engineered on the basis of each field's planting and sampling or imaged date. Furthermore, accumulated growing degree days were calculated using the same time period, a base temperature (B) of 16.9°C and a ceiling temperature (C) of 29.2°C (18), by

$$GDD = \begin{cases} 0 & \text{if Tmin} < \text{B} \\ \text{C-B} & \text{if Tmin} \geq B \text{ and Tmax} > C \\ \text{Tmax-B} & \text{if Tmin} \geq B \text{ and Tmax} \leq C. \end{cases} \tag{2.3}$$

Data Compilation

Weight and LWR, the response variables in our yield prediction models, were merged with their respective environmental and field management features. In total, 40 features (36 numerical and 4 categorical variables) could potentially explain the weights of the

sweetpotatoes and / or LWR. Two datasets (D) were compiled for analysis using the entire feature set. D1, where each observation represented information for a single sweetpotato in a sample image, a total of 6,647 observations, and D2, where each observation represented the average information of all sweetpotatoes in a sample image, 577 observations total. Figure 2.4 show the values for both targets plotted over days after planting using both datasets.

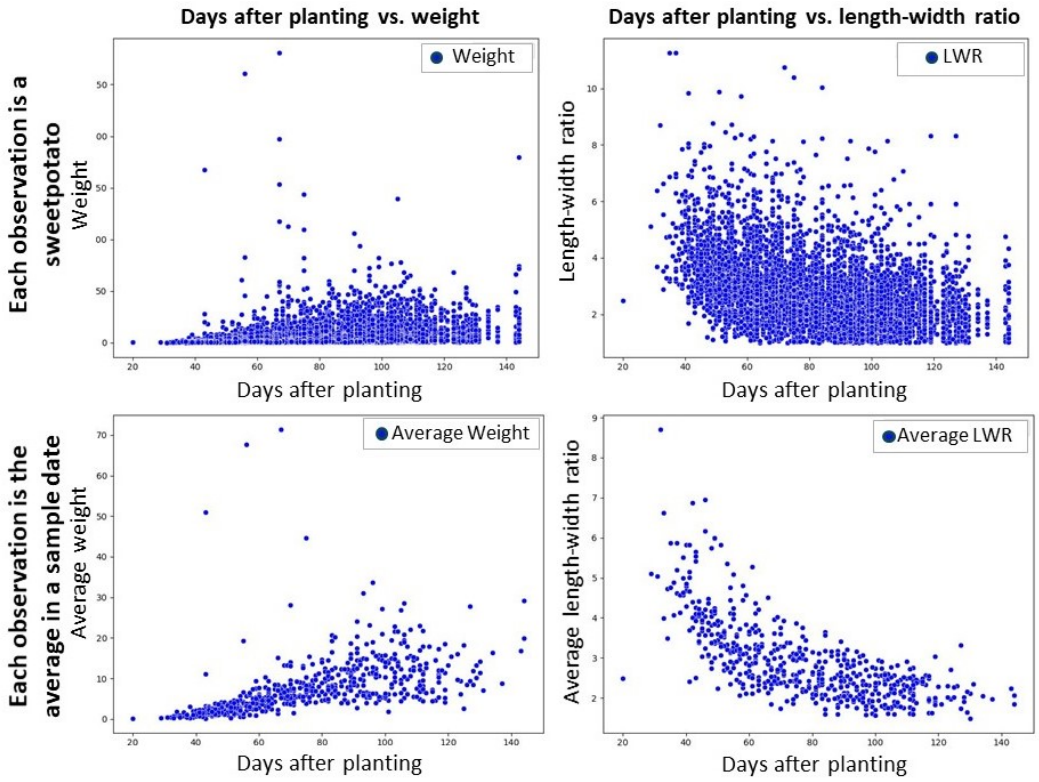

Figure 2.4: Scatter plot of days after planting (DAP) vs. weight and length-width (LWR) targets using: 1) D1 dataset, where each observation is a single sweetpotato (n= 6,647) and 2) D2 dataset, where each observation is the average of sweetpotato features in the same image (n= 577)

Data Transformation

Explanatory features were standardized using the MinMax scaler in Python prior to the implementation of the model. Categorical variables were transformed into dummy variables using dummy encoding (38). Dummy encoding is similar to one-hot encoding as it creates

new binary columns dedicated to each value of a category, except dummy encoding creates n-1 binary columns to avoid the issue of multicollinearity that arises with one-hot encoding. These new variables were labeled using the category variable name and the categorical value. For example, a texture that corresponds to soil texture would be texture_sandy_loam. The 4 categorical features in our dataset were translated into 31 dummy variables.

Data Splitting

Three cross-validation schemes were explored for splitting data into training and testing sets: random split, split by field location, and split by growing season. The latter two schemes were particularly evaluated to address field- and season-specific questions in agricultural data. Splitting by field and growing season enables the prediction of sweetpotato characteristics in unseen fields or future seasons, providing valuable insights for growers' short- and long-term decision-making.

2.2.3 Model Building

We implemented the most commonly used ML techniques for crop yield prediction as described by Klompenburg et al. (15): linear regression (LR), random forest (RF), support vector machine (SVM), and gradient increase (GB). For each model, we calculated the mean absolute error (MAE), root mean squared error (RMSE), and coefficient of determination (R^2) as well as their standard deviations to assess performance. When determining the best model, the focus was primarily RMSE and MAE for model comparison, using R^2 as a supplementary indicator due to its limitations in evaluating the goodness of fit in nonlinear models (39). The LR model is considered our baseline model for this research.

Feature Selection

Feature selection was used to avoid multicollinearity, reduce overfitting, enhance generalization, and improve interpretability. Specifically, a hybrid feature selection technique presented by Madhuri et al.(2022) (40) that includes correlation-based feature selection (CBFS) and recursive feature elimination (RFE) was used. Madhuri et al.(2022)(40) found that the performance of crop recommendation models increased when using the combination of a filter and wrapper feature selection technique. CBFS ensures that the chosen features are not highly correlated with each other but are highly correlated with the target. CBFS has two threshold parameters: a correlation threshold for the features associated

with the predicted variable and a redundancy threshold for the features that are highly correlated with each other. For this work, the correlation threshold was set to 0, and the redundancy threshold was set to values of 0.60 and 0.70 for comparison. RFE is considered a wrapper method for feature selection that begins with the full data set and works backward to eliminate the least important features with respect to model performance, leaving only features that are important for prediction (41). The term wrapper comes from the iterative process, where an ML model is used to process subsets of data until an optimal subset is reached. RFE was performed using a random forest regressor as its base estimator. RFE requires that the number of features selected be set before implementation. To determine the optimal number of features to select, we ran all models using 10, 15, and 20 features as parameter values and compared the model performances.

Hyper-parameter Tuning

Hyperparameter tuning for all models was conducted using grid search with 5-fold cross-validation on the training sets. The selected optimal hyperparameters are provided in the supplementary tables (A.1, **??**, A.3, A.4, A.5, A.6). After training and tuning, a final model was developed using the entire dataset and the best-performing algorithm, with permutation feature importance reported.

Nested K-Fold Cross Validation

Cross-validation is a data sampling technique used in ML to evaluate a model's ability to generalize to unseen data. In K-fold cross-validation, the dataset is divided into K distinct folds, with one fold used for validation and the remaining for training. This process is repeated K times, and performance metrics are calculated for each iteration. The model's average performance is then derived from these metrics. By utilizing multiple data combinations, cross-validation produces robust models and provides reliable estimates of predictive generalization performance.

In this chapter, a nested K-fold approach (with $k = 5$) was employed to assess model performance and facilitate model selection. This technique incorporates both an inner and outer K-fold cross-validation loop, where the outer loop evaluates model performance, and the inner loop handles model and feature selection. The outer loop operates like standard K-fold cross-validation, splitting the dataset into training and testing folds. Within each training fold, the data were further partitioned into training and validation sets using the inner loop. Hyperparameter tuning and feature selection were conducted in the inner loop,

minimizing the risk of overfitting and data leakage. This approach provides an unbiased estimate of model performance \ (42). For random and field splits, stratified k-fold cross-validation was used, ensuring balanced data according to data_year, and was used with $K = 5$. Ordinary k-fold cross-validation with $K = 3$ was used for the season split. The outer number of splits and the inner number of splits are equal for each data split. The pseudo-code in 1 illustrates the workflow for this analysis.

Algorithm 1 Nested K-Fold Cross-Validation Pseudo code

Require: Dataset X, Target y, Outer Folds K_{outer}, Inner Folds K_{inner}, Pipeline $pipeline$, Hyperparameter Grid $param\_grid$
Ensure: Predictive Model Performance Estimation
Initialize empty lists for performance metrics (RMSE, MAE, R^2)
Initialize $best\_global\_model$ and $best\_global\_rmse$
Define outer cross-validation $outer\_cv \leftarrow$ StratifiedKFold(K_{outer})
for each fold $i = 1$ to K_{outer} **do**
    Split data into training and test sets: $X_{\text{train}}, X_{\text{test}}, y_{\text{train}}, y_{\text{test}}$
    Define inner cross-validation $inner\_cv \leftarrow$ KFold(K_{inner})
    Initialize pipeline:
        1. Data scaling (MinMax Scaler)
        2. Feature selection (CBFS and RFE)
        3. Model training
    Perform hyperparameter tuning and Feature selection using GridSearchCV
    Train best model on the outer training set
    Evaluate best model on outer test set and store metrics (RMSE, MAE, R^2)
    Update global best model if current fold's RMSE is lower
end for
Return $best\_global\_model$ and performance metrics

2.3 Results

2.3.1 Data Compilation

The average metrics in all models for D1 -where an observation is a single sweetpotato ROI, and the predicted variable was weight, were MAE= 4.38, RMSE= 8.15, and R^2= 0.10, and when the predicted variable was LWR, the average metrics were MAE= 0.82, RMSE= 1.12, and R^2= 0.22. Similarly, for D2 -where each observation corresponds to the average of all sweetpotato ROIs in the same image- results for the predicted variable average weight were MAE= 2.52, RMSE= 4.48, and R^2= 0.28 and for average LWR were MAE= 0.415, RMSE=

0.58, and R^2= 0.61. Using the latter dataset resulted in a 42% decrease, 45% decrease, and 171% increase in average model metrics for the predicted variables weight and average weight, as measured by MAE, RMSE, and R^2, respectively. Similar trends were observed for the predicted variables LWR and average LWR, a 49% decrease, 48% decrease, and 179% increase for MAE, RMSE, and R^2, respectively. Since model performance metrics were higher when using dataset D2 compared to D1, we continued our analysis hereafter only using D2, similar to the approach taken by Liu et al. (3). Permutation feature importance and partial dependence plots were used to assess the importance of features for the models. Permutation feature importance measures how the features in a model contribute to the overall performance by randomly shuffling each feature and evaluating the decrease in model performance when permuted. Partial dependence plots are often used to visualize the relationship between a predicted variable and an input feature while marginalizing over all other input features.

2.3.2 Feature Selection

When implementing the CBFS, relevance thresholds of 0.6 and 0.7 were tested. A threshold of 0.6 was stricter than 0.7, as seen in the selection of features such as DAP and GDU, which are commonly included in yield prediction models but are highly correlated. With a threshold of 0.6, DAP emerged as the dominant feature due to its greater impact on the target average weight, whereas a threshold of 0.7 retained both features after selection. The primary results presented in this paper were based on a relevance threshold of 0.6, with results for a threshold of 0.7 provided in the supplementary tables (A.1, A.2).

The optimal number of features selected for both targets was determined to be 10. For predicting average LWR, the best model performance consistently occurred with 10 features across all data splits. In contrast, results for average weight varied by data split, but selecting 10 features yielded higher coefficients of determination (R^2) and comparable RMSE values, indicating that including additional features did not improve model performance. Detailed comparisons for feature selection and a correlation matrix of all features and response variables are provided in supplementary tables (A.1, A.2, A.1). Notably, a slight negative correlation (-0.42) was observed between the two response variables analyzed.

Across all data splits for the target average weight, the ten features chosen by hybrid feature selection were DAP, avg_temp, DAP_lf_app, min_rh, slope, min_rad, aspect, elevation, cec_5_15, and N_0_5. For the target average LWR the ten features chosen by the hybrid feature selection were DAP, min_rh, avg_ws, avg_temp, cec_0_5, elevation bd_5_15,

aspect, pH_5_15, and t1_crop_tobacco. The order of importance of the features varied for each target-split combination.

As expected, DAP has the greatest importance in both predicted variables. With a different order of importance, five of the ten features were common across all data splits for average weight and LWR: DAP, avg_temp, min_rh, elevation, and aspect. A more detailed analysis of how these features relate to the predicted variables is discussed later in the paper.

2.3.3 Model Performance

Performance metrics and permutation importance plots for all models in all splits are presented in Tables 2.1 and 2.2. The overall best-performing model for predicting average weight was RF using the random split, while for predicting average LWR, RF performed best with the field split. Across data splits, RF consistently outperformed other models for both average weight and LWR when the data were split randomly and by field. However, for the season split, GB performed best for average weight, and LR performed best for average LWR.

Table 2.1: 5-fold cross-validation results comparison table for average weight prediction

RMSE mean	RMSE std	MAE mean	MAE std	R^2 mean	R^2 std	ML algorithm	Data split
5.6354	1.6555	3.1602	0.4168	0.3668	0.1562	RF	random
5.8780	1.1538	3.2190	0.2839	0.2984	0.0558	RF	field
5.9467	1.7103	3.2700	0.4213	0.2970	0.1563	SVM	random
5.9795	1.4678	3.2074	0.3325	0.2784	0.1486	SVM	field
6.0021	2.1050	3.8622	0.6417	0.1865	0.1456	GB	season
6.0256	2.4689	3.9514	1.1393	0.1895	0.2659	SVM	season
6.0463	1.5032	3.7030	0.4151	0.2643	0.1205	LR	random
6.0673	1.4012	3.6144	0.3158	0.2549	0.1397	LR	field
6.0846	1.3788	3.4362	0.4225	0.2506	0.1095	GB	field
6.0871	2.0870	3.9485	0.8138	0.1590	0.1668	RF	season
6.1066	1.5981	3.5290	0.3743	0.2474	0.1613	GB	random
6.7948	2.1744	4.6218	1.0737	-0.0672	0.2589	LR	season

Table 2.2: 5-fold cross-validation results comparison table for average LWR prediction

RMSE mean	RMSE std	MAE mean	MAE std	R^2 mean	R^2 std	ML algorithm	Data split
0.6195	0.1013	0.4533	0.0528	0.5766	0.0801	RF	field
0.6315	0.1043	0.4539	0.0506	0.5611	0.0808	GB	field
0.6512	0.1117	0.4901	0.0621	0.5321	0.0977	LR	field
0.6534	0.0817	0.4780	0.0559	0.5248	0.1118	RF	random
0.6600	0.1183	0.4835	0.0714	0.5210	0.0998	SVM	field
0.6656	0.0779	0.4810	0.0507	0.5068	0.1072	GB	random
0.6685	0.0820	0.4846	0.0588	0.5051	0.1042	SVM	random
0.6797	0.0920	0.5079	0.0765	0.4883	0.1116	LR	random
1.0291	0.4480	0.8599	0.4493	-0.5290	1.3134	LR	season
1.0838	0.4686	0.9033	0.4575	-0.6355	1.3688	SVM	season
1.1109	0.5986	0.9456	0.6112	-0.9080	1.9252	RF	season
1.1750	0.6879	1.0048	0.6860	-1.2365	2.3896	GB	season

Feature Importance Analysis

The feature importance rankings for the best-performing models are presented in Figures 2.5 and 2.6, revealing a consistent pattern in the features that contribute to predictive performance across models and predicted variables. This analysis highlights key farming, temporal, and stationary factors critical to predicting sweetpotato shape attributes.

For average weight, consistently important features across all three data splits included days after planting, average temperature, days after planting of liquid fertilizer application, minimum relative humidity, slope, minimum radiation, aspect, elevation, and soil properties such as cation exchange capacity at a depth of 5-15 cm and soil nitrogen at a depth of 0-5 cm.

Similarly, several features important for average weight were also significant for average LWR, but notable differences offered insights into sweetpotato roundness. For average LWR, dominant features included days after planting, minimum relative humidity, average wind speed, average temperature, cation exchange capacity at a depth of 0-5 cm, elevation, soil bulk density at a depth of 5-15 cm, aspect, soil potential of hydrogen (pH) at a depth of 5-15 cm, and the previous year's crop being tobacco. Supplementary figures provide feature importance rankings for all models (A.1, A.2).

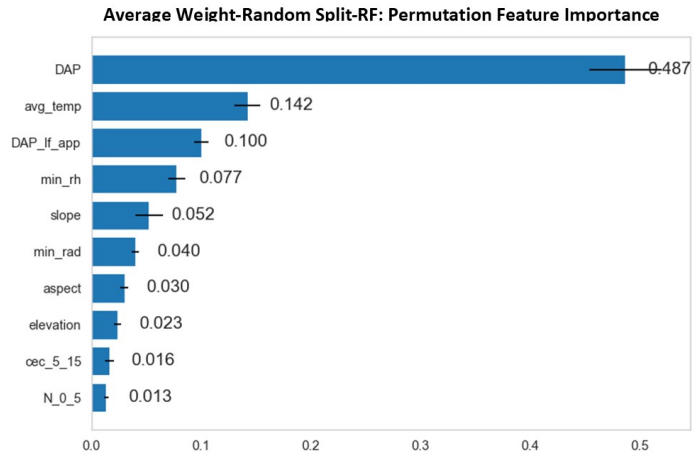

Figure 2.5: Top 10 important features for determining sweetpotato weight using random forest regression and random splits. Where, DAP= Days after planting, avg_tem= average temperature, DAP_lf_app= liquid fertilizer application in terms of days after planting, min_rh= minimum relative humidity, min_rad= minimum solar radiation, cec_5_15= soil's cation exchange capacity at depths of 5-15 cm, N_0_5= soil's nitrogen at depths of 0-5 cm, and slope, aspect, and elevation are topography characteristics of the field.

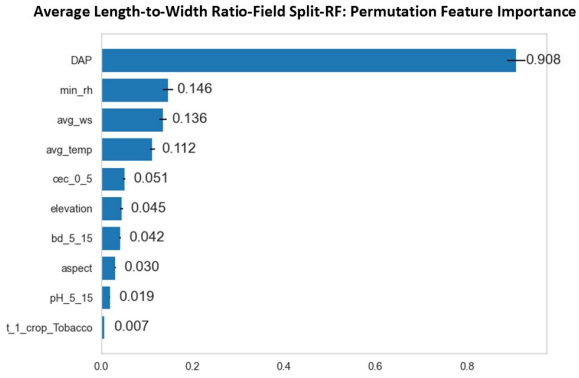

Figure 2.6: Top 10 important features for determining sweetpotato length-width ratio using random forest regression and random splits. Where, DAP= Days after planting, min_rh= minimum relative humidity, avg_ws= average wind speed, avg_temp= average temperature, cec_0_5= soil's cation exchange capacity at depths of 0-5 cm, bd_5_15= soil's bulk density at depths of 5-15 cm, ph_5_15= soil's potential of hydrogen at depths of 5-15 cm, t_1_crop_Tobacco = where tobacco was planted and harvested the year prior in that field, and elevation and aspect, are topography characteristics of the field.

The common features shared between both predicted variables are days after planting, average temperature, minimum relative humidity, elevation, and aspect. Although soil cation exchange capacity is important for both targets, the relevant depth varies. This overlap underscores the critical role of these features in sweetpotato development during the growing season and highlights the stability and reliability of the models, as these features consistently emerged as significant across different models, even if their order of importance varied.

Unique features also provide valuable insights. For average weight, days after planting of liquid fertilizer application is particularly important, while soil characteristics such as bulk density and soil pH are more relevant for predicting average length-width ratio (LWR). Furthermore, the importance of the previous year crop for predicting LWR suggests a potential role of crop rotation in sweetpotato development.

Partial dependence plots further elucidate these relationships. Figure 2.7 shows that for average weight: 1) DAP values of 50 and 65 lead to increases in predictions from 4–6.5 oz and 6.5–8.5 oz, respectively. 2) Average temperature increases from 76–77.5°F and 80–81°F result in decreases in predictions from 10–8 oz and 8–7 oz, respectively. 3) Liquid fertilizer application around 40 DAP increases the prediction of average weight from 8 to over 11 ounces. 4) Minimum relative humidity values of 55% or higher correspond to decreases in average weight predictions. 5) Decreasing trends in weight prediction are observed around 77°F and 80°F. For average LWR, Figure 2.8 shows that: 1) Increasing DAP results in a decreasing trend in LWR predictions. 2) Minimum relative humidity values above 55% lead to increases in LWR predictions. 3) Average wind speeds above 3.2 m/s and temperatures above 76°F are associated with increases in LWR predictions.

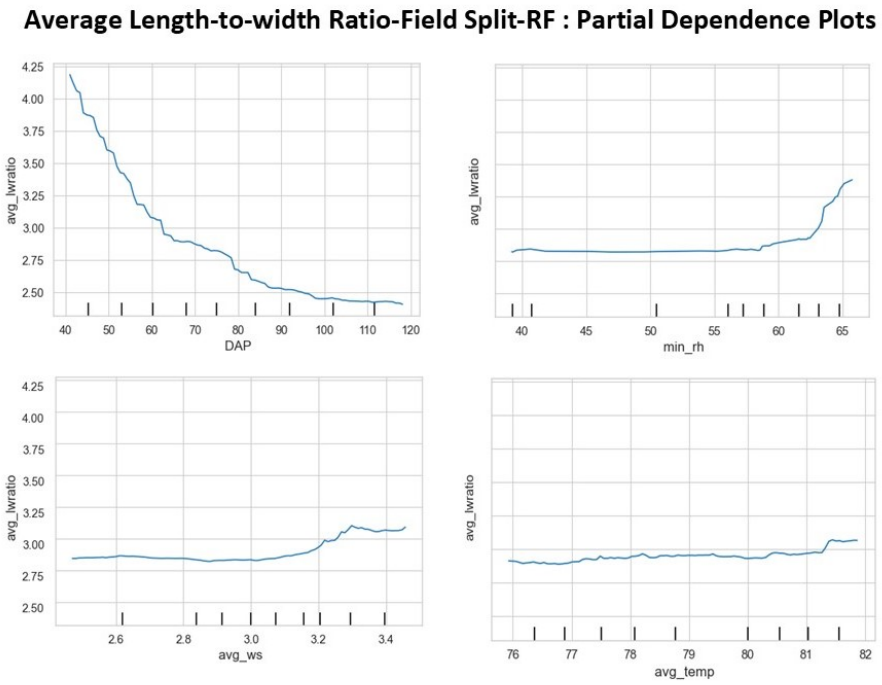

Figure 2.7: Partial dependence plots of the top 4 important features for predicting sweet-potato weight using random forest regression and random data splits. Where, DAP= Days after planting, avg_tem= average temperature, DAP_lf_app= liquid fertilizer application in terms of days after planting, and min_rh= minimum relative humidity.

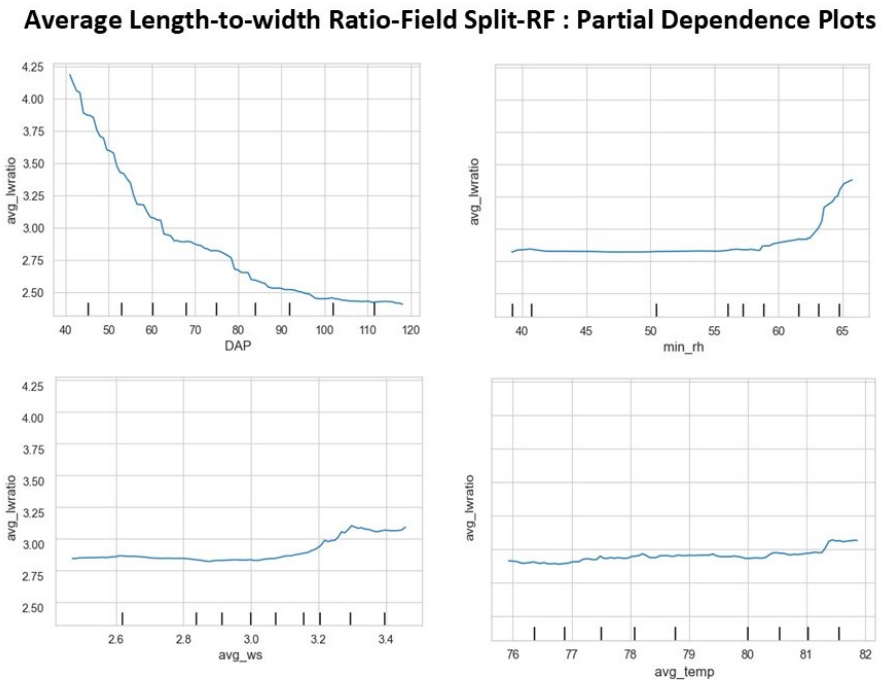

Figure 2.8: Partial dependence plots of the top 4 important features for predicting sweet-potato length-width ratio using random forest regression and by field data splits. Where, DAP= Days after planting, min_rh= minimum relative humidity, avg_ws= average wind speed, avg_temp= average temperature.

2.4 Discussion

2.4.1 Comparison of Results with Existing Literature

This study highlights the influence of environmental factors such as average temperature, minimum relative humidity, elevation, and aspect on the size and shape of sweetpotato. Field management practices, such as the application of liquid fertilizers, are crucial to determining weight, while soil characteristics and crop rotation are more significant for shape. These findings align with those of Carbajal-Carrasco et al. (5), who predicted sweetpotato yield at the county level.

Unlike previous studies (5; 3; 23), this analysis incorporates multiple data splits to emphasize field- and season-specific factors. Supplementary figures (A.1, A.2) reveal consistent feature importance across data splits, though the order of importance varies. For average weight, the random and field splits show similar feature rankings, with elevation and soil cation exchange capacity at a depth of 5–15 cm (cec_5_15) reversing in importance. In the season split, elevation gains prominence, with shuffling this feature reducing model performance by 0.026. For average LWR, random and field splits also have consistent rankings, but the importance of average wind speed (avg_ws) and minimum relative humidity (min_rh) is reversed. The season split highlights increased importance of average temperature (avg_temp), bulk density at 5–15 cm (bd_5_15), and when the previous year's crop was tobacco (t_1_crop_tobacco), with model performance decreases of 0.120, 0.045, and 0.028, respectively, when these features are shuffled.

A strong positive correlation between soil bulk density and sand content suggests that sandy soils, which typically exhibit higher bulk density, are favorable for sweetpotato growth (43; 44; 5). These two variables may be interchangeable in modeling as a high bulk density value indicates high sand content in soils.

Season and field data splits provide practical insights for growers, while the random split was used as the baseline. Table 2.3 shows average 5-fold model performance metrics, with the baseline highlighted in red. The season split produced the lowest coefficients of determination (R^2) for both targets, likely due to limited data (three growing seasons). Field splits increased R^2 by 60% and 209% for average weight and LWR, respectively, with RMSE decreasing by 2.1% and 40%. Random splits improved R^2 by 97% and 199% and reduced RMSE by 6% and 37%, respectively.

Liu et al. (3) predicted LWR using post-harvest data and agronomic features, achieving an RMSE of 0.192 and R^2 of 0.3996. In contrast, our RF model (field split) achieved

RMSE of 0.6195 and R^2 of 0.58, reflecting a broader range of LWR values given that our analysis included sweetpotatoes throughout a growing season (1.49–8.69 vs 1.4–2.6) and field-specific factors such as soil type rather than GPS coordinates. Gogineni et al. (23) estimated weight using image-based models reached an R^2 of 0.91 using neural networks. While Gogineni et al.'s (23) approach achieved higher prediction accuracy, it was limited to post-harvest applications and controlled conditions. The present study expands on these findings by addressing the prediction of sweetpotato characteristics earlier in the growing season, incorporating agronomic factors, and emphasizing the importance of data splitting for practical and field-specific applications.

To the best of our knowledge, this study addresses a gap by exploring how data splitting highlights agronomic and field management factors critical for predicting sweetpotato shape and size earlier in the growing season, providing actionable insights directly relevant to packing operations and market value.

Table 2.3: 5-fold cross-validation results comparison for both average weight (WT) and average length-width ratio (LWR) predictions

RMSE mean	MAE mean	R^2 mean	Algorithm	Data split	Variable
5.6354	3.1602	0.3668	RF	random	Average WT
5.8780	3.2190	0.2984	RF	field	Average WT
6.0021	3.8622	0.1865	GB	season	Average WT
0.6195	0.4533	0.5766	RF	field	Average LWR
0.6534	0.4780	0.5248	RF	random	Average LWR
1.0291	0.8599	-0.5290	LR	season	Average LWR

2.4.2 Practical Applications: A Use-Case for Growers

Using the best-performing models presented herein, growers can predict crop development during the growing season if they have field coordinates, weather, soil, and field management variables for a specific field. Figures **??** and 2.10 compare observed and predicted values for both targets using unseen data. For average weight, the model achieved RMSE = 3.35, MAE = 2.38, and $R^2 = 0.69$, indicating that 69% of the variance in average weight can be explained by the model. The RMSE value suggests an average prediction error of ±3.35 ounces for three sweetpotato plants. Figure 2.11 shows that the model predictions align closely with the observed values until 75 DAP, after which the accuracy decreases as the DAP increases. For average LWR, the model achieved an RMSE = 0.76, MAE = 0.40 MAE=0.40, and $R^2 = 0.56$, explaining 56% of the variance. The RMSE indicates an average error of

±0.76 in LWR predictions for three sweetpotato plants. Figure 2.12 highlights a contrasting trend: The model's predictions for average LWR improve as DAP increases, showing a closer alignment with observed values after 60 DAP.

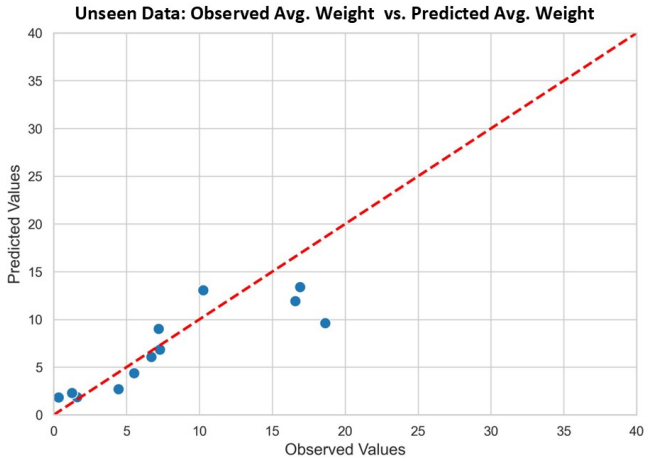

Figure 2.9: Predicted vs. observed validation plot when predicting average weight using the random split random forest regressor.

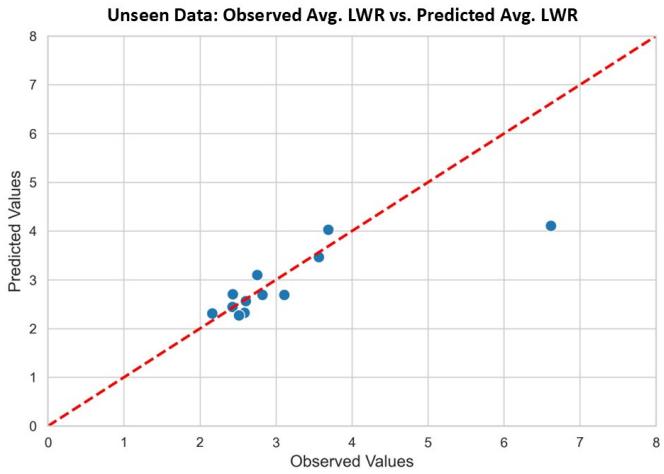

Figure 2.10: Predicted vs. observed validation plot when predicting average length-width ratio using the field split random forest regressor.

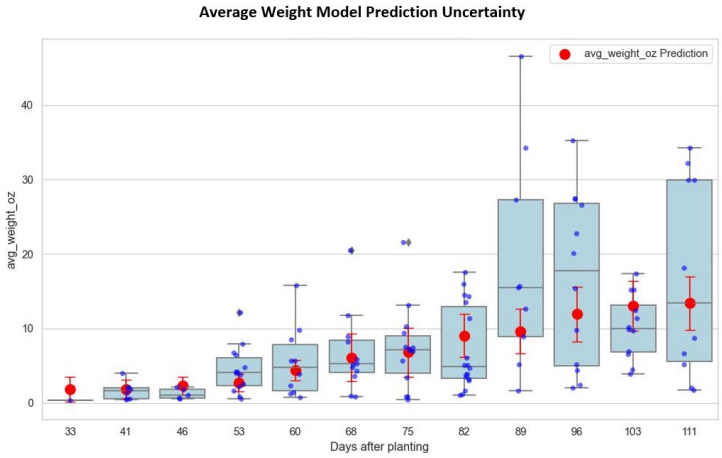

Figure 2.11: Comparison of the observed weight and average weight values superimposed with the average weight predictions for one holdout field

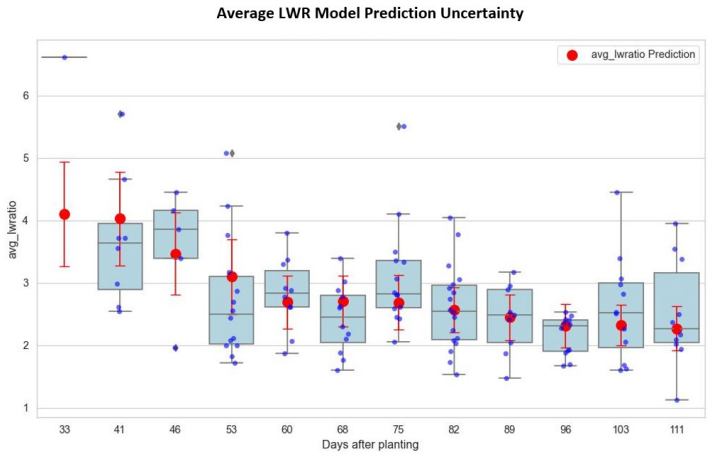

Figure 2.12: Comparison of the observed LWR and average LWR values superimposed with the average LWR predictions for one holdout field

The random and field splits yielded similar performance for both targets, suggesting potential interchangeability. However, the field split showed a 5.19% decrease in RMSE and a 10% increase in R^2 for average LWR compared to the random split. For average weight, the field split increased RMSE by 4.3% and decreased R^2 by 19%. While the random split is easier to implement without requiring field grouping, the field split provides marginal benefits for specific variables. Furthermore, growers can easily reproduce this analysis since all data required for these predictions are accessible using latitude and longitude coordinates.

2.4.3 Study Limitations

Limitations of this study include the assumption that each photo data point represents a potential harvest date. Additionally, due to time and resource constraints, only three sweetpotato plants were sampled per photo image. This approach was chosen to emphasize sampling from a diverse range of fields rather than collecting a large number of samples from a few fields to better capture field-specific variability. The analysis does not account for surface defects of sweetpotatoes, which may affect grading and quality predictions. Furthermore, the MaskRCNN model used in this study was trained exclusively on sweetpotatoes weighing between 3–32 ounces, limiting its applicability to sweetpotatoes outside this weight range.

Because the study relied on on-farm sampling without intervention in regular industry practices, field management varied over the years. In 2021, dry fertilizer was applied approximately 40 days before planting, liquid fertilizer about 35 days after planting, and herbicide and insecticide 40 days after planting. Management practices in 2022 and 2023 were more consistent, with dry and liquid fertilizers applied 5-7 days before planting, except in a field where the application occurred 60 days before, and herbicide and insecticide applied approximately 18 days after planting.

Future work will address these limitations and reorganize the dataset to incorporate the combined target variable for weight and length-width ratio. This integrated approach will be analyzed using long short-term memory (LSTM) neural networks to account for temporal dynamics. Additionally, reinforcement learning techniques will be explored to identify optimal crop management policies, providing actionable insights for growers and industry stakeholders.

CHAPTER

3

DATA AND MACHINE LEARNING PIPELINE FOR CROP DEVELOPMENT AND QUALITY PREDICTION

3.1 Introduction

During the methodology of Chapter 2, many methods, suggestions, and techniques can be applied when working with agricultural data. There was no general guideline to follow when preparing agricultural datasets for machine learning. In this chapter, we address the need for a detailed protocol for maintaining an efficient data pipeline and robust machine learning framework, as well as considerations for data splitting, handling missing data, and performing feature selection. The result of this chapter is a fully connected machine learning pipeline that takes a set of unique identifiers and xy coordinates as input, and outputs a model comparison of commonly used models in agriculture to aid in decision making that can be used by researchers, extension agents, and growers. The data pipeline presented herein is available at [add github link please].

3.2 Method Details

Our objective is to present and share a reproducible data processing and machine learning pipeline designed to integrate environmental, agronomic, and in-season crop data for predicting crop development and quality. This pipeline aims to be easily accessible for researchers and extension agents while also providing general guidelines for preparing agricultural datasets suitable for machine learning applications. Furthermore, this methodology presents a streamlined approach by consolidating common machine learning methodologies found in agricultural literature, such as feature scaling, feature selection, and hyperparameter tuning, into a cohesive pipeline. By streamlining these techniques, we offer users a robust framework that can be applied to root vegetables, such as sweetpotatoes, carrots, and beets, thereby enhancing decision-making in underground cropping systems.

3.2.1 Data Pipeline

Data Collection and Compilation

This section requires a CSV file containing columns for field location IDs, data collection dates, and latitude and longitude coordinates, which are essential for obtaining temporal information such as weather, and stationary data such as soil and topography. The subsequent sections will describe these processes in detail.

Farm Data

Farm data includes any manually collected information, small-plot research experiments, or automatic data collections from sensors in field machinery and stored in a tabular format that includes a series of dates for each observation.

Weather Data

Daily weather data can be extracted from NASA Power (33). For weather data collection, we present three interconnected R scripts - illustrated as A, B, and C in Figure 3.1, each building on the previous one through the source() function. Each script contains a function that performs a specific task within the workflow:

- weather_data_fetch: takes as input a unique location ID, latitude and longitude coordinates, and start and end dates of the data to be captured and connects to the API

to pull the daily weather values and returns a clean row of data representative of a single location.

- weather_data_pull: takes as input farm data in tabular form and the start and end dates of data collection. This function iterates through the rows of the given dataset and pulls weather data for all locations. The output is an integrated dataframe of daily weather values for all locations.

- weather_data_integrate: Integrates farm, field management, and weather data and performs feature engineering.

The daily weather variables extracted from NASA Power include total precipitation (mm), solar radiation (W/m^2), wind speed (m/s), relative humidity (%), and temperature (°C). Average, minimum, and maximum values for solar radiation, temperature, wind speed, and relative humidity could be calculated, along with cumulative values for precipitation and growing degree days.

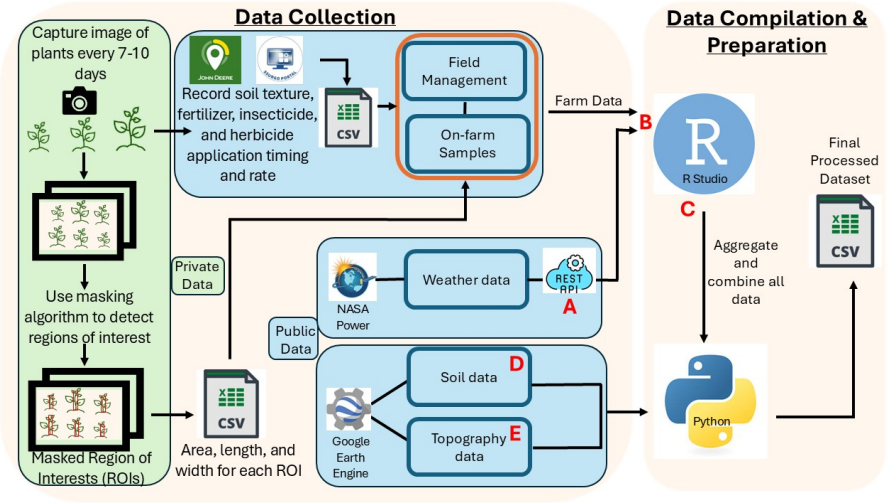

Figure 3.1: Overall workflow of the data pipeline to collect and combine private on-farm samples and field management data, extract and integrate public heterogeneous data, and prepare data for machine learning models. (A). Extracting raw weather data with the "weather_data_fetch" R script. (B). Pulling cleaned weather data using unique location IDs and xy coordinates from the given farm data with the "weather_data_pull" R script, (C). Integrating weather data and farm data with the "weather_data_intgerate" R script. (D). Pulling soil characteristics for a set of unique location IDs using Google Earth Engine. (E). Pulling topography data for a set of unique location IDs using Google Earth Engine.

Soil and Topography Data

We also present a Python script that extracts stationary soil characteristics using the latitude and longitude coordinates of each field from ISRIC World Soil Information SoilGrids (34) global gridded soil information using Google Earth Engine, 3.1 Figure 1(D). The soil characteristics include bulk density (bd, g/cm^3), cation exchange capacity (meq/100g), clay content (%), silt content (%), sand content (%), nitrogen (g/kg), pH, and soil organic carbon (g/kg). All soil variables are from depths of 0-5 and 5-15 cm at a resolution of 250 m. A duplicate of this script was adapted to incorporate the digital elevation model (DEM) from the Shuttle Radar Topographic Mission (SRTM), originally produced by NASA and later refined by the Consortium for Spatial Information (36), Figure 3.1 (E). This change added topographical features such as elevation (meters), aspect (degrees), and slope (degrees) to the stationary features.

3.2.2 Data Compilation

Datasets could be joined based on the field location name.

Data Splitting

When preparing data for machine learning, the dataset is split into training, validation, and testing subsets- each serving a distinct role in the model development and evaluation. The validation subset is used to validate the training of the model and provides a glimpse of how well the model is generalizing to unseen data, and the testing subset is used as a final evaluation of model performance. To ensure unbiased performance, these subsets must be representative and non-overlapping. Common splitting strategies include random, stratified, grouped, time-dependent, and cross-validation.

Random splitting partitions data based on specified ratios (e.g. 60/40, 70/30, 80/20), with the larger portion being the training set and the smaller portion the testing set. The training set is typically split further using ratios between 10-30% to obtain the validation set. Random splitting is most appropriate for balanced datasets, as it can introduce bias in unbalanced data by overrepresenting dominant categories.

Stratified splitting maintains consistent proportions of each category (e.g.year, true/false, or low/medium/high). Grouped splitting keeps all observations from the same group (e.g. location, farm) within a single set (training, validation, or testing) to prevent data leakage. Time-based splitting preserves the chronological order of observations and is commonly

used for time series or sequential data to prevent future data from influencing past predictions.

Cross-validation is a robust splitting strategy where the dataset is divided into multiple folds, and the model is trained and tested iteratively across these folds (45; 46).

This rotating process allows for a comprehensive evaluation of the model's true performance, in contrast to fixed train-test splits. Cross-validation can be customized by combining previously mentioned techniques, such as k-fold, stratified k-fold, and group k-fold (46; 47). K-fold divides data into k equally sized folds, trains the model on k-1 folds, and tests on the remaining fold, repeating this process for k iterations. Stratified and group k-fold are special cases of K-fold where the folds either preserve the distribution of a specific category or class, or ensure that all observations from a given group remain within a single fold.

Handling Missing Data

Missing or incomplete data can reduce model performance, making it essential to understand the underlying causes and apply appropriate handling strategies. Data is considered missing completely at random (MCAR) when the likelihood of missing values is unrelated to both the observed data and unobserved or missing data, meaning there is no systematic pattern to the missingness (48). For example, a weather station might have gaps in its data due to random power outages or internet failures, which are unrelated to the weather conditions or any other recorded variables. Data is considered missing at random (MAR) when the likelihood of missing values depends on the observed data but not on the missing data itself (48). This means the reason for the missing values is related to some variables in the dataset, but not the variable that is missing. For example, missing soil sensor readings may be associated with sensor location rather than the soil nutrient values, revealing a pattern tied to a known variable. In contrast, not missing at random (NMAR) occurs when missingness is related to both the observed data and the missing data (48). For example, crop damage being unreported due to its severity makes it difficult to account for using other variables.

Once the type of missingness is determined, the appropriate strategy for handling it depends on its nature and extent, as well as the importance of the variable in the analysis (49). Common strategies include deletion and imputation. Deletion removes rows or columns with missing values and is generally acceptable when the data are Missing Completely at Random (MCAR) and the percentage of missingness is low, since the remaining data still

represent a random subset of the original dataset (49). In contrast, imputation replaces missing values with the mean, median, or more sophisticated techniques like k-Nearest Neighbors (kNN), which leverages local similarity -averaging the k nearest neighbors to fill gaps. Imputation is preferred when data are Missing at Random (MAR), allowing one to exploit existing relationships in the data (49). However, when data are Not Missing at Random (NMAR), more advanced methods are required, including the inclusion of auxiliary variables that capture the mechanism behind the missingness (48).

3.2.3 Machine Learning Algorithms

The predictive algorithms included in the coding pipeline are Random Forest, Gradient Boosting, Support Vector Machine, and Linear Regression. These models were selected based on their popularity and demonstrated effectiveness in agricultural research for yield prediction (15). The pipeline, described in the following section, was used to evaluate the performance of each model.

3.2.4 Machine Learning Pipeline

Model Hyperparameter Tuning and Evaluation

Nested K-fold Cross-Validation Nested K-fold cross-validation is a robust approach for simultaneously tuning hyperparameters and assessing model performance (50; 51). This method provides a reliable estimate of generalization error and minimizes bias, enabling a more accurate and consistent evaluation of model reliability across different data partitions (52; 53; 50; 51). Nested K-fold cross-validation consists of an outer and an inner cross-validation loop. The outer loop estimates model performance, while the inner loop handles model development through hyperparameter tuning and feature selection. This process begins by initializing the outer loop and splitting the full dataset into k folds (typically 3, 5, or 10), as shown in Figure 3.2 (F). In each iteration of the outer loop, one of these folds is used as the test set, and the remaining $k-1$ folds are used to create the training set. Next, within the training set established by the outer loop, another k-fold cross-validation is performed in the inner loop, Figure 3.2 (G). Once the inner loop is initialized, each training subset is further split into training and validation subsets. Within the inner loop, several combinations of hyperparameters are used to obtain the best-performing model and hyperparameters, Figure 3.2 (H). Finally, the best-performing model chosen from the inner loop is then passed to the outer loop to evaluate the model using the testing subset

from the corresponding outer fold, Figure 3.2 (I, J). Once all iterations from the outer loop are finalized, average metrics like Root Mean Squared Error (RMSE), Mean Absolute Error (MAE), and the Coefficient of Determination (R^2) are calculated to estimate how the model would perform on unseen data, Figure 3.2 (K). While this method is more computationally expensive, it provides a less biased approach to optimizing hyperparameters and model evaluation compared to traditional cross-validation, as bias can be introduced during the initial training and testing partitions, or through feature or parameter selection (52; 53). This method allows the model to be trained on the entire dataset via smaller subsets of the full dataset, making it efficient when dealing with smaller datasets (51).

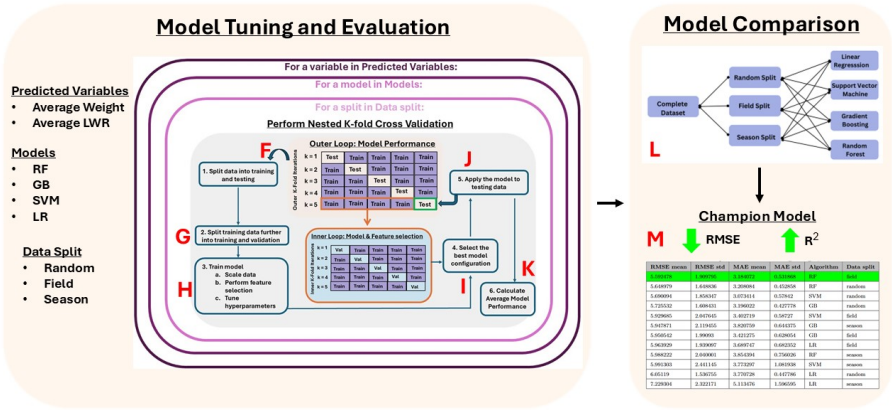

Figure 3.2: Overview of the machine learning pipeline. (F). Initial data splitting for the outer loop of nested K-fold cross-validation (G). Secondary data splitting for the inner loop of nested K-fold cross-validation (H). Model training, feature scaling, and hyperparameter optimization (I). Selection of the best-performing model combination from the inner loop (J). Model evaluation using the test set (K). Calculating the average metrics (RMSE, MAE, R^2) across outer folds (L). Example representation of models processed in the method validation use-case (M). Conditions used to choose the champion model

Grid Search Cross-Validation Grid search cross-validation is a method used to identify the best combination of hyperparameters for a machine learning model. It exhaustively tries all combinations of parameter values from a given grid of potential candidates, using cross-validation to evaluate how each set of parameters performs. The optimal parameters

are obtained by training and validating the model multiple times, and the combination that produces the highest average performance is selected. This helps improve the models' generalizability on unseen data and can be incorporated into the inner loop of nested K-fold cross-validation. Hyperparameter tuning is similar to having a tailor-made suit, as the parameters are specifically fitted to the data. Aside from grid search, other options for hyperparameter tuning are Randomized CV and Bayes CV. Randomized CV is similar to Grid Search CV, but randomly samples subsets of parameter combinations instead of performing an exhaustive search over the parameter grid (54). A search space is defined using a dictionary of distributions for each parameter, rather than exact values, and parameter values are sampled from these distributions, resulting in a faster run time when compared to Grid Search CV. Bayes' CV is based on the theory of Bayes' Theorem, where the probability of an outcome, known as the posterior, is updated as new information becomes available (54). Like Randomized CV, Bayes CV begins by specifying distributions for each parameter value, but these distributions are considered prior distributions. Each new iteration of the algorithm uses the results from the previous iterations to update and guide its search toward more promising hyperparameter values, making this technique more efficient and typically faster than Grid and Random search (54).

Feature Transformation Machine learning models can learn more effectively when input values are on a similar scale, so data are often transformed using mathematical functions. Feature transformations prevent any single numerical feature from dominating the model's output due to its magnitude. Rescaling features to smaller or more uniform ranges helps improve algorithm performance and stability during training. Minmax scaling or normalization has been a commonly used feature scaling method with sweetpotato yield and shape prediction research (5; 3). For that reason, all features were normalized to values between 0 and 1 using the Minmax scaler from Python. This means that after the data transformation, a feature variable's minimum value is converted to 0, its maximum value is converted to 1. All other values fall in between, ensuring that full range of each feature is retained without altering the relationships within the data. Outliers can affect the results of some feature transformation methods, so it is important to identify and appropriately address them before applying transformations. While many outlier detection methods exist, statistical detection methods tend to be a quick and practical solution (55). In particular, the interquartile range (IQR) method that is based on the five-number summary, minimum, first quartile (Q1), median or second quartile (Q2), third quartile (Q3), and maximum value provides a way to describe the full distribution of a variable. Boxplots are a common way

of visualizing this five-number summary (55). The IQR represents the distance between the first and third quartiles ($Q3-Q1$). Values that fall below $Q1-1.5*IQR$ and above $Q3+1.5*IQR$, are treated as outliers. This outlier detection method is especially relevant when the data distribution is not normal or when models, such as neural networks, require bounded input values.

Other commonly used feature transformations that were not included in this work are standardization and log-transformation. Standardization transforms the data to have a mean of 0 and a standard deviation of 1 (56). This method is less affected by outliers compared to MinMax scaling and is commonly used when the data are normally distributed. Log transformation alters the data by computing the logarithm of each variable (57). This method can be useful in reducing skewness and bringing the data closer to a normal distribution.

Feature Selection Feature selection involves reducing the number of feature variables by removing variables that are redundant or do not add much predictive value, leaving only the most informative features for predicting the response variable potentially improving model accuracy (58; 59). Four categories of feature selection are commonly used: filter, wrapper, embedded methods, and hybrid methods (58; 59; 60). Filter-based methods, which are often statistical, reduce the number of feature variables by analyzing their relationships with the response variable and themselves and removing variables that fall above or below a specific criterion (59). Examples include correlation and the chi-square test (60; 61). Wrapper-based methods involve training a machine learning model on different subsets of feature variables, iteratively adding or removing unimportant features until a desired number of features is obtained or the highest model performance (59). Common examples of wrapper-based methods include recursive feature elimination (60) which iteratively removes the least important features based on model performance, and the Boruta method (62), which identifies all relevant features by comparing them against randomized shadow features. Embedded methods automatically integrate feature selection into the model training process, typically minimizing the errors between observed and predicted variables (59)). Examples include Least Absolute Shrinkage and Selection Operator (LASSO), which applies L1 regularization to shrink less important feature coefficients to zero, and Elastic Net, which combines L1 and L2 regularization to balance sparsity and stability in feature selection (63). Hybrid methods combine filter and wrapper methods to leverage the benefits of both methods to improve model performance (59; 60). Published literature in agriculture reports that combining the benefits of both filter and wrapper-based methods enhances

model performance (60; 40). The case study presented in this manuscript highlights the use of a hybrid approach that combines correlation-based feature selection and random forest recursive feature elimination.

- Correlation-based feature selection (CBFS): To reduce feature dimensionality, one approach involves determining the correlation between all features and the response variable, with correlation values sorted in descending order. Next, correlations among features could be evaluated by applying a redundancy threshold (e.g., 0.6). For each pair of features with a Pearson correlation coefficient greater than or equal to the threshold, their individual correlations with the response variable are compared. The feature with the lower correlation to the response is removed from the analysis, thereby reducing redundancy while retaining the most informative features for predictions.

- Recursive feature elimination (RFE): To further reduce data dimensionality, recursive feature elimination begins by specifying the number of features to maintain at the end of the feature selection process (e.g., 10). Features passed from CBFS are ranked based on their importance scores using a random estimator (e.g., random forest), and the most important features are retained for modeling (e.g. top 10 most important).

3.2.5 Model Comparison

To evaluate performance, the average value of the following metrics is reported for each model: root mean squared error (RMSE, Equation 3.1), mean absolute error (MAE, Equation 3.2), and coefficient of determination (R^2, Equation 3.3). Specifically, RMSE and MAE both measure the difference between observed values and predicted values using the same units as the predicted variable, but differ in the way they are affected by outliers; R^2 represents how much variance can be explained from the feature variables (64). All algorithms- Linear Regression, Support Vector Machine, Random Forest, and Gradient Boosting- would be compared to select a champion model based on average performance metrics, with priority given to mode exhibiting the lowest RMSE value and highest R^2 value, Figure 3.2 (M).

$$\text{RMSE} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2} \tag{3.1}$$

$$\text{MAE} = \frac{1}{n}\sum_{i=1}^{n}|y_i - \hat{y}_i| \tag{3.2}$$

$$R^2 = 1 - \frac{\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}{\sum_{i=1}^{n}(y_i - \bar{y})^2} \tag{3.3}$$

3.3 Method Validation

The method described in this paper was applied by (25) to predict sweetpotato average weight and length-width ratio. The data were collected from 57 field locations over 3 growing seasons. Images of sweetpotatoes were taken every 7-10 days and processed using MaskRCNN (37; 1) to obtain the width, length, and area of each sweetpotato in an image. These image-derived features were then combined with field management information such as planting and harvest dates and fertilizer, insecticide, and herbicide application rates and times. Using each field's latitude and longitude, these data were combined with daily weather data, soil characteristics, and topography data. Four predictive models were produced and compared for each predicted variable, average weight and length-width ratio for three types of data splits, random, by field location, and by growing season. Model training and validation included feature scaling and selection implemented through a nested cross-validation splitting scheme for the four models: Linear Regression, Support Vector Machine, Gradient Boosting, and Random Forest. The predictive performance of the designed models was evaluated using the RMSE, MAE, and R^2. The random forest algorithm performed the best across both predicted variables. The results indicated that for average weight, the model achieved the following metrics: RMSE = 5.64, MAE = 3.16, and $R^2 = 0.37$, meaning that 37% of the variance in average weight can be explained by the model. For the average LWR, the model produced an RMSE = 0.62, MAE = 0.45, and $R^2 = 0.58$, indicating that 58% of the was explained.

3.4 Limitations

While nested K-fold cross-validation provides an unbiased approach to hyperparameter tuning and feature selection, it is computationally expensive and can make implementation challenging, particularly with large datasets or limited resources. In addition, correlation-based feature selection methods may rely on redundancy thresholds that are too strict. This can lead to the removal of features that, despite being correlated with others, still contain valuable predictive information. As a result, the final feature set may retain variables that appear weakly associated with the response, yet lack true predictive or causal relevance.

CHAPTER

4

PREDICTING SWEETPOTATO GROWTH AND QUALITY CHARACTERISTICS USING SPATIOTEMPORAL APPLICATIONS AND MACHINE LEARNING

4.1 Introduction

In Chapter 2, the data instances for each field location were considered independent observations, based on the assumption that each photo date was a potential harvest date. This approach, however, overlooked the temporal relationships between the different photo time steps for a specific field location. To address this limitation, we used Long Short-Term Memory (LSTM) recurrent neural networks to capture the temporal dependencies between photos obtained during the data collection process. Additionally, we analyzed whether an unsupervised clustering-based algorithm would enhance model performance. Lastly, model performance was compared between chapters 2 and 4 by including random forest

(RF) and extreme gradient boosting (XGB) models in the analysis.

4.2 Materials and Methods

4.2.1 Data Collection

The dataset used in this study was the same as McDowell et al. (25), collected over three growing seasons (2021–2023), comprising images of cleaned sweetpotato roots from three plants sampled every 7–10 days across 57 field-years; 21 in 2021, 18 in 2022, and 18 in 2023. After collection, McDowell et al. (25) aligned weather, soil, topological data, growing degree days (GDD), and days after planting (DAP) values with each observation using a mix of private and public sources.

4.2.2 Data Processing

Feature Engineering

To determine if the data were spatially dependent, spatial autocorrelation was utilized and confirmed using Moran's I and corresponding p-values. K-means clustering was applied to group the data based on location and other similar features. This process introduced a new feature, cluster ID, which served as a splitting mechanism during model training. To determine the optimal number of clusters (K) for K-Means clustering, the elbow method was employed to examine the relationship between the within-cluster sum of squares and the number of clusters.

Data Structure

To prepare the data for an LSTM, the dataset from Chapter 1 was restructured into the 3-dimensional format of samples, time steps, and features, where samples represented the number of field locations, and time steps corresponded to the number of photos taken at each location. The number of photos taken for each field location ranged from 5 to 15, as shown in Figure 1 4.1. This led to gaps in the data for field locations with shorter sequences. To handle this range variation, sequence padding was utilized to fill in the missing timestep information, ensuring that all sequences were the same length as the longest sequence. Sequences can be pre- and post-padded or truncated to obtain the same lengths. The maximum length of our field sequences was 15. Therefore, we post-padded each sequence

with missing values(Nan, or Not a Number)to a length of 15 to preserve the temporal order of the data. Nan values were used during padding so as not to skew the data values during feature scaling.

Features for each time step included in this analysis were based on the top ten features identified as most important for average weight and length-width ratio in McDowell et al. (25). Common features included in this analysis to predict average weight and length-width ratio were days after planting (DAP), average temperature (avg_temp), aspect, minimum relative humidity (min_rh), elevation, and cation exchange capacity (cec_5_15, cec_0_5) at different depths for average weight and length width ratio, respectively. Additionally, minimum radiation (min_rad), days after planting for liquid fertilizer application (DAP_lf_app), slope, soil nitrogen (N_0_5), and lagged values of average weight (avg_weight_oz) were unique variables used to predict average weight. Unique variables used to predict average length-width ratio were average wind speed (avg_ws), soil bulk density (bd_5_15), soil pH (pH_5_15), the previous year's crop being Tobacco (t_1_crop_Tobacco), and lagged values of average length-width ratio (avg_lwratio).

Our goal was to determine whether the previous two photo time steps could be used to predict average weight and length-width ratio at the next photo time step. This prediction or forecast is considered short-term as the prediction covers a time frame that is less than a year (65). Therefore, to align with our objective, each field location's sequence was further divided using the sliding window approach. For example, for a field location with 6 photos taken during the growing season, the average weight/length-width ratio values and features at time steps 1 and 2 were used to predict average weight/length-width ratio values at timestep 3, features at time steps 2 and 3 were used to predict the predicted value at timestep 4, and so on. For an original sequence of 6 time steps, this results in 4 sequence pairs of length 2 with 10 features at each time step, as seen in the example in Figure 4.2.

Varying features were relabeled as either from time step 1 or time step 2, based on the order of samples. For example, for a specific location, minimum radiation (min_rad) would be relabeled as min_rad_t1 and min_rad_t2, etc. Stationary features were not relabeled for each time step to avoid introducing redundant information.

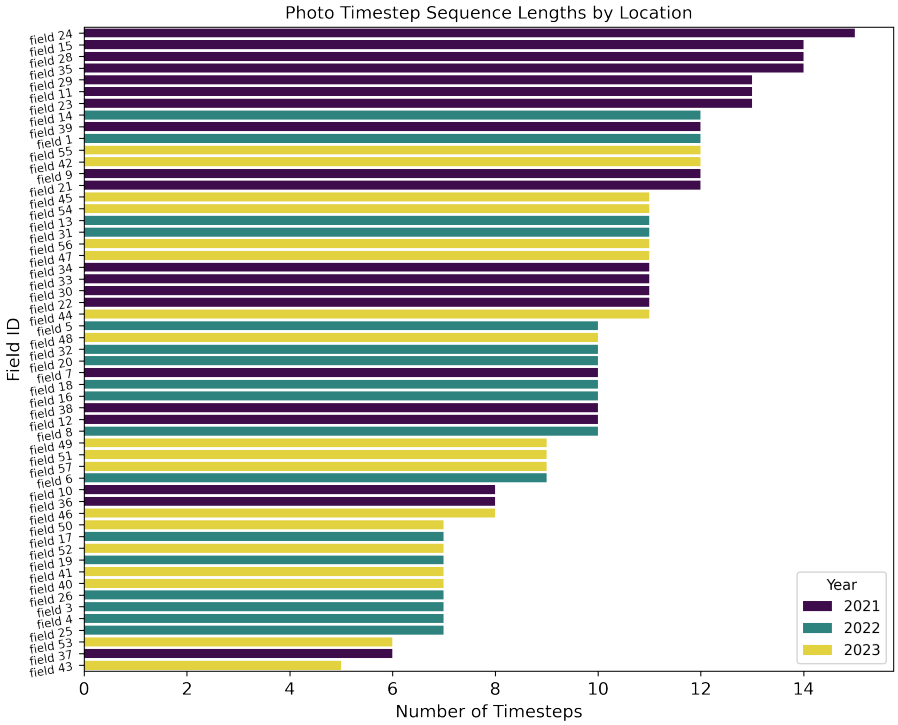

Figure 4.1: Photo timestep sequence lengths for each field location

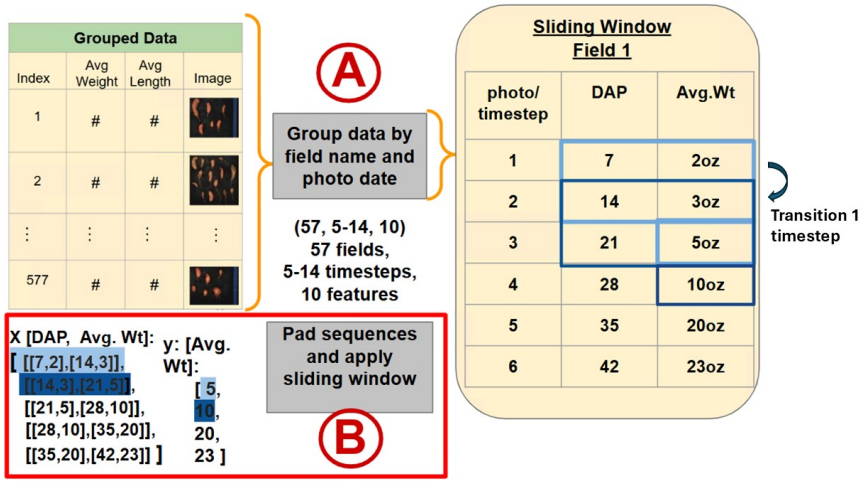

Figure 4.2: Photo timestep sequence lengths for each field location

Data Splitting

Two data splitting schemes were employed in this work: random and grouped splitting. For random splitting, the sequences were split randomly across all folds, and for the grouped split, all data belonging to a specific cluster, as determined by K-means clustering, ensured no data leakage. Both splitting techniques were used with nested K-fold cross-validation, a method also used in (25). For the random data split, K-fold cross-validation was used with $K = 5$ for the outer and inner split. Group K-fold cross-validation with $K = 3$ for the outer and inner splits was used for the grouped data split.

Feature Scaling

To ensure consistent feature scaling and improve model performance, this study applied normalization using Python's Min-Max Scaler, which scales feature values between 0 and 1. The scaler ignored Nan values introduced during sequence padding, preserving their positions and preventing distortion of feature distributions. These Nan values were then replaced with the value –1 (66), a value outside the normalized range, so they could be recognized and ignored by the LSTM's masking layer during training.

4.3 Model Building

Klompenburg et al. (15) reported that linear regression (LR), random forest (RF), support vector machine (SVM), and gradient boosting (GB) are the more widely used machine learning models for predicting crop yield. This work employed linear regression (LR), random forest (RF), and extreme gradient boosting (XGB). Additionally, Long short-term memory (LSTM) recurrent neural networks were also implemented for their memory capability and their ability to capture temporal dependencies within the data. For each model, we calculated the root mean squared error (RMSE), the mean absolute error (MAE), and the coefficient of determination (R^2) to evaluate model performance. When determining the best model, the focus was primarily on RMSE and MAE for model comparison, using R^2 as a supplementary indicator due to its limitations in evaluating the goodness of fit in nonlinear models (39). The LR model is considered a baseline or benchmark model for this work.

4.3.1 Model Architecture

LSTM Network Design

The model architecture selected for this study was built on the CNN-LSTM method described by (67), but only the LSTM architecture was considered, as CNN architecture was not needed for this research study. The predictive model consists of one LSTM layer followed by a dense layer with 64 units, a dropout layer with a rate of 0.5, and a final dense layer with one unit that was used to predict yield. The model also included the Rectified Linear Units (ReLU) activation function. This architecture was modified by incorporating a masking layer before the LSTM layer to identify padded values of -1 within the sequence, to ensure that this information is not learned during training.

Performance Metrics

The champion model for average weight and length-width ratio was chosen by ranking the model performance first based on the lowest RMSE and second on the highest R^2 value. MAE values were also included to evaluate model performance.

4.3.2 Naive Baseline

To evaluate the performance of the average weight and length-width ratio models, a naive baseline was implemented using the naive method (68). The naive method assumes the predicted variable at the next timestep will be equal to the last observed value of the predicted variable. More specifically, the average weight or length-width ratio of the current timestep will be equal to the last observed average weight or length-width ratio value, without incorporating any additional field management or agronomic variables. This method is a common benchmark to assess whether a more complex model adds predictive value. Performance was evaluated using RMSE, MAE, and R^2.

4.3.3 Hyperparameter Tuning

Bayesian Cross Validation

The Bayesian cross-validation search method was employed to determine the optimal hyperparameters. This technique begins by defining a parameter search space dictionary, assigning an upper and lower bound, and a prior distribution for each parameter of interest.

Each new iteration of BayesCV is informed by the previous iteration, potentially decreasing the training time when compared to other methods such as Grid Search and Random Search (54). Since the RF and XGB models are tree-based, some of their hyperparameters are the same, such as the number of decision trees (n_estimators), the learning rate (learning_rate), and the maximum depth of each decision tree (max_depth). The parameters tuned that were unique to the RF were the minimum sample required to be left at each node (min_samples_leaf), and the maximum number of features to be considered by the model when splitting at each node(max_features). The unique parameters tuned for XGB were the maximum number of leaf nodes in each tree (max_leaves), the minimum total data or weight of a child leaf node after splitting (min_child_weight), and the ratio of training instances (subsample). The parameters tuned for the LSTM were the node units of the LSTM layer (lstm_units), dropout rate (dropout), learning rate (learning_rate), epochs (epochs), and batch size (batch_size) were the parameters tuned during training.

4.3.4 Results

Spatial Autocorrelation

Moran's I values for average weight and length-width ratio were 0.2756 and 0.3242, respectively, with corresponding p-values of 0.002 and 0.001, both below the significance threshold of 0.05, indicating moderate statistically significant positive spatial autocorrelation for both predicted variables. This suggests that, across the 3 years, nearby field locations had similar values of average weight and length-width ratio, and that the high and low clusters found were not by random chance. As pictured in the Local Indicators of Spatial Autocorrelation (LISA) plots for average weight (4.3) and length-width ratio 4.4, high values clustered near other high values and low values near other low values, and low-high/high-low are areas to investigate to determine what is different about these locations.

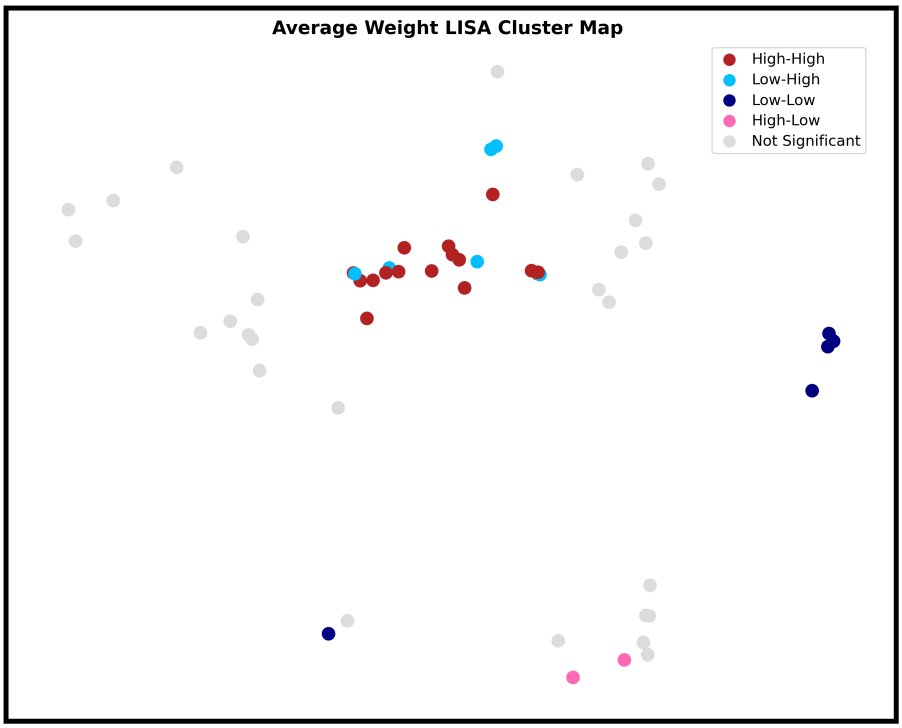

Figure 4.3: Local Indicators of Spatial Association (LISA) Cluster Map that identifies local spatial autocorrelation in the values of Average Weight. Each area is categorized based on the relationship between its value and the values of its neighbors. High-High (HH): corresponds to spatial clusters of high values, Low-Low (LL): corresponds to spatial clusters of low values, High-Low (HL): corresponds to spatial outliers, and Low-High (LH): corresponds to spatial outliers. Grey areas represent field locations with no statistically significant spatial autocorrelation at the significance level of 0.05.

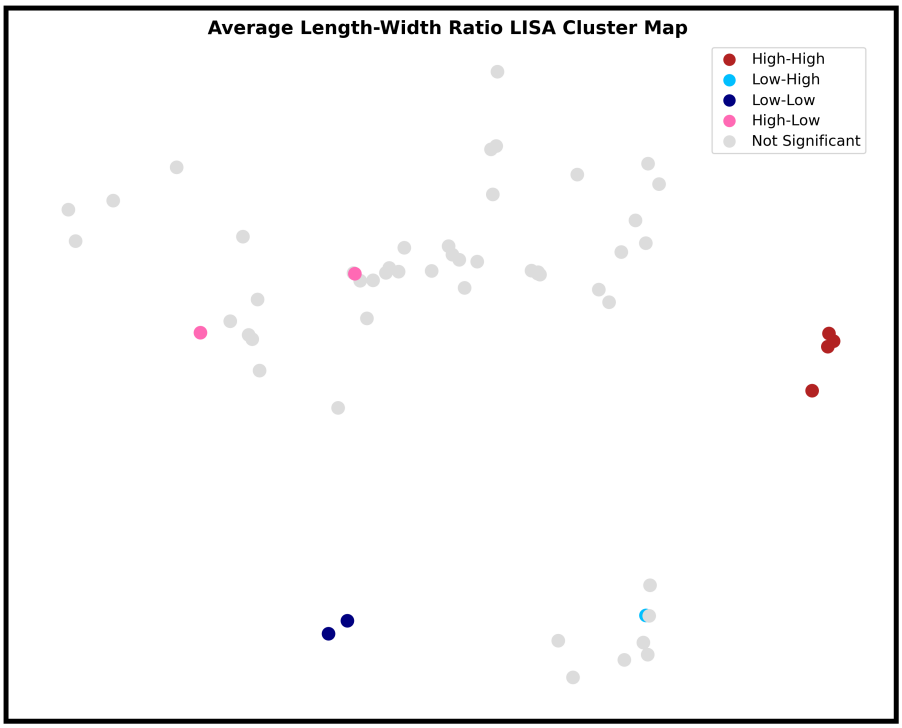

Figure 4.4: Local Indicators of Spatial Association (LISA) Cluster Map that identifies local spatial autocorrelation in the values of Average Length-Width Ratio. Each area is categorized based on the relationship between its value and the values of its neighbors. High-High(HH): corresponds to spatial clusters of high values, Low-Low (LL): corresponds to spatial clusters of low values, High-Low (HL): corresponds to spatial outliers, and Low-High (LH): corresponds to spatial outliers. Grey areas represent field locations with no statistically significant spatial autocorrelation at the significance level of 0.05.

K-Means Clustering

Spatial autocorrelation helped determine where similar values of average weight and length-to-width ratio occurred based on spatial proximity, but K-Means clustering helped determine why those spatial patterns exist by segmenting the data into zones with homogeneous field management, weather, and soil characteristics. The optimal number of K clusters determined by the elbow method is 3, as seen in Figure 4.7. Therefore, a new column, cluster_ID, was introduced to the dataset that contained the unique values, 0, 1, and 2. The percentage of data classified by each cluster is displayed in Figure 4.6. Figure 4.5 displays all fields and the latitude and longitude values colored by cluster_ID.

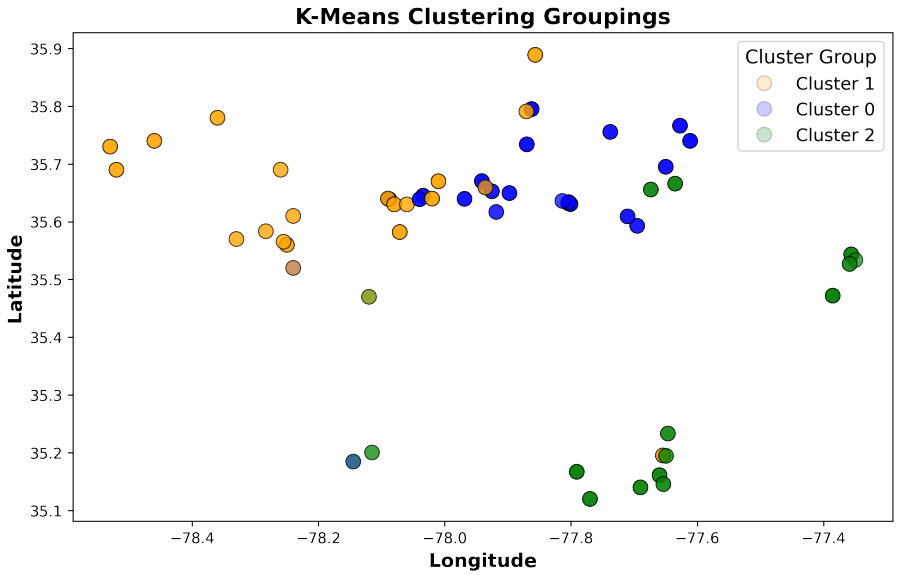

Figure 4.5: Spatial KMeans clustering results where K = 3 plotted over the latitude and longitude for all field locations over all field-years

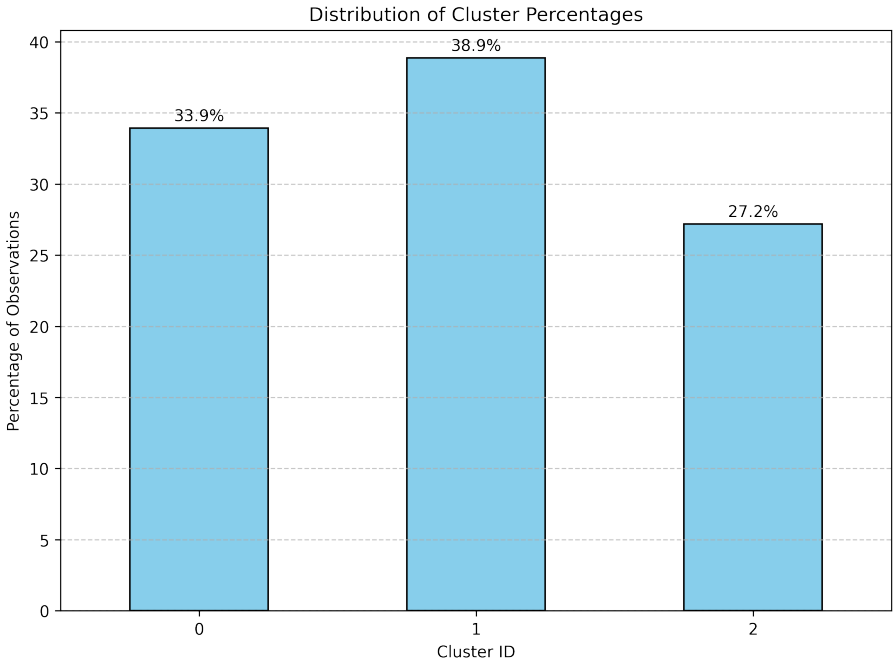

Figure 4.6: Distribution of data points across the identified Spatial KMeans clusters. Cluster 1 contains the highest proportion of observations (38.9%), followed by Cluster 0 (33.9%) and Cluster 2 (27.2%)

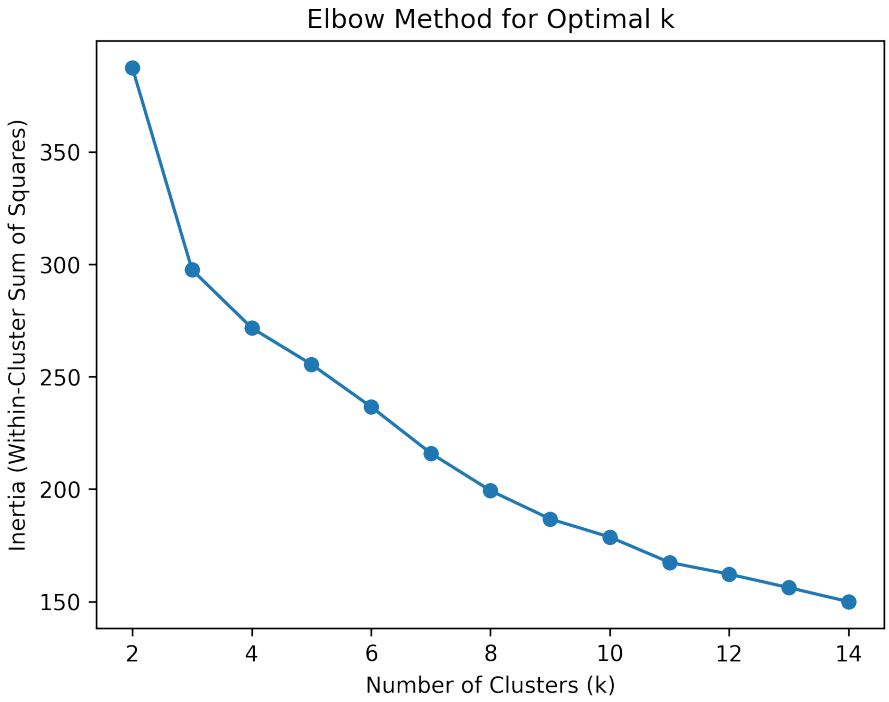

Figure 4.7: Elbow plot showing the relationship between the number of clusters (k) and the within-cluster sum of squares (inertia). The plot suggests an optimal k around 3–4, where the rate of decrease in inertia begins to level off.

4.3.5 Naive Baseline

Results for the naive baseline comparison can be found in tables 4.1 and 4.2. For both predicted variables, we see that the Test RMSE and MAE are lower than the Naive RMSE and MAE, and that Test R^2 is higher than Naive R^2, indicating that a more complex model that considers field management, environmental, and stationary features positively impacts model performance.

Target	Split	Model	Test RMSE	Test RMSE SD	Test MAE	Test MAE SD	Test R^2	Test R^2 SD
AW	Random	LR/RF/XGB	4.932	1.431	3.5	0.8349	0.1827	0.2502
AW	Group	LR/RF/XGB	4.795	1.616	3.408	1.132	0.1912	0.2667
AW	Random	LSTM	4.83	1.269	3.409	0.7338	0.202	0.2388
AW	Group	LSTM	14.11	7.367	12.01	7.667	-6.945	5.673

Table 4.1: Naive baseline results table for average weight prediction

Target	Split	Model	Test RMSE	Test RMSE SD	Test MAE	Test MAE SD	Test R^2	Test R^2 SD
ALWR	Random	LR/RF/XGB	0.5297	0.07056	0.411	0.05139	0.1722	0.2419
ALWR	Group	LR/RF/XGB	0.5373	0.04798	0.4135	0.04236	0.1494	0.1015
ALWR	Random	LSTM	0.6028	0.03324	0.4767	0.03248	-0.1136	0.37
ALWR	Group	LSTM	1.281	0.7856	1.077	0.7103	-5.124	7.064

Table 4.2: Naive baseline results table for average length-width ratio prediction

4.3.6 Model Performance

The randomly split models achieved lower RMSE and MAE values when compared to their grouped equivalents. All models across both targets and data splits perform better than the naive baseline models. The 5-fold and 3-fold average metrics for all models and data splits are presented in Tables 4.3 and 4.4. The percent change in performance between training and testing was calculated for all model metrics to assess how each model generalizes to unseen data and was included in each comparison table. When comparing RMSE, MAE,

and R^2 and predicting the average weight, the best-performing model is the RF using the random split. When considering the models' ability to generalize to unseen data (% percent change), the LR using the random split outperformed all models, followed by the grouped LR and random LSTM 4.3.

When predicting the average length-width ratio, the randomly split LR model exhibited the best model performance and the smallest gap between the model performance metrics. The randomly split RF and LSTM models did achieve similar model performance values but suffered from overfitting. Similarly, when considering RMSE percent change (rmse_pc), the LR using the random split outperformed all models, followed by the grouped LR and random LSTM 4.4.

Overall, the model results indicate that models using the random split consistently perform better than those using the grouped approach. The randomly split LR model has the smallest gap in model performance between the training and testing sets for both predicted variables. RF and XGB achieve better performance than LR when predicting average weight but suffer from some overfitting. When predicting average length-width ratio, the random RF achieves similar model performance results to the LR but suffers from overfitting more when compared to the RF. Given model performance, the randomly split RF and LR models were selected as the best models when predicting average weight and length-width ratio, respectively. Additionally, since the random RF had similar performance when predicting the average length-width ratio, this model was also included in the interpretation. Final models for both of the best-performing models for each predicted variable were trained on the full dataset without data partitions. The final model for predicting average weight was RF (RMSE: 2.756, MAE: 1.407, R^2: 0.7498), and when predicting average length-width ratio, LR (RMSE: 0.4140, MAE: 0.5654, R^2: 0.5294). Additionally, final model results for the random RF when predicting average length-width ratio (RMSE: 0.3657, MAE: 0.5316, R^2: 0.6328), which had higher model performance when compared to the final LR model, but the LR had higher overall performance during training.

Target	Split	Model	Train RMSE	Train RMSE SD	Train MAE	Train MAE SD	Train R^2	Train R^2 SD	Test RMSE	Test RMSE SD	Test MAE	Test MAE SD	Test R^2	Test R^2 SD	RMSE (% Change)	MAE (% Change)	R^2 (% Change)
AW	Random	RF	2.798	0.4144	2.013	0.2865	0.7404	0.0608	3.975	0.9995	2.909	0.5778	0.4683	0.121	42.08	44.46	-36.75
AW	Random	XGB	2.95	0.62	2.117	0.4613	0.7074	0.09417	4.037	1.028	2.932	0.614	0.4528	0.1241	36.86	38.48	-35.99
AW	Grouped	RF	2.784	0.4782	1.991	0.3508	0.7393	0.06963	4.096	1.147	2.97	0.7012	0.4046	0.1395	47.13	49.17	-45.28
AW	Grouped	XGB	3.312	0.4889	2.389	0.3736	0.6342	0.06198	4.136	1.225	2.963	0.7618	0.3943	0.158	24.91	24.05	-37.82
AW	Random	LR	4.085	0.2295	2.951	0.162	0.4498	0.02583	4.296	0.8113	3.173	0.5224	0.3752	0.0653	5.178	7.531	-16.59
AW	Random	LSTM	3.658	0.2281	2.625	0.155	0.5581	0.03471	4.315	0.7441	3.081	0.4981	0.3611	0.09112	17.96	17.39	-35.31
AW	Grouped	LR	3.975	0.5644	2.865	0.448	0.473	0.08386	4.579	1.355	3.472	0.8953	0.2611	0.1808	15.21	21.19	-44.8
AW	Grouped	LSTM	3.536	0.443	2.514	0.3381	0.5808	0.07711	5.72	0.5333	4.61	0.6876	-0.2861	0.6256	61.75	83.41	-149.3

Table 4.3: Cross-validation results comparison table for average weight prediction

Target	Split	Model	Train RMSE	Train RMSE SD	Train MAE	Train MAE SD	Train R^2	Train R^2 SD	Test RMSE	Test RMSE SD	Test MAE	Test MAE SD	Test R^2	Test R^2 SD	RMSE (% Change)	MAE (% Change)	R^2 (% Change)
ALWR	Random	LR	0.411	0.01286	0.3176	0.008507	0.5345	0.03448	0.4413	0.04353	0.3465	0.02386	0.4118	0.2269	7.381	9.106	-22.96
ALWR	Random	RF	0.3418	0.03617	0.262	0.02879	0.6761	0.06189	0.4453	0.05178	0.3418	0.03268	0.4192	0.1523	30.27	30.44	-38
ALWR	Grouped	LR	0.4092	0.009751	0.3161	0.01345	0.5334	0.02654	0.4545	0.008963	0.3516	0.01155	0.3824	0.1244	11.08	11.24	-28.3
ALWR	Random	XGB	0.3663	0.04843	0.2801	0.03937	0.6258	0.0915	0.456	0.04289	0.3539	0.02607	0.3927	0.1445	24.48	26.37	-37.25
ALWR	Random	LSTM	0.4071	0.01835	0.3142	0.01461	0.5431	0.03266	0.4607	0.05103	0.3579	0.03454	0.3686	0.1272	13.15	13.88	-32.14
ALWR	Grouped	RF	0.3305	0.007918	0.252	0.008283	0.694	0.0412	0.4726	0.02731	0.3708	0.01271	0.3346	0.1243	43.02	47.16	-51.8
ALWR	Grouped	XGB	0.3739	0.006886	0.2857	0.0114	0.609	0.04461	0.4731	0.02727	0.369	0.01693	0.3359	0.1037	26.53	29.16	-44.84
ALWR	Grouped	LSTM	0.4075	0.01404	0.3113	0.01482	0.5363	0.04598	0.5485	0.06237	0.4294	0.03447	0.1069	0.1945	34.58	37.96	-80.06

Table 4.4: Cross-validation results comparison table for average length-width ratio prediction

Feature Importance Analysis

Shapley Additive Explanations (SHAP values) were used to investigate feature importance and model interpretation using a combination of SHAP beeswarm feature importance and dependence plots. The SHAP beeswarm feature importance plots for the best-performing models for each predicted variable are shown in Figures 4.8 and 4.9, and dependence plots are shown in Figures 4.10, 4.11, and 4.12.

The top 5 most important features to consider when predicting average weight using the random RF are the average weight (avg_weight_oz_t2, avg_weight_oz$\widehat{1}$) from previous time steps, days after planting at the most recent timestep (DAP_t2), minimum relative humidity at the most recent timestep (min_rh_t2), and days after planting two time steps prior (DAP_t1) 4.8. The SHAP beeswarm plot 4.8, ranked average weight (avg_weight_oz_-t2) at timestep 2 as the top predictor for predicting average weight at the next time step. High values of average weight (avg_weight_oz_t2) at timestep 2 tend to increase average weight predictions, while low values of average weight (avg_weight_oz_t2) at timestep 2 tend to decrease average weight predictions. This relationship is also displayed in the SHAP dependence plots 4.10, where an increasing nonlinear relationship can be seen where the model switches from lower to higher predictions around an average weight (avg_-weight_oz_t2) value of 6 oz. This same relationship exists where high feature values increase average weight predictions for average weight (avg_weight_oz_t1) at timestep 1 and days after planting (DAP_t2, DAP_t1) at both time steps 1 and 2, while the opposite relationship exists for minimum relative humidity (min_rh_t2) at time step 2. Min to low values of minimum relative humidity (min_rh_t2) at time step 2 tend to increase average weight predictions, while high values tend to decrease average weight predictions. These results suggest that the combination of the most recent development and temporal conditions strongly influences average weight a week ahead.

When predicting average length-width ratio using the random LR, the top 5 features were days after planting (DAP_t2, DAP_t1) from previous time steps, average temperature (avg_temp_t2) at the most recent timestep, average length-width ratio (avg_lwratio_t2) from the most recent timestep, and average wind speed (avg_ws_t1) from the first timestep 4.9. Additionally, when predicting average length-width ratio using the random RF, the top 5 features were average length-width ratio (avg_lwratio_t2) from the most recent timestep, days after planting (DAP_t2) from the most recent timestep, (avg_lwratio_t1) from the first timestep, days after planting (DAP_t1) from the first time step, and cation exchange capacity at depths 0-5 cm (cec_0_5) 4.9. Shared features between the LR and RF when predicting

average length-width ratio are days after planting (DAP_t1, DAP_t2) at both time steps, and average length-width ratio (avg_lwratio_t2) at the most recent timestep indicating that both models prioritize the most recent shape and length of time the crop has been in the ground when making predictions for the next timestep. While the order of importance is different, when comparing the top 5 features for both LR and RF models, they display similar relationships for all variables except for days after planting (DAP_t1) at time step 1. The top predictor for the LR model is days after planting (DAP_t2) at time step 2 4.9, where low values of this feature positively impact average length-width ratio predictions and higher feature values negatively impact predictions 4.11. Suggesting that sweetpotatoes that have had less time to develop by time step 2 have longer roots, while sweetpotatoes that have had more time to develop by time step 2 are more round. This relationship is also present for the RF but appears as the second most important. The opposite of this relationship for days after planting (DAP_t2) exists when predicting average weight. Next, days after planting (DAP_t1) for LR suggests that high values of this feature positively impact average length-width ratio predictions, and lower feature values negatively impact predictions, which is the opposite of the relationship displayed by days after planting (DAP_t2) 4.11. The RF shows the opposite of this relationship for days after planting (DAP_t1); the decreasing trend for days after planting (DAP_t1) can be seen in Figure 4.12, where lower values of days after planting (DAP_t1) increase the average weight prediction. Similar to days after planting (DAP_t2), average temperature (avg_temp_t2) at time step 2 indicates that lower values tend to increase average length width ratio predictions, while high values decrease predictions. This suggests that locations with a lower average temperature by time step 2 tend to result in more elongated roots, while locations with higher average temperature produce more rounded roots. Although the importance of this variable falls outside the top 5 for the RF, the relationship is the same. The final two features for LR model are average length-width ratio (avg_lwratio_t2) from the most recent timestep and the average wind speed (avg_ws_t1) from the first timestep show a similar relationship where higher values increase the average-length width ratio prediction, while lower values decrease the prediction. Suggesting that if sweetpotatoes are more elongated at timestep 2, they tend to be elongated at the next time step, and locations with higher wind speeds at time step 1 tend to have a higher average-length width ratios. The RF also shows this same relationship 4.12. Two variables included in the top 5 RF model that were not included in the LR are average-length width ratio (avg_lwratio_t1) from the first timestep and cation exchange capacity at depths 0-5 cm (cec_0_5). Both variables follow the same relationship as average length-width ratio (avg_lwratio_t2) where larger values tend to increase prediction. When considering cation

exchange capacity, this suggests that field locations with a higher capacity to retain essential nutrients positively impact average length-width ratio predictions, while field locations with lower capacities decrease average weight.

Overall, both models display similar relationships for all features except days after planting (DAP_t1) and seem to prioritize higher average length-width ratio values to increase predictions rather than smaller values. The LR model prioritizes a combination of field management and temporal data to make predictions, while the RF prioritizes a combination of recent development, field management, and stationary soil data to make predictions. Given that the RF SHAP dependence plots display non-linear relationships for many of the variables and since the LR does not account for interactions that may exist between feature values, the RF may be the superior model between the two. These results suggest that the combination of the most recent development, field management, and environmental conditions strongly influences the average length-width ratio a week ahead.

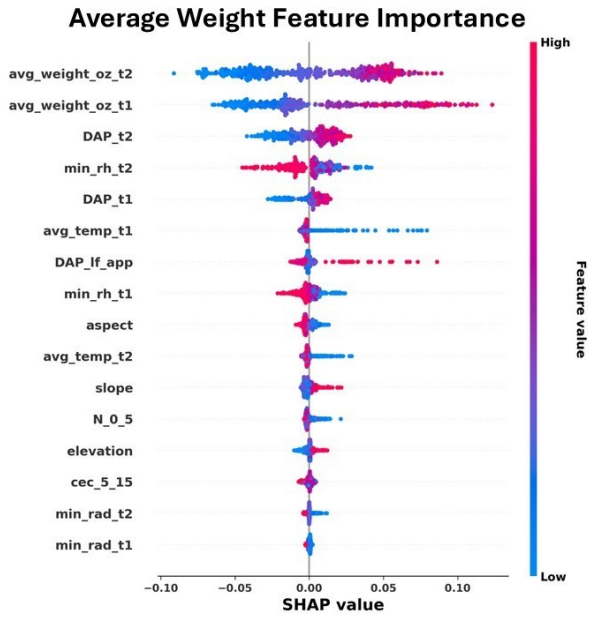

Figure 4.8: Important features for determining sweetpotato weight using random forest and random splits.

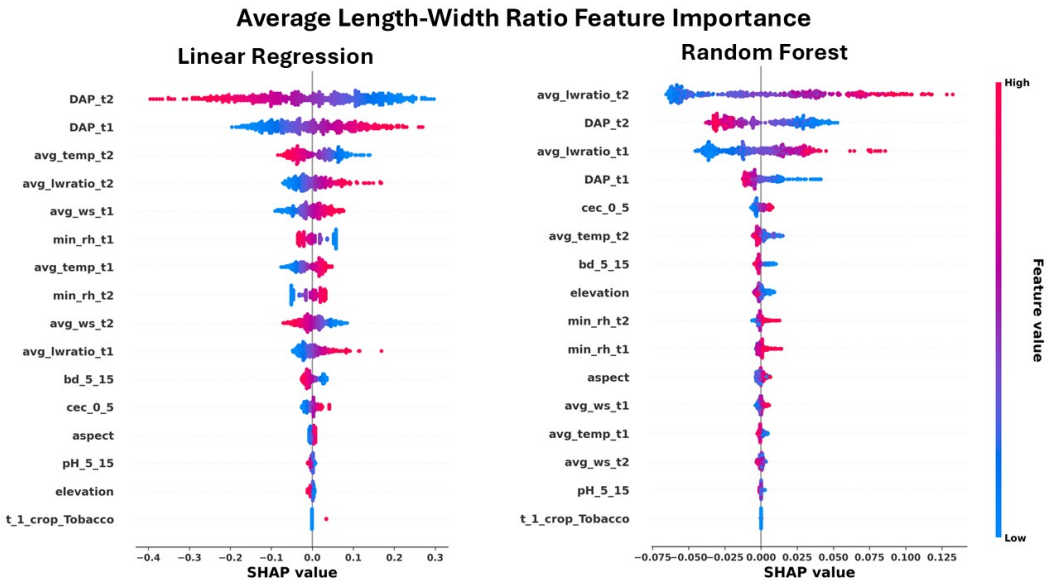

Figure 4.9: Important features for determining sweetpotato average length-width ratio using linear regression and random forest with random splits.

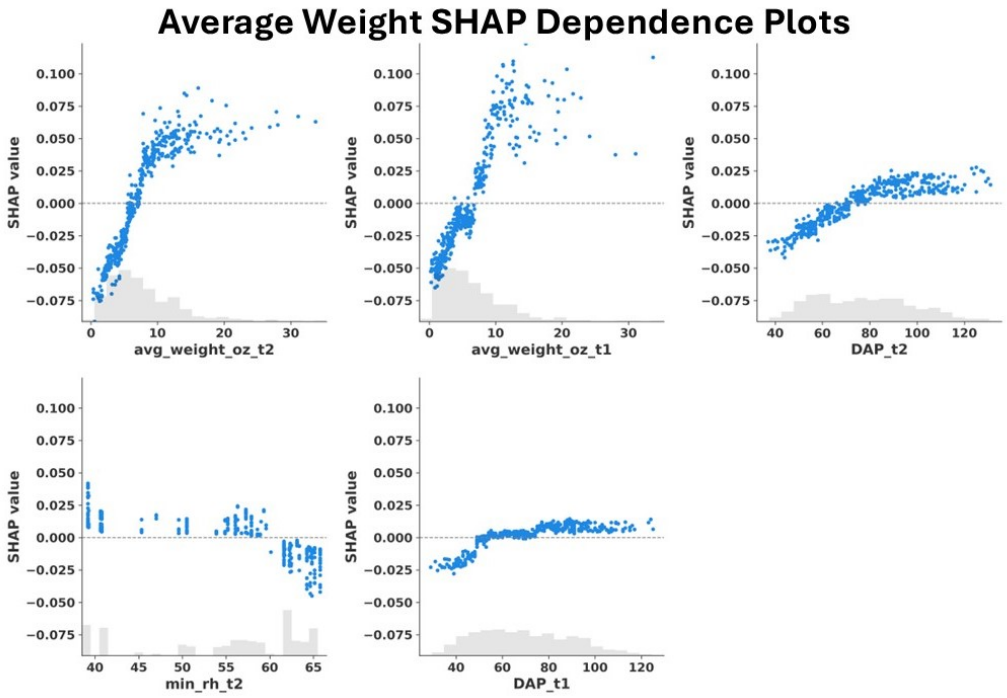

Figure 4.10: SHAP dependence plots for the top 5 features for predicting average weight using a random forest model with random splitting. Prediction values above $y = 0$ increase predictions, and values below decrease predictions.

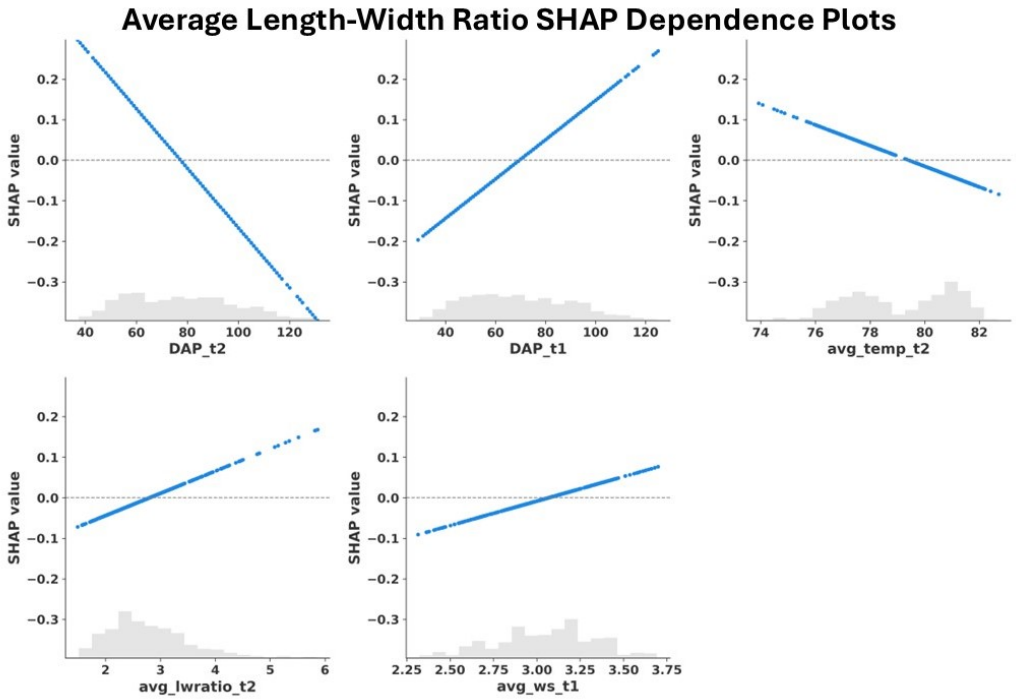

Figure 4.11: SHAP dependence plots for the top 5 features for predicting average length-width ratio using a random forest model with random splitting. Prediction values above $y = 0$ increase predictions, and values below decrease predictions.

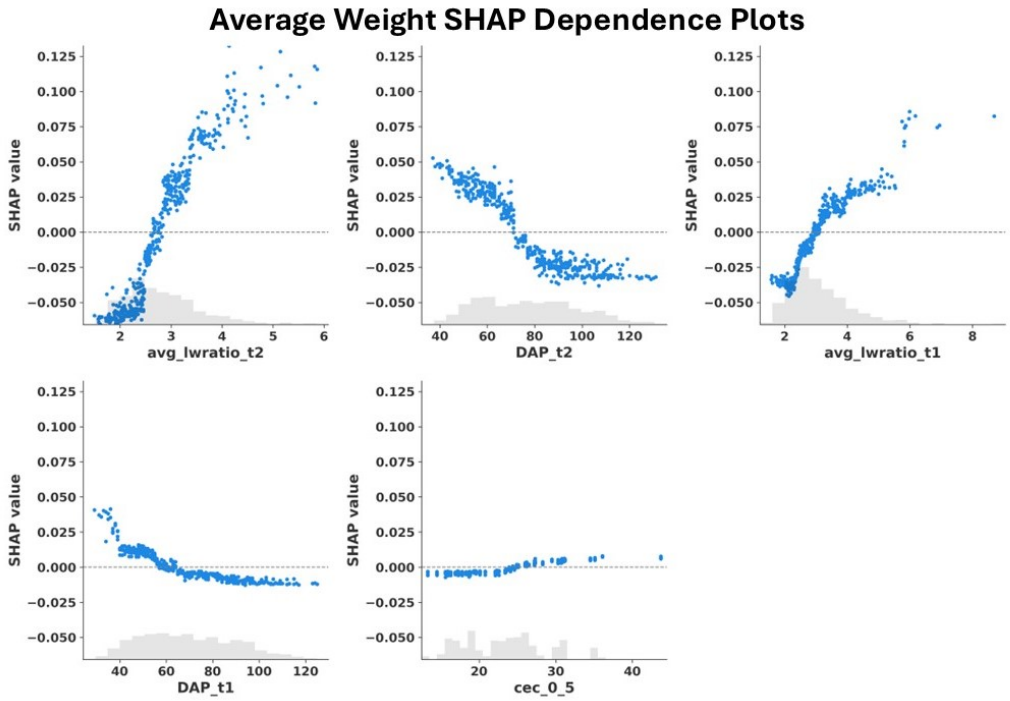

Figure 4.12: SHAP dependence plots for the top 5 features for predicting average length-width ratio using a random forest model with random splitting.Prediction values above $y = 0$ increase predictions, and values below decrease predictions.

4.3.7 Discussion

Our results show that static models achieve higher model performance values when compared to dynamic models when used to predict sweetpotato shape metrics. While the size of the data set does impact how the model performs, these results suggest that if you have a smaller dataset, model performances between less static models and dynamic models are similar. The LSTM models, both random and grouped, achieved some of the lowest RMSE, MAE, and R^2. Using random splitting led to higher model performances, with LR and RF models having achieved the highest performance when predicting average weight and length-width ratio, respectively. Results also suggest that a combination of the most recent crop development, field management, and temporal data consistently increases average weight predictions, while the most recent crop development, field management, and soil data consistently increase average length-width ratio predictions.

Comparison of Results with Existing Literature

Chapter 1 revealed that while the average length-width ratio had been predicted previously using machine learning models (3) (RMSE:0.6195 and R^2:0.58), the average weight of an individual crop is not often predicted; rather, yield is most commonly predicted. Currently, there exist no historical prediction values to compare our average weight values to, while (3) reported ranges for length-width ratio, but our data represent the entire length of the growing season rather than at the time of harvest (1.49–8.69 vs 1.4–2.6). Therefore, this work included a naive baseline for both predicted variables that compares the RMSE, MAE, and R^2 obtained by all models to a simpler model that assumes the predicted variable at the next time step is equal to the previous timestep. While this baseline is common in forecasting and prediction tasks, this baseline is essentially assuming that crop development is constant over time and will likely result in an unrealistic comparison for crops whose growth does not plateau over time. When compared to the naive baseline, the random RF used to predict average weight reduces RMSE by 19% and MAE by 17%, and increases R^2 by 156% when predicting average length-width ratio, the random LR reduces RMSE by 17% and MAE by 16%, and increases R^2 by 139% and the random RF reduces RMSE by 16% and MAE by 17%, and increases R^2 by 142%.

When comparing the model performance results of this work to the previous work on (25), the best-performing model when predicting average weight previously achieved (RMSE:5.6354, MAE:3.1602, R^2:0.3668), current predictions (RMSE:3.975, MAE:2.909, R^2:0.4683) show a 29% decrease in RMSE and a 8% decrease in MAE, and 28% increase in R^2. For the average length-width ratio the previous model achieved (RMSE:0.6195, MAE:0.4533, R^2:0.5766), current predictions using LR achieved (RMSE:0.4413, MAE:0.3465, R^2:0.4118) show a 29% decrease in RMSE, 24% decrease in MAE, and a 29% decrease in R^2 and current predictions using RF achieved (RMSE:0.4453, MAE:0.3418, R^2:0.4192) show a 28% decrease in RMSE, 24% decrease in MAE, and a 27% decrease in R^2. It is worth noting that while it is not ideal for the explained variance to decrease, the previous work predicted shape metrics considering the whole season of data. At the same time, this study limits the data the models see to a snapshot of roughly two weeks at a time.

Practical Applications: Use Case For Growers

Using the best-performing models presented herein, growers can predict crop development during the growing season 7-10 days ahead if they have field coordinates, weather, soil, field management variables, and crop development data for a specific field 14-20 days prior.

Figures 4.13, 4.14 compare observed and predicted values for both targets using unseen data. For average weight, the model achieved RMSE:4.2328, MAE:1.7309, and R^2:0.4330, indicating that 43% of the variance in average weight in 7-10 days can be explained by the model. The RMSE value suggests an average prediction error of ±4.23 ounces for three sweetpotato plants. Figure 4.15 reveals that the model predictions and observed values closely align up to 75 days after planting; however, the accuracy between them decreases as days after planting increase. For average LWR, the LR model achieved an RMSE = 0.2622, MAE = 0.4479, and $R^2 = 0.5192$, explaining 52% of the variance. The RMSE indicates an average error of ±0.25 in LWR predictions for three sweetpotato plants. Figure 4.16 displays the opposite relationships of average weight, as the model predictions do not begin to align until around 60 days after planting. Similarly, the RF model achieved an RMSE:0.3046, MAE:0.4764, and R^2:0.3512, explaining 35% of the variance. The RMSE indicates an average error of ±0.35 in LWR predictions for three sweetpotato plants. Figure 4.16 highlights a similar relationship as the results shown above for the linear regression, where the model predictions improve as days after planting increase.

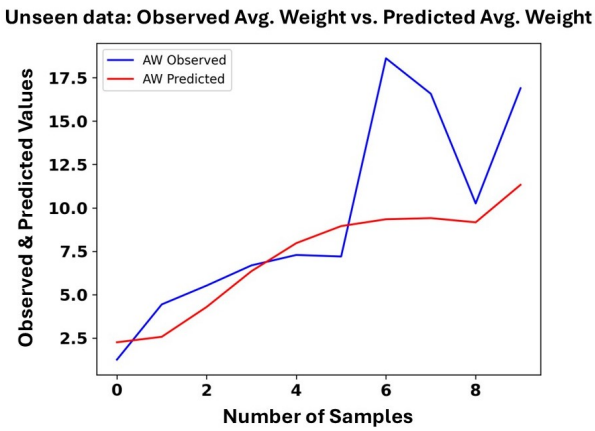

Figure 4.13: Predicted vs. observed validation plot when predicting average weight using the random split random forest regression.

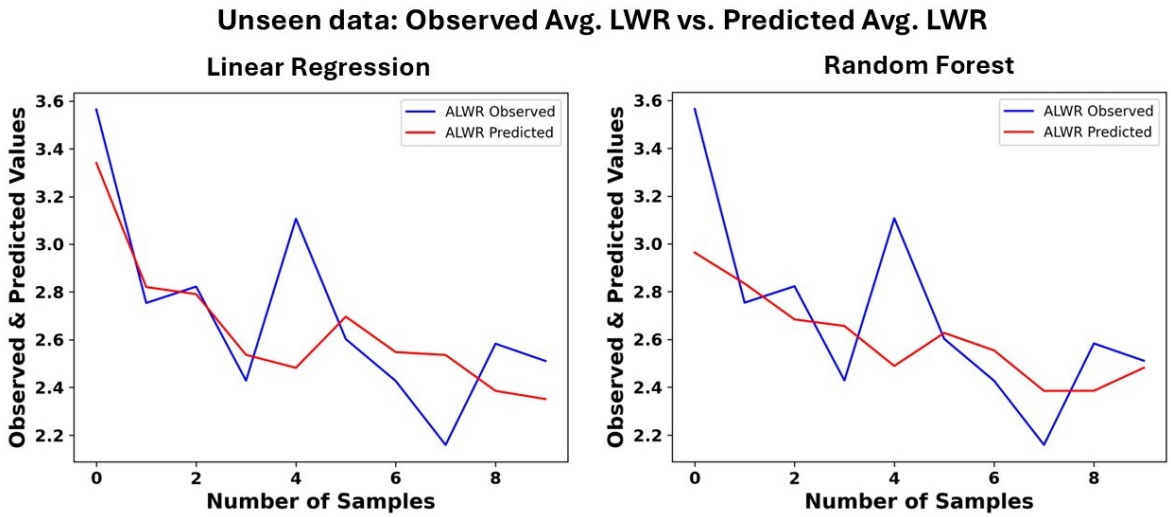

Figure 4.14: Predicted vs. observed validation plot when predicting average length-width ratio using linear regression and random forest with the random split.

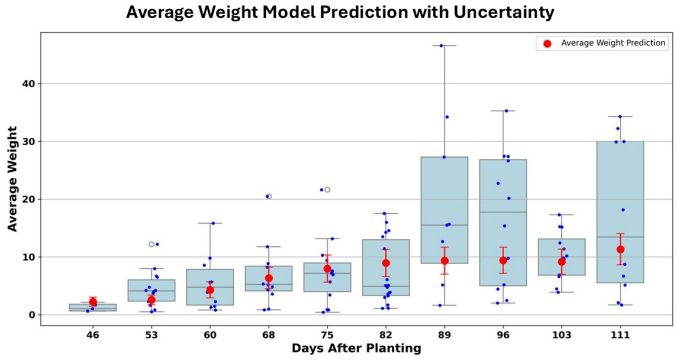

Figure 4.15: Comparison of the observed weight and average weight values superimposed with the average weight predictions for one holdout field.

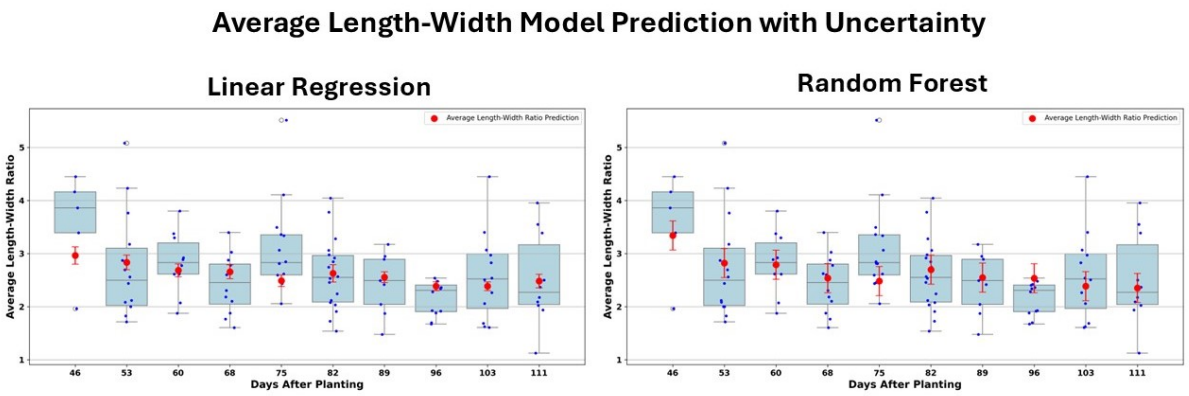

Figure 4.16: Comparison of the observed length-width ratio and average length-width ratio values superimposed with the average length-width ratio predictions for one holdout field.

Limitations

Some limitations or areas of improvement for this study are the refinement of the days between data collection, the number of previous time points used for average weight and length-width ratio predictions, and a deeper exploration of the point in time that each timestep represents to accurately pad sequences. Data collection intervals vary with the min, max, and average days between data collection being 2, 22, and 7.9, respectively 4.17. The decision to predict sweetpotato attributes a week ahead, given the previous two weeks of data, came from the fact that image data was collected on average every 7-10 days. Many agricultural decisions, such as fertilizer and insecticide applications, and harvesting, are made weekly, so this average simulated a consistent temporal pattern for decision making but could be refined by filling in gaps using interpolation or smoothing using aggregation to represent the realistic time between each timestep. Predictions are made using the previous two time steps, but comparing the performance of using more than 2 time steps could benefit model performance by allowing the models access to more past information at each timestep. It may also be beneficial to subset the data to make this prediction beginning at 65 DAP, which is the beginning of the growth phase 3, where more consistent patterns of growth can be observed. Lastly, it's important to examine how the sequences align to days after planting. While two sequences may be of similar lengths, they may not represent the same developmental stage. As a result, only post-padded sequences may allow one sequence to begin at a later point in development. In this case, the models are seeing both being the beginning timestep, which can negatively impact a shape prediction overall. A

combination of pre- and post-padding may be beneficial to improve model performance.

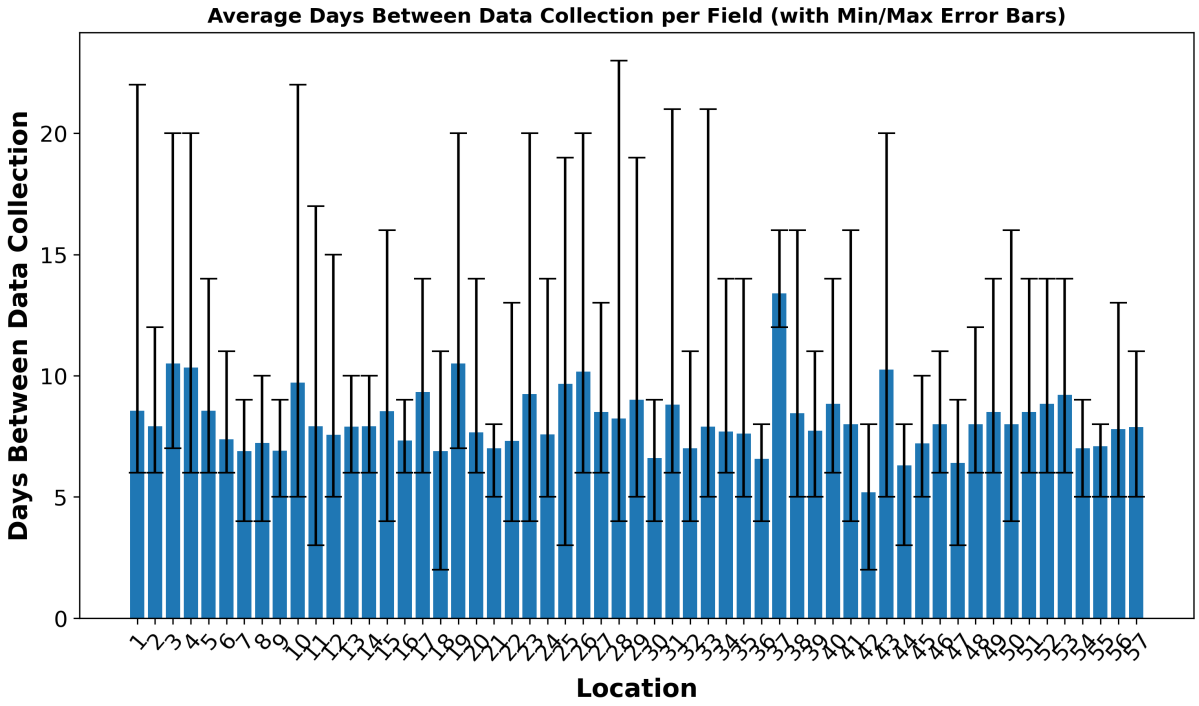

Figure 4.17: Barplots displaying the average number of days between data collection for each field location (blue) with the minimum and maximum values displayed and error bars (black)

CHAPTER

5

CONCLUSIONS

This work demonstrates that machine learning (ML) models can predict sweetpotato shape attributes during the growing season using key farming, temporal, and stationary data. Features such as days after planting, minimum relative humidity, average temperature, fertilizer application, and soil properties were identified as critical for predicting both average weight and length-width ratio (LWR) in a global sense, as well as when making predictions at a week ahead, and should be incorporated into on-farm decision-making processes. Additionally, we presented a use-case scenario to illustrate the practical application of these models, making the results more interpretable and actionable for farmers. This work also provides tools and guidelines for working with agricultural data that can be used to quickly analyze crop development.

5.1 Future Work

The time between the collection of data samples was generally between 7-10 days, but could be more. The limited data size presented a challenge in this research, as machine learning models prefer large amounts of data to learn from. The data in this research has roughly

577 grouped observations. Future work will address these limitations by using Generative Adversarial Networks (GAN) to augment the dataset and assess how this addition will impact the model performance of the LSTM in particular. Introducing a model that predicts multiple targets, such as average weight and length-width ratio, to investigate their effects and features of importance closer to the harvest time.

REFERENCES

[1] H. M. Nguyen, S. Gyurek, R. Mierop, K. V. Pecota, K. LaGamba, M. Boyette, G. C. Yencho, C. M. Williams, M. W. Kudenov, Deployment and analysis of instance segmentation algorithm for in-field grade estimation of sweetpotatoes, arXiv preprint arXiv:2308.08534 (2023).

[2] S. Haque, E. Lobaton, N. Nelson, G. C. Yencho, K. V. Pecota, R. Mierop, M. W. Kudenov, M. Boyette, C. M. Williams, Computer vision approach to characterize size and shape phenotypes of horticultural crops using high-throughput imagery, Computers and Electronics in Agriculture 182 (2021) 106011. doi:10.1016/j.compag.2021.106011.
URL https://www.sciencedirect.com/science/article/pii/S0168169921000296

[3] C. Y. K. P. R. M. C. W. D. J. Shelly Hunt, Hangjin Liu, Predicting crop shape-related features using agricultural inputs, Computers and Electronics in Agriculture Journal (2024).
URL https://papers.ssrn.com/sol3/papers.cfm?abstract_id=4732363

[4] E. E. P. Martinez, M. Kudenov, H. Nguyen, D. S. Jones, C. Williams, Evaluating two high-throughput phenotyping platforms at early stages of the post-harvest pipeline of sweetpotatoes, Smart Agricultural Technology (2024) 100469.

[5] C. W. Mariella Carbajal-Carrasco, Daniela Jones, N. Nelson, In-season sweetpotato yield forecasting using multitemporal remote sensing environmental observations and machine learning (2024).
URL https://d197for5662m48.cloudfront.net/documents/publicationstatus/205960/preprint_pdf/1d4906e6a908b1764d15327297fbb0f3.pdf

[6] M. D. Boyette, A. L. Tsirnikas, Evaluating the Shape and Size Characteristics of Sweetpotatoes Using Digital Image AnalysisMAG ID: 2765227104 (Jan. 2017). doi:10.13031/aim.201700038.

[7] S. T. Hingston, T. J. Noseworthy, On the epidemic of food waste: Idealized prototypes and the aversion to misshapen fruits and vegetables, Food Quality and Preference 86 (2020) 103999. doi:10.1016/j.foodqual.2020.103999.
URL https://www.sciencedirect.com/science/article/pii/S0950329320302688

[8] T. Taşkıner, B. Bilgen, Optimization Models for Harvest and Production Planning in Agri-Food Supply Chain: A Systematic Review, Logistics 5 (3) (2021) 52, number: 3 Publisher: Multidisciplinary Digital Publishing Institute. doi:10.3390/logistics5030052.
URL https://www.mdpi.com/2305-6290/5/3/52

[9] L. K. Johnson, J. D. Bloom, R. D. Dunning, C. C. Gunter, M. D. Boyette, N. G. Creamer, Farmer harvest decisions and vegetable loss in primary production, Agricultural Systems 176 (2019) 102672. doi:10.1016/j.agsy.2019.102672.
URL https://www.sciencedirect.com/science/article/pii/S0308521X18311387

[10] L. K. Johnson, R. D. Dunning, C. C. Gunter, J. Dara Bloom, M. D. Boyette, N. G. Creamer, Field measurement in vegetable crops indicates need for reevaluation of on-farm food loss estimates in North America, Agricultural Systems 167 (2018) 136–142. doi:10.1016/j.agsy.2018.09.008.
URL https://www.sciencedirect.com/science/article/pii/S0308521X18305687

[11] Sweetpotatoes grades and standards, https://www.ams.usda.gov/grades-standards/sweetpotatoes-grades-and-standards.

[12] R. A. Arancibia, C. D. Smith, D. R. LaBonte, J. L. Main, T. P. Smith, A. Q. Villordon, Optimizing sweetpotato production for fresh and processing markets through plant spacing and planting-harvest time, HortTechnology 24 (1) (2014) 16–24.

[13] D. Wallach, D. Makowski, J. W. Jones, F. Brun, Working with dynamic crop models: methods, tools and examples for agriculture and environment, Academic Press, 2018.

[14] I. Droutsas, A. Challinor, C. Deva, E. Wang, Integration of machine learning into process-based modeling to improve simulation of complex crop responses, in silico Plants (2022). doi:10.1093/insilicoplants/diac017.

[15] T. Klompenburg, A. Kassahun, C. Catal, Crop yield prediction using machine learning: A systematic literature review, Computers and Electronics in Agriculture 177 (2020) 105709. doi:10.1016/j.compag.2020.105709.

[16] P. D. C. Sanchez, N. Hashim, R. Shamsudin, M. Z. M. Nor, Applications of imaging and spectroscopy techniques for non-destructive quality evaluation of potatoes and sweet potatoes: A review, Trends in Food Science & Technology 96 (2020) 208–221.

[17] D. Tedesco, M. F. de Oliveira, A. F. dos Santos, E. H. Costa Silva, G. de Souza Rolim, R. P. da Silva, Use of remote sensing to characterize the phenological development and to predict sweet potato yield in two growing seasons, European Journal of Agronomy 129 (2021) 126337. doi:10.1016/j.eja.2021.126337.
URL https://www.sciencedirect.com/science/article/pii/S1161030121001088

[18] A. Villordon, C. Clark, D. Ferrin, D. LaBonte, Using growing degree days, agrometeorological variables, linear regression, and data mining methods to help improve prediction of sweetpotato harvest date in louisiana, HortTechnology 19 (2009) 133–144. doi:10.21273/HORTSCI.19.1.133.

[19] J. S. D. L. Villordon, Arthur, C. Clark, Development of a prototype bayesian network model representing the relationship between fresh market yield and some agroclimatic variables known to influence storage root initiation in sweetpotato, HortScience 45, no. 8 (2010): 1167-1177 (2010).

[20] M. E. Wright, J. H. Tappan, F. E. Sistler, The Size and Shape of Typical Sweet Potatoes, Transactions of the ASABE 29 (3) (1986) 0678–0682, mAG ID: 2330314609. doi:10.13031/2013.30212.

[21] A. Villordon, H. W. Carroll, Digital Image Analysis of Sweetpotato Storage Roots in Herbicide Trials, Hortscience 37 (4) (2002) 669–670, mAG ID: 2184535675. doi:10.21273/hortsci.37.4.669.

[22] A. Villordon, J. C. Gregorie, D. LaBonte, Direct Measurement of Sweetpotato Surface Area and Volume Using a Low-cost 3D Scanner for Identification of Shape Features Related to Processing Product Recovery, HortScience 55 (5) (2020) 722–728, publisher: American Society for Horticultural Science Section: HortScience. doi:10.21273/HORTSCI14964-20.
URL https://journals.ashs.org/hortsci/view/journals/hortsci/55/5/article-p722.xml

[23] S. Gogineni, J. G. White, J. Thomasson, P. Thompson, J. Wooten, M. Shankle, Image-based sweetpotato yield and grade monitor (2002) 1doi:10.13031/2013.10586.

[24] Exeter Engineering, Exeter engineering, https://www.exeterengineering.com/, accessed: April 2, 2024 (2023).

[25] S. McDowell, D. Jones, S. Carpenter, S. Hunt, M. Kudenov, C. M. Williams, Machine learning for predicting sweetpotato growth and quality: Integrating in-season root imaging with environmental and agronomic insights, Available at SSRN 5108983 (2025).

[26] S. J. Kays, Preharvest factors affecting appearance, Postharvest biology and technology 15 (3) (1999) 233–247.

[27] S. Haque, E. Lobaton, N. Nelson, G. C. Yencho, K. V. Pecota, R. Mierop, M. W. Kudenov, M. Boyette, C. M. Williams, Computer vision approach to characterize size and shape phenotypes of horticultural crops using high-throughput imagery, Computers and Electronics in Agriculture 182 (2021) 106011. doi:https://doi.org/10.1016/j.compag.2021.106011.
URL https://www.sciencedirect.com/science/article/pii/S0168169921000296

[28] J. Ping, C. Green, R. Zartman, K. Bronson, Exploring spatial dependence of cotton yield using global and local autocorrelation statistics, Field Crops Research 89 (2-3) (2004) 219–236.

[29] T. M. Koutsos, G. C. Menexes, A. P. Mamolos, The use of crop yield autocorrelation data as a sustainable approach to adjust agronomic inputs, Sustainability 13 (4) (2021) 2362.

[30] K. Kopczewska, Spatial machine learning: new opportunities for regional science, The Annals of Regional Science 68 (3) (2022) 713–755.

[31] A. M. Ikotun, A. E. Ezugwu, L. Abualigah, B. Abuhaija, J. Heming, K-means clustering algorithms: A comprehensive review, variants analysis, and advances in the era of big data, Information Sciences 622 (2023) 178–210.

[32] John Deere, John deere operations center, accessed: 2024-06-07 (2024). URL https://www.deere.com/en/technology-products/precision-ag-technology/data-management/operations-center/

[33] NASA Langley Research Center, The power project, https://power.larc.nasa.gov/, retrieved from: [2021-2023].

[34] L. Poggio, L. M. de Sousa, N. H. Batjes, G. B. M. Heuvelink, B. Kempen, E. Ribeiro, Soilgrids 2.0: producing soil information for the globe with quantified spatial uncertainty, SOIL (2021).

[35] Natural Resources Conservation Service, Map unit description by soil survey area and major components, https://www.nrcs.usda.gov/, accessed: January 11, 2024.

[36] A. Jarvis, E. Guevara, H. Reuter, A. Nelson, Hole-filled srtm for the globe : version 4 : data grid, published by CGIAR-CSI on 19 August 2008. (2008).

[37] K. He, G. Gkioxari, P. Dollar, R. Girshick, Mask R-CNN, pp. 2961–2969. URL https://openaccess.thecvf.com/content_iccv_2017/html/He_Mask_R-CNN_ICCV_2017_paper.html

[38] P. development team. (2023), pandas.get dummies pandas 2.2.3 documentation., https://pandas.pydata.org/docs/reference/api/pandas.get_dummies.html, accessed: November 20, 2024.

[39] A.-N. Spiess, N. Neumeyer, An evaluation of r2 as an inadequate measure for nonlinear models in pharmacological and biochemical research: a monte carlo approach, BMC pharmacology 10 (1) (2010) 1–11.

[40] J. Madhuri, M. Indiramma, Hybrid filter and wrapper methods based feature selection for crop recommendation, in: 2022 International Conference on Electronic Systems and Intelligent Computing (ICESIC), IEEE, 2022, pp. 247–252.

[41] S. P. Raja, B. Sawicka, Z. Stamenkovic, G. Mariammal, Crop Prediction Based on Characteristics of the Agricultural Environment Using Various Feature Selection Techniques and Classifiers, IEEE Access 10 (2022) 23625–23641. doi:10.1109/ACCESS.2022.

3154350.
URL https://ieeexplore.ieee.org/abstract/document/9721191

[42] D. Berrar, et al., Cross-validation. (2019).

[43] Bulk density measurement, accessed on 2024-11-22.
URL https://www.soilquality.org.au/factsheets/bulk-density-measurement

[44] L. Gatiboni, Soils and Plant Nutrients, Chapter 1, 2nd Edition, NC State Extension, Raleigh, NC, 2022, accessed on 2024-11-22.
URL https://content.ces.ncsu.edu/extension-gardener-handbook/1-soils-and-plant-nutrients

[45] P. Refaeilzadeh, L. Tang, H. Liu, Cross-validation, in: Encyclopedia of database systems, Springer, 2009, pp. 532–538.

[46] A. Seraj, M. Mohammadi-Khanaposhtani, R. Daneshfar, M. Naseri, M. Esmaeili, A. Baghban, S. Habibzadeh, S. Eslamian, Cross-validation, in: Handbook of Hydroinformatics, Elsevier, 2023, pp. 89–105.

[47] S. learn developers, Cross-validation: evaluating estimator performance, https://scikit-learn.org/stable/modules/cross_validation.html, accessed: 2025-06-24 (n.d.).

[48] C. K. Enders, Applied missing data analysis, Guilford Publications, 2022.

[49] D. A. Bennett, How can i deal with missing data in my study?, Australian and New Zealand journal of public health 25 (5) (2001) 464–469.

[50] I. Tsamardinos, A. Rakhshani, V. Lagani, Performance-estimation properties of cross-validation-based protocols with simultaneous hyper-parameter optimization, International Journal on Artificial Intelligence Tools 24 (05) (2015) 1540023.

[51] A. Vabalas, E. Gowen, E. Poliakoff, A. J. Casson, Machine learning algorithm validation with a limited sample size, PloS one 14 (11) (2019) e0224365.

[52] S. Varma, R. Simon, Bias in error estimation when using cross-validation for model selection, BMC bioinformatics 7 (2006) 1–8.

[53] D. Krstajic, L. J. Buturovic, D. E. Leahy, S. Thomas, Cross-validation pitfalls when selecting and assessing regression and classification models, Journal of cheminformatics 6 (2014) 1–15.

[54] A. H. Victoria, G. Maragatham, Automatic tuning of hyperparameters using bayesian optimization, Evolving Systems 12 (1) (2021) 217–223.

[55] A. Smiti, A critical overview of outlier detection methods, Computer Science Review 38 (2020) 100306.

[56] A. Jadhav, D. Dhaulakhandi, S. K. Shandilya, L. Malviya, A. Mewada, Data transformation: A preprocessing stage in machine learning regression problems, in: Artificial Intelligence Techniques in Power Systems Operations and Analysis, Auerbach Publications, 2023, pp. 183–194.

[57] F. Kamalov, S. Moussa, J. A. Reyes, Data transformation in machine learning: empirical analysis, in: 2023 International Conference on Innovation and Intelligence for Informatics, Computing, and Technologies (3ICT), IEEE, 2023, pp. 115–120.

[58] J. Li, K. Cheng, S. Wang, F. Morstatter, R. P. Trevino, J. Tang, H. Liu, Feature selection: A data perspective, ACM computing surveys (CSUR) 50 (6) (2017) 1–45.

[59] J. Cai, J. Luo, S. Wang, S. Yang, Feature selection in machine learning: A new perspective, Neurocomputing 300 (2018) 70–79.

[60] D. Elavarasan, D. R. Vincent PM, K. Srinivasan, C.-Y. Chang, A hybrid cfs filter and rf-rfe wrapper-based feature extraction for enhanced agricultural crop yield prediction modeling, Agriculture 10 (9) (2020) 400.

[61] I. S. Thaseen, C. A. Kumar, Intrusion detection model using fusion of chi-square feature selection and multi class svm, Journal of King Saud University-Computer and Information Sciences 29 (4) (2017) 462–472.

[62] M. B. Kursa, W. R. Rudnicki, Feature selection with the boruta package, Journal of statistical software 36 (2010) 1–13.

[63] V. Fonti, E. Belitser, Feature selection using lasso, VU Amsterdam research paper in business analytics 30 (2017) 1–25.

[64] D. Chicco, M. J. Warrens, G. Jurman, The coefficient of determination r-squared is more informative than smape, mae, mape, mse and rmse in regression analysis evaluation, Peerj computer science 7 (2021) e623.

[65] L. Li, S. Dai, Z. Cao, Deep long short-term memory (lstm) network with sliding-window approach in urban thermal analysis, in: 2019 IEEE/CIC International Conference on Communications Workshops in China (ICCC Workshops), 2019, pp. 222–227. doi:10.1109/ICCChinaW.2019.8849965.

[66] U. Bhimavarapu, G. Battineni, N. Chintalapudi, Improved optimization algorithm in lstm to predict crop yield. computers 12 (1), 10 (2023).

[67] J. Sun, L. Di, Z. Sun, Y. Shen, Z. Lai, County-level soybean yield prediction using deep cnn-lstm model, Sensors 19 (20) (2019) 4363.

[68] R. J. Hyndman, G. Athanasopoulos, Forecasting: principles and practice, OTexts, 2018.

APPENDIX

APPENDIX

A

CHAPTER 2 SUPPLEMENTAL INFORMATION

newpage

A.1 Full Model Performance Results

A.2 Model Performance Results when CBFS relevance threshold is .7

Table A.1: 5-fold cross-validation Results comparison table for average weight prediction with relevance threshold = 0.7

RMSE mean	RMSE std	MAE mean	MAE std	R^2 mean	R^2 std	Algorithm	Data split
5.5790	1.6103	3.1171	0.3886	0.3796	0.1448	RF	random
5.6883	1.3506	3.1242	0.2631	0.3484	0.1059	RF	field
5.8584	1.8667	3.1603	0.5830	0.3197	0.1965	SVM	random
5.9078	1.2736	3.3118	0.2001	0.2921	0.0991	GB	field
5.9604	1.6270	3.5710	0.4695	0.2910	0.1384	LR	random
6.0331	1.5059	3.1887	0.3043	0.2657	0.1551	SVM	field
6.0571	1.6624	3.5246	0.4557	0.2635	0.1645	GB	random
6.0743	1.4127	3.6968	0.3598	0.2538	0.1392	LR	field
6.0908	2.0972	3.9894	0.6364	0.1603	0.1357	GB	season
6.4101	1.7289	4.3007	0.3729	0.0348	0.0465	RF	season
6.6469	2.5449	4.5269	1.5350	-0.0403	0.4744	LR	season
6.7570	1.7429	4.6179	0.2521	-0.0787	0.0535	SVM	season

Table A.2: 5-fold cross-validation Results comparison table for average LWR prediction with relevance threshold = 0.7

RMSE mean	RMSE std	MAE mean	MAE std	R^2 mean	R^2 std	Algorithm	Data split
0.6180	0.1127	0.4522	0.0590	0.5788	0.0959	RF	field
0.6290	0.0663	0.4551	0.0313	0.5622	0.0741	RF	random
0.6453	0.1244	0.4666	0.0700	0.5409	0.1127	GB	field
0.6571	0.1059	0.4913	0.0527	0.5235	0.0909	LR	field
0.6575	0.0702	0.4713	0.0397	0.5213	0.0800	GB	random
0.6600	0.1025	0.4965	0.0588	0.5207	0.0797	SVM	field
0.6608	0.0480	0.4759	0.0323	0.5119	0.0925	SVM	random
0.6802	0.0673	0.5002	0.0494	0.4869	0.0928	LR	random
0.8508	0.2372	0.6926	0.2257	0.0883	0.4927	SVM	season
0.8989	0.3210	0.7517	0.3259	-0.1041	0.8021	LR	season
1.0316	0.5416	0.8718	0.5464	-0.6340	1.6204	RF	season
1.1270	0.6521	0.9562	0.6447	-1.0512	2.1752	GB	season

A.3 Feature Selection Number of Features Comparison

Table A.1: Number of features Results Comparison Table for Average Weight

Feature count	RMSE mean	R^2 mean	Algorithm	Data split
10	5.9	0.34	RF	random
20	5.8	0.33	RF	random
15	5.9	0.32	RF	random
15	6	0.3	RF	field
20	6	0.29	RF	field
10	6.1	0.29	SVM	field
10	6	0.23	SVM	season
15	6	0.19	RF	season
20	6.1	0.18	RF	season

Table A.2: Number of features Results Comparison Table for Average LWR

Feature_count	RMSE mean	R^2 mean	Algorithm	Data split
10	0.57	0.63	RF	random
10	0.57	0.63	RF	field
15	0.57	0.63	RF	field
20	0.57	0.63	RF	field
20	0.58	0.62	RF	random
15	0.58	0.61	RF	random
10	0.64	0.5	GB	season
15	0.64	0.49	GB	season
20	0.64	0.49	GB	season

A.4 Predictor Correlation Matrix

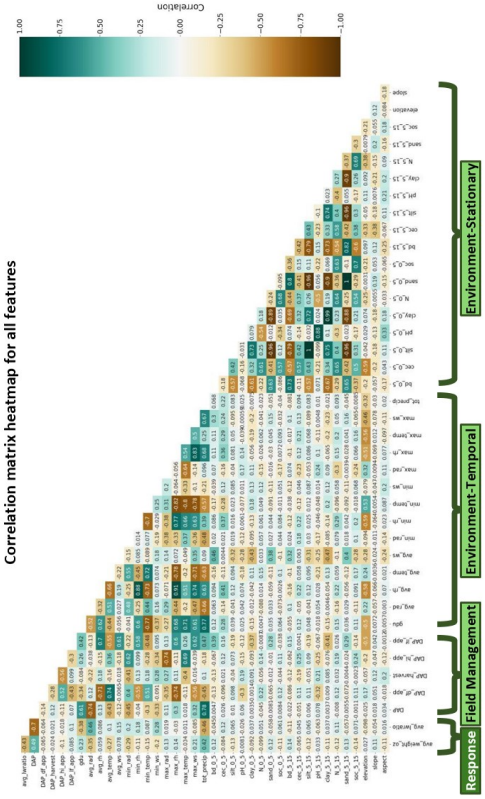

Figure A.1: Correlation matrix heatmap for response variables and all predictor variables

A.5 Summary Statistics

Here are the summary statistics for the predictors:

Table A.1: Summary Statistics

	Min	Max	Mean	St.dev
avg_weight_oz	0.17	71.31	8.08	7.12
avg_lwratio	1.49	8.70	2.91	0.97
DAP	20.00	144.00	77.32	24.92
DAP_df_app	-66.00	20.00	-12.63	28.98
DAP_harvest	99.00	147.00	122.45	11.63
DAP_hi_app	0.00	24.00	4.71	7.77
DAP_lf_app	-66.00	43.00	18.64	16.78
gdu	207.23	2514.42	1221.12	535.45
avg_rad	96.78	129.81	115.97	5.25
avg_rh	61.96	81.98	74.69	5.15
avg_temp	71.42	82.93	78.96	2.08
avg_ws	1.65	3.70	3.03	0.33
min_rad	17.77	82.43	38.72	12.66
min_rh	39.19	65.75	54.66	9.20
min_temp	31.17	84.15	61.70	12.07
min_ws	0.02	0.64	0.12	0.07
max_rad	143.41	159.98	154.15	4.89
max_rh	82.25	94.50	90.83	2.87
max_temp	76.19	101.32	89.83	8.94
max_ws	4.48	16.42	11.75	3.74

Table A.2: Summary Statistics (continued)

	Min	Max	Mean	Stdev
tot_precip	2.63	26.44	13.92	5.07
bd_0_5	1.28	1.40	1.35	0.03
cec_0_5	13.30	43.70	23.56	5.95
silt_0_5	20.50	37.90	28.08	4.78
pH_0_5	4.70	5.30	4.97	0.14
clay_0_5	8.50	20.20	12.47	2.95
N_0_5	37.86	70.24	49.17	7.99
sand_0_5	45.10	70.60	59.46	7.22
soc_0_5	34.30	101.10	52.86	12.83
bd_5_15	1.44	1.64	1.56	0.06
cec_5_15	5.10	19.20	10.75	2.28
silt_5_15	20.60	37.80	28.12	4.66
pH_5_15	4.80	5.40	5.08	0.12
clay_5_15	7.90	21.10	11.98	3.08
N_5_15	7.63	25.46	11.73	2.71
sand_5_15	45.40	70.60	59.89	7.25
soc_5_15	8.20	28.50	13.24	4.44
elevation	10.00	104.00	42.59	19.35
slope	0.31	5.99	1.70	0.98
aspect	0.00	346.25	181.32	100.09

A.6 Model Performance per fold

Table A.3: 5-Fold Model Performance for Linear Regression with predicted variable average LWR

FOLD	RMSE	MAE	R^2	Target	Algorithm	Data split
1	7.8654	4.0789	0.1257	avg_weight_oz	LR	random
2	6.9858	3.8634	0.2028	avg_weight_oz	LR	random
3	3.7332	2.8965	0.4850	avg_weight_oz	LR	random
4	6.7336	3.7762	0.2720	avg_weight_oz	LR	random
5	4.9134	3.9001	0.2360	avg_weight_oz	LR	random
1	4.3567	3.1080	0.4429	avg_weight_oz	LR	field
2	6.8769	3.6947	0.0870	avg_weight_oz	LR	field
3	6.5809	3.8963	0.1471	avg_weight_oz	LR	field
4	4.5441	3.4165	0.3951	avg_weight_oz	LR	field
5	7.9780	3.9567	0.2023	avg_weight_oz	LR	field
1	9.2994	5.3599	-0.1002	avg_weight_oz	LR	season
2	7.0874	5.4020	-0.3666	avg_weight_oz	LR	season
3	3.9975	3.1036	0.2651	avg_weight_oz	LR	season

Table A.4: 5-Fold Model Performance for Linear Regression with predicted variable average LWR

FOLD	RMSE	MAE	R^2	Target	Algorithm	Data split
1	0.7884	0.6030	0.3975	avg_lwratio	LR	random
2	0.6744	0.4432	0.5831	avg_lwratio	LR	random
3	0.7325	0.5613	0.3159	avg_lwratio	LR	random
4	0.6895	0.5357	0.5479	avg_lwratio	LR	random
5	0.5137	0.3966	0.5969	avg_lwratio	LR	random
1	0.5229	0.4159	0.6013	avg_lwratio	LR	field
2	0.8352	0.5970	0.3573	avg_lwratio	LR	field
3	0.6422	0.4980	0.6069	avg_lwratio	LR	field
4	0.5542	0.4426	0.6039	avg_lwratio	LR	field
5	0.7018	0.4970	0.4910	avg_lwratio	LR	field
1	0.8026	0.5669	0.4729	avg_lwratio	LR	season
2	1.6547	1.4947	-2.3844	avg_lwratio	LR	season
3	0.6298	0.5181	0.3245	avg_lwratio	LR	season

Table A.5: 5-Fold Model Performance for Support Vector Machine with predicted variable average weight

FOLD	RMSE	MAE	R^2	Target	Algorithm	Data split
1	7.9048	3.8001	0.1169	avg_weight_oz	SVM	random
2	7.0377	3.2526	0.1909	avg_weight_oz	SVM	random
3	3.5401	2.6874	0.5369	avg_weight_oz	SVM	random
4	6.9677	3.6686	0.2206	avg_weight_oz	SVM	random
5	4.2830	2.9416	0.4195	avg_weight_oz	SVM	random
1	4.3235	2.7832	0.4514	avg_weight_oz	SVM	field
2	6.6736	3.0145	0.1402	avg_weight_oz	SVM	field
3	6.5948	3.5819	0.1435	avg_weight_oz	SVM	field
4	4.2674	3.0404	0.4665	avg_weight_oz	SVM	field
5	8.0382	3.6168	0.1902	avg_weight_oz	SVM	field
1	9.1415	5.0747	-0.0631	avg_weight_oz	SVM	season
2	5.8322	4.3901	0.0746	avg_weight_oz	SVM	season
3	3.1032	2.3894	0.5571	avg_weight_oz	SVM	season

Table A.6: 5-Fold Model Performance for Support Vector Machine with predicted variable average LWR

FOLD	RMSE	MAE	R^2	Target	Algorithm	Data split
1	0.7123	0.5089	0.5082	avg_lwratio	SVM	random
2	0.6842	0.4573	0.5710	avg_lwratio	SVM	random
3	0.7369	0.5347	0.3075	avg_lwratio	SVM	random
4	0.7008	0.5395	0.5329	avg_lwratio	SVM	random
5	0.5080	0.3825	0.6057	avg_lwratio	SVM	random
1	0.4960	0.3867	0.6413	avg_lwratio	SVM	field
2	0.8449	0.5926	0.3423	avg_lwratio	SVM	field
3	0.6900	0.5313	0.5461	avg_lwratio	SVM	field
4	0.5765	0.4469	0.5713	avg_lwratio	SVM	field
5	0.6927	0.4598	0.5040	avg_lwratio	SVM	field
1	0.9910	0.7321	0.1965	avg_lwratio	SVM	season
2	1.6983	1.5293	-2.5651	avg_lwratio	SVM	season
3	0.5619	0.4486	0.4623	avg_lwratio	SVM	season

Table A.7: 5-Fold model performance for gradient boosting with predicted variable average weight

FOLD	RMSE	MAE	R^2	Target	Algorithm	Data split
1	8.2497	4.0196	0.0382	avg_weight_oz	GB	random
2	6.8171	3.4103	0.2409	avg_weight_oz	GB	random
3	3.5805	2.8924	0.5263	avg_weight_oz	GB	random
4	6.7294	3.7140	0.2730	avg_weight_oz	GB	random
5	5.1566	3.6087	0.1585	avg_weight_oz	GB	random
1	5.1623	3.0595	0.2178	avg_weight_oz	GB	field
2	6.4023	3.1921	0.2087	avg_weight_oz	GB	field
3	5.8755	3.3165	0.3201	avg_weight_oz	GB	field
4	4.4691	3.3579	0.4149	avg_weight_oz	GB	field
5	8.5136	4.2550	0.0916	avg_weight_oz	GB	field
1	8.7819	4.6319	0.0189	avg_weight_oz	GB	season
2	5.5345	3.8937	0.1667	avg_weight_oz	GB	season
3	3.6898	3.0609	0.3739	avg_weight_oz	GB	season

Table A.8: 5-Fold Model Performance for Gradient Boosting with predicted variable average LWR

FOLD	RMSE	MAE	R^2	Target	Algorithm	Data split
1	0.7578	0.5402	0.4434	avg_lwratio	GB	random
2	0.6489	0.4245	0.6141	avg_lwratio	GB	random
3	0.7268	0.5117	0.3264	avg_lwratio	GB	random
4	0.6624	0.5124	0.5827	avg_lwratio	GB	random
5	0.5322	0.4160	0.5673	avg_lwratio	GB	random
1	0.5175	0.4063	0.6095	avg_lwratio	GB	field
2	0.7978	0.5329	0.4136	avg_lwratio	GB	field
3	0.6589	0.4851	0.5861	avg_lwratio	GB	field
4	0.5216	0.3966	0.6491	avg_lwratio	GB	field
5	0.6619	0.4487	0.5472	avg_lwratio	GB	field
1	0.8509	0.6073	0.4077	avg_lwratio	GB	season
2	2.1315	1.9699	-4.6156	avg_lwratio	GB	season
3	0.5427	0.4372	0.4984	avg_lwratio	GB	season

Table A.9: 5-Fold model performance for random forest with predicted variable average weight

FOLD	RMSE	MAE	R^2	Target	Algorithm	Data split
1	7.8103	3.7615	0.1379	avg_weight_oz	RF	random
2	6.5686	3.1676	0.2952	avg_weight_oz	RF	random
3	3.2312	2.4608	0.6142	avg_weight_oz	RF	random
4	6.3019	3.2880	0.3624	avg_weight_oz	RF	random
5	4.2651	3.1233	0.4243	avg_weight_oz	RF	random
1	4.8454	3.0829	0.3109	avg_weight_oz	RF	field
2	6.2386	2.7688	0.2486	avg_weight_oz	RF	field
3	5.9684	3.2605	0.2985	avg_weight_oz	RF	field
4	4.5434	3.3659	0.3953	avg_weight_oz	RF	field
5	7.7939	3.6170	0.2387	avg_weight_oz	RF	field
1	8.7302	4.8922	0.0304	avg_weight_oz	RF	season
2	5.9028	4.0473	0.0521	avg_weight_oz	RF	season
3	3.6282	2.9061	0.3946	avg_weight_oz	RF	season

Table A.10: 5-Fold Model Performance for random forest with predicted variable average LWR

FOLD	RMSE	MAE	R^2	Target	Algorithm	Data split
1	0.7225	0.5231	0.4941	avg_lwratio	RF	random
2	0.6571	0.4319	0.6043	avg_lwratio	RF	random
3	0.7312	0.5366	0.3183	avg_lwratio	RF	random
4	0.6531	0.5056	0.5944	avg_lwratio	RF	random
5	0.5033	0.3927	0.6130	avg_lwratio	RF	random
1	0.5335	0.4131	0.5851	avg_lwratio	RF	field
2	0.7791	0.5413	0.4407	avg_lwratio	RF	field
3	0.6481	0.4792	0.5995	avg_lwratio	RF	field
4	0.4899	0.3918	0.6905	avg_lwratio	RF	field
5	0.6470	0.4409	0.5674	avg_lwratio	RF	field
1	0.8651	0.6095	0.3876	avg_lwratio	RF	season
2	1.9354	1.8033	-3.6296	avg_lwratio	RF	season
3	0.5321	0.4241	0.5178	avg_lwratio	RF	season

A.7 Permutation Importance Plots for All Models

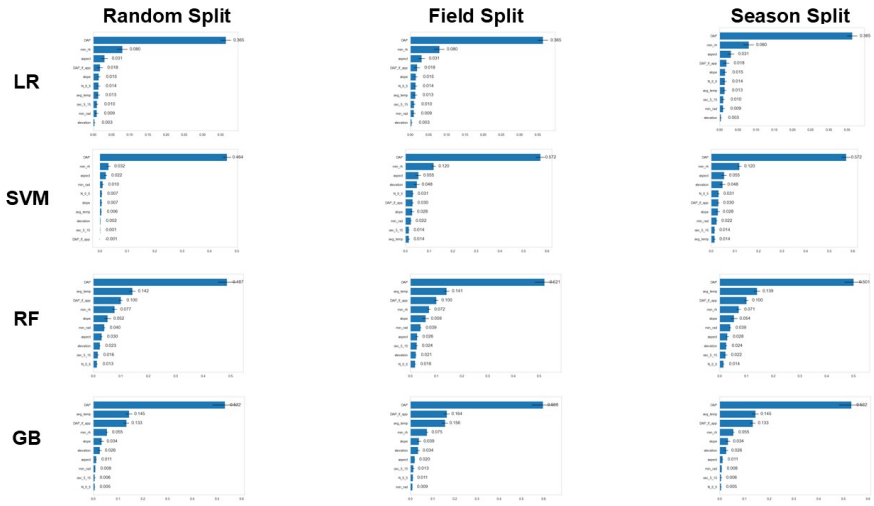

Figure A.1: Comparison of the top 10 important features for determining sweetpotato weight across all model-split combinations

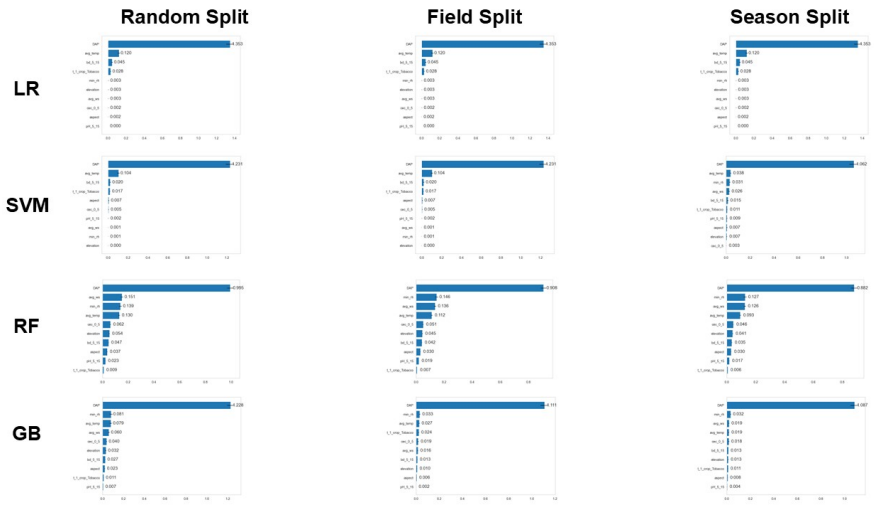

Figure A.2: Comparison of the top 10 important features for determining sweetpotato length-with-ratio across all model-split combinations

A.8 Optimal Model Parameters

We obtained the hyperparameters for each model using nested 5-fold cross-validation. The optimal parameters obtained are listed in the following tables.

A.8.1 Random Forest

Table A.1: Optimized hyperparameters for random forest with predicted variable average weight

Target	Data split	Parameter	Value
avg_weight	random	criterion	squared_error
		max_depth	10
		max_features	sqrt
		min_samples_leaf	4
		min_samples_split	10
		n_estimators	100
	field	criterion	squared_error
		max_depth	10
		max_features	sqrt
		min_samples_leaf	4
		min_samples_split	2
		n_estimators	100
	season	criterion	squared_error
		max_depth	20
		max_features	sqrt
		min_samples_leaf	4
		min_samples_split	10
		n_estimators	100

Table A.2: Optimized hyperparameters for random forest with predicted variable average LWR

Target	Data split	Parameter	Value
avg_lwr	random	criterion	squared_error
		max_depth	30
		max_features	sqrt
		min_samples_leaf	1
		min_samples_split	2
		n_estimators	100
	field	criterion	squared_error
		max_depth	30
		max_features	sqrt
		min_samples_leaf	1
		min_samples_split	5
		n_estimators	200
	season	criterion	squared_error
		max_depth	10
		max_features	sqrt
		min_samples_leaf	2
		min_samples_split	2
		n_estimators	500

A.8.2 Gradient Boosting

Table A.3: Optimized hyperparameters for gradient boosting with predicted variable average weight

Target	Data split	Parameter	Value
avg_weight	random	alpha	0.9
		criterion	friedman_mse
		learning_rate	0.01
		loss	squared_error
		max_depth	5
		min_samples_leaf	4
		min_samples_split	10
		n_estimators	100
	field	alpha	0.9
		criterion	friedman_mse
		learning_rate	0.01
		loss	squared_error
		max_depth	10
		min_samples_leaf	4
		min_samples_split	10
		n_estimators	100
	season	alpha	0.9
		criterion	friedman_mse
		learning_rate	0.01
		loss	squared_error
		max_depth	5
		min_samples_leaf	4
		min_samples_split	10
		n_estimators	100

Table A.4: Optimized Hyperparameters for Gradient Boosting with predicted variable average LWR

Target	Data split	alpha	0.9
avg_lwr	random	criterion	friedman_mse
		learning_rate	0.01
		loss	squared_error
		max_depth	5
		min_samples_leaf	2
		min_samples_split	5
		n_estimators	200
	field	alpha	0.9
		criterion	friedman_mse
		learning_rate	0.01
		loss	squared_error
		max_depth	3
		min_samples_leaf	2
		min_samples_split	10
		n_estimators	200
	season	alpha	0.9
		criterion	friedman_mse
		learning_rate	0.01
		loss	squared_error
		max_depth	3
		min_samples_leaf	4
		min_samples_split	10
		n_estimators	200

A.8.3 Support Vector Machine

Table A.5: Optimized Hyperparameters for Support Vector Machine with predicted variable average WT

Target	Data split	Parameter	Value
avg_weight	random	C	10
		gamma	auto
		kernel	rbf
	field	C	10
		gamma	scale
		kernel	rbf
	season	C	10
		gamma	scale
		kernel	rbf

Table A.6: Optimized Hyperparameters for Support Vector Machine with predicted variable average LWR

Target	Data split	Parameter	Value
avg_lwr	random	C	10
		gamma	scale
		kernel	linear
	field	C	10
		gamma	scale
		kernel	linear
	season	C	10
		gamma	auto
		kernel	rbf