MACHINE LEARNING AND FEATURE SELECTION FOR BIOMASS YIELD PREDICTION USING WEATHER AND PLANTING DATA

by

CHRISTOPHER DUNCAN WHITMIRE

(Under the Direction of Khaled M. Rasheed)

ABSTRACT

Using a relatively small amount of accessible data, we developed machine learning models to predict alfalfa yield and compared how different sets of features affected their error. We also compared the regression tree (RT), random forest (RF), neural network, support vector machine (SVR), k-nearest neighbors (KNN), Bayesian ridge regression, and linear regression methods. These methods were trained and evaluated with cross validation. The best set of features consisted of the Julian day of the harvest, the number of days between the sown date and the harvest date, and the cumulative amount of solar radiation and rainfall the crop received since the previous harvest. The RF, KNN, RT, and SVR methods obtained results that, when averaged, did not vary significantly from each other. The best individual model was a RF with a R^2 of 0.941. This model had the highest R^2 value compared to the best results from similar studies.

INDEX WORDS: Crop Yield Prediction, Biomass Yield Prediction, Machine Learning, Regression, Alfalfa, Feature Selection, Precision Agriculture

MACHINE LEARNING AND FEATURE SELECTION FOR BIOMASS YIELD PREDICTION USING WEATHER AND PLANTING DATA

by

CHRISTOPHER DUNCAN WHITMIRE

BS, Berry College, 2017

A Thesis Submitted to the Graduate Faculty of The University of Georgia in Partial Fulfillment of the Requirements for the Degree

MASTER OF SCIENCE

ATHENS, GEORGIA

2019

MACHINE LEARNING AND FEATURE SELECTION FOR BIOMASS YIELD PREDICTION USING WEATHER AND PLANTING DATA

by

CHRISTOPHER DUNCAN WHITMIRE

Major Professor: Khaled Rasheed
Committee: Ali Missaoui
Frederick Maier

Electronic Version Approved:

Suzanne Barbour
Dean of the Graduate School
The University of Georgia
August 2019

DEDICATION

I would like to dedicate this study to my parents. I could not have gotten where I am if not for their constant love and support.

ACKNOWLEDGEMENTS

I would like to thank Hend Rasheed who worked diligently in gathering and processing data, as well as contributing to the overall work. I would also like to thank Dr. Khaled Rasheed and Dr. Frederick Maier, who continually advised and guided me while working on this project. I would like to thank Dr. Ali Missaoui whose knowledge in plant sciences was invaluable. Finally, thanks to Jonathan Markham who helped provide the data and who helped me understand the data.

TABLE OF CONTENTS

APPENDICES

LIST OF TABLES

Page

LIST OF FIGURES

Page

CHAPTER 1

INTRODUCTION

This project seeks to explore the use of machine learning for crop yield prediction. First, the motivation of this work and an overview of related work is covered in Chapter 2. It will also draw connections between related work and the efforts of this study.

Chapter 3 presents a system for developing machine learning models for the purpose of biomass yield prediction and explores how different sets of features affect the error of these models. It does this by analyzing the results found by machine learning models trained on features found by different feature selection methods. The models are trained to predict alfalfa biomass yield. In doing this, a method for developing optimized machine learning models for the field of biomass yield prediction is demonstrated. The hope is that this system for developing machine learning models will allow plant scientists and agricultural planners to use machine learning for crop yield prediction, without needing a thorough background in machine learning. This chapter will also show how feature selection can provide insights into what attributes most affect alfalfa crop yield. It concludes by showing that the feature selection method that found the best set of features was a correlation-based method that minimized the correlation between the chosen features and maximized the correlation between that set of features and the target. The set of features this method found included the Julian day of the harvest, the number of days between the sown date and harvest date of the crop, and the cumulative solar radiation and rainfall the crop received since its previous harvest.

Chapter 4 begins by describing the motivation behind researching crop yield prediction. It then gives a brief summary of recent studies that also use machine learning to predict crop yield. Then, it expands on the work done in Chapter 3 by using the best set of features from Chapter 3 to predict alfalfa yield with a variety of machine learning methods. Regression trees, random forests, neural networks, support vector machines, k-nearest neighbors, Bayesian ridge regression, and linear regression are all used and compared. These models are evaluated with a variety of metrics and the results are compared to recent other studies. This project's results are comparable to the best results from similar studies, and the R^2 values of this project's models were the highest. This project's methods also used simpler data and more accessible features than many recent works. Specifically, the best results found by this study were found by a random forest with a mean absolute error of 162.01 lbs/acre, and a R^2 of 0.941.

CHAPTER 2

LITERATURE REVIEW

Agricultural planning is used to ensure that enough crops are produced, and is thus important for the economy, humanitarian efforts, and fighting world starvation and poverty (Dodds & Bartram, 2016; Rosegrant, Magalhaes, Valmonte-Santos, & Mason-D'Croz, 2018). It has also been found that research and development on increasing crop yields would result in $34 worth of benefit for every 1$ spent (Lomberg, 2015). One important tool in researching ways of increasing crop yield is yield prediction. Not only can it inform planners, but it could also potentially be used to streamline efforts in crop variety development.

There are many factors that affect crop yield, and many of these factors vary both spatially and temporally. Given this, it can be difficult to predict a crop's yield at a specific time. However, farmers often rely on their own experience to predict their yield (RuB, 2009). Given that personal experience can be unreliable, it would be beneficial to know what features have the largest impact on yield prediction. Work has been done showing that feature selection can improve the performance of machine learning models for crop yield prediction (Bocca & Rodrigues, 2016). This suggests that some factors play a larger role in affecting crop yield then others. Chapter 3 will expand on this idea by investigating how different factors affect predictive accuracy of machine learning models for alfalfa yield. This study will also reveal what factors more largely impact alfalfa yield itself.

There have been a variety of methods used to predict crop yield. The USDA makes monthly forecasts of crop yield in the United States by using survey techniques. They have achieved a very low percentage error (You, Li, Low, Lobell, & Ermon, 2017) but this method is very resource intensive, requiring thousands of phone calls and hundreds of inspections every month (National Agricultural Statistics Service, 2018; Johnson, 2014). Other studies have used remote sensing image data to train machine learning algorithms to predict crop yield (Chlingaryan, Sukkarieh, & Whelan, 2018). These methods have been successful (You et al., 2017; Johnson, 2014; Panda, Ames, & Panigrahi, 2010), but they require a large amount of processing of data from different platforms (Chlingaryan et al., 2018). There is also no particular piece of remote sensing data that universally works for all applications (Xue & Su, 2017). Remote sensing also cannot be used to make predictions until images are available to act as inputs. This means that models trained on remote sensing data cannot make predictions until the season starts, which is often too late to be useful (Cunha, Silva, & Netto, 2018).

Finally, other work has been done on using weather and soil features to train machine learning models to predict crop yield (González Sánchez, Frausto Solís, & Ojeda Bustamante, 2014; Ayoubi & Sahrawat, 2011; Jeong et al., 2016; Chlingaryan et al., 2018). Chapter 4 will expand on this by developing machine learning that not only uses accessible features like solar, rainfall, and planting data, but will also demonstrate that accurate machine learning models can be made using a small amount of this accessible data.

CHAPTER 3

THE IMPACT OF FEATURE SELECTION ON MACHINE LEARNING METHODS FOR BIOMASS YIELD PREDICTION USING WEATHER AND PLANTING DATA[1]

[1] Whitmire, C.D., Rasheed, H.K., Missaoui, A., Rasheed, K.M., & Maier, F.W. To be submitted to *Applications in Plant Sciences*

ABSTRACT

Predicting biomass and crop yield is important, and many features could be used to train machine learning models for yield prediction. Using yield data of different alfalfa varieties from multiple years in Kentucky and Georgia, we compared the impact of different feature selection methods on machine learning models trained to predict alfalfa yield. Linear regression, regression trees, support vector machines, neural networks, Bayesian regression, and nearest neighbors were all developed with cross validation. The features used included weather data, historical yield data, and the sown date. The feature selection methods that were compared included a correlation-based method, the ReliefF method, and a wrapper method. It was found that the best method was the correlation based method, and the feature set it found consisted of the Julian day of the harvest, the number of days between the sown and harvest dates, cumulative solar radiation since the previous harvest, and cumulative rainfall since the previous harvest. Using these features, the k-nearest neighbor and random forest methods achieved an average R value over 0.95 and average mean absolute error less than 200 lbs./acre. This work could be used to develop and improve efforts for biomass and crop yield prediction.

INTRODUCTION

In 2015, the United Nations developed 17 goals for the world to reach by the year 2030 (United Nations, 2015). These goals are meant to focus nations' efforts on solving the world's biggest problems, such as reducing worldwide poverty, improving physical health, reducing social inequalities, and improving environmental conditions. In order to evaluate whether those 17 goals were achieved, 169 targets were made (United Nations, 2015). However, these goals were not prioritized, and 85% of the proposals for these goals did not consider economic costs or benefits (Copenhagen Consensus Center, 2015). In response to this, the Copenhagen Consensus Center performed cost-benefit analyses on these 169 targets and ranked them according to the cost benefit ratio. One of their findings found that increasing research and development in increasing crop yields would be one of the most cost-effective ways of achieving some of these goals (Rosegrant, Magalhaes, Valmonte-Santos, & Mason-D'Croz, 2018). Specifically, every $1 spent on this kind of R&D would result in $34 worth of benefit. (Lomberg, 2015)

Improvements in agricultural planning and R&D on crop variety testing would increase crop yields, so work in these areas would help achieve some of the UN's goals. Machine learning techniques can be used for crop yield predictions, and these predictions can improve efforts in agricultural planning and crop variety testing. Specifically, by predicting a community's potential crop yield given certain conditions, farmers can better plan what to plant. This can help humanitarian efforts as well, by showing what communities should be receiving crops (Dodds & Bartram, 2016). Also, machine learning can help with crop variety testing. This testing is done to test the short-term and long-term yield of new varieties of crops. Having a prediction of a variety's yield may give agricultural scientists some insight into what varieties may be successful, allowing them to develop better varieties more efficiently.

Bocca and Rodrigues showed that feature selection can improve the predictive accuracy of machine learning models for crop yield prediction, while also simplifying the models (2016). This is because decreasing the number of features that are used to train a machine learning model can reduce noise in the data. This helps the performance of the machine learning models, but it can also help scientists understand what factors most impact crop yield. Because of this fact, this study will use alfalfa data from Georgia and Kentucky to make machine learning models to predict alfalfa yield. Then this study will explore the effect different feature selection methods have on the performance of these machine learning models. This will also provide information that may lead to insight into what factors most impact alfalfa yield in the Southeastern United States.

This paper will also present a method to develop optimized machine learning models for biomass and crop yield prediction. It is the hope of the authors that this will help readers, especially plant scientists and agricultural planners, develop their own machine learning models for crop yield prediction without requiring a background in machine learning.

Linear Regression

There are several diverse machine learning methods that can be used for crop yield prediction. Linear regression can be considered a machine learning technique and is often used as a baseline whose results are compared to the results of other techniques. Conceptually, linear regression finds a linear function that minimizes the squared error between the predictions of that function and the true values (Russell & Norvig, 2016).

This function will have the form $y_i = w_o + \sum_{i=1}^{k} w_i \, x_i$, where k is the number of features, the x_i is the value of a data point's ith feature, w_i is a coefficient associated with the ith feature, w_o is the intercept, and y_i is the prediction of the linear regression.

Neural Networks

Neural Networks, like linear regression, learn a function that minimizes the error between the predictions of the function and the true values. However, neural networks are capable of learning nonlinear functions of any complexity. It does this by roughly imitating the structure of the human nervous system (Rojas, 1996). A neural network is made up of multiple layers of nodes. Each node takes in inputs from a previous layer, performs a mathematical operation on those inputs, and outputs the results of that mathematical operation to the nodes in the next layer. The last layer outputs the final prediction. Typically, each node will output n where $n = A\left(\sum_{j=1}^{t} w_j \, m_j\right)$ with t being the number of inputs for that layer, m_j being the value of the jth input, w_j being the learned coefficient for the jth input, and A being a predefined nonlinear function. To train a neural network, all the coefficients (w_j's) are initialized with random values. Then the training data is fed to the network and predictions are found. An error is calculated by finding the difference between the prediction and the true value. By finding the gradient of the error, the neural network can iteratively change the coefficients of each node to minimize the overall error. By changing the number of layers and nodes, a neural network can approximate many different functions (Mitchell, 1997).

Support Vector Machines

Another approach is done by support vector machines (SVMs). SVMs attempt to make a linear best fit line that keeps all the predictions within a certain error threshold from that best fit line. However, this technique can fit nonlinear data by projecting the data into a higher dimensional space. In this higher dimensional space, that data will appear more linear, so a linear best fit line can be made in this higher dimensional space. The best fit line is then projected back to the original space where it no longer appears linear (Gonzalez, Frausto, & Ojeda, 2014). This is called the 'kernel trick' (Russell & Norvig, 2016).

K-Nearest Neighbors

The k-nearest neighbor (kNN) method is another spatially based machine learning method. This method remembers all the data it has been shown before, and when it receives an input X, it looks at the distance between X and all those other points. It then finds the k closest points to X and uses them to make a prediction. The prediction is found by calculating a normalized weighted sum of the values of the k closest points. The weights are often proportional to the distance between the saved point and X (Gonzalez et al., 2014), but all the weights could be equal. If this case, kNN is finding the average value of the k closest points.

Regression Trees

Regression trees learn patterns by recursively breaking up the sample space into different regions where each region gives a certain prediction. Note that regression trees tend to split the space into many regions, so it can make many predictions. (Quinlan, 1992) It does all of this by forming a tree of nodes. Each node asks a certain question about one of the input's features. For example, a node may ask whether the input data point has a solar radiation value greater than 600 MJ/m^2. If the answer is yes, then it will go to another node and ask another question. If the answer is no, it will go to a different node. This process continues until an answer is given. In order to learn what questions to ask, the regression tree will minimize some impurity measure (Gonzalez et al., 2014). Note that a random forest is a collection of multiple regression trees, and the final output of a random forest is the average result of all its regression trees.

Bayesian Ridge Regression

Bayesian ridge regression is a probabilistic method that is like linear regression. But instead of making a linear function, a probability distribution is made based on the training data. Using Bayes rule, this method outputs the most likely value given the input values (Gelman, 2013). Since this is a ridge regression, a cost is added to the error if the coefficients are above a certain threshold. This encourages the model to not become too complicated and overfit the data.

Feature Selection

These machine learning methods use a variety of different techniques to make predictions, and the effect different feature selection methods have on their results will be compared. Correlation based feature selection (Cfs) will be done, and its effect on each model will be shown. Cfs methods look at the correlation between each feature and the target as well as the correlation between the features. It then finds the set of features that maximizes the correlation between the feature set and the target while also minimizing the correlation between the chosen features (Dash and Liu, 1997; Hall, 1999). By minimizing the intra-correlation between features, Cfs reduces redundancy and noise and can show what relatively independent processes contribute to the target's value.

Another feature selection method is the ReliefF method. It develops weights for each of the features and adjusts those weights depending on the similarity of feature values among clustered data points. It does this by first initializing each weight to be zero. Then, it picks a random point from the dataset and finds the point in the dataset that has the closest target value to that random point. Then the features between these two points are compared. For every feature, if the values of that feature are similar among those two points, the weight for that feature is increased. However, if the values are dissimilar, then the weight of that feature is decreased (Kononenko, 1994).

Cfs and ReliefF are both filter feature selection methods. This means that they look at characteristics of the features themselves and uses that information to decide what features should be used. Wrapper feature selection methods on the other hand, use a machine learning algorithm to learn what sets of features lead to the best results. This paper will use a wrapper method using a ZeroR classifier. The ZeroR classifier uses the average value of each feature to predict the target. The effects of Cfs, ReliefF feature selection, and the wrapper method on the results of machine learning models for alfalfa biomass yield will all be analyzed and compared.

METHODS

The programming language used to clean the data, make visualizations, apply feature selection methods, and to make the machine learning models was Python (Python Software Foundation) within the Anaconda environment (Anaconda Software Distribution). Many packages for python were used. Pandas was used to clean and organize the data (McKinney, 2010), matplotlib was used to make the visualizations

(Hunter, 2007), seaborn was used to make a heat map showing the correlation between features (Waskom et al., 2016), sci-kit learn was used for all of the machine learning and the SelectKBest feature selection operations (Pedregosa et al., 2011), and finally, numpy was used for general mathematical operations (Oliphant, 2006; Van Der Walt, Colbert, & Varoquaux, 2011). Weka was used for the CfsSubsetEval (Cfs), ReliefFAttributeEval (ReliefF), and WrapperSubsetEval (Wrapper) feature selection operators (Witten, Frank, Hall, & Pal, 2016).

The features used in training our machine learning models were the Julian day of the harvest, the number of days between the harvest and the sown date of the crop, the number of days between the current harvest and the previous harvest, the total amount of solar radiation and rainfall since the last harvest, the percent cover and day length at the time of the harvest, the average air temperature since the previous harvest, the average minimum air temperature since the last harvest, and the average maximum air temperature since the previous harvest, and the average soil moisture since the last harvest (Table 3.1). All the features that are averages were formed by obtaining daily values and averaging over every daily value. For example, the average air temperature feature was found by getting the average temperature for each day between the crop's previous harvest and current harvest. Then all the daily values were averaged resulting in the final value for the average air temperature feature.

These features were constructed from various datasets. All the data sources are shown in Appendix 1. Alfalfa yield and harvest data were obtained from alfalfa variety trials done by the University of Georgia (UGA) and University of Kentucky (UKY). This data contained the yield (tons/acre) of multiple varieties of alfalfa. UGA's data was from

Athens and Tifton, Georgia from the years 2008 to 2010 and included data points from April to December. UKY's data contained yield data from Lexington, Kentucky ranging from 2013 to 2018 and contains data from May to September. Each data set contained the yield, harvest date, and sown date for multiple varieties over time. The percent cover was also given along with the dates it was measured, but the percent cover was measured on different dates than when the crop was harvested.

For every data point, the date the crop was harvested was converted into a Julian date. For each harvest, the number of days since the crop's sown date and the number of days since the last harvest were calculated. The percent cover of the crop at its harvest date was found using interpolation.

Daily weather data was also found. Data for Tifton and Watkinsville, which is about 13 miles from Athens, GA, was retrieved from the Georgia Automated environmental network. Similar data was found for Versailles, which is nearby Lexington, KY, from the National Oceanic and Atmospheric Administration (NOAA). These weather data sets contained the daily amount of solar radiation and rainfall, as well as the average air temperature, minimum and maximum air temperature, and the soil moisture. The day length was found using the United States Naval Observatory website.

By using the weather data for the dates corresponding with the alfalfa harvest times, we calculated for each harvest: the total amount of solar radiation and rainfall that location had received since the previous harvest, and the average temperature, minimum temperature, maximum temperature, and soil moisture since the previous harvest.

TABLE 3.1: Sample data point. A data point with the same features as the data used to train our machine learning models.

Feature Name	Value	Abbreviation
Julian day of harvest	249.00	JD
Number of days since the crop was sown	643.00	DSS
Number of days since last harvest	30.00	DSH
Total solar radiation since the previous harvest (MJ/m^2)	610.29	Sol
Total rainfall since the previous harvest(mm)	98.83	Rain
Avg air temp since the previous harvest (C)	25.33	T
Avg max air temp since the previous harvest (C)	31.25	MaxT
Avg min air temp since the previous harvest (C)	19.1	MinT
Avg soil moisture since the previous harvest (%)	0.11	SM
Interpolated percent cover for the day of the harvest (%)	78.82	PC
Day length on the day of the harvest (hrs)	12.62	DL

Once the data was gathered, all the data which had invalid values were disregarded. Also, all the data points that had harvest dates that happened in the same year as the sown date were filtered out. Similarly, the first harvest of every season was filtered out. This is because the amount of time since the previous harvest would be much larger for this harvest relative to subsequent harvests. After this cleaning process, 770 data points were left. Athens had 108 corresponding data points, Tifton had 70, and Lexington had 592.

Before training the models, we applied feature selection and standardized the data. For feature selection, we first used Sci-Kit Learn’s SelectKBest to show how changing the number of features changes the average R of each method. Feature selection with Weka's CFsSubsetEval (Cfs), ReliefFAttributeEval (ReliefF), and WrapperSubsetEval (Wrapper) operators was then used to train machine learning models, and their results were compared. Then all the features were standardized according to the formula $x_{new} = \frac{x_{old} - x_{mean}}{x_{SDev}}$ where x_{old} is the value of the feature before standardization, x_{mean} is the average value of the features, and x_{SDev} is the standard deviation of the values for that feature.

The following was done for each method. Before training the models, the data was shuffled and split into ten folds to be used for 10-fold cross validation. For each iteration of cross validation, one of the ten folds was used as a testing set while the other nine folds were used to train the machine learning model. Each fold was a testing set for one of the 10 iterations and was not used as the testing set more than once. Then for each iteration of the cross validation, a machine learning model was initialized. A grid search (Appendix 2) with 5-fold cross validation was done to find the hyperparameters for the model that most minimized the mean absolute error. Only the training set for this iteration was used here. Once the hyperparameters were found, the machine learning model was trained on the training set and was evaluated against the testing set. The mean absolute error (MAE), R value, and R squared value were all found and recorded. This was done for each of the ten iterations. Note that this means that ten different models were made for each method. The average MAE, R, and R squared value over all ten models were also found and recorded.

This process was done to train and evaluate regression tree, random forest regression, k-nearest neighbor regression, support vector regression, neural networks, Bayesian Ridge regression, and linear regression. Once all the machine learning models were trained and evaluated for the different sets of features found by the different feature selection operators, a two-tailed unpaired t test was performed between the results. This was used to determine if any of the feature selection operators picked feature subsets that led to significantly better results.

RESULTS

For every feature selection method, the average MAE, R, and R^2 value for each model over the ten iterations are shown. Note that the average yield in the dataset is 2020 lbs./acre.

Using the SelectKBest feature selection method, we made all features available for feature selection and compared the results for K=3 to K= 11. Notice that as K increases, the R value increases, but the increase in R levels off at around K=6 features (Fig 3.1). These 6 features were the Julian day, number of days since the crop was sown, total solar radiation, average soil moisture, day length, and percent cover. The results of the models with no feature selection are shown in Fig 3.2 and Table C.1. Here, the support vector regression model had the highest average R of 0.948.

We used Weka's Cfs method for feature selection. If all features were made available for feature selection, it found that the best features to use would both maximize the correlation between the features to the target and minimize the correlation between the features were the Julian day, total solar radiation, total rainfall, and the percent cover. The results from training the models using just these features are shown in Fig 3.3 and Table C.2. The random forest method had achieved the highest R

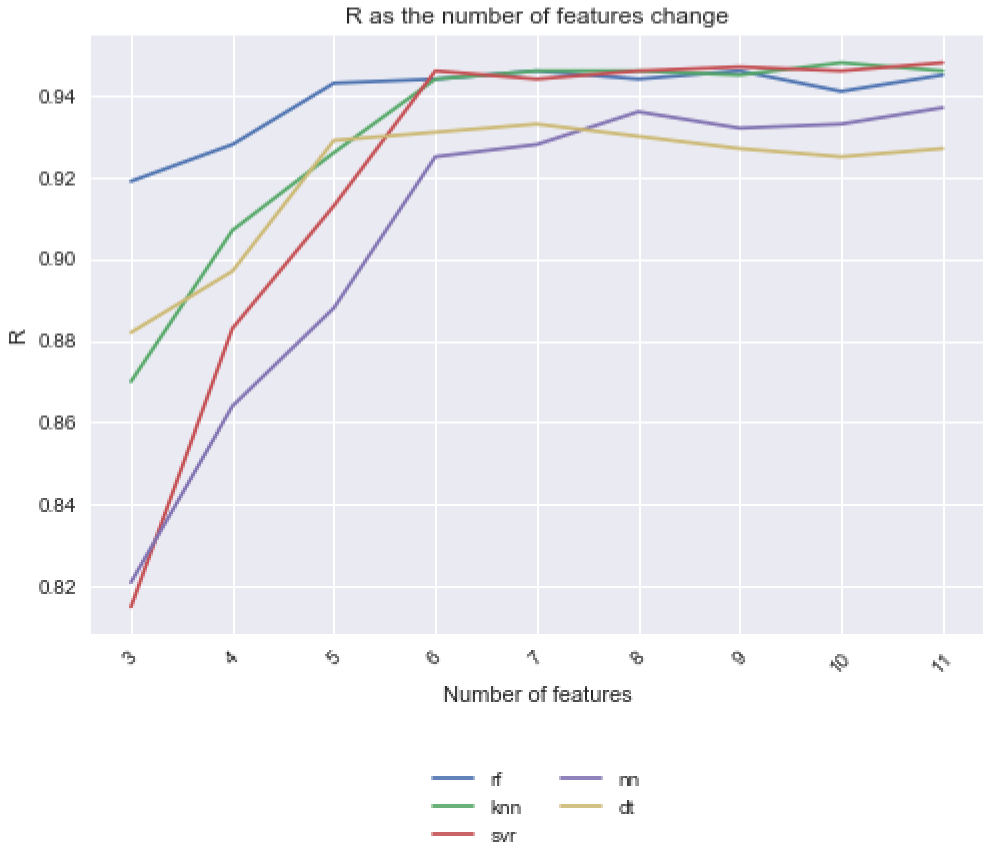


FIGURE 3.1: Performance of models with k features and all features made available for feature selection. The average R value of the models is shown. SelectKBest feature selection was used with K values from K=3 to K=11. Note that the average R value for Bayesian Ridge Regression and linear regression were much lower than any of the other models, so they were not shown here.

with a R of 0.933. The correlations between the features and target are shown in Fig C.1.

However, because it may not be easy to get an accurate value of percent cover, we did another experiment with Weka's Cfs method for feature selection. In this experiment, we made all the features available for feature selection except for percent cover. It found that the best set of features to use in this case were the Julian day, total solar radiation, total rainfall, and the number of days since the sown date. The results of evaluating the models trained on just these features are shown in Fig. 3.4 and Table C.3.

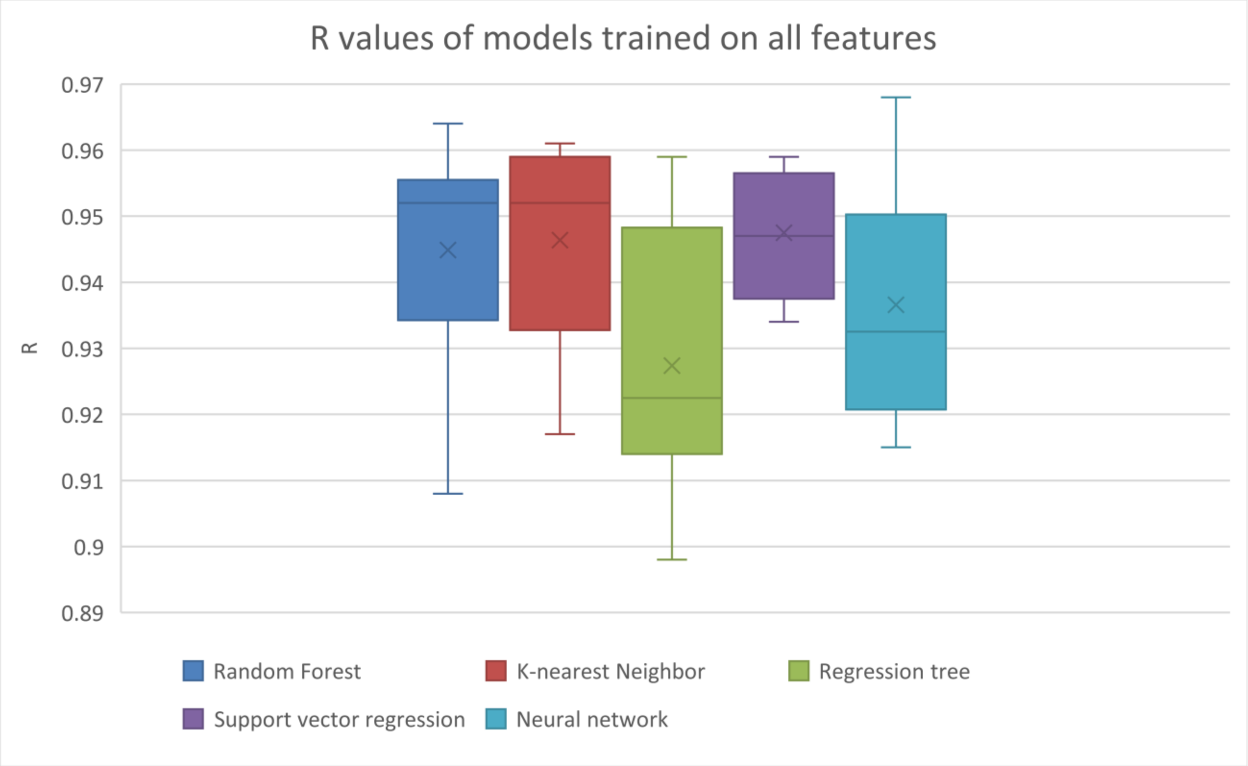


FIGURE 3.2: R values found with no feature selection. The results from linear regression and Bayesian ridge regression were much lower than the other models, so their results are not shown here. The results are shown explicitly in Table C.1.

The k-nearest neighbor and random forest methods both achieved the best average R with this set of features by obtaining an average R of 0.952.

To compare the results obtained from using the two sets of features found by Cfs, an unpaired two-tailed t test was performed between the R values of the models trained with the features chosen by the Cfs operator (Table 3.2). The random forest, k-nearest neighbor, and regression tree methods performed significantly better using the feature set that excluded percent cover from being available for selection. The other methods did not vary significantly across the two sets of results. Because excluding percent cover led to results that were significantly better or the same when compared to not excluding percent cover, only the results found by Cfs without percent cover will be considered for the rest of this work.

The ReliefF operator found that the best features were the number of days between the crop's sown date and harvest date, the cumulative amount of rainfall the crop got since the previous harvest, and the average minimum daily temperature since the previous harvest. The results from training the machine learning models with these features are shown in Fig. 3.5 and Table C.4. In this case, k-nearest neighbors achieved the highest average of R with a value of 0.953.

The Wrapper operator reported that the best features were number of days between the crop's sown date and harvest date, the cumulative amount of rainfall since the previous harvest, the day length at the time of the harvest, and the Julian day of the harvest. The results of the machine learning models trained on these features is shown in Fig. 3.6 and Table C.5. The best R value of these methods was also k-nearest neighbors getting an average R of 0.952.

Unpaired two-tail t tests were done between the R values of the methods which used all the features, the Cfs features (without percent cover), the ReliefF features, and the Wrapper features (Table 3.3). To show these results more clearly, Table 3.4 shows what feature selection operator led to the best results for each machine learning method. There is no significant difference in the results given by the feature selection operators in the same row of Table 3.4.

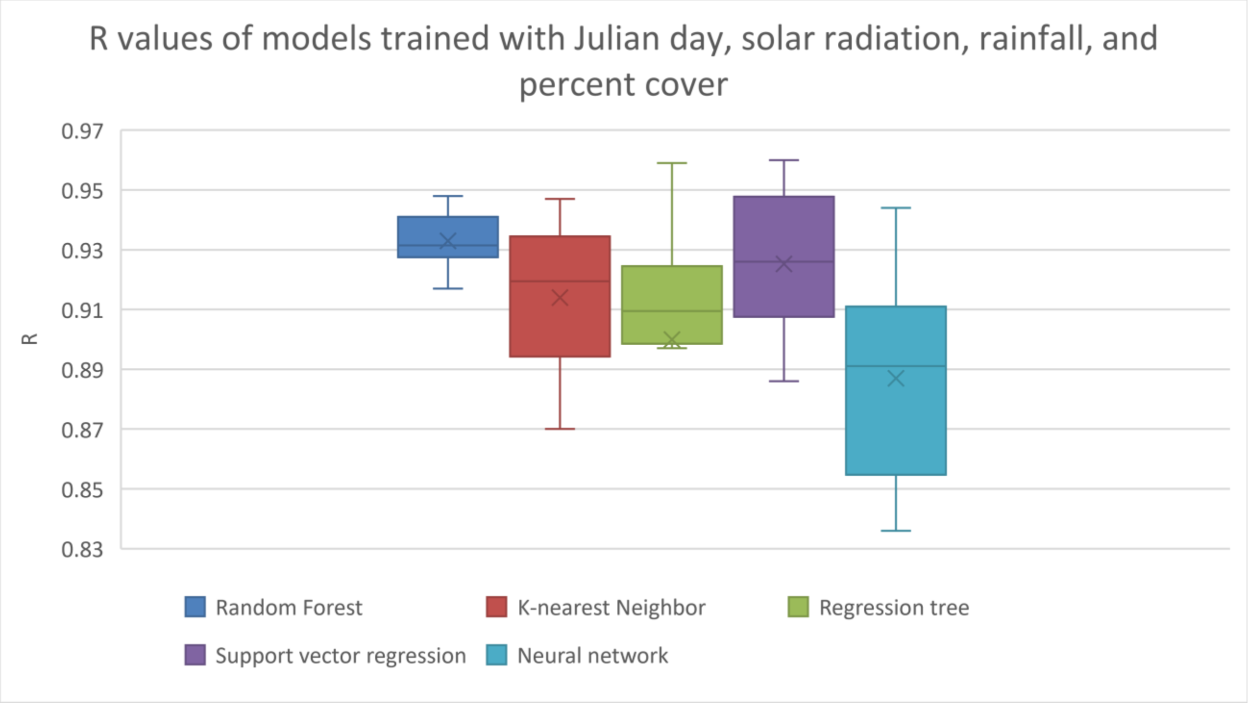


FIGURE 3.3: Results from Cfs feature selection with all features. The results from linear regression and Bayesian ridge regression were much lower than the other models, so their results are not shown here. The results are shown explicitly in Table C.2.

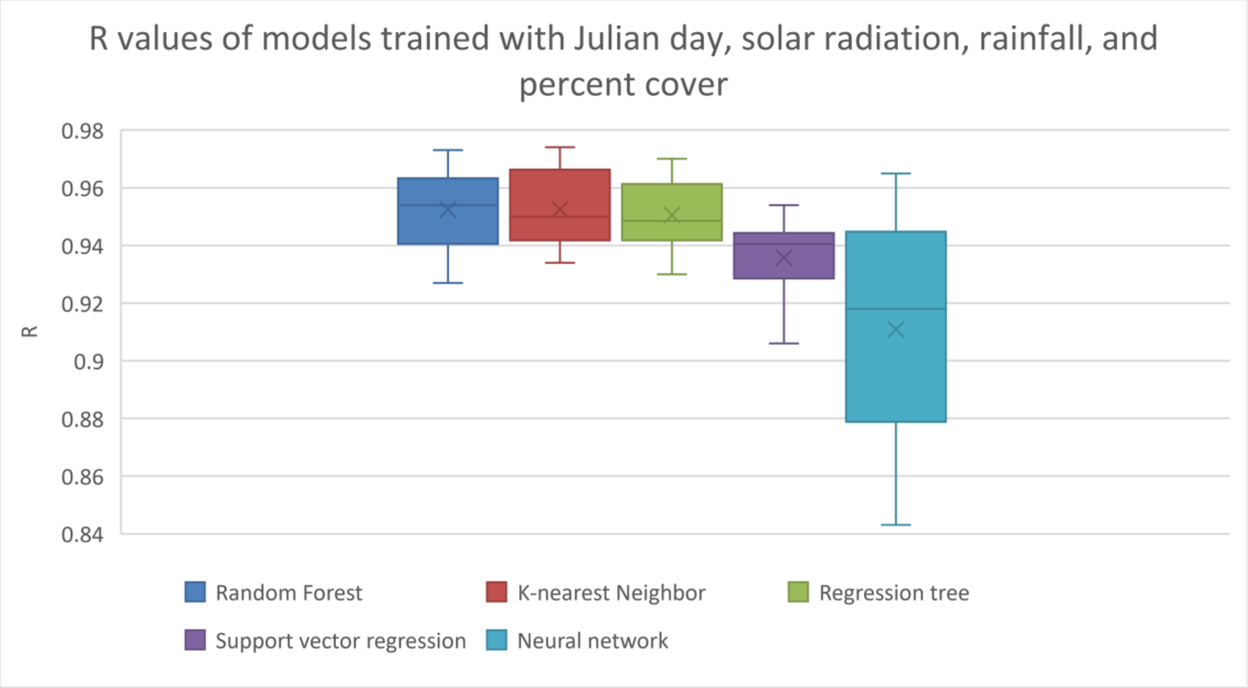


FIGURE 3.4: **Results from Cfs feature felection with no percent cover.** The results from linear regression and Bayesian ridge regression were too low to show. The results are shown explicitly in Table C.3.

TABLE 3.2: P-values between the R² values of the models trained by the two CfsSubsetEval feature sets. The results were found by doing unpaired two-tailed t tests. The first feature set contained the Julian day, total solar radiation, total rainfall, and percent cover. The second feature set contained the Julian day, the number of days since the sown date, total solar radiation, and the total rainfall. Significant results are shown in bold.

Model	**T test results**
Random forest	**0.0046**
K-nearest neighbor	**0.0007**
Regression tree	**0.0103**
Support vector regression	0.2820
Neural network	0.2070
Linear regression	0.8940
Bayesian ridge regression	0.7481

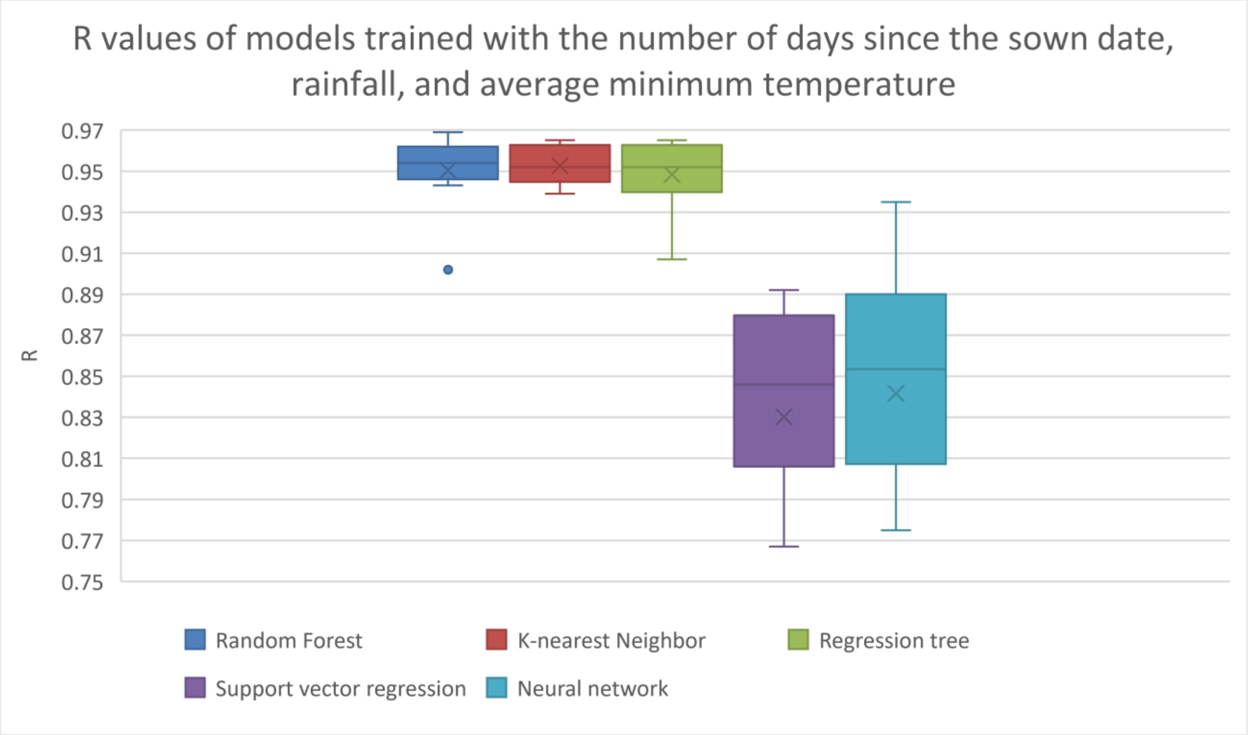


FIGURE 3.5: Results from ReliefF feature selection. The results from linear regression and Bayesian ridge regression were much lower than the other models, so their results are not shown here. The results are shown explicitly in Table C.4.

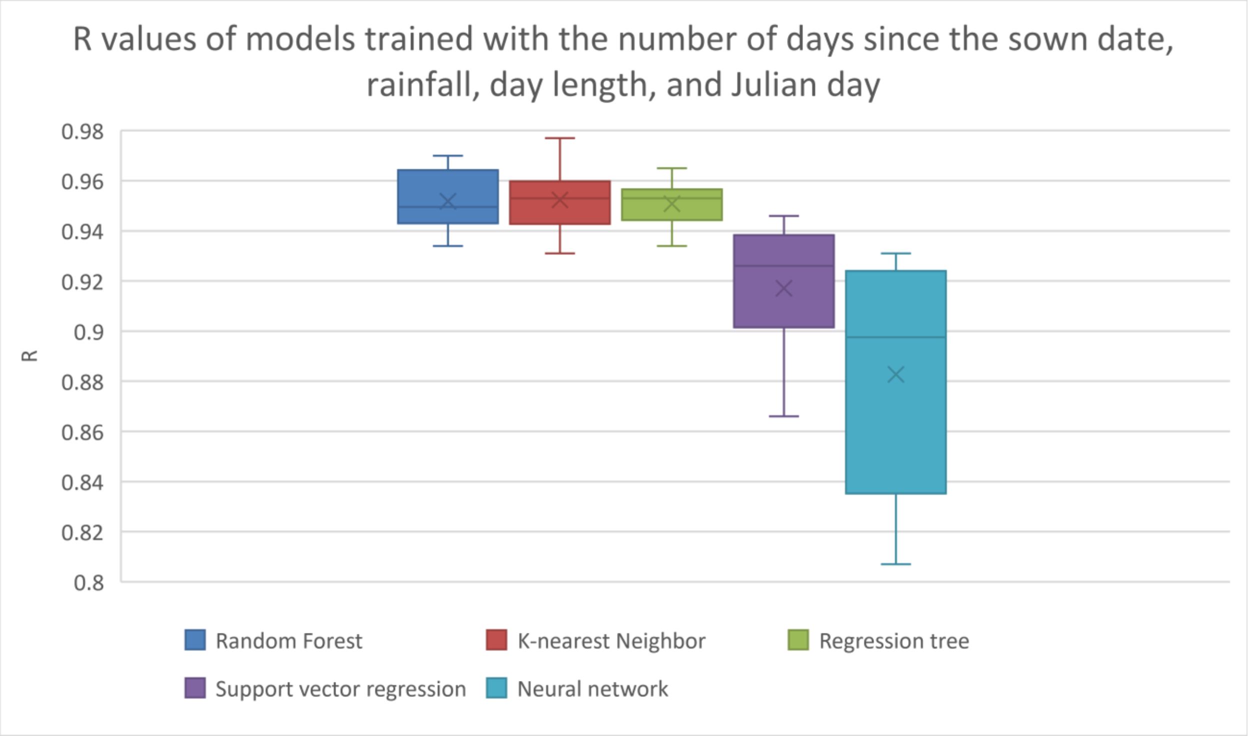


FIGURE 3.6: Results from Wrapper feature selection operator. The results from linear regression and Bayesian ridge regression were much lower than the other models, so their results are not shown here. The results are shown explicitly in Table C.5.

Table 3.3: P-values between R^2 values of different feature selection operators. Results from unpaired two-tail t tests. 'All' represents the results from Table C.1, 'Cfs' represents the results which used the features from Fig 3.4/Table C.3, 'ReliefF' represents the results from Fig 3.5/Table C.4, and 'Wrapper' represents the results from Fig 3.6/Table C.5. If a p-value is followed by a parenthesis, the value in the parentheses is an abbreviation of the feature selection method that resulted in the higher average R^2 value.

T test	**RF**	**KNN**	**RT**	**SVR**	**NN**	**Lin**	**Bayes**
All vs Cfs	0.2973	0.3303	**0.0086 (C)**	0.0559	0.0871	0.3758	0.3795
All vs Relieff	0.4631	0.2306	**0.0140 (R)**	**0.0001 (A)**	**0.0010 (A)**	**2E-13 (A)**	**3E-15 (A)**
All vs Wrapper	0.2398	0.3321	**0.0045 (W)**	**0.0038 (A)**	**0.0035 (A)**	0.7555	0.3569
Cfs vs Relieff	0.8331	0.9179	0.8967	**0.0002 (C)**	**0.0156 (C)**	**3E-12 (C)**	**3E-11 (C)**
Cfs vs Wrapper	0.9867	0.9804	0.7840	0.0685	0.2196	0.6726	0.9486
Relieff vs Wrapper	0.8057	0.8924	0.6999	**0.0014 (W)**	0.1052	**5E-10 (W)**	**8E-13 (W)**

Table 3.4: Best feature selection operators for each machine learning method. There is no significant difference between the results in the same cell. 'All' refers to all features being used, 'Cfs' refers to the set of features found by CfsSubsetEval, 'ReliefF' refers to the set of features found by ReliefFAttributeEval, and 'Wrapper' refers to the set of features found by 'WrapperSubsetEval'.

Machine learning method	**Feature selection operator that led to the best results**
Random forest	All, Cfs, ReliefF, Wrapper
K-nearest neighbors	All, Cfs, ReliefF, Wrapper
Regression tree	Cfs, ReliefF, Wrapper
Support vector regression	All, Cfs
Neural network	All, Cfs
Linear regression	All, Cfs, Wrap
Bayesian ridge regression	All, Cfs, Wrap

DISCUSSION

The Cfs operator was the best overall feature selection method because it led to the best results for each method. None of the other feature selection operators led to the best results for each method. The feature set that the Cfs operator found consisted of the Julian day, the number of days between the sown and harvest date, the cumulative solar radiation since the previous harvest, and the cumulative rainfall since the last harvest.

There was no significant difference in any of the random forest results, no matter the feature selection method. The same is true for k-nearest neighbors. Even though using all features does not result in a significant difference from using a feature selection operator, it would still be beneficial to use a feature selection operator. Doing so would lower computational time and could simplify the models. The same can be said for support vector regression and the neural network, which got the best results from using either all the features or Cfs. For the regression tree, using any of the three feature selection methods resulted in better results than if all the features were used. In this case, even though fewer features are used, the results improved. This may be because different features can embed the same information. For example, the Julian day of the harvest and the day length features both refer to seasonal information, therefore they would have a high correlation with each other (Fig C.1). Thus, including both the Julian day of the harvest and the day length could add noise to the model. For linear regression and Bayesian ridge regression, using anything but the ReliefF operator led to the best results. This is probably because forming a linear prediction function with only three features is not appropriate for this domain.

This work may be helpful because it describes a framework that can be applied to other machine learning problems in predicting crop and biomass yield. This work also shows what features are most important for predicting alfalfa yield in the Southeast United States from Spring to the end of Fall. The best results came from training the models with the Julian day, amount of solar radiation and rainfall since the previous harvest, and the number of days since the crop was sown. This is useful because gathering data is resource intensive and knowing the best features can help make data collecting more efficient. These four features are also relatively easy to obtain. The Julian day and amount of time since the crop was sown are trivial to retrieve, and the amount of solar radiation and rainfall can be obtained from weather data sources.

Also, besides possibly improving the results of the models, feature selection can provide insight into the problem domain (Dash & Liu, 1997). By understanding what features are most important for predicting yield, one may gain insight into what factors most impact a crop's yield. The cumulative rainfall since the previous harvest and the number of days between the harvest date and sown date were chosen by all the feature selection methods, so this is evidence that they may be the most important features for this problem. Similarly, the Julian day was chosen by two out of three feature selection methods, so this is evidence that it is also an important feature.

This work could be extended by providing this framework to alfalfa crops grown in other locations besides Georgia and Kentucky. It could also be improved by incorporating more data from other locations in the Southeast United States.

CHAPTER 4

COMPARING MACHINE LEARNING METHODS FOR BIOMASS YIELD PREDICTION USING WEATHER AND PLANTING DATA[2]

[2] Whitmire, C.D., Rasheed, H.K., Missaoui, A., Rasheed, K.M., & Maier, F.W. To be submitted to *Computers and Electronics in Agriculture*

ABSTRACT

Predicting crop yield is important for agricultural planning and humanitarian efforts. Efforts had been made to use remote sensing, weather, planting, and soil data to train machine learning models for yield prediction. However, remote sensing, though successful, requires large amounts of data be processed, and the models cannot make predictions until the harvesting season begins. Using weather and planting data from alfalfa variety trials in Kentucky and Georgia, we developed machine learning models to predict biomass yield. Linear regression, regression trees, support vector machines, neural networks, k-nearest neighbor and Bayesian ridge regression methods were all used. Cross validation was used to find the optimal hyperparameters and to evaluate the methods. There was no significant difference between the results of the random forest, k-nearest neighbor, regression tree, and support vector regression when the results for each model were averaged. We compared the results of our methods to the results of other studies. We achieved results that were comparable with the best results of the studies we examined, but our models used a small amount of data and accessible features. Our best individual model was a random forest with a mean absolute error of 162.01 lbs/acre and a R^2 of 0.941.

INTRODUCTION

With the intent of directing world leaders towards solving some of the world's biggest problems, the United Nations has recently developed 17 goals and 169 targets. The hope is that the world will reach these goals by the year 2030 (United Nations, 2015). However, it is the opinion of the Copenhagen Consensus Center (CCC), a think tank, that prioritizing these goals will make it more likely that the goals will be reached (Copenhagen Consensus Center, 2015). The CCC has performed a cost-benefit analysis on all these targets and ranked them accordingly. One of their findings was that increasing research and development of increasing crop yields would be one of the most cost-effective ways of achieving the UN's goals (Rosegrant, Magalhaes, Valmonte-Santos, & Mason-D'Croz, 2018). Specifically, every $1 spent on this kind of R&D would result in $34 worth of benefit worldwide. (Lomberg, 2015)

One possible way to increase yields is to improve agricultural planning. This would help ensure that there are sufficient yields of particular crops. At the start of every season, agricultural planners need to estimate the yields of different agricultural plans (Frausto-Solis, Gonzalez-Sanchez, & Larre, 2009). Often, farmers rely on their own personal experiences of history to predict what their yields will be, but this can be inaccurate (RuB, 2009). Given that crop yield varies spatially and temporally, and are sensitive to varying conditions like weather, better prediction methods should be investigated.

The USDA, with its National Agricultural Statistics Service branch, makes monthly forecasts of crop yields in the United States. It does this by conducting two surveys, a farm operator survey and an objective survey. The farm operator survey is done by calling farmers at random and asking them what they think their predicted yield for the next month will be. The objective survey involves an investigator going out and surveying random fields and recording data on the output of those fields. The findings of these surveys are compared to previous historical data to confirm that the findings are consistent with previous harvests with similar conditions. The final predicted yields then come from the results of these surveys (National Agricultural Statistics Service, 2018; Johnson, 2014). The findings of this methodology, when compared to the ground truth, have had very low errors (You, Li, Low, Lobell, & Ermon, 2017; National Agricultural Statistics Service, 2018). However, it is very resource intensive. The farm operator survey is done primarily over the phone, and the objective survey requires measurements to be taken in person at hundreds of farms every month (National Agricultural Statistics Service, 2018; Johnson, 2014).

An alternative approach is to use remote sensing (RS) data. RS techniques use images achieved primarily from aircraft of satellites, and these images will record spectral, spatial, and temporal information (Chlingaryan, Sukkarieh, & Whelan, 2018). Mathematical operations can be performed on these images to form vegetation indices (VIs), which can be used as inputs into machine learning algorithms (Xue and Su, 2017). Recent work has been done to use VIs to predict crop yield. You et al., had great success at predicting county level soybean yield in the United States using remote sensing data as input for a convolutional neural network and a LSTM, both with a Gaussian Process component (2017). Panda, Ames, & Panigrahi used several different VIs as an input to a neural network to predict corn yield (2010). Johnson did something similar but used regression trees to predict both corn and soybean yield (2014). However, despite these successes, there are difficulties with making machine learning models based on remote sensing data. This is because using remote sensing data means depends on the processing of large amounts of data across different platforms (Chlingaryan, 2018). These models also cannot make a prediction unless there are images available for input, which means that this model cannot begin making predictions until the season has started (Cunha, Silva, & Netto, 2018). Xue and Su also compared over one hundred different vegetation indices and found that no VI is universally better than the others. Each is more suitable to certain situations, and each has their own limitations (2017). This means that it may be difficult to know the optimal VI to be used in any particular case.

Weather, spatial, and soil features have also been used to train machine learning models to predict crop yield (González Sánchez, Frausto Solís, & Ojeda Bustamante, 2014; Ayoubi & Sahrawat, 2011; Jeong et al., 2016; Chlingaryan et al., 2018). These kinds of data also require less processing than remote sensing data and can be used to make predictions before the season starts. They also have the potential to use weather forecasting results to make predictions before the season begins, making it more convenient for planning purposes than using remote sensing data. This paper will use weather and planting data to develop a variety of machine learning models and will compare the results.

METHODS

The Python programming language was used throughout this research (Python Software Foundation). Specifically, Python as provided within the Anaconda environment was used (Anaconda Software Distribution). The following packages were used: Pandas for data cleaning and preparation (McKinney, 2010), matplotlib (Hunter, 2007) and seaborn (Waskom et al., 2016) for visualizations, sci-kit learn to make and evaluate the machine learning models (Pedregosa et al., 2011), and finally, numpy for general mathematical operations (Oliphant, 2006; Van Der Walt, Colbert, & Varoquaux, 2011).

The features used in training our machine learning models were the Julian day of the harvest, the amount of days between the harvest and the sown date of the crop, the cumulative solar radiation since the previous harvest, and the cumulative rainfall since the last harvest. The cumulative solar radiation and rainfall values were found by summing daily values.

All the data sources for this work are presented in Appendix 1. Alfalfa harvest data was obtained from variety trials done by the University of Georgia (UGA) and University of Kentucky (UKY). This data contained the yield (tons/acre) of multiple varieties of alfalfa. UGA's data came from Athens and Tifton, Georgia from the years 2008 to 2010. Harvests were done here from April to December. UKY's data contained yield data from Lexington, Kentucky ranging from 2013 to 2018 and contains data from the months of May through September. Each data set contained the yield, harvest date, and sown date for alfalfa crop.

Daily weather data was found for each location. Data for Tifton and Watkinsville, which is about 13 miles from Athens, GA, was retrieved from the Georgia Automated environmental network. Similar data was found for Versailles, which is nearby

Lexington, KY, from the National Oceanic and Atmospheric Administration (NOAA). These weather data were made up of daily weather data.

All the data which had invalid values were disregarded. Also, all the data points that had harvest dates with the same year as the sown date were filtered out. Similarly, the first harvest of every season was removed because the amount of time since the previous harvest would be much larger for this harvest relative to subsequent harvests. After this cleaning process, 770 data points were left. Athens had 108 corresponding data points, Tifton had 70, and Lexington had 592.

Before training the models, we standardized the data. All of the features were standardized according to the formula $x_{new} = \frac{x_{old} - x_{mean}}{x_{SDev}}$ where x_{old} was the original value of the feature, x_{mean} is the average value of the features, and x_{SDev} is the standard deviation of the values for that feature.

Before training the models, the data was shuffled and split into ten folds to be used for 10-fold cross validation. For each fold, a machine learning model was initialized. This means that ten models were made for each method, one model for each fold. Then, within this outer fold, a grid search (Appendix 2) with 5-fold cross validation was done to find the hyperparameters for the model that most minimized the mean absolute error. Once the hyperparameters were found, the machine learning model was trained on the training set and was evaluated against the testing set. The mean absolute error (MAE), mean absolute percent error (MAPE), root mean square error (RMSE), R value, and R squared value were all found and recorded (Table 4.1). The average errors, percent error, R, and R squared value over the ten iterations was found and recorded, and the results of the best model were also recorded along with their standard deviations.

This process was done to train and evaluate the following methods: regression tree, random forest regression, k-nearest neighbors, support vector machines, neural networks, Bayesian Ridge regression, and linear regression. Once the results for each method were obtained, an unpaired two-tailed t test was used to find the p-value between the average R^2 values of each method.

TABLE 4.1: Evaluation metric definitions. The metrics used to evaluate each method. For each case, n is the number of total data points, $true_i$ is ground truth value for the ith data point, $pred_i$ is the predicted value for the ith data point, $\overline{y_t}$ is the average yield value from the dataset, and $\overline{y_p}$ is the average value of the predictions.

Metric	Equation
Mean absolute error (MAE)	$\frac{1}{n}\sum_{i=1}^{n} true_i - pred_i$
Mean absolute percent error (MAPE)	$\frac{100}{n}\sum_{i=1}^{n}\left\lvert\frac{true_i - pred_i}{true_i}\right\rvert$
Root mean square error (RMSE)	$\sqrt{\frac{\sum_{\mathrm{i=1}}^{\mathrm{n}}(\mathrm{true_i} - \mathrm{pred_i})^2}{\mathrm{n}}}$
R	$\frac{\sum_{i=1}^{n}(true_i - \overline{y_t})(pred_i - \overline{y_p})}{\sqrt{\sum_{i=1}^{n}(true_i - \overline{y_t})^2 \cdot \sum_{i=1}^{n}(pred_i - \overline{y_p})^2}}$
R^2	$1 - \frac{\sum_{i=1}^{n}(true_i - pred_i)^2}{\sum_{i=1}^{n}(true - \overline{y_t})^2}$

RESULTS

For each method, ten models were made and evaluated. These results are shown in Fig 4.1 and Table 4.2. Also, for each method, the results for the model with the best R² value out of the ten models were recorded (Table 4.3). Note that the average yield from the entire dataset was 2020 lbs./acre. The p-values between the average results of each method is shown in Table 4.4.

FIGURE 4.1: Average results. The average results found over the 10 iterations for each type of model. The results from linear regression and Bayesian ridge regression were much lower than the other models, so their results are not shown here.

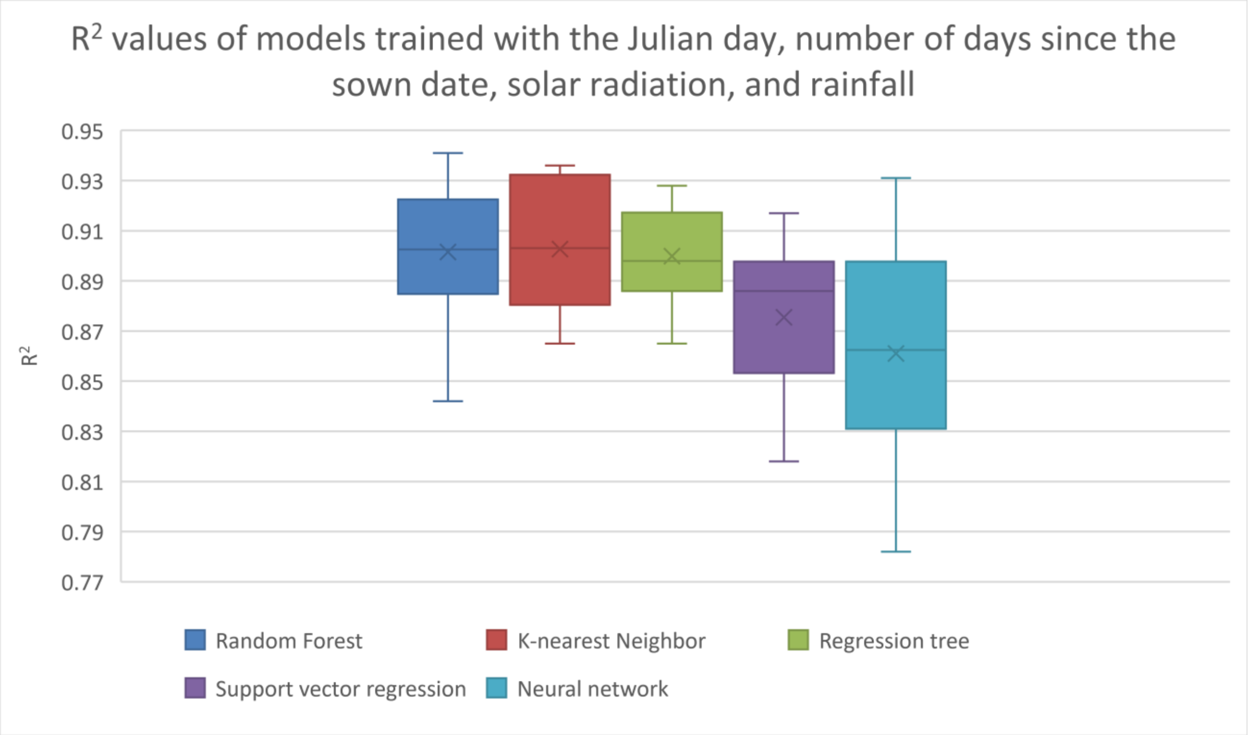


TABLE 4.2: Average results. The average results found over the 10 iterations for each type of model. Each result is shown as 'average results +/- 2σ', where σ is the standard deviation. The best result in each column is shown in bold.

Model	MAE (lbs./acre)	MAPE (%)	RMSE (lbs./acre)	R	R^2
Regression Tree	199.87 +/- 29.884	12.742 +/- 5.15	272.085 +/- 60.982	0.951 +/- 0.02	0.9 +/- 0.042
Random Forest	197.508 +/- 34.128	12.728 +/- 6.916	**267.067 +/- 54.412**	0.95 +/- 0.03	0.902 +/- 0.058
K-Nearest Neighbors	**194.558 +/- 42.612**	**12.725 +/- 5.2**	267.363 +/- 54.572	**0.952 +/- 0.026**	**0.903 +/- 0.052**
Support Vector Machines	227.375 +/- 56.136	17.093 +/- 7.372	301.198 +/- 59.65	0.937 +/- 0.034	0.876 +/- 0.068
Neural Network	242.95 +/- 61.886	15.874 +/- 5.608	316.218 +/- 89.124	0.932 +/- 0.042	0.861 +/- 0.088
Bayesian Ridge Regression	372.139 +/- 72.446	25.798 +/- 9.84	518.463 +/- 177.618	0.802 +/- 0.118	0.642 +/- 0.188
Linear Regression	371.836 +/- 92.538	25.521 +/- 10.176	518.365 +/- 173.14	0.802 +/- 0.096	0.638 +/- 0.146

TABLE 4.3: Results of best models. The results from the model with the highest R^2 value. The best result in each column is shown in bold.

Model	**MAE (lbs./acre)**	**MAPE (%)**	**RMSE (lbs./acre)**	**R**	**R^2**
Regression Tree	182.078	14.632	248.418	0.963	0.928
Random Forest	**162.01**	**9.892**	**218.913**	**0.97**	**0.941**
K-Nearest Neighbors	181.082	13.264	231.769	0.968	0.936
Support Vector Machines	188.365	14.016	245.468	0.958	0.917
Neural Network	184.816	15.561	239.856	0.965	0.931
Bayesian Ridge Regression	294.055	21.723	380.825	0.882	0.777
Linear Regression	320.906	34.528	474.567	0.851	0.723

Table 4.4: P-values between different machine learning methods. Significant values are shown in bold.

	RF	SVR	KNN	RT	NN	Linear	Bayes
RF	1	0.080	0.923	0.888	**0.027**	**2E-07**	**5E-06**
SVR	0.080	1	0.058	0.069	0.421	**4E-07**	**1E-05**
KNN	0.923	0.058	1	0.792	**0.021**	**2E-07**	**5E-06**
RT	0.888	0.069	0.792	1	**0.025**	**4E-07**	**7E-06**
NN	**0.027**	0.421	**0.021**	**0.025**	1	**5E-07**	**1E-05**
Linear	**2E-07**	**4E-07**	**2E-07**	**4E-07**	**5E-07**	1	0.903
Bayes	**5E-06**	**1E-05**	**5E-06**	**7E-06**	**1E-05**	0.903	1

DISCUSSION

Linear regression is commonly used as a baseline, and all other methods performed better than it except for the Bayesian ridge regression method. On average, the k-nearest neighbor method had the best MAE, MAPE, R value, and R^2 value, and the random forest had the best average RMSE. However, K-nearest neighbor, random forest, regression tree, and support vector regression all had average results that did not differ significantly from each other. The best individual model overall was a random forest model. It performed the best according to all metrics.

It can be difficult to compare results between papers given that different metrics are used in different papers. Some metrics are also not suitable for comparing two models if the models used different datasets or are working in different contexts. However, we have attempted to compare our results with the best results of other work that used machine learning to predict crop yield by using the R, R^2, and MAPE values (Table 4.5). These values are inherently normalized to the data used to train each model.

Note that You et al, Johnson, Panda et al., and Kuwata & Shibasaki all used remote sensing data to train their machine learning models. González Sánchez et al., Ayoubi & Sahrawat, and Jeong et al. used weather, planting, and/or soil data as features for their machine learning models.

Our results are better or at least comparable to the findings of other studies (Table 4.5). Our procedure also uses features that are easy to find and require little and no processing, unlike remote sensing data. Our method has the potential to make predictions before the harvesting season begins, while remote sensing cannot make any predictions until data from the harvesting season has been recorded (Cunha et al., 2018). In this way, our procedure for developing machine learning algorithms for crop yield prediction is more convenient. As we and others have demonstrated, good results can be obtained with these simpler features that do not use remote sensing data.

A weakness of our method is that is only applicable to a specific region. Our models were trained with alfalfa data in Kentucky and Georgia, USA, and they would not be able to make reliable predictions for alfalfa in other parts of the world. However, some studies have worked to make more universal models, and with great success (You et al., 2017). Further work could be done to compare the results of a universal model against the results of several regional models, using similar datasets.

TABLE 4.5: Results comparison. A comparison between different studies on using machine learning for crop yield prediction. A dash means the study did not use that metric. Note that the best results from each study is shown here.

Study	R	R²	MAPE (%)
Our Study's Average Results: RF	0.95	0.902	12.728
Our Study's best Results: RF	0.97	0.941	9.892
You et al., 2017: CNN and LSTM with GP	-	-	3.19
Johnson, 2014: DT	-	0.93	-
Panda et al., 2010: NN	-	0.72	7
Kuwata & Shibasaki, 2015: NN	0.81	-	-
González Sánchez et al., 2014: DT	0.74	-	-
Ayoubi & Sahrawat, 2011: NN	-	0.93	-
Jeong et al., 2016: RF	0.98	-	-

CONCLUSION

Predicting crop yield is essential for agricultural preparation and can be helpful in reaching some of the worldwide goals established by the United Nations. Much work has been done using remote sensing, weather, planting, and soil data to predict crop yield. We have proposed a procedure that uses only four features that are easy to find and process, and this procedure results in good results regarding alfalfa yield in the Southeast United States. The four features used were the Julian day of the harvest, the number of days between when the crop was sown and when it was harvested, the cumulative solar radiation since the previous harvest, and the cumulative rainfall since the previous harvest. K-nearest neighbor, random forest, a regression tree, and support vector regression had average results which did not vary significantly from each other. The best single model was a random forest, which achieved a MAE of 162.01 lbs./acre and a R^2 value of 0.941.

CHAPTER 5

CONCLUSION

This project was successful in exploring the effect of feature selection on machine learning models for biomass yield prediction and in developing machine learning models with high R^2 values and low percentage errors using a relatively small amount of accessible data. The models were made to predict alfalfa yield in the Southeastern United States. After doing feature selection, the optimal features found were the Julian day, the number of days between the sown date and harvest date, the cumulative solar radiation since the crop's previous harvest, and the cumulative rainfall since the crop's previous harvest. K-nearest neighbor, random forest, a regression tree, and support vector regression performed the best and did not vary significantly from each other. The best average result found was obtained by k-nearest neighbor, and it had a MAE of 194.558 lbs/acre and a R^2 of 0.903. The best individual model was a random forest model which achieved a MAE of 162.01 lbs/acre and R^2 of 0.941.

This work could be expanded by using more data. The methods described here could also be used to develop predictive models for other crops in other regions. It would also be interesting to determine if the set of features that were found to be optimal in our study were also optimal for other regions and other crops. A direct comparison between using weather and historical planting data, and vegetative indices would also be insightful. By using the same region and time, these differing sets of features could be better compared. An exploration of using a combination of remote sensing, historical planting, and weather data together would also be useful.

This work could be further expanded by exploring more hyperparameters for the different methods. Neural networks may especially benefit from this given that there has been so much recent work in developing successful deep learning neural networks for a variety of applications.

Finally, work on using plant characteristics as features may help to make a universal prediction model that could work across different species and regions. Features such as leaf size, root depth, and temperature constraints, along with weather and soil features, may could be used to make universal models.

REFERENCES

Anaconda Software Distribution. Computer software. Version 4.6.8 Anaconda, Nov. 2016. Website https://anaconda.com

Ayoubi, S., & Sahrawat, K. L. (2011). Comparing multivariate regression and artificial neural network to predict barley production from soil characteristics in northern Iran. *Archives of Agronomy and Soil Science*, *57*(5), 549-565.

Bocca, F. F., & Rodrigues, L. H. A. (2016). The effect of tuning, feature engineering, and feature selection in data mining applied to rainfed sugarcane yield modelling. *Computers and electronics in agriculture*, *128*, 67-76.

Chlingaryan, A., Sukkarieh, S., & Whelan, B. (2018). Machine learning approaches for crop yield prediction and nitrogen status estimation in precision agriculture: A review. *Computers and electronics in agriculture*, *151*, 61-69.

Copenhagen Consensus Center. (2015). *Background*. Retrieved from https://www.copenhagenconsensus.com/post-2015-consensus/background

Cunha, R. L., Silva, B., & Netto, M. A. (2018, October). A Scalable Machine Learning System for Pre-Season Agriculture Yield Forecast. In *2018 IEEE 14th International Conference on e-Science (e-Science)* (pp. 423-430). IEEE.

Dash, M., & Liu, H. (1997). Feature selection for classification. *Intelligent data analysis*, *1*(1-4), 131-156.

Dodds, F., & Bartram, J. (Eds.). (2016). *The water, food, energy and climate Nexus: Challenges and an Agenda for action*. Routledge.

Frausto-Solis, J., Gonzalez-Sanchez, A., & Larre, M. (2009, November). A new method for optimal cropping pattern. In *Mexican International Conference on Artificial Intelligence* (pp. 566-577). Springer, Berlin, Heidelberg.

Gelman, A., Stern, H. S., Carlin, J. B., Dunson, D. B., Vehtari, A., & Rubin, D. B. (2013). *Bayesian data analysis*. Chapman and Hall/CRC.

González Sánchez, A., Frausto Solís, J., & Ojeda Bustamante, W. (2014). Predictive ability of machine learning methods for massive crop yield prediction.

Hall, M. A. (1999). Correlation-based feature selection for machine learning.

Hunter, J. D. (2007). Matplotlib: A 2D graphics environment. *Computing in science & engineering*, *9*(3), 90.

Jeong, J. H., Resop, J. P., Mueller, N. D., Fleisher, D. H., Yun, K., Butler, E. E., Timlin, D.J., Shim, K., Gerber, J.S., Reddy, V.R., & Kim, S. H. (2016). Random forests for global and regional crop yield predictions. *PLoS One*, *11*(6), e0156571.

Johnson, D. M. (2014). An assessment of pre-and within-season remotely sensed variables for forecasting corn and soybean yields in the United States. *Remote Sensing of Environment*, *141*, 116-128.

Kononenko, I. (1994, April). Estimating attributes: analysis and extensions of RELIEF. In *European conference on machine learning* (pp. 171-182). Springer, Berlin, Heidelberg.

Lomborg, B. (2015). *The Nobel Laureates' Guide to the Smartest Targets for the World: 2016-2030*. Copenhagen Consensus Center USA.

Kuwata, K., & Shibasaki, R. (2015, July). Estimating crop yields with deep learning and remotely sensed data. In *2015 IEEE International Geoscience and Remote Sensing Symposium (IGARSS)* (pp. 858-861). IEEE.

McKinney, W. (2010, June). Data structures for statistical computing in python. In *Proceedings of the 9th Python in Science Conference* (Vol. 445, pp. 51-56).

Miao, Y., Mulla, D. J., & Robert, P. C. (2006). Identifying important factors influencing corn yield and grain quality variability using artificial neural networks. *Precision Agriculture*, *7*(2), 117-135.

Mitchell, Tom. (1997) *Machine Learning*. United States of America, McGraw-Hill.

National Agricultural Statistics Service. (2018, November). *Crop Production*. Retrieved from https://www.nass.usda.gov/Publications/Todays_Reports/reports/crop1118.pdf

Oliphant, T. E. (2006). *A guide to NumPy* (Vol. 1, p. 85). USA: Trelgol Publishing.

Panda, S. S., Ames, D. P., & Panigrahi, S. (2010). Application of vegetation indices for agricultural crop yield prediction using neural network techniques. *Remote Sensing*, *2*(3), 673-696.

Pantazi, X. E., Moshou, D., Alexandridis, T., Whetton, R. L., & Mouazen, A. M. (2016). Wheat yield prediction using machine learning and advanced sensing techniques. *Computers and Electronics in Agriculture*, *121*, 57-65.

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

Python Software Foundation. Python Language Reference, version 3.6.8. Website http://www.python.org

Quinlan, J. R. (1992, November). Learning with continuous classes. In *5th Australian joint conference on artificial intelligence* (Vol. 92, pp. 343-348).

Rojas, R. (1996). Neural Networks-A Systematic Introduction Springer-Verlag. *New York*.

Rosegrant, M. W., Magalhaes, E., Valmonte-Santos, R. A., & Mason-D'Croz, D. (2018). Returns to investment in reducing postharvest food losses and increasing agricultural productivity growth. *Prioritizing Development: A Cost Benefit Analysis of the United Nations' Sustainable Development Goals*, 322.

Ruß, G. (2009, July). Data mining of agricultural yield data: A comparison of regression models. In *Industrial Conference on Data Mining* (pp. 24-37). Springer, Berlin, Heidelberg.

Russell, S. J., & Norvig, P. (2016). *Artificial intelligence: a modern approach*. Malaysia; Pearson Education Limited.

United Nations. (2015). Transforming our world: The 2030 agenda for sustainable development. *Resolution adopted by the General Assembly*.

Van Der Walt, S., Colbert, S. C., & Varoquaux, G. (2011). The NumPy array: a structure for efficient numerical computation. *Computing in Science & Engineering*, *13*(2), 22.

Waskom, M., Botvinnik, O., drewokane, Hobson, P., David, Halchenko, Y., ... & Lee, A., (2016). seaborn: v0. 7.1 (June 2016). *Zenodo. doi*, 10.5281/zenodo.54844

Witten, I. H., Frank, E., Hall, M. A., & Pal, C. J. (2016). *Data Mining: Practical machine learning tools and techniques*. Morgan Kaufmann.

Xue, J., & Su, B. (2017). Significant remote sensing vegetation indices: a review of developments and applications. *Journal of Sensors*, *2017*.

You, J., Li, X., Low, M., Lobell, D., & Ermon, S. (2017, February). Deep gaussian process for crop yield prediction based on remote sensing data. In *Thirty-First AAAI Conference on Artificial Intelligence*.

APPENDIX A

CODE AND DATA ACCESSIBILITY

The code used for this project can be found at

https://github.com/chriswhitmire/alfalfa-yield-prediction

The University of Georgia alfalfa yield data can be found here:

https://georgiaforages.caes.uga.edu/species-and-varieties/cool-season/alfalfa.html

The University of Kentucky alfalfa yield data can be found as progress reports on this page: http://dept.ca.uky.edu/agc/pub_prefix.asp?series=PR

Note that the only data that was used from the University of Kentucky was the non-roundup ready alfalfa varieties that were first harvested in the year 2013 or later.

The daily weather data for Kentucky was found on the National Oceanic and Atmospheric Administration website: https://www.ncdc.noaa.gov/crn/qcdatasets.html

The daily weather data for Georgia was given to us by the Georgia Automated Environmental Monitoring Network.

The day length was found from the United States Naval Observatory's website:

https://aa.usno.navy.mil/data/docs/Dur_OneYear.php

APPENDIX B

HYPERPARAMETER GRID VALUES

The grid for the hyperparameters of each model is as follows:

Regression Tree-

- ‘criterion': ['mae'],
- ‘max_depth': [5,10,25,50,100]

Random forest -

- 'n_estimators': [5, 10, 25, 50, 100],
- 'max_depth': [5, 10, 15, 20],
- 'criterion': ["mae"]

K-nearest neighbors-

- 'n_neighbors': [2,5,10],
- 'weights': ['uniform', 'distance'],
- 'leaf_size': [5, 10, 30, 50]

Support vector machine-

- 'kernel': ['linear', 'poly', 'rbf', 'sigmoid'],
- 'C': [0.1, 1.0, 5.0, 10.0],
- 'gamma': ["scale", "auto"],
- 'degree': [2,3,4,5]

Neural Network-

- 'hidden_layer_sizes':[(3), (5), (10), (3,3), (5,5), (10,10)],

- 'solver': ['sgd', 'adam'],
- 'learning_rate' : ['constant', 'invscaling', 'adaptive'],
- 'learning_rate_init': [0.1, 0.01, 0.001]

Bayesian ridge regression-

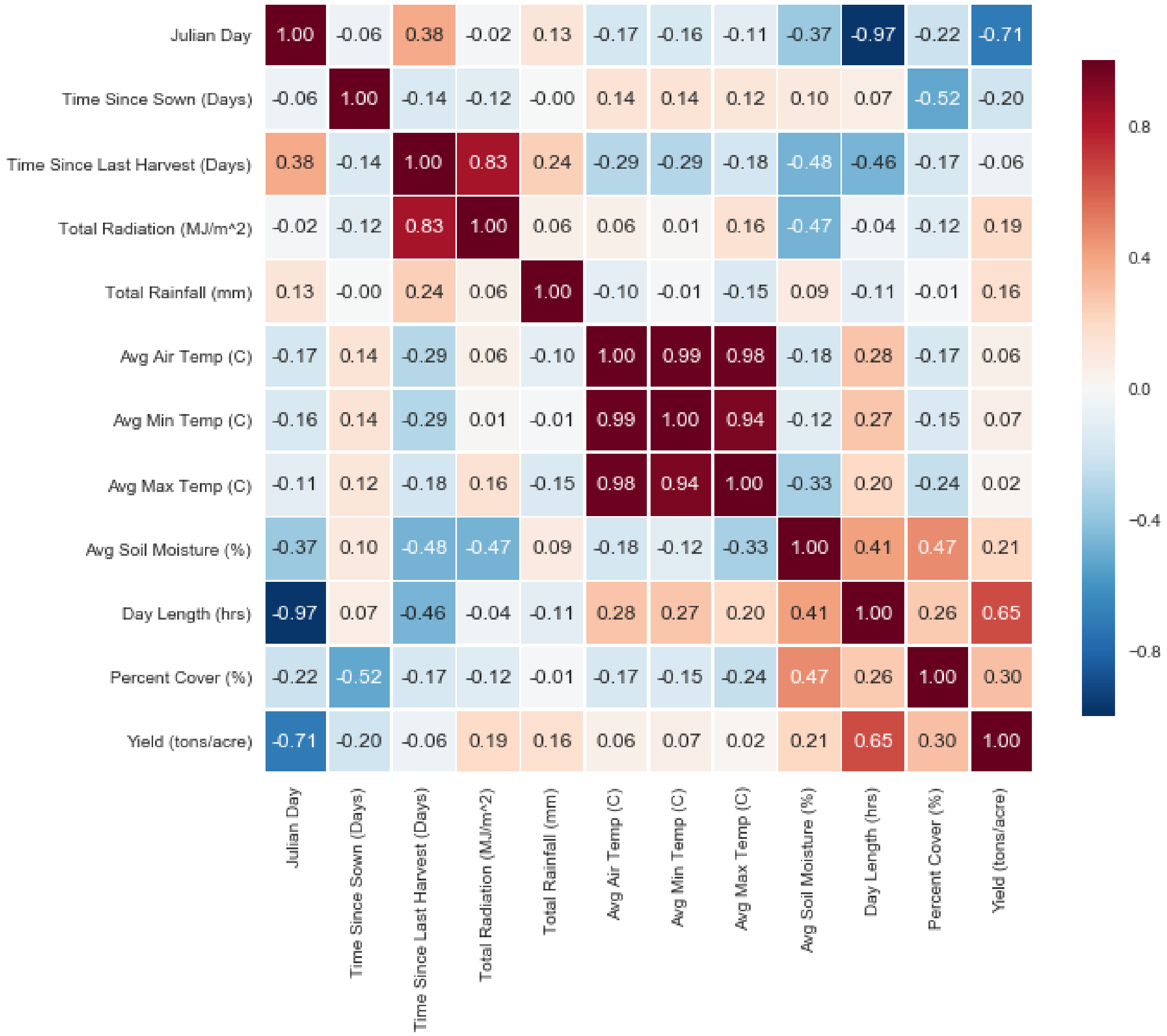
- 'n_iter':[100,300,500],
- 'lambda_1': [1.e-6, 1.e-4, 1.e-2, 1, 10],
- 'lambda_1': [1.e-6, 1.e-4, 1.e-2, 1, 10]

Linear Regression- no hyperparameters

APPENDIX C

CHAPTER 3 ADDITIONAL TABLES AND FIGURES

Each table is organized in descending order based on the model’s R value. Each result is given as ‘average result +/- 2σ’, where σ is the standard deviation.



	Julian Day	Time Since Sown (Days)	Time Since Last Harvest (Days)	Total Radiation (MJ/m^2)	Total Rainfall (mm)	Avg Air Temp (C)	Avg Min Temp (C)	Avg Max Temp (C)	Avg Soil Moisture (%)	Day Length (hrs)	Percent Cover (%)	Yield (tons/acre)
Julian Day	1.00	-0.06	0.38	-0.02	0.13	-0.17	-0.16	-0.11	-0.37	-0.97	-0.22	-0.71
Time Since Sown (Days)	-0.06	1.00	-0.14	-0.12	-0.00	0.14	0.14	0.12	0.10	0.07	-0.52	-0.20
Time Since Last Harvest (Days)	0.38	-0.14	1.00	0.83	0.24	-0.29	-0.29	-0.18	-0.48	-0.46	-0.17	-0.06
Total Radiation (MJ/m^2)	-0.02	-0.12	0.83	1.00	0.06	0.06	0.01	0.16	-0.47	-0.04	-0.12	0.19
Total Rainfall (mm)	0.13	-0.00	0.24	0.06	1.00	-0.10	-0.01	-0.15	0.09	-0.11	-0.01	0.16
Avg Air Temp (C)	-0.17	0.14	-0.29	0.06	-0.10	1.00	0.99	0.98	-0.18	0.28	-0.17	0.06
Avg Min Temp (C)	-0.16	0.14	-0.29	0.01	-0.01	0.99	1.00	0.94	-0.12	0.27	-0.15	0.07
Avg Max Temp (C)	-0.11	0.12	-0.18	0.16	-0.15	0.98	0.94	1.00	-0.33	0.20	-0.24	0.02
Avg Soil Moisture (%)	-0.37	0.10	-0.48	-0.47	0.09	-0.18	-0.12	-0.33	1.00	0.41	0.47	0.21
Day Length (hrs)	-0.97	0.07	-0.46	-0.04	-0.11	0.28	0.27	0.20	0.41	1.00	0.26	0.65
Percent Cover (%)	-0.22	-0.52	-0.17	-0.12	-0.01	-0.17	-0.15	-0.24	0.47	0.26	1.00	0.30
Yield (tons/acre)	-0.71	-0.20	-0.06	0.19	0.16	0.06	0.07	0.02	0.21	0.65	0.30	1.00

FIGURE C.1: Correlation heat map between features. A heat map showing the value of the correlation coefficient between each possible pair of features.

TABLE C.1: Results with no feature selection. The results from training the models with all possible features. Each result is shown as 'average results +/- 2σ', where σ is the standard deviation.

Model	Mean absolute error (lbs./acre)	R	R²
Support vector Machine	209.888 +/- 43.626	0.948 +/- 0.018	0.895 +/- 0.034
K-nearest neighbors	205.418 +/- 20.214	0.946 +/- 0.032	0.891 +/- 0.06
Random Forest	207.448 +/- 42.568	0.945 +/- 0.034	0.887 +/- 0.068
Neural network	232.937 +/- 49.23	0.937 +/- 0.036	0.873 +/- 0.068
Regression Tree	236.039 +/- 58.794	0.927 +/- 0.042	0.849 +/- 0.088
Linear Regression	358.454 +/- 80.506	0.818 +/- 0.094	0.664 +/- 0.15
Bayesian ridge Regression	357.686 +/- 67.776	0.818 +/- 0.07	0.663 +/- 0.11

TABLE C.2: Results from Cfs feature selection with all features. The results are from using the features Julian day, total solar radiation, total rainfall, and percent cover. Each result is shown as 'average results +/- 2σ', where σ is the standard deviation.

Model	Mean absolute error (lbs./acre)	R	R²
Random Forest	228.651 +/- 60.952	0.933 +/- 0.018	0.865 +/- 0.04
Support vector Machine	248.458 +/- 50.402	0.925 +/- 0.048	0.851 +/- 0.094
K-nearest neighbors	251.494 +/- 78.648	0.914 +/- 0.05	0.831 +/- 0.094
Regression Tree	272.247 +/- 87.004	0.9 +/- 0.106	0.8 +/- 0.192
Neural network	293.606 +/- 74.538	0.887 +/- 0.068	0.778 +/- 0.136
Linear Regression	382.928 +/- 91.962	0.792 +/- 0.104	0.627 +/- 0.164
Bayesian ridge Regression	383.459 +/- 73.826	0.79 +/- 0.096	0.619 +/- 0.162

TABLE C.3: Results from Cfs Feature Selection with no percent cover. The results from using the features Julian day, number of days since the sown date, total solar radiation, and total rainfall. Each result is shown as 'average results +/- 2σ', where σ is the standard deviation.

Model	Mean absolute error (lbs./acre)	R	R²
K-nearest neighbors	193.938 +/- 50.358	0.952 +/- 0.028	0.904 +/- 0.054
Random Forest	196.539 +/- 43.768	0.952 +/- 0.028	0.903 +/- 0.06
Regression Tree	200.052 +/- 39.43	0.95 +/- 0.026	0.899 +/- 0.056
Support vector Machine	231.222 +/- 60.97	0.936 +/- 0.032	0.871 +/- 0.064
Neural network	260.651 +/- 96.324	0.911 +/- 0.084	0.821 +/- 0.162
Bayesian ridge Regression	372.945 +/- 56.526	0.8 +/- 0.122	0.632 +/- 0.186
Linear Regression	372.547 +/- 51.848	0.798 +/- 0.096	0.632 +/- 0.166

TABLE C.4: Results from ReliefF Feature Selection. The results from using the features number of days since the sown date, total rainfall, and the average minimum temperature since the previous harvest. Each result is shown as 'average results +/- 2σ', where σ is the standard deviation.

Model	Mean absolute error (lbs./acre)	R	R²
K-nearest neighbors	195.86 +/- 44.704	0.953 +/- 0.018	0.905 +/- 0.038
Random Forest	197.026 +/- 49.294	0.95 +/- 0.038	0.9 +/- 0.076
Regression Tree	199.584 +/- 34.374	0.948 +/- 0.036	0.897 +/- 0.066
Neural network	357.532 +/- 133.118	0.842 +/- 0.146	0.7 +/- 0.234
Support vector Machine	344.604 +/- 104.482	0.83 +/- 0.128	0.688 +/- 0.206
Linear Regression	667.121 +/- 104.778	0.262 +/- 0.176	0.05 +/- 0.12
Bayesian ridge Regression	666.844 +/- 73.35	0.258 +/- 0.236	0.049 +/- 0.114

TABLE C.5: Results from Wrapper feature selection operator. The results from using the features number of days since the sown date, total rainfall, day length, and the Julian day. Each result is shown as 'average results +/- 2σ', where σ is the standard deviation.

Model	Mean absolute error (lbs./acre)	R	R²
K-nearest neighbors	199.28 +/- 69.822	0.952 +/- 0.026	0.904 +/- 0.052
Random Forest	197.782 +/- 51.598	0.952 +/- 0.024	0.903 +/- 0.044
Regression Tree	200.208 +/- 41.574	0.951 +/- 0.018	0.902 +/- 0.036
Support vector Machine	261.395 +/- 56.65	0.917 +/- 0.054	0.835 +/- 0.098
Neural network	300.245 +/- 84.178	0.883 +/- 0.088	0.776 +/- 0.156
Linear Regression	370.509 +/- 108.572	0.807 +/- 0.144	0.651 +/- 0.22
Bayesian ridge Regression	372.011 +/- 59.48	0.8 +/- 0.092	0.634 +/- 0.154