– Independent Work Report Spring 2019 –

Predicting Crop Yield and Disease Trends in Central New Jersey Farms

Lauren Johnston
Adviser: Barbara Engelhardt

Abstract

Specialized farming technology is on the rise but efforts to accurately predict crop yield and identify crop disease are still in their infancy. In this project, I aimed to find a solution to both problems with a caveat: I wanted to ensure that yield prediction was cost-effective. I studied tomato yield at three New Jersey farms and attempted to predict yield based off of common crop measurements like normalized difference vegetation index (NDVI), evapotranspiration, and growing degree days (GDD). I found that elastic net was the best regression method due to both its feature selection and accuracy ($R^2 = .39$). In the second phase of my experiment, in which I sought to distinguish different types of crop disease from drone images, my results indicated that further data collection was needed to create a more even distribution of labels and allow for better model training.

1. Introduction

In this project [1], I sought to accurately predict tomato yield using common crop health metrics like normalized difference vegetation index (NDVI), evapotranspiration, and growing degree days (GDD). I wanted to determine which metrics were most predictive in order to identify the minimum number of measurements that need to be made in order to accurately predict yield. In particular, I wanted to assess the accuracy of current GDD models and explore potential variations in the GDD formula. Furthermore, I pursued a secondary goal of detecting and classifying crop disease using a series of drone RGB and NIR images that I collected from farm fields. I wanted to develop a classification model that could distinguish between crop decay and animal damage.

[1] Parts of the text in this final project report are adapted from (and at times exactly the same as) passages that appear in Lauren Johnston, Spring Independent Work Proposal, 2019 [8]

Crop damage erodes farm economic viability and is often caused by diseases such as blight and mildew, animals, or natural decay at the end of the growing season. Figures 1 and 2 show the disparity between healthy and diseased crops. Diseased crops, such as the tomatoes shown in Figure 2, are difficult to sell and often farmers are forced to compost them or feed them to livestock instead of earning a profit. The New Jersey farmers that participated in this study cited crop damage as a major source of profit loss and anxiety during the growing season. If farmers were able to check an accurate summary of crop disease in their farm and farms in surrounding areas, they would be able to alter their planting dates or switch crops to avoid disease affecting a particular crop.

In addition, finding a formula for GDD that is an accurate predictor of yield would provide a solution to a major agriculture-technology problem. GDD is simple and inexpensive to measure using temperature, so a technology solution can easily be implemented in farms. If farmers are able to more accurately predict yield, they can proactively alter budgets and planting strategies during the growing season.

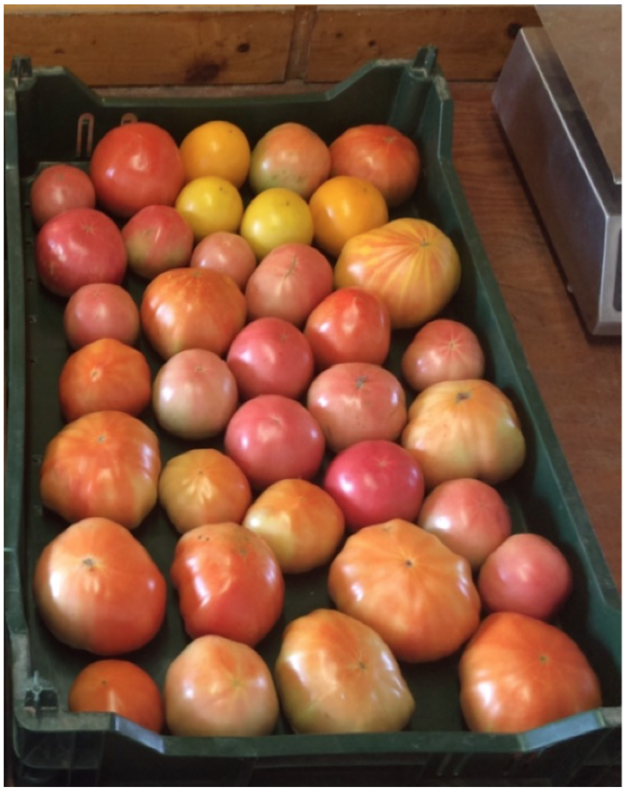

Figure 1: Healthy tomatoes (heirloom variety).

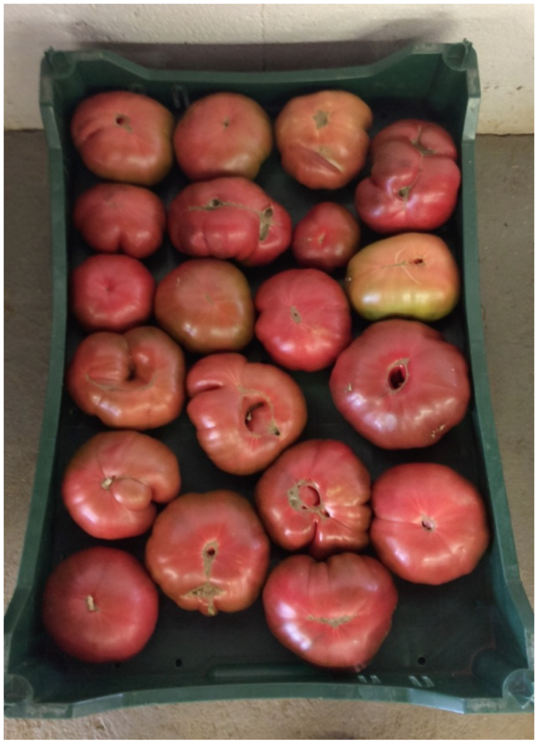

Figure 2: Diseased tomatoes (heirloom variety).

2. Problem Background and Related Work

2.1. Yield regression models

Crop evaluation metrics such as NDVI, evapotranspiration, and GDD were developed in the mid 20th century and remain relevant today in modified forms. NDVI was developed from a study in the Great Plains that used "multispectral remote sensing" to generate images of agricultural fields [18]. This study used supervised machine learning methods to classify vegetation cover including "boundary enhancement" by thresholding, "chromaticity transformation", and "automatic classification of ground cover types" [18]. The standard equation used for NDVI used currently is:

$$\frac{\alpha_{nir} - \alpha_{vis}}{\alpha_{nir} + \alpha_{vis}}$$

In this equation, α_{nir} is the mean "surface reflectance" over the IR spectrum and α_{vis} is the mean surface reflectance in the visual or "red" spectrum [2]. NDVI has been used to predict crop yield in wheat, corn, rice, and cotton [17] and recently tomato yield [9, 5]. Due to the limitations of NDVI described chiefly by Wang et al., including the "effect of canopy background" and other forms of noise, NDVI alone is often not a successful predictor for yield [19]. However, alongside evapotranspiration and GDD measures, an accurate model could potentially be built to predict yield.

A formula for evapotranspiration (the combination of evaporation and transpiration) was first derived from physics formulae by Penman in 1948 using parameters such as solar radiation, vapor pressure and temperature [15]. GDD, on the other hand, is a simple formula that takes mean daily temperature minimums and maximums and cumulatively calculates the number of viable growing days.

$$GDD = \frac{max(T) + min(T)}{2} - T_0$$

In the above equation for GDD, T_0 (the base temperature) is often 10 degrees Celsius [16]. According to McMaster and Wilheim, however, there are two diverging methods of calculating

GDD that yield different results [11]. The first involves setting T_0 to $mean(T)$ if $mean(T) < T_0$ and the second entails setting $max(T)$ and $min(T)$ to T_0 if $(max(T)$ or $min(T)) < T_0$ [11]. In this project, I will explore the efficacy of both calculations in regards to tomato yield prediction.

2.2. Crop damage classification models

In regards to computer vision techniques for classifying crop damage, there are two key models that have been used previously for agricultural image applications: random forest (RF) classification and convolutional neural networks (CNNs) [7, 4]. A previous study conducted by Huang et al. used CNNs to create maps of weed areas in crops [7]. This particular application could be extended to exclude weed areas of crop fields in order to focus on crop disease. On the other hand, a study by Castro et al. used a combination of object-based image analysis (OBIA) and RF classification to perform automatic weed mapping [4]. This study combined both RGB and Infrared data and therefore enhances the potential of analysis beyond the visual spectrum and allows for NDVI computation. In this project, I will endeavor to extend the previously attempted use of CNNs in weed mapping in order to classify crop disease.

3. Approach

3.1. Yield Prediction

For the tomato yield prediction phase of my project, I decided to collect NDVI, evapotranspiration, GDD measures, and other crop health measurements from local farms. These crop health measurements would act as descriptive features to predict yield, the target feature. To create a yield prediction model, I hoped to obtain cumulative tomato yield data from three different New Jersey farms collected over the same time interval to train the model. Training the model entails splitting the instances by day into a training and test set and then applying appropriate regression methods to the training set before testing on the test set.

I decided to attempt canonical machine learning regression methods such as ridge, Lasso, and elastic net. Given that these methods are known for their feature selection abilities, I hoped to use them to generate model coefficient sets that eliminated less predictive factors. Through analysis of the efficacy of the combined NDVI, evapotranspiration, and GDD features I endeavored to determine an improved method for forecasting crop yield. I believed this approach would be successful with robust data collection over the growing season, the placement of sensors to collect crop health metrics directly in tomato fields at each farm in order to directly link the data to the yield numbers, and the use of proven machine learning methods in order to create a sparse prediction model.

3.2. Spotlight Yield Regression Method: Elastic Net

One particular method that was critical to the approach of the tomato yield prediction phase of my project was elastic net. Elastic net was developed in 2003 by Zou and Hastie as a response to limitations in ordinary least squares (OLS) regression and related methods such as ridge and Lasso regression [21].

Both ridge and Lasso regression have benefits and pitfalls. Ridge regression was created to generate model coefficients with most coefficients being very small except for the most predictive [12]. It shrinks continuously by minimizing the residual sum of squares (RSS) in a squared term (L2-regularization). However, it does not perform effective subset selection because most coefficients are not reduced to zero. On the other hand, Lasso regression generates a sparse model coefficient set. Lasso uses the L1-norm or the absolute value of the RSS to generate coefficients [12]. There is a definite benefit to Lasso: it zeroes some coefficients and effectively eliminates them from the model. Unfortunately, however, it is known for arbitrarily selecting only one of two variables with high pairwise correlation [21]. Therefore, even a feature that is highly predictive may be eliminated due to its correlation with another variable. With these shortcomings in mind, Zou and Hastie saw the need to improve upon current linear regression techniques and develop elastic net regularization.

Elastic net regularization is defined in Zou and Hastie's paper [21] as:

$$L(\lambda_1, \lambda_2, \beta) = |y - X\beta|^2 + \lambda_2|\beta|^2 + \lambda_1|\beta|_1$$

In the above equation, y is the set of response variables and X is the model matrix with each X corresponding to a set of predictors. In naive elastic net, the equation is optimized with respect to β:

$$\hat{B} = argmin\{L(\lambda_1, lambda_2, \beta)\}$$

This can be simplified to the following equation, with a penalty:

$$\hat{B} = argmin\{|y - X\beta|^2\}, given : (1-\alpha)|\beta|_1 + \alpha|\beta|^2$$

In practice, the elastic net is solved in a two-part process by first performing L2-norm (ridge) regularization and then L1-norm (Lasso) shrinkage. This leads to the desired sparse coefficient set (a property of Lasso) while also balancing bias and variance (a property of ridge regression). Furthermore, elastic net improves on both methods because it exhibits the grouping effect [21]. The grouping effect is a property that means that given closely-correlated variables in a feature set, variables will be consistently assigned similar (and same-signed) coefficients. This effect results in a consistent set of features, a benefit that is highly desirable in this application because many features in the Arable data set are closely correlated. Figure 3, taken directly from Zou and Hastie, explains why the grouping effect occurs. It shows the total convexity of the elastic net solution in relation to ridge and Lasso's.

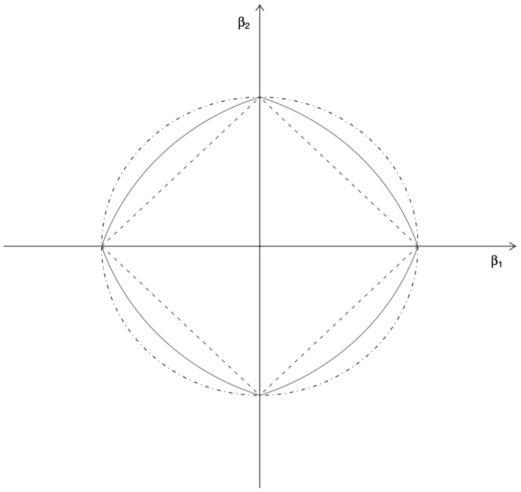

Figure 3: "Two-dimensional contour plots (level 1) (- . - . -, shape of the ridge penalty; - - - - -, contour of the lasso penalty; ——-, contour of the elastic net penalty with $\alpha = 0.5$): we see that singularities at the vertices and the edges are strictly convex; the strength of convexity varies with α" [21]

3.3. Crop Damage Classification

With regards to the crop damage classification phase of the project, I decided to collect drone image (RGB and Infrared) data using a Sentera-modified drone over summer 2018. Given Huang's study that employed CNNs to generate maps of weed patterns, I decided to use one of the CNNs that was important to the study: AlexNet [7].

3.4. Spotlight Image Classification Method: AlexNet CNN

For the crop disease classification phase of my project, the AlexNet CNN model was the most appropriate for the task due to its success in unsupervised image classification tasks [10]. AlexNet, invented by Krizhevsky et al. in 2012, was initially trained on the LSVRC-2010 ImageNet photo set and was both efficient and low in error.

AlexNet's architecture built upon previous CNN designs while adding certain unique modifications. The AlexNet network consists of eight layers as shown in Figure 4 below, taken directly from Krizhevsky et al. [10].

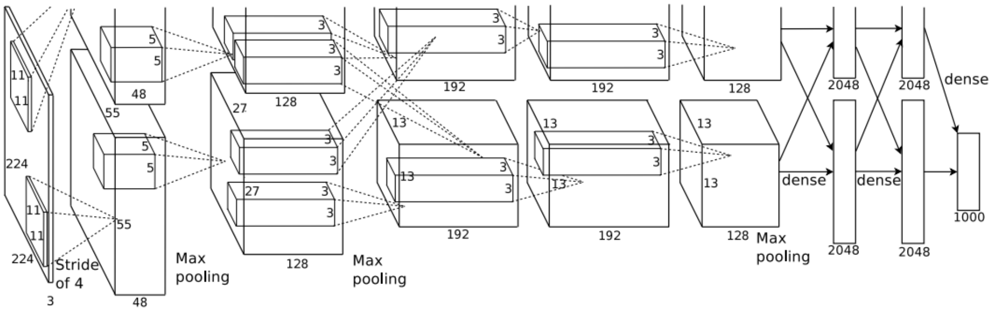

Figure 4: AlexNet 8-layer architecture.

The five leftmost layers in the architecture figure are convolutional layers while the final three on the right are fully-connected. The convolutional layers, computed in parallel, generate a series of linear activations [6]. Then, AlexNet's rectified linear activation function (ReLU) processes the linear activations. Finally, these activations are connected by the max pooling function. Max pooling is a function designed by Zhou and Chellappa in 1988 that maximizes the output in a kernel region by selecting the maximum value [20]. In early max pooling functions, kernel regions were covered non-overlapping portions of the input. Meanwhile, in AlexNet Zhou and Chellappa were able to reduce overfitting-caused error by introducing overlap to max pooling applied on the kernel regions. A combination of architecture and modification of functions led to AlexNet's success at image classification tasks. Furthermore, as Huang et al. proved in 2018, AlexNet is useful in agriculture applications including weed mapping [7].

4. Implementation

4.1. Yield Prediction

4.1.1. Data Collection Only 3 out of 7 farms that participated in the Farm Project conducted in the Rubenstein Lab at Princeton's Ecology and Evolutionary Biology Department grew standard size tomatoes. As a result, I chose those three farms to compare tomato yield numbers and crop health metrics (Table 1). In order to measure crop health metrics such as NDVI, evapotranspiration, and GDD, I helped the Farm Project team install Arable Mark sensors at the three farms.

Farm	Organic	Arable collection dates	Sale Method	Notes
Cherry Grove	Yes	6/6 to 9/22	CSA membership	Composting + crop rotation.
Honey Brook	Yes	6/8 to 9/19	CSA, boxed shares.	Tomatoes covered by tunnels.
Kerr's Korn Stand	No	6/7 to 9/24	Road-side stand	Compost + synthetic fertilizer.

Table 1: Farms chosen for comparison in tomato yield prediction.

4.1.2. Data Collection I completed the data collection phase of this project from June to October 2018. Using Arable Mark sensors, I collected information on temperature, evapotranspiration, solar radiation, GDD, NDVI, rainfall, and more [2]. Arable calculates GDD using the standard equation [16] and computes NDVI using the Penman equation [1].

Furthermore, I collected tomato sales data from farmers and measurements on the average tomato weight and circumference per box sold. From these measurements, I calculated the total daily yield in pounds per day.

4.1.3. Models I chose five methods of linear regression for the yield prediction phase of my project: ridge, Lasso, Lasso with cross validation, Lasso computed via LARS, and elastic net. I used the Python Scikit-learn implementation for all 5 methods [14]:

- ***Ridge regression:*** Ridge(alpha=.5)
- ***Lasso regression:*** Lasso(alpha = 0.01)
- ***Lasso with 10 rounds of cross validation:*** LassoCV(cv=10, random_state=0)
- ***Lasso computed via LARS:*** LassoLars(alpha=0.01)
- ***Elastic net:*** ElasticNet(alpha=0.01, l1_ratio=0.7)

As discussed in the Related Work section, I chose elastic net because it generates a sparse model (thus performing feature selection) and also balances this sparsity with the bias-variance tradeoff. I wanted to compare elastic net with forms of Lasso, which also generates sparse coefficients, for accuracy. However, I knew that Lasso was not well-suited to handling features with high pairwise correlation. Thus, I included ridge regression to compare sparse models with non-sparse models for accuracy.

For ridge regression, I chose 0.5 alpha (default is the max, 1.0) in order to perform regularization

[2] For a full list of Arable data features, go to: https://www.arable.com/solutions_weather/

with moderate strength. As for Lasso and elastic net I chose a 0.01 alpha through trial and error. Higher values for alpha yielded solutions that were too sparse in both cases.

In addition to linear regression methods, I performed principle component analysis (PCA) as an alternative method of feature selection. I calculated PCA with 1 to 10 components and compared the explained variance ratio for each in order to find an ideal drop-off point. At the ideal drop-off point, feature selection can occur with minimal reduction in prediction accuracy. I used the Sci-kit learn implementation of PCA with all default parameters except for the number of components (10) [14].

4.2. Crop Damage Classification

4.2.1. Data Collection In this phase of the project, I collected aerial images (both RGB and NIR) using a Sentera-modified drone during summer 2018. These images were collected at 400ft altitude using the Digital Surface Model (DSM)/Ortho setting with 80% overlap. In addition, the drone images were taken on different dates with varying amounts of light and on different crops, which increased the amount of variation of the dataset. In total, 913 images were collected on 11 separate occasions. The fields that I flew over were corn and soybean primarily due to the near-total crop coverage of both types of fields from above. Choosing corn and soybeans eliminated the task of separating crops row by row so as not to include bare ground in prediction.

4.2.2. Models I used the Pytorch implementation of AlexNet and Pytorch dataloader documentation in order to classify crop disease in my image set [13]. Instead of using the trained model of Alexnet, I used the untrained version because I wanted the ability to train the network on the class labels. I designed a custom Pytorch dataloader for the drone dataset.

The parameters I chose for the Alexnet model are as follows:

```
torchvision.models.alexnet(pretrained=False, num_classes=6)
```

In the next section, I will discuss the specifics of the AlexNet model training and evaluation process.

5. Evaluation

5.1. Yield Prediction

5.1.1. Data pre-processing The first step in pre-processing yield data was calculating the weekly yield total and distributing it evenly over each day of the week. This step is essential due to the fact that certain days are more productive then others because some days have more rigorous harvest schedules or better weather for picking. Therefore, it made the most sense to eliminate day-to-day fluctuation by allowing each day to be smoothed according to the weekly yield. I accomplished this smoothing by dividing the measured total weekly weight picked by the number of days. Next, I normalized the yield for each farm using the formula (where w is the total weight picked that day and W is the set of all weights):

$$\frac{w - min(W)}{max(W) - min(W)}$$

This normalization was necessary due to the variability between farms in terms of scale of operation and thus total yield. However, normalizing does make it difficult to compare between farm and assess the impact of individual farming methods. It would have been preferable to divide yield by total area, but unfortunately the Farm Project was not able to obtain this data for all three farms.

Another important pre-processing step was removing "time", "location", "device", "latitude", "longitude", and "rainfall". All except for rainfall were removed before regression modeling due to the small number of farms that would not produce meaningful location comparison. Rainfall, on the other hand, was removed because Honey Brook Chesterfield tomatoes were under cover, in a hot house. This meant that rain measures would not be viable for cross-farm comparison.

5.2. Crop Damage Classification

5.2.1. Data labelling In order to classify the drone images, I decided to manually generate multi-label classification sets for each image. I chose to specify six unique target labels: animal damage, disease, decay, trees, bare ground, and road. Each of these was a binary (1 or 0) for presence in the image or lack thereof. The images below (Figures 5 and 6) show labels on two sample images from the drone RGB image set. The labels themselves are binary rather than region-dependent, but I overlayed the classes on the image where they occur for the purpose of demonstrating classification decisions.

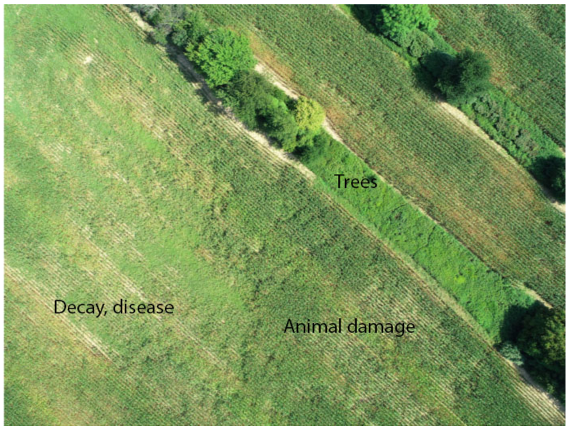

Figure 5: Sample RGB drone image with decay, disease, trees, and animal damage present.

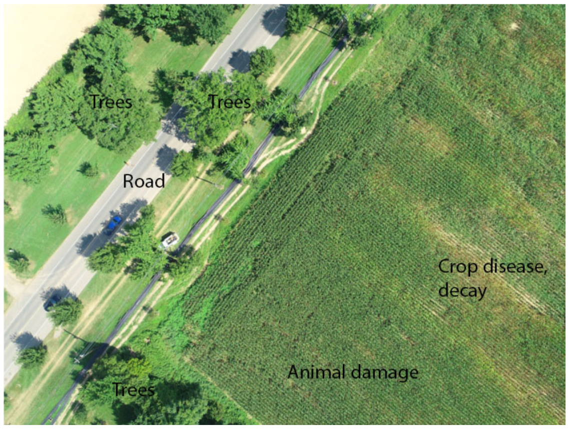

Figure 6: Sample RGB drone image with road, decay, disease, trees, animal damage present.

5.2.2. Data pre-processing Each drone image was subjected to a series of transformations during the loading process. First, I resized each image to 256 pixels along one side. Then, I performed a center crop so that the resulting image was 256 x 256 pixels. This resizing and center-crop were originally employed by Krizhevsky et al. [10]. Together, the methods help improve performance due to the dimension reduction of model inputs. Finally, I transformed the image into a tensor and then normalized it to a mean of 0.5 and a standard deviation of 0.5. This transform, implemented in Pytorch, works as follows (X is the image, μ is the mean and σ is the standard deviation):

$$X' = \frac{X - \mu}{\sigma}$$

The resulting image is in the range [-1,1]. This new, reduced image range diminishes skew in the data and minimizes bias in the model.

5.2.3. Experiment design First of all, I randomly split all images and labels into a 70/30 train-test split. I used the random_split function in Pytorch [13] to perform random subset sampling without replacement. Next, I trained the Alexnet model using gradient descent with batches of 25 images each, randomly shuffled. For the gradient descent loss function, I used binary cross entropy loss, which has independent vectors, to update model weights. Finally, I inputted the test data into the trained model and compared the labels with the classification predictions.

6. Results

6.1. Yield Prediction

By fitting yield regression models with normalized tomato yield and Arable sensor data, I was able to generate useful subsets of descriptive features from the Arable data. I found that cumulative GDD was by far the most predictive feature for yield, with coefficients ranging from 0.347 in Lasso with LARS to 0.920 in Lasso with cross validation. This is unsurprising given the high correlation between yield and cumulative growing degree days, as shown in Figure 10. Furthermore, growing degree days essentially measures how productive a day is with regards to temperature and plant maturation, so it is natural that it is highly predictive of yield. However, it is surprising that NDVI is completely eliminated as a feature in Lasso, Lasso LARS, and elastic net given previous research to indicate its ability to predict yield. This disparity could be due to a genuine lack of usefulness as a yield predictor in tomatoes or perhaps due to inaccuracies in the measurements made by the Arable sensor.

NDVI may be less useful as a yield predictor in tomatoes due to the uneven cover of tomatoes on the ground. In some farms, there are large (1-2ft.) gaps between plants and in others the tomato plants may be so overgrown that no bare ground can be seen. Depending on where NDVI (which depends on mean surface reflectance) is measured, there could be a lot of variability that does not reflect the health of the plant but rather the spacing.

Following GDD, chlorophyll index (CI) was typically the second highest coefficient across the models. Chlorophyll index is an indication of the amount of chlorophyll in a plant's leaves, so this correlation is not surprising [3]. It is calculated from down-welling spectrometer bands, meaning that if chlorophyll index were included in the final model for yield prediction, a spectrometer would need to be used for any further calculations [3]. As NDVI also requires a spectrometer, it would be preferable to either include or eliminate both.

Feature	**Ridge**	**Lasso**	**Lasso CV**	**Lasso LARS**	**Elastic net**
GDD (cumulative)	0.885973	0.629643	0.920113	0.347498	0.650474
Chlorophyll index	0.242645	0.127821	0.221249	0	0.179403
Evapotranspiration	0.029076	0	0	0	0
GDD for day	-0.045835	0	0	0	0
LfAirDelta	-0.113918	0	-0.040104	0	0
NDVI	0.107615	0	0.254656	0	0
shortwave DW radiation	-0.03192	0	0	0	0
maxT	-0.121239	0	0	0	0
meanT	0.113613	0	0.004334	0	0
minT	0.131308	0	0.112843	0	0.015313
sea level pressure	-0.139502	-0.117789	-0.113361	0	-0.125171
crop coefficient Kc	0.107623	0	0	0	0
crop evapotranspiration	0.044176	0	0	0	0

Table 2: Regression coefficients for Arable data features by model.

I found that Lasso regression with cross validation yielded the most accurate results with an R^2 score of 0.417 (Table 3). Meanwhile, ridge regression was a close second at 0.403 and elastic net was third at 0.390. The success of Lasso with cross validation is intuitive because cross validation prevents over-fitting on the training set. With 10 rounds of cross validation, the model coefficients are less likely to be susceptible to noise in the training data.

Model	R^2 **Score**	**Time (train)**	**Time (predict)**
Ridge	0.403	0.0016	0.0002
Lasso	0.386	0.0013	0.0002
Lasso CV=10	0.417	0.1156	0.0001
LARS Lasso	0.218	0.0038	0.0002
Elastic Net	0.390	0.0012	0.0007

Table 3: Accuracy and speed tests for yield regression models (with normalized yield).

As shown in Figure 7, ridge regression has the most non-zero coefficients. This is to be expected because it doesn't have the property of reducing coefficients to zero like Lasso and elastic net.

Interestingly, Lasso computed via LARS only has a single non-zero coefficient (growing degree days). This extreme feature selection hurt the R^2 score which, at 0.218, is the lowest of all models tested.

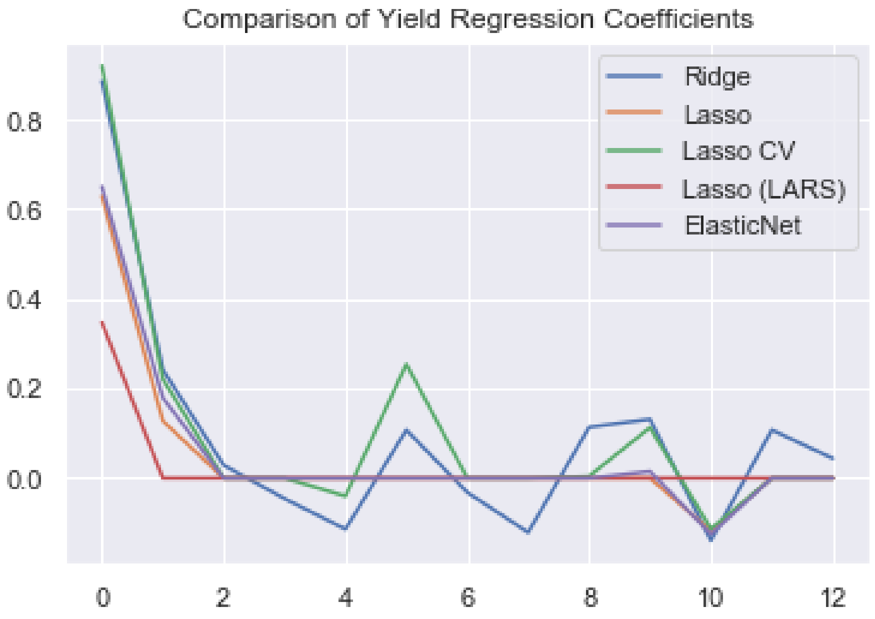

Figure 7: Yield regression coefficients across all models.

As an extension on the descriptive features provided by Arable, I calculated GDD using the two popular methods described in McMaster and Wilheim [11]. These methods clamp the minimum/maximum and the mean respectively. Table 4 shows the results of running regression models with alternative calculations of GDD included. Surprisingly, the results reaffirm that GDD with no clamping is the most successful predictor of yield. The cumulative GDD Max/Min Clamp has a slightly higher coefficient than GDD Mean Clamp (Cumulative) in most models. However, cumulative GDD ranges from 0.40-.85 while the two clamping GDD methods had a 0.27 coefficient at maximum.

GDD Coefficient	Ridge	Lasso	Lasso CV	Lasso LARS	Elastic Net
GDD Max/Min Clamp	-0.021	0	0	0	0
GDD Mean Clamp	-0.021	0	0	0	0
GDD Max/Min Clamp (Cumulative)	0.224	0.266	0.253	0	0.148
GDD Mean Clamp (Cumulative)	0.224	0.000	0.000	0.048	0.146
GDD (Cumulative)	0.765	0.618	0.847	0.400	0.607
GDD	-0.021	0	0	0	0

Table 4: GDD coefficients by method of GDD calculation.

In addition to linear regression methods, I performed principle component analysis (PCA) in search of a method to perform feature selection. Figure 8, shown below, displays the results. There is a promising drop off between 3-4 components that suggest 3-4 components would be an ideal number to select in the final model. However, when I used the PCA output with 3-4 components to train regression models, I noticed a significant decrease in the performance. For ridge regression, for example, the R^2 value went from 0.38 at 10 components to 0.13 at 3 components. Therefore, I ultimately decided against using PCA in my yield prediction model.

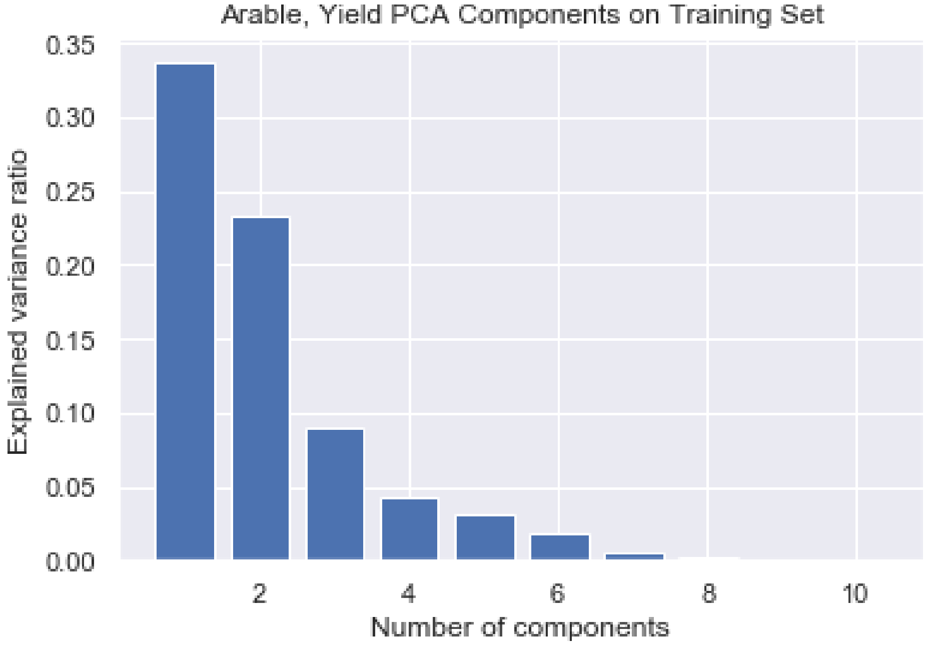

Figure 8: Explained variance by number of PCA components.

6.2. Crop Damage Classification

In the crop damage classification portion of my experiment, I found that error did not significantly decrease over training epochs with a learning rate of $1e^{-3}$. Table 5 shows this slow progression in loss decrease that seems to be nearly stagnant from the beginning. However, I tested the trained model on a held-out test set and found a 56.1% accuracy across class labels. In order to calculate the accuracy, I calculated the sum of all correctly predicted individual labels divided by the total number of labels. Given the lack of improvement over epochs, this is a surprisingly high result.

Epoch	Average BCELoss	Time (s)	Epoch	Average BCELoss	Time (s)
1	0.693	13.68	14	0.692	14.37
2	0.693	13.6	15	0.692	13.59
3	0.693	13.6	16	0.692	16.52
4	0.693	13.63	17	0.692	14.62
5	0.693	13.75	18	0.691	14.7
6	0.692	13.52	19	0.691	15.71
7	0.692	13.58	20	0.691	15.92
8	0.692	13.56	21	0.691	14.37
9	0.692	13.53	22	0.691	14.49
10	0.692	13.51	23	0.691	14.82
11	0.692	13.61	24	0.691	14.58
12	0.692	13.52	25	0.691	14.95

Table 5: Binary cross entropy (BCE) loss over training epochs.

Table 5 also shows the average speed of each batch in an epoch. The roughly 15 seconds per epoch is to be expected and would be improved on a machine that allows Pytorch GPU functionality.

7. Summary

From the tomato yield prediction phase of my experiment, I determined that Lasso regression with cross validation, ridge regression, and elastic net regression generate the most accurate yield models. Of these three, Lasso with cross validation and elastic net are the most promising due to the sparsity of their solutions. However, given that the highest R^2 score was 0.417, there likely needs to be further data collection and exploration of new crop health metrics to create a more accurate yield prediction method. These results, despite their limitations, have still generated interesting feature subsets that may be used to reduce farming technology in future. In particular, Table 2 indicates that cumulative GDD and Chlorophyll index are important while NDVI eliminated in three out of five models. Furthermore, by testing two alternative clamping methods of GDD I was able to show that with the current data it is unclear if there is any benefit to clamping GDD methods over traditional GDD calculations.

With regards to the crop disease classification portion of my experiment, I found that Alexnet had a prediction accuracy of 56.1% across all classes. I concluded that this was likely due to the uneven distribution of class labels in the training data. I also found that there was no significant decrease in BCE loss over epochs during model training.

Given the opportunity to further develop this project, I would consider performing image segmentation before training my model with Alexnet. Furthermore, I would make use of the NIR images collected and add them as an additional input dimension to the model. This could potentially act as well as boundary segmentation due to the high contrast of NIR that showcase areas of healthy vs. unhealthy crops. Additionally, I would reconsider the class labels that I chose for the model. It was difficult to differentiate between crop disease and crop decay, so often I simply ended up classifying images as both. Part of the issue in differentiating classes was due to the fact that images were taken from the high altitude of 400ft. If I were to collect more data, I could consider flying at a much lower altitude (around 100-200ft.) so that images contained only 1 or 2 classes rather than most of them.

Acknowledgments

I would like to thank Professor Engelhardt for her guidance as well as Greg Gundersen for his help understanding CNNs and Pytorch. Furthermore, the research for this project was conducted as part of the Farm Project at the Rubenstein Lab in the Ecology and Evolutionary Biology (EEB) department. Many thanks to Professor Daniel Rubenstein, Gina Talt (Lab Fellow), Mina Arashloo (Graduate student), Axel Haenssen (OIT drone specialist), and the EEB students who worked on this project over the summer.

Honor Code

This paper represents my own work in accordance with university regulations.

/s/ Lauren Johnston

References

[1] Arable, "Exploring evapotranspiration," Feb 2019. [Online]. Available: https://www.arable.com/2018/04/06/exploring-evapotranspiration/

[2] T. N. Carlson and D. A. Ripley, "On the relation between NDVI, fractional vegetation cover, and leaf area index," *Remote sensing of Environment*, vol. 62, no. 3, pp. 241–252, 1997.

[3] J. Dash and P. Curran, "The meris terrestrial chlorophyll index," 2004.

[4] I. de Castro, Ana, J. Torres-Sanchez, M. Peña, Jose, F. Jimenez-Brenes, O. Csillik, and F. López-Granados, "An automatic random forest-OBIA algorithm for early weed mapping between and within crop rows using uav imagery," *Remote Sensing*, vol. 10, no. 2, p. 285, 2018. [Online]. Available: https://search-proquest-com.ezproxy.princeton.edu/docview/2014750644?accountid=13314

[5] R. Fortes, M. H. Prieto, J. M. Terrón, J. Blanco, S. Millán, and C. Campillo, "Using apparent electric conductivity and NDVI measurements for yield estimation of processing tomato crop," *Transactions of the ASABE*, vol. 57, no. 3, pp. 827–835, 2014.

[6] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. MIT Press, 2016, http://www.deeplearningbook.org.

[7] H. Huang, J. Deng, Y. Lan, A. Yang, X. Deng, and L. Zhang, "A fully convolutional network for weed mapping of unmanned aerial vehicle (UAV) imagery," *PLoS One*, vol. 13, no. 4, 04 2018. [Online]. Available: https://search-proquest-com.ezproxy.princeton.edu/docview/2031413122?accountid=13314

[8] L. Johnston, "Predicting crop yield and disease trends in Central New Jersey farms," Feb 2019.

[9] M. Koller and S. Upadhyaya, "Prediction of processing tomato yield using a crop growth model and remotely sensed aerial images," *Transactions of the ASAE*, vol. 48, no. 6, pp. 2335–2341, 2005.

[10] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in Neural Information Processing Systems 25*, F. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger, Eds. Curran Associates, Inc., 2012, pp. 1097–1105. [Online]. Available: http://papers.nips.cc/paper/4824-imagenet-classification-with-deep-convolutional-neural-networks.pdf

[11] G. S. McMaster and W. Wilhelm, "Growing degree-days: one equation, two interpretations," *Agricultural and forest meteorology*, vol. 87, no. 4, pp. 291–300, 1997.

[12] K. P. Murphy, *Machine Learning : A Probabilistic Perspective*. Cambridge: MIT Press, 2012. [Online]. Available: http://ebookcentral.proquest.com/lib/princeton/detail.action?docID=3339490

[13] A. Paszke, S. Gross, S. Chintala, G. Chanan, E. Yang, Z. DeVito, Z. Lin, A. Desmaison, L. Antiga, and A. Lerer, "Automatic differentiation in pytorch," in *NIPS-W*, 2017.

[14] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay, "Scikit-learn: Machine learning in Python," *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.

[15] H. L. Penman, "Natural evaporation from open water, bare soil and grass," *Proceedings of the Royal Society of London. Series A. Mathematical and Physical Sciences*, vol. 193, no. 1032, pp. 120–145, 1948.

[16] I. C. Prentice, W. Cramer, S. P. Harrison, R. Leemans, R. A. Monserud, and A. M. Solomon, "Special paper: a global biome model based on plant physiology and dominance, soil properties and climate," *Journal of biogeography*, pp. 117–134, 1992.

[17] N. Quarmby, M. Milnes, T. Hindle, and N. Silleos, "The use of multi-temporal NDVI measurements from AVHRR data for crop yield estimation and prediction," *International Journal of Remote Sensing*, vol. 14, no. 2, pp. 199–210, 1993.

[18] J. Rouse Jr, R. Haas, J. Schell, and D. Deering, "Monitoring vegetation systems in the great plains with ERTS," pp. 1,29–31, 1974.

[19] Z. Wang, C. Liu, and A. Huete, "From AVHRR-NDVI to MODIS-EVI: Advances in vegetation index research," *Acta ecologica sinica*, vol. 23, no. 5, pp. 979–987, 2003.

[20] Y. Zhou and R. Chellappa, "Computation of optical flow using a neural network," 08 1988, pp. 71 – 78 vol.2.

[21] H. Zou and T. Hastie, "Regularization and variable selection via the elastic net," *Journal of the royal statistical society: series B (statistical methodology)*, vol. 67, no. 2, pp. 301–320, 2005.

A. Appendix

```
AlexNet(
  (features): Sequential(
    (0): Conv2d(3, 64, kernel_size=(11, 11), stride=(4, 4), padding=(2, 2))
    (1): ReLU(inplace)
    (2): MaxPool2d(kernel_size=3, stride=2, padding=0,
    dilation=1, ceil_mode=False)
    (3): Conv2d(64, 192, kernel_size=(5, 5), stride=(1, 1), padding=(2, 2))
    (4): ReLU(inplace)
    (5): MaxPool2d(kernel_size=3, stride=2, padding=0,
    dilation=1, ceil_mode=False)
    (6): Conv2d(192, 384, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1))
    (7): ReLU(inplace)
    (8): Conv2d(384, 256, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1))
    (9): ReLU(inplace)
    (10): Conv2d(256, 256, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1))
    (11): ReLU(inplace)
    (12): MaxPool2d(kernel_size=3, stride=2, padding=0,
    dilation=1, ceil_mode=False)
  )
  (avgpool): AdaptiveAvgPool2d(output_size=(6, 6))
  (classifier): Sequential(
    (0): Dropout(p=0.5)
    (1): Linear(in_features=9216, out_features=4096, bias=True)
    (2): ReLU(inplace)
    (3): Dropout(p=0.5)
    (4): Linear(in_features=4096, out_features=4096, bias=True)
    (5): ReLU(inplace)
    (6): Linear(in_features=4096, out_features=6, bias=True)
  )
)
```

Figure 9: Pytorch Alexnet architecture, tailored to 6 class labels.

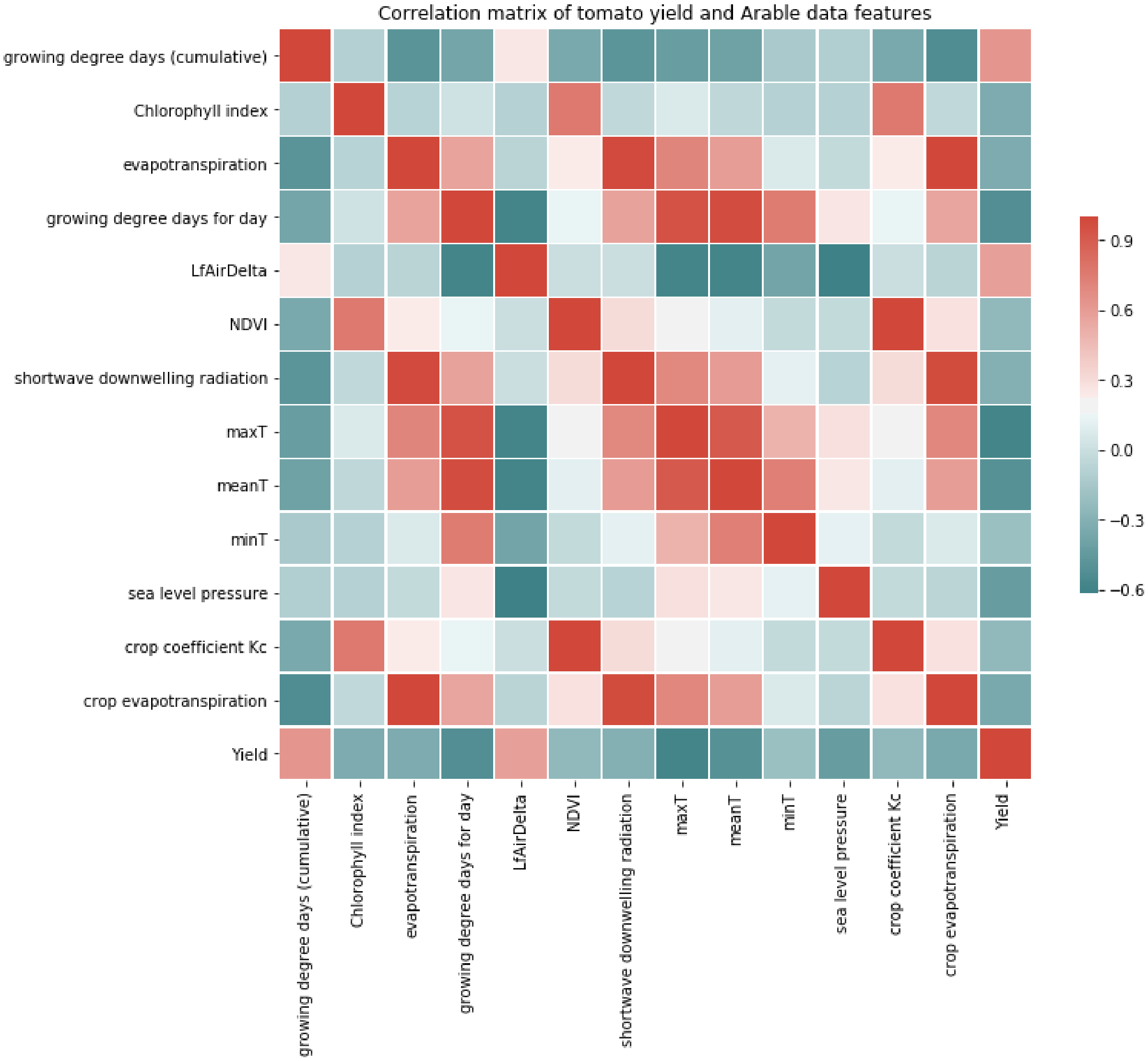

Figure 10: Correlation matrix of Arable sensor data and yield