FarmView: Regression Analysis of 2016 Sorghum Composition

Ben Parr
bparr@cmu.edu
Machine Learning Department
Carnegie Mellon University

Background. FarmView is a joint project between Carnegie Mellon University, Clemson University and Near Earth Autonomy whose goal is to increase the bioenergy yield of sorghum, a drought and heat tolerant grass with a diverse gene pool. Clemson University harvested a field of sorghum in 2016 which had a ground robot and an aerial drone gather data as the sorghum grew.

Aim. I determined the predictive power of the 2016 ground robot, aerial drone, harvest, accession and field location data in predicting the composition of the harvested sorghum.

Data. There are 698 subplots in the sorghum field with known composition. Each subplot also has corresponding accession features (such as the country of origin), GPS field location, robot-generated features, drone-generated features, harvest phenotypes and composition features. The dataset has many missing values, with more than half of the samples missing at least 24% of the input features.

Methods. I trained a random forest, and other regressors, on the 2016 dataset using 10-fold cross validation. I augmented the training set so there were more training samples with missingness identical to the missingness in the test set. Finally, I trained on three different views of the dataset to determine the incremental increase in predictive power from the robot-generated plus drone-generated features, as well as the harvest phenotypes.

Results The random forest regressor preformed the best with an overall r^2 score of 0.436. This includes the 2.5% increase in r^2 score from my missing value augmentation

Conclusions. The ground robot and aerial drone features improved predictive power of the composition features. This predictive power should also increase in the upcoming sorghum fields.

Intellectual merit. The 2016 dataset only included 698 samples, each with 29 input features that have a significant amount of missing values. The relatively small and sparse dataset allowed for an extensive regressor search, combined with a new missing value augmentation.

Broader impacts. Bioenergy sorghum breeders expect to double yield in the next five years alone. The Global Harvest Initiative reported that by 2050, the demand for food will outpace production if food productivity does not advance. FarmView can help bioenergy sorghum breeders reach their expectations, and also help increase food productivity by translating the developed science and technology to food crops.

Keywords. bioenergy, sorghum, phenotyping robot

1 Introduction

Sorghum bicolor has become a preferred bioenergy candidate because it is drought-tolerant and highly productive. It is an adaptable grass with a diverse gene pool containing over 40,000 genetic varieties. Production of sorghum is rising, having grown by 66% in the last 50 years. Sorghum breeders expect to double yield in the next five years alone (Schmitmeyer (2016)).

FarmView is a joint project between Carnegie Mellon University, Clemson University and Near Earth Autonomy. In 2016, biologists at Clemson University cultivated a field of sorghum in the clay soil of Pendleton, South Carolina. Carnegie Mellon University created a ground robot that is able to traverse up and down the rows of sorghum. Near Earth Autonomy created an aerial drone that also gathered data about the growing sorghum. Clemson University harvested the matured sorghum, and then gathered phenotypic data about the plants, such as the weight of the harvested plants. Finally, Clemson University conducted extensive composition experiments on the harvested sorghum.

FarmView's goals are multifaceted. From the biology side, one goal is to discover which factors affect sorghum bioenergy composition and yield, and exploit them during the breeding process to create higher yielding varieties. From the robotics side, one goal is to develop affordable robotic systems for specialty crop management. For sorghum, these robots gather data as the sorghum grows and improve composition predictions, which could be used when determining which sorghum plants to breed. The FarmView efforts have only just begun, with the ground robot and aerial drone producing their first features for the 2016 sorghum field.

Finally, the Global Harvest Initiative reported that by 2050, the demand for food will outpace production if food productivity does not advance (Koba (2014)). The science and technology developed as part of the FarmView project could be translated to food crops, and help resolve the forecasted food shortage.

Figure 1: Ground robot created by Carnegie Mellon University, and aerial drone created by Near Earth Autonomy. Credit: Saswati Raw from Carnegie Mellon's AutonLab, and Paul Bartlett from Near Earth Autonomy, respectively.

2 Problem Statement

A primary goal of the FarmView project is to use the ground robot and aerial drone to predict the composition of harvested sorghum. This paper analyzes the composition predictive power of these robot-generated and drone-generated features, as well as the predictive power of the available accession, field location and harvest features.

3 Related Work

Machine learning algorithms have already started to shape agricultural research. Heckmann et al. (2017) published results this year on predicting photosynthetic capacity of cabbage and corn using the reflectance spectra of mature leaves. The best performing predictor was a combination of first performing recursive feature elimination on raw spectra data, and then least squares regression to make the final photosynthetic capacity predictions. Unlike this cabbage and corn project, the FarmView project has analyzed a non-food crop. Sorghum does not compete with food crop such as corn and cabbage, because sorghum can grow in harsher ground. Also, the 2016 sorghum dataset does not include raw spectra values, since the Carnegie Mellon Robotics team and Clemson biologists have already processed the raw data.

The FarmView biologists at Clemson also analyzed sorghum grown in Florence, South Carolina in 2013 and 2014 (Brenton et al. (2016)). Instead of using phenotypic features to predict composition, the Clemson biologists performed genome-wide association analysis by first creating and characterizing an association panel of the 390 diverse sorghum varieties with 232,303 genetic markers. They identified genes that are potentially associated with increased non-fibrous carbohydrates (NFC). If the genes are experimentally verified to cause an increase in NFC, sorghum breeders could then breed for those genes and thus increase sorghum bioenergy yield. Given the long-term nature of genetic experiments, and contrasting with the short-term nature of a Data Analysis Project, this paper does not use the genetic dataset created by the Clemson biologists. Instead, this paper uses the newly available robot-generated, drone-generated and harvest phenotype features.

Saswati Ray from Carnegie Mellon's AutonLab analyzed the 2014 sorghum dataset, which had 390 samples with nine input features, and four composition features as the output features. She found that random forests performed best at predicting the four composition features, achieving an average r^2 score of 0.50. There was no 2015 dataset. Ray also analyzed a preliminary 2016 dataset. Ray used random forest regression to predict harvest features from robot generated-features, achieving an average r^2 score of 0.18. Although Ray completed an analysis of the partial 2016 dataset, this paper is the first to analyze the complete 2016 dataset with accession, field location, robot-generated, drone-generated, harvest and composition features.

Random forest regression is an ensemble method which makes predictions by averaging the outputs of its constructed regression decision trees. They were first formally defined by Breiman (2001). Fernández-Delgado et al. (2014) evaluated 179 classifiers on 121 datasets and found that random forests achieved the best overall accuracy. Caruana et al. (2008) also came to the same conclusion when they evaluated 10 families of classifiers on 11 binary classification problems with high dimen-

sionality. Random forests had the highest performance on average across all considered datasets, outperforming SVMs, neural nets, logistic regression, etc. Random forests are fast and perform well in practice.

4 Data

The 2016 sorghum dataset was created by sorghum cultivated at Simpson Farm located in Pendleton, South Carolina. This field is partitioned into 5 meter by 1 meter subplots, where each subplot contains sorghum plants that are all genetic siblings. Each sample of the 2016 sorghum dataset describes a single subplot. The dataset contains 698 samples with accession, field location, robot-generated, drone-generated, harvest and composition features.

Table 1: Timeline of data availability in the 2016 sorghum dataset. The 2016 field was planted on May 26 - 27, 2016 and harvested on October 3 - 6, 2016 . Note that this is the first year the ground robot and aerial drone produced features. So, the bold Robot and Aerial datasets are expected to become available much sooner in the upcoming 2017 dataset.

Feature Type	Description	Date First Available
Accession	Biological categorization of sorghum.	Before planting
Field Location	GPS location of subplot centers.	Before planting
Robot	Automatically generated features from ground robot.	November 16, 2016
Aerial	Automatically generated features from aerial drone.	August 2, 2017
Harvest	Phenotypic features of fully grown sorghum.	January 12, 2017
Composition	Chemical composition of harvested sorghum.	July 5, 2017

Prior to planting, the accession and field location of each subplot is known. There are four accession features: *accession_photoperiod*, *accession_origin*, *accession_type*, and *accession_race*. *accession_photoperiod* is a boolean value for whether the subplot's sorghum is sensitive to the length of day or night. *accession_origin* is the sorghum's country of origin. *accession_type* and *accession_race* are are biological classifications of the sorghum, such as Sweet sorghum and Grain sorghum. Finally, the field location features include the GPS Easting and Northing of the center point of the subplot.

Carnegie Mellon created a ground robot with multispectral cameras that traverses the rows of sorghum. Members of Carnegie Mellon's Robotics Institute then used computer vision to automatically compute leaf necrosis, vegetation index, leaf area, plant height and light interception from the raw sensor values. The ground robot traversed the field in July, August and September of 2016. Near Earth Autonomy created an aerial drone that is able to measure macro-scale field growth. The 2016 dataset only includes the average saturation and hue from an overhead photograph of the field.

Generating the composition data involves conducting experiments which chemically break down harvested sorghum. So, unlike the data generated by the ground robot and the aerial drone,

the composition data is not expected to become available much earlier in the upcoming sorghum datasets.

Appendix A contains a complete list of the features used in the regression analysis. The full dataset can be found at `https://raw.githubusercontent.com/bparr/dap/master/2016.merged.csv`. Note that this file includes unused features (e.g. Notes). Also, 66 of the subplots in 2016.merged.csv file were ignored for regression analysis because they were missing composition data.

5 Methods

5.1 Training Procedure

The 2016 dataset is relatively small with 698 samples. So, I trained multiple types of regressors, such as Random Forests, Support Vector Machines (SVM), and Kernel Ridge. For each regressor, I used 10-fold cross validation, which randomly partitions the dataset into 10 equally-sized subsamples. Each subsample is used as the test set for a regressor trained on the remaining nine subsamples. I trained models for three different views of the 2016 dataset, so that I could compute the improvements from including the ground robot and aerial drone features, as well as from including the harvest features.

5.2 Missing Values

The sorghum dataset unfortunately has a significant amount of missing values, as shown in Figure 2. For example, more than half of the samples are missing at least 7 feature values (i.e. missing at least 24% of the total inputs). During regression analysis, -1 was used for these missing values.

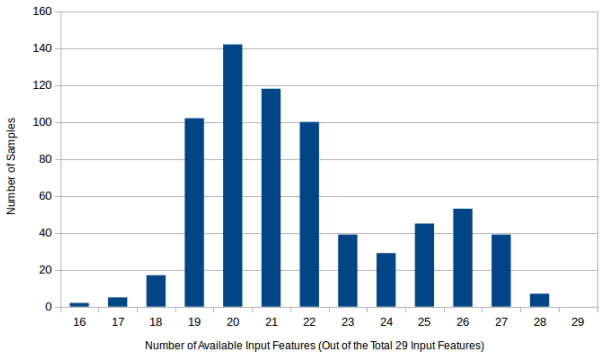


Figure 2: Missingness in the dataset. The dataset has no missing output values. So all missing values are from the 29 input features.

5.3 String Values

The accession features are string values, which regressors can not naturally train on. Before training, each accession feature was mapped to a one-hot vector using the SciPy DictVectorizer. The

accession features also include missing values, which are simply represented as the first dimension in the one-hot vector. So, a missing accession feature would be mapped to a vector with a 1 in the first dimension, and 0 for the remaining dimensions.

5.4 Apparatus and Instrumentation

I trained the regressors on an AutonLab machine with 48 cores and 500 GB of RAM. All regressors can also be trained on much smaller machines, although slower. The AutonLab machine runs Python v3.4.5, NumPy v1.13.0, and SciPy v0.19.1.

6 Results

Random forests performed best, with an Overall r^2 score of 0.436. Figure 3 includes the final r^2 score results for each composition feature, as well as the multi-dimensional Overall r^2 score.

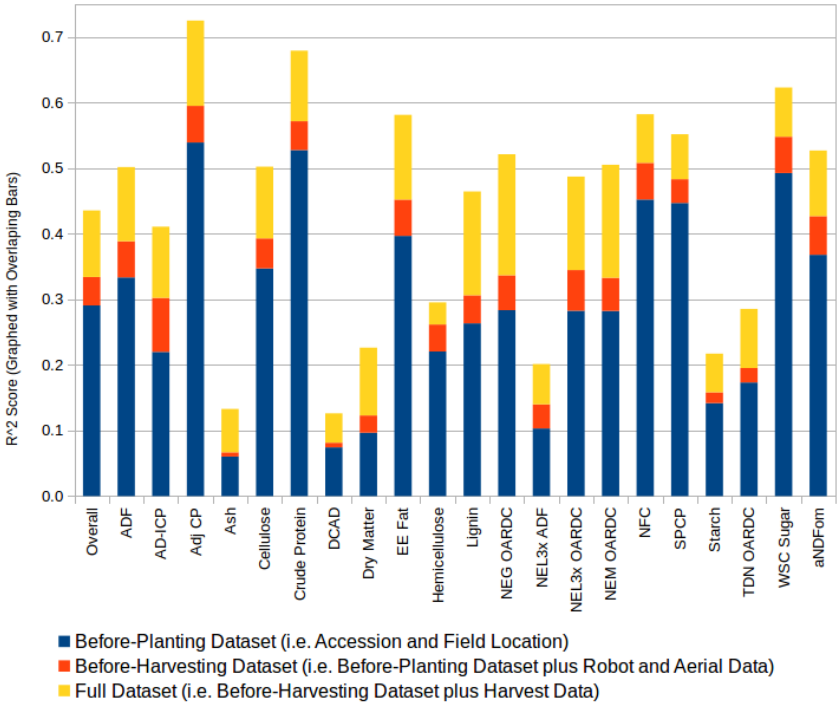


Figure 3: Final results for each composition feature, and Overall. Each bar represents three separate dataset views, with overlapping bars. The blue part of the bar represents the r^2 score when using just Accession and Field Location. The red part of the bar represents the increase in r^2 when using the Robot and Aerial data, along with the Accession and Field Location data. And finally, the yellow part of the bar represents the increase in r^2 score when using the Harvest data, along with the Accession, Field Location, Robot and Aerial data. Note that the y-axis only goes to 0.75 instead of the maximum r^2 score of 1.0.

These results were achieved by using my missingness augmentation along with the SciPy RandomForestRegressor. The random forest used 100 trees, each with a maximum depth of 10, and also used $\sqrt{\text{number of input features}}$ as the number of features to consider when looking for the best split. Finally, the random forest only split if the node had at least 10 samples in it. These changes from the default SciPy settings are based on the default settings for the AutonLab implementation of Random Forest regressor.

6.1 Missing Value Data Augmentation

I increased the Overall r^2 score by 2.5% (from 0.425 without augmentation, to 0.436 with augmentation) by augmenting the training set with training samples that had selected values removed and replaced with missing values. The idea was inspired by the image recognition improvements from augmenting training images, by shifting the image for example. By augmenting the training set, there were more training samples with missingness identical to missingness in the test set.

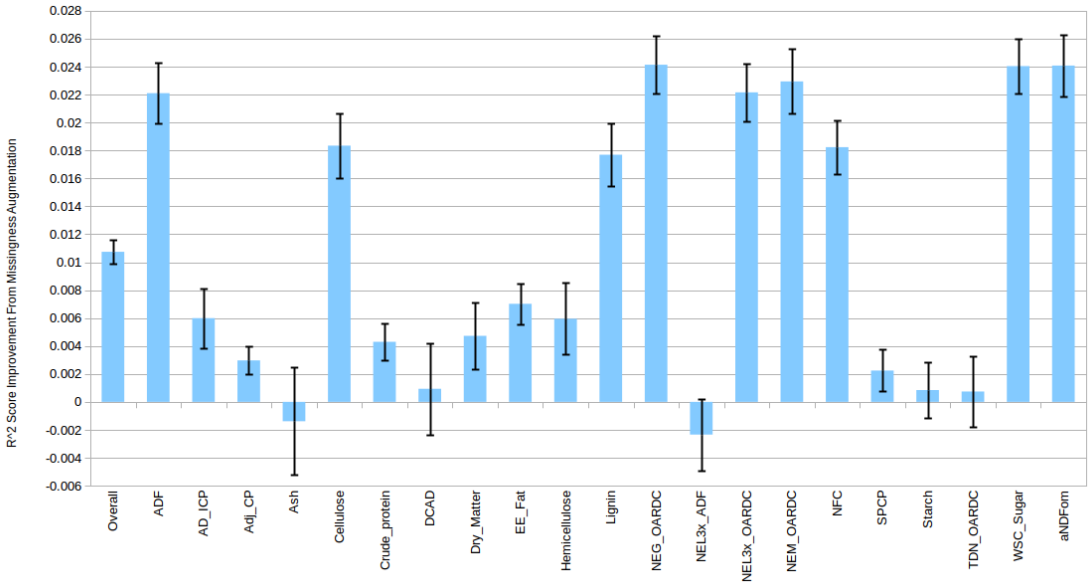


Figure 4: For the full 2016 dataset, the difference between the mean r^2 scores with augmentation and mean r^2 scores without augmentation. So greater than 0.0 is an improvement from augmentation. With augmentation and without augmentation were both run for 100 trials. The figure also includes the 99% confidence interval for the difference in mean r^2 scores. Only five individual composition features (Ash, DCAD, NEL3x ADF, Starch and TDN OARDC) did not see a significant increase in r^2 score from augmentation. No composition feature saw a significant decrease in r^2 score.

Before training, I ran the below missingness_augment() Python function (minified for readability):

```
def missingness_augment(X_train, y_train):
  missings = set()  # Set of True/False tuples.
  for x in X_train:
    missings.add(tuple(is_missing(value) for value in x))

  augmented = []  # List of (x_train, y_train, sample_weight) tuples.

  for x, y in zip(X_train, y_train):
    augmented_samples = set([tuple(x)])  # Always include original.

    for missing in missings:
      augmented_samples.add(tuple([
          (MISSING_VALUE if b else a) for a, b in zip(x, missing)]))

    # Note that the sum of sample weights for each original x = 1.0.
    augmented.append(x, y, 0.5 + 0.5 / len(augmented_samples))
    for augmented_sample in (augmented_samples - set([tuple(x)])):
      augmented.append(augmented_sample, y, 0.5 / len(augmented_samples))

  return augmented
```

For example, consider a dataset with only three input values, and where X_train only contains three samples with values $[MISSING\_VALUE,\ 2.0,\ 3.0], [4.0,\ 5.0,\ MISSING\_VALUE]$ and $[7.0,\ 8.0,\ 9.0]$. Augmenting the sample with value $[7.0,\ 8.0,\ 9.0]$ would result in three augmented values:

- $[7.0,\ 8.0,\ 9.0]$ with $sample\_weight = \frac{4}{6}$
- $[MISSING\_VALUE,\ 8.0,\ 9.0]$ with $sample\_weight = \frac{1}{6}$
- $[7.0,\ 8.0,\ MISSING\_VALUE]$ with $sample\_weight = \frac{1}{6}$

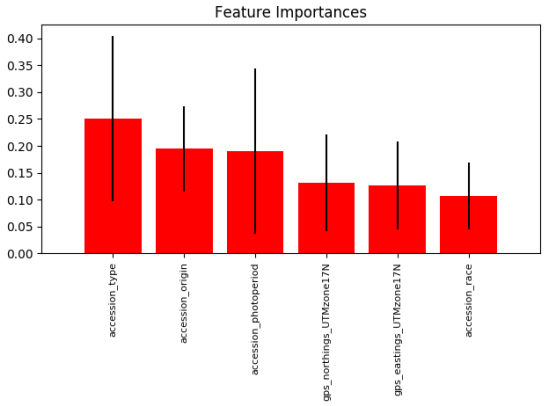
Since the missingness_augment() function uses sets, there are no duplicate entries when augmenting a training sample. Since each original training sample results in training samples with sample weights that sum to one, the original weight of a sample remains unchanged. Without using sample weights, samples with fewer missing values and thus more augmented samples would have a larger effect in test predictions. In practice, this caused worse results in the 2016 dataset when compared to running without augmentation. I experimented with a few different sample weight schemes, and saw no significant difference. So I settled on simply preallocating 0.5 of the sample weight to the original sample, and evenly distributing the remaining 0.5 sample weight among the augmented samples.

The 2016 sorghum dataset has only 71 different missingness patterns. On average, there are about 69 different missingness patterns in the training set. The augmented training set was on average 32

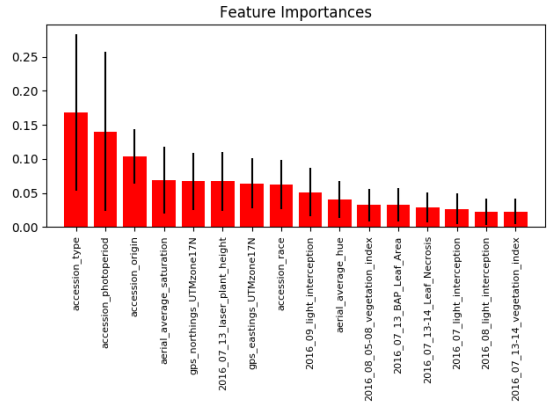
times larger than the original training set. Regardless, the augmented training set is still relatively small, with an entire 10-fold cross validation still taking less than a minute to complete.

6.2 Feature Importance

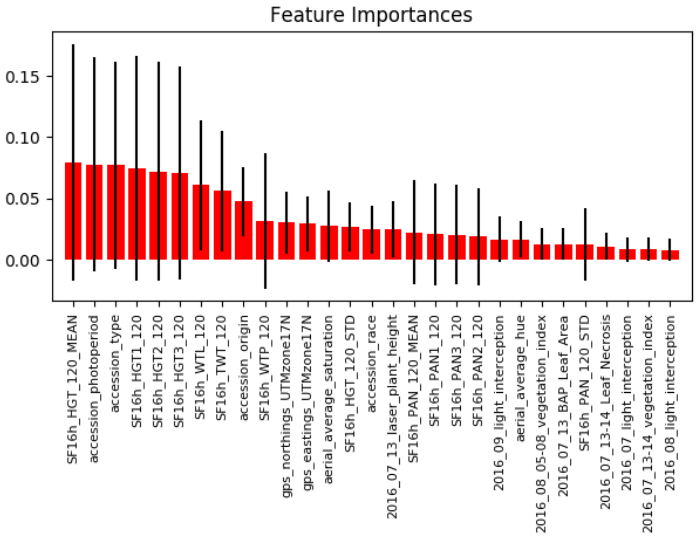
The SciPy implementation of random forests includes feature importances. See Figure 5 for the mean and standard deviation of each feature's importance across all trees. For the full dataset, harvest height and accession features were the most important.



(a) Before-Planting Dataset



(b) Before-Harvesting Dataset



(c) Full Dataset

Figure 5: Mean and standard deviation feature importances across all trees in random forests trained on the 2016 dataset. Feature importance is computed relative to all features, so that the sum of all importances is 1.0.

6.3 Regressors Besides Random Forests

My final results use random forests because they performed the best with an Overall r^2 score of 0.436 on the full 2016 dataset. I also tried other regressors available in SciPy, including Support Vector Machines (SVM), Nearest Neighbors, Kernel Ridge and other decision tree regressors. However, with only 698 samples and a relatively large amount of missing data, many of the regressors performed poorly. After the random forest regressor, the next best four regressors were GradientBoostingRegressor, ExtraTreesRegressor, AdaBoostRegressor and KernelRidge, which respectively achieved an Overall r^2 score of 0.407, 0.346, 0.345 and 0.330 on the full 2016 dataset.

6.4 Reproducibility

All input files, source code and output files can be found at `https://github.com/bparr/dap/`.

7 Discussion

As previously mentioned, 2016 was the first year that the ground robot and aerial drone produced features. So there were some unexpected difficulties. For example, the aerial features available in 2016 are only the tip of the iceberg. Near Earth Autonomy also produced elevation maps, infrared maps and RGB maps of the 2016 sorghum field. Unfortunately, these maps could not be segmented into subplots because of GPS differences between Near Earth Autonomy's data and the rest of the dataset. The 2017 field should not have this problem, and thus there will be more aerial-generated features that will need to be incorporated into the 2017 sorghum composition regression. By incorporating these upcoming features, aerial-generated predictive power should increase.

Predictive power in 2017 should also increase from improvements in the ground robot. For example, the 2017 dataset should include hyperspectral data of the sorghum leaves gathered by shining a broadband light source directly on the leaves. These raw spectra features could be incorporated into 2017 composition regression similar to Heckmann et al. (2017), and thus the chemical composition of the sorghum leaves could be more accurately predicted.

8 Acknowledgements

This research was made possible by the U.S. Department of Energy under award DE-AR0000595: Breeding High Yielding Bioenergy Sorghum for the New Bioenergy Belt. Also, thank you to Dr. Artur Dubrawski for advising me on this project, and Saswati Ray and Simon Heath for their project feedback.

References

Breiman, L. (2001). Random forests. *Machine learning*, 45(1):5–32.

Brenton, Z. W., Cooper, E. A., Myers, M. T., Boyles, R. E., Shakoor, N., Zielinski, K. J., Rauh, B. L., Bridges, W. C., Morris, G. P., and Kresovich, S. (2016). A genomic resource for the development, improvement, and exploitation of sorghum for bioenergy. *Genetics*, 204(1):21–33.

Caruana, R., Karampatziakis, N., and Yessenalina, A. (2008). An empirical evaluation of supervised learning in high dimensions. In *Proceedings of the 25th International Conference on Machine Learning*, pages 96–103. ACM.

Fernández-Delgado, M., Cernadas, E., Barro, S., and Amorim, D. (2014). Do we need hundreds of classifiers to solve real world classification problems. *J. Mach. Learn. Res*, 15(1):3133–3181.

Heckmann, D., Schlüter, U., and Weber, A. P. (2017). Machine learning techniques for predicting crop photosynthetic capacity from leaf reflectance spectra. *Molecular Plant*.

Koba, M. (2014). World may not have enough food by 2050: Report. *CNBC*.

Schmitmeyer, L. (2016). FarmView Uses AI To Feed a Growing Planet. *The Link, School of Computer Science, Carnegie Mellon University*, (Winter):6–9.

Appendix A: Description of Dataset Features

Feature Type	Feature Name	Description
Accession	accession_photoperiod	Boolean value for whether the subplot's sorghum is sensitive to the length of day or night.
	accession_type	Predominant type for sorghum germplasm (e.g. Grain).
	accession_origin	Country of origin (e.g. Ethiopia).
	accession_race	Taxonomic rank below subspecies (e.g. Durra-bicolor).
Field Location	gps_eastings_UTMzone17N gps_northings_UTMzone17N	GPS Easting and Northing of the subplot's center (in UTM).
Robot	2016_07_13-14_Leaf_Necrosis	Cell death or degeneration in leaves. Measure of plant stress, calculated as a ratio of healthy/unhealthy pixels.
	2016_07_13-14_vegetation_index 2016_08_05-08_vegetation_index	Measure of live green vegetation.
	2016_07_13_BAP_Leaf_Area	Exposed leaf area.
	2016_07_13_laser_plant_height	Plant height according to the ground robot.
	2016_07_light_interception 2016_08_light_interception 2016_09_light_interception	Ratio of Leaf Area / Area of Sky, using an upwards facing camera.
Aerial	aerial_average_hue aerial_average_saturation	Average hue and saturation of the subplot, using an overhead photograph of the entire field.
Harvest	SF16h_TWT_120	Plant weight, in kilograms.
	SF16h_WTP_120	Weight without panicle, in kilograms.
	SF16h_WTL_120	Weight without leaves, in kilograms.
	SF16h_HGT1_120 SF16h_HGT2_120 SF16h_HGT3_120 SF16h_HGT_120_MEAN SF16h_HGT_120_STD	Plant height, in cm, measured from ground to apex of panicle. Three plants were measured. MEAN and STD are calculated from these three measurements.

	SF16h_PAN1_120 SF16h_PAN2_120 SF16h_PAN3_120 SF16h_PAN_120_MEAN SF16h_PAN_120_STD	Length of panicle, in cm. Three panicles were measured. MEAN and STD are calculated from these three measurements.
Composition	ADF	Acid Detergent Fiber.
	AD-ICP	Acid Detergent Insoluble Nitrogen.
	Adj CP	Adjusted Crude Protein (e.g. adjusted for excessive heat damage).
	Ash	Total mineral content.
	Cellulose	C6 carbohydrate found in cell walls.
	Crude Protein	Approximate amount of protein.
	DCAD	Dietary Cation-Anion Difference.
	Dry Matter	Weight with water removed.
	EE Fat	Crude fat content.
	Hemicellulose	C5 carbohydrate found in cell walls.
	Lignin	Strengthening material in the cell walls.
	NEG OARDC	Net Energy of Gain. Energy available to produce a weight gain.
	NEL3x ADF	Net Energy of Lactation.
	NEL3x OARDC	OARDC Net Energy of Lactation.
	NEM OARDC	Net Energy Maintenance.
	NFC	Non-Fiber Carbohydrates.
	SPCP	Water soluble protein.
	Starch	A non-fiber carbohydrate mainly used for storing energy.
	TDN OARDC	Total Digestible Nutrients.
	WSC Sugar	Water Soluble Carbohydrates.
	aNDFom	Ash free natural detergent fiber.