SMART FARM DATA ANALYTICS

MINOR THESIS

Adekola Oluwatade Job

Deakin University

Project Title: Smart Farm Data Analytics
Project Supervisor: Prof. Gang Li
Unit Chair: Dr. Shui Yu
Student Name: Adekola Oluwatade Job
Student ID: 215383256

Table of Content

Table of Figures

Acknowledgements

I would like to express my earnest gratitude and appreciation to some people without whom this thesis would not have been completed.

Big thanks goes to Professor Gang Li, my main supervisor, for guiding me every step of the way and mentoring me right from the onset of this thesis to this very moment. His advice, insights and professionalism in putting me through the rudiments of thesis writing were outstanding and I was able to learn a lot.

I'm also thankful to my colleague, Mohit, with whom I also got some insights when I got stuck at some point during the analysis stage of the thesis. I was able to learn so much from all the independent research I made and some online videos on how to carry out the research methods appropriately.

Also, thanks to the unit chair for providing such a welcoming session of exploratory skills. I was able to learn more about report writing and also learn about the various suitable report format for different kinds of report and paper writing. The breakdown of tasks and submissions made it easier for me to articulate the research and it structure into one submission.

Ultimately, big thanks to God for making it quite a successful one.

Project Abstract

The evolution of smart farm is based on the development of Information technology across all facets of study. Although the technologies involving the use of data analytics in farming have just been gaining wave, this thesis will cover how predictive analytics techniques can be adequately used to predict crop yield across different times of the year given some common growth factor. The crop dataset was collected from different farmers from different geographic locations. The data set includes the amount of crop yields per some particular period of time and a list of numerous factors that influences these yields. The result of the prediction will help in most importantly forecasting and predicting:

1. The times of the year in which crops blossom the most.

2. The most important factors that hugely impact on the yield of crops.

3. Factors that most influences growth at different weeks of crop growth.

Some other methods that will be implemented to the dataset will include the use of cross validation (K fold) to increase the accuracy of the accuracy of the prediction.

Penalties in the form of regularization will be applied to model building so as to reduce noise, overfitting and prediction errors. These concepts only help to improve prediction accuracy and also provide a very good platform for learning various kinds machine learning algorithms.

Introduction

This research aims to ultimately predict crop yield. The research area will cover data collection in which the dataset is a combination of several data collected from different farms. After the data is collected and pre-processed, analysis of the data will follow in which suitable features that might influence crop yield will be identified using correlation and also independent research on likely resources that explains different predictors that affects yield of crops. Factors with the most recurrent influence on growth throughout the crop growth period. This is done in a weekly basis.

This research problem completely aims to discover better ways in satisfying customer demand and saving cost by accurately forecasting the amount of yield a set of farm crops grown in a particular period will produce. The innovative approach to farming will help the agricultural sector embrace the use of data and analytics to improve yield and even optimise business processes within the agricultural sector. The outcome of the thesis will advance the application of machine learning to agriculture. Smart farming has been a very core aspect of agriculture. Many large-scale farms require computer operated machines to aid production and minimise cost. But today,

technology has so improved to the extent that every data about farming operations that'd have been disposed of in the past are now very important to develop insights about trends and anomalies in farming operations.

Huge investments have been made in the area of analytics by several organisations and this has really paid dividend as lots of data mining techniques have evolved in their applications. Leading IT companies like Oracle, SAS, IBM, SAP and lots more have developed solutions in analytics that can be easily utilised by 3rd parties to generate results, derive insights and other hidden patterns in their data that could help in performance increase.
Another technological innovation is the IOT introduction into farming. It's expected that in the nearest future, the need for more food will rise as the population of the earth continues to grow. Current systems won't be able to aid faster food production. The employment of IOT devices will help farming faster, provide insights about soil topography, weather forecasts and even the nutrients in the soil. IOT will enable farmers to monitor their farming operations through their mobile devices, get interesting stats about crop and livestock growths as well. It's estimated that self-driving tractors will soon be introduced. These tractors will be capable of ploughing large areas of land without much supervision from farmers. All these will improve production and reduce cost in the long run. Some other benefits are in the areas of satellite usage and drones. Drones are now being adopted by several farmers to improve security and as well several other tasks. Drones can be used in the farm to provide more camera covers to areas within a farming environment where cameras can't reach. Sometimes drones can be employed to scan or take an aerial photograph of the farm. Imaging yet another advancement in smart farming. This approach uses advanced cameras to take photos of crops over some periods of time. This is often used to ensure that crops are not growing yellow which signifies lack of nutrient. The images collected can also be used for image processing and estimating growth. It can also be used in cases of prediction of estimated yielding time.

Multi agent systems that makes use of machine learning principle are also being used in most industrial farms. These systems make use of intelligent agents to learn all activities that have common trends and then implement them at the necessary times. This concept has been applied to many large-scale farms to ease and reduce labour and as well reduce cost.

Background

This background review takes into consideration all past works that have been carried out on the integration of Technology into agriculture. Different aspects of technological research have been done on how to optimally improve the way farming and agricultural processes are being carried out. Lots of journals have been published on different areas in which advanced technologies can be used in improving all aspects of agriculture. Most notable systems that have been integrated is the precision farming technique. This farming technique applies the use of satellites and sensors to study agricultural environments through imaging and comparison algorithms.

The literature synthesis will cover the review on past works that have been done on the precision farming. It will highlight the strengths and also the weaknesses of precision farming and why a much more suitable approach that when augmented with precision farming will bring about a huge impact on the way we farm and agriculture as a whole. Then subsequent analysis will be done on the approach that brought about intelligent farming into practice. The use of robotics and intelligent machines to aid farming processes will also be reviewed and traced back to the onset of their usage and how science improved on them through research and studies.

The more recent smart farming techniques which involves the use of analytics and machine learning techniques will be reviewed, current challenges will be analysed and future works that could make the technique much better will also be discussed.

Finally, this literature synthesis will cover the research methodologies that would be used in this research and all the processes involved such as data collection in which the dataset is a combination of several data collected from different farms. Then data normalization which involves normalising the data dimensions into same scale for easy analysis. Other analytical processes will then be used to ensure that the Ultimately, factors with the most recurrent influence on growth throughout the crop growth period will be reviewed and more analysis will be done afterwards.

The research problem completely aims to discover better ways in satisfying customer demand and saving cost by accurately forecasting the amount of yield a set of farm crops grown in a particular period will produce. The innovative approach to farming will help the agricultural sector embrace the use of data and analytics to improve yield and even optimise business processes within the agricultural sector. The outcome of the thesis will advance the application of machine learning to agriculture.

Research Motivation

The motivation behind this research is based on the evolving trends and technology in data and analytics. The agricultural industry has long been using Information technology in its day to day activities and some technologies like precision farming, image processing etc. These technologies have so long been very pivotal in every phase of farming and as data usage continue to increase, processing of these data to gain meaningful insights to improve the way we farm became a goal. This research was majorly started to achieve some breakthrough in big data application to farming. This research will focus on how predictive models can be built from data collected from different farms on the growth of tomato plant and their environmental factors. The model built can be used to predict yield and help in anticipating the amount of profit that can be estimated from a set of tomato plantation at any given time. This can also provide insights on what environmental factor best influences the growth of tomato plants.

Also, looking at the significance of the success of this project, another great motivation behind carrying out this project was because of how high a necessity that the impact of this solution would be and massive and begin a new era of smart farming. Farmers would be able to understand what physical and environmental factors affect crop growth.

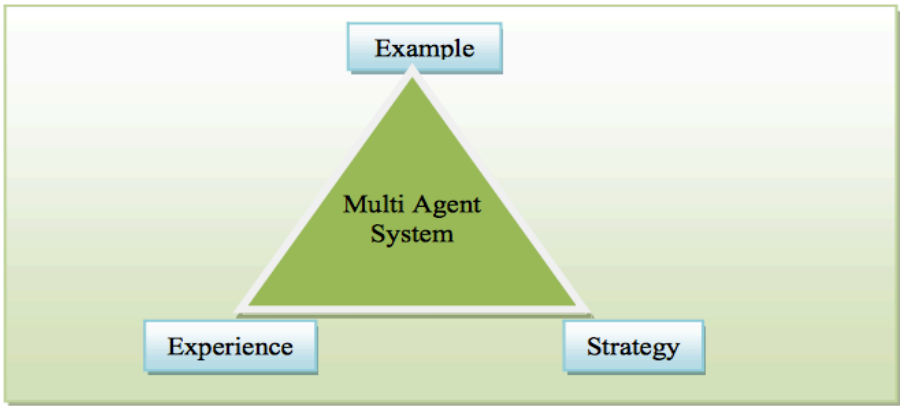
Literature Review

It's been a long time coming since the pre-birth of smart farming. It all started with the aim of implementing the advances in Information technology in order to improve the way farming and agriculture as a whole is done. Precision farming was one of the early techniques to be used in this case. According to Shekhar S (6), It was introduced as a mechanism that uses several integrated technological tools. The most prominent one global positioning systems such as sensors to monitor the growth and health of crops, map visualization systems to understand the variabilities in crop structure and texture, databases to collect and store farm data and decision support systems to aid decision making to optimize yield. The need for optimisation later brought about the introduction of descriptive, predictive, prescriptive and proactive levels (17). Descriptive levels would enable farmers understand crop and soil traits from historical growth nature and help in better farm management. Predictive level would use historical data as well as soil, crop and water model to forecast crop yield (concept to which project is based), Prescriptive levels would be used to determine farm

management interventions in cases of useful generated insights. This would give farmer ability to prescribe procedures to use based on some knowledge derived from this area.

Another development was suggested in 2012 by (Gutta A et al). In this journal, much overviews were made regarding the use intelligent multi agent systems to provide access to large farming related data. This would very much reduce information overload and improve from paper-based systems to much more intuitive and business intelligent agents who can help in making fast and reliable decisions. The methodology proposed for this system is based on 3 approaches which can be used to optimize knowledge systems and improve the intelligent systems used to improve farming. These approaches are 1. The use of example; to building reasoning and learning systems for intelligent agents. 2. Implementation of strategies; this is developed from explicit learning from past and historical data. 3. The experience aspect which is hugely utilized in decision making (6). This system is seen in the diagram below:

Figure 1



Also, most recently, some work has hugely been done on precision livestock farming and this has vastly improved so many aspect of agriculture and improved humanity. Precision farming, an aspect of smart farming that utilizes satellite imaging to monitor variations in agricultural fields to grow more food using fewer resources and reducing production cost (11). A review was done by T.M Banhazi on the application of precision farming to livestock. Here, the very important effect of using precision farming were outlined and discussed in detail as it improves livestock welfare on farm and also reduces greenhouse effect (10). The journal targets the current state of the art improvements and also the commercialisation aspects of precision farming with the sole purpose of suggesting more advancements between its technological and commercial aspect. The issues faced in this areas as a result of the will to use precision farming was in the area of data collection where different data are gathered from

variety of sources including media, government advisors, magazines etc. This was one of the major stumbling block to implementing precision farming at the time as it was difficult to apply readily these varieties of unstructured data (10). Another stumbling block was that farmers tend to ignore areas where their expertise is not covered and tend to neglect areas that are seemingly the driving force to productivity and profitability. There were also likely thoughts that huge risks might be involved in adopting the precision system as it involves spending a huge chunk of money what has not been widely proven and adopted by many.

Very recently, after the emergence of Big data began, more grounds were gained in the effort to implement big data into agriculture which now surpasses not just the implementation of precision farming but also the use of robotics and drones. This is because farming processes are now becoming more data driven and IOT driven (12). It integrates several properties of data analytics to drive the improvement if agriculture in an all-round manner. Areas that are vastly utilized by big data smart farming include real time data processing, weather and real time disease data. Other properties through the aid of IOT that has made a significant improvement in smart farming include the interconnection of digital devices used in the farm to aid faster data processing and insights discovery (12). For instance, devices like tractors, rain gauges, irrigation machines etc consist of sensors that interconnects them and comprises pf built-in intelligence which are very capable of enabling these devices perform autonomous tasks. This in turn leaves a whole chunk of data as these complex processes are being carried out. Deriving insights from these data is what the research will try to achieve.

Related Work and Current Research

This research is hugely narrowed down to a specific aspect in the area of smart farming which involves the use of data analytics and machine learning algorithms to predict certain outcomes in plant growth and development. Not only does it involve prediction but also involves identifying certain environmental condition for which certain crops will flourish and also identifying accurately their yield times. Here, very related works that have been done in the past under this scope will be reviewed.

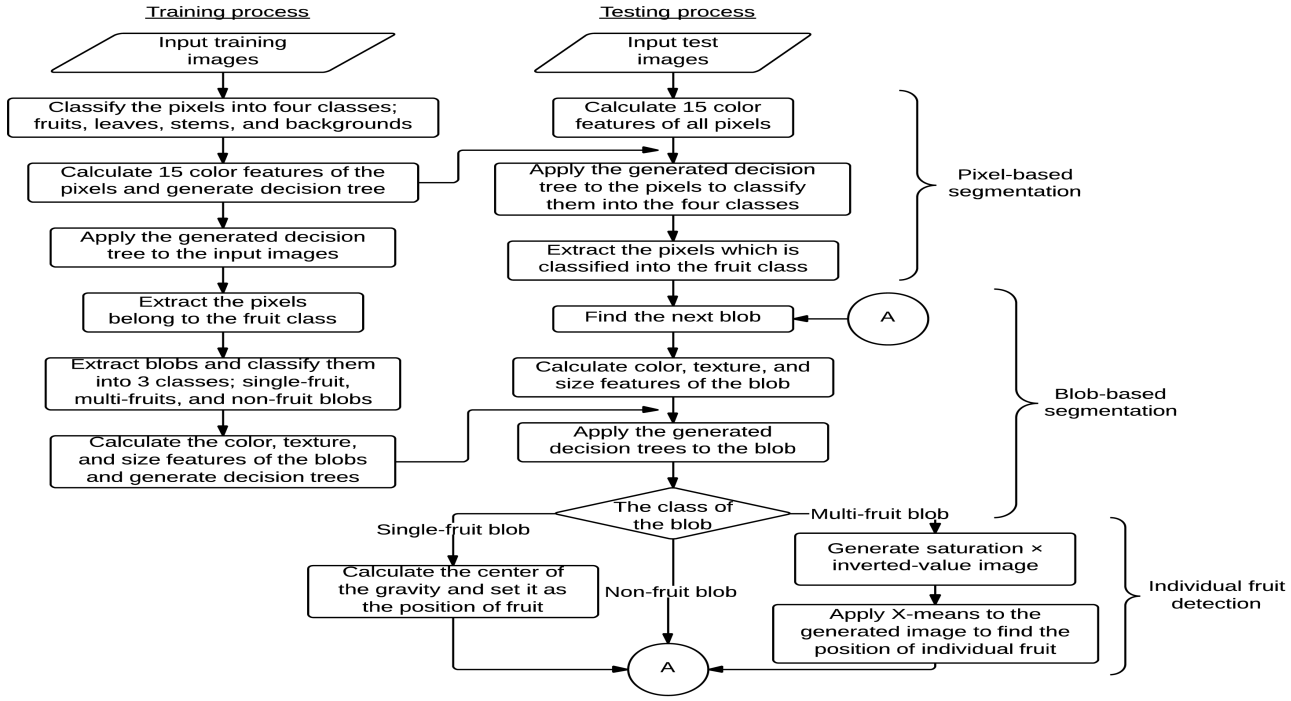
Japan has hugely impressed in the application of machine learning algorithm to aiding farming. The employment of computational intelligence systems for both farming productions and their subsequent environments coupled with intelligent robots has been tremendous (13). Also, biosystem-derived algorithms for photosynthetic analysis and their comparisons with Neural networks is another complex farming advancement

that has been unveiled in Japan. These techniques were employed basically for solving optimisation problems and aiding plant growth. Here, the neural network algorithm is used in identifying how plants are affected by different concentration of farm nutrients. After insights have been generated using this approach, an optimal step that aims to maximise plant growth will be developed using identified neural network models and genetic algorithms (19). Other methodologies such as navigations, sensors, imaging etc. are currently proposed and being carried out in most Japanese universities but due to limited funds, little advancement have been made in these approaches. Many research work are being done on the use of photosynthesis.

Another important research area was on image processing and detection of disease on crops and plants (7). This technique was employed to help and effectively improve good and healthy growth in crops. This technique consists of two image databases. One is used for storing data of already trained disease model and the other is used for querying images. The neural network back propagation is used here (15). The images are classified according to their disease categories and this is done on the basis of three features ie. Colour, morphology and texture. This technique further helps in disease detection using these properties and help in improving production significantly.

So many work have been done in the past to create a platform for a smart farm but not many have utilized the techniques of predictive analytics to solve the research problem. In Al-Gaadi, Hassaballa (1) research, a prediction was done on potato crop yield using precision technique. Here, remote and GIS techniques were employed to monitor the growth and development of potato crop yield and also to predict yield. This was implemented to about three different farms in the east region of Saudi Arabia and the images collected from the satellites were used to develop vegetation index maps to understand the health status of the crops. Few days prior to harvest, prediction yield samples were collected and were correlated to the results of the two imaging systems. Prediction algorithms were developed as a result of the correlation analysis and this was used to generate prediction yield maps (20). The result of the study gave further insights on variations across the three different potato fields and in turn help farmers in decision making when managing their practices.
Another closely related work was done by K. Yamamoto from the University of Tokyo (2). Here, the aim was to differentiate between intact tomato fruits using different methods like shape, colour, size etc. prior to harvest. This was implemented using image analysis and machine learning methods but the problem with this technique is that it depends on the threshold used and this varies with images. It could optimally differentiate between mature and immature fruits but could not estimate the number of young fruits which is important for prediction of long term fluctuation in yields.
The flow chart for the developed method is seen below

Figure 2



Key Challenges and Questions

The research problem associated with this work involves the use of a better predictive algorithm approach to better predict crop yield. Prior to proposing this project work, some Initial works were done on the dataset with the use of multiple regression techniques that involves using correlation level of the response variable with the predictors to predict likely future outcome became ineffective as it was discovered that patterns in factors affecting crop growth and yield changes based on other hidden factors. So, there was a need for a more robust algorithm that could handle both supervised learning and unsupervised learning. This brought about the idea of using cross validation technique to improve accuracy.

Also, another problem with the project is that other methods that have been used do not actually explain the factors that influences the growth of the tomato crop at different stages of their growth but instead gives the factors that influences growth at the 8^{th} week i.e. yield week.

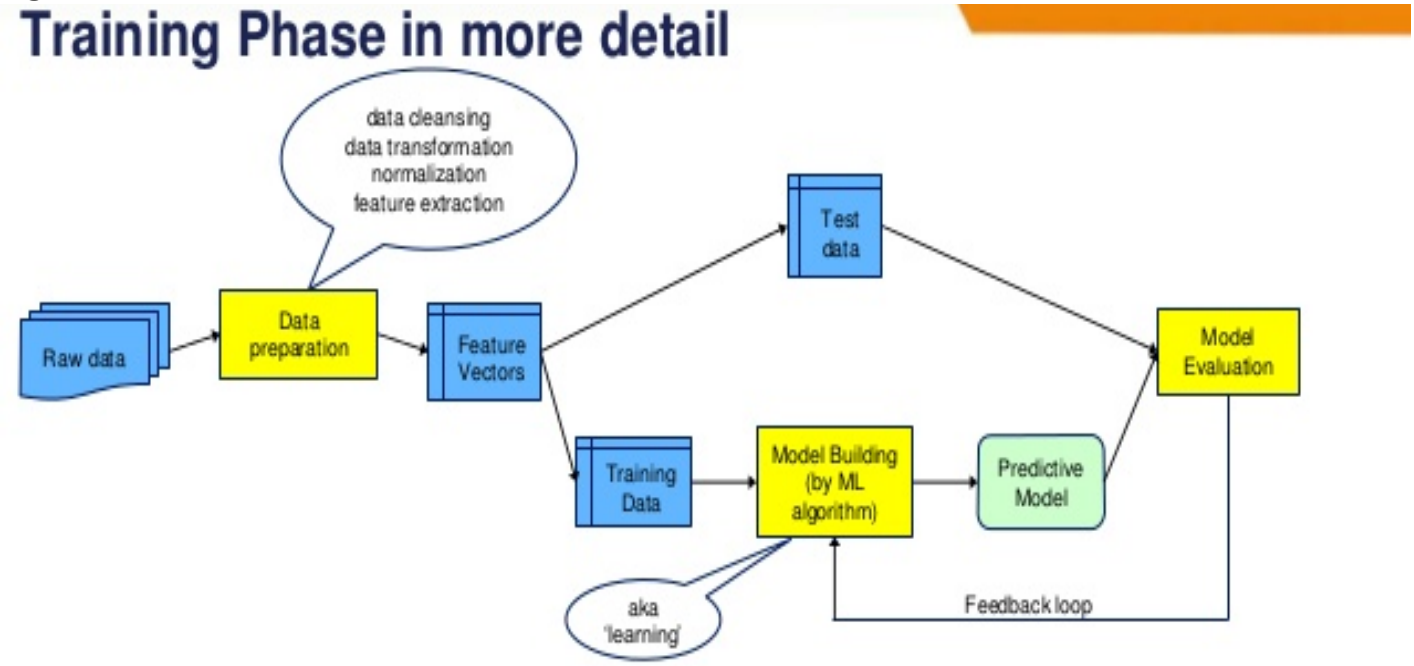
Future research could be done on the implementation of deep learning to the project. This is a more effective approach but require a lot of time and input.

In conclusion, some of the research question will include but not limited to the following:

1. What analytic tool is best suitable in giving the insight we are trying to get?
2. Another question is finding the more robust machine learning algorithm that best gives best prediction.
3. Can we rely on just predictive models or can we implement some model evaluation techniques to boost predictive accuracy?
4. How can we best identify the most prominent factors that affect the crops at different stages of growth I.e. each week.
5. What measures best help in checking predictive accuracy of existing models?
6. Choosing the best strategy so that the overall yield is maximized.

The methodology that would be used for this project will include the use of machine learning algorithm and follow the approach shown in the figure below:

Figure 3



The methodology includes

1. Data collection and pre-processing: This stage involves the collection of data and storing in a readable file format for programming and analysis to be done. The pre-processing processes involves feature extraction, imputing missing values, scaling, dimensionality reduction and correlation analysis
2. Data Normalisation and Transformation: Data normalisation and transformation is very important in data analysis as several dimensions of data might be in different unit and for models to be built and analysis to be carried out, the data must be in the same unit to prevent bias results. In this research, logarithmic transformation will be done on the dataset and then reduction of data will be

done using PCA approach. This would involve a step by step procedure that determines the specific dimensions that hold a very high variance. These components will be selected and will be used for model building

3. Training and Testing data: For basically all machine learning approaches and model building, training and testing data are required. This is because a segment of the dataset which is called training is used to train the model to understand and build knowledge on the data by looking for trends, correlation etc. Then the testing part helps us to testing the trained model to check for its accuracy and understand performance. Here, we'll be using regression, XGboost and comparing their performances.
4. Model building: Model building is the aspect of analytics that makes use of some learning system to understand a dataset and intelligently understand relationship between target(s) and predictor(s). In machine learning, there are different model building algorithm but in this research, we'll focus on few.
5. Evaluation of existing Models: This takes into consideration all the existing models that have been built in doing analysis on the dataset. The evaluation technique uses different criterion to assess how a model performed and which model can be good for prediction. Some of the criterion include misclassification rate, P value, AIC and BIC, RMSE (root mean square error) etc. This will be used to evaluate which model to use in this research.
6. Using Cross Validation Method to Check for model Accuracy: This is one of the model evaluating techniques that ensures that each data is tested at least once and training is done adequately. So, in this project, the model evaluation technique that would be used is 10-fold validation.
7. Generating Insights: This is when insights are generated and predicted yields can be estimated from a future perspective.

This methodology will include both qualitative and quantitative approach. Where the qualitative approach includes the analysis and pre-processing of data and quantitative involves the use of data analytics tool like Python to build models. Using Python. programming which is one of the most suitable programming language in data science.

Research Methodology

The research methodology carried out in this project involves the use several machine learning algorithms and trying to evaluate the best performing algorithm that can be used to improve yield prediction.

The datasets were collected from five different tomato plantation farmers in South Korea and these datasets contains the same amount of data attributes and features. The first three columns contain the ID number, the weeks of growth and the yield outcome (Target Variable). Every other variable are the predictors which include both internal and external factors that influences crop growth.

Dataset Description and Techniques used

There are two types of recordings can be identified from this data set. Those are environmental recordings and plant factors. All the attributes summarised in the Table 1 below.

Outside data:

Attributes x1 to x4 represent outside environment factors. They are average temperature for 24hours [average outside temperature for many years and aver- age outside temperature of this year (year not specified) in Korean], maximum ambient temperature, minimum ambient temperature and 24h Radiation sum(J) [Cumulative solar radiation average for several years and cumulative solar radiance per day (average cumulative irradiance per day * 7) in Korean.

Inside environment data:

Recording of environment data inside the plant house can be categorise in to six main factors. They are temperature, humidity, CO2 level, water, EC and pH. Average temperature for 24 hours, average temperature(day) and average temperature(night) represented by x5 to x7 respectively. x8 to x11 represent average humidity(day), average humidity (night), maximum humidity and minimum humidity reactively. CO2 level day(ppm) represented by x12. Next ten attributes about water supply, intake and derange. They are x13 to x22 accordingly Water (gift-dripper), Water (gift-no), Water (gift), Water (gift)/, Water (drain)/slab, Water (drain)/, Water (cc/J)/, Water uptake/, Water drain (c- c/J)/and Water (drain/gift). Gift EC (Electric conductivity?) [this is not clear (G-EC Supply Average, in Korean)]and Gift pH represent attributes x23 and x24 respectively. x25 and x26 are Slab EC and Slab pH. Here, the slab means these plants are grown on the slab.

Plant data:

Several factors of the plants were recorded throughout the year. No data for some attributes during the remaining few weeks. Plant growth, leaf measurements, fruiting and harvest are the main observations. x27 and x28 are growth length and cumulative growth length. Leaf measurements were recorded in x29 to x31. they are length of leaf, width of leaf and no of leaves respectively. Thickness of stem recorded

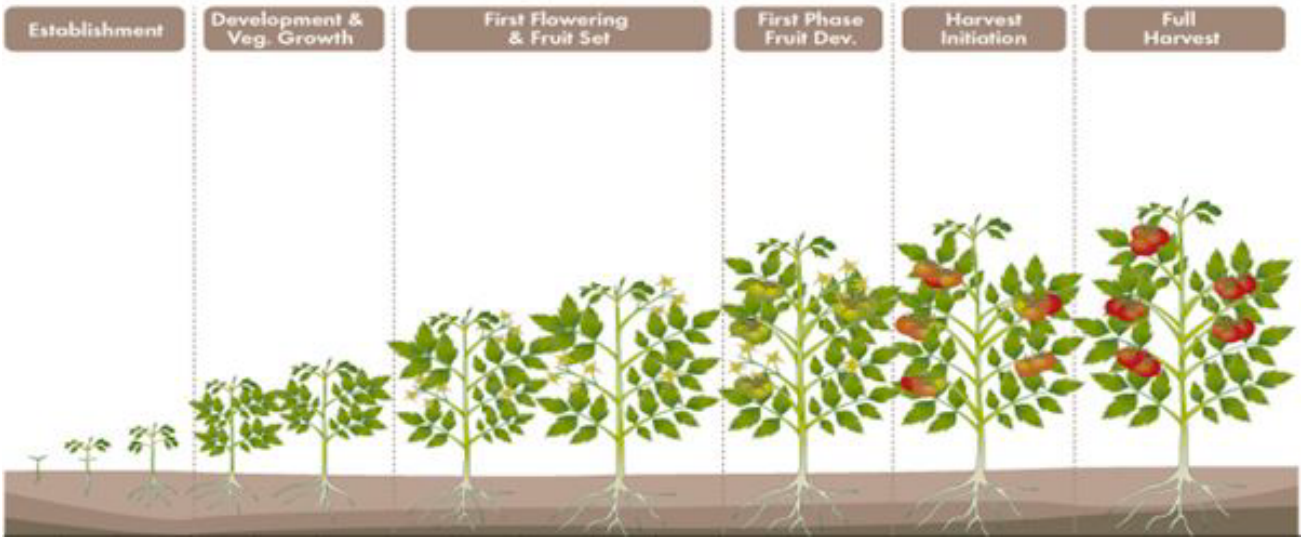
in x32.Days of yield, x37; no of fruits, x38; no of fruits per unit, x39 and fruit factor in x43. However, the 'factor' is not defined. Flowering speed and set speed (fruit set) were recorded in x40 and x41 but the method of speed taken unknown. Next attribute is Leaf Area Index (LAI) it is described as follows;

$$LAI = \frac{(\text{Leaf length})\ (\text{Leaf width})\ (0.5)\ (\text{Leaf number})\ (\text{DENSITY})}{(10000) + (\text{Penetration area})}$$

Attributes x44 and x45 explained as trans1 and trans2 accordingly.
Trans 1 = 24hr Radiation sum (J) * Transmittance: (Transmittance = 0.85)
Trans 2 = (LAI)3 - (0.133 * Leaf area (LAI)2) +(0.606 * Leaf area (LAI)+0.003)
PED (attribute x46) also explained in formula:
PED = ((Number of fruits=)
DENSITY Factor reference light + basic metabolism 7
The last attribute x47 explained average weight per unit.
*LAI is a potent factor in plant physiology
* x34, x35 and x36 critical factors according to Pro Na and they are tabulated in different file.
Following figures give more details about tomato plant growth, plant parts and the measurements.

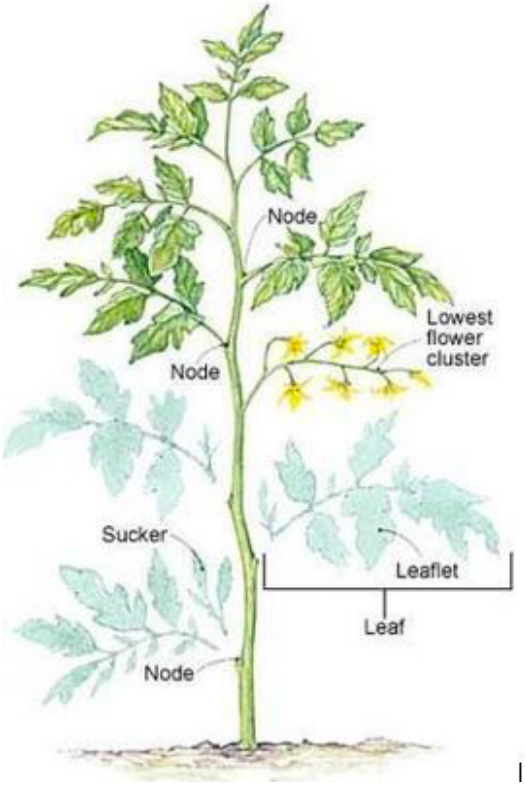
Growth

Figure 4



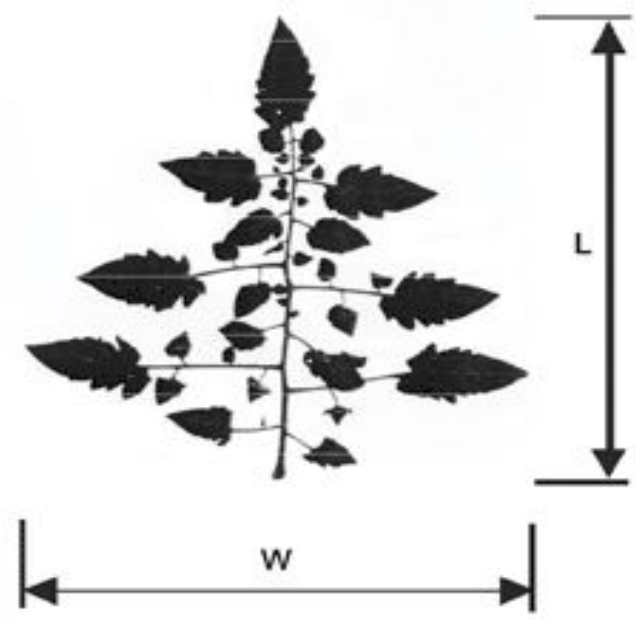
Plant Parts

Figure 5



Plant Measurements

Figure 6



The dataset description table is seen in the Figure below

Table 1: Tomato attributes summary

Attribute No	Attribute name	Description	Data type
1	ID		
2	week	week of the year	Time
3	y	Yield	plant Factor
4	x1	average temperature for 24hours	Outside environment
5	x2	maximum ambient temperature	Outside environment
6	x3	minimum ambient temperature	Outside environment
7	x4	24h Radiation sum(J)	Outside environment
8	x5	Average temperature for 24 hours	Inside environment
9	x6	average temperature(day)	Inside environment
10	x7	average temperature(night)	Inside environment
11	x8	average humidity(day)	Inside environment
12	x9	average humidity (night)	Inside environment
13	x10	average maximum humidity	Inside environment
14	x11	average minimum humidity	Inside environment
15	x12	average CO2 level day(ppm)	Inside environment
16	x13	Water (gift-driper)	water supply
17	x14	Water (gift-no)	water supply
18	x15	Water (gift)	water supply
19	x16	Water (gift)/	water supply
20	x17	Water (drain)/slab	water supply
21	x18	Water (drain)/	water supply
22	x19	Water (cc/J)/	water supply
23	x20	Water uptake/	water supply
24	x21	Water drain (cc/J)/	water supply
25	x22	Water (drain/gift)	water supply
26	x23	Gift EC	water supply
27	x24	Gift pH	water supply
28	x25	Slab EC	water supply
29	x26	Slab pH	water supply
30	x27	growth length	Plant factors
31	x28	cumulative growth	Plant factors
32	x29	length of leaf	Plant factors
33	x30	width of leaf	Plant factors
34	x31	no of leaves	Plant factors
35	x32	Thickness of stem	Plant factors
36	x33	height of flower	Plant factors

Table 2: Tomato attributes summary

Attribute No	Attribute name	Description	Data type
37	x34*		Plant factors
38	x35*		Plant factors
39	x36*		Plant factors
40	x37	Days of yield	Plant factors
41	x38	no of fruits	Plant factors
42	x39	fruit factor per unit	Plant factors
43	x40	Flowering speed	Plant factors
44	x41	set speed	Plant factors
45	x42	$LAI = \frac{(Leaflength)*(Leafwidth)*(0.5)*(Leafnumber)*(DENSITY)}{(10000)+(Penetrationarea)}$	Plant factors
46	x43	Factor of fruit	Plant factors
47	x44	$Trans1 = 24hrRadiationsum(J) * Transmittance : (Transmittance = 0.85)$	Plant factors
48	x45	$Trans2 = (LAI)3 - (0.133 * Leafarea(LAI)2) + (0.606 * Leafarea(LAI) + 0.003)$	Plant factors
49	x46	$PED = \frac{((Numberoffruits/)}{DENSITY} * Factor * referencelight + basicmetabolism * 7$	Plant factors
50	x47	average weight per unit	Plant factors

Problem Analysis

The problem analysis stage reveals the major research problems and the statistical steps carried out in preparing the data for processing. This stages shows how the data was read into python and how the pre-processing procedures were carried out before finally building a model and then evaluating results.

Applying Machine Learning Approaches

The programming language selected for this data analysing the dataset is Python. And all data cleaning, pre-processing and analysis were done with the python framework and analytics libraries. The major machine learning approaches are broken down below:

1. Data pre-processing
2. Applying PCA to reduce dimensions
3. Applying Regression and other algorithms with the selected components after PCA
4. Apply K-fold Cross-validation to data without PCA and build XGBOOST algorithm
5. Apply other algorithms like Random Forest and SVM
6. Apply LASSO Feature Selection Attribute to select important features
7. Build models with selected Features
8. Check for the best performing model of all models and use for Prediction

Step 1. Data Pre-processing

In this stage of data analysis, data cleaning and transformation is done to prepare data for proper statistical analytical processing and model building. Firstly, the data is read into python using Panda library and then stored in a variable. After reading the data, the shape of the data is checked and also missing values. All missing values are replaced with zeros as this represents no yield at those times. This is seen in the image below.

Dataset before Removing N/A

Figure 7

	week	y	x1	x2	x3	x4	x5	x6	x7	x8	...	x39	x40	x41	x42	x43
0	32	NaN	0.000000	0	0	0.000000	0.000000	0.000000	0.000000	0.000000	...	0.000000	0.00000	0.0000	0.000000	0.000000
1	33	NaN	0.000000	0	0	0.000000	0.000000	0.000000	0.000000	0.000000	...	0.000000	0.00000	0.0000	0.000000	0.000000
2	34	NaN	0.000000	0	0	0.000000	0.000000	0.000000	0.000000	0.000000	...	0.000000	0.00000	0.0000	0.000000	0.000000
3	35	NaN	0.000000	0	0	0.000000	0.000000	0.000000	0.000000	0.000000	...	0.000000	0.00000	0.0000	0.000000	0.000000
4	36	NaN	0.000000	0	0	0.000000	0.000000	0.000000	0.000000	0.000000	...	0.000000	0.00000	0.0000	0.000000	0.000000
5	37	NaN	0.000000	0	0	0.000000	0.000000	0.000000	0.000000	0.000000	...	0.000000	0.00000	0.0000	0.000000	0.000000
6	38	NaN	0.000000	0	0	0.000000	0.000000	0.000000	0.000000	0.000000	...	0.000000	0.00000	0.0000	0.000000	0.000000
7	39	NaN	21.833333	32	11	12638.000000	21.285714	30.571429	17.666667	48.500000	...	10.902933	3.00000	2.1250	2.407731	0.550000
8	40	NaN	19.571429	32	8	10040.000000	20.571429	26.000000	18.714286	76.285714	...	26.045894	1.00000	1.1250	3.557824	0.985417
9	41	NaN	16.250000	31	8	11110.750000	19.500000	25.750000	17.250000	64.250000	...	27.257331	1.00000	1.2500	3.853392	0.825000
10	42	NaN	12.833333	25	2	9341.500000	17.833333	23.166667	15.500000	61.333333	...	35.131672	0.25000	0.1875	3.166586	1.012698
11	43	NaN	15.428571	26	4	7651.000000	18.714286	24.285714	16.428571	73.142857	...	0.000000	0.00000	0.0000	0.000000	0.000000
12	44	NaN	12.000000	22	1	9483.600000	17.400000	23.000000	15.000000	59.500000	...	0.000000	0.00000	0.0000	0.000000	0.000000
13	45	0.213183	10.428571	19	1	6584.000000	17.142857	22.142857	15.714286	65.000000	...	56.331818	9.00000	8.0000	2.076842	0.990278
14	46	0.648232	7.571429	15	1	6823.000000	16.714286	22.000000	15.000000	69.428571	...	54.514663	1.49500	1.1650	4.063160	0.821820
15	47	0.732154	7.285714	17	1	5753.000000	16.857143	22.714286	15.000000	68.428571	...	54.514663	0.00000	0.0000	4.063160	0.821820

Dataset after Removing N/A

Figure 8

	week	y	x1	x2	x3	x4	x5	x6	x7	x8	...	x39	x40	x41	x42	x43	
0	32	0.000000	0.000000	0	0	0.000000	0.000000	0.000000	0.000000	0.000000	...	0.000000	0.00000	0.0000	0.000000	0.000000	0.0
1	33	0.000000	0.000000	0	0	0.000000	0.000000	0.000000	0.000000	0.000000	...	0.000000	0.00000	0.0000	0.000000	0.000000	0.0
2	34	0.000000	0.000000	0	0	0.000000	0.000000	0.000000	0.000000	0.000000	...	0.000000	0.00000	0.0000	0.000000	0.000000	0.0
3	35	0.000000	0.000000	0	0	0.000000	0.000000	0.000000	0.000000	0.000000	...	0.000000	0.00000	0.0000	0.000000	0.000000	0.0
4	36	0.000000	0.000000	0	0	0.000000	0.000000	0.000000	0.000000	0.000000	...	0.000000	0.00000	0.0000	0.000000	0.000000	0.0
5	37	0.000000	0.000000	0	0	0.000000	0.000000	0.000000	0.000000	0.000000	...	0.000000	0.00000	0.0000	0.000000	0.000000	0.0
6	38	0.000000	0.000000	0	0	0.000000	0.000000	0.000000	0.000000	0.000000	...	0.000000	0.00000	0.0000	0.000000	0.000000	0.0
7	39	0.000000	21.833333	32	11	12638.000000	21.285714	30.571429	17.666667	48.500000	...	10.902933	3.00000	2.1250	2.407731	0.550000	10742.3
8	40	0.000000	19.571429	32	8	10040.000000	20.571429	26.000000	18.714286	76.285714	...	26.045894	1.00000	1.1250	3.557824	0.985417	8534.0
9	41	0.000000	16.250000	31	8	11110.750000	19.500000	25.750000	17.250000	64.250000	...	27.257331	1.00000	1.2500	3.853392	0.825000	9444.1
10	42	0.000000	12.833333	25	2	9341.500000	17.833333	23.166667	15.500000	61.333333	...	35.131672	0.25000	0.1875	3.166586	1.012698	7940.2
11	43	0.000000	15.428571	26	4	7651.000000	18.714286	24.285714	16.428571	73.142857	...	0.000000	0.00000	0.0000	0.000000	0.000000	6503.3
12	44	0.000000	12.000000	22	1	9483.600000	17.400000	23.000000	15.000000	59.500000	...	0.000000	0.00000	0.0000	0.000000	0.000000	8061.0
13	45	0.213183	10.428571	19	1	6584.000000	17.142857	22.142857	15.714286	65.000000	...	56.331818	9.00000	8.0000	2.076842	0.990278	5596.4
14	46	0.648232	7.571429	15	1	6823.000000	16.714286	22.000000	15.000000	69.428571	...	54.514663	1.49500	1.1650	4.063160	0.821820	5799.5
15	47	0.732154	7.285714	17	1	5753.000000	16.857143	22.714286	15.000000	68.428571	...	54.514663	0.00000	0.0000	4.063160	0.821820	4890.0
16	48	0.722830	10.571429	17	2	5199.000000	16.571429	21.428571	15.000000	77.142857	...	50.274633	0.41750	0.8325	4.263955	0.728904	4419.1
17	49	0.458521	3.000000	13	-5	4121.000000	15.857143	20.571429	15.000000	81.428571	...	52.091789	0.75000	0.2525	3.422385	0.553055	3502.8

A panda data frame is created with Target specified. This is done so as to be able to set the Target data during training and testing. After this is done, the data is normalized so as to allow all features to be in the same range of scale for easy analysis and this also helps to reduce bias in the model building phase. The formula used is done by first calculating the mean of each column and then also calculating the standard deviation of each column. So, for each cell, the new value will be the difference between the

actual value minus mean, all divided by the standard deviation. This is evident in the screenshot below:

Figure 9

```
mu    = df.mean(axis=0) # mean of each col
sigma = df.std(axis=0)  # std dev of each col

Xnorm = (df - mu)/sigma
print Xnorm
```

```
1   0.362922 -1.232147 -1.362064 -2.239464 -0.486737 -2.129202 -2.324188
2   0.428908 -1.232147 -1.362064 -2.239464 -0.486737 -2.129202 -2.324188
3   0.494894 -1.232147  1.418878  1.305360  1.612316  0.772087  1.380080
4   0.560880 -1.232147  1.272014  1.026322  1.439930  0.649821  1.220904
5   0.626866 -1.232147  1.246747  1.098665  1.429790  0.739794  1.132473
6   0.692852 -1.232147  0.916698  0.726614  0.719965  0.560050  0.702110
7   0.758838 -1.232147  0.831422  0.788622  0.750386  0.868560  0.647087
8   0.824824 -1.232147  0.913540  0.943644  0.983614  0.602092  0.645121
9   0.890809 -1.232147  0.861426  1.067661  0.598281  0.392899  0.637261
10  0.956795 -1.232147  0.493477  0.530253  0.365053  0.534460  0.495772
```

Step 2. Applying PCA to Reduce Dimension

During the collection of data, virtually all physical and environmental factors influencing Tomato yield were collected and added to the dataset. But as it's known, the more the feature the more tendency for the predictive model to overfit. This is because, as more and more data are added as feature, the predictive model would always want to fit its line on all points and this could include noise as well. In this situation, the model will tend to overfit. This would indicate great performance during training. But if a data outside the training set is used to test the model, it's will perform horribly bad.

In order to avoid this, PCA is applied. Principal component analysis is a dimensionality reduction technique that allows for the selection of principal components that which when selected improves model accuracy and as well avoid overfitting. PCA reduces the number of features to be included in the training set. The selection criteria are often based on features with the highest variation or covariance.

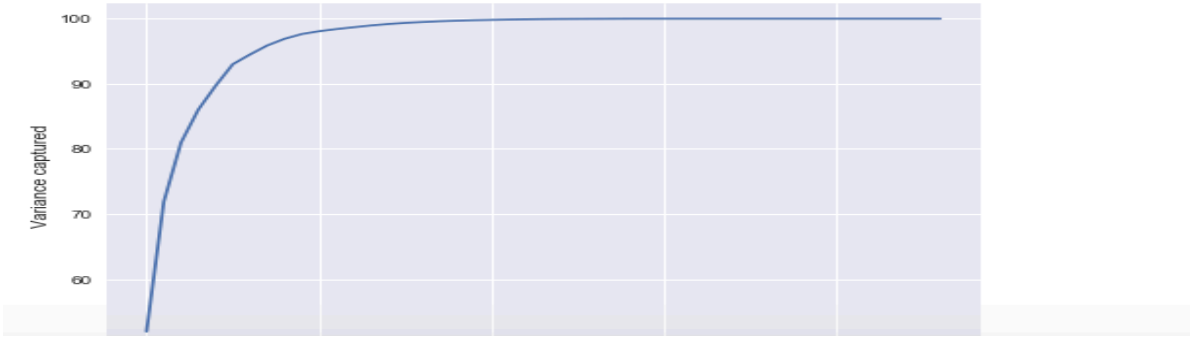
After data normalization, the matrix Z created is transposed. Then the eigen vectors and their corresponding eigen values are calculated as well. After this, the eigen values are sorted from largest to smallest. Then the features are selected by calculating the proportion of variance explained for each feature, pick a threshold and add features until their cumulative addition hit a good threshold (often between 90-95). Ultimately, the PCA give a perfect orthogonal representation of each feature to another one. This represents high independence amongst features and ensures that there is no multi-

collinearity with target variable. The selection of the feature for the tomato data is seen below:

Figure 10

```
#Cumulative Variance explains
var1=np.cumsum(np.round(pca.explained_variance_ratio_, decimals=4)*100)
print var1
plt.plot(var1)
plt.xlabel("Principal components")
plt.ylabel("Variance captured")
```

```
[ 52.    71.9   80.97  86.01  89.66  92.97  94.47  95.85  96.87  97.6
  98.03  98.36  98.64  98.91  99.13  99.31  99.44  99.56  99.65  99.72
  99.77  99.82  99.86  99.89  99.91  99.92  99.93  99.94  99.95  99.95
  99.95  99.95  99.95  99.95  99.95  99.95  99.95  99.95  99.95  99.95
  99.95  99.95  99.95  99.95  99.95  99.95  99.95]
<matplotlib.text.Text at 0x1194d1bd0>
```



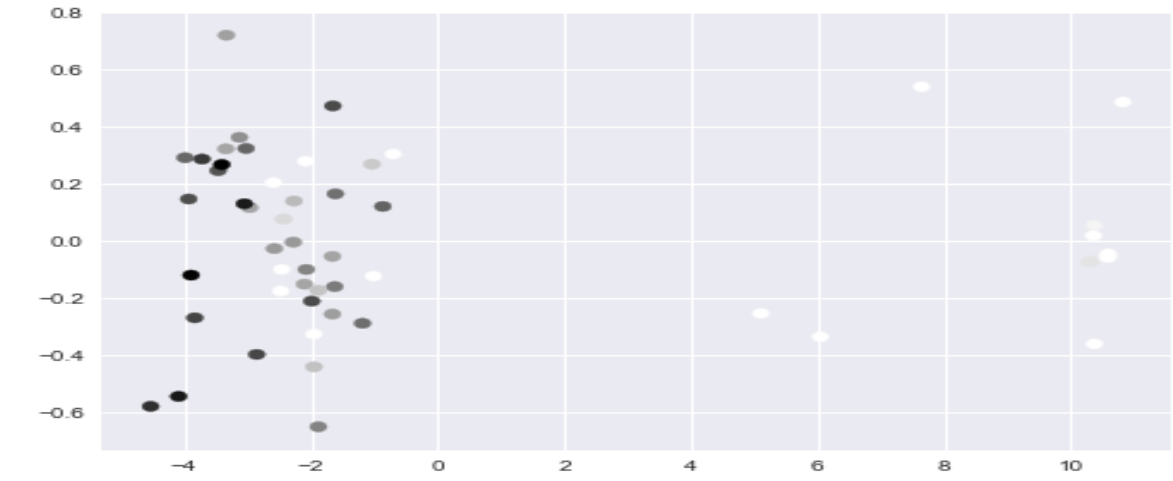
Step 3. Applying Regression to Selected Components

After applying PCA, 15 components were selected based on their threshold and then used to train the model. The target variable was included as well in training the model. The plot below shows a scatter plot of the 15 selected components.

Figure 11

```
import pylab as pl
pl.scatter(Zred[:, 0], Zred[:, 15], c=df.target )
```

```
<matplotlib.collections.PathCollection at 0x119b066d0>
```



The scatter plot shows all the selected components that were selected via PCA through their high covariance. The next step reveals the training stage of the selected variables. 60 percent of the data were split into training while the remaining 40 percent were split into test set. The outcome of the training is seen below.

Figure 12

```
import pandas as pd
from sklearn import datasets, linear_model
from sklearn.model_selection import train_test_split
from matplotlib import pyplot as plt
Xe = Zred
ye = dfNorm.target.values

Xe_train, Xe_test, ye_train, ye_test = train_test_split(Xe, ye, test_size=0.4, random_state=500)


# fit a model
lm = linear_model.LinearRegression()
model = lm.fit(Xe_train, ye_train)
predictions = lm.predict(Xe_test)
predictions[0:50]
```

```
array([-1.4764024 ,  1.22104949,  1.10874703,  1.38489894, -0.68458477,
        2.12706704,  2.27229877,  4.21602473,  0.84644918, -2.7148482 ,
       -1.23214726, -0.22928963,  0.32891567,  1.18753333, -2.31331086,
       -0.44544969,  0.39494067, -0.14252473,  0.7491898 , -3.18587075,
        0.6906357 ])
```

The above diagram shows how the Selected principal components are used to train the linear model using the train_test_split library from sklearn. The ouputs below shows the output of the prediction. The accuracy therefore, is not as good as we were only able to achieve an 80 percent accuracy.

Results

The results for this project work is highlighted from steps 4 to step 5 where the output of the various prediction accuracies were previewed and accompanied by the selection of the best algorithm.

Step 4. Applying 10-fold Cross-validation to data without PCA and build XGBOOST algorithm

After a not too successful prediction using PCA, another approach was attempted. This time it involves all the data from the original data set and applying K-fold cross validation (a model evaluation technique). This technique would ensure a better model being built. 10-fold cross validation approach was applied. Meaning that the data is partitioned into 10 equal subsamples. Of the 10 subsamples, a single subsample is retained and used as the validation set for testing the model. So, the remaining 9

subsamples are used as training set. This continues until all subsamples are used for training and one subsample is used for testing at least once. After cross validation, other machine learning algorithm can then be used for prediction.

Here a very versatile machine learning algorithm will be implemented and then evaluation will be done on the model performance.
XGBoost algorithm is a machine learning algorithm often used in training data with multiple features. In this case, if implemented well, it'll give a very good output with optimal prediction accuracy. Another of its advantages is its natural ability to control overfitting of a predictive model. It also a term that refers to an ensemble of trees an always often applies some penalties to improve accuracy. This is all seen in the diagram below:

Figure 13

Kfold with XGboost

```
from sklearn.model_selection import KFold
from sklearn.model_selection import cross_val_score
import xgboost
import math
from __future__ import division
from scipy.stats import pearsonr
from sklearn.linear_model import LinearRegression
from sklearn import cross_validation, tree, linear_model
from sklearn.model_selection import train_test_split
from sklearn.metrics import explained_variance_score
```

Figure 14

```
from sklearn import cross_validation
from sklearn.model_selection import KFold


lm = linear_model.LinearRegression()
cv = cross_validation.KFold(len(X), n_folds=10)
xgb = xgboost.XGBRegressor(n_estimators=100, learning_rate=0.08, gamma=0, subsample=0.75,
                           colsample_bytree=1, max_depth=7)
results = []
# "Error_function" can be replaced by the error function of your analysis
for Xcv, ycv in cv:
        probas = xgb.fit(X[Xcv], y[Xcv])
        predictions = probas.predict(X[ycv])
        #results.append( Error_function )
#print "Results: " + str( np.array(results).mean() )
print model.fit(X[Xcv], y[Xcv]).score(X[ycv], y[ycv])
```

```
1.0
```

From the above, after implementing 10-fold cross validation, it is realised that the prediction accuracy was 100%. This signifies the efficiency of cross validation when used for training and validation test.
Below is an image of XGboost implemented to the ordinary data without applying cross validation. The accuracy in this case is lesser than that which was done with cross validation. The figure is seen below.

Figure 15

Xgboost

```
xgb = xgboost.XGBRegressor(n_estimators=100, learning_rate=0.08, gamma=0, subsample=0.75,
                           colsample_bytree=1, max_depth=7)
```

```
traindf, testdf = train_test_split(X_train, test_size = 0.3)
xgb.fit(X_train,y_train)
```

```
XGBRegressor(base_score=0.5, colsample_bylevel=1, colsample_bytree=1, gamma=0,
       learning_rate=0.08, max_delta_step=0, max_depth=7,
       min_child_weight=1, missing=None, n_estimators=100, nthread=-1,
       objective='reg:linear', reg_alpha=0, reg_lambda=1,
       scale_pos_weight=1, seed=0, silent=True, subsample=0.75)
```

```
predictions = xgb.predict(X_test)
print(explained_variance_score(predictions,y_test))
```

```
0.998219962588
```

Step 5. Applying other algorithms like Random Forest and SVM

Looking at applying other types of machine learning algorithm, Random forest was selected to check how it's predictive performance will be with the given dataset. Firstly, Random forest random forest is a combination of tree predictors. It uses averaging to improve predictive accuracy and controls overfitting. it's basically a group of weak learners that come together to form a stronger one.

Random Forest is very useful because of the following: 1) It improves predictive accuracy. 2) It runs efficiently on large data sets. 3) It can handle thousands of input variables without deleting or omitting any one. 4) It gives estimates of what variables are important in the classification 5) It controls overfitting.

Therefore, after applying random forest algorithm, the model seemed to have performed better than PCA with regression. The figure below shows the output of the predictive model accuracy after being trained with random forest:

Figure 16

Random Forest

```
import pandas as pd
import numpy as np
from sklearn.preprocessing import Imputer
from sklearn.ensemble import RandomForestClassifier, RandomForestRegressor, ExtraTreesRegressor, GradientBoostingRegres
from sklearn.cross_validation import train_test_split
from sklearn.metrics import accuracy_score
from sklearn import tree

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size = 0.2)

clf = RandomForestRegressor(n_estimators=10)

clf.fit(X_train, y_train)
y_pred = clf.predict(X_test)
#accuracy_score(y_test, y_pred)
clf.score(X_test, y_test)
```

```
0.96714209627068304
```

From the above figure, it's seen that the random forest regressor algorithm was used as this problem is a linear problem. After the algorithm was split and trained using the train_test_split function, the model was fitted and built. The accuracy score given after testing the trained data was 96%. This is also another alternative algorithm that could be used.

Step 5. Applying LASSO Feature Selection Attribute to select important features

Regularization is another important aspect of statistics that helps to avoid overfitting. Regularization adds complexity penalty to the loss function by adding a multiple of an L1L1 (LASSO) or an L2L2(Ridge) norm of your weights vector ww (it is the vector of the learned parameters in the linear regression). All this is done so as to reduce overfitting. There are two types of regularization. They are L1 regularisation (LASSO) and L2 regularization (ridge).

LASSO regularization also known as **L1 regularisation is used in this project** and used to reduce the number of features used in training the model. The model selection process is often based on some factors. This is done in such a way that it imposes a constraint on the model parameters that causes regression coefficients for some variables to shrink toward zero. Variables with a regression coefficient equal to zero after the shrinkage process are excluded from the model. Variables with non-zero regression coefficients variables are most strongly associated with the response

variable. This is seen in the figure below where after applying LASSO, the output of the selected data with coefficients is displayed:

Figure 17

Feature Selection With Lasso

```
X = df_.values
y = dfNorm.target.values

from sklearn.cross_validation import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.4)
names = df_.columns
from sklearn.cross_validation import train_test_split
from sklearn.linear_model import LassoLarsCV
# specify the lasso regression model
model=LassoLarsCV(cv=10, precompute=False).fit(X,y)

# print variable names and regression coefficients
dict(zip(names, model.coef_))
```

The above code shows how LASSO was used with python on the tomato data. First, the X which signifies the predictors and y, which signifies the target were trained using the train_test_split function. Then the variable names were called from the dataframe created to store the data. Next is using the LASSO function with k fold cross validation (in this instance. K=10). After running the code, the output below was returned.

Figure 18

```
'x24': 0.0,
'x25': 0.0,
'x26': 0.0,
'x27': 0.0,
'x28': 0.13564905213830614,
'x29': 0.0,
'x3': 0.0,
'x30': 0.0,
'x31': 0.0,
'x32': 0.0,
'x33': 0.0,
'x34': 0.0,
'x35': 0.0,
'x36': 0.0,
'x37': 0.59757583002183434,
'x38': 0.0,
'x39': 0.0,
'x4': 0.0,
'x40': 0.0,
'x41': 0.0,
'x42': 0.0,
'x43': 0.093255302261392906,
'x44': 0.0,
'x45': 0.0,
'x46': 0.1179478262790566,
'x47': 0.0,
'x5': 0.0,
'x6': 0.0,
'x7': 0.0,
'x8': 0.0,
'x9': 0.0}
```

The above figure shows the variables that were shrinked towards zero have been dropped and the remaining variables with coefficients were retained and used for building the model.

In an attempt to look for more variables that wouldn't have been selected using LASSO, the **Randomized LASSO** was used. Randomised lasso carries out its selection feature in a more randomised manner and it tends to implement the strategies of regression setting. It's more like a method that alleviates the problems of ordinary LASSO. The outcome of the method is seen in the figure below:

Figure 19

Randomised LASSO

```
#*************************************************L1 Regularization*******************************************************#
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import accuracy_score
from sklearn.metrics import classification_report
from sklearn.metrics import confusion_matrix
from sklearn.metrics import recall_score
from sklearn.metrics import precision_score
from sklearn.linear_model import RandomizedLasso

# For L1 regularization, alpha is given as 0.025
from sklearn.cross_validation import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.4)

rlasso = RandomizedLasso(alpha=0.025)
rlasso.fit(X_train, y_train)
names = df_.columns

#Now, we'll  be checking for the classification accuracy

print "Features sorted by their score:"
print sorted(zip(map(lambda x: round(x, 6), rlasso.scores_),
                 names), reverse=True)
```

```
Features sorted by their score:
[(0.525, 'x37'), (0.475, 'x28'), (0.39, 'x46'), (0.275, 'x34'), (0.2, 'x39'), (0.2, 'x38'), (0.16, 'x14'), (0.15, 'x3
5'), (0.14, 'x36'), (0.125, 'x26'), (0.07, 'x43'), (0.07, 'x41'), (0.055, 'x47'), (0.03, 'x45'), (0.03, 'x31'), (0.03
, 'x12'), (0.02, 'x17'), (0.015, 'x40'), (0.015, 'x3'), (0.015, 'x18'), (0.015, 'x16'), (0.01, 'x42'), (0.01, 'x24'),
(0.005, 'x22'), (0.005, 'x15'), (0.005, 'week'), (0.0, 'x9'), (0.0, 'x8'), (0.0, 'x7'), (0.0, 'x6'), (0.0, 'x5'), (0.
0, 'x44'), (0.0, 'x4'), (0.0, 'x33'), (0.0, 'x32'), (0.0, 'x30'), (0.0, 'x29'), (0.0, 'x27'), (0.0, 'x25'), (0.0, 'x2
3'), (0.0, 'x21'), (0.0, 'x20'), (0.0, 'x2'), (0.0, 'x19'), (0.0, 'x13'), (0.0, 'x11'), (0.0, 'x10'), (0.0, 'x1')]
```

As seen in the figure above, up to 15 feature were selected using Randomised LASSO. This signifies it's preferredness to ordinary LASSO to solve some machine learning task.

Conclusion

After implementing several machine learning algorithms on the tomato dataset, so many insights were drawn from the predictive models. It's evidently shown from the application of PCA and LASSO that not all the features in the dataset directly influence the yield of the tomato plants. For instance, some insights gotten from LASSO selection feature implies that variables with label "x28", "x37", "x43", "x46" have a very significantly high influence on yield "y". Looking at the actual label of the data from the data dictionary, we'll see the true labels below.

x28 represents **cumulative length of growth**
x37 **r**epresents **days of yield**
x43 represents **factor of fruit**
x46 represents **PED** (number of fruits/density)

From obvious and research indications, we can conclude that these factors that were selected by LASSO seem to be genuinely right when compared to past researches and facts. For instance, the cumulative length of growth of a plant signifies that the plant is actually growing to maturity. This is one of the most undisputed physical fact about

determining a plant's growth [23]. Also, days of yield, which shows the number of days of the tomato plant yield also makes sense in the real world. Factor of fruit and the number of fruits also seem to be highly significant. It's proven that the more the yield of a particular plant, the longer it takes to produce yield. Some plants that produces one yield per plant often produces yield faster.

The PCA approach also reduced the number of features that were used to build the predictive model, making it easier to work with smaller dataset and thus reducing the risks of overfitting.

In conclusion, the research tries to implement and evaluates several supervised learning algorithms, it shows how accurate they can be when applied to the tomato dataset and the best machine learning algorithm to use. For this research, the **10-fold cross validation technique with Xgboost algorithm** far out performs every other approach used. The output is once again seen below:

Figure 20

```
from sklearn import cross_validation
from sklearn.model_selection import KFold


lm = linear_model.LinearRegression()
cv = cross_validation.KFold(len(X), n_folds=10)
xgb = xgboost.XGBRegressor(n_estimators=100, learning_rate=0.08, gamma=0, subsample=0.75,
                           colsample_bytree=1, max_depth=7)
results = []
# "Error_function" can be replaced by the error function of your analysis
for Xcv, ycv in cv:
        probas = xgb.fit(X[Xcv], y[Xcv])
        predictions = probas.predict(X[ycv])
        #results.append( Error_function )
#print "Results: " + str( np.array(results).mean() )
print model.fit(X[Xcv], y[Xcv]).score(X[ycv], y[ycv])
```

```
1.0
```

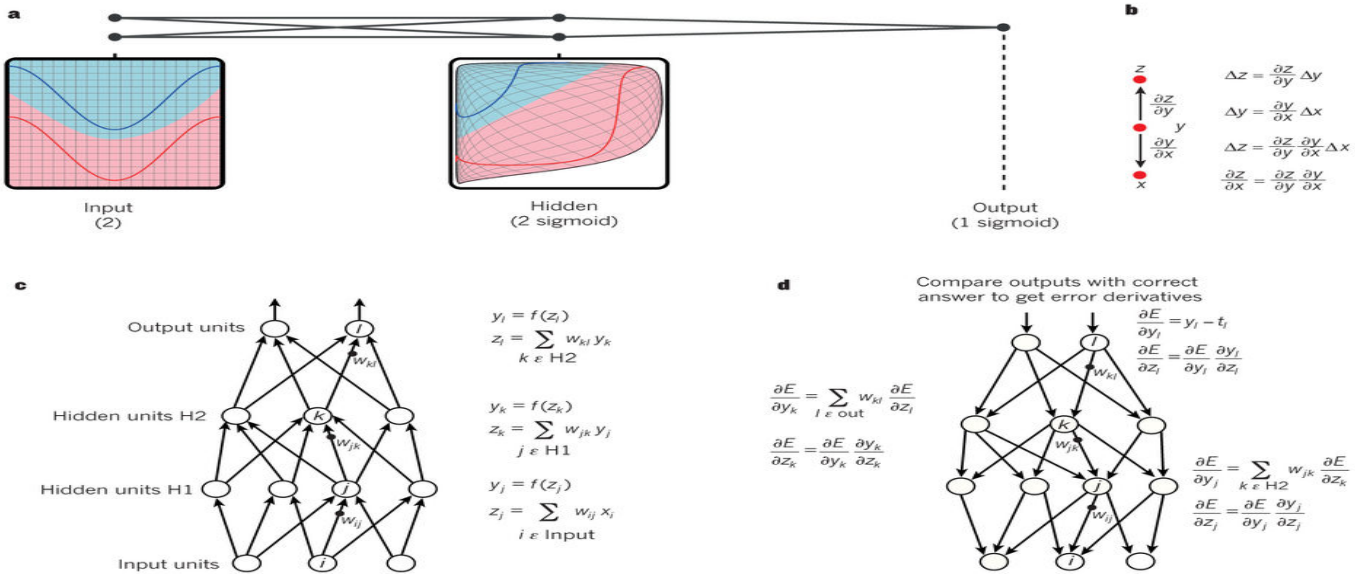
The result from this gives a 100% prediction accuracy. This can be further utilised for future data.

Future Work

The use of regression, XGboost, random forest and SVM algorithms seem not to be very effective in the in trying to implement a very accurate predictive algorithm. Looking at the tomato dataset, it's seen that data is collected weekly and this continues for as long as possible until yields are produced. This could go on for years and more insights would be needed to automatically be generated as more data is collected [21]. The best machine learning algorithm for this problem is Deep Learning. Deep Learning is an area of machine learning that uses Neural network (artificial intelligence) approach to learn. It works tremendously well with growing data size. It's also a hybrid machine learning approach that can be used for both unsupervised and supervised learning tasks. Deep learning allows computational models that are composed of multiple processing layers to learn data representation with diverse levels of abstractions. In large datasets, it uses back propagation techniques to discover intricate structures about the data.

In our tomato dataset, there's definitely going to be increase in the data as years run by and an algorithm as robust and versatile as deep learning will be very important to train the data and to discover intricate structures in the data [21]. Representation learning is also another area that's applied to deep learning that is pivotal to smart farming. Not in all case will collected data will be labelled. In situation where unlabelled farm data are needed for analysis, Deep learning algorithm can be very pivotal. Here, raw data are fed into the algorithm and then it automatically discovers the representations needed for detection and classification.

Figure 21



The figure above shows the working mechanism of deep learning algorithm. The algorithm comprises of several hidden layer of units that ensure proper training of the data.
Appropriate use of deep learning technique will solve most problems of the current machine learning algorithms.
Also, for application of deep neural network, google developed a software library called **Tensor flow**. This library was adopted for solving numerical problems using data flow graphs. This comprises of several nodes which denotes mathematical operations and the edges of the graph represents a multidimensional array of data. This tool will play a major role in applying deep neural network algorithm to solve the tomato data analysis problem.

Also, another approach which is fairly advantageous is the use of **Ensemble learning algorithm**. This is also another approach that is often very effective in getting the best out of models. It works in such a way that it combines several in order to produce improved results. In a more detailed explanation, it combines it constructs a set of different hypotheses and combine them efficiently to generate the best result. It always contains a number of weak learners called the **base learners** and also a set of strong learners. These weak learners, when boosted by the ensemble tend to provide accurate predictions than a random guess by a strong learner. The most commonly used ensemble methods are voting and averaging. Voting is mostly used for solving classification problems while averaging is often used in solving linear problems.

These approaches; Deep learning and Ensemble learning can also be combined together to produce an even much more accurate prediction. This could be applied in such a way that that deep learning could be used to extract features from the dataset and then use the output of the n-1 layer as input to an ensemble of classifiers. This is often regarded as hybrid multiple regressor or classifier system.

Reference

1. Al-Gaadi KA, Hassaballa AA, Tola E, Kayad AG, Madugundu R, Alblewi B, et al. Prediction of Potato Crop Yield Using Precision Agriculture Techniques. PLoS One. 2016;11(9):e0162219.
2. Yamamoto K, Guo W, Yoshioka Y, Ninomiya S. On plant detection of intact tomato fruits using image analysis and machine learning methods. Sensors (Basel). 2014;14(7):12191-206.
3. Mondal P, Tewari VK. Present status of precision farming: A review. Int. J. Agric. Res. 2007;2:1-0.
4. Kaewmard N, Saiyod S. Sensor data collection and irrigation control on vegetable crop using smart phone and wireless sensor networks for smart farm. InWireless Sensors (ICWiSE), 2014 IEEE Conference on 2014 Oct 26 (pp. 106-112). IEEE.
5. Shekhar S, Colletti J, Muñoz-Arriola F, Ramaswamy L, Krintz C, Varshney L, Richardson D. Intelligent Infrastructure for Smart Agriculture: An Integrated Food, Energy and Water System. arXiv preprint arXiv:1705.01993. 2017 May.
6. Gutta A, Sajja PS. Intelligent farm expert multi agent system. International Journal on Computer Science and Engineering. 2012 Feb 1;4(2):166.
7. Cruz F, Pereira A, Valente P, Duarte P, Reis LP. Intelligent farmer agent for multi-agent ecological simulations optimization. InPortuguese Conference on Artificial Intelligence 2007 Dec 3 (pp. 593-604). Springer, Berlin, Heidelberg.
8. Jhuria M, Kumar A, Borse R. Image processing for smart farming: Detection of disease and fruit grading. InImage Information Processing (ICIIP), 2013 IEEE Second International Conference on 2013 Dec 9 (pp. 521-526). IEEE.
9. Yeom TH, Park SM, Kwon HI, Hwang DK, Kim J. A smart farming system based on visible light communications. The Journal of Korean Institute of Communications and Information Sciences. 2013;38(5):479-85.
10. Banhazi TM, Lehr H, Black JL, Crabtree H, Schofield P, Tscharke M, Berckmans D. Precision livestock farming: An international review of scientific and commercial aspects. International Journal of Agricultural and Biological Engineering. 2012 Sep 22;5(3):1-9.
11. Precision Farming - producing more with less [Internet]. Cema-agri.org. 2017. Available from: http://www.cema-agri.org/page/precision-farming-0
12. Wolfert S, Ge L, Verdouw C, Bogaardt MJ. Big Data in Smart Farming–A review. Agricultural Systems. 2017 May 31; 153:69-80.
13. Hashimoto Y, Murase H, Morimoto T, Torii T. Intelligent systems for agriculture in Japan. IEEE Control Systems. 2001 Oct;21(5):71-85.
14. Tripicchio P, Satler M, Dabisias G, Ruffaldi E, Avizzano CA. Towards smart farming and sustainable agriculture with drones. InIntelligent Environments (IE), 2015 International Conference on 2015 Jul 15 (pp. 140-143). IEEE.

15. Jhuria M, Kumar A, Borse R. Image processing for smart farming: Detection of disease and fruit grading. InImage Information Processing (ICIIP), 2013 IEEE Second International Conference on 2013 Dec 9 (pp. 521-526). IEEE.
16. Barnaghi P, Sheth A, Henson C. From data to actionable knowledge: Big data challenges in the web of things [Guest Editors' Introduction]. IEEE Intelligent Systems. 2013 Nov;28(6):6-11.
17. Kanjilal D, Singh D, Reddy R, Mathew PJ. Smart farm: extending automation to the farm level. International Journal of Scientific & Technology Research. 2014 Jul;3(7).
18. Porter ME, Heppelmann JE. How smart, connected products are transforming competition. Harvard Business Review. 2014 Nov 1;92(11):64-88.
19. Diamantoulakis PD, Kapinas VM, Karagiannidis GK. Big data analytics for dynamic energy management in smart grids. Big Data Research. 2015 Sep 30;2(3):94-101.
20. Zhou K, Fu C, Yang S. Big data driven smart energy management: From big data to big insights. Renewable and Sustainable Energy Reviews. 2016 Apr 30;56:215-25.
21. LeCun Y, Bengio Y, Hinton G. Deep learning. Nature. 2015 May 28;521(7553):436-44.
22. Judd LA, Jackson BE, Fonteno WC. Advancements in root growth measurement technologies and observation capabilities for container-grown plants. Plants. 2015 Jul 3;4(3):369-92.