Estimating nutrient mix concentration for an autonomous hydroponic vertical farm

ABSTRACT

Hydroponics is a paradigm of agriculture in which crops grow without the use of soil, instead using water as a medium to provide nutrients. It is known to be less wasteful than conventional agriculture, requiring less land, consuming less water, and emitting lower levels of greenhouse gasses. Hydroponic systems require close monitoring of the system's water to maintain optimal growth conditions, including healthy nutrient levels, which can be difficult to measure directly. This analysis aims to provide a model for estimating the nutrient levels in a hydroponics system actively producing various microgreen crops.

KEYWORDS: Smart agriculture; Hydroponics; Internet of Things; Sustainability

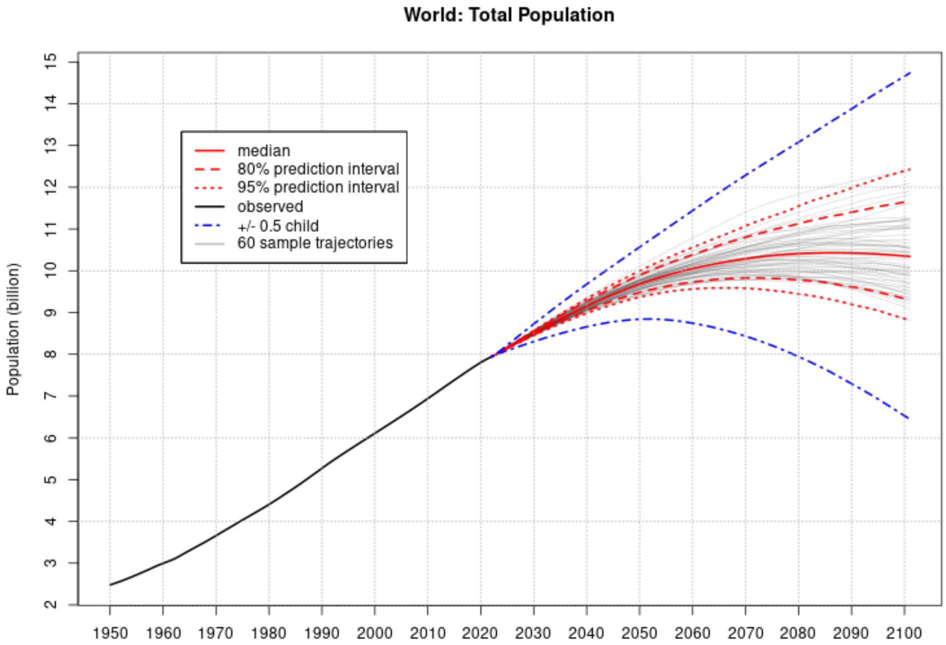
Author Information

- Alex Rosenblum, DePaul University, arosenblum1@gmail.com
- Harnain Kaur Sardarni, DePaul University, hsardar1@gmail.com

INTRODUCTION

While conventional agriculture has sustained the human population sufficiently throughout history, there are several reasons why conventional agriculture alone is not sustainable in the long term when considering its land use, water consumption, and impact on air quality. Pomoni, et al., 2022, states that conventional agriculture occupies 38% of total land used globally, consumes 70% of the world's water resources, and accounts for 13.5-29% of total greenhouse gas emissions.[2]

These issues are further exacerbated by rapid population growth; while current population levels are already straining the global infrastructure required to feed them, the 2022 revision of the World Population Prospects from the United Nations estimates that the world population will reach 10 billion by the year 2060.[1] As populations grow and demand increases, the environmental damage done by the necessities of conventional agriculture will only get worse.


World: Total Population

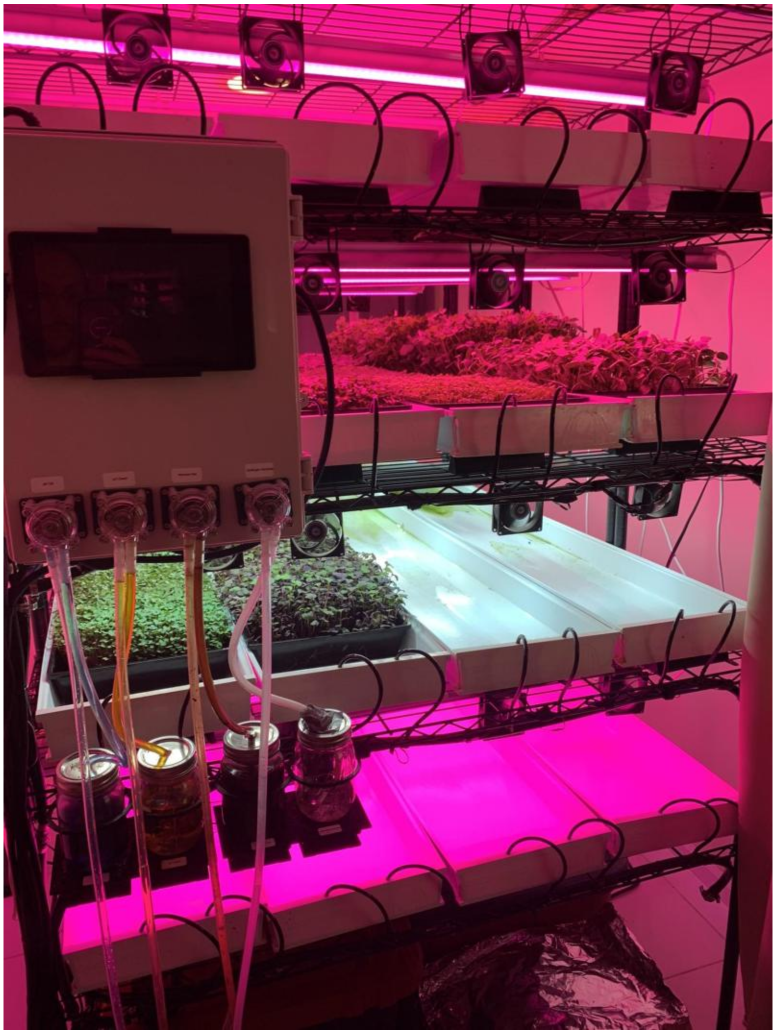
© 2022 United Nations, DESA, Population Division. Licensed under Creative Commons license CC BY 3.0 IGO.
United Nations, DESA, Population Division. *World Population Prospects 2022*. http://population.un.org/wpp/

Vertical agriculture, namely hydroponic systems, presents a way to relieve the pressure straining the global agricultural infrastructure. While conventional agriculture is limited by land area, hydroponic systems allow for crop cultivation in urban centers. Hydroponic systems also show up to 33% reduction in water consumption when compared to conventional agriculture and produce 52% lower CO2 emissions.[2]

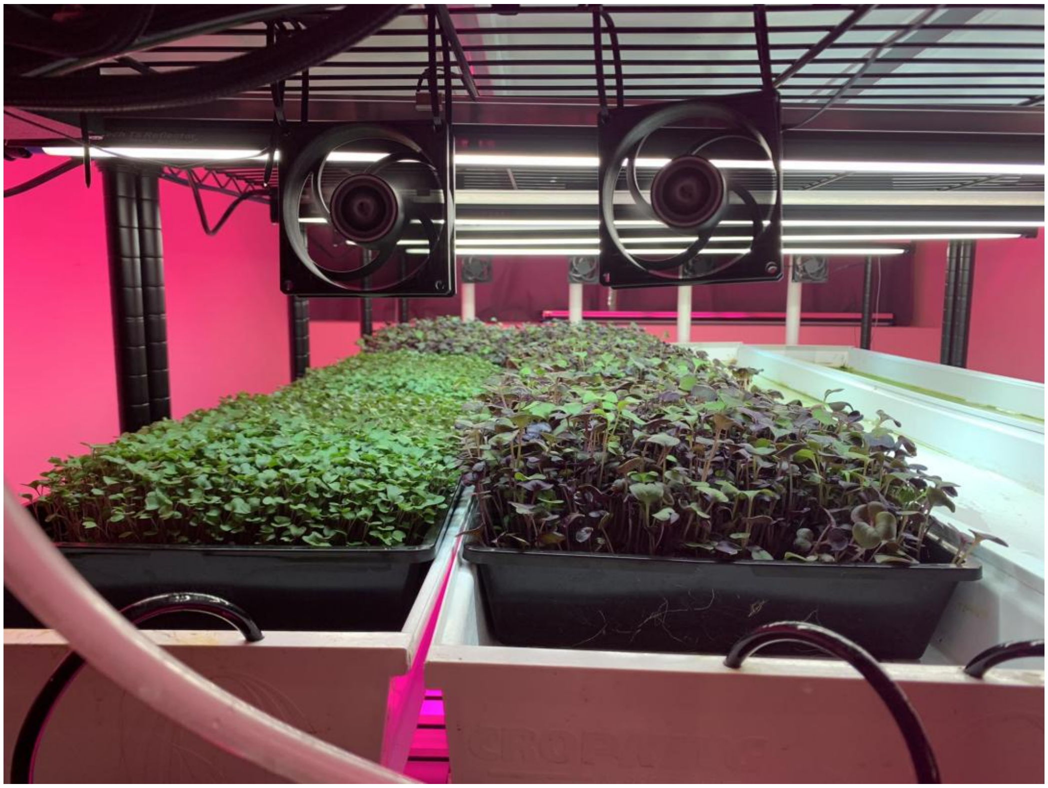
So what is a hydroponic system? Hydroponic agriculture is an agricultural design paradigm that uses no soil, relying instead on a solution of nutrients and additives, delivered to crops via a system of pipes and pumps. The hydroponic solution is carefully monitored to maintain optimal conditions for crop growth. One modern approach for hydroponic systems, like the system analyzed in this paper, involves using an Internet of Things (IoT) network of sensors to monitor solution conditions such as temperature, water volume, pH, and conductivity, and automatically administer additive doses as prescribed by a schedule or algorithm.

As mentioned above, this paper will be an analysis of one such system, which is designed to grow microgreens such as pea shoots and purple kohlrabi. While the aforementioned measurements allow for a respectable degree of control over the system, there is some information missing in our case. While conductivity is useful as a proxy measurement for estimating the nutrient levels in the system, there is currently no way to directly measure nutrient concentration, which is necessary information for optimizing growth yield and economic viability. Using several weeks of raw data collected during production, we aim to train a machine learning model to provide nutrient level estimates in real time.

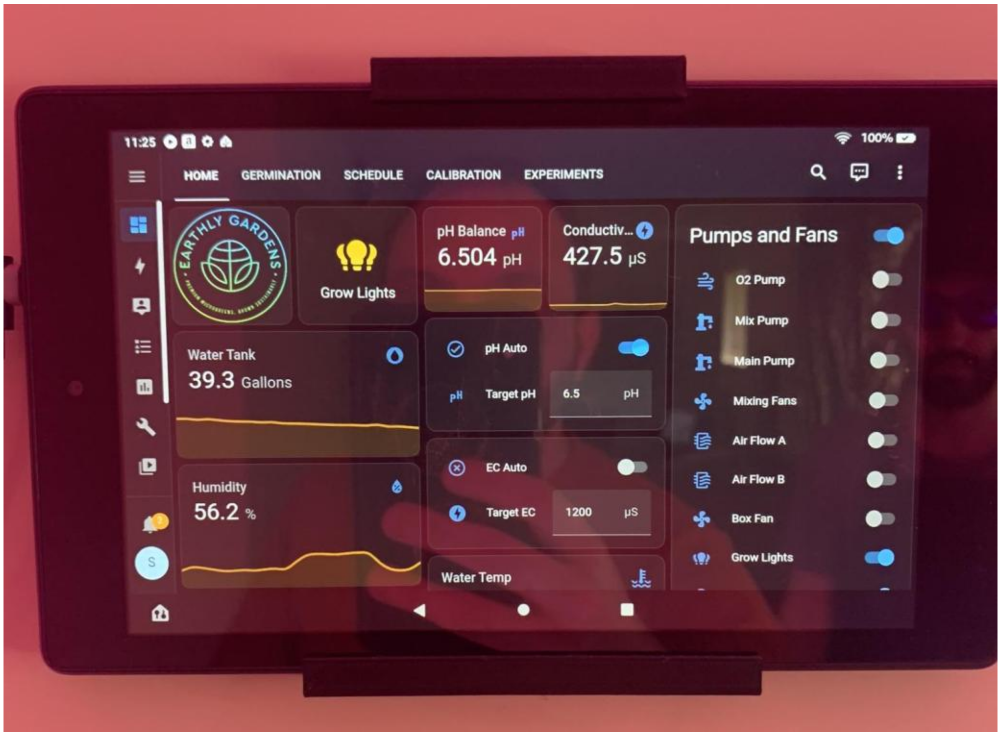
Some pictures of the farm from which the data is sourced can be found below and on the following page.



An automated hydroponic microgreen farm, with control panel (upper left), additive jars connected to peristaltic pumps (lower left), and 55 gallon tub with sensors for pH, Conductivity, temperature, and volume (off camera, lower left)



Pea shoots (left) and purple kohlrabi (right) ready for harvest



pH Balance pH 6.504 pH

Conductiv... 427.5 µS

Water Tank 39.3 Gallons

Humidity 56.2 %

pH Auto

Target pH 6.5 pH

EC Auto

Target EC 1200 µS

Water Temp

Pumps and Fans

O2 Pump

Mix Pump

Main Pump

Mixing Fans

Air Flow A

Air Flow B

Box Fan

Grow Lights

Close-up of control panel home screen

LITERATURE REVIEW

Pomoni, et. al., provides a review of the impact on energy and water consumption, land use, and environmental conditions of both hydroponic and conventional agricultural practices. They cite studies reporting that conventional agriculture occupies 38% of total land used globally, consumes 70% of the world's water resources, and accounts for 13.5-29% of total greenhouse gas emissions. In comparison, hydroponic agriculture reduces the required arable land by up to 25%, has up to 52.2% lower CO_2 emissions and shows a 33% reduction in water consumption, all while presenting 30-50% faster growth rates. [2]

Wortman reports a study comparing hydroponic systems with aquaponic systems, which utilizes a naturalistic design employing the use of fish to provide nutrients to crops in a closed system. Results showed that hydroponic systems gave crops with increased size at full maturity and increased marketable yield for all four of kale, basil, cherry tomato, and chipotle pepper crops. [3]

Kaewwiset, et. al., provides a study relating pH and Conductivity, giving a model for how the former affects changes in the latter. [4]

Akhter, et. al., utilizes IoT systems and machine learning to provide a predictive model to identify high risk conditions for development of the "apple scab" crop disease. [5]

DATA

Overview

To start, 8 weeks' worth of tabular data, collected non-continuously between the dates of 9/30/2023 and 12/10/2023, was provided. The data was acquired and minimally pre-processed from raw unstructured IoT data into tabular .xlsx files via a REST API that was built prior to the commencement of this analysis. The data was received from the farm's owner as 8 separate Excel files, containing over 2 million rows in total.

The data useful to our analysis fell, with one exception, into two major categories. The first category contains the measurements discussed above, those being pH, Electrical Conductivity (hereafter referred to as "EC" and provided in the unit of micro-Siemens, or μS), Water Volume (Gallons), and Water Temp (°C), as well as the environmental measurements of Humidity and Room Temp (°C). The second category is binary status columns, which refer to the various pumps and fans necessary to mix and circulate the hydroponic solution through the system to the crops and back again, and to keep the air around the crops at consistent temperature and humidity levels.

The final column we'll use is the Dosing Log, which is a record of dosing events for the system's pre-mixed additives, including pH Up, pH Down, and our target additive, Nutrient Mix. The Dosing Log column is an encoded string of the format "Dosing:X:YY:ZZZ", where X refers to the additive ID, YY is the volume of the dosage amount in mL, and ZZZ refers to the pump speed in mL/s.

Critically missing from our analysis is the target variable of Nutrient Mix Concentration. While future work may include sampling of the hydroponic solution throughout a growth cycle for chemical testing to provide data labels, that information is currently unavailable. As such, this analysis will use simulated data labels as model inputs. Generation of this simulated data will be discussed below.

Preprocessing

Typecasting

As mentioned above, the data was received as 8 files totalling over 2 million rows. Reading just the first file into Python resulted in an unwieldy dataframe using over 51MB of memory. This was, in part, due to the `object` dtype that most of the numerical variables were automatically cast to.

Inspection of these variables showed that the reason for this casting is the API's treatment of missing values, which were given as either `unknown` or `unavailable` - both of which were obviously interpreted as strings, causing whole columns to be misinterpreted as non-numerical. A simple reassignment to `None` using Pandas indexing syntax allowed recasting these columns to `float` or Pandas' built-in `category` type, as needed.

Recasting brought the memory usage for our first file down to an improved (but still not great) 29MB. Now that the data is appropriately interpreted, we can drive that number way down by downsampling.

Downsampling & Concatenating

As is the nature of IoT systems, we began with a very large dataset with altogether too many rows and much redundant data, as the multitude of components reporting data independently from one another populated a new row any time one component reported a new value while maintaining the previous value for all other components. While this results in an unnecessarily large dataset, it also brings the problem of incorrect weighting of the data; if one tried to train a model with the data in this state, some arbitrary value may be overrepresented as more components may report results in a given interval compared to other intervals, resulting in more rows for that arbitrary value than it merits.

To deal with this, we downsampled the dataframe to the consistent interval of 1 minute per row. Pandas provides a convenient `resample` dataframe method for this process. However, our different data types require different downsampling calculations, so `resample` had to be applied separately to each set of columns.

For the `float` columns, which contain sensor measurements, we simply took the mean for a given minute using the built-in `mean` method on the `resample` object, much in the same way as is done on `groupby` objects.

For the `category` columns, which contain a binary on/off value for the system's pumps and fans, a calculation of the mode (or most frequent) value made more sense as the value with the highest count in a given minute is likely the state in which a given component spent most of that minute. This calculation is not provided by Pandas, so we defined it ourselves and applied it via Pandas' `apply` dataframe method.

Finally came time to downsample the string column `Dosing Log`, still cast as `object` type. This column is sparse, populated with non-NA values only when the water bath is dosed with an additive. For context, the first file only had 71 non-NA values for this column in over 250k rows. Because doses are never less than 2 minutes apart, the downsampling for this column consisted of simply detecting a single non-NA value in a given minute and, if detected, returning that value. This was done in a similar fashion to the `category` column, with a custom function passed into the `apply` method.

All of this done, our single-file dataset of 250k rows was brought down to under 13k rows, down from 51MB to 3MB. With this pipeline in place, we processed the other 7 files and combined the data into a single dataframe. Since the multiple data files contained some overlapping data from inconsistent API requests, we also dropped the rows with duplicate indices at this time.

Parsing the Dosing Log variable

Next came time to parse the `Dosing Log` string variable. As described above, this column was encoded in the format "Dosing:X:YY:ZZZ", where X refers to the additive ID, YY is the volume of the dosage amount in mL, and ZZZ refers to the pump speed in mL/s. The pump speed is not of great interest to us, so we drop it here.

The first step in the parsing process is to apply the `split` method on the column using the str accessor, which allows access to many of Python's standard string methods in relation to Pandas `Series` objects. We called the `split` method, with `separator = ":"` to identify that the values are delimited by the colon character. Normally, this would return a series of lists, but the Pandas implementation of the `str.split` method allows the additional argument of `expand = True`, which adjusts the method to return the delimited values as separate columns in a dataframe. Since the first value returned will simply be the word "Dosing" for every value, we cut off this first column and maintained the other 3, saving it as `df_dosing` and naming the columns `Additive` and `Volume added (mL)` accordingly.

Next, we defined the `additive_mapper` dictionary to act as a mapper for the additive IDs so we could interpret the "X" of the dosing log, which was given as one of 0, 1, or 2, with 0 referring to the pH Up additive, 1 being pH Down, and 2 being Nutrient Mix. The system also allows for a value of 3 for Peroxide, and though there were no Peroxide doses in our data, this code will likely be used for further analysis of this system outside the scope of this project, so that was included in the mapper as well. We overwrite the `Additive` column of the `df_dosing` dataframe we produced earlier using the `map` dataframe method, passing in the `additive_mapper` dictionary to interpret the additive IDs.

We then reshaped the data to be a bit more convenient to work with in a model. To do this, we applied the `pivot` method, setting `columns = 'Additive'` and `values = 'Volume added (mL)'`, which re-imagines the information in those columns such that each additive is its own column, with the values being the volume in mL for each dosing event. Finally `df_dosing` was left-joined to our main dataframe.

Feature engineering

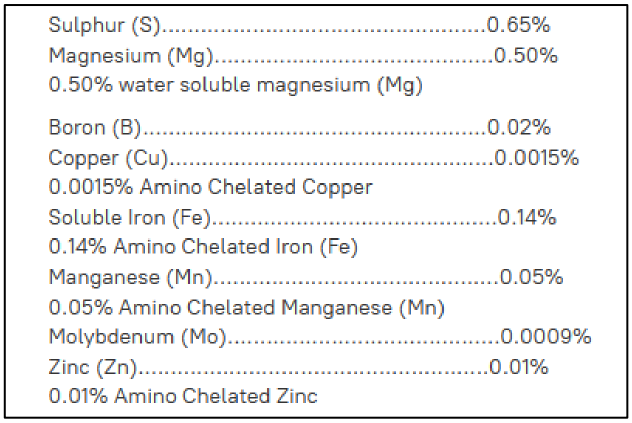
It's a common sense assumption that the amount of nutrients in the water will steadily decrease over time as the plants consume nutrients to fuel their growth. To account for this, we will add a predictor that counts the minutes since the most recent dose of Nutrient Mix. We first looked for an array operation solution to accomplish this task quickly, but a simple for-loop turned out to be the more convenient option, and ran without too much delay.

First, the column `minutes_since_last_dose_NM` was defined as an NA column in our main dataframe. A numerical variable called `count_since_last_additive` was also defined, to be incremented inside the for-loop. Finally, the loop was defined. For each row, if the `Nutrient Mix` column generated by the `pivot` operation above equaled a value other than 0, then `count_since_last_additive` was set to 0. Otherwise, `count_since_last_additive` was incremented by 1. Finally, the row's value for the `minutes_since_last_dose_NM` column was set equal to the current value of `count_since_last_additive`. This is a viable algorithm since our data is indexed at an interval of 1-minute-per-row; if the interval were to change at a later date – say, to be aggregated in 5-minute intervals instead – either the increment value will have to be changed, or the interpretation of the column will have to be adjusted accordingly.

One other minor calculated column was added. Since the dosing amounts were given in mL, it didn't make sense to keep the `Water Tank (G)` variable, which measures the volume of the system’s water, in Gallons. So the new column `Water Tank (L)` was added using the simple calculation of `df[“Water Tank (G)”] * 3.78541`, the conversion ratio of Gallons to Liters.

Generating the target data

As mentioned in the Overview subsection above, a critical piece of missing information for this analysis is the data labels – the true value of Nutrient Mix concentration on which to train models. While there are tests available to measure the chemical content of nutrient solutions, such tests can be expensive and time-consuming. While this process may be included in future work, for this analysis we will instead generate simulated target data for use in model training.



Sulphur (S)..0.65%
Magnesium (Mg)..0.50%
0.50% water soluble magnesium (Mg)

Boron (B)..0.02%
Copper (Cu)...0.0015%
0.0015% Amino Chelated Copper
Soluble Iron (Fe)..0.14%
0.14% Amino Chelated Iron (Fe)
Manganese (Mn)..0.05%
0.05% Amino Chelated Manganese (Mn)
Molybdenum (Mo)..0.0009%
Zinc (Zn)...0.01%
0.01% Amino Chelated Zinc

Before getting into how this simulated data was generated, a quick note about its unit. Because of the complex composition of the Nutrient Mix (see image to the right), the decision was made to abstract the concentration, treating it instead as a proportion of the dosing mass and, in turn, its concentration prior to its addition to the water bath.

To illustrate – a full dose consists of stock nutrient mix diluted 1:1 totaling 15mL. Both the stock mix and dose (after dilution) will be considered to have a **mass** of 1.000 dose. After adding the dose to the main water bath, the mass of Nutrient Mix present in the system will increase by 1.000. In this example, the **concentration** of the diluted dose is 1.000 and the stock mix is 2.000, since it has twice the concentration of the diluted dose. The concentration in the main water bath is the mass of nutrients divided by the water bath’s volume.

Now to explain how the target data was generated. A few assumptions were made; first, that the rate of Nutrient Mix consumption by the plants was equal to a full dose per day, or .000695 per minute. Second, that the mass of Nutrient Mix in the system fluctuates roughly between 0.500 and 1.500. With that said, the generation algorithm follows Equations 1 and 2, below:

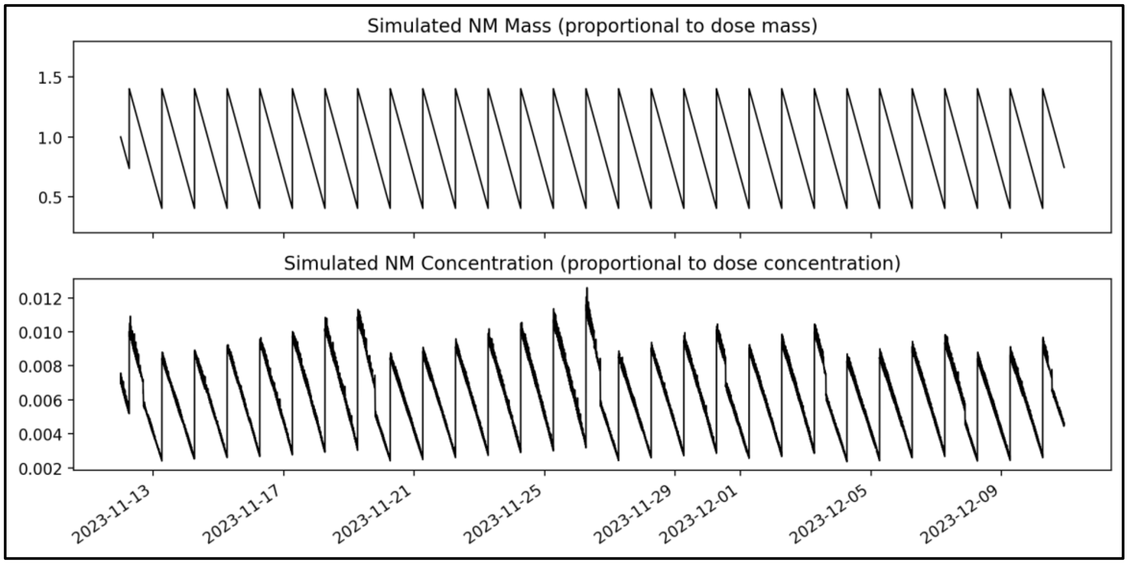
$$m_{NM,i} = m_{NM,i-1} - 0.000695 + D_{NM,i} \tag{1}$$

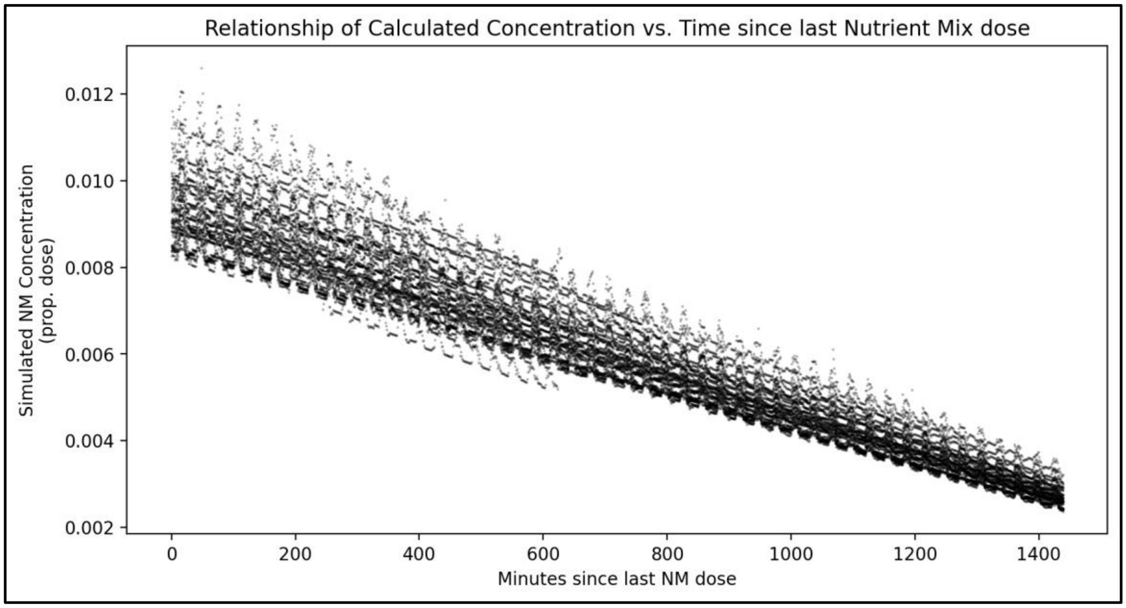
$$C_{NM,i} = \frac{m_{NM,i}}{V_i} \tag{2}$$

Where $m_{NM,i}$ is the mass of Nutrient Mix for a given row, $m_{NM,i-1}$ is the mass the row before, $D_{NM,i}$ is the Nutrient Mix dose amount for a given row (1 if a dose was added, 0 if not), $C_{NM,i}$ is the concentration of Nutrient Mix for a given row, and V is the volume of the water tank in Liters.

Equation 1 was calculated similarly as the `minutes_since_last_dose_NM` variable discussed above, using a for-loop increment row-by-row and saved as a Series named `target_mass`. The Equation 2 calculation was performed as an array operation between the `target_mass` series and the `Water Tank (L)` dataframe column.

The results of this calculation can be seen in the plots below. The first image shows both the `target_mass` and the `target_concentration` columns plotted over time. The second image shows the `target_concentration` column plotted against the `minutes_since_last_dose_NM` column, showing a strong inverse relationship and indicating that this timing column will be an important predictor when the time comes for modeling.

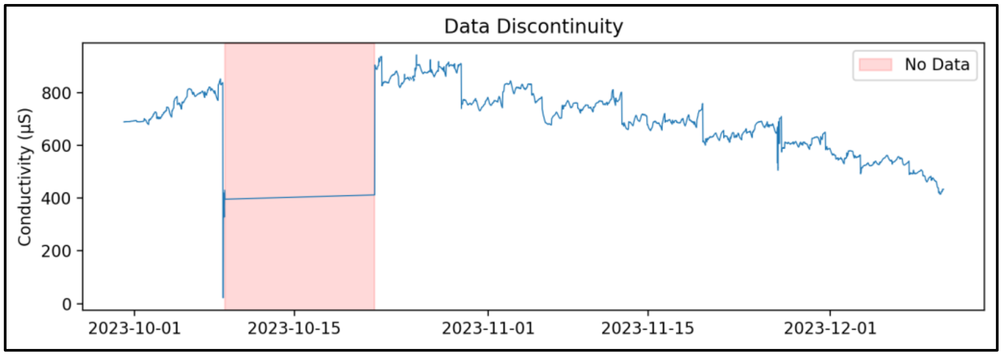




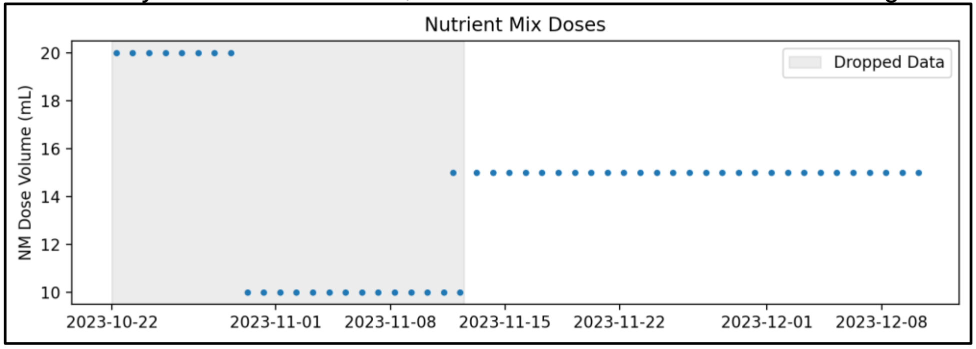
METHODOLOGY

Exploration

Early in the data preparation process it became apparent that the data provided was not continuous. The graph below of Conductivity (EC) over the course of the data collection window shows this discontinuity starkly; further inspection shows that the first data file provided was collected several weeks before the second file. In order to keep time-series calculations streamlined, the decision was made to drop the first file from the analysis, with the truncated dataset now beginning on 10/22.



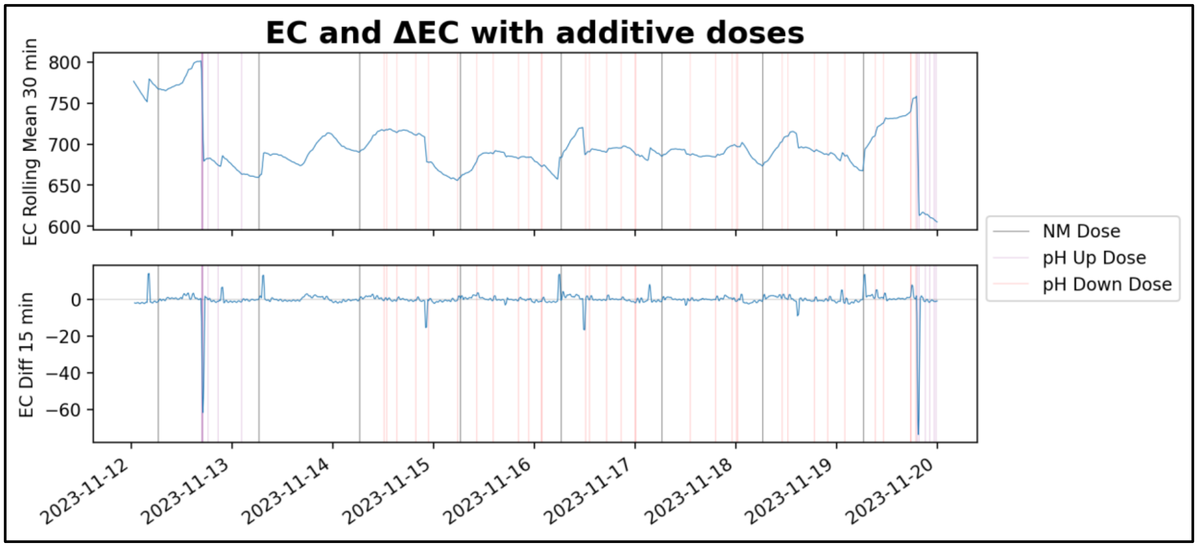
The window was further narrowed after an inspection of the Nutrient Mix doses revealed that the dose volume varied throughout the data window, with some dosing events adding 20mL of Nutrient Mix, some adding 10mL, eventually settling on a standard value of 15mL. While the effect of varying dose volumes would no doubt provide useful information, this consideration is outside the scope of this analysis. For this reason, the data was further truncated to begin on 11/12.



The graph above of the Electrical Conductivity also highlights some major trending in the remaining data. To account for this, we can take the first-difference of this column. The visualization below shows one week's worth of data - the top half shows the direct reading from the EC sensor, and the bottom half shows the difference with a window of 15 minutes (in other words, the change in EC from 15 minutes before). Also seen are some vertical line annotations, marking the dosing for different additives denoted by color (gray for Nutrient Mix, purple for pH Up, red for pH Down).

This visualization allows us to identify the effect of these different doses on EC, with the idea being that the dissolved ions in the Nutrient Mix will have a consistent and noticeable effect on the Conductivity, which would allow us to quantify a dose of Nutrient Mix in terms of EC.

Unfortunately, this did not pan out, as the graph below shows several points where the EC is virtually unaffected by a dose of nutrients.



Modeling

In the pursuit of optimizing hydroponic system management to facilitate superior plant growth outcomes, the strategic utilization of predictive modeling assumes paramount importance. Here, we embark on a meticulous examination of three distinct predictive models i.e., “`Linear Regression`”, “`Gradient Boosting`”, and “`Artificial Neural Networks (ANN)`” each meticulously evaluated for its proficiency in forecasting the nutrient concentration within hydroponic environments. This segment serves as a cornerstone of our project report, illuminating the methodological rigor and the technical intricacies that underpin our predictive modeling endeavors.

These models play a pivotal role in determining which one exhibits a superior predictive performance concerning our target variable, the concentration of nutrients within the hydroponic environment. Our data encompasses a diverse array of metrics, including electrical conductivity (EC), pH level, humidity, room temperature, water levels, and temperatures, alongside the operational statuses of various system components. Among these metrics, the *`TARGET - NM Concentration (%)`* emerges as the primary focus of our predictive modeling efforts.

<u>Model Selection:</u>

1. Linear Regression:
Linear Regression, is a classic statistical method, it was chosen as a fundamental model to establish a baseline prediction of our nutrient concentration. Its simplicity, interpretability, and computational efficiency makes it an ideal for the initial exploration. It assumes a linear relationship between the input features and its target variable, making it well suited for capturing direct and proportional effects of environmental parameters on nutrient concentration.

2. Gradient Boosting Model:
Gradient Boosting, being an ensemble method, was selected for its remarkable predictive power and its ability to handle complex, nonlinear relationships inherent in hydroponic systems. It leverages decision trees as weak learners, constructing an ensemble model that sequentially corrects errors made by preceding trees. This iterative process enables the model to capture the intricate patterns and interactions within the data, therefore enhancing the predictive accuracy.

3. Artificial Neural Networks (ANN):
Lastly, we chose ANN, inspired by its biological neural networks of the human brain, it was included for its capacity to discern intricate patterns and nonlinear relationships from vast datasets. As a deep learning architecture, ANN excels in not only capturing complex, hierarchical representations, but also potentially uncovering nuanced predictors of nutrient concentration. ANN comprises interconnected layers of artificial neurons, each performing weighted transformations on the data and passing the results to subsequent layers. Through backpropagation and gradient descent, ANN adjusts the weights of these transformations iteratively to minimize the prediction error.

<u>Model Evaluation:</u>

The evaluation phase of our predictive modeling endeavors played a pivotal role in discerning the efficacy of each model in accurately forecasting the nutrient concentration within hydroponic systems. Lets dive into a detailed exploration of the model evaluation process, shedding light on the comprehensive metrics utilized and the insights gleaned from the assessment.

1. Linear Regression:
Linear Regression, being a foundational statistical technique, served as the initial model for predicting nutrient concentration. The evaluation commenced with training the model on a subset of the "`model data`" and subsequently testing its predictive capabilities on the unseen data. The Root Mean Square Error (RMSE) emerged as the primary metric for evaluating the model's performance. With an achieved `RMSE` of approximately `0.0002`, Linear Regression exhibited a commendable predictive accuracy, providing us a baseline benchmark for comparison with more sophisticated models.

2. Gradient Boosting Model:
The Gradient Boosting model, characterized by its ensemble nature and iterative error correction mechanism, underwent rigorous evaluation to assess its predictive prowess. Leveraging decision trees as weak learners, the model sequentially corrected errors made by the preceding trees, and helped in capturing intricate patterns and nonlinear relationships within the data. Upon evaluation, the Gradient Boosting model demonstrated a significantly lower `RMSE` of approximately `0.0001` compared to Linear Regression. This marks improvement by underscoring the model's superior ability to discern complex data patterns, making it a compelling choice for the accurate nutrient concentration prediction.

3. Artificial Neural Networks (ANN):
Coming towards Artificial Neural Networks, it being inspired by the biological neural networks of the human brain, presented an intriguing avenue for predicting nutrient concentration in hydroponic systems. Despite its computational complexity and potential for discerning nonlinear relationships, the ANN model exhibited a higher `RMSE` of approximately `0.2304` compared to Gradient Boosting. This outcome suggested challenges associated with effectively training deep neural networks and highlighted the areas for further optimization.

The evaluation results emphasized on the superior predictive performance of the "**`Gradient Boosting`**" model compared to Linear Regression and Artificial Neural Networks. The ensemble nature and iterative learning approach of Gradient Boosting enabled it to effectively capture nuanced data patterns, leading to more precise predictions of nutrient concentration. The evaluation process provided us valuable insights into the strengths and limitations of each model, guiding towards subsequent steps in model refinement and optimization.

Hyperparameter Tuning:

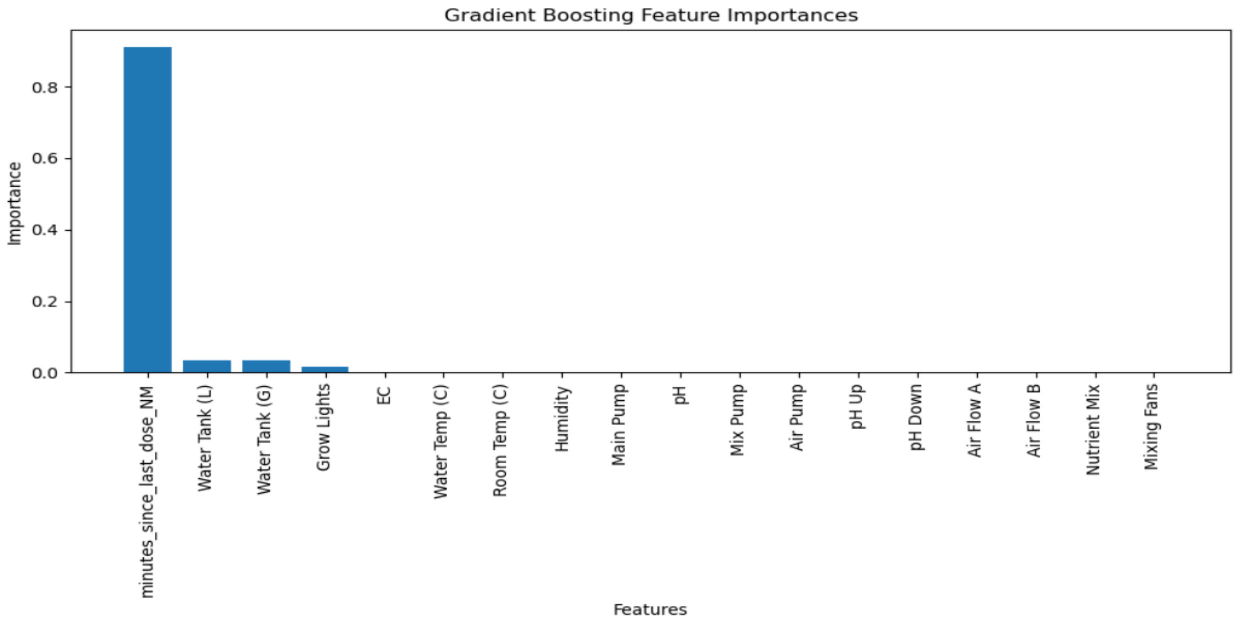
To further improve the model performance, we conducted hyperparameter tuning specifically for the Gradient Boosting model. This iterative process involved systematically exploring different combinations of hyperparameters, such as the `number of estimators, maximum depth of trees,` and `learning rate`. The objective was to identify the optimal parameter configuration that minimizes the prediction error and maximizes predictive accuracy.

Following the comprehensive parameter tuning, the best parameter configuration `{'learning_rate': 0.1, 'max_depth': 5, 'n_estimators': 100}` was selected based on its ability to *minimize* the `RMSE` to `0.00010083929808762843`. This refined configuration underscored the efficacy of Gradient Boosting as the preferred model for predicting nutrient concentration in hydroponic systems, further solidifying its position as the optimal choice for conducting informed decision making in hydroponic system management.

Model Interpretation:

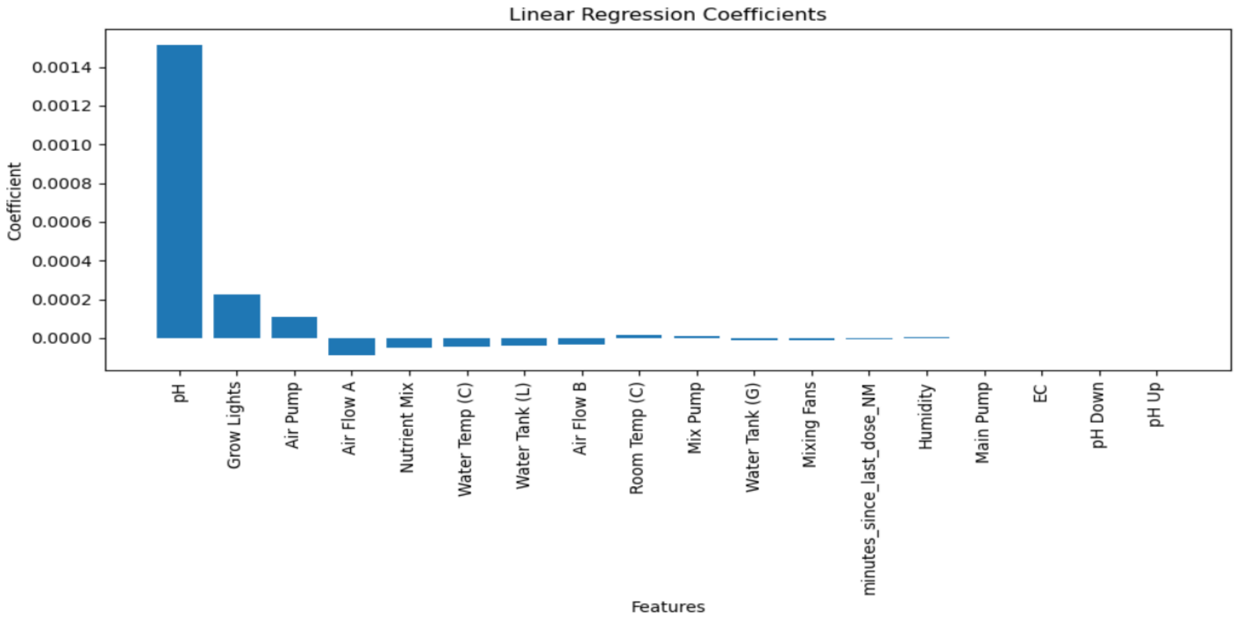
1. Gradient Boosting:

The feature importance plot generated for the Gradient Boosting model unveils a rich tapestry of insights into the intricate interplay between environmental parameters and nutrient concentration dynamics. Notably, the prominence of the "`minutes_since_last_dose_NM`" feature underscores the temporal dimension of nutrient uptake, emphasizing the criticality of the timely nutrient supplementation in maintaining the optimal nutrient levels for plant growth. This underscores the model's ability to capture nuanced temporal dependencies and its capacity to adapt to the evolving hydroponic conditions.



2. Linear Regression:

Contrastingly, the coefficient plot derived from the Linear Regression model provides a linear perspective on feature importance, revealing the direct and proportional relationships between input features and nutrient concentration. Here, the salience of the "`pH`" feature underscores its role in modulating the nutrient absorption rates, highlighting the importance of pH regulation for optimizing nutrient availability and fostering robust plant growth. This underscores the model's interpretability and its efficacy in discerning linear relationships within the model data.



Model Comparison:

An exhaustive comparison of model performance, encompassing the metrics such as `R-squared value, Mean Absolute Error (MAE),` and `Adjusted R-squared` offers nuanced insights into the efficacy of each model:

"Gradient Boosting" emerges as the front runner, boasting the *highest R-squared value* and the *lowest MAE* among the three models. Its ensemble learning framework and iterative error correction mechanisms empower it to capture the complex data patterns and its nonlinear relationships, rendering it to adapt at the forecasting nutrient concentration with the unparalleled precision and accuracy.

"Linear Regression", while commendable in its own right, *lags* slightly behind Gradient Boosting in terms of predictive performance. Nonetheless, its simplicity and interpretability make it a pragmatic choice for the simpler predictive tasks, offering a clear and transparent framework for analyzing linear relationships within the data.

In contrast to them, the "Artificial Neural Network" model exhibits significant shortcomings, characterized by an abysmally *low R-squared value and a disproportionately high MAE*. This underscores the inherent challenges associated with training deep neural networks for hydroponic system modeling, including the issues of over fitting, data scarcity, and the parameter sensitivity.

The "**Gradient Boosting**" emerges as the undisputed champion, offering unparalleled predictive accuracy and robust performance for forecasting nutrient concentration in hydroponic systems. Gradient Boosting paves the way for informed decision making and optimized hydroponic system management, driving towards maximal plant growth and yield.

Model Refinement and Performance Evaluation:

We then went ahead with “Round 2” for modeling and evaluation. In this iterative phase, our focus shifted towards enhancing the predictive prowess of our models through meticulous feature selection and refinement. By isolating the most influential features, we aimed to streamline the modeling process and boost predictive accuracy.

Feature Selection and Model Refinement:
Drawing upon the Gradient Boosting model’s feature importances, we established a significance *threshold of 0.05*. Features surpassing this threshold were deemed sufficiently impactful, to be included in our refined models. This strategic selection process allowed us to distill the data to its most salient components, by minimizing noise and maximizing the predictive power. With the subset of important features identified, we proceeded to retrain both the Gradient Boosting and Linear Regression models using only these selected variables. This refinement step ensured that our models were optimized to focus solely on the key drivers of nutrient concentration in hydroponic systems.

After the refinement, a comprehensive evaluation of the refined models was conducted using established performance metrics:

```
Gradient Boosting Model:
- RMSE: 0.000482
- R-squared: 0.935
- MAE: 0.000388

Linear Regression Model:
- RMSE: 0.000491
- R-squared: 0.932
- MAE: 0.000393
```

The refined Gradient Boosting model continued to exhibit superior predictive performance compared to its Linear Regression counterpart, albeit by a slight margin. With a higher R-squared value and lower RMSE and MAE, the Gradient Boosting model demonstrated its robustness in accurately forecasting nutrient concentration in hydroponic environments, even after the feature selection and refinement process. While the Linear Regression model showcased a respectable performance, its predictive accuracy fell slightly short of the Gradient Boosting model.

The “**Gradient Boosting**” model, in particular, has emerged as a formidable tool for hydroponic system management, underscoring the significance of meticulous optimization strategies in achieving optimal outcomes in plant cultivation.

Recurrent Neural Network Implementation:

The deployment of Recurrent Neural Networks (RNN) for time series data analysis holds immense promise in capturing the temporal dependencies and extracting important patterns for predicting the model. In our hydroponic system management project, we harnessed the power of RNNs to “forecast nutrient concentration dynamics with high precision”.

Model Architecture and Training:
Our RNN architecture comprises a single *SimpleRNN layer with 64 units*, it is tailored to accommodate the sequential nature of our time series data. The input shape was configured to

match the dimensions of the training data, ensuring the compatibility with the temporal structure. The model was then compiled using the Adam optimizer and mean squared error (MSE) loss function. Training commenced with "*10 epochs and a batch size of 32*", with a validation split of 0.1 for monitoring model performance during training. Through iterative optimization, the model learned to discern intricate temporal patterns within the hydroponic system data, gradually refining its predictive capabilities.

Performance Evaluation:
Upon the completion of training, the model underwent rigorous evaluation to gauge its predictive prowess. The test Mean Squared Error (MSE) was computed to quantify the disparity between predicted and actual nutrient concentration values. For the RNN model, the test MSE was approximately "7.92e-05", indicating a commendable predictive performance. This metric underscores the model's ability to generalize well to unseen data and make accurate forecasts, thereby enhancing the efficacy of hydroponic system management.

The successful implementation of RNN for nutrient concentration prediction signifies a significant advancement in our predictive modeling endeavors. By leveraging the temporal dynamics inherent in time series data, the RNN model demonstrates its potential to revolutionize hydroponic system management, paving the way for more efficient and sustainable agricultural practices

Conclusion:

In conclusion, our endeavor to optimize hydroponic system management through predictive modeling has yielded significant advancements in enhancing plant growth outcomes and fostering sustainability in agriculture. Through the meticulous examination and evaluation of the three distinct predictive models "Linear Regression", "Gradient Boosting", and "Artificial Neural Networks", we have not only illuminated the methodological rigor underlying our modeling approach but also discerned the most effective techniques for forecasting the nutrient concentration dynamics within hydroponic environments.

The comprehensive evaluation of our models revealed "**Gradient Boosting**" as the standout performer, showcasing a superior predictive accuracy and robustness in capturing complex data patterns. With its ensemble learning framework and iterative error correction mechanisms, Gradient Boosting emerged as the optimal choice for informed decision making in hydroponic systems, offering us unparalleled precision and accuracy in forecasting the nutrient concentrations. Furthermore, the refinement of our models through feature selection and optimization strategies further bolstered their predictive prowess, in culminating the enhanced performance metrics and streamlined modeling processes. The deployment of Recurrent Neural Networks for time series data analysis represented a significant milestone, enabling us to capture the temporal dependencies and extraction of crucial patterns for precise nutrient concentration forecasting.

Through these advancements, our predictive models not only facilitate dynamic adjustment of nutrient dosing for optimal plant growth but also offer valuable insights into nutrient uptake dynamics, thereby contributing to the evolution of autonomous farming technology. Moving forward, the implementation of our solutions holds the potential to revolutionize sustainable agricultural practices in indoor farming environments, fostering efficiency, productivity, and ultimately, a more sustainable future for food production.

CONTRIBUTIONS

Alex Rosenblum is responsible for the Abstract, Introduction, Literature Review, Data, and Future Work sections as well as the Exploration subsection of Methodology. Harnain Kaur Sardarni is responsible for the Modeling and Conclusion subsections of Methodology.

FUTURE WORK

The obvious next step for this analysis is to find some way to collect new data labeled with ground truth values that can be used to train the models we've developed. The work that we've done here to build preprocessing and modeling pipelines will streamline modeling when labeled data is acquired. Methods for labeling this data are currently being explored.

It's also possible that the best solution to this particular problem lies outside of machine learning. An alternative avenue of exploration will be to build a hand-written estimator that is more robust than a simple daily schedule, but requires less set-up than a machine learning model.

Finally, the last piece for this work will be integrating the final working model into the system such that it can facilitate decision-making automatically, administering Nutrient Mix doses based on model output.

REFERENCES

[1] https://population.un.org/wpp/Graphs/Probabilistic/POP/TOT/900

[2] Pomoni DI, Koukou MK, Vrachopoulos MG, Vasiliadis L. A Review of Hydroponics and Conventional Agriculture Based on Energy and Water Consumption, Environmental Impact, and Land Use. *Energies*. 2023; 16(4):1690. https://doi.org/10.3390/en16041690

[3] Sam E. Wortman, Crop physiological response to nutrient solution electrical conductivity and pH in an ebb-and-flow hydroponic system, Scientia Horticulturae, Volume 194, 2015, Pages 34-42, ISSN 0304-4238, https://doi.org/10.1016/j.scienta.2015.07.045

[4] T. Kaewwiset and T. Yooyativong, "Estimation of electrical conductivity and pH in hydroponic nutrient mixing system using Linear Regression algorithm," *2017 International Conference on Digital Arts, Media and Technology (ICDAMT)*, Chiang Mai, Thailand, 2017, pp. 1-5, doi: 10.1109/ICDAMT.2017.7904922

[5] Ravesa Akhter, Shabir Ahmad Sofi, Precision agriculture using IoT data analytics and machine learning, Journal of King Saud University - Computer and Information Sciences, Volume 34, Issue 8, Part B, 2022, Pages 5602-5618, ISSN 1319-1578, https://doi.org/10.1016/j.jksuci.2021.05.013