UNIVERSITY OF SCIENCE AND TECHNOLOGY OF HANOI

MASTER'S DEGREE

Research and Development

MASTER THESIS

by

PHAM Phan Bach

M21.ICT.002

Information and Communication Technology

Title:

Regression methods for analysis of rice spectral reflectance data

Supervisors: Dr. TRAN Giang Son

Lab name: USTH's ICT lab

Hanoi. Oct 2024

Attestation

I hereby, PHAM Phan Bach, certify that my thesis doesn't contain plagiarism (copy/paste) from other sources.

In case of plagiarism in my thesis, I know the consequences and I understand that my thesis won't be evaluated. In this case, my M2 internship will be noted as "fail".

Oct, 23th, 2023
Signature

PHAM Phan Bach

Acknowledgements

Contents

List of Figures

List of Tables

Abstract

Chapter 1

Introduction

1.1 Context & Motivation

Rice is well known as an indispensable ingredient providing daily nourishment for more than half of the world's population. With the upward trends of the world's population projections, it has become more and more essential to not only maintain but also increase the rice yield efficiency. The health of rice production has been proven to be linked directly to the amount of nutrients in the plants, especially Chlorophyll (Chlo), Phosphorus(P), and Potassium (K). A lower level of concentration means a diminished yield. A sufficient amount of Nitrogen will improve plant size, spikelet, panicle health, and yields. Phosphorus affects the rice's flower blooming cycle, helps root growth, and plants immunity against diseases and drought. Potassium increases rice's culm strength and prevents plants from topping on each other. However, fertilizing crops excessively can lead to reducing the branch strength and attracting unwanted vermin, fungi, and other foreign organisms. The appearance of leaves, including their color, shape, and sheath, provides valuable information about a plant's nutrient and health status, which is closely tied to its nutritional content. The measurement of the chemical components present in a plant can be determined through its leaves and canopy's spectral reflectance signature.

While analyzing spectral reflectance gives the most accurate result, this method can only be done on a small scale due to its high-cost, time-consuming, labor-intensive process. The fertilization also needs to be carried out in a timely manner during the plant growth stage. Remote sensing is one of the more practical approaches for monitoring hundreds of hectares of rice fields. It can be swiftly deployed over larger areas with minimum impact on the crops. The plant's nutrition content can be analyzed through hyper-spectral imaging techniques. Drone armed with multi-spectral cameras with very high spatial resolution are flown over the fields

and gather images in hundreds of different bandwidths. Scientists from many fields have been using these hyperspectral images to analyze many chemical components in crop leaves. In spite of their usefulness, the task of processing the data and accuracy of crops determined nutrition values leaves a lot to be desired.

One of the more popular method is ultilising normalized difference vegetation index also known as NDVI. However, since it constists of 2 specified bandwidth red and infrared, there a thounsands of combination pairs. Therefore it is quite a challenges to maximize the method effectiveness.

1.2 Objectives

The objective of this project is first to develop multiple model capable of identifying the concentrations of Phosphorus (P), potassium (K), and Chlorophyll-a based on reflectance data collected from replicates and their sub-replicates. Afterwards, these regression technique is extensively examined and analysed to find out correspondingly the most appropriate pairs of bandwidth . In summary, we aim to comprehensively assess the accuracy and precision of our models, providing valuable insights into their performance and suitability for its practical applications.

1.3 Related Work

During the time working on this study, multiple documents with similar topics and challenges was used as reference documents. In the research "Spectral Reflectance Characteristics and Chlorophyll Content Estimation Model of Quercus aquifolioides Leaves at Different Altitudes in Seijila Mountain" Zhu, J et al, by using a dataset of 60 samples of spectral parameters explored regression models to predict Chlorophyll content. Their analysis contains 9 different spectral characteristics of plants as predictor variables, with Chlorophyll content serving as the response variable. They applied univariate regression techniques including index, linear, and quadratic polynomial models, to control accurate estimation models. As for the result, the R-squared for the RGP model seems to be the greatest one with remarkable values of 0.8523. Another work that was used as reference is the study of "Predicting Nitrogen Content in Rice Leaves Using Multi-Spectral Images with a Hybrid Radial Basis Function Neutral Network and Partial Least-Squares Regression" is the combination of Multi-Spectral Imagse, a Hybrid Radial Basis Function Neural Network, and Partial Least-Squares Regression. The evaluation metrics are MAE, MAPE, and RMSE (similar to MSE but different when RMSE is the square root of MSE). The most remarkable performance goes to the RBFNN model which is outstanding in both the growing and mature stages. During the growing stage, it achieves a MAPE

of 0.5399, while with the mature stage, it records a MAPE of 0.1566. These results when compared with GRL seem to be surpassed when this model has the MAPE of 1.0545 with the growing stage and 0.7399 with the mature stage. Which also performs well is the GRM with 1.2395 of MAPE in the growing stage and 1.2272 in the mature stage.

1.4 Thesis organization

The thesis is structured as follows:

- Section 1: Introduction & Objectives
- Section 2: Materials and methods
- Section 3: Result
- Section 4: Conclusion & Future works

Chapter 2

Material & Methodology

2.1 material

The experiements is conducted in Lam Thao District, Phu Tho province, Vietnam. This part is favorable for its location within the Red River delta, under the influence of tropical weather and sustain 1720 mm of annual rainfall. It have been ultilised by local residents for cultivating rice for years. Our sample rice field is sepperated into 54 part of 100m2 square parcels. With the help of locals farmer, the parcels are fertilized into differents amounts of N, P, K which will cause variation in leaf's spectral reflectance. The farmers was also request for their expertises to take care of the rice field during its growth stages to minimize the crops harms because of water stress or diseases. In each divided parcels, the data is extracted 3 times using specialised machine for measuring spectral reflectance.

The data is captured into a folder with two types of files: a set of sed files and csv file. The sed files are taken from the specialised machine while contains a great deal of information, most of them are unnecessary for the project. Each row in the csv file is correspond to a sed file. For the information to be usable, both type of files need to be combined to a comprehensive and informative dataset

Structure of .sed files Number of .sed files: 260 files.

- Content in a .sed file:
- Version: 2.3 [1.2.6250C]
- File Name
- Instrument: PSR-2500-SN1726293 [2]

- Detectors: 512,0,256
- Measurement: REFLECTANCE
- Date
- Time
- Temperature (c)
- Battery Voltage
- Averages: 10,10
- Integration: 10,30,20,30
- Dark Mode: AUTO, AUTO
- Foreotopic: LENS RADIANCE, LENS4 RADIANCE
- Radiometric Calibration: RADIANCE
- Units: W/m2 /sr/nm
- Wavelength Range: 350,2500
- Latitude
- Longtitude
- Altitude
- GPS Time
- Satellites
- Calibrated Reference Correction File: none
- Channels: 2151
- Columns [4]

Each file also contain 2151 data, corresponding to wavelength channel captured from 350-2500

- Wavelength (Wvl)
- Reference Radian (Rad. (Ref.))

- Target Radian (Rad. (Target.))
- Reflect (Reflect.)

Structure of csv files There are 61 replicates, with 171 data row

- Latitude
- Longitude
- Replicate, Sub-replicate
- Concentration of P, K (mg/kg)
- Chlorophyll-a

Nutrients	Sample	Mean	Min	Max	St. Deviation
Chlorophyll	171	41.84	33.0	48.6	3.10
P concentration	171	4317.11	1124.0	7740.0	805.68
K concentration	171	35975.34	12620.0	59730.0	10578.94

Table 2.1: Table of Nutrients' statistics description

2.2 Methodology

In this section, several necessary knowledge is discussed before the reader could understand deeply on our work problem and solution.

2.2.1 Spectral Reflectance

The reflectance is the effectiveness in reflecting radio energy of a surface, and different radio wavelength has different reflectance on the same surface. By combining multiple wavelength reflectance, we gain a spectral reflectance of the targeted surface.

It is a highlight that different material provides different reflectance spectrum basing on the material physical characteristics, chemical conditions, or environmental surroundings. This interesting fact plays a key role in the material recognition through their spectral reflectance. The hypothesis is that by collecting enough sample reflectance of a surface, it is possible to analysis a list of materials existed there. Even more advanced, we could approximately guess about the quantitative of those materials.

Actually, this hypothesis is quite robustness through a bunch of its application in the astronomy

domains. For instances, the NASA scientists uses the spectral reflectance to analyze materials existed in a planet, star, or even a galaxy. Therefore, it is a potential question about its application in the smart agriculture also.

More detail, in this work, we try to examine that question by an afford to find out the relationship between the spectral reflectance of the rice leave with the concentration level of three important chemicals: phosphrus, potassium, and chlorophyll. Our final target is a quality prediction of these chemicals based on their reflection from the sunlight which could be easy captured through UAV devices. However, because of accuracy, in this work context, we use a high quality device to collect the leave spectral reflectance. In other words, we manually go to each rice, then capture their leave reflection by an expensive hand device. This interesting collective method is discussed in other part of report and be illustrated to our accuracy of their relationship found if having any.

2.2.2 Regression Analysis

Regression analysis is a statistical technique for estimating the relevancy between a dependent variable and one or multiple base variables. The technique gain this relevancy by applying following steps: Firstly, researchers a model that is believed to match to the observed samples. Commonly, the model is a mathematical map/function:

$$R^n \mapsto R$$

where R^n presents n dimension inputs or base variables and R presents scalar space of the dependent variable.

In this map, there are also multiple internal parameters of transformation called weights. The researchers apply their chosen method to estimate these weights values that satisfies the output of map from the corresponding inputs is approximately resemble of the observed samples.

The model, provided by regression analysis, could be considered as an approximate representation of the relationship between the dependent variable and its corresponding based variables. Noting that this statement is only true if the premised model is carefully chosen and the process weight esitmation go right. Then, this statistical model is used to examine our understanding on the data, or applying in the future anticipated of important events.

In our work, this statistical approach is quite potential to analyze our spectral reflectance data. We have a large amount reflected band/wavelength recored for a rice leave, and we have a corresponding data of several concerned chemicals data collected directly from those rice leave through laboratory aproach. These data relationship are fuzzy as well as their amount of factors causing difficulty of directly deducing. Therefore, we try to use several common regression

model applying in our data hoping with luck if the relationship could be discovered here.

Phosphorus

Phosphorus (P) is an important nutrient for plants that supports root growth, energy production, storage, and transfer, and produces flowers and fruits. The lack of P can affect negatively the growth and overall health of plants.

Potassium

Potassium (K) is an essential nutrient for plant growth and development. It plays an irreplaceable role in the physiological processes within plants. It helps control tiny openings and closing pores called stomata on the surface of the leaves. It also helps move water and nutrients inside plants.

Chlorophyll

Chlorophyll is a plant's special reaction that takes part in photosynthesis. It plays an important role in the process of photosynthesis which is how plants convert light energy into chemical energy. Chlorophyll's green color comes from its ability to absorb light in the blue and red parts of the electromagnetic spectrum while reflecting green.

Chlorophyll-A is a special kind of chlorophyll. It grabs most of the energy from violet-blue and orange-red light but it is not very good at catching green light. Instead of reflecting green light, it mostly uses other colors. Chlorophyll-A is the main pigment that helps plants make food from light.

Normalized difference vegetation index (NDVI)

NDVI is a widely-known measurement for monitoring the health and density of vegetation using sensor data. As introduction, the metric used the spectrometric data at 2 specic bands: red and near-infrared(NIR) for its calculation. Most of thesse spectrometric data is obtained through satellites or in our cases, an UAV Drone. Popularized in the industry, it has been proven to have high correlation with the true state of plants. Determine the

$$NDVI = \frac{NIR-Red}{NIR+Red}$$

Red is a visible light, has a fairly long waves, with wavelegth around 625 to 750nm. Infra-red, however, is an electromagnetic radiation spectrum with wavelegth above red which is between 780 nm and 1mm. Together, they could create thounsands of combination. Data limited by these pairs have many affect on the result of the models. Determine the most compatible pairs

for each individual to get the most desirable outcome will be one of the main focus of this study.

2.2.3 Regression Analysis

Regression analysis is a statistical technique for estimating the relevancy between a dependent variable and one or multiple base variables. The technique gain this relevancy by applying following steps: Firstly, researchers a model that is believed to match to the observed samples. Commonly, the model is a mathematical map/function:

$$R^n \mapsto R$$

where R^n presents n dimension inputs or base variables and R presents scalar space of the dependent variable.

In this map, there are also multiple internal parameters of transformation called weights. The researchers apply their chosen method to estimate these weights values that satisfies the output of map from the corresponding inputs is approximately resemble of the observed samples.

The model, provided by regression analysis, could be considered as an approximate representation of the relationship between the dependent variable and its corresponding based variables. Noting that this statement is only true if the premised model is carefully chosen and the process weight esitmation go right. Then, this statistical model is used to examine our understanding on the data, or applying in the future anticipated of important events.

In our work, this statistical approach is quite potential to analyze our spectral reflectance data. We have a large amount reflected band/wavelength recored for a rice leave, and we have a corresponding data of several concerned chemicals data collected directly from those rice leave through laboratory aproach. These data relationship are fuzzy as well as their amount of factors causing difficulty of directly deducing. Therefore, we try to use several common regression model applying in our data hoping with luck if the relationship could be discovered here.

2.2.4 Dimensionality Reduction

Dimensionality reduction is a technique aimed at representing a dataset with a reduced number of features (or dimensions) while preserving the essential characteristics of the original data. This process involves the elimination of irrelevant or redundant features, as well as the removal of noisy data, resulting in a model with fewer variables. Dimensionality reduction includes a range of feature selection and data compression methods employed during preprocessing. Although these methods vary in their approaches, they share the common goal of transforming high-dimensional spaces into low-dimensional ones through the extraction or combination of

variables.

2.2.5 Scikit-learn

Scikit-learn is an open-source machine learning library for python which supports vast amount of data analysis and machine learning algorithms. It propose many utilities components and tools to interact with machine leanring tasks like classification, clustering, and regression analysis as well as data preprocessing tasks like dimension reduction, null data cleaner, etc.

Besides, it is notable that Scikit-learn is designed to interoperate with others populer strongly python library and framework as Numpy (algebraic computation framework) or Scipy (signal computation framework). It allows to transperantly develope a full data processing pipeline with minimal effort on convesion.

In our work, the experiment on different aspect of data is necessary to discover any potential relation between reflectance data to the concetrate level of NPK chemicals. So, for productivity, a universal and reliable framework like Scikit-learn which proposes multiple predefined data analytics solutions readily to apply into an expected data is an irreplaceable option.

2.2.6 PCA

PCA is a method used to reduce the dimensionality of the dataset by transforming variables into smaller ones but still contains important features of the dataset. Principal components are variables represented as linear combinations of the initial elements in the dataset. the dataset after simplification is easier to understand and easier to visualize. Various fields of science have already applied PCA to their data like genetics or climate science where it reduces the complexity of the data.

This is how the PCA works step-by-step:

Step 1: Standardization In this step, the goal is to standardize the variables to have the same impact on the analysis by their ranges. This is a necessary step when PCA. If some variables have much larger ranges, they can unfairly dominate the results. Standardizing involves adjusting data to a common scale by subtracting the average and dividing by the standard deviation for each value of each variable.

$$z = \frac{value - mean}{standard\ deviation}$$

Step 2: Covariance matrix computation This step is to find connections between variables by comparing how they change from their average to each other. This helps spot repeated info. We use the covariance matrix for this

$$cov(X,Y) = \frac{1}{n} \sum_{i=1}^{n} (X - \underline{X})(Y - \underline{Y})$$

- cov(X, Y) is the covariance between X & Y variables
- x & y are members of X and Y variables
- $\underline{x}$ and $\underline{y}$ are mean of X & Y variables
- n is the number of members

It's a chart that shows how variables connect based on their changes. For example, in 3D data with x, y, and z variables, the covariance matrix is a 3x3 grid.

The diagonal of the matrix shows each variable's variance because a variable's covariance with itself is its variance (Cov(a, a)=Var(a)). Moreover, due to the commutative property of covariance (Cov(a,b)=Cov(b, a)), the covariance matrix is symmetric which means that the upper and lower triangular sections mirror each other across the main diagonal.

Step 3: Eigenvectors and eigenvalues Eigenvectors and eigenvalues are linear algebra concepts used to compute from the covariance matrix to determine the principal components of the data. Because these two values always connect in pairs and their number is equal to the dimensions of the data.

The eigenvectors of the covariance matrix contain the most variance of the data and that is called principal components. The eigenvalues are the coefficients of the eigenvectors and thus represent the amount of variance in each principal component.

This step process primarily removes ignorable components while keeping the significant variables and convert into new variables. The principle components indicate the directions of the data.

$$Av = \lambda v$$

- A is the matrix
- V is a special vector
- λ is an eigenvalue

This step process primarily removes ignorable components while keeping the significant variables and convert into new variables. The principle components indicate the directions of the data. The importance of the principal components can be determined by ordering the eigenvec-

tors by their eigenvalues decreasing.

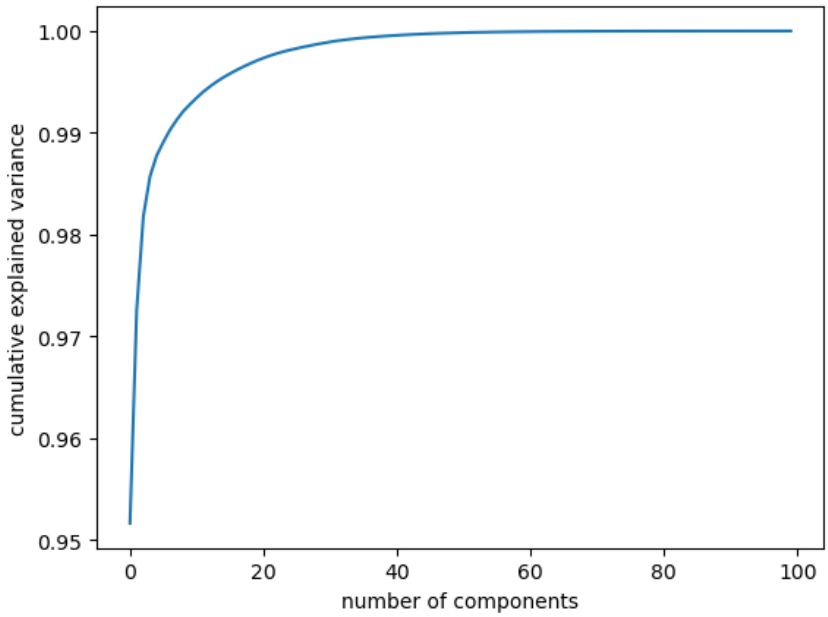
Step 4: Feature vector By computing and sorting eigenvectors by their eigenvalues, we identify principal components in order of importance. The feature vector is a matrix formed from the chosen eigenvectors, serving as the first step in dimensionality reduction. Selecting p eigenvectors out of n reduces the dataset to p dimensions.

Step 5: Reorient the principal components axes

$$FinalDataSet = FeatureVector^T * StandardizedOriginalDataSet^T$$

Throughout the first 4 steps, The data only had been standardiazation without any others change and only select the principal components of the data and the data still remains in the same original axes. Lastly, we aim to reorient the data from the original axes into a new axes which represented by the principal components.

Our dataset in this project is pretty large and contains a high number of dimensions, so we apply PCA to reduce the dimension and increase the model's efficiency. To find the best number of components that will be well-suited to the data, we use the result of the explained variance and the cumulative variance The figure below shows the information on the explained variance of the dataset.



(a)

Figure 2.1: The number of components needed to explain variance

From the figure, five number of component should be enough to get most of the dataset variance coverage,

2.2.7 Machine Learning

Ridge

Ridge regression is also a technique used in linear regression, the same as lasso, which is a tool that helps fix tuning models when dealing with closely related data, called a multicollinearity problem. (9) It applies L2 regularization to handle this. When data has multicollinearity, standard least-square techniques stay unbiased, but variation grows, which leads to substantial differences between predicted parameter estimation. To deal with this, ridge regression adds some bias to improve the accuracy with which parameters are estimated. The cost function for ridge regression can be performed like this:

$$J(\theta) = \frac{1}{2m} \sum_{i=1}^{m}((h_\theta(x^{(i)} - y^{(i)})^2 + \lambda \sum_{j=1}^{n} |\theta_j|)$$

- $J(\theta)$: cost function
- m number of the training set
- $h_\theta(x^{(i)})$: predicted output value of i$^{\text{th}}$ training examples
- λ: regularization term,
- n: number of features,
- θ_j: weight of j$^{\text{th}}$ feature.

Ridge regression works best when having more predicted variables than the observations in the data. This is especially useful in cases where the standard assumptions of linear regression might not be valid. It finds a middle ground between effectively capturing relationships within the data and preventing overfitting issues.

Lasso

Lasso regression is a technique used in regression analysis. Like Ridge Regression, it is a way to shrink and select coefficients in linear regression models, especially with many predictors or multicollinearity. It adds a penalty based on absolute coefficient values (L1 regularization), which can precisely zero out some coefficients. Effectively performing feature selection. This prevents overfitting, simplifies models, and is great when only a subset of predictors is vital.

$$J(\theta) = \frac{1}{2m} \sum_{i=1}^{m}((h_\theta(x^{(i)} - y^{(i)})^2 + \lambda_1 \sum_{j=1}^{n} \theta_j^2 + \lambda_2 \sum_{j=1}^{n} \theta_j^2)$$

- $J(\theta)$: cost function
- m: number of the training set

- $h_\theta(x^{(i)})$: predicted output value of ith training examples
- λ: regularization term,
- n: number of features,
- θ_j: weight of jth feature.

What makes Lasso different from Ridge is that while Lasso can lead to zero coefficients, Ridge keeps predictors, although their magnitude is reduced. By ignoring some specific predictors, Lasso is best for its simplicity and clarity, whereas Ridge is well-suited for situations where numerous influential predictors are involved

ElasticNet

ElasticNet is a powerful and flexible regularization method which have both characteristics of Lasso and Ridge regression, it merges the capability of dealing with high-dimensionality and the need for robust regularization techniques which means that ElasticNet is the combination of L1 and L2 regularization. By mixing L1 and L2 regularization, it aims to strike a balance between simplicity and retaining relevant predictors. ElasticNet provides two parameters, alpha, and lambda, to balance these two types of regularization. The alpha parameter controls the mix of L1 and L2 regularization, allowing us to emphasize one over the other or find an equilibrium between the two. The lambda parameter controls the overall strength of regularization, controlling how much the coefficients shrink

$$J(\theta) = \frac{1}{2m}((h_\theta(x^{(i)} - y^{(i)})^2 + \lambda_1\theta_j^2 + \lambda_2\theta_j^2)$$

- $J(\theta)$: cost function
- m: number of the training set
- $h_\theta(x^{(i)})$: predicted output value of ith training examples
- $y^{(i)}$: real output value of ith training examples,
- λ_1, λ_2:: regularization terms,
- n: number of features,
- θ_j: weight of jth feature.

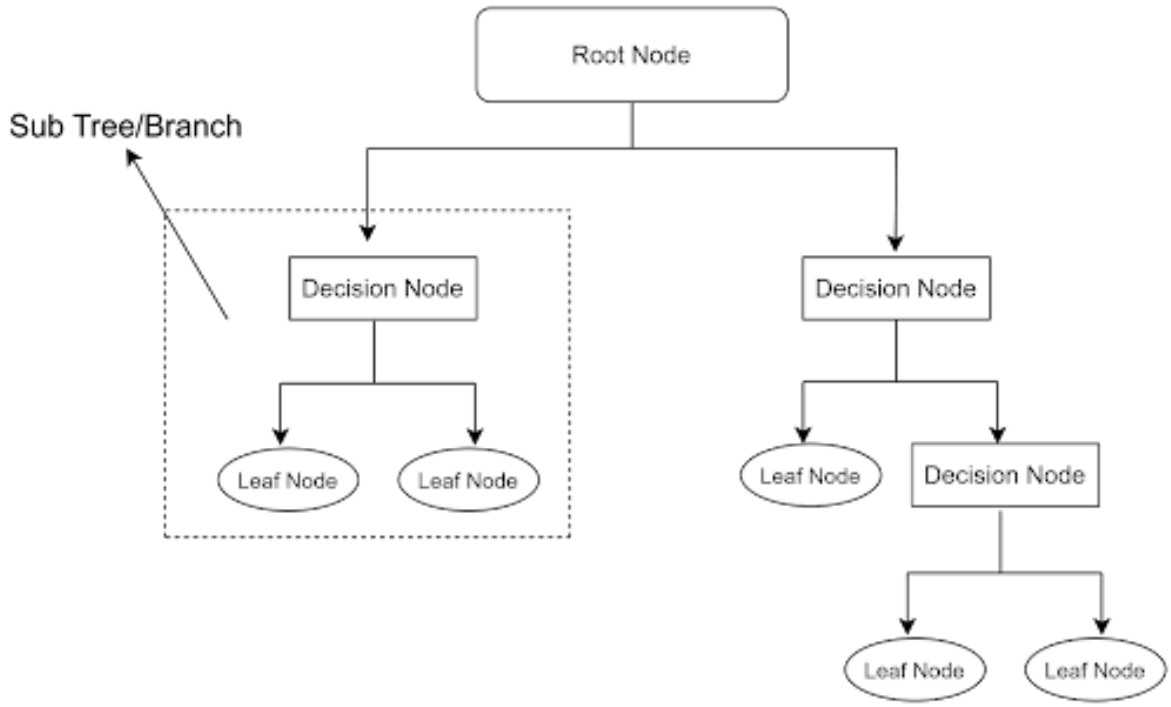
Decision Tree

Decision tree is a versatile algorithm used for classification and regression tasks. A decision tree is a supervised learning technique mainly applied for classification task machine learning, where the data is represented in a tree-like structure.

The Decision tree components include Root Node which is the whole dataset where the tree starts. From the root node, the tree will be divided into a subtree known as decision nodes. At the end of a branch will be a leaf node and cannot be split into another branch.

The main advantage of using the decision tree is the algorithm works like a human thinks and makes decisions and is easy to understand due to the tree-like structure. this approach also requires fewer resources to pre-process than other algorithms. However, the number of branches is higher the more complex the data is and contains lots of layers. The bigger the tree, the more the output will be overfitting.

Pruning - With a big tree model, to prevent the data from overfitting, the data will be pruned which is a process to remove branches containing non-important decisions/output of the tree. Cost complexity pruning and reduced error pruning are the most well-known processes. The other known way to reduce the chance of overfitting is using a random forest algorithm.



(a)

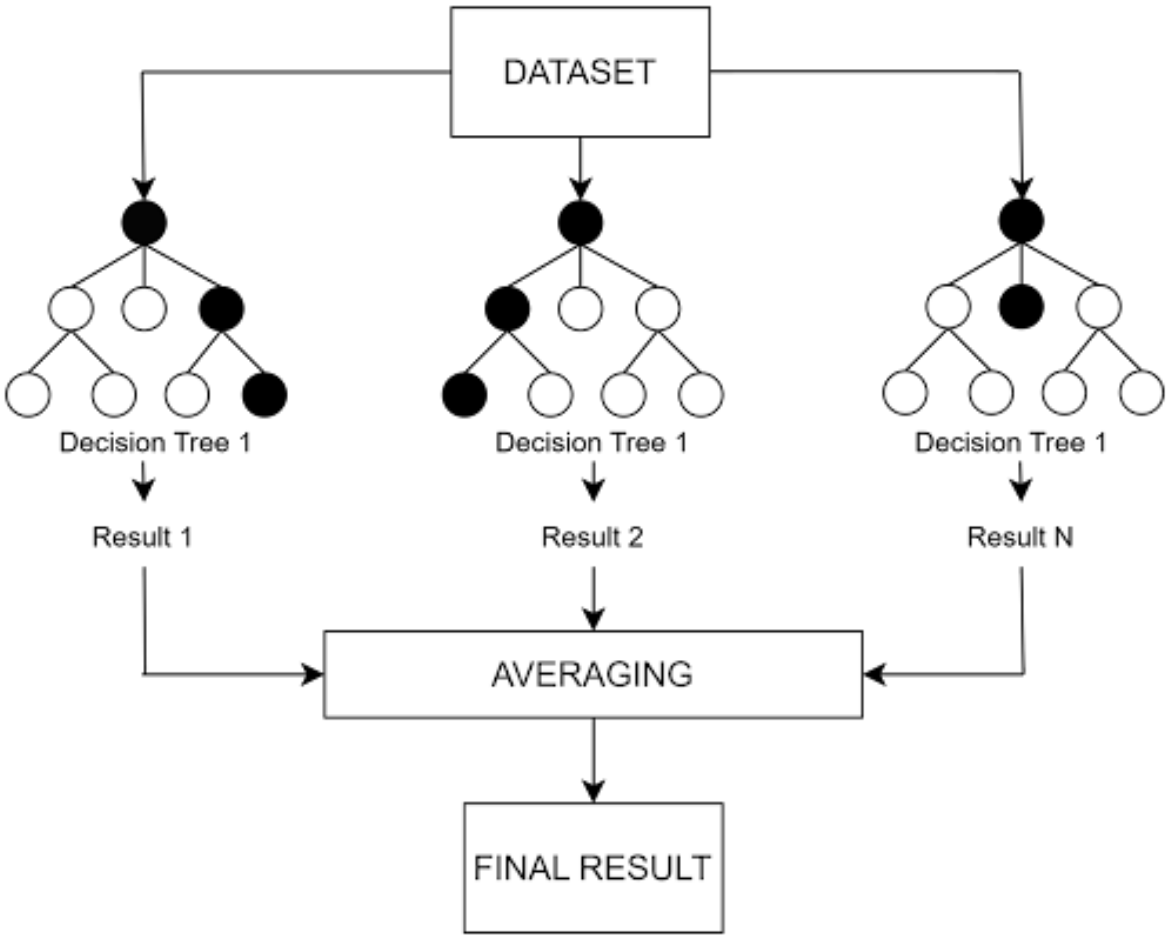
Figure 2.2: Decision Tree workflow

The decision tree starts with data preparation, where a dataset with input features and a target

variable is collected. It selects initial features as the root node, splits the data based on feature values, and calculates the reduction at each step. When the process chooses the best split, which will become the first split of the tree, it will repeat from the beginning to this step. To make predictions, it goes from branches based on feature values to reach leaf nodes, which contain target value predictions that have averaged. After evaluation with metrics like MSE or R squared, it may apply pruning to refine the tree's structure.

Random Forest

Random forest is a learning method for regression or classification tasks that construct a decision tree when training. The random forest algorithms train data while applying the technique of bootstrap aggregating to tree branches to deal with an uncorrelated ensemble of the decision trees. this technique will make sure that each decision tree in the forest is set for a different set of features, by randomly selecting a subset of features from the dataset.



(a)

Figure 2.3: Random Forest workflow

The Random Forest algorithm runs through a bunch of systematic steps to create a predictive

model. It starts with data preparation and then uses bootstrapping to create diverse subsets of the data. For each subset, it constructs a decision tree that contains information about the feature randomness, which means that there is only a random subset of features at each node for splitting. Predictions from these trees are combined through averaging for regression and voting for classification to get to the final prediction. The Random Forest model can be working great through parameter adjustments and evaluated using suitable metrics, while also giving a view of feature importance.

Random Forest brings users several advantages such as their versatility and accuracy when handling diverse data types including binary, numerical, and categorical features, making them robust against outliers and nonlinear features. One of the most useful capabilities is its ability to balance errors in populations and unbalanced datasets, measuring feature importance is straightforward. However, they can be slower due to building many trees, limiting real-time use. The predictions rely on past data and may not work well with different ranges. Also, they are not easy to understand and the decisions cannot be explained easily.

Support Vector Regression

Support Vector Regression (SVR) is a specialized type of support vector machine (SVM) used for regression tasks, targeted to predict continuous output values based on given input data. It can be used both for linear and non-linear kernels, with a simple dot product between input vectors for linear kernels while capturing more complex data patterns for non-linear kernels. Choosing between linear and non-linear is based on data characteristics and task complexity. SVR is based on SVM principles, with the same main role being error minimization while finding a hyperplane with a margin that allows for some error. This minimization process for the parameter 'w' in the equation is akin to maximizing the margin:

$$\min ||w||^2 + C \sum_i^n (\xi_i^+ + \ \xi_i^-)$$

In order to reduce this error, we utilize the following equation, where the summation component represents an empirical error.

To minimize the error, we use the following equation:

$$f(x) = \ \sum_i^n (\alpha_i^* + \ \alpha_i) K \ (x, x_i) + B$$

And to calculate the kernel K we can use the following equation:

$$K\,(x, x_i) = \ \gamma\,(x * \ x_i + 1)^d$$

Research with other algorithms shows that SVR acquires better results when working with Linear Regression or ElasticNet. The algorithms also show high accuracy results when working

with large datasets with a high number of variables. The SVR has high compatibility when used alongside functions such as geometric, transmission, or data generalization. Implement standardization is highly recommended when ensuring unbiases evaluations.

Bayesian Ridge

Bayesian linear regression predicts the mean of one variable using a weighted sum of others. Its goal is to calculate the posterior probability of regression coefficients and other distribution parameters based on observed predictors. Bayesian regression suits datasets with sparse or poorly distributed data, as it derives the posterior distribution of model parameters rather than estimating them directly.

$$p(y \mid X, w, a) \;=\; N\left(y \mid X_w, a\right)$$

Bayesian Ridge regression, which is the most widely used type of Bayesian regression, models regression problems by incorporating probability estimates. This helps account for uncertainty in predictions, making it useful for situations with limited data or noise. The prior for the coefficients w is given by spherical Gaussian as follows.

$$p(w \mid \lambda) \;=\; N\left(w \mid 0,\; \lambda^{-1} I_p\right)$$

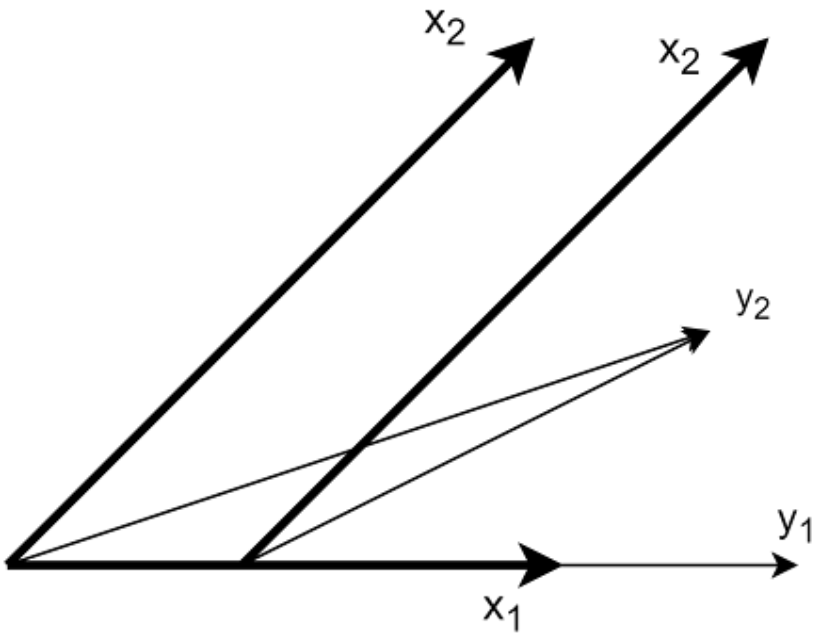
Lasso Lars

Lasso Lars regression is the mixture of two techniques: Lasso and Lars.

Lasso is primarily used for feature selection and addressing multicollinearity in linear regression models. On the other hand, Lars is an algorithm used for efficiently selecting variables in a high-dimensional dataset. Combining these two techniques, Lasso Lars inherits the feature selection capabilities of Lasso and the efficient variable selection process of Lars. This makes it particularly well suited for linear regression tasks involving datasets with many predictor variables or situations where multicollinearity is present. It starts with all coefficients at zero and selects predictors based on their correlation with the target variable. As it progresses, it employs Lasso's feature selection, encouraging some coefficients to become zero, simplifying the model. This approach excels in handling high-dimensional data efficiently while building interpretable and accurate linear regression models.

Lars Least Angle Regression (Lars) is used for feature selection and model building, which is specially designed for high dimensional datasets. Its main task is to select and incorporate the most correlated contributions into the model without overfitting. To fit the models, we start by normalizing all values. Then we choose the most highly correlated variable with the residual

and adjust the regression line. This process continues until we have used all data or created a satisfactory model.



(a)

Figure 2.4: Lars Skrinkage

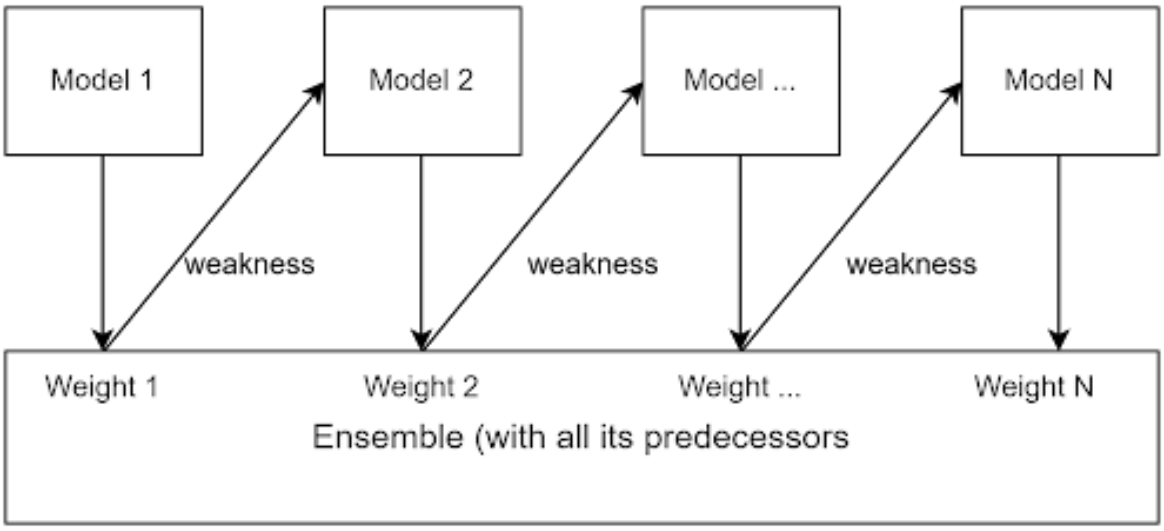
- y2 is the projection of y onto L(x1, x2)
- Two covariates x1 and x2 and the space L(x1, x2) that is spanned by them
- Start at $\mu_0 = 0$

Firstly, we start with all coefficients as zero and find the predictor variable xj that correlates most with the target variable y. Then we increase the coefficients Bj in the direction of this correlation until another predictor variable xk with equal or higher correlation is found. The coefficients (Bj, Bk) are adjusted so that they have the same angle with xj and xk. This process continues until all predictor variables are included in the model.

Boosting

Boosting is a powerful ensemble meta-algorithm in machine learning that reduces bias and variance in supervised learning. It transforms weak learners into strong ones by combining them iteratively and adjusting their weights based on accuracy. Boosting emerged from the idea of enhancing weak learners to create strong ones. Boosting methods assume training weak classifiers sequentially, making it an essential concept in machine learning and statistics. By

each stage of adding, a process called ‘re-weighting’, misclassified data points to gain higher weight, allowing weak learners to focus on them. There are a large number of types of boosting algorithms but in this project, we are supposed to use just the most popular which are AdaBoost, XGBoost, LGBM, CatBoost, and GradientBoost.



(a)

Figure 2.5: Boosting algorithm workflow

The boosting algorithm contains several advantages, including improved accuracy achieved by combining predictions from weak models, robustness against overfitting by giving more weight to misclassified data points, and effective handling of imbalanced datasets. However, boosting algorithms also have some limitations which can be named as the sensitivity which will make them less suitable when working with real-time applications. Unlike others focused on high-quality predictions, boosting algorithms rely on weak models, each addressing the predecessor’s weaknesses.

Gradient Boosting

Gradient Boosting is a robust boosting algorithm that combines weak learners into strong ones by training each new model to minimize the loss function of the previous model using gradient descent. In each iteration, it calculates the gradient of the loss function regarding the predictions of the current ensemble and trains a new weak model to minimize this gradient. This method often uses decision trees as weak learners to transform the data. The iterative procedure involves computing residuals, which represent the difference between predictions and actual values, and training models to map features to these residuals. These steps are to improve the overall predictive performance of the model.

We can express how GradientBoosting works through these steps:

Step 1: Assume X and Y are the input and target with N samples. The goal is to find a function f(x) that maps input features X to target variables Y. The loss function quantifies the difference between actual and predicted values

$$L(f) = \sum_{i=1}^{N} L(y_i,\ f\,(x_i))$$

Step 2: Focuses on minimizing L(f) with respect to f f0x=argmin f Lf= argmin f i=1NL(yi, fxi) In our gradient boosting algorithm with M stages, we improve the model fm by introducing additional estimators denoted as hm, where m ranges from 1 to M

$$\widehat{f_0}(x) = argmin\ f\ \ L(f) = \ argmin\ f\ \ \sum_{i=1}^{N} L(y_i,\ f\,(x_i))$$

Step 3: Steepest Descent

$$g_{im} = \ -\lfloor \frac{\partial L(y_i,\ f(x_i))}{\partial f(x_i)} \rfloor_{f(x_i) = \ f_{m-1}\ (x_i)}$$

With the M stage of Gradient Boosting, the Steepest Descent technique is used to determine the hm which is an important component. This is the combination of 2 elements: constant termed as step length, and the gradient of the loss function gm. The key point of the step length is to scale the gradient of the loss function L(f).

Step 4: we update the solution iteratively using

$$f_m(x) = \ f_{m-1}(x) + \left(argmin\ h_{m \in H}\ \ \left[\sum_{i=1}^{N} L\,(y_i,\ f_{m-1}\,(x_i) + \ h_m\,(x_i))\right]\right)(x)$$

This process continues for M trees, refining the model at each stage to achieve a more accurate prediction. The solution can also be written as:

$$f = \ f_{m-1} - \ \rho_m g_m$$

Adaptive Boosting

AdaBoost, short for Adaptive Boosting is a machine learning algorithm that combines the outputs of weak learners to create a strong classifier. It is known for its adaptability and the ability to handle various base learners including weak ones like decision stumps or even strong learners like deep decision trees, making it versatile. AdaBoost adapts by giving more emphasis to instances that previous learners misclassified, reducing the risk of overfitting. It assigns different weights to errors, influencing the importance of weak learners in the final model. This is an example of how AdaBoost works through a pseudocode:

Input: Data set $D = \{(x_1,\ y_1)\,,\ (x_2,\ y_2)\,,\ \ldots\ (x_m,\ y_m)\}\,;$

Base learning algorithm L;

Output: $H(x) = sign(\sum_{t=1}^{T} \alpha_t h_t(x)$

Number of learning round T.

Process:

$D_1(i) = 1/m$ % Initialize the weight distribution

for $t = 1,\ \ldots\ ,\ T$;

$h_t = L\,(D,\ D_t)\,;$ % Train a weak learner h_t from D using distribution D_t

$\epsilon_t = \Pr_{i\sim Di}[h_t(x_i \neq y_i)]$; % Measure the error of h_t

$\alpha_t = \frac{1}{2} \ln \ln \left(\frac{1-\epsilon_t}{\epsilon_t}\right)$; % Determine the weight of the h_t

$D_{t+1} = \frac{D_t(i)}{Z_t} \times \{\exp\ (-\alpha_t)\ \ if\ h_t(\,x_i) = y_i\ exp\ (\alpha_t)\ \ if\ h_t(\,x_i) \neq y_i$

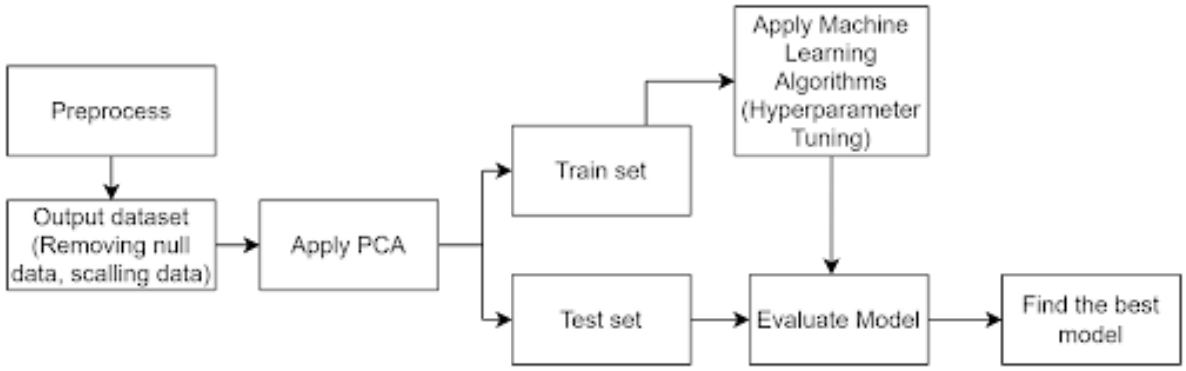
$= \frac{D_t(i) exp(-\alpha_t y_i h_t(x_i))}{Z_t}$ % update the distribution, where Z_t is

% a normalization factor which enables D_{t+1} be a distribution end.

AdaBoost offers several advantages, notably its ease of use with minimal parameter tuning, in contrast to more complex algorithms like SVM. However there are some disadvantages, AdaBoost's progressive learning process demands high-quality data as it's sensitive to noise and outliers.

2.3 Implementation

2.3.1 Study Pipeline



(a)

Figure 2.6: The models workflow

First of all, the data need to be prepared before applying with any model. 260 reflectance information files each including 2150 rows of figures. After proccessing the raw data and removing all the null data, the result is an output csv file with 168 row and 2154 collumns. The obtained data may contains features with a variety of dimensions and scales. These differences leads to a biased predictions results in terms of accuracy rates. To avoid this, it is necessary to implement standardization scale to the data before applying to modelin. The next step, the main works of this project, is the analysis and study the relationship of the reflectance data with the plants nutrients. The dataset is splitted into a train set and a test set with a 80-20 portions. To make the model more effective in exploring such a high-dimensional datasets, two methods is proposed. One is using PCA which have been widely used with machine learning algorithms. Others method, limiting which reflectance bandwidth is selected to ran through the models can show the correlation betwween the bandwidth and the plant's nutrient. Since multple models are used in this study, optuna is applied as a tool to automatically find the best hyperparameter for each corresponding models. Finally, the models is continously executed using the dataset that is limited by the red and infra-red bandwidth. From all the collected result, the prediction data and desirable wavelegth, we can conclude which one will be the best model and technique for resolving our challenge.

2.3.2 Tools & Library

For this project, most of the work is ran using ICTlab servers.

- Scikit-learn: Scikitlearn is an open-source machine learning library for Python. It offers several tools for various machine learning tasks including classification, regression, clustering, dimensional reduction, model selection, and data preprocessing.

- Optuna: This is an open-source hyperparameter optimization for machine learning models, which helps in finding the best hyperparameters, making it easier to improve performance and accuracy. Optuna contains various optimization algorithms to search for the optimal hyperparameter in a defined space.

- Numpy, Matplotlib: used for scientific computing and visualization. Numpy is used for numerical computing and Matplotlib is a tool for creating data visualization.

2.3.3 Prepare the dataset

As mentioned above, in this study, two type of dataset need to be put together. First is the main dataset which is the combination of 2 file. The csv file, which contain the measured data of the plant nutrients and its position. The sed file contain most of the information of the vegetation spectral reflectance, however it needs to filter a lot of noise data before merging. Secondly,

another dataset need to generated, the steps remain largely the same, except to analyse which bandwidth is most compatible, the spectral reflectance data need to be limited by the corresponding red and infra red bandwidth. While the highest bandwidth in the dataset is only 2500 nm, the number of time need to ran for each model is over 20000 combination. Most simple model can run through the experiements just fine, However for more advance model, this method could be quite taxing procedure.

2.3.4 Hyperparameter Optimization

Hyperparameter Optimization in Machine learning is the tool to select the best parameters for the learning algorithm. These hyperparameters find the values that can improve the efficiency of model performance. The main idea is to tune parameters to ensure that the model can handle data patterns and minimize a predefined loss function. Cross-validation is commonly used to estimate performance and choose the hyperparameter values that maximize it. There are several ways to optimize hyperparameters such as Grid Search, Random Search, etc. Grid Search is working when specifying a range of parameter values and testing them to see which will work best. This process requires performance metrics like cross-validation to guide the choice. However, when dealing with parameters that can take on real or unbounded values, we may need to set boundaries and discrete values before conducting the grid search. Grid Search is not such a great choice when working with high dimensional parameters space. So to work with more advance model likes AdaBoost, we choose Optuna, which is an automatic hyperparameter optimization framework, as a backup plan. Optuna runs based on the define-by-run approach, which will bring flexibility when working with high-dimensional spaces for hyperparameters.

2.3.5 Model Evaluation

Mean Squared Error (MSE)

MSE quantifies the average squared difference between estimated values and actual values. In machine learning, it is one of the popular metrics when evaluating the quality of predictors or estimators. MSE is a non-negative value increasing as errors grow in the model. It considers both variance (how wide estimates vary across data samples) and bias (how far the average between estimates and true value). For an unbiased estimator, MSE equals the variance of the estimator. MSE can be given as the following equation:

$$MSE = \frac{1}{n} \sum_{i=1}^{n} (Y_i - \widehat{Y}_i)^2$$

- Y_i: the i_th observed value,
- $\widehat{Y}_i$: the corresponding predicted value,

- n = the number of observations.

R square

R squared or R2 (coefficient of determination) is one of the metrics to understand how the output values (variance of a dependent variable) are explained by an independent variable. The value ranges from 0 to 1 and can be negative if the model performs worse than the average fit. An R2 above 0.7 can be signified as a strong correlation, while below 0.4 signified a weaker one.

$$R^2 = 1 - \frac{SS_{Regression}}{SS_{Total}}$$

- $SS_{Regression}$: the sum of squares due to regression,
- SS_{Total} : the total sum of squares.

Mean Absolute Percentage Error (MAPE)

Mean Absolute Percentage Error (MAPE) is a metric for evaluating the accuracy of predicting methods, especially when we are working with large and nonzero dataset values. It calculates the percentage error between predicted and actual values. A MAPE below 5% is a highly accurate prediction, while a MAPE between 10% and 25% is acceptable. But when it exceeds 25%, it means that the results come out have very low accuracy, equivalent to an unacceptable prediction.

$$MAPE = \frac{1}{n} \sum_{t=1}^{n} \left| \frac{A_t - F_t}{A_t} \right|$$

- n : sample size,
- A_t: the actual data value,
- F_t: the forecasted data value.

Chapter 3

Result & Discussion

3.1 Evaluation

3.1.1 Chlorophyll Model Prediction

Models	Without PCA			PCA (5 components)		
	MSE	R^2	MAPE	MSE	R^2	MAPE
Linear Regression	10.16	-0.00	6.38%	9.71	0.04	6.30%
Lasso Regression	10.64	-0.01	6.57%	10.17	0.00	6.38%
Decision Tree	11.53	-0.09	6.69%	10.45	-0.03	6.57%
Random Forest	**9.41**	**0.07**	**6.27%**	9.67	0.05	6.32%
Ridge	10.60	0.00	6.65%	9.95	0.02	6.48%
ElasticNet	10.41	0.02	6.57%	10.17	0.00	6.38%
SVR	10.23	-0.01	6.42%	10.24	-0.01	6.42%
Bayesian Ridge	10.17	0.00	6.38%	10.20	-0.01	6.44%
Gradient Boosting	9.97	0.02	6.42%	10.84	-0.07	6.74%
Lasso Lars	10.17	0.00	6.38%	10.18	0.00	6.41%
Lars	10.64	-0.01	6.57%	9.95	0.02	6.48%
AdaBoost	11.10	-0.05	6.52%	**9.55**	**0.06**	**6.28%**

Table 3.1: Comparison of Learning Models Performance in Chlorophyll Prediction

The Chlorophyll performance with the dataset basically is strong with a quite low result in

MAPE of less than 7% for most of the models, some have higher MAPE result, especially after applying 5 PCA component. Among the models, the Random Forest model without applying PCA seems to be the top performer (with a score of 9.41 of MSE, the highest R^2 score, and a MAPE of 6.27% which is also the best one). Though after using PCA with 5 components, while it still have decent score, Random Forest (with 9.41 MSE) is outperformed by AdaBoost which have 6.28% MAPE score. With this result, we can conclude that the boosting algorithms are the most well-suited to the dataset when . For the machine learning algorithm, there is nothing better than the result of Random Forest without PCA when it has 6.27% in MAPE, 9.41 in MSE and the R^2 score is 0.07 which is also a good score. When taking a look at the higher number of PCA components, we can see that the result seems to be better in some models, especially with AdaBoosting when the MSE score increases impressively (from 11.10 to 9.55 in MSE score) through each higher stage of PCA. The results in MAPE and MSE are more stable than with the lower number of components which is around 10 in MSE and around 6.5 in MSE. If choosing one best model for 5 PCA components, AdaBoost, and Linear Regression, among them, Linear Regression seems to have the overall results pretty well. The worst performance is Decision Tree without applying PCA's results (with -0.82 in R^2 score, the highest MSE 11.53, and the highest MAPE 6.69%.). After applying PCA the results are much better but still not very desirable MSE: 10.45, R^2: -0.03, MAPE: 6.57%. The other noticable bad performance is the Gradient Boosting after applying 5 component of PCA MSE: 10.84, R^2: -0.07, MAPE: 6.74%.

Overall Chlorophyll seems to have the best performance for machine learning models because of having the best result when compared with P and K concentrations.

Methods	Red bw	IR bw	R square	MSE	MAPE
Gradient Boost	735	1084	0.03	8.299285863	5.28%
Gradient Boost	738	1093	0.03	8.28778986	5.30%
Gradient Boost	730	1084	0.03	8.275941423	5.30%
Gradient Boost	730	1094	0.03	8.286555199	5.30%
Gradient Boost	737	1084	0.03	8.311230227	5.30%
Random Forrest	680	1665	-0.01	8.67683125	5.53%
Random Forrest	650	1067	0.01	8.493064482	5.54%
Random Forrest	642	764	0	8.588484947	5.54%
Random Forrest	677	937	0	8.516774513	5.55%
Forrest	637	1300	0	8.527544172	5.55%

Methods	Red bw	IR bw	R square	MSE	MAPE
SVR	625	750	-0.19	10.15272789	5.77%
SVR	625	750	-0.19	10.15272789	5.77%
SVR	625	766	-0.16	9.890171071	5.78%
SVR	625	765	-0.16	9.910342117	5.78%
SVR	625	764	-0.16	9.928388891	5.78%
Linear Regression	740	758	-0.22	10.46803389	5.83%
Linear Regression	740	757	-0.22	10.47387994	5.86%
Ridge	740	756	-0.17	10.02904318	5.91%
Ridge	740	750	-0.17	10.02936277	5.91%
Ridge	740	751	-0.17	10.03365566	5.91%
Ridge	740	752	-0.17	10.03727319	5.91%
Ridge	740	755	-0.17	10.03919053	5.91%
Linear Regression	740	759	-0.26	10.7475421	5.92%
Linear Regression	740	754	-0.27	10.83335362	5.94%
Regression	740	756	-0.27	10.8463624	5.98%
Elasticnet	660	1426	-0.03	10.31318804	6.33%
Elasticnet	661	1426	-0.03	10.31316867	6.33%
Elasticnet	659	1426	-0.03	10.31320883	6.33%
Elasticnet	658	1426	-0.03	10.31320323	6.33%
Elasticnet	657	1426	-0.03	10.31320132	6.33%

Table 3.2: Best NDVI result from Learning Models in Chlorophyll Prediction

While applying PCA have quite a good influence on the result, manually limitting the spectral bandwidth showed even greatter improvement to the result. After running through multiple model with different combination of red and infrared bandwidth, the best scores is consistantly achieved using GradientBoosting model in 735-1084nm bandwidth. While the R square result still not as good as using PCA, the model's others score received quite substantial improvement with MSE: 8.3 and MAPE: 5.28%. Overall, eventhough the improvements are vary, all other regression has shown possitive developments, each in a different bandwidth range: Random Forest at 680-1665nm, SVR at 625-750, Linear Regression: 740-758nm, Ridge: 740-756nm and ElasticNet: 660-1426nm.

3.1.2 Phosphorus Model Prediction

Linear Regression	600400.27	0.01	17.85%	602786.91	0.01	17.58%
Lasso Regression	809576.62	0.00	21.79%	606748.22	0.00	17.61%
Decision Tree	798576.62	0.01	21.57%	642453.51	-0.06	18.60%
Random Forest	673434.06	-0.11	18,71%	685801.38	-0.13	18.53%
Ridge	779860.59	0.03	21.29%	622814.65	-0.03	17.81%
ElasticNet	805652.41	0.00	21.52%	614590.82	-0.01	17.72%
SVR	661599.95	-0.09	18.24%	636115.22	-0.05	17.94%
Bayesian Ridge	**591150.94**	**0.03**	**17.56%**	**591923.27**	**0.03**	**17.57%**
Gradient Boosting	615169.00	0.02	17.94%	644217.55	-0.06	18.41%
Lasso Lars	627916.94	-0.33	18.21%	622582.18	-0.02	17.81%
Lars	831396.92	-0.03	22.05%	622791.61	-0.03	17.81%
AdaBoost	793464.37	0.02	23.31%	647898.54	-0.07	18.49%

Table 3.3: Comparison of Learning Models Performance in Phosphorus Prediction

Next moving to the performance of Phosphorus concentration. Generally, the MAPE scores is quite good (20%), it seems worse than Chlorophyll's performance but still acceptable. The MSE and the R^2 score are unfortunately high, some of the R^2 scores are very low. After using PCA with 5 components, we can pick out some great results improvement which are the score belonging to AdaBoost and Lasso Regression, However, the score of Bayesian Ridge model still remain the best result. Even though the overall performance is great there are still some not-great results with a little high MAPE (¿20%) without applying PCA, such as Lasso, Ridge, AdaBoost, ElasticNet, and Lars, but Lars seems to have the worst performance when compared in total (with highest MSE and a R^2 score remain unimproved, and the MAPE is also high but not as high as AdaBoost's MAPE).

Method	Red bw	IR bw	R square	MSE	MAPE
Elasticnet	660	1426	0.06	672,150.97	16.51%
Elasticnet	661	1426	0.06	672,136.34	16.51%
Elasticnet	659	1426	0.06	672,216.12	16.51%
Elasticnet	658	1426	0.06	672,347.03	16.51%
Elasticnet	657	1426	0.06	672,455.44	16.51%

Method	Red bw	IR bw	R square	MSE	MAPE
Random Forrest	642	764	0.01	679,799.94	19.96%
Random Forrest	680	1665	-0.03	706,177.95	20.01%
Forrest	637	1300	-0.03	702,782.86	20.25%
Ridge	740	750	-0.09	745,210.37	20.39%
Ridge	740	751	-0.09	745,749.16	20.40%
Ridge	740	752	-0.09	746,331.77	20.41%
Ridge	740	756	-0.09	746,576.25	20.41%
Random Ridge	740	755	-0.09	747,475.01	20.42%
Forrest	650	1067	-0.12	764,132.21	20.53%
Gradient Boost	730	1094	-0.12	764,676.10	20.60%
Gradient Boost	730	1084	-0.12	766,628.28	20.63%
Gradient Boost	737	1084	-0.12	766,559.43	20.64%
Gradient Boost	735	1084	-0.12	766,860.31	20.64%
Gradient Boost	738	1093	-0.12	765,972.87	20.65%
Linear Regression	740	754	-0.17	803,237.16	21.26%
SVR	625	766	-0.18	809,677.58	21.27%
SVR	625	765	-0.18	810,264.58	21.27%
SVR	625	764	-0.19	811,001.53	21.28%
SVR	625	750	-0.2	823,641.89	21.51%
Random SVR	625	750	-0.2	823,641.89	21.51%
Linear Forrest	677	937	-0.2	820,829.62	21.62%
Linear Regression	740	756	-0.33	911,833.28	21.86%
Linear Regression	740	757	-0.34	917,429.71	21.99%
Linear Regression	740	758	-0.47	1,002,756.80	23.36%
Regression	740	759	-0.49	1,019,264.06	23.54%

Table 3.4: Best NDVI result from Learning Models in Phosphorus Prediction

Just like with Chlorophyll, the models is analysed even futher to find their most compatible bandwidth , ElasticNet in the bandwidth 660- 1426, show the most improvement and have the best result even in R^2, MSE: 672150.97, R^2: 0.06, MAPE: 16.51%. However, others models performance was not very desirable, all of them failed to surpass their original scores.

3.1.3 Potassium Model Prediction

Models	Without PCA			PCA (5 components)		
	MSE	R^2	MAPE	MSE	R^2	MAPE
Linear Regression	107236125.48	-0.01	27.88%	107373204.57	-0.02	27.97%
Lasso Regression	127745308.26	-0.12	36.57%	103388230.49	0.02	27.22%
Decision Tree Random	134777212.87	-0.18	38.00%	**98501105.81**	**0.07**	**26.34%**
Forest	**95390644.85**	**0.10**	**25.69%**	107056102 1.26	-0.01	27.54%
Ridge	127064172.96	-0.11	35.70%	106784938.14	-0.01	27.62%
ElasticNet	119845469.73	-0.05	35.21%	103690201.94	0.02	27.24%
SVR Bayesian	106589283.16	-0.01	27.52%	106614398.08	-0.01	27.54%
Ridge Gradient	107221952.93	-0.01	27.88%	107221946.63	-0.02	27.88%
Boosting	96605006.46	0.09	26.19%	99717999.88	0.06	26.52%
Lasso Lars	107233510.81	-0.02	27.88%	103450924.98	0.02	27.17%
Lars	126196026.94	-0.11	36.11%	106815493.01	-0.01	27.63%
AdaBoost	148137256.36	-0.30	38.73%	98205464.34	0.07	26.76%

Table 3.5: Comparison of Learning Models Performance in Potassium Prediction

Finally on with the Potassium prediction, in general, the MAPE is not so good, especially when compared with Chlorophyll and P. The amount in MAPE is quite high but still never exceeds 40% though MSE is not as good as we expect. This is maybe because the quality of our dataset doesn't have a good measure. Overall, to give a comment on these results, we could say that almost every R^2 score when applying PCA is better than the original. We can see that Random Forest is the best model even without applying PCA, MSE: 95390644.85, R^2: 0.10, MAPE: 25.69% After PCA implementation, the Decision Tree model shows the most improvement (0.07 in R^2 score, 26.34% in MAPE, and 98501105.81 in MSE). Another model that has very good performance is AdaBoost with PCA applied. However, AdaBoost without

PCA became the worst model when having such high results in MSE and MAPE, and the lowest R^2 (148137256.36 in MSE, -0.30 in R^2 score and surprisingly high MAPE 38.73%).

We can conclude that with K concentration, applying PCA improves the overall performance of all the models, making the outcomes more stable and better.

Method	Red bw	IR bw	R square	MSE	MAPE
Elasticnet	661	1426	-0.03	105,099,723.87	24.94%
Elasticnet	660	1426	-0.03	105,110,232.58	24.94%
Elasticnet	657	1426	-0.03	105,119,786.92	24.94%
Elasticnet	659	1426	-0.03	105,120,528.25	24.94%
Elasticnet	658	1426	-0.03	105,125,719.86	24.94%
SVR	625	750	-0.04	110,605,690.54	30.89%
SVR	625	750	-0.04	110,605,690.54	30.89%
Ridge	740	755	-0.07	113,855,424.25	30.93%
Ridge	740	756	-0.07	113,875,397.26	30.93%
Ridge	740	752	-0.07	113,922,615.78	30.94%
Ridge	740	751	-0.07	113,950,018.38	30.94%
Ridge	740	750	-0.07	113,971,957.88	30.95%
SVR	625	764	-0.05	111,531,206.47	31.07%
SVR	625	765	-0.05	111,604,128.04	31.09%
SVR Linear	625	766	-0.05	111,676,384.01	31.10%
Regression	740	754	-0.42	150,687,608.47	32.18%
Gradient Boost	730	1094	-0.24	131,477,468.56	33.66%
Gradient Boost	730	1084	-0.25	132,642,489.61	33.77%
Gradient Boost	737	1084	-0.25	132,591,840.63	33.78%
Gradient Boost	735	1084	-0.26	133,296,164.45	33.81%
Gradient Boost Random	738	1093	-0.26	133,595,962.28	33.92%
Forrest Linear	680	1665	-0.36	144,501,023.36	34.53%
Regression Random	740	759	-1.03	215,001,655.07	35.02%
Forrest Linear	650	1067	-0.49	158,479,178.69	35.74%
Regression Random	740	758	-1.16	228,513,700.11	35.99%
Forrest Random	642	764	-0.51	160,232,536.88	36.23%
Forrest	637	1300	-0.52	160,836,192.42	36.49%

Method	Red bw	IR bw	R square	MSE	MAPE
Random Forrest	677	937	-0.51	160,469,704.42	36.99%
Linear Regression	740	756	-1.26	239,595,865.40	37.30%
Linear Regression	740	757	-1.27	240,574,703.41	37.54%

Table 3.6: Best NDVI result from Learning Models in Potassium Prediction

When applying the K regression models to find the optimal red-IR range for each individual, ElasticNet surprisingly remain as the technique not only with the highest score but also the best improvement. At 660-1426nm bandwidth, its MAPE score jumps from 27.24% to 16.51% and the R2 also boosted to 0.06. Others methods, however, received quite a penalty hit on their result.

Chapter 4

Conclusion

4.1 Conclusion

In this study, we work with the prediction of multiple kinds of , chlorophyll content, Phosphorus concentration, and Potassium concentration. From the above comparison, we can conclude that the best result is the prediction of the chlorophyll content. This is because the dataset of Chlorophyll is pretty well which can be concluded that the wavelength when collecting data is good. The P and K concentration on the other hand is not as great as with Chlorophyll, especially the worst one among the fields is the K concentration because of its poor performance in MAPE and MSE. Luckily the R2 score was not the greatest but it is still not so bad when compared to P concentration and Chlorophyll content. The possible causes for this bad performance in P and K may be because of the imbalance in measures P and K concentrations, or an insufficient amount of training data available for learning models. When comparing between applying or without utilising PCA, with 5 component can be a good choice to have better results. However, some of them show a decrease in prediction quality. No models stand out too much to be use exclusively with all three nutrient concentration. To pick out the best model for each concentration: Chlorophyll predictions is best compatible with AdaBoost model; Bayesian Ridge model with or without PCA's result remain consistently best for P regression, and for K prediction, Random Forest model have the best calculation. We also learned that PCA is not the only method to reduce the complexity of our dataset. Limiting bandwidth has proven that it can generate even better result: GradientBoosting models at 735-1084nm has the best result when it comes to predict Chlorophyll content; and ElasticNet has shown us that it excel others at both P, K prediction and even utilising very similar frequency: 660-1426nm.

In the modern age, the integration of advanced technologies into agriculture has gained a great amount of popularity. Plant health is important in agriculture, and these technological advance-

ments provide farmers with huge support. Farmers may optimize their operations by using the potential of these technologies, resulting in savings in both time and effort.

4.2 Future work

In future work, we would like to implement the proposed pipeline with the increase the sample in the dataset. Because the current dataset is in one season, just having 3 kinds of concentrations is not enough to train a good model. Besides, studying and applying more models and different kinds of learning algorithms such as the models of Deep Learning may work and get better results. What we want to work with more is to work with other kinds of objects such as vegetable leaves and also find the best model among a large number of concentrations. We also want to work with not just the PCA but other techniques to deal with the high dimensional data issues