

UNIVERSIDAD AUTÓNOMA AGRARIA ANTONIO NARRO
POSTGRADUATE SUBDIRECTORATE



APPLICATION (SOFTWARE) AND ARTIFICIAL INTELLIGENCE ALGORITHMS
DEVELOPMENT FOR AUTOMATED PRESCRIPTION GENERATION IN
PRECISION AGRICULTURE

Thesis

Presented by ANDRÉS CADENA DÍAZ

In partial fulfillment of the requirement for the degree of:
MASTER OF SCIENCE IN PRODUCTION SYSTEMS ENGINEERING

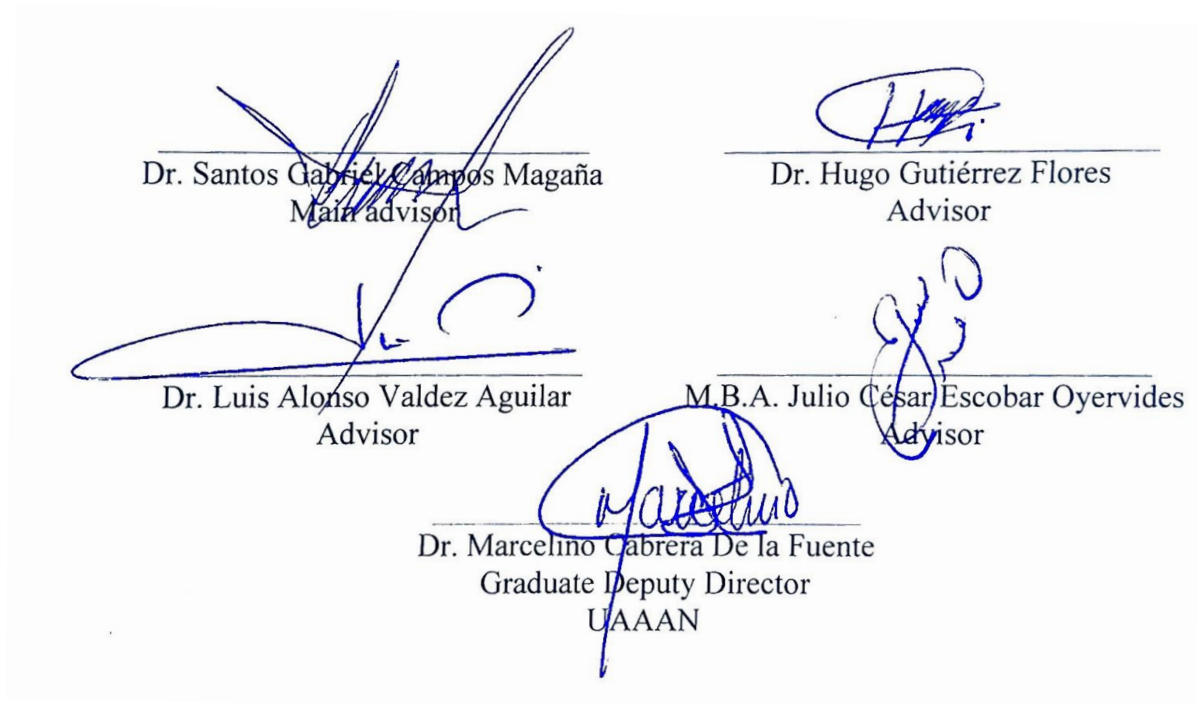
Saltillo, Coahuila

June 2022

APPLICATION (SOFTWARE) AND ARTIFICIAL INTELLIGENCE ALGORITHMS
DEVELOPMENT FOR AUTOMATED PRESCRIPTION GENERATION IN
PRECISION AGRICULTURE

Thesis

Elaborated by ANDRÉS CADENA DÍAZ in partial fulfillment of the requirement for the degree of Master of Science in Production Systems Engineering with supervision and approval by the advisory committee.



Dr. Santos Gabriel Campos Magaña
Main advisor

Dr. Hugo Gutiérrez Flores
Advisor

Dr. Luis Alonso Valdez Aguilar
Advisor

M.B.A. Julio César Escobar Oyervides
Advisor

Dr. Marcelino Cabrera De la Fuente
Graduate Deputy Director
UAAAN

Acknowledgements

To God Almighty for the gift of life and many blessings in the form of health, a beautiful family, professional satisfaction, among others.

To **UAAAN** for the opportunity to study my master's degree in its facilities and being a means for my professional success.

To **Dr. Santos Gabriel Campos Magaña** for his support during my postgraduate studies especially on the first semester since this professional area was completely new for me.

To **Dr. Hugo Gutiérrez Flores** for his support in this research project and guiding me through the technical aspects of Artificial Intelligence and application development. Additionally, for encouraging me to always strive for the best.

To **Dr. Luis Alonso Valdez Aguilar** for kindly providing his expertise in the plant nutrition area of this project, which provided the basis for the agronomic criteria for fertilizer prescription generation.

To **M.B.A. Julio César Escobar Oyervides** for his contribution in the conceptualization of this project, shaping it into a usable product with great market potential.

To the diverse professors of UAAAN, especially **Dr. Martín Cadena Zapata, Dr. Alejandro Zermeño González, Dr. Jorge Méndez González, Dr. Mario Alberto Méndez Dorado, M.Sc. Juan Antonio López López, and M.Sc. Gilbert Fresh López López** for sharing their knowledge and helping shape my postgraduate studies.

To my fellow postgraduate students **Solangel Lescieur López, Juan Carlos Rangel Treviño, Vladimir Mejía, Rocío Mendieta Oviedo, and Juan Manuel Pérez Gómez** thank you all for your sincere friendship.

Dedication

To my family which has supported me in all my endeavors, especially my wife **Linda Sarahí Rojas Uresti**, my parents **Martín Cadena Zapata** and **María de los Ángeles Díaz Altamirano**, and my siblings **Pedro, Juan, and María Cadena Díaz**.

To my advisory committee, professors and fellow students who shaped my master's studies experience.

TABLE OF CONTENTS

1	INTRODUCTION	1
1.1	Contribution.....	2
1.2	Objectives.....	3
1.2.1	Main objective	3
1.2.2	Specific objectives.....	3
1.3	Hypothesis.....	3
2	LITERATURE REVIEW.....	4
2.1	Background	4
2.1.1	Problem: Crop-input cost and ecological footprint	4
2.1.2	Precision agriculture	5
2.2	Target market.....	14
2.2.1	Corn production in Mexico, a representative crop.....	14
2.2.2	High yield corn production.....	16
3	MATERIALS, EQUIPMENT, AND METHODS.....	21
3.1	Materials	21
3.1.1	Fields A, B, and C.....	21
3.1.2	Field D	22
3.2	Equipment.....	25
3.2.1	Software	25
3.2.2	Hardware.....	27
3.3	Methods.....	29
3.3.1	Data preparation.....	29
3.3.2	Yield modeling.....	33
3.3.3	Variable-rate recommendations.....	34
3.3.4	AgStat application	38
4	RESULTS AND DISCUSSION	56
4.1	Yield modeling.....	56
4.1.1	Fields A, B, and C (ANN Predictions)	56

4.1.2	Field D (Fixed exponential model)	61
4.2	Variable-rate recommendations.....	61
4.2.1	Fields A, B, and C (ANN predictions)	61
4.2.2	Field D (Linear model)	66
4.3	AgStat application	68
4.4	Discussion.....	73
5	CONCLUSIONS.....	77
6	REFERENCES.....	78
7	ANNEXES	86

LIST OF FIGURES

Figure 1. Mexican farmers' main problematics.	4
Figure 2. Information process flow of Precision Agriculture.	5
Figure 3. Hierarchy of Precision Agriculture Technologies	6
Figure 4. Overview of variable rate herbicide implementation in a vineyard.	8
Figure 5. Generic example of a fully connected ANN structure.....	12
Figure 6. Activation functions.....	14
Figure 7. Mexican farmers' classification.....	15
Figure 8. Average agricultural machinery age in Mexico 2019.	17
Figure 9. Mexico's white corn production classes 2019.....	18
Figure10. Mexico's sweet corn production classes 2019.....	18
Figure 11. Target corn market location.....	19
Figure 12. Planted sorghum surface per municipality 2020.	19
Figure 13. Sorghum yield per municipality 2020.	20
Figure 14. Alternate market location.	20
Figure 15. Location map (Fields A, B, and C).....	21
Figure 16. Location map (Field D).	22
Figure 17. Raw photo (left) orthophoto (right).	23
Figure 18. NDVI orthomosaic.....	23
Figure 19. Soil sampling distribution.....	24
Figure 20. Yield samples' spatial distribution	25
Figure 21. Jetson Nano Board.....	27
Figure 22. Original image (left), Manually digitized polygon features (right).....	29
Figure 23. Procedure followed to convert digitized polygons to point features	30

Figure 24. Point features ready to be interpolated.	31
Figure 25. Normalized X, Y coordinates.	32
Figure 26 Exponential model fitted to yield and NDVI data.	34
Figure 27. Response of two corn hybrids to seeding rate and environmental index.	35
Figure 28. Field A, ECa range 0.4 – 0.5	36
Figure 29. Average yield for each ECa range and seeding rate in field A.....	36
Figure 30. ANN prediction’s data processing.....	37
Figure 31. AgStat’s color palette	39
Figure 32. Analysis menu	39
Figure 33. Prescription processing steps.	41
Figure 34. Data visualization utility screens.	42
Figure 35. Data acquisition system.....	43
Figure 36. Files utility.	44
Figure 37. Settings menu interface	45
Figure 38. Client database’s general structure.	46
Figure 39. Proposed folder structure.....	47
Figure 40. Integrated data processing functionalities.	51
Figure 41. File systems and FolderLabel objects.....	54
Figure 42. Interpolated surfaces derived from point features (field A)	56
Figure 43. Interpolated surfaces derived from point features (field B).....	56
Figure 44. Interpolated surfaces derived from point features. (field C).....	57
Figure 45. Interpolated database (ECa vs Yield)	57
Figure 46. Interpolated database (altitude vs yield).....	58
Figure 47. Interpolated database (Seeding rate vs yield).....	58
Figure 48. Model loss.....	59

Figure 49. Yield prediction results by ANN (fields A, B, and C).....	60
Figure 50. Estimated yield map generated by regression (field D).....	60
Figure 51. Model 1 response.....	62
Figure 52. Seeding rate recommendations from Model 1.....	62
Figure 53. Optimized seeding rate functions.....	63
Figure 54. Model 2 Loss.....	63
Figure 55. Seed prediction results from model 2.....	64
Figure 56. Seeding rate recommendations from model 2 (field B).....	64
Figure 57. Yield and seeding differences.....	65
Figure 58. Screenshot of urea prescription in AgStat.....	66
Figure 59. Comparison between nitrates distribution and urea prescription.....	67
Figure 60. AgStat's main menu.....	68
Figure 61. Prescription generation process screens.....	69
Figure 62. Data visualization screens.....	70
Figure 63. Data acquisition screens.....	71
Figure 64. File management screen.....	72
Figure 65. Settings section menu.....	73

LIST OF TABLES

Table 1. Production goal for 2030	16
Table 2. Corn yield classes.....	16
Table 3. Annual production surface and value from segments mid-high and high	17
Table 4. Jetson Nano Board's specifications	28
Table 5. Variables' correlation with yield	59
Table 6. Comparison between actual yield values and estimated yield values.....	61
Table 7. Prescription results compared to original data.....	66
Table 8. AgStat's functionalities compared to similar applications'	74

LIST OF EQUATIONS

Equation 1. Exponential model	33
Equation 2. Nitrogen rate model	37

Abstract

APPLICATION (SOFTWARE) AND ARTIFICIAL INTELLIGENCE ALGORITHMS
DEVELOPMENT FOR AUTOMATED PRESCRIPTION GENERATION IN
PRECISION AGRICULTURE

BY

ANDRÉS CADENA DÍAZ

MASTER OF SCIENCE IN PRODUCTION SYSTEMS ENGINEERING.

DR. SANTOS GABRIEL CAMPOS MAGAÑA - ADVISOR

SALTILLO, COAHUILA

JUNE 2022

Mexican farmers report that their main problematic is the high cost of crop-inputs and services. Additionally, with the recent increases in fertilizer prices, is necessary to optimize their usage. Precision Agriculture Technologies, and particularly Information Intensive Technologies (IITs) offer data driven solutions for farmers to optimize their crop-input usage.

However, the adoption rate of IITs for precision agriculture (soil mapping, statistical analysis, prescription generation) is low because IITs require the user to have specialized knowledge and skills. Unlike embodied knowledge technologies (EKTs), whose adoption rates are much higher (autosteering, section control, among others) since the operator does not need to be an expert to use them. Many of IITs' potential users have low information technology literacy, and to encourage them to use and benefit from software programs, easy-to-use applications are needed.

AgStat is an application that was designed to provide farmers, researchers, and advisors with an integral, user-friendly system to perform automated data processing. The application was developed in python language and incorporates field surveying, soil mapping, data visualization functionalities, and a fully automated variable-rate prescription generation (VRPG) algorithm for nitrogen application. The algorithms developed for prescription generation include Artificial Neural Networks (ANN) for corn seeding recommendations and an expert-provided fixed model for nitrogen rate recommendations in a silage cornfield. Unlike the nitrogen recommendation model, ANN resulting models were not viable for use since the training data lacked quality and quantity, and therefore were not integrated on the application.

AgStat's VRPG results in a ready-to-use ESRI shapefile that can be exported to an external USB drive, which can be used to transfer the file to a variable-rate machine in the field. AgStat provides the insights for decision-making of an IIT without requiring the user to be a specialized geographic information systems operator or an expert agronomic advisor.

Keywords: User-Friendly application, Spatial variability mapping, Agronomic prescription, Precision agriculture, GIS.

1 INTRODUCTION

Precision Agriculture (PA) or site-specific management involves data analytics to address the spatial and temporal variability within agricultural plots to improve yields, reduce costs, and mitigate the farm's environmental footprint. The International Society of Precision Agriculture (ISPA) defines it as a "Management strategy that gathers, processes and analyzes temporal, spatial and individual data and combines it with other information to support management decisions according to estimated variability for improved resource use efficiency, productivity, quality, profitability and sustainability of agricultural production." Specialized equipment and tools have been developed to help farmers to make data-driven decisions in every site of their fields, automate labors and optimize their production. However, data analytics tools, also called "information intensive technologies" (IIT) for PA require specialized knowledge and skills to operate. Farmers, and even agronomic advisors, often lack those skills, which has resulted in a limited adoption of this technologies. This work aims to create a user-friendly application that automates data processing, encouraging the adoption of IITs by farmers and agronomic advisors. This chapter will discuss the background and context of this thesis, followed by the research problem, aims and objectives, significance and finally, the limitations.

The concept of PA as we know it emerged in the 1980's decade and was totally focused on the spatial variability involved in crop production. With the years, new variability components and other agricultural sectors were related to the concept (Gili et al., 2018). Some of the technologies on which PA relies are Global Positioning Systems (GPS), Geographic Information Systems (GIS), and variable-rate equipment. Combining the mentioned technologies, farmers can manage their operation dividing each field into subplots according to their characteristics. The variable rate machinery then can apply a different amount of crop-input (seed, fertilizer, herbicide, among others) based on the productivity potential of each subplot in the field, saving costs and improving yields. The file that instructs the variable-rate equipment how much crop-input to apply and where to apply it is called an agronomic prescription. To create adequate agronomic prescriptions, the user needs proper agronomic knowledge, geostatistical analysis skills, and strong

information-technology proficiency. Erickson et al. (2018) reported the difficulties farmers and PA agencies in the United States face to find personnel with the previously mentioned skills, which has resulted in a low adoption rate for IITs. An effective solution to this problem is to create user-friendly applications (Demiris et al., 2004). This solution proposal has taken the form of applications created specially to facilitate the duties of farmers (Rahmayanti et al., 2020; Songsupakit et al., 2019; Ali et al., 2018). In the context of PA, applications have been developed to automatically process data and delineate subplots within a field, also called management zones (MZ). Paccioretti (2020) programmed an application that automatically generates MZ by aggregating data with k-means clustering processing. The application created by Alborno et al. (2018), additionally to delineating MZ, generates a prescription file, but requires the user to enter the rates' values manually. These applications represent great advances in easy-to-use IITs for PA users, but none of them fully automates the agronomic prescription process, requiring the user to have the pertinent knowledge or access to a specialist to decide the crop-input rates.

This document describes the creation of the AgStat application which fully automates the prescription generation process. The literature review explores PA concepts, terminology, and previous works in software development for PA. In the Materials, Equipment, and Methods section, the source data is presented which includes three grain corn producing cornfields from Argentina and one silage corn field from Mexico, which were the data available at the time of this project's development. Additionally, the software packages used and the steps taken into the design of the application are discussed. The results and discussion section presents performance data, maps obtained and the resulting functionalities of the application. Development decisions based on model performance are discussed as well in the aforementioned section. Finally, the conclusions, references, and annexes are presented.

1.1 Contribution

The present work will contribute to the adoption of IITs by providing an easy-to-use tool "AgStat" to farmers, researchers, and agronomic advisors. The fact that the AgStat user does not need to be a skilled specialist to generate ready-to-use prescriptions can lead to

an increase in the adoption of IITs. With more farmers implementing PA strategies in their operations, the elevated fertilizer prices (Beghin & Nogueira, 2021) can be addressed to maintain productivity and profitability.

1.2 Objectives

Given the lack of IITs that fully automate the agronomic prescription generation process, this work has the following aim:

1.2.1 Main objective

Develop a user-friendly application to automatically generate variable-rate prescriptions and agricultural data visualizations.

1.2.2 Specific objectives

1. Model corn yield variability with Artificial Neural Networks and regression analysis
2. Use Artificial Neural Networks to prescribe variable seeding rates for corn
3. Develop an algorithm to create variable-rate nitrogen prescriptions
4. Design a user-friendly interface for an embedded application
5. Design and program application's functionalities
6. Deploy application into a Jetson Nano Board

1.3 Hypothesis

It is possible to fully automate an agronomic prescription generation process with fixed models and artificial neural networks via a portable graphic application

2 LITERATURE REVIEW

2.1 Background

2.1.1 Problem: Crop-input cost and ecological footprint

Since the beginning of agriculture, humanity has developed technologies to improve its productivity and efficiency (Khana & Kaur, 2019; Bhakta et al., 2019; Isioye, 2013). The agricultural mechanization revolution in the 19th century and the green revolution in the 20th century resulted in increased productivity never seen before (Van Zanden, 1991). As a result, a minimum portion of individuals began to have the possibility of producing food for the rest of the population. The use of machinery, improved seeds, and agrochemicals such as fertilizers, herbicides, and pesticides have notably increased the competitiveness of agricultural companies (Evenson & Gollin, 2003; Thompson & Blank, 2000). However, the methodology that has been used and continues to be used by most Mexican companies with agricultural machinery is not sustainable in the long term. Excessive applications of agrochemicals generate leaching (Bergström et al., 2005), which results in contamination of soils and aquifers.

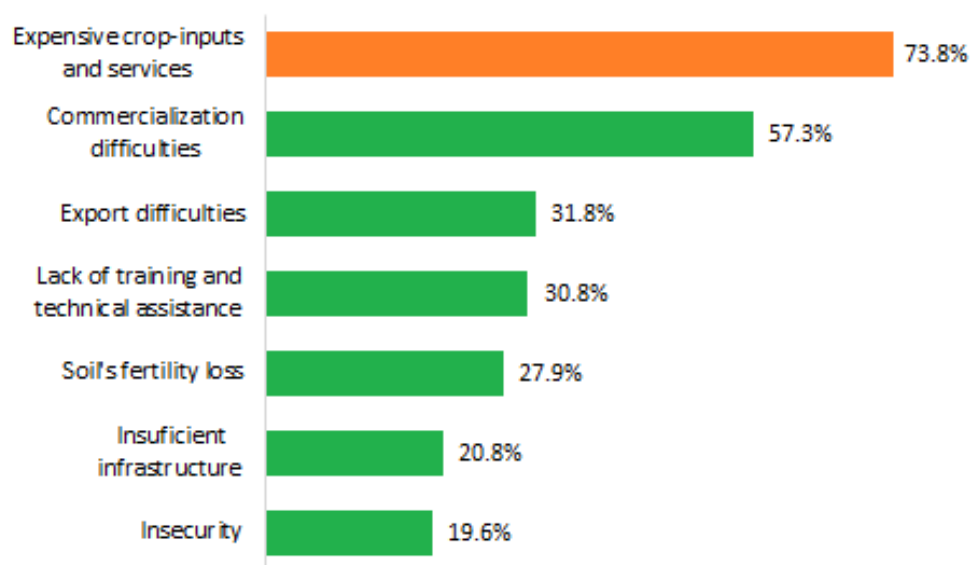


Figure 1. Mexican farmers' main problematics. Most farmers reported trouble with high crop-input and services costs. (from INEGI, 2020).

Additionally, the excessive use of agrochemicals is reflected in the production costs of agricultural companies, reducing their profitability, even more so with the rise in fertilizer prices (Beghin and Nogueira, 2021). According to INEGI (2020) expensive crop-inputs is the Mexican farmers' main issue (Figure 1).

2.1.2 Precision agriculture

Also called "site-specific management", this strategy optimizes crop-input applications and increases a farm's profitability by addressing the continuous variability in the soil, topography, and climate.

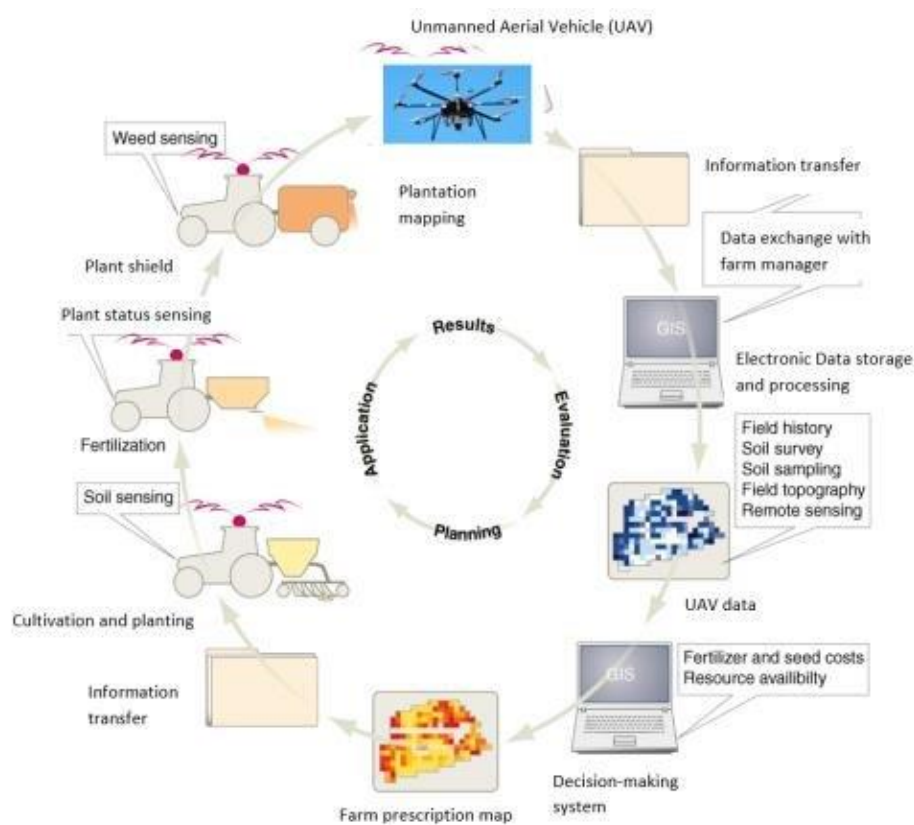


Figure 2. Information process flow of Precision Agriculture. Includes data collection from the crop, soil, weather, etc. The information is processed, and site-specific recommendations are generated and applied, then data collection starts again. (from Abdullahi & Zubair, 2017)

Despite the innovative appearances of PA practices, their same principles were used before the agrarian revolution by small-scale farmers. Onyango et al. (2021) point out that

in the past, sub-Saharan Africa's small-scale farmers planned their management strategies addressing the soil condition's variability of their plots to optimize their resources.

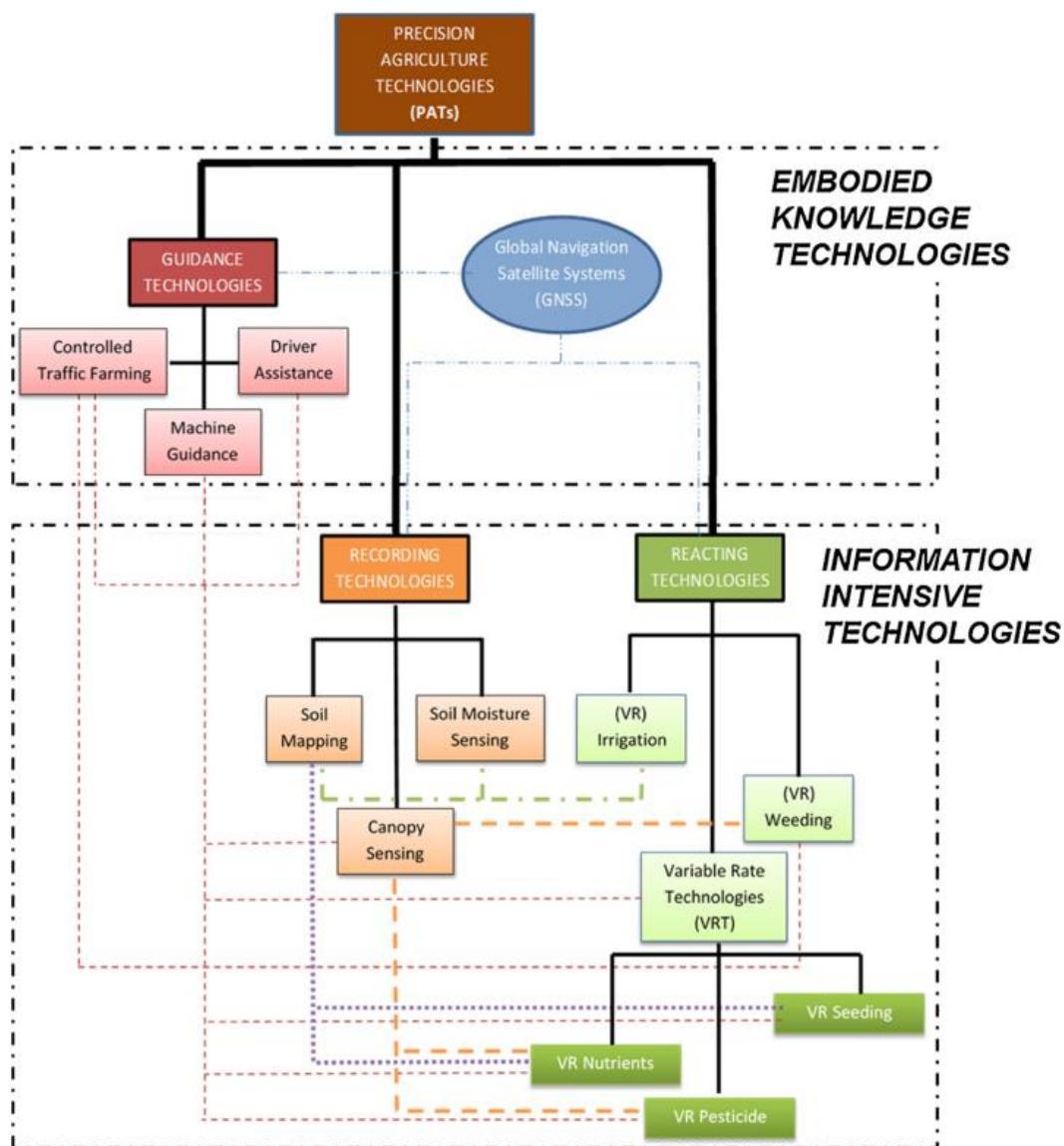


Figure 3. Hierarchy of Precision Agriculture Technologies (from Barnes et al., 2019)

These practices were largely lost since the introduction of inorganic fertilizers generalized agronomic recommendations, and the introduction of large-scale mechanized agricultural production. In recent decades, economic and environmental factors have encouraged farmers to optimize their operations by addressing the spatial and temporal variability in their fields. Since the 1990's, precision agriculture technologies (PATs) such automated

guidance, variable-rate technology, yield mapping, unmanned aerial vehicles and multispectral sensing have gained popularity (DeLay et al., 2022).

PATs rely on data processing (Figure 2) to help farmers optimize their resources and increase their productivity (Bhakta et al., 2019). Barnes et al. (2019) cited numerous studies that have classified PATs into two major categories (Figure 3) based on the level of interaction and knowledge required to operate them: Embodied Knowledge Technologies (EKTs) and Information Intensive Technologies (IITs). Autosteering and section control are good examples of EKTs. They have been proved to be adopted at a considerably higher rate than non-automated technology since, once installed, they do not need specialized skills or technical knowledge to operate. On the other hand, IITs like data geoprocessing and soil characterization, need a higher level of knowledge and human capabilities to benefit from them (Erickson et al., 2018).

2.1.2.1 Variable-rate technology

Most farmers in Mexico intend to apply crop production inputs uniformly to their fields and some only adjust rates between fields, but in reality the seeding rate varies continuously due to variations in speed, seed size, among other factors.. This management strategy does not maximize the crop-input use efficiency nor profitability. The previously mentioned premise leads to variable-rate technologies (VRT). Sawyer (1994) provides five assumptions needed to conceptualize VRT:

- 1) Important factors that affect crop yield vary spatially within fields
- 2) Spatial variability does influence crop yield
- 3) Spatial variability can be identified, quantified, and mapped
- 4) Precise crop response models are available to determine appropriate variable input rates

Implementing VRT in an agricultural operation is costly, the money invested in equipment, software, and trained personnel to carry out the pertinent analysis and applications needs to be surpassed by the improvements in crop yield or savings in crop inputs to make this technology viable. The value of PA needs to be adequately quantified and demonstrated to promote adoption between farmers. Figure 4 shows an example of a

VRT implementation cycle, which starts with crop monitoring via a drone equipped with a multispectral camera. The imagery captured by the drone can be used to extract a Normalized Difference Vegetation Index map (NDVI, Rouse et al., 1974) which strongly correlates with the plant vigor.

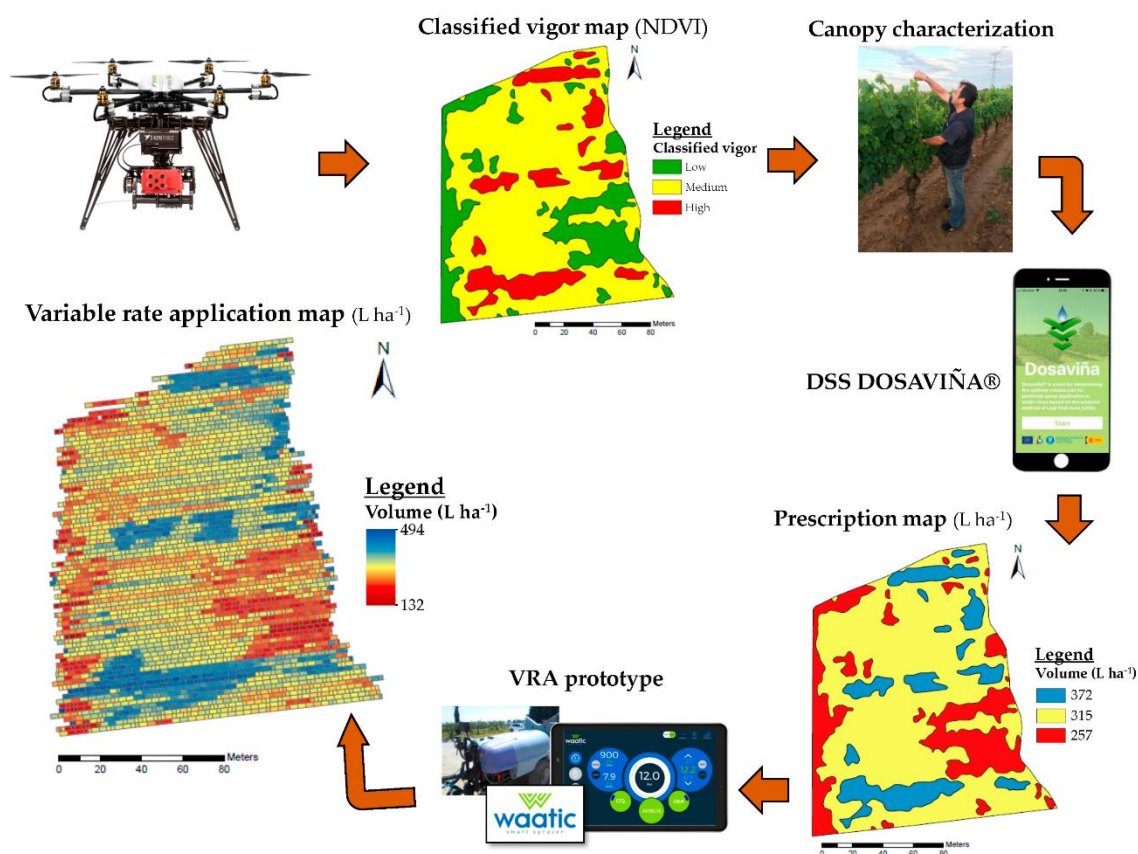


Figure 4. Overview of variable rate herbicide implementation in a vineyard. The classified NDVI map zones are used to apply different herbicide rates. (from Campos et al., 2020)

The NDVI map needs to be classified (Figure 4, Classified vigor map) to form MZ which then are used to assign herbicide rates (Figure 4, Prescription map) according to the plant vigor class and the pertinent agronomic criteria. In this case, the canopy of the plants in each zone measured to obtain the Leaf Wall Area (LWA), and then the rates were prescribed by the decision support system “Dosaviña ®” based on the LWA parameter. Once the prescription is ready, the herbicide was applied with variable-rate equipment (Figure 4, VRA prototype). The variable-rate application map of Figure 4 shows the actual

rates applied to the vineyard by the sprayer. The application, despite following generally the management zones, is not perfect because the equipment cannot change its application rate instantaneously when changing management zone. This illustrates the need for management zones with smooth changes between them to obtain a greater fidelity in the crop-input application. Once applied, the effect of the crop-input in the plants can be evaluated again with the drone, and the cycle starts one more time.

2.1.2.1.1 Prescription maps

The prescription map's fundamental components are the variable management zones and the corresponding crop-input rates. In other words, a prescription map indicates the machinery how much and where to apply a crop-input (seed, fertilizer, herbicide, among others). The management zones in the previously mentioned example were derived from NDVI data, but there is no correct methodology to delineate them. A common conclusion among publications regarding management zone delineation is that the methodology regarded as best in that specific study might not be optimal for other fields, crops or time frames. Management zones can be derived from a single variable i.e. biomass (Breunig et al., 2020), apparent electrical conductivity (Millan et al., 2019), among others. Some authors have delineated management zones using more than one variable by forming spatial clusters (Ohana-Levi et al., 2021).

A prescription file normally has a shapefile format (ESRI, 1998). A shapefile is formed by multiple files with the same name, but different extension. Although a shapefile can be composed of 10+ files, only three of them are mandatory 'SHP', 'DBF' and 'SHX' files. The SHP file contains the geometry of its features, in prescription maps those features are polygons corresponding to the management zones. DBF files are tables that store the information of the features, in precision agriculture context, the information are the crop-input rates. SHX files stores the index of the polygons and functions as a link between the geometry (SHP) and the database information (DBF) files.

2.1.2.2 Process automation

2.1.2.2.1 Precision agriculture apps

Many potential PATs users lack an information technology affinity. With the advent increasingly complex analytics systems for agriculture, automated data processing is

needed to promote the adoption of PATs. Along these lines, SAGARPA and CONACYT (2014) established mobile application development for automated data-processing, seeding, and fertilization as priority research lines in Mexico. Demiris et al. (2004) found that a user-friendly interface (UI) can encourage users that lack information technology proficiency to use and benefit from software applications. UI and user experience (UX) design significantly impact the adoption of technology, satisfaction, and user loyalty (Kim & Mc. Fadden, 2020; Nugraha & Dhewanto, 2019). Darejeh & Singh (2013) compiled UI/UX solutions for users with low computer literacy: clear and straightforward navigation paths, reduce the number of features available at any given time, avoid the use of complicated terminology, use of similar functions for different jobs, available user's guide and help in the software, among others.

Some applications have been developed to address the need for user-friendly systems for farmers (Rahmayanti et al., 2020; Songsupakit et al., 2019; Ali et al., 2018). For example, Paccioretti et al. (2020) and Albornoz et al. (2018) designed and developed user-centered / easy-to-use applications for automatic management zone delineation via fuzzy K-means clustering, considering multiple data layers. The system developed by Albornoz et al. (2018) was programmed to generate an ESRI shapefile as the last step in its processing sequence. Nevertheless, the user manually enters the rate values for each management zone, so technical knowledge or access to an expert is required.

2.1.2.2.2 Artificial Neural Networks

Artificial Intelligence (AI) is found in numerous and varied aspects of contemporary life. Thanks to the access to Big Data and the processing power of modern computers, it has gone from a mere concept to a reality that is transforming organizations in many sectors. De Mauro et. al. (2016) proposed a formal definition of Big Data based on its essential features: "Big Data is the Information asset characterized by such a High Volume, Velocity and Variety to require specific Technology and Analytical Methods for its transformation into Value."

Artificial Intelligence, which encompasses concepts such as Machine Learning and Deep Learning, dates back to the 50s decade of the twentieth century. Until a few years ago, its use was limited to research laboratories (Arrabales, 2016). Thanks to Big Data, there has been a digital revolution in organizations and companies with diverse lines of business. In this revolution, there is substantial demand for intelligent systems capable of processing, analyzing, and interpreting the numerous data generated by millions of individuals thanks to social networks, online shopping, among others. Deep Learning (DL) is a subfield of Machine Learning (Janiesch et al., 2021), whose concept appeared in the doctoral thesis of Paul Werbos in 1974 and was rediscovered in 1980 by Geoffrey Hinton, who was devoted to the implementation of the human learning model applied in a machine (Smart Panel, 2019).

Among the systems that use DL are artificial neural networks (ANN): computer systems based on an analogy to biological neural networks. They consist of multiple layers formed of simple processors called "nodes" or "neurons" (Agatibivuc-Kustrin & Beresford, 2000) that are interconnected with each other and operate in parallel. Depending on the type of ANN, connections between layers are arranged in many forms. A common ANN type is "fully connected" (Figure 5) but not all ANNs are internally connected like that, in some cases, neighboring layers are not necessarily connected. Values passing through each neuron are multiplied by a coefficient or weight and summed at the output of that neuron. These weights affect the functioning of adjacent neurons.

Another type of ANN is the Convolutional Neural Network (CNN), whose functioning is similar to that of the visual cortex. Its hidden layers can extract and identify attributes (lines, curves, textures, among others.) or specific features (eyes, the silhouette of an animal, a plant species) in a two-dimensional matrix. They are widely used for image processing and classification (Ghosh et al., 2020). Recurrent Neural Networks (RNN) allow the creation of feedback loops between neurons, which allows the network to have memory, and it is because of this attribute that this type of learning is potent for the analysis of sequences such as text, audio, video, etc. (Calvo, 2018). Self-Organizing Maps (SOM) were invented to represent multidimensional numerical data in spaces of reduced dimensions, usually in 2 or 3. A common way to demonstrate their operation is by

arranging 3d vectors formed by the RGB components of the color in a two-dimensional space, where colors are grouped with others having similar values. Rulaningtyas et al. (2017) used a self-organizing map to classify secretions caused by tuberculosis based on the color of the secretions.

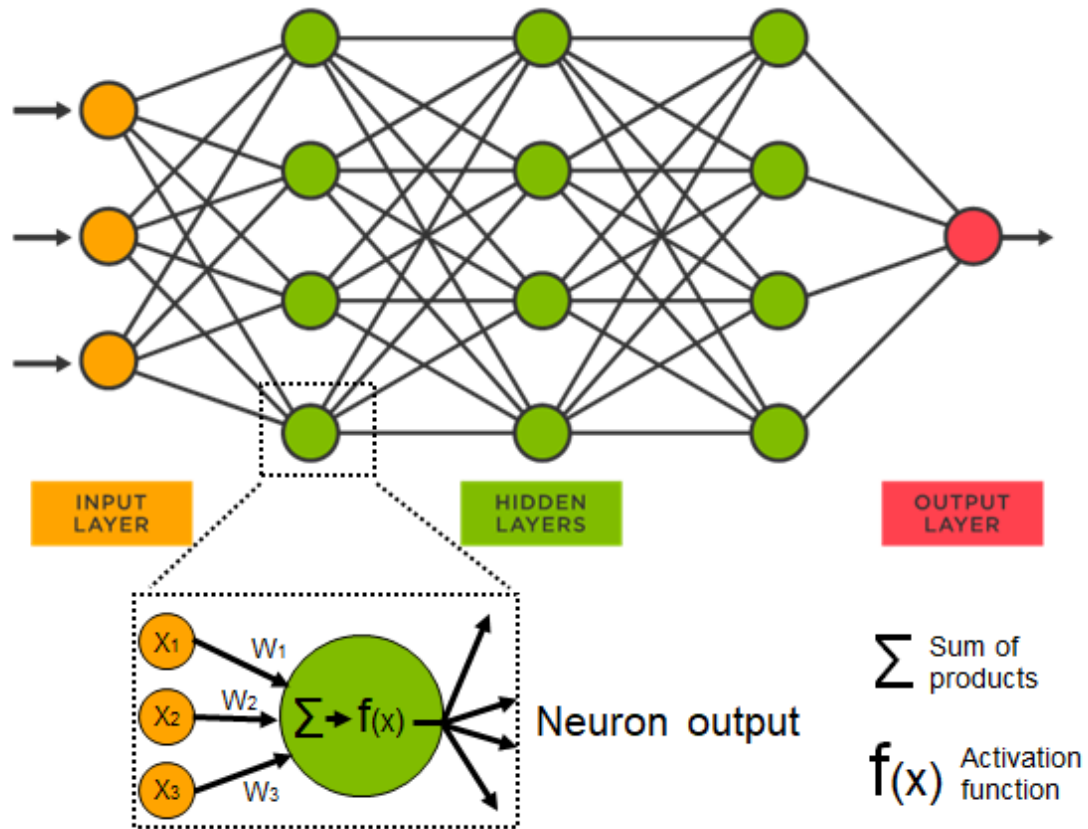


Figure 5. Generic example of a fully connected ANN structure. (adapted from Tibco, 2022)

For an ANN to work correctly, training must be performed through which the neural network identifies the patterns that relate the input variables with the output variable. Before training, data preprocessing is performed, dividing the data into a training section and a test section (80% and 20% are typical values) and then scaling the values for proper processing. Once these steps are completed, the network training is performed. A vector containing normalized and scaled values is introduced through the input layer. Each value is assigned to its corresponding input neuron or node. The ANN used for this example is a fully connected network, so each neuron in the hidden layers is connected to all the

nodes in the layer before and after the layer where it is located. Each of the input connections to the neurons has an assigned weight, i.e., in Figure 5, each neuron of the first hidden layer has three inputs and a weight corresponding to each of them. In comparison, the neurons of the other hidden layers and the output layer have four inputs. At the beginning of training, each weight is randomly assigned a value close to 0. The first vector enters the neural network through the input layer and is processed by each neuron. The processing that occurs in each neuron is as follows:

Each input variable x_n is multiplied by its assigned weight w_n (Figure 5) Then the node obtains the sum of the products of these variables with their weights. At the output of each neuron is an activation function that is applied to the sum of products and whose resulting value corresponds to the output of the neuron. The neuron's output will be propagated to the next layer of the neural network to be processed again. Figure 6 shows some examples of activation functions. The "x" axis corresponds to the possible values of the neuron's sum of products, and the "y" axis corresponds to the value at the neuron's output. A commonly used activation function for hidden layers is "ReLU" (Figure 6) with an output range from 0 to infinity. For the output node, a popular choice is "sigmoid" function since often the expected result is between 0 and 1.

Once all the neurons process the variables, the output layer generates the prediction result. Then, the prediction is compared with the actual value, and the prediction's error is computed. The error value is used to evaluate the cost function, which is sought to be minimized through the gradient descent method. This method evaluates the slope of the cost function for the prediction so that the system eventually reaches a prediction value that minimizes the error in the prediction (a prediction close enough to the actual value).

Is common for neural networks to use the batch learning model, which means that the above processes are repeated for n vectors or database entries where n is a parameter called "batch size." Each time batch processing is completed, the weights in the neurons are updated through a process called "backpropagation". When all batches of training data are processed, an epoch is completed. The user determines the number of epochs he/she considers appropriate so that the accuracy of the neural network tends to increase over time without stagnating since, after a certain number of epochs, the accuracy stops

increasing. When the neural network training finishes, the testing phase is executed, where the section of the database reserved for this purpose is introduced to the network. Predictions are made, and the results are compared with the actual values, obtaining a measurable error for the network.

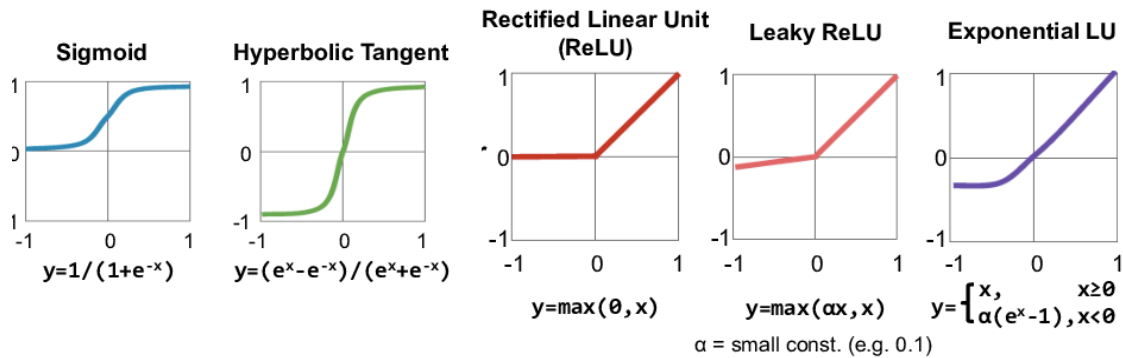


Figure 6. Activation functions. Each function has unique properties and behavior and are suitable for different applications (adapted from Sze et al., 2017)

After a successful testing phase is over, the ANN can be deployed or integrated into an external system, where it can help users to make data-driven decisions. DL is a potent and versatile tool that is transforming the way of analyzing data and understanding phenomena, particularly those of complex and multivariate nature, such as those related to agriculture (Kamilaris & Penafreta-Boldú, 2018).

2.2 Target market

2.2.1 Corn production in Mexico, a representative crop

Corn (*Zea mays*) is the cereal with the greatest production volume. Only in 2021, 1.2 metric tons of corn were harvested worldwide (Shahbandeh, 2022). Corn surpassed rice and wheat about ten years ago, becoming the most extensively cultivated cereal in the world (Garcia-Lara & Serna-Sandoval, 2019). This crop is an autochthonous Mexican species (CONABIO, 2006). The earliest paleoethnobotanical evidence of its domestication is located in the Mexican state of Tamaulipas and dates between from 6000 and 20000 years BC (Garcia-Lara & Serna-Sandoval, 2019). Despite the Mexican origins of corn and the fact that nearly a third of the national agricultural surface is destined to its

production, our country ranks 8th in the production of this commodity worldwide. Grain corn represents 89.1% of Mexican grain production. In 2020, Mexico produced 27.4 million metric tons of this cereal worth \$114,911 million MXN in 7.4 million hectares, being Sinaloa and Jalisco the top producing states (SIAP, 2022). One of the main reasons for which Mexico, despite being the original corn land, shared only 2.25% to global corn production in 2020 (Shahbandeh, 2021) is that nine out of ten Mexican farmers produce on a small-medium scale (FAO, 2018).

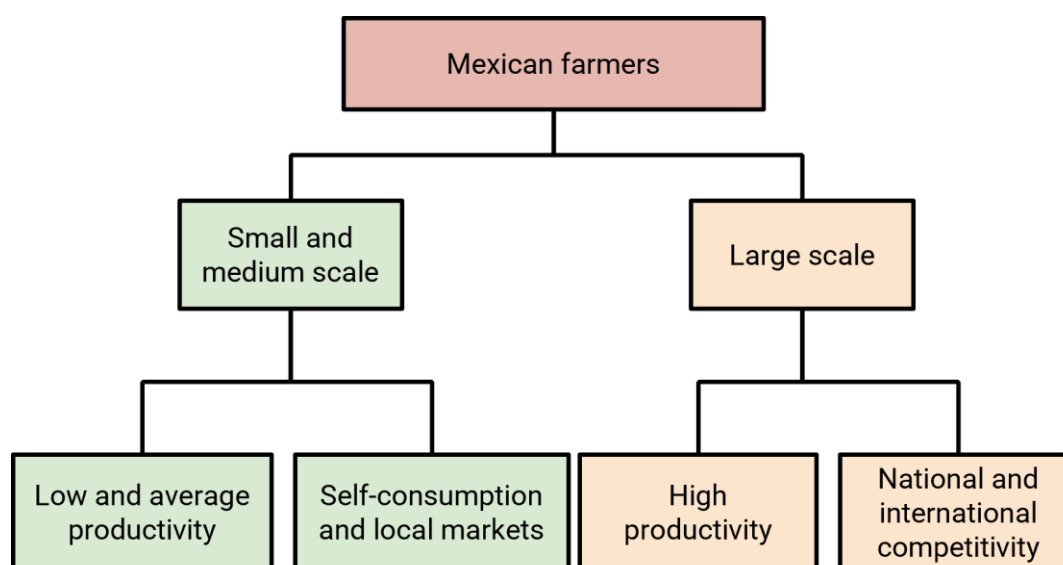


Figure 7. Mexican farmers' classification (with data from Diaz & Lozano, 2019)

The small-scale farmer majority in Mexico reflects also in the average grain corn yield, which was 3.8 tons per hectare in 2020 (SIAP, 2022) compared to large-scale farming in the United States: 11.2 tons per hectare in 2020, according to USDA (2022). Irrigation systems notoriously also affect Mexico's corn production. 77.3% of the surface destined for corn farming used a rainfed system, but only contributed to 47.5 of national corn production. A corn production deficit exists currently in Mexico, to compensate for it 15.9 million tons were imported on 2020, most of them was sweet (yellow) corn from the United States (SIAP, 2022). Addressing this issue, SAGARPA (2016) established production goals for 2030 (Table 1).

Table 1. Production goal for 2030 (from SAGARPA, 2016)

Crop	Production (thousands of tons)		Cumulative Growth
	2015	2030 Goal	2015-2030
Sweet corn	2,469.4	19,841.56	703.5%
White corn	22,224.64	26,376.19	18.7%

2.2.2 High yield corn production

The nature of this research project requires the potential users to have access to relatively new technologies. In Mexico, most tractors are 15 or more years old (Figure 8), only 27% have 10 or less years of use (INEGI, 2020). The use of modern machinery and technology are required to perform PA operations and are usually correlated with high yield. To identify farmers that can benefit from the product of this research project, Cadena & Escobar (2020) classified Mexican municipalities according to the average corn yield of their farmers (table 2).

Table 2. Corn yield classes (from Cadena & Escobar, 2020)

Class	Low	Low-Med	Medium	Med-High	High
Average yield (t/ha)	< 3	3 – 5	5 – 8	8 – 10	> 10

According to FAO (2018) large-scale farmers, despite being minority, are responsible for half of the national agricultural production in Mexico. Since yield and technology use are closely related, it is highly probable that farmers located in municipalities with average yield greater than 8 t/ha (mid-high and high yield segments) can benefit from the technology developed in this project. Figures 9 & 10 suggest that the average sweet corn farmer have access to better technology than the average white corn farmer. The value of this market is summarized in table 3. Municipalities highlighted in figure 11 correspond to the selected yield segments with high potential to benefit from this research project, almost all of them located in the states of Chihuahua and Sinaloa.

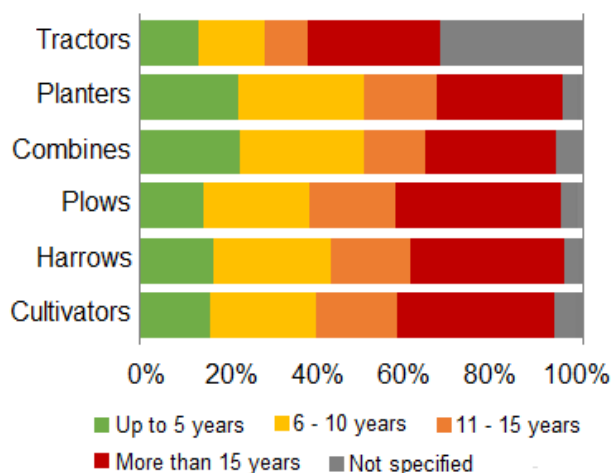


Figure 8. Average agricultural machinery age in Mexico 2019. Tractors appear to be not as old as their implements. (with data from INEGI, 2020)

Table 3. Annual production surface and value from segments mid-high and high (with data from SIAP and SIACON, 2020)

Crop	Surface (ha)	Value \$USD
White corn	750,000	1.5 billion
Sweet corn	170,000	307 million

A complementary market to corn farmers is sorghum farmers. According to data from SIAP (2021) the highest concentration of surface used to grow sorghum in Mexico is in the municipalities of San Fernando, Matamoros, and Rio Bravo (Figure 12). Only in 2020, 819450 hectares of sorghum were planted in the state of Tamaulipas. Despite being heavily farmed, sorghum production in Tamaulipas has relatively low yields (Figure 13). Precision Agriculture adoption programs exist in the region and this project can assist Tamaulipas' sorghum farmers to optimize their resources and be more competitive. Figure 2 illustrates more alternate markets in high yielding municipalities that produce with high-value crops. Most of this market are export crops produced in the Bajío region.

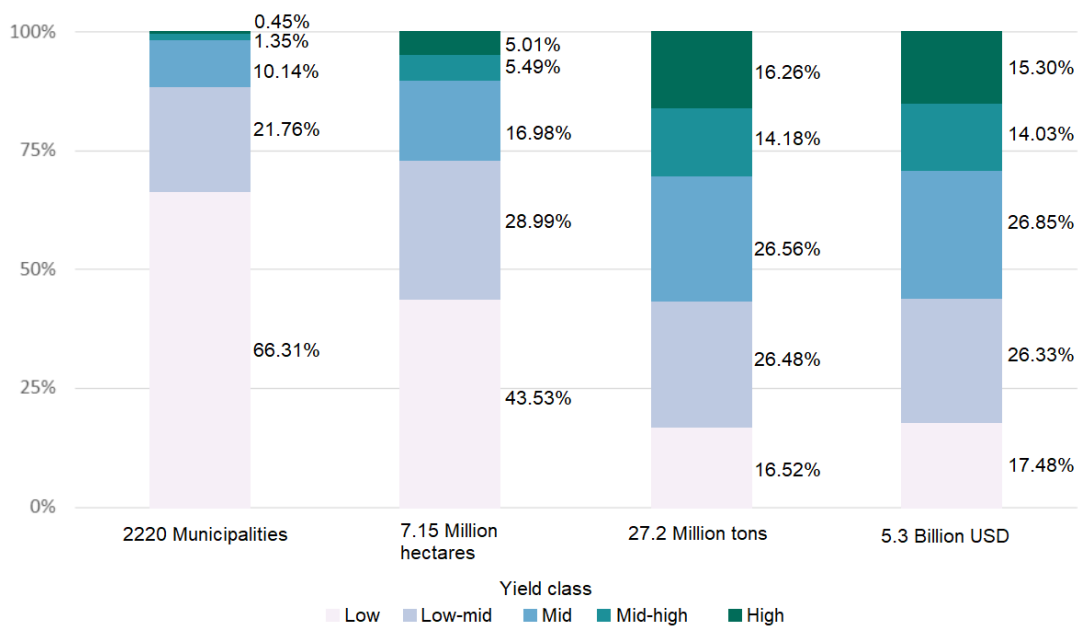


Figure 9. Mexico’s white corn production classes 2019. Top 2% of municipalities produce nearly 30% of grain corn. (from Cadena & Escobar, 2020)

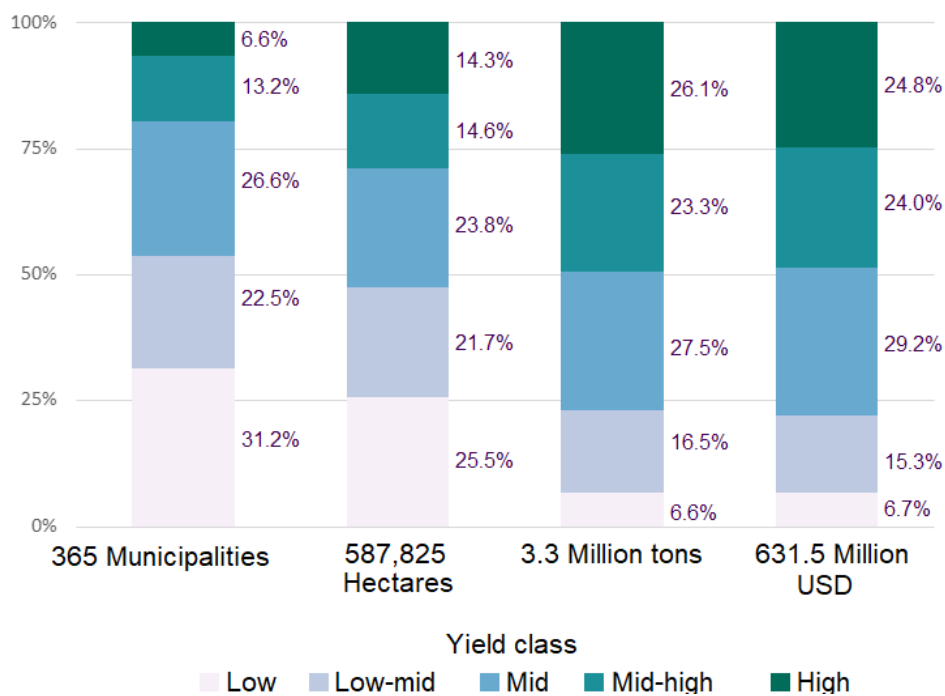


Figure10. Mexico’s sweet corn production classes 2019. Top 6.6% municipalities produce nearly 25% of sweet corn.

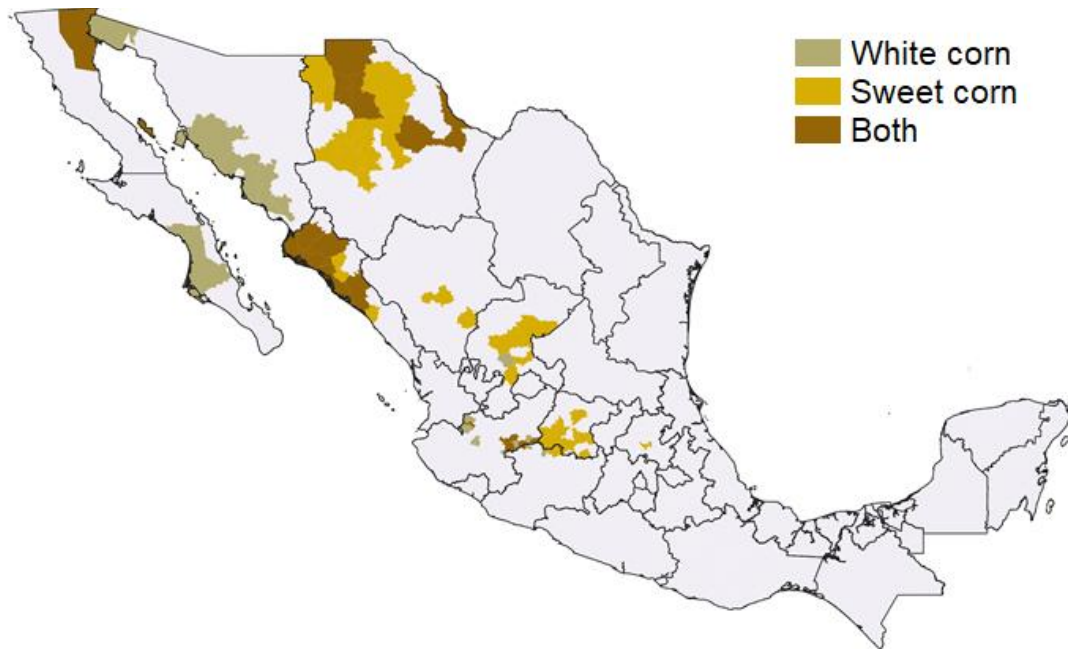


Figure 11. Target corn market location. Almost all corn market is located in Mexico's northwest (from Cadena & Escobar, 2020)

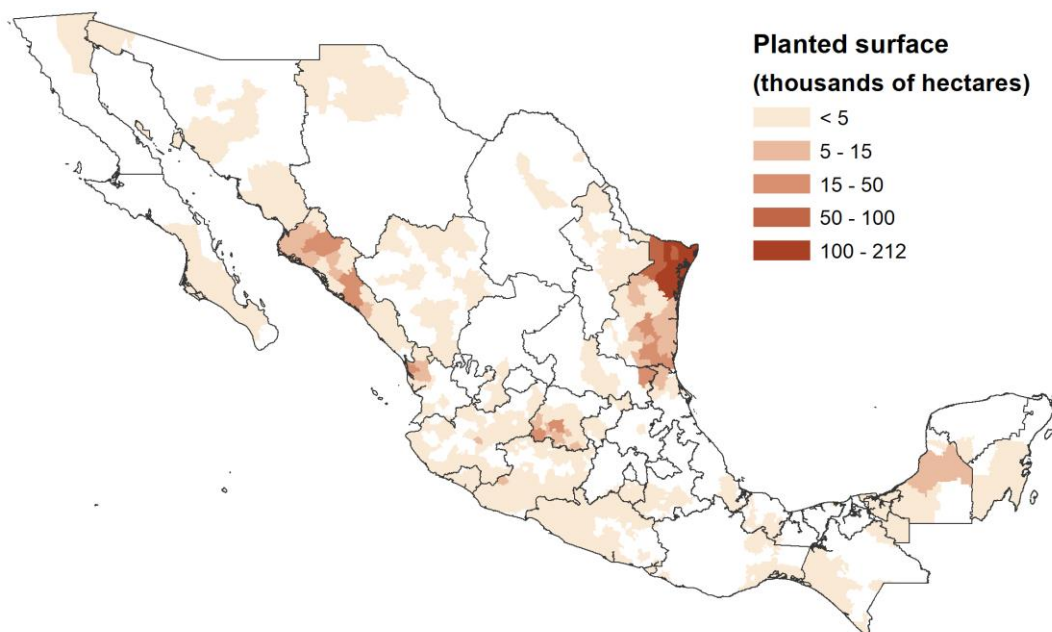


Figure 12. Planted sorghum surface per municipality 2020. Almost all concentrated in Tamaulipas state.

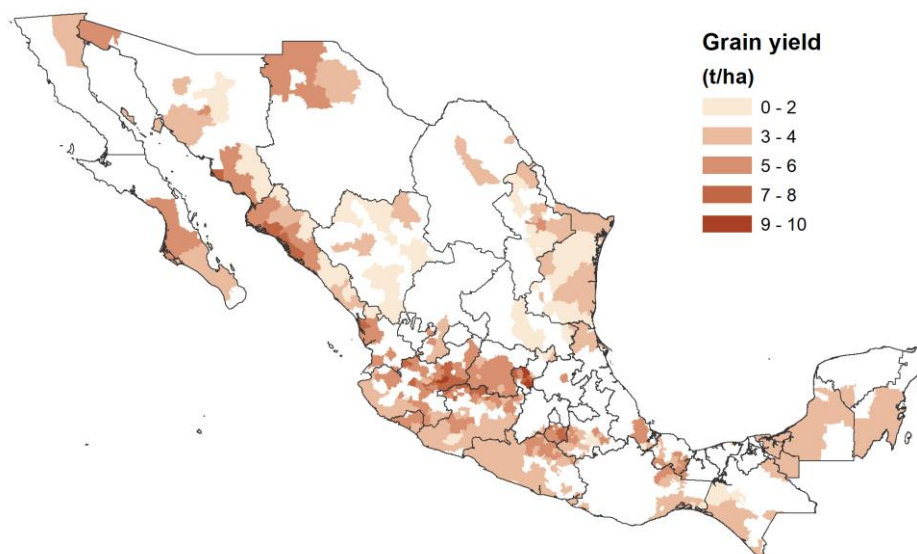


Figure 13. Sorghum yield per municipality 2020. Highest yields in Querétaro and Jalisco states.

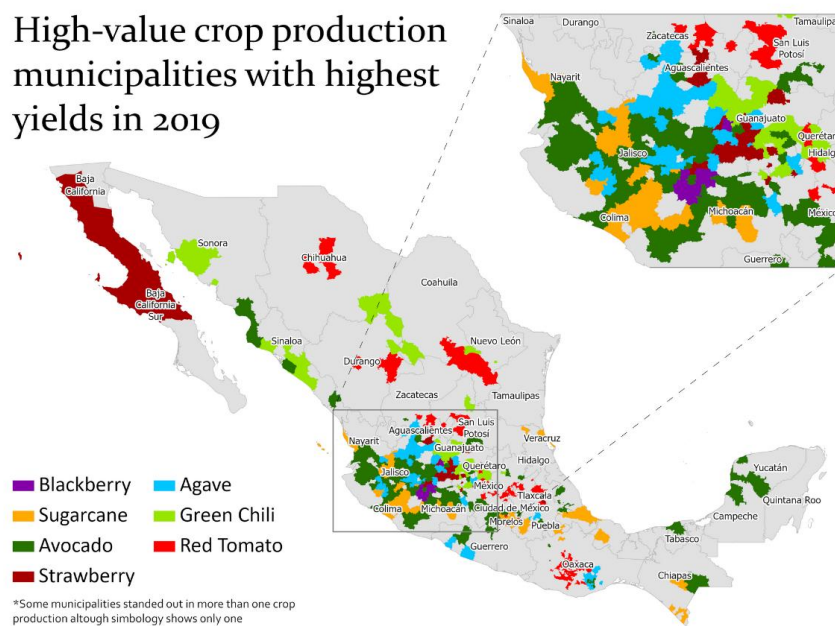


Figure 14. Alternate market location. Mostly consisting in export crops grown in the Bajío region. (from Cadena & Escobar, 2020)

3 MATERIALS, EQUIPMENT, AND METHODS

3.1 Materials

This research work used data from four cornfields, three of them are in Argentina (Figure 15) and are used to grow grain corn. The fourth field is from Mexico and is used to produce silage corn. The Argentinian fields were used for ANN analysis, and the Mexican field was used for regression analysis.

3.1.1 Fields A, B, and C

3.1.1.1 Field Imagery

The information available for the three fields included Apparent Electrical Conductivity (ECa), Altitude, Seed rate, and Yield. The data was acquired from Easy Agro (2019) as imagery containing maps (annex 1), which were georeferenced and digitized for analysis.

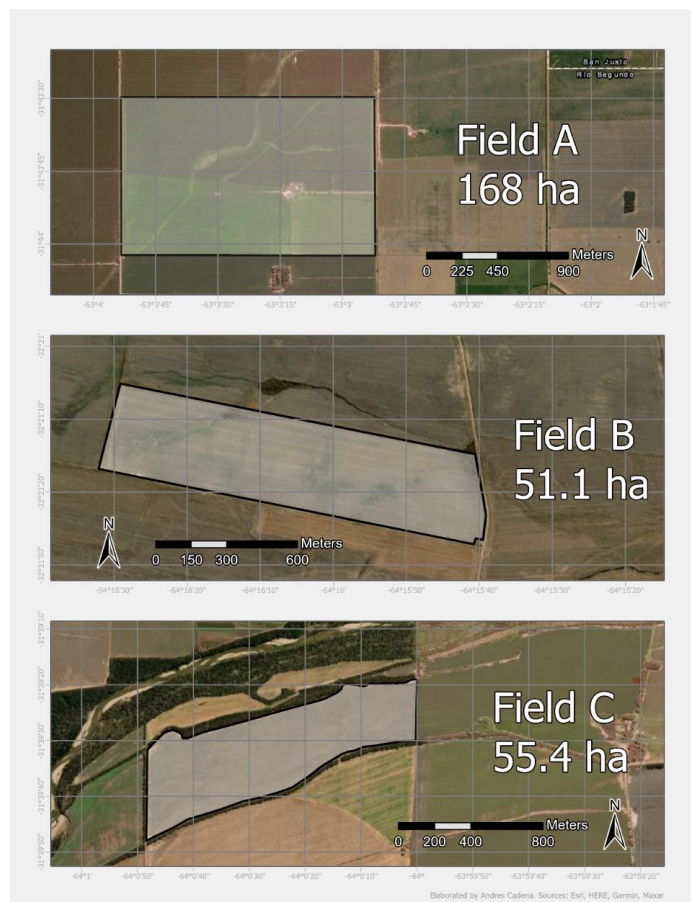


Figure 15. Location map (Fields A, B, and C). All located in Argentina.

3.1.2 Field D

3.1.2.1 Orthomosaics

For regression analysis, data from a center-pivot cornfield in Aguascalientes, Mexico was used (Figure 16). The silage corn produced in the field is used to feed livestock in the same farm. The field has a diameter of approximately 530 m and covers a surface of 21.1 ha. A dirt road within the field is used to access the center of the pivot.

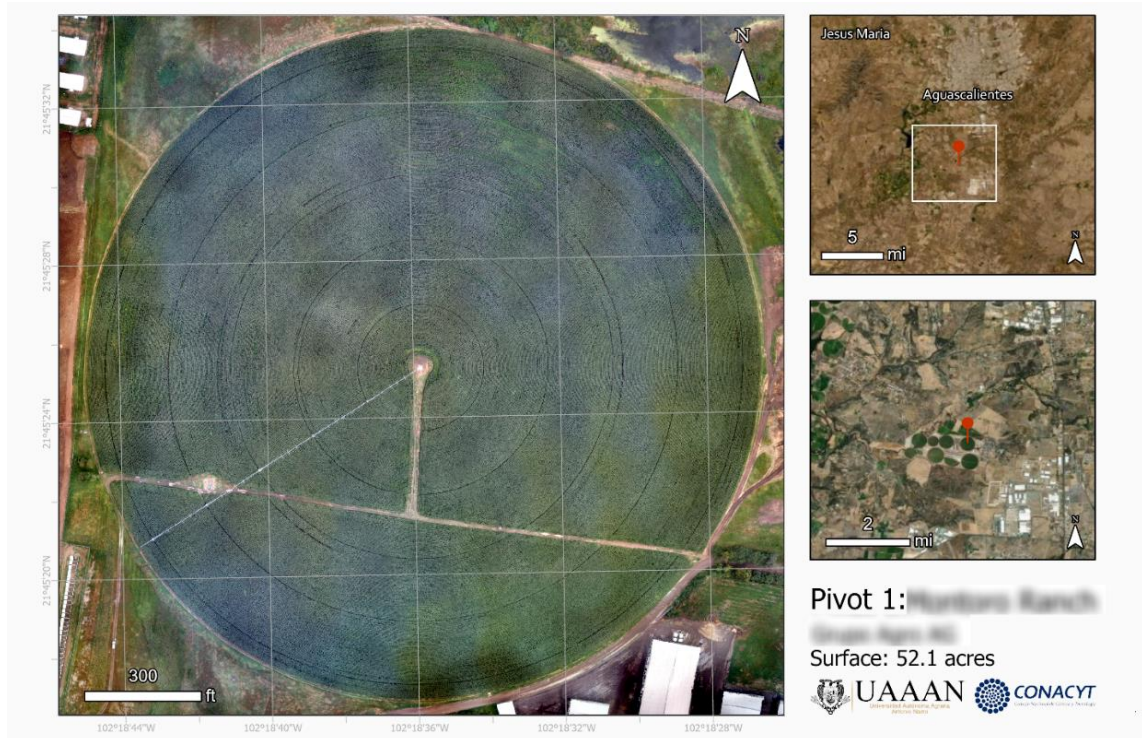


Figure 16. Location map (Field D). Located in Aguascalientes, Mexico.

The crop in this field was monitored via multispectral aerial orthophotos captured by an eBee drone (SenseFly, 2018). Orthophotos differ from ordinary photos since they are processed to correct lens distortion, camera tilt, perspective, and topographic relief (Dogget, n.d.), so they result in a perfectly straight-down view of the objects in the frame (Figure 17). To create map-quality images, orthophotos are combined into an orthomosaic, where additional geometric and color corrections are performed (ESRI, n.d.).



Figure 17. Raw photo (left) orthophoto (right). (Günay et al. 2007)

A NDVI orthomosaic from "pivot 1" field was created with Pix4DMapper software (Pix4D SA, 2022), combining the orthophotos captured by an eBee drone (SenseFly, 2018) with a parrot sequoia multispectral sensor (Parrot, 2018). The orthomosaic has a resolution of 10cm per pixel and was provided in Geo tiff format (Figure 18).

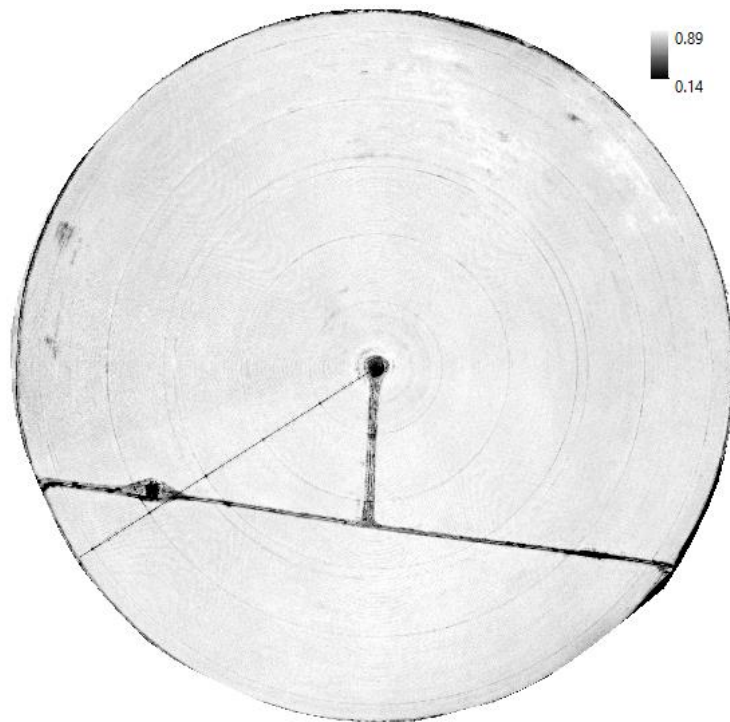


Figure 18. NDVI orthomosaic collected by eBee drone equipped with multispectral camera

3.1.2.2 Soil samples

To estimate the soil properties' spatial variability within the Mexican field, 25 samples were taken averaging 0.84 hectares per sample. The sampling locations were selected in a random distribution within the field. Pixels that corresponded to the dirt road and the pivot structure were filtered out to preserve only the crop's data (Figure 19). The preprocessing steps will be discussed further in the methods section.

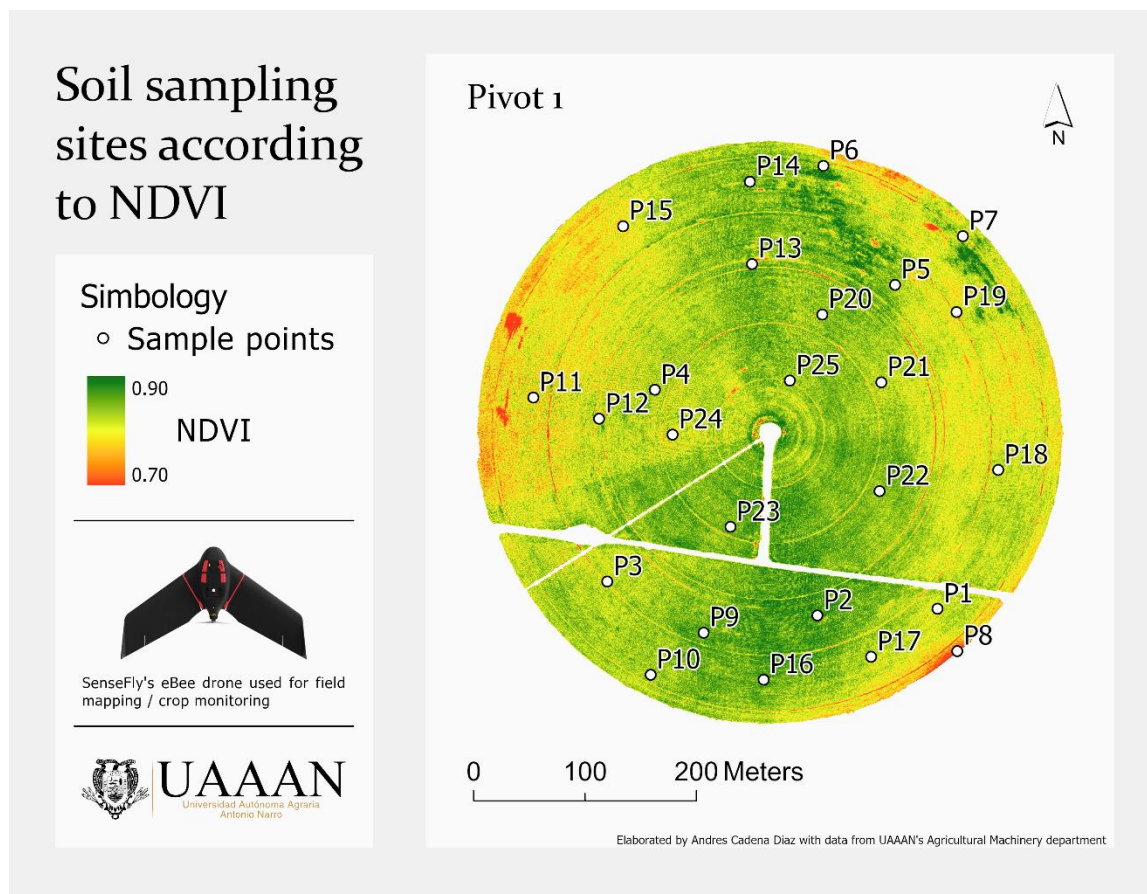


Figure 19. Soil sampling distribution

3.1.2.3 Yield samples

Yield is one of the most important parameters in agriculture, and generally the most important to farmers, since it is directly linked to their income. To assess the relationship between soil properties and yield, five samples were taken manually in the locations

shown in Figure 20. The samples were taken in locations near to the field boundaries because intense rain made impossible to reach into the field.

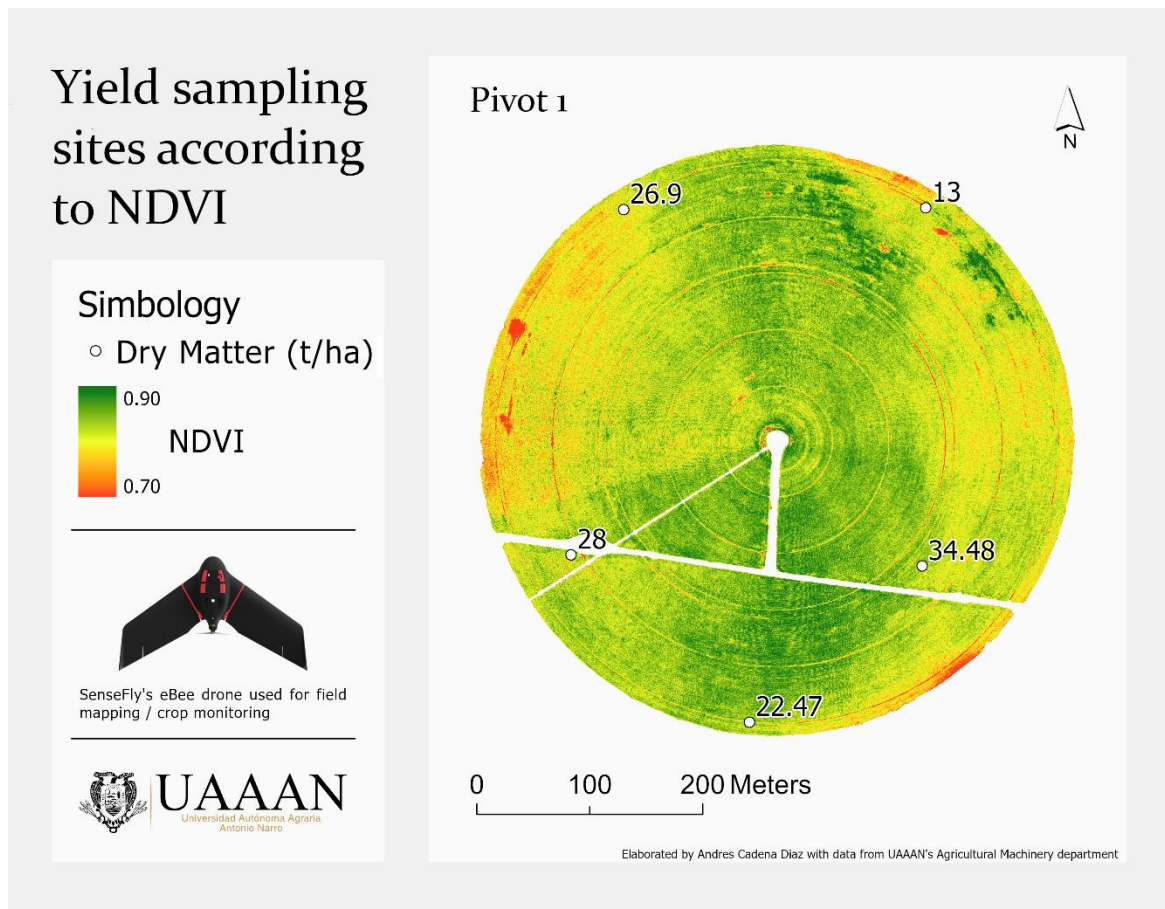


Figure 20. Yield samples' spatial distribution

3.2 Equipment

3.2.1 Software

3.2.1.1 Python

It is an interpreted programming language designed in 1991 by Guido van Rossum. Python is multi-paradigm and fully supports object-oriented programming and structured programming. Its philosophy highlights code readability and simplicity.

3.2.1.1.1 NumPy

Python library that supports high level mathematical functions and fast multi-dimension array computing. This work relied on NumPy to process big tridimensional databases in a fast and efficient way.

3.2.1.1.2 Pandas

Python Data Analysis “Pandas” is a software library for data analytics. Its “Data Frame” object supports fast manipulation, reshaping, indexing, merging, splitting, among other functions. Databases were stored and modified into Data Frame objects for this project.

3.2.1.1.3 GDAL

Geospatial Data Abstraction Library. Developed by the Open-Source Geospatial Foundation, supports vector and raster processing. Its python APIs were used to process tabular data into georeferenced raster data and vice versa.

3.2.1.1.4 Geopandas

Provides a high-level interface to process geospatial vector data, extends the pandas’ Data Frame object into Geo Data Frame to allow spatial processing and to store geometric information. Map plotting and geometric processing in this project was achieved with Geopandas.

3.2.1.1.5 PyQt5

Developed by Riverbank Computing, PyQt offers many GUI widgets to develop applications in an intuitive and fast manner. The user interface and basic functionality of the application developed in this project was created with QtDesigner and PyQt5.

3.2.1.1.6 Matplotlib

Plotting library that provides an object-oriented API to embed plots in applications (i.e. Qt). Geopandas’ plotting functionality depends on matplotlib. Scatterplots and histograms in the applications were produced with matplotlib as well.

3.2.1.1.7 Seaborn

Offers a high-level interface for plotting visually pleasing and informative plots with matplotlib objects. Regression plots on this work were produced with seaborn.

3.2.1.2 ArcMap

Geographic Information System developed by ESRI, its functionalities include viewing, editing, creating, and analyzing geospatial data. Its interface allows the user to symbolize features and create professional map layouts. ArcMap was used to process the NDVI orthomosaic, process data before developing the application, and to create the maps presented in this document.

3.2.2 Hardware

3.2.2.1 Jetson Nano Board

This product is a small computer (Figure 21) that can run multiple ANNs for a variety of applications. A decisive feature for using this piece of hardware is its 128 core Graphics Processing Unit (GPU), since most small computer units in the market do not have one. ANN's computations run in parallel. GPUs specialize in this type of computations and are ideal for running ANNs (the processing tasks are divided among the GPU's cores). Additionally, this portable board has the technical specifications mentioned in Table 4.



Figure 21. Jetson Nano Board (From Nvidia, 2022)

Table 4. Jetson Nano Board's specifications (from NVIDIA, 2022)

GPU	128-core Maxwell
CPU	Quad-core ARM A57 @ 1.43 GHz
AI Performance	472 GFLOPs
Memory	4 GB 64-bit LPDDR4 25.6 GB/s
Storage	microSD (not included)
Video Encode	4K @ 30 4x 1080p @ 30 9x 720p @ 30 (H.264/H.265)
Video Decode	4K @ 60 2x 4K @ 30 8x 1080p @ 30 18x 720p @ 30 (H.264/H.265)
Camera	2x MIPI CSI-2 DPHY lanes
Connectivity	Gigabit Ethernet, M.2 Key E
Display	HDMI and display port
USB	4x USB 3.0, USB 2.0 Micro-B
Others	GPIO, I ² C, I ² S, SPI, UART
Mechanical	69 mm x 45 mm, 260-pin edge connector

3.2.2.2 Generic Display

This display was required for the user to interact with the application via a capacitive tactile interface, which comes with high touch sensibility and good visibility in sunlight. The screen has a resolution of 800 x 480 pixels and measures 7". The size, resolution and visibility of this screen make it practical for use in the field. In future works, displays with

a resistive tactile interface may be evaluated since they have a better resistance to dust and water (Li, 2018).

3.3 Methods

3.3.1 Data preparation

3.3.1.1 Fields A, B, and C

The map images were imported into ArcMap, and georeferenced via control points. Then, Polygon feature layers were manually digitized following the shapes in the maps (Figure 22). Each polygon corresponds to a range of values, but only one value can be stored as the identifier value in its attribute table. To solve this problem, the lower limit of each range was assigned to the “value” field of the attribute table for plotting purposes. The locations where the upper limit value of the range was necessary were edited manually. The editing process is discussed further in this chapter. The ranges shown in the legend correspond to a string type field in the attribute table of the polygon features.

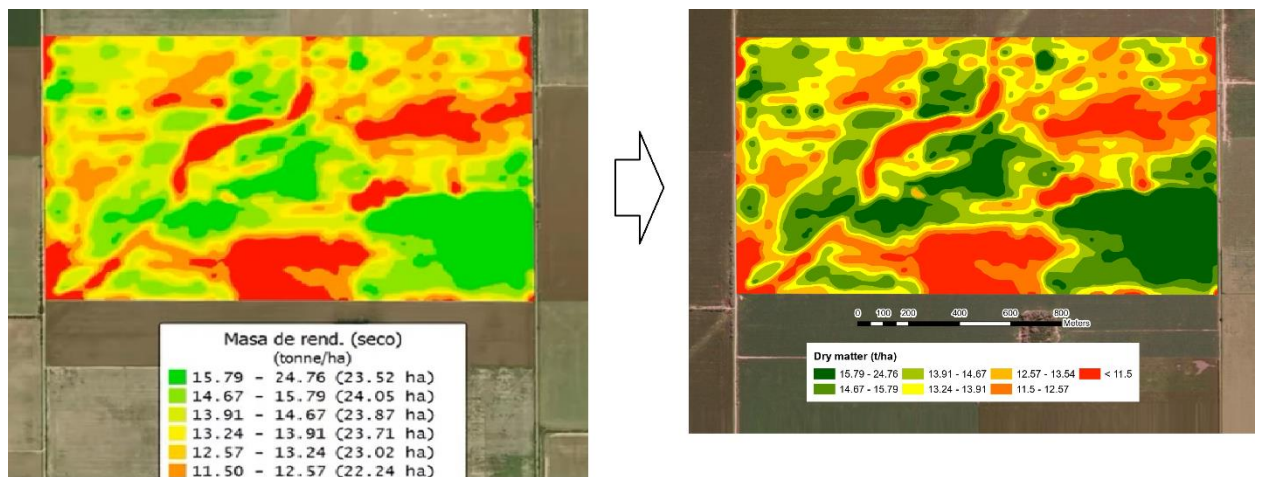


Figure 22. Original image (left), Manually digitized polygon features (right)

To generate the databases, an interpolated surface was needed. To achieve this, the polygons were converted to point features following the procedure shown in Figure 23.

Since the polygon features share vertices with their neighboring range polygons, half of the polygon ranges were deleted following an alternate pattern (i.e. if the map had five ranges, the second and fourth ranges were deleted). This methodology preserves all the vertices needed to create contours without having duplicate points in the same location due to neighboring polygons sharing same vertices. After that, the polygon vertices were converted to point features (Figure 23. c)). The value field of the points was edited when necessary to preserve the spatial gradient of the variable (Figure 23. d). The points linked with lines in the pane d) of Figure 23 share the same value. A good manner to think of this operation is to think in the linked points as elevation curves. Finally, additional points were created inside the polygons that correspond to the extreme ranges. The extra points located inside the space that corresponds to the highest range polygons were assigned the upper limit of that range. On the other hand, the extra points located inside the space that corresponds with the lowest range polygons were assigned to the lower limit of that range.

Other points were added in the boundaries of the field to cover the whole area, like the point shown in the upper-right corner of pane e) in Figure 23. The value assigned to the last-mentioned points was the value of the nearest “elevation curve”. A good example of the followed procedure’s results is shown in Figure 24.

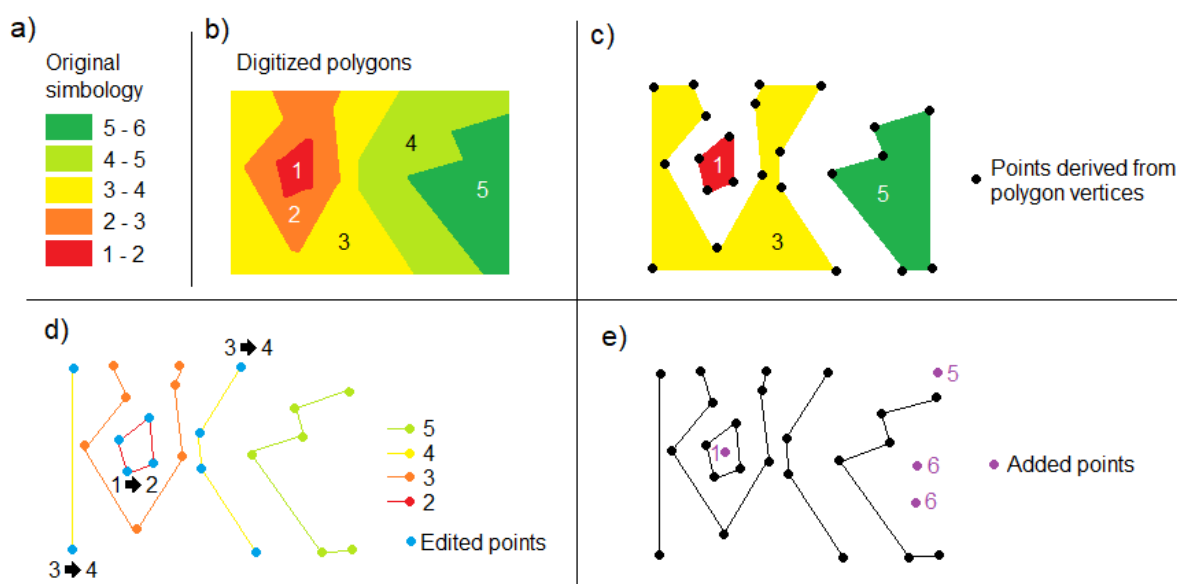


Figure 23. Procedure followed to convert digitized polygons to point features

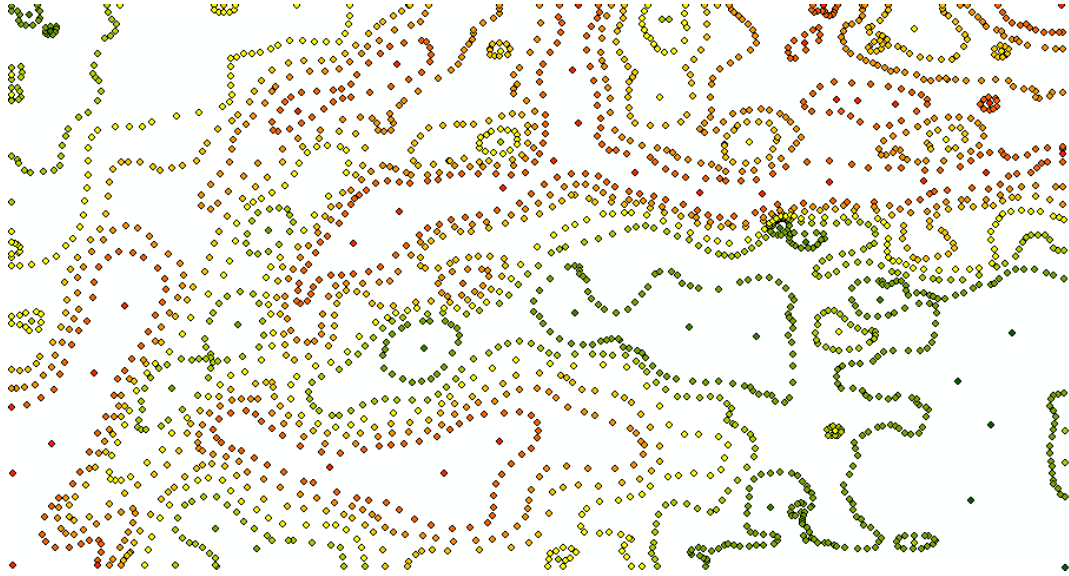


Figure 24. Point features ready to be interpolated. Color indicates the value of the parameter of interest.

The point values were then interpolated using the Kriging tool in ArcMap. An important exception to this procedure is the seeding rate data, since it was applied in discrete intervals, the polygon features were converted to raster without intermediate steps.

The interpolated datapoints from each field were normalized with min-max methodology, and then concatenated into one large database with +100,000 entries and stored as a “.csv” file. Normalization ensures a common scale is used across all data entries and fields, while preserving the relative differences between values. The X and Y coordinates were normalized after concatenating the three field’s data to preserve the relative field sizes (Figure 25). Coordinates, Altitude, ECa, seeding rate, and yield were the variables registered for each observation. Additionally, the field variable was encoded into three columns, one for each field (i.e., for data observations that corresponded to field “A”, the column “Field A” was assigned with a value of 1, and the columns “Field B” and “Field C” were assigned a value of 0). This encoding processing is necessary because machine learning models only can process numerical data, and since the “Field” value is categorical, it must be converted to a format that the model can work with. After encoding, the three field columns were added to the database.

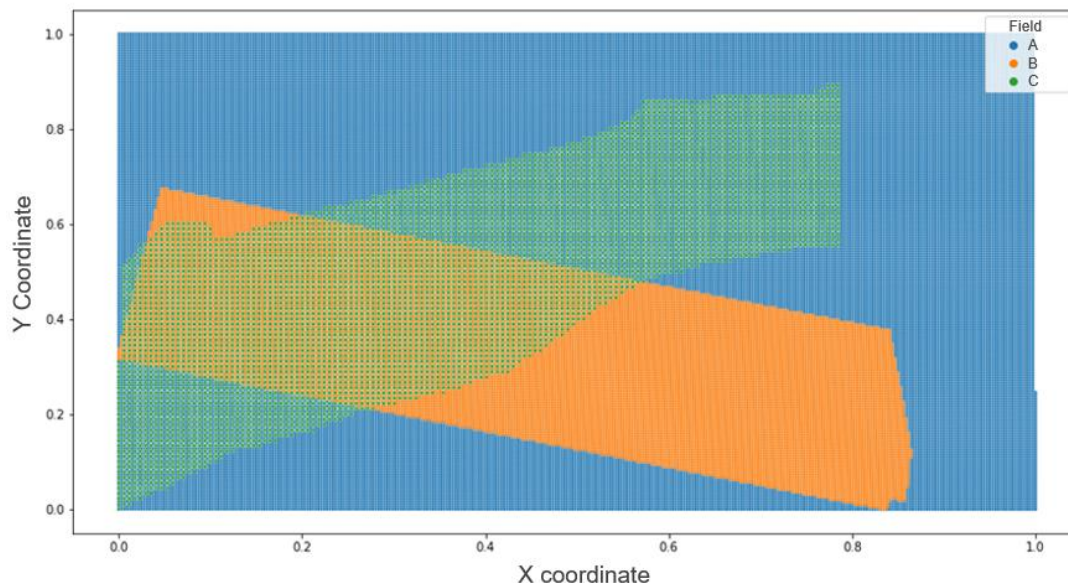


Figure 25. Normalized X, Y coordinates. The proportions between the sizes of the fields are preserved in this way.

3.3.1.2 Field D

The raw NDVI orthomosaic (Figure 18) was imported into ArcMap, the image covered the whole field, including areas that did not contain corn plants, like the dirt road in the middle of the field, and the metallic pivot structure. The NDVI values of interest were those of the corn plants, so a filter was applied with the “Con” tool to remove all the pixels with values below 0.7 from the raster. This way, the outer pixels, the pivot structure, and the dirt road were removed. Some pixels inside the field were also removed as a side effect of the filtering process, so a vector mask was created from the previously filter raster, then, the “holes” corresponding to the eliminated pixels that were in crop area were removed. The “Extract by mask” tool was used to clip the original raster preserving only the areas occupied by the crop, resulting in the maps of Figures 19 and 20.

3.3.2 Yield modeling

3.3.2.1 Yield predictions with ANN (Fields A, B, and C)

Among the main goals of Precision Agriculture are optimizing crop inputs and maximizing yields. Many tools are available to do this optimization processes. One of the most popular tools today is TensorFlow, which is an open-source and high-level library which specializes in machine learning. One of TensorFlow's classes is the "Dense" Artificial Neural Network, which was used to make yield predictions and find the Seeding rate value that maximizes yield. That optimized Seeding rate value is considered the seeding prescription. The full normalized database was divided in training and test set 80% and 20% respectively, and yield was set as target variable. Once the model (ANN1) was trained, an algorithm was developed to find the seeding rate value that maximizes yield for each datapoint. This was achieved by making ten yield predictions for each point, varying the normalized seeding rate from 0 to 1 with 0.1 increments and locking the rest of the input variables. Then, the seeding rate that generated the highest yield prediction was output as the "prescribed seed rate.

3.3.2.2 Yield modeling with regression analysis (Field D)

Since Field D is used to grow silage corn, NDVI is a good indicator of yield, which is not necessarily the case when working with grain corn. In this case to estimate yield, the dry plants collected in sampling areas of 4 x 4 m were weighed, then the dry mass was multiplied by a factor of 625 to scale it up to the equivalent mass in one hectare. NDVI values were aggregated in cells of 4 x 4 m, averaging the values of the 10 x 10 cm original cells. Once the NDVI cells were aggregated, the tool "sample" was used to collect the NDVI values corresponding to the yield sample locations. Linear, polynomial, logarithmic, and exponential regression models were fitted to the data. The outlier point highlighted in red was not considered. The model with highest r^2 value was the exponential model, with $r^2 = 0.98$ (Figure 26 and equation 1).

$$\text{Dry matter} \left(\frac{t}{ha} \right) = 0.0045e^{10.606NDVI}$$

Equation 1. Exponential model

The model was applied to the NDVI raster, and an Estimated Yield raster was obtained, then, the raster was classified in five-ton ranges, and converted to a polygon shapefile

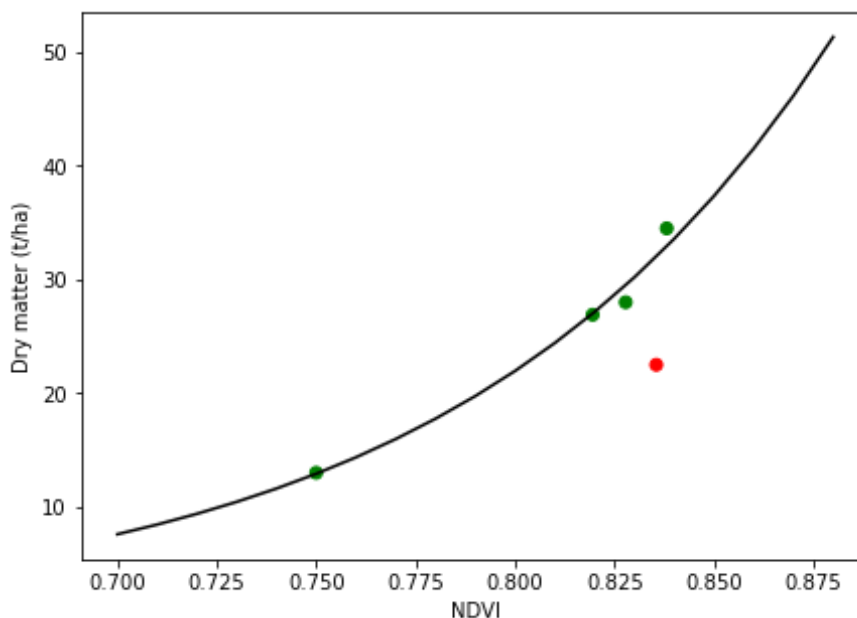


Figure 26 Exponential model fitted to yield and NDVI data. Atypical datapoint marked in red was not considered.

3.3.3 Variable-rate recommendations

3.3.3.1 Seeding rate recommendations with ANN (Fields A, B, and C)

The second model (ANN2) aimed to prescribe the seeding rate according to a variable that represents the soil's productive potential / environmental index. This involved extracting insights from the database before training the model. Talano (2016) analyzed the corn response to different seeding rates according to an environmental index. As Figure 27 shows, each hybrid responds differently to seed rate and environmental index. The plot on the left of Figure 27 shows a way to find the optimum seed rate according to hybrid and environmental index. The variable designated to represent the environment index in this case was the Apparent Electrical conductivity, since it correlates strongly with important soil properties (Corwin & Lesch, 2005).

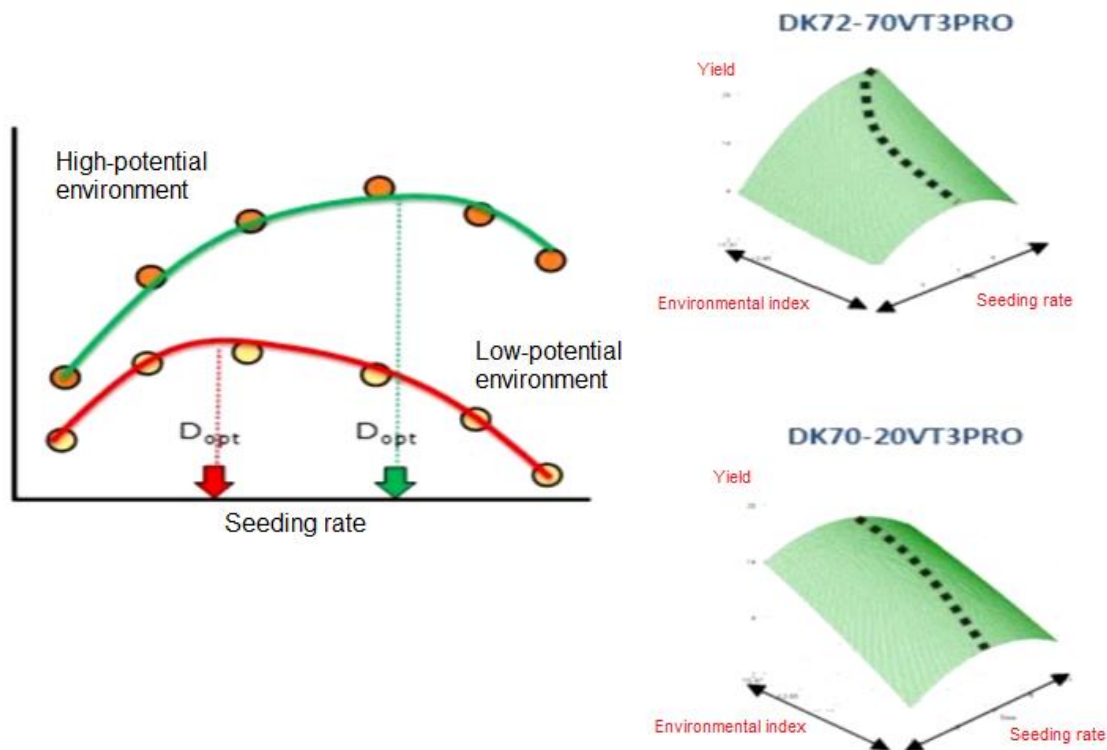


Figure 27. Response of two corn hybrids to seeding rate and environmental index. Yield increases along seeding rate, until excessive plants start competing for nutrients and yield decreases again. From Talano (2016)

To follow the methodology proposed by Talano (2016), the datapoints were divided by field to account for the unique properties and behavior of each field. Then, the datapoints were grouped by ECa value in ten evenly spaced ranges to have a good generalization of the low, medium, and high ECa values and in which the groups had at least four different yield values. For each group, the seeding rate was plotted against the yield. The next step was to compute the average yield per seeding rate value. In Figure 28, the greatest average yield value is 0.37, this corresponds to a 0.82 seeding rate value, which is considered the optimum rate and hence, prescribed value. This operation was repeated for each ECa range in each field (Figure 29).

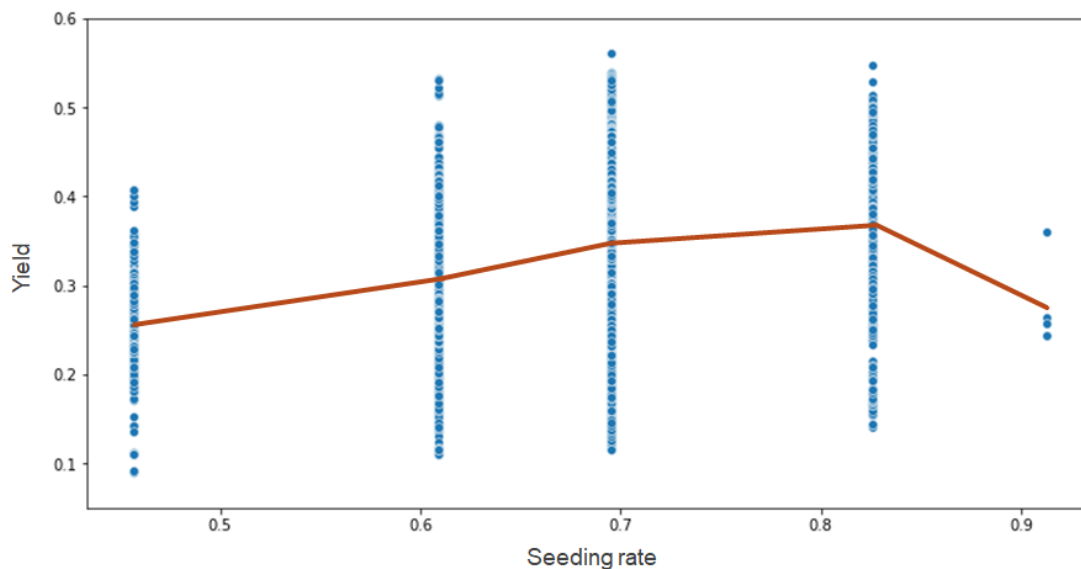


Figure 28. Field A, ECa range 0.4 – 0.5 (datapoints in blue, yield average in brown).
Observed behavior is similar to the one described by Talano (2016)

Once the optimum values for each ECa range in every field were obtained, the intermediate values were interpolated to obtain a function for each field that maximizes yield based on seeding rate.

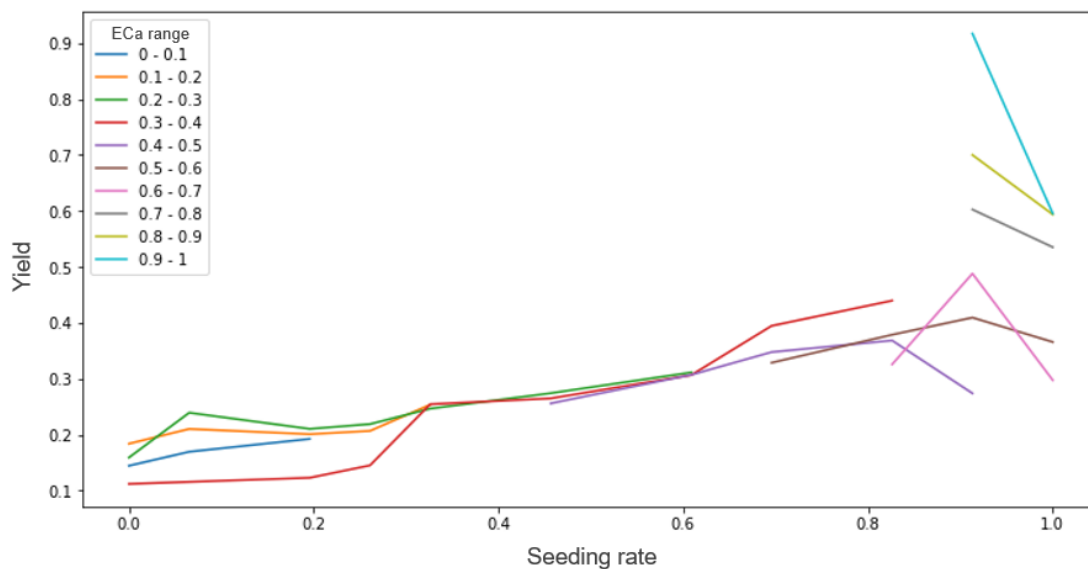


Figure 29. Average yield for each ECa range and seeding rate in field A

The interpolation was performed with the “interpolate” method of a pandas dataframe object (Pandas’ development team, 2022). The interpolation method was polynomial of order three. A new field “Optimum seed rate”, was added to the database and the optimized values were assigned to each of the datapoints. Then, the database was divided into training and test sets 80% and 20% respectively, and artificial neural networks of “dense” class were trained to predict optimum seed rates. Once trained and tested, the ANN2 was used to predict the optimum seed rate for all the data points. ANN1 predicted the corn yield using the seeding rates recommended by the ANN2 as shown on Figure 30.

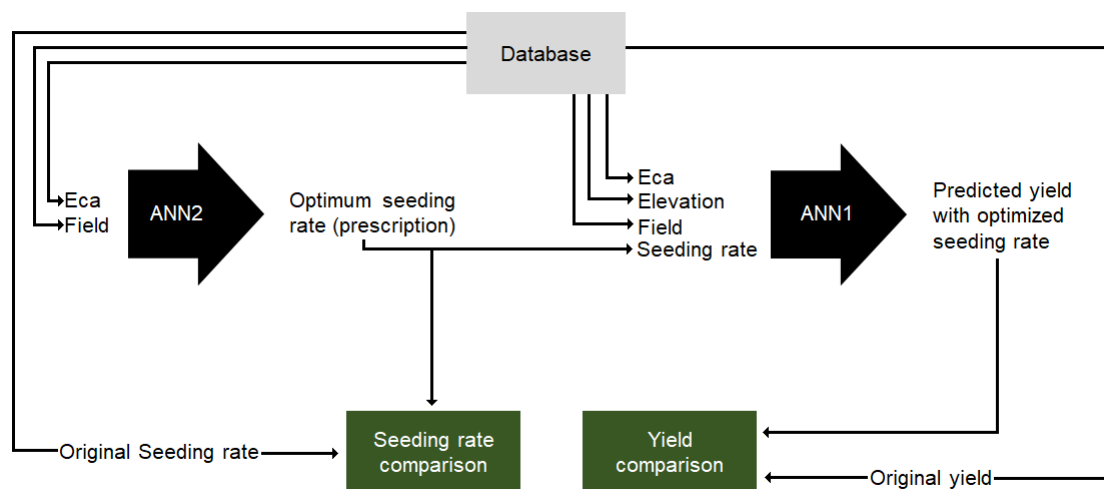


Figure 30. ANN prediction’s data processing

3.3.3.2 Nitrogen rate recommendations with linear model (Field D)

A formula (equation 2) to compute nitrogen rate was adapted from Stanford (1973). The model considers the nitrogen extraction needs of the crop according to a yield goal, the nitrogen present in the soil, the nitrogen assimilation efficiency of the crop, the fertilization method, and the fertilizer to optimize the nitrogen rate for each specific site.

$$\text{Nitrogen Rate}(kgha^{-1}) = \frac{(Yg \times N_e) - ((N_m + N_i + N_r + N_o) \times Ef_N)}{Ef_{fert}}$$

Equation 2. Nitrogen rate model

Where: Y_g is the yield goal of the crop in $t\ ha^{-1}$, N_e is the nitrogen extracted by yield ton, N_m is the mineralized nitrogen in the soil, N_i is the inorganic nitrogen in the soil's profile, N_r is the residual nitrogen on the soil, N_o is the organic nitrogen in the soil, E_{fN} is the crop's nitrogen assimilation efficiency, and E_{fert} is the fertilization method's efficiency. A goal of $24\ t\ ha^{-1}$ of dry-matter corn was established, as it is a common value for high yielding silage corn in the area, this goal was also supported by the laboratory's analysis. The considered nitrogen extraction needs for silage corn crop was $12\ kg\ ha / t^{-1}$ (Stanford, 1973), a nitrogen assimilation constant of 60%, and 80% efficiency for nitrogen fertigation via center pivot. Residual and organic nitrogen were considered as $0\ kg\ ha^{-1}$ since the whole plants are harvested and no crop residue is left on the field. The fertilizer selected to apply nitrogen was urea (46-00-00).

3.3.4 AgStat application

3.3.4.1 User interface design

To facilitate the user-machine interactions, a clean looking user interface was designed. The interface layout was created in QtDesigner (QT, 2022) which provides a graphical user interface to arrange the elements on the screen. QtDesigner's files extension is ".ui". To link the icon files to their respective objects in the interface, a resources file ".qrc" was created, in this way, the resources (icons, sounds, among others) can be compiled to and read directly from a python file. Each screen contains only the strictly needed elements to perform the tasks, if a process requires more than five steps, the steps are divided in various screens with four to five steps each. Elements on the screen were arranged consistently over the screen layout to facilitate operation and navigation. The color palette chosen for the interface is shown in Figure 31. High-contrasting colors and bright background were chosen to facilitate the visualization and of screen elements in outdoor conditions with intense sunlight.

Layout	Interactive
#ECEFF0	#4A8FE7
#A0B532	#98BFF2
#C4D668	#E84444
#D9D9D9	#F08C8C
#404040	#000000

Figure 31. AgStat's color palette. The use of consistent colors favors an adequate usability.

3.3.4.1.1 Analysis

The data analysis menu was designed in a minimalist form (Figure 32), to let the user decide to visualize stored field data or generate new data (agronomic prescriptions) by processing the stored data. Most of the space in the screen was given to the menu buttons. Icons and text were added for clarity.



Figure 32. Analysis menu. Interface designed with big and easy-to-interpret buttons.

3.3.4.1.1.1 Prescription generation

The prescription generation process was divided into the steps/screens shown on Figure 33.

- On steps 1,2, and 3, the user enters/selects the information and parameters needed to generate the prescription. Input cost is optional (Figure 33, screen 3)
- Screen 4 summarizes the information provided so the user can verify if the data entered is correct and go back if necessary. If the user is satisfied with the information shown, then the “generate” button can be pressed to run the tool.
- On screen 5, the prescription map is plotted. If an area does not need input application, it will be colored in gray. The map’s legend shows the input rate and the area covered by the corresponding management zone. The user has the option to generate an economic estimate if the input cost was provided on screen 3.
- The economic estimate offers a summarized dashboard in which the user can see the monetary value of the variable-rate input application. The first element of the monetary value is the difference in expected income from selling the harvest product. This is achieved by predicting the yield obtained with the recommended crop-input rates, multiplied by the market price of the crop, the expected income is compared to the income of the previous cycle. The second element of the monetary value is the crop-input savings, the total mass of the input (i.e. metric tons of Nitrogen) are multiplied by their unitary cost, and then compared to the previous cycle costs. The sum of the income from selling the harvest, and the savings in crop-input is the monetary value displayed in the application (Figure 33, screen 6).
- Finally, the prescription shapefile can be exported to an external drive. The interface offers the option to remove previous files in the usb drive, since some field-monitor-controller brands require it to operate the prescription properly (Figure 33, screen 7). When the file transfer is complete, a dialog appears to notify the user (Figure 33, screen 8). When clicked on “accept”, the dialog takes the user to the main menu.

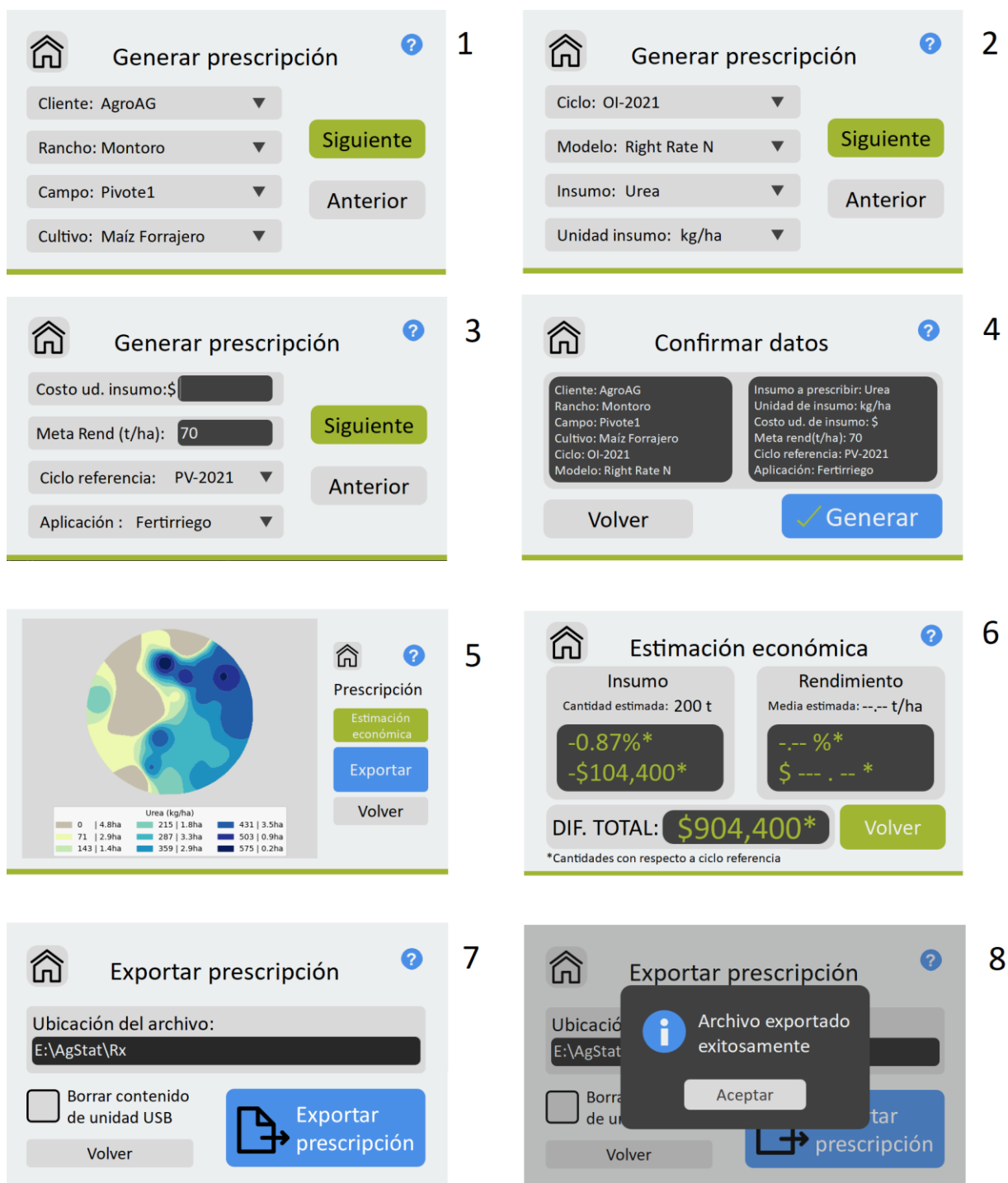


Figure 33. Prescription processing steps. Easy-to-follow interface during the processing steps.

3.3.4.1.1.2 Visualizations

The data visualization section helps the user to generate useful plots of the acquired data. The plots are generated with minimal parameter selection, and a blue call-to-action button. The visualization menu (Figure 34, screen 1) displays the four plotting options with icons related to the type of plot.



Figure 34. Data visualization utility screens. Menu with direct access to the different visualization tools.

- The map plot section shows the spatial distribution of the selected soil attribute, dividing it in five equally spaced ranges (Figure 34, screen 2).

- On the histogram plot section, the users can select the number of bins according to their needs (Figure 34, screen 3).
- The scatterplot and regression plots allow the user to select the “x” and “y” axis variables. The regression plot also permits the selection of the polynomial order (Figure 34, screens 4 and 5).

3.3.4.1.2 Data acquisition

Data enters the system via this section. It can be through the Field Survey Utility (FSU), or the user can enter a .csv file that contains scattered or interpolated data points (Figure 35, screen 1).



Figure 35. Data acquisition system

The user needs to specify the client, farm, and field the data corresponds to, as well as pertinent information like the variable to acquire or the directory in which the information will be stored (Figure 35, screens 2,3, and 4). The FSU has two operation modes, punctual, and on-the-go. Depending on the nature of the survey instrument, the user may select the mode that suits the survey needs. The punctual survey allows the user to take individual samples by pressing one button (Figure 35, screen 5). Meanwhile, the operation of on-the-go survey consists in toggling the data capture, with a central button (Figure 35, screen 6). On the top of the capture screen, the interface displays the client, farm, field, and variable name. The screen also displays the sample number, the current reading, and status of RTK and satellite signal.

3.3.4.1.3 Files

The files utility provides an easy interface to extract data from the AgStat system. It displays the folder trees of AgStat on the left, and the external USB drive on the right. The user must select the source file or folder, and the destination folder. Finally, with the press of the central button, a copy of the file or folder is transferred to the new directory (Figure 36).

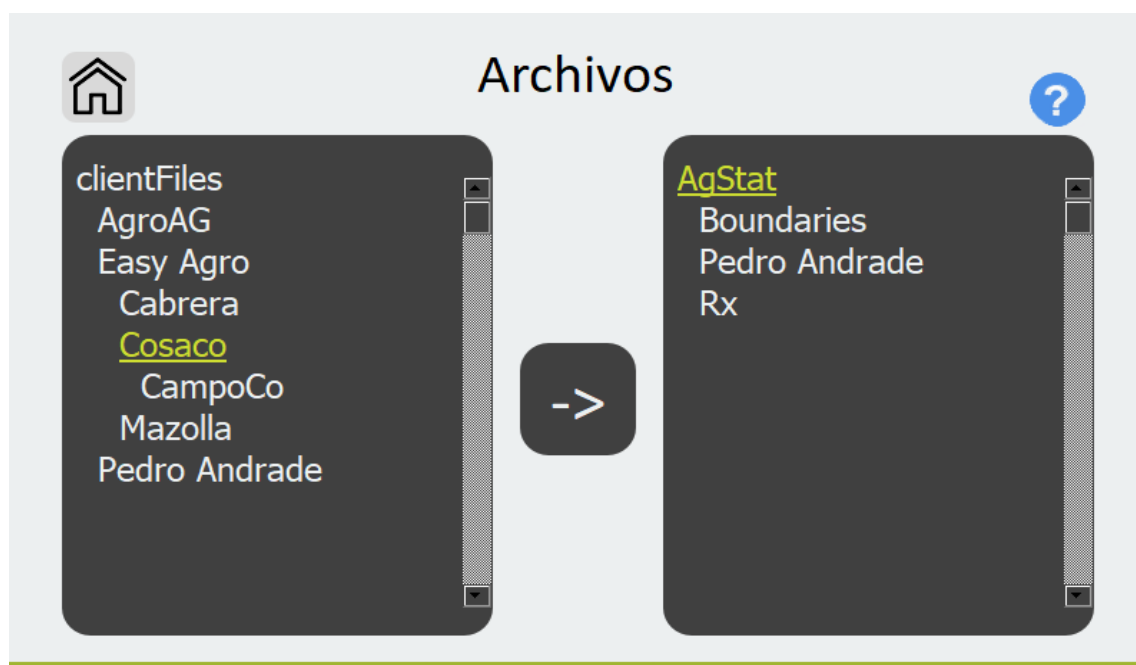


Figure 36. Files utility. Created to easily extract files and folders from the system

3.3.4.1.4 Settings

This section (Figure 37) was created to customize system parameters, user profile editing, wireless connectivity, which includes Global Navigation Satellite Systems (GPS, GLONASS, and Galileo), and Global System Mobile communications (GSM). The settings menu offers four clear options to the user from which specific functionalities can be accessed.



Figure 37. Settings menu interface

3.3.4.2 Programming

AgStat was developed in Microsoft Windows, the necessary python code to run it is contained in nine separate files which comprehend the different functionality aspects of the project. In some cases, more than one file will fit into the following categories. Since the number of objects contained in the python files' code (buttons, labels, frames, among others) is very high, the author established a naming and abbreviation system: section_subsection_class_objectname

i.e: an_gRx2_pushButton_home

In which “an” is the abbreviation for “analysis”, “gRx2” stands for “generate prescription, page 2”, “pushButton” indicates that the object is a push button, and “home” indicates that the button is meant to take the user to the home screen when it is pressed. The full list of abbreviations can be found at the annex 9.

3.3.4.2.1 Client database

This file contains the definition and structure of objects that will be recurrently mentioned in the next sections. This application was designed to store client profiles, similarly to systems like John Deere’s Operations Center, and Trimble Fmx display. The database structure is summarized in Figure 38.

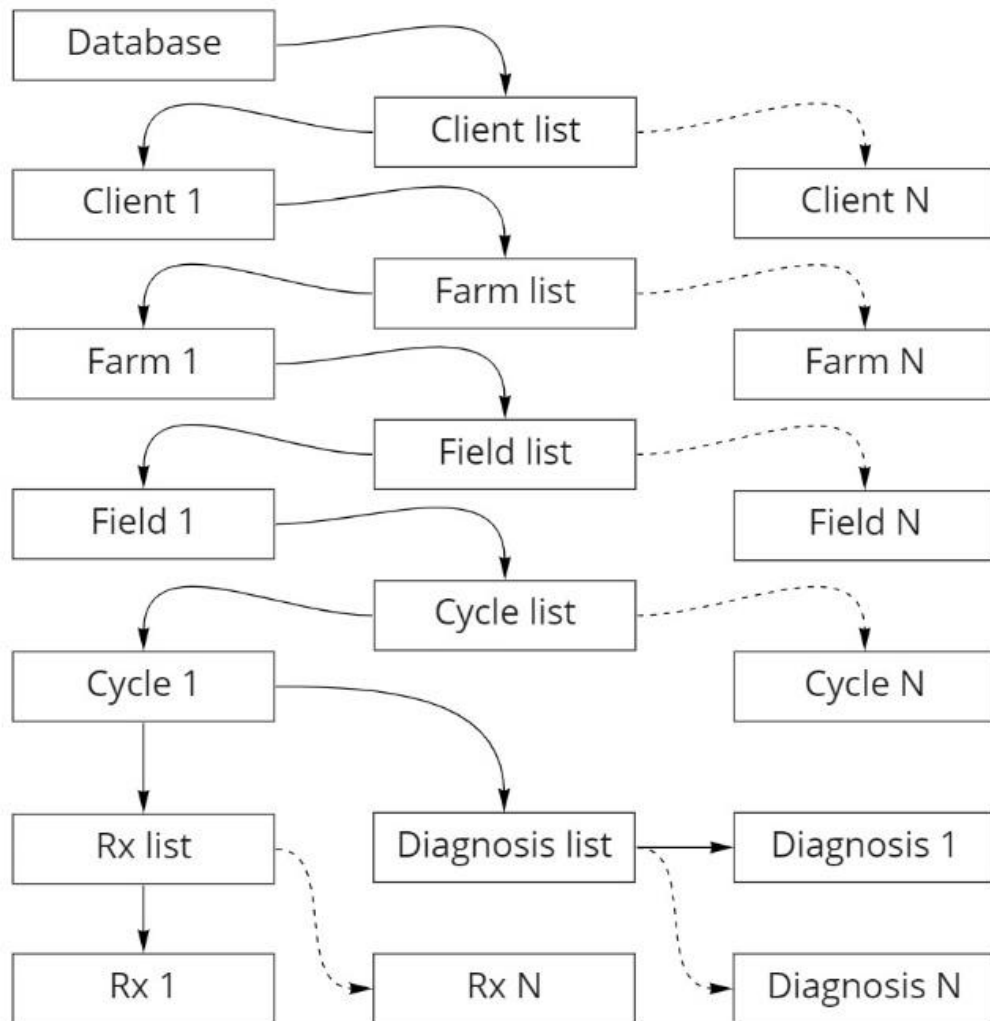


Figure 38. Client database’s general structure.

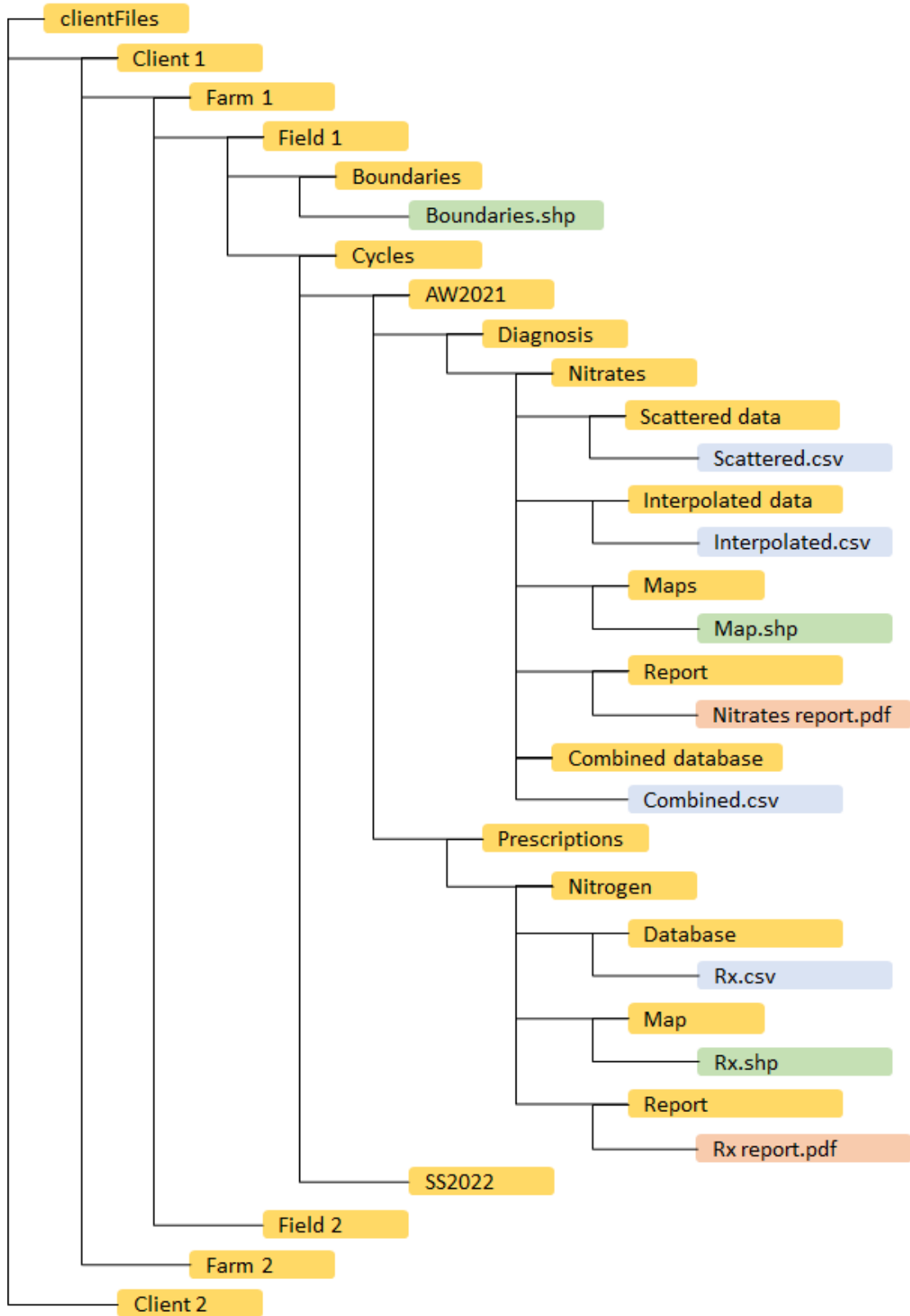


Figure 39. Proposed folder structure

The proposed folder structure is shown in Figure 39, it follows a similar logic to the database structure, it is designed to ease the process of storing and finding files within the system. The data reports in pdf are not implemented yet but are included in the structure's proposal since they would be a useful feature for the user. The objects are programmed to create their corresponding folders in the system automatically.

3.3.4.2.2 AgStat main

The file "AgStatMain.py" integrates the functionality of the rest of the files. The application runs from the code in this file. Most of the code in this file is in the "MainWindow" class definition, which is a subclass of "QMainWindow" and "UI_MainWindow". The different screens of the app are accessed changing the index of a Stacked Widget object. In this object, every screen or page has an assigned index, i.e., main menu screen is assigned to index 0.

The MainWindow class definition has the following structure

init function definition

Global indices setup: to load the first page of the stacked widget, as well as client, farm, and field information when the application starts.

Set working directories for the internal system and external USB drive, for now the external drive needs to be connected to the hardware before AgStat starts in order to be detected. If the external drive is connected after the reading function was executed, it will not be detected. This functionality will be improved in the future to detect external drives at any moment.

Navigation connections: Links between the navigation button objects and the navigation functions to browse the application pages. All the connections are made with signal-slot syntax.

Action button connections: Here, the action buttons (Generate histogram, generate prescription, export prescription, among others) are connected to their respective functions.

Combo boxes connections: The connections in this section allow all the drop-down menus (comboBox class) to synchronize their contents over all the application sections, i.e., if the user selects one client, all client combo boxes update to show the same client.

Navigation functions definition: Help the user to go to the different screens of the application by setting a determined index in the stacked widget.

Combo boxes functions: These functions set the elements of the drop-down menus, reset them, or change the selected element. All the combo boxes send a signal when the user selects one element. Depending on the combo box level (client, farm, field, or cycle) the rest of the combo boxes are updated i.e.:

The user selected a different farm:

1. All farm combo boxes update their selected value
2. Since the new selected farm has different fields and cycles, all the field and cycle combo boxes are emptied and then filled with the new fields.
3. The field combo boxes automatically display the first field of the selected farm
4. The cycle combo boxes automatically display the first cycle of the first field.

Plot functions definition: In this section, the system takes the values selected by the user, and then calls the plotting functions from the “AgStatPlots.py” file to visually represent the selected data i.e.:

The user pressed the “generate” button on the histogram section

1. The system reads the “client”, “farm”, “field”, “cycle”, and “bins” combo boxes to extract the values selected by the user. The values are stored in variables.
2. The plotting function is called, and the variables are passed as arguments.
3. A histogram with the specified parameters is generated

Dialog functions definition: These functions execute dialogs to notify the user that a task has been completed, i.e. prescription export.

Miscellaneous functions definition: A variety of functions is defined in this section. i.e. A function to fill the values in the confirmation page of the prescription generation process (Figure 33, screen 4). File transfer functions are also included in this category.

After the “MainWindow” class definition, the “InformationDialog” class is defined. This last class’s purpose is to visually notify the user with a pop-up dialog when a process has been completed (Figure 33. Screen 8). Finally, the last lines on the “AgStatMain.py” initiate and execute the application.

3.3.4.2.3 User interface

The file “AgStat_ui.py” was generated automatically by the calling following function in the Windows’ command prompt (after setting the prompt’s working directory to the folder containing the .ui file):

```
pyuic5 -x AgStat.ui -o AgStat_ui.py
```

The above-mentioned function takes the Qt designer’s “.ui” file as an input and converts it to a python code file that contains all the objects, their position, colors, size, among other attributes. If this file is executed, the created interface will be displayed but the objects will not function properly until connecting specific functions with signals and slots (AgStatMain.py). The resources file “.qrc” was converted to python file as well. To perform this task, the following function was called in the command prompt, similarly to the previously mentioned function:

```
pyrcc5 -o AgStatResources_rc.py AgStatResources.qrc
```

Without the proper resource’s python file, the interface’s objects will not display their icons when executed.

The information dialog’s interface was designed following the same methodology as the main window, the python file obtained is “InfoDlg.py”

3.3.4.2.4 Processing

The file “AgStatProcessing.py” contains the code to process geospatial and tabular data. Figure 40 shows the functionality that results from combining the functions included in this file.

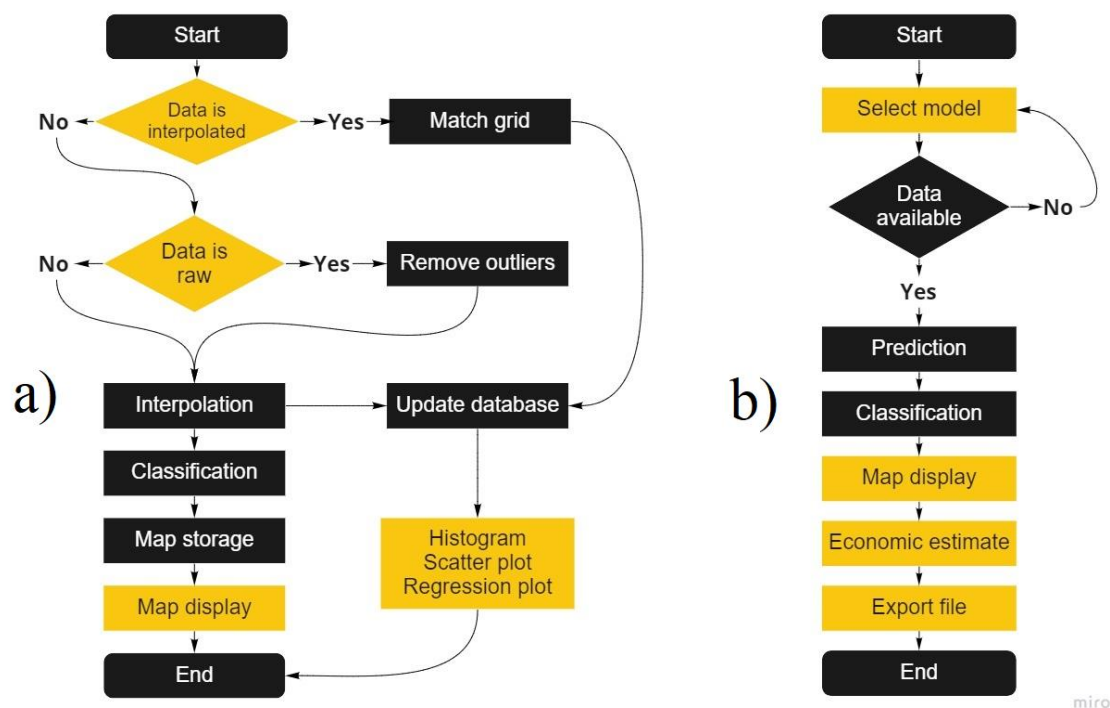


Figure 40. Integrated data processing functionalities. To enter data in the system and create maps, and to generate prescriptions.

Figure 40a describes the data acquisition, storage, and display processes. Figure 40b shows the prescription generation process, the economic estimate functions are still under development. Now, the specific functions will be described in order as they appear in the file:

`getBoundariesDir`: Takes a diagnosis object as input and returns its field’s boundaries shapefile directory (Figure 39).

`paddedGrid`: adds a padding of approximately 40 m to the bounds of the file. Its purpose is to avoid missing data after interpolation, since it is a common problem with some

libraries. The excess data outside the boundaries of the field is discarded after interpolation.

`scatterToDiagnosisMap`: takes a diagnosis object as input and generates a diagnosis map based on scattered data points. This function accesses a folder named “unused”, its purpose is to store temporary files that are generated in intermediate processing steps, but are useless afterwards, so they are deleted automatically. The processing steps of this function are briefly described below:

1. Set directories: Generates strings containing the routes to the source files, and the files to be generated.
2. Delete previous files in the directories if they exist.
3. Read datapoints, and delete values up to percentile 5, and from percentile 95 if the number of datapoints is greater than 100. The threshold of 100 values was established to prevent the algorithm from automatically deleting 10% of the datapoints (and losing information) if the number of observations is small (less than 100).
4. Interpolate datapoints via `gdal.Grid()` and remove data outside boundaries with `gdal.Warp()`. Then save the interpolated data as a “.csv” file
5. Classify interpolated points into five evenly spaced classes, and store class breakpoints into a “.csv” file. This classification is done for plotting purposes and facilitates map interpretation, by displaying two high ranges, two low ranges, and a middle range.
6. Generate a raster “.tif” file from the classified datapoints.
7. Convert classified raster to polygon features with `gdal.Polygonize()`
8. Trim the edges of the resulting shapefile using the boundaries shapefile as reference.
9. Delete garbage and unused files.

`generateRx`: This function contains the Prescription Generation Algorithm (PGA). The inputs are “cycle”, “crop”, “agricultural input”, “application method”, and “yield goal”. The prescription values are obtained applying a vectorized function to a dataframe. A brief description of the processing steps is:

1. Set directories to read boundaries shapefile, and to store prescription shapefile.
2. Read cycle's interpolated database into a data frame
3. Add constant columns to dataframe according to conditions (efficiency, crop, yield goal, and agricultural input)
4. Apply vectorized prescription formula to dataframe and create a column with the results
5. Set nine evenly spaced values (rates) between the maximum and minimum prescription value. This classification was designed to allow a relatively smooth change between different rates by maintaining consistent rate changes over the field.
6. Snap/round prescription values to the nearest rate value
7. Call generateMap function to create prescription shapefile

generateMap: Reads classified values file, generates a raster, which is then converted to a shapefile. Finally cuts the shapefile using the boundaries file and stores it in the assigned directory.

refreshDatabase: This function removes the previous combined database from the cycle's diagnosis folder, and creates a new file, joining the different diagnosis interpolated files present at that moment. This is useful when a new variable is added to the cycle's diagnosis.

3.3.4.2.5 Plots

Matplotlib is connected to PyQt5 via "mplwidget.py" file which contains the classes that allow the developer to embed matplotlib objects into PyQt5 widgets. In this application, QFrame widgets were located where the plots were supposed to be, the QFrames were then promoted to inherit the methods of the MplWidget class, so they could display the actual plot.

"AgStatPlots.py" contains functions based on matplotlib to help visualize the data on an easy manner.

truncate_colormap: delimits a colormap for aesthetic purposes.

plotDiagnosisMap: reads diagnosis shapefile, configures the legend, and plots the map on the corresponding axes object.

plotRx: reads prescription shapefile, computes the management zones' areas, configures the legend, and plots the map on the corresponding axes object. For now, is configured for Urea prescriptions only.

3.3.4.2.6 File system

This functionality was developed to extract files from the application's folders to an external disk. Files cannot be transferred from external drives to the application's storage folders via the file system to protect the integrity and structure of folders and files (the DAA performs this operation automatically). To extract files, the user must select the source folder/file in the File System 1, and the destination folder in the File System 2, and then click on the central button to perform the transfer. The following descriptions correspond to the classes from "AgStatFileSystem.py" file. Together, the objects in this system form a folder tree.

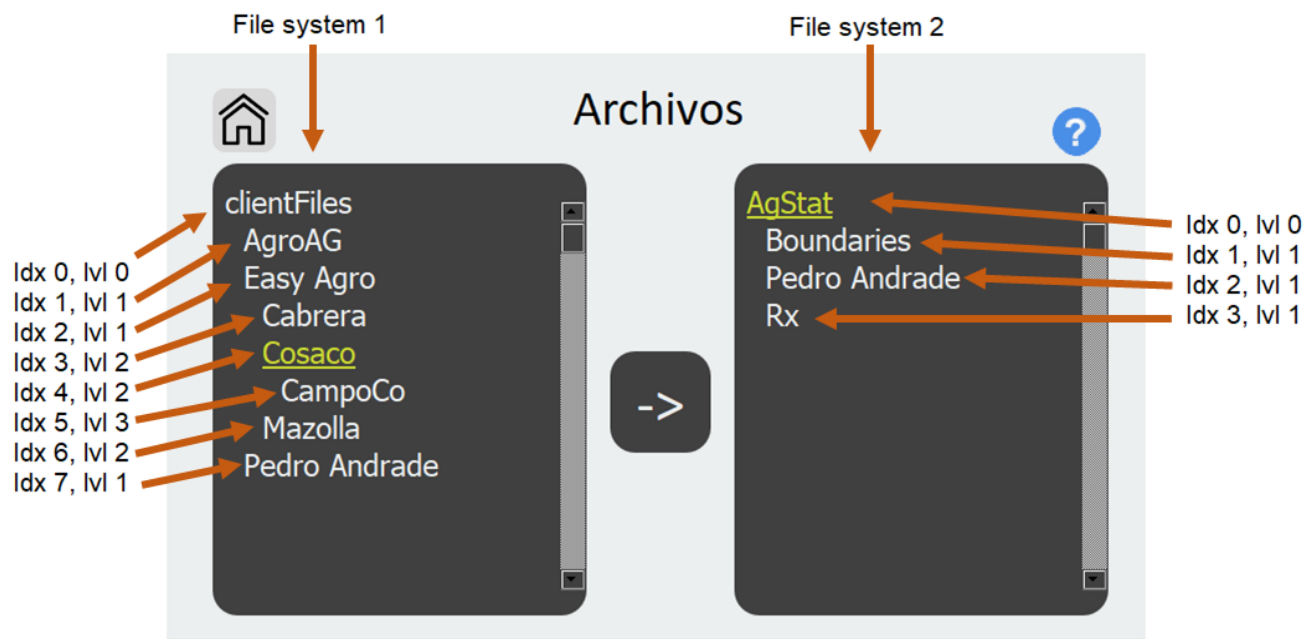


Figure 41. File systems and FolderLabel objects visual representation. Text outside screen and brown arrows are not part of the actual screen layout.

FolderLabel: This class functions as a button and as label. Folders and files are represented visually in the system with FolderLabel objects (Figure 41). When clicked, FolderLabel objects toggle between active and inactive state, this visually expands or collapses the sub folders contained in it. Its functions include:

- click: Toggles active and inactive state of the object. When the folder is activated, the setSubFolders function is called:
- setSubFolders: Iterates through the folder's content and fills the sub folder list with the folder and file names.

FileSystem: This class was created to manage and display the FolderLabel objects. Its main component is a list called "rows" in which FolderLabel objects are stored and removed according to the user's needs. The position of the FolderLabel is determined by its index in the list (Idx), and the folder level (lvl), every folder level is "the level of its parent folder + 1". This hierarchy is represented visually with indentation (Figure 41). Two file systems are used in AgStat, one for the application's folders, and the other for the external disk's folders. The following functions are methods for the FileSystem class:

- createHomeFolder: This function sets the root folder for the system (lvl 0) from which the rest of the folders are derived
- nextFolderIndex: Returns the index of the next folder on the same level as the input folder, if there is no next folder, it returns the last index of the rows list. The purpose of this function is to delimit the range of folders that will be visually removed when a parent folder is collapsed.
- updateRows: This function adjusts the position of the folders in the screen after expanding/collapsing a folder.

3.3.4.2.7 Settings

The functionalities of this section are still under development. They will include functions to modify the clients database and high level connectivity functionalities to receive data from GNSS and engage with the GSM network. Additionally display functionalities will be implemented to modify screen appearance according to working conditions to improve usability.

4 RESULTS AND DISCUSSION

4.1 Yield modeling

4.1.1 Fields A, B, and C (ANN Predictions)

4.1.1.1 Interpolated database

The resulting interpolated maps are shown on Figures 42, 43, and 44. The spatial gradient of the variables is clear and has the appearance of a continuous surface, except for the seeding rate maps, which are distributed in discrete steps.

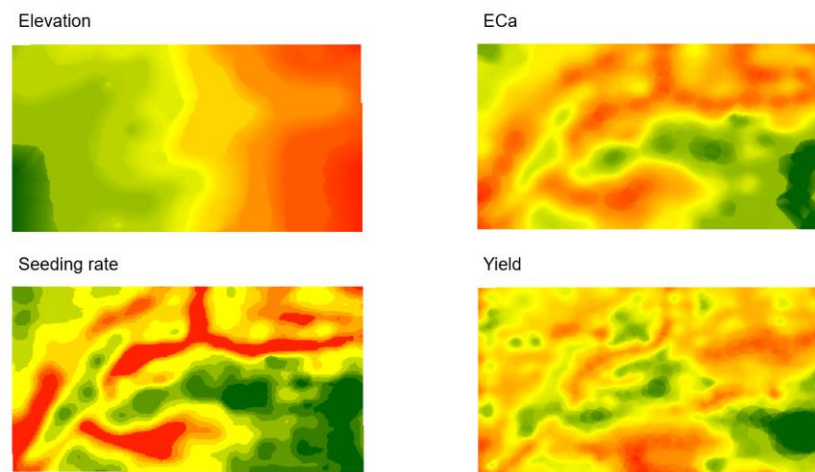


Figure 42. Interpolated surfaces derived from point features (field A)

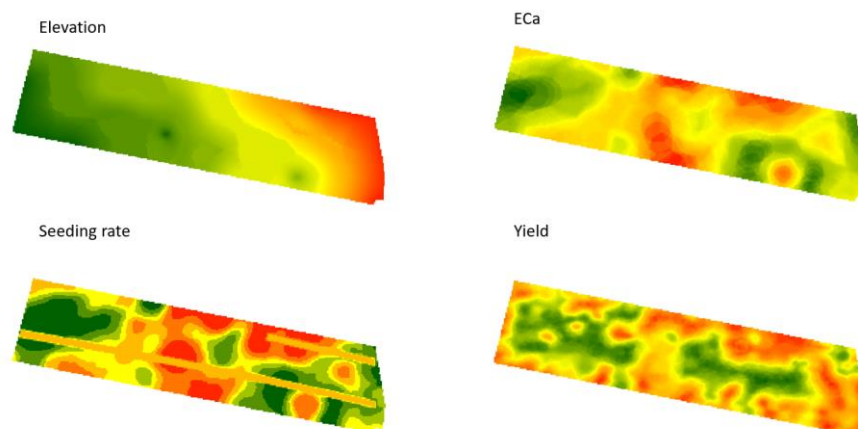


Figure 43. Interpolated surfaces derived from point features (field B)

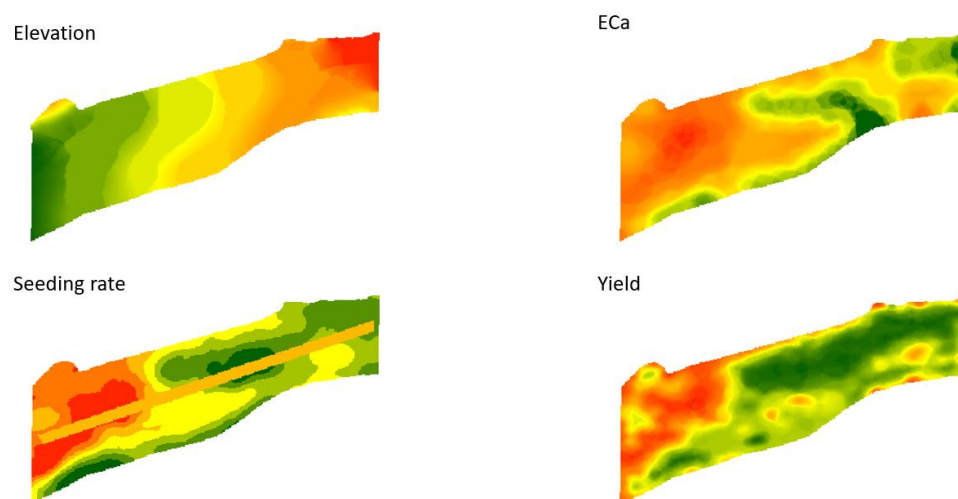


Figure 44. Interpolated surfaces derived from point features. (field C)

The value distribution of the datapoints is shown in Figures 45, 46, and 47. Generally speaking, the variables have low correlation coefficients with yield, which may derive from the large number of factors that influence the soil and plant behavior. The specific correlation values are shown on Table 5:

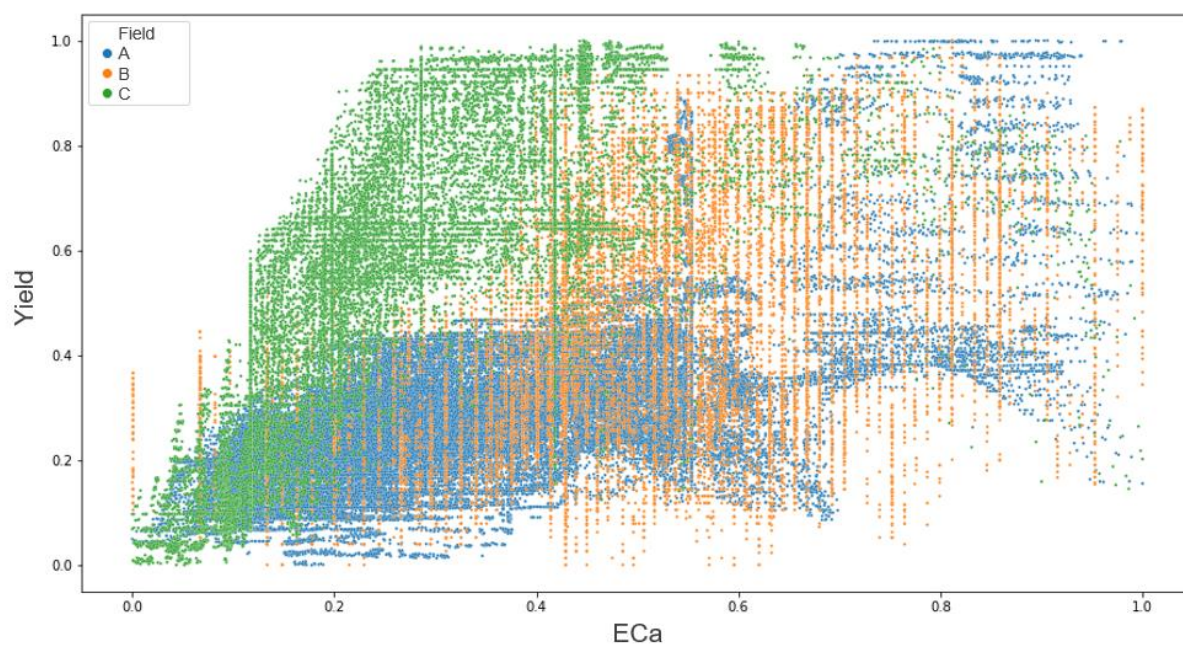


Figure 45. Interpolated database (ECa vs Yield)

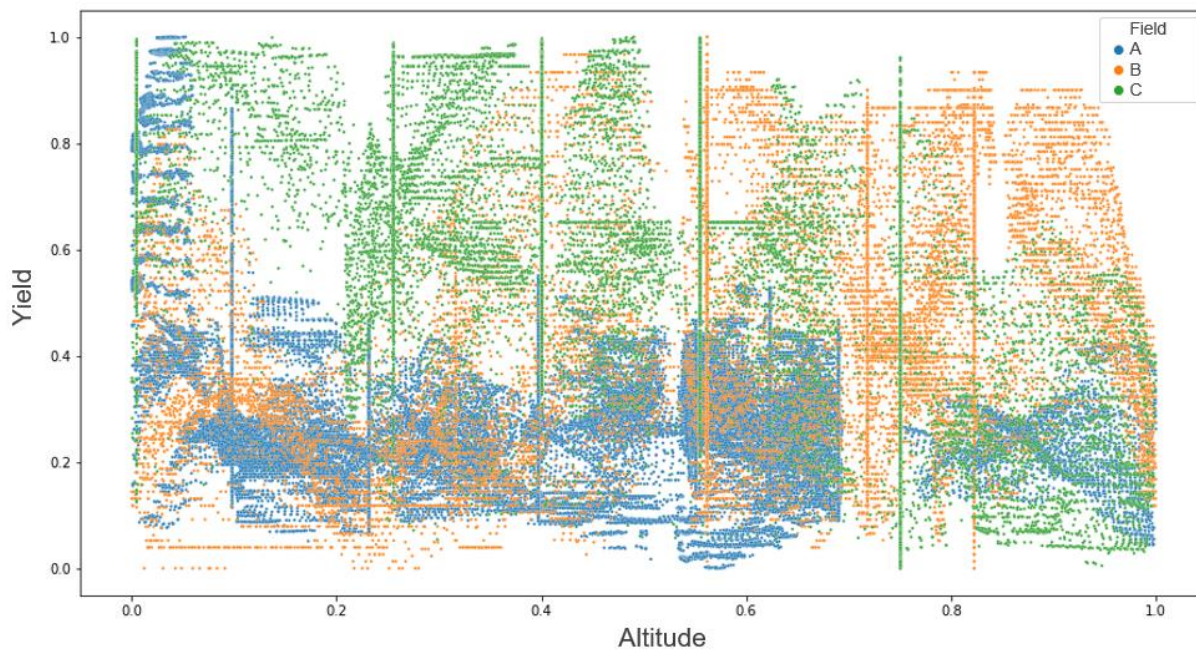


Figure 46. Interpolated database (altitude vs yield)

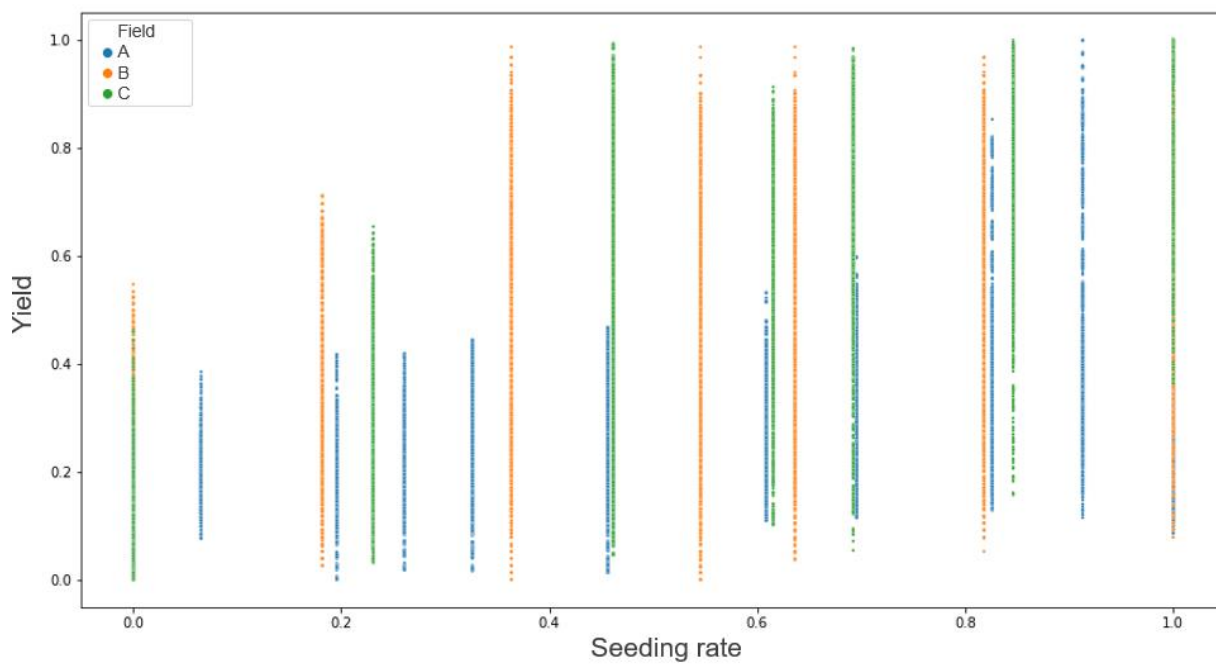


Figure 47. Interpolated database (Seeding rate vs yield)

Table 5. Variables' correlation with yield

Variables	Correlation
ECa – Yield	0.44
Altitude – Yield	-0.07
Seeding rate - Yield	0.54

4.1.1.2 Model predictions

The best prediction results ($r^2 = 0.954$) of the Dense ANN were achieved in 1200 epochs with 6 hidden layers of 6, 20,15,10,10, and 6 neurons respectively (Figures 48 and 49). After 1200 epochs, the error stayed relatively constant, so this number of epochs was specified to save computation time. The validation loss presented some noise, possibly due to important ANN weights adjusted in a “wrong” manner and thus increasing the error before correcting in the following epoch. Successful crop yield predictions have been done in the past in a similar manner. Dahikar et al. (2014) predicted yield from several crops using soil parameters like pH, nitrogen content, organic carbon, among others with one and two hidden layers. On the other hand, and with more similarities with this work, Ehsani et al. (2005) predicted corn yield based on ECa data and seeding rates with one hidden layer.

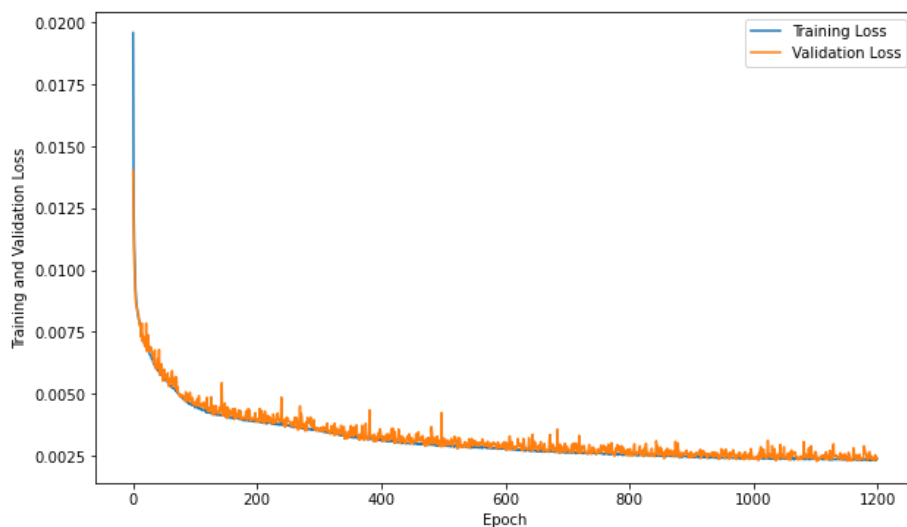


Figure 48. Model loss (Mean Squared Error) progress during training. The error decreases as expected. Validation loss presents some noise

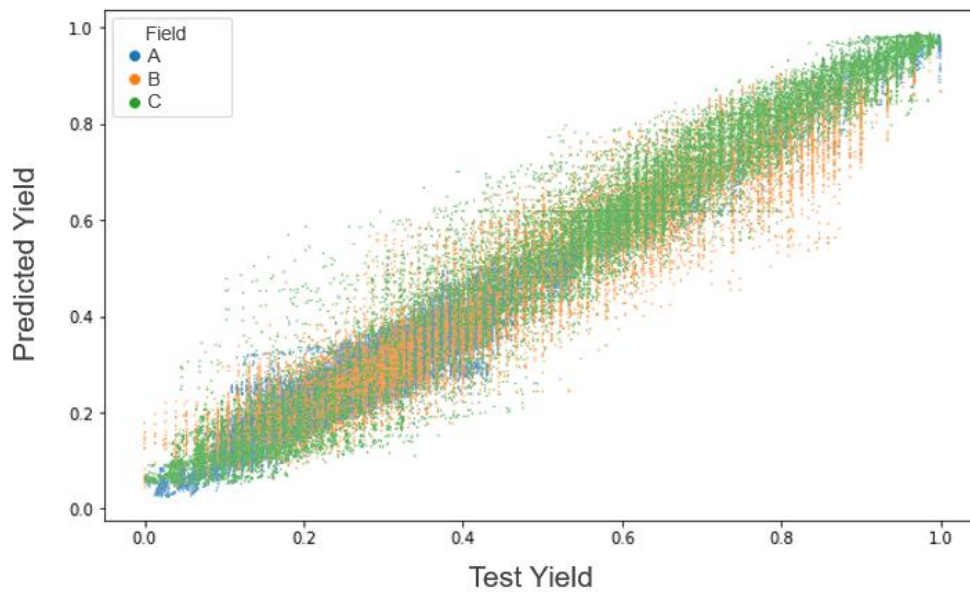


Figure 49. Yield prediction results by ANN (fields A, B, and C)

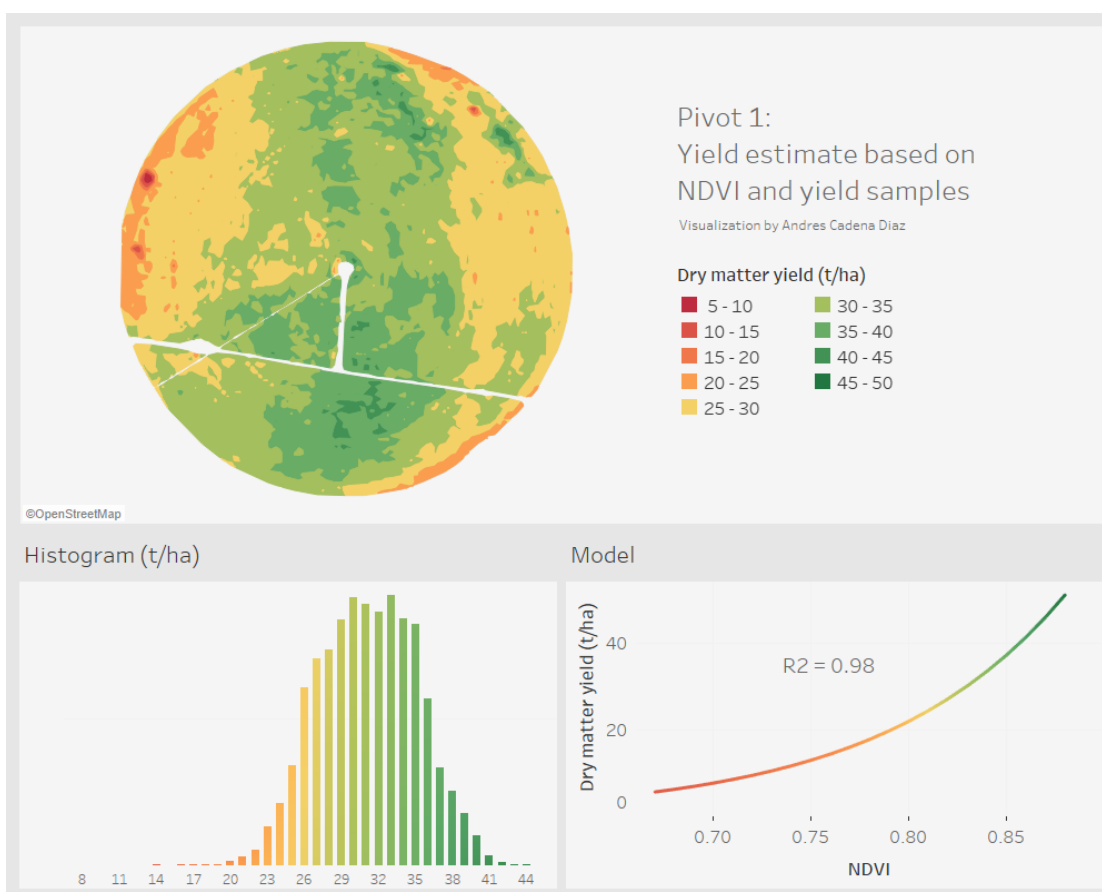


Figure 50. Estimated yield map generated by regression (field D)

4.1.2 Field D (Fixed exponential model)

The obtained yield map (Figure 50) has reasonable values that are close to the actual yield values (Table 6), although more samples are needed to fully validate it. A high-yielding belt is located at the center of the field, particularly in the southern end, which has the highest altitude. The average estimate yield is 26 t/ha of dry matter, with some outlier values reaching 44 t ha⁻¹. Some spots presented poor plant vigor as indicated in the original NDVI raster and translated into the estimated yield map. Further research is needed to identify the precise causes for this phenomenon and prevent it from affecting larger areas in the field.

Table 6. Comparison between actual yield values and estimated yield values

Sample	NDVI	Actual yield (t/ha)	Estimated yield (t/ha)
1	0.75	13	12.9
2	0.83	28	30.2
3	0.84	34.5	33.6
4	0.82	26.9	27.1

4.2 Variable-rate recommendations

4.2.1 Fields A, B, and C (ANN predictions)

4.2.1.1 Derived from ANN1

The prescriptions generated by the ANN1 presented a notable prediction variability between trainings, as shown on Figure 51. This situation makes them impractical and unreliable. The variability may be caused by the random initialization of the neuron's weights and can be addressed by setting a fixed "random seed" (RS) value. It should be noted that RSs are values that are used in this case to initialize the weights of an ANN to get reproducible results and this concept has nothing to do with plant seeds. Nevertheless, if the training data presented a clear pattern, the predictions would be similar enough among trainings, but they are even contradictory in some zones, so there is no way to

know which trained model is the best, and therefore there is no good reason to use RSs. The plots on the right of Figure 52 were obtained by plotting the predicted yield for the seeding rate values between 0 and 1, while locking ECa and Altitude at 0.5.

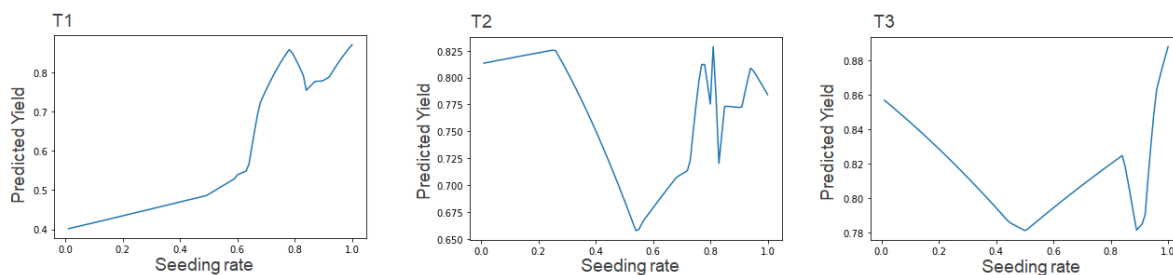


Figure 51. Model 1 response to variation in seeding rate during three different trainings. Notable variability is present in the models due to random initialization and no clear patterns in the data

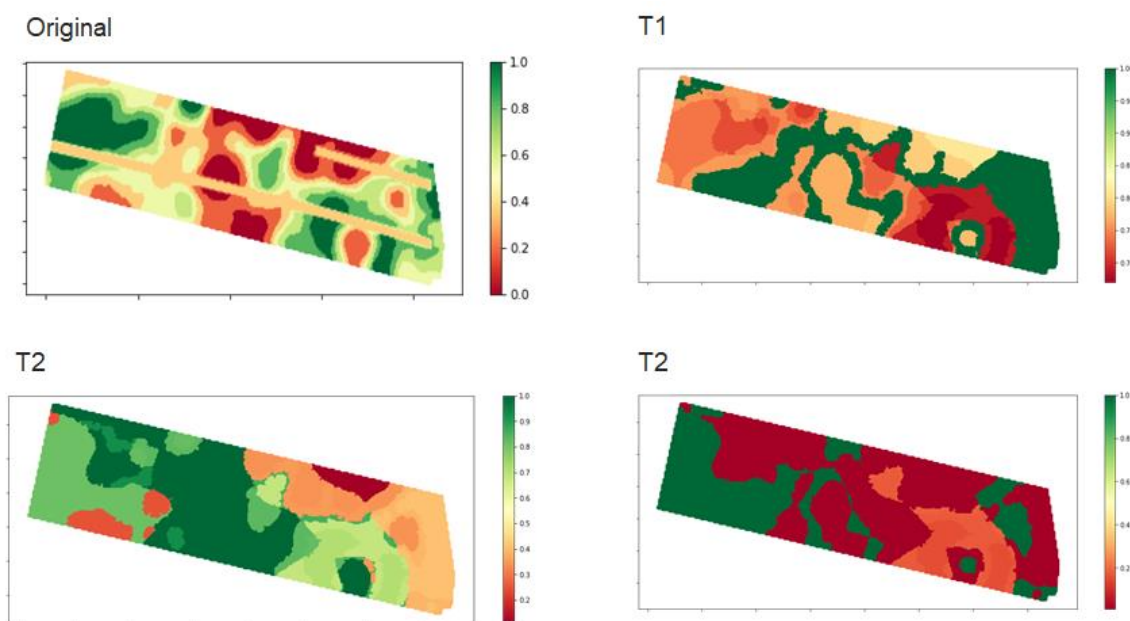


Figure 52. Seeding rate recommendations from Model 1. Notable differences are observed between maps

4.2.1.2 Recommendations from model 2

The analysis that was manually carried out to find the seeding rate that maximizes yield resulted in the three models, one for each field, shown in Figure 53, those values were used to train the model.

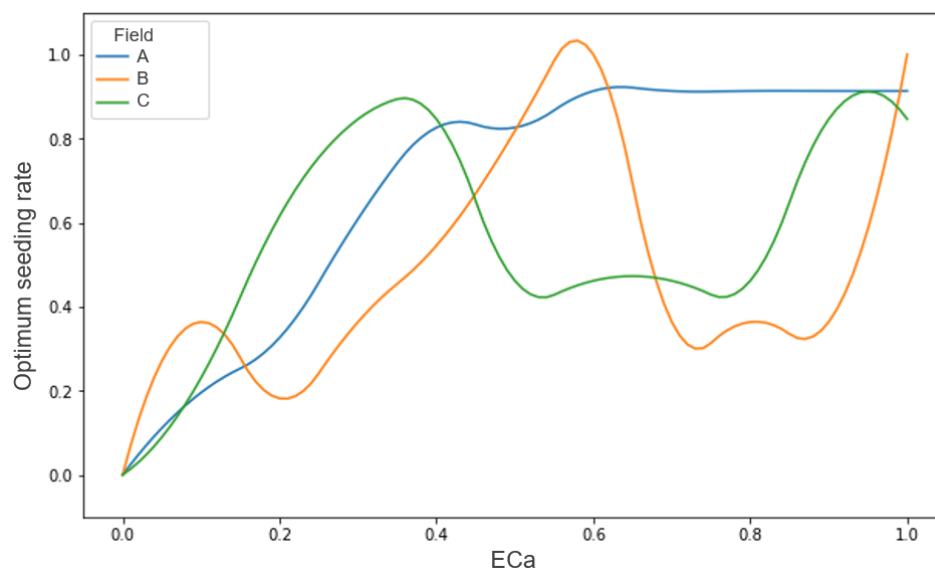


Figure 53. Optimized seeding rate functions for each field

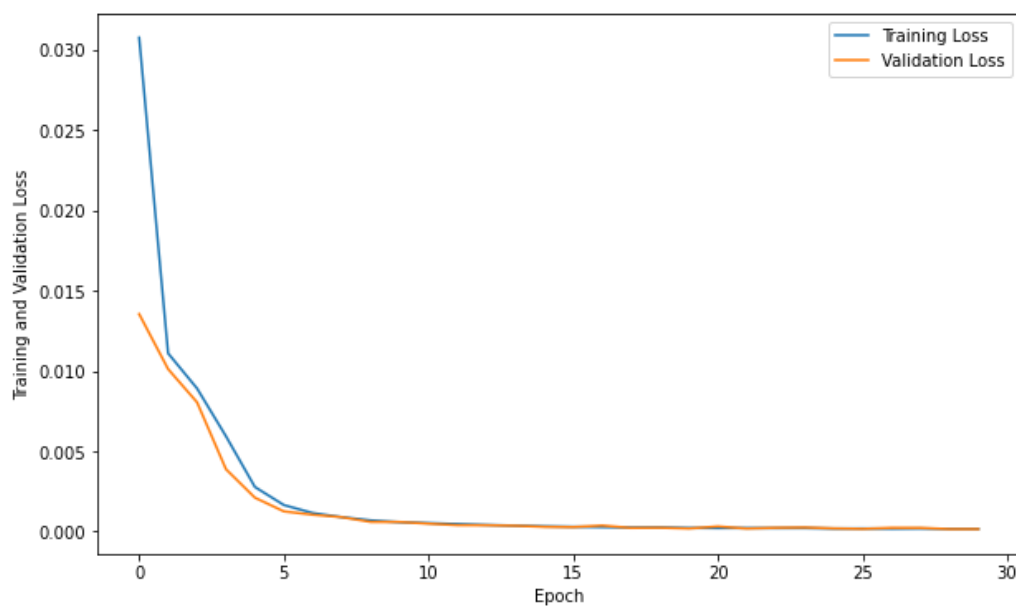


Figure 54. Model 2 Loss (Mean Squared Error) progress during training

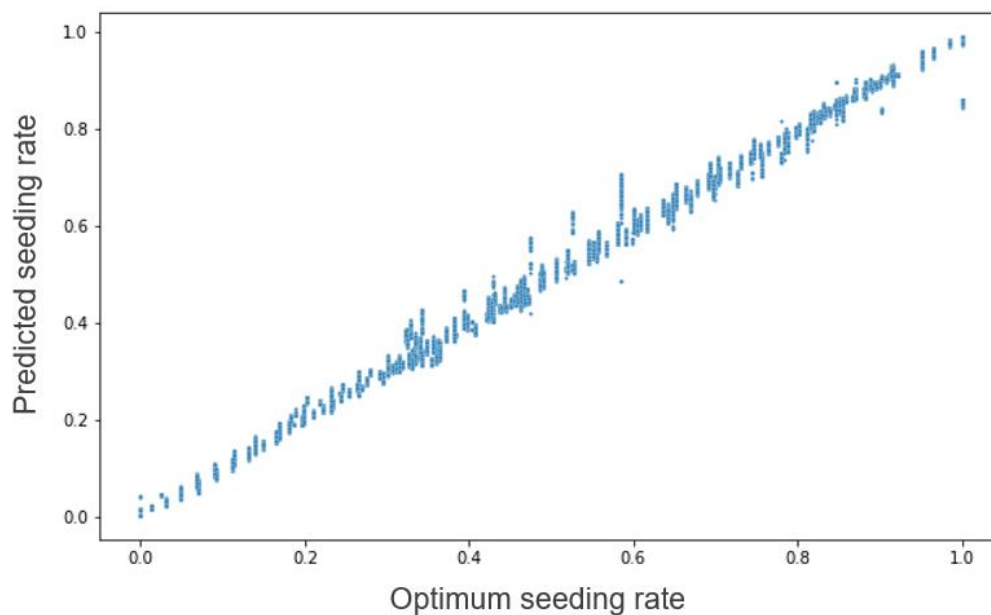


Figure 55. Seed prediction results from model 2

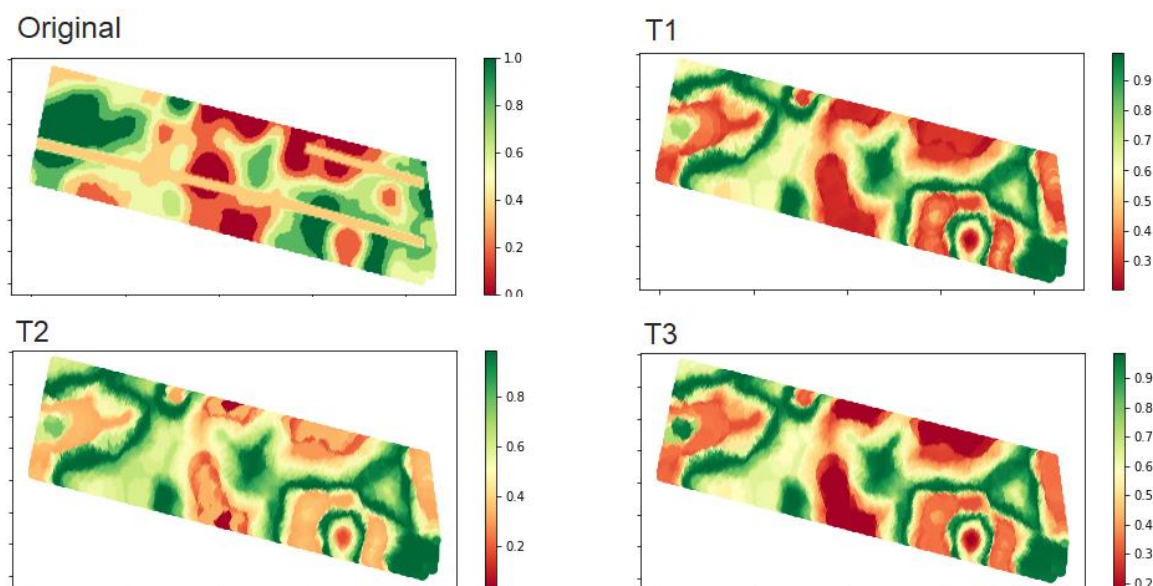


Figure 56. Seeding rate recommendations from model 2 (field B) The rest of the resulting maps can be found at annexes 4 and 5.

The best results were obtained in 15 epochs with six hidden layers of 5, 20, 15, 10, 10, and 6 neurons respectively. Figure 54 shows that above 15 epochs, the error stays almost the same. The recommended seeding rate was plotted for each field (Figure 55).

On contrast with the ANN1, the ANN2 generated more consistent results, as shown on Figure 56. This was achieved because the training data was pre-processed and generalized to obtain polynomial functions (Figure 53), whose clear patterns are easily recreated by the ANN. The most noticeable differences from the original map are the continuous nature of the new prescriptions, and the absence of the control strips, which were added manually by the farm's advisor.

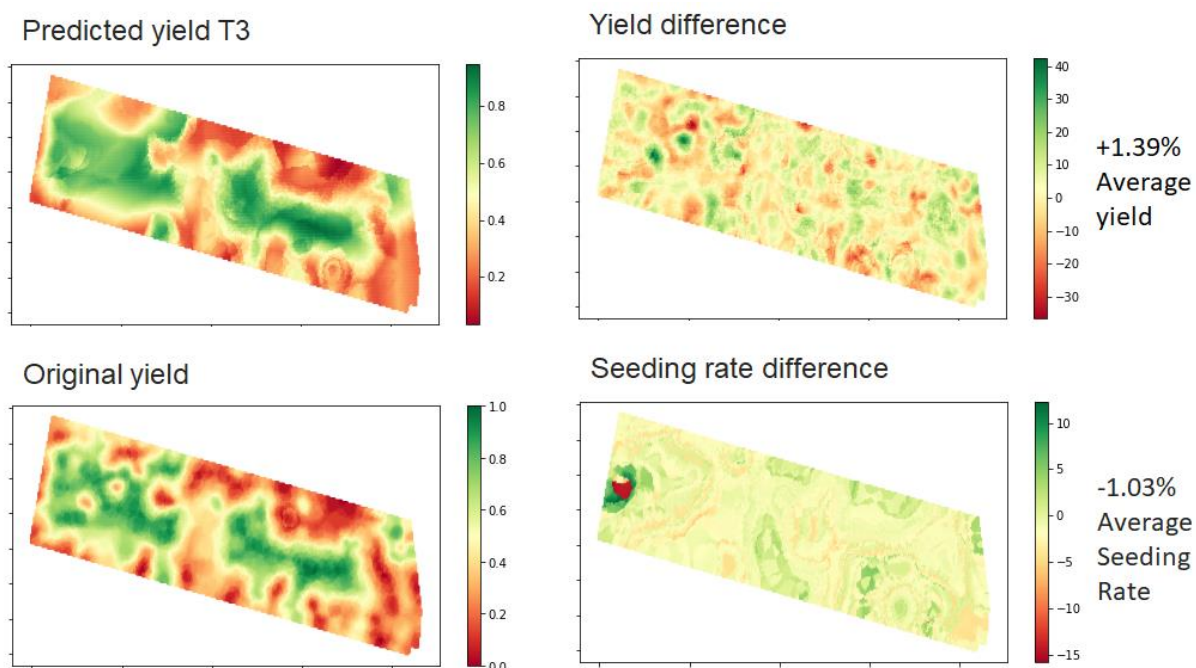


Figure 57. Yield and seeding differences between original values and generated prescription (field B). The rest of the resulting maps can be found at annexes 6 and 7.

The ANN1 was used to predict the yield that would be obtained using the recommendations of model 2 (Figure 30). The original yield and seeding rates were compared to the predicted yield and the recommended seeding rate (Table 7 and Figure 57). Despite the desired consistency in the seeding rate predictions, the overall differences in yield and seeding rate are not good enough. Results show that the average yields and seeding rate changes are close to 0%. In some zones, the predicted yield increased as desired, but decreased in other zones. The same situation applies to the seeding rates, the fundamental difference is that for seeding rate, a reduction is the desired effect. Given the wide range of values that compensate each other, the total effect is very little. Possible

causes for this wide variation in the predicted values include inaccuracies in the original data or while digitizing the images. There may be factors that were not considered, like pest incidences, which can notably affect yield. Further research is needed to optimize the models and find a way to keep the desired datapoints while avoiding the undesired ones, or at least keep the original values ($\Delta = 0$) to prevent their negative effects.

Table 7. Prescription results compared to original data

	Field A	Field B	Field C
Average Δ Yield	+0.56%	+1.39%	+0.35%
Max yield increase	+15%	+40%	+75%
Max yield decrease	-20%	-35%	-75%
Average Δ Seeding Rate	+0.015%	-1.03%	+0.4%
Max Seeding Rate increase	+4%	+12%	+80%
Max Seeding Rate decrease	-2%	-15%	-80%

4.2.2 Field D (Linear model)

The map obtained from applying the Nitrogen Rate model (Equation 2) was classified into 9 management zones and plotted as shown in Figures 58 and 59.

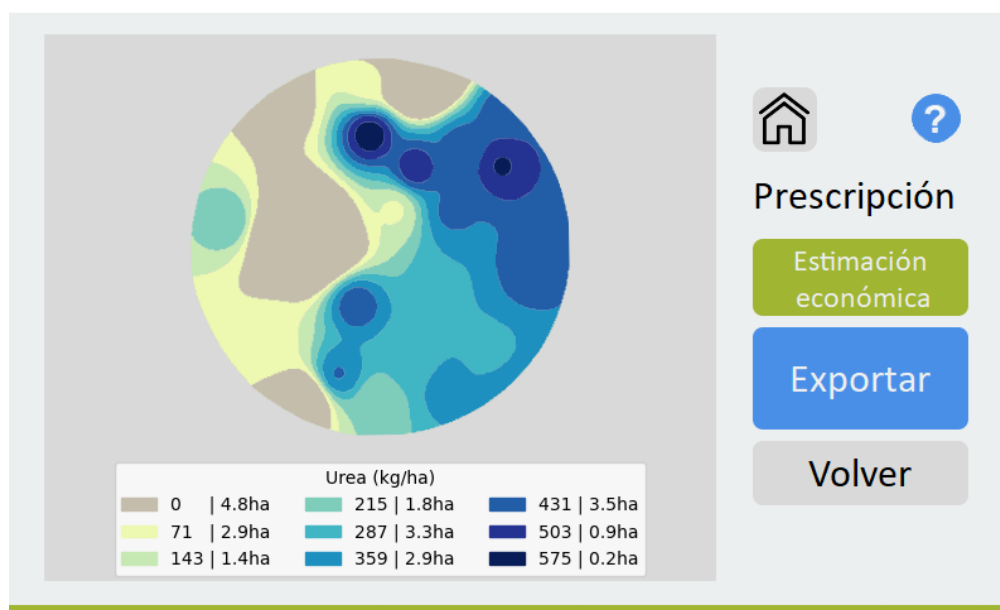


Figure 58. Screenshot of urea prescription in AgStat

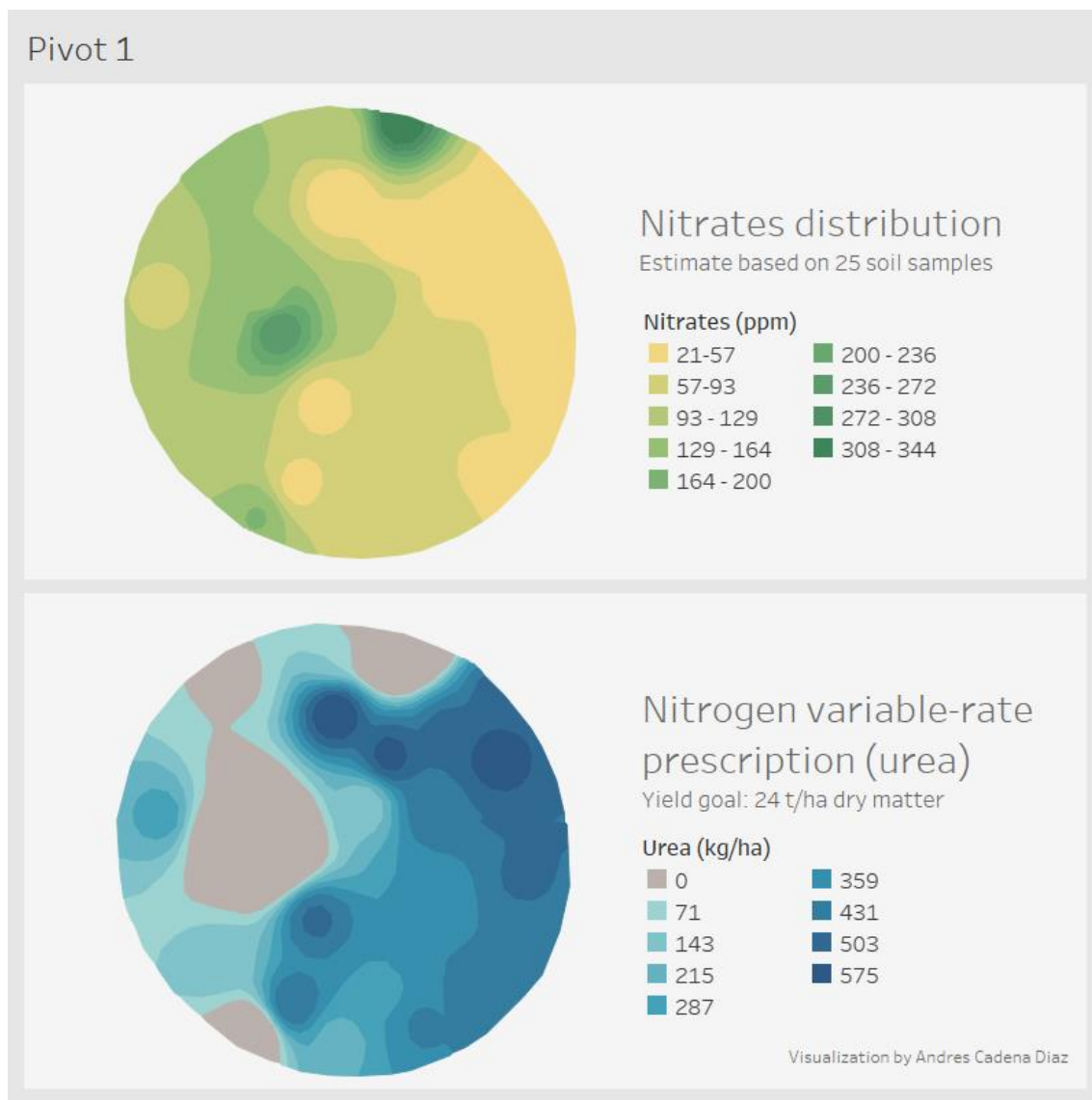


Figure 59. Comparison between nitrates distribution and urea prescription

4.3 AgStat application



Figure 60. AgStat's main menu

AgStat is deployed in a Jetson nano © board from NVIDIA (2021) with a 7" generic compatible touchscreen display. When executed, the main menu is shown (Figure 60) and the user can select between the four options offered

- Analysis:
 - Prescription generation: Guides the user through an easy-to-follow process (Figure 61) to generate a variable rate nitrogen prescription according to the field's attributes and equation 2. The seed recommendation models were not added to the application since their current state is not optimal. The application's system automatically predicts the input rate for each point in the grid, classifies them into nine equally spaced values, and generates the polygons. A preview map with the surface assigned to each rate and an economic estimate is shown to the user to assess the savings achieved by variable rate. Finally, the ESRI shapefile is exported to an external USB drive to transfer it to the variable rate equipment in the field.

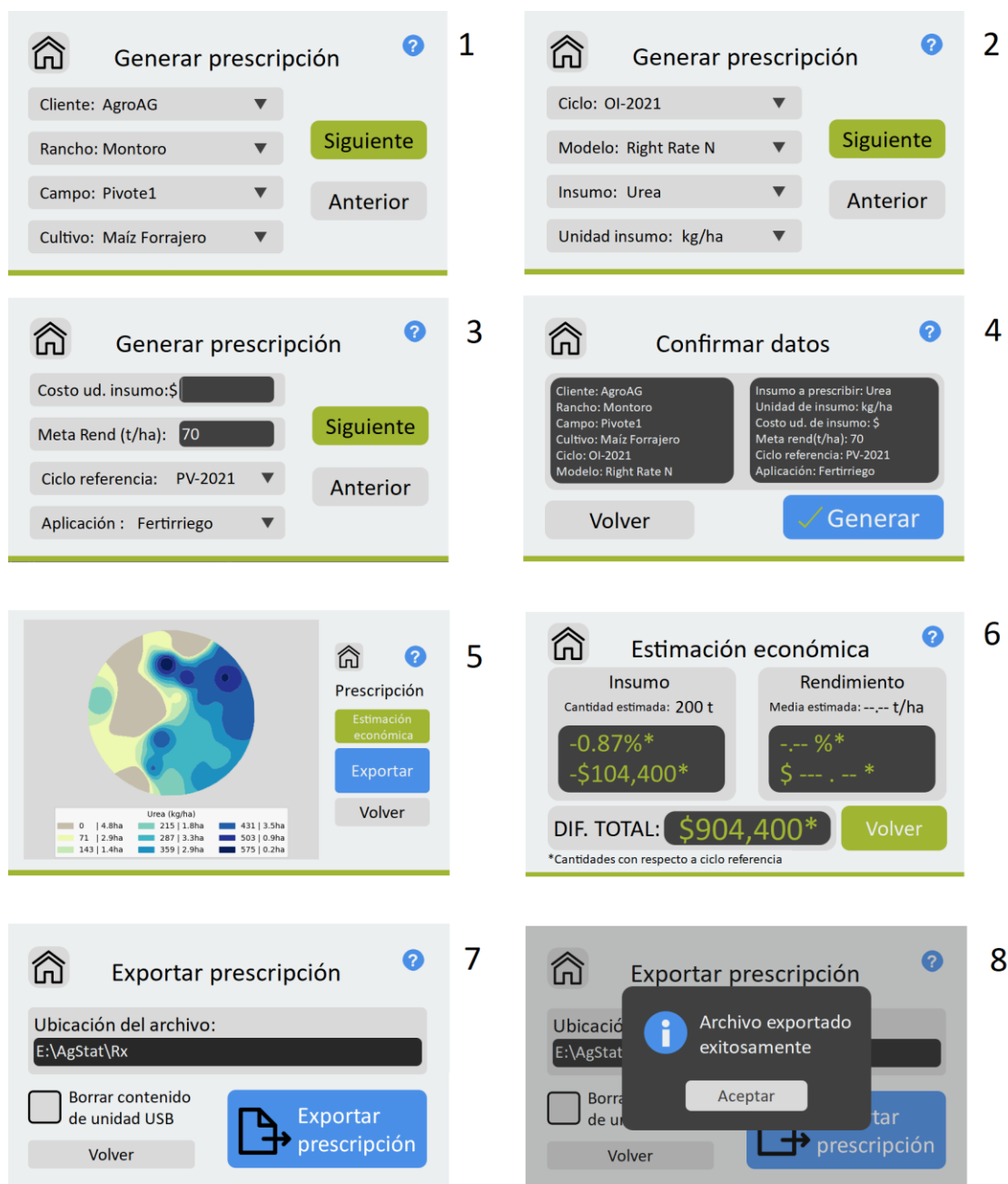


Figure 61. Prescription generation process screens

- Data visualization: Provides users with a friendly interface to visualize their field's data. The plots are generated with the interpolated RG database.
 - Maps: Provides the user with a visualization of the spatial distribution of their field's soil attributes, classified in 5 equally

spaced ranges. This classification facilitates map interpretation, by displaying two high ranges, two low ranges, and a middle range (Figure 62, screen 2).

- Histogram: Single variable distribution plot, the user can select the number of bins from a dropdown menu, ranging from 4 to 200 (Figure 62, screen 3).
- Scatter and regression plots: Displays the interaction between two soil attributes. AgStat offers polynomial regression from orders 1 to 5, which the user selects (Figure 62, screens 4 and 5).



Figure 62. Data visualization screens

- Acquisition: Functionalities in this section (Figure 63) were developed to assist the user to add external databases to the system. In this way, the folder-file structure and required objects to analyze and display the data are created properly.



Figure 63. Data acquisition screens

- Scattered data: Processes the data with the interpolation algorithm and adds the new variable(s) to the database.
- Interpolated data: When a third-party source has already interpolated the data, AgStat only adjusts the data points to match the standard grid and adds them to the database

- Field survey: Provides the user with a simple and intuitive interface (Figure 1d) for georeferenced data acquisition via punctual sampling or on-the-go sampling. The required hardware to acquire the interest variable, like sensors, signal processing circuitry, and the GNSS with RTK correction, are connected via USB and GPIO connectors of the Jetson nano board. The data is then processed by the DAA and added to the database.
- Files: If the user wants to extract specific files or folders from the system, it can be done with this friendly interface (Figure 64), which shows folder structure as a tree hierarchy. The user selects the source and destination folder and makes the transference with the press of a button.

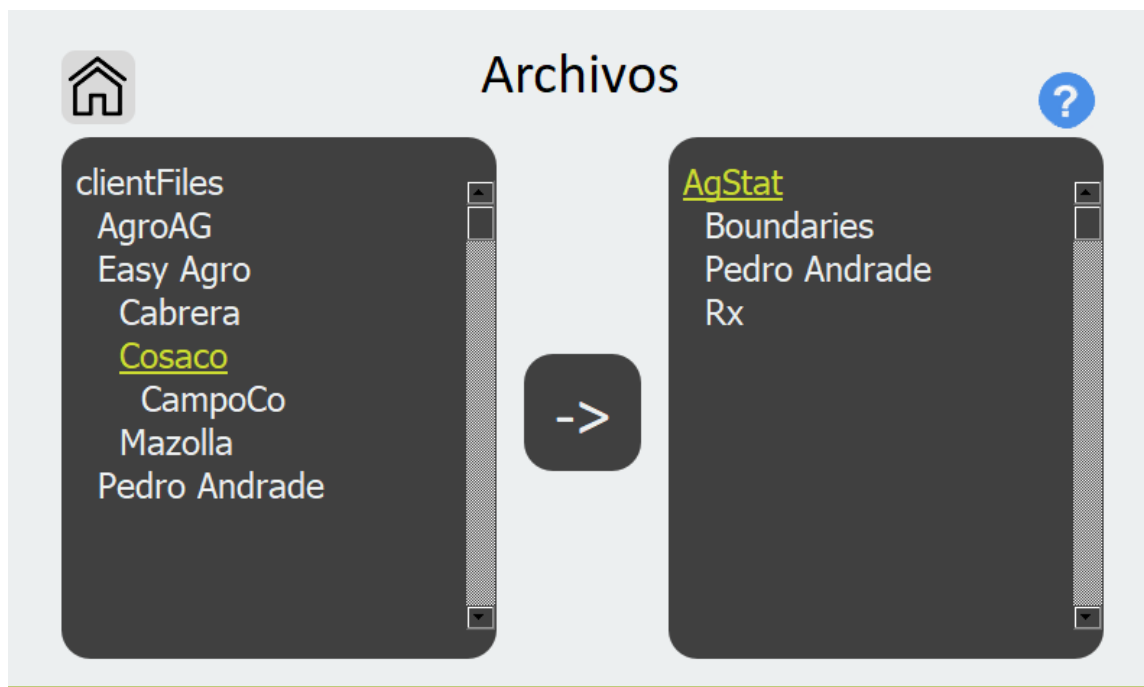


Figure 64. File management screen

For all data operations, the user must specify the client, farm, field, and agricultural cycle of interest. That selection remains the same across the different sections of the application (visualization, prescription generation, survey) unless the user selects another profile, farm, field, or cycle.

- Settings: Although the interface of this section was designed (Figure 65), its functionality is still under development.



Figure 65. Settings section menu

4.4 Discussion

Automatic data processing applications for precision agriculture have been developed in recent years. Some are designed to define management zones and their statistical insights (Paccioretti et al., 2020; Albornoz et al., 2018), and generate thematic maps, like AgDataBox (Borges et al., 2020; Dall’Agnoll et al., 2020; Michelon et al., 2019). In general terms, the previously mentioned applications were designed to transform raw data into meaningful information layers to support farmers' management decisions. However, only GeoFis (Leroux et al., 2018) and the system developed by Albornoz et al. (2018) take the data to the decision layers. GeoFis enables specialists to incorporate their knowledge via expert rules to process data layers and support decision making.

This system allowed the GeoFis’s developers to create a ‘risk of having sub-optimal practices’ map of a vineyard. On the other hand, the application created by Albornoz et al. (2018) automatically generates ESRI shapefiles that contain the polygons of

management zones and offers a friendly interface to enter the input rate value for each zone. The latter methodology has great flexibility since any input can be prescribed using the same interface. However, the user needs to have specific technical knowledge or have access to an expert to decide the proper input rates. A general overview of AgStat's functionalities compared to similar applications is shown in Table 8. The use of ANNs applied to agriculture has been successfully implemented in many stages of agricultural production including production forecasting, pest and disease detection, and harvest quality control (Kujawa & Niedvala, 2021).

Table 8. AgStat's functionalities compared to similar applications'

	Leroux et al. (2018)	Albornoz et al. (2018)	Michelon et al. (2019)	Paccioreti et al. (2020)	AgStat (this work)
Field surveying					
Automated global outlier's removal					
Automated spatial outliers' removal					
Thematic mapping					
Statistical plotting					
Spatial Clustering (ie. Fuzzy K means)					
Management zone generation					
Shapefile generation					
Expert fuzzy rules					
Automated rate assignation to shapefile					

Works using ANNs to recommend seeding rates were not found, however Ehsani et al. (2005) used seeding rates as an independent variable to predict yield. Dahikar et al. (2014) is another good example of agricultural yield prediction based on soil properties. In this particular work, the best results were obtained with a fixed nitrogen model adapted from Stanford (1973) in contrast of the ANNs' poor results. ANNs have the advantage learning dynamically, improving by processing new datasets and have been successfully used in many research and practical areas. The low amount and accuracy of the data from the fields A, B, and C are possibly the root causes of the ANN results in this work. However,

the same algorithms used in this work can process a larger volume of better quality data in the future, improving results considerably and putting ANNs at direct service of the PA users.

This work proposes a fully automated prescription-generation processing that decides the appropriate input rate for each site, based on the needs of the crop and the spatial distribution of the variable of interest (i.e., nitrogen). The AgStat's user does not need to be an expert to generate VR prescriptions. Instead, the user only enters the necessary information into the system and lets the application predict the rates, classify them into management zones, and export a ready-to-work shapefile. However, this approach is limited by the models and embodied knowledge available in the system. A nitrogen-only macronutrient prescription was adequate for this case study since phosphorus and potassium application was not advised due to high concentrations of those elements in the field. An expert in plant nutrition has supervised both models used for this work. They were approved since they followed the same logic as the expert's recommendation. Nevertheless, the actual effect on crop yield due to the prescriptions generated by the system needs to be quantified to validate the models fully. Additionally, the adequate execution of the prescriptions would depend on how precisely the variable-rate machinery responds to rate changes. With further development, prescription maps tailored to specific brands or models of VRT machinery can be made by storing the rate-change speeds of each model into the application. In this manner, the prescription would be executed in the field with improved precision.

Although this work is aimed to be universal, for now the prescription generation scope it is limited to the models that were developed. The application resulting of this work needs further development and testing to optimize its processes and stability before launching it to the market. At the present stage, the developed PGA outputs single variable prescriptions (nitrogen). This approach has the advantage of managing each input precisely. However, it is limited when more generalized management zones want to be created (i.e., Fuzzy K-Means) to apply more than one input in one machinery pass (i.e., variable-rate NPK fertilization). Therefore, generalized management-zone models

(Paccioretti et al., 2020; Albornoz et al., 2018; Fridgen et al., 2003) are planned to be implemented, along with functionalities like spatial-outliers removal.

With future research, numerous models can be developed and tested to predict optimum rates for a variety of inputs for different crops and conditions, including NPK and micronutrient formulas. Adding those models to the system would favor the adoption of information-intensive technology since it would not be very knowledge-demanding. The average PA user could take advantage of such tools with simple training.

5 CONCLUSIONS

To address the lack fully automated prescription generation tools in PA, AgStat was designed and developed. The product was created as an integrated PA tool designed for intuitive and automated data processing, visualization, acquisition, and prescription generation for non-expert users. This tool can perform tasks from georeferenced data survey in the field, data cleaning, interpolation, to automatically apply expert knowledge to generate agronomic prescriptions. The system embodies the GIS operation skills and technical criteria a PA specialist needs to perform some of PA's fundamental operations.

During the development process, corn yield was modeled using both ANNs and regression analysis. ANNs were trained to prescribe corn seeding rates and a fixed model was used to prescribe nitrogen rates. Artificial Neural Networks were used to predict yield and optimum seeding rate for three fields. Yield predictions were adequate but seeding rate predictions were not due to datapoints with undesired results compensating for the desired ones. A larger volume of better data is needed to obtain a model that will truly optimize the seeding rate via a Deep Learning model. ANNs depend on a large amount of good quality data to produce adequate results, which unfortunately lacked in this case. But the main algorithms were designed properly and can be trained with better data in the future. In this way, the application will put the processing power of ANNs to the service of PA users. On the other hand, nitrogen recommendations obtained with the fixed model were appropriate and proved the adequate functionality of the application and internal algorithms, resulting in an easy-to-use and powerful tool. Farmers, advisors, and researchers can benefit from this application which can assist them from data collection to generating a ready-to-use prescription file.

The reduced costs due to use of open-source software in AgStat's development presents an advantage over commercial equipment, which can favor its adoption among farmers. Future research, model testing, and development are needed to improve the system's capabilities and add more advanced functionalities into an intuitive user experience. The author hope that products like AgStat and similar applications improve IIT adoption rates by PA users, leading to more sustainable and productive agricultural systems.

6 REFERENCES

- Abdullahi, H., & Zubair, O. (2017). Advances of image processing in precision agriculture: Using deep learning convolution neural network for soil nutrient classification. *Journal of Multidisciplinary Engineering Science and Technology*, 4(17) 7981-7987.
- Agatonovic-Kustrin, S., & Beresford, R. (2000). Basic concepts of artificial neural network (ANN) modeling and its application in pharmaceutical research. *Journal of Pharmaceutical and Biomedical Analysis*, 22(5), 717–727. [https://doi.org/10.1016/s0731-7085\(99\)00272-1](https://doi.org/10.1016/s0731-7085(99)00272-1)
- Aizenman, H. (2018, January 8). *Working with Tabular Geospatial Data using Pandas & Geopandas*. 98th American Meteorological Society Annual Meeting. <https://ams.confex.com/ams/98Annual/meetingapp.cgi/Paper/341756>
- Albornoz, E. M., Kemerer, A. C., Galarza, R., Mastaglia, N., Melchiori, R., & Martínez, C. E. (2018). Development and evaluation of an automatic software for management zone delineation. *Precision Agriculture*, 19(3), 463–476. <https://doi.org/10.1007/s11119-017-9530-9>
- Ali, T. A. A., Choksi, V., & Potdar, M. B. (2018). Precision agriculture monitoring system using green internet of things (G-IoT). *2018 2nd International Conference on Trends in Electronics and Informatics (ICOEI)*, 481–487. <https://doi.org/10.1109/ICOEI.2018.8553866>
- Appel, M., Lahn, F., Buytaert, W., & Pebesma, E. (2018). Open and scalable analytics of large Earth observation datasets: From scenes to multidimensional arrays using SciDB and GDAL. *ISPRS Journal of Photogrammetry and Remote Sensing*, 138, 47–56. <https://doi.org/10.1016/j.isprsjprs.2018.01.014>
- Arrabales, R. (2016). *Deep Learning: Qué es y por qué va a ser una tecnología clave en el futuro de la inteligencia artificial*. Xataka. <https://www.xataka.com/robotica-e-ia/deep-learning-que-es-y-por-que-va-a-ser-una-tecnologia-clave-en-el-futuro-de-la-inteligencia-artificial>
- Barnes, A. P., Soto, I., Eory, V., Beck, B., Balafoutis, A., Sánchez, B., Vangeyte, J., Fountas, S., van der Wal, T., & Gómez-Barbero, M. (2019). Exploring the adoption of precision agricultural technologies: A cross regional study of EU farmers. *Land Use Policy*, 80, 163–174. <https://doi.org/10.1016/j.landusepol.2018.10.004>
- Bauer, M., & Garland, M. (2019). Legate NumPy: Accelerated and distributed array computing. *Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis*, 1–23. <https://doi.org/10.1145/3295500.3356175>
- Beghin, J. C., & Nogueira, L. (2021). A perfect storm in fertilizer markets. *AgEcon Search*. <https://doi.org/10.22004/ag.econ.316020>

- Bergström, L., Bowman, B. t., & Sims, J. t. (2005). Definition of sustainable and unsustainable issues in nutrient management of modern agriculture. *Soil Use and Management*, 21(1), 76–81. <https://doi.org/10.1111/j.1475-2743.2005.tb00111.x>
- Bhakta, I., Phadikar, S., & Majumder, K. (2019). State-of-the-art technologies in precision agriculture: A systematic review. *Journal of the Science of Food and Agriculture*, 99(11), 4878–4888. <https://doi.org/10.1002/jsfa.9693>
- Bisong, E. (2019). Matplotlib and Seaborn. In E. Bisong (Ed.), *Building Machine Learning and Deep Learning Models on Google Cloud Platform: A Comprehensive Guide for Beginners* (pp. 151–165). Apress. https://doi.org/10.1007/978-1-4842-4470-8_12
- Borges, L. F., Bazzi, C. L., Souza, E. G. de, Magalhães, P. S. G., & Michelin, G. K. (2020). Web software to create thematic maps for precision agriculture. *Pesquisa Agropecuária Brasileira*, 55, e00735. <https://doi.org/10.1590/s1678-3921.pab2020.v55.00735>
- Breunig, F. M., Galvão, L. S., Dalagnol, R., Dauve, C. E., Parraga, A., Santi, A. L., Della Flora, D. P., & Chen, S. (2020). Delineation of management zones in agricultural fields using cover–crop biomass estimates from PlanetScope data. *International Journal of Applied Earth Observation and Geoinformation*, 85, 102004. <https://doi.org/10.1016/j.jag.2019.102004>
- Calvo, D. (2018, December 9). Red Neuronal Recurrente—RNN. *Diego Calvo*. <https://www.diegocalvo.es/red-neuronal-recurrente/>
- Campos, J., Gallart, M., Llop, J., Ortega, P., Salcedo, R., & Gil, E. (2020). On-Farm Evaluation of Prescription Map-Based Variable Rate Application of Pesticides in Vineyards. *Agronomy*, 10(1), 102. <https://doi.org/10.3390/agronomy10010102>
- Chen, D. Y. (2017). *Pandas for Everyone: Python Data Analysis*. Addison-Wesley Professional.
- Coelho, A. L. F., Querioz, D. M., Valente, D. S. M., & Pinto, F. A. C. (2020). *Development of a Variable-Rate Controller for a Low-Cost Precision Planter*. <https://doi.org/10.13031/aea.13784>
- Comisión Nacional para el Conocimiento y Uso de la Biodiversidad. (2006). Elementos para la determinación de centros de origen y centros de diversidad en general y el caso específico de la liberación experimental de maíz transgénico al ambiente en México. México, D.F. https://www.biodiversidad.gob.mx/media/1/genes/files/Elementos_2006.pdf (last accessed 18 May 2022).
- Consejo Nacional de Ciencia y Tecnología. (2014). Agenda de Innovación de la Región Norte.
- Consejo Nacional de Ciencia y Tecnología. (2014). Agenda de Innovación de la Región Sur Sureste.

- Corwin, D. L., & Lesch, S. M. (2003). Application of Soil Electrical Conductivity to Precision Agriculture. *Agronomy Journal*, 95(3), 455–471. <https://doi.org/10.2134/agronj2003.4550>
- Dahikar, M. S. S., S, D., & Rode, E. V. (2014). *Agricultural Crop Yield Prediction Using Artificial Neural Network Approach*.
- Dall’Agnol, R. W., Michelon, G. K., Bazzi, C. L., Magalhães, P. S. G., Souza, E. G. de, Betzek, N. M., & Sobjak, R. (2020). Web applications for spatial analyses and thematic map generation. *Computers and Electronics in Agriculture*, 172, 105374. <https://doi.org/10.1016/j.compag.2020.105374>
- Darejeh, A., & Singh, D. (2013). A review on user interface design principles to increase software usability for users with less computer literacy. *Journal of Computer Science*, 9(11), 1443–1450. <https://doi.org/10.3844/jcssp.2013.1443.1450>
- DeLay, N. D., Thompson, N. M., & Mintert, J. R. (2022). Precision agriculture technology adoption and technical efficiency. *Journal of Agricultural Economics*, 73(1), 195–219. <https://doi.org/10.1111/1477-9552.12440>
- Demiris, G., Rantz, M. J., Aud, M. A., Marek, K. D., Tyrer, H. W., Skubic, M., & Hussam, A. A. (2004). Older adults’ attitudes towards and perceptions of ‘smart home’ technologies: A pilot study. *Medical Informatics and the Internet in Medicine*, 29(2), 87–94. <https://doi.org/10.1080/14639230410001684387>
- Diaz, E., & Lozano, G. (2019). *Retos y tendencias en el sector agropecuario en México*. https://www.ey.com/es_mx/consumer-products-retail/retos-y-tendencias-en-el-sector-agropecuario-en-mexico
- Ehsani, M. R., Durairaj, C. D., Woods, S., & Sullivan, M. (2005). *Potential Application of Electrical Conductivity (EC) Map for Variable Rate Seeding*. <https://ecommons.cornell.edu/handle/1813/10453>
- Erickson, B., Fausti, S., Clay, D., & Clay, S. (2018). Knowledge, Skills, and Abilities in the Precision Agriculture Workforce: An Industry Survey. *Natural Sciences Education*, 47(1), 180010. <https://doi.org/10.4195/nse2018.04.0010>
- Escobar-Oyervides, J.C. (2003). Análisis de la adopción de la tecnología Agricultura de Precisión en el mercado de producción de granos comercial en México. Unpublished master’s thesis, Instituto Tecnológico y de Estudios Superiores de Monterrey (pp. 1-39).
- ESRI. (1998). *ESRI Shapefile* [Web page]. <https://www.loc.gov/preservation/digital/formats/fdd/fdd000280.shtml>
- Evenson, R. E., & Gollin, D. (2003). Assessing the Impact of the Green Revolution, 1960 to 2000. *Science*, 300(5620), 758–762. <https://doi.org/10.1126/science.1078710>
- The Food and Agriculture Organization. (2018). *México rural del Siglo XXI*. <https://www.fao.org/family-farming/detail/es/c/1147129/>

- Fertilab (2021). Soil analysis laboratory. <https://www.fertilab.com.mx/>
- García-Lara, S., & Serna-Saldivar, S. O. (2019). Corn History and Culture. In *Corn* (pp. 1–18). Elsevier. <https://doi.org/10.1016/B978-0-12-811971-6.00001-2>
- GDAL/OGR contributors. (2022). *GDAL/OGR} Geospatial Data Abstraction software Library*. Open Source Geospatial Foundation. <https://gdal.org/>
- Ghosh, A., Sufian, A., Sultana, F., Chakrabarti, A., & De, D. (2020). Fundamental Concepts of Convolutional Neural Network. In V. E. Balas, R. Kumar, & R. Srivastava (Eds.), *Recent Trends and Advances in Artificial Intelligence and Internet of Things* (pp. 519–567). Springer International Publishing. https://doi.org/10.1007/978-3-030-32644-9_36
- Günay, A., Arefi, H., & Hahn, M. (2007). *True Orthophoto production using Lidar data*.
- Harris, C. R., Millman, K. J., van der Walt, S. J., Gommers, R., Virtanen, P., Cournapeau, D., Wieser, E., Taylor, J., Berg, S., Smith, N. J., Kern, R., Picus, M., Hoyer, S., van Kerkwijk, M. H., Brett, M., Haldane, A., del Río, J. F., Wiebe, M., Peterson, P., ... Oliphant, T. E. (2020). Array programming with NumPy. *Nature*, 585(7825), 357–362. <https://doi.org/10.1038/s41586-020-2649-2>
- Hu, L., Di, L., & Yue, P. (2021). Design and Implementation of Geospatial Data Services for Agriculture. In L. Di & B. Üstündağ (Eds.), *Agro-geoinformatics: Theory and Practice* (pp. 385–403). Springer International Publishing. https://doi.org/10.1007/978-3-030-66387-2_18
- Hunter, J. D. (2007). Matplotlib: A 2D Graphics Environment. *Computing in Science Engineering*, 9(3), 90–95. <https://doi.org/10.1109/MCSE.2007.55>
- Instituto Nacional de Estadística y Geografía, Secretaría de Agricultura y Desarrollo Rural. (2020). Encuesta Nacional Agropecuaria 2019
- Instituto Nacional de Estadística y Geografía, Secretaría de Agricultura y Desarrollo Rural. (2019). Encuesta Nacional Agropecuaria 2017.
- Isioye, O. (2013). Precision Agriculture: Applicability and Opportunities for Nigerian Agriculture. *Middle East Journal of Scientific Research*, 13, 1230–1237. <https://doi.org/10.5829/idosi.mejsr.2013.13.9.1004>
- Janiesch, C., Zschech, P., & Heinrich, K. (2021). Machine learning and deep learning. *Electronic Markets*, 31(3), 685–695. <https://doi.org/10.1007/s12525-021-00475-2>
- Jordahl, K., Bossche, J. V. den, Fleischmann, M., Wasserman, J., McBride, J., Gerard, J., Tratner, J., Perry, M., Badaracco, A. G., Farmer, C., Hjelle, G. A., Snow, A. D., Cochran, M., Gillies, S., Culbertson, L., Bartos, M., Eubank, N., maxalbert, Bilogur, A., ... Leblanc,

- F. (2020). *geopandas/geopandas: V0.8.1*. Zenodo. <https://doi.org/10.5281/zenodo.3946761>
- Jüstel, A., Endlein Correira, A., Wellmann, F., & Pischke, M. (2021). *GemGIS - GemPy Geographic: Open-Source Spatial Data Processing for Geological Modeling*. EGU21-4613.
- Kale, S. S., & Patil, P. S. (2019). A Machine Learning Approach to Predict Crop Yield and Success Rate. *2019 IEEE Pune Section International Conference (PuneCon)*, 1–5. <https://doi.org/10.1109/PuneCon46936.2019.9105741>
- Kamilaris, A., & Prenafeta-Boldú, F. X. (2018). Deep learning in agriculture: A survey. *Computers and Electronics in Agriculture*, 147, 70–90. <https://doi.org/10.1016/j.compag.2018.02.016>
- Khanna, A., & Kaur, S. (2019). Evolution of Internet of Things (IoT) and its significant impact in the field of Precision Agriculture. *Computers and Electronics in Agriculture*, 157, 218–231. <https://doi.org/10.1016/j.compag.2018.12.039>
- Kim, S. Y., & McFadden, Ed. (2020). Using Established UX Design Techniques and Visual Enhancements to Redesign an Enterprise Mobile App and Improve Employee Productivity and Engagement. In T. Ahram & C. Falcão (Eds.), *Advances in Usability and User Experience* (pp. 169–176). Springer International Publishing. https://doi.org/10.1007/978-3-030-19135-1_17
- Lemenkova, P. (2019). PROCESSING OCEANOGRAPHIC DATA BY PYTHON LIBRARIES NUMPY, SCIPY AND PANDAS. *Aquatic Research*, 73–91. <https://doi.org/10.3153/AR19009>
- Lemenkova, P. (2020a). *Python Libraries Matplotlib, Seaborn and Pandas for Visualization Geo-spatial Datasets Generated by QGIS* (SSRN Scholarly Paper ID 3699706). Social Science Research Network. <https://papers.ssrn.com/abstract=3699706>
- Lemenkova, P. (2020b). *NOAA Marine Geophysical Data and a GEBCO Grid for the Topographical Analysis of Japanese Archipelago by Means of GRASS GIS and GDAL Library* (SSRN Scholarly Paper ID 3732734). Social Science Research Network. <https://papers.ssrn.com/abstract=3732734>
- Leroux, C., Jones, H., Pichon, L., Guillaume, S., Lamour, J., Taylor, J., Naud, O., Crestey, T., Lablee, J.-L., & Tisseyre, B. (2018). GeoFIS: An Open Source, Decision-Support Tool for Precision Agriculture Data. *Agriculture*, 8(6), 73. <https://doi.org/10.3390/agriculture8060073>
- Ma, X., Longley, I., Salmond, J., & Gao, J. (2020). PyLUR: Efficient software for land use regression modeling the spatial distribution of air pollutants using GDAL/OGR library in Python. *Frontiers of Environmental Science & Engineering*, 14(3), 44. <https://doi.org/10.1007/s11783-020-1221-5>
- Maresma, A., Chamberlain, L., Tagarakis, A., Kharel, T., Godwin, G., Czymbek, K. J., Shields, E., & Ketterings, Q. M. (2020). Accuracy of NDVI-derived corn yield predictions

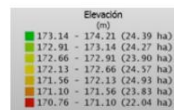
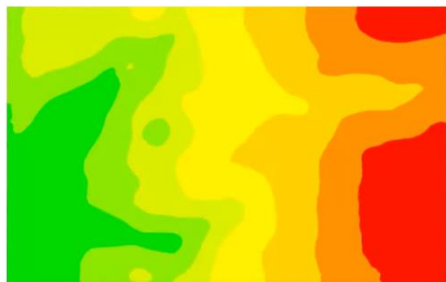
- is impacted by time of sensing. *Computers and Electronics in Agriculture*, 169, 105236. <https://doi.org/10.1016/j.compag.2020.105236>
- Michelon, G. K., Bazzi, C. L., Upadhyaya, S., de Souza, E. G., Magalhães, P. S. G., Borges, L. F., Schenatto, K., Sobjak, R., Gavioli, A., & Betzek, N. M. (2019). Software AgDataBox-Map to precision agriculture management. *SoftwareX*, 10, 100320. <https://doi.org/10.1016/j.softx.2019.100320>
- Millan, S., Moral, F. H., Prieto, M. H., Perez-Rodriguez, J. M., & Campillo, C. (2019). Mapping Soil Properties and Delineating Management Zones Based on Electrical Conductivity in a Hedgerow Olive Grove. *Transactions of the ASABE*, 62(3), 749–760.
- Nishant, P. S., Sai Venkat, P., Avinash, B. L., & Jabber, B. (2020). Crop Yield Prediction based on Indian Agriculture using Machine Learning. *2020 International Conference for Emerging Technology (INCET)*, 1–4. <https://doi.org/10.1109/INCET49848.2020.9154036>
- Nugraha, H. K., & Dhewanto, W. (2019). Exploring of Factors Related to the Technology Adoption in Purchasing System of Product Through Mobile Application. *International Journal of Business and Administrative Studies*, 5(2). <https://doi.org/10.20469/ijbas.5.10005-2>
- NVIDIA. (2021). Jetson Nano Developer Kit. *NVIDIA Developer*. <https://developer.nvidia.com/embedded/jetson-nano-developer-kit>
- Ohana-Levi, N., Ben-Gal, A., Peeters, A., Termin, D., Linker, R., Baram, S., Raveh, E., & Paz-Kagan, T. (2021). A comparison between spatial clustering models for determining N-fertilization management zones in orchards. *Precision Agriculture*, 22(1), 99–123. <https://doi.org/10.1007/s11119-020-09731-5>
- Onyango, C. M., Nyaga, J. M., Wetterlind, J., Söderström, M., & Piikki, K. (2021). Precision Agriculture for Resource Use Efficiency in Smallholder Farming Systems in Sub-Saharan Africa: A Systematic Review. *Sustainability*, 13(3), 1158. <https://doi.org/10.3390/su13031158>
- Paccioretti, P., Córdoba, M., & Balzarini, M. (2020). FastMapping: Software to create field maps and identify management zones in precision agriculture. *Computers and Electronics in Agriculture*, 175, 105556. <https://doi.org/10.1016/j.compag.2020.105556>
- Parrot (2018). Sequoia: multispectral sensor for drones. *Parrot*. <https://www.parrot.com/en/shop/accessories-spare-parts/other-drones/sequoia>
- Rahmayanti, P., Nurhadryani, Y., & Hermadi, I. (2020). Analysis and Design of Agricultural Marketplace Using User Experience. *2020 International Conference on Informatics, Multimedia, Cyber and Information System (ICIMCIS)*, 207–212. <https://doi.org/10.1109/ICIMCIS51567.2020.9354292>

- Reback, J., jbrockmendel, McKinney, W., Bossche, J. V. den, Augspurger, T., Cloud, P., Hawkins, S., Roeschke, M., gfyong, Sinhrks, Klein, A., Hoefler, P., Petersen, T., Tratner, J., She, C., Ayd, W., Naveh, S., Garcia, M., Darbyshire, J. H. M., ... Seibold, S. (2022). *pandas-dev/pandas: Pandas 1.4.0rc0*. Zenodo. <https://doi.org/10.5281/zenodo.5824773>
- Rouse, J. W., Haas, R. H., Schell, J. A., & Deering, D. W. (1974, January 1). *Monitoring vegetation systems in the Great Plains with ERTS*. <https://ntrs.nasa.gov/citations/19740022614>
- Rulaningtyas, R., Suksmono, A. B., Mengko, T. L. R., & Saptawati, P. (2017). Colour segmentation of multi variants tuberculosis sputum images using self organizing map. *Journal of Physics: Conference Series*, 853, 012012. <https://doi.org/10.1088/1742-6596/853/1/012012>
- Sawyer, J. E. (1994). Concepts of Variable Rate Technology with Considerations for Fertilizer Application. *Journal of Production Agriculture*, 7(2), 195–201. <https://doi.org/10.2134/jpa1994.0195>
- Secretaría de Agricultura, Ganadería, Desarrollo Rural, Pesca y Alimentación. (2016). *Planeación Agrícola Nacional 2016-2030*.
- Secretaría de Agricultura, Ganadería, Desarrollo Rural, Pesca y Alimentación. (2017). *Planeación Agrícola Nacional 2017-2030*.
- Secretaría de Agricultura, Ganadería, Desarrollo Rural, Pesca y Alimentación y Organización de las Naciones Unidas para la Alimentación y la Agricultura. (2014). *Diagnóstico del sector rural y pesquero 2012*.
- SenseFly (2018). eBee: Fixed-wing Mapping Drone for Agriculture. *SenseFly*. <https://www.sensefly.com/drone/ebec-ag/>
- Servicio de Información Agroalimentaria y Pesquera. (2021). *Panorama Agroalimentario 2020*.
- Servicio de Información Agroalimentaria y Pesquera. (2021). *Estadística de la producción agrícola 2020*.
- Setianto, A., & Triandini, T. (2015). Comparison of kriging and inverse distance weighted (idw) interpolation methods in lineament extraction and analysis. *Journal of Applied Geology*, 5(1), Article 1. <https://doi.org/10.22146/jag.7204>
- Shahbandeh, M. (2022). *Corn in the U.S*. Statista. <https://www.statista.com/study/11479/corn-in-the-us-statista-dossier/>
- Smart Panel. (2018, April 10). ¿Qué es el Deep Learning? *SmartPanel*. <https://www.smartpanel.com/que-es-deep-learning/>

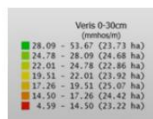
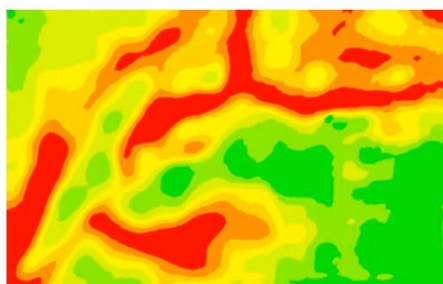
- Songsupakit, K., Phisanbut, N., Watanapongse, P., Piamsa-nga, P., Koonawootrittriron, S., Suwanasopee, T., & Jattawa, D. (2019). UI/UX-centric Design of In-the-Field Agricultural Data Acquisition System. *2019 23rd International Computer Science and Engineering Conference (ICSEC)*, 390–393. <https://doi.org/10.1109/ICSEC47112.2019.8974767>
- Swamy, K. R., & Venkataprasad, G. (2018). DATA Analytics of Agriculture Production, Wages and Income in Rural Areas of India using Big Data and Python Matplot Lib. *International Journal on Future Revolution in Computer Science & Communication Engineering*, 4(1), 193–199.
- Talano. (2019). *Selección de genética por ambiente—17° Curso Internacional de Agricultura y Ganadería de Precisión*. <https://www.youtube.com/watch?v=Jaf3KT-XyXM>
- Thompson, J., & Blank, S. (2000). Harvest mechanization helps agriculture remain competitive. *California Agriculture*, 54(3), 51–56.
- Tibco. (2022). *What is a Neural Network?* TIBCO Software. <https://www.tibco.com/reference-center/what-is-a-neural-network>
- USDA. (2022). *Corn: Yield by Year, US*. https://www.nass.usda.gov/Charts_and_Maps/Field_Crops/cornyield.php
- van Zanden, J. L. (1991). The First Green Revolution: The Growth of Production and Productivity in European Agriculture, 1870-1914. *The Economic History Review*, 44(2), 215–239. <https://doi.org/10.2307/2598294>
- Waskom, M. L. (2021). seaborn: Statistical data visualization. *Journal of Open Source Software*, 6(60), 3021. <https://doi.org/10.21105/joss.03021>
- Willman, J. (2021). Overview of PyQt5. In J. Willman (Ed.), *Modern PyQt: Create GUI Applications for Project Management, Computer Vision, and Data Analysis* (pp. 1–42). Apress. https://doi.org/10.1007/978-1-4842-6603-8_1

7 ANNEXES

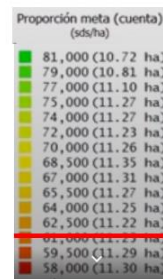
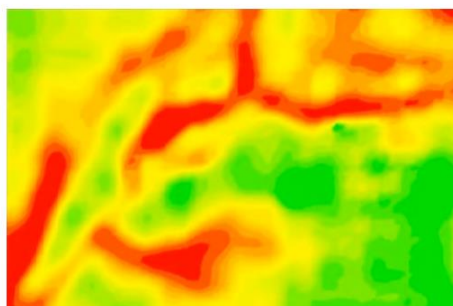
Elevation



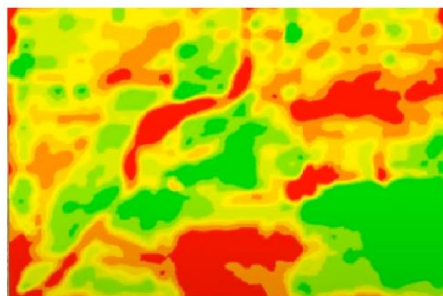
ECa



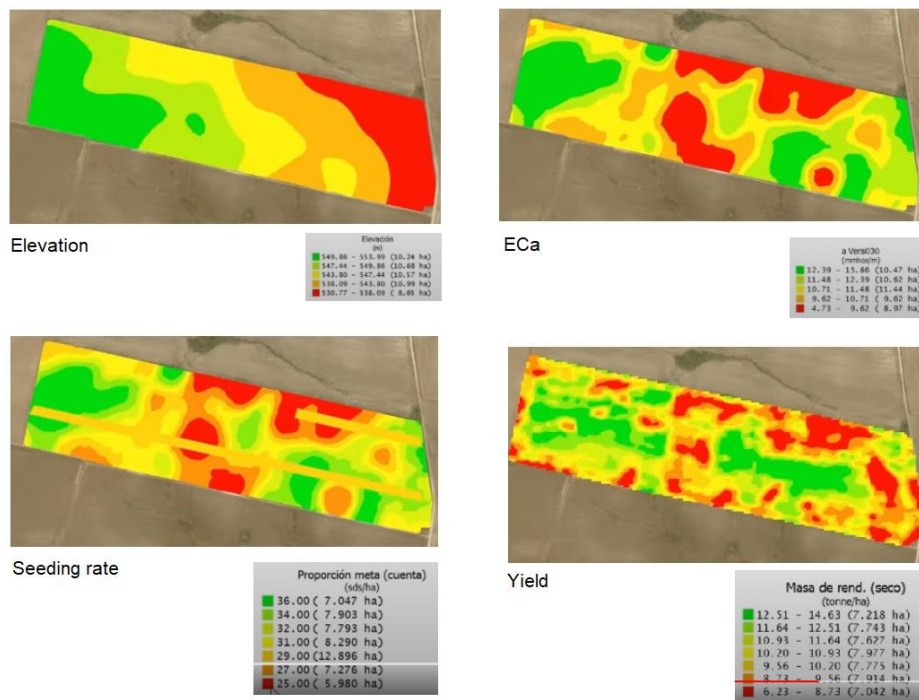
Seeding rate



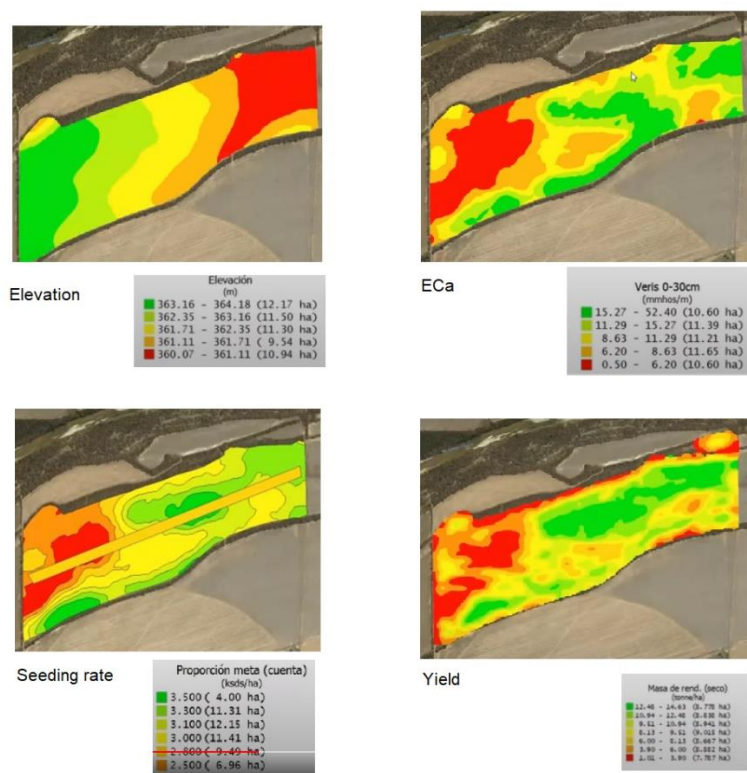
Yield



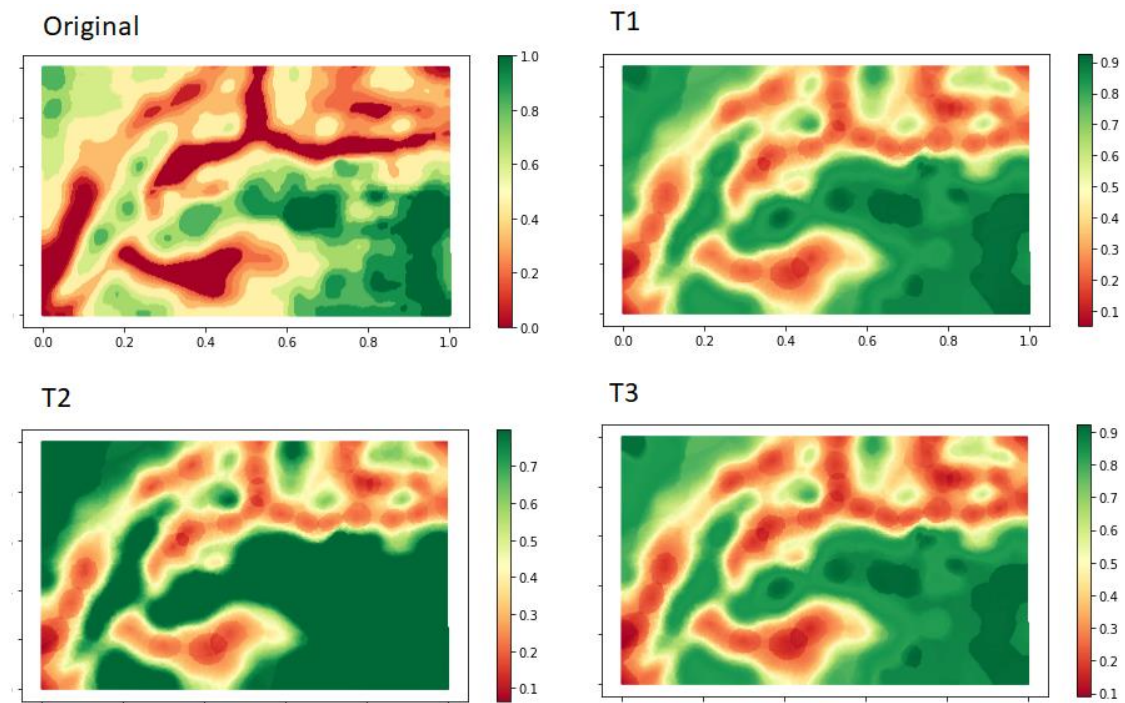
Annex 1. Original imagery (Field A)



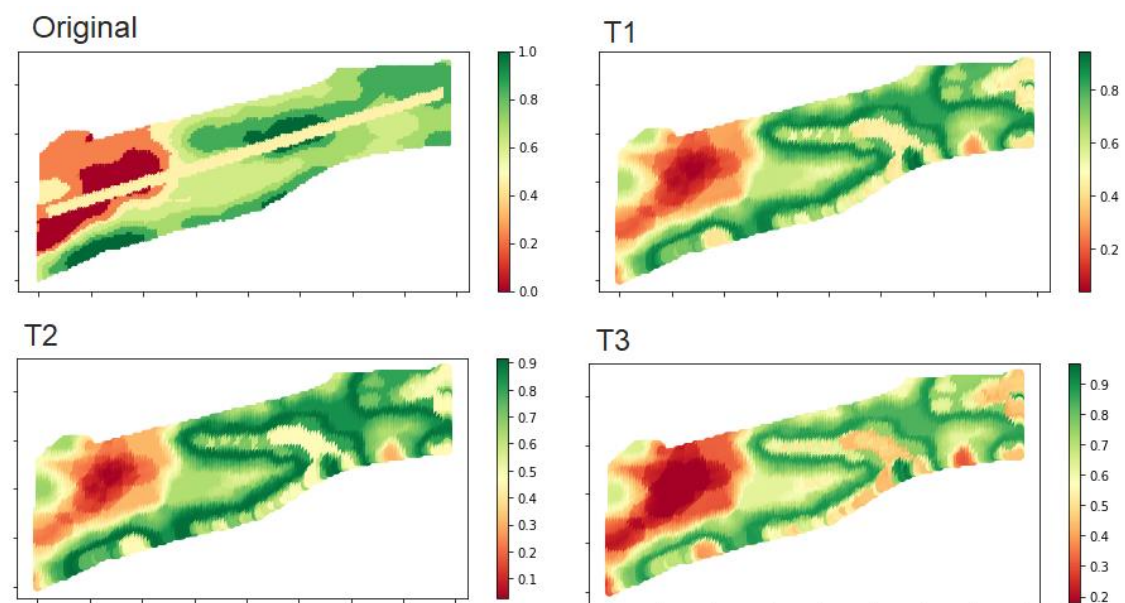
Annex 2. Original Imagery (Field B)



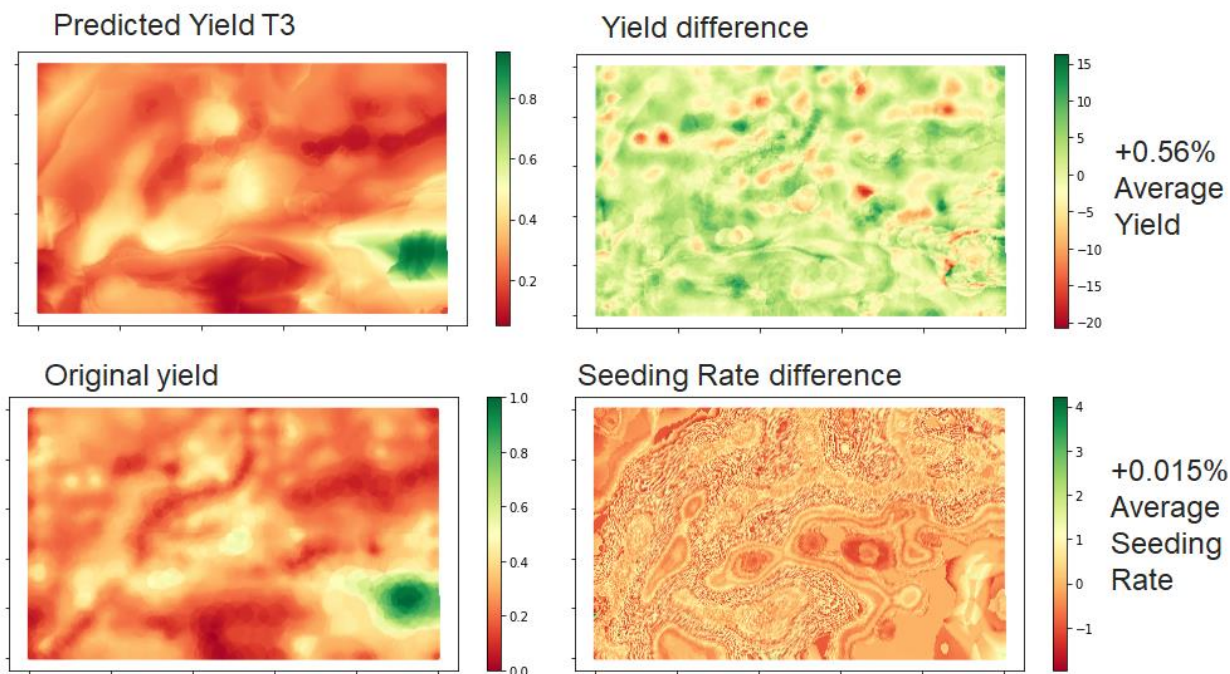
Annex 3. Original imagery (Field C)



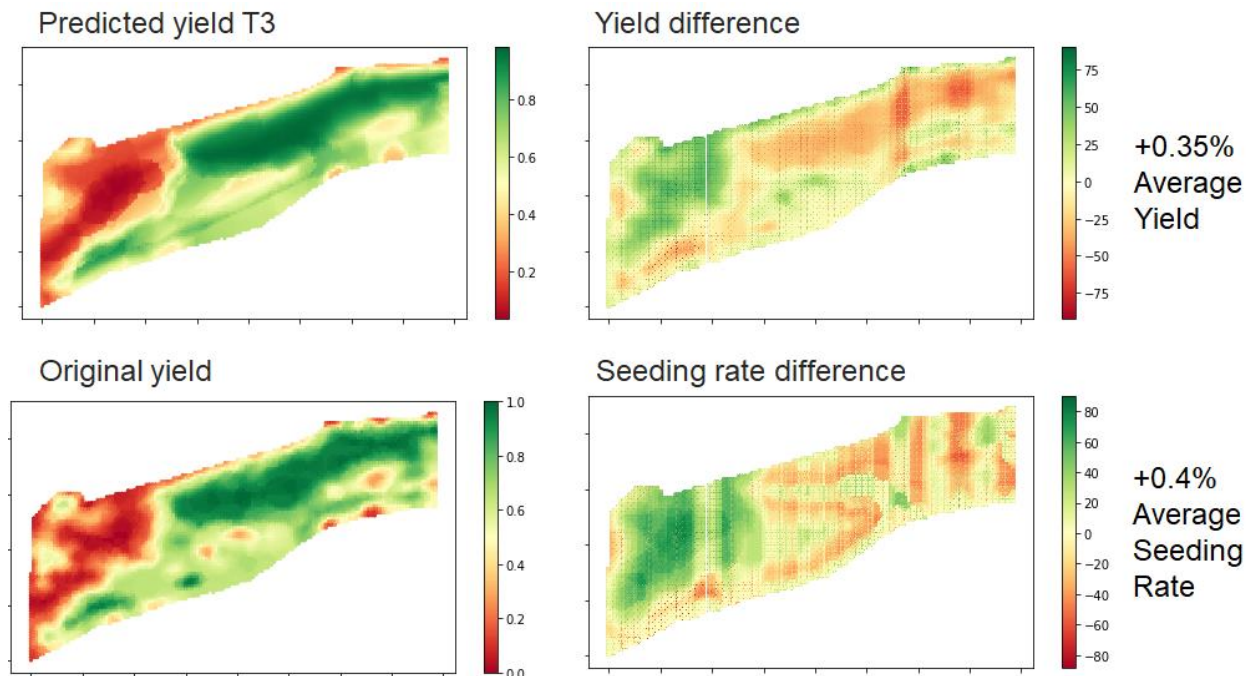
Annex 4. Model 2 Seeding Rate recommendations (Field A)



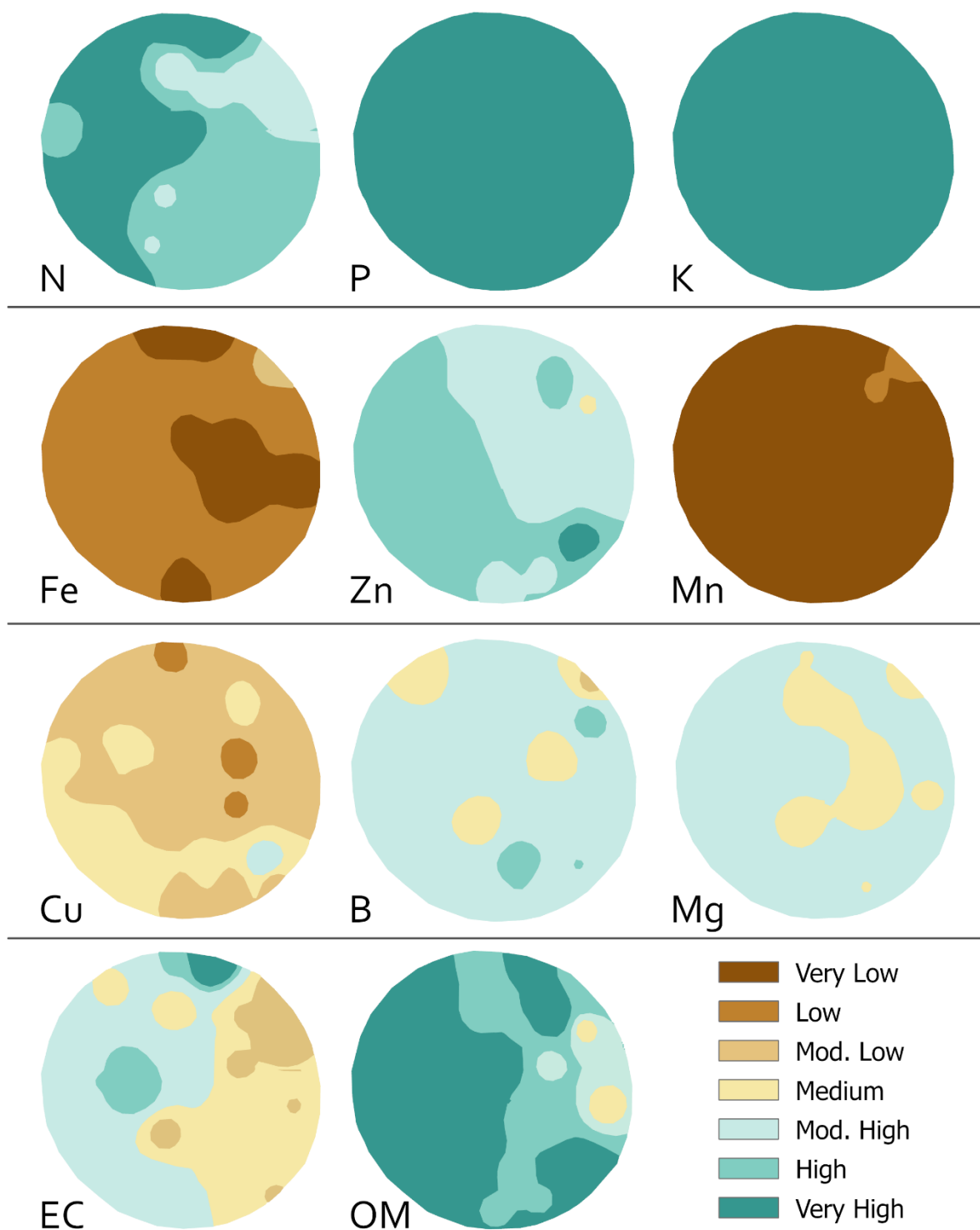
Annex 5. Model 2 Seeding Rate recommendations (Field C)



Annex 6. Yield and seeding differences between original values and generated prescription (Field A)



Annex 7. Yield and seeding differences between original values and generated prescription (Field C)



Annex 8. Spatial distribution of soil properties (Field D)

Word	Abbreviation
Menu	me
Analysis	an
Survey	su
Files	fi
Settings	se
Generate Prescription	gRx
Visualization	vi
Diagnosis maps	dMa
Distribution plot	di
Scatter plot	sc
Regression plot	re
Static sampling	sSa
On-the-go sampling	oSa
Estimate	es
Economic estimate	ecEs

Annex 9. List of abbreviations for AgStat's object naming