Concept demonstrator for a decision support tool for agricultural applications

by

Juanita Claudia van Rooyen

Department of Industrial Engineering

Stellenbosch University

Thesis presented in fulfilment of the requirements for
the degree of Master of Engineering (Engineering Management)
in the Faculty of Engineering at Stellenbosch University

Supervisor: Prof CWI Pistorius
Co-supervisor: Prof SS Grobbelaar

April 2022

Declaration

By submitting this dissertation electronically, I declare that the entirety of the work contained therein is my own, original work, that I am the sole author thereof (save to the extent explicitly otherwise stated), that reproduction and publication thereof by Stellenbosch University will not infringe any third-party rights and that I have not previously in its entirety or in part submitted it for obtaining any qualification.

Date: April 2022

Abstract

Farmers face daily challenges, and there are numerous factors to consider to produce crops profitably. For example, large amounts of data can be overwhelming and complex if not utilised correctly. However, tools such as decision support systems can be incorporated to support the decision-making process. Precision Agriculture presents several opportunities and challenges.

An industry partner, *Company A*, was approached to identify and test a real-world PA problem. The manual element of analysing several data layers is time-consuming and require a more user-friendly way to display data. This research study developed and presented a concept demonstrator of a decision support tool to illustrate how several components can be used to improve the decision-making process. Soil- and nutrient classification data were provided by the use case, *Farm X,* which produces winter wheat in a summer rainfall area in South Africa. Chlorophyll data from 2017 to 2020 were provided by the Airbus Verde service of *Company A*. The assumption was made to add historical and current meteorological data acquired from the South African Weather Services. QGIS was used to extract soil and nutrients classification and chlorophyll data from 296 GPS-specific points on the crop circle. The data table consisted of 85 soil and nutrient and weather features.

A major challenge was presented when no GPS-specific yield was available for *Farm X*. A third (11 088) of the total chlorophyll data were missing, and only 24 849 data points were available for analysis. Nevertheless, Python was used to clean and analyse the available data to provide one chlorophyll value per month for every 296 points. After careful consideration, it was decided to use all features to identify agricultural trends and predict chlorophyll values on a crop circle. A sequential forward feature selector was used to determine which features influence chlorophyll values. A lazy regressor was used to determine the best performing algorithms for feature selection and chlorophyll prediction. The algorithms included the (i) Random Forest regressor, (ii) HistGradientBoost regressor, (iii) XGB regressor and (iv) Extra Trees regressor. The latter outperformed the other algorithms and achieved an R^2 value of 0.86 to predict chlorophyll values for August and September. Operational validation was done using 80% of the data set for training and 20% for testing. The model was then presented with an unknown years data table used for testing to predict chlorophyll for August and September. An R^2 value of 0.273 was achieved. This was to be expected due to the data quality issues and the absence of yield data. The model was provided with at most two chlorophyll values to train with and monthly weather values (instead of daily) to predict a time-series value. The model achieved a positive R^2 value.

The concept demonstrator was successfully developed and tested on a real-world use case. It illustrated how different data sets, machine learning algorithms, predictions and visualization tools could be integrated and used in a decision support tool for agricultural purposes.

Opsomming

Boere word deur daaglikse uitdagings in die gesig gestaar en daar is talle faktore wat in ag geneem moet word om gewasse winsgewend te produseer. Groot hoeveelhede data kan oorweldigend en kompleks wees as dit nie reg aangewend word nie. Hulpmiddels soos besluitondersteuningstelsels kan egter geïnkorporeer word om die besluitnemingsproses te ondersteun. Presisielandbou bied verskeie geleenthede asook uitdagings aan.

'n Bedryfsvennoot, *Maatskappy A*, is genader om 'n werklike PA-probleem te identifiseer en te toets. Die handmatige element van die ontleding van verskeie datalae is tydrowend en vereis 'n meer gebruikersvriendelike manier om data te vertoon. Hierdie navorsingsstudie het 'n konsepdemonstrator van 'n besluitondersteuningsinstrument ontwikkel en aangebied om te illustreer hoe verskeie komponente gebruik kan word om die besluitnemingsproses te verbeter. *Maatskappy A* het grond- en voedingstofklassifikasiedata van *Plaas X* verskaf, wat winterkoring in 'n somerreënvalgebied in Suid-Afrika produseer. Chlorofildata van 2017 tot 2020 is verskaf deur die Airbus Verde-diens van *Maatskappy A*. Die aanname is gemaak om historiese en huidige meteorologiese data by te voeg wat van die Suid-Afrikaanse Weerdienste verkry is. QGIS sagteware is gebruik om grond- en voedingstofklassifikasie data asook chlorofildata van 296 GPS-spesifieke punte op die oessirkel te onttrek. Die datatabel het uit 85 grond- en voedingstof- en weerkenmerke bestaan.

’n Groot uitdaging het na vore gekom toe geen GPS-spesifieke opbrengs data vir *Plaas X* beskikbaar was nie. ’n Derde (11 088) van die totale chlorofildata was vermis en slegs 24 849 datapunte vir ontleding was beskikbaar. Nietemin, is Python gebruik om die data skoon te maak en die beskikbare data te ontleed om een chlorofilwaarde per maand vir elk van die 296 punte te verskaf. Die besluit is geneem om die data patrone te ontleed en om chlorofilwaardes vir Augistus en September op 'n oessirkel te voorspel. 'n “Sequential forward feature selector” metode is gebruik om te bepaal watter veranderlikes chlorofilwaardes beïnvloed. 'n “Lazy regressor” is gebruik om die beste presterende algoritmes te bepaal om te gebruik vir die keuse van veranderlikes en chlorofilvoorspelling. Die algoritmes het die (i) Random Forest regressor, (ii) HistGradientBoost regressor, (iii) XGB regressor en die (iv) Extra trees regressor ingesluit. Laasgenoemde het beter as die ander algoritmes gevaar en 'n R-kwadraatwaarde van 0.86 behaal om chlorofilwaardes vir Augustus en September te voorspel. Operasionele validering is gedoen deur 80% van die data vir die leerproses en 20% van die datastel vir die toetsproses te gebruik. ‘n Onbekende datatabel van ‘n spesifieke jaar is vir die model gegee wat gebruik is vir die toetsproses om Chlorofil vir Augustus en September te voorspel. 'n R^2 van 0,273 is behaal. Dit was te verwagte weens die datakwaliteitkwessies en die afwesigheid van opbrengsdata. Die model is voorsien van hoogstens twee chlorofilwaardes om mee te leer en maandelikse weerdata (in plaas van daagliks) om 'n tydreekswaarde te voorspel. Steeds het die model 'n positiewe R^2 behaal.

Die konsepdemonstrator is suksesvol ontwikkel en getoets op 'n werklike gebruiksgeval. Daar is geïllustreer hoe verskillende datastelle, masjienleeralgoritmes, voorspellings en visualiseringsinstrumente geïntegreer en gebruik kan word in 'n besluitondersteuningsinstrument vir landbou.

Acknowledgements

The author wishes to acknowledge the following people and institutions for their various contributions towards completing this work.

Thank you to my supervisor, Prof Calie Pistorius, for his continuous support and guidance. Thank you for your patience and encouragement throughout the project. I would also like to thank my co-supervisor, Prof Sara Grobbelaar, for her support and check-ins during the completion of my thesis.

I want to thank the industry partner for being involved in this master's thesis and providing me with data to use in this project.

Thank you to Mr OA Esterhuizen at Stellenbosch University for your help with the non-disclosure agreements.

Thank you to my father for affording me the opportunity to further my studies and complete my master's degree. Thank you to my mother for her continuous support, love and always making sure that I eat well and get home safe when travelling from the faculty at night. Thank you to my brother for his advice and support during my most stressed times.

Thank you to my friends, especially Leanne, Jennimi and Ané, for your constant support, advice and coffee breaks. Finally, thank you to Chase for your support and motivation.

Table of contents

List of figures

List of tables

List of acronyms and abbreviations

R^2	Coefficient of Determination
3D	Three-Dimensional
ADSS	Agricultural Decision Support Systems
AI	Artificial Intelligence
ANFIS	Adaptive Neuro-Fuzzy Inference Systems
ANN	Artificial Neural Network
APSIM	Agriculture Production Systems Simulator
BI	Business Intelligence
B	Boron
BPNN	Back Propagation Neural Network
Ca	Calcium
CI	Chlorophyll Index
CPS	Cyber-Physical Systems
CRISP-DM	Cross-Industry Standard for Data Mining
Cu	Copper
CV	Cross-Validation
CYEI	Comprehensive Yield Evaluation Indicator
DAFF	Department of Agriculture, Forestry and Fisheries
DM	Data Mining
DSS	Decision Support System
DSSAT	Decision Support System for Agrotechnology Transfer
DSSI	Damage Sensitive Spectral Index
EMLRM	Enhanced Multiple Linear Regression Model
ETR	Extra Trees Regressor
EWS	Early Warning System
FCover	Fraction of Vegetation Cover
GIS	Geographical Information System
GMM	Gaussian Mixture Modelling
GOES	Geostationary Operational Environmental Satellites
GPS	Geographical Positioning System
GUI	Graphical User Interface
hPa	hectopascal
IoT	Internet of Things
IPI	Irrigation Performance Index
IR	Infrared
IT	Information Technology

K	Potassium
KDD	Knowledge Discovery in Database
LAI	Leaf Area Index
LASSO	Least Absolute Shrinkage and Selection Operator
LGBM	Light Gradient Boosting Machine
LIDAR	Light Detection And Ranging
LRDSI_1	Leaf Rust Disease Severity Index 1
LRDSI_2	Leaf Rust Disease Severity Index 2
LST	Land Surface Temperature
LSTM	Long Short-Term Memory
MAE	Mean Absolute Error
MCARI	Modified Chlorophyll Absorption in Reflective Index
Mg	Magnesium
ML	Machine Learning
mm	millimetre
MMBD	Multimedia Big Data
MSAVI	Modified Soil-Adjusted Vegetation Index
MSE	Mean Squared Error
Na	Natrium
NDRE	Normalised Difference RedEdge Index
NDVI	Normalised Difference Vegetation Index
NDWI	Normalised Difference Water Index
NIR	Near-Infrared
NPCI	Normalised Pigment Chlorophyll Ratio Index
NRMSE	Normalised RMSE
P	Phosphorous
PA	Precision Agriculture
PAT	Precision Agriculture Technologies
PAWC	Plant Available Water Content
PBray1	Phosphorous
PEU	Perceived Ease of Use
pH	Potential of Hydrogen
POAMA	Predictive Ocean Atmosphere Model for Australia
PU	Perceived Usefulness
RGB	Red Green Blue
RMSE	Root Mean Squared Error
S	Sulphur
SAVI	Soil Adjusted Vegetation Index

SAWS	South African Weather Services
SEMMA	Sample, Explore, Modify and Access
SHI	Soil Health Indicators
SIF	Solar-Induced Flurescence
SIPI	Structure Insesitive Pigment Index
SME	Subject Matter Expert
SQ	Soil Quality
SQL	Structured Query Language
SQuaRe	Software Quality Requirements and Evaluation
SVM	Support Vector Machine
TAM	Technology Acceptance Model
TDSP	Team Data Science Process
UAV	Unmanned Aerial Vehicle
VTOL	Vertical Take-Off and Landing
Zn	Zinc

Chapter 1
Introduction

This chapter provides a contextual background of the current trends and challenges of precision architecture (PA), followed by the problem statement and research objectives. Next, the scope of the project is discussed, along with the research approach and strategy. Finally, the thesis outline and chapter summary are presented.

1.1 Context

Farmers face daily challenges such as finite natural resources, external factors (e.g., exchange rate and oil price), climate and environmental changes, as well as diseases and pests (Goldblatt, 2013). There are numerous factors to consider to produce crops profitably, which makes the management of a farm a formidable task. Fluctuating market demand can also be challenging for farmers. Agriculture has undergone and continues to experience significant changes to keep up with demand and remain competitive. Precision agriculture (PA) is regarded by many as the fourth agricultural revolution. CropOM (2021) describes PA as a data-driven enterprise that aims to improve efficiency and optimise production processes to increase profitability. Data is collected from several sources, including the Internet of Things (IoT) sensors, weather stations, geographical positioning systems (GPS), and remote sensing technologies (Rehman, 2015). The data is stored in a database, typically located in the cloud, where it can be analysed and transformed into actionable intelligence. Large amounts of data can be overwhelming and complex if not utilised correctly. However, tools such as decision support systems (DSSs) can be incorporated to support the decision-making process and ultimately increase production efficiency.

DSSs support organisational decision-making activities by collecting and analysing data (Power, 2002). It is a prevalent technology used in many sectors, including manufacturing and logistics, to assist with tasks such as inventory planning and production schedules. Other technologies are also incorporated into the DSS to improve functionality, including artificial intelligence (AI) and machine learning (ML). Business intelligence (BI) software enables users to visualise valuable data to assist in decision-making activities. The software can be used to display real-time dashboards, which can help to improve the efficiency of a business and contribute to decision support.

PA incorporates several Fourth Industrial Revolution-related emerging technologies, often in combination. IoT sensors can provide a plethora of data, including air humidity, temperature, soil moisture, potential of hydrogen (pH) levels and water levels, to name but a few. Remote sensing is used in several applications such as aerospace, land surveying, military, commercial planning and agriculture. Unmanned aerial vehicles (UAV) and satellites can assist in detecting diseases and pests, predicting yields, estimating harvest timing, and analysing water and nutrient status (Jarman & Dimmock, 2018). Various papers explore the use of remote sensing in the agricultural sector. A study conducted by Ballesteros, Intrigliolo, Ortega, Ramírez-Cuesta, Buesa and Moren (2020), for example, combined remote sensing, computer vision and artificial neural network (ANN) techniques to estimate vineyard yield in Spain. A UAV was used to gather multispectral imagery, and the data were analysed to extract valuable information. The experimental research resulted in accurate yield predictions and has shown that the proposed system supports decision-making. Another study utilised remote sensing to detect the properties of soil in specific areas in the United States. Soil property indicators identified by the remote sensing technology included texture, organic and inorganic carbon content, moisture content, pH and iron (Ge, Thomasson & Sui, 2011).

Meteorological data is one of the most important data sets in the agricultural sector. Weather collection devices include thermometers, rain gauges, barometers, radars, UAVs and satellites (Lumen Learning, 2020). The data can help users to identify trends and make predictions to mitigate future risks. Meteorological data specific to a geographical location is known as “weather data” and can better help farmers understand their immediate environment. Weather from a specific region averaged over a long period is known as “climate data”. The University of Minho conducted a study using meteorological data to predict forest fires (which also present significant threats to farmers) using data mining techniques (Cortez & Morais, 2007). Another study examined how meteorological data can be used for efficient irrigation. Weather, irrigation, yield and soil characteristic data were utilised to predict a weekly irrigation schedule. The predicted schedule was compared to the company’s agronomists suggested schedule, and the best performing ML algorithm resulted in a 93% accuracy (Goldstein, Fink, Meitin, Bohadana, Lutenberg & Ravid, 2018).

There are various applications of precision agriculture technology (PAT). Still, it is important to note that PA is continuously being researched and improved to address new and existing agricultural problems and challenges.

1.2 Problem statement

This research primarily uses remote sensing and related data to inform decision support platforms for agricultural purposes, including farmers' early warning systems (EWSs). A concept demonstrator is used as an instrument to explore the research issues.

An industry partner specialises in plant nutrition products and precision farming services offered locally and globally. They have developed a cloud-based product that collates large volumes of data from various sources. Data sources include remote sensing, IoT devices such as soil moisture probes and tracking devices, as well as pest traps and laboratory soil samples. Displayed data include physical and chemical soil maps, pest monitoring, leaf and tissue analysis, yield maps, water analysis and Airbus Verde biophysical parameters. The client can access multiple layers of information and analyse the data to make informed decisions and track progress.

Analysing the various data sets and transforming the information into actionable intelligence and decision support in a client/user-friendly format can be time-consuming for a user. There is thus an opportunity/need to enhance the system in this regard. In this research study, the industry partner will be referred to as "*Company A". Real data collected by *Company A* from "*Farm X"* were used to develop and assess the concept demonstrator. *Company A* provided access to their system and data pertaining to *Farm X* for the purposes of this research subject to a non-disclosure agreement concluded between themselves and Stellenbosch University.

The study aims to develop and validate a conceptual decision-support tool to address the information overload that can improve the decision-making process of the current system, *inter alia,* by reducing the time and effort of manually analysing and optimising many data layers. The farm data supplied by *Company A* were incorporated in the concept demonstrator to demonstrate how a decision-support tool can be utilised in real-life scenarios and potentially mitigate risks. The data relates to winter wheat data in a summer rainfall area in South Africa. The literature investigated PA technologies and applications, whereas the field research provided the context for the use case farm. The knowledge gained from the research was combined with statistics, data analytics and decision support principles to develop the concept demonstrator.

The following research questions emerged and were used to guide the development of the concept demonstrator successfully. The questions are addressed in various chapters of this research study.

1. Are farmers using remote sensing technology on their farms, and if so, which sources are they using (including data from satellites and UAVs)? What are the pros/cons as well as restrictions and limitations of remote sensing technology?

2. Which types of data sets and associated sensing technologies are used by farmers?

3. Which diseases/problems are potentially preventable if early detection and forecasting can be done, and which crops best lend themselves to this application?

4. Which data sets can be added to the tool to improve decision support, such as meteorological data and IoT sensors inputs?

5. What additional information needs to be collected to perform data analysis?

6. How can the conceptual tool improve the current technologies used by the industry partner in the agriculture sector?

7. How will the decision support be presented to the project partner effectively?

8. How can all the components be integrated to develop the decision support tool?

9. Can predictive analytics be used to detect patterns in the data?

10. What are the most important features/variables that affect decision-making?

1.3 Research objectives

The primary objective of this research study is to develop a better understanding of the main components and applications of PA through the study of a real-life PA scenario to develop a DSS tool. The research objectives have been grouped into (i) a literature study and (ii) field research objectives, which were researched in parallel.

1.3.1 Literature study

To fully understand PA and its changing environment, a comprehensive literature study was conducted to gain more knowledge to develop the concept demonstrator. The literature study researched applications in the entire agriculture sector with the primary focus on wheat. The following points are addressed:

- Background research on PA, its components and applications
- Adoption of the technology by farmers
- The use of satellites and UAVs in agriculture
- Research on the Airbus Verde satellite used by the industry partner, including its:
    - Specifications
    - Advantages/disadvantages
- Prevalent diseases and pests detectable via remote sensing
- Crop management definition, components and applications
- Other data inputs, including laboratory tests of crops and IoT sensors
- Software:
    - Software to display and manipulate remote sensing data
    - Data analytics software
    - Visualisation tools and software
- Determine best practices for displaying agricultural data to users
- DSSs:
    - Basic principles and components
    - Applications
    - Features and benefits
    - Technology and sources used
    - Commercial and/or experimental DSSs currently available

- Other helpful complementary information
    - Other technology solutions
    - Additional complementary data sources
    - Relevant emerging technologies

1.3.2 Field research

The field research was conducted parallel to the literature study to investigate a real-world example of PA and gain more knowledge regarding winter wheat crop management. Field research aims to investigate the industry partner's existing product and identify the farmer's specific requirements to design and implement a decision support tool. Thus, improving the efficiency of the software by removing the "manual element" of decision-making when analysing the multi-layered data of the client. The following main points are used to address the field research objectives:

- Conduct background research on *Company A* and *Farm X* to accurately define the project requirements of this research study. The research includes:
    - Services and software
    - Data sources used for data acquisition
    - Type and quality of the data
    - Crop health indicators displayed on the platform (features and variables)
- Research the crop characteristics of the data provided from the clients
    - Ideal growing conditions are required for successful crop production, for example, weather, soil, irrigation, etc.
    - Wheat crop lifecycle, including pre- and post-harvest management
    - Diseases and pests associated with the crop
    - Other variables that can potentially influence crop health
- Determine how the partner(s) is currently utilising the product.
    - Research the systems/technologies currently used on the farm(s): What remote sensing, IoT technology is used to collect data?
    - What IoT devices are used, and what data do they collect?
    - Pathology/lab testing/horticulture
    - Which indicators are provided by the platform?
- Refine the needs and requirements of the project partner
- Determine which diseases are prevalent on the project-partner farm.

1.4 Scope

The project scope is constrained by the data and information that are available and accessible for analysis from *Farm X*. This project did not consider the entire agricultural supply chain (e.g., procurement, import, export) but only focused on the factors influencing the crop lifecycle (e.g., pre-harvest, crop growth and post-harvest). The literature study includes a broader research approach, whereas the field research focuses on winter wheat in a specific province in South Africa. The data analysis and any additional data sets acquired also focused on the specific region. The project scope was impacted by:

- The size of the farm
- Type of crop
- Annual or perennial crop
- Quantity of available historical remote sensing data from *Farm X*
- Quality of the data
- The requirements of the stakeholders
- The two-year timeframe between 2020 – 2021 to complete the project

Related assumptions are discussed in more detail in Chapter 3 and Chapter 4, where more information regarding the use case and the available data is explored.

1.5 Research approach and strategy

The research approach followed in this research study is demonstrated in Figure 1.1 (see overleaf). A brief literature review was conducted to identify the trends and challenges in PA and used to define the initial research topic. A potential industry partner(s) was approached with the initial research questions to determine whether a more detailed real-world problem regarding a specific use case scenario exists. This step did not have to be repeated as the first industry partner approached was able to present a viable real-world PA problem. The initial problem statement and research questions were refined. Thereafter, the user requirements were defined and used to formulate the research objectives. Brymann and Bell's (2011) quantitative outline was used as a guideline for the research methodology approach.

A more comprehensive secondary literature review was conducted to gain the necessary knowledge on existing technologies and data approaches. In contrast, the field research provided information about the project partner environment and analysed data. The findings from the literature review were used to analyse the data and develop a concept demonstrator that is applied to a real-world example. The concept demonstrator aimed to fulfil the research questions and satisfy the research objectives.

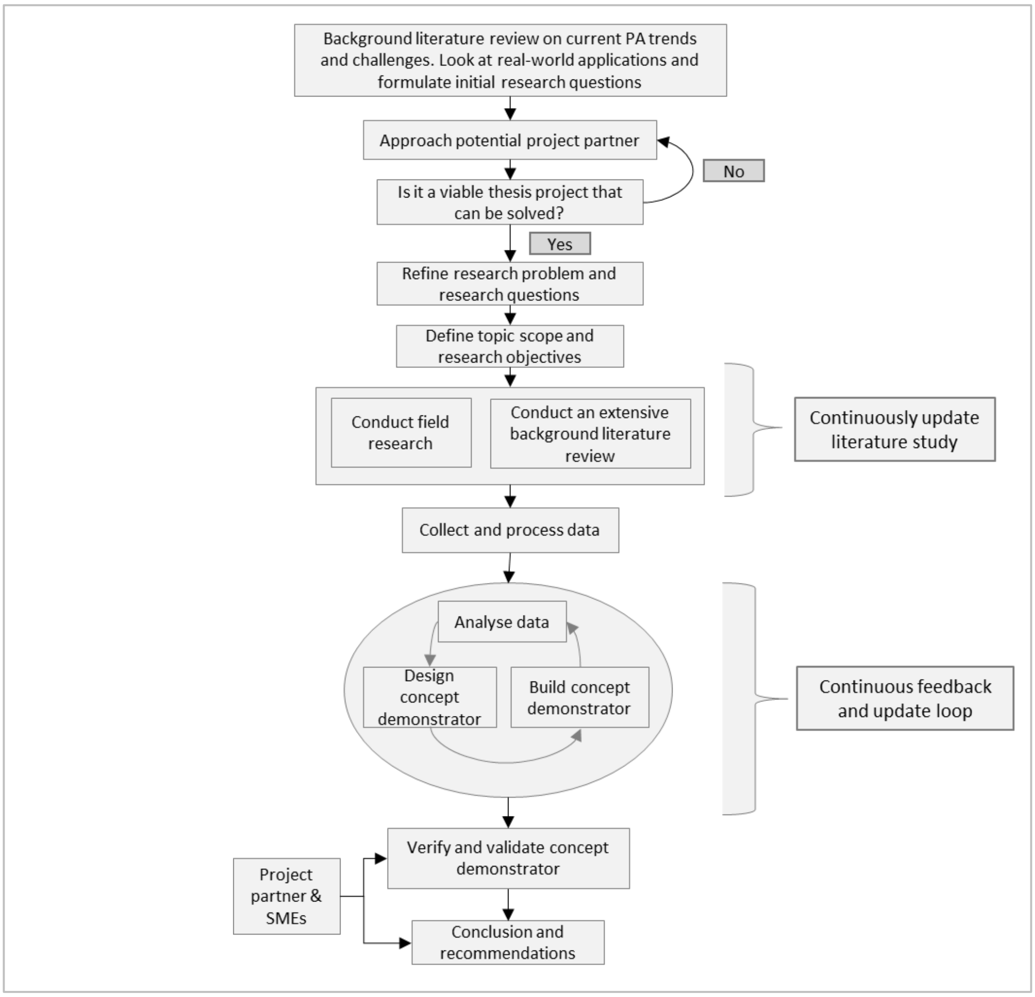

Background literature review on current PA trends and challenges. Look at real-world applications and formulate initial research questions
Approach potential project partner
No
Is it a viable thesis project that can be solved?
Yes
Refine research problem and research questions
Define topic scope and research objectives
Conduct field research
Conduct an extensive background literature review
Continuously update literature study
Collect and process data
Analyse data
Design concept demonstrator
Build concept demonstrator
Continuous feedback and update loop
Verify and validate concept demonstrator
Project partner & SMEs
Conclusion and recommendations

Figure 1.1: Research methodology mind map

This chapter provided context on how the research topic was formulated by conducting research on PA and identifying appropriate research questions. The project partner was consulted to provide a more in-depth use case scenario. The problem statement and research questions were refined to address the specific use case of *Farm X*. The research objectives were grouped into a literature study and field research to gain the necessary knowledge to design a concept demonstrator.

The conceptual model was customised according to the requirements of the project stakeholders and was designed to be adaptable to more scenarios. The ultimate goal was to design and develop a conceptual model to improve efficiency and add value to the industry partner involved. Furthermore, incorporating an innovative industry partner added immense value to the learning of the researcher.

1.6 Thesis outline

Chapter 2: Literature study

The information gleaned from the literature study provides a better understanding of PA, DSSs and current and emerging technologies used. The chapter opens with the principles of PA and the adoption of technology. Various remote sensing technologies and applications are discussed, followed by the use of meteorological data in agriculture for decision-making. Next, the importance of big data, data methodologies and visualisation tools are discussed. Lastly, the chapter focuses on ML algorithms, agriculture DSSs and crop management application examples.

Chapter 3: Field research

Chapter 3 provides information regarding *Company A* and *Farm X* to understand the current services and data better. In addition, the ideal growing conditions were researched to determine which factors can potentially influence decision-making.

Chapter 4: Data analysis

The data analytics methods discussed in the literature study and the knowledge gained from the field research will be used to perform the data analysis.

Chapter 5: Concept development

This chapter discusses the various sections in the document to indicate how they were used to develop the concept demonstrator tool.

Chapter 6: **Next-generation decision support tool**

Chapter 6 discusses two prediction scenarios of using known and unknown test data to make chlorophyll-related predictions. The data sets used in the algorithm and the prediction accuracy are presented.

Chapter 7: Validation and verification

This chapter discusses how the model incorporated into the concept demonstrator is validated and includes the questionnaire used to gather input from subject matter experts (SMEs) to test validity.

Chapter 8: Summary, recommendations and conclusion

The final chapter provides a summary of the study and discusses the research findings. Finally, recommendations for future work are made.

Chapter 2
Literature study

A systematic literature review was conducted to provide a foundation for understanding and developing a conceptual decision support tool for agriculture. Section 2.1 provides a brief discussion on PA and the adoption of new technologies to place the rest of the discussion into context. A review of remote sensing is presented in Section 2.2, including a discussion on current developments in the IoT, sensors, geographical information systems (GIS), and climate and meteorological data. The concepts of big data, data analysis and visualisation tools are discussed in Section 2.4, along with a brief discussion on AI and a more extensive review of ML concepts and algorithms in Section 2.5.

Numerous literature articles were considered to research the components and compile tables that summarise existing use case applications in crop management (Section 2.6) and DDSs (Section 2.7). A summary is presented in Section 2.8.

2.1 Precision agriculture

The Fourth Industrial Revolution, also sometimes popularly referred to as “Industry 4.0”, is evolving rapidly and is disrupting many industries. Cyber-Physical Systems (CPSs) play a pivotal role in this technological transformation and have several applications in the manufacturing, automotive, healthcare, military, entertainment, and agriculture sectors. Agriculture-related applications of the Fourth Industrial Revolution are commonly referred to as PA.

The World Economic Forum’s research indicates that Africa’s population growth will triple by 2050. The United Nations’ (UN) most recent estimation projects that the world population growth could reach 11 billion by 2100 (Hajjar, 2020; UN, 2019). There is a dire need for more sustainable farming practices and increased food production to accommodate the rapid population growth and overcome the challenges farmers face. PA, also known as precision farming or smart farming, is thought to be the solution. However, farmers generally face many challenges, including scarcity of fresh water, climate change, pests and diseases and other socio-economic factors.

One of the primary goals of PA is to use advanced technologies to precisely measure the variation in the field (Verma, Bhatia, Chug & Singh, 2020). Advanced technologies such as remote sensing, IoT, sensors, big data, AI, UAVs (also called drones) and cloud computing are utilised for farm management activities. Applications include production scheduling, crop monitoring, livestock tracking, variable rate application, and pest and disease monitoring. Farm data can be used to analyse trends and make predictions that can provide valuable insight to the farmer. The farmer can utilise a data-driven approach by utilising the data collected from various sources to support decision-making and ultimately increase profitability.

Despite the interminable possibilities of PA, several challenges and limitations present themselves, including access to power and Internet connectivity in remote and rural areas and a lack of training and expertise (Microsoft Research, 2021).

Farmer adoption of PA is another major challenge. As part of this research, the literature was studied to understand better the factors that influence the adoption of PA. Sheng Tey and Brindal (2012) conducted a review of ten studies to investigate farmer adoption in developed countries with respect to PATs, focusing on (i) GPS, (ii) remote sensing, (iii) soil sampling, (iv) yield monitoring and (v) variable-rate applicators. They concluded that 34 factors explained the adoptive decision-making of PATs, grouped into seven categories, viz. socio-economic, agro-ecological, institutional, informational, farmer perception, behavioural and technological.

Pierpaoli, Carli, Pignatti and Canavari (2013) conducted a study to determine the factors influencing farmer adoption with regard to PATs. Their research focused on two main groups, viz. (1) factors that influenced farmers that have already adopted PATs and (2) factors influencing farmers with the intention to adopt PATs. The most important factors that influenced the adoption of the first group can be seen in Figure 2.1 below. Farm size and confidence with computers and technology were the most frequently cited parameters affecting the use of PATs. Other important factors include farmer age, farmer education and a high farm income. Figure 2.2 illustrates the factors affecting farmers' attitudes to adopt PATs. The Technology Acceptance Model (TAM) was used to explain which drivers could affect a potential user's behaviour to adopt or not adopt PA technologies. The main themes that influenced the behaviour to adopt were Perceived Ease of Use (PEU) and Perceived Usefulness (PU). Factors such as farm size, education and cost-benefit analysis can contribute to these perceptions; however, it was discovered that technology demonstrations and free-trails encouraged positive behaviour toward PAT adoption (Pierpaoli *et al.*, 2013).

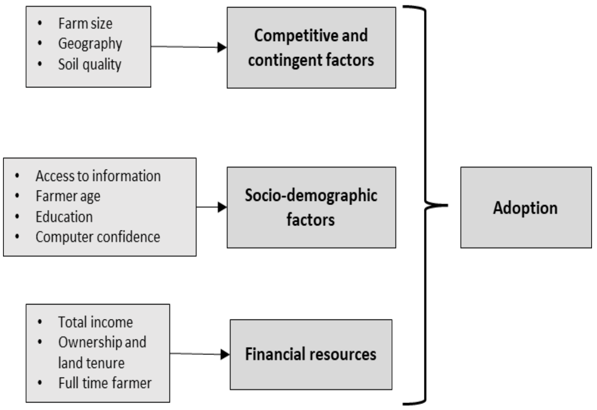


Figure 2.1: Factors that influenced PAT adoption

Adapted from Pierpaoli *et al.* (2013)

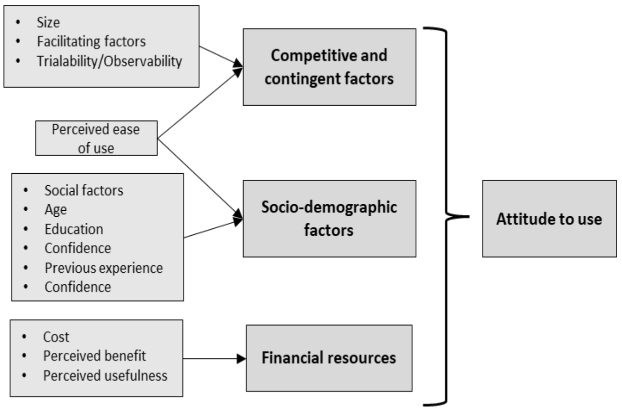


Figure 2.2: Factors affecting attitude to adopt

Adapted from Pierpaoli *et al.* (2013)

PAT adoption challenges are often overcome by implementing technologies systematically, with a phased approach. PrecisionAg Alliance (2020) defines six levels of PA adoption, which are briefly discussed below. The levels were used in Chapter 5 to determine where *Farm X* lies in this spectrum and aid in designing the decision support tool.

Level 0: Equipment efficiency and basic automation
The main focus of this level is efficiency-technologies such as automation steering. There is little to no data collection, and any available data is used for operations but not for production planning.

Level 1: Basic georeferenced data collection
Spatial data is collected to assist in inter-field and sub-field assessments and year-over-year fertility plans. The field data is collected and analysed but not necessarily fully utilised in decision-making.

Level 2: Advanced georeferenced data collection
Imagery, weather data and other information sources are used to capture data to support operational decisions. Outside expertise is often used for data collection and aggregation.

Level 3: In-season decision-making
Level 3 adoption integrates multiple data layers to provide an evidence-based approach to decision-making.

Level 4: Digital and process mastery
Having operated at level 3 for a few years, the grower has accumulated multiple data layers and can make yearly comparisons to assist with in-season operational decisions.

Level 5: Continuous improvement and systems mastery
Level 5 includes exploring new technologies and continuous improvement by utilising the integrated technologies and data sets for effective decision-making. Level 5 adoption typically implements imagery, weather- and soil moisture sensors, as well as pests- and disease monitoring systems.

The remaining part of the literature focuses on the leading technologies used in PA, with practical examples. The main focus areas are remote sensing, IoT and sensors, UAVs, GISs, crop health indicators, meteorological data, big data, AI and DSSs. This research, along with field research, were used to gain the necessary knowledge to design the concept demonstrator. The remaining literature discusses the definitions, backgrounds, and components and examines current technologies used.

2.2 Remote sensing

Remote sensing is the science of obtaining information without physical contact with the observed object. Remote sensing is considered a primary means of acquiring spatial data and measures the energy or electromagnetic radiation interacting with objects (Zhu, Suomalainen, Liu, *et al.*, 2017).

The sun's energy is reflected, absorbed or transmitted by the material's surface in certain regions of the spectrum. The relationship between the reflected, absorbed, and transmitted energy is used to determine the spectral signatures of objects. Remote sensing uses these unique spectral signatures to distinguish between vegetation, water, soil and other features (Nowatzki, Andres & Kyllo, 2017).

The term "remote sensing" was first introduced by Fischer in the 1960s when the new technologies surpassed traditional aerial photography and required a more comprehensive term to define emerging technologies. The shift from aeroplanes to satellites ensured more regular land space cover (Baumann, 2009). Remote sensing applications in agriculture include detecting and monitoring the physical characteristics of soil and plant material (Mulla, 2013).

The type of sensors and imaging systems were researched to better understand the data collection process and type of agricultural data acquired from *Company A*. The research was also used to investigate applications of current technologies and how they can be used in a decision-support tool.

2.2.1 Sensors and resolution

The two types of remote sensing sensors relevant to this research study are active and passive sensors. Passive remote sensing records reflected electromagnetic radiation (e.g., visible light and near-infrared (NIR) light) or emitted electromagnetic radiation (e.g., thermal infrared light) from the surface of an object. Active remote sensing emits radiation and provides its own source of energy to illuminate the objects observed. The rapid advancement in sensors has led to the integration of passive and active sensors. Both imaging sensors and non-imaging sensors can be used in remote sensing instruments.

2.2.1.1 Non-imaging

Non-imaging sensors include radiometers, spectrometers, altimeters and LIDAR (light detection and ranging). The sensors typically operate in visible light, infrared (IR), and microwave spectral bands and can determine temperature, height, wind speed and other atmospheric measurements. Red laser non-imaging is commonly used for vegetation measurements and LIDAR for three-dimensional (3D) topographic mapping (Zhu *et al.*, 2017).

2.2.1.2 Imaging sensors

Imaging sensors include (i) optical imaging, (ii) thermal imaging and (iii) radar imaging sensors.

(i) Optical remote sensing
Optical imaging sensors operate in the visible and reflective IR range and include panchromatic, multispectral and hyperspectral imaging systems (Zhu *et al.*, 2017). Optical images in the visible spectrum cannot be acquired at night (although IR can overcome this limitation) or when obstructed

by cloud cover. Table 2.1 below summarises the difference between optical remote sensing platforms and their applications in agriculture.

Table 2.1: Optical remote sensing and satellite applications (Zhu *et al.*, 2017)

Features	Panchromatic	Multispectral	Hyperspectral
Spatial resolution	Submeter	1-2m	2m
Satellites	QuickBird, SPOT, IKONOS	SPOT, QuickBird, IKONOS, Landsat, SPOT, RapidEye, Worldview-2 and 3.	TRW Lewis, EO-1
Spectral range (nm)	430-720	430-720; 750-950	470-2000
Applications	Earth observation and reconnaissance applications	Red-green-blue: visual analysis Green-red IR: vegetation and camouflage detection Blue-NIR-MIR: visualising water depth, vegetation coverage, soil moisture content, and the presence of fires, all in a single image.	Agriculture Food processing Mineralogy Surveillance Physics Astronomy Chemical imaging

Multispectral imaging has a high spectral resolution. Panchromatic images, which have a high spatial resolution, are often combined or fused for improved visual image interpretation and information retrieval. This is known as pan-sharpening or intensity substitution. It combines three bands from the multispectral image with the panchromatic image to produce an output with both image types' spatial and spectral properties. Pan sharpening is useful for object-based image analysis such as farm boundaries (STARS project, n.d.). The narrow bands of hyperspectral imagery are more sensitive to variations in energy wavelengths and, therefore, have a greater potential to detect crop stress than multi-spectral imagery.

(ii) Thermal imaging

Thermal sensors typically operate between the mid-to-far-IR and microwave spectrum ranges. It does not require illumination from solar radiation and can provide imaging in the day or night-time. Thermal sensors can be used in livestock tracking and forest fire and threat detection.

(iii) Radar sensors

Radar sensors typically operate in the 1mm – 1m spectrum range. Radar can show the difference in surface roughness and soil moisture and is often used in conjunction with IR, identifying minerals and vegetation types.

Remote sensing imaging sensors are generated based on four types of resolutions:

1. *Spatial resolution* - This refers to the size of the smallest object that can be detected in an image and is usually presented by a value representing the length of one side of a square. A spatial resolution of 100m means that one pixel represents a 100m x 100m square on the ground.

2. *Spectral resolution* - The sensors' ability to measure the width of the wavelengths and number of bands of the electromagnetic spectrum.

3. *Radiometric resolution* - The sensitivity of the sensor to detect variations in the reflection on land surfaces, and it is measured in bits. The more bit values an image has, the more grey-scale values can be stored to differentiate reflectance (FIS, 2020).

4. *Temporal resolution* - The frequency of images of the same geographical area. Geo-stationary satellites continuously provide sensing, while orbiting satellites can only provide images each time they pass over an area. In addition, cloud cover can interfere with the data from a scheduled remotely sensed data system.

2.2.2 Remote sensing imaging systems

Remote sensing can be grouped into ground-, air- and satellite-based imaging systems. Imaging applications in agriculture include pest control, crop irrigation, disease monitoring and other agriculture-related activities.

2.2.2.1 Ground-based imaging systems

Ground-based remote sensing uses a variety of geophysical surveying to scan below the surface and is useful in field monitoring for detecting biotic and abiotic crop stresses. Ground-based sensors can be used in handheld devices or can be attached to machinery. They are efficient to evaluate small areas, whereas airborne and satellite-based remote sensing is preferred when large-area sensing is required.

2.2.2.2 Air-based imaging systems

UAVs, also known as drones, are robots that can fly in manual, semi-autonomous and autonomous modes without a pilot on board. They are categorised into (i) multi-rotor, (ii) fixed-wing, (iii) single rotor, and (iv) hybrid Vertical Take-Off and Landing (VTOL) systems (Yinka-Banjo & Ajayi, 2019).

The cost of acquiring UAV imagery or purchasing UAV technology is currently a major challenge in adopting UAVs. UAVs have a limited flight time and are currently not competitive against non-battery-operated UAVs and satellites when considering large areas of cover in a time-constrained scenario. On this point, it is important to keep in mind that, as is typically the case with emerging technologies, the performance of drones will increase, and prices will drop. The solar-powered hybrid fixed-wing UAV could solve the current problem (Yinka-Banjo & Ajayi, 2019).

Table 2.2 below illustrates several applications of UAVs in agriculture, such as animal mustering, crop monitoring, pest and herbicide spraying and disease detection.

Table 2.2: UAV applications in agriculture

Application	Country	UAV	Use	Technology and sensors	Resource
Mustering (cattle)	Australia	Quadcopter drone	• Reduced labour and reduced risk of using quadbikes and horses for animal mustering. • Approximate cost of $20 (Aus.) of drone mustering in a 600-hectare paddock.	Not specified	Bolton (2020)
Monitoring and identification	Several countries	Not specified	• Monitoring endangered animals. • Cameras fitted on drone can scan RFID and identify animal.	RFID tags, QR codes	Yinka-Banjo & Ajayi (2019)
Monitoring and risk detection (livestock)	Several countries	Single- and multirotor drones	• Monitoring the impact of feral animals and invasive predators on livestock, especially at night. • Tracking stolen and missing livestock by using radio-tracking drones.	Radio sensors, RFID	Wildlife Drones (2020)
Crop monitoring	Colombia	Quadrotor	• Biomass estimation in rice by modelling the relationship of selected vegetation indices.	Multispectral NIR	Abdulridha, Ampatzidis, Kakarla & Roberts (2019)
Crop and spot spraying	India	VTOL Quadcopter	• UAV used to spray pesticides to reduce pesticide contact with humans. • Controlled spraying by utilising imaging sensors and spraying areas not easily accessible to humans.	Multispectral camera QGIS software	(Meivel, Gandhiraj, Srinivasan & Maguteeswaran (2016)
Disease detection	Florida, USA	DJI Matrice drone	• Used UAV remote sensing to distinguish between target spot and bacterial spot infected tomato plants at different disease development stages.	Resonon Pika-L2.4 hyperspectral sensor	Abdulridha *et al.* (2019)
Herbicides	Brazil	eBeeX fixed-wing drone	• Mapped 500 hectares and detect weed infestations areas. • Generated application maps and reduced herbicides by 52% on Soybean farm.	Xarvio field manager	Pinguet (2021)
Planting	USA	AeroSeeder-Octocopter	• Autonomous drone equipped with an 18kg sack and terrain-following sensors to release seeds. • Focused on cover crops and eliminates risk of damaging main crops with heavy ground machinery. • Can cover 40.5 hectares in eight hours.	Not specified	Coxworth (2020)

2.2.2.3 Satellite-based imaging systems

Satellite sensing is widely used in forestry, oil and gas, agriculture, mining, construction, oceanography, insurance and finance, and medicine. Geostationary satellites travel at the same rate as the Earth's rotation and provide continuous coverage of one specific area on Earth. Geostationary operational environmental satellites (GOES), otherwise known as weather satellites, are examples of such satellites. Popular remote sensing satellites, their applications in agriculture and spatial resolution are discussed in more detail in Appendix A1.

The data provided by *Company A* is acquired from the Airbus Verde service that delivers detailed crop analytics from satellite imagery. Verde can be integrated into any PA portal with any satellite used as a source. The imagery is cropped to the parameters of the farm and de-clouded, which is used to detect anomalies, optimise field scouting, irrigation, fertilisation and seeding (Airbus, 2019). It provides 15 indicators, including leaf area index (LAI), leaf chlorophyll content, leaf water content, and normalised difference vegetation index (NDVI). Verde collaborates with a UK agritech company to combine service and link metrics such as soil chemistry, weather and ecological indicators (African Farming, 2020). The type of data provided by the Verde service for this use case is discussed further in the field research (see Section 3.2).

Choosing the best remote sensing technology depends on the type of application and imagery required by the farmer. Airborne-based and satellite-based remote sensing gather information in different ways and scales. It is often not an “either-or” but rather an “if-then” decision when it comes to deciding which technology to choose (Barnes, 2018). Airborne and satellite remote sensing are often combined to utilise the full potential of both technologies. Aerial photography has a higher resolution but is currently more expensive per square meter. MicaSense and their South African partner, Aerobotics, are examples of companies that incorporate both satellite and multispectral data to provide different levels of information and data analytics solutions to help farmers detect pests and diseases (MicaSense & Aerobotics, 2021).

2.2.3 GIS and GPS

GIS refers to computer software that visualises information gathered from remote sensing and GPSs. It captures, stores and displays data related to positions on the Earth’s surface and integrates the data captured from remote sensing to show data, such as streets, buildings and vegetation, on a map. Popular GIS software includes ArcGIS, Google Earth Pro, Google Maps API, ArcGIS, QGIS, PostGIS, Global Mapper and gvSIG (G2, 2021; GISGeography, 2021).

2.2.4 IoT and sensors

Agriculture management requires timeous data on several factors such as soil quality, fertilisers, irrigation and meteorological data. Sensors can be used to collect these, including temperature, soil moisture, light and pH sensors. The IoT and edge devices consolidate various communication technologies to create an intelligent system that interacts with the real world and digital world, connecting (smart) devices with another, computers, and people. Sensors can be combined with several other technologies to provide a complete integrated monitoring system (Verma *et al.*, 2020). Kumar, Mishra, Gupta and Dutta (2021) compiled a detailed figure (see Figure 2.3 below) to show applications of IoT sensors in PA, including soil health monitoring, irrigation, disease identification and crop yield monitoring.

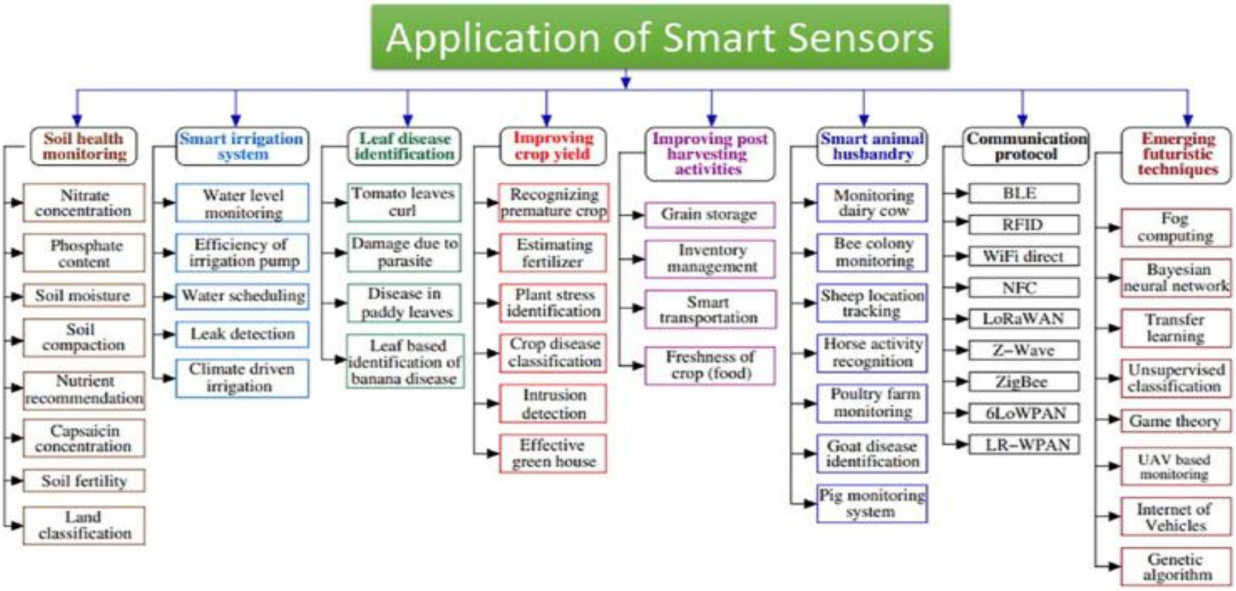


Figure 2.3: Applications of smart sensors in precision agriculture (Kumar *et al.*, 2021)

2.3 Climate and meteorological data

Climate and meteorological data is critical to the success of agriculture production and profits. Climate data provide valuable insight to assist the farmer in decision-making regarding factors such as crop selection, pesticides and harvesting. It is also important to compare historical, current and forecasted weather data to ensure more accurate decision-making. Historical weather data can provide valuable insight into past weather patterns and seasonal data, whilst current weather data can help plan day-to-day and short-term operational strategies. Using both historical and present data can help to predict future trends by utilising appropriate weather forecasting techniques.

Meteorological data also play an important role in managing pest and disease control, thereby helping to mitigate these risks. Copious amounts of literature explore the effects of climate change and weather data on the agricultural sector across the globe. One study, for example, aimed to develop an adaptive model for forecasting seasonal rainfall using predictive analytics. A framework called the “Enhanced Multiple Linear Regression Model” (EMLRM) was proposed, including a rainfall prediction model (Reddy & Sureshbabu, 2019).

Han, Baethgen, Ines, Mer, Souza, Berterretche, Atunez and Barreira (2019) developed a decision support tool that compares several input variables to climate conditions. It allows, for example, the user to input planting dates, crop variety and fertiliser application and then choose a historical, forecasted or hindcasted climate option. The results of the input variables are simulated against the climate option selected, and the output results are used to aid in the decision-making process. The tool, named SIMAGRI, was customised for maize, soybean and wheat crop production in Uruguay but can be modified to be applicable in other countries. The weather data used in the tool include long-term data of minimum and maximum air temperature, solar radiation and precipitation data of Uruguay.

Frisvold and Murugesan (2013) conducted a study that used a subsample of 284 farms in Arizona to assess the use of weather data for agricultural decision-making. Two of the main research questions explored the (i) importance and (ii) use of different types of weather data for production and marketing decisions. Part of the study asked farmers to indicate the importance of weather data on their decision-making. The farmers' responses were recorded on a Likert scale and analysed. Table 2.3 below summarises the type of management decisions made from specific weather information.

Table 2.3: Types of weather data used for agricultural decision-making (Frisvold & Murugesan, 2013)

Type of weather data	Agricultural decisions
Temperature	Planting, harvesting, defoliation, crop modelling, disease risk, pest control
Precipitation	Planting, harvesting, fertiliser applications, cultivation, spraying, irrigation, disease risk
Soil moisture	Planting, harvesting, fertilising, transplants, spraying, irrigation, monitoring growing conditions, measuring plant stress
Soil temperature	Planting, pest overwintering conditions, transplanting, fertilising
Relative humidity	Planting, irrigation, pest control, harvesting, pollination, spraying, drying conditions, crop stress potential
Wind speed	Defoliation, harvesting, freeze potential/ protection, pest control, pruning, spraying or dusting, pollination, dust drift, pesticide drift
Wind direction	Freeze potential/protection, cold or warm air advection over crop areas, pesticide drift, dust drift

WeatherPlot is a site-specific precision weather and soil analytics mobile application built on Iteris' ClearAg platform. It can provide hourly and daily weather information and 30 years of historical and forecasted data with soil-related information. The application also provides advisory services assisting in pests and diseases, crop nutrition, irrigation and planting and harvest timing.

Climate and meteorological data is discussed in Section 2.6.3 concerning yield prediction and Decision support in agriculture (see Section 2.7.1).

2.4 Big data, analytics and visualisation

Big data, data analytics and visualisation can be integrated to collect and extract value from data and present it in a useful and user-friendly format. Large amounts of data is often complex, and visualisation tools are required to provide easily interpretable visualisations and dashboard displays. The information in this section was used in the design and development of the concept demonstrator discussed in Chapter 5.

2.4.1 The nature of big data

Big data can be described as data that are too large, fast and complex for traditional methods to be used to process (SAS, 2021). The term “big data” gained momentum in the early 2000s. The definition of the 3Vs was first introduced by the analyst Dough Laney (Marbella International University Centre (MIUC), 2020) viz. volume, velocity and variety, which he described as:

> *“Big data is high-volume, high-velocity and/or high-variety information assets that demand cost-effective, innovative forms of information processing that enable enhanced insight, decision-making, and process automation.”*

Although the concept of big data is not new, the tools and techniques used to analyse large data sets are becoming increasingly powerful and sophisticated. Laney’s definition is widely accepted, although some authors have attempted to expand the definition by adding additional Vs such as variability, veracity and value (MIUC, 2020; Oracle, 2021; SAS, 2021).

2.4.1.1 Why big data?

With the advent of the digital revolution and fusion of big data, cloud computing and IoT devices, sensor-based technologies are becoming more affordable and accessible. As a result, a plethora of data is generated, collected, and used for various industries, including automotive, healthcare, military, manufacturing, and PA sectors. According to Kumari, Tanwar, Tyagi, Kumar, Maasberg and Choo (2018), the primary challenge of big data is dealing with and utilising the vast amounts of Multimedia Big Data (MMBD). The data acquired from multiple sources are often unstructured and requires data pre-processing and complex algorithms to extract valuable data. This adds to the complexity of data storage, processing capabilities and analysis techniques. Therefore, conventional data processing tools are not sufficient. Instead, big data mining techniques can be used to uncover patterns and enhance decision-making by providing knowledgeable insight to the stakeholders. Figure 2.4 below shows the basic framework of MMBD computing and its essential processes (Verma *et al.*, 2020).

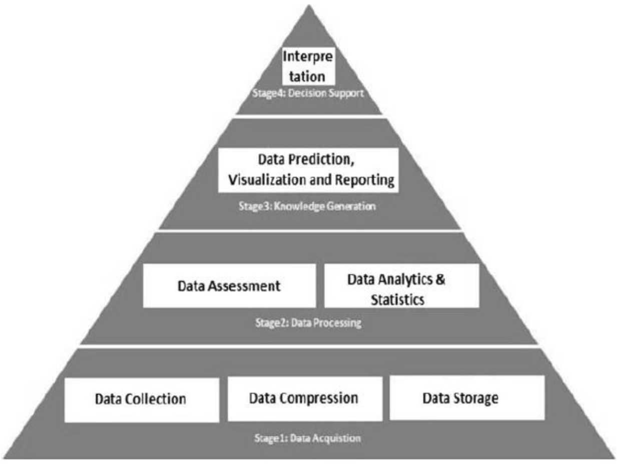


Figure 2.4: The basic framework of MMBD computing and its processes (Verma *et al.*, 2020)

The framework consists of four stages: data acquisition, data processing, knowledge generation, and decision support. Data acquisition (stage 1) includes collecting raw and unstructured data, which are processed and then stored. In the data processing stage (stage 2), the data is analysed using data analytics tools. The analysed data is then used in the knowledge generation stage (stage 3) to make predictions and visualisations. Finally, the knowledge gathered from the data can be used to derive conclusions and improve decision support (stage 4).

2.4.1.2 The challenges of big data

Two of the major challenges in implementing big data in agriculture are the initial investment cost of the infrastructure and the proper training of farmers (Verma *et al.*, 2020). According to Wolfert, Ge, Verdouw and Bogaardt (2017), these can be considered technical challenges and are regarded as the first type of challenges encountered with big data in agriculture. They relate to installing the technological devices, information technology infrastructure and maintaining the power supply and intranet. On the other hand, organisational challenges are challenges related to infrastructure, lack of expertise*,* and the overall management of information technology (IT) systems. The accuracy and privacy of the data being captured are additional issues. Validation and verification methods can be used to authenticate the data, and strict policies should be in place to ensure data security and user anonymity (Carbonell, 2016; Verma *et al.*, 2020). The challenges in big data relate to the factors that influence PA adoption of farmers (Section 2.1). Mindful of these challenges, it is thus helpful to research the challenges related to agriculture and technology adoption before designing a demonstrator tool, such as the one proposed in this research study.

2.4.2 Data analytics

Data analytics is the process of analysing raw data by using various techniques to uncover patterns in the data. There are four main types of data analytics, viz.:

1. *Descriptive Analytics* - Uses data to describe the performance of an entity. It includes data collection, processing, analysis and data visualisation (Schaap, 2020). Charts, graphs, maps, and diagrams can visually represent the data and enable the user to gain insight into past events (Du Preez, 2020).

2. *Diagnostic Analytics* – Also known as “exploratory analytics”, uses the findings from the descriptive stage to determine why something has happened (2U, 2021; Frankenfield, 2021). It attempts to discover unexpected relationships, patterns and trends and detect anomalies in the data (Du Preez, 2020).

3. *Predictive Analytics* - Uses known data by utilising statistical and ML techniques to determine what will most likely happen in the future (2U, 2021). Historical data is used to detect patterns and the relationship between the input and output variables and can perform forecasting,

prediction and estimation to infer what is most likely to happen (Du Preez, 2020; Schaap, 2020).

4. *Prescriptive Analytics* - Seen as the most challenging but most valuable form of analytics. It aims to answer the questions about what should be done (2U, 2021; Schaap, 2020). Prescriptive analytics uses ML techniques to analyse and find patterns to estimate the various outcomes and support data-driven decision-making.

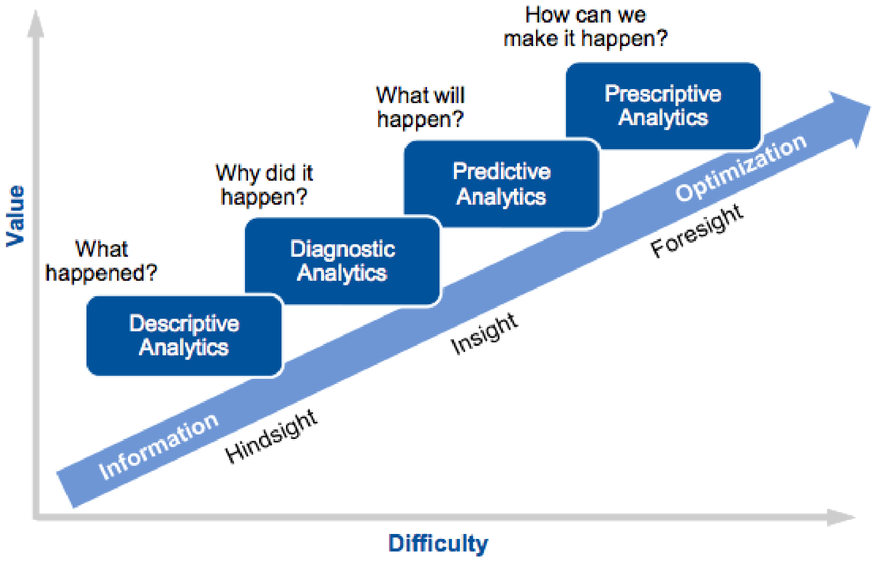


Figure 2.5: Gartner's analytics ascendency model (Schaap, 2020)

The field research was used to explore the use case's nature further and determine where it lies within Gartner's ascendency model. This was used in the data analysis and development of the concept demonstrator.

2.4.2.1 Data analytics methodologies and processes

A few industry-standard methodologies exist for data analytics, but the main methodology remains the CRoss-Industry Standard for Data Mining (CRISP-DM). Other methodologies include Microsoft's Team Data Science Process (TDSP), Knowledge Discovery in Database (KDD) and Sample, Explore, Modify and Access (SEMMA). Several methodologies are discussed for comparative purposes below, emphasising the CRISP-DM methodology used in Chapter 4.

Knowledge discovery in database (KDD)

The term "knowledge discovery in database" was coined in 1989 to refer to the broad process of using data mining (DM) methods to find knowledge in data according to the specification of measures and thresholds (Azevedo & Santos, 2008). KDD is an interactive and iterative process involving using a database along with any pre-processing, sub-sampling and transformation of the data (Shafique & Qaiser, 2014).

Team data science process (TDSP)

Microsoft built the agile and iterative methodology in 2016 to facilitate the successful implementation of data science projects. It includes best practices and structures from industry leaders and comprises six main components (Deeper Insights, n.d.; Microsoft, n.d.). The last stage differentiates the TDSP methodology from the CRISP-DM method. It includes system validation to confirm that the client requirements have been met and ensures smooth roll-out within a company.

Sample, Explore, Modify and Access (SEMMA)

The SAS Institute developed the SEMMA process to describe the process of conducting a data analysis project. It comprises five main stages (Azevedo & Santos, 2008; Shafique & Qaiser, 2014).

CRISP-DM

The CRISP-DM methodology is a hierarchical process model comprising four major phases, generic tasks, specialised tasks and process instances. The CRISP-DM methodology distinguishes between the reference model and the user guide (see Figure 2.6). The reference model provides a quick overview of phases, tasks and outputs, whereas the user guide provides more detailed descriptions of each phase and depicts how to do a data mining project. For the purpose of this study, only the user guide will be discussed below. The CRISP-DM phases discussed below are adapted from Chapman, Clinton, Kerber, Khabaza, Reinartz, Shearer and Wirth (1999), Nisbet, Elder and Miner (2009), and Wirth and Hipp (2000):

1. Business understanding

   This phase aims to assess the requirements, resources, and constraints to understand the problem at hand and determine the business goals and objectives. It also involves compiling a list of risks and potential actions taken and determining the data mining goals in technical terms. Moreover, a detailed project plan can be used to specify the intended duration, resources and iterations of the project, and an assessment of the initial selection of tools and techniques can also be included.

2. Data understanding

   In the data understanding phase, the data is collected, described, and examined to better understand the data. After the data have been explored and the type of data has been identified, any data quality problems and potential solutions are listed. The business understanding and data understanding phases are iterative processes, and after collecting the initial data, some of the business objectives and approach strategies could change.

3. Data preparation

   Data preparation includes all the activities required to construct the final data set used in the model. Preparation activities include data cleaning, transformation, data imputation, data reduction and data derivation. Data can also be integrated by merging tables and aggregating new data records.

4. Modelling

   The first step in the modelling phase involves choosing the actual modelling technique(s) used and listing the specific models' data assumptions and parameter settings. A procedure is then generated to describe the testing that will be done to validate the model performance. For example, planning is required when using a classification algorithm to divide the data set into training, test and validation data sets. The results of the model outcomes should be evaluated, and parameters or data sets can be revised to improve the model results (Chapman *et al.*, 1999; Smart vision, 2021).

5. Evaluation

   The results are summarised and assessed during this stage to determine whether the model has achieved the desired business objectives. Any important risk factors discovered in previous phases should be highlighted, and recommendations for improvement and future work can be reported in this phase.

6. Deployment

   In the final stage, the evaluation results are used to determine a strategy for the deployment and the necessary steps on how to perform them. A monitoring and maintenance plan should also be compiled and included in the final report that will be presented to the concerned stakeholders.

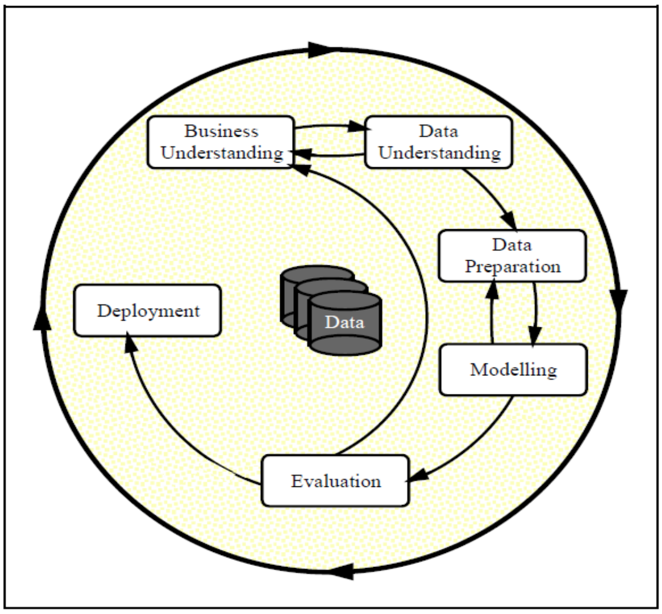


Figure 2.6: The CRISP-DM methodology (Wirth & Hipp, 2000)

2.5 Artificial intelligence (AI) and machine learning (ML)

AI and ML can provide farmers with real-time insight into the farm and are used to better understand the data-intensive processes and environments of agriculture. With the advancement of high-performance computing, AI and ML became popular tools to assist in complex analysis and decision-making, which are often time-consuming for the farmer. Automating some manual elements and simplifying the data analysis and decision processes allow the farmer to prioritise specific farming activities and make data-driven decisions. The definition and few applications of AI are discussed below, followed by the definition, types of ML and popular programming languages available. This research and data analysis methodologies in Section 2.4.2 are used in Chapter 4 for the data analysis.

2.5.1 Artificial intelligence (AI)

The term “artificial intelligence” had appeared in literature as early as the 1950s when Alan Turing published his work “Computing machinery and intelligence” (cited in McCarthy, 2004). McCarthy (2004) defines AI as the science and engineering of making intelligent machines that do not have to confine themselves to the biological methods of a human. ML and deep learning are frequently mentioned in conjunction with AI and are combined with knowledge from computer science, engineering and statistics to try and simulate human intelligence to solve problems (IBM Cloud Education, 2020; Master’s in data analytics, 2020).

AI requires a colossal amount of data to train the machine and can be used in complex speech and facial recognition, weather prediction and medical diagnostics. AI applications in agriculture include irrigation, pest management, livestock tracking, disease detection and yield prediction. Several factors can influence production. A major challenge in agriculture is that it can be a timely process to construct a robust model. For example, crop-specific data is collected on an annual basis. Production performance of each year can vary vastly due to a combination of factors, such as climate, pesticides, fungicides, crop type and soil type. Therefore, a substantial number of years’ worth of data collection may be necessary to provide sufficient data to train the AI models (Dharmaraj & Vijayanand, 2018).

2.5.2 Machine learning (ML)

Liakos, Busato, Moshou, Pearson and Bochtis (2018) define ML as processes that learn from training data to perform specified tasks. A more detailed definition by Du Preez (2020) (cited in Al Sonosy, Rady, Badr and Hashem, 2016) claims that ML is defined as a set of rules that uses mathematical and statistical techniques to assist in identifying patterns and trends and learn from existing data to make predictions and decisions, without explicitly being programmed to do so. Choosing the appropriate type of ML algorithm will depend on the data availability, data quality and desired outcome of the user.

2.5.2.1 Types of machine learning

There are four types of ML algorithms, viz. supervised, semi-supervised, unsupervised, and reinforcement learning, that can be used. Data within this project scope can be categorised as labelled or unlabelled data. Labelled data contain informative labels to provide context to learn from an ML model (AWS, 2021). On the other hand, unlabelled data do not contain tagged labels to classify and identify characteristics.

2.5.2.2 Supervised learning

Supervised learning uses labelled data as input and maps it to the desired known output. The data sets include the correct outputs to allow the model to learn and make predictions. The model's accuracy is measured through a loss function and will adjust until the error is minimised and an acceptable level of performance is achieved (Du Preez, 2020). The two main types are grouped into classification and regression problems, and popular algorithms such as linear regression, random forest and Support Vector Machine (SVM) can be used to address these problems (Brownlee, 2020a).

2.5.2.3 Unsupervised learning

Unlabelled data is given as input without any output (target) data. The algorithm tries to find associations between the given inputs and groups them to predict the desired output. Unsupervised learning problems can be grouped into clustering and association problems, and popular algorithms include k-means for clustering and the Apriori algorithm for association rule learning problems (Brownlee, 2020a).

2.5.2.4 Semi-supervised learning

Semi-supervised learning is seen as a hybrid between supervised and unsupervised learning as the learning model receives both labelled and unlabelled data (Pykes, n.d.). It is often costly and difficult to obtain labelled data, and semi-supervised learning is particularly useful in scenarios where labelled data is scarce (van Engelen & Hoos, 2019). The model is trained on the labelled data, and pseudo-labelling can be used to label the unlabelled data based on the predicted outcomes.

2.5.2.5 Reinforcement learning

Reinforcement learning is a unique type of learning because it does not receive data to solve a problem. An agent needs to navigate an environment and try to achieve a goal or set of goals to achieve the greatest reward (Grokking, 2019; Singh, 2018). The environment reveals itself to the agent as states (s) while the agent influences the environment and takes actions (a) through a trial-

and-error basis. Reinforcement learning is applied in fields such as games, robotics and self-driving cars. (SPIME Analytics, 2020).

2.5.3 Machine learning languages

There are numerous ML languages available. This raises the question of what program language is best for ML? To answer this question, one must consider the problem that needs to be solved. The type of learning problem, data quality, computational power, compatibility, support, and available language libraries all play a role when choosing the right programming language. Five of the main ML languages are discussed below.

2.5.3.1 Python

Python (www.python.org) is one of the preferred programming languages due to its simplicity, consistency and excellent community support and documentation. It is easy to learn, and once you know the basics of Python programming, you can start using the libraries. It boasts a vast number of libraries and tools to support various ML tasks. Python can run on multiple platforms, such as Linux, Windows and macOS (CFI, 2018). Popular ML libraries include Pandas, TensorFlow, Pytorch, Scikit-learn and Matplotlib (ActiveState, 2020). Python is easily integrable with Microsoft PowerBI, Excel spreadsheets, Orange and databases like MySQL and PostgreSQL.

2.5.3.2 R

R (www.r-project.org) is one of the most popular open-source programming languages for statistical modelling and analysis. Various packages are available for data analysis, data sampling, data visualisation, model evaluation, supervised and unsupervised ML applications. R is also a cross-platform language that can easily run on Linux, Windows and Mac and is highly compatible with other languages like Python, Java, C and C++ (DataFlair, 2021; Springboard, 2020).

R supports the natural implementation of matric arithmetic and other data structures like vectors and is often preferred over its Python competitors, NumPy. Where R dominates in some areas, there are also some limitations and disadvantages of the language. R lacks basic security and, therefore, has several restrictions and cannot be embedded into a web application. The language has a steep learning curve compared to Python, and first-time users may find it difficult to learn. It is also much slower than Python, Java and Julia. R utilizes more memory because the physical memory stores the objects and can pose problems in working with large data sets (DataFlair, 2021; Krill, 2015). A few ML libraries for R include dplyr, tidyr, ggplot2, lubridate, mice, lattice and caret (Springboard, 2020).

2.5.3.3 Java and JavaScript

Java (www.java.com) is an older open-source programming language that provides a powerful base for application development with efficient coding and debugging. It is characterised by its static-typing syntax, which is much faster than a language like Python, which has a dynamic-typed syntax. Java codes are often described as overly complex and long. Some AI libraries include: TensorFlow.js, Keras.js, RapidMiner, JGAP, Watchmaker, Apache Jena, Jenetics and Deeplearning4j.

2.5.3.4 Julia

Whilst many programming languages were developed between the 1960 - 1980s, Julia (www.julialang.org) was only introduced in 2012. The developers wanted to address the disadvantages of other programming languages and create a language that incorporates the same computational capabilities as MATLAB, be comparably fast as C and be as simple as Python. Julia was created for complex linear algebra, data science and ML (Medina, 2020). Julia is object-orientated, and the syntax is easy to understand and is effective in computational statistics and numerical calculations (SPEC INDIA, 2021).

2.5.3.5 Scala

Scala (www.scala-lang.org) is a well-known language that combines object-orientated and functional programming. Its static types help avoid bugs and are also highly compatible with Java frameworks and libraries. In addition, it has a strong backend and can manage enormous amounts of data and dataflows (SPEC INDIA, 2021). Popular libraries include Breeze, Spire, Saddle and DeepLearning.scala (Krykowski, 2021).

2.5.3.6 Summary

The main advantages and disadvantages of each programming language are summarised in Table 2.4 below. Previously, R was the preferred ML programming language, but other languages have gained momentum in the ML domain in recent years. R and Python are still considered the top competitors and are very similar in ability and performance.

Table 2.4: Advantages and disadvantages of ML programming languages.

Programming Language	**Advantages**	**Disadvantages**
Python	• Easy to learn • Powerful libraries • Cross-platform compatibility and integration (SQL, PowerBI) • Free and open source • Community support	• Slower than most languages • Struggles to support multithreading
R	• Large variety of libraries	• Steep learning curve

Programming Language	Advantages	Disadvantages
	• Free and open source • Cross-platform compatibility • Excellent community support • Does not require a compiler	• Poor memory management • Slow speed • Poor security
Java	• Object-orientated • Robust, secure and platform-independent • Community support • Uses multi-threaded environment to run various threads separately	• Moderate learning curve with high entry point • Requires more memory • More costly due to higher processing and memory requirements
Julia	• Object-orientated and functional • Easy syntax • Free and open source • Less libraries and scientific tools than Python and R	• Less resources and community support than other languages • Libraries are not well-maintained
Scala	• Compatible with Java • High performance	• Slow compiling • Steep learning curve • Limited commercial support and documentation

2.5.4 Visualisation

It is often difficult for the human eye to detect patterns and relationships when exploring large data sets without statistics and visualisation tools. Different data types can be visualised by using heat maps, bar charts, radar charts, pie charts, histograms and clustering, to name but a few. Moreover, dashboards can be used to integrate the visualisations to assist with the interpretation of data and decision support.

2.5.4.1 Orange

Orange (www.orangedatamining.com) offers many data visualisation options and helps users gain insight into the data rather than the programming. It is interactive software that allows the user to select data subsets from graphs, plots and tables. Some visualisation widgets include scatter plots, box plots, histograms, heat maps, classification trees and even silhouette plots, mosaic and sieve diagrams, which much other software do not include. These widgets are easily customisable. Orange also offers a clever reporting option that compiles a summary of the desired data and visualisations. The widgets are used to build the base layer upon which data sets can be connected to the infrastructure to be visualised. Orange is compatible with Microsoft Excel and Python, and it has excellent user support on its website.

Figure 2.7 displays the drag-and-drop widgets that can be selected on the user interface to make predictions with classification trees and logistic regression. Figure 2.8 displays a simple box plot based on the popular Iris data set.

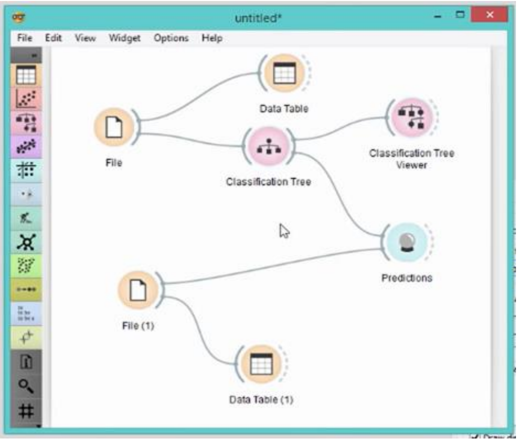


Figure 2.7: An example of widgets used for data analysis in Orange data mining (Orange, 2021)

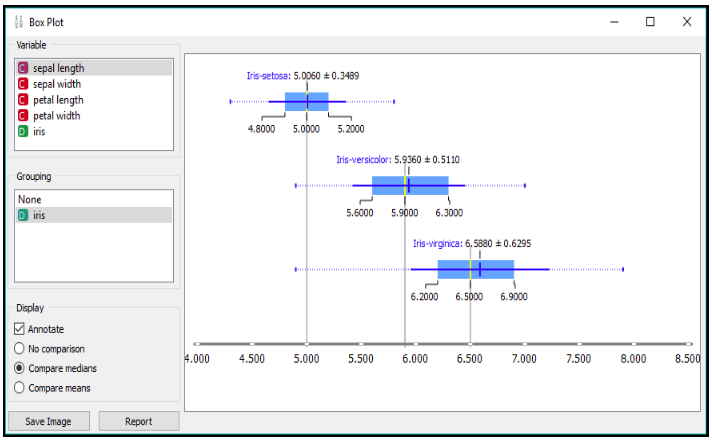


Figure 2.8: Box plot - An example of widgets used for data analysis in Orange data mining (Orange,2021)

2.5.4.2 Microsoft PowerBI

Power BI (www.powerbi.microsoft.com) is a Microsoft-based real-time BI software program that offers on-premises and cloud access to data. The software has built-in visuals, allows for customised visualisation, and the user can even publish the desktop dashboard online. Users can collaborate and share the visualisations and data, and there is also large support and an assisting community if the user requires help. The software is free, and additional features can be purchased if needed. PowerBI's powerful Excel integration allows the user to select, filter or slice data in a PowerBI report or dashboard and transfer it back into Microsoft Excel. More useful features include the mobile application and easy integration with the Structured Query Language (SQL) server.

One of the limitations of the software is that the free version's data storage is limited to 2 GB and requires a purchase upgrade if larger volumes of data processing are required. This can be solved by accessing data directly from the server. In addition, PowerBI is not ideal for visualising complex relationships between tables on the dashboard (Van Rooyen, 2019). Figure 2.9 shows a conceptual management dashboard that was designed and developed for the Stellenbosch Learning Factory to enable a user to view critical KPI's.

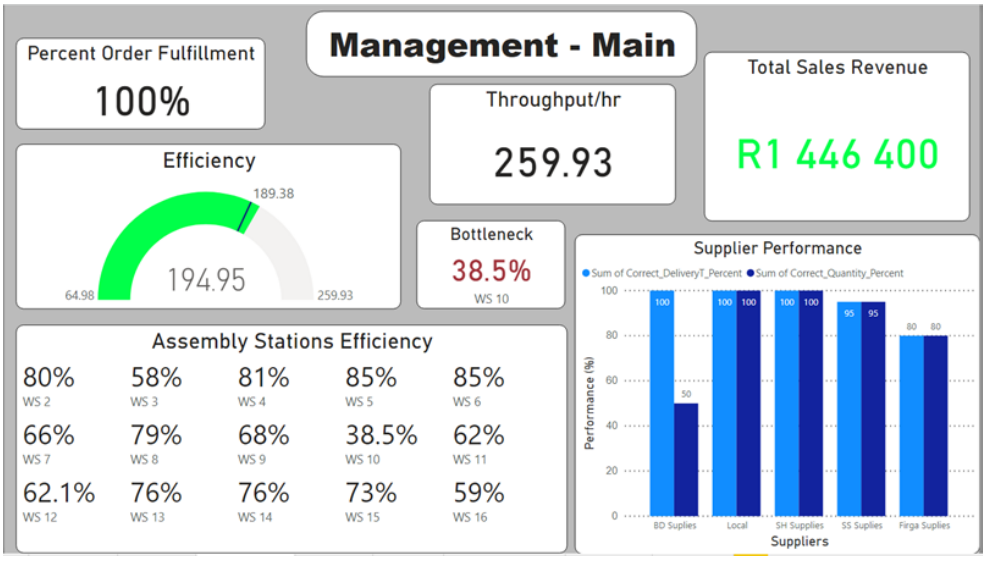


Figure 2.9: Management dashboard designed in PowerBI

2.5.4.3 Tableau

Tableau Public (www.tableau.com) is a popular free BI software program that includes a drag-and-drop interface to customise user dashboards. The user can publish the dashboards and share them live on the web or smart mobile devices. Tableau does not require any programming knowledge and provides real-time data visualisation. Tableau also provides a collaborative working environment where dashboards can be shared with chosen users (Tableau, 2003). An example of a graph from Tableau's website can be seen in Figure 2.10 (see overleaf).

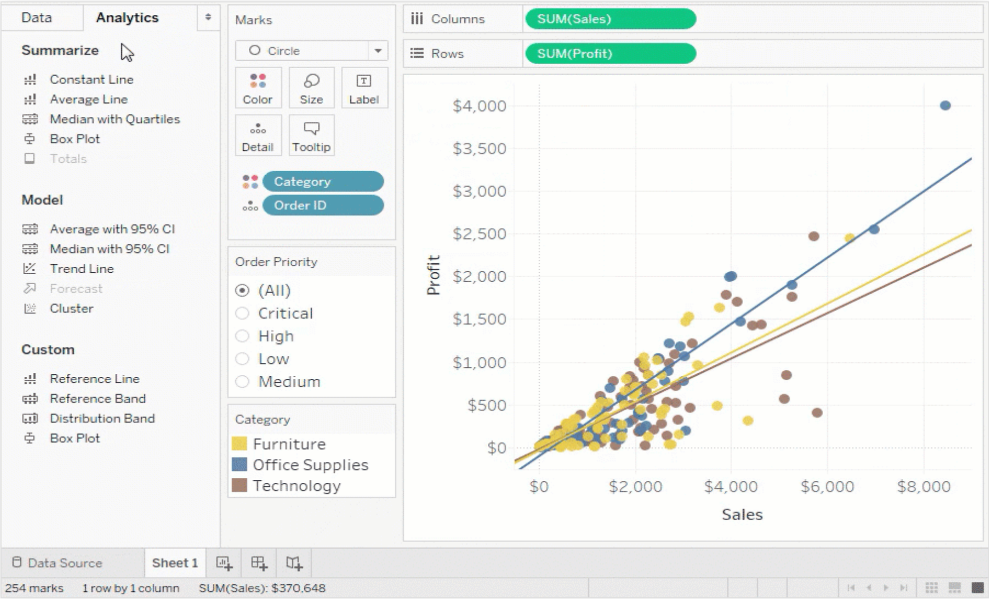


Figure 2.10: Tableau concept visualisation indicating profit versus sale data

2.5.4.4 Looker

Looker (www.looker.com) is an entirely web-based platform that provides a free educational version (Looker, 2021). It has its own proprietary modelling language called LookML, which is seen as an improved version of SQL and defining queries (Nine Boards, 2020).

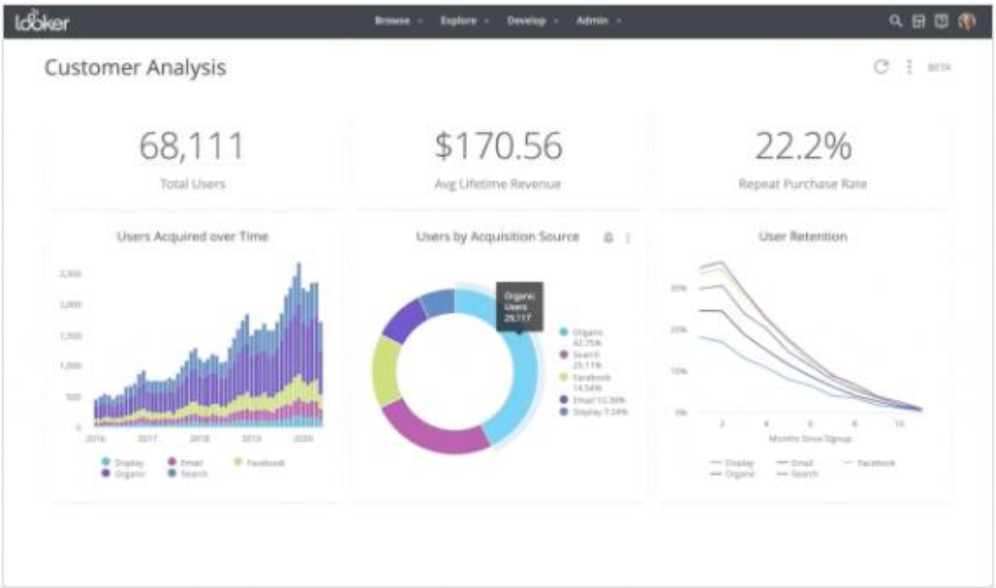


Figure 2.11: Looker concept dashboard

All of the software mentioned above presents powerful features for data insight and visualisation. However, the chosen visualisation tool will depend on the client requirements and objectives of the study to ensure that the desired outcomes are met.

2.6 Crop and production management

Crop management refers to all the agricultural processes involved to ensure optimal productivity in the field. Processes such as soil preparation, planting, fertiliser, irrigation, pest and disease management, harvesting and post-harvesting activities can help to provide accurate and up-to-date field crop records. Vegetation indices, pests- and disease management, as well as yield prediction are discussed below.

2.6.1 Vegetation indices

Soil quality (SQ) assessments are fundamental for increasing agricultural productivity and designing more sustainable land management practices. SQ depends on factors such as climate, soil and the type of crop planted. Soil health indicators (SHI) can be used to monitor the SQ and play a vital role in the communications between the land managers and other stakeholders involved (Eze, Dougill, Banwart, Sallu, Smith, Tripathi, Mgohele & Senkoro, 2021; Viana, Farhate, de Souza, Cherubin & Carneiro, 2020). Remote sensing can be used to monitor biotic and abiotic stresses in plants.

2.6.1.1 Biotic health indicators

Soil samples can be taken and used to collect information about the condition of the soil. Data such as magnesium (Mg), potassium (K), calcium (Ca), phosphorus (P) and pH can be used. Remote sensing, however, can tell us more about the state of the land and crops. A few of the major crop health indicators are discussed with their formulas below. Multispectral sensors are used to capture indices such as NDVI, Modified Chlorophyll Absorption in Reflective Index (MCARI), Normalised Difference RedEdge Index (NDRE) and soil moisture levels.

Normalised difference vegetation index (NDVI)

NDVI is one of the most common indicators in agriculture and is used to assess whether an area contains live green vegetation by capturing how much more infrared (IR) light is reflected compared to visible red light. NDVI can be used to differentiate between crops and crop stages, differentiate bare soil from grass or forest, and detect plants under stress (Nuno, 2014). The value varies between -1.0 and +1.0, with zero indicating no green vegetation and values close to +1 showing high-density green leaves (NASA EO, 2000). Vegetation properties such as LAI, biomass and chlorophyll can be derived from the index.

Soil adjusted vegetation index (SAVI)

The soil adjusted vegetation index (SAVI) accounts for the variation in soil type and soil properties. Areas of low vegetative cover influence light reflectance in the visible red and NIR spectra (< 40%). This can be problematic when different soil types and crops are being evaluated due to the difference in reflectance of red and IR wavelengths. The accuracy of the NDVI decreases with variables such as soil colour, soil moisture and saturation from high-density vegetation. SAVI was developed to improve the shortcomings of NDVI and minimise the influence of soil brightness in the red and NIR wavelengths (Olukayode, Blesing, Rotimi & Oguntola, 2018; The landscape toolbox, 2012). SAVI ranges between -1 and +1, and a lower value indicates the amount of green vegetation (Olukayode *et al.*, 2018).

Modified soil-adjusted vegetation index (MSAVI, MSAVI2)

MSAVI, later revised as MSAVI2, was developed by (Qi, Chehbouni, Huete, Kerr & Sorooshian, 1994) to address some of the limitations of NDVI for areas with high exposed bare soil due to minimal vegetation or a lack of chlorophyll. SAVI requires specifying the soil-brightness correction factor(L) for the vegetation. The problem with this is that it is based on a trial-and-error specific to the amount of vegetation in the study area. Still, most use the standard L value of 0.5, leading to inaccurate calculations (The landscape toolbox, 2012).

Leaf Area Index (LAI)

LAI is the total leaf area per unit ground surface area. It tells us how many layers of leaves would be on the ground if it were to fall and be arranged exactly side-by-side. The leaves in the canopy are arranged randomly, and, therefore, light can still often reach the ground surface with an LAI value greater than one (1) (Gabron, n.d.). LAI is dimensionless and measured as a ratio of leaf area per ground surface area [m^2/m^2]. An LAI value of three (3) means that the study area has a leaf area to ground surface area ratio of 3:1. Some desert ecosystems would have an LAI value of less than one

(1), while shrublands typically have values between three (3) and six (6). Tracking the LAI of a maize farm from seeding to maturity could range from zero (0) to six (6) (Campbell, n.d.).

Chlorophyll index (CI)

The index incorporates the CIgreen and CIred-edge spectrum bands to calculate the total chlorophyll in plants and provide meaningful insight into plant health. The two bands respond to variations in chlorophyll content and are consistent for most plants (EOS, 2021). Common uses of CI include yield prediction, improving crop distribution uniformity, identifying nutrient deficiencies and assisting in target tissue sampling. Patterns detected in CI were found to be highly correlated with final crop yield in the fall (Ceres, 2021). Other relevant vegetation indices are shown in Table 2.5 below.

Table 2.5: Other relevant vegetation indices (Kulbacki, Segen, Knieć, Klempous, Kluwak, Nikodem, Kulbacka & Serester, 2018)

Index	Formula	Spectral bands	Sensor	Application	Source
Leaf rust disease severity index 1 (LRDSI_1) and 2 (LRDSI_2)	$LRDSI\_1 = 6.9\frac{RED1}{BLUE} - 1.2$ $LRDSI\_2 = 4.2\frac{RED2}{BLUE} - 0.38$	*BLUE*: 455 *RED*: 605 *RED*: 695	Ground-based FieldSpec - spectrometer	Detection of wheat leaf rust	Ashourloo, Mobasheri & Huete (2014)
Normalised Pigment Chlorophyll Ratio Index (NPCI)	$NPCI = \frac{RED1 - BLUE1}{RED2 + BLUE2}$	*BLUE*: 460 *RED*: 660	Ground-based radiometers	Estimation of leaf chlorophyll content	Hatfield & Prueger (2010)
Normalised Difference Water Index (NDWI)	$NDWI = \frac{NIR1 - NIR2}{NIR1 + NIR2}$	*NIR*1: 841 - 876 *NIR*2: 1230–1250	Satellite (MODIS)	Estimation of plant water content	Zarco-Tejada, Rueda & Ustin (2003)
Structure Insensitive Pigment Index (SIPI)	$SIPI = \frac{NIR - BLUE}{NIR - RED}$	*BLUE*: 445 *RED*: 680 *NIR*: 800	Handheld Spectroradio-meter	Determine the sunn pest damage on wheat	Genc, Genc, Turhan, Smith & Nation (2010)
Damage Sensitive Spectral Index (DSSI)	$DSSI = \frac{RED - NIR - BLUE - GREEN}{(RED - NIR) + (BLUE - GREEN)}$	*BLUE*: 509 *GREEN*: 537 *RED*: 719 *NIR*: 873	Handheld Spectroradio-meter	Determine the sunn pest damage on wheat	Genc *et al.* (2010)

There are numerous vegetation indices available that serve different purposes. Figure 2.12 (see overleaf) visually illustrates how PA components such as remote sensing, IoT, GIS and vegetation indices can be integrated to capture and display different layers that can be used in farm management. Indicators such as NDVI, weather and crop moisture layers can be utilised to gain more insight and assist the farmer in the decision-making process.

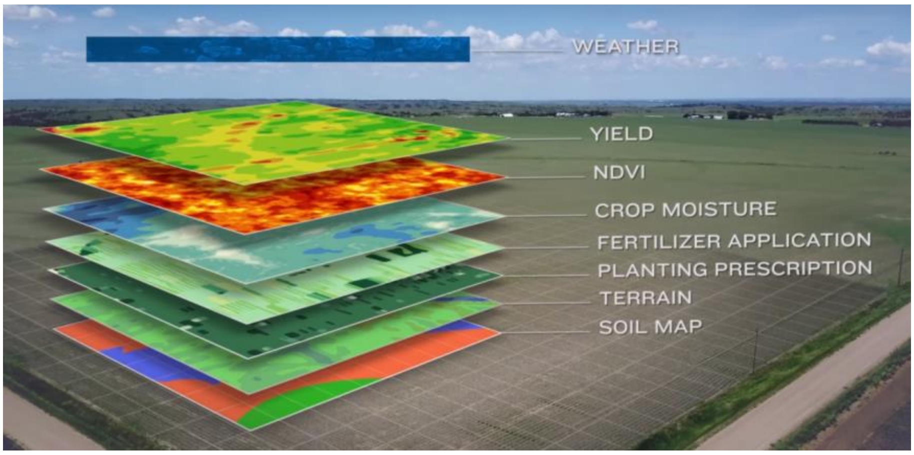


Figure 2.12: Multiple layers for precision farming applications (Loizos, 2017)

2.6.1.2 Abiotic health indicators

Abiotic health indicators are physical, non-infectious factors contributing to plant health. Moisture and temperature extremes, soil properties, fertility imbalance, physical injuries and chemical toxicity are common examples of abiotic disorders. The soil structure determines the ability to hold water, oxygen, and nutrients and its availability to plants. Compaction is a common issue in soil structure, which accounts for the pore space for root growth. Compaction can occur from heavy farming equipment traffic, impact from rain and minimal crop rotation. Clay soils are especially known to have smaller pore space and can easily become compacted, which can cause low oxygen levels for the root respiration system (Kennelly, O'Mara, Rivard, Miller & Smith, 2012).

2.6.2 Disease and pest management

Plant pathology refers to managing plant disease by studying the interaction between the organisms and the varying environmental conditions and the effects on plant growth, yield, and quality (University of Stellenbosch, 2013). Pest and disease management is essential to effective crop production. Quantifying the impact of pests and disease on crop performance is still a challenge for the scientific community (Donatelli, Magarey, Bregaglio, Willocquet, Whish & Savary, 2017). Farmers should regularly inspect their lands to identify insects and disease problems and to stop potential problems. Certain pests and diseases may be treated curatively, while others should be treated preventatively. Farmers can counter these problems by using a combination of farming practices such as crop rotation, pest tolerant cultivars, certified seed, pesticides and proper soil preparation and management.

Referring to Section 2.6.1, soil quality plays a vital role in plant health. Organisms planted in good soils can withstand more environmental stress and diseases than those planted in poor soils. Climate change can also influence insect populations and disease outbreaks, thus creating the need for farmers to constantly assess their crops and make timely decisions (South Africa, 2021). The South African Agricultural Research Centre identifies the diseases and pests shown in Table 2.6 as the most important in wheat disease and pest management.

Table 2.6: Typical pests and diseases found in South African wheat (Agriculture Research Council, 2014).

Disease	Pests
Powdery mildew	Russian wheat aphid
Rust	Other aphids
Tan spot	Brown wheat mite
Bacterial streak	False wireworm
Black chaff	Bollworm
Ergot disease	False armyworm
Basal glume rot	Leaf miner
Eyespot	Black maize beetle

Several studies regarding pest and disease management have been published. Yang, Rao, Elliott, Kindler & Popham (2009) conducted a study to determine the feasibility of remote sensing techniques to detect two different stresses in wheat caused by Russian aphid and greenbug infestation. Ratio-based vegetation indices were used to differentiate the two stresses in wheat, and researched the use of deep-learning architectures to classify soybean pest images achieved accuracies of up to 93.82% (Tetila, Machado, Astolfi, de Souza Belete, Amorim, Roel & Pistori, 2020). Ali Al-windi, Abbas and Mahmood (2021) developed a new method for detecting wheat stem rest disease. Image processing was used to convert Red Green Blue (RGB) to hue saturation value and performed feature selection to improve the accuracy of the chosen neural network.

2.6.3 Yield prediction

Yield prediction is an essential component in PA and can help farmers decide which crops to grow and when to grow them. Yield prediction can be used in yield mapping in conjunction with demand requirements and expected profitability. Many studies have used growth status and trend monitoring, but most are based on a single agronomic parameter. A few studies have combined multiple parameters into a more comprehensive yield estimation system.

Jégo, Pattey and Liu (2012) conducted a study in Canada to evaluate the conditions regarding the application of re-initialisation (e.g., number of image acquisitions and spatial resolution). Remote sensing was used to provide LAI data to re-initialise STICS, a crop prediction model, to evaluate the performance of biomass and yield prediction. Green LAI was estimated with the modified transformative vegetation index using airborne hyperspectral sensors and multispectral satellite sensors. Re-initialisation of seeding data, seeding density and field capacity greatly improved the prediction with a root mean squared error (RMSE) of 13% for yield and 23% for biomass. Another study by Brown, Hochman, Holzworth and Horan (2018) explored the benefits of using the Predictive Ocean Atmosphere Model for Australia (POAMA) climate model over historical climate to predict wheat yield in Australia. POAMA consists of daily temperature, radiation and rainfall and was used as inputs for the agriculture production systems simulator (APSIM) crop model to help predict yield. The climate model forecasts have a narrower prediction range but at the expense of a higher number of misleading forecasts. The study concluded that the climate model and historical climate are both useful but provide different advantages and should be combined in future research.

Several papers use a combination of vegetation indices and climate data to conduct crop yield predictions. Some indicate the use of existing crop simulation models and add variables to test the performance of the yield predictions. Hao, Ryu, Western, Perry, Bogena and Franssen (2021) used the APSIM classic-wheat model and conducted a meta-analysis on 30 simulations containing observed yield. APSIM simulates soil water, nutrients and crop growth processes under varying environmental and management conditions. APSIM's 'WHEAT' model also includes water stress, nitrogen stress and heat stress to investigate the factors influencing the yield prediction performance. Heat- and frost stress were found to cause large discrepancies in grain yield prediction. Grain is particularly sensitive to short-term heat stress in the anthesis and grain-filling stages. The study hypothesised that the discrepancies could be due to the use of mean daily temperatures. Site-specific calibration of the model resulted in an RMSE of smaller than 1t/ha and normalised RMSE (NRMSE) of 28%. Hassanijalilian, Igathinathane, Doetkott, Bajwa, Nowatzki and Haji Esmaeili (2020) developed a low-cost infield method to measure chlorophyll using smartphone digital imaging and ML models. The chlorophyll content is indicative of plant growth and health issues. The researchers claim that the method can easily be extended to other crop types and large-scale aerial imaging platforms. Additional research regarding variables and technologies, which influence yield prediction are summarised in Table 2.7 below.

Table 2.7: Examples of yield prediction research in literature

Application	**Summary**	**Source**
Benefit of satellite-based solar-induced chlorophyll fluorescence (SIF) in crop yield prediction (USA)	• The performance of using SIF data for yield prediction was compared to satellite-based vegetation indices performance - NDVI, NIRv, and land surface temperature (LST). • Five ML algorithms were used to evaluate the performance of remote-sensing and climate-remote-sensing predictions – LASSO, Ridge, SVM, ANN and RF.	Peng, Guan, Zhou, Jiang, Frankenberg, Sun, He & Köhler (2020)
Yield evaluation indicator based on hyperspectral improved fuzzy method (China)	• Development of a new comprehensive yield evaluation indicator (CYEI) that monitors crop growth and yield estimation. • Used winter-wheat data between 2012 - 2018 with different soil moisture and nitrogen fertiliser treatments. LAI, biomass, leaf nitrogen and leaf water content was used with the CYEI indicator to monitor crop growth and estimate yield.	Xu, Nie, Jin, Li, Zhu, Xu, Wang & Zhao (2021)

Application	Summary	Source
NDVI, rainfall and temperature data to predict wheat grain yield (Morocco)	• Used regression models and 10-daily NDVI, rainfall sums and average monthly temperatures to predict provincial and national wheat yields. • NDVI was the most important predictor influencing yield prediction and explained 40% of yield variation in the provincial study. Rainfall and temperature gained more significance in arid areas.	Balaghi, Tychon, Eerens & Jlibene (2008)

Stemming from the above, it is evident that a combination of high-resolution remote sensing information, soil properties, climate- and yield data, and ML can contribute to improved performance of yield prediction models. Chlorophyll, NDVI and LAI indices are featured in yield prediction studies, especially wheat yield prediction.

2.7 Decision support systems (DSS) and early warning systems (EWS)

DSSs have been investigated and implemented for almost 40 years since the widespread use of computers. Holsapple and Burstein (2008:22) define a DSS as a “computer-based system that represents and processes knowledge” that assists in more agile and innovative decision-making. Marin (2008) describes it as an information system that collects and analyses data supporting business and organisational decision-making activities by providing access to information and the appropriate analysis tools. The accuracy of the decisions is based on the quality of the data and the analysis process to discover trends to create solutions and strategies.

A typical DSS consists of a knowledge base data management system, model management system and user interface (CFI, 2015). The knowledge base includes data collected from several sources, whereas the model management system holds the models used for decision-making. The user interface is the output data after the data have been processed and the decisions have been made. A DSS assists users in evaluating historical and present data, forecasting future trends, considering alternative decisions and potentially helping an organisation to make optimal decisions. Decision support applications that only collect and organise data and do not suggest specific decisions are called passive models. Active DSSs collect, analyse and then incorporate human input to revise the model (Marin, 2008). The types of decision-making can be grouped into strategic, tactical, and operational decisions and continuous monitoring. According to Power (2002), there are five types of DSSs: communication, data, knowledge, model and document-driven, as shown in Figure 2.13 below. These DSSs are used to provide group, knowledge-based or organisational support. Marin (2008) also defines three DSS levels: technology, human and computer inputs, and the developmental approach to designing the DSS.

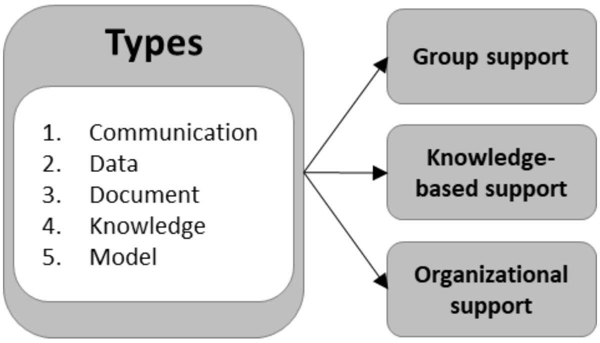


Figure 2.13: Types of DSSs and the support they provide (adapted from Marin, 2008; Power, 2002)

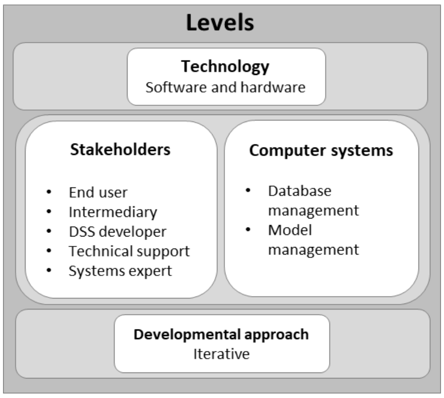


Figure 2.14: The three levels of a DSS (Marin, 2008)

Decision support and EWS often work hand-in-hand to identify and mitigate potential risks. EWSs is often used in disasters risk management applications, which mostly involve natural disasters such as landslides, earthquakes, floods and tsunamis. EWSs use forecasting and prediction strategies to alert the user or affected parties.

2.7.1 Decision support in agriculture

With the recent advancements in technology and overwhelming amounts of data, farmers are faced with difficult decision-making choices. DSSs can help by suggesting evidence-based and precise decisions to address the challenge of transforming data into knowledge and actionable intelligence (Zhai, Martínez, Beltran and Martínez, 2020). Agricultural decision support systems (ADSS) are used for decision support in various agricultural applications. The IBM Watson decision support platform for agriculture, for example, is a popular AI-driven ADSS. It provides accurate weather data (historic, near real-time and forecasted), soil data (soil type and moisture, nutrient content and fertility), equipment data (IoT sensors) and imagery (satellites and UAVs). The platform utilises AI, ML, and advanced analytics to extract valuable insight and generate guidance in decision support (IBM, 2018). Microsoft's Azure FarmBeats (www.microsoft.com) was developed in 2014 and provides the farmer with data-driven insights. The system creates digital maps from data collected from various remote sensing devices. AI and ML models are used to make predictions and provide the farmer with actionable insight (Agrawal, 2020). Other popular agriculture decision support software includes AgVend, Bayer-ClimateFieldView, Taranis and FluroSat.

Tyrychtr and Vostrovsky (2017) researched ADSSs and used the Software Quality Requirements and Evaluation (SQuaRe) standard as evaluation criteria. This standard examines the accessibility, scalability, interoperability, functionality and completeness of an ADSS. The graphical user interface (GUI) should be easy to understand (accessible), while scalability refers to adding new sensors to improve the system's functionality. ADSSs can easily integrate with external sources, for example, external weather stations, and have high interoperability. There are several benefits of implementing

ADSSs, but it is important also to examine its challenges and limitations.

DSSs are designed to eliminate bias when making decisions. However, this can also have the opposite effect, and the user can become too dependent on the system to make proper decisions. This is because certain assumptions are made when the DSS is designed, and it can sometimes be difficult to quantify certain data in the system. Thus, a DSS may, for example, lead to information overload for the user, as it considers a vast amount of data and alternatives that are not always necessary for certain decisions to be made. This is exactly why it is referred to as “decision support”, i.e., the users should use the system to guide them with the decision process (CFI, 2015; Juneja, n.d.).

Zhai *et al.* (2020) mention additional obstacles regarding DSSs:

- Not all farmers are confident in using new technology, and complex DSSs often require training and expert experience.
- Several factors can influence decision-making, and hence, there is a need to develop customisable ADSSs that are scalable and can adapt to various crop types.
- Many ADSSs are limited or task-specific, and the farmer often has to combine several ADSSs to manage agricultural activities.
- Fundamental factors such as climate change, drought and pests can lead to irregular patterns and trends that can cause the decision tool to suggest inaccurate decisions.
- ADSSs require mass data to improve decision-making and accuracy.
- Current ADSSs have not yet reached fully autonomous intelligence, and hence it is necessary to incorporate human expert knowledge.

Ways in which ADSSs are currently being implemented were researched, including comparisons of the components that are utilised in practice. The description, applications, and components of the examined ADSSs are summarised in Table 2.8 (overleaf) and are grouped by application: (i) Water resource management and irrigation; (ii) pests and disease, (iii) management zones;(iv) climate and GIS and (v) livestock.

Table 2.8: Applications of DSSs in agriculture

Application	DSS name	Crop or animal	What it does	Components and sources	AI, ML	Resource
Water and irrigation	Smart irrigation decision support system (SIDSS)	Citrus trees (Spain)	• Estimates weekly irrigation needs of a plantation using meteorological data, crop characteristics and soil measurements. • Provides an irrigation report with water usage and irrigation time. • Uses ML algorithms to remove redundant variables and minimise estimated errors.	Soil sensors Weather stations SIDSS	Partial least squares regression, Adaptive neuro-fuzzy inference systems (ANFIS)	Navarro-Hellín, Martínez-del-Rincon, Domingo-Miguel, Soto-Valles & Torres-Sánchez (2016)
Water and irrigation	Fuzzy decision support system (FDSS)	Corn Kiwi Potato	• Determines the irrigation amount based on the growing degree days, total water applied to the crop, and crop evapotranspiration. • A fuzzy soil moisture model was applied to IRRINET and calibrated with data from IRRINET crop database. • Consists of three main parts: Predictive soil moisture model, irrigation inference system deciding timing and amount. Irrigation performance index (IPI) consists of the sum of past irrigations.	IRRINET agro-meteorological database	Inference system developed in MATLAB (fuzzy C-means algorithm	Giusti & Marsili-Libelli (2015)
Pests, disease and weed management	Integrated Pest management system (IPM)	Vineyard	• Strategic, tactical, and operational levels decision support for pesticide application. • Reduce risk to human health and environment. • Reduce labour and pesticide cost as well as increase crop quality and quantity.	UAV and satellite sensing	Not specified	Rossi, Caffi & Salinari (2012)
Livestock	Not specified	Cattle, pigs, sheep, chickens	• Paper introducing data-driven DSS and challenges for ADSS animal health and greenhouse gas emissions. • Incorporates ML, statistical analysis and simulation tools. • Research articles include applications in cattle behaviour, growth trajectories of chickens and pig waste disease detection.	UAV, RFID, other sensors	Bagging ensemble with tree learning, Gaussian Mixture Modelling (GMM), SVM and other algorithms.	Niloofar, Francis, Lazarova-Molnar, Vulpe, Vochin, Suciu, Balanescu, Anestis & Bartzana (2021)
Management	Fast Mapping	Wheat (Argentina)	• Interactive web application that automatically cleans raw spatial data, generates and creates field maps to identify management zones using multivariate classification.	R language	Not specified	Paccioretti, Córdoba & Balzarini (2020)
Climate	'Simulador de Agricultura' (SIMAGRI)	Maize Soybean Wheat (Uruguay)	• An agro-climate DSS that supports strategic and tactical decisions in crop production by utilising historical climate and probabilistic seasonal forecast data. • Comparisons of management practices (planting dates, crops, fertilisers etc.) and environmental conditions.	[1]DSSAT models, GUI developed in Python	Not specified	Han *et al.* (2019)

[1] DSSAT – Decision Support Systems for Agrotechnology Transfer. A modular based application package of various models that can simulate crop growth for 42 different crops under specific management practices (www.dssat.net).

Application	DSS name	Crop or animal	What it does	Components and sources	AI, ML	Resource
Climate and GIS	CROPGRO-Peanut model	Groundnuts (India)	• Response to climate change scenarios. • Simulated crop yield and crop maps	GIS, GPS, crop models, weather stations, DSSAT	Prediction models and simulation	Kadiyala, Nedumaran, Singh, Irshad & Bantilan (2015)

2.8 Summary

A global need for more sustainable and efficient agricultural output to meet the world's food demand is evident. PA is seen as an answer to this problem. PA consolidates various technologies to help the farmer improve farm management activities by providing data-driven and evidence-based decision support. Various remote sensing platforms and IoT sensors can be employed to collect different types of farm data. Several studies highlight the value of real-time, historical and forecasted climatic data in PA. Big data, data analysis, and ML can be used to process and analyse the data to extract information and transform it into actionable intelligence. Section 2.4.2 discussed the different types of data analytics and presented four data analytic methodologies for processing data used in Chapter 4. A brief explanation of AI was followed by a review of various types of ML algorithms and a comparison of the most popular programming languages used for ML.

Important factors influencing crop management and yield prediction, supported by several practical examples discussed in the literature, were expanded upon in Section 2.6. Sections 2.2 to 2.6 explored how these can be used in DSSs. The types of decisions that can be made and the components of a DSS were discussed in Section 2.7. A more extensive review was conducted on the use of DSSs for ADSSs. It was found that due to the numerous factors influencing farming production, most ADSSs only focus on singular applications such as irrigation, pests, disease, climate, crop production, and livestock management. Thus, there exists a demand for more integrated and customizable ADSSs.

Chapter 3
Field research

The object of the field research was to examine the product of *Company A* and the data it provides to clients, as well as the client farm, referred to as '*Farm X'*. The knowledge gathered from the literature study and field research was applied in the data analysis section and used in the design of the concept demonstrator. The field research shows what is currently being done and used to improve the 'manual element' of the decision-making process.

The rest of the chapter discusses the process of acquiring sufficient and accurate weather data and explores wheat and soybean conditions in summer rainfall areas. The grain research was used in the data analysis phase (see Chapter 4). This contributes towards a better understanding of the data and aid in decision support development.

3.1 Company A background

Company A specialises in agronomy, horticulture, soil science, microbiology, geographic information systems (GIS), chemistry and process and production engineering. *Company A* provides its clients with an interactive, cloud-based platform that collates large volumes of data captured from various sources, including remote sensing from satellites, soil moisture probes, tracking devices, pest traps and laboratory soil samples. The sources are used to collect and display data such as soil physical and chemical maps, pest monitoring, leaf and tissue analysis, yield maps, water analysis and Airbus Verde biophysical parameters. After consulting the data on the platform, clients can request advisors to assist them with the interpretation and decision-making. One of the research goals of this study was to investigate how and to what extent the latter actions can be automated and/or accelerated. As indicated before, the concept demonstrator was used for this purpose.

Company A's web-based platform provides the client with an aerial view of the farm. The client can then use the platform to overlay soil physical and chemical maps, pest monitoring data, leaf and tissue analysis, yield maps, water analysis and remote sensing biophysical parameters. The client can also select s number of desired features to superimpose on the aerial view. However, as indicated above, inspecting the overlays with several features can be a timely and complex process, and finding optimised solutions is not a trivial exercise.

The literature study indicated clearly that agricultural decisions strongly rely on meteorological data. *Company A* provides the user with a daily weather update from OpenWeatherMap. This provides the user with current temperature, pressure, humidity, windspeed, sunrise time, sunset time, and weekly minimum and maximum temperatures. However, historical weather data is crucial for forecasting and pattern prediction. As part of this research, the South African Weather Services

(SAWS) was approached to provide the historical weather data specifically relevant for *Farm X*.

3.2 Farm X

Farm X was chosen by *Company A* as the subject farm, as significant data for this farm are available on their cloud platform. *Farm X* is located in the Limpopo province of South Africa and has a summer rainfall climate. From 2016 to 2020, soybeans were planted in summer, while wheat was planted during the winter season.

The data and data sources available on *Company A*'s platform for this farm are briefly discussed below:

Soil classification – Soil samples and probes were used in 2015 to determine the soil physical elements. Full soil classification can be costly, and according to *Company A,* the modus operandi is to complete a full analysis every three years. These values are considered to be "static" variables.

Pest and diseases – A worm infestation broke out during 2019 and 2020, but no coordinate specific data is available.

Remote sensing – The client can select an Airbus Verde subscription, which provides soil health indicator data, such as chlorophyll, as a time series. The satellite provides images when passing over the farm, but they are affected when obstructed by cloud cover. The Verde service can provide de-clouded images, but this results in missing values and a random spread of time-series data. Only chlorophyll data is provided for *Farm X*, with no other crop indicators or yield data.

3.3 Acquiring meteorological data

The SAWS (www.weathersa.co.za) provides a Google Earth file (kmz file) on their website under their "climate services", which indicates all the weather stations found in South Africa (see Figure 3.1). The station code in the vicinity of *Farm X* was chosen, and after completing the contract formalities, the SAWS provided the data. However, after close inspection of the data, it was discovered that the main area station incurred technical difficulties during 2018 and 2019. Regrettably, no data were available during this time from the specific station. The next nearest station was then chosen as an alternative data source. The distance between the weather stations are approximately 100km. For the purpose of this research, the assumption was made that the alternative station data closely approximates that of the main station data.

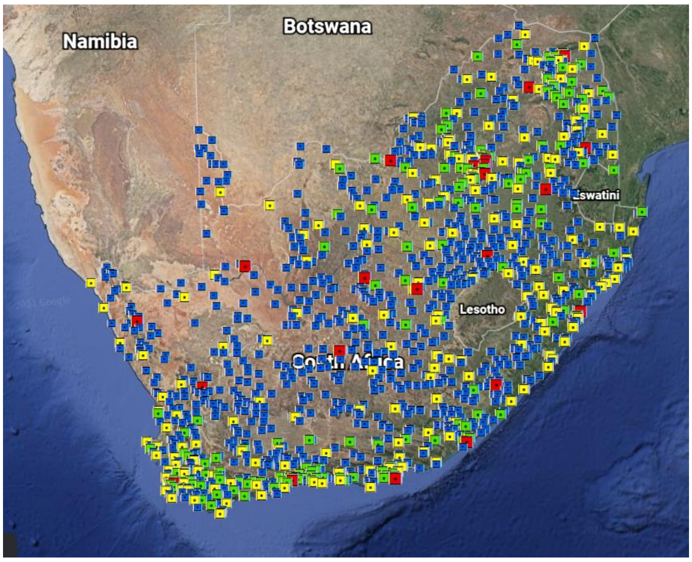


Figure 3.1: KMZ file displaying the available weather stations in South Africa

3.4 Winter wheat conditions in South Africa

The production guidelines for wheat compiled by the Department of Agriculture, Forestry and Fisheries (DAFF) of South Africa were consulted to better understand the crop conditions of *Farm X*. It is important to note that the choice of the cultivar can significantly impact the yield and be affected by factors such as soil type and geographical location (DAFF, 2010).

3.4.1 Planting and harvesting timeframe

Winter wheat is planted from mid-April to mid-June and is usually harvested from August to November. It can only be harvested when the grain moisture is about 16% and fully ripened. The planting date is important since early planting can stimulate excessive vegetative growth, later leading to lodging, whereas late planting can lead to insufficient vegetative growth and ultimately lower yield.

3.4.2 Temperature and rainfall requirements

The ideal climate for growing wheat is a cool temperature with plenty of rain followed by a dry period for harvesting. Most parts of South Africa have a summer rainfall climate, and wheat grown in these areas depends on sufficient rain in the previous season to ensure adequate residual soil moisture (this is essential when incorporating weather data in the wheat data analysis). Winter wheat requires temperatures between 5°C to 25°C and an annual rainfall of about 600mm per annum. Frost and hail can result in serious damage and low yield. (DAFF, 2010).

3.4.3 Soil requirements

Well-drained fertile loam to sandy loam with a pH of 6,0 to 7,5 is preferred for wheat production. Wheat is adversely affected by acidic soil, particularly during the early development stages and can cause the soil nutrients to be fixed or unavailable. Cu, Mn, Zn, and Boron (B) are essential for wheat's normal development and growth (DAFF, 2010).

3.4.4 Irrigation and fertilisation

It is important to continue irrigation until the plant is almost discoloured. Wheat requires sufficient soil moisture during planting and germination, lowered moisture during flowering and increased moisture during pod filling. Irrigation should be ceased during ripening, and wet weather during harvesting can contribute to diseases and quality deterioration of the grains. Proper irrigation scheduling can also minimise lodging and disease occurrence (DAFF, 2010).

3.4.5 Diseases and pests

Several weeds, diseases and pests can affect wheat production. Some weeds can limit yields by a staggering 20% annually. Cultivars, weather, irrigation, and soil conditions can play a major role in the prevalence of diseases and pests. Crop rotation and herbicides can be used to manage potential problems (DAFF, 2010).

3.5 Data research

It is difficult to define “normal” or “ideal” growing conditions when several variables can impact crop production. Instead, it is helpful to consider the entire life cycle of the crop at specific instances in time.

Company A’s platform provides more than 85 soil features available to the client to choose from. The database pertaining to *Farm X* does not have sufficient NDVI data but has Fraction of Vegetation Cover (FCover), LAI and chlorophyll data from 2016 to 2020. *Company A* advised using the chlorophyll time-series data and six main soil features to start the analysis. The six main soil features that were identified are:

- Soil type
- Depth of potential root development
- Plant available water content (PAWC) effective depth
- Magnesium percentage (Mg %)

- Sodium-Potassium (Na:K)
- Phosphorus (PBray 1)

After consulting the DAFF guidelines, it was decided to add Cu, K, Zn, Mn, S and pH to the data analysis to investigate if they could add further value to the analysis.

3.6 Summary

In this chapter, *Company A* and *Farm X* were discussed to understand better the purpose and application of the concept demonstrator tool to be developed. Field research was done to show how meteorological data can be obtained for research purposes, and the DAFF guidelines were studied to gain more knowledge regarding winter wheat growing conditions. The data and features identified in this chapter were used as a starting point for the following chapter.

Chapter 4
Data analysis

Chapter 2 and Chapter 3 were used to gain the necessary knowledge regarding PA and the data provided for this research study. Several research papers explore factors such as NDVI, LAI and yield data for crop yield estimation. Since there is no yield data available for *Farm X*, this chapter focuses on the chlorophyll data available for the initial analysis. The various data analysis methodologies and processes were discussed in Chapter 2, Section 2.4.2.

The CRISP-DM methodology is still the most popular method and was deemed suitable for the purpose of this study. It was used as a guideline to conduct the data analysis and forms the basis of the concept demonstrator discussed in Chapters 5 and 6. The CRISP-DM methodology was adapted slightly to fit the requirements of this study, as illustrated in Figure 4.1. Each section is discussed separately in this chapter. The process starts with a business understanding and data collection, followed by data understanding, data preparation, and modelling. The final stage (deployment) does not fall within the scope of this research study.

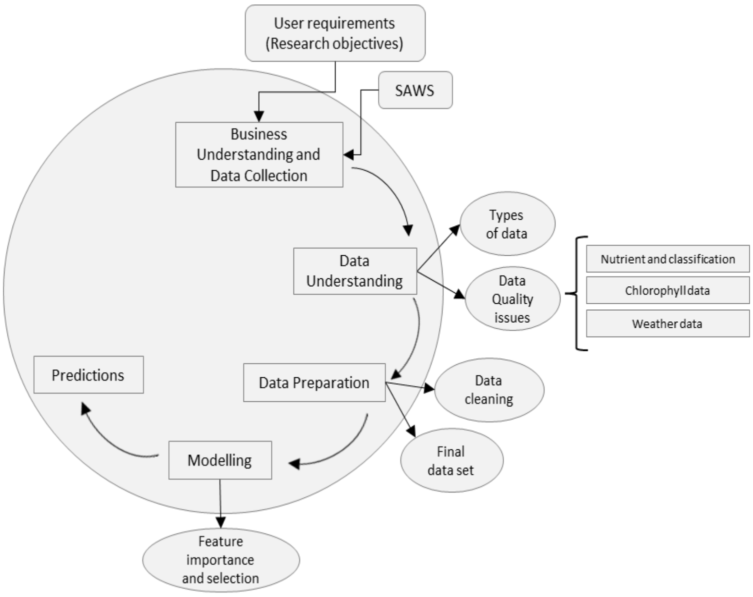


Figure 4.1: An illustration of the CRISP-DM method used in this study

4.1 Business understanding and data collection

Business understanding involves understanding the objectives and requirements from a business perspective. The research objectives discussed in Chapter 1 (see Section 1.3) and the information gained from the literature study and field research served as the first part in understanding the problem. The next step was the data collection process.

Company A supplied a folder with TIFF files containing chlorophyll data from *Farm X,* collected from 2016 to 2020, as shown in Figure 4.2 (see overleaf). The shapes indicated in this figure map onto specific geographical areas on *Farm X* where crops are planted.

To process the data, all the files were imported into the QGIS[2] software to view the chlorophyll data shown in Figure 4.3. QGIS was chosen from the list of software mentioned in Chapter 2, Section 2.2.3, as it is a popular open-source software that is easy to learn for basic applications. A single file containing the nutrient and soil classification data was also imported and superimposed on the TIFF files shown in Figure 4.4. Figure 4.5 displays a point shape file with 296 points (later referred to as 'instances') created in QGIS to extract and present all the available data per point on the crop circle in one image.

[2] QGIS: A free and open-source cross-platform desktop GIS application that supports viewing, editing, and analysis of geospatial data (www.qgis.org).

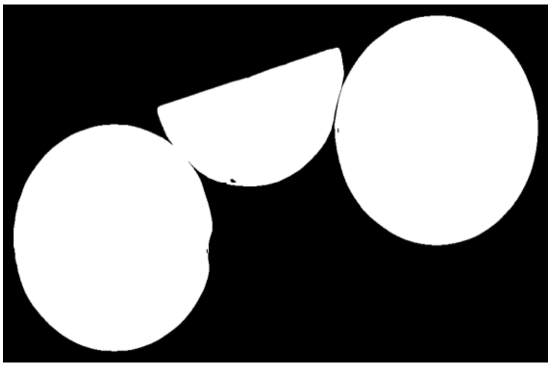


Figure 4.2: Example of a TIFF file

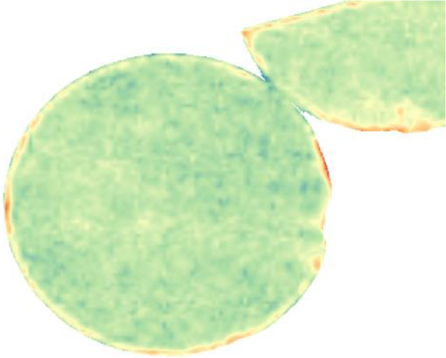


Figure 4.3: TIFF file imported into QGIS

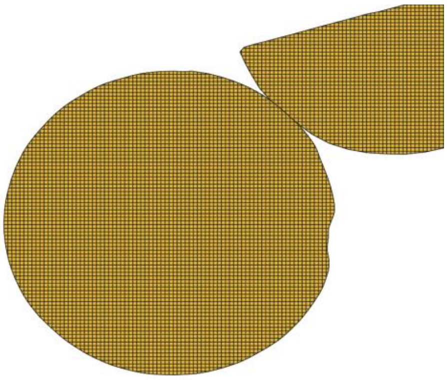


Figure 4.4: Soil classification .dbf file

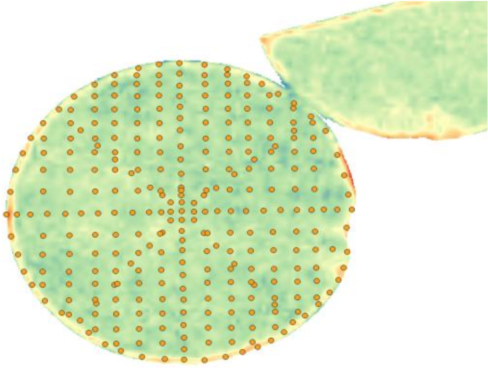


Figure 4.5: Point shape file superimposed onto a TIFF file

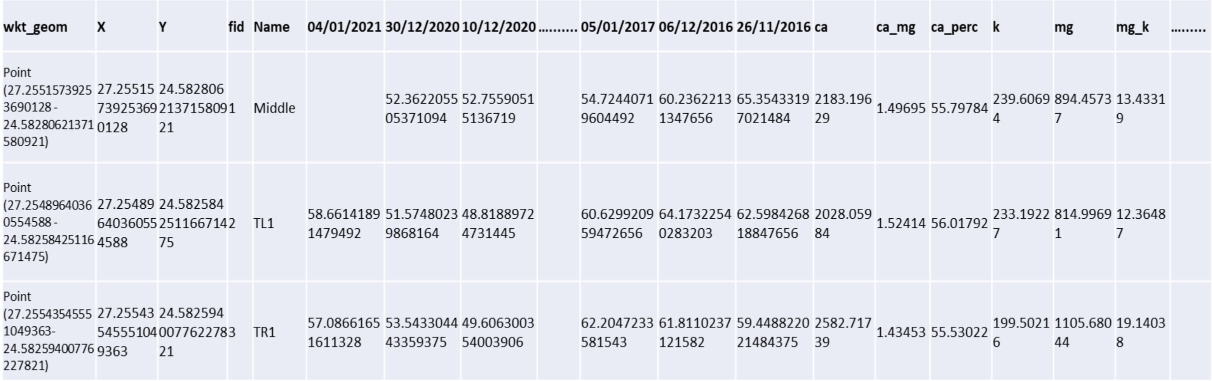
4.2 Data understanding and data exploration

Data exploration is used to understand the characteristics of the data and determine if any data quality issues might affect the model.

The data extracted from QGIS was copied into a Microsoft Excel workbook, which was used to examine the data. An extract of the data can be seen in Table 4.1 (see overleaf) and consists of 296 GPS point coordinates relating to nutrient and soil classification feature values. A total of 26 features were extracted. These features were discussed in Chapter 3, Section 3.5.

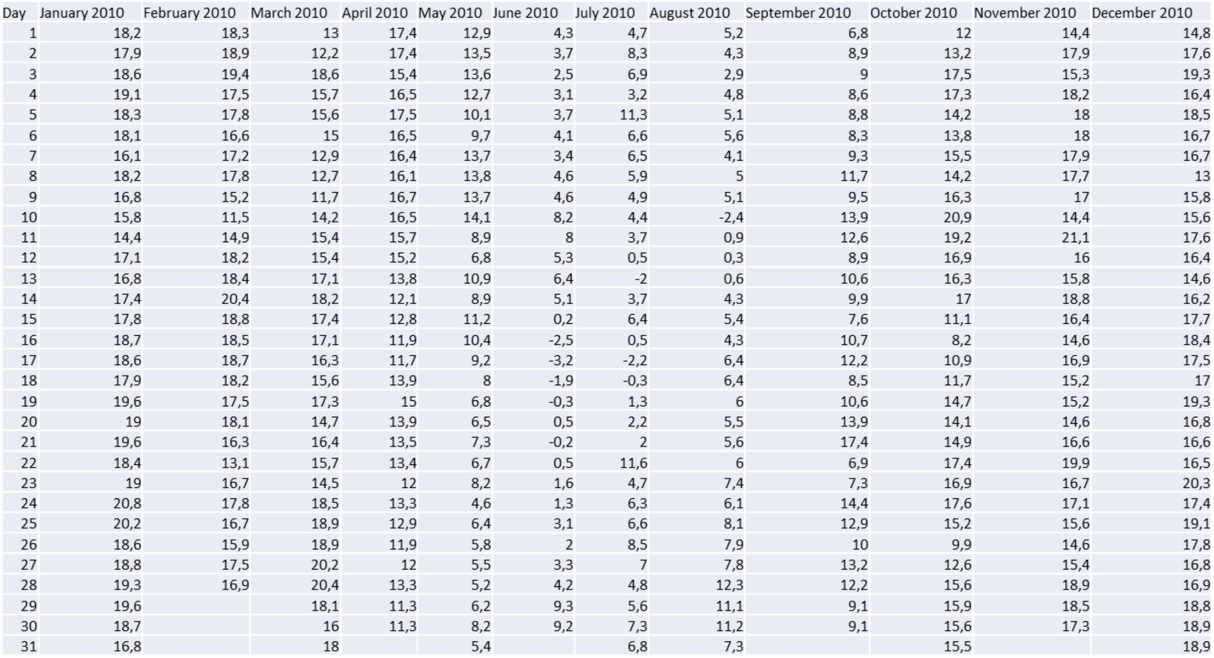
The second data set consists of the data supplied by the SAWS in the form of a 1997 - 2003 Excel-format workbook. The sheet data consisted of poorly structured vertical tables, which had to be transformed into a more user-friendly horizontal time-series data table. Table 4.2 (overleaf) illustrates a sample of data converted to a table format.

Table 4.1: Extract of raw data from QGIS



wkt_geom	X	Y	fid	Name	04/01/2021	30/12/2020	10/12/2020	....	05/01/2017	06/12/2016	26/11/2016	ca	ca_mg	ca_perc	k	mg	mg_k	....
Point (27.25515739253690128 -24.58280621371580921)	27.25515739253690128	24.582806213715809121	1	Middle		52.362205505371094	52.75590515136719		54.72440719604492	60.23622131347656	65.35433197021484	2183.19629	1.49695	55.79784	239.60694	894.45737	13.43319	
Point (27.25489640360554588 -24.58258425116671475)	27.25489640360554588	24.58258425116671475	2	TL1	58.66141891479492	51.57480239868164	48.81889724731445		60.629920959472656	64.17322540283203	62.598426818847656	2028.05984	1.52414	56.01792	233.19227	814.99691	12.36487	
Point (27.25543545551049363-24.58259400776227821)	27.25543545551049363	24.582594007762278321	3	TR1	57.08661651611328	53.543304443359375	49.606300354003906		62.2047233581543	61.8110237121582	59.448822021484375	2582.71739	1.43453	55.53022	199.50216	1105.68044	19.14038	

Table 4.2: Minimum daily temperature for the year 2010 in ˚C

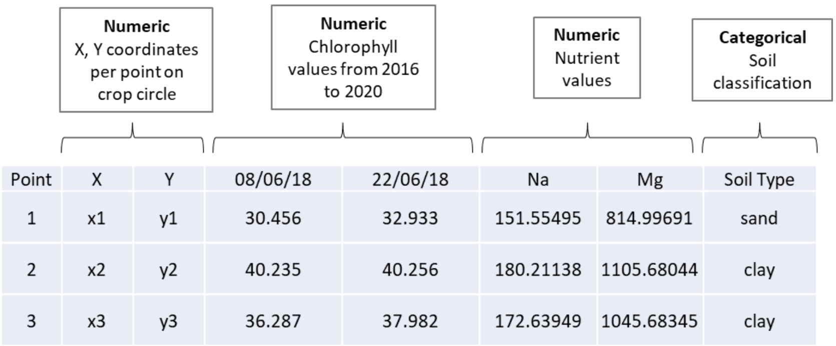


Day	January 2010	February 2010	March 2010	April 2010	May 2010	June 2010	July 2010	August 2010	September 2010	October 2010	November 2010	December 2010
1	18,2	18,3	13	17,4	12,9	4,3	4,7	5,2	6,8	12	14,4	14,8
2	17,9	18,9	12,2	17,4	13,5	3,7	8,3	4,3	8,9	13,2	17,9	17,6
3	18,6	19,4	18,6	15,4	13,6	2,5	6,9	2,9	9	17,5	15,3	19,3
4	19,1	17,5	15,7	16,5	12,7	3,1	3,2	4,8	8,6	17,3	18,2	16,4
5	18,3	17,8	15,6	17,5	10,1	3,7	11,3	5,1	8,8	14,2	18	18,5
6	18,1	16,6	15	16,5	9,7	4,1	6,6	5,6	8,3	13,8	18	16,7
7	16,1	17,2	12,9	16,4	13,7	3,4	6,5	4,1	9,3	15,5	17,9	16,7
8	18,2	17,8	12,7	16,1	13,8	4,6	5,9	5	11,7	14,2	17,7	13
9	16,8	15,2	11,7	16,7	13,7	4,6	4,9	5,1	9,5	16,3	17	15,8
10	15,8	11,5	14,2	16,5	14,1	8,2	4,4	-2,4	13,9	20,9	14,4	15,6
11	14,4	14,9	15,4	15,7	8,9	8	3,7	0,9	12,6	19,2	21,1	17,6
12	17,1	18,2	15,4	15,2	6,8	5,3	0,5	0,3	8,9	16,9	16	16,4
13	16,8	18,4	17,1	13,8	10,9	6,4	-2	0,6	10,6	16,3	15,8	14,6
14	17,4	20,4	18,2	12,1	8,9	5,1	3,7	4,3	9,9	17	18,8	16,2
15	17,8	18,8	17,4	12,8	11,2	0,2	6,4	5,4	7,6	11,1	16,4	17,7
16	18,7	18,5	17,1	11,9	10,4	-2,5	0,5	4,3	10,7	8,2	14,6	18,4
17	18,6	18,7	16,3	11,7	9,2	-3,2	-2,2	6,4	12,2	10,9	16,9	17,5
18	17,9	18,2	15,6	13,9	8	-1,9	-0,3	6,4	8,5	11,7	15,2	17
19	19,6	17,5	17,3	15	6,8	-0,3	1,3	6	10,6	14,7	15,2	19,3
20	19	18,1	14,7	13,9	6,5	0,5	2,2	5,5	13,9	14,1	14,6	16,8
21	19,6	16,3	16,4	13,5	7,3	-0,2	2	5,6	17,4	14,9	16,6	16,6
22	18,4	13,1	15,7	13,4	6,7	0,5	11,6	6	6,9	17,4	19,9	16,5
23	19	16,7	14,5	12	8,2	1,6	4,7	7,4	7,3	16,9	16,7	20,3
24	20,8	17,8	18,5	13,3	4,6	1,3	6,3	6,1	14,4	17,6	17,1	17,4
25	20,2	16,7	18,9	12,9	6,4	3,1	6,6	8,1	12,9	15,2	15,6	19,1
26	18,6	15,9	18,9	11,9	5,8	2	8,5	7,9	10	9,9	14,6	17,8
27	18,8	17,5	20,2	12	5,5	3,3	7	7,8	13,2	12,6	15,4	16,8
28	19,3	16,9	20,4	13,3	5,2	4,2	4,8	12,3	12,2	15,6	18,9	16,9
29	19,6		18,1	11,3	6,2	9,3	5,6	11,1	9,1	15,9	18,5	18,8
30	18,7		16	11,3	8,2	9,2	7,3	11,2	9,1	15,6	17,3	18,9
31	16,8		18		5,4		6,8	7,3		15,5		18,9

4.2.1 Types of data

Knowing the data types is crucial to understanding the data and using the correct methods to approach data quality issues and process the data correctly. The nutrient and soil classification data did not change during the analysed period and can be considered “static” or stationary data. Most of the stationary data is numeric in nature, with five categorical feature columns. The chlorophyll and meteorological data is numeric and non-stationary data and change over time. Table 4.3 (overleaf) is populated with artificial data to illustrate the various data types of the original data table.

Table 4.3: Conceptual table illustrating data types



Numeric
X, Y coordinates per point on crop circle

Numeric
Chlorophyll values from 2016 to 2020

Numeric
Nutrient values

Categorical
Soil classification

Point	X	Y	08/06/18	22/06/18	Na	Mg	Soil Type
1	x1	y1	30.456	32.933	151.55495	814.99691	sand
2	x2	y2	40.235	40.256	180.21138	1105.68044	clay
3	x3	y3	36.287	37.982	172.63949	1045.68345	clay

4.2.2 Data quality issues

After the initial inspection of the data in QGIS, it was discovered that several of the TIFF files provided did not contain complete data. One of the reasons is that the Airbus Verde satellite is highly sensitive towards cloud cover and only takes photographs when it passes a requested area. The photographs do not occur at equally spaced time intervals and are randomly dispersed. Some months might, for example, have six photographs per month for a given area and other months might have two. Given the infrequency of these photographs, more TIFF files do not necessarily equate to good quality data, and the TIFF files are not always useable. Figure 4.6 below shows a photograph taken on a clear day with complete data. In contrast, Figure 4.7 (see overleaf) shows an example of a photo with cloud cover and very little useable data.

Data quality issues can be addressed by compiling data quality reports to analyse continuous and categorical data. The three main data groups, viz. nutrient and soil classification, chlorophyll and meteorological data, are analysed separately.

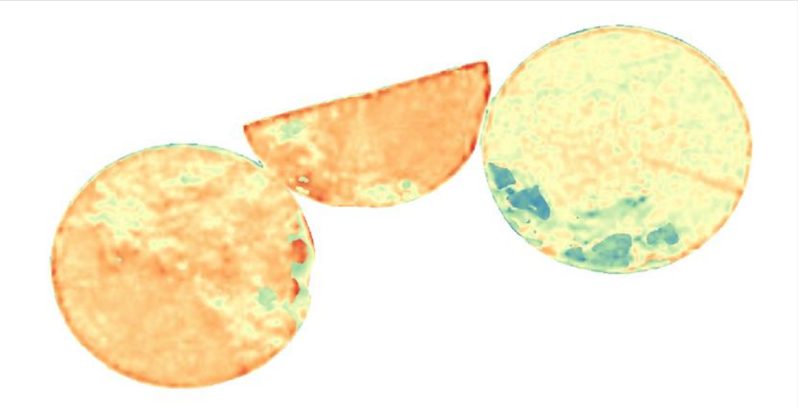


Figure 4.6: TIFF file displayed in QGIS with no cloud cover

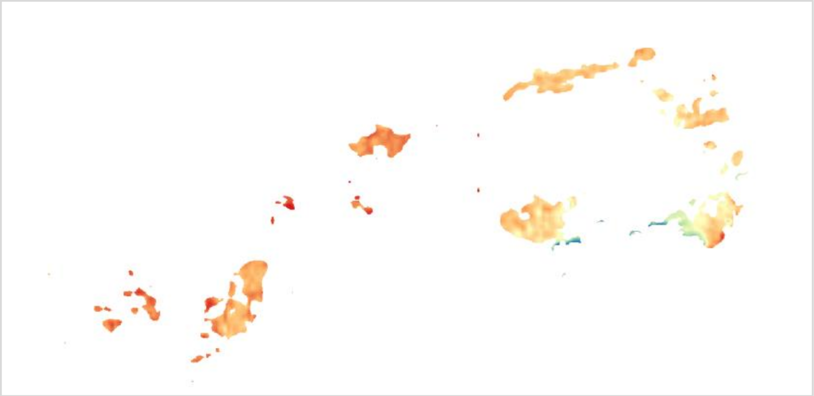


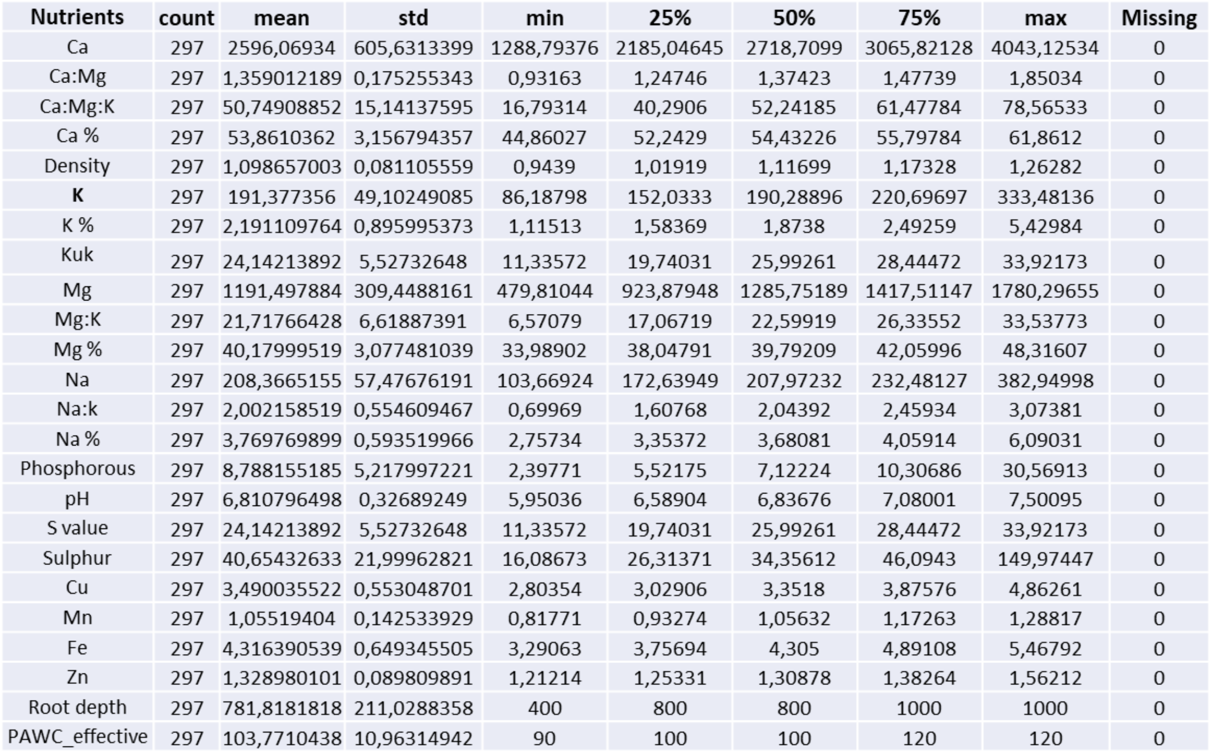
Figure 4.7: TIFF file in QGIS with substantial cloud cover

4.2.2.1 Data quality – Nutrient and soil classification data

The nutrient and soil classification data quality report is shown in Table 4.4 below. Note that it does not contain any missing values, and the ranges of the features differ dramatically and should be considered during the data preparation phase. Feature scaling can be applied to eliminate potential bias to affect the outcome of the model.

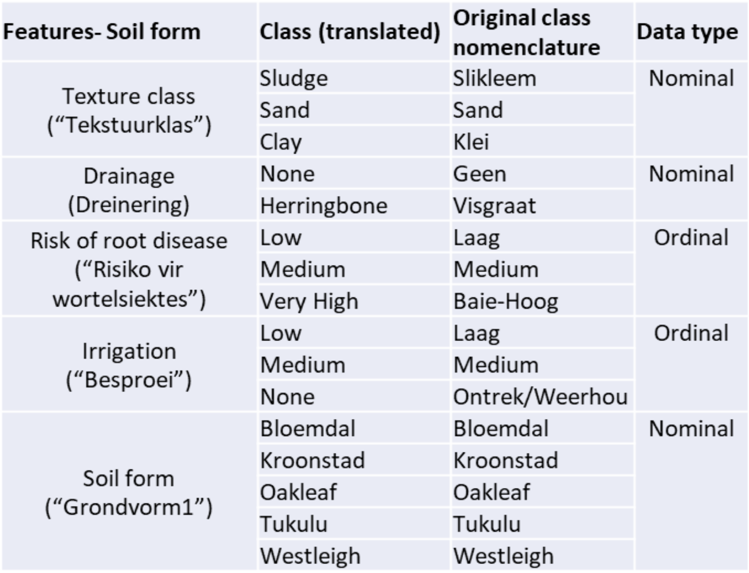
The five categorical features and their classes can be seen in Table 4.5 (see overleaf). The feature names have been translated to English to eliminate any confusion. Table 4.6 contains the data quality report for the categorical features and indicates no missing values. The mode, mode frequency, and mode percentage are also displayed to understand the prominent classes better.

Table 4.4: Data quality report for continuous features



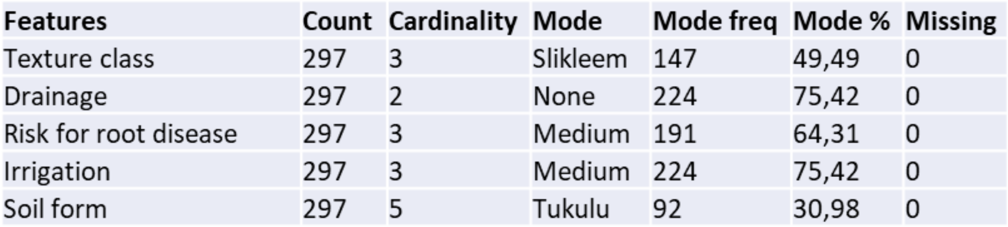
Nutrients	count	mean	std	min	25%	50%	75%	max	Missing
Ca	297	2596,06934	605,6313399	1288,79376	2185,04645	2718,7099	3065,82128	4043,12534	0
Ca:Mg	297	1,359012189	0,175255343	0,93163	1,24746	1,37423	1,47739	1,85034	0
Ca:Mg:K	297	50,74908852	15,14137595	16,79314	40,2906	52,24185	61,47784	78,56533	0
Ca %	297	53,8610362	3,156794357	44,86027	52,2429	54,43226	55,79784	61,8612	0
Density	297	1,098657003	0,081105559	0,9439	1,01919	1,11699	1,17328	1,26282	0
K	297	191,377356	49,10249085	86,18798	152,0333	190,28896	220,69697	333,48136	0
K %	297	2,191109764	0,895995373	1,11513	1,58369	1,8738	2,49259	5,42984	0
Kuk	297	24,14213892	5,52732648	11,33572	19,74031	25,99261	28,44472	33,92173	0
Mg	297	1191,497884	309,4488161	479,81044	923,87948	1285,75189	1417,51147	1780,29655	0
Mg:K	297	21,71766428	6,61887391	6,57079	17,06719	22,59919	26,33552	33,53773	0
Mg %	297	40,17999519	3,077481039	33,98902	38,04791	39,79209	42,05996	48,31607	0
Na	297	208,3665155	57,47676191	103,66924	172,63949	207,97232	232,48127	382,94998	0
Na:k	297	2,002158519	0,554609467	0,69969	1,60768	2,04392	2,45934	3,07381	0
Na %	297	3,769769899	0,593519966	2,75734	3,35372	3,68081	4,05914	6,09031	0
Phosphorous	297	8,788155185	5,217997221	2,39771	5,52175	7,12224	10,30686	30,56913	0
pH	297	6,810796498	0,32689249	5,95036	6,58904	6,83676	7,08001	7,50095	0
S value	297	24,14213892	5,52732648	11,33572	19,74031	25,99261	28,44472	33,92173	0
Sulphur	297	40,65432633	21,99962821	16,08673	26,31371	34,35612	46,0943	149,97447	0
Cu	297	3,490035522	0,553048701	2,80354	3,02906	3,3518	3,87576	4,86261	0
Mn	297	1,05519404	0,142533929	0,81771	0,93274	1,05632	1,17263	1,28817	0
Fe	297	4,316390539	0,649345505	3,29063	3,75694	4,305	4,89108	5,46792	0
Zn	297	1,328980101	0,089809891	1,21214	1,25331	1,30878	1,38264	1,56212	0
Root depth	297	781,8181818	211,0288358	400	800	800	1000	1000	0
PAWC_effective	297	103,7710438	10,96314942	90	100	100	120	120	0

Table 4.5: Categorical classes



Features- Soil form	Class (translated)	Original class nomenclature	Data type
Texture class (“Tekstuurklas”)	Sludge	Slikleem	Nominal
	Sand	Sand	
	Clay	Klei	
Drainage (Dreinering)	None	Geen	Nominal
	Herringbone	Visgraat	
Risk of root disease (“Risiko vir wortelsiektes”)	Low	Laag	Ordinal
	Medium	Medium	
	Very High	Baie-Hoog	
Irrigation (“Besproei”)	Low	Laag	Ordinal
	Medium	Medium	
	None	Ontrek/Weerhou	
Soil form (“Grondvorm1”)	Bloemdal	Bloemdal	Nominal
	Kroonstad	Kroonstad	
	Oakleaf	Oakleaf	
	Tukulu	Tukulu	
	Westleigh	Westleigh	

Table 4.6: Data quality report for categorical features



Features	Count	Cardinality	Mode	Mode freq	Mode %	Missing
Texture class	297	3	Slikleem	147	49,49	0
Drainage	297	2	None	224	75,42	0
Risk for root disease	297	3	Medium	191	64,31	0
Irrigation	297	3	Medium	224	75,42	0
Soil form	297	5	Tukulu	92	30,98	0

4.2.2.2 Data quality – Chlorophyll data

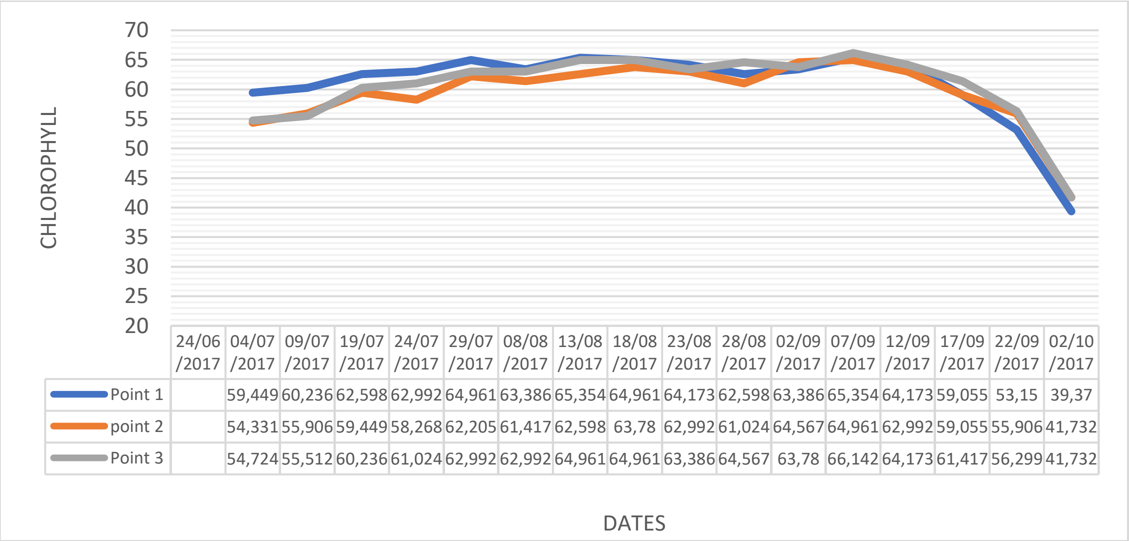
The chlorophyll time-series data contains data for the period 2016 to 2020. Out of the 120 TIFF files provided and 296 instances (crop circle points), there are 11 088 missing values and 24 849 usable values. The wheat data were divided into yearly seasons to investigate each year separately. An example of the 2018 wheat season chlorophyll data can be seen in Table 4.7. An empty TIFF file was supplied for “24/06/2018” and thus contained no data. This is most likely due to a significant amount of cloud cover during the imaging process. Similarly, only 5 data points on the entire crop circle (5 out of a total of 296 points) were available to supply data on “29/06/2018” and “07/10/2018”.

Table 4.7: Data quality report of 2018 chlorophyll values

Dates	count	mean	std	min	25%	50%	75%	max	Missing
24/06/2018	0	-	-	-	-	-	-	-	297
29/06/2018	5	46,29921188	1,099545663	44,4881897	46,0629921	46,85039139	46,85039139	47,24409485	292
04/07/2018	105	54,81439808	2,352252157	46,0629921	53,93700409	55,11810684	56,29921722	59,05511856	192
14/07/2018	294	54,25437773	1,835983185	47,63779449	53,54330444	54,52755737	55,5118103	61,02362442	3
19/07/2018	294	55,16631799	1,46942603	50	54,33070755	55,11810684	56,29921722	59,05511856	3
24/07/2018	295	58,33711615	1,814774853	50,78739929	57,87401962	58,66141891	59,44882202	62,20472336	2
29/07/2018	296	57,54682027	1,95303681	49,21260071	57,08661652	57,87401962	58,66141891	60,62992096	1
08/08/2018	297	64,02608716	2,04720195	55,90550995	63,38582611	64,56692505	65,35433197	66,92913055	0
13/08/2018	297	61,81632633	1,700767593	53,93700409	61,41732407	61,81102371	62,99212646	65,35433197	0
18/08/2018	297	62,39428376	2,214391294	51,96850204	62,20472336	62,99212646	63,77952576	66,14173126	0
23/08/2018	297	64,72865082	1,915912825	56,69291687	64,1732254	64,96063232	65,74803162	68,11023712	0
28/08/2018	296	62,45477747	2,318718424	52,36220551	61,81102371	62,99212646	63,77952576	66,14173126	1
02/09/2018	297	63,63901416	2,568263473	53,1496048	63,38582611	64,1732254	64,96063232	67,71652985	0
07/09/2018	284	63,9029047	1,626658268	57,48031616	62,99212646	63,77952576	64,56692505	68,89764404	13
12/09/2018	297	62,41019092	2,85667804	49,60630035	61,81102371	62,99212646	64,1732254	67,3228302	0
22/09/2018	280	56,34420797	1,722459776	48,4251976	55,5118103	56,29921722	57,48031616	62,59842682	17
27/09/2018	251	51,25325406	2,317900305	41,3385849	50,39369965	51,5748024	52,75590515	57,48031616	46
02/10/2018	233	37,90510604	2,413527507	30,70866203	36,22047424	37,79527664	39,37007904	45,27558899	64
07/10/2018	5	31,18110237	1,679584183	28,34645844	31,10236168	31,88976288	31,88976288	32,67716599	292

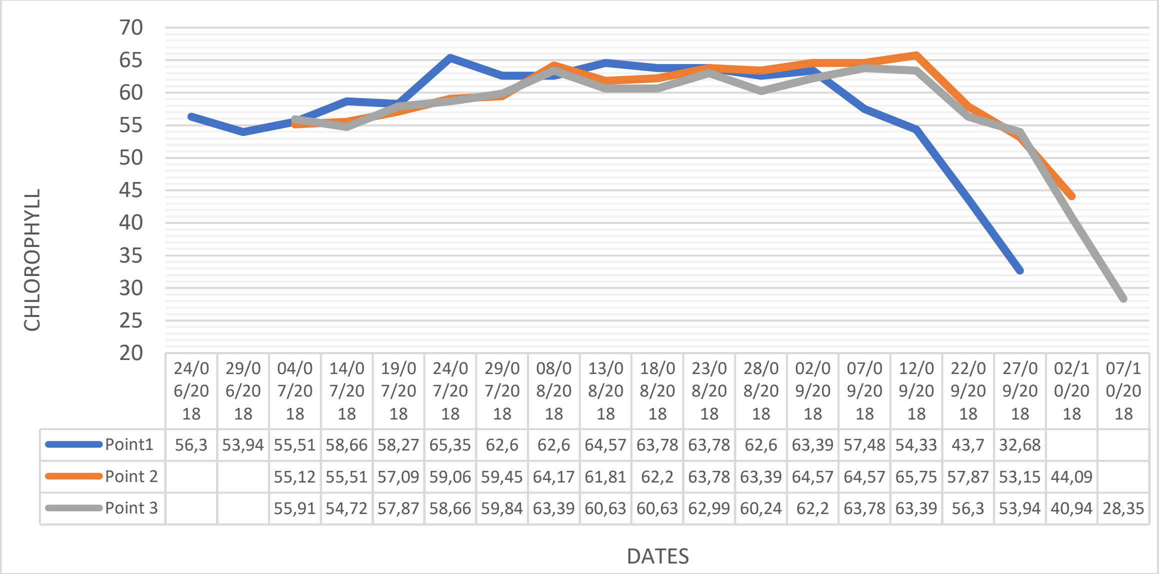
All the crop circle points were used to plot a graph to better understand the behaviour of the wheat chlorophyll values.

Figure 4.8 and Figure 4.9 represent the chlorophyll values for the wheat season during 2017 and 2018. For better visualisation, only three crop circle points were used as points on the line graphs below. All the points on the crop circle follow a similar pattern. The wheat is planted in May, starts to show chlorophyll values in June and July as the plant grows, peaks in September and starts to decline in October as the plant dies and the wheat is harvested.



	24/06 /2017	04/07 /2017	09/07 /2017	19/07 /2017	24/07 /2017	29/07 /2017	08/08 /2017	13/08 /2017	18/08 /2017	23/08 /2017	28/08 /2017	02/09 /2017	07/09 /2017	12/09 /2017	17/09 /2017	22/09 /2017	02/10 /2017
Point 1		59,449	60,236	62,598	62,992	64,961	63,386	65,354	64,961	64,173	62,598	63,386	65,354	64,173	59,055	53,15	39,37
point 2		54,331	55,906	59,449	58,268	62,205	61,417	62,598	63,78	62,992	61,024	64,567	64,961	62,992	59,055	55,906	41,732
Point 3		54,724	55,512	60,236	61,024	62,992	62,992	64,961	64,961	63,386	64,567	63,78	66,142	64,173	61,417	56,299	41,732

Figure 4.8: 2017 Wheat season – Chlorophyll



	24/06/2018	29/06/2018	04/07/2018	14/07/2018	19/07/2018	24/07/2018	29/07/2018	08/08/2018	13/08/2018	18/08/2018	23/08/2018	28/08/2018	02/09/2018	07/09/2018	12/09/2018	22/09/2018	27/09/2018	02/10/2018	07/10/2018
Point1	56,3	53,94	55,51	58,66	58,27	65,35	62,6	62,6	64,57	63,78	63,78	62,6	63,39	57,48	54,33	43,7	32,68		
Point 2			55,12	55,51	57,09	59,06	59,45	64,17	61,81	62,2	63,78	63,39	64,57	64,57	65,75	57,87	53,15	44,09	
Point 3			55,91	54,72	57,87	58,66	59,84	63,39	60,63	60,63	62,99	60,24	62,2	63,78	63,39	56,3	53,94	40,94	28,35

Figure 4.9: 2018 Wheat season – Chlorophyll

4.2.2.3 Data quality - Meteorological data

The data supplied by the SAWS contained data from January 1939 to May 2021 and included the following:

- The daily minimum temperature in degrees Celsius (˚C)
- The daily maximum temperature in degrees Celsius (˚C)
- The daily rainfall in millimetre (mm)
- The daily humidity (%)
- The daily windspeed in meter per second (m/s)
- The daily pressure in hectopascal (hPa)

It was decided to isolate the period 2010 to May 2021 for analysis purposes. The data have no missing values. The minimum and maximum temperature trends can be seen in Figure 4.10 below. The years follow a similar trend throughout, varying slightly for each month. Figure 4.11 shows a bar chart of the mean rainfall per month, colour-coded per year of a typical summer rainfall climate trend in South Africa. As discussed in Chapter 3, Section 3.4.2, wheat production heavily relies on the previous rain season. The rain season is approximately from October until April, followed by very little to no rain from May to September.

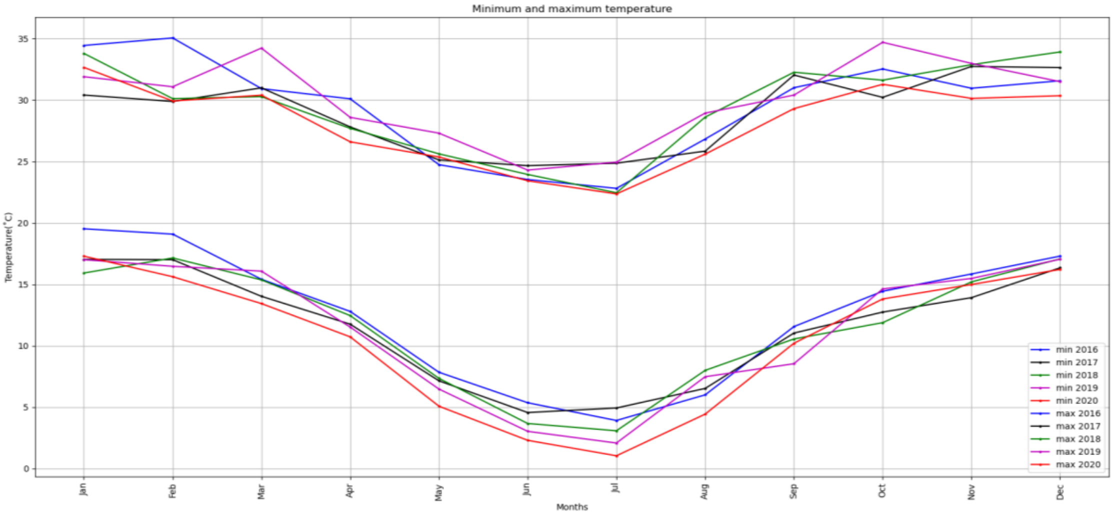


Figure 4.10: Minimum and maximum temperature from 2016 to 2020 for the approximate location of *Farm X*

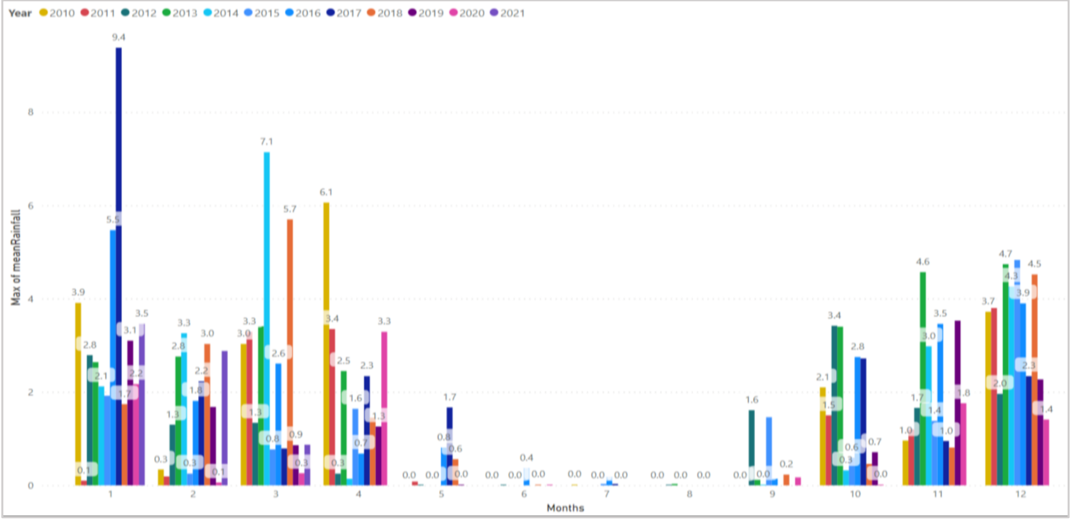


Figure 4.11: Mean rainfall per month grouped by years from 2010 to 2021

4.3 Data preparation

Data preparation includes data cleaning and constructing the final data tables to prepare the data for the model. The data quality issues for the sets were identified and were dealt with in the cleaning phase to ensure the model runs as efficiently as possible.

4.3.1 Data cleaning

Data cleaning was done to mitigate the data quality issues that were identified in the quality reports and initial data exploration graphs. The categorical data were converted into artificial variables to prepare the data for the ML algorithms. The missing values for the chlorophyll were dealt with in the following ways:

- Column dates with no values were removed.
- Missing values were replaced with date averages and not point (instance) averages as the

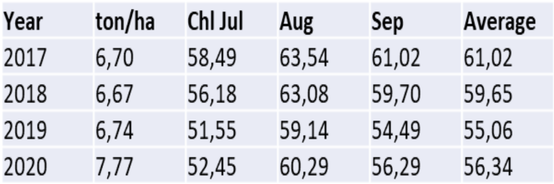
seasonal chlorophyll values vary between 30 and 65. The given crop circle points have a chlorophyll standard deviation between 1 to 2.5 on any specific date.

- Months with more than one observation per month were grouped together, and the mean per month was determined per point. By standardising the chlorophyll observations to one value per month simplified the analysis and graphing.

4.3.2 Constructing final datasets from the initial raw data

No GPS-specific yield was available to add to the final dataset. The chlorophyll data was compared to the average tonnage per hectare to explore the relationship between these values. The average yield per hectare for each year is shown in a bar chart in Figure 4.12. Note that there is no clear correlation between the average chlorophyll of 296 points per month and the overall crop circle yield presented in Table 4.8. It is important to note that more accurate yield along with other features should be considered to determine the correlation between chlorophyll. The total yield for the season was supplied, but simply dividing the total yield with the area of the crop circle will only provide an average yield value for each varying chlorophyll point and will not be accurate to include in the final data table. Results show that 2017 had the highest chlorophyll average but the second-lowest yield of 6.7 ton/ha. The year 2020 had a much lower average chlorophyll of 56.34 but had the highest yield of 7.77 ton/ha. Ideally, more accurate yield data, such as GPS-specific yield, would be more useful to test the predictions and confirm the correlation between chlorophyll and yield. However, considering the real-world scenario and utilising the available data, a detailed analysis can be performed to test the predictions of monthly average chlorophyll values instead.

Table 4.8: Ton/ha and average chlorophyll per year



Year	ton/ha	Chl Jul	Aug	Sep	Average
2017	6,70	58,49	63,54	61,02	61,02
2018	6,67	56,18	63,08	59,70	59,65
2019	6,74	51,55	59,14	54,49	55,06
2020	7,77	52,45	60,29	56,29	56,34

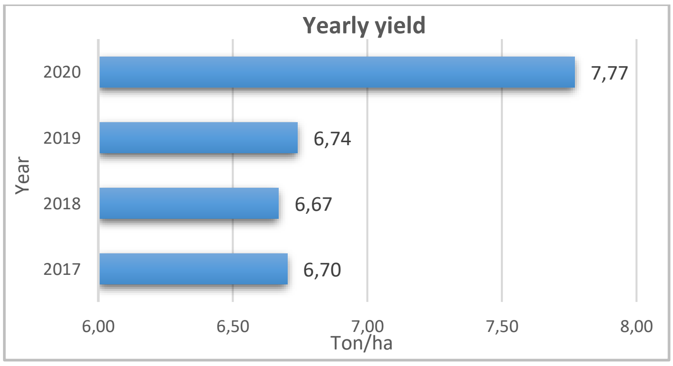
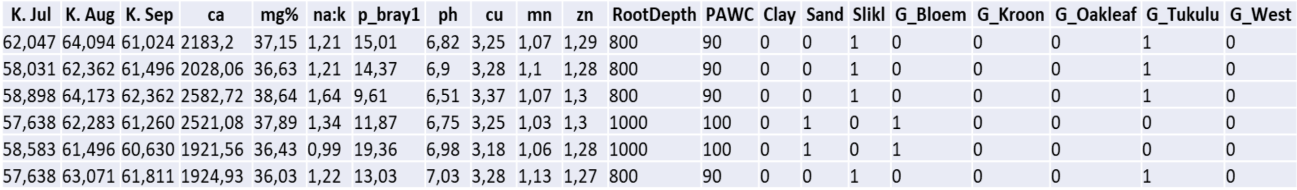


Figure 4.12: Bar chart of yearly average ton/ha

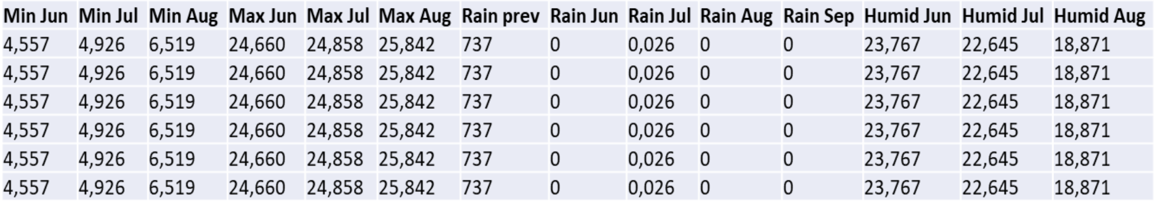
The decision was made to set the September chlorophyll values as the target feature values, as most points reached their peak chlorophyll in September. This decision was based on the research done in Chapter 2, Section 2.6.3, regarding the correlation between chlorophyll and yield values. Furthermore, it was assumed that if the "peak" chlorophyll value can be predicted in September, it can be used as a reasonable indication of the crop yield in October. The final table consists of useable chlorophyll data, nutrient and soil classification, and meteorological data. The meteorological data consist of the seasonal monthly means and the previous rain season total in millimetres from October to May. The previous season total is included because of wheat's dependency on the residual moisture in the soil from the previous rain season. A sample of the final table constructed from the raw data can be seen in Table 4.9 and Table 4.10. The aforesaid tables were split into part 1 and part 2 due to the width of the table.

Table 4.9: Sample of the final wheat table - part 1



K. Jul	K. Aug	K. Sep	ca	mg%	na:k	p_bray1	ph	cu	mn	zn	RootDepth	PAWC	Clay	Sand	Slikl	G_Bloem	G_Kroon	G_Oakleaf	G_Tukulu	G_West
62,047	64,094	61,024	2183,2	37,15	1,21	15,01	6,82	3,25	1,07	1,29	800	90	0	0	1	0	0	0	1	0
58,031	62,362	61,496	2028,06	36,63	1,21	14,37	6,9	3,28	1,1	1,28	800	90	0	0	1	0	0	0	1	0
58,898	64,173	62,362	2582,72	38,64	1,64	9,61	6,51	3,37	1,07	1,3	800	90	0	0	1	0	0	0	1	0
57,638	62,283	61,260	2521,08	37,89	1,34	11,87	6,75	3,25	1,03	1,3	1000	100	0	1	0	1	0	0	0	0
58,583	61,496	60,630	1921,56	36,43	0,99	19,36	6,98	3,18	1,06	1,28	1000	100	0	1	0	1	0	0	0	0
57,638	63,071	61,811	1924,93	36,03	1,22	13,03	7,03	3,28	1,13	1,27	800	90	0	0	1	0	0	0	1	0

Table 4.10: Sample of the final wheat table - part 2

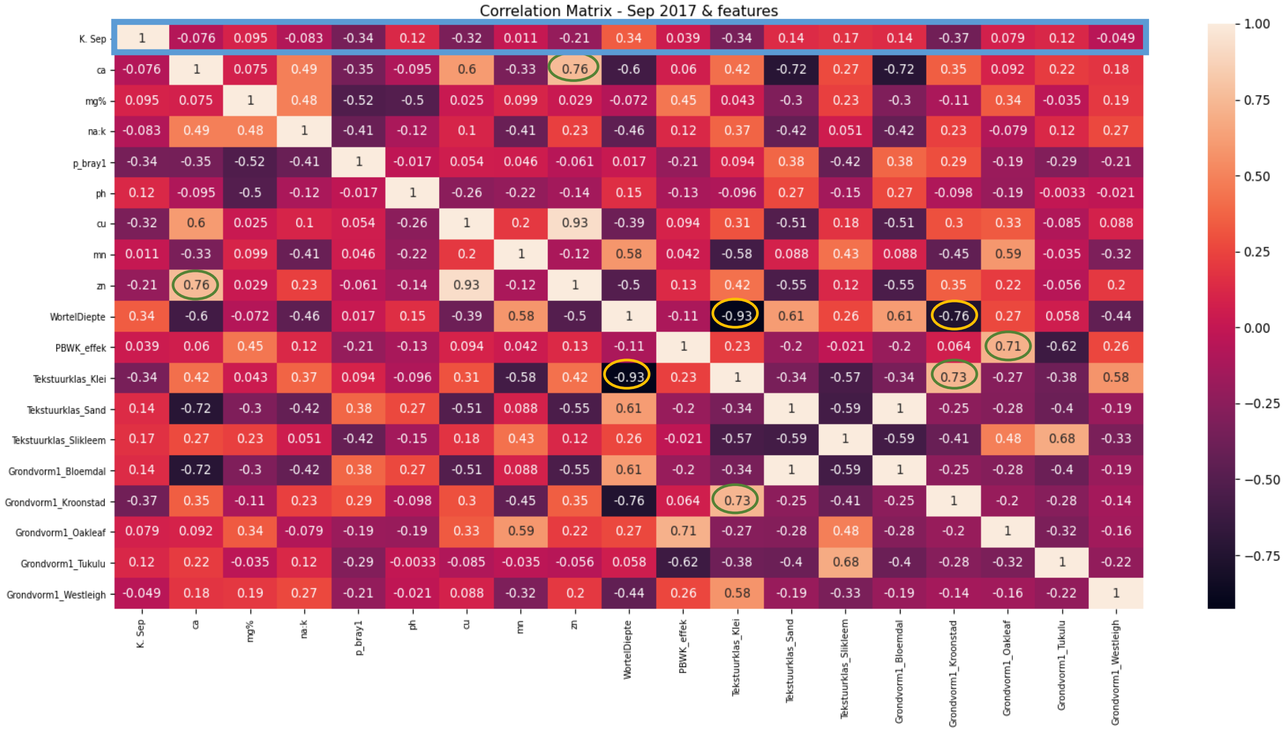


Min Jun	Min Jul	Min Aug	Max Jun	Max Jul	Max Aug	Rain prev	Rain Jun	Rain Jul	Rain Aug	Rain Sep	Humid Jun	Humid Jul	Humid Aug
4,557	4,926	6,519	24,660	24,858	25,842	737	0	0,026	0	0	23,767	22,645	18,871
4,557	4,926	6,519	24,660	24,858	25,842	737	0	0,026	0	0	23,767	22,645	18,871
4,557	4,926	6,519	24,660	24,858	25,842	737	0	0,026	0	0	23,767	22,645	18,871
4,557	4,926	6,519	24,660	24,858	25,842	737	0	0,026	0	0	23,767	22,645	18,871
4,557	4,926	6,519	24,660	24,858	25,842	737	0	0,026	0	0	23,767	22,645	18,871
4,557	4,926	6,519	24,660	24,858	25,842	737	0	0,026	0	0	23,767	22,645	18,871

4.4 Modelling

The modelling stage includes the steps to analyse the data by the use of data science methods. Firstly, correlation matrices were used to inspect the correlation between the chlorophyll values in September and all the nutrient and soil classification features. Figure 4.13 shows an example of a correlation matrix of the chlorophyll values of September 2017. It can be seen that there are no strong correlations between any features and the target feature, September chlorophyll. There are, however, strong positive correlations between features such as (i) Ca and Zn - with a correlation of 0.76 and (ii) Oakleaf soil-form and PAWC - with a strong positive correlation of 0.73. Root depth and "clay texture class" presented a strong negative correlation of -0.93. The correlation matrices were a reasonable starting point to get to know the data, but further analysis was required.

At this stage, it was unclear which features were best suited for accurate predictions. Hence, it was decided that the features could not be analysed linearly but should rather be considered in subsets. Features such as pH do not follow a linear relationship and cannot be analysed using linear regression to determine the relationship with the target values. Feature importance and selection can be used to analyse the features with various methods to determine the relationship and contribution of features.



	K. Sep	ca	mg%	na:k	p_bray1	ph	cu	mn	zn	WortelDiepte	PBWK_effek	Tekstuurklas_Klei	Tekstuurklas_Sand	Tekstuurklas_Slikleem	Grondvorm1_Bloemdal	Grondvorm1_Kroonstad	Grondvorm1_Oakleaf	Grondvorm1_Tukulu	Grondvorm1_Westleigh
K. Sep	1	-0.076	0.095	-0.083	-0.34	0.12	-0.32	0.011	-0.21	0.34	0.039	-0.34	0.14	0.17	0.14	-0.37	0.079	0.12	-0.049
ca	-0.076	1	0.075	0.49	-0.35	-0.095	0.6	-0.33	0.76	-0.6	0.06	0.42	-0.72	0.27	-0.72	0.35	0.092	0.22	0.18
mg%	0.095	0.075	1	0.48	-0.52	-0.5	0.025	0.099	0.029	-0.072	0.45	0.043	-0.3	0.23	-0.3	-0.11	0.34	-0.035	0.19
na:k	-0.083	0.49	0.48	1	-0.41	-0.12	0.1	-0.41	0.23	-0.46	0.12	0.37	-0.42	0.051	-0.42	0.23	-0.079	0.12	0.27
p_bray1	-0.34	-0.35	-0.52	-0.41	1	-0.017	0.054	0.046	-0.061	0.017	-0.21	0.094	0.38	-0.42	0.38	0.29	-0.19	-0.29	-0.21
ph	0.12	-0.095	-0.5	-0.12	-0.017	1	-0.26	-0.22	-0.14	0.15	-0.13	-0.096	0.27	-0.15	0.27	-0.098	-0.19	-0.0033	-0.021
cu	-0.32	0.6	0.025	0.1	0.054	-0.26	1	0.2	0.93	-0.39	0.094	0.31	-0.51	0.18	-0.51	0.3	0.33	-0.085	0.088
mn	0.011	-0.33	0.099	-0.41	0.046	-0.22	0.2	1	-0.12	0.58	0.042	-0.58	0.088	0.43	0.088	-0.45	0.59	-0.035	-0.32
zn	-0.21	0.76	0.029	0.23	-0.061	-0.14	0.93	-0.12	1	-0.5	0.13	0.42	-0.55	0.12	-0.55	0.35	0.22	-0.056	0.2
WortelDiepte	0.34	-0.6	-0.072	-0.46	0.017	0.15	-0.39	0.58	-0.5	1	-0.11	-0.93	0.61	0.26	0.61	-0.76	0.27	0.058	-0.44
PBWK_effek	0.039	0.06	0.45	0.12	-0.21	-0.13	0.094	0.042	0.13	-0.11	1	0.23	-0.2	-0.021	-0.2	0.064	0.71	-0.62	0.26
Tekstuurklas_Klei	-0.34	0.42	0.043	0.37	0.094	-0.096	0.31	-0.58	0.42	-0.93	0.23	1	-0.34	-0.57	-0.34	0.73	-0.27	-0.38	0.58
Tekstuurklas_Sand	0.14	-0.72	-0.3	-0.42	0.38	0.27	-0.51	0.088	-0.55	0.61	-0.2	-0.34	1	-0.59	1	-0.25	-0.28	-0.4	-0.19
Tekstuurklas_Slikleem	0.17	0.27	0.23	0.051	-0.42	-0.15	0.18	0.43	0.12	0.26	-0.021	-0.57	-0.59	1	-0.59	-0.41	0.48	0.68	-0.33
Grondvorm1_Bloemdal	0.14	-0.72	-0.3	-0.42	0.38	0.27	-0.51	0.088	-0.55	0.61	-0.2	-0.34	1	-0.59	1	-0.25	-0.28	-0.4	-0.19
Grondvorm1_Kroonstad	-0.37	0.35	-0.11	0.23	0.29	-0.098	0.3	-0.45	0.35	-0.76	0.064	0.73	-0.25	-0.41	-0.25	1	-0.2	-0.28	-0.14
Grondvorm1_Oakleaf	0.079	0.092	0.34	-0.079	-0.19	-0.19	0.33	0.59	0.22	0.27	0.71	-0.27	-0.28	0.48	-0.28	-0.2	1	-0.32	-0.16
Grondvorm1_Tukulu	0.12	0.22	-0.035	0.12	-0.29	-0.0033	-0.085	-0.035	-0.056	0.058	-0.62	-0.38	-0.4	0.68	-0.4	-0.28	-0.32	1	-0.22
Grondvorm1_Westleigh	-0.049	0.18	0.19	0.27	-0.21	-0.021	0.088	-0.32	0.2	-0.44	0.26	0.58	-0.19	-0.33	-0.19	-0.14	-0.16	-0.22	1

Figure 4.13: Correlation matrix for September chlorophyll 2017 and nutrient and soil features

4.4.1 Feature importance and selection

"Feature importance and selection" is the method of evaluating the importance of features and choosing a subset of the most relevant features that perform the best. Feature selection can be approached by using filter, wrapper or embedded methods. The selection depends on the type of data set and the required predictions.

Initially, a lazy regressor was run on the wheat table to analyse the performance of several algorithms on all of the features. The built-in lazy regressor performance metrics are shown in Figure 4.14 below and are adjusted-R^2, R^2 and Root-Mean Squared Error (RMSE), and the time taken by the algorithm to be completed. The equation for R^2 is shown in (4.1), with y_i being the actual y value, $\hat{y}_i$ the predicted y value and $\bar{y}$ the mean of all the y values. Adjusted R^2 is shown in (4.2) with N being the number of observations and K the number of independent variables in the model. Lastly RMSE can be seen in (4.3). Let y_t be the observed value and $\tilde{y}_t$ the predicted value, with T fitted points in the time series.

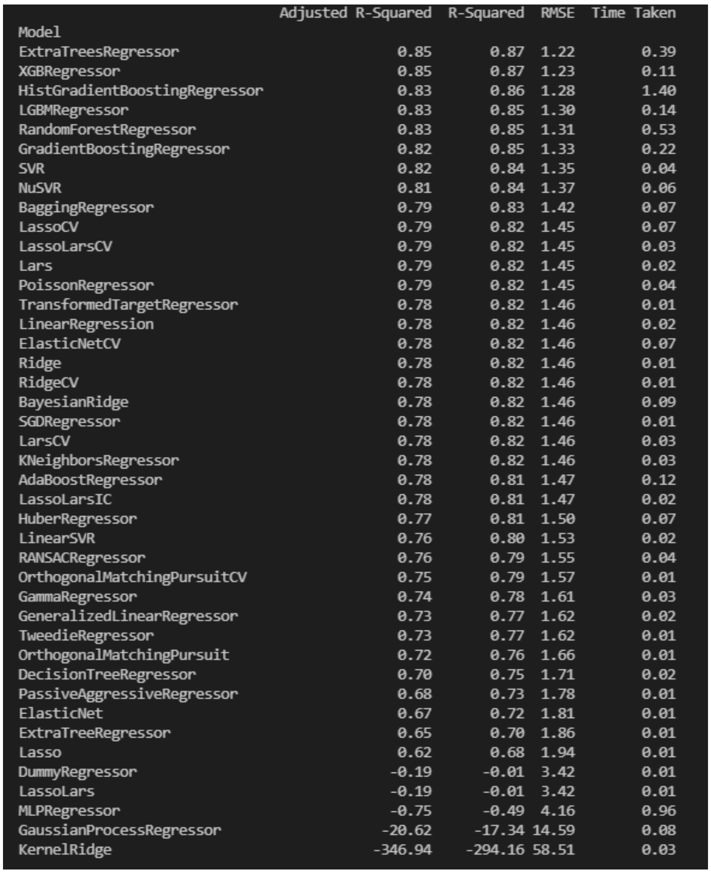
$$R^2 = 1 - \frac{sum\ squared\ regression\ (SSR)}{total\ sum\ of\ squares\ (SST)} \tag{4.1}$$

$$= 1 - \frac{\sum i(yi - \hat{yi})^2}{\sum i(yi - \bar{y})^2}$$

$$R^2adj = 1 - \frac{(1 - R^2)(N - 1)}{N - p - 1} \tag{4.2}$$

$$RMSE = \sqrt{\frac{\sum_{t=1}^{T}(yt - \tilde{y}t)^2}{T}} \tag{4.3}$$

The most common interpretation of a R^2 value indicates how well the regression model fits the observed data. Adjusted-R^2 also indicates how well the regression model fits the observed data, but adjusts for the number of terms in a model. RMSE is the standard deviation of the residuals or prediction errors and are a measure of how far the data points are from the regression line. The top-performing algorithms from the lazy regressor with their respective R^2 scores were (i) ETR with 0.87, (ii) XGBoost regressor with 0.87, (iii) HistGradientBoost regressor with 0.86, (iv) Light Gradient Boosting Machine (LGBM) regressor with 0.85 and the (v) Random Forest regressor with 0.85.

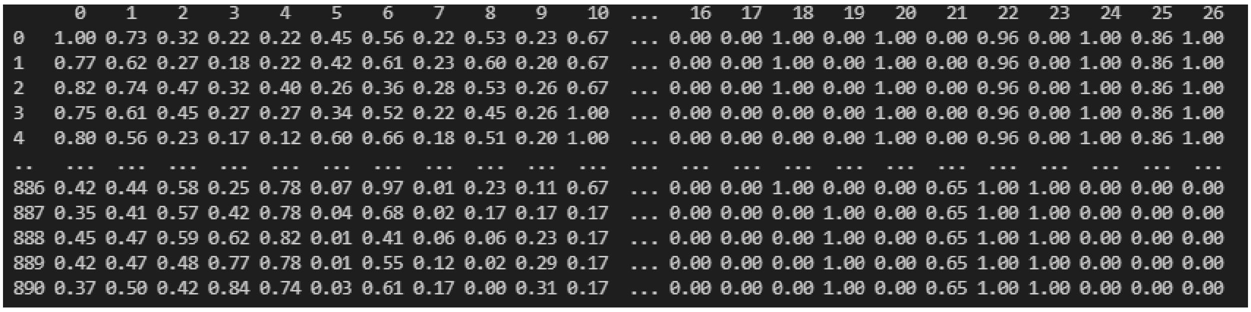


```
                               Adjusted R-Squared  R-Squared  RMSE  Time Taken
Model
ExtraTreesRegressor                          0.85       0.87  1.22        0.39
XGBRegressor                                 0.85       0.87  1.23        0.11
HistGradientBoostingRegressor                0.83       0.86  1.28        1.40
LGBMRegressor                                0.83       0.85  1.30        0.14
RandomForestRegressor                        0.83       0.85  1.31        0.53
GradientBoostingRegressor                    0.82       0.85  1.33        0.22
SVR                                          0.82       0.84  1.35        0.04
NuSVR                                        0.81       0.84  1.37        0.06
BaggingRegressor                             0.79       0.83  1.42        0.07
LassoCV                                      0.79       0.82  1.45        0.07
LassoLarsCV                                  0.79       0.82  1.45        0.03
Lars                                         0.79       0.82  1.45        0.02
PoissonRegressor                             0.79       0.82  1.45        0.04
TransformedTargetRegressor                   0.78       0.82  1.46        0.01
LinearRegression                             0.78       0.82  1.46        0.02
ElasticNetCV                                 0.78       0.82  1.46        0.07
Ridge                                        0.78       0.82  1.46        0.01
RidgeCV                                      0.78       0.82  1.46        0.01
BayesianRidge                                0.78       0.82  1.46        0.09
SGDRegressor                                 0.78       0.82  1.46        0.01
LarsCV                                       0.78       0.82  1.46        0.03
KNeighborsRegressor                          0.78       0.82  1.46        0.03
AdaBoostRegressor                            0.78       0.81  1.47        0.12
LassoLarsIC                                  0.78       0.81  1.47        0.02
HuberRegressor                               0.77       0.81  1.50        0.07
LinearSVR                                    0.76       0.80  1.53        0.02
RANSACRegressor                              0.76       0.79  1.55        0.04
OrthogonalMatchingPursuitCV                  0.75       0.79  1.57        0.01
GammaRegressor                               0.74       0.78  1.61        0.03
GeneralizedLinearRegressor                   0.73       0.77  1.62        0.02
TweedieRegressor                             0.73       0.77  1.62        0.01
OrthogonalMatchingPursuit                    0.72       0.76  1.66        0.01
DecisionTreeRegressor                        0.70       0.75  1.71        0.02
PassiveAggressiveRegressor                   0.68       0.73  1.78        0.01
ElasticNet                                   0.67       0.72  1.81        0.01
ExtraTreeRegressor                           0.65       0.70  1.86        0.01
Lasso                                        0.62       0.68  1.94        0.01
DummyRegressor                              -0.19      -0.01  3.42        0.01
LassoLars                                   -0.19      -0.01  3.42        0.01
MLPRegressor                                -0.75      -0.49  4.16        0.96
GaussianProcessRegressor                   -20.62     -17.34 14.59        0.08
KernelRidge                               -346.94    -294.16 58.51        0.03
```

Figure 4.14: Lazy regressor model performance on all features

Some of the features might be dependent on others. In such cases, simply analysing the relationship between each feature and the target feature (September chlorophyll) would not be sufficient. Thus, wrapper methods were used to detect the interaction between features and analyse the best performing feature subset. This means that the lowest-scoring features are not necessarily disregarded because they might increase the performance when used in a subset with other features. Wrapper search methods included forward selection, backward elimination, exhaustive selection and stepwise or bidirectional selection (Charfaoui, 2020; Verma, 2020).

While wrapper methods have many advantages, one of the main disadvantages is a high chance of over-fitting. The final wheat data set provides a good candidate for using scaler transforms as the variables have different minimum and maximum values and different data distributions and ranges. Some algorithms might not be as effective when the data is not scaled, as variables that are measured at different scales do not contribute equally to the model fitting & model learned function and might end up creating a bias as they might consider ranges such as pH (ranging between 1 and 14) to contribute a smaller weight than Ca that ranges within the thousands. Thus, scaling or normalising the data can help to deal with this problem. Standardisation is usually preferred when the data follow a Gaussian distribution, which was not the case for this data set (Brownlee, 2020b). Thus, the MinMaxScaler function in Python was used to normalise the data to values between 0 and 1, as seen in Figure 4.15. The MinMax equation can be seen in (4.4). Let *Xmin* be the minimum in the range and *Xmax* the maximum in the range.



```
       0    1    2    3    4    5    6    7    8    9    10  ...   16   17   18   19   20   21   22   23   24   25   26
0   1.00 0.73 0.32 0.22 0.22 0.45 0.56 0.22 0.53 0.23 0.67  ... 0.00 0.00 1.00 0.00 1.00 0.00 0.96 0.00 1.00 0.86 1.00
1   0.77 0.62 0.27 0.18 0.22 0.42 0.61 0.23 0.60 0.20 0.67  ... 0.00 0.00 1.00 0.00 1.00 0.00 0.96 0.00 1.00 0.86 1.00
2   0.82 0.74 0.47 0.32 0.40 0.26 0.36 0.28 0.53 0.26 0.67  ... 0.00 0.00 1.00 0.00 1.00 0.00 0.96 0.00 1.00 0.86 1.00
3   0.75 0.61 0.45 0.27 0.27 0.34 0.52 0.22 0.45 0.26 1.00  ... 0.00 0.00 0.00 0.00 1.00 0.00 0.96 0.00 1.00 0.86 1.00
4   0.80 0.56 0.23 0.17 0.12 0.60 0.66 0.18 0.51 0.20 1.00  ... 0.00 0.00 0.00 0.00 1.00 0.00 0.96 0.00 1.00 0.86 1.00
..   ...  ...  ...  ...  ...  ...  ...  ...  ...  ...  ...  ...  ...  ...  ...  ...  ...  ...  ...  ...  ...  ...  ...
886 0.42 0.44 0.58 0.25 0.78 0.07 0.97 0.01 0.23 0.11 0.67  ... 0.00 0.00 1.00 0.00 0.00 0.65 1.00 1.00 0.00 0.00 0.00
887 0.35 0.41 0.57 0.42 0.78 0.04 0.68 0.02 0.17 0.17 0.17  ... 0.00 0.00 0.00 1.00 0.00 0.65 1.00 1.00 0.00 0.00 0.00
888 0.45 0.47 0.59 0.62 0.82 0.01 0.41 0.06 0.06 0.23 0.17  ... 0.00 0.00 0.00 1.00 0.00 0.65 1.00 1.00 0.00 0.00 0.00
889 0.42 0.47 0.48 0.77 0.78 0.01 0.55 0.12 0.02 0.29 0.17  ... 0.00 0.00 0.00 1.00 0.00 0.65 1.00 1.00 0.00 0.00 0.00
890 0.37 0.50 0.42 0.84 0.74 0.03 0.61 0.17 0.00 0.31 0.17  ... 0.00 0.00 0.00 1.00 0.00 0.65 1.00 1.00 0.00 0.00 0.00
```

Figure 4.15: Wheat table normalised

$$Xscaled = \frac{X - Xmin}{Xmax - Xmin} \tag{4.4}$$

A sequential forward selector was used to determine the optimal number of features to include in the subset. Figure 4.16 indicates that the intersection occurs at nine features. Choosing more than nine features will not necessarily increase the model performance but require more computing power and increase the computing time. Thus, it is a point of diminishing return.

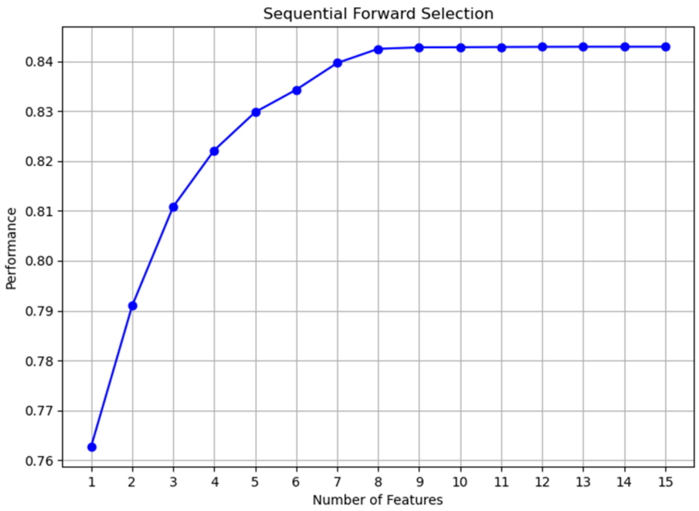


Figure 4.16: Graph for analysing the optimal number of features

The scoring argument specifies the evaluation criterion to be used. For regression problems, there is only an R^2 score in default implementation. Similarly, for classification, it can be accuracy, precision, recall, f1-score, etc. (Verma, 2020).

After choosing the optimal number of features, the four wrapper methods were run on the top-performing algorithms. The LGBM regressor was not used due to the similarity to the other chosen algorithms. The three most commonly used error metrics for evaluating performance of regression models are Mean Absolute Error (MAE), Mean Squared Error (MSE) and RMSE. MAE measures the average magnitude of the errors in a set of predictions without considering their direction and is less biased for higher values. The equation can be seen in (4.5), with *yi* being the ith expected value in the dataset, *ŷi* the ith predicted value and n the total number of data points. MSE tells you how close a regression line is to a set of points and is preferred to MAE when accounting for outliers. The equation is seen in (4.6), with *yi* is the ith expected value in the dataset, *ŷi* is the ith predicted value. The results are shown in Table 4.11. A smaller MAE and MSE indicate a better model, whereas a value close to one (1) for R^2 is desired. It is clear that the ETR outperforms the other algorithms in each method. Gradient boosting and decision tree-based algorithms are usually robust against scaling and normalisation problems. The algorithms were also tested with and without normalisation to compare the performance. Normalising the data did not improve the boosting regressor accuracy, but minor changes were observed with the Random Forest and ETRs because of the random nature of the algorithm. The sequential forward selection wrapper method with the Extra Trees algorithm was chosen. The R^2 value of 0.86 indicates that the variance of the independent variable explains 86% of the variance of the dependent variable being studied.

$$MAE = \frac{\sum_{i=1}^{n} |yi - \hat{yi}|}{n} \tag{4.5}$$

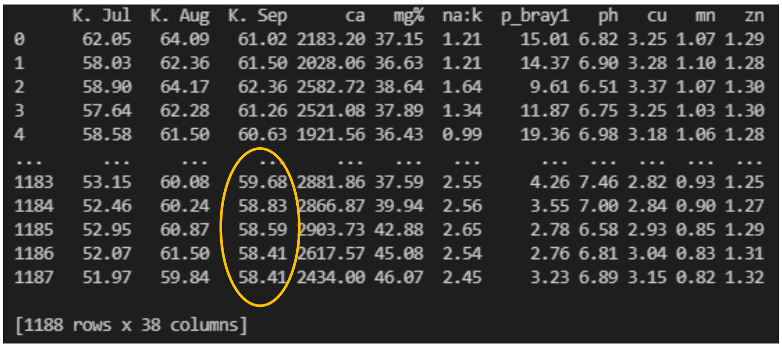
$$MSE = \frac{1}{n}\sum_{i=1}^{n}(yi - \hat{y}i)^2 \tag{4.6}$$

Table 4.11: Wrapper method and algorithm results table

Method	**MAE**	**Normalised**	**MSE**	**Normalised**	**r2**	**Normalised**
Sequantial forward selection:						
Randomforest Regressor	0,9364	0,9432	1,8441	1,7841	0,8410	0,8462
HistgradientBoost Regressor	0,9450	0,9450	1,8098	1,8098	0,8440	0,8440
XGB Regressor	0,9725	0,9725	1,9672	1,9672	0,8304	0,8304
Extra Trees Regressor	0,9128	**0,8832**	1,7415	**1,5897**	0,8499	**0,8629**
Sequantial backward selection:						
Randomforest Regressor	0,9442	0,9553	1,7900	1,8867	0,8457	0,8373
HistgradientBoost Regressor	0,9381	0,9381	1,7234	1,7234	0,8514	0,8514
XGB Regressor	0,9318	0,9318	1,7310	1,7310	0,8508	0,8508
Extra Trees Regressor	0,9007	0,9220	1,6463	1,7062	0,8581	0,8529
Sequantial float forward selection:						
Randomforest Regressor	0,9611	0,9366	1,9547	1,7467	0,8315	0,8494
HistgradientBoost Regressor	0,9450	0,9450	1,8098	1,8098	0,8440	0,8440
XGB Regressor	0,9371	0,9371	1,7559	1,7559	0,8486	0,8486
Extra Trees Regressor	0,8925	0,8949	1,6607	1,6368	0,8568	0,8589
Sequantial float backward selection:						
Randomforest Regressor	0,9318	0,9363	1,7830	1,7919	0,8463	0,8455
HistgradientBoost Regressor	0,9222	0,9222	1,7262	1,7262	0,8512	0,8512
XGB Regressor	0,9569	0,9569	1,8587	1,8587	0,8398	0,8398
Extra Trees Regressor	0,8912	0,8887	1,6390	1,6036	0,8587	0,8617

4.4.2 Predictions

After the best performing method and algorithm were chosen, the wheat data for the year 2020 were fed into the ETR algorithm. Only data from June, July and August were available, but the data for September 2020 are missing from the original TIFF files. This presented an interesting opportunity to ‘test’ the prediction model in an unconventional way. The previous data from 2017, 2018 and 2019 were used as the entire training set, and the new 2020 data were used as the “x_test” data to predict chlorophyll values for September in 2020 (y_predict). As referred to above, no data were available for September 2020 (y_test). Consequently, the accuracy of the model cannot be calculated. The predicted values were stored in a column along with the original 2020 data and were used in the visualisations to compare the predicted values to previous years’ September chlorophyll by using statistical methods such as the mean and standard deviation (see Figure 4.17).

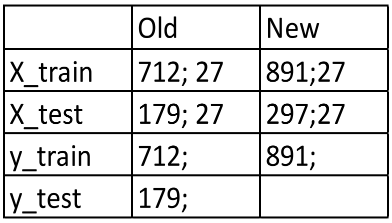


```
      K. Jul  K. Aug  K. Sep      ca    mg%  na:k  p_bray1   ph   cu   mn   zn
0      62.05   64.09   61.02 2183.20 37.15  1.21    15.01 6.82 3.25 1.07 1.29
1      58.03   62.36   61.50 2028.06 36.63  1.21    14.37 6.90 3.28 1.10 1.28
2      58.90   64.17   62.36 2582.72 38.64  1.64     9.61 6.51 3.37 1.07 1.30
3      57.64   62.28   61.26 2521.08 37.89  1.34    11.87 6.75 3.25 1.03 1.30
4      58.58   61.50   60.63 1921.56 36.43  0.99    19.36 6.98 3.18 1.06 1.28
...      ...     ...     ...     ...   ...   ...      ...  ...  ...  ...  ...
1183   53.15   60.08   59.68 2881.86 37.59  2.55     4.26 7.46 2.82 0.93 1.25
1184   52.46   60.24   58.83 2866.87 39.94  2.56     3.55 7.00 2.84 0.90 1.27
1185   52.95   60.87   58.59 2903.73 42.88  2.65     2.78 6.58 2.93 0.85 1.29
1186   52.07   61.50   58.41 2617.57 45.08  2.54     2.76 6.81 3.04 0.83 1.31
1187   51.97   59.84   58.41 2434.00 46.07  2.45     3.23 6.89 3.15 0.82 1.32

[1188 rows x 38 columns]
```

Figure 4.17: September 2020 predicted chlorophyll added to the original wheat table

The train and test X and Y sizes can be seen in Table 4.12 below. Combining data for years 2017, 2018 and 2019 provides a data set of almost 900 training points. This was split into a 712-point rows and 27 feature columns (X_train) data table. The test set, X_test, consisted of 179 rows and 27 feature columns (20% of the original data table values). The model predicted 179 chlorophyll values and compared them to the X_test data table. By adding 2020 data, the set increased by another 296 points. Still, due to the unavailability of September satellite data in 2020, no y_test could be used to compare the predicted values and measure the model's prediction accuracy.

Table 4.12: Train and test sets



	Old	New
X_train	712; 27	891;27
X_test	179; 27	297;27
y_train	712;	891;
y_test	179;	

4.5 Summary

The CRISP-DM method was discussed and used as a guideline to analyse the data. Four wrapper methods were used for feature importance and selection to determine the final feature subset in the prediction model (see Chapter 6). The feature subset was used as input to determine chlorophyll values for September. The ETR algorithms were given a data table of 2017, 2018 and 2019 split into 80% training and 20% testing data. The sequential forward selector chose the best feature subset, used the chosen features to make chlorophyll predictions, and achieved an accuracy of 86%. The model was also given a test set from 2020 to predict September chlorophyll values, which was then saved in a table used in Chapter 6 for visualisation and validation of the model.

Chapter 5
Concept demonstrator development

This chapter incorporates all the previous chapters' research that was used to develop the concept demonstrator for the use case. In addition, the level of technology adoption of *Farm X* and the data used in the visualisation are discussed below.

5.1 Level of precision agriculture adoption

Recall the discussion of the six levels of PA adoption in Chapter 2, Section 2.1 and note the relevant snippet shown in Figure 5.1 (see overleaf). *Farm X* lies between level 3 and level 4.

The chlorophyll data layers have been collected over four years (2016 - 2020). The imagery is affected by cloud cover and only provides approximately one to two useable images per month from June to October. However, in-season operational decision-making requires more frequent imagery to assist with near real-time decisions. Level 5 PA adoption typically implements imagery, weather- and soil moisture sensors, and pests- and disease monitoring systems. *Farm X* data include a nutrient and soil characteristic data layer, but it is not updated annually.

The sixth level of PA adoption should be considered along with the type of data analytic systems discussed in Chapter 2, Section 2.4.2. Descriptive and diagnostic analytics lies within levels 1 to level 3 of PA adoption. Predictive analytics can be used in level 4 adoption, and prescriptive analytics can be used in level 5 adoption for more automated decision support.

Company A uses the collected data to provide agronomic advice to the client but does not include a predictive- or prescriptive analytics service. The concept demonstrator thus aimed to utilise predictive analytics and demonstrated how the collected data layers could be used for in-season yearly comparisons to form the basis for a prescriptive decision support tool. This can be achieved in future work when more data have been collected, and more data layers (pest, disease and other vegetation indices) have been integrated into the system. Ideally, yield data would be preferred to test the relationship between chlorophyll predictions and yield data, but the study included a real-world example.

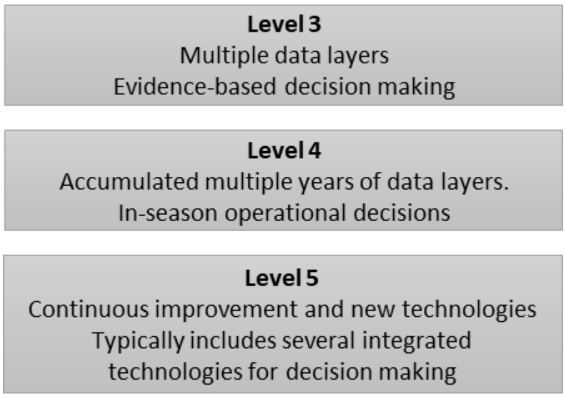

Level 3
Multiple data layers
Evidence-based decision making
Level 4
Accumulated multiple years of data layers.
In-season operational decisions
Level 5
Continuous improvement and new technologies
Typically includes several integrated
technologies for decision making

Figure 5.1: The six levels of PA adoption (levels 3 to 5)

5.2 Visualising agriculture data and concepts

As indicated in previous sections, a major goal of this research study is to improve the efficiency of analysing the layers, ideally by automating the process and yielding decision support intelligence.

After cleaning the farm data and constructing the final data table in Chapter 4, it was discovered that too many features complicate the decision-making process. Therefore, a feature importance and selection algorithm was used to select a subset of features for the ML prediction algorithm. It was found that nine features were an optimal number and that using more led to diminishing returns.

Decision-support systems were discussed in Chapter 2, Section 2.7. The concept demonstrator can include crop management zones, weather, and water management decision-support by utilising the available data. However, the current system used by *Farm X* does not compare yearly data. The farmer might thus surmise that the crops are performing well, but in reality, the current conditions of the crops might be performing below average compared to the crops a year ago on the same day.

Providing and analysing historical data can provide the farmer with valuable comparison data to calibrate the farm's performance. August 2017 and August 2018 were used as an example to illustrate this, as they had the most available data with four satellite image observations per month. Missing data points were replaced with the average of the total crop circle points of the same day and were not interpolated. Interpolating the data would not accurately represent the chlorophyll as it differs significantly within weekly timeframes. Figure 5.2 and Figure 5.3 display the chlorophyll for 8 August 2017 and 2018, respectively. They are plotted within the same colour range. Comparing the two images, the farmer can now visually see how the farm performed in 2018, compared to August of the previous year (2017). Python was used to compare each point value with the previous year's point value on the same day. The point colour is displayed according to the performance of 2018 compared to 2017 (see Figure 5.4). Figure 5.5 displays the chlorophyll values of 2019 compared to the performance of 2017 and 2018. The chlorophyll values on 8 August 2019 is performing poorly compared to the previous year's chlorophyll values on 8 August.

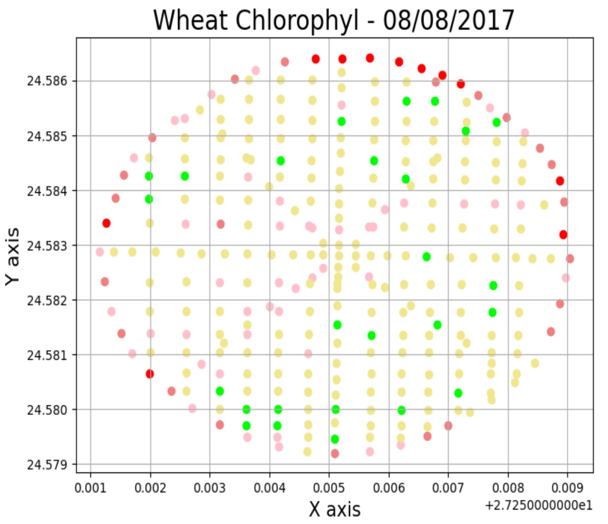


Figure 5.2: Chlorophyll per point for 8 August 2017

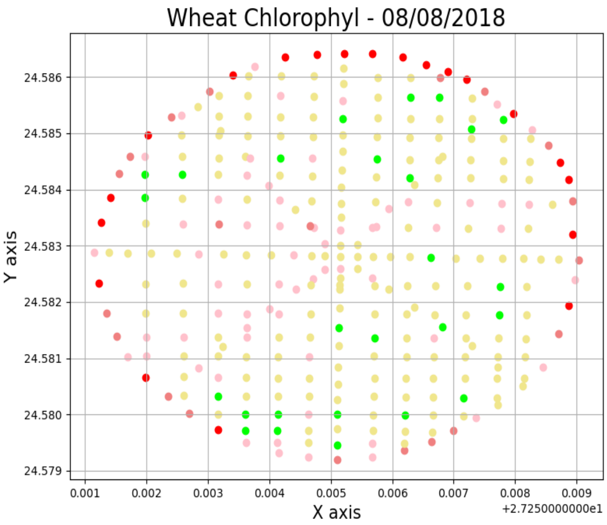


Figure 5.3: Chlorophyll per point for 8 August 2018

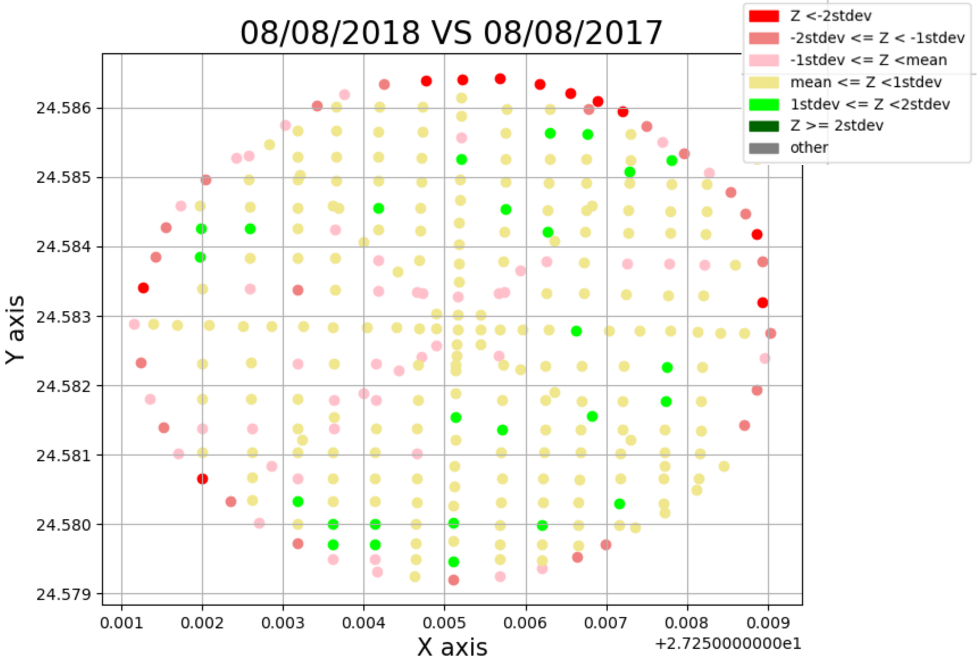


Figure 5.4: 2018 chlorophyll values compared to 2017 chlorophyll values per point

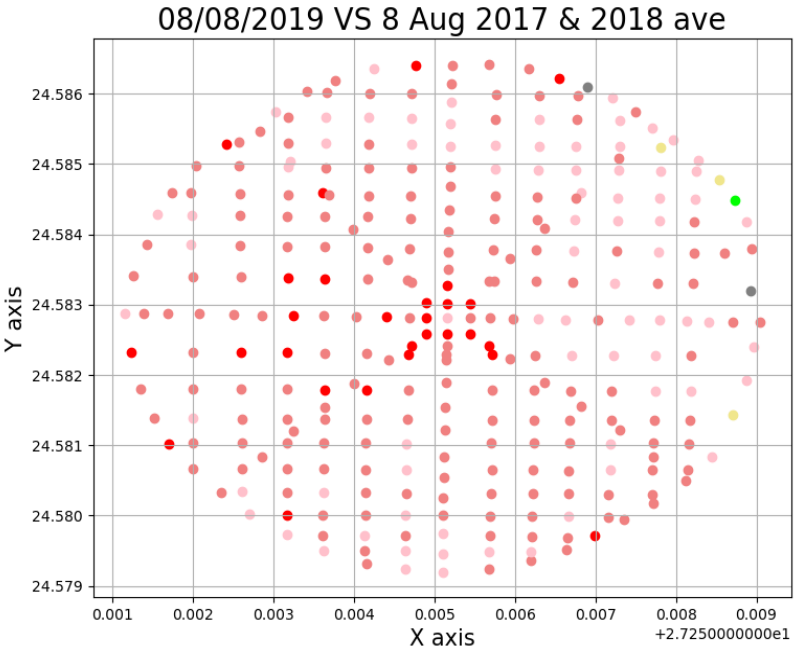


Figure 5.5: 2019 chlorophyll values compared to 2018 and 2017 chlorophyll values per point

5.3 Summary

This chapter explained how all the previous sections are combined to develop the concept demonstrator decision support tool. In particular, new comparison and prediction functions were included. The next chapter discusses the components of the decision support tool and illustrates how it can be used in a dashboard to improve decision-making.

Chapter 6
A next-generation decision support tool

Chapter 4 described the CRISP-DM methodology that was used as a guideline to clean and manipulate the data used in the prediction algorithm. The feature selector algorithm was used to choose a subset of nine features used in chlorophyll predictions when given known data for training and testing. This chapter discusses predictions made with known and unknown data. In Chapter 5, it was shown that *Farm X* lies between levels 3 and 4 of PA adoption and that it does not currently utilise predictive analytics.

This chapter examines how predictive analytics can be added and used in a decision support tool and how data can be displayed in a dashboard. Conceptual dashboards were developed to present information in a user-friendly way, particularly the "current status of the farm" and predictions for the following months derived from the algorithms and comparisons with previous years' performance.

6.1 Data for predictions

The real monthly chlorophyll averages for each year are shown in Figure 6.1, including the predicted September 2020 average presented in Chapter 4. All four years follow a similar chlorophyll trend, except for the chlorophyll values in October 2019, probably because the wheat was most likely only harvested in November. The data of each year were also examined separately to investigate the values in more detail. The average chlorophyll per month (blue line) and the standard deviation (red shading) for each month were calculated and graphed in Figure 6.2, Figure 6.3 and Figure 6.4. It was decided to isolate the data from 2018 for the predictions since it has the most "complete" data set.

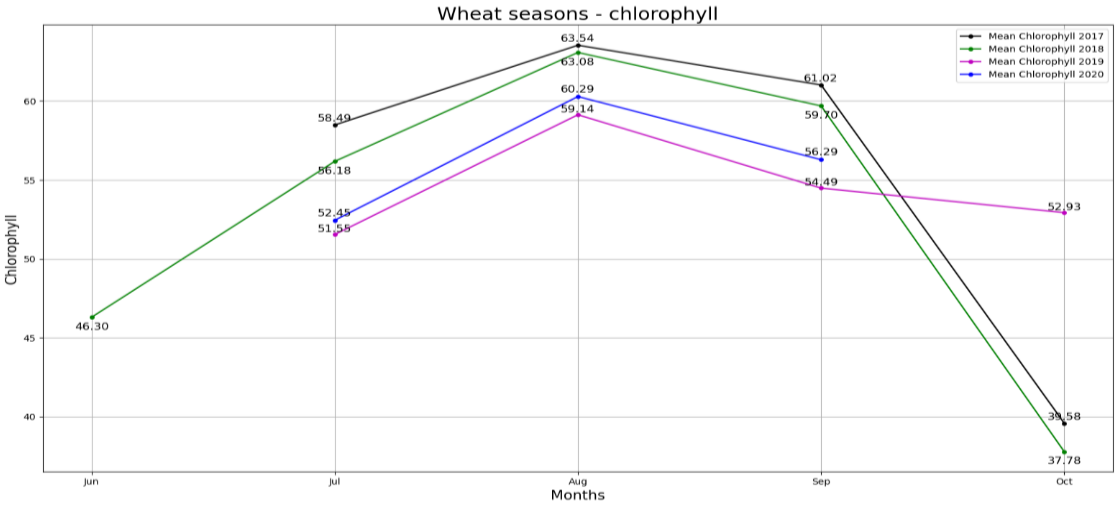


Figure 6.1: Monthly chlorophyll averages for the years 2017, 2018, 2019 and 2020

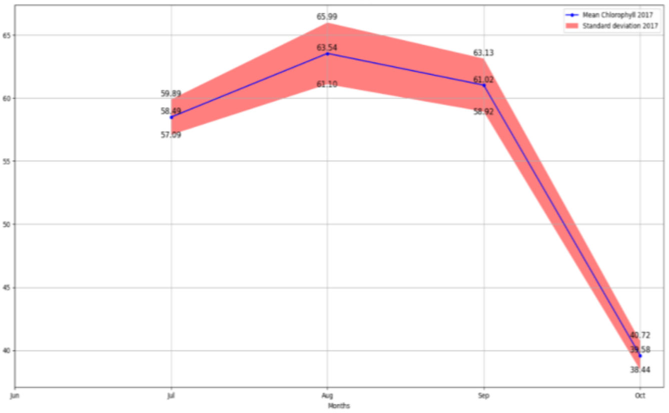


Figure 6.2: 2017 Chlorophyll monthly averages and standard deviation

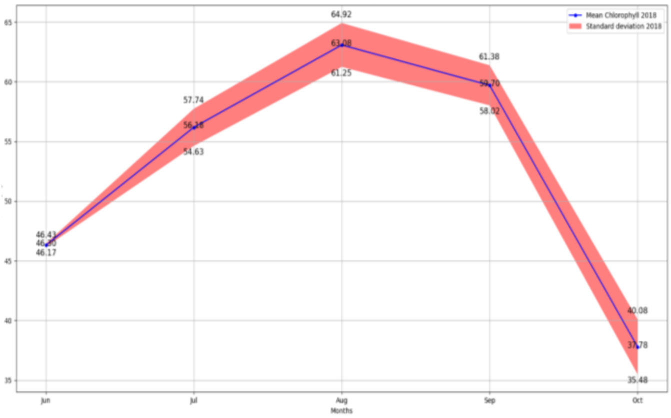


Figure 6.3: 2018 Chlorophyll monthly averages and standard deviation

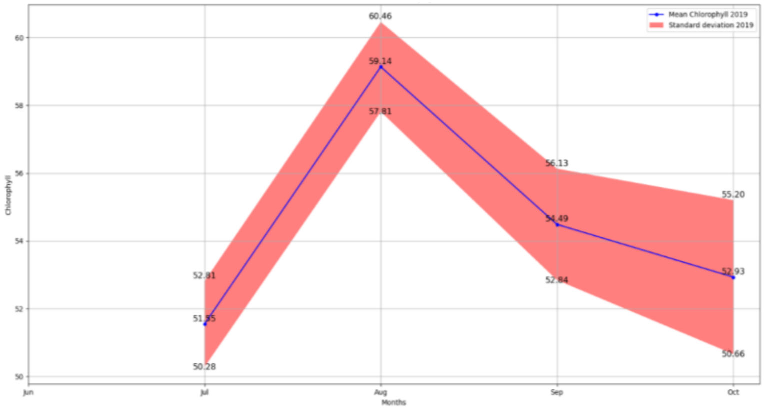


Figure 6.4: 2019 Chlorophyll monthly averages and standard deviation

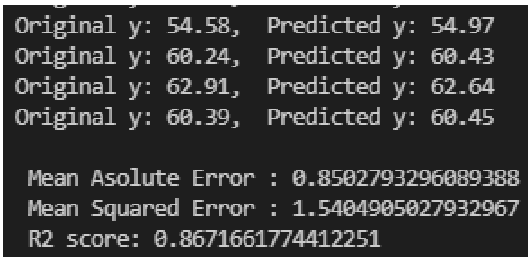
6.2 Test scenarios

The chlorophyll predictions were divided into two subsections using known and unknown test data. The data used for training and testing are discussed as well as the results and prediction accuracy. The regressors accuracy was measured by the coefficient of determination (R^2), MAE and MSE.

6.2.1 Known test data

A significant part of the work described in Chapter 4 was spent on data cleaning and data manipulation to prepare the data for the chosen prediction algorithm. The available data from 2016 to 2020 were used to run the feature selectors. After analysing all the wrapper methods, the ETR algorithm was chosen for the chlorophyll predictions. Despite the algorithms random nature, it produced the best R^2 value with every iteration. The train, test, split function was used in Python to split the data into 80% for training and 20% for testing with a random state of 42. It was discovered that with various scenarios, the model could make accurate chlorophyll predictions given a variety of nutrient, soil, chlorophyll and weather features.

An example of September predictions for the year combination 2017, 2018 and 2019 can be seen in the snippet in Figure 6.5. The training set consisted of 712 rows and 29 feature columns, and the X_test data table consisted of 179 rows and 29 feature columns. With X being the entire data set without the September chlorophyll column and Y being the September chlorophyll column.



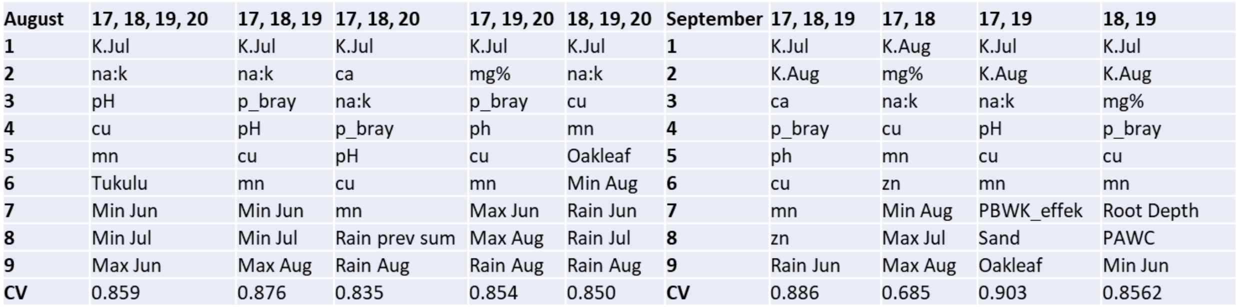
```
Original y: 54.58,  Predicted y: 54.97
Original y: 60.24,  Predicted y: 60.43
Original y: 62.91,  Predicted y: 62.64
Original y: 60.39,  Predicted y: 60.45

 Mean Asolute Error : 0.8502793296089388
 Mean Squared Error : 1.5404905027932967
 R2 score: 0.8671661774412251
```

Figure 6.5: Example of September prediction accuracy for the year combination 2017, 2018 and 2019

The various year combinations produced different feature subsets (see Table 6.1). Still, in most cases, the top five features were almost always the same nutrient and soil features, with variation in the last three features, including meteorological and soil type features. The assumptions from Chapter 3, Section 3.5 to include the extra features such as pH, Mn and Zn in the feature selection model proved successful.

For example, it was noted that Mn appeared in all of the feature selection subsets and pH in 6/9 of the feature subsets. Since the chlorophyll is presented in monthly instances, instead of, for example, weekly instances, the average monthly meteorological is too generalised for time-series data to have a substantial impact on the model. Only the minimum temperature, maximum temperature and rain seemed to influence the model's feature subset selection.

Table 6.1: Selected features and cross-validation (CV) score for various year combination data

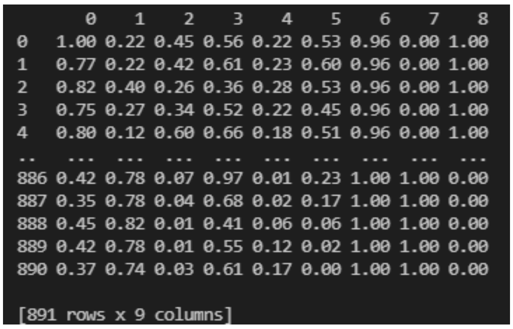


August	17, 18, 19, 20	17, 18, 19	17, 18, 20	17, 19, 20	18, 19, 20	September	17, 18, 19	17, 18	17, 19	18, 19
1	K.Jul	K.Jul	K.Jul	K.Jul	K.Jul	1	K.Jul	K.Aug	K.Jul	K.Jul
2	na:k	na:k	ca	mg%	na:k	2	K.Aug	mg%	K.Aug	K.Aug
3	pH	p_bray	na:k	p_bray	cu	3	ca	na:k	na:k	mg%
4	cu	pH	p_bray	ph	mn	4	p_bray	cu	pH	p_bray
5	mn	cu	pH	cu	Oakleaf	5	ph	mn	cu	cu
6	Tukulu	mn	cu	mn	Min Aug	6	cu	zn	mn	mn
7	Min Jun	Min Jun	mn	Max Jun	Rain Jun	7	mn	Min Aug	PBWK_effek	Root Depth
8	Min Jul	Min Jul	Rain prev sum	Max Aug	Rain Jul	8	zn	Max Jul	Sand	PAWC
9	Max Jun	Max Aug	Rain Aug	Rain Aug	Rain Aug	9	Rain Jun	Max Aug	Oakleaf	Min Jun
CV	0.859	0.876	0.835	0.854	0.850	CV	0.886	0.685	0.903	0.8562

6.2.2 Unknown test data

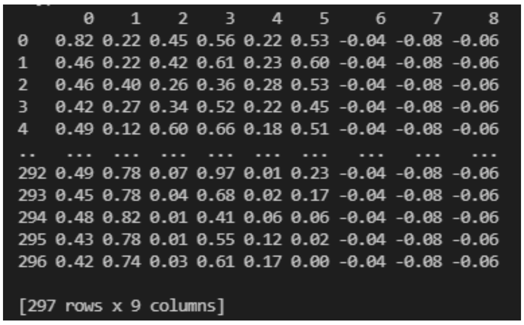
After exploring the model performance on known data, the ETR algorithm used various yearly scenario combinations to train the model with "unknown data". The unknown data refers to a chosen year that was not used in the training of the model. The various scenarios consisted of combinations of various years' 296 chlorophyll data points and features as training data (*X_train*). A separate year (*X_test*) was presented as unknown data. The X_train set was normalised using the MinMaxScaler as shown in Figure 6.6, and the *X_test* was transformed based on the normalisation of the *X_train* table (see Figure 6.7). After the features were selected, the *X_train* and *X_test* tables were transformed to include only the selected features from the selector. *X_test* was given to the model to make chlorophyll predictions, and the original chlorophyll values (*y_test*) was used to calculate the prediction accuracy.

There were three prediction options, where the first and second used July to make predictions for August and September (in the absence of August chlorophyll data). The third type of prediction was made with July and August chlorophyll data included in the *X_train* table to predict chlorophyll values for September.



```
        0    1    2    3    4    5    6    7    8
0    1.00 0.22 0.45 0.56 0.22 0.53 0.96 0.00 1.00
1    0.77 0.22 0.42 0.61 0.23 0.60 0.96 0.00 1.00
2    0.82 0.40 0.26 0.36 0.28 0.53 0.96 0.00 1.00
3    0.75 0.27 0.34 0.52 0.22 0.45 0.96 0.00 1.00
4    0.80 0.12 0.60 0.66 0.18 0.51 0.96 0.00 1.00
..    ...  ...  ...  ...  ...  ...  ...  ...  ...
886  0.42 0.78 0.07 0.97 0.01 0.23 1.00 1.00 0.00
887  0.35 0.78 0.04 0.68 0.02 0.17 1.00 1.00 0.00
888  0.45 0.82 0.01 0.41 0.06 0.06 1.00 1.00 0.00
889  0.42 0.78 0.01 0.55 0.12 0.02 1.00 1.00 0.00
890  0.37 0.74 0.03 0.61 0.17 0.00 1.00 1.00 0.00

[891 rows x 9 columns]
```

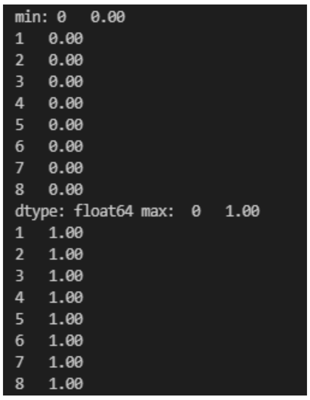
Figure 6.6: Normalised X_train table (2017, 2018, 2019)



```
        0    1    2    3    4    5     6     7     8
0    0.82 0.22 0.45 0.56 0.22 0.53 -0.04 -0.08 -0.06
1    0.46 0.22 0.42 0.61 0.23 0.60 -0.04 -0.08 -0.06
2    0.46 0.40 0.26 0.36 0.28 0.53 -0.04 -0.08 -0.06
3    0.42 0.27 0.34 0.52 0.22 0.45 -0.04 -0.08 -0.06
4    0.49 0.12 0.60 0.66 0.18 0.51 -0.04 -0.08 -0.06
..    ...  ...  ...  ...  ...  ...   ...   ...   ...
292  0.49 0.78 0.07 0.97 0.01 0.23 -0.04 -0.08 -0.06
293  0.45 0.78 0.04 0.68 0.02 0.17 -0.04 -0.08 -0.06
294  0.48 0.82 0.01 0.41 0.06 0.06 -0.04 -0.08 -0.06
295  0.43 0.78 0.01 0.55 0.12 0.02 -0.04 -0.08 -0.06
296  0.42 0.74 0.03 0.61 0.17 0.00 -0.04 -0.08 -0.06

[297 rows x 9 columns]
```

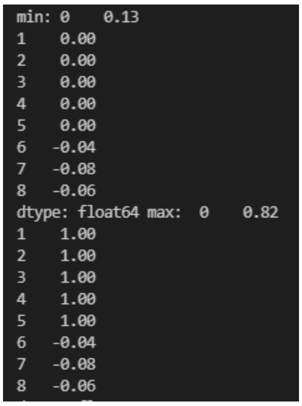
Figure 6.7: Normalised X_test table (2020)

The various scenarios were fed into the feature selector algorithm described in Section 4.4. It was found that the chosen features did not produce the same accuracy when presented with a data table from an unknown year, compared to the initial predictions with known data discussed in Section 6.2.1. These discrepancies illustrate the sensitivity of the model with regard to the completeness of the data set, in this case (i) the absence and quality of chlorophyll data captured as well as the (ii) nutrient and soil classification values that were not updated for each growing season. The unknown test year data table values are seen as “out of sample”, and after normalising the data, any “out of sample features” significantly influenced the accuracy of the prediction model. Figure 6.8 and Figure 6.9 show the minimum and maximum of each feature for X_train and X_test after it was transformed with the selected features. Transforming a data table entail removing the unwanted feature columns and only keeping the selected feature column chosen by the feature selector.



```
min: 0    0.00
1   0.00
2   0.00
3   0.00
4   0.00
5   0.00
6   0.00
7   0.00
8   0.00
dtype: float64 max:  0    1.00
1   1.00
2   1.00
3   1.00
4   1.00
5   1.00
6   1.00
7   1.00
8   1.00
```

Figure 6.8: Minimum and maximum of normalised and feature-transformed X_train



```
min: 0    0.13
1    0.00
2    0.00
3    0.00
4    0.00
5    0.00
6   -0.04
7   -0.08
8   -0.06
dtype: float64 max:  0    0.82
1    1.00
2    1.00
3    1.00
4    1.00
5    1.00
6   -0.04
7   -0.08
8   -0.06
```

Figure 6.9: Minimum and maximum of normalised and feature-transformed X_test based on X_train normalisation

6.3 Monthly chlorophyll predictions for 2018

After examining all the prediction outcomes of the various year combination scenarios, 2018 was used to plot the three prediction options. The blue line in Figure 6.10 represents the real average chlorophyll value, and the red shading represents the true standard deviation for 2018. The dashed line and dotted lines represent the average of the predicted values for August and September independently.

6.3.1 Using July data to predict August and September values

Figure 6.10 shows a potential user view for July. The average crop circle chlorophyll for June and July are displayed together with the standard deviation of the crop circle points. The average prediction points for August and September are shown by the dashed and dotted lines. In an ideal scenario, as time progresses, the model will update the blue line as true values and adjust the August and September predictions.

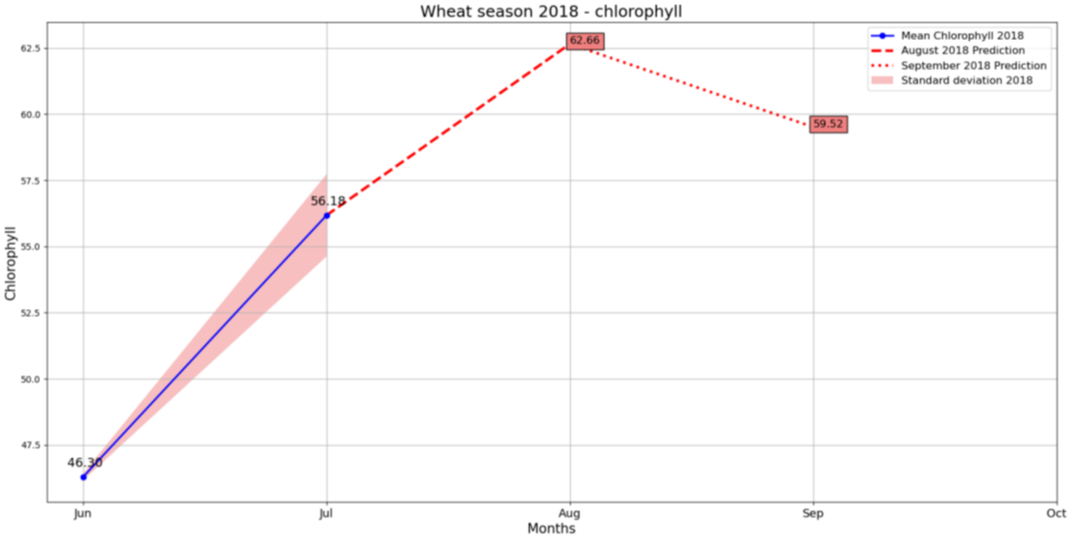


Figure 6.10: August and September chlorophyll prediction from July

6.3.2 Using August data to predict September values

The average August prediction of chlorophyll value of 62.66 can be compared to the actual average of 63.08. This seems close until one examines the R^2 value of -0.001, which provides a worse alternative to simply using the average to make a prediction. The model updated the table with the actual chlorophyll values for 2018 and then adjusted the chlorophyll predictions for September, with a mean of 59.37 compared to the July prediction average of 59.52.

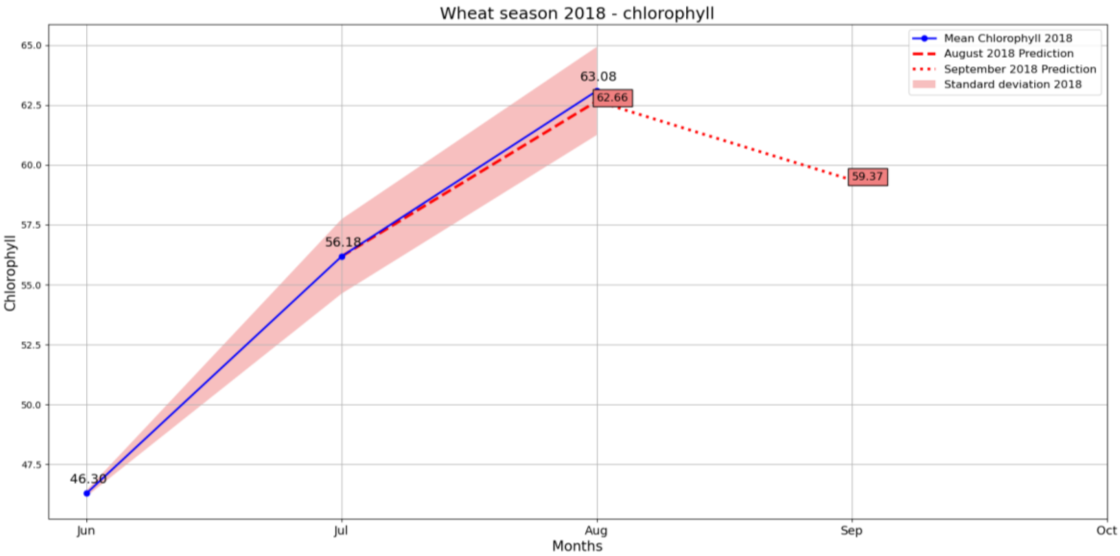


Figure 6.11: September chlorophyll prediction from July and August

6.3.3 September – compare predictions to true values

The model coped well with the August predictions for September and the mean of 59. The value of 37 was close to the true mean of 59.70. The R^2 value also improved to 0.15.

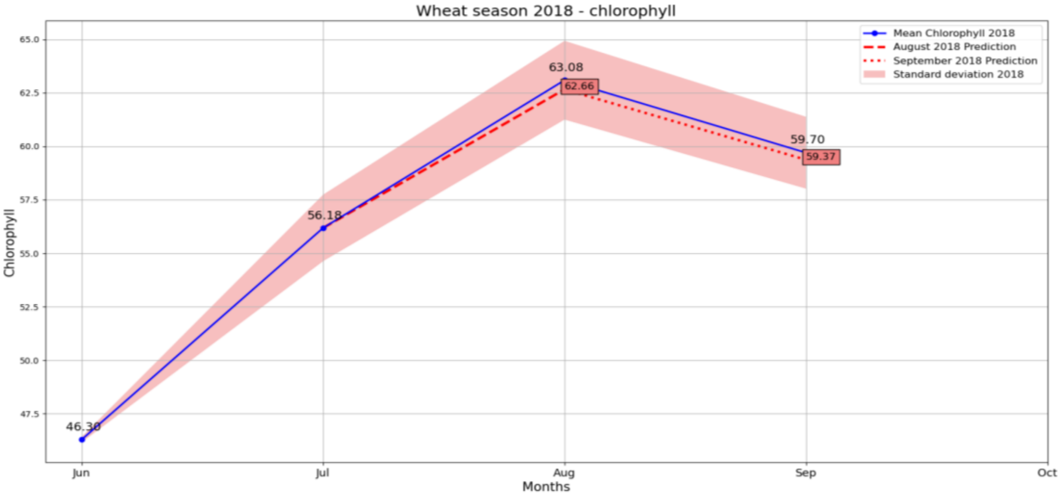
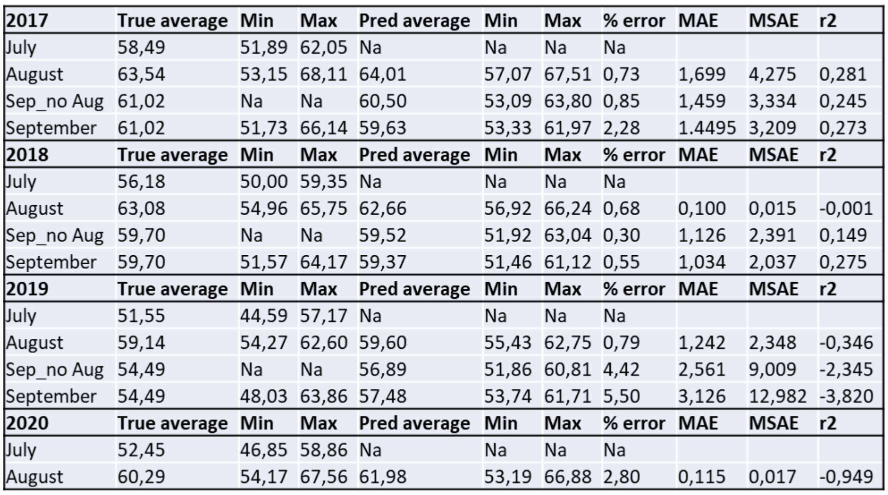


Figure 6.12: True and predicted monthly chlorophyll for August and September

Similarly, the other year combinations were tested and produced the results displayed in Table 6.2 below. One may be tempted to assume that the model is accurate because the predicted average is very close to the true average. However, by examining the R^2 values, it is evident that some predictions performed much better than others. This is due to the previously mentioned out of sample meteorological and chlorophyll data. The year 2019 performed the worst compared to the other years. Referring back to Figure 6.1: Monthly chlorophyll averages for the years 2017, 2018, 2019 and 2020, 2019 have the lowest average chlorophyll per month and an abnormal harvesting season for October.

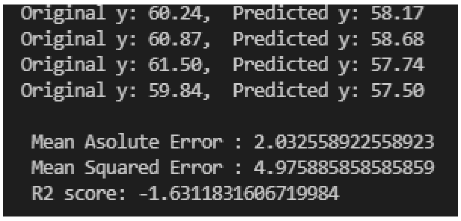
Table 6.2: True and predicted values for August and September for each year combination



2017	True average	Min	Max	Pred average	Min	Max	% error	MAE	MSAE	r2
July	58,49	51,89	62,05	Na	Na	Na	Na			
August	63,54	53,15	68,11	64,01	57,07	67,51	0,73	1,699	4,275	0,281
Sep_no Aug	61,02	Na	Na	60,50	53,09	63,80	0,85	1,459	3,334	0,245
September	61,02	51,73	66,14	59,63	53,33	61,97	2,28	1.4495	3,209	0,273
2018	**True average**	**Min**	**Max**	**Pred average**	**Min**	**Max**	**% error**	**MAE**	**MSAE**	**r2**
July	56,18	50,00	59,35	Na	Na	Na	Na			
August	63,08	54,96	65,75	62,66	56,92	66,24	0,68	0,100	0,015	-0,001
Sep_no Aug	59,70	Na	Na	59,52	51,92	63,04	0,30	1,126	2,391	0,149
September	59,70	51,57	64,17	59,37	51,46	61,12	0,55	1,034	2,037	0,275
2019	**True average**	**Min**	**Max**	**Pred average**	**Min**	**Max**	**% error**	**MAE**	**MSAE**	**r2**
July	51,55	44,59	57,17	Na	Na	Na	Na			
August	59,14	54,27	62,60	59,60	55,43	62,75	0,79	1,242	2,348	-0,346
Sep_no Aug	54,49	Na	Na	56,89	51,86	60,81	4,42	2,561	9,009	-2,345
September	54,49	48,03	63,86	57,48	53,74	61,71	5,50	3,126	12,982	-3,820
2020	**True average**	**Min**	**Max**	**Pred average**	**Min**	**Max**	**% error**	**MAE**	**MSAE**	**r2**
July	52,45	46,85	58,86	Na	Na	Na	Na			
August	60,29	54,17	67,56	61,98	53,19	66,88	2,80	0,115	0,017	-0,949

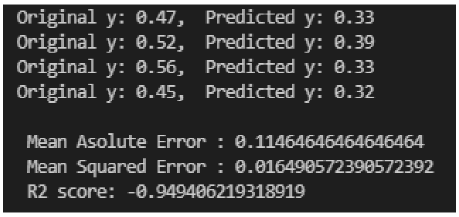
6.3.4 Exploring model parameter and performance

Algorithms related to the decision tree family rarely require normalisation, as the algorithm often handles the differing features ranges well. The prediction accuracy was tested by feeding the algorithm only normalised X table data (Figure 6.13) and normalised X and Y data (Figure 6.14). In every scenario, normalising both X and Y values improved the accuracy significantly.



```
Original y: 60.24,  Predicted y: 58.17
Original y: 60.87,  Predicted y: 58.68
Original y: 61.50,  Predicted y: 57.74
Original y: 59.84,  Predicted y: 57.50

 Mean Asolute Error : 2.032558922558923
 Mean Squared Error : 4.975885858585859
 R2 score: -1.6311831606719984
```

Figure 6.13: 17,18,19 Pred Aug20 - Only X normalised



```
Original y: 0.47,  Predicted y: 0.33
Original y: 0.52,  Predicted y: 0.39
Original y: 0.56,  Predicted y: 0.33
Original y: 0.45,  Predicted y: 0.32

 Mean Asolute Error : 0.11464646464646464
 Mean Squared Error : 0.016490572390572392
 R2 score: -0.949406219318919
```

Figure 6.14: 17,18,19 Pred Aug20 - X and Y normalised

6.4 Conceptual user dashboards

The research proposal discussed in Chapter 1, Section 1.2 highlighted the plethora of information presented to the user as one of the main issues to be addressed. To do so, the dashboard concept visualisations were designed to show how the analysed data and predictions could be incorporated to be useful to the user. The dashboards provide a summary of the features and calendar timeframe chosen by the user. The dashboard consists of two main views, viz. the “overview” and “weather” dashboards, displaying the current status and historical data. In some instances, the dashboard will show a warning related to a current or potential problem. Although the current status of the farm could look acceptable, it is important to compare the current farm status to previous years’ performances to “calibrate” the performance. Viable month and year dates were chosen for visualisation purposes and will be discussed below. Orange circle markers were added to the dashboards to simplify the explanations.

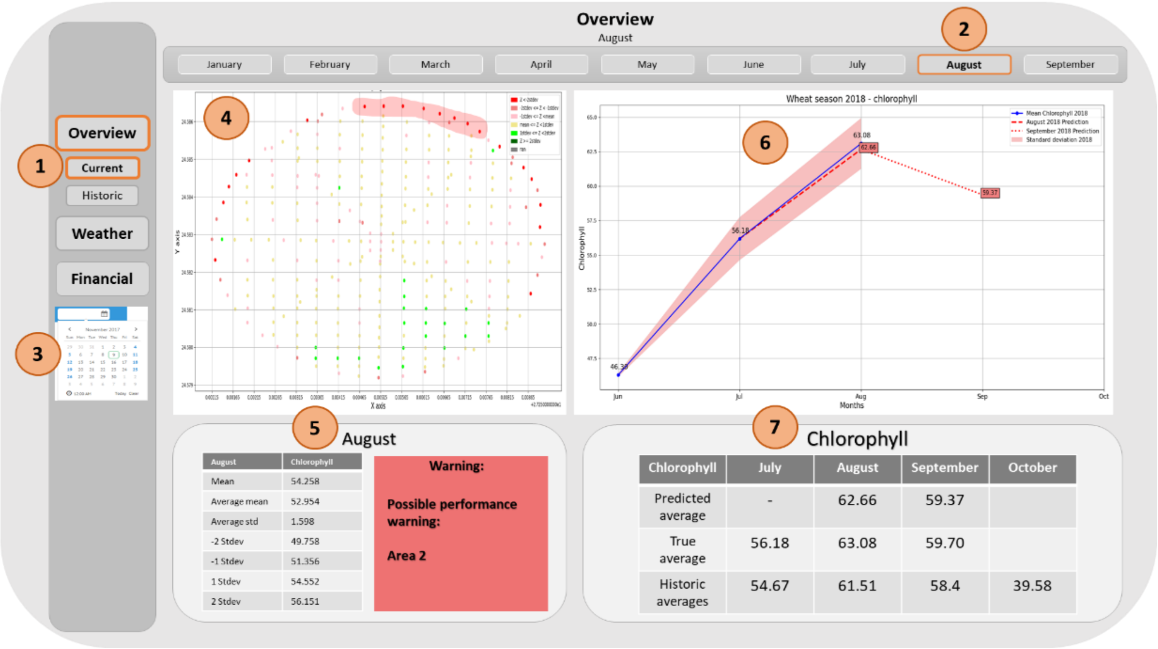
6.4.1 Overview of the dashboard

In the overview view of the dashboard, the user can choose to display the current status of the farm, view the historical performance and trends, or compare the current status performance (of chlorophyll) to historic chlorophyll data simultaneously.

Note: In the discussion below, numbers between brackets refer to the orange circle markers.

6.4.1.1 Current year performance – August 2018

The current status overview (1) was selected to display August 2018 (2) as an example for the current status dashboard. The dashboard is used to identify how the farm is currently performing in the specific month (in this case, August 2018) compared to previous monthly and yearly chlorophyll data. The timeframe to compare the current performance can be selected at (3), and in this example, the data is compared to 2017’s and 2019’s combined chlorophyll performance. Then, (4) visually shows the 298 individual geographical points’ performance compared to the chosen timeframe point performances. In other words, 2018’s point 3 chlorophyll value is compared to the average of both point 3’s chlorophyll from 2017 and 2019. This is an added feature to address the original research problem and aims to help the user identify the true performance of the crops. Suppose there is no individual historical point to compare the point data with. In that case, it will compare it with the standard deviation of the current date to investigate how it is performing compared to the rest of the crop circle. Ideally, this will not be necessary (as it would be preferred to have no missing data), but variables such as cloud cover affect the data quality in the real world. The model will also calculate when an area on the crop circle is severely underperforming based on the historical patterns and trends (4), (5). A monthly view of the average and predicted average monthly chlorophyll could be viewed in (6), accompanied by the tabulated values in (7), indicating the historical monthly average for all the previous years.



August	Chlorophyll
Mean	54.258
Average mean	52.954
Average std	1.598
-2 Stdev	49.758
-1 Stdev	51.356
1 Stdev	54.552
2 Stdev	56.151

Warning:

Possible performance warning:

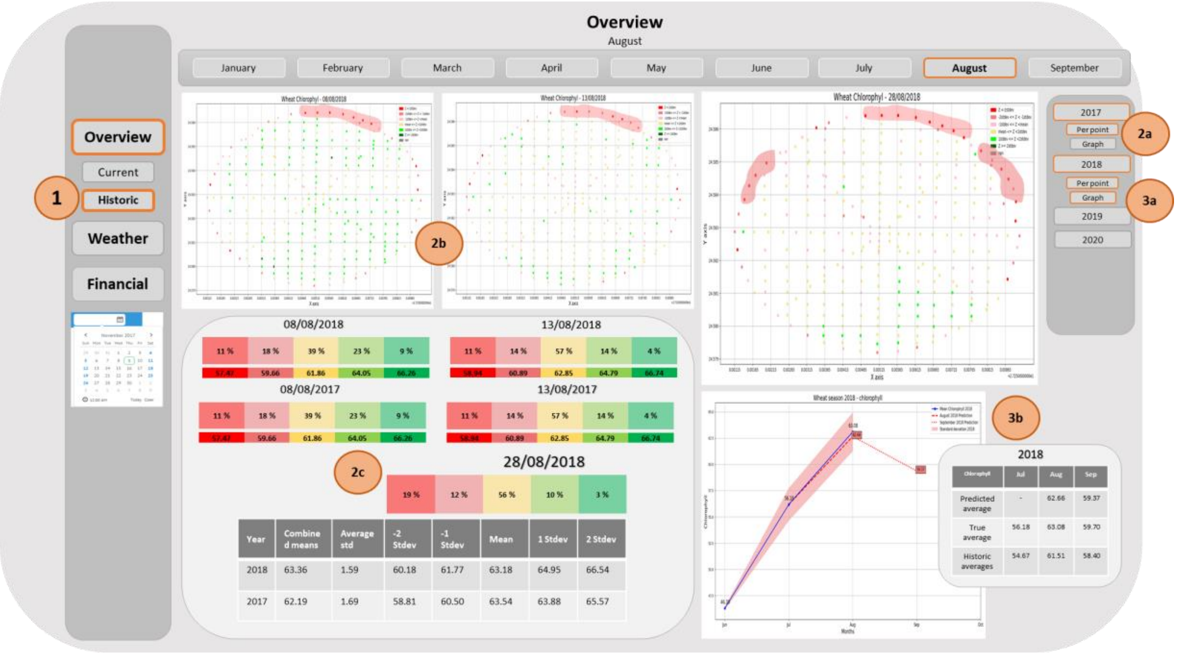
Area 2

Chlorophyll	July	August	September	October
Predicted average	-	62.66	59.37	
True average	56.18	63.08	59.70	
Historic averages	54.67	61.51	58.4	39.58

Figure 6.15: Current overview dashboard for August 2018

6.4.1.2 Historic performance

Figure 6.16 displays what a user will typically see on the dashboard when selecting the historical (1) option for August 2018. There are three satellite data files (in tiff format) for August 2018 that were manipulated and visualised in Python and displayed in (2b). In this case, the user compared the 2018's to 2017's (2a) performance seen at (2c). Lastly, the user also chose to see a graph (3a) displaying the average performance during August 2018, compared to the monthly averages of 2017 and previous years' monthly averages.



08/08/2018

11 %	18 %	39 %	23 %	9 %
57.47	59.66	61.86	64.05	66.26

13/08/2018

11 %	14 %	57 %	14 %	4 %
58.94	60.89	62.85	64.79	66.74

08/08/2017

11 %	18 %	39 %	23 %	9 %
57.47	59.66	61.86	64.05	66.26

13/08/2017

11 %	14 %	57 %	14 %	4 %
58.94	60.89	62.85	64.79	66.74

28/08/2018

19 %	12 %	56 %	10 %	3 %

Year	Combined means	Average std	-2 Stdev	-1 Stdev	Mean	1 Stdev	2 Stdev
2018	63.36	1.59	60.18	61.77	63.18	64.95	66.54
2017	62.19	1.69	58.81	60.50	63.54	63.88	65.57

2018

Chlorophyll	Jul	Aug	Sep
Predicted average	-	62.66	59.37
True average	56.18	63.08	59.70
Historic averages	54.67	61.51	58.40

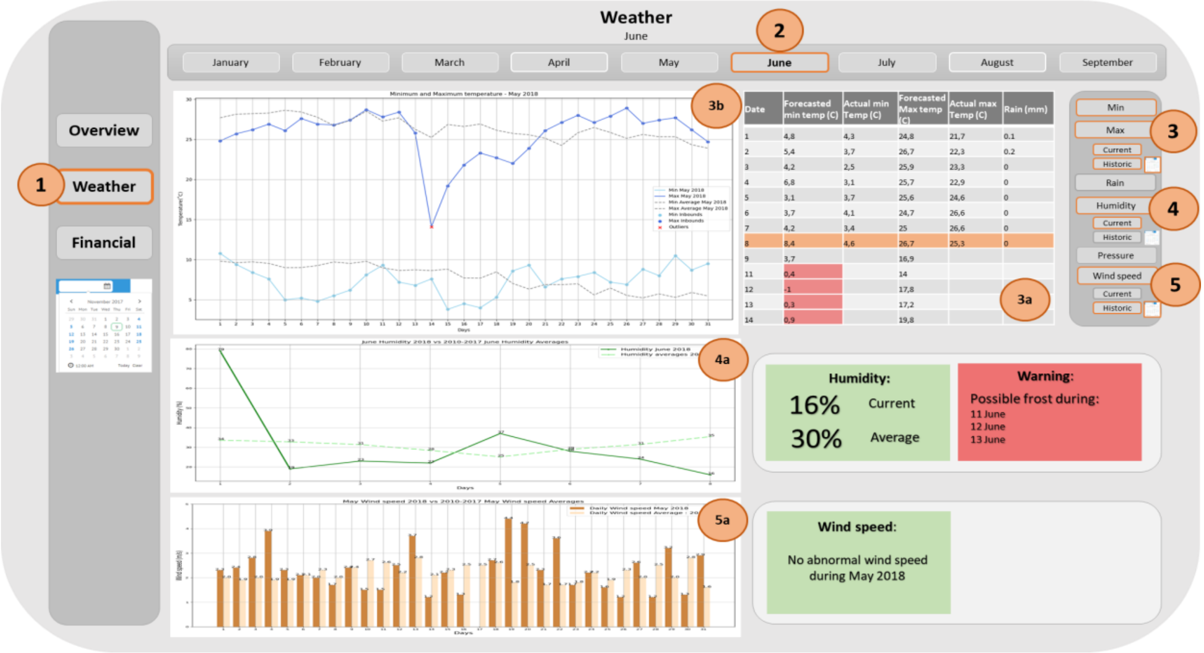
Figure 6.16: Concept dashboard: Current and historical overview for August 2018

6.4.2 Weather/Meteorological data

After exploring the importance of meteorological data in the literature discussed in Section 2.3 and the growing conditions in Chapter 3, Section 3.4, it was also decided to include this data in the dashboard design. The two dashboards below conceptually show the potential value of meteorological data to support decision-making.

6.4.2.1 Weather example A – 8 June 2018

The weather (1) dashboard for 8 June 2018 (2) displays a combination of current and historical weather data. The timeframes for each weather feature can be chosen next to the "Historic" button as seen next to (3), (4) and (5). As previously mentioned in Chapter 3, the South African grain guidelines suggest that the ideal growing temperature for winter wheat is between 5 - 25˚C. The daily minimum and maximum temperature for May 2018 is shown in the line graph at (3b) and compared to the daily average of 2010 to 2017. It can be observed that on 14 May 2018, the maximum temperature was below 15˚C, which is considered an outlier for May. The table at (3a) shows the current temperatures as well as the forecasted temperature. The red blocks are forecasted temperature values from the SAWS and indicate possible frost from 11 June 2018. This could be important since the wheat seedlings are extremely sensitive to hail and frost. When wheat is planted during May, it is also important to examine the windspeed for potential seed loss or lodging (5a).



Date	Forecasted min temp (C)	Actual min Temp (C)	Forecasted Max temp (C)	Actual max Temp (C)	Rain (mm)
1	4,8	4,3	24,8	21,7	0.1
2	5,4	3,7	26,7	22,3	0.2
3	4,2	2,5	25,9	23,3	0
4	6,8	3,1	25,7	22,9	0
5	3,1	3,7	25,6	24,6	0
6	3,7	4,1	24,7	26,6	0
7	4,2	3,4	25	26,6	0
8	8,4	4,6	26,7	25,3	0
9	3,7		16,9		
11	0,4		14		
12	-1		17,8		
13	0,3		17,2		
14	0,9		19,8		

Humidity:
16% Current
30% Average

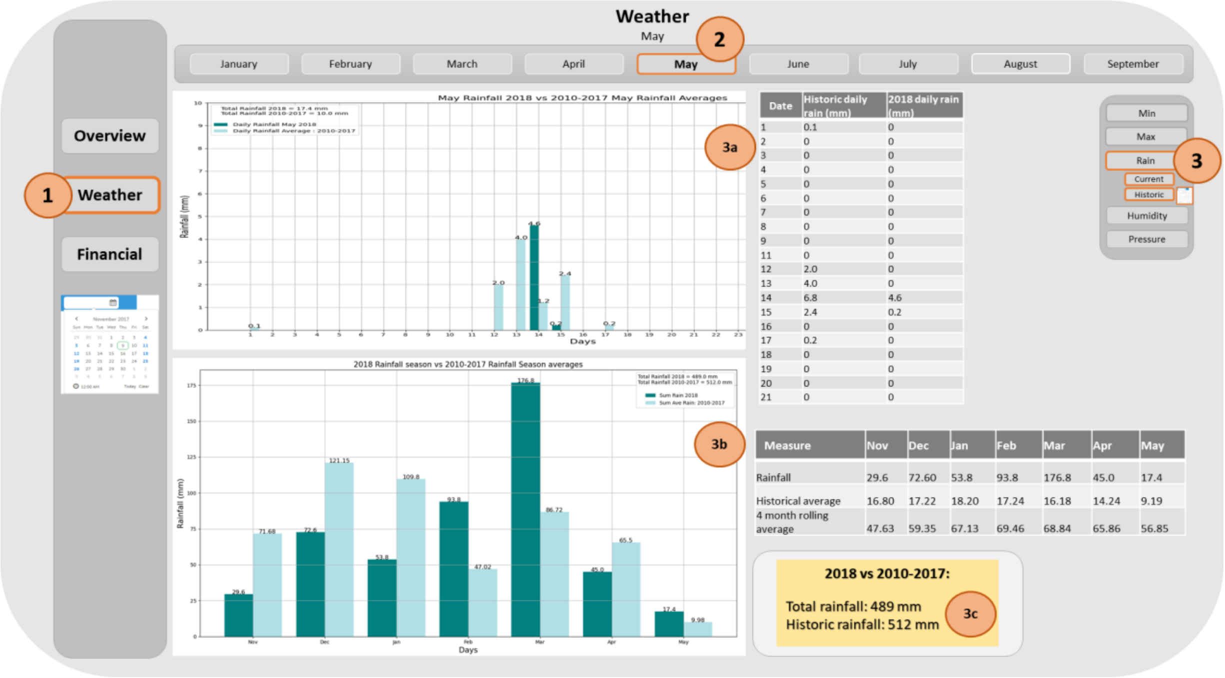
Warning:
Possible frost during:
11 June
12 June
13 June

Wind speed:
No abnormal wind speed during May 2018

Figure 6.17: Concept dashboard: 8 June 2018 current and historical weather

6.4.2.2 Weather example B – 21 May 2018

An examination of the South African Grain Guidelines indicates that winter wheat relies heavily on the residual soil moisture of the previous rain season. Thus, this dashboard supports the user in related decision-making such as irrigation scheduling. The bar chart (3a-left) and table (3a-right) show the daily rain in May 2018 and the daily averages from 2010 to 2017. Another bar chart at (3b) displays the total monthly rainfall for the previous six months and the historical monthly averages, selected at the button and calendar icon next to (3). Finally, the text box at (3c) is displayed in yellow since a previous season rainfall of 600 mm is preferred when planting wheat in May, but 2018 only produced a total rainfall of 489 mm from November 2017 to April 2018.



Date	Historic daily rain (mm)	2018 daily rain (mm)
1	0.1	0
2	0	0
3	0	0
4	0	0
5	0	0
6	0	0
7	0	0
8	0	0
9	0	0
11	0	0
12	2.0	0
13	4.0	0
14	6.8	4.6
15	2.4	0.2
16	0	0
17	0.2	0
18	0	0
19	0	0
20	0	0
21	0	0

Measure	Nov	Dec	Jan	Feb	Mar	Apr	May
Rainfall	29.6	72.60	53.8	93.8	176.8	45.0	17.4
Historical average	16.80	17.22	18.20	17.24	16.18	14.24	9.19
4 month rolling average	47.63	59.35	67.13	69.46	68.84	65.86	56.85

2018 vs 2010-2017:

Total rainfall: 489 mm
Historic rainfall: 512 mm

Figure 6.18: Concept dashboard: 21 May 2018 current and historic rain

6.5 Summary

This chapter discussed the testing scenarios for predicting chlorophyll values given known and unknown data tables. This predictor function will be able to provide the farmer with “early warning” estimates of the future performance of the crops.

The ETR algorithm produced an R^2 value of 0.85 when the algorithm was given known data for training and testing. When the algorithm was used to train on known data but test on unknown data, the R^2 values dropped significantly due to the model regarding small changes in the data as “out of sample”. The ETR algorithm performed well for predicting August and September chlorophyll for 2017 with R^2 values of 0.281 and 0.273, respectively. The models produced negative R^2 values for the August and September predictions for 2019 and 2020. By referring back to the monthly chlorophyll displayed in Figure 6.1, it can be observed that the monthly chlorophyll was lower than 2017 and 2018 for each month.

Chapter 7
Validation and verification

This chapter discusses the validation process of the concept demonstrator decision support tool. Model validation refers to the process of determining whether the model accurately represents a real-world system. Three popular types of simulation validation are mentioned by Law (2015), which are also applicable within the context of this concept demonstrator: conceptual validity, operational validity and credibility.

7.1 Operational validity

Operational validity determines whether the model's output represents that of a real-world system and is typically confirmed by means of results validation. The testing data set is a separate portion of the same data set from which the training set is derived. The main purpose of using the testing data set is to test the generalisation ability of a trained model.

The project differs slightly since it is only tested on real-world data to confirm operational validity. The operational validity was done in Chapter 4 and Chapter 6, which discussed the use of known and unknown data to test the model. The various feature subsets from each of the four algorithms each produced different results for the predictions of chlorophyll. The performance of the algorithms was measured by comparing the actual data with the predicted chlorophyll data. MAE, MSE and R^2 were used to measure the accuracy of the regressor algorithm performances.

7.2 Conceptual validity and credibility

Conceptual validity is used to determine whether a model is a valid representation of the real world. Face validation is the most popular technique, which involves asking knowledgeable individuals if the model is comparable to the real world. Three SMEs were consulted to evaluate the approach and technical aspects of the decision support tool. The researcher presented an online presentation, and the SMEs were provided with the (i) thesis outline, (ii) problem statement, research objectives, project scope and limitations, (iii) data analysis approach, (iv) ML algorithms and test scenario predictions, (v) concept dashboards and (vi) recommendations for future work. They were supplied with a short questionnaire and asked to provide their professional opinions regarding the research approach and development of the concept demonstrator. The names of the SMEs are disclosed, as agreed in the NDA contract, and will be referred to as SME1, SME2 and SME3. However, they do have several combined years of experience in data analytics, AI, and digital solutions in agriculture. The questions were divided into categories and will be discussed below.

7.2.1 Approach, data collection and data analysis

The SMEs were asked to comment on the following points regarding the research approach and data analysis:

1. **The use of QGIS to superimpose and extract the data into an Excel workbook.** The experts agreed that consolidating the tiff files containing the chlorophyll and the soil and nutrient data layer is useful for further analysis. SME2 felt that the Python plug-in in QGIS should be utilised to import the data table directly into Python. SME3 agreed with SME2 that exporting the data into an Excel workbook might not be optimal for future work since adding additional data, such as yield data, might contain millions of 'pixels' yield points. SME1 preferred the tabular data in Excel to better visualise the data and an easier method of initial data inspection and cleaning for this use case containing a third missing value for chlorophyll.

2. The use of the CRISP-DM method as the basis for the data analysis and the method of dealing with missing values and the decision to consider feature subsets instead of linear relationships for between features.
   All three SMEs believe that an appropriate systematic approach to the data analysis was followed and that analysing the relationship between features when choosing the best performing subset is important.

3. **The approach to determine the best ML algorithm for the feature subset selection and the chlorophyll predictions.**
   The SMEs were interested in seeing the top-performing algorithms for the feature subset selection and chlorophyll predictions for the known and unknown data. SME1 suggested further exploration into the use of minimum, maximum and average values of the weather data for when more chlorophyll data and yield data is available for analysis and predictions.

7.2.2 Concept demonstrator

The SMEs were presented with the four concept dashboards to illustrate how the given components (chlorophyll data, weather data, ML algorithms, visualisation tools) can be used in a decision support tool. They were asked whether they agreed with the use of visualisation tools such as graphs, tables and point-specific heatmaps to illustrate the features, predictions, weather trends and derived values. They were also asked whether they think that the proposed concept demonstrator could be useful to a potential end user and improve the decision-making process. The comments are summarised below.

SME1 feels that a decision-making tool, such as this one, can assist the user in identifying problem areas and whether crops are performing as expected. Farmers are provided with better insights into existing data, which would otherwise be hard to interpret. SME2 felt that it can also be extremely useful for subject experts, such as agronomists, who have intrinsic knowledge about the data and know what to look out for. SME3 stated that a high number of variables with geospatial and temporal variability requires tools to analyse, simplify, and visualise the data for farmers and their expert advisors.

“The power of visualisation is under-estimated”- SME3

7.2.3 Recommendations

The researcher’s recommendations for future work were presented to the SMEs, and they were asked to rate it according to a Likert scale and provide additional comments. The recommendations are presented in questions and statements.

1. **Updating soil and nutrient classification data yearly can assist in the analysis and prediction of chlorophyll and yield.**
   Soil classification data is a static data set (except in the case of major earthworks). SME1 agreed with the research suggestion to update the nutrient and soil chemistry data sets seasonally. SME2 and SME3 believe that updating it annually will not add obvious value to the analysis and that it should be conducted every 2-3 years.

2. **Implementing GPS specific yield (e.g., GPS systems in tractors) can improve the prediction model.**
   All three SMEs strongly agreed with this recommendation. SME1 believes that adding GPS-specific yield can relate the model more closely to the variable of interest from a business point of view.

3. **More frequent imagery can add value by improving predictions and early warning.**
   All of the SMEs agreed that increasing the number of images will add value and improve the model's predictability but did comment that the optimal number of images is unknown. SME2 believes six times a month is financially sensible, but daily imagery could add more value to real-time decision support and early warning tools.

4. Literature suggests that adding crop indicators such as FCover, MSAVI, and LAI can improve yield prediction. Do you agree that adding more indicators can improve yield prediction?
   The SMEs all expressed their curiosity about the potential benefit of adding more crop indicators to a prediction model (such as the one discussed in this document). SME2 commented that more vigour and growth might not always be correlated with yield, as in some cases, the plant pushes more energy into the leaf and not into the fruit.

5. **Adding pest and disease data can add value to a decision support system?**

SME1 suggests that adding pest and disease data can improve the tool's performance and offer a better explanation of the chlorophyll trend (potentially yield too). SME3 states that it could be more useful for some crops than others but agrees with the recommendation to add the pest and disease data.

6. **Adding meteorological data sets (historical, current and forecasted) can improve decision support?**
   The SMEs all agreed that adding more detailed weather data can add considerable value to a decision support tool. It can assist the users in detecting anomalies, taking preventative actions, and increasing the accuracy of expert advice to the farmer. SME3 states that it is well-known that micro-climate has a significant impact on crop production.

7. Integrating market-related data (demand, price etc.) can provide estimations of predicted profits and assist the farmers with crop selection.
   SME1 and SME3 agreed that it could be useful to add in the presence of yield data and more specific yield predictions. However, not all farmers can switch crops on short notice, and SME2 stated that the recommended feature might not be useful in all situations.

7.2.4 Additional research

The SMEs were also asked to provide their expert opinions on the adoption of PA technologies.

1. What are the challenges and limitations that influence the adoption of advanced technologies in agriculture? (Global or South African perspective)
   The most important factors influencing the adoption of PA according to SME1 is the lack of high-quality data, high implementation costs and the understanding and familiarity with data-driven decision-making. SME2 explained that the average age of farmers is increasing and that very few young people are taking up farming as a career, which often leads to another challenge - resistance to change and adoption of new technologies. Both SME2 and SME3 mention the challenge of Internet connectivity in rural areas, critical for some PA technologies. SME3 states that hardware in a laboratory or factory often does not last in actual farming environments due to a farm's “rugged” environment. Many advanced technologies have become affordable (i.e., have a high financial return), but SME3 feels that some technologies are still far too expensive for commercial adoption. Another challenge is the disparate data formats from various sources such as satellites, yield monitoring devices, IoT devices and lab results that require manipulation and integration into a single tool.

2. Please provide your view on the following statement: “It is not financially viable for small and medium-scale farmers in developing countries to adopt smart farming technologies and decision support tools.”
   All three SMEs disagreed with this statement. It is not a binary question of “adoption” or “no adoption”, but rather a situation investigating which technologies would make most financial sense for small- and medium-scale farmers. The key business decisions and requirements should be used to prioritise and determine which technologies should be implemented.

7.3 Summary

This chapter discussed operational validation and focussed on the conceptual validation of the concept demonstrator tool. The general feedback regarding the research study was overwhelmingly positive. The final comments suggested testing the dashboards and early warning components with farmers to get further input from a different user perspective. This was not part of the project scope due to the anonymity of the farmer agreed to in the NDA contract, but it will be useful for the next stage. The major points derived from the SME feedback refer to the importance of adding meteorological data and yield data to a decision support tool. The factors influencing PA adoption mentioned in the feedback related to the literature study's research (Section 2.1). The challenges of PA adoption should be considered when developing a decision-support tool such as the one described in this thesis. Some aspects of the tool still require improvement and further study to make it a tool that can be commercially rolled out in the future.

Chapter 8
Summary, recommendations and conclusion

This chapter provides a summary of the research study. The three main recommendations for future work regarding soil and nutrient data, yield data and GIS work are discussed. Finally, the chapter ends with the conclusion and fulfilment of research questions discussed in the problem statement in Chapter 1 (Section 1.2).

8.1 Summary

The research described in this study followed a systematic approach to develop a concept demonstrator for a decision support tool that can be used in agriculture. The initial literature study and research questions were refined and adapted to be applicable to a real-world problem related to *Farm X*, producing winter wheat in South Africa. Thereafter, a comprehensive literature review was conducted in parallel with the necessary field research to understand the nature of the real-world problem and develop a concept demonstrator. The use of weather and climate data in current PA applications were researched. The assumption was made that weather data should be added to a decision support tool to improve decision-making activities, and in this case, chlorophyll predictions. The weather data from the specific region of *Farm X* were acquired from the SAWS. The SMEs later validated the assumptions, which suggested that weather data can add enormous value to a decision-support tool.

The CRISP-DM methodology served as a guideline for the data analysis. QGIS software was used to extract the data into a table in Excel, which was then imported into Python for further analysis. The weather data were also cleaned and combined with the soil and nutrient and chlorophyll data table. After constructing the final data table, a sequential forward feature selector was used to select the subset features utilised in the prediction algorithm. The top-performing algorithms were (i) Random Forest regressor, (ii) HistgradientBoost regressor, (iii) XGB regressor and the (iv) ETR. The algorithms were compared and delivered R^2 values of 0.846, 0.844, 0.830 and 0.863, respectively. The main features selected by the four algorithms were July chlorophyll, August chlorophyll, Mn and pH. The weather features chosen by the feature selector were minimum temperature, maximum temperature and rainfall. The ETR produced the best results in each prediction iteration and was chosen for further data analysis described in Chapter 6. Thereafter, the model was presented with a data table from an unknown year to predict the chlorophyll values for August and September. The model's accuracy decreased from 0.86 to 0.273, which was expected due to a third of the chlorophyll time-series data missing and the soil and nutrient layer not being updated within the standard two-to three-year period. The model's prediction accuracy completed in Chapter 4 and Chapter 6 served as the operational validation of the tool. It shows how a model reacts to the real world and which factors potentially influence decision-making when available through chlorophyll, soil and nutrient data. Three SMEs were approached to explore the conceptual validity of the model, and the general feedback was positive. The SMEs agreed that adding yield data will improve the decision-support tool and assist in exploring predictions and the relationship between variables.

8.2 Recommendations

The study was modelled on a real-world use case. Though the objective was to develop a concept demonstrator, there is ample opportunity for improvement for future work. Suggestions for possible future work that transpired during the concept model research and development are discussed below.

8.2.1 Soil and nutrient classification layer

The field research conducted in Chapter 3 indicated that it is common practice to update the soil and nutrient classification layer once every three years. The data from *Farm X* is only updated every five years, and the soil and nutrient features were considered static features. The research done in the literature study indicated how biophysical parameters such as soil moisture and pH could be used for better crop management. The model struggled to find strong correlations between chlorophyll and the soil and nutrient layer features, as the chlorophyll changed over time, whilst the soil layer remained static. It is thus suggested to update the layer annually. The farm can even implement soil sensors and utilise Airbus Verde’s soil analysis service to decrease manual in-field data collection time and cost.

8.2.2 Yield

It is evident from the literature (Section 2.6.3) that yield is a valuable factor contributing to a farm's success. Yield prediction is an essential component in PA and can help farmers decide which crops to grow and when to grow them. Yield prediction can be used in yield mapping and conjunction with demand requirements and expected profitability. A recommendation for future work would be implementing a GPS yield monitoring device to collect more specific yield data. This can be used to determine factors that directly influence the yield and potentially warn the farmer if a problem is identified in the field.

8.2.3 Data issues

Several problems arose during the data analysis described in Chapter 4. A third of the extracted data had to be discarded as no data points were available on the crop circle. The cloud cover often completely obstructed the farm, and no chlorophyll data could be supplied. The average chlorophyll per month was used to ensure at least one value per point per month of chlorophyll. It is thus recommended to explore the potential use of a drone to increase the frequency of the remote sensing imagery. It would be helpful to collect imagery one to two times a week and test the model on the improved data to explore if it could improve the prediction accuracy. It is also recommended to explore other indicators such as NDVI, FCover and LAI in a model such as the concept demonstrator presented in the thesis.

8.3 Conclusion

The concept demonstrator was successfully developed in this research study. It illustrated how different data sets, ML algorithms, predictions and visualisation tools could be integrated and used in a decision support tool (RQ4, RQ5, RQ6, RQ8). The decision support tool was presented in the form of conceptual dashboards that displayed chlorophyll predictions and weather data effectively (RQ7). The chlorophyll data analysis and predictions provided better insight into the existing data by analysing specific GPS points on the crop circle and comparing them to previous years. Users can identify the exact location of problem areas and determine whether the crops are performing as expected. In addition, the study showed how predictive analytics can be used to detect patterns in agricultural data and that ML algorithms can determine which features/variables are important in prediction and decision-making (RQ9, RQ10).

References

2U. 2021. *What is data analytics?* [Online]. Available: https://www.mastersindatascience.org/learning/what-is-data-analytics/ [2021, October 14].

Abdulridha, J., Ampatzidis, Y., Kakarla, S.C. & Roberts, P. 2019. Detection of target spot and bacterial spot diseases in tomato using UAV-based and benchtop-based hyperspectral imaging techniques. *Precision Agriculture*, 21(5):955-978.

ActiveState. 2020. *Top 10 Python packages for machine learning* [Online]. Available: https://www.activestate.com/blog/top-10-python-machine-learning-packages/ [2021, May 14].

African Farming. 2020. *Airbus launches AI-powered crop analytics service* [Online]. Available: https://www.africanfarming.net/crops/agriculture/airbus-launches-ai-powered-crop-analytics-service [2021, November 04].

Agrawal, R. 2020. *How Microsoft is building new tech to bring precision agriculture to the world's poorest farmers* [Online]. Available: https://news.microsoft.com/en-in/features/microsoft-farmbeats-building-tech-precision-agriculture-world-poorest-farmers/ [2021, October 25].

Agriculture Research Council. 2014. *Insect pests of leaves and stems* [Online]. Available: https://www.arc.agric.za/arc-sgi/Pages/Crop Protection/Insect-pests-of-leaves-and-stems.aspx [2021, May 21].

Airbus. 2019. *Airbus adds new service Verde to its precision farming portfolio* [Online]. Available: https://www.airbus.com/en/newsroom/press-releases/2019-02-airbus-adds-new-service-verde-to-its-precision-farming-portfolio [2021, November 03].

Amazon Web Services. 2021. *What is data labeling for machine learning?* [Online]. Available: https://aws.amazon.com/sagemaker/groundtruth/what-is-data-labeling/ [2021, October 14].

Ashourloo, D., Mobasheri, M.R. & Huete, A. 2014. Developing two spectral disease indices for detection of wheat leaf rust (Pucciniatriticina). *Remote Sensing*, 6(6):4723-4740.

Azevedo, A. & Santos, M.F. 2008. KDD, semma and CRISP-DM: A parallel overview. *MCCSIS'08 - IADIS multi conference on computer science and information systems; Proceedings of informatics 2008 and data mining 2008*, 182–185.

Balaghi, R., Tychon, B., Eerens, H. & Jlibene, M. 2008. Empirical regression models using NDVI, rainfall and temperature data for the early prediction of wheat grain yields in Morocco. *International Journal of Applied Earth Observation and Geoinformation*, 10(4):438-452.

Ballesteros, R., Intrigliolo, D.S., Ortega, J.F., Ramírez-Cuesta, J.M., Buesa, I. & Moreno, M.A. 2020. Vineyard yield estimation by combining remote sensing, computer vision and artificial neural network techniques. *Precision Agriculture*, 21:1242-1262.

Barnes, J. 2018. *Drones vs Satellites: Competitive or Complementary?* [Online]. Available: https://www.commercialuavnews.com/infrastructure/drones-vs-satellites-competitive-complimentary [2021, October 24].

Baumann, P.R. 2009. *History of remote sensing, satellite imagery* [Online]. Available: http://employees.oneonta.edu/baumanpr/geosat2/rs history ii/rs-history-part-2.html [2021, April 20].

Bolton, M. 2020. Drones for mustering improves safety and efficiency on rural properties. *ABC News*. 20 July [Online]. Available: https://www.abc.net.au/news/rural/2020-07-21/drone-musters-cattle-safely-at-calliope-station/12468058 [2021, October 24].

Brown, J.N., Hochman, Z., Holzworth, D. & Horan, H. 2018. Seasonal climate forecasts provide more definitive and accurate crop yield predictions. *Agricultural and Forest Meteorology*, 260–261:247–254.

Brownlee, J. 2020a. *Supervised and unsupervised machine learning algorithms* [Online]. Available: https://machinelearningmastery.com/supervised-and-unsupervised-machine-learning-algorithms/ [2021, October 17].

Brownlee, J. 2020b. *How to use StandardScaler and MinMaxScaler transforms in Python* [Online]. Available: https://machinelearningmastery.com/standardscaler-and-minmaxscaler-transforms-in-python/ [2021, September 10].

Brymann, A. & Bell, E. 2011. *Research Methodology*.

Campbell, D.G.S. n.d. *Leaf Area Index (LAI): The researcher's complete guide* [Online]. Available: https://www.metergroup.com/environment/articles/lp80-pain-free-leaf-area-index-lai/ [2021, May 01].

Carbonell, I.M. 2016. The ethics of big data in big agriculture. *Internet Policy Review*, 5(1):1-13.

Ceres. 2021. *Chlorophyll Index* [Online]. Available: https://www.ceresimaging.net/en/chlorophyll-index [2021, October 25].

CFI. 2018. *Python (Machine Learning) - Overview, advantages* [Online]. Available: https://corporatefinanceinstitute.com/resources/knowledge/other/python-in-machine-learning/ [2021, May 14].

CFI, C. finance institute. 2015. *Decision Support System (DSS) - Overview, components, types* [Online]. Available: https://corporatefinanceinstitute.com/resources/knowledge/other/decision-support-system-dss/ [2021, May 06].

Chapman, P., Clinton, J., Kerber, R., Khabaza, T., Reinartz, T., Shearer, C. & Wirth, R. 1999. *CRISP-DM 1.0 Step-by-step data mining guide*. DaimlerChrysler.

Charfaoui, Y. 2020. *Hands-on with feature selection techniques* [Online]. Available: https://heartbeat.fritz.ai/hands-on-with-feature-selection-techniques-wrapper-methods-5bb6d99b1274 [2021, September 10].

Cortez, P. & Morais, A. 2007. A data mining approach to predict forest fires using meteorological data. *Proceedings of 13th Portugese Conference on Artificial Intelligence*, 512–523 [Online]. Available: http://www.dsi.uminho.pt/~pcortez/fires.pdf.

Coxworth, B. 2020. *AeroSeeder drone designed to speed the seeding of cover crops*. [Online], Available: https://newatlas.com/drones/aeroseeder-drone-cover-crops/?utm_source=NewAtlasSubscribers&utm_campaign=9b88dd6ba8-EMAIL_CAMPAIGN_2020_08_14_01_58&utm_medium=email&utm_term=0_65b67362bd-9b88dd6ba8-92937789 [2021, October 23].

CropOM. 2021. *How Precision Agriculture is Revolutionizing the Agricultural Sector*. [Online], Available: https://cropom.com/articles/how-precision-agriculture-is-revolutionizing-the-agricultural-sector.

DAFF. 2010. *Wheat Production guideline*.

DataFlair. 2021. *Pros and Cons of R Programming Language - Unveil the Essential Aspects!* [Online], Available: https://data-flair.training/blogs/pros-and-cons-of-r-programming-language/ [2021, May 14].

Deeper Insights. n.d. *How to run a Data Science Team: TDSP and CRISP Methodologies* . [Online], Available: https://deeperinsights.com/how-to-run-a-data-science-team/ [2021, October 13].

Dharmaraj, V. & Vijayanand, C. 2018. Artificial Intelligence (AI) in Agriculture. *International Journal of Current Microbiology and Applied Sciences*. 7(12):2122–2128.

Donatelli, M., Magarey, R.D., Bregaglio, S., Willocquet, L., Whish, J.P.M. & Savary, S. 2017. Modelling the impacts of pests and diseases on agricultural systems. *Agricultural Systems*. 155:213–224.

van Engelen, J.E. & Hoos, H.H. 2019. A survey on semi-supervised learning. *Machine Learning*

2019 109:2. 109(2):373–440.

EOS. 2021. *Chlorophyll Index*. [Online], Available: https://eos.com/make-an-analysis/ci/ [2021, October 25].

Eze, S., Dougill, A.J., Banwart, S.A., Sallu, S.M., Smith, H.E., Tripathi, H.G., Mgohele, R.N. & Senkoro, C.J. 2021. Farmers' indicators of soil health in the African highlands.

FIS. 2020. *Radiometric Resolution*. [Online], Available: https://www.fis.uni-bonn.de/en/recherchetools/infobox/professionals/resolution/radiometric-resolution [2021, October 23].

Frankenfield, J. 2021. *Data Analytics*. [Online], Available: https://www.investopedia.com/terms/d/data-analytics.asp [2021, October 14].

Frisvold, G.B. & Murugesan, A. 2013. Use of weather information for agricultural decision making. *Weather, Climate, and Society*. 5(1):55–69.

G2. 2021. *Best GIS Software 202*. [Online], Available: https://www.g2.com/categories/gis [2021, October 25].

Gabron, N. n.d. *LEAF AREA INDEX (LAI)*. [Online], Available: www.fao.org/gtos/tems [2021, May 01].

Ge, Y., Thomasson, J.A. & Sui, R. 2011. Remote sensing of soil properties in precision agriculture: A review. *Frontiers of Earth Science*. 5(3):229–238.

Genc, H., Genc, L., Turhan, H., Smith, S. & Nation, J. 2010. Vegetation indices as indicators of damage by the sunn pest (Hemiptera: Scutelleridae) to field grown wheat. *African Journal of Biotechnology*. 7(2):173–180.

GISGeography. 2021. *30 Best GIS Software Applications*. [Online], Available: https://gisgeography.com/best-gis-software/ [2021, October 25].

Giusti, E. & Marsili-Libelli, S. 2015. A Fuzzy Decision Support System for irrigation and water conservation in agriculture. *Environmental Modelling & Software*. 63:73–86.

Goldblatt, A. 2013. AGRICULTURE: FACTS & TRENDS South Africa 1. [Online], Available: http://awsassets.wwf.org.za/downloads/facts_brochure_mockup_04_b.pdf.

Goldstein, A., Fink, L., Meitin, A., Bohadana, S., Lutenberg, O. & Ravid, G. 2018. Applying machine learning on sensor data for irrigation recommendations: revealing the agronomist's tacit knowledge. *Precision Agriculture*. 19(3):421–444.

Grokking. 2019. Types of machine learning. in *Manning*. [Online], Available: https://livebook.manning.com/book/grokking-machine-learning/welcome/v-4/ [2021, October 17].

Hajjar, B. 2020. *Africa's population growth will triple by 2050*. [Online], Available: https://www.weforum.org/agenda/2020/01/the-children-s-continent/ [2021, October 22].

Han, E., Baethgen, W.E., Ines, V.M., Mer, F., Souza, J.S., Berterretche, M., Atunez, G. & Barreira, C. 2019. SIMAGRI: An agro-climate decision support tool. *Computers and electronics in agriculture*. 161:241–251.

Hao, S., Ryu, D., Western, A., Perry, E., Bogena, H. & Franssen, H.J.H. 2021. Performance of a wheat yield prediction model and factors influencing the performance: A review and meta-analysis. *Agricultural Systems*. 194:103278.

Hassanijalilian, O., Igathinathane, C., Doetkott, C., Bajwa, S., Nowatzki, J. & Haji Esmaeili, S.A. 2020. Chlorophyll estimation in soybean leaves infield with smartphone digital imaging and machine learning. *Computers and Electronics in Agriculture*. 174:105433.

Hatfield, J.L. & Prueger, J.H. 2010. Value of Using Different Vegetative Indices to Quantify Agricultural Crop Characteristics at Different Growth Stages under Varying Management Practices. *Remote Sensing 2010*. 2(2):562–578.

Holsapple, C.W. & Burstein, F. 2008. Decisions and Knowledge. in *Handbook on Decision Support Systems* Springer Berlin Heidelberg. 21–53.

IBM. 2018. *Watson Decision Platform for Agriculture*. [Online], Available: http://www.ibm.com/legal/us/en/copytrade.shtml [2021, October 18].

IBM Cloud Education. 2020. *What is Artificial Intelligence (AI)?* [Online], Available: https://www.ibm.com/cloud/learn/what-is-artificial-intelligence [2021, October 17].

Jarman, M.;Dimmock, J. 2018. [Online], Available: https://projectblue.blob.core.windows.net/media/Default/Imported Publication Docs/SatellitesForAgriculture1825_181217_WEB.pdf.

Jégo, G., Pattey, E. & Liu, J. 2012. Using Leaf Area Index, retrieved from optical imagery, in the STICS crop model for predicting yield and biomass of field crops. *Field Crops Research*. 131:63–74.

Juneja, P. n.d. *Limitations & Disadvantages of Decision Support Systems*. [Online], Available: https://www.managementstudyguide.com/limitations-and-disadvantages-of-decision-support-systems.htm [2021, May 11].

Kadiyala, M.D.M., Nedumaran, S., Singh, P., S., C., Irshad, M.A. & Bantilan, M.C.S. 2015. An integrated crop model and GIS decision support system for assisting agronomic decision making under climate change. *Science of The Total Environment*. 521–522:123–134.

Kennelly, M., O'Mara, J., Rivard, C., Miller, G.L. & Smith, D. 2012. Introduction to Abiotic Disorders in Plants. *The Plant Health Instructor*.

Krill, P. 2015. *Why R? The pros and cons of the R language*. [Online], Available: https://www.infoworld.com/article/2940864/r-programming-language-statistical-data-analysis.html [2021, May 14].

Krykowski, M. 2021. *Top 15 Scala Libraries for Data Science in 2021*. [Online], Available: https://scalac.io/blog/top-15-scala-libraries-for-data-science-in-2021/ [2021, October 17].

Kulbacki, M., Segen, J., Knieć, W., Klempous, R., Kluwak, K., Nikodem, J., Kulbacka, J. & Serester, A. 2018. Survey of Drones for Agriculture Automation from Planting to Harvest. *INES 2018 - IEEE 22nd International Conference on Intelligent Engineering Systems, Proceedings*. 000353–000358.

Kumar, R., Mishra, R., Gupta, H.P. & Dutta, T. 2021. Smart Sensing for Agriculture: Applications, Advancements, and Challenges. *IEEE Consumer Electronics Magazine*. 10(4):51–56.

Kumari, A., Tanwar, S., Tyagi, S., Kumar, N., Maasberg, M. & Choo, K.K.R. 2018. Multimedia big data computing and Internet of Things applications: A taxonomy and process model. *Journal of Network and Computer Applications*. 124:169–195.

Law, A.M. 2015. *Simulation Modeling and Analysis*. fifth ed. New York: McGraw-Hill Education. [Online], Available: www.averill-law.com [2021, November 08].

Liakos, K.G., Busato, P., Moshou, D., Pearson, S. & Bochtis, D. 2018. Machine learning in agriculture: A review. *Sensors (Switzerland)*. 18(8):1–29.

Loizos, C. 2017. *Farmers Business Network just raked in a whopping $110 million in Series D funding*. [Online], Available: https://techcrunch.com/2017/11/30/farmers-business-network-just-raked-in-a-whopping-110-million-in-series-d-funding/.

Looker. 2021. *Business Intelligence (BI) & Data Analytics Platform*. [Online], Available: https://looker.com/ [2021, May 26].

Lumen Learning. 2020. [Online], Available: https://courses.lumenlearning.com/geophysical/chapter/collecting-weather-data/.

Marbella International University Centre. 2020. *The V's of Big Data*. [Online], Available: https://miuc.org/vs-big-data/ [2021, October 13].

Marin, G. 2008. Decision support systems. *Journal of Information Systems & Operations Management*. 2(2):513–520. [Online], Available: ftp://ftp.repec.org/opt/ReDIF/RePEc/rau/jisomg/FA08/JISOM-FA08-A19.pdf.

Master's in data analytics. 2020. *Big Data and Artificial Intelligence: How They Work Together*. [Online], Available: https://online.maryville.edu/blog/big-data-is-too-big-without-ai/ [2021, October 18].

Mccarthy, J. 2004. WHAT IS ARTIFICIAL INTELLIGENCE? (November, 24). [Online], Available: http://www-formal.stanford.edu/jmc/ [2021, October 18].

Medina, A. 2020. *What is better for data science learning and work: Julia or python?* [Online], Available: https://www.analyticsvidhya.com/blog/2020/08/what-is-better-for-data-science-learning-and-work-julia-or-python/ [2021, May 14].

Meivel, S., Gandhiraj, N., Srinivasan, G. & Maguteeswaran, D.R. 2016. Quadcopter UAV Based Fertilizer and Pesticide Spraying System. *Journal of Engineering Sciences*. 1(1):8–12. [Online], Available: https://www.researchgate.net/publication/303453086 [2021, October 25].

MicaSense & Aerobotics. 2021. *Satellite vs. Drone Imagery in Vegetation Mapping*. [Online], Available: https://micasense.com/satellite-vs-drone-imagery-in-vegetation-mapping/ [2021, October 24].

Microsoft. n.d. *What is the Team Data Science Process?* . [Online], Available: https://docs.microsoft.com/en-us/azure/architecture/data-science-process/overview [2021, October 13].

Microsoft Research. 2021. [Online], Available: https://www.youtube.com/watch?v=LzKl4q2vLuk [2021, October 22].

Mohammed Ali Al-windi, B.K., Abbas, A.H. & Shakir Mahmood, M. 2021. Wheat stem rust leaf disease detection using image processing. in *Materials Today: Proceedings* Elsevier.

Mulla, D.J. 2013. Twenty five years of remote sensing in precision agriculture: Key advances and remaining knowledge gaps. *Biosystems Engineering*. 114(4):358–371.

NASA EO. 2000. Measuring Vegetation (NDVI & EVI). (August, 30).

Navarro-Hellín, H., Martínez-del-Rincon, J., Domingo-Miguel, R., Soto-Valles, F. & Torres-Sánchez, R. 2016. A decision support system for managing irrigation in agriculture. *Computers and*

Electronics in Agriculture. 124:121–131.

Niloofar, P., Francis, D.P., Lazarova-Molnar, S., Vulpe, A., Vochin, M.C., Suciu, G., Balanescu, M., Anestis, V., et al. 2021. Data-driven decision support in livestock farming for improved animal health, welfare and greenhouse gas emissions: Overview and challenges. *Computers and Electronics in Agriculture*. 190:106406.

Nine Boards. 2020. *Tableau vs. Power BI vs. Looker - Which tool is better for your Business?* . [Online], Available: https://nineboards.com/tableau-vs-power-bi-vs-looker-which-tool-is-better-for-your-business/ [2021, May 26].

Nisbet, R., Elder, J. & Miner, G.D. 2009. *Handbook of Statistical Analysis and Data Mining Applications - Robert Nisbet, John Elder, Gary D. Miner - Google Books*. [Online], Available: https://books.google.co.za/books?hl=en&lr=&id=U5np34a5fmQC&oi=fnd&pg=PP1&dq=nisbet+et+al+2009&ots=Ss0ZETClDO&sig=2USZWDFZestOie9JckjHz3H9K6Q#v=onepage&q=nisbet et al 2009&f=false [2021, October 13].

Nowatzki, J., Andres, R. & Kyllo, K. 2017. *Agricultural Remote Sensing Basics*. [Online], Available: https://www.ag.ndsu.edu/publications/crops/agricultural-remote-sensing-basics [2021, October 23].

Nuno, B.S.-A. 2014. *Visualizing NDVI for Agriculture*. [Online], Available: https://blog.mapbox.com/visualizing-ndvi-for-agriculture-ad35d7c5f27e [2021, April 30].

Oracle. 2021. *What Is Big Data?* [Online], Available: https://www.oracle.com/za/big-data/what-is-big-data/ [2021, October 13].

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.

Peng, B., Guan, K., Zhou, W., Jiang, C., Frankenberg, C., Sun, Y., He, L. & Köhler, P. 2020. Assessing the benefit of satellite-based Solar-Induced Chlorophyll Fluorescence in crop yield prediction. *International Journal of Applied Earth Observation and Geoinformation*. 90:102126.

Pierpaoli, E., Carli, G., Pignatti, E. & Canavari, M. 2013. Drivers of Precision Agriculture Technologies Adoption: A Literature Review. *Procedia Technology*. 8:61–69.

Pinguet, B. 2021. *The Role of Drone Technology in Sustainable Agriculture* . [Online], Available: https://www.precisionag.com/in-field-technologies/drones-uavs/the-role-of-drone-technology-in-sustainable-agriculture/?e=info@deltahedron.co.uk&utm_source=omail&utm_medium=newsletter&utm_ca

mpaign=pagnews05282021 [2021, October 23].

Power, D.J. 2002. *Decision Support Systems: Concepts and Resources for Managers*.

PrecisionAg Alliance. 2020. *Six Levels of Precision Agriculture Adoption Identified*. [Online], Available: https://www.precisionagalliance.com/digital-farming/six-levels-of-precision-agriculture-adoption-identified-by-the-precisionag-institute/ [2021, October 22].

Du Preez, A. 2020. A decision support framework for machine learning applications. Stellenbosch : Stellenbosch University. [Online], Available: https://scholar.sun.ac.za:443/handle/10019.1/109222 [2021, October 14].

Pykes, K. *Semi-Supervised Machine Learning Explained*. [Online], Available: https://towardsdatascience.com/semi-supervised-machine-learning-explained-c1a6e1e934c7 [2021, October 17].

Qi, J., Chehbouni, A., Huete, A.R., Kerr, Y.H. & Sorooshian, S. 1994. *A Modified Soil Adjusted Vegetation Index*.

Reddy, P.C.S. & Sureshbabu, A. 2019. An adaptive model for forecasting seasonal rainfall using predictive analytics. *International Journal of Intelligent Engineering and Systems*. 12(5):22–32.

Rehman, A. 2015. *Smart Agriculture: An Approach Towards Better Agriculture Management*.

Van Rooyen, J.C. 2019. Digital shop floor monitoring for the Stellenbosch learning factory. Stellenbosch University.

Rossi, V., Caffi, T. & Salinari, F. 2012. Helping farmers face the increasing complexity of decision-making for crop protection. *Phytopathologia Mediterranea*. 51(3):457–479.

SAS. 2021. *Big Data: What it is and why it matters*. [Online], Available: https://www.sas.com/en_za/insights/big-data/what-is-big-data.html [2021, October 13].

Schaap, P. 2020. *4 Levels Of Data Maturity Every Manager Must Know*. [Online], Available: https://computd.nl/demystification/4-levels-of-data-maturity/ [2021, October 14].

Shafique, U. & Qaiser, H. 2014. A Comparative Study of Data Mining Process Models (KDD, CRISP-DM and SEMMA). *International Journal of Innovation and Scientific Research*. 12(1):217–222. [Online], Available: http://www.ijisr.issr-journals.org/ [2021, October 13].

Sheng Tey, Y. & Brindal, M. 2012. Factors influencing the adoption of precision agricultural technologies: a review for policy implications. *Precision Agric*. 13(6):713–730.

Singh, A. 2018. *Introduction to Reinforcement Learning*. [Online], Available: https://www.datacamp.com/community/tutorials/introduction-reinforcement-learning?utm_source=adwords_ppc&utm_campaignid=1455363063&utm_adgroupid=65083631748&utm_device=c&utm_keyword=&utm_matchtype=b&utm_network=g&utm_adpostion=&utm_creative=332602034358&ut [2021, October 17].

Smart vision. 2021. *Crisp DM methodology*. [Online], Available: https://www.sv-europe.com/crisp-dm-methodology/ [2021, October 14].

Al Sonosy, O., Rady, S., Badr, N.L. & Hashem, M. 2016. Machine learning techniques for mining location-based social networks for business predictions. in *ACM International Conference Proceeding Series* Vols 09-11-May-2016. Association for Computing Machinery. 185–190.

South African government. 2021. *Wheat Pest Management, South Africa*. [Online], Available: https://southafrica.co.za/wheat-pest-management.html [2021, May 21].

SPEC INDIA. 2021. *Top 10 Best Programming Languages For Machine Learning In 2021*. [Online], Available: https://www.spec-india.com/blog/programming-languages-for-machine-learning [2021, October 17].

SPIME Analytics. 2020. *Reinforcement Learning: Introduction*. [Online], Available: https://www.youtube.com/watch?v=4bgIgJZnQWs [2021, October 17].

Springboard. 2020. *Best language for Machine Learning: Which Programming Language to Learn*. [Online], Available: https://in.springboard.com/blog/best-language-for-machine-learning/ [2021, May 14].

stars project. n.d. *Multispectral and panchromatic images* . [Online], Available: https://www.stars-project.org/en/knowledgeportal/magazine/remote-sensing-technology/introduction/multispectral-and-panchromatic-images/ [2021, April 30].

Sunday Olukayode, O., Olamidotun Blesing, L., Dauda Rotimi, A. & Ayodeji Oguntola, E. 2018. Assessment of plant health status using remote sensing and GIS techniques. *Advances in Plants & Agriculture Research*. Volume 8(Issue 6).

Tableau. 2003. *Business intelligence and analytics software*. [Online], Available: https://www.tableau.com/en-gb?_ga=2.125131980.711503524.1622015806-1380853709.1622015806&_gl=1*ur2z3d*_ga*MTM4MDg1MzcwOS4xNjIyMDE1ODA2*_ga_8YLN0SNXVS*MTYyMjAxODMzNC4yLjAuMTYyMjAxODMzNC4w&_fsi=uKiWHjbt [2021, May 26].

Tetila, E.C., Machado, B.B., Astolfi, G., Belete, N.A. de S., Amorim, W.P., Roel, A.R. & Pistori, H.

2020. Detection and classification of soybean pests using deep learning with UAV images. *Computers and Electronics in Agriculture*. 179:105836.

The landscape toolbox. 2012. *Modified Soil-adjusted Vegetation Index*. [Online], Available: https://wiki.landscapetoolbox.org/doku.php/remote_sensing_methods:modified_soil-adjusted_vegetation_index [2021, April 23].

Tyrychtr, J. & Vostrovsky, V. 2017. The current state of the issue of information needs and dispositions among small Czech farms. *Agricultural Economics (Czech Republic)*. 63(4):164–174.

United Nations. 2019. *Population*. [Online], Available: https://www.un.org/en/global-issues/population [2021, October 22].

University of Stellenbosch. 2013. *Department of Plant Pathology*. [Online], Available: http://www.sun.ac.za/english/faculty/agri/departments1/plantpathology [2021, May 20].

Verma, V. 2020. *A comprehensive guide to feature selection using wrapper methods in Python*. [Online], Available: https://www.analyticsvidhya.com/blog/2020/10/a-comprehensive-guide-to-feature-selection-using-wrapper-methods-in-python/ [2021, September 09].

Verma, S., Bhatia, A., Chug, A. & Singh, A.P. 2020. Recent Advancements in Multimedia Big Data Computing for IoT Applications in Precision Agriculture: Opportunities, Issues, and Challenges. *Intelligent Systems Reference Library*. 163:391–416.

Viana, C., Farhate, V., de Souza, Z.M., Cherubin, M.R. & Carneiro, M.P. 2020. Abiotic Soil Health Indicators that Respond to Sustainable Management Practices in Sugarcane Cultivation. 1–19.

Wildlife Drones. 2020. *4 ways drones can help you on the farm*. [Online], Available: https://wildlifedrones.net/4-ways-drones-can-help-you-on-the-farm/ [2021, October 25].

Wirth, R. & Hipp, J. 2000. CRISP-DM: Towards a Standard Process Model for Data Mining. *Computer Science*.

Wolfert, S., Ge, L., Verdouw, C. & Bogaardt, M.J. 2017. Big Data in Smart Farming – A review. *Agricultural Systems*. 153:69–80.

Xu, X., Nie, C., Jin, X., Li, Z., Zhu, H., Xu, H., Wang, J., Zhao, Y., et al. 2021. A comprehensive yield evaluation indicator based on an improved fuzzy comprehensive evaluation method and hyperspectral data. *Field Crops Research*. 270:108204.

Yang, Z., Rao, M.N., Elliott, N.C., Kindler, S.D. & Popham, T.W. 2009. Differentiating stress induced by greenbugs and Russian wheat aphids in wheat using remote sensing. *Computers and*

Electronics in Agriculture. 67(1–2):64–70.

Yinka-Banjo, C. & Ajayi, O. 2019. Sky-Farmers: Applications of Unmanned Aerial Vehicles (UAV) in Agriculture. in *Autonomous Vehicles* G. Dekoulis (ed.). IntechOpen G. Dekoulis (ed.).

Zarco-Tejada, P.J., Rueda, C.A. & Ustin, S.L. 2003. Water content estimation in vegetation with MODIS reflectance data and model inversion methods. *Remote Sensing of Environment*. 85(1):109–124.

Zhai, Z., Martínez, J.F., Beltran, V. & Martínez, N.L. 2020. Decision support systems for agriculture 4.0: Survey and challenges. *Computers and Electronics in Agriculture*. 170.

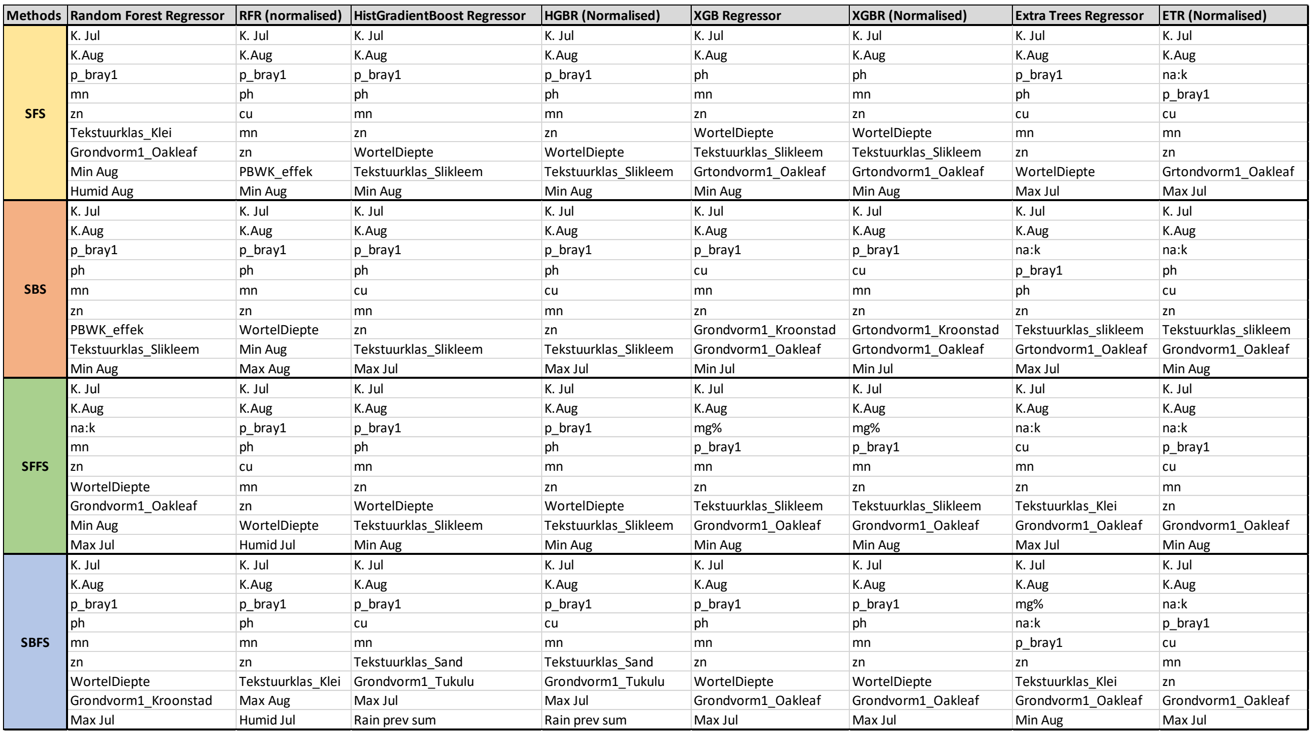
Zhu, L., Suomalainen, J., Liu, J., Hyyppä, J., Kaartinen, H. & Haggren, H. 2017. A Review: Remote Sensing Sensors. in *Multi-purposeful Application of Geospatial Data* InTech.

Appendix A

Table A 1: Satellites used in agriculture

Satellite	Launch Year	Sensors	Height of orbit	Swath (km)	Revisit (days)	Channels	Spatial resolution
Landsat	1972, 1975, 1978, 1992, 1984, 1993, 1999, 2013, 2020	Panchromatic and multispectral sensor	705	185, 183	16	7–11	120 m, 100 m, 60 m, 30 m, 15 m
Spot	1986, 1990, 1993, 1998, 2002, 2012	Imaging spectroradiometer	694	60	1-3	Panchromatic, B, G, R, NIR	2.5 m, 5 m, 10 m, 20 m
ERS	1991, 1995	IR radiometer, microwave sounder, Radiometer, SAR	782–785	5–100 km (AMI) - 500 km (ATSR)	3, 35, 336	SAR	26 m across track and 6–30 m along track
RADARSAT	1995, 2007, 2018	SAR	793–821, 798, 592.7	45–100, 18–500, 5–500	1	SAR	8–100 m, 3–100 m, 3–100 m
MODIS	1999, 2002	Imaging spectroradiometer	705		1	36	1000 m, 500 m, 250 m
IKONOS	1999	Imaging spectroradiometer	681		3	Panchromatic, B, G, R, NIR	Panchromatic:80 cm B, G, R, NIR:3.2 m
QuickBird	2000, 2001	Imaging spectroradiometer	482, 450		2.4–5.9	Panchromatic, B, G, R, NIR	Panchromatic:65 cm/61 cm B, G, R, NIR:2.62 m/2.44 m
Envisat	2002	ASAR, MERIS, AATSR, RA-2, MWR, GOMOS, MIPAS, SCIAMACHY, DORIS, LRR	790		35	15 bands (VIS, NIR), C-band	300 m, 30–150 m
GeoEye	2008	Imaging spectroradiometer	681		8.3	Panchromatic, B, G, R, NIR	Panchromatic:41 cm B, G, R, NIR: 1.65 m

Satellite	Launch Year	Sensors	Height of orbit	Swath (km)	Revisit (days)	Channels	Spatial resolution
WorldView 1-3	2007, 2009, 2014, 2016	Imaging spectroradiometer, Laser altimeter	496, 770, 617, 681	17.6 km 16.4 km 13.1 km 14.5 km	1.7 1.1 <1 3	Panchromatic; Panchromatic and eight multispectral. Panchromatic and eight multispectral. Panchromatic, B, G, R, NIR	Panchromatic 0.5 m. Panchromatic and stereo images: 0.46 m multispectral: 1.84 m. Panchromatic 0.34 m and multispectral 1.36 m
Sentinel 1-6	2014, 2015, 2016, 2017, 2021	Radar and super-spectral imaging	693, 786, 814	250 km 290 km, 250 km	12, 10, 27	C-SAR, 12 bands (VIS, NIR, SWIR), 21 bands (VIS, NIR), S-band & X-band	5–20 m, 5–40 m, 10 m & 20 m & 60 m



Methods	Random Forest Regressor	RFR (normalised)	HistGradientBoost Regressor	HGBR (Normalised)	XGB Regressor	XGBR (Normalised)	Extra Trees Regressor	ETR (Normalised)
SFS	K. Jul	K. Jul	K. Jul	K. Jul	K. Jul	K. Jul	K. Jul	K. Jul
	K.Aug	K.Aug	K.Aug	K.Aug	K.Aug	K.Aug	K.Aug	K.Aug
	p_bray1	p_bray1	p_bray1	p_bray1	ph	ph	p_bray1	na:k
	mn	ph	ph	ph	mn	mn	ph	p_bray1
	zn	cu	mn	mn	zn	zn	cu	cu
	Tekstuurklas_Klei	mn	zn	zn	WortelDiepte	WortelDiepte	mn	mn
	Grondvorm1_Oakleaf	zn	WortelDiepte	WortelDiepte	Tekstuurklas_Slikleem	Tekstuurklas_Slikleem	zn	zn
	Min Aug	PBWK_effek	Tekstuurklas_Slikleem	Tekstuurklas_Slikleem	Grtondvorm1_Oakleaf	Grtondvorm1_Oakleaf	WortelDiepte	Grtondvorm1_Oakleaf
	Humid Aug	Min Aug	Min Aug	Min Aug	Min Aug	Min Aug	Max Jul	Max Jul
SBS	K. Jul	K. Jul	K. Jul	K. Jul	K. Jul	K. Jul	K. Jul	K. Jul
	K.Aug	K.Aug	K.Aug	K.Aug	K.Aug	K.Aug	K.Aug	K.Aug
	p_bray1	p_bray1	p_bray1	p_bray1	p_bray1	p_bray1	na:k	na:k
	ph	ph	ph	ph	cu	cu	p_bray1	ph
	mn	mn	cu	cu	mn	mn	ph	cu
	zn	zn	mn	mn	zn	zn	zn	zn
	PBWK_effek	WortelDiepte	zn	zn	Grondvorm1_Kroonstad	Grtondvorm1_Kroonstad	Tekstuurklas_slikleem	Tekstuurklas_slikleem
	Tekstuurklas_Slikleem	Min Aug	Tekstuurklas_Slikleem	Tekstuurklas_Slikleem	Grondvorm1_Oakleaf	Grtondvorm1_Oakleaf	Grtondvorm1_Oakleaf	Grondvorm1_Oakleaf
	Min Aug	Max Aug	Max Jul	Max Jul	Min Jul	Min Jul	Max Jul	Min Aug
SFFS	K. Jul	K. Jul	K. Jul	K. Jul	K. Jul	K. Jul	K. Jul	K. Jul
	K.Aug	K.Aug	K.Aug	K.Aug	K.Aug	K.Aug	K.Aug	K.Aug
	na:k	p_bray1	p_bray1	p_bray1	mg%	mg%	na:k	na:k
	mn	ph	ph	ph	p_bray1	p_bray1	cu	p_bray1
	zn	cu	mn	mn	mn	mn	mn	cu
	WortelDiepte	mn	zn	zn	zn	zn	zn	mn
	Grondvorm1_Oakleaf	zn	WortelDiepte	WortelDiepte	Tekstuurklas_Slikleem	Tekstuurklas_Slikleem	Tekstuurklas_Klei	zn
	Min Aug	WortelDiepte	Tekstuurklas_Slikleem	Tekstuurklas_Slikleem	Grondvorm1_Oakleaf	Grondvorm1_Oakleaf	Grondvorm1_Oakleaf	Grondvorm1_Oakleaf
	Max Jul	Humid Jul	Min Aug	Min Aug	Min Aug	Min Aug	Max Jul	Min Aug
SBFS	K. Jul	K. Jul	K. Jul	K. Jul	K. Jul	K. Jul	K. Jul	K. Jul
	K.Aug	K.Aug	K.Aug	K.Aug	K.Aug	K.Aug	K.Aug	K.Aug
	p_bray1	p_bray1	p_bray1	p_bray1	p_bray1	p_bray1	mg%	na:k
	ph	ph	cu	cu	ph	ph	na:k	p_bray1
	mn	mn	mn	mn	mn	mn	p_bray1	cu
	zn	zn	Tekstuurklas_Sand	Tekstuurklas_Sand	zn	zn	zn	mn
	WortelDiepte	Tekstuurklas_Klei	Grondvorm1_Tukulu	Grondvorm1_Tukulu	WortelDiepte	WortelDiepte	Tekstuurklas_Klei	zn
	Grondvorm1_Kroonstad	Max Aug	Max Jul	Max Jul	Grondvorm1_Oakleaf	Grondvorm1_Oakleaf	Grondvorm1_Oakleaf	Grondvorm1_Oakleaf
	Max Jul	Humid Jul	Rain prev sum	Rain prev sum	Max Jul	Max Jul	Min Aug	Max Jul

Figure A 1: Summary of features selected from the various wrapper selection methods

Appendix B – Python Code

```
## IMPORTS
from os import stat
from matplotlib.colors import Normalize
from numpy.core.function_base import linspace
import pandas as pd
import numpy as np
from pandas.core.reshape.concat import concat
from scipy.stats import kurtosis, skew
from scipy.interpolate import interp1d
import statistics
import matplotlib as mpl
import matplotlib.pyplot as plt
from scipy.stats.stats import hmean, mode
import seaborn as sns
from sklearn import feature_selection
from sklearn.metrics import roc_auc_score, r2_score,mean_squared_error

from statsmodels.tsa.seasonal import seasonal_decompose   #decompose time-series data CHL
from sklearn.utils._testing import all_estimators
from sklearn.linear_model import LinearRegression
from sklearn.preprocessing import StandardScaler,MinMaxScaler,RobustScaler,Normalizer
#before running algorithms and feature selection

#LAZY PREDICT
import lazypredict
from lazypredict.Supervised import LazyRegressor    #from lazypredict.Supervised import LazyClassifier
from sklearn.model_selection import train_test_split

# plot feature importance manually
import xgboost as xgb
from xgboost import plot_importance
from sklearn.metrics import accuracy_score

#Decision Tree
from sklearn.tree import DecisionTreeRegressor

#Extra Trees
from sklearn.ensemble import ExtraTreesRegressor

#Random Forest
#from sklearn.ensemble import RandomForestClassifier
from sklearn.ensemble import RandomForestRegressor

#Boruta
#import xgboost as xgb
from boruta import BorutaPy
```

```
#permutation with sklearn
from sklearn.model_selection import train_test_split
from sklearn.inspection import permutation_importance
#from sklearn.ensemble import RandomForestClassifier

#RFE
from numpy import floating, mean
#from numpy import st
from sklearn.model_selection import cross_val_score
from sklearn.model_selection import RepeatedStratifiedKFold
from sklearn.tree import DecisionTreeClassifier
from sklearn.feature_selection import RFE
from sklearn.pipeline import Pipeline

#Exhaustive feature selection
from mlxtend.feature_selection import ExhaustiveFeatureSelector,SequentialFeatureSelector

#HistGradBoostReg
from sklearn.ensemble import HistGradientBoostingRegressor
from xgboost.sklearn import XGBRegressor
from sklearn.metrics import mean_absolute_error

##DATASETS FROM EXCEL AND QGIS
rawdata = pd.read_excel(r'C:\Users\jclau\Documents\Documents\AMasters-2020\Masters
Project\Data\Excel\07_21_FinalTable_Python.xlsx',sheet_name='Final')
dfdailyMin = pd.read_excel(r'C:\Users\jclau\Documents\Documents\AMasters-2020\Masters
Project\Data\Weather data\Daily Warmbad_Reworked_Final.xlsx',sheet_name='DailyMin')
dfdailyMin = dfdailyMin.round(2)
dfdailyMax = pd.read_excel(r'C:\Users\jclau\Documents\Documents\AMasters-2020\Masters
Project\Data\Weather data\Daily Warmbad_Reworked_Final.xlsx',sheet_name='DailyMax')
dfdailyMax = dfdailyMax.round(2)
dfdailyRf = pd.read_excel(r'C:\Users\jclau\Documents\Documents\AMasters-2020\Masters
Project\Data\Weather data\Daily Warmbad_Reworked_Final.xlsx',sheet_name='DailyRainfall')
dfdailyRf = dfdailyRf.round(2)
dfhumidity = pd.read_excel(r'C:\Users\jclau\Documents\Documents\AMasters-2020\Masters
Project\Data\Weather data\Daily Warmbad_Reworked_Final.xlsx',sheet_name='Humidity')
dfPressure = pd.read_excel(r'C:\Users\jclau\Documents\Documents\AMasters-2020\Masters
Project\Data\Weather data\Daily Warmbad_Reworked_Final.xlsx',sheet_name='Pressure')
dfWindSpeed = pd.read_excel(r'C:\Users\jclau\Documents\Documents\AMasters-2020\Masters
Project\Data\Weather data\Daily
Warmbad_Reworked_Final.xlsx',sheet_name='WindSpeed',converters={'Date': str})
dfdailyRainfall = dfdailyRf.fillna(0)
#dfWindSpeed.columns = dfWindSpeed.columns.datetime.strptime()
nutrients = rawdata.loc[:,'ca':'Grondvorm1']
#Continious features for data quality report
nutrients1 = rawdata.loc[:,'ca':'PBWK_effek']
nutrients_con = nutrients1.drop(["Tekstuurklas"],axis=1)
nutrients_con_Desc = nutrients_con.describe().T
#Categorical features for data quality report
```

```
nutrients_cat = rawdata.loc[:,'Tekstuurklas':'Grondvorm1']
nutrients_cat = nutrients_cat.drop(['PBWK_effek'],axis=1)
#
mainNutrients = pd.concat([nutrients.loc[:,['ca','mg%','na:k','p_bray1','ph','cu','mn','zn','WortelDiepte','PBWK_effek']]], axis=1)
mainNutrients = mainNutrients.round(2)
dummies = pd.get_dummies(nutrients[['Tekstuurklas','Grondvorm1']])
#dummies = pd.get_dummies(nutrients[['Tekstuurklas','Dreinering','Risiko_vir','Besproei_1','Grondvorm1']])
dataDummies = pd.concat([mainNutrients,dummies],axis=1)
df = pd.DataFrame()
df1 = pd.DataFrame()


#\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\
######\\\\\\\\\\\\\\\ ISOLATE MONTHS FOR MEAN - KORING \\\\\\\\\\\\\\\#####
#\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\

#####\\\\\2017\\\\\\#####
koring17Jun = rawdata.loc[:,'24/06/2017']
koring17Jul = rawdata.loc[:,'04/07/2017':'29/07/2017']
koring17Aug = rawdata.loc[:,'08/08/2017':'28/08/2017']
koring17Sep = rawdata.loc[:,'02/09/2017':'22/09/2017']
koring17Oct = rawdata.loc[:,'02/10/2017']


koring17JunMean = pd.Series(koring17Jun,name="K. Jun")      #only 1 column, thus no mean calculation
koring17JunMean.fillna(koring17JunMean.mean(),inplace=True)
koring17JulMean = pd.Series(koring17Jul.mean(axis=1),name="K. Jul")
koring17JulMean.fillna(koring17JulMean.mean(),inplace=True)
koring17AugMean = pd.Series(koring17Aug.mean(axis=1),name="K. Aug")
koring17AugMean.fillna(koring17AugMean.mean(),inplace=True)
koring17SepMean = pd.Series(koring17Sep.mean(axis=1),name="K. Sep")
koring17SepMean.fillna(koring17SepMean.mean(),inplace=True)
koring17OctMean = pd.Series(koring17Oct,name="K. Oct")
koring17OctMean.fillna(koring17OctMean.mean(),inplace=True)

koring17PerMonthChl = pd.concat([koring17JunMean,koring17JulMean,koring17AugMean,koring17SepMean,koring17OctMean],axis=1)

#####\\\\\\\\ 2018\\\\\\\\\\#####
koring18Jun = rawdata.loc[:,'24/06/2018':'29/06/2018']
koring18Jul = rawdata.loc[:,'04/07/2018':'29/07/2018']
koring18Aug = rawdata.loc[:,'08/08/2018':'28/08/2018']
koring18Sep = rawdata.loc[:,'02/09/2018':'27/09/2018']
koring18Oct = rawdata.loc[:,'02/10/2018':'07/10/2018']
```

```
koring18JunMean = pd.Series(koring18Jun.mean(axis=1),name="K. Jun")
koring18JunMean.fillna(koring18JunMean.mean(),inplace=True)
koring18JulMean = pd.Series(koring18Jul.mean(axis=1),name="K. Jul")
koring18JulMean.fillna(koring18JulMean.mean(),inplace=True)
koring18AugMean = pd.Series(koring18Aug.mean(axis=1),name="K. Aug")
koring18AugMean.fillna(koring18AugMean.mean(),inplace=True)
koring18SepMean = pd.Series(koring18Sep.mean(axis=1),name="K. Sep")
koring18SepMean.fillna(koring18SepMean.mean(),inplace=True)
koring18OctMean = pd.Series(koring18Oct.mean(axis=1),name="K. Oct")

koring18_MonthChl =
pd.concat([koring18JunMean,koring18JulMean,koring18AugMean,koring18SepMean,koring18Oct
Mean],axis=1)

#####\\\\\\\\ 2019 \\\\\\\\\\#####
Etc.…


#\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\
######################### WEATHER DATA ########################
#\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\

df['meanMin'] = round((dfdailyMin.iloc[:,1:]).mean(),2)      # Mean from 2016-2020
df['meanMax'] = round((dfdailyMax.iloc[:,1:]).mean(),2)       # Mean from 2016-2020
df['meanRainfall'] = round(dfdailyRainfall.mean(),2)
df['meanHumidity'] = round((dfhumidity.iloc[:,1:]).mean(),2)
df['meanWindSpeed'] = round((dfhumidity.iloc[:,1:]).mean(),2)
koringSeason =
pd.Series(['Dec0','Jan','Feb','Mar','Apr','May','Jun','Jul','Aug','Sep','Oct'],name="Koring Season")
#df['columndates'] = df.index.to_series(index=None)
soyaSeason = pd.Series(['Oct0','Nov','Dec','Jan','Feb','Mar'],name="Soya Season")


#\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\
#######\\\\\\\\\\ MIN PER YEAR \\\\\\\\\\\\#########
#\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\

#####\\\\\\\\ 2016 \\\\\\\\\\#####
t0 = dfdailyMin.loc[:,"December 2015":"October 2016"]
tMean = t0.mean()
t0.columns = range(t0.shape[1])
meanMin16 = pd.Series(t0.mean(),name="meanMin16")
meanMin16Jan = dfdailyMin.loc[:,'January 2016'].mean()
meanMin16Feb = dfdailyMin.loc[:,'February 2016'].mean()
meanMin16Mar = dfdailyMin.loc[:,'March 2016'].mean()
meanMin16Apr = dfdailyMin.loc[:,'April 2016'].mean()
meanMin16May = dfdailyMin.loc[:,'May 2016'].mean()
meanMin16Jun = dfdailyMin.loc[:,'June 2016'].mean()
meanMin16Jul = dfdailyMin.loc[:,'July 2016'].mean()
meanMin16Aug = dfdailyMin.loc[:,'August 2016'].mean()
```

```
meanMin16Sep = dfdailyMin.loc[:,'September 2016'].mean()
meanMin16Oct = dfdailyMin.loc[:,'October 2016'].mean()
meanMin16Nov = dfdailyMin.loc[:,'November 2016'].mean()
meanMin16Dec = dfdailyMin.loc[:,'December 2016'].mean()

#####\\\\\\\\ 2017 \\\\\\\\\\#####
t = dfdailyMin.loc[:,"December 2016":"October 2017"]
tMean = t.mean()
t.columns = range(t.shape[1])
meanMin17 = pd.Series(t.mean(),name="meanMin17")
meanMin17Jan = dfdailyMin.loc[:,'January 2017'].mean()
meanMin17Feb = dfdailyMin.loc[:,'February 2017'].mean()
meanMin17Mar = dfdailyMin.loc[:,'March 2017'].mean()
meanMin17Apr = dfdailyMin.loc[:,'April 2017'].mean()
meanMin17May = dfdailyMin.loc[:,'May 2017'].mean()
meanMin17Jun = dfdailyMin.loc[:,'June 2017'].mean()
meanMin17Jul = dfdailyMin.loc[:,'July 2017'].mean()
meanMin17Aug = dfdailyMin.loc[:,'August 2017'].mean()
meanMin17Sep = dfdailyMin.loc[:,'September 2017'].mean()
meanMin17Oct = dfdailyMin.loc[:,'October 2017'].mean()
meanMin17Nov = dfdailyMin.loc[:,'November 2017'].mean()
meanMin17Dec = dfdailyMin.loc[:,'December 2017'].mean()

#####\\\\\\\\ 2018 \\\\\\\\\\#####
t1 = dfdailyMin.loc[:,"December 2017":"October 2018"]
t1.columns = range(t1.shape[1])
meanMin18 = pd.Series(t1.mean(),name="meanMin18")
meanMin18Jan = dfdailyMin.loc[:,'January 2018'].mean()
meanMin18Feb = dfdailyMin.loc[:,'February 2018'].mean()
meanMin18Mar = dfdailyMin.loc[:,'March 2018'].mean()
meanMin18Apr = dfdailyMin.loc[:,'April 2018'].mean()
meanMin18May = dfdailyMin.loc[:,'May 2018'].mean()
meanMin18Jun = dfdailyMin.loc[:,'June 2018'].mean()
meanMin18Jul = dfdailyMin.loc[:,'July 2018'].mean()
meanMin18Aug = dfdailyMin.loc[:,'August 2018'].mean()
meanMin18Sep = dfdailyMin.loc[:,'September 2018'].mean()
meanMin18Oct = dfdailyMin.loc[:,'October 2018'].mean()
meanMin18Nov = dfdailyMin.loc[:,'November 2018'].mean()
meanMin18Dec = dfdailyMin.loc[:,'December 2018'].mean()

#####\\\\\\\\ 2019 \\\\\\\\\\#####
t2 = dfdailyMin.loc[:,"December 2018":"October 2019"]
t2.columns = range(t2.shape[1])
meanMin19 = pd.Series(t2.mean(),name="meanMin19")
meanMin19Jan = dfdailyMin.loc[:,'January 2019'].mean()
meanMin19Feb = dfdailyMin.loc[:,'February 2019'].mean()
meanMin19Mar = dfdailyMin.loc[:,'March 2019'].mean()
meanMin19Apr = dfdailyMin.loc[:,'April 2019'].mean()
meanMin19May = dfdailyMin.loc[:,'May 2019'].mean()
meanMin19Jun = dfdailyMin.loc[:,'June 2019'].mean()
```

```
meanMin19Jul = dfdailyMin.loc[:,'July 2019'].mean()
meanMin19Aug = dfdailyMin.loc[:,'August 2019'].mean()
meanMin19Sep = dfdailyMin.loc[:,'September 2019'].mean()
meanMin19Oct = dfdailyMin.loc[:,'October 2019'].mean()
meanMin19Nov = dfdailyMin.loc[:,'November 2019'].mean()
meanMin19Dec = dfdailyMin.loc[:,'December 2019'].mean()

######\\\\\\\\ 2020 \\\\\\\\\\######
t3 = dfdailyMin.loc[:,"December 2019":"October 2020"]
t3.columns = range(t3.shape[1])
meanMin20 = pd.Series(t3.mean(),name="meanMin20")
meanMin20Jan = dfdailyMin.loc[:,'January 2020'].mean()
meanMin20Feb = dfdailyMin.loc[:,'February 2020'].mean()
meanMin20Mar = dfdailyMin.loc[:,'March 2020'].mean()
meanMin20Apr = dfdailyMin.loc[:,'April 2020'].mean()
meanMin20May = dfdailyMin.loc[:,'May 2020'].mean()
meanMin20Jun = dfdailyMin.loc[:,'June 2020'].mean()
meanMin20Jul = dfdailyMin.loc[:,'July 2020'].mean()
meanMin20Aug = dfdailyMin.loc[:,'August 2020'].mean()
meanMin20Sep = dfdailyMin.loc[:,'September 2020'].mean()
meanMin20Oct = dfdailyMin.loc[:,'October 2020'].mean()
meanMin20Nov = dfdailyMin.loc[:,'November 2020'].mean()
meanMin20Dec = dfdailyMin.loc[:,'December 2020'].mean()

#\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\
########\\\\\\\\\\ MAX PER YEAR \\\\\\\\\\\\###########
#\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\

######\\\\\\\\ 2016 \\\\\\\\\\######
r0 = dfdailyMax.loc[:,"December 2015":"October 2016"]
r0Mean = r0.mean()
r0.columns = range(r0.shape[1])
meanMax16 = pd.Series(r0.mean(),name="meanMax16")
meanMax16Jan = dfdailyMin.loc[:,'January 2016'].mean()
meanMax16Feb = dfdailyMin.loc[:,'February 2016'].mean()
meanMax16Mar = dfdailyMin.loc[:,'March 2016'].mean()
meanMax16Apr = dfdailyMin.loc[:,'April 2016'].mean()
meanMax16May = dfdailyMax.loc[:,'May 2016'].mean()
meanMax16Jun = dfdailyMax.loc[:,'June 2016'].mean()
meanMax16Jul = dfdailyMax.loc[:,'July 2016'].mean()
meanMax16Aug = dfdailyMax.loc[:,'August 2016'].mean()
meanMax16Sep = dfdailyMin.loc[:,'September 2016'].mean()
meanMax16Oct = dfdailyMin.loc[:,'October 2016'].mean()
meanMax16Nov = dfdailyMin.loc[:,'November 2016'].mean()
meanMax16Dec = dfdailyMin.loc[:,'December 2016'].mean()

######\\\\\\\\ 2017 \\\\\\\\\\######
r = dfdailyMax.loc[:,"December 2016":"October 2017"]     #t3 = dfdailyMax.loc[:,"October
2016":"October 2017"]
rMean = r.mean()
```

```
r.columns = range(r.shape[1])
meanMax17 = pd.Series(r.mean(),name="meanMax17")
meanMax17Jan = dfdailyMin.loc[:,'January 2017'].mean()
meanMax17Feb = dfdailyMin.loc[:,'February 2017'].mean()
meanMax17Mar = dfdailyMin.loc[:,'March 2017'].mean()
meanMax17Apr = dfdailyMin.loc[:,'April 2017'].mean()
meanMax17May = dfdailyMax.loc[:,'May 2017'].mean()
meanMax17Jun = dfdailyMax.loc[:,'June 2017'].mean()
meanMax17Jul = dfdailyMax.loc[:,'July 2017'].mean()
meanMax17Aug = dfdailyMax.loc[:,'August 2017'].mean()
meanMax17Sep = dfdailyMin.loc[:,'September 2017'].mean()
meanMax17Oct = dfdailyMin.loc[:,'October 2017'].mean()
meanMax17Nov = dfdailyMin.loc[:,'November 2017'].mean()
meanMax17Dec = dfdailyMin.loc[:,'December 2017'].mean()

#####\\\\\\\\ 2018 \\\\\\\\\\#####
r1 = dfdailyMax.loc[:,"December 2017":"October 2018"]
r1Mean = r1.mean()
r1.columns = range(r1.shape[1])
meanMax18 = pd.Series(r1.mean(),name="meanMax18")
meanMax18Jan = dfdailyMin.loc[:,'January 2018'].mean()
meanMax18Feb = dfdailyMin.loc[:,'February 2018'].mean()
meanMax18Mar = dfdailyMin.loc[:,'March 2018'].mean()
meanMax18Apr = dfdailyMin.loc[:,'April 2018'].mean()
meanMax18May = dfdailyMax.loc[:,'May 2018'].mean()
meanMax18Jun = dfdailyMax.loc[:,'June 2018'].mean()
meanMax18Jul = dfdailyMax.loc[:,'July 2018'].mean()
meanMax18Aug = dfdailyMax.loc[:,'August 2018'].mean()
meanMax18Sep = dfdailyMin.loc[:,'September 2018'].mean()
meanMax18Oct = dfdailyMin.loc[:,'October 2018'].mean()
meanMax18Nov = dfdailyMin.loc[:,'November 2018'].mean()
meanMax18Dec = dfdailyMin.loc[:,'December 2018'].mean()

#####\\\\\\\\ 2019 \\\\\\\\\\#####
r2 = dfdailyMax.loc[:,"December 2018":"October 2019"]
r2Mean = r2.mean()
r2.columns = range(r2.shape[1])
meanMax19 = pd.Series(r2.mean(),name="meanMax19")
meanMax19Jan = dfdailyMin.loc[:,'January 2019'].mean()
meanMax19Feb = dfdailyMin.loc[:,'February 2019'].mean()
meanMax19Mar = dfdailyMin.loc[:,'March 2019'].mean()
meanMax19Apr = dfdailyMin.loc[:,'April 2019'].mean()
meanMax19May = dfdailyMax.loc[:,'May 2019'].mean()
meanMax19Jun = dfdailyMax.loc[:,'June 2019'].mean()
meanMax19Jul = dfdailyMax.loc[:,'July 2019'].mean()
meanMax19Aug = dfdailyMax.loc[:,'August 2019'].mean()
meanMax19Sep = dfdailyMin.loc[:,'September 2019'].mean()
meanMax19Oct = dfdailyMin.loc[:,'October 2019'].mean()
meanMax19Nov = dfdailyMin.loc[:,'November 2019'].mean()
meanMax19Dec = dfdailyMin.loc[:,'December 2019'].mean()
```

```
#####\\\\\\\\ 2020 \\\\\\\\\\#####
r3 = dfdailyMax.loc[:,"December 2019":"October 2020"]
r3Mean = r3.mean()
r3.columns = range(r3.shape[1])
meanMax20 = pd.Series(r3.mean(),name="meanMax20")
meanMax20Jan = dfdailyMin.loc[:,'January 2020'].mean()
meanMax20Feb = dfdailyMin.loc[:,'February 2020'].mean()
meanMax20Mar = dfdailyMin.loc[:,'March 2020'].mean()
meanMax20Apr = dfdailyMin.loc[:,'April 2020'].mean()
meanMax20May = dfdailyMax.loc[:,'May 2020'].mean()
meanMax20Jun = dfdailyMax.loc[:,'June 2020'].mean()
meanMax20Jul = dfdailyMax.loc[:,'July 2020'].mean()
meanMax20Aug = dfdailyMax.loc[:,'August 2020'].mean()
meanMax20Sep = dfdailyMin.loc[:,'September 2020'].mean()
meanMax20Oct = dfdailyMin.loc[:,'October 2020'].mean()
meanMax20Nov = dfdailyMin.loc[:,'November 2020'].mean()
meanMax20Dec = dfdailyMin.loc[:,'December 2020'].mean()

#\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\
#######\\\\\\\\\\\\ RAINFALL \\\\\\\\\\##########
#\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\

#####\\\\\\\\ 2016 \\\\\\\\\\#####
#previous rain season for Wheat
s0 = dfdailyRainfall.loc[:,"November 2015":"May 2016"]
s0Sum = s0.sum()
sumRainfall16 = s0Sum.sum()
#s0.columns = range(s0.shape[1])
#sumRainfall16 = pd.Series(s0Sum.sum(),name="sumRainfall 16")
#Koring_tablesJoin
meanRain16Jan = dfdailyRainfall.loc[:,'January 2016'].mean()
meanRain16Feb = dfdailyRainfall.loc[:,'February 2016'].mean()
meanRain16Mar = dfdailyRainfall.loc[:,'March 2016'].mean()
meanRain16Apr = dfdailyRainfall.loc[:,'April 2016'].mean()
meanRain16May = dfdailyRainfall.loc[:,'May 2016'].mean()
meanRain16Jun = dfdailyRainfall.loc[:,'June 2016'].mean()
meanRain16Jul = dfdailyRainfall.loc[:,'July 2016'].mean()
meanRain16Aug = dfdailyRainfall.loc[:,'August 2016'].mean()
meanRain16Sep = dfdailyRainfall.loc[:,'September 2016'].mean()
meanRain16Oct = dfdailyRainfall.loc[:,'October 2016'].mean()
meanRain16Nov = dfdailyRainfall.loc[:,'November 2016'].mean()
meanRain16Dec = dfdailyRainfall.loc[:,'December 2016'].mean()


#####\\\\\\\\ 2017 \\\\\\\\\\#####
#previous rain season for Wheat
s = dfdailyRainfall.loc[:,"November 2016":"May 2017"]
sSum = s.sum()
sumRainfall17 = sSum.sum()
```

```
#s.columns = range(s.shape[1])
#sumRainfall17 = pd.Series(sSum.sum(),name="sumRainfall 17")
#Koring_tablesJoin
meanRain17Jan = dfdailyRainfall.loc[:,'January 2017'].mean()
meanRain17Feb = dfdailyRainfall.loc[:,'February 2017'].mean()
meanRain17Mar = dfdailyRainfall.loc[:,'March 2017'].mean()
meanRain17Apr = dfdailyRainfall.loc[:,'April 2017'].mean()
meanRain17May = dfdailyRainfall.loc[:,'May 2017'].mean()
meanRain17Jun = dfdailyRainfall.loc[:,'June 2017'].mean()
meanRain17Jul = dfdailyRainfall.loc[:,'July 2017'].mean()
meanRain17Aug = dfdailyRainfall.loc[:,'August 2017'].mean()
meanRain17Sep = dfdailyRainfall.loc[:,'September 2017'].mean()
meanRain17Oct = dfdailyRainfall.loc[:,'October 2017'].mean()
meanRain17Nov = dfdailyRainfall.loc[:,'November 2017'].mean()
meanRain17Dec = dfdailyRainfall.loc[:,'December 2017'].mean()

#####\\\\\\\\ 2018 \\\\\\\\\\#####
#previous rain season for Wheat
s1 = dfdailyRainfall.loc[:,"November 2017":"May 2018"]
s1Sum = s1.sum()
sumRainfall18 = s1Sum.sum()
#Koring_tablesJoin
meanRain18Jan = dfdailyRainfall.loc[:,'January 2018'].mean()
meanRain18Feb = dfdailyRainfall.loc[:,'February 2018'].mean()
meanRain18Mar = dfdailyRainfall.loc[:,'March 2018'].mean()
meanRain18Apr = dfdailyRainfall.loc[:,'April 2018'].mean()
meanRain18May = dfdailyRainfall.loc[:,'May 2018'].mean()
meanRain18Jun = dfdailyRainfall.loc[:,'June 2018'].mean()
meanRain18Jul = dfdailyRainfall.loc[:,'July 2018'].mean()
meanRain18Aug = dfdailyRainfall.loc[:,'August 2018'].mean()
meanRain18Sep = dfdailyRainfall.loc[:,'September 2018'].mean()
meanRain18Oct = dfdailyRainfall.loc[:,'October 2018'].mean()
meanRain18Nov = dfdailyRainfall.loc[:,'November 2018'].mean()
meanRain18Dec = dfdailyRainfall.loc[:,'December 2018'].mean()

#####\\\\\\\\ 2019 \\\\\\\\\\#####
s2 = dfdailyRainfall.loc[:,"November 2018":"May 2019"]
s2Sum = s2.sum()
sumRainfall19 = s2Sum.sum()
#Koring_tablesJoin
meanRain19Jan = dfdailyRainfall.loc[:,'January 2019'].mean()
meanRain19Feb = dfdailyRainfall.loc[:,'February 2019'].mean()
meanRain19Mar = dfdailyRainfall.loc[:,'March 2019'].mean()
meanRain19Apr = dfdailyRainfall.loc[:,'April 2019'].mean()
meanRain19May = dfdailyRainfall.loc[:,'May 2019'].mean()
meanRain19Jun = dfdailyRainfall.loc[:,'June 2019'].mean()
meanRain19Jul = dfdailyRainfall.loc[:,'July 2019'].mean()
meanRain19Aug = dfdailyRainfall.loc[:,'August 2019'].mean()
meanRain19Sep = dfdailyRainfall.loc[:,'September 2019'].mean()
meanRain19Oct = dfdailyRainfall.loc[:,'October 2019'].mean()
```

```
meanRain19Nov = dfdailyRainfall.loc[:,'November 2019'].mean()
meanRain19Dec = dfdailyRainfall.loc[:,'December 2019'].mean()

#####\\\\\\\\ 2020 \\\\\\\\\\#####
s3 = dfdailyRainfall.loc[:,"November 2019":"May 2020"]
s3Sum = s3.sum()
sumRainfall20 = s3Sum.sum()
#Koring_tablesJoin
meanRain20Jan = dfdailyRainfall.loc[:,'January 2020'].mean()
meanRain20Feb = dfdailyRainfall.loc[:,'February 2020'].mean()
meanRain20Mar = dfdailyRainfall.loc[:,'March 2020'].mean()
meanRain20Apr = dfdailyRainfall.loc[:,'April 2020'].mean()
meanRain20May = dfdailyRainfall.loc[:,'May 2020'].mean()
meanRain20Jun = dfdailyRainfall.loc[:,'June 2020'].mean()
meanRain20Jul = dfdailyRainfall.loc[:,'July 2020'].mean()
meanRain20Aug = dfdailyRainfall.loc[:,'August 2020'].mean()
meanRain20Sep = dfdailyRainfall.loc[:,'September 2020'].mean()
meanRain20Oct = dfdailyRainfall.loc[:,'October 2020'].mean()
meanRain20Nov = dfdailyRainfall.loc[:,'November 2020'].mean()
meanRain20Dec = dfdailyRainfall.loc[:,'December 2020'].mean()

#\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\
#######\\\\\\\\\\\\ HUMIDITY \\\\\\\\\\##########
#\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\

#####\\\\\\\\ 2016 \\\\\\\\\\#####
h0 = dfhumidity.loc[:,"December 2015":"October 2016"]
h0.columns = range(h0.shape[1])
meanHumidity16 = pd.Series(h0.mean(),name="meanHumidity 16")
meanHumid16Jan = dfhumidity.loc[:,'January 2016'].mean()
meanHumid16Feb = dfhumidity.loc[:,'February 2016'].mean()
meanHumid16Mar = dfhumidity.loc[:,'March 2016'].mean()
meanHumid16Apr = dfhumidity.loc[:,'April 2016'].mean()
meanHumid16May = dfhumidity.loc[:,'May 2016'].mean()
meanHumid16Jun = dfhumidity.loc[:,'June 2016'].mean()
meanHumid16Jul = dfhumidity.loc[:,'July 2016'].mean()
meanHumid16Aug = dfhumidity.loc[:,'August 2016'].mean()
meanHumid16Sep = dfhumidity.loc[:,'September 2016'].mean()
meanHumid16Oct = dfhumidity.loc[:,'October 2016'].mean()
meanHumid16Nov = dfhumidity.loc[:,'November 2016'].mean()
meanHumid16Dec = dfhumidity.loc[:,'December 2016'].mean()

#####\\\\\\\\ 2017 \\\\\\\\\\#####
h = dfhumidity.loc[:,"December 2016":"October 2017"]
h.columns = range(h.shape[1])
meanHumidity17 = pd.Series(h.mean(),name="meanHumidity 17")
meanHumid17Jan = dfhumidity.loc[:,'January 2017'].mean()
meanHumid17Feb = dfhumidity.loc[:,'February 2017'].mean()
meanHumid17Mar = dfhumidity.loc[:,'March 2017'].mean()
meanHumid17Apr = dfhumidity.loc[:,'April 2017'].mean()
```

```
meanHumid17May = dfhumidity.loc[:,'May 2017'].mean()
meanHumid17Jun = dfhumidity.loc[:,'June 2017'].mean()
meanHumid17Jul = dfhumidity.loc[:,'July 2017'].mean()
meanHumid17Aug = dfhumidity.loc[:,'August 2017'].mean()
meanHumid17Sep = dfhumidity.loc[:,'September 2017'].mean()
meanHumid17Oct = dfhumidity.loc[:,'October 2017'].mean()
meanHumid17Nov = dfhumidity.loc[:,'November 2017'].mean()
meanHumid17Dec = dfhumidity.loc[:,'December 2017'].mean()

#####\\\\\\\\ 2018 \\\\\\\\\\#####
h1 = dfhumidity.loc[:,"December 2017":"October 2018"]
h1.columns = range(h1.shape[1])
meanHumidity18 = pd.Series(h1.mean(),name="meanHumidity 18")
meanHumid18Jan = dfhumidity.loc[:,'January 2018'].mean()
meanHumid18Feb = dfhumidity.loc[:,'February 2018'].mean()
meanHumid18Mar = dfhumidity.loc[:,'March 2018'].mean()
meanHumid18Apr = dfhumidity.loc[:,'April 2018'].mean()
meanHumid18May = dfhumidity.loc[:,'May 2018'].mean()
meanHumid18Jun = dfhumidity.loc[:,'June 2018'].mean()
meanHumid18Jul = dfhumidity.loc[:,'July 2018'].mean()
meanHumid18Aug = dfhumidity.loc[:,'August 2018'].mean()
meanHumid18Sep = dfhumidity.loc[:,'September 2018'].mean()
meanHumid18Oct = dfhumidity.loc[:,'October 2018'].mean()
meanHumid18Nov = dfhumidity.loc[:,'November 2018'].mean()
meanHumid18Dec = dfhumidity.loc[:,'December 2018'].mean()

#####\\\\\\\\ 2019 \\\\\\\\\\#####
h2 = dfhumidity.loc[:,"December 2018":"October 2019"]
h2.columns = range(h2.shape[1])
meanHumidity19 = pd.Series(h2.mean(),name="meanHumidity 19")
meanHumid19Jan = dfhumidity.loc[:,'January 2019'].mean()
meanHumid19Feb = dfhumidity.loc[:,'February 2019'].mean()
meanHumid19Mar = dfhumidity.loc[:,'March 2019'].mean()
meanHumid19Apr = dfhumidity.loc[:,'April 2019'].mean()
meanHumid19May = dfhumidity.loc[:,'May 2019'].mean()
meanHumid19Jun = dfhumidity.loc[:,'June 2019'].mean()
meanHumid19Jul = dfhumidity.loc[:,'July 2019'].mean()
meanHumid19Aug = dfhumidity.loc[:,'August 2019'].mean()
meanHumid19Sep = dfhumidity.loc[:,'September 2019'].mean()
meanHumid19Oct = dfhumidity.loc[:,'October 2019'].mean()
meanHumid19Nov = dfhumidity.loc[:,'November 2019'].mean()
meanHumid19Dec = dfhumidity.loc[:,'December 2019'].mean()

#####\\\\\\\\ 2020 \\\\\\\\\\#####
h3 = dfhumidity.loc[:,"December 2019":"October 2020"]
h3.columns = range(h3.shape[1])
meanHumidity20 = pd.Series(h3.mean(),name="meanHumidity 20")
meanHumid20Jan = dfhumidity.loc[:,'January 2020'].mean()
meanHumid20Feb = dfhumidity.loc[:,'February 2020'].mean()
meanHumid20Mar = dfhumidity.loc[:,'March 2020'].mean()
```

```
meanHumid20Apr = dfhumidity.loc[:,'April 2020'].mean()
meanHumid20May = dfhumidity.loc[:,'May 2020'].mean()
meanHumid20Jun = dfhumidity.loc[:,'June 2020'].mean()
meanHumid20Jul = dfhumidity.loc[:,'July 2020'].mean()
meanHumid20Aug = dfhumidity.loc[:,'August 2020'].mean()
meanHumid20Sep = dfhumidity.loc[:,'September 2020'].mean()
meanHumid20Oct = dfhumidity.loc[:,'October 2020'].mean()
meanHumid20Nov = dfhumidity.loc[:,'November 2020'].mean()
meanHumid20Dec = dfhumidity.loc[:,'December 2020'].mean()


#\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\
#######\\\\\\\\\\\\ PRESSURE \\\\\\\\\\##########
#\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\

#####\\\\\\\\ 2016 \\\\\\\\\\#####
p0 = dfPressure.loc[:,'December 2015':"October 2016"]
p0.columns = range(p0.shape[1])
meanPressure16 = pd.Series(p0.mean(),name="Pressure 16")
meanPressure16Jan = dfPressure.loc[:,'January 2016'].mean()
meanPressure16Feb = dfPressure.loc[:,'February 2016'].mean()
meanPressure16Mar = dfPressure.loc[:,'March 2016'].mean()
meanPressure16Apr = dfPressure.loc[:,'April 2016'].mean()
meanPressure16May = dfPressure.loc[:,'May 2016'].mean()
meanPressure16Jun = dfPressure.loc[:,'June 2016'].mean()
meanPressure16Jul = dfPressure.loc[:,'July 2016'].mean()
meanPressure16Aug = dfPressure.loc[:,'August 2016'].mean()
meanPressure16Sep = dfPressure.loc[:,'September 2016'].mean()
meanPressure16Oct = dfPressure.loc[:,'October 2016'].mean()
meanPressure16Nov = dfPressure.loc[:,'November 2016'].mean()
meanPressure16Dec = dfPressure.loc[:,'December 2016'].mean()
```

ETC…. for all the weather features

```
#\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\
### YEARLY TABLES FOR LAZY REGRESSOR CALCS, MEAN CHL, KPI & WEATHER ##
#\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\

#table2017 =
pd.concat([koring17JulMean,koring17AugMean,koring17SepMean,dataDummies],axis=1)
#dataDummies or mainNutrients
table2017 = pd.concat([koring17JulMean,koring17AugMean,mainNutrients],axis=1)  ## For 2019
Aug prediction
#table2017.loc[:,'Min Jun'] = meanMin17Jun
table2017.loc[:,'Min Jul'] = meanMin17Jul
table2017.loc[:,'Min Aug'] = meanMin17Aug
#table2017.loc[:,'Max Jun'] = meanMax17Jun
table2017.loc[:,'Max Jul'] = meanMax17Jul
table2017.loc[:,'Max Aug'] = meanMax17Aug
```

```
table2017.loc[:,'Rain prev sum'] = sumRainfall17
#table2017.loc[:,'Rain Jun'] = meanRain17Jun
# table2017.loc[:,'Rain Jul'] = meanRain17Jul
# table2017.loc[:,'Rain Aug'] = meanRain17Aug
# table2017.loc[:,'Rain Sep'] = meanRain17Sep
#table2017.loc[:,'Humid Jun'] = meanHumid17Jun
table2017.loc[:,'Humid Jul'] = meanHumid17Jul
table2017.loc[:,'Humid Aug'] = meanHumid17Aug
#table2017.loc[:,'Wind Aug'] = meanWindSpeed17Aug


#table2018 = pd.concat([koring18JulMean,koring18AugMean,
koring18SepMean,dataDummies],axis=1)
table2018 = pd.concat([koring18JulMean,koring18AugMean,mainNutrients],axis=1)   ## For 2019
Aug prediction
#table2018.loc[:,'Min Jun'] = meanMin18Jun
table2018.loc[:,'Min Jul'] = meanMin18Jul
table2018.loc[:,'Min Aug'] = meanMin18Aug
#table2018.loc[:,'Max Jun'] = meanMax18Jun
table2018.loc[:,'Max Jul'] = meanMax18Jul
table2018.loc[:,'Max Aug'] = meanMax18Aug
table2018.loc[:,'Rain prev sum'] = sumRainfall18
# table2018.loc[:,'Rain Jun'] = meanRain18Jun
# table2018.loc[:,'Rain Jul'] = meanRain18Jul
# table2018.loc[:,'Rain Aug'] = meanRain18Aug
# table2018.loc[:,'Rain Sep'] = meanRain18Sep
#table2018.loc[:,'Humid Jun'] = meanHumid18Jun
table2018.loc[:,'Humid Jul'] = meanHumid18Jul
table2018.loc[:,'Humid Aug'] = meanHumid18Aug
#table2018.loc[:,'Wind Aug'] = meanWindSpeed18Aug

#table2019 =
pd.concat([koring19JulMean,koring19AugMean,koring19SepMean,dataDummies],axis=1)
table2019 = pd.concat([koring19JulMean,koring19AugMean,mainNutrients],axis=1)   ## For 2019
Aug prediction
#table2019.loc[:,'Min Jun'] = meanMin19Jun
table2019.loc[:,'Min Jul'] = meanMin19Jul
table2019.loc[:,'Min Aug'] = meanMin19Aug
#table2019.loc[:,'Max Jun'] = meanMax19Jun
table2019.loc[:,'Max Jul'] = meanMax19Jul
table2019.loc[:,'Max Aug'] = meanMax19Aug
table2019.loc[:,'Rain prev sum'] = sumRainfall19
# table2019.loc[:,'Rain Jun'] = meanRain19Jun
# table2019.loc[:,'Rain Jul'] = meanRain19Jul
# table2019.loc[:,'Rain Aug'] = meanRain19Aug
# table2019.loc[:,'Rain Sep'] = meanRain19Sep
#table2019.loc[:,'Humid Jun'] = meanHumid19Jun
table2019.loc[:,'Humid Jul'] = meanHumid19Jul
table2019.loc[:,'Humid Aug'] = meanHumid19Aug
#table2019.loc[:,'Wind Aug'] = meanWindSpeed19Aug
```

```
table2020 = pd.concat([koring20JulMean,koring20AugMean,mainNutrients],axis=1)   #los
koring20SepMean uit want X TEST (with filtered)
#table2020.loc[:,'Min Jun'] = meanMin20Jun
table2020.loc[:,'Min Jul'] = meanMin20Jul
table2020.loc[:,'Min Aug'] = meanMin20Aug
#table2020.loc[:,'Max Jun'] = meanMax20Jun
table2020.loc[:,'Max Jul'] = meanMax20Jul
table2020.loc[:,'Max Aug'] = meanMax20Aug
table2020.loc[:,'Rain prev sum'] = sumRainfall20
# table2020.loc[:,'Rain Jun'] = meanRain20Jun
# table2020.loc[:,'Rain Jul'] = meanRain20Jul
# table2020.loc[:,'Rain Aug'] = meanRain20Aug
# table2020.loc[:,'Rain Sep'] = meanRain20Sep
#table2020.loc[:,'Humid Jun'] = meanHumid20Jun
table2020.loc[:,'Humid Jul'] = meanHumid20Jul
table2020.loc[:,'Humid Aug'] = meanHumid20Aug
#table2020.loc[:,'Wind Aug'] = meanWindSpeed20Aug
print(table2020)

# # FOR 2019 Aug PREDICTIONS
koring_tablesJoin =
pd.concat([table2017,table2018,table2019],keys=["2017","2018","2019"],ignore_index=True)  #for
predictions and 2020
#koring_tablesJoin.to_excel(r'C:\Users\jclau\Documents\Documents\AMasters-2020\Masters
Project\Data\Excel\KoringTableAUG19.xlsx', index = False)

# \\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\
##################### HEATMAP  ########################
# \\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\

table2017_cor = pd.concat([koring17SepMean,dataDummies],axis=1)    #dataDummies or
mainNutrients
# table2017_cor.loc[:,'Min Sep'] = meanMin17Sep
# table2017_cor.loc[:,'Max Sep'] = meanMax17Sep
# table2017_cor.loc[:,'Rain prev sum'] = sumRainfall17

fig, ax = plt.subplots(figsize=(10,10))
dataCorr = table2017_cor.corr()
corrMatrix = sns.heatmap(dataCorr, annot = True, linewidths=.8, ax=ax)       #plot correlation matrix
plt.xticks(fontsize=10)
plt.yticks(fontsize=10)
plt.title("Correlation Matrix - Sep 2017 & features")
plt.tight_layout()
plt.show()

#\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\
############################## LAZY PREDICT #################
#\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\
# y = koring_tablesJoin['K. Sep']
# X = koring_tablesJoin.drop(["K. Sep"],axis=1)
# scaler_norm = MinMaxScaler()
# X = scaler_norm.fit_transform(X)
# y = scaler_norm.fit_transform(y)

# X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=.2, random_state=42)
# # #fit all models
# reg = LazyRegressor(predictions=True)
# models, predictions = reg.fit(X_train, X_test, y_train, y_test)

# print(models)

#\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\
############ SEQUENCIAL FEATURE SELECTION #################
#\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\

from mlxtend.plotting import plot_sequential_feature_selection as plot_sfs
#||||||||||||||||||||||||||||||||||||||||||||||||||||||
# OPTIMAL NUMBER OF FEATURES:
# ||||||||||||||||||||||||||||||||||||||||||||||||||||||
# hgbr = HistGradientBoostingRegressor()
# rfr = RandomForestRegressor()
# xgbr = XGBRegressor()
# y = koring_tablesJoin['K. Sep']
# X = koring_tablesJoin.drop(["K. Sep"],axis=1)
# #scaler_norm = MinMaxScaler()
# #X = scaler_norm.fit_transform(X)


# sfs = SequentialFeatureSelector(hgbr,
#        k_features=15,
#        forward=True,
#        floating=False,
#        scoring='r2',
#        cv=5)

# # fit the object to the training data
# sfs.fit(X,y)

# fig1 = plot_sfs(sfs.get_metric_dict(),kind='std_dev')
# plt.title('Sequential Forward Selection std_err')
# plt.grid()
# plt.show()

###############################################################################
##################
# y = koring_tablesJoin['K. Sep']
# X = koring_tablesJoin.drop(["K. Sep"],axis=1)
# X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)
```

```
#print('X_train: ',X_train.shape,'\nX_test: ',X_test.shape,'\ny_train: ',y_train.shape,'\ny_test:
',y_test.shape)  #y_test type = Series
#scaler_std = StandardScaler() #doesnt work for continous values
#scaler_rob = RobustScaler()
#X = scaler_std.fit_transform(X)
#scaler_norm = MinMaxScaler()
#X = scaler_norm.fit_transform(X)
#scaled_X = pd.DataFrame(X)
#X = scaler_rob.fit_transform(X)
hgbr = HistGradientBoostingRegressor()
rfr = RandomForestRegressor()
xgbr = XGBRegressor()
etr = ExtraTreesRegressor()


# # #||||||||||||||||||||||||||||||||||||||||||||||||||||||
# # #Sequential Forward Selection
# sfs = SequentialFeatureSelector(etr,
#        k_features=12,
#        forward=True,
#        floating=False,
#        verbose = 1,
#        scoring = 'r2',
#        cv=5)

# # fit the object to the training data. It calculates the parameters or weights on the training data
# sfs.fit(X_train,y_train)

# print('Forward sequential feauture selection index:',sfs.k_feature_idx_,' Feature name:
',sfs.k_feature_names_)
# print('\nForward sequential feature selector with Extra Trees Regressor (Normalised):')
# feature_ranks = list(zip(sfs.k_feature_idx_,sfs.k_feature_names_))
# for feat in feature_ranks:
#     print('Feature Index: {},  Names: {}'.format(feat[0], feat[1]))
# print('\nCV Score:',sfs.k_score_)

# # Now use the subset of selected features to fit model on training data
# X_train_sfs = sfs.transform(X_train)
# x_test_sfs = sfs.transform(X_test)

# # Fit the estimator using the new feature subset
# # and make a prediction on the test data
# newModel = etr.fit(X_train_sfs,y_train)        # MODELS !!!!!!!
# y_predict = etr.predict(x_test_sfs)

# ypredict = list(zip(y_test,y_predict))
# for preds in ypredict:
#     print('Original y: {:.0f},  Predicted y: {:.0f}'.format(preds[0], preds[1]))
# accmae = mean_absolute_error(y_test,y_predict)
# accmse = mean_squared_error(y_test,y_predict)
# accr2 = r2_score(y_test,y_predict)
# print('\n Mean Asolute Error :',accmae,'\n Mean Squared Error :',accmse,'\n R2 score:',accr2)


# Compute the accuracy of the prediction
#acc = float((y_test == y_pred).sum()) / y_pred.shape[0]
#print('Test set accuracy: %.2f %%' % (acc * 100))
# #Confusion Matrix - verify accuracy of each class
# from sklearn.metrics import confusion_matrix
# cm = confusion_matrix(y_test, prediction_hist)
# print(cm)
# sns.heatmap(cm, annot=True)
#acc = accuracy_score(y_test,y_predict)    #normalise=False return nr of correct predictions,
otherwise fraction(TRUE) -for classification

#||||||||||||||||||||||||||||||||||||||||||||||||||||||
# Sequential Backward Selection
# sbs = SequentialFeatureSelector(etr,
#         k_features=9,
#         forward=False,
#         floating=False,
#         verbose = 1,
#         scoring = 'r2',
#         cv=5)

# sbs.fit(X_train,y_train)
# print('\nBackward sequential feature selector with ETR :')      #RANDOM FOREST REGRESSOR
# feature_ranks1 = list(zip(sbs.k_feature_idx_,sbs.k_feature_names_))
# for feat in feature_ranks1:
#     print('Feature Index: {},  Names: {}'.format(feat[0], feat[1]))
# print('\nCV Score:',sbs.k_score_)

# # Now use the subset of selected features to fit model on training data
# X_train_sbs = sbs.transform(X_train)
# x_test_sbs = sbs.transform(X_test)

# # Fit the estimator using the new feature subset
# # and make a prediction on the test data
# newModel = etr.fit(X_train_sbs,y_train)
# y_predict = etr.predict(x_test_sbs)

# ypredict = list(zip(y_test,y_predict))
# for preds in ypredict:
#     print('Original y: {:.0f},  Predicted y: {:.0f}'.format(preds[0], preds[1]))

# accmae = mean_absolute_error(y_test,y_predict)
# accmse = mean_squared_error(y_test,y_predict)
# accr2 = r2_score(y_test,y_predict)
# print('\n Mean Asolute Error :',accmae,'\n Mean Squared Error :',accmse,'\n R2 :',accr2)

# #|||||||||||||||||||||||||||||||||||||||||||||||||||||
# # ## Sequential Forward floating Selection
# sffs = SequentialFeatureSelector(etr,
#         k_features=9,
#         forward=True,
#         floating=True,
#         verbose = 1,
#         scoring = 'r2',
#         cv=5)

# sffs.fit(X_train,y_train)
# print('\nForward floating sequential feature selector with ETR:')      #RANDOM FOREST
REGRESSOR
# feature_ranks2 = list(zip(sffs.k_feature_idx_,sffs.k_feature_names_))
# for feat in feature_ranks2:
#     print('Feature Index: {},  Names: {}'.format(feat[0], feat[1]))
# print('\nCV Score:',sffs.k_score_)

# # #Now use the subset of selected features to fit model on training data
# X_train_sffs = sffs.transform(X_train)
# x_test_sffs = sffs.transform(X_test)

# # #Fit the estimator using the new feature subset
# # #and make a prediction on the test data
# newModel = etr.fit(X_train_sffs,y_train)
# y_predict = etr.predict(x_test_sffs)

# ypredict = list(zip(y_test,y_predict))
# for preds in ypredict:
#     print('Original y: {:.0f},  Predicted y: {:.0f}'.format(preds[0], preds[1]))

# accmae = mean_absolute_error(y_test,y_predict)
# accmse = mean_squared_error(y_test,y_predict)
# accr2 = r2_score(y_test,y_predict)
# print('\n Mean Asolute Error :',accmae,'\n Mean Squared Error :',accmse,'\n R2 :',accr2)


# #|||||||||||||||||||||||||||||||||||||||||||||||||||||
# # Sequential Backward floating Selection
# sbfs = SequentialFeatureSelector(etr,
#         k_features=9,
#         forward=False,
#         floating=True,
#         verbose = 1,
#         scoring = 'r2',
#         cv=5)

# sbfs.fit(X_train,y_train)
# print('\nBackward floating sequential feature selector with ETR(norm):')    # HistgradboostRegr
Extra Trees Regr (Normalised
# feature_ranks3 = list(zip(sbfs.k_feature_idx_,sbfs.k_feature_names_))
# for feat in feature_ranks3:
#     print('Feature Index: {},  Names: {}'.format(feat[0], feat[1]))
# print('\nCV Score:',sbfs.k_score_)

# # #Now use the subset of selected features to fit model on training data
# X_train_sbfs = sbfs.transform(X_train)
# x_test_sbfs = sbfs.transform(X_test)

# # #Fit the estimator using the new feature subset
# # #and make a prediction on the test data
# newModel = etr.fit(X_train_sbfs,y_train)
# y_predict = etr.predict(x_test_sbfs)

# ypredict = list(zip(y_test,y_predict))
# for preds in ypredict:
#     print('Original y: {:.0f},  Predicted y: {:.0f}'.format(preds[0], preds[1]))

# accmae = mean_absolute_error(y_test,y_predict)
# accmse = mean_squared_error(y_test,y_predict)
# accr2 = r2_score(y_test,y_predict)
# print('\n Mean Asolute Error :',accmae,'\n Mean Squared Error :',accmse,'\n R2 :',accr2)


#|||||||||||||||||||||||||||||||||||||||||||||||||||||||||
#|||||||||||||||||| PREDICTION - 2020 |||||||||||||||||||||||  USE SFFS and ETR not normalised
#|||||||||||||||||||||||||||||||||||||||||||||||||||||||||
# y = koring_tablesJoin['K. Sep']
# X = koring_tablesJoin.drop(["K. Sep"],axis=1)
# X20_test = table2020
# scaler_norm = MinMaxScaler()
# X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)
# print('X_train: ',X_train.shape,'\nX_test: ',X_test.shape,'\ny_train: ',y_train.shape,'\ny_test:
',y_test.shape)  #y_test type = Series
# X = scaler_norm.fit_transform(X)
# X20_test = scaler_norm.fit_transform(X20_test)


# ## Sequential Forward Selection
# sfs = SequentialFeatureSelector(etr,
#        k_features=9,
#        forward=True,
#        floating=False,
#        verbose = 1,
#        scoring = 'r2',
#        cv=5)

# ## fit the object to the training data. It calculates the parameters or weights on the training data
# sfs.fit(X_train,y_train)

# print('Forward sequential feauture selection index:',sfs.k_feature_idx_,' Feature name:
',sfs.k_feature_names_)
# print('\nForward sequential feature selector with Extra Trees Regressor (Normalised):')
# feature_ranks = list(zip(sfs.k_feature_idx_,sfs.k_feature_names_))
# for feat in feature_ranks:
#     print('Feature Index: {},  Names: {}'.format(feat[0], feat[1]))
# print('\nCV Score:',sfs.k_score_)

# ## Now use the subset of selected features to fit model on training data
# #X becomes the new X_train. we want to train 2017,18 & 19
# # ***** FOR 2020 PREDICTION ****
# X_train_sfs = sfs.transform(X)
# x20_test_sfs = sfs.transform(X20_test)

# # Fit the estimator using the new feature subset
# # and make a prediction on the test data
# # ***** FOR 2020 PREDICTION ****
# Model_2020 = etr.fit(X_train_sfs,y)
# y_predict2020 = etr.predict(x20_test_sfs)

# print('X_train: ',X.shape,'\nX_test: ',X20_test.shape,'\ny_train: ',y.shape)
# print('\nPredictions for September 2020:\n',y_predict2020)

# koring20SepMean = pd.Series(y_predict2020,name="K. Sep")
# print(koring20SepMean)


# \\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\
###### \\\\\\\\\\\\\ CHLOROPHYLL PREDICTIONS – GRAPHING  \\\\\\\\\\\\\\ #####
# \\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\

#///// CHL
augmean = koring17Aug_1.mean()  #koring17Aug_1.mean()  (koring17Aug_1.mean() +
koring18Aug_1.mean())/2
aug1stdev = koring17Aug_1.std()   #koring17Aug_1.std()  (koring17Aug_1.std() +
koring18Aug_1.std())/2
std_low = augmean- aug1stdev
std_low2 = (augmean-(2*aug1stdev))
std_high = augmean+aug1stdev
std_high2 = (augmean+(2*aug1stdev))

x = round(rawdata["X"],5)
y = round(rawdata["Y"],5)
z = koring18Aug_1       # ----- koring18Aug_1
x_ax = np.arange(x.min(),x.max(),0.0005)
fig1, ax1 = plt.subplots()

# ####loop through every item in the series
```

```
col = []
for a in range(0,len(z),1):
    if z[a] < std_low2:
        col.append('red')
    elif (z[a] >= std_low2) and (z[a] < std_low) :
        col.append('lightcoral')
    elif (z[a] >= std_low) and (z[a] < augmean):   #-----
        col.append('pink')
    elif (z[a] >= augmean) and (z[a] < std_high):  #-----
        col.append('khaki')
    elif (z[a] >= std_high) and (z[a] < std_high2):
        col.append('lime')
    elif (z[a] >= std_high2):
        col.append('darkgreen')
    else:
        col.append('gray')

red_patch = mpatches.Patch(color='red', label='Z <-2stdev')
coral_patch = mpatches.Patch(color='lightcoral', label='-2stdev <= Z < -1stdev')
pink_patch = mpatches.Patch(color='pink', label='-1stdev <= Z <mean')
khaki_patch = mpatches.Patch(color='khaki', label='mean <= Z <1stdev')
lime_patch = mpatches.Patch(color='lime', label='1stdev <= Z <2stdev')
dgreen_patch = mpatches.Patch(color='darkgreen', label='Z >= 2stdev')
gray_patch = mpatches.Patch(color='gray', label='nan')

for i in range(0,len(z),1):
    ax1.scatter(x[i],y[i],c=col[i])
plt.xlabel('X axis',fontsize=15)
plt.ylabel('Y axis',fontsize=15)
plt.title('08/08/2018 VS 08/08/2017',fontsize=20)  # ------
#plt.xticks(x_ax)
#plt.legend(handles=[red_patch,coral_patch,pink_patch,khaki_patch,lime_patch,dgreen_patch,gra
y_patch])
plt.tight_layout()
plt.grid()
plt.show()
```