APSIM Crop Production Model

Siddharth Krishnakumar (s4651743)

Faculty of Engineering, Architecture, and Information Technology,
The University of Queensland.

Department of Environment and Science, Government of Queensland.

Master of Data Science – Capstone Project 2 (DATA7902)

Supervisor: Dr. Ross McVinish

THE UNIVERSITY OF QUEENSLAND
AUSTRALIA

Queensland Government

Department of Environment and Science

Declaration

I hereby declare that the thesis titled “APSIM Crop Production Model" submitted by me, for the award of the *Master in Data Science* degree at The University of Queensland is a record of bona fide work carried out under the supervision of Dr Ross McVinish. The work in this thesis contains due references to the authors for the information obtained in Literature.

I further declare that the work reported in this thesis is submitted for the course of ‘Data Science Capstone Project 2 [DATA7902]’. This work has not been submitted and will not be submitted, either in part or in full, for the award of any other degree or diploma in this institute or any other institute or university.

Place: **Brisbane**

Date: **06/06/2022**

Siddharth Krishnakumar

Acknowledgement

I want to express my sincere gratitude to several individuals and organisations for supporting me throughout my Postgraduate study. Firstly, I would like to express my sincere gratitude to the Department of Environment and Science for providing the opportunity to work on this project. I would also like to thank The University of Queensland at St. Lucia for accepting me into their Master of Data Science Program and providing a pleasant working environment.

I would also like to thank my supervisor, Dr Ross McVinish, for his patience, insightful comments, unceasing ideas, and continuous support. I have received tremendous help at all times and in the writing of this thesis. His immense knowledge, profound experience and professional expertise have enabled me to complete this thesis successfully.

I would also like to thank the school of EAIT (Engineering, Architecture, and Information Technology) for providing the required skills and materials to pursue this project.

Last but not least, I would like to thank my peers, friends and family who continuously supported me during this endeavor. This would not have been possible without their constant motivation and encouragement.

Table of Contents

Table of Contents

List of Figures

List of Tables

Executive Summary

The APSIM Crop Production Model is a tool that is used to simulate an agricultural ecosystem. It consists of various components, such as plant, soil, and climate modules. The soil module is based on several soil properties, such as the amount of clay, organic carbon content, fine sand particle size, and inter-particle spaces. Due to complications in obtaining manual readings for specific properties (*such as Bulk Density*), there is a need to establish a relation between such variables and other soil properties for their prediction. A relation between the soil properties and Bulk Density should be established using Pedo-Transfer functions.

Bulk Density is defined as the mass of the material's particles divided by the total occupancy volume. The dataset comprises of soil data from various sources (Queensland Govt. and others). Inconsistencies (duplicate data, missing observations) must be removed from the dataset. Mathematical tools (Splines) can be implemented to obtain consistent readings for all soil properties, at every interval. Machine Learning models are used to fit the data to observe patterns between the input variables and the target variable (Bulk Density). The model that best fits the data and which has the lowest prediction errors is used to obtain Pedo-Transfer functions, which is used for Bulk Density Predictions.

Introduction

The APSIM (Agricultural Production System sIMulator) Crop Production Model is a software tool used to simulate an agricultural ecosystem. It consists of various modules, such as plant, soil, climate, and water modules. The soil ecosystem forms the basic framework for the soil module (Holzworth *et al.*, 2014). Various soil factors and information is present in the soil module of the APSIM toolbox (Fig. 2). The main impact of the APSIM model is to help tackle issues such as climate change and food security and to increase ecosystem productivity. The APSIM model is utilised primarily by researchers to evaluate farming practices, assess climate adaptation methods, nutrient quality management (soil), and more. The APSIM model was built based on other models and incorporated development from various groups. The process that led to the formation of APSIM is highlighted in Fig. 3.

The mapping of soil properties plays a crucial role in agriculture. The nature and constituents of soil vary from one location to another. Furthermore, soil properties vary with an increase in depth from the soil surface (examples include organic carbon content, % of Sulphur, % of Nitrogen, number of fine particles and amount of clay). In this case, no trend can be observed, and changes are very generic. Measuring soil properties physically can be a time and resource-consuming process and expensive. Additionally, specific soil properties cannot be estimated directly. Such properties could have an underlying relationship with other soil variables.

One such soil parameter is bulk density. It can be defined as the mass of particles of a material, divided by the volume the material occupies. Its direct estimation is an intensive process and, therefore, not efficient. Identifying its underlying relationship with other soil variables would help estimate it quickly. Methods are adapted from Data Science to predict such values. With the help of Pedo-Transfer functions, Bulk Density values can be predicted.

Specific soil properties that cannot be easily measured are estimated using Pedo-Transfer Functions (PTFs). The main reason for using Pedo-Transfer functions is that Bulk-Density measurements are time-consuming and lacking in various soil datasets (Martin *et al.*, 2009). Additionally, Pedo-Transfer functions are predictive functions that help translate raw soil data into usable information.

Previous Pedo-Transfer functions were used to help predict Bulk-Density. Current soil data demand higher accuracies for the estimation of soil properties. Pedo-Transfer functions have a history of being applied only to a specific soil type and are not generalised. Novel Data Science methods are used to develop new PTFs that are more accurate to help improve predictions of Bulk Density.

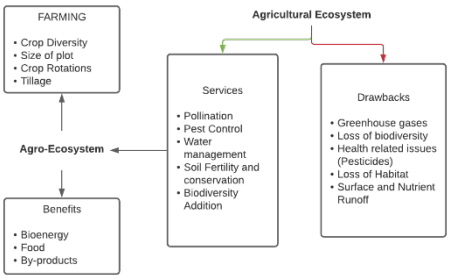

Fig. 1 - Components of an agricultural system.

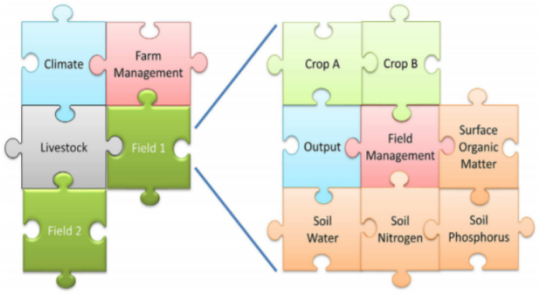

Fig. 2 – APSIM Toolbox (Holzworth et al., 2014)

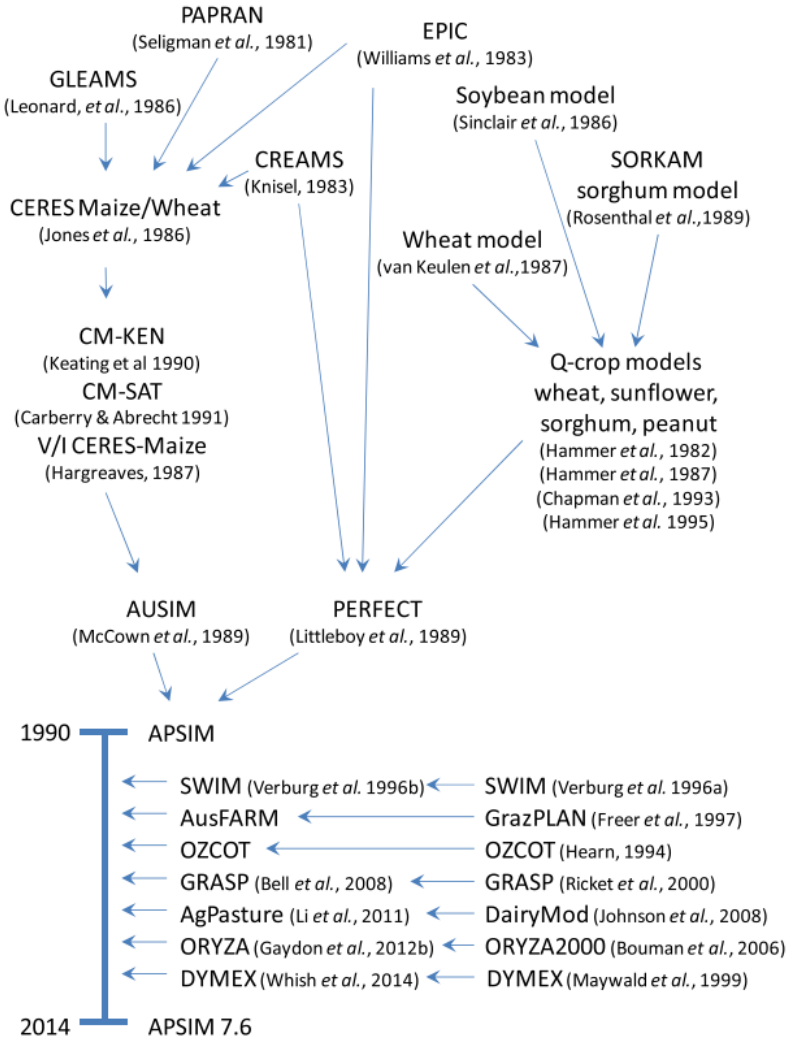

Fig. 3 – Model Pedigree of APSIM & Models that have influenced the inception of APSIM. (Holzworth et al., 2014)

Problem Statement

The APSIM model consists of various components (examples include soil, climate, water, and plant modules). To simulate the entirety of the agricultural ecosystem, data should be complete and in a format that the model can directly ingest. Concerning the soil module, Bulk Density measurements are not permanently recorded and are often missing in various datasets. Hence to complete datasets for the APSIM model, Bulk Density values should be estimated with the help of Pedo-Transfer functions. Novel Data Science methods can improve the existing Pedo-Transfer functions to predict Bulk Density with higher accuracy.

- The soil dataset in the APSIM Model is incomplete due to the absence of various data.
- Relationship between Bulk Density and other soil parameters is not well known.
- Although existing Pedo-Transfer functions predict Bulk Density, their performance and accuracy are unknown regarding current Data Science methods.
- Since Bulk Density is not always estimated, measures should be taken to provide accurate Bulk Density predictions from other soil parameters.

Objective

- Incomplete data in the soil database makes it difficult for the APSIM model to simulate an ecosystem accurately. **Filling in the incomplete data would produce accurate results from the APSIM model.**
- Certain soil parameters (such as Bulk Density) cannot be estimated directly without posing a strain on time, money, and resources. Such parameters could have underlying relations with other soil variables. **The connection between Bulk Density and other parameters should be established to assist in accurate predictions.**
- Pedo-Transfer functions are used to estimate soil variables dependent on other variables. **Novel Pedo-Transfer functions can help in more accurate predictions of Bulk Density.** Additionally, this also helps complete the soil data and spatial observations.
- **Incorporation of any models (used to develop PTFs) should be tested for their performance, using specific metrics such as goodness of fit (R^2 metric) and Standard Deviation of Residuals (RMSE – Root Mean Squared Error).**

Brief – APSIM & Soil Model

APSIM Model

It becomes increasingly harder to address issues such as food security. This is due to the increase in the number of uncontrollable factors and unpredictable factors. The APSIM model would help address these issues, mainly climate change and food security. More than 20 years of research and development have been incorporated to create APSIM, which provides simulation tools to address various challenges (Holzworth *et al.*, 2014).

Continuous improvements are incorporated into APSIM to address the pressure on farming systems. It contains various components to account for many factors in a live ecosystem (soil parameters, crop-related information, and different biophysical processes).

Some aspects of the APSIM model are (not limited to these):

- A platform to share information between the various constituents of APSIM.
- A user interface that helps clients utilize the models for their benefit.
- Software tools to facilitate data exchange between various APSIM models.
- Biophysical models that help address the technical issues of the modelled ecosystem (a model refers to a set of processes, which could be either physical or chemical or both).

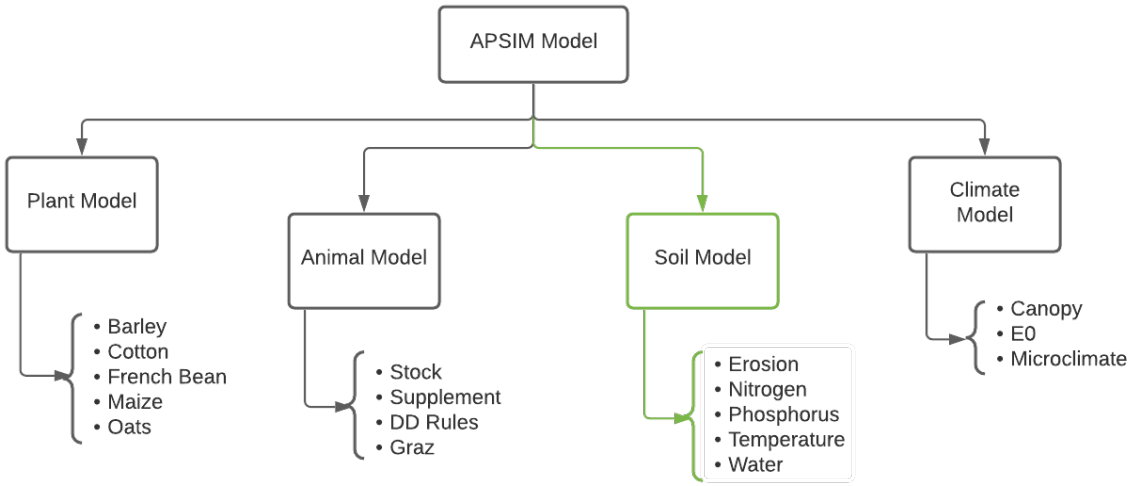

Fig. 4 – **Constituents of various models in APSIM**

Soil Model

The soil model is one of the primary models in the APSIM software. It contains all the relevant information that is recorded from the soil. Each soil parameter can be classified as dependent or independent. Some independent parameters include Particle size and nature, amount of clay, N_2 content, soil pH (power of Hydrogen) and dependent parameters include Bulk Density and water retention capacity of the soil. It is crucial to note that soil parameters do not have a fixed trend of variation with an increase in depth from the soil surface (Fig. 5). Since the physical estimation of such parameters could be inefficient, prediction methods are used (Pedo-Transfer functions).

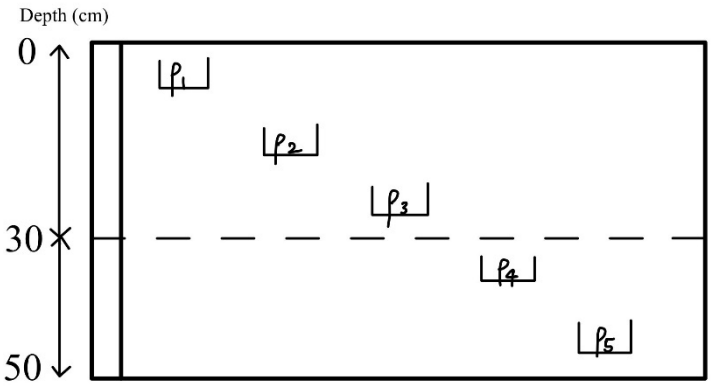

Fig. 5 – **Variation of Bulk Density in the soil**

Literature Review – Pedo-Transfer Functions and Bulk Density

Pedo-Transfer Functions (PTFs) are used to measure those properties of the soil that cannot be easily calculated from other properties which are already available. One such ideal example is Bulk Density, which, in short, is measured as the dry weight of soil per unit volume. This volume comprises the volume of particles and the soil pores. Estimating Bulk density is time-consuming, and PTFs have been developed to help predict Bulk Density. These functions cannot be applied to all possible types of soil. PTFs can be obtained by analysing a specific set of soils with a somewhat similar nature compared to all possible categories. Various methods have been developed and proposed to use PTFs on multiple soil types. These methods describe the kind of data on which PTFs have been developed and the reasoning.

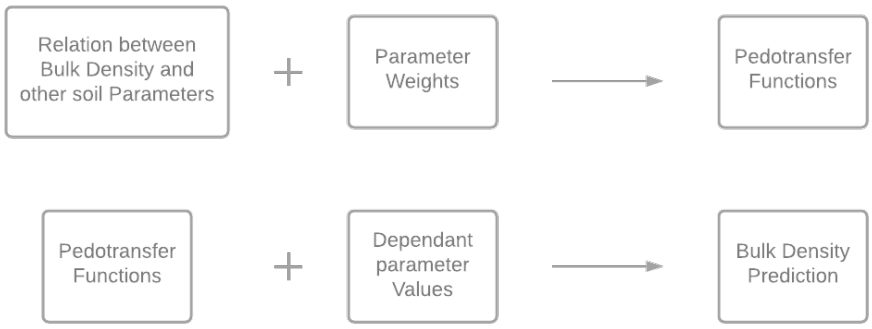

Fig. 6 – Pedo-Transfer functions & Bulk Density Prediction Mechanism

Empirical Model to predict soil Bulk Density

PTFs have been referred to as predictive functions to obtain a property (Bulk Density) based on independent and dependent parameters. This proposed model considers that soil bulk density is a function of soil structure and mineral packing. The considered soil samples contained about 80% of sand (approx.). Limits were set for the bulk density values (g cm^{-3}) between 0.7 and 1.8 since this would help rule out uncommon soil types in Australia. The carbon content % was also capped at 12% for the same reason. This analysis would help obtain the prominent factors affecting soil bulk density (Tranter *et al.*, 2007).

Apart from developing PTFs to estimate soil Bulk Density, the aim was to propose a conceptual model and establish the factors that influence the Bulk Density of Australian soil (Tranter *et al.*, 2007).

The previously interpreted studies found that bulk density was predicted using various factors: the content of clay and silt in the soil, arrangement of particles (packing) in the soil, pore spaces between

soil particles, position, and orientation, etc. This formed the basis of the equation for the conceptual model:

$$\rho_b = \rho_m + \Delta\rho + \varepsilon$$

Fig. 7 – **Bulk Density Pedo-Transfer function** ***(Tranter* et al.*, 2007)***

ρ_b - Bulk Density of the soil (g cm^{-3})

ρ_m - Variation of the Bulk Density of soil as a function of depth and mineral content (mineral matter bulk density) (g cm^{-3})

$\Delta\rho$ - variation in Bulk Density (as a function of the change of soil structure) (g cm^{-3})

ε - residual (g cm^{-3})

Dataset for the Empirical Model

While exploring the data, it was found that ρ_m exhibited a negative correlation (no direct relation) concerning organic carbon content, which previous studies have backed. Organic carbon also plays a vital role in the development of soil structure. Hence the model aims at using PTFs to help predict ρ_m residuals, which would help obtain the Bulk Density. Specific parameters trends were observed and identified using Scatter Plots and considered for further study. The limits set for Bulk Density and Carbon Content are strongly considered.

Subsequently, the model's data was obtained from SALI (Soil and Land Information) and other reliable sources in Australia. Two final datasets were used to train and validate the model's performance. Each set consisted of training and validation data.

Implementation of models

Two novel approaches that had not been used were implemented to help predict Bulk Density - Regression Trees (RTs) and Artificial Neural Networks (ANNs). The conceptual model was compared to the standard traditional models as mentioned above. Additionally, a variation of a second model was also considered (Stewart, Adams, & Abdulla, 1970). An adjustment was made to a previously used relation, where one of the variables was substituted by another - which considered depth and particle size distribution (ρ_m).

All models were trained using a data set and validated using another data set from the same source. The performance of the models was depicted using Root Mean Squared Errors (RMSE) and the R^2 statistic. A higher value of RMSE indicates a higher error value.

Results of Regression Trees and Artificial Neural Networks

- A stepwise analysis could not indicate a well-defined correlation between clay and silt size regarding Bulk Density. Previous research had shown that a well-defined relationship was obtained using silt and clay (Heinonen, 1977). On the contrary, a better performance was seen when the 'depth' parameter was transformed onto the logarithmic scale, which indicated an exponential growth of Bulk Density with an increase in depth from the surface.
- A model built on Multiple Linear Regression (MLR-A) outperformed the Regression Tree and performed comparably with the Artificial Neural Network (ANN) - A Model (the A implies the dataset used on the model's training). The performance was compared using the R^2 statistic (a higher value indicates better performance). **It is vital to note that the inclusion of Organic Carbon data resulted in improved Bulk Density predictions compared to only considering depth and particle size.**

Stepwise Regression also had similar results; Organic Carbon was transformed on the logarithmic scale, which proved to be the best Bulk Density predictor, while hardly any relation was obtained from silt-size and clay.

The models trained using the dataset 'B' also showed similar accuracies in terms of R^2 score and RMSE values, apart from the regression Trees (RT-B). It is also important to note that ANN-B1 (Dataset B is divided into portions - B1 and B2) showed a higher error value and could not fit the data as well as MLR-B, even though the former used more parameters in the model.

Including the clay and silt parameters in the ANN Model resulted in lower model accuracy. This could be because the model was overfitting the data (Overfitting is a term used when a model is not good at generalising the data anymore).

- **The limitations in the accuracy of the regression equations suggest that either alternative better predictors would have to be selected or improve the models used to predict bulk density.**

Adapted Adams-Stewart Model

- The ρ_m component associates Bulk Density with mineral content and depth. This was introduced into this model. It was observed that a maximum Bulk Density corresponded to a high sand content on structureless and loose sediments.
- This proves that the sediments influenced the soil structure (unstructured). A suggestion from previous studies (Koltermann & Gorelick, 1995) was that Bulk Density was observed to reach a maximum when the voids between larger particles were filled in by smaller ones (clay and silt-sized particles)

The Bulk density of organic matter was retained (field condition = 0.225 g cm^3), and the model was only trained on the ρ_m component. On validation, it was seen that the model had a better fit for the data, as compared to the previous cases, but also had the highest errors. Since this model was only trained on a single parameter, it is highly expected that this could perform better on thoroughly trained models and reduce the error (RMSE). This could mainly be applied to a smaller set of data, where the data for the organic matter could be underwhelming to identify relationships with the Bulk Density.

Proposed Conceptual Model

From the set of traditional models, it was observed that MLR-A was the best option concerning depth and particle size. Hence, it was used as the ρ_m substitute for the conceptual model. The trends reflected by the ρ_m component were recorded (Koltermann & Gorelick, 1995), but the soil structure influenced the more considerable variance.

Organic Carbon was found to influence $\Delta\rho$ on the basis that the former enhances the structure of soil via aggregation stability. As a result, a correlation was observed between Carbon and ρ_m residuals. Transforming depth and Carbon data onto the log scale further improved the predictions of $\Delta\rho$.

The model prediction was greatly improved within the limits of 1.2 and 1.6 g/cm^3 of Bulk Density. The model tended to either overestimate or underestimate values beyond these limits. Overall, this model's performance was comparable with other traditional models, even though the latter were much more complex than the conceptual model.

One crucial result proposed from this study was that the negative correlation observed before (between Carbon and Bulk Density) was due to soil aggregation (unstructured soil). In the case of Structured soils, the collection of soil due to the organic matter had a more significant effect on the Bulk Density due to a higher number of pore spaces.

General relations between various soil parameters are well known, but their quantification does not seem practical due to the complex interaction. Morphological Data could provide better descriptions of specific parameters and improve model predictions for Bulk Density compared to Lab-models.

Results

- From the analysis conducted on traditional models, it is seen that more complex models need not necessarily improve the accuracy for the prediction of Bulk Density (MLR model was found to be more accurate and have a better fit, as compared to the ANN and RT models). However, this trend was only observed on smaller datasets, which could be subject to change when the dataset size varies.
- The adapted model had the best fit for the data but also had the highest error. This was because it was only fit concerning a single parameter (ρ_m). While it is inconclusive that ρ_m directly shows improvement, the soil structure data helps reduce the error of the estimate $\Delta\rho$, which helps in improving Bulk Density Predictions.
- It was predicted that the pore space of the soil structure, along with better descriptions, would help in increasing the accuracy of the PTFs for a variety of applications.

Optimizing PTFs using Boosted Regression Trees

- Bulk Density of soil was mainly used for identifying the extent of soil compaction. It isn't very convenient to use older data to help answer novel environmental questions. Older data could have incomplete information regarding Bulk Density, which would have to be predicted again using PTFs. These functions have been based on the soil samples that have been studied to predict Bulk Density (Martin et al., 2009).
- While most functions (PTFs) are meant for specific soil samples, general functions that extend to a more extensive soil range have also been proposed. The parameters considered for the PTFs (for soil Bulk Density Predictions) were mostly Organic carbon content, particle size distributions, and fine-earth and coarse-element bulk densities.
- A new method was tested to help build a PTF. This method is called the MART (Multiple additive Regression Trees) method (Friedman, 2001, 2002; Friedman & Meulman, 2003), which has provided promising results in other areas of science. This method can also handle a comprehensive nature of data - quantitative or qualitative values, missing parameter values, highly correlated predictor features, and is not significantly affected by the presence of outliers.

Datasets

- The data for Bulk Density was obtained from a wide variety of soil samples, which were present in the RMQS Database (Soils of France). This database also had a wide variety of soil types and covered a comprehensive set of physical conditions (Martin *et al.*, 2009).
- Samples were taken from the topsoil layer and the bottom soil layer within a given area of a soil site.
- For each soil site, samples were mixed from each layer to obtain a resulting model and then air-dried.
- From the reference area of each soil site, three bulk density measurements were made from the topsoil and the subsoil (the depth of the topsoil was 30cm from the surface).
- Depending on the nature of the resulting soil sample, various methods were used to measure the bulk density.
- From the data obtained, the following variables were marked for further analysis: depth, organic matter content, silt and clay particle size distribution, gravel %, and the type of layer from where the soil was obtained (subsoil/topsoil).

MART Model

This model (Multiple Additive Regression Trees) was used to predict the value of Bulk Density (output) based on a set of input factors (predictor variables). It works based on regression trees, extending it further using Boosting. The model predicts the output value after the inputs are split into a disjoint region. When observations are poorly predicted, increased weight is given to them, and the model combines them additively, moving forward. This is an iterative process.

This also has increased accuracy compared to conventional decision trees due to Boosting, reducing the extent of overfitting.

Two main factors that affect this model are **tree size** and the **learning rate**. Learning rate is how the model should be changed depending on the error each time the weights are changed. Tree size is the number of nodes present in each tree. Each node is responsible for an outcome based on a condition. The variation in the tree size allows for the inclusion of more variables. Based on the number of times a variable is used as the splitting criteria for each tree, it is assigned an Index (Importance Index).

The model was configured in a way that the best iteration of the model does not affect the entire outcome since there is a set number of total iterations the model could perform.

Comparison of MART and known PTFs

The performance of the MART model was compared with a set of known PTFs (De Vos, Van Meirvenne, Quataert, Deckers, & Muys, 2005). Five different groups of PTFs were considered for the comparison. Variables were used from the RMQS Dataset to predict Bulk Density. This was applied to two models.

Model 1: The variables considered for Bulk Density estimation were: Clay and Silt data and the Organic content of the sample. This model outperformed all the PTF equation groups and was given the label - Model ‘m’.

Model 2: The variables considered for Bulk Density estimation were: Clay and Silt Data, Gravel %, Organic content of the sample, layer of soil and depth. This model was given the label - Model ‘M’.

The number of parameters considered for each model is the main difference between the two.

Validation

Both the models and the known PTFs were validated using two measures – Cross-Validation and Goodness of Fit. The latter measure helped compare the results obtained before. The difference between Observed and Predicted Bulk Densities was calculated, and this formed the basis of several numerical figures - **RMSPE** (Root mean Squared prediction error), **MPE** (mean prediction error) and **SDPE** (standard deviation prediction error). The R^2 estimate, which depicted the linear relationship between these values, was also calculated.

As a part of Cross-Validation, a random number of samples were selected from the Data and used to train the model, and the accuracy of the training data was recorded. The remaining samples constituted the testing data, which would measure the testing accuracy. This entire process was varied with the number of samples in the training and testing datasets, and the RMSPE, MPE and SDPE were noted correspondingly.

Results

- All the PTFs resulted in low R^2 values. On further analysis of each PTF, it was seen that the first and last PTFs had the lowest and highest values. The remaining three PTFs had values that were similar to each other.
- Since the last PTF had the highest $R^{2.}$ Value, it had the best fit for the data. This could be since the silt and clay data were considered.

MART PTFs vs. Equation-Based PTFs

Mart models ‘m’ and ‘M’ had exceedingly high R^2 values of 0.828 and 0.944, which suggest that they fit the data extremely well. The MPE and RMSPE values were shallow, resulting in a far better performance than the Equation-based PTFs.

Model E (the last PTF with the best R^2 value of the lot) produced accurate results for low Bulk Density values but could not accurately have higher Bulk Density Values.

The ‘m’ and ‘M’ models, on the other hand, produced accurate results throughout the entire range of Bulk Density Values.

The MART Model and Equation based PTFs were tested on a data set for another comparison. The latter did not perform well even though they were adjusted to the data before testing. This exposes their low versatile nature.

This was also the case with the MART Models, but they performed better than the standard PTF equations. The former had higher values of positive correlation between observed values and residuals. Model ‘m’, which was only trained with three parameters, performed better than Model ‘M’. Even though the latter performed better, that was attributed to the presence of more variables for prediction. In both models, Organic Content was the first for variable importance.
The MART Model allows the parameters to existing in qualitative and quantitative forms.

Conclusion

- The MART Model was proposed for estimating Bulk Density by developing new PTFs. These were compared with known PTFs.
- The MART Model PTFs (MMPTFs) produced better results with higher accuracy than the known PTFs. The MMPTFs are based on boosted regression trees, are easy to use, and have good prediction capability. Cross-Validation helped validate the results.
- Multiple Regressions are commonly used when the dataset is small and produces accurate results to a reasonable extent. However, more accurate and precise Bulk Density Estimations can be obtained using MART instead of Multiple Regression Algorithms.

Workflow

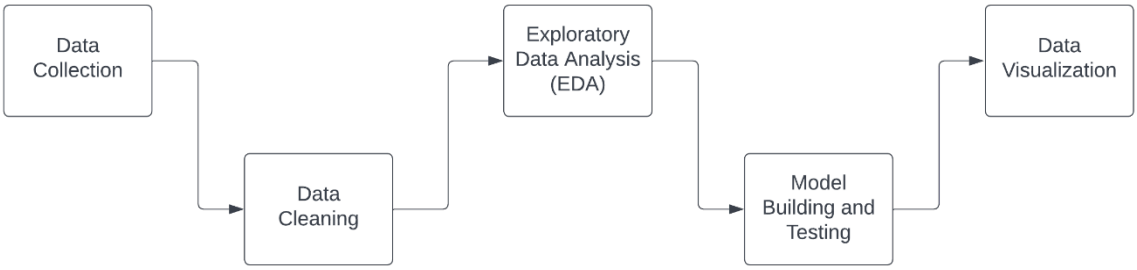

Fig. 8 – **Data Science Process implemented**

Data Collection

The dataset that has been utilised for analysis is from the Soil Data Federator API (Application Programming Interface), which the public can access. The dataset comprises data from various databases operated by multiple sources. They include CSIRO, Governments of the Northern Territory, Queensland, South Australia, Tasmania and Western Australia, and TERN. Various databases have been used by the sources (SALI – Soil and Land Information, ASRIS, etc.) to store relevant soil information and properties. The collected data consists of Australian soils. Data is extracted and used for analysis.

Limiting the extent of soil study to specific locations will not provide a comprehensive picture of the complete scenario. Although information can be obtained, it will not be generalised due to the restricted access to soil (concerning location). Moreover, the soil could also be very similar in these locations, which would provide similar results. Having soil from many sources will increase the diversity of soil types, and it would also help create a more generalised model to help predict soil characteristics. Each soil sample is given a unique identifier to map the parameters for each accordingly.

Although highly relevant, the location of the soil sample is not the only important factor. Other factors, such as the depth mapping of information (depth at which specific soil characteristics are recorded), are highly relevant since soil properties vary with depth. The soil characteristics are uniform and can vary significantly from one layer to another. In the absence of depth mapping, the values of various soil properties would have to be more generalised, which would reduce the accuracy of the analysis. The data contains the corresponding upper and lower depths (as an interval), at which the parameters and values are recorded.

Other essential parts of the metadata are the ‘Extraction Time’, ‘Observed Property’, ‘Unit’ and the Description for Observed Property. The time at which data is collected plays a key role. Without this metadata, it would be hard to estimate when the data has been collected since very old or recent data alone would not provide many insights into the dependence of soil properties on each other. The Observed Properties and Units are the main parts of the metadata since these are used to estimate the existence of a relationship between various soil parameters.

A secondary file is used as the key better to understanding specific technical aspects of the primary soil dataset.

	DataStore	Dataset	Provider	Location_ID	Layer_ID	SampleID	SampleDate	Longitude	Latitude	UpperDepth	...	QualSpatialAc
0	SALI	QLDGovernment	QLDGovernment	QLD_3MC_153_1	2.0	2.0	01-01-1993	151.160172	-24.85381	0	...	
1	SALI	QLDGovernment	QLDGovernment	QLD_3MC_153_1	4.0	4.0	01-01-1993	151.160172	-24.85381	0.15	...	
2	SALI	QLDGovernment	QLDGovernment	QLD_3MC_153_1	5.0	5.0	01-01-1993	151.160172	-24.85381	0.2	...	
3	SALI	QLDGovernment	QLDGovernment	QLD_3MC_153_1	8.0	8.0	01-01-1993	151.160172	-24.85381	0.5	...	
4	SALI	QLDGovernment	QLDGovernment	QLD_3MC_153_1	11.0	11.0	01-01-1993	151.160172	-24.85381	0.8	...	
...	...	...	...	...	...	...	...	...	...	...	...	
2039867	ASRIS	NatSoil	CSIRO	899_SSM_SSM9_1	1.0	3.0	31-01-1991	148.798660	-32.64053	0.05	...	
2039868	ASRIS	NatSoil	CSIRO	899_SSM_SSM9_1	2.0	1.0	31-01-1991	148.798660	-32.64053	0.1	...	
2039869	ASRIS	NatSoil	CSIRO	899_SSM_SSM9_1	3.0	1.0	31-01-1991	148.798660	-32.64053	0.15	...	
2039870	ASRIS	NatSoil	CSIRO	899_SSM_SSM9_1	4.0	1.0	31-01-1991	148.798660	-32.64053	0.25	...	
2039871	ASRIS	NatSoil	CSIRO	899_SSM_SSM9_1	6.0	1.0	31-01-1991	148.798660	-32.64053	0.7	...	

2039872 rows × 28 columns

Fig. 9 (a) **– Primary Dataset (a)**

PropertyType	ObservedProperty	Value	Units	QualCollection	QualSpatialAggregation	QualManagement	QualSpatialAccuracy	ExtractTime
LaboratoryMeasurement	10A1	0.06	%	3.0	2.0	5.0	3.0	2021-10-01T04:39:52
LaboratoryMeasurement	10A1	0.03	%	3.0	2.0	5.0	3.0	2021-10-01T04:39:52
LaboratoryMeasurement	10A1	0.03	%	3.0	2.0	5.0	3.0	2021-10-01T04:39:52
LaboratoryMeasurement	10A1	0.02	%	3.0	2.0	5.0	3.0	2021-10-01T04:39:52
LaboratoryMeasurement	10A1	0.03	%	3.0	2.0	5.0	3.0	2021-10-01T04:39:52
...	...	...	...	...	...	...	...	...
LaboratoryMeasurement	XRD_C_Vm	0	NaN	3.0	2.0	5.0	3.0	2021-10-01T19:30:27
LaboratoryMeasurement	XRD_C_Vm	0	NaN	3.0	2.0	5.0	3.0	2021-10-01T19:30:27
LaboratoryMeasurement	XRD_C_Vm	0	NaN	3.0	2.0	5.0	3.0	2021-10-01T19:30:27
LaboratoryMeasurement	XRD_C_Vm	0	NaN	3.0	2.0	5.0	3.0	2021-10-01T19:30:27
LaboratoryMeasurement	XRD_C_Vm	0	NaN	3.0	2.0	5.0	3.0	2021-10-01T19:30:27

Fig. 9 (b) – **Primary Dataset (b)**

	LAB_METH_CODE	LAB_METH_NAME	LAB_METH_SHORT_NAME	CREATED_BY	CREATION_DATE	LAST_UPDATED_BY	LAST_UPDATE_DATE	UNITS
0	10A1	Total sulfur - X-ray fluorescence	Total S XRF	SALI	11-Mar-03	SALI	11-Mar-03	%
1	10A3	C N S; Dumas	CNS	SALI	22-Jan-07	SALI	22-Jan-07	%
2	10A_NR	Total element - S (%) - Not recorded (CSIRO La...	Total S	CHRISTNG	1-Jun-21	CHRISTNG	1-Jun-21	NaN
3	10B1	Calcium phosphate-extractable sulfur - manual ...	CaPhos Extr S - manual	SALI	11-Mar-03	SALI	11-Mar-03	NaN
4	10B2	Calcium phosphate-extractable sulfur - automat...	CaPhos Extr S - automated	SALI	11-Mar-03	SALI	11-Mar-03	NaN
...	...	...	...	...	...	...	...	...
770	s-23F	TAA - calculated as equivalent pyrite S%	TAA - pyrite equiv	SALIP	28-Jul-05	SALIP	28-Jul-05	%
771	s-23G	TPA - calculated as equivalent pyrite S%	TPA - pyrite equiv	SALIP	28-Jul-05	SALIP	28-Jul-05	%
772	s-23H	TSA - calculated as equivalent pyrite S%	TSA - pyrite equiv	SALIP	28-Jul-05	SALIP	28-Jul-05	%
773	s-23R	Residual acid extractable S after peroxide exp...	RAS - pyrite equiv	SALIP	28-Jul-05	SALIP	28-Jul-05	%
774	s_23H	TSA - calculated as equivalent pyrite S%	TSA - pyrite equiv	SALI	27-Mar-19	SALI	27-Mar-19	%

775 rows × 8 columns

Fig. 10 – **Secondary Dataset (key)**

Descriptions of specific columns:

- Datastore: Name of the database in which soil data is stored.
- Provider: Source of the data
- Location_ID: Unique ID given to each soil, based on their location
- Latitude/Longitude: Coordinates of the given soil
- UpperDepth/LowerDepth: Boundary between which parameter readings are recorded
- ObservedProperty: Soil property, whose value is measured between the given boundaries.
- Value/Unit: Numeric value and corresponding units of the measured property
- LAB_METH_NAME: Name of the Property that is measured.

Data Cleaning

Following the data collection phase is data cleaning. In the entire Data Science Process, this is by far the most time-consuming. It can be defined as the process of identifying missing, inaccurate, or corrupt data from the obtained dataset and modifying or handling this data. Common issues that are identified in the soil dataset are:

- Missing values for certain variables
- Data type inconsistencies
- Inconsistent data observations
- Absence of mandatory constraints
- Corrupt data (erroneous characters in data columns)
- Duplicate data observations
- Irrelevant data observations

The most crucial obstacle for further analysis is that soil properties are not recorded at 'regular' intervals. Various data providers have recorded multiple properties at different intervals, which are inconsistent between providers. The relevance of properties to each other can only be studied and concluded if there are standard intervals that are used consistently throughout the study. This causes a slight deviation, where standard intervals must be established, and values at these intervals (if missing) should be recorded. This stems from the fact that soil data varies continuously with depth.

The main goal of the data cleaning phase is to identify and handle such issues with the dataset to ensure that a usable dataset is obtained for EDA purposes and further analysis. Improper data cleaning can lead to inaccurate results, most likely resulting in disastrous implications.

	DataStore	Dataset	Provider	Location_ID	Layer_ID	SampleID	SampleDate	Longitude	Latitude	UpperDepth	...	QualSpatialAccur
93	SALI	QLDGovernment	QLDGovernment	QLD_ABC_111_1	30.0	30.0	01-01-1981	NaN	NaN	NaN	...	
97	SALI	QLDGovernment	QLDGovernment	QLD_ABC_116_1	30.0	30.0	01-01-1981	NaN	NaN	NaN	...	
103	SALI	QLDGovernment	QLDGovernment	QLD_ABC_117_1	30.0	30.0	01-01-1981	NaN	NaN	NaN	...	
106	SALI	QLDGovernment	QLDGovernment	QLD_ABC_142_1	30.0	30.0	01-01-1981	NaN	NaN	NaN	...	
112	SALI	QLDGovernment	QLDGovernment	QLD_ABC_149_1	30.0	30.0	01-01-1981	NaN	NaN	NaN	...	
...	...	...	...	...	...	...	...	...	...	...	...	
2017683	ASRIS	NatSoil	CSIRO	599_WIM_P397_1	5.0	1.0	29-09-1958	NaN	NaN	0.43	...	
2017684	ASRIS	NatSoil	CSIRO	599_WIM_P397_1	6.0	1.0	29-09-1958	NaN	NaN	0.74	...	
2017685	ASRIS	NatSoil	CSIRO	599_WIM_P397_1	7.0	1.0	29-09-1958	NaN	NaN	0.96	...	
2017686	ASRIS	NatSoil	CSIRO	599_WIM_P397_1	8.0	1.0	29-09-1958	NaN	NaN	1.22	...	
2017687	ASRIS	NatSoil	CSIRO	599_WIM_P397_1	9.0	1.0	29-09-1958	NaN	NaN	1.37	...	

102756 rows × 28 columns

Fig. 11 – **Subset of data containing NaN (missing) values**

Exploratory Data Analysis (EDA)

The dataset that is used for EDA is from the Soil Data Federator. It is a consolidated dataset from various providers and consists of a wide variety of soil data. This data is collected from multiple soil samples throughout Australia.

The soil data is listed according to various properties, which correspond to soil parameters and contain additional information such as region of occurrence (UpperDepth – LowerDepth = region of occurrence) and coordinates of the soil (Latitude, Longitude).

- The primary parameter of interest is Bulk Density. Measuring Bulk Density values is not very practical since it is time-consuming and expensive.
- Based on the nature of Bulk Density Data, a suitable model can be used to obtain the Pedo-Transfer Function.

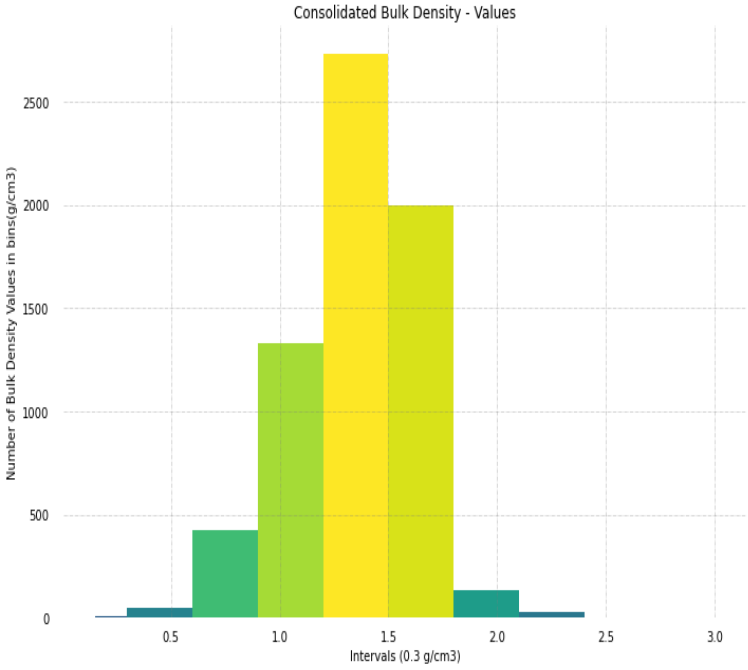

Fig. 12 – **Number of occurrences of Bulk Density (numerical)**

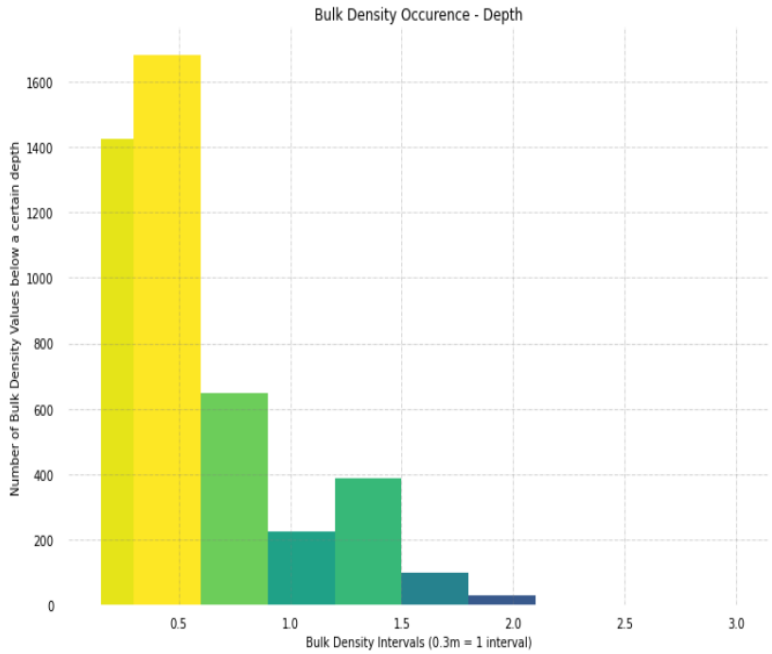

Fig. 13 – **Intervals at which Bulk Density is recorded**

- From the distribution of Bulk Density values, the majority is concentrated at 0.9-1.8 (g/cm^3) from various locations.
- A significant part of the Bulk Density dataset is obtained from the soil within 1m of its surface.
- Based on this information from the Bulk Density Data itself, a suitable algorithm can be used to analyze this information, which can be obtained from related research, to assist in creating a Pedo-Transfer function for future predictions.

Regions of occurrence of Bulk Density (within a range of values):

Fig. 14 (a) – **Bulk Density < = 0.3 g/cm^3**

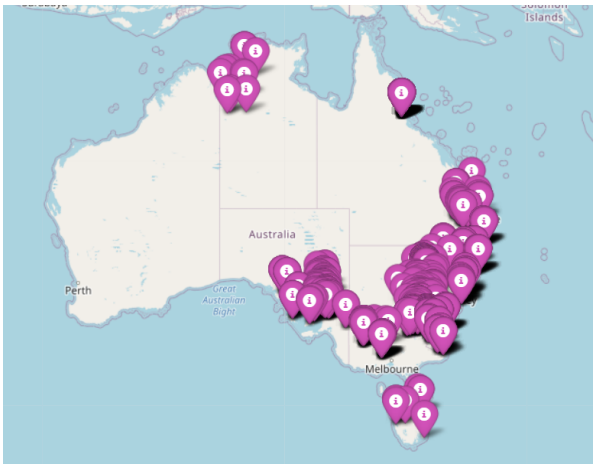

Fig. 14 (b) – **0.3 gm/cm^3 < Bulk Density < = 0.6 g/cm^3**

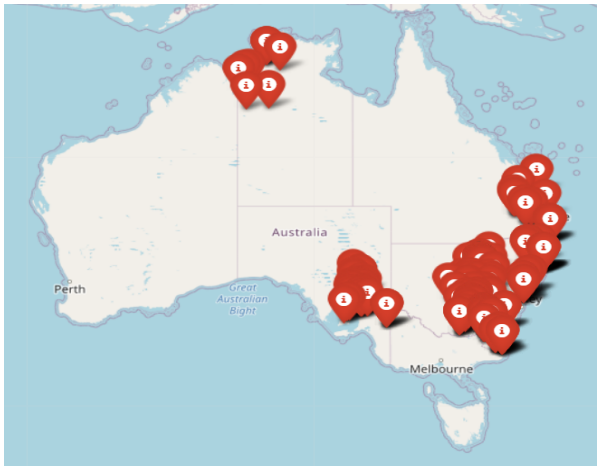

Fig. 14 (c) – **0.6 gm/cm^3 < Bulk Density < = 0.9 g/cm^3**

Fig. 14 (d) – **0.9 gm/cm^3 < Bulk Density < = 1.2 g/cm^3**

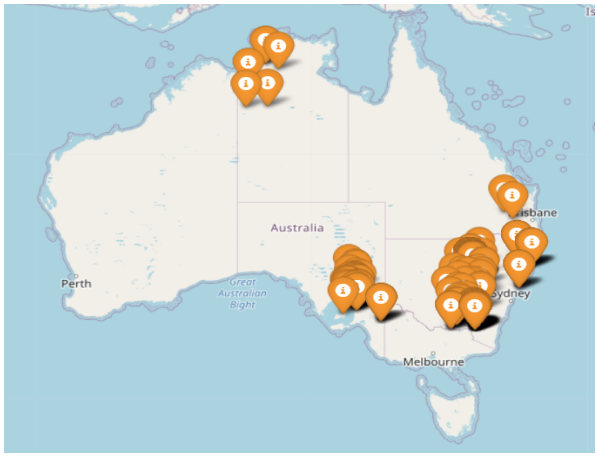

Fig. 14 (e) – **1.2 gm/cm^3 < Bulk Density < = 1.5 g/cm^3**

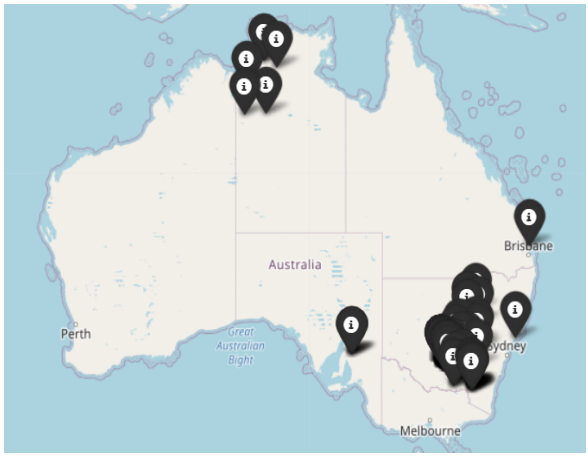

Fig. 14 (f) – **1.2 gm/cm^3 < Bulk Density < = 1.5 g/cm^3**

The literature survey provided an insight into the relation between Bulk Density, clay content and organic carbon. There was a relatively strong relationship between Bulk Density and the two variables.

Clay Content

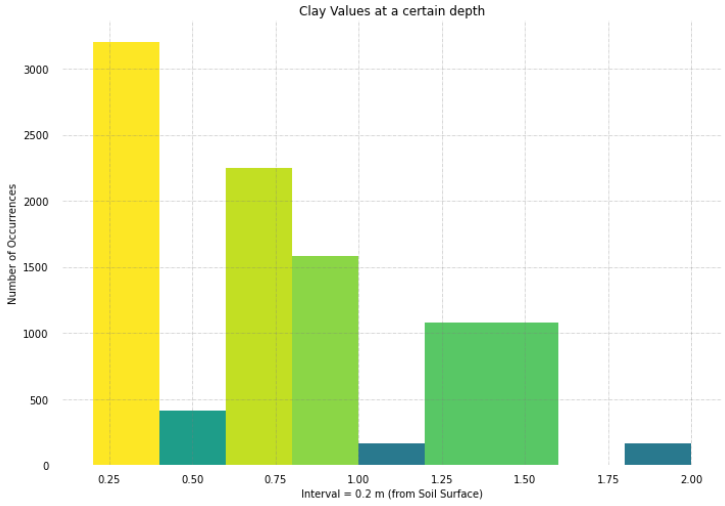

Fig. 15 – **Recorded Intervals of Clay Content in Soil**

The above histogram provides a view of the distribution of Clay in the soil.

- The relationship between Bulk Density and clay content in the soil can be further analysed to estimate whether the latter can help predict Bulk Density. This can be done using the Pearson Correlation test.
- If a high value (close to 1.0) is obtained for the same, Clay content can be labelled as a predictor for Bulk Density and used in the Pedo-Transfer functions.

- Clay content has a higher occurrence within 1m of the soil (from the surface). The trend between Clay and Bulk Density can be observed to conclude whether its effect slowly reduces towards increasing depth from the soil surface.

Organic Carbon

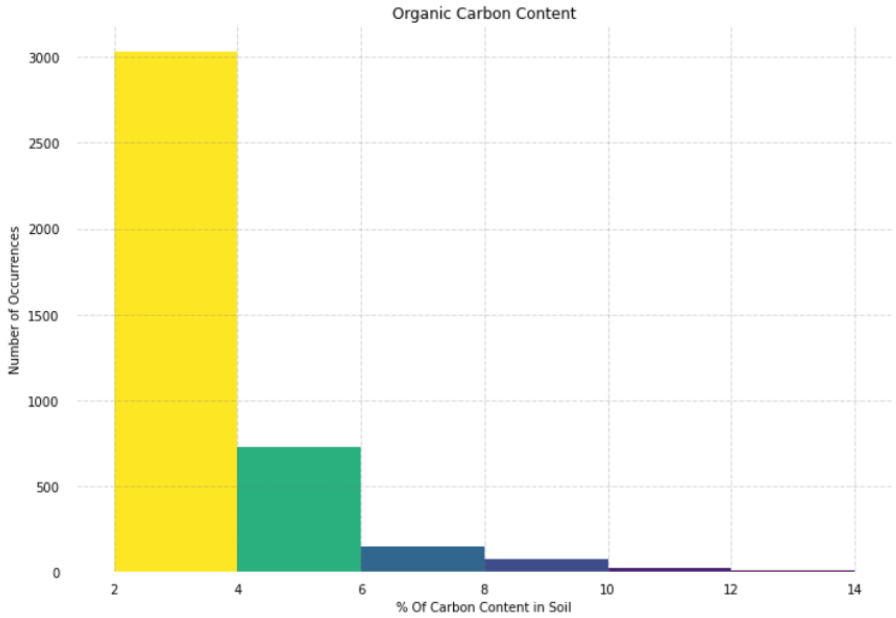

Fig. 16 (a) – **Intervals of Percentage of Organic Carbon Content in Soil**

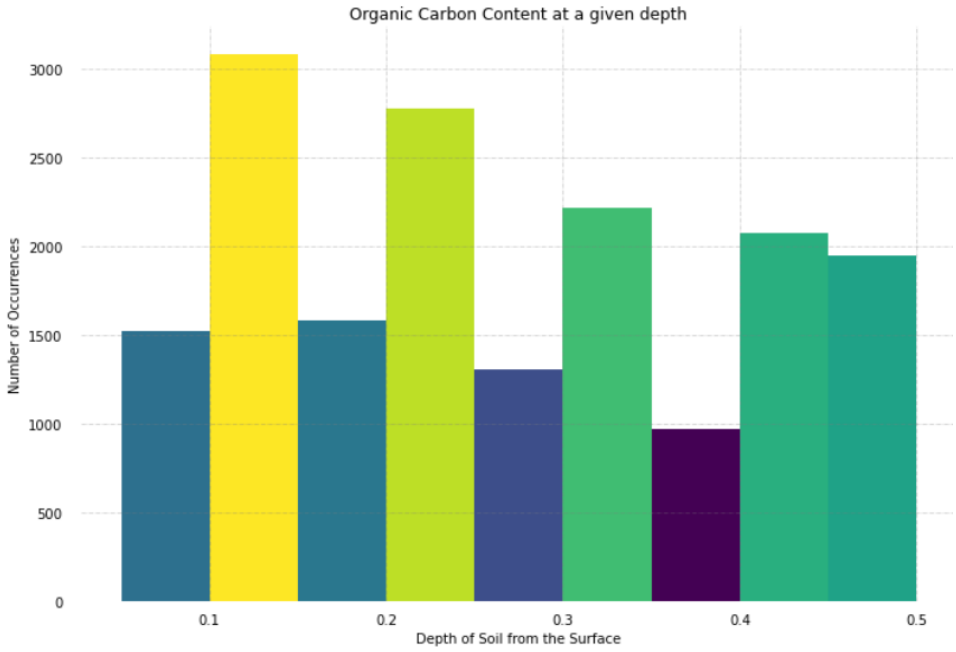

Fig. 16 (b)– **Recorded Intervals of Organic Carbon Content (%) in Soil**

- From the histograms, the carbon content is seen to occur in the top layer of soil, and it can be implied that carbon content might not play such a key role in Bulk Density Predictions as we move more profound in the soil layer.

Properties such as Air-water moisture content (5A2) and Electric Conductivity (3A1) have the most recorded observations in the soil dataset compared to other soil parameters. Therefore, there is a good chance that Bulk Density would be related to these variables.

Air-Dry Moisture Content (%)

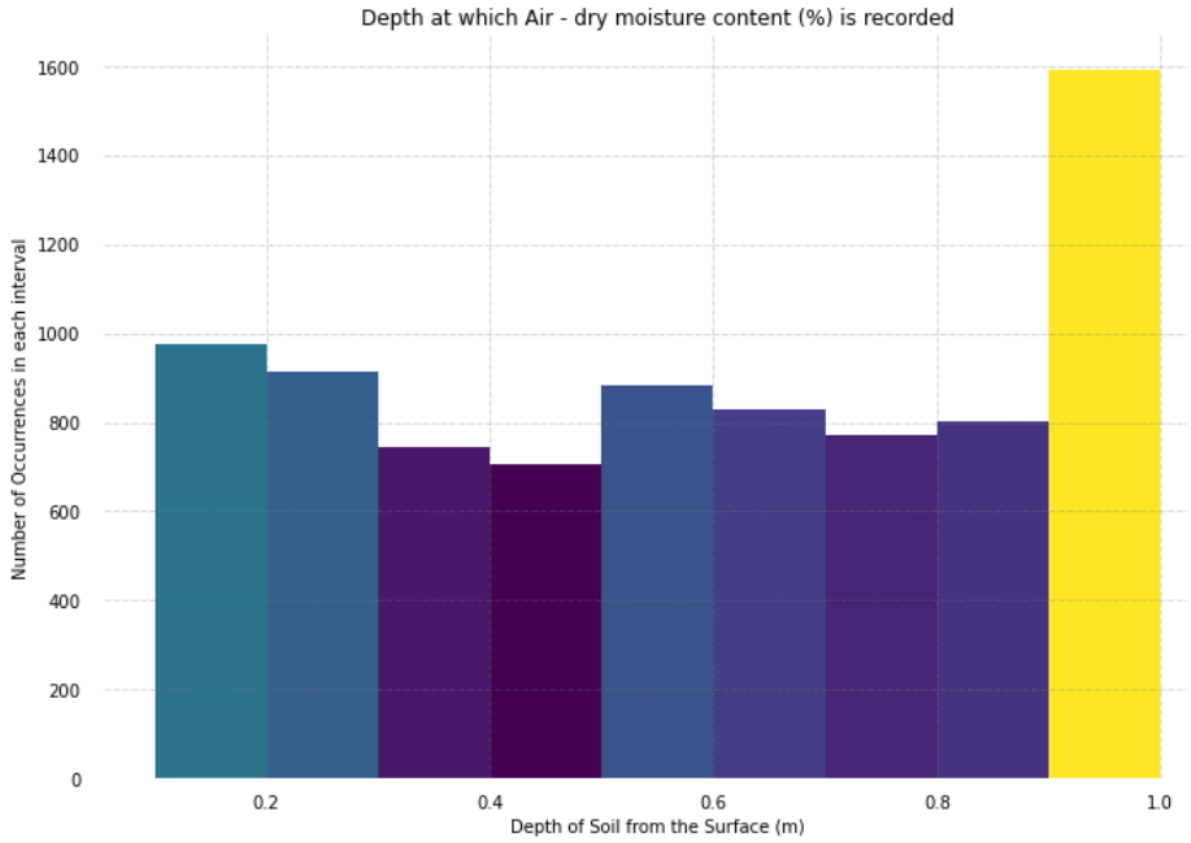

Fig. 17 – **Recorded Intervals of Air – dry moisture content (%) in Soil**

- From the distribution of air-dry moisture content, there is a high possibility that the value of bulk density is directly related to the occurrence of the dry moisture content since it increases the relative density of the soil, thereby increasing the Bulk Density.

Electrical Conductivity (EC) – 1:5 soil/water extract (mg/kg)

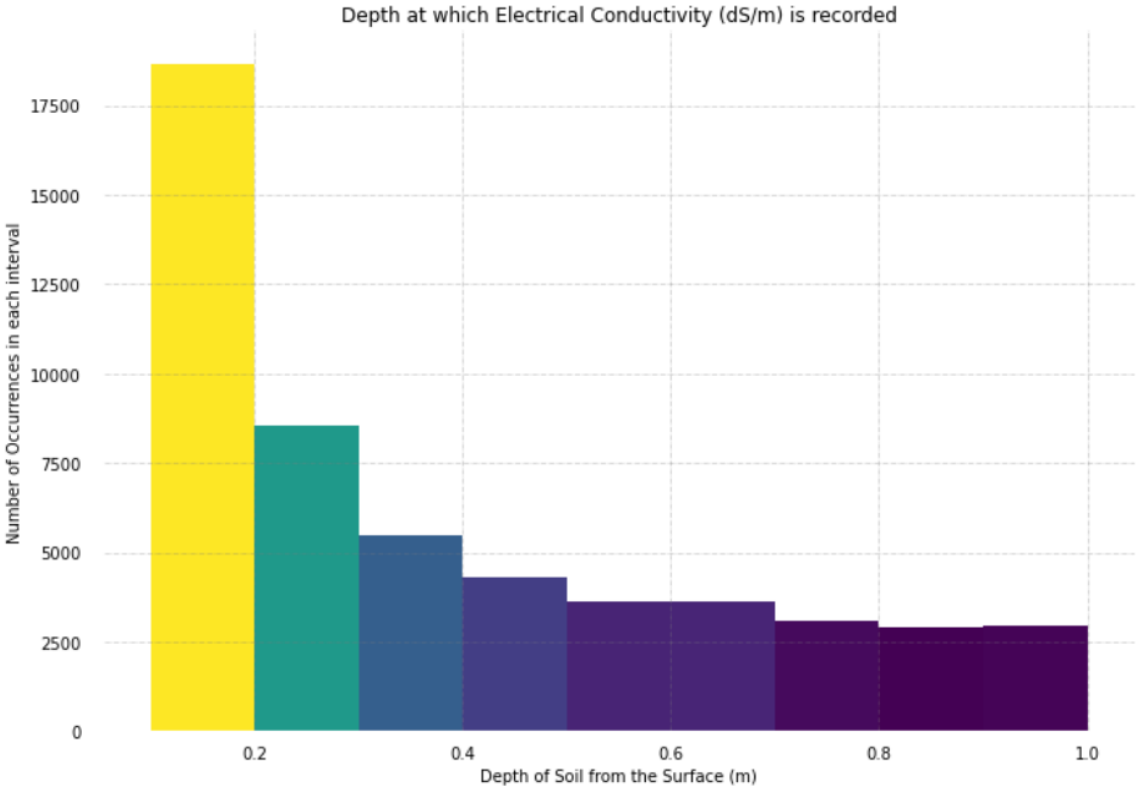

Fig. 18 (a) – **Recorded Intervals of Electrical Conductivity (dS/m) in soil**

- Electrical conductivity in soils decreases with depth, which would mostly be inversely related to the Bulk Density of soil since the Bulk Density of soil usually increases with depth as the number of particles per unit volume increases.

Bulk Density vs. Input Variables

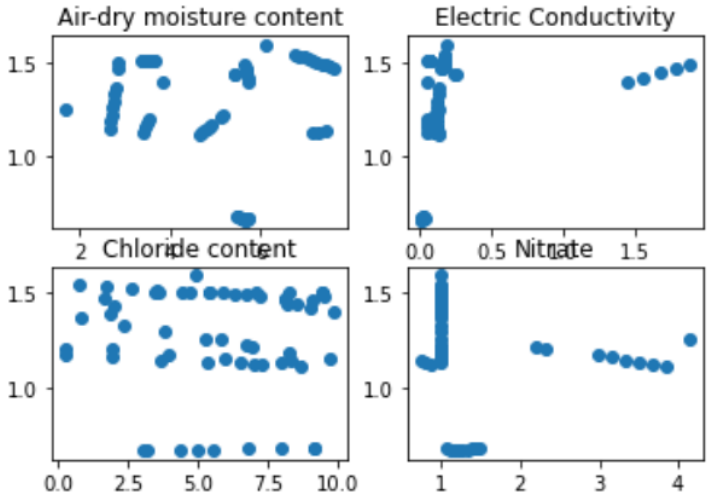

Fig. 18 (b) – **Bulk Density (y-axis) vs. Input Variables (x-axis)**

Model Building and Testing

Before a Machine Learning Model can be applied to our dataset to obtain the Pedo-Transfer function, the primary criteria is to ensure that all soil properties are recorded in the same intervals.

Latitude	Longitude	UpperDepth	LowerDepth	Value	UNITS
-24.85381	151.160172	0	0.05	6.2	%
-24.85381	151.160172	0.15	0.2	7.7	%
-24.85381	151.160172	0.2	0.3	7.5	%
-24.85381	151.160172	0.5	0.6	8.3	%
-24.85381	151.160172	0.8	0.9	10.4	%

Fig. 19 (a) – **Table of Air – dry moisture content with values at specified intervals**

Latitude	Longitude	UpperDepth	LowerDepth	Value	UNITS
-24.85381	151.160172	0	0.05	28	mg/kg
-24.85381	151.160172	0.05	0.15	25	mg/kg
-24.85381	151.160172	0.15	0.2	25	mg/kg
-24.85381	151.160172	0.2	0.3	25	mg/kg
-24.85381	151.160172	0.3	0.4	73	mg/kg

Fig. 19 (b) – **Table of Chloride content with values at specified intervals**

From fig. 19 (a) and fig. 19 (b), the intervals of the record parameters are not consistent. To obtain data at a specific soil depth interval can be considered an application of interpolation of data between upper and lower limits of soil depth. When it comes to interpolation, splines are a useful mathematical tool to solve interpolation-related problems.

Splines

Piecewise functions are defined by various smaller functions, where each function is applied to a different portion of the domain. This is a way to express the function itself rather than its characteristic. If a property holds piecewise (for a function), then it is implied that the domain can be split into various intervals. This further indicates that this is a property of the function itself.

Piecewise functions are continuous in an interval if the subsequent conditions are satisfied:

- No discontinuity is present at the given endpoints of the subdomain.
- Constituent functions of the piecewise function are continuous in each sub-domain.

A spline is a special function in mathematics defined by polynomials (piecewise). The spline corresponds with a polynomial in every interval throughout the entire domain. In the case of interpolation, spline interpolation is the preferred method over polynomial interpolation in the field of soil science (Bishop, McBratney, & Laslett, 1999). It is preferred to higher degree polynomials and yields similar results in the case of lower degree polynomials. Splines can be applied to single or multi-dimensional data.

Equal Area Spline (Mass-Preserving)

Properties of soil vary with an increase in depth from the surface. Soil data is recorded on the basis of soil horizons, but there are cases where values at arbitrary depths need to be estimated. **A soil Horizon is defined as a layer of soil parallel to the soil surface whose properties (physical, chemical, and biological) vary from the layers above and below it.** Additionally, properties seem to be relatively consistent across a soil horizon but can differ vastly between horizons. As depicted in fig. 20, Horizons are defined in terms of color and texture.

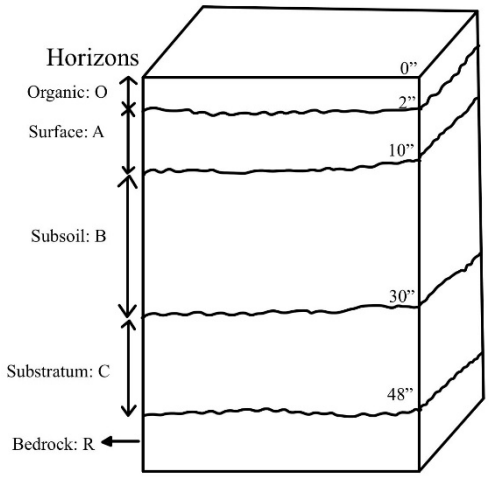

Fig. 20 – **Soil Horizons**

The equal-area spline is a variation of the normal spline function. This was proposed in order to account for the damping effect of incorporating the soil horizon data, to model depth functions (Ponce-Hernandez et al., 1986). The main properties of the equal-area spline are:

- It is composed of quadratic polynomials, where the knots are located at the boundaries of the horizons.
- For any given horizon, the area of the left portion of the spline above the horizon (X), is equal to the area of the right portion of the spline below the horizon (Y). This ensures that the average horizon value is constant.

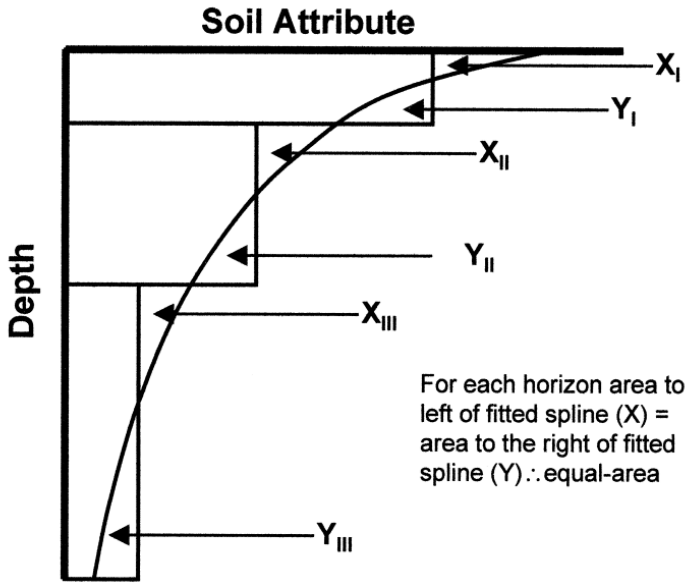

Fig. 21 – **Equal Area Spline (Ponce-Hernandez, Marriott, & Beckett, 1986)**

An attempt was made to check for any improvements regarding the quality of predictions of the equal-area spline (Ponce-Hernandez et al., 1986). This was done using the data from the selected soil horizon, as well as the samples from the soil profiles above and below it. Additional soil samples would fix boundary conditions and would thus help in improving the predictions at the end of the soil profile near its boundaries. Additionally, an extra sample was recorded from the top of the soil profile for specific soil attributes where the most variation existed beside the soil surface. This was done to improve the fit. There was no room for error in the data.

An additional feature of the spline function is its **mass-preserving** nature (the original data is stored, and the integration of the continuous spline can retrieve it). The spline parameters are the soil attribute values at standard depths (user-specified).

The mass-preserving spline was fit to a single soil profile with a specific parameter. The spline can only be used for interpolation and values at any depth that exceeds the maximum soil depth cannot be obtained. The following data is required to fit the spline on a single soil profile:

- Soil Profile Number
- Upper Boundary
- Lower Boundary
- Numerical values at each boundary

	SoilProfile	UpperDepth	LowerDepth	float_values
1	30	0	10	0.06
2	30	20	30	0.10
3	30	50	60	0.18
4	30	80	90	0.38
5	30	110	120	0.62

***Fig. 22* – Data frame incorporated into the Mass-Preserving Spline**

- From the above table, the gap in the soil depth for a single soil profile can be seen.
- The mass-preserving spline predicts a continuous function from the start of the soil profile till the maximum depth is reached. During this, interpolations occur within observed depths and at the regions where no observations are recorded.

The default parameters of the spline function can be accepted, but they are modified in this case. The lambda parameter controls the spline smoothness. The spline becomes more rigid if lambda increases and becomes smoother as lambda approaches 0. An optimal lambda value of 0.1 works for most soil properties (default value) (Malone, Minasny, & McBratney, 2017) .

Another parameter of the spline function is the depth parameter. It is the desired intervals where soil values need to be recorded. These values can be entered as per the user's requirements. The Spline function is first fit to the data, and the integral of the function helps obtain the values of the soil parameter at the desired depths.

```
> str(eaFit)
List of 4
 $ harmonised    :'data.frame': 1 obs. of  9 variables:
  ..$ id         : num 30
  ..$ 0-5 cm     : num 0.0587
  ..$ 5-15 cm    : num 0.0694
  ..$ 15-30 cm   : num 0.0959
  ..$ 30-60 cm   : num 0.154
  ..$ 60-90 cm   : num 0.315
  ..$ 90-130 cm  : num 0.542
  ..$ 130-150 cm: num NA
  ..$ soil depth: num 120
 $ obs.preds     :'data.frame': 5 obs. of  6 variables:
  ..$ SoilProfile : num [1:5] 30 30 30 30 30
  ..$ UpperDepth  : num [1:5] 0 20 50 80 110
  ..$ LowerDepth  : num [1:5] 10 30 60 90 120
  ..$ float_values: num [1:5] 0.06 0.1 0.18 0.38 0.62
  ..$ predicted   : num [1:5] 0.061 0.1 0.182 0.381 0.616
  ..$ FID         : num [1:5] 1 1 1 1 1
 $ splineFitError:'data.frame': 1 obs. of  2 variables:
  ..$ rmse   : num 0.00219
  ..$ rmseiqr: num 0.00781
 $ var.1cm       : num [1:150, 1] 0.0577 0.058 0.0585 0.0592 0.0602 ...
```

Fig. 23 – **Mass Preserving Spline fit per soil Profile (Information)**

```
> eaFit[1]
$harmonised
  id    0-5 cm    5-15 cm   15-30 cm  30-60 cm 60-90 cm 90-130 cm 130-150 cm soil depth
1 30 0.0587063 0.06940554 0.09585319 0.1538503 0.314844 0.5417726         NA        120

> eaFit[2]
$obs.preds
  SoilProfile UpperDepth LowerDepth float_values  predicted FID
1          30          0         10         0.06 0.06104895   1
2          30         20         30         0.10 0.10020989   1
3          30         50         60         0.18 0.18212566   1
4          30         80         90         0.38 0.38081014   1
5          30        110        120         0.62 0.61580537   1

> eaFit[3]
$splineFitError
         rmse     rmseiqr
1 0.002186962 0.007810579

> eaFit[4]
$var.1cm
             [,1]
  [1,] 0.05765736
  [2,] 0.05797204
  [3,] 0.05849651
  [4,] 0.05923077
  [5,] 0.06017482
  [6,] 0.06132866
  [7,] 0.06269229
  [8,] 0.06426571
  [9,] 0.06604892
 [10,] 0.06804191
 [11,] 0.07013980
 [12,] 0.07223769
 [13,] 0.07433558
 [14,] 0.07643347
 [15,] 0.07853136
 [16,] 0.08062925
 [17,] 0.08272714
 [18,] 0.08482503
 [19,] 0.08692292
 [20,] 0.08902082
 [21,] 0.09113969
 [22,] 0.09330055
 [23,] 0.09550338
```

Fig. 24 – **Mass Preserving Spline List – Split Components**

The *plot_ea_spline* function is used to fit the mass-preserving spline to the soil profile. The **'eaFit'** variable is the output of the spline that has been fit to the data. The output of the spline function is a list. The list consists of 4 components:

- eaFit[1] – A Harmonized Data frame, which consists of the spline predicted estimates at the mentioned depth intervals.
- eaFit[2] – A data frame consisting of the soil data and the spline predictions at the actual observation depths for the soil Profile.
- eaFit[3] – Stores the value of the root mean squared error (RMSE) of the spline fit, for the given soil profile. This represents the error magnitude between the actual observed and predicted values. This helps in gauging the performance of the spline. A lower RMSE value indicates higher accuracy of predictions.
- eaFit[4] – A matrix that stores the spline predictions. The resolution is set to '1 cm'. The first column represents the depth from the soil surface in increments of 1 cm, and the second column represents the predicted values of the spline, either up to the maximum depth or the depth till which values need to be extracted.

The 'plot_ea_spline' function is used to observe the spline's performance and fit on the data. This function does not have much control over plotting parameters. There are three possible plots that can be obtained from this function.

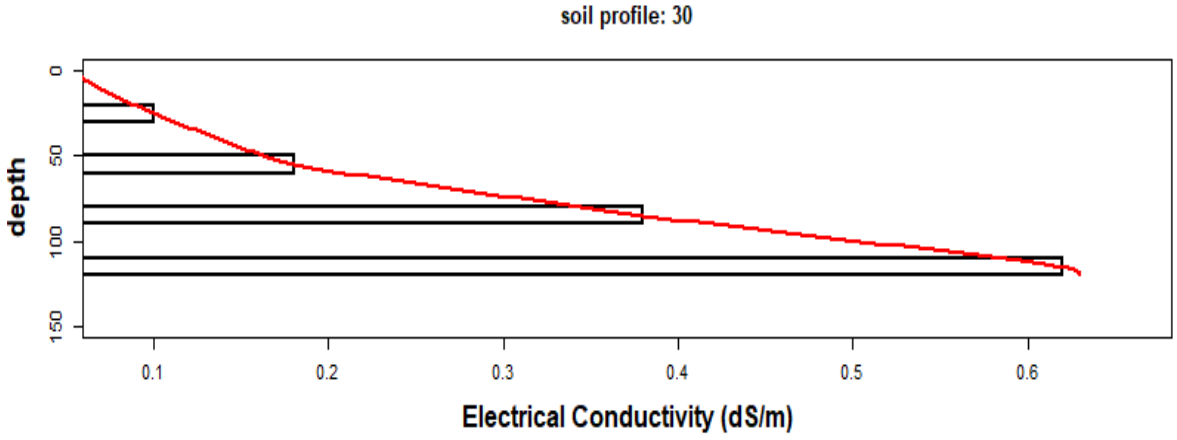

Fig. 25 (a) – **Spline Plot (a)**

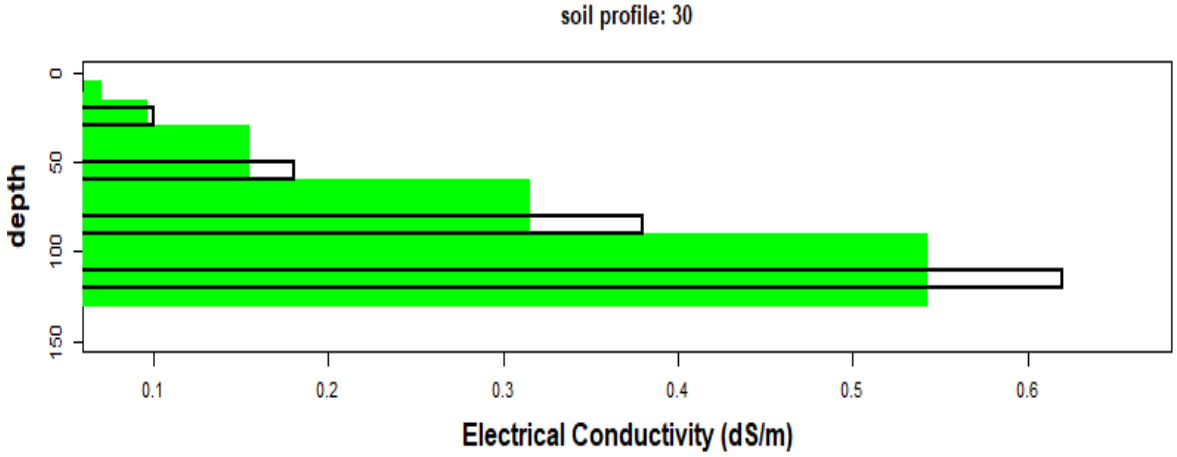

Fig. 25 (b) – Spline Plot (b)

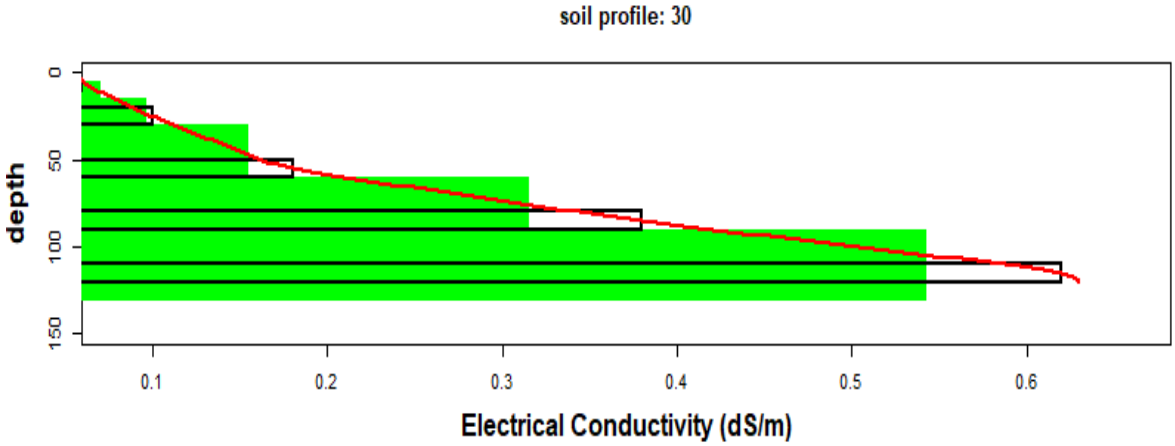

Fig. 25 (c) – Spline Plot (c)

- *Fig. 25 (a)* – Spline plot that represents the soil data + continuous spline.

- *Fig. 25 (b)* – Spline plot that returns the soil data and the averages of the spline at the specified depth intervals.

- *Fig. 25 (c)* – Spline plot that returns the soil data, continuous spline and the averages of the spline at the specified depth intervals.

Machine Learning Models

The dataset now has observations for all soil parameters in increments of 1 cm. Now that the dataset is more uniform, machine learning models can help obtain the relationship between Bulk Density and other soil parameters. Additionally, these models can also help obtain Pedo-Transfer functions.

In Machine Learning (ML), a model represents the output of an algorithm that has been executed on data. It can help identify underlying patterns from a dataset. A Machine Learning model leverages data to increase performance efficiency on a specific task/set of tasks.

- The ML model is trained on a set of data called the training dataset. The training dataset is a subset of the original dataset (usually consists of around 70-80% of the original dataset). This dataset is provided to the model, from which patterns or predictions are to be made.
- The remaining portion of the original dataset is labelled as the testing data, which is kept unseen from the Machine Learning Model.
- A method called Cross-Validation may be performed to estimate the ability of an ML model. A standard method of Cross-Validation (CV) is k-fold CV (k – a number greater than 0).
- Once the model has been fit to the training data, the algorithm identifies patterns in the data.
- The trained model can be applied to the testing data to evaluate model performance. Evaluation metrics that are commonly used are accuracy (%), RMSE (Root Mean Squared Error), and R^2 (goodness of fit).

The process of training an ML Model depends on the nature of the Algorithm. Machine Learning algorithms can be of 3 types:

- Supervised Learning
    - It is a subcategory of Machine Learning that incorporates labelled datasets to train ML algorithms. The algorithms can be used to classify or predict data.
    - Training data consisting of training examples are used to infer a function. Each training example consists of a vector (input variables) and the corresponding output value.
    - The learning algorithm analyses such data from the training examples to map a function, which can then be used to predict new examples.
    - In an ideal case, the algorithm correctly predicts the output for unseen examples. The learning algorithm is able to generalize from the training data and apply it to unseen data.
    - The dataset used for training usually consists of around 70-80% of the data, while the testing dataset is made up of the remaining data from the original dataset.

- Once the model is fit to the training data, it can then be used to predict samples from the testing data. We can estimate the model's accuracy by comparing the predicted and actual values in the test dataset.

- Unsupervised Learning
  - It is a subcategory of Machine Learning that can identify underlying patterns from unlabeled and unclassified data.
  - The primary goal of the algorithm is to categorize unsorted information according to any patterns or similarities without any prior data training.
  - The model acts on the training dataset, without any supervision from the user.
  - Common Classification or Regression problems cannot make use of unsupervised learning.

- Reinforcement Learning
  - It is a subcategory of Machine Learning where the model learns according to its feedback.
  - The primary goal is to identify the best behavior in any particular situation.
  - In the case of supervised learning, the training dataset consists of outputs and hence the model gets trained with the result. However, training datasets provided to reinforcement learning algorithms do not contain the output, but the algorithm's performance depends on the feedback provided by the reinforcement agent. The types of reinforcement provided to the model can either be positive or negative, where each has its own benefit.

Since numeric values of Bulk Density must be predicted, this falls under the category of Regression. Four methods of Regression that have been used are:

- Multiple Linear Regression
- Regression Tree
- Multiple Additive Regression Trees (MART)
- Random Forest Regression.

The performance of the models (Random Forest and MART) can be improved by tuning their hyperparameters. This is done using GridSearchCV.

Hyperparameter optimisation using GridSearchCV

Many Machine Learning Algorithms, such as Decision Trees and Random, are represented by model parameters. Training such models effectively involve selecting the optimal values for each hyperparameter. The learning algorithm incorporates these optimal values to map features (explanatory variables) to the target variable(s).

Hyperparameters (top-level parameters) are the parameters that influence the learning process and are responsible for determining the model parameter values. The hyperparameter values are set for the algorithm before the model is trained. The values of the hyperparameters cannot be altered during the model learning or training phase and are hence considered external to the model. The hyperparameters are incorporated during the learning phase, but it does not form a part of the model. The values of hyperparameters that are used to train a model cannot be extracted from it.

Examples of hyperparameters include:

- Split ratio (train-test)
- Selection of loss function for the model
- Hidden layers in a Neural Network
- Activation function of a Neural Network

Optimum hyperparameter values help in producing the best prediction results from the model. GridSearch implements different combinations of all hyperparameters values of the model and assesses its performance. This results in a time-consuming process based on the number of hyperparameters of the model.

Cross-Validation (k-fold) is also performed as a part of GridSearch and is hence known as GridSearchCV.

GridSearchCV provides a value for the model's R^2 metric (performance measure/goodness of fit). The performance of the model can be compared before and after implementing GridSearchCV.

Multiple Linear Regression (MLR)

Multiple Linear Regression is a statistical technique used to predict the value of a variable based on the values of two or more explanatory variables (if there is only a single explanatory variable, it would be referred to as Linear Regression). In this case, MLR is used to predict Bulk Density values based on the values of other soil properties. It is also referred to as Multiple Regression. It falls under the category of supervised learning.

The primary goal of regression is to model the underlying linear relationship between the independent variables (explanatory variable) and the dependent variable (response variable). The dependent variable may exhibit a linear relationship with two or more input variables. It can also be non-linear.

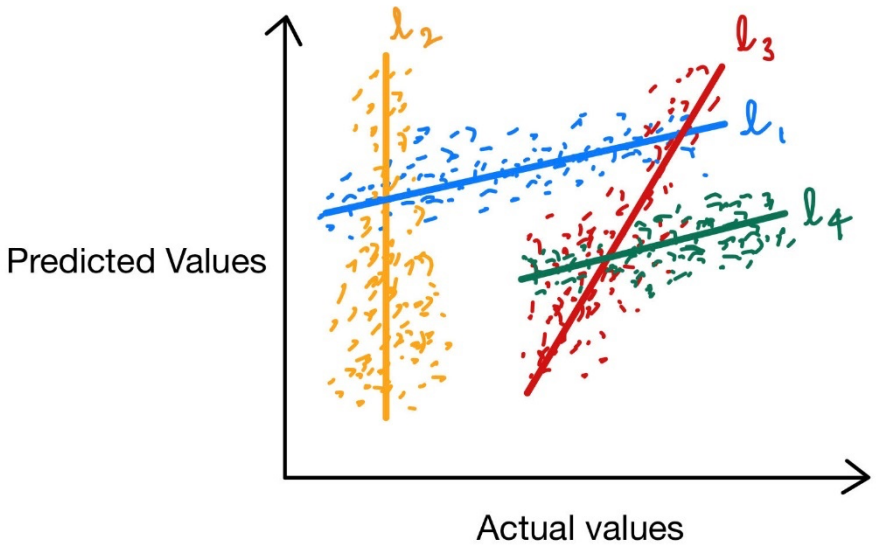

Fig. 26 – **Graph depicting linear relationships with multiple variables**

$$y_i = \beta_0 + \beta_1 x_{1,i} + \beta_2 x_{2,i} + \cdots + \beta_n x_{n,i} + c$$

Fig. 27 – **General Multiple Linear Regression Formula**

y_i - the observation at the 'i^{th}' position, where 0 <= i <= n (there are a total of 'n' observations)

β_0 – y-intercept

β_i – Slope coefficient for each independent variable

x_i – 'i^{th}' independent variable

c – residual term of the model

Multiple Linear Regression is based on the following assumptions (Nathans, Oswald, & Nimon, 2012):

a) There exists a linear relationship between the independent and dependent variables
   - The ideal way to visualise the relationship would be to create scatter plots and inspect them for linearity.
   - If no linear relationship exists between the independent and dependent variables, then non-linear regression can be performed.

b) There is no high correlation between the independent variables
   - Multicollinearity occurs when the independent (explanatory) variables exhibit a high correlation with each other.
   - In the case of multicollinearity, a problem will arise in identifying the specific independent variable that could contribute to the variance in the dependent variable.

c) Constant variance of the residuals
   - At each point in the linear model, the assumption is that the amount of error is similar (homoscedasticity).
   - This can be verified by plotting the standardised residuals against predicted values (scatterplot).

d) Observation Independence
   - The MLR model assumes that each observation in the dataset is independent of the others (independence of residual values).

e) Multivariate Normality
   - When residuals have a normal distribution, it is called Multivariate Normality.

The distribution of the residuals can be viewed using a histogram with a Normal Probability plot.

Regression Tree (RT)

A decision tree is an algorithm in Machine Learning that can be used for the purpose of either classification or regression. Falls under the category of Supervised Learning. A decision tree is a hierarchical structure which consists of nodes and edges. Edges are formed between the nodes of a decision tree. A decision tree imposes a condition at each node, which influences the final outcome.

An outcome is obtained when a leaf node is reached (does not have any children). A regression tree is a decision tree that is used for the purpose of regression, which predicts continuous values as the output instead of discrete values. MSE (mean squared error) is the standard metric to traverse from one node to another. MSE is the deviation of the predictions as compared to the original value.

$$MSE = \frac{1}{n} * \sum (observed - predicted)^2$$

Fig. 28 – **Mean Squared Error**

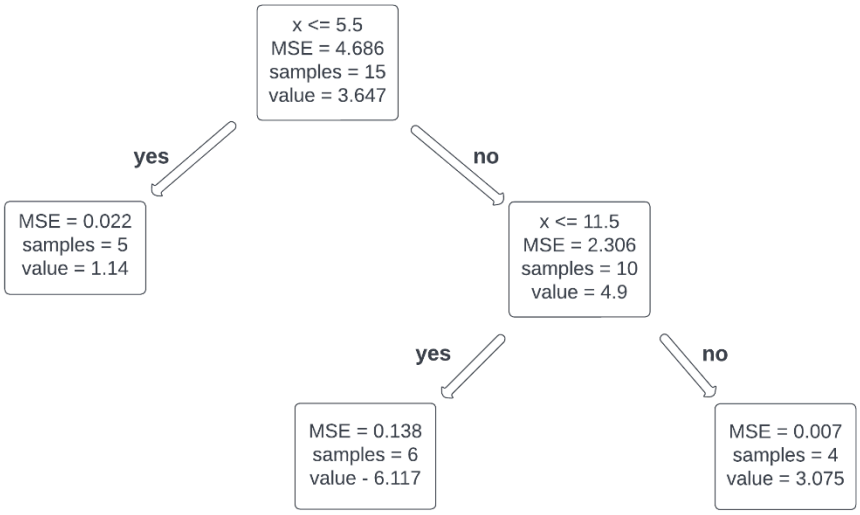

Fig. 29 – **Example of a Regression Tree and Node Information**

- The main goal of the algorithm is to find the point where the dataset can be split into different parts (MSE is minimised at that point). This occurs repetitively, and hence a tree is formed.
- The process of splitting the dataset continues, until it is no longer possible.

Overfitting and stopping criteria:

Overfitting is a term that is used to describe a model that has learnt to the extent that it impacts its performance on unseen data. Any outliers/noise in the training data are picked up and learnt by the model (the training data has been fit to the extent that the model cannot generalize new data). Overfitting is a common phenomenon in non-linear models, such as decision trees.

Overfitting can be addressed by pruning the decision tree to remove some details it has learned. Pruning is a process where sections of the decision tree are removed. It helps in reducing complexity, prevents overfitting and improves the accuracy of prediction (either pre-pruning or post-pruning can be performed). If the tree continuously grows to the point where the lowest impurities are corresponded by each leaf node, the data is said to be overfit. On the other hand, if splitting has a premature stop condition, the performance of the model will get impacted due to high bias.

A dataset with high dimensionality will affect the structure of the tree. Additionally, the time consumed to find the splitting criterion will also increase. Feature importance can be estimated by estimating the normalized sum at each level. Feature selection occurs in such a way that the entropy reduces by a large margin. Hence, the feature with the highest normalized sum is the most important.

Random Forest Regressor

Random Forest is an ensemble learning method that can be used for classification or regression. There are two types of Ensemble Learning – Bagging (Bootstrap Aggregation) and boosting. Random sample with replacement is referred to as Bootstrap. Bootstrap facilitates a better understanding of the bias and variance in the dataset (Castro-Franco, Costa, Peralta, & Aparicio, 2015). A small subset of data is randomly sampled from the original dataset. This is a process used to reduce the variance, which is a problem with decision trees. Bagging helps each model run independently, whose outputs are aggregated without bias to a specific model.

Ensemble learning is the use of multiple models that are trained using specific data. An accurate prediction is obtained by averaging the results of each model, whereas a classification result is the majority class selected from each decision tree in the random forest. In the case of ensemble learning, the errors of each model are independent of each other.

Random Forest, unlike decision trees, does not usually overfit the training data. There is no interaction between the trees of a Random Forest model while it is built and when the model is trained on the dataset.

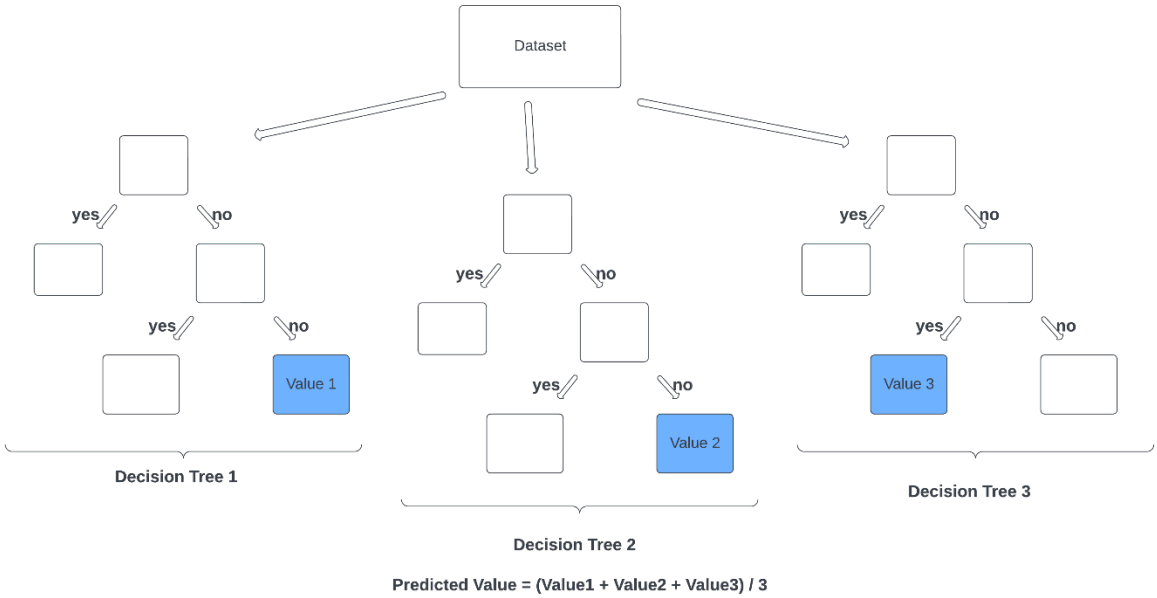

Fig. 30 – **Random Forest (with 3 decision trees) predicting a value.**

- Random forest is one of the most accurate ML algorithms (classification or prediction).
- It can run efficiently on vast datasets and handle many input variables (thousands).

Multiple Additive Regression Trees (MART)

The Multiple Additive Regression Trees (MART) model is an ensemble of Boosted Regression Trees (Vinayak & Gilad-Bachrach, 2015) (check Appendix I).

Boosting is a widely used algorithm in Machine Learning to primarily reduce bias and variance. It helps convert weak learning algorithms to strong ones (an example is Decision Trees). It helps improve the overall predictions of the model. Boosted Trees is a kind of additive model which predicts by combining decisions from a set of base models. It can be represented mathematically as:

$$g(x) = f_0(x) + f_1(x) + f_2(x) + \cdots$$

Fig. 31 – Classifier Equation

g(x) – the additive sum of base classifiers.
f (x) – base classifier.

In the case of Boosted Regression Trees (BRTs), each base classifier is a decision tree. The final Boosted Regression Tree is termed an additive regression model. Boosted Regression Trees generally produce better results than Random Forest (Boosting > Bagging).

Boosted Regression Trees are resilient to data overfitting. There are three types of Boosting – Adaptive, Gradient and XGBoost.The boosting used to train the decision trees of the MART model is **Gradient Boosting.** In the case of Gradient boosting, multiple weak learners (decision trees) are combined to form a strong learner. Each tree is serially connected to reduce errors, making this algorithm a slow learner.

To improve the model's efficiency, the weak learners are arranged so that each subsequent learner fits into the previously obtained residuals. Results are aggregated at each step thereby creating a strong learner. A loss function (such as MSE) is used to detect residuals. When any new tree is introduced into the model, the existing structure of the model remains unchanged. The newly introduced trees fit residuals of the current model. The hyperparameters that control the accuracy and performance of Boosted Regression Trees are – the learning rate and number of estimators.

Learning rate indicates how fast a model learns (denoted by α). When new trees are added to the existing model, the extent of modification of the overall model is determined by the learning rate. Lower values imply that models learn slowly.

Slow learning increases the efficiency and robustness of the model. It usually indicates better performance than a model with a higher learning rate.

The number of estimators is the number of decision trees that have to be added to the model. The drawback of having too many trees is the overfitting of the model on the data. Additionally, lower learning rates require additional estimators in the model for efficient learning.

- In the case of Random Forest, each observation in the dataset has an equal probability as the other observations, of getting selected. In the case of Boosted Regression Trees, the input data is weighted in subsequent trees. Data that was modelled poorly by previous trees has a higher chance of getting selected by the subsequent trees, thereby continuously improving accuracy.

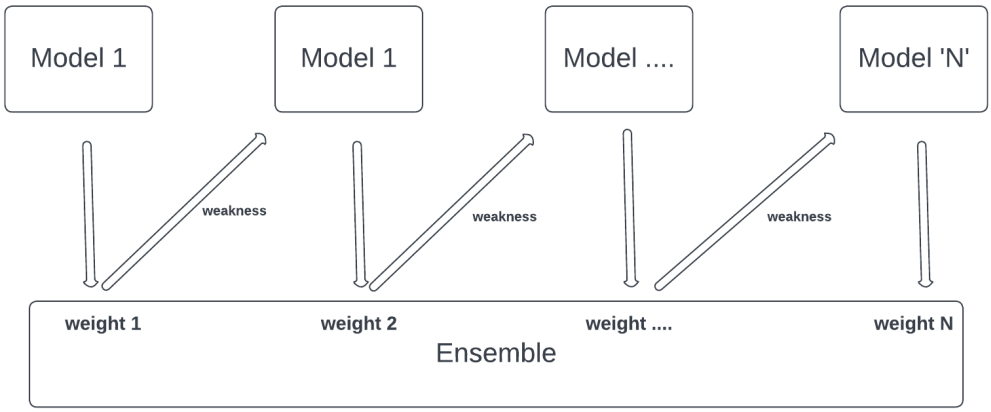

Fig. 32 – **Process of Boosting (Strong Learner Conversion)**

Results and Discussion (Data Visualization)

- The soil properties that have been considered to exhibit a relation with Bulk Density are:

Name of the Soil Property	Lab Method Code	Unit
Air-dry moisture content	2A1	%
Electric conductivity of 1:5 soil/water extract	3A1	dS/m
Chloride 1:5 soil/water extract	5A2	mg/kg
Water Soluble Nitrate	7B1	mg/kg
Bulk Density (Small Intact Soil Core)	503.01a	g/cm^3

Table 1

- The above properties, apart from Bulk Density, have been recorded for many soil profiles.

- A correlation plot between these variables is depicted in *Fig. 33 (a)* and the correlation values are represented in *Fig. 33 (b)*. The highest correlation is between Bulk Density and an explanatory variable is **Electrical Conductivity (3A1).**

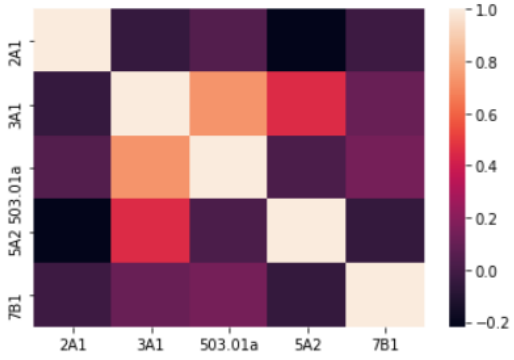

Fig. 33 (a) - Correlation Plot (all-variables)

	2A1	3A1	503.01a	5A2	7B1
2A1	1.000000	-0.043959	0.044067	-0.217898	-0.019493
3A1	-0.043959	1.000000	0.733098	0.455976	0.106786
503.01a	0.044067	0.733098	1.000000	0.024374	0.146211
5A2	-0.217898	0.455976	0.024374	1.000000	-0.050855
7B1	-0.019493	0.106786	0.146211	-0.050855	1.000000

Fig. 33 (b) – Correlation Values

Evaluation Metrics – Mass Preserving Spline

Metric	Value Obtained
Least RMSE of all fit splines	$1.226635 * 10^{-18}$
Maximum RMSE of all fit splines	370.893
Mean RMSE of all fit Splines	4.113986
Median RMSE of all fit Splines	0.02885843
Variance (RMSE) of all fit Splines	596.5366
Standard Deviation (RMSE) of all fit Splines	24.4241

Table 2

- The most negligible value of RMSE indicates that the observed and predicted values are very close to each other. Lower RMSE Values produce more accurate results.
- The Maximum RMSE Value was observed in a soil profile which had very little data to fit the spline. As a result, the predictions produced by the spline have a lower accuracy and hence a greater RMSE value.
- The Median Value indicates that most of the RMSE values are low, proving that there are multiple great fits with accurate predictions produced by the splines. However, some soil profiles have insufficient data to produce highly accurate predictions after fitting the spline, resulting in higher RMSE values.

- Each Machine Learning Algorithm – Multiple Linear Regression, Decision Tree Regressor, Random Forest Regressor, and Multiple Additive Regression Trees were fit to the data, in order to obtain the best possible fit and the lowest RMSE score.
- The Evaluation metrics concerned with Machine Learning Algorithms are – RMSE and R^2.

- R^2 score – a statistical measure of fit that indicates the proportion of variance of a target variable (dependent) that can be explained by explanatory variables (independent) in regression. If the R^2 value is 0.5, then half of the variation of one variable can be explained by input variables.

$$R^2 = 1 - \frac{RSS}{}$$

Fig 34 - R^2 estimation

RSS – Residual Sum of Squares

TSS – Total Sum of Squares

$$RMSE = \sqrt{\frac{1}{n} * \sum_{i=1}^{n} (o_i - e_i)^2}$$

Fig. 35 - Root Mean Squared Error (RMSE)

Evaluation Metrics – Multiple Linear Regression (MLR)

Hyperparameter	Value
Training dataset size	80% of the total dataset
Testing dataset size	20% of the total dataset

Table 3

The metrics are calculated over 5 iterations of varying training and testing datasets. Soil Samples are unique to the training or testing datasets (Simple Random Sampling without Replacement). This is done to ensure that there is complete independence of data.

Evaluation Metric	Average Value (5 iterations)
R^2 score	-7.318468501
MSE (Mean Squared Error)	0.392205172
RMSE (Root Mean Squared Error)	0.625599563

Table 4

- The R^2 value is negative. It compares the fit of a model with a horizontal straight line (mean of the observations). If the selected model fits the data worse than the horizontal line, a negative value of R^2 is obtained. The target variable does not seem to show any linear relationship with the explanatory variables.
- Since the Multiple Linear Regression Model seems to fit the data poorly, there is a more significant difference between predicted and actual values.
- Since the model is a poor fit, it results in inaccurate predictions, which increases the RMSE.

Hence, the Multiple Linear Regression Model is not a good fit for the data and is not used to predict Bulk Density or generate Pedo-Transfer functions.

Evaluation Metrics – Decision Tree Regressor

Hyperparameter	Value
Training dataset size	80% of the total dataset
Testing dataset size	20% of the total dataset

Table 5

- To reproduce the results, a random state was initialized to the '*train_test_split*' and to the Decision Tree before the training phase.

The metrics are calculated over 5 iterations of varying training and testing datasets. Soil Samples are unique to the training or testing datasets (Simple Random Sampling without Replacement). This is done to ensure that there is complete independence of data.

Evaluation Metric	Average Value (5 iterations)
R^2 score	-0.480667314
MSE (Mean Squared Error)	0.069756583
RMSE (Root Mean Squared Error)	0.262748496

Table 6

- The R^2 score obtained for the Decision Tree Regressor is higher as compared to the Multiple Linear Regression Model. However, this still seems to fit the data worse than the horizontal line (mean of the observations). This could be a case of overfitting and having a small size of the test set.
- Since the Decision Tree is a comparatively better fit than Multiple Linear Regression, it can predict data better and as a result has lower errors.
- Hence, this model has lower RMSE scores than the Multiple Linear Regression.
- A decision tree has low bias but high variance since it can change easily with a small change in the input. It may not have the ability to generalise patterns. There is a trade-off between the accuracy of its predictions and generalisations of patterns outside the training data.

Evaluation Metrics – Random Forest Regressor

- The Random Forest model was implemented before and after using GridSearchCV. As a result, there is a difference in the model fit and prediction accuracy before and after GridSearchCV.

Hyperparameter	**Value**
Training dataset size	80% of the total dataset
Testing dataset size	20% of the total dataset

Table 7

- To reproduce the results, a random state was initialized to the '*train_test_split*' and to the Random Forest model before the training phase.
- The condition for the Random Forest estimators was initialised to 300(arbitrary).
- The remaining hyperparameters are used in their default state.

The metrics are calculated over 5 iterations of varying training and testing datasets. Soil Samples are unique to the training or testing datasets (Simple Random Sampling without Replacement). This is done to ensure that there is complete independence of data.

For each iteration, the same datasets are subject to GridSearchCV.

- Prior to incorporating GridSearchCV, the Random Forest Regressor obtained the following results.

Evaluation Metric	**Average Value (5 iterations)**
R^2 score	0.004946418
MSE (Mean Squared Error)	0.047435287
RMSE (Root Mean Squared Error)	0.217419587

Table 8

- GridSearchCV is now implemented to produce optimal model parameters for the given data.
    - Number of iterations of GridSearchCV = 100
    - Value of 'k', to perform k-fold Cross Validation = 3
    - Random state is also assigned to reproduce results.

➢ Optimal Hyperparameters produced by GridSearchCV are depicted in *fig. 36.*

```
{'n_estimators': 400,
 'min_samples_split': 2,
 'min_samples_leaf': 1,
 'max_features': 'sqrt',
 'max_depth': None,
 'bootstrap': False}
```

Fig 36 – GridSearchCV Optimal Parameters (Single Iteration – Random Forest Regressor

- *n_estimators* (number of trees in the random forest), *max_features* (sqrt - the square root of the total number of input features) and *max_depth* (None) are the primary hyperparameters that are considered.
- These hyperparameters vary for each iteration, depending on the training and testing data.

During each iteration, the Random Forest Algorithm was executed on the testing data after modifying the hyperparameters to optimal values, which produced the results in *Table 9.*

Evaluation Metric	**Average Value (5 iterations)**
R^2 score	0.102644096
MSE (Mean Squared Error)	0.042102241
RMSE (Root Mean Squared Error)	0.20462071

Table 9

- GridSearchCV estimates optimal hyperparameter values that improve the model's performance. This increases the accuracy of the model which reduces errors between predicted and actual observations.

Hyperparameter tuning produced a better R^2 score and lower results for MSE and RMSE for the Random Forest Regressor.

Evaluation Metrics – Multiple Additive Regression Trees

- The MART model was implemented before and after using GridSearchCV. As a result, there is a difference in the model fit and prediction accuracy.
- The MART model is an ensemble of Gradient Boosted Regression Trees.

Hyperparameter	Value
Training dataset size	80% of the total dataset
Testing dataset size	20% of the total dataset

Table 10

- Before the training phase, a random state was initialized to the 'train_test_split' and the MART mode to reproduce the results.
- The condition for the number of estimators was initialised to 1000 (arbitrary).
- The learning rate was initialized to 0.56 (arbitrary).
- The maximum depth was set to 1.
- The remaining hyperparameters are used in their default state.

The metrics are calculated over 5 iterations of varying training and testing datasets. Soil Samples are unique to the training or testing datasets (Simple Random Sampling without Replacement). This is done to ensure that there is complete independence of data.

For each iteration, the same datasets are subject to GridSearchCV.

- Prior to incorporating GridSearchCV, the MART model obtained the following results.

Evaluation Metric	Average Value (5 iterations)
R^2 score	0.17490088
MSE (Mean Squared Error)	0.040032949
RMSE (Root Mean Squared Error)	0.199730502

Table 11

The R^2 score indicates that the current model (with existing hyperparameters) might not be a good fit for the data.

The mean squared error is higher compared to the Random Forest Regressor.

- GridSearchCV is now implemented to produce optimal model parameters for the given data.
    - Value of 'k', to perform k-fold Cross Validation = 3

```
{'learning_rate': 0.04,
 'max_depth': 10,
 'n_estimators': 1500,
 'subsample': 0.5}
```

Fig 37 – GridSearchCV Optimal Parameters (single Iteration) – MART

- *n_estimators* (number of trees in the random forest), *max_depth* (of the boosted regression trees), *learning_rate* (the speed at which the model learns), and *subsample* are primary hyperparameters that are considered.

These hyperparameters vary for each iteration, depending on the training and testing data.

During each iteration, the MART Algorithm was executed on the testing data after modifying the hyperparameters to optimal values, which produced the results in *Table 12.*

Evaluation Metric	**Average Value (5 iterations)**
R^2 score	0.175525115
MSE (Mean Squared Error)	0.039086059
RMSE (Root Mean Squared Error)	0.197186133

Table 12

- GridSearchCV estimates optimal hyperparameter values (per iteration) that improve the model's performance. This increases the accuracy of the model which reduces errors between predicted and actual observations.

Since the model better fits the data, the prediction accuracy increased, thereby resulting in lower MSE and RMSE values.

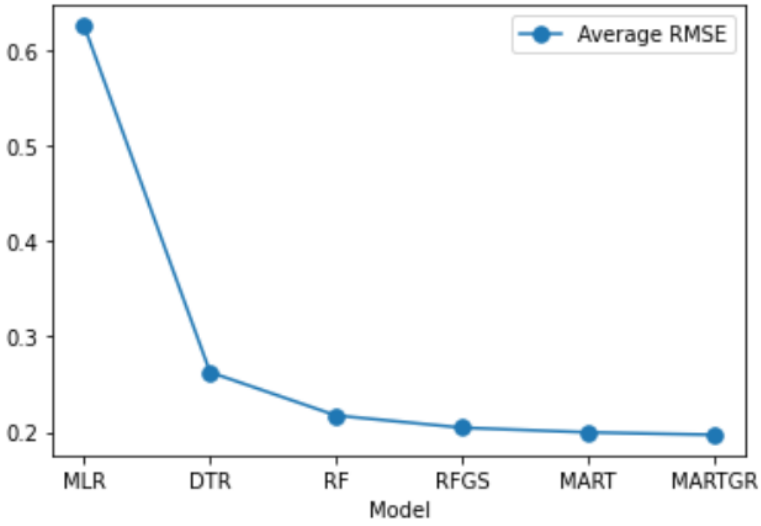

Fig. 38 - RMSE Values for each model

Comparing the results obtained by the Random Forest and MART Models, the **MART** model has lower RMSE values (indicates better predictions) and has a higher R^2 score. Both the MART and Random Forest Models produce comparable results, but the **preferred model in this case, is the MART Model**.

Feature Importance (MART Model) [Average of 5 iterations]:

Property	**2A1**	**3A1**	**5A2**	**7B1**
Feature Importance	**0.251997236**	**0.237749416**	**0.002706008**	**0.507547338**

Table 13

5A2 has a minimal contribution to the prediction of Bulk Density and can be neglected in the Pedo-Transfer function.

$$\rho = \beta_0 a + \beta_1 b + \beta_2 c$$

Fig. 39 – GridSearchCV Optimal Parameters – MART

a – Air-dry moisture content (β_0 is the coefficient of air-dry moisture content)

b – Electrical conductivity (β_1 is the coefficient of Electrical Conductivity)

c – Water soluble nitrate (β_2 is the coefficient of water-soluble nitrate)

Conclusion

The soil data that has been collated from various sources consists of a vast amount of information that can be used to predict Bulk Density. Exploratory data analysis (EDA) was performed on the cleaned dataset to observe any patterns and get a better understanding of the dataset. Observations for each soil parameter were recorded at different depths. Mass preserving Splines were fit to each soil profile in order to obtain a continuous set of values from the surface to the maximum depth, thereby making the data more uniform. A set of the most common properties on soils is extracted and a correlation plot is obtained to visualise their relationship. Machine Learning Models are applied to the dataset to observe the RMSE values and obtain Pedo-Transfer functions.

- **The incomplete dataset was cleaned and transformed into a complete dataset that can be incorporated directly into the APSIM Model.**
- **After the dataset was cleaned, the relation between Bulk Density and soil parameters was observed by observing the correlation of the entire dataset.**
- **A Pedo-Transfer function was obtained from the model that best fit the data (MART Model). This can be used to predict Bulk Density.**
- **The performance of each model trained on the data (Multiple Linear Regression, Decision Tree Regressor, Random Forest Regressor, and Multiple Additive Regression Trees) has been evaluated, and the model with the best fit and least error is to develop a PTF for Bulk Density.**
    - **The MART models showed the best fit for the soil data.**
    - **The MART Model outperformed the Random Forest Model and was used to develop the Pedo-Transfer Function.**

Future Scope and Recommendations

The primary objective was to efficiently predict Bulk Density and develop Pedo-Transfer functions for Bulk Density estimation.

- The original dataset collated from various sources can specify minimum requirements to ensure that each provider of data records required observations for effective analysis.
- For each soil profile, observations for a given set of parameters can be recorded to maintain uniformity.
- Soil data from southern and central Australia can be included to increase the generalisation of the applied models.
- A more significant number of soils can be considered. Larger datasets enable effective learning from Deep Learning models.
- Neural Networks (ANNs) can be trained on a large diverse dataset to check for model fit and prediction accuracy.
- Other interpolation methods can be tested on incomplete soil profile data (including other spline methods).

The suggestions mentioned above consider a more comprehensive set of conditions and methods to predict Bulk Density accurately. This widens the nature of soil data and facilitates quicker and more straightforward data analysis.

References

Bishop, T., McBratney, A., & Laslett, G. (1999). Modelling soil attribute depth functions with equal-area quadratic smoothing splines. *Geoderma, 91*(1-2), 27-45.

Castro-Franco, M., Costa, J. L., Peralta, N., & Aparicio, V. (2015). Prediction of soil properties at farm scale using a model-based soil sampling scheme and random forest. *Soil science, 180*(2), 74-85.

De Vos, B., Van Meirvenne, M., Quataert, P., Deckers, J., & Muys, B. (2005). Predictive quality of pedotransfer functions for estimating bulk density of forest soils. *Soil Science Society of America Journal, 69*(2), 500-510.

Friedman, J. H. (2001). Greedy function approximation: a gradient boosting machine. *Annals of statistics*, 1189-1232.

Friedman, J. H. (2002). Stochastic gradient boosting. *Computational statistics & data analysis, 38*(4), 367-378.

Friedman, J. H., & Meulman, J. J. (2003). Multiple additive regression trees with application in epidemiology. *Statistics in medicine, 22*(9), 1365-1381.

Heinonen, R. (1977). Towards "normal" soil bulk density. *Soil Science Society of America Journal, 41*(6), 1214-1215.

Holzworth, D. P., Huth, N. I., deVoil, P. G., Zurcher, E. J., Herrmann, N. I., McLean, G., . . . Murphy, C. (2014). APSIM–evolution towards a new generation of agricultural systems simulation. *Environmental Modelling & Software, 62*, 327-350.

Koltermann, C. E., & Gorelick, S. M. (1995). Fractional packing model for hydraulic conductivity derived from sediment mixtures. *Water Resources Research, 31*(12), 3283-3297.

Malone, B. P., Minasny, B., & McBratney, A. B. (2017). *Using R for digital soil mapping* (Vol. 35): Springer.

Martin, M., Lo Seen, D., Boulonne, L., Jolivet, C., Nair, K., Bourgeon, G., & Arrouays, D. (2009). Optimizing pedotransfer functions for estimating soil bulk density using boosted regression trees. *Soil Science Society of America Journal, 73*(2), 485-493.

Nathans, L. L., Oswald, F. L., & Nimon, K. (2012). Interpreting multiple linear regression: a guidebook of variable importance. *Practical assessment, research & evaluation, 17*(9), n9.

Ponce-Hernandez, R., Marriott, F., & Beckett, P. (1986). An improved method for reconstructing a soil profile from analyses of a small number of samples. *Journal of Soil Science, 37*(3), 455-467.

Stewart, V., Adams, W., & Abdulla, H. (1970). Quantitative pedological studies on soils derived from Silurian mudstones: II. The relationship between stone content and the apparent density of the fine earth. *Journal of Soil Science, 21*(2), 248-255.

Tranter, G., Minasny, B., McBratney, A., Murphy, B., McKenzie, N., Grundy, M., & Brough, D. (2007). Building and testing conceptual and empirical models for predicting soil bulk density. *Soil Use and Management, 23*(4), 437-443.

Vinayak, R. K., & Gilad-Bachrach, R. (2015). *Dart: Dropouts meet multiple additive regression trees.* Paper presented at the Artificial Intelligence and Statistics.

Appendix I

$$F_0(\mathbf{x}) = \arg\min_{r} \sum_{i=1}^{n} L(y_i, \gamma)$$

For $m = 1$ to M,

$$\tilde{y}_{im} = -\left\{ \frac{\partial L[y_i, F(\mathbf{x}_i)]}{\partial F(\mathbf{x}_i)} \right\}_{F(\mathbf{x}) = F_{m-1}(\mathbf{x})} \qquad i = 1 \ldots n$$

$$\{R_{jm}\}_1^J = J\text{-terminal node tree based on } \{\tilde{y}_{im}, \mathbf{x}_i\}$$

$$\gamma_{jm} = \arg\min_{\gamma} \sum_{\mathbf{x}_i \in R_{jm}} L[y_i, F_{m-1}(\mathbf{x}_i) + \gamma]$$

$$F_m(\mathbf{x}) = F_{m-1}(\mathbf{x}) + \nu\gamma_{jm} 1(\mathbf{x} \in R_{jm})$$

Fig. 40 - The MART Algorithm (Friedman, 2001)