Estimating Pore Fluid Saturation in an Oil Sands Reservoir using Ensemble Tree Machine Learning Algorithms

By

Gagan Kapoor

A Thesis Submitted to
Saint Mary's University, Halifax, Nova Scotia
in Partial Fulfilment of the Requirements for
the Degree of Bachelor of Sciences, Honours.

May 2017, Halifax, Nova Scotia

Approved: Dr. Andrew MacRae

Dr. Pawan Lingras

Date: 22/04/2017

Estimating Pore Fluid Saturation in an Oil Sands Reservoir using Ensemble Tree Machine Learning Algorithms

by Gagan Kapoor

Abstract

This thesis aims to estimate pore fluid saturation values in an oil sands reservoir using ensemble tree based machine learning models. Oil sands reservoirs provide an interesting opportunity to explore a relatively new technique in petrophysical analysis. The specific reservoir used in this study has high heterogeneity with discrete muddy layers that are difficult and time consuming to incorporate into a conventional petrophysical model. In addition, due to strong well control and sufficient well log data, the reservoir is a perfect candidate to test out a data-driven model by using techniques in Machine Learning – a subfield of Artificial Intelligence. Specifically, Random Forests and Extreme Gradient Boosted Trees are combined, which are two different ways to implement a decision-tree based model structure. The two algorithms have rapidly gained popularity in the machine learning community due to their robustness when dealing with outliers and/or bad data combined with a comparative immunity against over-fitting. The final aim of this thesis is to obtain comparable or superior results to the Modified Simandoux Equation method and analyze the shortcomings and advantages of the two methods in a real petroleum field.

22/04/2017

Acknowledgements

The data and the technical support required for this thesis was provided by an industry partner that must remain unnamed for confidentiality reasons. I thank all the people at the company for their continuous support and assistance.

My two academic supervisors have played a huge role in this thesis – Dr. Andrew MacRae and Dr. Pawan Lingras. I thank Dr. Pawan Lingras from the Computer Science department for teaching me and encouraging me to pursue my interest in Machine Learning/Artificial Intelligence. Dr. Andrew MacRae was very kind to take me up as his honours student on short notice. I really appreciate him giving after-hour access to his lab and his personal workstation (the fastest computer in our department!). I am lucky to have him as a supervisor due to his extensive knowledge not just in sedimentary geology but also in coding. I thank Dr. Pierre Jutras for sparking my interest in sedimentology and its applications in the oil/gas industry. I thank Dr. Ahlem Hajjem from HEC Montreal, for her guidance on combining Mixed Effects with Random Forests.

I thank the external reviewer for their evaluation and feedback on the written thesis.

Table of Contents

List of Figures

Chapter1: Introduction

This study focusses on the Lower Cretaceous Mannville Group, which holds the majority of heavy oil deposits in Alberta (Hayes et al., 1994). The oil sands of the Western Canada Sedimentary Basin (Fig. 1) make up the largest deposits of heavy crude bitumen in the world, and show a high degree of heterogeneity in terms of reservoir properties such as fluid saturation and distribution, geometry, porosity, permeability, mineralogy and other properties (Fustic et al., 2006). The variance in reservoir properties arises from the complex depositional history combined with the in-situ post-depositional diagenetic processes that these reservoirs have experienced (Fustic et al., 2006). The various autocyclical and allocyclical controls that have acted upon the complex assemblage of non-marine, marginal marine and shallow marine strata in the Mannville Group has encouraged extensive research to decipher the stratigraphic architecture of these deposits (McCrimmon & Arnott, 2002). Understanding and better estimating reservoir properties plays an integral role in every phase of exploration, development and production of an oil sands project and accurate predictions are of great economic value.

Previous studies have established that for reservoir analysis and to combat reservoir heterogeneities, multi-log methods are more useful compared to methods that use single well logs (Helle et al., 2001; Wendt et al., 1986). These methods include multiple linear regression models where the correlation coefficient between predicted and actual values increases with an increasing number of well logs as input (Wendt et al., 1986). The accepted workflow to determine agreeable oil : water fluid saturation (the fraction of pore volume occupied by water or oil/gas) values within the industry is to align the conventional petrophysical models with the specific reserovir by using derived petrophysical parameters from laboratory testing on drill core samples. In cases of more complex reservoirs involving thin interbedded layers of mudstone and detrital clay, further well-specific adjustments need to be made to models in order to arrive at accurate saturation estimates at reservoir scale. These include picking unique well-specific 'Gamma-Clean' and 'Gamma-Shale' values (often multiple values for a single well) in order to align model outputs with laboratory values obtained from cores. This thesis aims to construct an alternate and a more streamlined model to predict fluid saturation by using Machine Learning techniques. Machine Learning algorithms enable the use of the maximum amount of the data available, thus honouring Wendt et al. (2001)'s conclusions that multi-log methods are more useful than methods that use a single well log. Additionally, building more flexible models that are constructed from scratch using the intrinsic nature of the specific reservoir. Such an approach (once calibrated) will take into account the complexity and the heterogenity of specific reservoir without the tedious manual adjustments and without the need of using petrophysical parameters derived from lab results.

There is some noteable pre-existing research that has been aimed at using Machine Learning for predicting reservoir properties. However, the research has been widely concentrated on developing Artificial Neural Networks (ANN). For instance Helle et al. (2001) used a committee Neural Network and Amiri et al. (2016) used a neural network optimized by imperialist competitive algorithm. Others include, committee neural network with weight optimization using genetic algorithms (Chen and Lin, 2006), observational learning algorithm on neural networks (Wong et al., 2000) and feed forward back propagation (Mohaghegh et al., 2002). Thus, artificial neural networks have gained wide-spread popularity amongst geoscientists practicing machine learning. This can be attributed to the relatively early birth of ANNs as an algorithm combined with the numerous optimization techniques avaialable, and also to the extensive implementation of it in popular geoscientific software.

Tree based ensemble learning algorithms, although relatively new, have witnessed an exponential growth in terms of popularity and success rate, and are starting to be widely used by data scientists working in diverse fields. For instance, use of random forests in spectral data (Svetnik et al., 2003; Pal, 2005), time series analysis (Shen et al., 2007) and image segmentation (Yao et al., 2011). However, their usage by geoscientists remains scarce to nil. This thesis uses Random Forest (Breiman, 2001) and Extreme Gradient Boosted Trees (Chen, 2014), both of which are ensemble techniques based on the decision tree algorithm. One advantage of using a tree based approach over neural networks is the idea of a ‘black-box model’. Neural Networks due to their inherent complexity have been regarded as black boxes where the reasons for their function are difficult to understand, whereas Random Forest and Boosted Trees give the potential to extract the underlying patterns between input features based on the principles of ‘information gain’ (discussed later). Further, the aim of this thesis is to construct a simpler model with a less tedious workflow compared to the traditional petrophsyical model. The goal of a simpler model will not be met with a Neural Network technique because of the complexity when tuning/optimizing it. In the process of building a Machine Learning model, ‘feature engineering’ has been used to aid the model with derivation of more accurate output values. Results were considerbaly better when feature engineering was implemented compared to when it was not (See Results chapter).

In summary, the present work aims to build a pore fluid saturation model using Machine Learning, trained by using well logs as input, to predict fluid saturation values equivalent to values dervied using the ‘Dean Stark Laboratory Analysis’ from core as output. The algorithms to be used are a modified Random Forest (Breiman, 2001) and Extreme Gradient Boosted Trees (Chen, 2014). The results from these two algorithms will be compared with the values from the Modified Simandoux Equation (Bardon and Peid, 1969) which is currently one of the standard petrophysical models within the industry for shaly sandstone sequences.

The present work will attempt to answer the following questions using the Random Forest and Extreme Gradient Boosted Trees algorithms:

1. Can a Machine Learning pore fluid saturation model, trained by using well logs as input, accurately predict 'Dean Stark Laboratory Analysis' fluid saturation values from core?

2. Do the results from the Random Forests and Extreme Gradient Boosted Trees algorithms outperform the results from the standard Modified Simandoux Equation petrophysical model?

An important constraint on this project and its presentation is the matter of confidentiality of data and the ongoing assistance that has been provided by an unnamed industry partner. For this reason the exact location and dataset employed for this project can not be disclosed. Several details that would be routinely provided in a geological study of this type (e.g., a detailed location map showing wells) also can not be shown. However, in general the study is located in the Cold Lake area of Alberta, and published maps from other sources are provided as a proxy of the scale and type of data that was typically used for the project here.

Chapter 2: Background

The following chapter comprises the necessary information required to understand the objectives and the results of this thesis. The general geological setting of the study area is described, followed by an explanation of the conventional petrophysical models used to estimate fluid saturation. Lastly, the necessary techniques in Machine Learning are described with an aim to describe the workings of the underlying algorithms in the proposed models. As mentioned in the Introduction, the exact formation studied can not be specified for reasons of confidentiality, but the regional geology and the general character of the Mannville Group in the study area (Cold Lake) can be (see below).

2.1: Geological Setting

2.1.1: The Western Canada Sedimentary Basin

The study is focussed in the Western Canada Sedimentary Basin (WCSB; Fig. 1), which consists of Mesozoic sediments unconformably overlying Paleozoic carbonate-dominated successions and Precambrian granitic and metamorphic rocks (Cant and Stockmal, 1989). The overall dip direction of the basin is towards the southwest, expanding to the south and connecting with the cratonic Williston Basin to the southeast (Cant and Stockmal, 1989).

Due to tectonic compression at the western margins of the Western Canada Sedimentary Basin in the Jurassic and Early Cretaceous, older rocks were thrusted onto the continental margin (Hayes et al., 1994). The thrust loading on the craton edge led to basin subsidence, which provided accommodation space for the deposition of sediments (Hayes et al., 1994). Sediment supply during the Jurassic to Early Cretaceous was a mix of material from the erosion of the rising

Rocky Mountains in the west and the continued erosion of the craton of North America in the east (Hayes et al., 1994). Basin deposition eventually ceased later in the Cenozoic towards the end of the Laramide Orogeny and became dominated by erosion in the post-Eocene to the present day.

Variation in the deposition of the Early Cretaceous Mannville Group (see below) across the basin is due to multiple tectonic/structural elements (Hayes et al., 1994). These are the Rocky Mountain foredeep in the west, the Liard Basin and the Peace River Arch in the northwest, and the Sweetgrass Arch to the southeast, separating the foredeep from the Williston Basin further to the southeast (Hayes et al. 1994). Each of the structural arches separated thicker basin depocentres within this part of the WCSB during the Early Cretaceous. The thickest Mannville succession is recorded at the convergence of the southern flank of the Peace River Arch and the foredeep in addition to the relatively unexplored fault-bounded Liard Basin on the north flank of the Peace River Arch (Hayes et al., 1994).

The Punnichy Arch, located at the northeastern edge of the Williston Basin and joining with the Sweetgrass Arch further to the west, formed due to the dissolution of the Devonian Prairie Evaporite on either side of its flanks (Hayes et al., 1994). This dissolution provided additional accommodation space north of the Punnichy Arch, resulting in extensive clastic deposition in the Early Cretaceous that was sourced from the Precambrian Shield (Jackson, 1984; Hayes et al., 1994). The western edge of the Williston Basin, the Sweetgrass Arch, remained structurally high until late Mannville deposition (Hayes et al., 1994). Thus, the Mannville strata over the Sweetgrass Arch, including in the Cold Lake area, remained relatively thin compared to the adjacent Punnichy Arch or the main WCSB depocentres.

2.1.2: The Mannville Group

The formation of interest consists of poorly consolidated sands occurring in stacked incised valley fill complexes, in a brackish to marine, tide-dominated deltaic setting, and is a part of the Mannville Group (Fig.2). Figure 3 shows an analogous formation with similar stacked incised valley fill sequences.

The Mannville Group in the Cold Lake area consists of the McMurray, the Clearwater and the Grand Rapids formations in stratigraphic order, with the McMurray Formation in the lower Mannville consisting mainly of quartz arenite compared to the feldspathic litharenite of the Clearwater Formation (Hayes et al., 1994). This change in overall sand composition reflects variations in depositional settings (McCrimmon and Arnott, 2002). Differences in provenance, as inferred from detrital zircon dating (Benyon et al., 2014; Blum and Pecha, 2014), is thought to be the main driver of the variations in mineralogy of the Mannville sands.

These divisions within the Mannville are the result of previous work, starting with Naus (1945) who coined the term ‘Mannville Formation’, followed by Badgley’s (1952) proposal to group the sediments above the Paleozoic unconformity and overlain by the Colorado Group shales as the ‘Mannville Group’ (McCrimmon, 1996). Glaister (1959)**,** Loranger (1951), Rudkin (1964) and Jackson (1984) contributed more detailed work to subdivide the Mannville into upper, middle and lower based on different depositional settings (McCrimmon, 1996). This was followed by the combined efforts of Vigrass (1965) and Clack (1967) that gave rise to the formal labels that are used today both within the industry and academia. For example, in the oil sands area, the McMurray Formation was initially labelled as unit D, the Clearwater Formation was previously called unit C and the Grand Rapids Formation was informally referred to as units A and B (McCrimmon, 1996; Fig. 2).

The McMurray Formation is dominated by fluvial-point-bar and channel-fill deposits consisting mainly of very fine- to fine-grained, mature quartz sandstone unconformably overlying the Devonian carbonates (Jardine, 1974; McCrimmon, 1996). These units are overlain by inter-bedded sandstones and shales deposited in a tidal flat setting when the Boreal sea transgressed from the north (Jardine, 1974; Harrison et al., 1981; McCrimmon, 1996). This was followed by sea-level fall that marked the end of the McMurray deposition. The McMurray Formation is unconformably overlain by the basal Wabiskaw Member (0-10 metres thick) which is dominated by glauconitic sandstone with a small fraction of shale interbeds. The Wabiskaw is in-turn overlain by the deltaic deposits of the Clearwater Formation under a transgressive regime (McCrimmon, 1996).

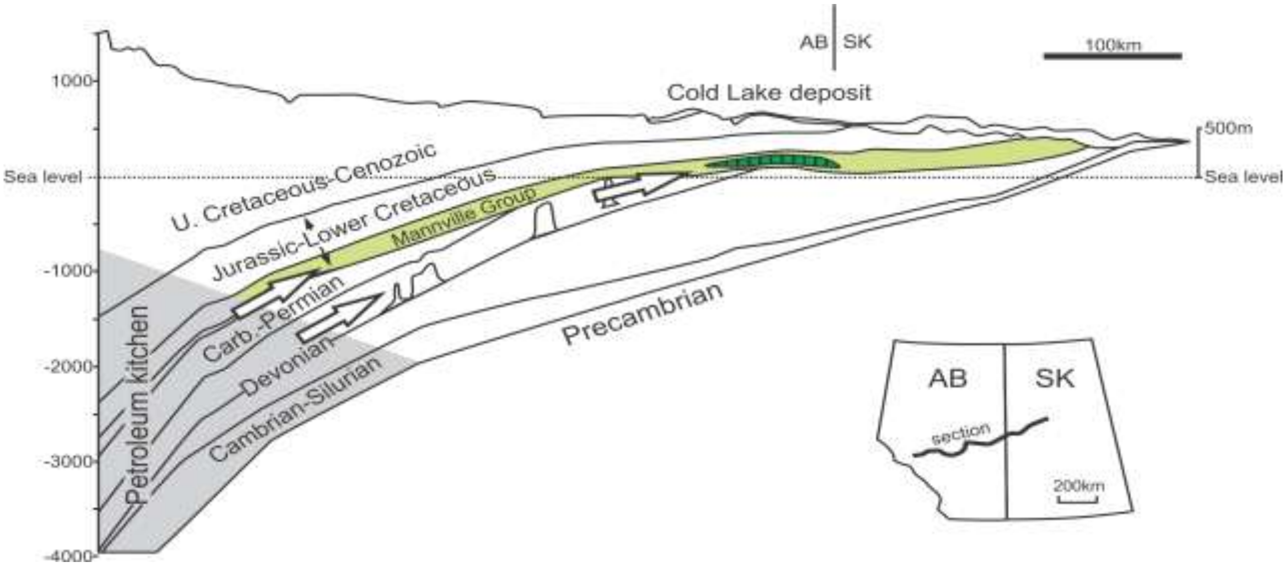


Figure 1: The Western Canada Sedimentary Basin. A SW-NE cross-section depicting the major stratigraphic units. Modified from Mossop & Shetsen (1994). Arrows represent generalized direction of petroleum migration from the petroleum kitchen in the west towards the eventual emplacement in the Cold Lake Oil Sands area.

The Clearwater Formation is dominated by fine-to medium-grained, moderate-to poorly consolidated, feldspathic litharenite and ranges from 40 to 104 metres in thickness (McCrimmon, 1996). The Clearwater sands are capped by a relatively thin shale unit in the Cold Lake area where the Clearwater Formation is the main bitumen reservoir (McCrimmon, 1996).

The Grand Rapids Formation varies from 75 to 134 metres in thickness and unconformably overlies the Clearwater Formation, in turn being overlain by the Colorado Group shales (McCrimmon, 1996). The Grand Rapids Formation consists of well-to poorly-consolidated, fine-to medium-grained, feldspathic and lithic sandstone interbedded with shale (McCrimmon, 1996). The Lower Grand Rapids as described by Milken (1974) consists of sediments deposited in a deltaic setting prograding towards the north into the Boreal Sea (McCrimmon, 1996). It is primarily comprised of thick sandstone and siltstone strata with shale and minor coal interbeds. The Upper Grand Rapids Member, again described by Minken (1974), was said to be deposited in a beach and shallow-marine environment under a transgressive regime. However, more recent sedimentological, ichnological and palynological studies have inferred a restricted to brackish environment (Benyon and Pemberton, 1992).

The top of the Grand Rapids Formation is marked by an erosion surface due to the fall of relative sea level (McCrimmon, 1996). This was later followed by a period of transgression and the formation of the Western Continental Seaway which led to the deposition of the Joli Fou Shales (Jackson, 1984).

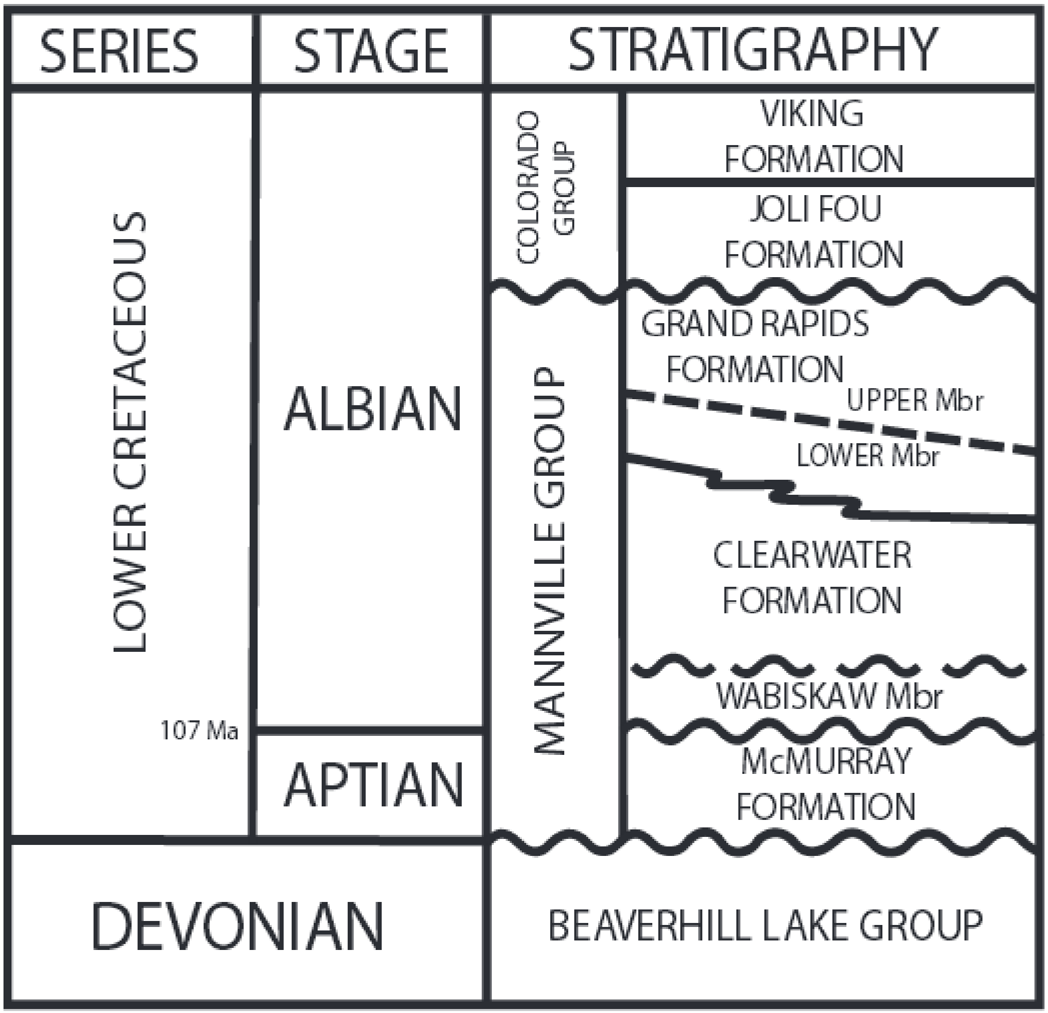


Figure 2: Lower Cretaceous Stratigraphy in the Cold Lake Area, northeastern Alberta (McCrimmon and Arnott, 2002).

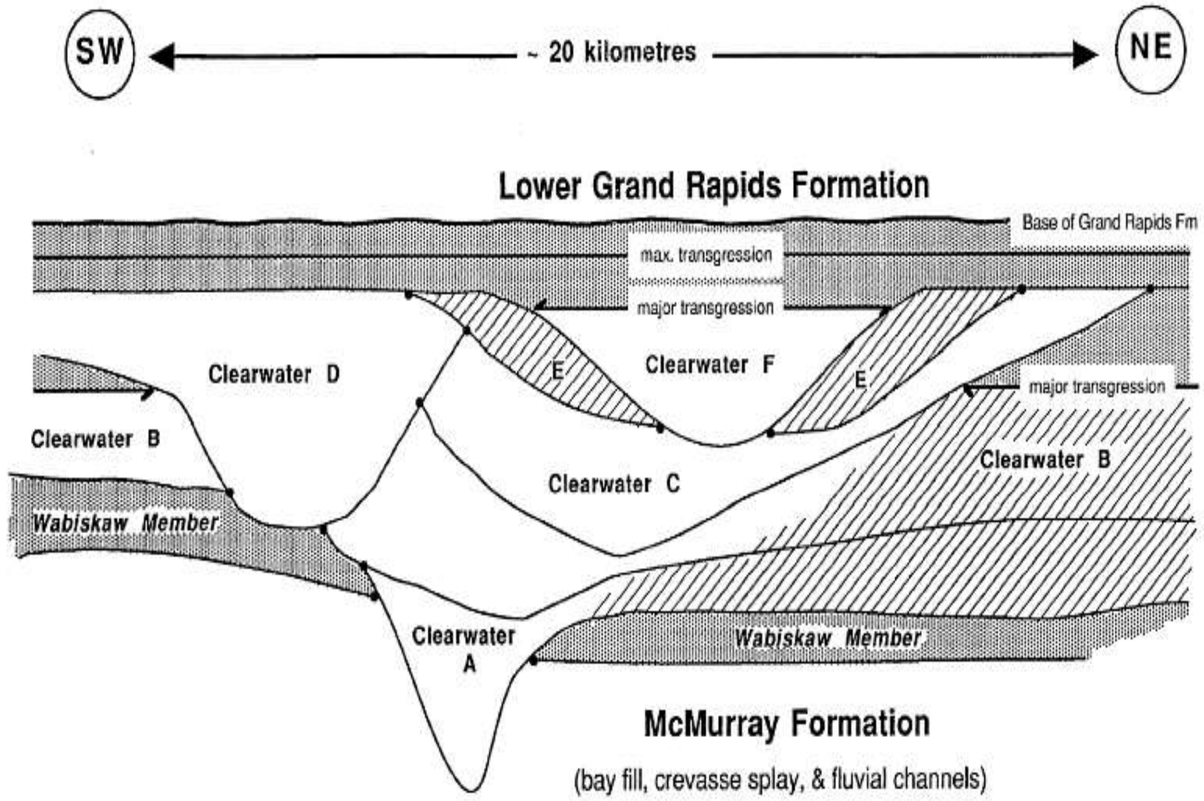


Figure 3: Schematic cross section of an analogous geological formation – Clearwater Formation in the Mannville Group. White: Proximal estuarine fill; Diagonal hachured: Distal Estuarine; Heavily stippled: Open Marine (Hein et al., 2007; Cheadle et al., 1995).

2.1.3: The Petroleum System

The petroleum system of the Western Canada Sedimentary Basin is complex and consists of multiple discrete systems linked to a number of different source rocks. Mature source rocks are primarily found in the western margins of the basin, where burial depth has led to optimal pressure and temperature conditions for maturation/generation. The Western Canada Sedimentary Basin is said to be a supercharged, laterally drained, low impedance basin (Creaney et al., 1994).

The source rocks of the heavy oil within our study area have been a matter of debate. The sheer volume of the reserve in place makes it impossible to be linked with one source rock.

Additionally, the high amount of biodegradation has made geochemical correlation difficult. However, the two prime candidates for source rocks are the Upper Devonian-Lower Mississippian Exshaw Formation and the Lower Jurassic Nordegg Member of the Fernie Group (Creaney and Allan, 1992). The Exshaw Formation is a black, laminated, slightly phosphatic organic rich mudshale (Meijer Dries and Johnston, 1996). It is classified as a type II source rock with a total organic content up to 14 weight percent (Meijer Dries and Johnston, 1996). The Exshaw is interpreted to be deposited in an offshore, outer continental shelf anoxic setting (Meijer Dries and Johnston, 1996). The Nordegg is a fine grained, organic rich, fossil rich, phosphatic calcareous mudstone deposited in deep anoxic restricted bottom water conditions (Riedieger et al., 1990). It is classified as type I/II source rock with total organic carbon content near 28 weight percent (Riedieger et al., 1990).

The principle phase of oil generation occurred during the Late-Cretaceous to Early Tertiary, resulting from the Cordilleran tectonism or the Laramide orogeny (Creaney et al., 1994). The thrusting and faulting led to basin subsidence and clastic loading which provided optimal pressure and temperature conditions for oil generation. An up-dip directed dynamic pressure was created which led to the migration of these hydrocarbons towards the east (Fig.4; Creaney et al., 1994).

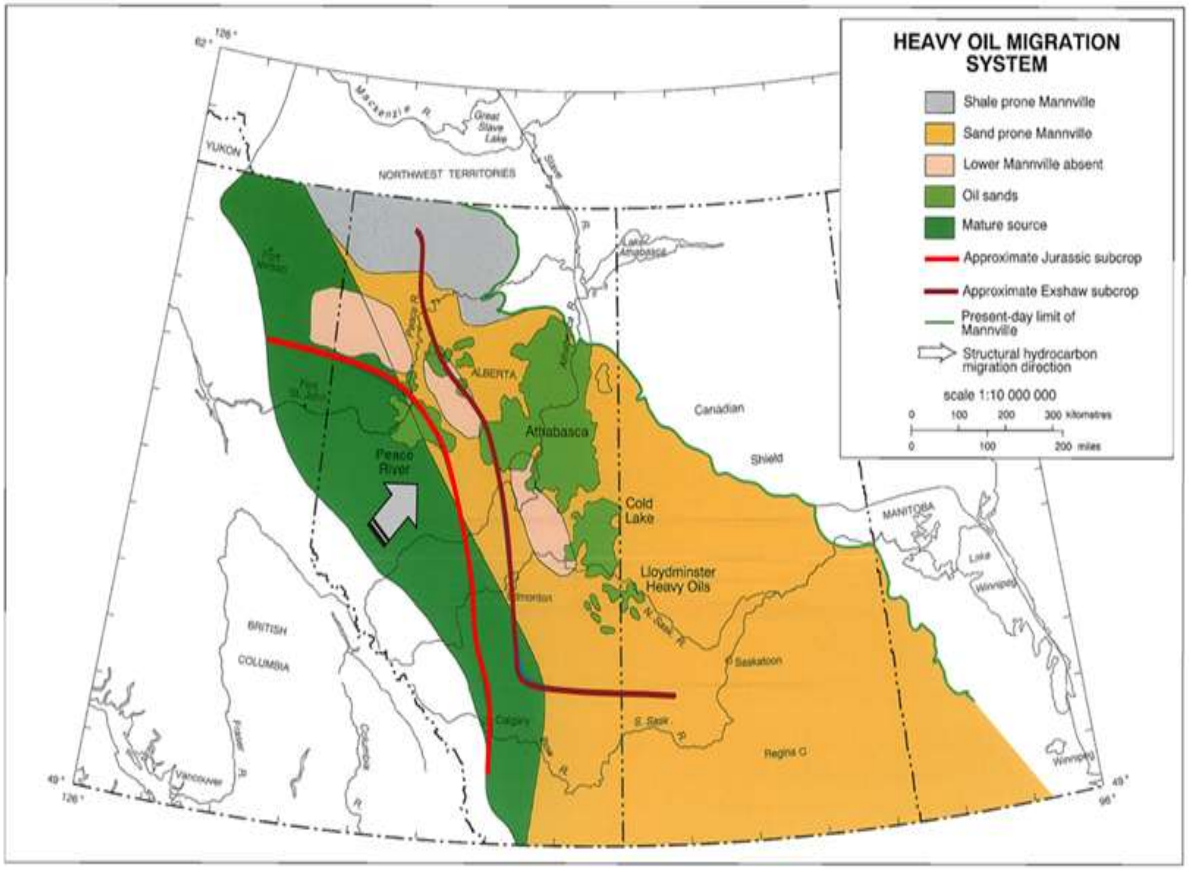


Figure 4: Petroleum migration direction in the WCSB with the Nordegg and Exshaw subcrop. Modified from Mossop & Shetsen (1994).

The majority of the oil was emplaced into shallow lower Cretaceous reservoirs like the McMurray Formation in the Athabasca region and the Clearwater Formation in the Cold Lake area. The oil was trapped with the help of a combination of structural and stratigraphic traps as described by Fustic (2013). Additionally, the bitumen also plays a part in acting as a seal itself due to its high viscosity that restricts flow at shallow subsurface temperatures.

2.2: Oil Sands Development in the Western Canada Sedimentary Basin

The oil sands are considered to be an amalgamation of sand, water clay and bitumen – a type of oil that is similar to the viscosity of molasses at room temperature. The occurrence of this natural resource is primarily in the subsurface which comprises 97% of the total oil sands surface area, the remaining 3% is accounted by mining operations where the oil sands occur as outcrops (CAPP, 2017).

The Athabasca oil sands are described by the Canadian Association of Petroleum Producers (2017) as the world's largest, most developed and the most technologically advanced operation of its kind.

In the last 50 years, Canadian Oil Sands have witnessed exponential growth which is credited to a parallel increase in technological development and economic factors. Quantifiably, daily production has increased from 30,000 barrels of oil per day in the 1970s to over 1.7 million barrels a day (Paskey et al., 2013).

Without a doubt, the oil sands have become an integral part of Canada's economy by spearheading its energy sector while accounting for more than half of the total crude oil produced in the country (Paskey et al., 2013).

In-situ production is mainly done through a process called Steam Assisted Gravity Drainage (SAGD) (Fig. 5). In this technique, a pair of horizontal wells is drilled into the reservoir of interest with a vertical spacing of about 4 to 6 metres (Energy Alberta, 2017). The pair consists of an injector well through which steam is injected into the formation in order to heat up the bitumen in the pore-space and decrease its viscosity making it flow into the producer well placed below the injector (Energy Alberta, 2017). SAGD has become the most widely used recovery method in Alberta and has a considerably smaller environmental footprint compared to surficial mining (Energy Alberta, 2017).

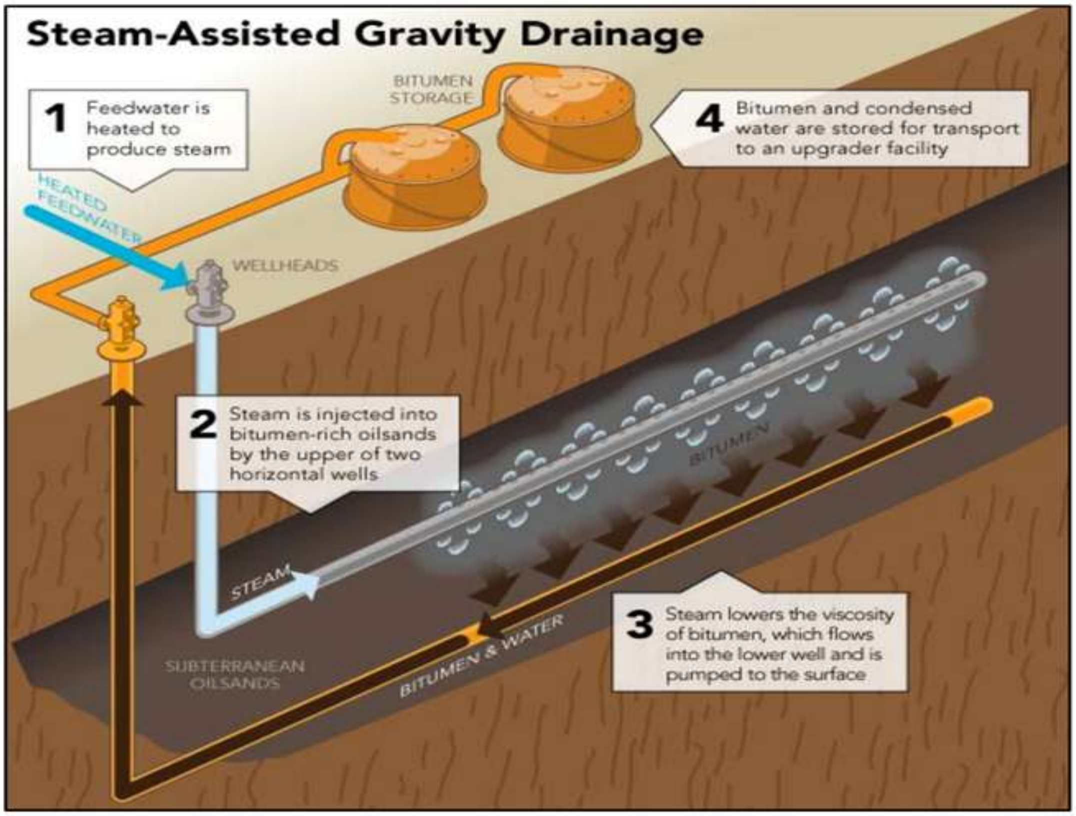

Steam-Assisted Gravity Drainage
1 Feedwater is heated to produce steam
BITUMEN STORAGE
4 Bitumen and condensed water are stored for transport to an upgrader facility
HEATED FEEDWATER
WELLHEADS
2 Steam is injected into bitumen-rich oilsands by the upper of two horizontal wells
BITUMEN
STEAM
3 Steam lowers the viscosity of bitumen, which flows into the lower well and is pumped to the surface
SUBTERRANEAN OILSANDS
BITUMEN & WATER

Figure 5: Steam Assisted Gravity Drainage (SAGD) (Energy Alberta, 2017).

In-situ oil sands projects have a relatively high level of well control, which suits the construction of a data driven model. Fig. 6 shows an analogous oil sands project in a different area with a similar well control to the area of the present study, and showing the typical scale of fluvial and estuarine channel systems.

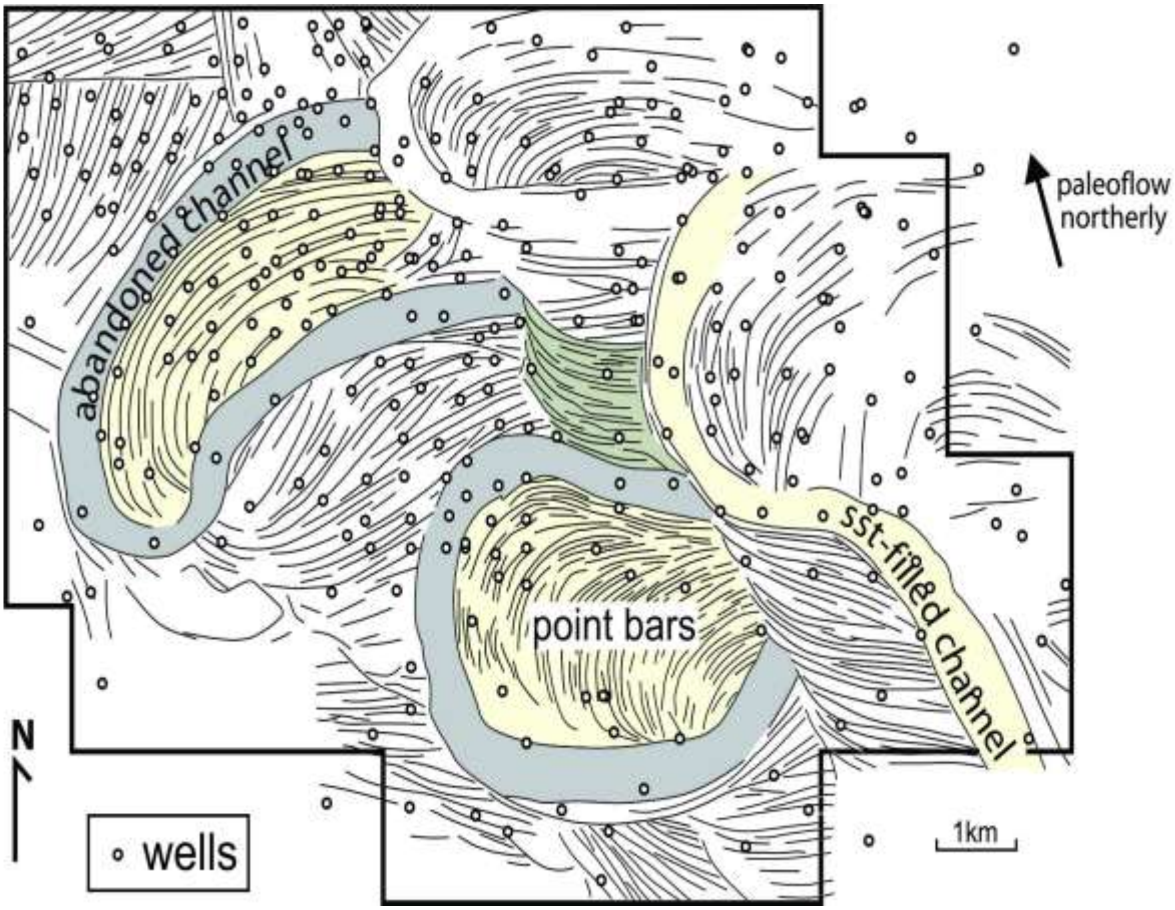


Figure 6: Analogous Oil Sands Project in terms of well control. Based on a 3D seismic time slice from the Long Lake Lease area, about 8ms (~8m) below the top of the McMurray Formation. Modified from Hubbard et al. (2011).

2.3: Modified Simandoux Equation for Water Saturation Estimation from Well Logs

Archie (1942) coined the term 'formation factor' (F), which was found to be a constant ratio of the resistivity of brine-saturated rock (R_o) to the resistivity of the brine (R_w) for a given rock (Archie, 1942). And it was further shown by Archie (1942) that:

$$F = \frac{1}{\emptyset^m}$$

Where, ϕ = Porosity, and m depends on the consolidation of the rock and is thus called the cementation factor.

Archie also proposed that the resistivity of a rock partially saturated with brine (R_t) over the resistivity of the same rock fully saturated with brine (R_o) could be equated to the water saturation (the fraction of water versus hydrocarbon in the pore space - Sw) with the following relationship:

$$S_w^{-n} = \frac{R_t}{R_o}$$

Where, n is called the saturation exponent. From the above equations, one can find the value for water saturation, which is called the Archie Equation:

$$S_w = \sqrt[n]{\frac{a.R_w}{R_t.\emptyset^m}}$$

The 'a', which is called the 'tortuosity index' has been developed from the works of Winsauer et al. (1952).

Several water saturation models exist and are used in the industry, the selection of the appropriate model depending on the nature of the reservoir. The different models are all derived from the basic Archie model, which works well for a completely 'clean' sand reservoir with little or no clay present (Archie, 1942). This is because Archie's equation assumes that the electrical conductivity due to the rock itself is negligible, an assumption violated by the presence of clays due to their cation exchange capacity. Thus, when clay is present, the Archie Equation over-estimates water saturation values. One of the methods typically used for shaly-sand systems is the Modified Simandoux Equation to calculate water saturation from well logs, proposed by Bardon and Peid (1969):

$$\frac{1}{R_t} = \frac{S_w^2}{F.R_w} + \frac{V_{sh}S_w}{R_{sh}}$$

Or,

$$\mathrm{S_w} = \left\{\left(\frac{\mathrm{F.R_w}}{\mathrm{R_t}}\right) + \mathrm{V_{sh}}\left(\frac{\mathrm{F.R_w}}{2\mathrm{R_{sh}}}\right)\right\}^{1/2} - \mathrm{V_{sh}}\left(\frac{\mathrm{F.R_w}}{2\mathrm{R_{sh}}}\right)$$

Where,

R_{sh} = Shale Resistivity, V_{sh} = Volumetric Fraction of Shale. The Modified Simandoux Equation is a member of a group of water saturation models that assume shale as a homogeneous conductive material, and formulises equations between the resistivity of shale and the volumetric fraction of shale present.

As this is the standard technique, water saturation values for the wells used in this study were calculated using the above equation as a comparison, in order to assess the relative performance of the machine-learning model.

2.4: Dean Stark Analysis

To determine pore fluid saturation from core samples, the Dean-Stark extraction method (Dean & Stark, 1920) was used in the lab, where fluid saturations are calculated using distillation extraction. The method is usually adopted for unconsolidated sediments such as oil sands. The cores are kept frozen during transport and cut in two halves. One is kept for descriptions and photography while the other half is sent for analyses. Water is driven off by applying heat through a boiling solvent (e.g. Toluene), condensed and then collected. The volume of water is thus measured, followed by sending the condensed solvent to flow back over the sample to extract the oil. Once all the fluid is driven off, usually after a time period of approximately two days, the difference in the initial weight of the rock/sample and the final weight of the sample plus the weight of the water is calculated to find out the volume of oil, with adjustments for oil and water density as part of the calculation. In this study, values of water saturation and bitumen weight determined by Dean-Stark analysis are assumed to be the ‘accurate’ values for the cored intervals.

2.5: Machine Learning and Workflow

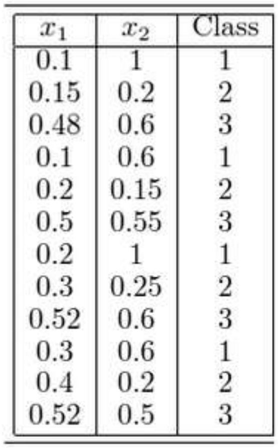
Data analysis has witnessed a paradigm shift in the last couple of decades with the advent of advanced processing systems with superior storage capabilities. Statistical algorithms which have existed for almost a century have now been combined with machines to iteratively learn from the vast amounts of data being produced today.

This coupling of robust statistics with equally robust processing machines/computers has given rise to the idea of Machine Learning. Machine Learning, which is a sub-field of artificial intelligence, not only paves the way to develop intelligent machines and thus promoting automation, but also gives us the ability to explore hidden patterns in data.

The area where machines and computers hold a clear advantage over their human counterparts is the realm of ‘multi-dimensionality’. Human beings are typically restricted to or are comfortable working in three dimensions or less. Add a fourth dimension or more, and things start getting complicated. Thus there exists a large number of real world problems, which are addressed through old theoretical models that are restricted to a limited number of dimensions.

By contrast, the real world is messy and multidimensional, thus constraining any natural model in theory to one or two dimensions often does not do it justice.

Machine learning has come forward as a tool to break down these dimensionality barriers, and various fields of science such as Biology, Astronomy, and Medical Sciences to name a few, are reaping the benefits of adopting Machine Learning techniques (Marsland, 2009). Marsland (2009) gives a good illustration in Fig. 7 of how not just the sheer scale and size of contemporary data but also how it is stored (i.e. numerical rows and columns) leaves the human mind at a disadvantage, the numerical data can be represented efficiently on a 2D Cartesian plane as long as the dimensions are less than or equal to three, however, interpretations become exponentially hard as the dimensions increase beyond the scope of simple visual representation.



x_1	x_2	Class
0.1	1	1
0.15	0.2	2
0.48	0.6	3
0.1	0.6	1
0.2	0.15	2
0.5	0.55	3
0.2	1	1
0.3	0.25	2
0.52	0.6	3
0.3	0.6	1
0.4	0.2	2
0.52	0.5	3

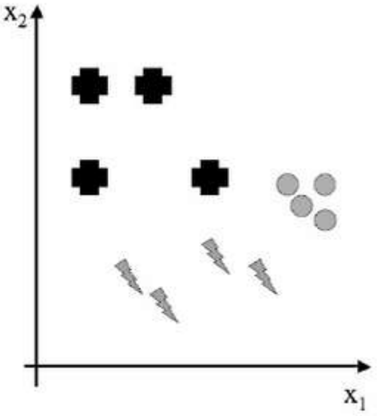


Figure 7: Numerical data points in rows and columns on the left, plotted as a graph on the right. The human mind finds it much easier to visually analyze data (Marsland, 2009).

The approach to constructing a machine learning algorithm is essentially derived from the learning techniques of animals i.e. learning from experience. The three pillars of successful learning are – remembering, adapting and generalising. Shwartz and David (2014) draw wonderful parallels from animal behaviour experiments to explain how these key segments have been demonstrated to be integral to learning, and how they form the basis of developing 'intelligent' machines.

From here on, the focus will be on the general workflow of producing a machine learning model and the different components associated with it. Assuming the data has been collected and digitized, from then on, the broad steps of the workflow are pointed out in Fig. 8.

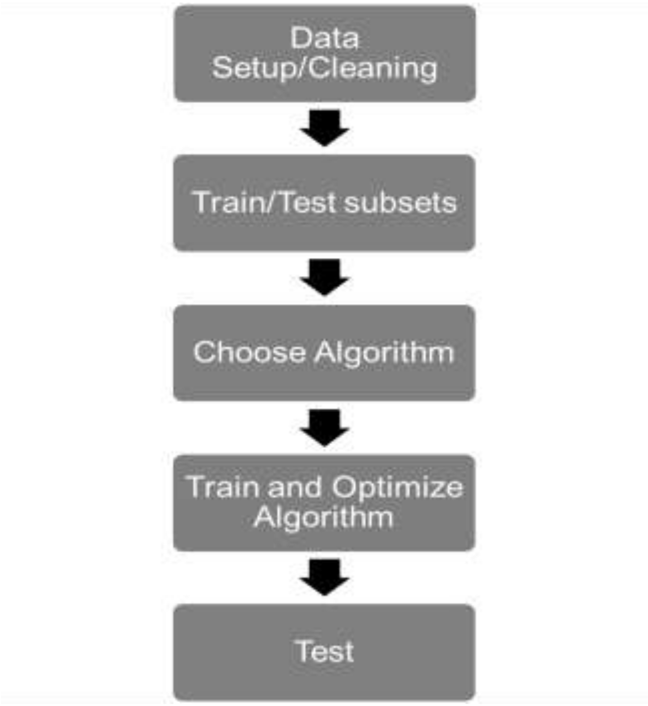


Figure 8: Overview of the steps of building a Machine Learning Model

Data Setup: Data setup is important for two broad reasons: understandably, machine learning models are data-driven, and the success of one's model lies on the strength of one's data, strength being proportional to how representative the data is to the end goal and also how clean and consistent the data is. Another important reason for data setup is from the programmer's point of view. It is extremely essential to understand one’s own data. The more extensively one studies the data the more one learns about it and discovers things that were previously unseen, including features that could make or break a predictive model. The specific steps that were followed in the data setup process for this study are discussed in Chapter 3 of this thesis.

Train, Validate and Test:

According to Ripley (1996), when model selection has to be simultaneously computed along with true error estimates, the dataset needs to be split into three disjoint sets, namely:

1. Training Set: This is a subset of the dataset that contains your predictor variables (e.g. the different geophysical well logs in our study) along with the output variable i.e. the value to be predicted (pore fluid saturation). The training set forms the foundation of the ‘learning’, and is a set of examples that help the algorithm to formulate a function that will lead it to predict the outcome variable using the inputs provided. To prevent training the algorithm on erroneous data such as parts of the logs with hole condition problems or cased versus uncased parts of the hole, data cleaning is a vital step and should be done before the data is split into a training set.
2. Validation Set: Validation set is essentially a subset of the training set, which is used to tune the parameters of an algorithm. These parameters differ depending on the algorithm being used, for instance in a multi-layered perceptron, one might use a validation set to find the optimal number of hidden layers (Rumelhart et al., 1985). In the case of an ensemble tree algorithm (Section 2.5.2), the number of trees to be used is optimized using a validation set. The random forest algorithm employed in this thesis project inherently creates ‘out-of-bag’ validation sets and thus, the manual creation of validation sets was

deemed not necessary because it is automatically addressed in the R package (Breiman, 2001); see Chapter 3: Methods.

3. Testing Set: Another subset of the dataset, with the same input and output variables as the training subset, which is used to assess the performance of the now tuned model. The final performance metrics reported in this thesis will be based on the test set/sets (cross-validation).

To further describe the process of creating a machine learning model, assume a dataset containing:

Input variables X = $x_1, x_2, \ldots. x_n$

Response Variables Y = $y_1, y_2, \ldots y_n$

The definition of a model refers to the mathematical workflow of making a prediction $x_i \rightarrow y_i$ (Chen, 2014). The prediction is usually governed on the ability to learn previously undetermined parameters θ (Chen, 2014).

To find the best parameters from a set of examples in the training set, one needs to define a cost function or an objective function (Chen, 2014). The objective function $G(\theta)$, consists of 'training loss' and 'regularization' and can be defined as (Chen, 2014):

$$G(\theta) = L(\theta) + \Omega(\theta)$$

Where, L = Training Loss Function, Ω = Regularization Term

The training loss function varies depending on the need of the task and the nature of the data. The training loss function is a measure of the prediction accuracy of the model on the training set, whereas the regularization term controls the 'complexity' of the model. If a model is too complex it gets prone to over-fitting, i.e. high accuracy on the training set and a very low accuracy on the test set, analogous to rote-memorization for an exam as opposed to constructive learning.

Chen (2014) presents a good explanation of Training loss and regularization in Fig. 9.

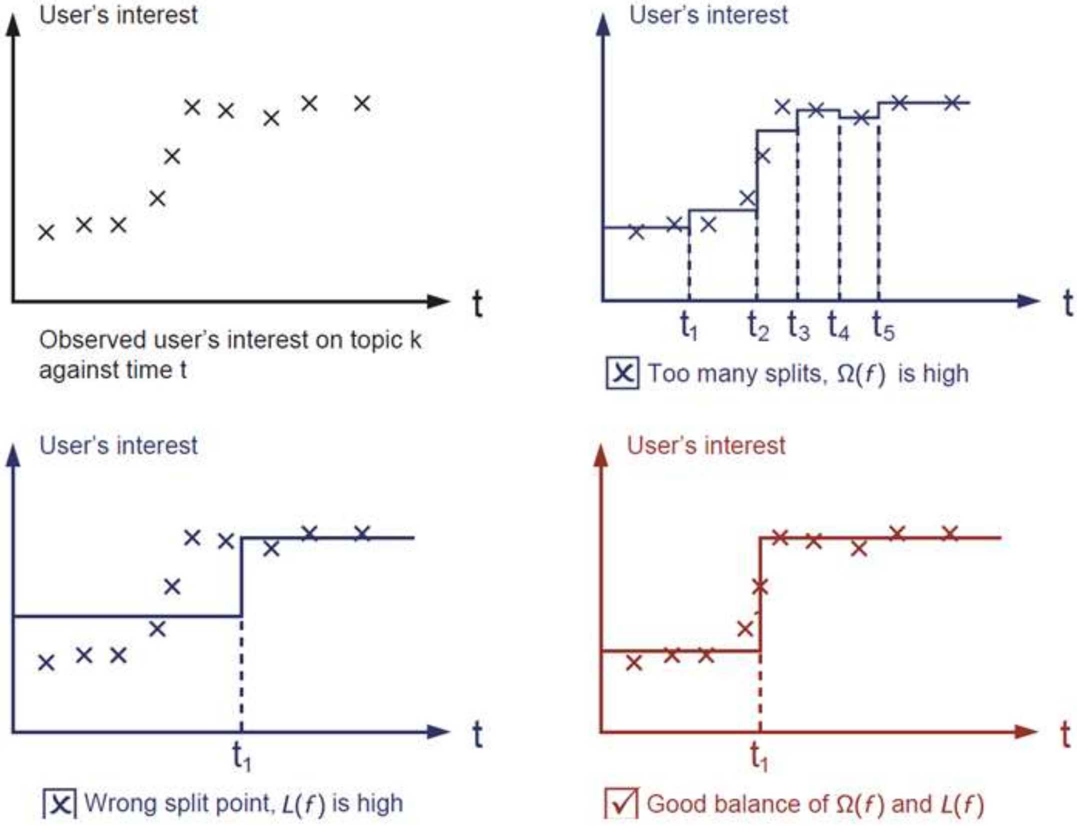


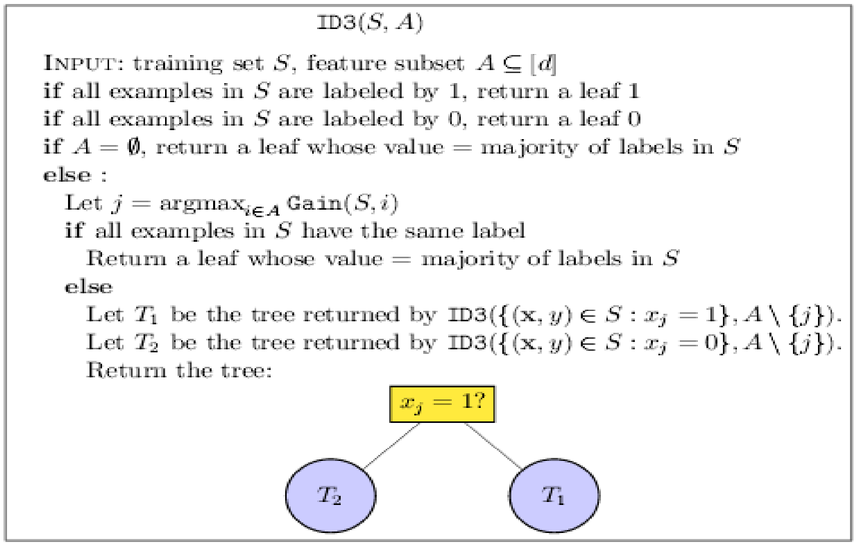
Figure 9: Visual Representation by Chen (2014) of the bias-variance trade-off: Visually fitting a line, given the data points provided (Chen , 2014).

2.5.1: The Decision Tree Algorithm

Decision Trees or prediction trees aim to predict an outcome or class Y from inputs or features $X_{1,}$ $X_{2,}$ $X_{3,}$ ….. $X_{p.}$ The tree is grown by travelling from a root node of a tree to a leaf (Shalev-

Shwartz & Ben-David, 2014). At each node in a tree, a test is applied to X$_p$ which determines the direction of travel, i.e. right sub-branch or the left sub-branch. This is done iteratively, until a leaf is reached where the prediction can be made (Shalev-Shwartz & Ben-David, 2014).

There are different implementations to achieve the above mentioned goal, for example: the 'Iterative Dichotomizer 3' (ID3) (Quinlan, 1986) , C4.5 (Quinlan, 1996) and CART (Breiman et al., 1984). The 'test' described in the former paragraph is essentially a quantifiable measure of the improvement due to a certain split. It is essential to understand the concept of how this is done. The pseudocode for the ID3 algorithm taken directly from Shalev-Shwartz & Ben-David (2014) is shown in Figure 10, where the ID3 returns a decision tree after taking an input of training set S and an index i.



```
ID3($S$, $A$)

INPUT: training set $S$, feature subset $A \subseteq [d]$
if all examples in $S$ are labeled by 1, return a leaf 1
if all examples in $S$ are labeled by 0, return a leaf 0
if $A = \emptyset$, return a leaf whose value = majority of labels in $S$
else :
  Let $j = \text{argmax}_{i \in A}$ Gain($S, i$)
  if all examples in $S$ have the same label
    Return a leaf whose value = majority of labels in $S$
  else
    Let $T_1$ be the tree returned by ID3($\{(\mathbf{x}, y) \in S : x_j = 1\}, A \setminus \{j\}$).
    Let $T_2$ be the tree returned by ID3($\{(\mathbf{x}, y) \in S : x_j = 0\}, A \setminus \{j\}$).
    Return the tree:
```

Figure 10: Pseudocode for the 'Iterative Dichotomizer 3' (ID3) (Shalev-Shwartz & Ben-David , 2014).

The Gain (S,i) function used above differs with the implementation of the algorithm used. ID3 uses 'Information Gain', while CART uses 'Gini Index'. However, for this study's particular problem, which is a regression problem (where the output variable takes continuous values), Information Gain and Gini Index do not apply and are both restricted to classification type problems containing discrete output classes.

When the output variable is continuous, i.e. response vector Y for each observation in variable matrix X, regression trees are used. The Gain (S,i) is based on *squared residuals minimization algorithm* which essentially aims to reduce the standard deviation of the outcome variable after a split is made using a feature.

It should be noted that the concept of how a decision tree works remains consistent if one is dealing with a classification or a regression problem. Gain (S, i) for classification aims to reduce the entropy/randomness of the outcome variable, while Gain (S,i) in a regression problem aims to reduce the standard deviation, thus both are essentially trying to converge towards a point of homogeneity in the target variable, although the method of measuring said homogeneity is different.

In summary, decision trees partition the data space into smaller discrete regions, and then apply a separate function to each region as opposed to polynomial regression methods that aim to build a global function and smooth it over the complete dataspace.

2.5.2: Random Forests

Assuming the concepts that govern the workings of a single decision tree are now clear, we can move ahead to Random Forests. The basic idea behind a random forest or for that matter any other ‘ensemble learning method’ in Machine Learning and Statistics comes from the approach that a group of weak models when combined together can produce a more robust model that produces stronger predictive performance (Opitz & Maclin, 1999). The reason a single decision tree is considered a weak predictor is because of the concept of the bias-variance trade-off (James et al., 2013).

Random Forests, as the name implies, is a group of decision trees that work towards achieving a common predictive task. The first algorithm for this approach of grouping multiple decision trees was developed by Tim Kam Ho (1995), which was a method to implement the ‘stochastic discrimination’ approach as proposed by Eugene Kleinberg (1996).

Leo Breiman and Adele Cutler (2001) developed an extension to the algorithm (now formally known as Random Forests) which has become the most widely used form of the algorithm today, and is also the algorithm that has been used in this thesis.

To get to the next step of Random Forests from decision trees, one needs to understand the technique of bootstrap aggregation. Bootstrap aggregation or ‘bagging’ proposed by Leo Breiman (1994), aims to improve the accuracy of Machine Learning models:

Assuming a training set of:

Input variables $X = x_1, x_2, \ldots. x_n$

Response Variables $Y = y_1, y_2, \ldots y_n$

A random subset of the training set is selected M times, and a decision tree is fit to each M subset. If the training set M contains a subset of X and Y, namely X_m, Y_m, and if a regression tree called f_m *is fitted to M.* The prediction of the overall model that combines the output of M regression trees on the samples that were left out (‘out-of-bag’ samples) M* from M subsets, will be the average of each tree:

$$\hat{f} = \frac{1}{M}\sum_{m=1}^{M} f_m(M^*)$$

Random Forests (Breiman, 2001) differs from the above algorithm of combining trees in that it uses a subsample of the input features at each split in the learning workflow. The number of features to be used at each split can be varied, however, the documentation of the algorithm recommends using p/3 (at each tree) features if the total number of features in the input space is p.

The above method is called ‘feature bagging’ as an extension to the general bagging or bootstrap aggregation, and this is done so that the correlation between the trees in the ensemble model is reduced. The reader is referred to Ho (2002) for an explanation as to why correlation between predictive trees is harmful to the final accuracy and the reason for using random subspace projection.

2.5.3: Extreme Gradient Boosted Trees

Extreme Gradient Boosted Trees (XGBoost) is a machine learning algorithm developed by Tianqi Chen (2014) that provides a framework for gradient boosting (Friedman, 2001) and was one the main tools used in this study.

The simplest way to think about the difference in Gradient Boosted Trees and Random Forests is that Random Forests essentially builds individual trees in parallel whereas in Boosted Trees the trees were built in a linear sequence. The learning method of Gradient Boosted Trees is slightly more complicated compared to Random Forests, and thus it re-ignites the concern about 'black-boxes'. With that in mind, an attempt at a simple explanation is made here by mostly following Chen's documentation (2014) of XGBoost.

Coming back to our objective function and assuming our learning parameters (θ) are a function of two variables i.e. the real outputs in our dataset y_i, and the prediction of our model $\hat{f}$. The regularization term is dependent on the prediction function f_m. Thus the objective function $obj(\theta)$ (as defined in section 2.5: Machine Learning and Workflow) can be re-written as (Chen, 2014):

$$obj(\theta) = \sum_{i}^{n} L\left(y_i, \hat{f}\right) + \sum_{m=1}^{M} \Omega(f_m)$$

Where, L = Training Loss Function, Ω = Regularization Term

As previously stated, gradient boosted trees are built in a sequential order, they thus assume a method of 'additive training' (Chen, 2014). At step 't' of the sequence the prediction of the model is said to be $\hat{y}_i^{\overline{(t)}}$, thus we have (Chen, 2014):

$$\hat{f}_i^{(0)} = 0$$
$$\hat{f}_i^{(1)} = f_1(x_i) = \hat{f}_i^{(0)} + f_1(x_i)$$
$$\hat{f}_i^{(2)} = f_1(x_i) + f_2(x_i) = \hat{f}_i^{(1)} + f_2(x_i)$$

…..

$$\hat{f}_i^{(t)} = \sum_{k=1}^{t} f_k(x_i) = \hat{f}_i^{(t-1)} + f_t(x_i)$$

Thus the objective function becomes (Chen, 2014):

$$obj^{t} = \sum_{i}^{n} L\left(y_i, \widehat{f}_i^{(t-1)} + f_t(x_i)\right) + \sum_{m=1}^{M} \Omega(f_m) + constant$$

As mentioned, before L can vary and can take different forms ranging in complexity. For instance an MSE function will assume an easy form after a differential with a first order term and a quadratic term. However, there might be more complex functions such as the logistic loss function below, and according to Chen (2014) XGBoost has the flexibility to deal with a variable loss function.

$$L(\theta) = \sum_{i} [y_i \ln(1 + e^{-\hat{f}i}) + (1 - y_i) \ln(1 + e^{\widehat{\hat{f}\imath}})]$$

Converting a general loss function ‘l’ into a Taylor series of the second order (Chen, 2014):

$$obj^{t} = \sum_{i=1}^{n} [l\left(y_i, \widehat{y}_i^{(t-1)}\right) + g_i f_t(x_i) + \frac{1}{2} h_i f_t^2(x_i)] + \Omega\left(f_t\right) + constant$$

Where,

$$g_i = \partial_{\hat{f}_{i(t-1)}} l(y_i, \hat{f}_i^{(t-1)})$$
$$h_i = \partial^2{}_{\hat{f}_{i(t-1)}} l(y_i, \hat{f}_i^{(t-1)})$$

Removing the constants, the objective function at a specific step ‘t’ becomes:

$$\textstyle\sum_{i=1}^{n} [\, g_i f_t(x_i) + \frac{1}{2} h_i f_t^2(x_i) + \Omega(f_t)]$$

Thus the objective function simplifies and only depends on g_i and h_i (first and second order derivatives of the loss function) . This enables xgboost to deal with custom loss functions by only taking g_i and h_i as input (Chen, 2014). For a more detailed description and the deduction of the complexity term$\Omega(f_m)$, the reader is referred to Chen (2014) and Friedman (2001).

2.5.4: Combining Mixed Effects with Random Forest

As noted before, the reservoir in the study area consists of stacked incised valley fills. During the process of building the model, it was realized that the well logs had slightly different responses based on the different valley fills and varying depositional settings. There have been three distinct valley fills recognized in the study area. The lower two valleys have comparatively cleaner sands and have been deposited in a high energy tidal flat setting near the margin of the channels, whereas the topmost valley has been interpreted to be deposited in a tidal channel with more interbedded mud.

It was important for the model to recognize this facies difference. Thus another feature was created that would categorize from which valley fill the log readings were derived. The values ranged from 1-3 for each valley fill category. However, as the model depends on the variance of an input variable to determine its importance, a categorical input feature with only three distinct values would rank the lowest in terms of variable importance within the model, and thus would be useless. Tree based models fail to recognize the clustered nature of longitudinal data.

In order to combat this problem, it was decided to implement a Linear Mixed Effect Model with the Random Forest algorithm. The inspiration for this approach came from the field of biology with the work of Wang et al. (2016), where they combined a mixed effect model with random forests in order to impute uncollected gene expression data in multi-tissues taken from different populations. The different populations in their study are analogous to our different valley fills in terms of their effect on the algorithm.

Mixed-effects models in a broad sense value the relationship between the target/output variable and the inputs, with coefficients that depend on the grouping of the data thus honouring the clustered nature of the data. Thus, Mixed-Effects models have two parts: 1) random effects, which are associated with data units drawn at random from a population i.e. the valley fills, and 2) fixed effects, associated with the input variables or well logs.

The Random Forest algorithm will inherently eliminate inputs that have high correlation, thus the need for a dimensionality reduction technique – e.g. Principal Component Analysis, is often deemed unnecessary in the case of Random Forests. The mixed effect model used here essentially becomes a random effects model. The reader is encouraged to read Pinherio and Bates (2004) for a detailed description on Mixed Effects modelling.

In summary, the water saturation values are predicted using Random Forest and Mixed Effects model combined. These predicted water saturation values are then used as an input to XGboost to estimate bitumen weight (Fig. 11). The reader is referred to Schlumberger (1991) and Crain (2017) for more details on the various well logs listed as inputs (Fig.11)

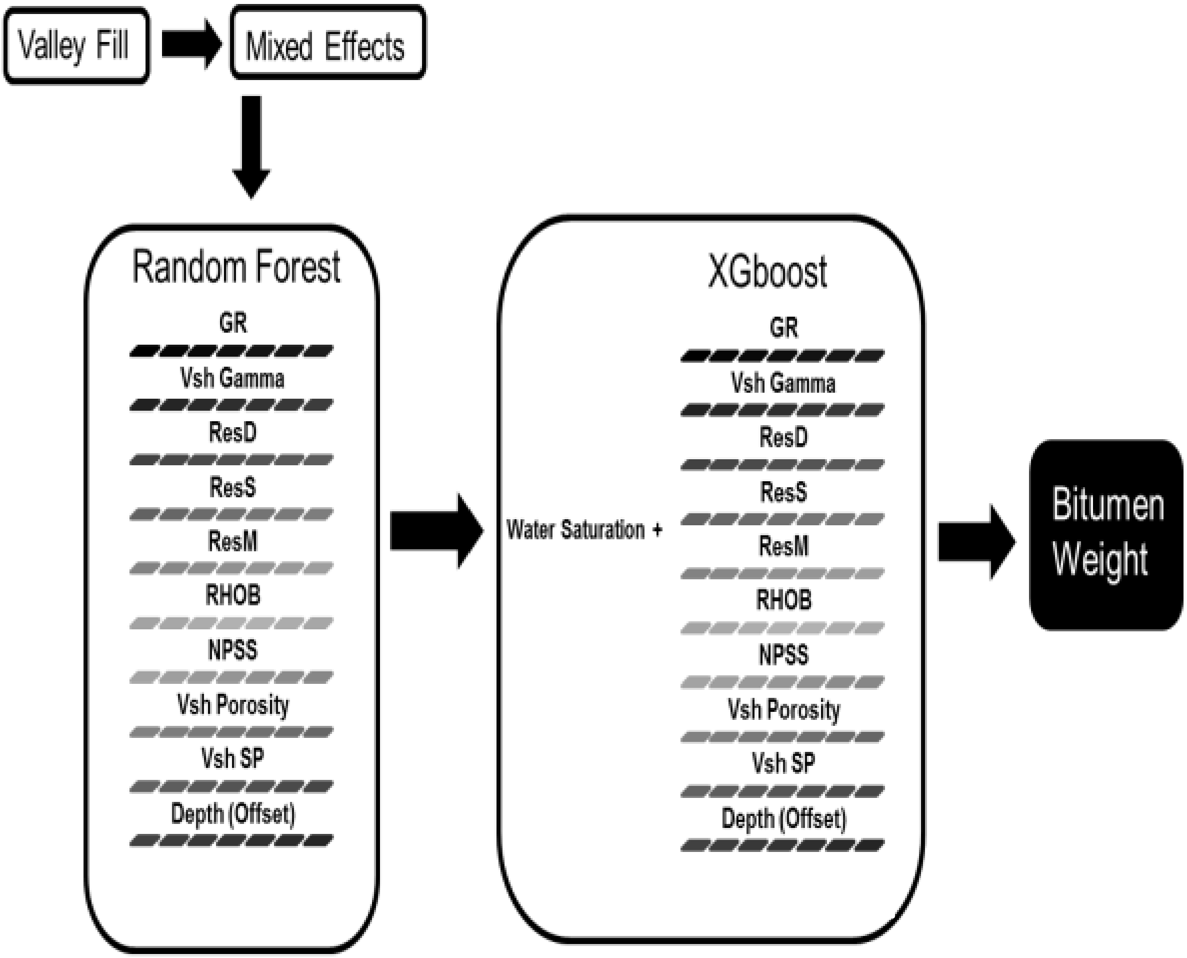


Figure 11: Algorithm workflow, showing the inputs for the model and the location of each algorithm in the overall workflow. GR: Gamma Ray, Vsh Gamma: Vshale from Gamma Ray. ResD: Deep Resistivity, ResS: Shallow Resistivity, ResM: Medium Resistivity, RHOB: Bulk Density, NPSS: Neutron Porosity, Vsh Porosity: Vshale from Neutron-Density Porosity, Vsh SP: Vshale from Spontaneous Potential

Chapter 3: Methods

3.1: Data Preparation and Normalization

After data cleaning, 40 wells were used in this study. There were 3724 Dean-Stark sample data points for comparison to the well log data. The well logs were visually inspected initially using depth plots, and any discrepancies in formation top picks were fixed. Average gamma ray log values in three different marine shale formations were computed by picking representative depths and averaging the corresponding values. The three shale units are: Fish Scale Formation shale, Joli Fou shale and the Clearwater capping shale (Fig. 12).

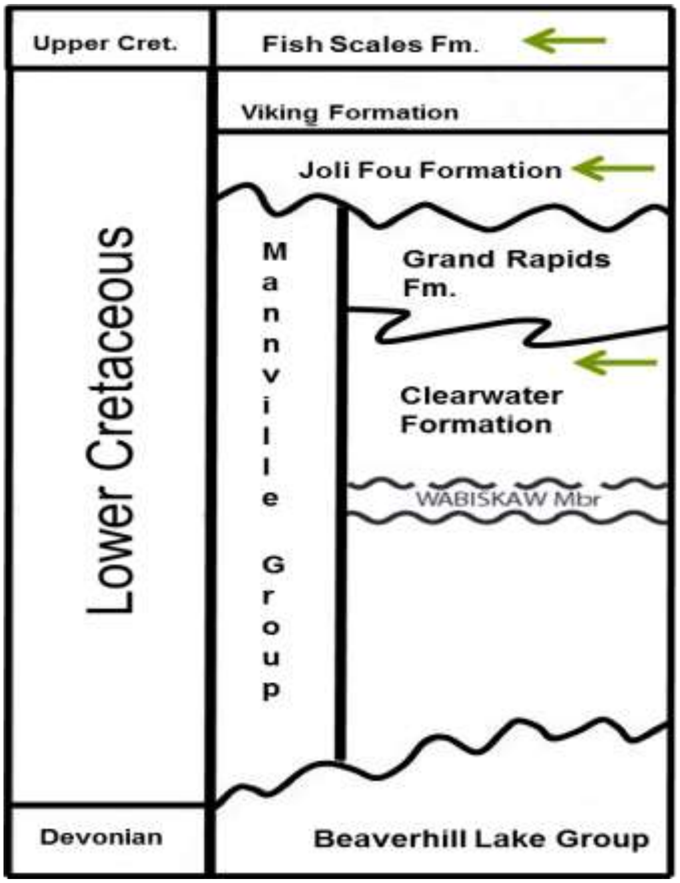
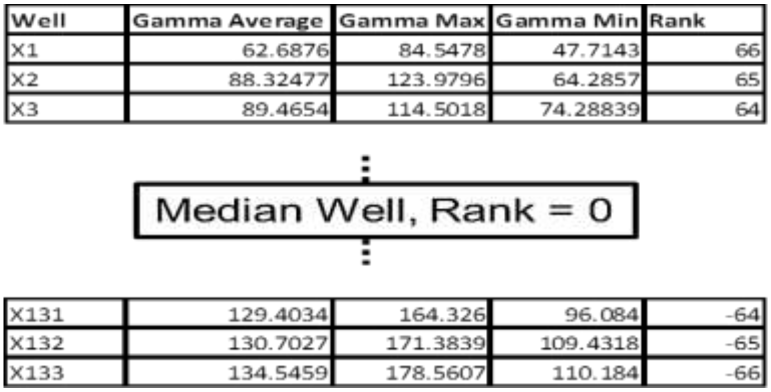


Figure 12: Stratigraphic location of the marine shales used for Gamma Log normalization. Modified from McCrimmon and Arnott (2002).

The wells were then ranked for all the three formations separately. The median well in the average gamma distribution getting a rank of zero, wells having an average gamma value less than the median received negative ranks from -1, -2, -3,…n. Wells with gamma average values above the median received positive ranks from 1, 2, 3…n (Fig. 13).

Rankings for all the three sets of wells (based on the 3 formations/intervals) were combined together and the well with lowest cumulative rank was labelled as the 'type well' that hypothetically represents a roughly 'median' sample of the variations in the area.

Once the type well was selected, 'Gamma Clean' and 'Gamma Shale' values were selected for each well in the project. 'Gamma Clean' is a value in the reservoir sands that corresponds to the lowest gamma log value. The 'Gamma Shale' value corresponds to the gamma value recorded in the shale unit (caprock) just above the reservoir. This normalization process ensures that discrepancies in log values from a well-to-well basis due to various reasons are taken out. These reasons could be due to wells having logs of different vintages, difference in logging parameters used by different companies while drilling, variations in hole condition, drilling mud or mud cake development that are not properly corrected etc.



Well	Gamma Average	Gamma Max	Gamma Min	Rank
X1	62.6876	84.5478	47.7143	66
X2	88.32477	123.9796	64.2857	65
X3	89.4654	114.5018	74.28839	64

Median Well, Rank = 0

X131	129.4034	164.326	96.084	-64
X132	130.7027	171.3839	109.4318	-65
X133	134.5459	178.5607	110.184	-66

Figure 13: Ranking Method based on Average Gamma Values in three different marine shales. Exact well identification labels have been removed for reasons of confidentiality.

Once all the gamma clean and gamma shale values were picked, the next step was to apply the 'Stretch and Squeeze' method for log normalization, as defined by Crain (2017):

$$NGamma = minG + (maxG - minG) * \frac{(G - lowG)}{(highG - lowG)}$$

Where,

NGamma = Normalized Gamma Log Value

minG = Gamma Clean in Type Well

maxG = Gamma Shale in Type Well

lowG = Gamma Clean in Well to be normalized

highG = Gamma Shale in Well to be normalized

The same approach was applied to the Spontaneous Potential log, which was also used an input to the model. Figure 14 shows SP log distribution before and after normalization. A flooding surface at the top of the reservoir was chosen as a common datum, and the original depth values were offset according it.

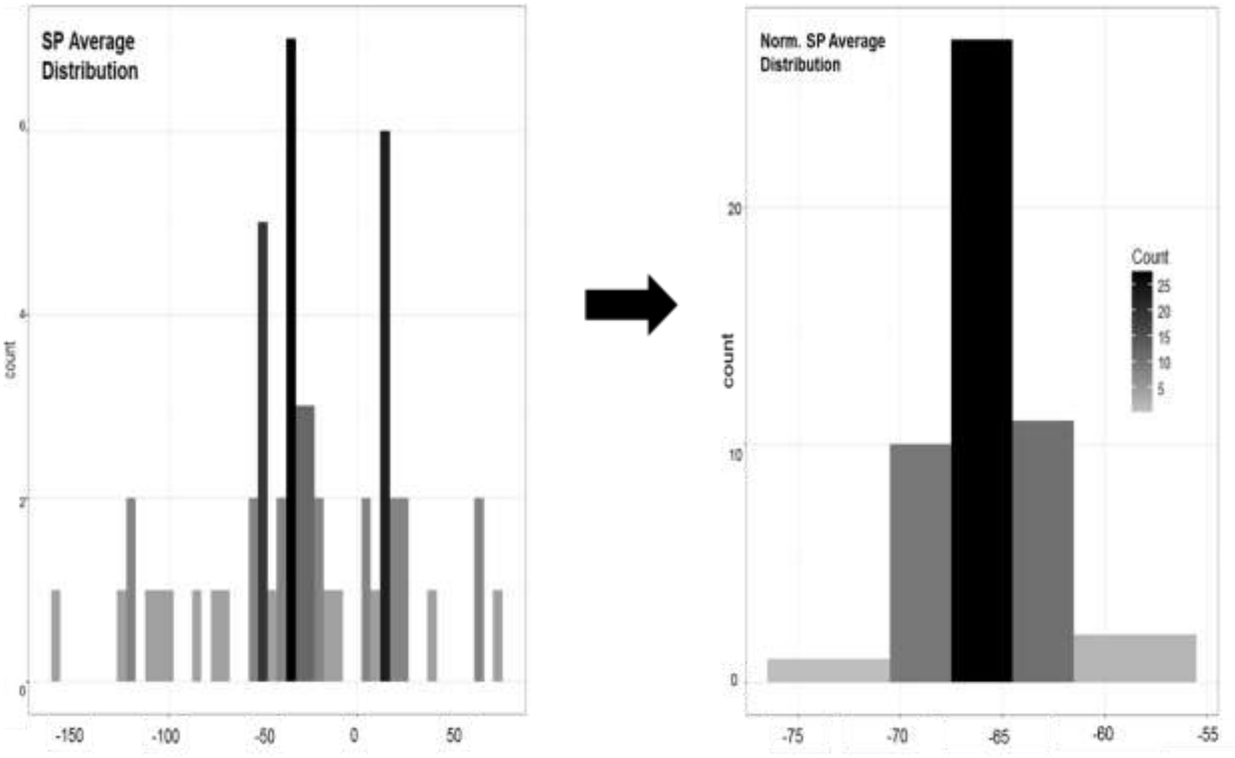


Figure 14: Distribution of Spontaneous Potential log before (left) and after (right) 'Stretch and Squeeze' normalization.

3.2: Feature Engineering

Shale volume estimation is one of the key steps for correcting porosity and water saturation due to the effects of clay bound water (Crain, 2017).

Once the logs were normalized, features for the data model were generated. These included Volume Shale (Vshale) calculations based on three different well logs:

1. Gamma Ray Log
2. Spontaneous Potential
3. Neutron Porosity – Density Porosity

The reason for using three different logs was to capture different sets of information. The Vshale values were calculated using the following equations:

$$Vsh_{(neut,dens)} = \frac{\emptyset_{neutron} - \emptyset_{density}}{\emptyset_{neutron(shale)} - \emptyset_{density(shale)}}$$

$$Vsh_{(Gamma)} = \frac{GR_{LogSignal} - GR_{CleanRock}}{GR_{Shale} - GR_{CleanRock}}$$

$$Vsh_{(SP)} = \frac{SP_{LogSignal} - SP_{CleanRock}}{SP_{Shale} - SP_{CleanRock}}$$

In addition, SP shale values were picked from the interbedded muds within the reservoir sands. This was done to designate the interbedded sands because the interbedding increases the variance of the SP log near the muds, and the algorithm is directly affected by the variance of the inputs. This step will make it easier for the model to pick up the muddy interbeds that are often blanketed out in the traditional Vshale calculations. Figure 15 shows the calculated Vshale values derived from the logs, compared to visual core facies description. The same technique was applied for Vsh (neutron, density) by picking neutron-shale and neutron-density values from the type well.

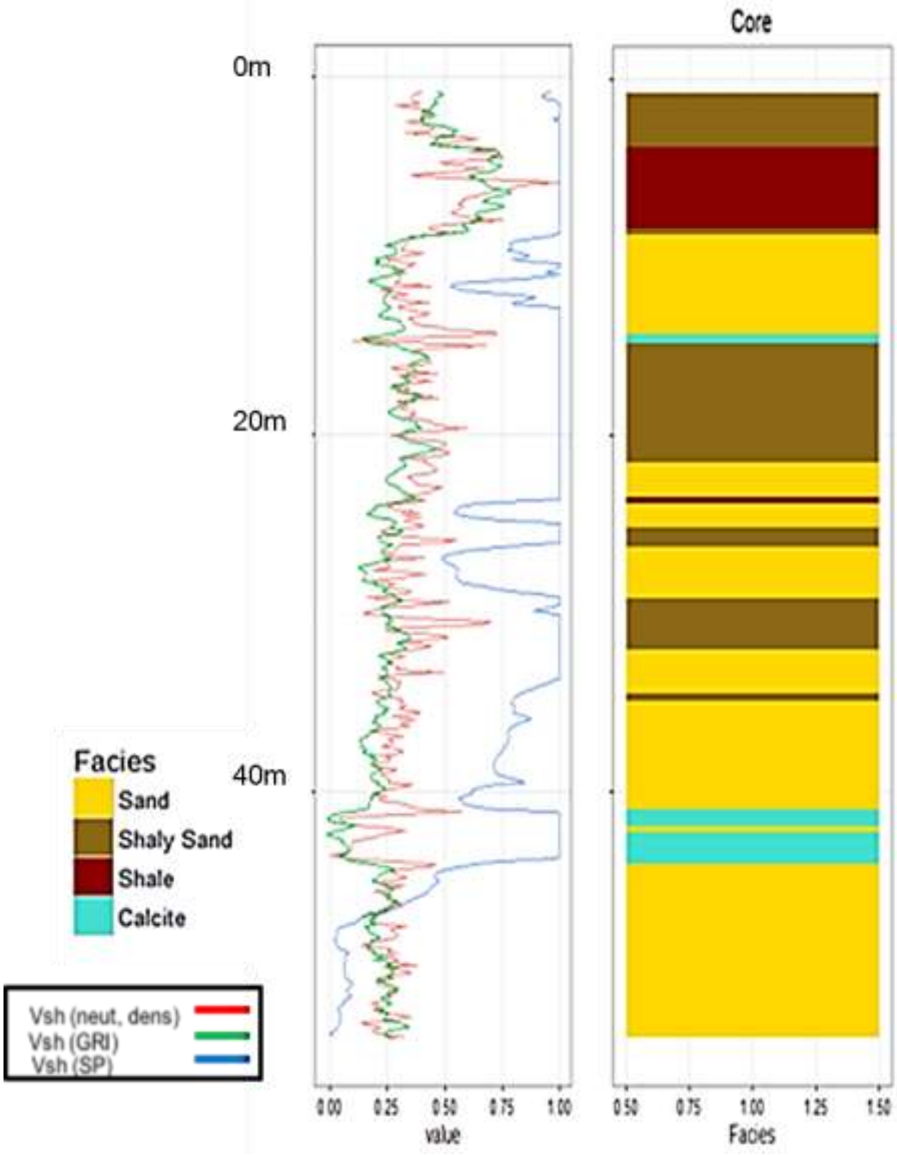


Figure 15: Comparison of the different Volume Fraction of Shale calculated versus Depth, using different well logs (left) against core facies descriptions (right). Calcite refers to intervals of sand cemented by calcite.

It is essential to capture the maximum amount of information on the volume of shale present in the reservoir. Vshale values greatly affect prediction accuracy, as shown in Figure 16. Average Vshale values calculated from SP log (picked on the mud interbeds) for each well were plotted against R-squared values for water saturation from the Modified Simandoux Equation versus core values (Dean Stark Analysis) in Figure 16. There are two clusters of values: wells with high

Vshale values have a lower prediction accuracy compared to wells with lower Vshale values. This shows that that the Modified Simandoux Equation has lower prediction accuracy for wells with higher amounts of mud interbeds (Vshale from SP was picked on the mud interbeds).

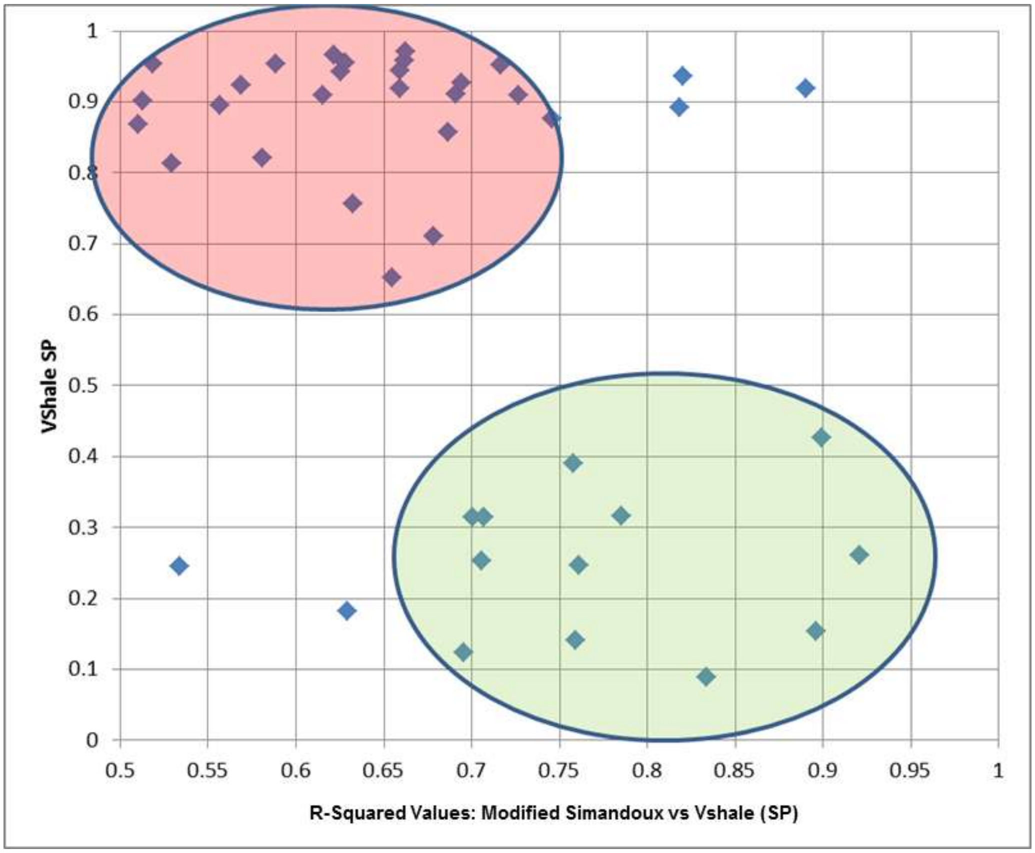


Figure 16: Vshale-SP plotted against R-Squared values for Water Saturation (Modified Simandoux compared to Dean Stark Analysis).

3.3: Optimization

For Random Forest optimization, the number of variables randomly sampled as candidates at each split was held constant at P/3 (P = total number of inputs). The number of trees to be built was tested over a range from 1 to 10,000, and the out-of-bag (Section 2.5.2) error rate was plotted versus the number of trees. In general, prediction accuracy increases with the number of trees used. However, the rate of improvement decreases as the number of trees increases. Thus, the benefit in prediction accuracy from using more trees will be lower than the cost in computation time for learning these additional trees. This is evident when the out-of-bag error rate is plotted versus the number of trees (Fig. 17). The rate of increase in prediction accuracy suddenly decreases close to 100 trees in the current model. Although a proper computational time cost analysis was not done for the study, the optimal number of trees judging by the graph was set to 500. The benefit of adding additional trees was deemed minimal, and individual training runs took approximately 3 hours. Many trials were necessary while testing various workflows. Thus, models with too many trees proved tedious to work with.

For XGBoost parameters, a grid containing learning rate values (eta) of {0.01, 0.05, 0.1 }, maximum depth (of a single tree) = {2,3,4,6,8,10,14} was tested using cross validation in the CARET package **(**Kuhn, 2008**).** The CARET package provides an automatic workflow for optimization of machine learning algorithms and thus saves much time especially when working with algorithms that have not been modified (XGboost in our study). The reader is referred to Kuhn (2008) for an introduction to the CARET package. Optimum value for eta (through the CARET package) was found to be 0.01 and maximum depth was found to be 3.

The code for this study was written in R programming language (R Core Team, 2016). Version: 3.3.1 (2016-06-21). Platform: x86_64-w64-mingw32/x64 (64-bit)

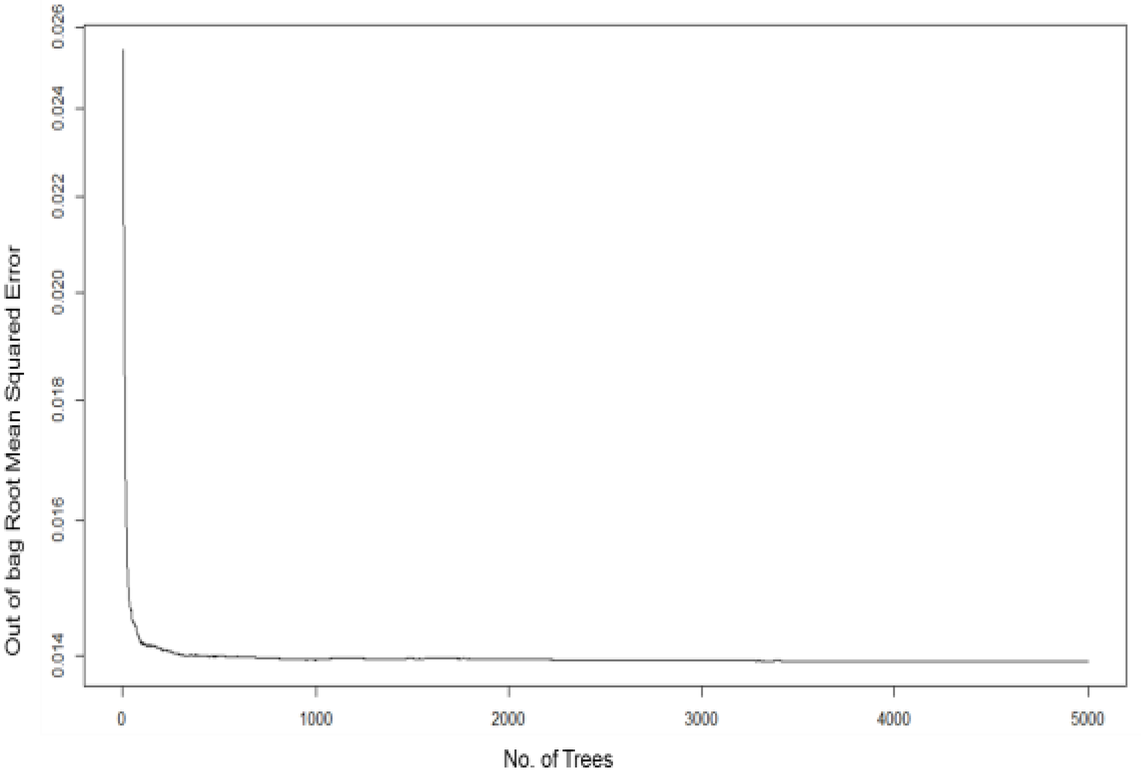


Figure 17: Out-of-bag error rate vs. number of trees, used for estimating the optimum number of trees.

Chapter 4: Results

The predicted water saturation and bitumen weight values were compared to their corresponding values derived from the Dean-Stark analysis done in the lab. The assumption of this study is that the Dean-Stark analysis values are correct. It is understood that this assumption might not always hold true, as there could be various sources of error in the Dean-Stark procedure. Differences in log derived and core derived values are unavoidable. Especially in a reservoir with thin beds, conventional wireline logging is prone to errors due to the limitations of vertical resolution and the different scale of sampling compared to core measurements.

To analyze the performance of our model, two phases of testing were implemented. The first phase included taking out 4 wells that had the best correlation (i.e. highest R-squared) between the measured core water saturation values and the values obtained by the Modified Simandoux Equation (Table 1). These wells were not used in the training set. For the second phase, the model was tested by multi-fold cross validation: with the number of folds equal to the number of wells in the data set (Table 2). Each well was left out of the training set once – the model was trained with all the other wells and the prediction was done on the well that was left out. The mean R-squared values were analyzed along with the root mean squared error (RMSE). Average R-squared and RMSE values for all the test wells across each fold show that the Machine Learning model had higher R-squared values and lower RMSE. Figure 18 shows a visual representation of the results for one of the wells. Predicted bitumen weight values from Machine Learning (ML) are in green, Modified Simandoux (MS) are in blue, with values derived from Dean Stark Analysis on core samples asblack dots. Tracks on the left show 4 out of the 11 inputs used in the model: Gamma Ray (GR), Deep Resistivity (ResD), Bulk Density (RHOB) and Neutron Porosity (NPSS).

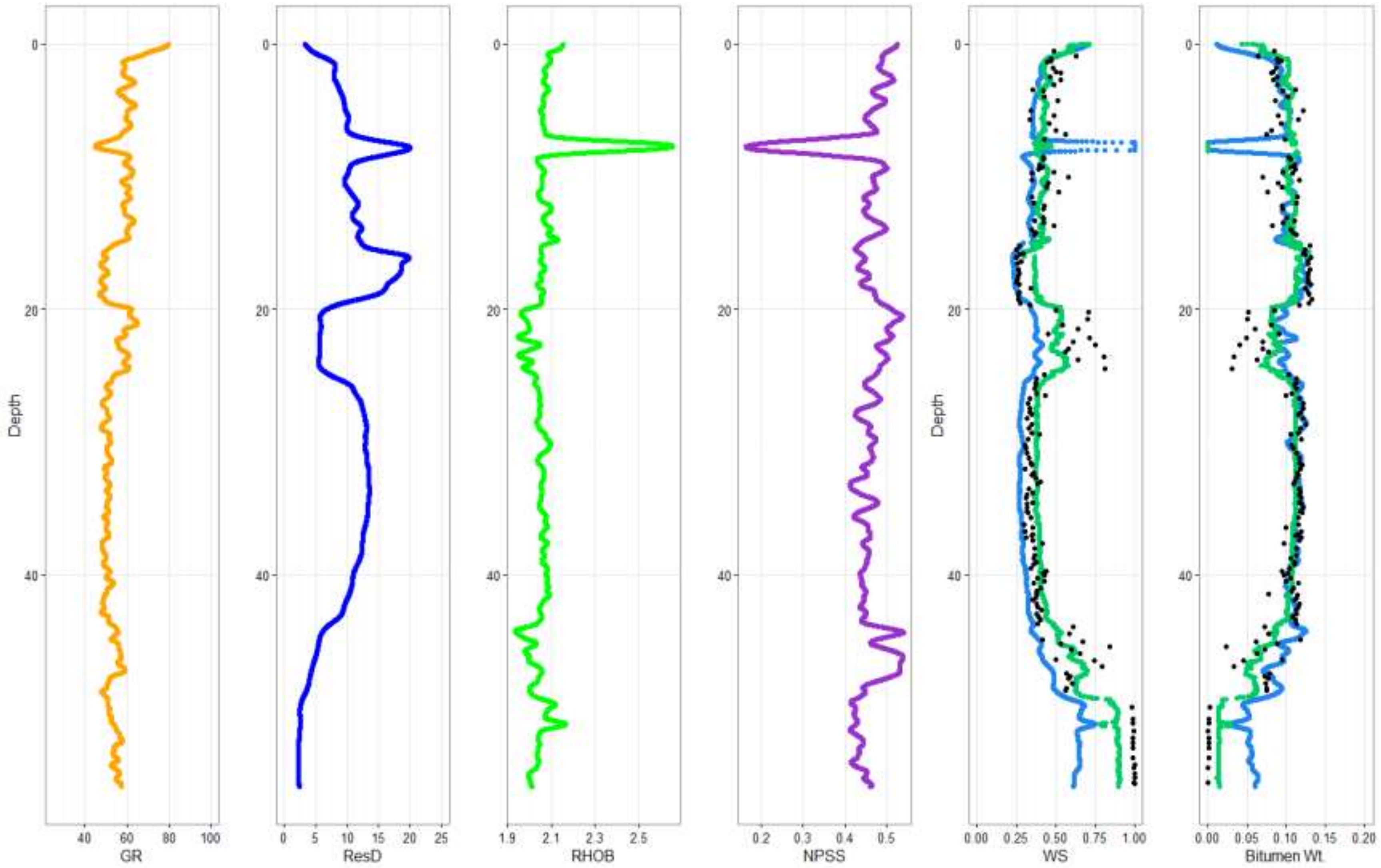


Figure 18: Predicted bitumen weight values from Machine Learning (ML - Green), Modified Simandoux (MS - Blue) with values derived from Dean Stark Analysis on core samples. Tracks on the left show 4 out of the 11 inputs used in the model: GR, ResD, RHOB, NPSS

4.1: Regular Test Set Results

	Water Saturation - Dean Stark vs Predicted		Bitumen Weight - Dean Stark vs Predicted			
	R-Squared		R-Squared		RMSE	
Well	MS	ML	MS	ML	MS	ML
1	0.89	0.93	0.79	0.9	0.022	0.014
2	0.92	0.87	0.8	0.68	0.015	0.018
3	0.9	0.84	0.83	0.87	0.026	0.025
4	0.9	0.91	0.87	0.9	0.021	0.023
Mean	0.9	0.89	0.82	0.84	0.021	0.02

Table 1: Results for regular test for 4 wells with the highest R-squared values using the Modified Simandoux Equation used as test wells. WS: Water Saturation, BW: Bitumen Weight, MS: Modified Simandoux, ML: Machine Learning. Last two columns show RMSE

4.2: Multi Fold Cross Validation Results

The full cross validation results are shown in the appendix, the mean values for accuracy metrics are reported below.

Water Saturation - Dean Stark vs Predicted		Bitumen Weight - Dean Stark vs Predicted			
R-Squared		R-Squared		RMSE	
MS	ML	MS	ML	MS	ML
0.65	0.68	0.58	0.66	0.024	0.022

Table 2: Mean values for cross validation results. Abbreviations same as Table 1.

Chapter 5: Discussion

Judging by the R-squared and RMSE values for the test sets (see Table 1 and Table 2), the machine learning model performed equally as well if not better than the Modified Simandoux Equation. As mentioned in methods, it is important to note that comparing water saturation results from well-log derived models and core analyses is challenging in the best of circumstances. The reason for this is the difference in scale and methodology between porosity derived from well logs and the porosity derived from the cores. It is probably unrealistic to expect them to match perfectly. Differences in the two values can arise from various reasons. For example, especially when clay is present in the reservoir, the core sample may not dry completely during the Dean-Stark process. For this reason, petrophysicists compare bitumen weight values between well logs and core because they are more likely to match, and for the same reason we provide accuracy metrics on both output variables. The latter point is evident from Figure 18, which shows closer overlap between the different models of bitumen weight compared to predictions for water saturation.

Although cross-validation is a more robust method for analyzing the accuracy of a predictive machine learning model, the regular test process was done keeping in mind the fairly large temporal gap between the drilling dates of these wells. Some of the wells have core analysis done in discrete intervals and thus might yield unreliable core saturation results, especially considering the heterogeneity in the reservoir due to interbedded mud beds. This is probably the reason why some of the wells in the cross-validation results have a substantially lower accuracy metric for both the conventional and the machine learning model. It is probably not a coincidence that the regular test set of wells with the best matches are wells that have a comparatively younger drill date with more consistent and continuously tested cores.

Analyzing the residuals (Observed core values – Predicted modelled values) of a predictive model gives important insights and is an important part of result analysis. Plotting residuals against observed values can give details about the nature of the prediction (overestimation/underestimation).

Figures 19 and 20 show residuals plotted against observed bitumen weight values (Dean-Stark Analysis) for Modified Simandoux and Machine Learning respectively. Judging by the plots we can see that both the models tend to underestimate high values of observed bitumen weight and also overestimate low values of bitumen weight. Both of these observations expected around a regression line bounded by an upper and a lower limit.

The colour in the plots reflects the different valley fill units in the reservoir. There is a segregation of residuals with respect to the valley fills. Valley B which has the lowest oil saturation is expectedly offset towards the lower bitumen weight values. Comparing the two predictive models we observe that the Machine Learning model does a better job in separating out the valley fills, and also has better prediction accuracy in the D valley. D valley is more tidally influenced and has more mud interbeds. Mud interbeds are a challenge for the Modified Simandoux Equation as shown in Figure16. This demonstrates that implementing the different valley fills and combining the mixed effects model with Random Forest improved the prediction accuracy. Achieving similar consistency with the Modified Simandoux equation would probably require time-consuming empirical adjustment for each reservoir facies.

The values close to zero bitumen weight are samples near the bottom water (i.e. below the bitumen-bearing zone, where the reservoir is almost entirely water saturated), which is evident from Figures 21 and 22. The data points in the two figures have been coloured by deep-resistivity and depth (offset from the top of the reservoir) respectively. With additional time these could be easily filtered out of the input data and likely allow a more fine-tuned Machine Learning model for the bitumen-bearing zone.

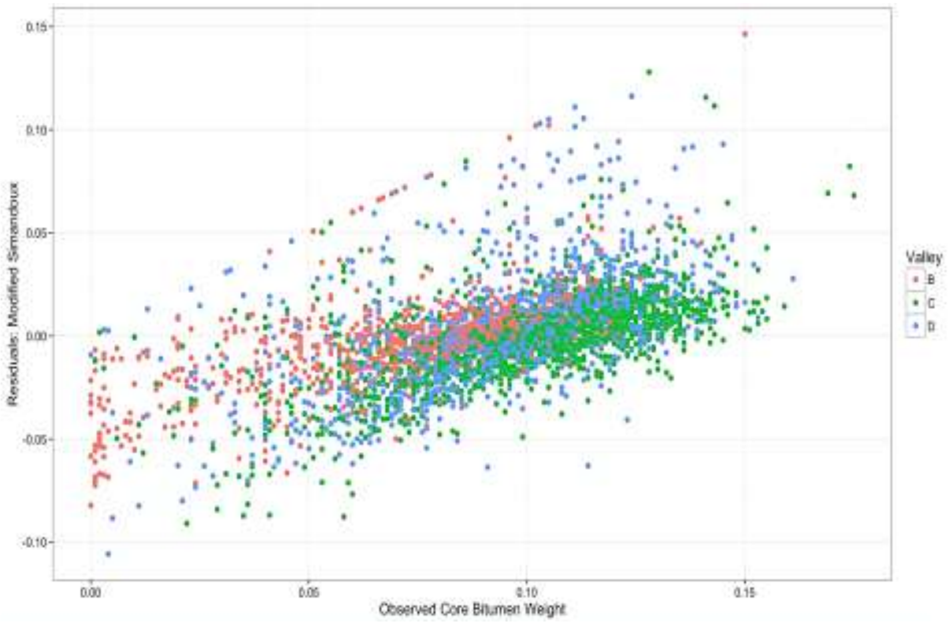


Figure 19: Observed Core Bitumen Weight (x:axis) vs. Residuals for the Modified Simandoux Equation (Observed – Predicted, y:axis) coloured by the different valley fills. Valley B: Red, Valley C: Green, Valley D: Blue.

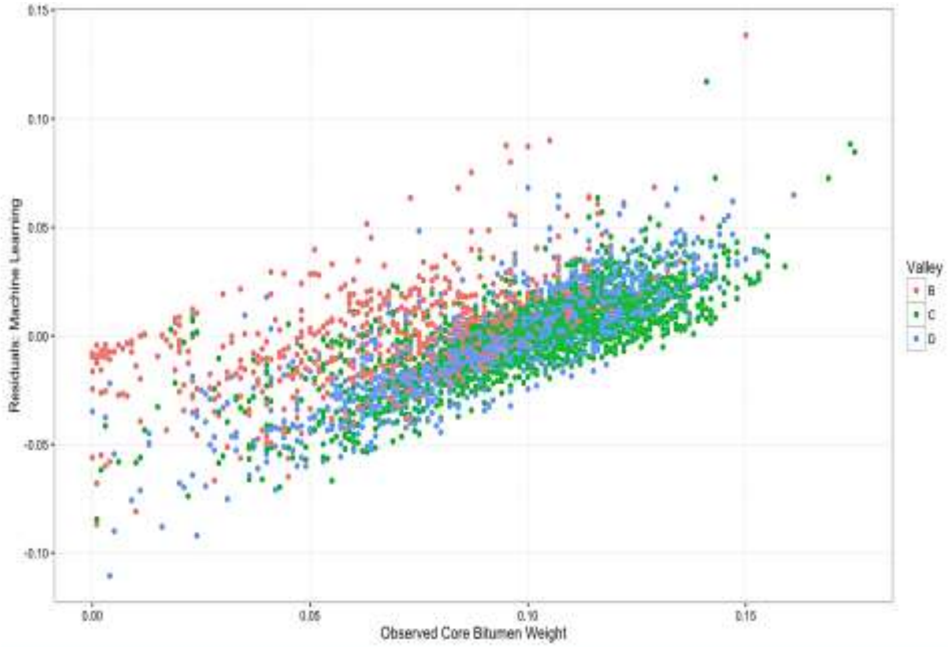


Figure 20: Observed Core Bitumen Weight (x:axis) vs. Residuals for the Machine Learning model (Observed – Predicted, y:axis) coloured by the different valley fills. Valley B: Red, Valley C: Green, Valley D: Blue.

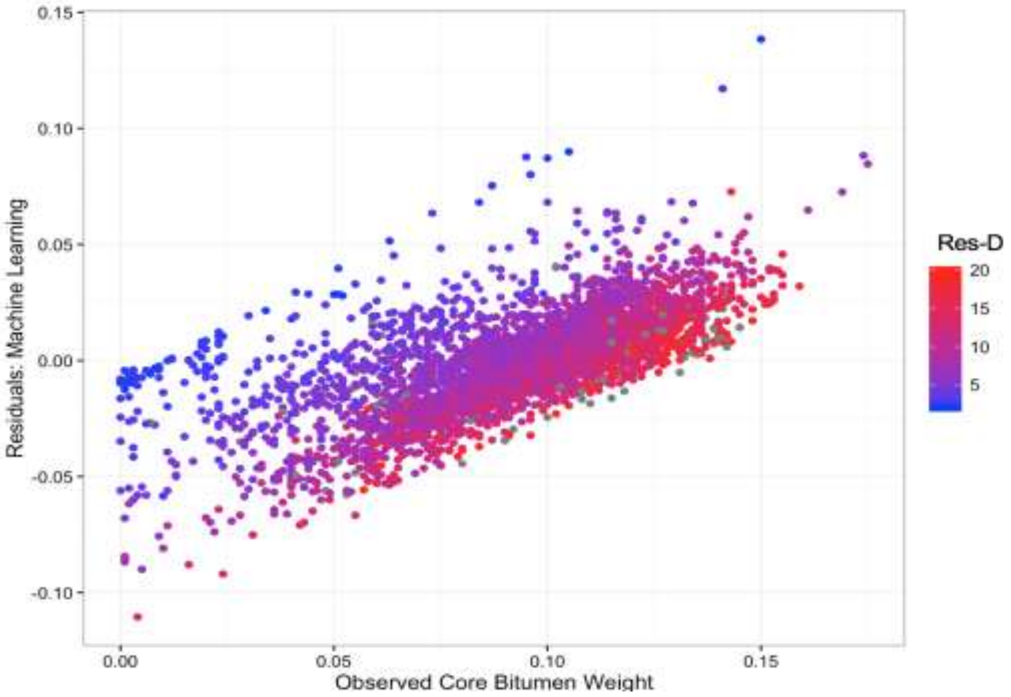


Figure 21: Observed Core Bitumen Weight (x:axis) vs. Residuals for the Machine Learning model (Observed – Predicted, y:axis) coloured by Deep-Resistivity values. Red: High Resistivity Blue: Low Resistivity, the grey dots are outside the range of the Resistivity values on the colour gradient (0-20).

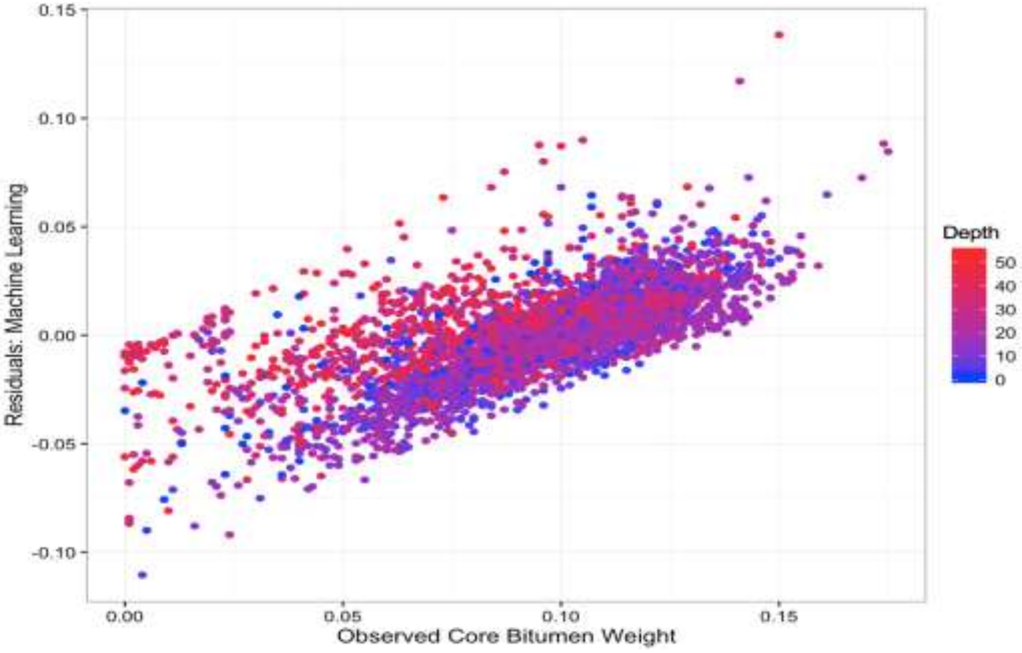


Figure 22: Observed Core Bitumen Weight (x:axis) vs. Residuals for the Machine Learning model (Observed – Predicted, y:axis) coloured by Depth (offset) values. Red: Deeper Blue: Shallower.

More data generally is a good thing when it comes to building a data-driven model. However, with geology, it also means accounting for more variation. Thus blanketing a large number of wells in a complex depositional environment with a model (Modified Simandoux or Machine Learning) is always going to be difficult due to the heterogeneity of the setting. In addition to the geology, the issue of legacy wells in the area and the logging inconsistency between wells is also an issue. This, combined with the resolution limits of the logging tools that makes it difficult to account for the thin mud interbeds, all lead to the reduction in prediction accuracies. Nevertheless, this study shows great potential for application of Machine Learning to similar types of problems in petroleum geology settings.

Any discussion about Machine Learning is incomplete without the argument of black-boxes, and although a very good attempt was made in this study to ensure the transparency of the model by using simpler algorithms, there is no doubt that the Modified Simandoux equation is the more transparent model, giving the petrophysicist more control. Machine Learning, although fast and seemingly more accurate, remains a black box to some extent and it is up to the user to decide which model works best for the problem in hand.

In the words of George Box (1976): “All models are wrong, some models are useful….

It would be remarkable if any system existing in the real world could be exactly represented by any simple model. However, cunningly chosen parsimonious models often do provide remarkably useful approximations. For example, the law PV = RT relating pressure P, volume V and temperature T of an "ideal" gas via a constant R is not exactly true for any real gas, but it frequently provides a useful approximation and furthermore its structure is informative since it springs from a physical view of the behavior of gas molecules.

For such a model there is no need to ask the question "Is the model true?" If "truth" is to be the "whole truth" the answer must be "No". The only question of interest is "Is the model illuminating and useful?” In that respect, in the author’s opinion, Machine Learning has been “illuminating and useful” in the present application.

5.1: Future Work

There is potential for future work using similar approaches that include Machine Learning and well log data. Machine Learning has been used to effectively generate electro-facies (Bhatt and Helle, 2002; Dubois et al., 2007). The generated electro-facies can then be used as inputs by creating a second tier of mixed effects and combining it with the saturation model. Feature engineering plays an important role in the accuracy of a model, as demonstrated above. Features that can amplify the interbedded muds prove to be extremely useful. One such feature that can be generated and used is a derivative of the gamma ray log. Gamma ray log derivatives have been used in the past for stratigraphic studies (Reid et al., 1989; Vermeer and Alkemade, 1992). The derivative log can be used to amplify the response of lithology contacts within a well. Such a log can be constructed by using the following equation of the slope of a line:

$$Gamma_{Derivative} = \frac{GR_2 - GR_1}{Depth_2 - Depth_1}$$

Where, $Depth_2$ and $Depth_1$ are the consecutive depth readings with corresponding gamma ray log values of GR_2 and GR_1 respectively. Applying this equation to the entire gamma ray log in a well generates a derivative of the gamma ray log.

Figure 23 shows a derivative log for the well used in Figure 18. Positive spikes indicate a transition from sand to mud (low radioactivity to high radioactivity) . A negative spike represents a transition from mud to sand. The derivative log can then be used as an ‘engineered input’ to a machine learning model that predicts either facies or fluid saturation.

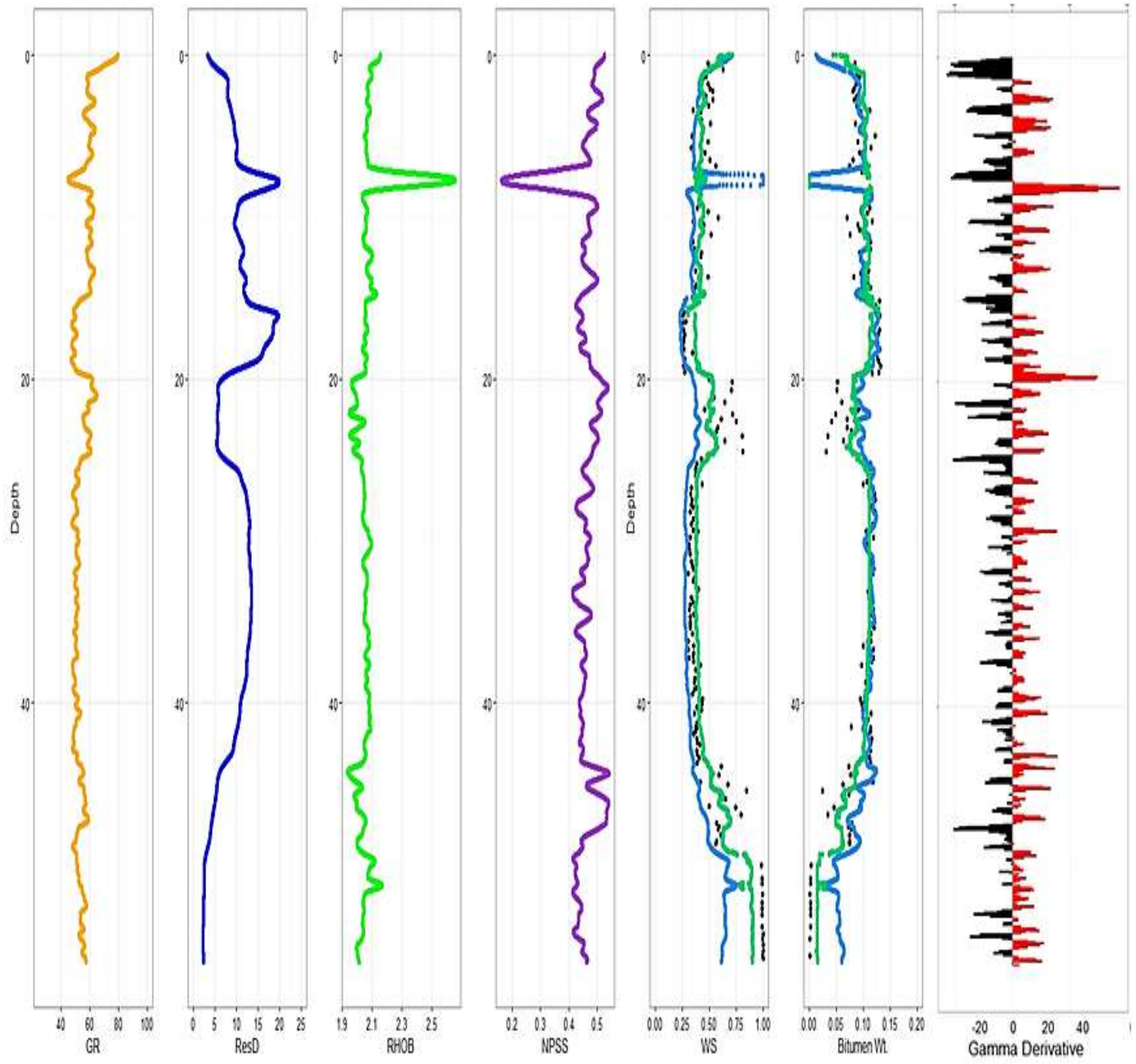


Figure 23: A derivate of the gamma ray log (seventh track) showing points of contact between sand and mud. Red: Positive - sand to mud. Black: Negative - mud to sand. Other tracks are same as Figure 18.

Chapter 6: Conclusions

In summary, the machine learning model is a supplemental guide for fluid saturation prediction instead of a replacement. Some parallels can be drawn between the Archie Equation and the machine learning model in the sense that they both involve an empirically driven process.

Coming back to the two questions posed in the introduction of this thesis:

1) Judging by the accuracy metrics of R-Squared and RMSE on both the testing phases, it can be concluded that Machine Learning was successful in predicting fluid saturation values and had comparable if not better prediction accuracy than the Modified Simandoux Equation.
2) Some of the variation in R-squared values is related to variations in the Dean-Stark analysis values. These variations are not captured in the well logs and could be due to various reasons. The most obvious is some mud interbeds are too thin to be captured as discrete beds in well logs due to resolution limits of the tools. There could also be discrepancies in a few core samples from the lab, which could be outliers. Additionally, there is a lack of consistency arising from the temporal gap due to legacy wells, which could also reduce the prediction accuracy of both models. Finally, there is also variation in the accuracy due to the different valley fills in the reservoir. Thus, implementing the mixed effect models proved to be an important step in the workflow.

References

Amiri, M., Ghiasi-Freez, J., Golkar, B. and Hatampourd, A. (2015): Improving water saturation estimation in a tight shaly sandstone reservoir using artificial neural network optimized by imperialist competitive algorithm - A case study; Journal of Petroleum Science and Engineering, v. 127, p. 347-358.

Archie, G. (1942): The electrical resistivity log as an aid in determining some reservoir characteristics; Transactions of the American Institute of Mechanical Engineers, v. 146, p. 54-62.

Badgley, P. C. (1952): Notes on the subsurface stratigraphy and oil and gas geology of the Lower Cretaceous series in central Alberta (Report and seven figures); Geological Survey of Canada, Paper No. 52-11, p. 12.

Bardon, C. & Pied, B. (1969): Formation water saturation in shaly sands; Society of Professional Well Log Analysts 10th Annual Logging Symposium Transactions, p. 19.

Benyon, B.M. and Pemberton, S.G. (1992): Ichnological signature of a brackish water deposit: An example from the Lower Cretaceous Grand Rapids Formation, Cold Lake oil sands area, Alberta. In: Pemberton, S.G. (ed.). Application of Ichnology to Petroleum Exploration: A Core Workshop, SEPM Core Workshop, v. 17, p. 199-221.

Benyon, C., Leier, A., Leckie, D.A., Webb, A., Hubbard, S.M., Gehrels, G. (2014): Provenance of the cretaceous Athabasca oil sands, Canada: Implications for continental-scale sediment transport; Journal of Sedimentary Research, v. 84, p. 136-143.

Bhatt, A., Helle, H.B. (2002): Determination of facies from well logs using modular neural networks; Petroleum Geoscience, v. 8(3), p. 217-228.

Blum, M. and Pecha, M. (2014): Mid-Cretaceous to Paleocene North American drainage reorganization from detrital zircons; Geology, v. 42(7), p. 607-610.

Box, G. E. (1976): Science and Statistics; Journal of the American Statistical Association, v. 71(356), p. 791-799.

Breiman, L. (2001): Random Forests; Machine Learning, v. 45(1), p. 5-32.

Breiman, L., Friedman, J., Olshen, R. & Stone, C. (1984): Classification and regression trees. Monterey, CA: Wadsworth & Brooks/Cole Advanced Books & software.

Canadian Association of Petroleum Producers. (2017): What are Oil Sands? http://www.capp.ca/canadian-oil-and-natural-gas/oil-sands/what-are-oil-sands

Cant, D. J. and Stockmal, G. S. (1989): The Alberta foreland basin: relationship between stratigraphy and Cordilleran terrane-accretion events; Canadian Journal of Earth Sciences, v. 26(10), p. 1964-1975.

Cheadle, B.A., Dudley, J.S., Eastwood, J.E., Lovell, R.W.W., Reed, K.W., Stancliffe, R.P.W. and Van Wagoner, J.C. (1995): Integrated reservoir description for resource management at Cold Lake, Alberta; In: Proceedings, Exploration, Evaluation, and Exploitation 1995, The Economic Integration of Geology and Formation Evaluation, Canadian Society of Petroleum Geologists/Canadian Well Logging Society (CSPG/CWLS) Joint Symposium, Core Conference, June 1-2, Calgary, p. 14.

Chen, C., Lin, Z. (2006): A committee machine with empirical formulas for permeability prediction; Computers & Geoscience, v. 32, p. 485-496

Chen, T. (2014): Introduction to Boosted Trees; http://homes.cs.washington.edu/~tqchen/pdf/BoostedTree.pdf

Clack, W. (1967): Sedimentology of the Mannville Group in the Cold Lake area, Alberta; Unpublished M.Sc. thesis. University of Calgary, Alberta. 95p.

Crain, E. (2017): Crain's Petrophysical Handbook. https://www.spec2000.net/index.htm

Creaney, S. and Allan, J. 1992. Petroleum systems in the foreland basin of Western Canada. In: Foreland basins and foldbelts. R.W. Macqueen and D.A. Leckie (eds.). American Association of Petroleum Geologists, Memoir 55, p. 279-308.

Creaney, S., Cole, K.S., Brooks, P.W., Fowler, M.G., Osadetz, K.G., Macqueen, R.W., Snowdon, L.R., Roedoeger, C.L. (1994): Petroleum Generation and Migration in the Western Canada Sedimentary Basin; In: Geological Atlas of the Western Canada Sedimentary Basin, G.D. Mossop and I. Shetsen (comp.), Canadian Society of Petroleum Geologists and Alberta Research Council. http://www.ags.gov.ab.ca/publications/wcsb_atlas/atlas.html

Dean, E. and Stark, D. D. (1920): A convenient method for determination of water in petroleum and other organic emulsions; The Journal of Industrial & Engineering Chemistry, v. 12(5), p. 486-490.

Dubois, M.K., Bohling, G.C., Chakrabarti, S. (2007): Comparison of four approaches to a rock facies classification; Computer & Geosciences, v. 33(5), p. 599-617.

Energy Alberta. (2017): Talk about SAGD. http://www.energy.alberta.ca/OilSands/pdfs/FS_SAGD

Friedman, J. H. (2001): Greedy Function Approximation: A Gradient Boosting Machine; The Annals of Statistics, v. 29(5), p. 1189-1232.

Fustic, M., (2013): The Athabasca Oil Sands Deposit From Basin to Molecular Scale – Recent Insights and Emerging Questions. http://www.cspg.org/documents/Technical/Webcasts/webcast%20slides/2013/fustic%20presentation.pdf

Fustic, M., Ahmed, K., Brough, S., Bennett, B., Bloom, L., Asgar-Deen, M., Jakonala, O., Spencer, R., Larter, S. (2006): Reservoir and Bitumen Heterogeneity in Athabasca Oil Sands; In: Geological Controls on Reservoir and Bitumen Heterogeneities in Athabasca Oil Sands Deposit, Fustic, M., Doctoral thesis, Calgary, p. 640-652.

Glaister, R. (1959): Lower Cretaceous of southern Alberta and adjacent areas. American Association of Petroleum Geologists, Bulletin, v.43, p. 590-640.

Harrison, D., Glaister, R. and Nelson, H. (1959): Reservoir description of the Clearwater oil sand, Cold Lake, Alberta. In: Meyer R.F. and Steele C.T. (eds.). The Future of Heavy Crude Oils and Tar Sands. McGraw Hill. p. 264-280.

Hayes, B.J.R., Christopher, J.E., McKercher, B., Minken, D., Tremblay, D.W.M., Fennell, J., Smith, D.G. (1994): Cretaceous Mannville Group of the Western Canada Sedimentary Basin; In: Geological Atlas of the Western Canada Sedimentary Basin, G.D. Mossop and I. Shetsen (comp.), Canadian Society of Petroleum Geologists and Alberta Research Council. http://www.ags.gov.ab.ca/publications/wcsb_atlas/atlas.html

Helle, H., Bhatt, A. and Ursin, B. (2001): Porosity and permeability prediction from wireline logs using artificial neural networks: a North Sea case study; Geophysical Prospecting, v. 49(4), p. 431-444.

Hein, F. Weiss, J., and Berhane, M. (2007): Cold Lake Oil Sands Area: Formation Picks and Correlation of Associated Stratigraphy; Alberta Energy and Utilities Board, EUB/AGS Geo-Note 2006-03.

Ho, T. K. (1995): Random Decision Forests. Montreal; Proceedings of the 3rd International Conference on Document Analysis and Recognition, Montreal, QC, v.1, p. 278-282.

Ho, T. K. (2002): A Data Complexity Analysis of Comparative Advantages of Decision Forest Constructors; Pattern Analysis and Applications, v.5, p. 102-112.

Hubbard, M. S., Smith, D.G., Nielsen, H., Leckie, D.A., Fustic, M., Spencer, R.J., Bloom, L. (2011): Seismic geomorphology and sedimentology of a tidally influenced river deposit, Lower Cretaceous Athabasca oil sands, Alberta, Canada; The American Association of Petroleum Geologists, v. 95(7), p. 1123-1145

Jackson, P. (1984): Paleogeography of the Lower Cretaceous Mannville Group of Western Canada. In: Elmworth - Case Study of a Deep Basin Gas Field. J.A. Masters (ed.). Tulsa, American Association of Petroleum Geologists, Memoir 38, p.49-78.

James, G., Witten, D., Hastie, T. and Tibshirani, R. (2013): An Introduction to Statistical Learning, Springer.

Jardine, D. (1974): Cretaceous oil sands of Western Canada. In: Hills, L.V. (ed.). Oil Sands, Fuel of the Future; Canadian Society of Petroleum Geologists, Memoir 3, p. 50-67.

Kleinberg, E. (1996): An Overtraining-Resistant Stochastic Modeling Method for Pattern Recognition; Annals of Statistics, v. 24, p. 2319-2349.

Kuhn, M., (2008): Building predictive models in R using the caret package; Journal of Statistical Software, v. 28(5), p. 1-26.

Labrecque, P. A., Hubbard, S. M., Jensen, J. L. and Nielsen, H. (2011): Sedimentology and stratigraphy architecture of a point bar deposit, Lower Cretaceous McMurray Formation, Alberta, Canada; Bulletin of Canadian Petroleum Geology, June, v. 59, p. 147-171.

Loranger, D. (1951): Useful Blairmore microfossil zones in central and southern Alberta, Canada; Bulletin of American Association of Petroleum Geologists, v. 35, p. 2348-2367.

Marsland, S. (2014): Machine Learning. Edition-2. NY: CRC Press.

Meijer Dries, N.C. and Johnston, D.I. 1996. Famennian and Tournaisian biostratigraphy of the Big Valley, Exshaw and Bakken Formations, southeastern Alberta and Southwestern Saskatchewan. Bulletin of Canadian Petroleum Geology, v. 44(4), p. 683-694.

McCrimmon, G. (1996): Sedimentology and sequence stratigraphy of the Lower Cretaceous Clearwater Formation, Cold Lake, Alberta; Unpublished M.Sc. thesis, University of Ottawa

McCrimmon, G. G. and Arnott, R. (2002): The Clearwater Formation, Cold Lake, Alberta: a worldclass hydrocarbon reservoir hosted in a complex succession of tide-dominated deltaic deposits; The Bulletin of Canadian Petroleum Geology, v.50, p. 370-392.

Minken, D.F. (1974): The Cold Lake oil sands: geology and a reserve estimate. In: Hills, L.V., (ed.), Oil Sands, Fuel of the Future; Canadian Society of Petroleum Geologists, Memoir 3, p. 84-99.

Mohaghegh, S., Arefi, R., Ameri, S. (1996): Petroleum reservoir characterization with the aid of artificial neural networks; Journal of Petroleum Science and Engineering, v. 16, p. 263-274

Mossop, G.D. and Shetsen, I., comp. (1994): Geological atlas of the Western Canada Sedimentary Basin; Canadian Society of Petroleum Geologists and Alberta Research Council. http://ags.aer.ca/reports/atlas-of-the-western-canada-sedimentary-basin.htm

Naus, A. W. (1945): Cretaceous stratigraphy of Vermilion area, Alberta, Canada; American Association of Petroleum Geologists (AAPG), AAPG Bulletin, v. 29, p. 1605-1629.

Opitz, D. & Maclin, R. (1999): Popular ensemble methods: An empirical study; Journal of Artificial Intelligence Research, v.11, p. 169-198.

Pal, M. (2005): Random forest classifier for remote sensing classification; International Journal of Remote Sensing, v. 26(1), p. 217-222

Paskey, J., Steward, G. and Williams, A. (2013): The Alberta Oil Sands Then and Now: An Investigation of the Economic; Environmental and Socials Disclosures Across Four Decades., Edmonton: OSRIN.

Pinherio, J. and Rogers, J. H. (2004): Mixed Effects Models in S and S-Plus; Statistics and Computing Series, Springer.

Quinlan, J. R. (1986): Induction of Decision Trees; Machine Learning, v.1, p. 81-106.

Quinlan, J. R. (1996): Improved use of continuous attributes in c4.5; Journal of Artificial Intelligence Research, v. 4, p. 77-90.

R Core Team (2016): R: A language and environment for statistical computing. R Foundation for Statistical Computing, Vienna, Austria.
https://www.R-project.org/.

Riediger, C.L. (1990): Rock-Eval/TOC data from the Lower Jurassic “Nordegg Member”, and the Lower and Middle Triassic Doig and Montney formations, Western Canada Sedimentary Basin, Alberta and British Columbia; Geological Survey of Canada, Open File Report 2308.

Reid, I., Linsey, T. and Frostick, L.E. (1994): Automatic bedding discriminator for use with digital wireline logs; Marine and Petroleum Geology, v.6, p. 364-369.

Ripley, B.D. (1996): Pattern Recognition and Neural Networks; Cambridge University Press, Cambridge, p.403.

Rudkin, R. (1964): Lower Cretaceous; In: Geologic History of Western Canada. R.G. McCrossan and R.P. Glaister (eds.). Calgary, Alberta Society of Petroleum Geologists, p. 156-168.

Rumelhart, D.E., Hinton, G.E. and Williams, R.J., (1985): Learning internal representations by error propagation (No. ICS-8606). California University San Diego La Jolla Institute for Cognitive Science

Schlumberger (1991): Log interpretation principles/application; Schlumberger educational series seventh printing.

Shalev-Shwartz, S. and Ben-David, S. (2014): Understanding Machine Learning: From Theory to Algorithms; Cambridge University Press.

Shen, K.Q., Ong, C.J., Li, X.P., Zheng, H., Wilder-Smith, E.P.V. (2007): A feature selection method for multi-level mental fatigue EEG classification.; IEEE Transactions on Biomedical Engineering, v. 54(7), p. 1231-7

Svetnik, V., Liaw, A., Tong, C., Culberson, J.C., Sheridan, R.P., Feuston, B.P. (2003): Random Forest: A Classification and Regression Tool for Compound Classification and QSAR Modeling; Journal of Chemical Information and Modelling, v. 43, p. 1947-58.

Vermeer, P.L., Alkemade, J.A.H. (1992): Multiscale segmentation of well logs; Mathematical Geology, v.24(1), p. 27-43

Vigrass, L. (1965): General geology of Lower Cretaceous heavy oil accumulations in western Canada; Journal of Canadian Petroleum Technology. v. 4, p. 168-176.

Wang, J. et al. (2016): Imputing Gene Expression in Uncollected Tissues Within and Beyond GTEx; American Journal of Human Genetics, v. 98(4), p. 697-708.

Wendt, W., Sakurai, S. and Nelson, P. (1986) Permeability prediction from well logs using multiple regression; In: L.W. Lake and H.B. Carroll Jr.(eds.). Reservoir Characterization; Academic Press, p. 181-221.

Wong, P.M., Jang, M., Cho, S., Gedeon, T.D. (2000): Multiple permeability predictions using an observational learning algorithm; Computers & Geoscience, v. 26, p. 907-913

Winsauer, W., Shearin, J. H., Masson, P. and Williams, M. (1952): Resistivity of brine-saturated sands in relation to pore geometry; American Association of Petroleum Geologists Bulletin, v. 36, p. 253-277.

Yao, B., Koshla, A., Fei-Fei, L. (2011): Combining randomization and discrimination for fine-grained image categorization; IEEE Conference on Computer Vision and Pattern Recognition (CVPR)

Appendix 1: Cross Validation Results:

	Water Saturation – Dean Stark vs Predicted		Bitumen Weight – Dean Stark vs Predicted			
	R-Squared		R-Squared		RMSE	
Well	MS	ML	MS	ML	MS	ML
1	0.66	0.70	0.58	0.69	0.02	0.02
2	0.75	0.81	0.74	0.80	0.02	0.02
3	0.46	0.29	0.40	0.33	0.03	0.03
4	0.67	0.59	0.70	0.66	0.02	0.02
5	0.62	0.60	0.56	0.61	0.02	0.02
6	0.51	0.43	0.25	0.31	0.02	0.02
7	0.55	0.65	0.35	0.37	0.04	0.03
8	0.38	0.60	0.56	0.73	0.03	0.02
9	0.45	0.70	0.37	0.72	0.03	0.03
10	0.21	0.43	0.24	0.41	0.02	0.02
11	0.81	0.82	0.76	0.83	0.03	0.02
12	0.82	0.91	0.59	0.81	0.02	0.02
13	0.89	0.93	0.79	0.90	0.02	0.01
14	0.69	0.74	0.64	0.74	0.02	0.02
15	0.51	0.60	0.54	0.60	0.02	0.03
16	0.69	0.74	0.59	0.72	0.02	0.02
17	0.92	0.87	0.80	0.68	0.02	0.02

18	0.71	0.71	0.58	0.57	0.03	0.03
19	0.57	0.60	0.66	0.60	0.02	0.02
20	0.70	0.81	0.74	0.80	0.02	0.02
21	0.61	0.69	0.66	0.69	0.02	0.02
22	0.76	0.87	0.76	0.87	0.02	0.02
23	0.63	0.67	0.64	0.59	0.02	0.02
24	0.90	0.84	0.83	0.87	0.03	0.02
25	0.68	0.62	0.71	0.62	0.02	0.04
26	0.83	0.79	0.70	0.75	0.02	0.02
27	0.53	0.54	0.23	0.39	0.03	0.03
28	0.65	0.68	0.54	0.65	0.03	0.03
29	0.59	0.69	0.54	0.68	0.03	0.02
30	0.58	0.68	0.63	0.73	0.04	0.03
31	0.90	0.91	0.87	0.90	0.02	0.02
32	0.70	0.75	0.74	0.63	0.02	0.03
33	0.79	0.81	0.59	0.78	0.02	0.02
34	0.38	0.34	0.38	0.34	0.02	0.02
35	0.72	0.74	0.64	0.75	0.02	0.02
36	0.63	0.64	0.71	0.72	0.02	0.02
37	0.69	0.82	0.32	0.73	0.02	0.02
38	0.62	0.66	0.61	0.68	0.02	0.02
39	0.63	0.67	0.51	0.63	0.02	0.02
40	0.53	0.42	0.27	0.48	0.03	0.02

Appendix 2: Plots of a few wells showing the modelled values vs Core values

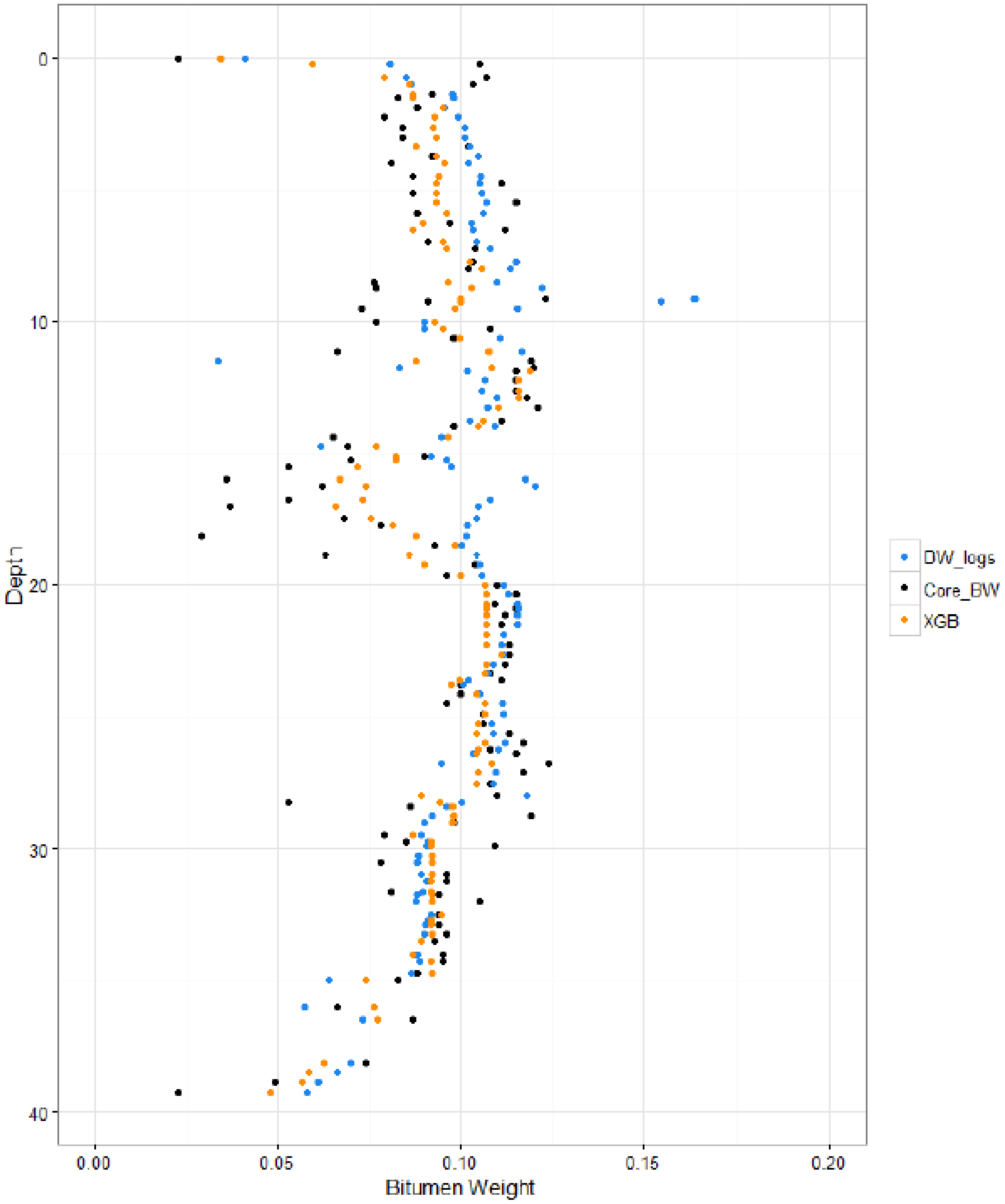
Legend:

BW_ Logs (Blue) : Bitumen Weight using Water Saturation Values derived from the Modified Simandoux Equation

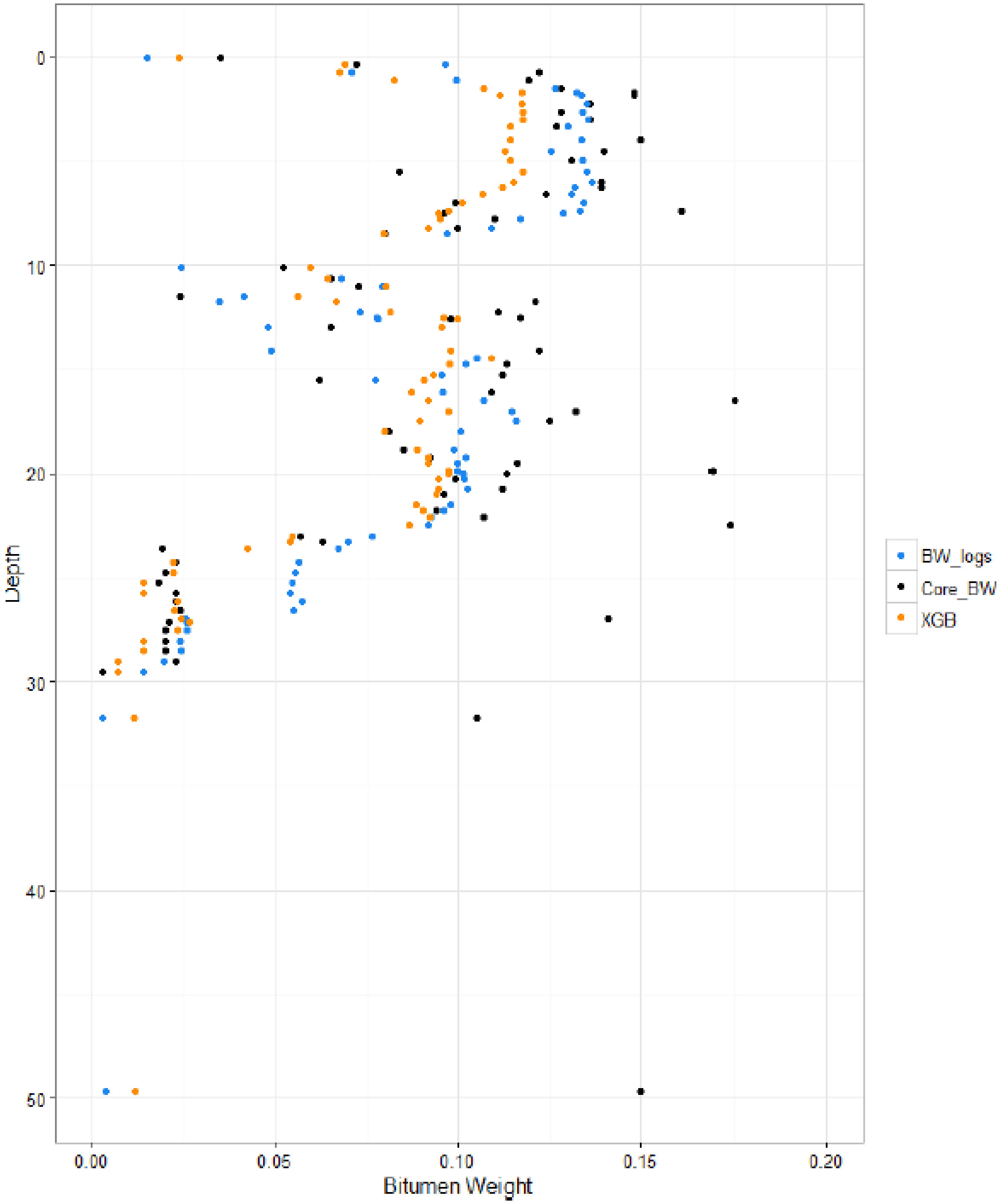
Core_BW (Black): Bitumen Weight values derived from the Dean Stark Analysis on core samples

XGB (Orange): Bitumen Weight values using Machine Learning

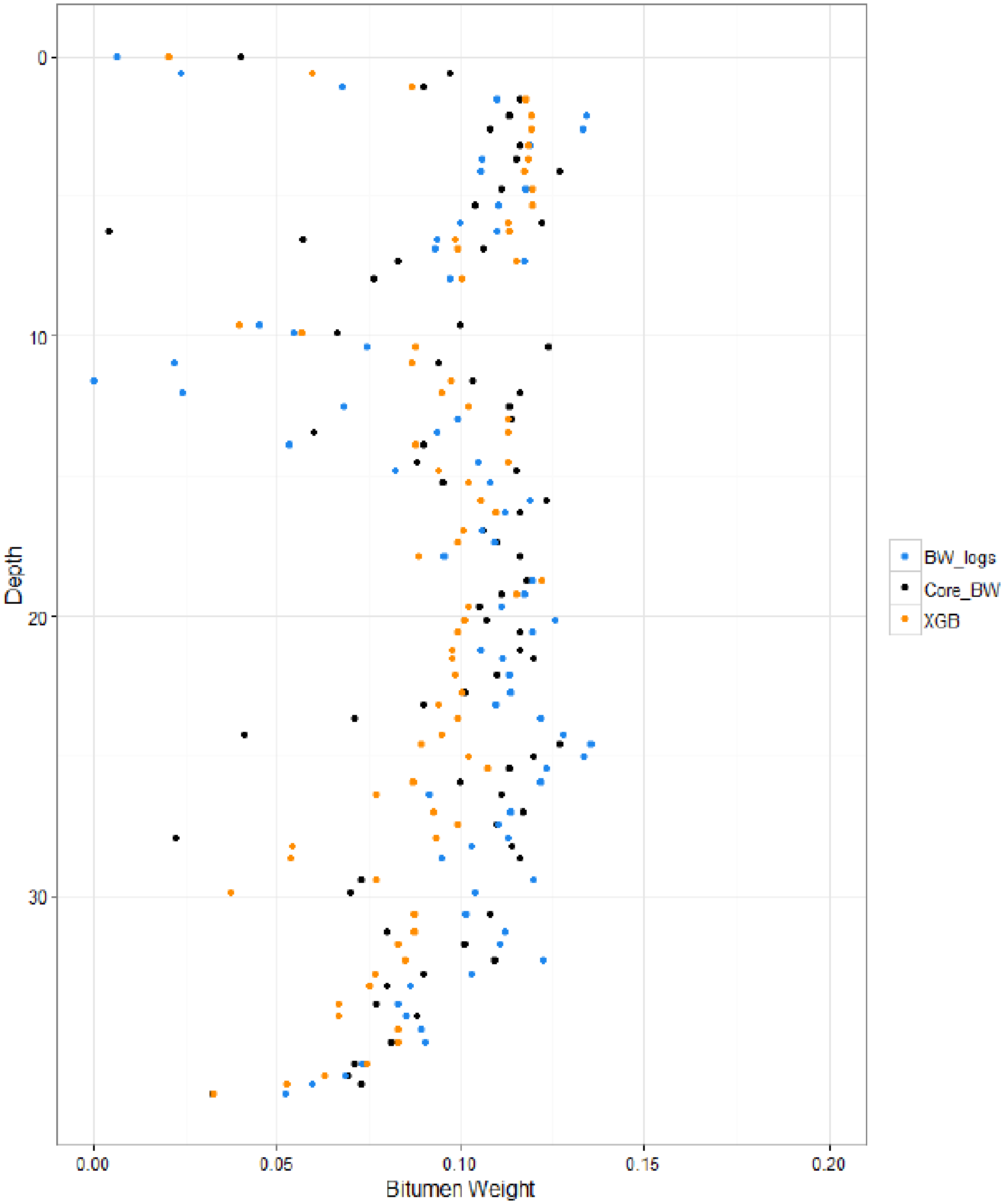
Well X1:


Depth
0
10
20
30
40
0.00
0.05
0.10
0.15
0.20
Bitumen Weight
● DW_logs
● Core_BW
● XGB

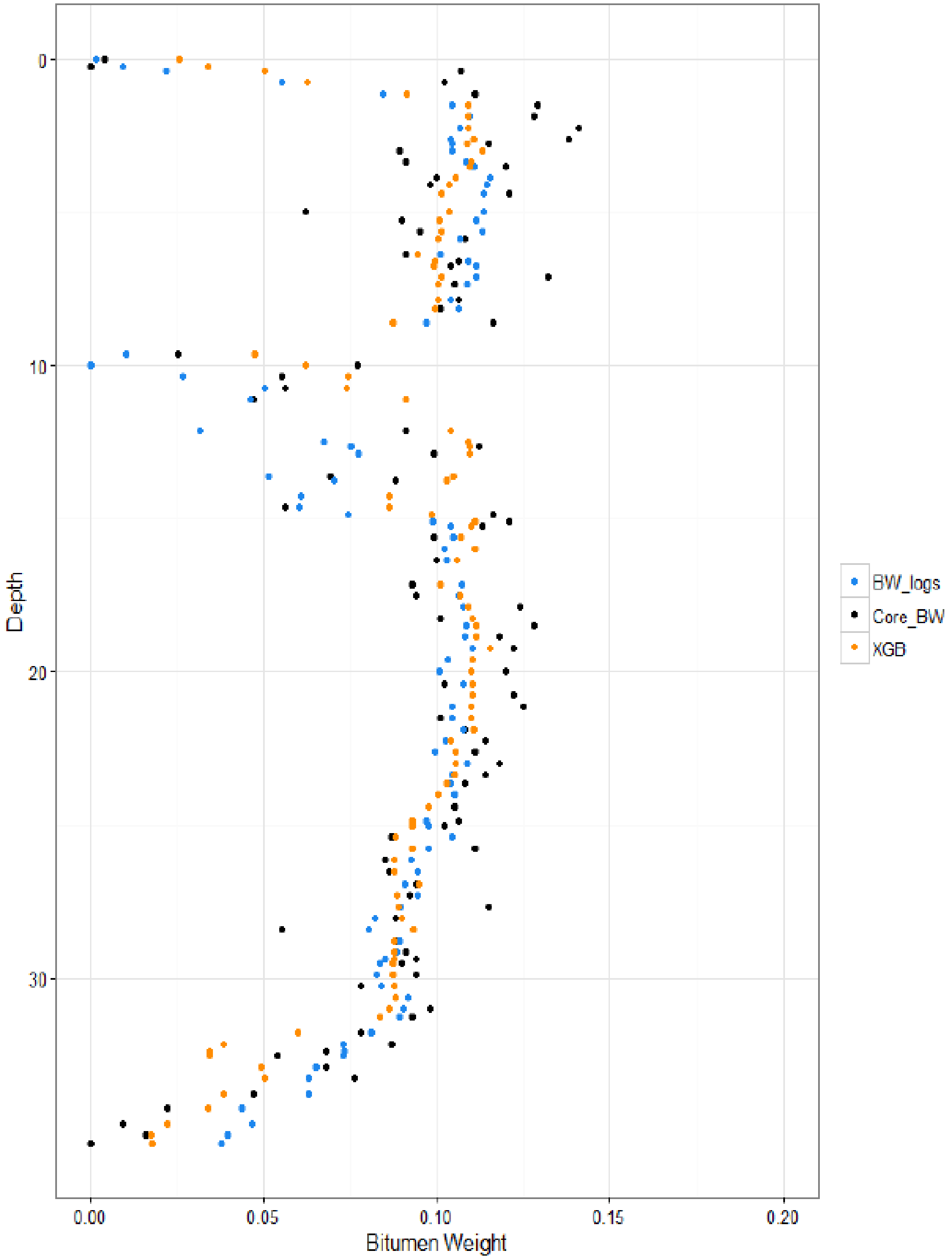
Well X2:


Depth
0
10
20
30
40
50
0.00
0.05
0.10
0.15
0.20
Bitumen Weight
● BW_logs
● Core_BW
● XGB

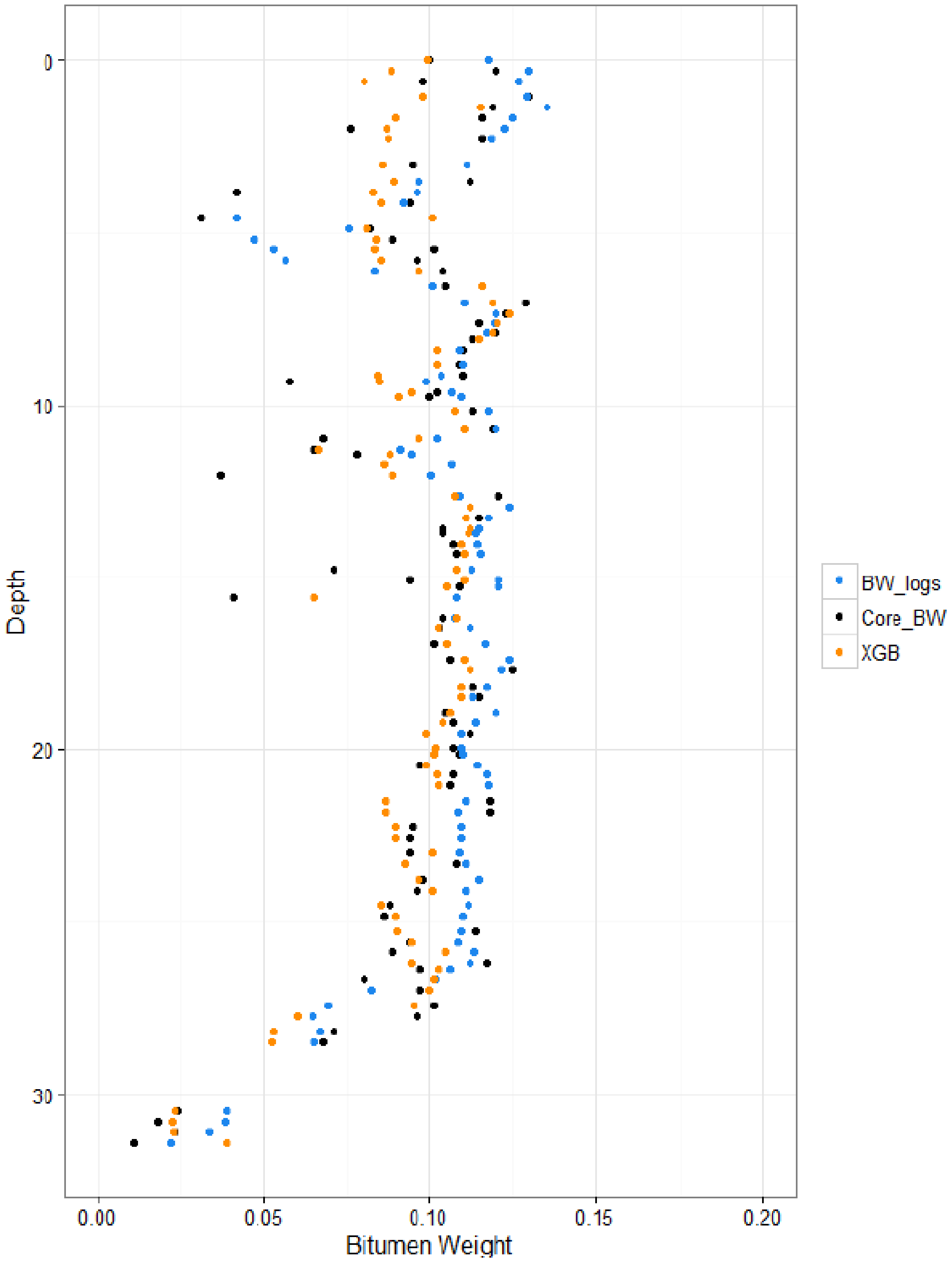
Well X3:


Depth
0
10
20
30
0.00
0.05
0.10
0.15
0.20
Bitumen Weight
BW_logs
Core_BW
XGB

Well X4:


Depth
0
10
20
30
0.00
0.05
0.10
0.15
0.20
Bitumen Weight
● BW_logs
● Core_BW
● XGB

Well X5:


0
10
20
30
Depth
0.00
0.05
0.10
0.15
0.20
Bitumen Weight
BW_logs
Core_BW
XGB

Appendix 3: Code

Function for Combining Mixed Effects with Random Forest. Modified from mixRF package

```
function(Y, X, random, data, initialRandomEffects=0,
         ErrorTolerance=0.001, MaxIterations=1000) {

  #Y = Target Variable, X = Predictors, random = random effects (Valleys)

  Target = Y

  # Condition that indicates the loop has not converged or run out of
iterations
  ContinueCondition = TRUE

  iterations <- 0

  # Get initial values
  AdjustedTarget <- Target - initialRandomEffects
  oldLogLik <- -Inf

  while(ContinueCondition){

    iterations <- iterations+1


    #randomForest
    rf = randomForest(X, AdjustedTarget, ntree = 500)

    # y - X*beta (out-of-bag prediction)
    resi = Target - rf$predicted

    ## Estimate New Random Effects and Errors using lmer
    f0 = as.formula(paste0('resi ~ -1 + ',random))
    lmefit <- lmer(f0, data=data)

    # check convergence
    newLogLik <- as.numeric(logLik(lmefit))

    ContinueCondition <- (abs(newLogLik-oldLogLik)>ErrorTolerance &
iterations < MaxIterations)
    oldLogLik <- newLogLik

    # Extract random effects to make the new adjusted target
    AllEffects <- predict(lmefit)

    #  y-Zb
    AdjustedTarget <- Target - AllEffects
  }

  result <- list(forest=rf, MixedModel=lmefit, RandomEffects=ranef(lmefit),
                 IterationsUsed=iterations)
  return(result)
}
```

Random Forest and XGboost:

```
Wells <- data.frame(unique(facies_sw_dn$.id)) #Store all Well UWIs
Wells[,1] <- as.character(Wells[,1])          #convert to characters


for (i in 1:nrow(Wells)) {

  Test_well <- Wells[i,1] #select well to be used as test set with i

  Train_mrf <- subset(facies_sw_XGB_narm, !facies_sw_XGB_narm$.id %in%
Test_well) #Create Train set without Test_Well

  #Only select features that we want to use
  Train_mrf2 <- Train_mrf[,c("Core_Sw_old" ,"GR.x" , "iGR", "ResD.x", "ResM"
,"RHOB.x" , "NPSS.x" , "ResS.x" , "VshPhi2" , "vshSP2" ,  "N_DEPT",
"Valley")]

  Train_mrf <- na.omit(Train_mrf) #take out any NA values

  Test_mrf <- subset(facies_sw_XGB_narm, facies_sw_XGB_narm$.id %in%
Test_well) #Create Test Set


  Test_mrf2 <- Test_mrf[,c("Core_Sw_old","GR.x" , "iGR", "ResD.x", "ResM"
,"RHOB.x" , "NPSS.x" , "ResS.x" , "VshPhi2" , "vshSP2" , "pred_final2",
"N_DEPT", "Valley" )]

  facies_sw_XGB_PR2 <- facies_sw_XGB_narm[,c("Core_Sw_old","GR.x" , "iGR",
"ResD.x", "ResM" ,"RHOB.x" , "NPSS.x" , "ResS.x" , "VshPhi2" , "vshSP2" ,
"pred_final2", "N_DEPT", "Valley" )]


  #Random Forest with Mixed Effects model:

  mrf_VFac2 = MixXGB(Y = Train_mrf2$Core_Sw_old, X = as.data.frame(Train_mrf2
%>%
select(-Core_Sw_old)), random = "(1|Valley)", data = Train_mrf2,
initialRandomEffects = 0, ErrorTolerance = 0.01, MaxIterations = 5)

  pred_mixrf2 <- predict(mrf_VFac2$forest, Test_mrf2) #Make Prediction on the
test set

  pred_full_mrf <- predict(mrf_VFac2$forest, facies_sw_XGB_PR2) #Predict
Water Saturation (From Random Forest-Mixed Effects) for all the wells - to be
used by XGBoost for Bit Wt.



  #store R2 values for test set, Col. 2: Modified Simandoux vs Dean Stark,
Col. 3: Machine Learning vs Dean Stark
  mrf_Results_BW[i,1] <- Test_well #store test UWI
  mrf_Results_BW[i,2] <-  cor(Test_mrf$Core_Sw_old, Test_mrf$SwMS)
  mrf_Results_BW[i,3] <-  cor(Test_mrf$Core_Sw_old, Test_mrf$pred_mixrf)


  facies_bw_XGB_narm <- cbind(facies_sw_XGB_narm, pred_full_mrf) #bind WS
predictions with datatable

  facies_bw_XGB <- subset(facies_bw_XGB_narm,
!facies_bw_XGB_narm$Core_UD1_old %in% -999.250) #Remove depths that done have
core samples
  Test_well <- Wells[i,1] #select well to be used as test set with i

  Train_mrf <- subset(facies_bw_XGB, !facies_bw_XGB$.id %in% Test_well)
#Create Train set without Test_Well
  Train_mrf <- Train_mrf[,c("Core_UD1_old" ,"GR.x" , "iGR", "ResD.x", "ResM"
,"RHOB.x" , "NPSS.x" , "ResS.x" , "VshPhi2" , "vshSP2" , "pred_final2",
"N_DEPT", "Valley", "pred_full_mrf" )]

  #Only select features that we want to use
  Train_mrf2 <- Train_mrf[,c("Core_UD1_old" ,"GR.x" , "iGR", "ResD.x", "ResM"
,"RHOB.x" , "NPSS.x" , "ResS.x" , "VshPhi2" , "vshSP2" ,  "N_DEPT", "Valley",
"pred_full_mrf" )]

  Train_mrf <- na.omit(Train_mrf) #take out any NA values

  Test_mrf <- subset(facies_bw_XGB, facies_bw_XGB$.id %in% Test_well) #Create
Test Set

  Test_mrf <- Test_mrf[,c("Core_UD1_old", "GR.x" , "iGR", "ResD.x", "ResM"
,"RHOB.x" , "NPSS.x" , "ResS.x" , "VshPhi2" , "vshSP2" , "N_DEPT",
"pred_full_mrf"  )]



  #Run XGBOOST

  Train_mrf2$Valley <- as.numeric(Train_mrf2$Valley)

  xgb_bw = xgboost(data = as.matrix(Train_mrf2 %>%
                                        select(-Core_UD1_old)),
                   label = Train_mrf2$Core_UD1_old,
                   params = xgb_params_1,
                   nrounds = 1000,
# max number of trees to build
                   verbose = TRUE,
                   print.every.n = 1,
                   early.stop.round = 10
# stop if no improvement within 10 trees
  )




  xgbpred_bw <- predict(xgb_bw, as.matrix(Test_mrf %>%
                                              select(-Core_UD1_old))) #Make
```

```
Predictions on test set

  Test_mrf$Bw_logs <-(Test_mrf$PHIE * (1 - Test_mrf$SwMS))/(((1-
Test_mrf$PHIE)*2.65) + (Test_mrf$PHIE)) #bitumen weight from Modified
Simandoux Predictions

  Test_mrf <- Test_mrf[,c("Core_UD1_old", "GR.x" , "iGR", "ResD.x", "ResM"
,"RHOB.x" , "NPSS.x" , "ResS.x" , "VshPhi2" , "vshSP2" , "N_DEPT", "Valley",
"Bw_logs" )]
  Test_mrf <- cbind(Test_mrf,  xgbpred_bw) #bind XGboost predictions with
Testset

  mrf_Results_BW[i,4] <-  cor(Test_mrf$Core_UD1_old, Test_mrf$Bw_logs)
#Record R-squared values for Models vs Dean Stark Bit. Wt., Col. 4: Mod.
Sim., Col. 5: Machine Learning
  mrf_Results_BW[i,5] <-  cor(Test_mrf$Core_UD1_old, Test_mrf$xgbpred_bw) #
  mrf_Results_BW[i,6] <- sqrt(mean((Test_mrf$Core_UD1_old -
Test_mrf$Bw_logs)^2)) #Record RMSE values for Models vs Dean Stark Bit. Wt.,
Col. 6: Mod. Sim., Col. 7: Machine Learning
  mrf_Results_BW[i,7] <- sqrt(mean((Test_mrf$Core_UD1_old -
Test_mrf$xgbpred_bw))


}
```

Feature Abbreviations:

Core_UD1_old = Core bitumen weight, GR.x = Gamma Ray Log, iGR = Vshale (Gamma), ResD.x = Deep Resistivity, ResM = Medium Resistivity, RHOB.x = Bulk Density, NPSS.x = Neutron Porosity, ResS.x = Shallow Resistivity, VshPhi2 = Vshale (Porosity Logs), N_DEPT = Depth (Offset), Valley = Valley Fills, Bw_logs = Bitumen Weight from Mod. Sim

Libraries/Packages Used:

MixRF, version 1.0
randomForest, version 4.6-12
xgboost, version 0.6-2
plyr, version 1.8.4
dplyr, version 0.5.0
ggplot2, version 2.1.0

R Core Team (2016). R: A language and environment for statistical computing. R Foundation for Statistical Computing, Vienna, Austria. URL https://www.R-project.org/