University of
Stavanger

FACULTY OF SCIENCE AND TECHNOLOGY

MASTER'S THESIS

Study program/ Specialization: Petroleum Geoscience Engineering	Spring semester, 2021 Open
Writer: Mohammed Omar Abdelaziz Ali	Signature: Mohammed Omar Ali

Faculty supervisor: Nestor Cardozo, University of Stavanger

External supervisor: Wim Lekens, GeoProvider A/S

Thesis title:

Machine learning based shale volume prediction from the Norwegian North Sea.

Credits (ECTS): 30

Keywords: *Shale volume* *Machine Learning* *Norwegian North Sea* *Well logs* *Core-image grayscale* *Supervised ML*	Pages: 81 Stavanger, 15th June, 2021

Machine learning based shale volume prediction from the Norwegian North Sea

By

Mohammed Omar Abdelaziz Ali

MSc Thesis

Presented to the Faculty of Science and Technology
University of Stavanger

University of Stavanger

June 2021

Acknowledgements

First, I would like to express my gratitude to my family for their encouragement, motivation, gentle patience, and magnificent love that guides me joyfully every day.

I would like to thank my supervisors, Professor Nestor Cardozo and Dr. Wim Lekens, for their amazing mentorship, from insightful feedback and comments, knowledge sharing, engagement and guidance throughout this thesis. I would also want to thank FORCE and Geoprovider A/S for providing the data that made this thesis possible.

Abstract

Petroleum geosciences, like other fields, has entered the era of new advanced technologies to handle problems related to complex massive data sets and decision making. The growing quantity of subsurface datasets has created numerous challenges in terms of data analysis reliability, speed, accuracy, and consistency. As a result, data science and machine learning (ML) are quickly gaining traction in the oil industry as a strategy for adding value.

This thesis investigates the application of ML in the field of petrophysics, specifically in the estimation of shale volume (Vsh). The percentage or fraction of shale in the rock, or Vsh, is an important input parameter for some geological applications such as reservoir calculations (e.g., net to gross and average reservoir porosity), fault seal analysis (e.g., shale gouge ratio), and geomechanics.

The motivation for this study is to obtain a fast, reliable, and consistent Vsh model using ML and well data from the Norwegian northern North Sea. For this purpose, several objectives were defined. As the first most important step, a dataset was constructed by generating the Vsh label to train the ML models and selecting the features used by the models. The Vsh label was generated after comparing and validating the different petrophysical methods with core data. Secondly, a supervised regression ML technique was used to build the predictive models.

Classical petrophysical methods for Vsh estimation can be used with a relatively low margin of error, with the nonlinear GR method being the most optimal. In addition, core image grayscale pixel values are effective as a Vsh estimation. The Vsh from the non-linear GR method was used as label for the ML methods. The ML results indicate outstanding performance of the XGBoost algorithm with a validation set root mean square error (RMSE) of 0.078 for the best model. Tuning of the model hyperparameters, adding meaningful features such as lithology index, and making the ML models aware of the wells' locations, increases the model accuracy (reduces RMSE) as much as 15%. An interesting ML model was built based only on location and depth features, with a validation set RMSE of 0.13. This model could be improved and used to determine Vsh at any location and depth before drilling a well.

Table of Contents

List of Figures

List of Tables

Glossary

Baseline model: A baseline is a simple model that produces reasonable results on a task and requires little knowledge or time to construct and serve as a comparison point for how well another model performs.

Categorical data: Features with a finite number of possible values. For example, in the current study it is groups and formations features.

Classification model: A discrete-valued model.

Correlation matrix: A correlation matrix is a table that displays the correlation coefficients between variables.

Clustering: The process of grouping data points into groups or clusters so that data points in the same cluster are more similar to each other than those in the other clusters.

Continuous feature: A feature that can take on any number of different values.

Cross-validation: Testing a model against one or more non-overlapping data subsets excluded from the training set to estimate how well it will generalize to new data.

Data cleaning: The process of increasing the quality of data by changing its form or substance, such as eliminating or correcting inaccurate data values.

Decision tree: A model in the form of a series of branching statements.

Discrete feature: A feature that is discontinuous and have a definite set of possible values.

Feature: An input variable that is utilized to make predictions. The feature set is the set of features on which the machine learning model trains.

Feature engineering / Feature extraction: The process of deciding which features would be beneficial in model training and then converting raw data from log files and other sources into those features.

Hyperparameter: It is a parameter whose value controls the learning process.

Label: The desired prediction (synonym is target).

Learning rate: A scalar that is used to train a model by gradient descent. The gradient descent technique multiplies the learning rate by the gradient at each iteration. The gradient step is the name given to the final product. The learning rate is a key hyperparameter.

Machine learning: A system that uses input data to create a predictive model. The trained model is then used to make predictions from new data that has the same distribution as the data used to train it.

Model: The output of a machine learning system after it has learned from the training data.

Overfitting: Building a model that is so similar to the training data that it fails to generate accurate predictions on new data.

Parameter: A model variable that is self-taught by the machine learning system.

Prediction: An algorithm's output after it has been trained on historical data and applied to new data.

Prediction bias: The distance between the average of predictions and the average of label/s. The concept of bias is that the model prioritizes some features in order to generalize across a broader dataset with a variety of other features.

Regression model: A continuous-valued model.

Root Mean Squared Error (RMSE): The square root of the Mean Squared Error (MSE). The **MSE** of a model in relation to a test set is the average of all squared prediction errors in the test set.

Scaling: Feature scaling is a technique for normalizing the range of features by converting an actual range of numbers into a standard range of values, such as 0 to 1.

Scikit-learn: An open-source machine learning platform that is widely used (See www.scikit-learn.org).

Supervised machine learning: The process of training a model using input data and labels.

Test set: An independent data set that is used to test the model after it has been trained and validated on an initial training data set. It is new data but follows the same probability distribution as the training dataset.

Training: Is the process of providing a training dataset for a machine learning (ML) system to learn from.

Training set: The subset of the dataset used to train a model.

Underfitting: Poor predictive model because the model does not capture the complexity of the training data.

Validation: A method for evaluating the quality of a machine learning model using the validation set as part of the training process.

Validation set: A subset of the dataset that is disjoint from the training set.

List of Abbreviations

Abbreviation	Description
CV	Cross-Validation
Fm	Formation
GR	Gamma ray log
IP	Interactive Petrophysics software
KNN	K-Nearest Neighbor algorithm
ML	Machine Learning
m	Meter
MD	Measured depth
ms	Millisecond
N/G	Net pay to gross sand
NPHI	Neutron Porosity log
PE	Photo-Electric Factor
QC	Quality Check
RF	Random Forest algorithm
RMSE	Root Mean Square Error
RDEP	Deep resistivity log
RMED	Medium resistivity log
RHOB	Bulk Density log
SGR	Shale gouge ratio
SVM	Support Vector Machine algorithm
Vsh	Shale volume
VSH_GR	ML label
Vsh (GR)	Average of nonlinear GR shale volumes
Vsh (N)	Shale volume from neutron log
Vsh (SP)	Shale volume from spontaneous potential log
Vsh (N/D)	Shale volume from neutron and density cross plot
Vsh (N/S)	Shale volume from neutron and sonic cross plot
Vsh (S/D)	Shale volume from sonic and density cross plot
XGBoost	Extreme Gradient Boosting algorithm
XRD	X-Ray Diffraction

1. Introduction

In a world of ever-growing subsurface datasets and the need for faster and better decisions, there is a need in the oil and gas industry for high-quality fast data analysis that can replace the current manual ways of working. As in many other fields, petroleum geoscience has entered the era of data science and machine learning (ML) to deal with the size and complexity of datasets. The ML methodology is a collection of algorithms that enable data to be transformed into reasonable predictions.

The motivation behind this study is the application of ML for the estimation of shale volume (Vsh) from well data. Vsh is a relatively simple petrophysical parameter that is very important for several geological applications. For example, Vsh estimation is an essential early-stage process for net reservoir calculations and fault sealing analysis, as it lays the foundation for further calculations such as porosity, saturation, and shale gouge ratio (SGR). These calculations form an important part of assessing the petroleum potential and hydrocarbon reserves in an area. Vsh can be challenging to determine, for example in thin beds and laminated sands. According to Syed et al. (2020), ML has been increasingly used in several studies related to shales over the last decade, with a particular emphasis on shale gas reserves (2157 papers) and geomechanics (971 papers). The large number of Vsh-related studies emphasizes the need for having a fast and reliable Vsh model.

Calculating Vsh using classical petrophysical methods brings uncertainties related to multiple calculations, as different logs have specific measurement issues and errors which are highlighted in this study. This introduces a high level of interpretation and potential bias because the petrophysicist needs to determine which data to use and which method to select. Recently, more research has been conducted to reduce Vsh uncertainties (e.g., Soto et al., 2010; Mabrouk and Kamel, 2011; Naeini et al., 2019). To test the potential of ML on Vsh predictions, an area in the Norwegian northern North Sea with well-known geology and available well data was chosen.

1.1 Goals of the Study

There are two main goals behind this study:

- Building a fast, reliable, and consistent Vsh ML model using multi well logs.
- Obtaining a better understanding of how ML performs predictions on Vsh based on well logs.

1.2 Objectives

The main steps of this study are:

- Perform classical petrophysical calculations of Vsh using well logs.
- Attempt to derive a “true” Vsh from core data.
- Comparison between the petrophysical Vsh results and the Vsh model from core data. Given the limitations of each method, the best method will be chosen to create the label for the ML phase.
- Implement, train and test ML predictive models for Vsh estimation.
- Apply new ideas for Vsh estimation that can add value to the energy industry.

2. Regional setting

2.1 Location

The North Sea is one of the world's major hydrocarbon provinces. Geographically, the current study area is situated in the Norwegian Sector of the northern North Sea and covers blocks in the quadrants 15, 16, 17, 25, 26, 29, 30, 31, 32, 33, 34 and 35 (Figure 1), spanning the southern and northern Viking graben. The study area includes large fields such as Troll, Oseberg Øst, Brage, Statfjord, Gullfaks Sør, Valemon and Kvitebjørn (Figure 1).

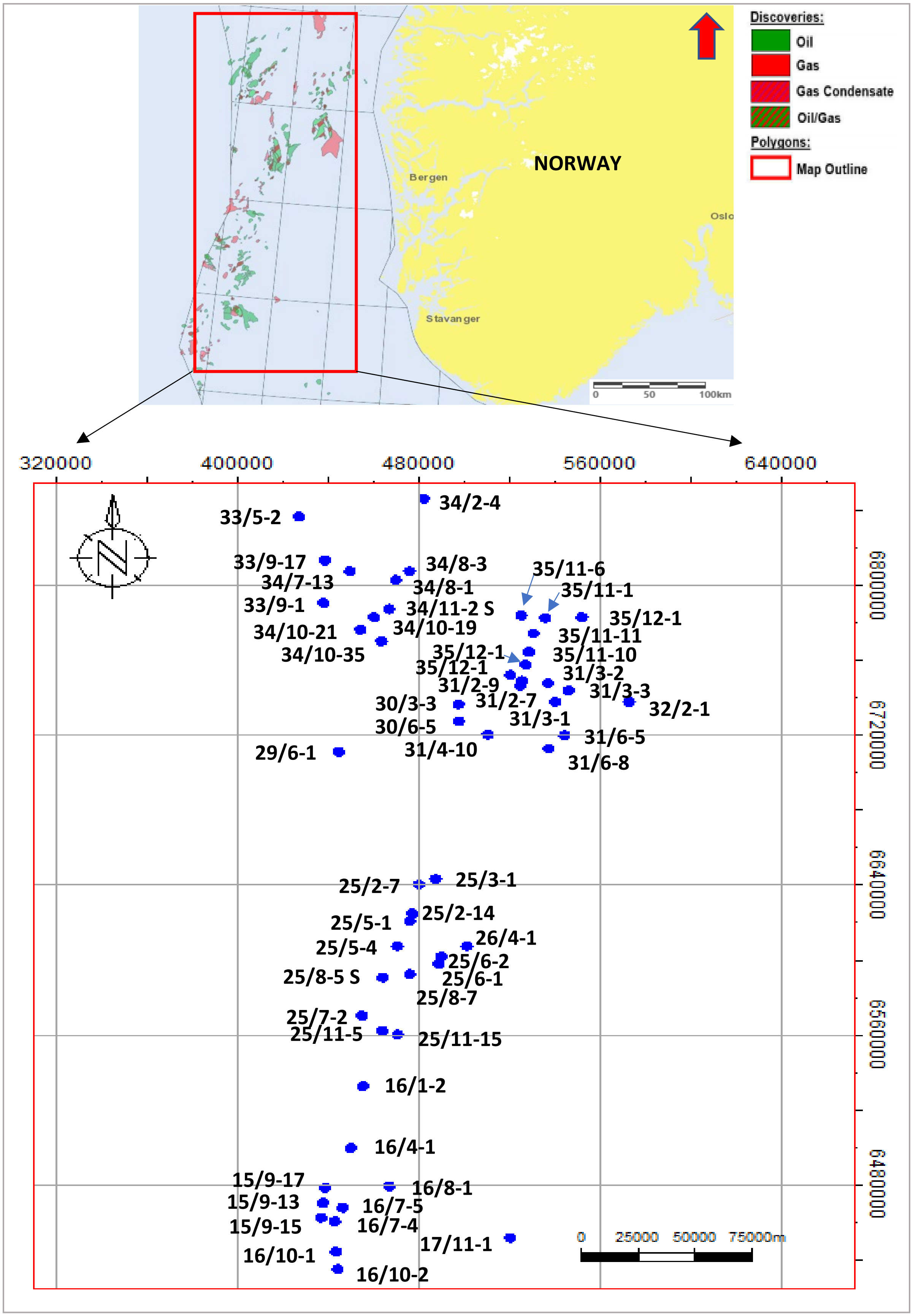


Figure 1: Outline of study area within the Norwegian northern North Sea. The map below shows the wells' locations.

2.2 Structural setting

The main structures of the North Sea are primarily related to the Upper Jurassic-Lower Cretaceous rifting, with older structural elements contributing less (Halland et al., 2014). The northern North Sea basin is bounded by the Øygarden Fault Complex (Permo-Triassic) to the east, and normal faults (mainly Jurassic) to the west, separating the area from the East Shetland platform (Færseth, 1996) (Figure 2). Since the end of the Caledonian orogeny, the northern North Sea has been subjected to several post-Caledonian phases of crustal extension. Two main phases of rifting are recognized: the Permo–Early Triassic and the mid Jurassic–Early Cretaceous (Badley et al., 1988; Yielding et al., 1992; Færseth, 1996; Odinsen et al., 2000; Fossen, 2016). The northern North Sea basin consists of a post-Caledonian graben system dominated by normal faults trending north and northeast (Færseth, 1996).

The Viking graben and the Sogn graben dominate the northern North Sea structural evolution (Yielding et al., 1992; Bjørlykke, 2015) (Figure 2). The Viking Graben structural style is characterized by tilted fault blocks that are bounded by large normal faults (Figure 3).

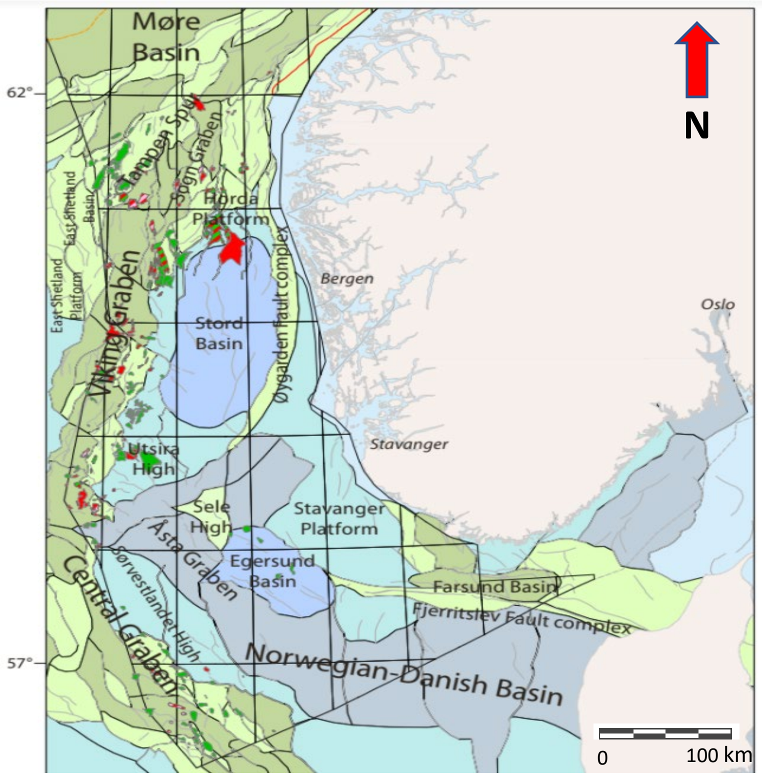


Figure 2: Main structural elements of the Norwegian northern North Sea (modified after Halland et al., 2014)

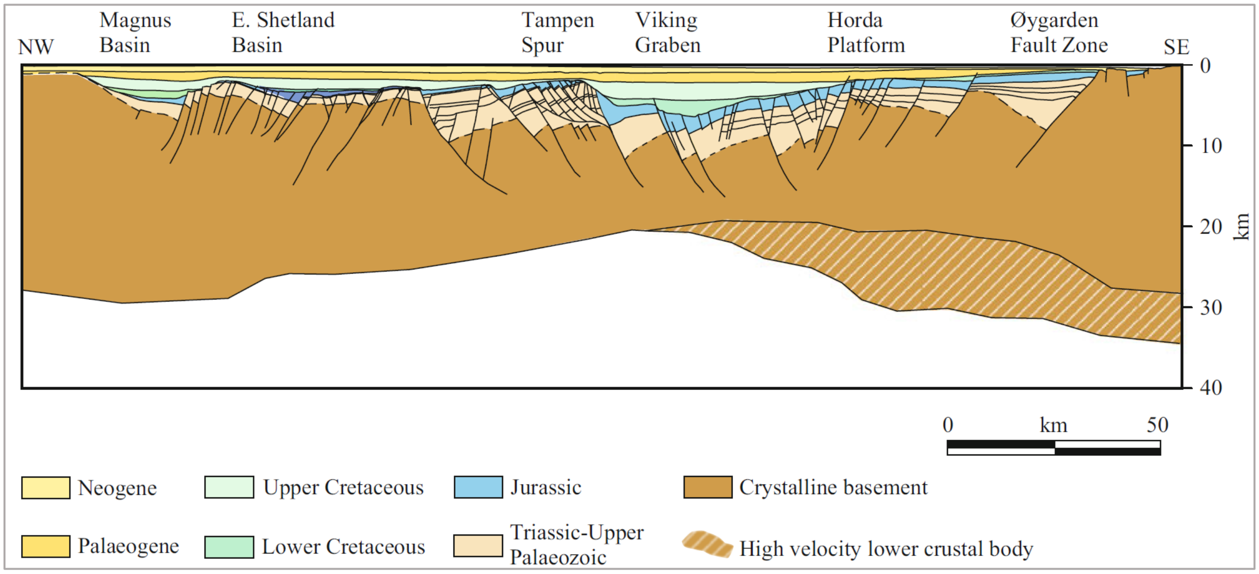


Figure 3: Regional section across the northern North Sea. Modified from Bjørlykke (2015) after Christiansson et al. (2000).

2.3 Stratigraphy

The stratigraphy of the northern North Sea is well described in numerous studies (e.g., Bowen 1975; Helland-Hansen et al. 1992; Yielding et al. 1992; Ryseth, 2000). This study focuses on the Brent Group reservoir and the estimation of Vsh in this unit. Since the discovery of the Brent field in 1971 by Shell, the Brent Group has been the most important hydrocarbon reservoir in the northern North Sea (Morton et al., 1992; Brennand et al. 1990).

The Brent Group consists of Middle Jurassic deltaic sediments. These sediments were deposited during the late phase of post-rift subsidence following the Late Permian-Early Triassic rifting, because of regressive-transgressive cycles of sedimentation (Helland-Hansen et al. 1992; Ryseth, 2000). The deposition of the group was controlled by Middle Jurassic thermal subsidence and Late Jurassic-Early Cretaceous rifting (Yielding et al. 1992). The Brent Group is located at a various range of depths because of faulting, uplift, erosion, and differential subsidence. Also, parts of the group may be missed over the crests of rotated fault blocks (Halland et al., 2014). The upper boundary of the Brent Group is defined by marine shales of the Heather Fm, while the lower boundary consists of marine shales of the Dunlin Group.

The Brent Group is divided into five formations: the Broom, Rannoch, Etive, Ness and Tarbert formations (Bowen 1975; Vollset & Doré 1984) (Figure 4). Generally, it consists of clastic sediments including sandstones, siltstones, shales, and coals. The upper formations represent a significant regressive (Rannoch, Etive and Ness formations) to transgressive (Ness and Tarbert formations) clastic wedge, whereas the lowest Broom Formation represents the early lateral infill of the basin (Helland-Hansen et al., 1992).

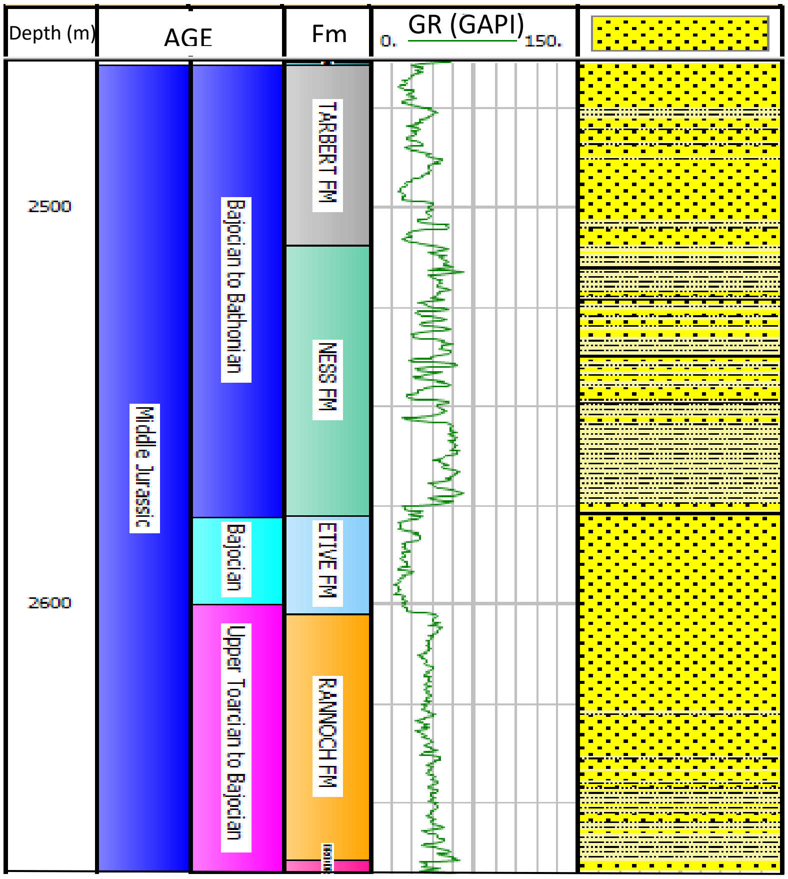


Figure 4: Stratigraphic column of the Brent Group together with Gamma ray log. Created from well 33/9-1 whose location is in Figure 1.

3. Data and Methodology

The dataset used for this thesis consists of well logs and core images. The well logs are provided by the FORCE 2020 machine learning contest (https://xeek.ai/challenges/force-well-logs/overview) and they consist of data from 108 wells from the Norwegian continental shelf, northern North Sea, of which 55 wells were selected for this study (Table 1). Core images were obtained from Diskos and the NPD factpages, and processed by Geoprovider. The processing included cutting, cropping, and indexing individual core pictures. Well logs include gamma ray (GR), caliper, resistivity (shallow, medium and deep), density (ROHB), neutron (NPHI), sonic (shear and compressional), spontaneous potential (SP), photoelectric (PE), in addition to lithology labels and groups / formations intervals. The Python programming language and Interactive Petrophysics (IP) were used for analyzing, processing, interpreting, and presenting the data.

Table 1: Well data used in this study. A means available and N means not available.

Index	Well	X-coordinate	Y-coordinate	Core images	Well logs and Lithology index
1	15/9-13	437653.70	6470978.02	N	A
2	15/9-15	436823.95	6462994.43	N	A
3	15/9-17	438602.65	6478951.28	N	A
4	16/1-2	455231.21	6533324.63	N	A
5	16/4-1	449959.99	6500262.06	N	A
6	16/7-4	442828.36	6461118.04	N	A
7	16/7-5	446284.86	6468386.64	N	A
8	16/8-1	466879.22	6479866.89	N	A
9	16/10-1	444163.52	6435546.75	N	A
10	16/10-2	443327.64	6444927.85	N	A
11	17/11-1	520158.20	6452288.88	N	A
12	25/2-7	479891.01	6641115.53	N	A
13	25/2-14	476953.03	6625451.08	N	A
14	25/3-1	487288.97	6643946.88	N	A
15	25/5-1	475841.98	6621549.92	N	A
16	25/5-4	470403.24	6608163.57	N	A
17	25/6-1	488717.11	6598746.73	N	A
18	25/6-2	489849.43	6602696.66	N	A
19	25/7-2	454816.14	6571051.25	N	A

20	25/8-5 S	463972.36	6591324.83	N	A
21	25/8-7	475896.00	6593127.02	N	A
22	25/11-5	463755.93	6562934.90	N	A
23	25/11-15	470534.45	6560840.35	N	A
24	26/4-1	501148.45	6608189.83	N	A
25	29/6-1	444581.73	6711948.14	N	A
26	30/3-3	497360.47	6737312.21	A	A
27	30/6-5	497437.43	6728313.49	A	A
28	31/2-7	524587.90	6747126.56	A	A
29	31/2-8	526925.96	6758502.04	N	A
30	31/2-9	525386.67	6749867.74	N	A
31	31/2-19 S	520149.18	6753026.19	A	A
32	31/3-1	539989.88	6738661.73	A	A
33	31/3-2	536836.48	6748639.40	A	A
34	31/3-3	545884.32	6744748.95	A	A
35	31/4-5	503125.28	6713782.15	A	A
36	31/4-10	510277.82	6721185.71	N	A
37	31/6-5	544106.97	6720730.31	A	A
38	31/6-8	537156.05	6713676.33	A	A
39	32/2-1	572630.04	6738602.99	N	A
40	33/5-2	427002.08	6837585.98	A	A
41	33/9-1	437780.23	6791573.15	N	A
42	33/9-17	438481.99	6814193.73	A	A
43	34/2-4	482072.43	6847163.49	N	A
44	34/7-13	449458.44	6808466.37	N	A
45	34/8-1	469666.18	6803722.25	N	A
46	34/8-3	475751.35	6808456.80	N	A
47	34/10-19	460128.36	6783931.44	N	A
48	34/10-21	454125.77	6777264.06	A	A
49	34/10-35	463374.05	6771063.85	A	A
50	34/11-2 S	466835.28	6788267.62	N	A
51	35/11-1	535626.01	6783527.21	N	A
52	35/11-6	525017.66	6784876.16	N	A
53	35/11-10	528323.08	6765243.58	N	A
54	35/11-11	530381.59	6775189.73	N	A
55	35/12-1	551762.89	6783875.97	N	A

3.1 Data handling

The well logs are provided in text files (LAS and ASCII formats), where in each file, each row represents the measurements at a specific depth in the well. Before the analysis, a data quality check was performed on each set of well logs. For almost 60% of the wells, the shear sonic, shallow resistivity and photoelectric logs were missing or not acquired. Therefore, it was decided not to use these logs in the study. No major well log editing was required, as all the well logs are environmentally corrected and merged, also called high quality log data (HQLD). In general, except minor editing, the data are considered suitable to conduct the research. Most of the wells are near-vertical exploration wells with no sidetracks and are plugged and abandoned (P&A).

3.2 Research workflow

To accomplish this study, the workflow was divided into two main parts (Figure 5). The left branch describes the petrophysical analysis of Vsh using classical methods. The right branch involves building and optimizing the ML Vsh predictive models.

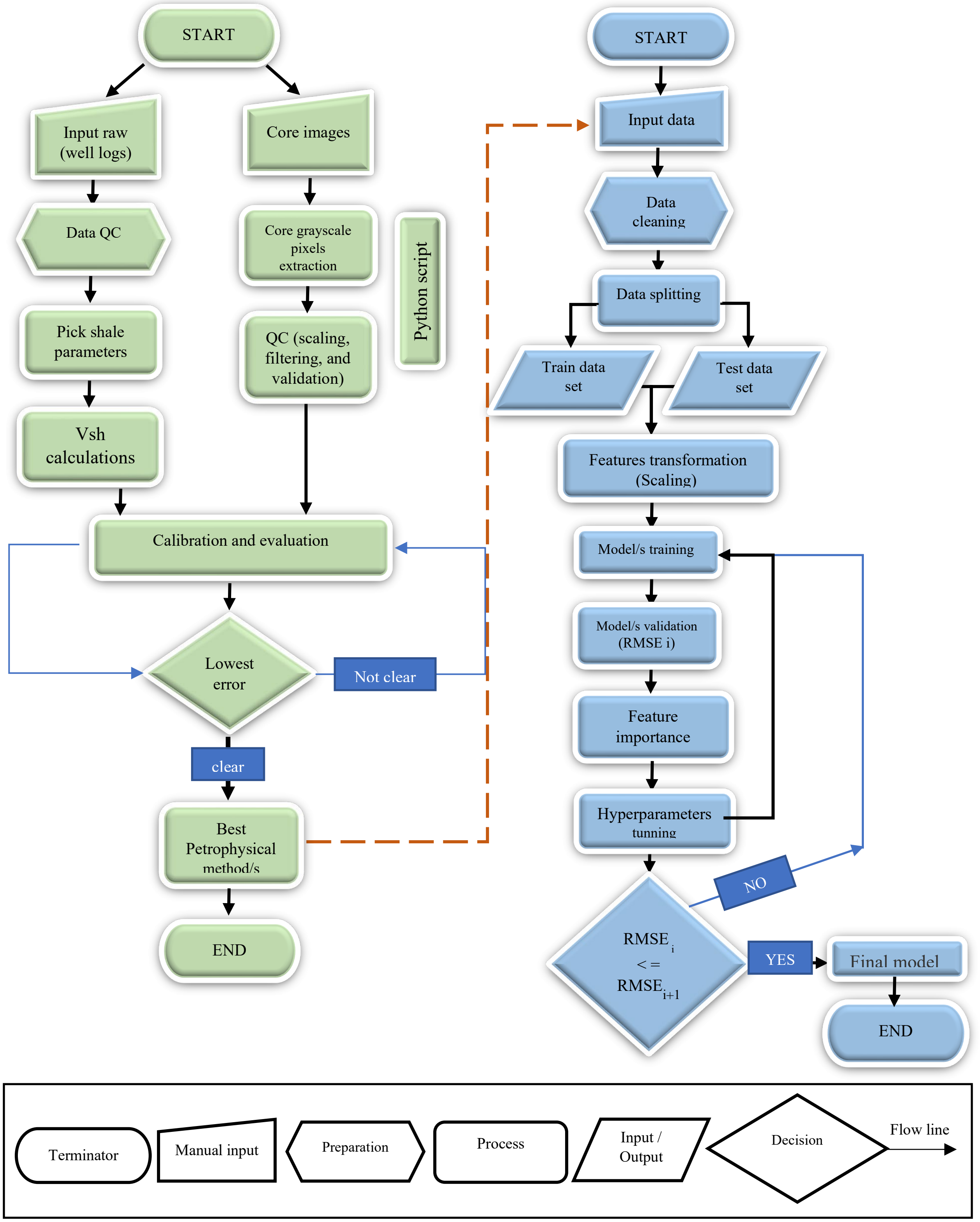


Figure 5: Workflow of the thesis. The left branch (green) consists of the Vsh classical methods. The right branch (blue) consists of the ML Vsh methods. The legend shows the meaning of the different polygons.

3.3 Shale volume (Vsh) or clay volume (Vcl)?

Before conducting the work, it is important to clarify the difference between shale and clay volume. Usually, when it comes to the use of shale volume (Vsh) and clay volume (Vcl) terms, there is a misunderstanding as these terms are considered the same, but they are not. This may lead to incorrect reservoir calculations in shaly sandstone reservoirs.

Shales are fine-grained sedimentary rocks with a high clay content (more than 40%) consisting of clay-sized particles and clay minerals. The clay minerals (e.g. illite, smectite, chlorite, and kaolinite) account for at least 25% of the total rock volume (Picard, 1953; Shaw and Weaver, 1965; Jones et al., 1989). Shales are generally high porous and low permeable rocks. The term Vcl is meant to refer to the clay mineral volume in the shale.

Regarding the petrophysical calculations (well logs analysis), Vsh is what is estimated, not Vcl. From these definitions of Vsh and Vcl, a Vsh of 100% does not imply that the shale is made entirely of clay minerals. When it comes to Vcl estimations, they are not easy to do from well logs as they require lab analyses to study the shale mineral composition, mostly using thin sections and X-Ray Diffraction (XRD).

3.4 Classic shale volume calculations

Various classical Vsh calculation methods are introduced in several studies (e.g., Poupon & Gaymard, 1970; Clavier et al, 1971; Steiber, 1970; Worthington, 1985; Kamel & Mabrouk, 2003). This section aims to find the Vsh calculation method with the least uncertainties so that it can be used to produce the label (desired output) for the ML training step. In general, two types of indicators have been used to measure the Vsh: single indicators like gamma ray (GR), spontaneous potential (SP), neutron, and deep resistivity logs, and double indicators like Neutron/Density, Neutron/Sonic, and Sonic/Density logs. The Vsh can be expressed as a decimal fraction or as percentage.

Various analytical calibrations (e.g., XRD results) can be used to validate the Vsh results from this step. However, these kinds of data were not available. Also, such calibrations have poor coverage, and a margin of error as well. Since core data are more readily available and have better coverage, these are more suitable for calibration and validation of the results from this step by deriving a curve that could reflect the core visible Vsh. For this purpose, the core-images grayscale pixel values were extracted as a trial of the true Vsh. After evaluating the credibility and resolution of the classical Vsh calculation methods and comparing these to the core-images grayscale pixel values and core-visible shale descriptions, the Vsh results with the least uncertainties were selected as the label for the ML step.

3.4.1 Single Indicators

Gamma-Ray Methods

The gamma ray log is the most widely used Vsh indicator. The log responds to changes in natural radiation emitted by the rock formation. As shale content increases, the gamma ray log response increases because of the concentration of radioactive material in the shale. Two gamma ray responses are important for the calculations: maximum gamma ray response, which represents the shale line, and minimum gamma ray response, which represents the clean sand line. There are typically two kinds of equations available to calculate the Vsh from gamma ray logs: linear and nonlinear (Figure 6). The main difference between the linear and nonlinear methods is that the nonlinear response is based on geographic area, formation age, or any other available local information (Asquith and Krygowski, 2004).

The Vsh from GR can be calculated using the following methods:

Linear Method:

$$Vsh = ((\text{GR log} - \text{GR min})/(\text{GR max - GR min}))$$

Where:

GR log = Gamma ray reading

GR min = minimum gamma ray (clean sand or carbonate)

GR max = maximum gamma ray (shale)

This relationship is also known as the gamma ray index (IGR).

Non-linear Method:

For Tertiary rocks, Larionov (1969):

$$Vsh = 0.083\ (2^{(3.7\ IGR)} - 1)$$

Steiber (1970):

$$Vsh = \frac{IGR}{3 - 2\ IGR}$$

Clavier (1971):

$$VSh = 1.7 - \sqrt{3.38 - (IGR - 0.7)^2}$$

For older rocks, Larionov (1969):

$$Vsh = 0.33(2^{(2\ IGR)} - 1)$$

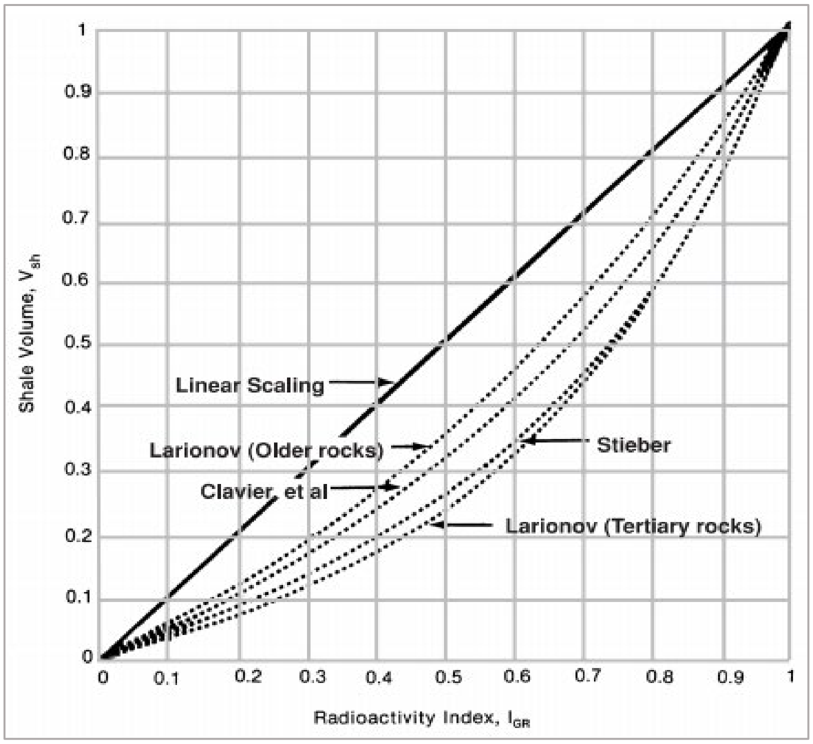


Figure 6: Linear and nonlinear Vsh from GR (Western Atlas, 1995)

Spontaneous Potential (SP) Log Method

The SP log measures the natural potential differences (self-potentials) between an electrode in the borehole and a reference electrode at the surface, with no artificial current applied (Rider, 2002). The SP Vsh formula is a linear interpolation between the minimum and maximum SP readings. The presence of shale in a permeable formation reduces the SP signal. In water bearing zones, the amount of SP reduction is related to the amount of shale in the formation. The SP log can be used to calculate the Vsh in a permeable zone (Asquith and Krygowski, 2004) using the following formula:

$$Vsh(SP) = (\mathrm{SP} - \mathrm{SP0}) \,/\, (\mathrm{SP100} - \mathrm{SP0})$$

where:

Vsh(SP) = Vsh from spontaneous potential

SP = spontaneous potential reading

SP0 = spontaneous potential reading in 100% clean (no shale) formation

SP100 = spontaneous potential reading in shale

Neutron Log Method

The neutron log measures the hydrogen content of the formation. It records higher values in shaly formations and lower values in sandy formations because shales contain more water in the pores than sandstones. Using the neutron log to calculate Vsh is more accurate in the case of high clay content and low effective porosity.

The neutron Vsh calculation formula is:

$$Vsh(N) = \sqrt{\left(\frac{\mathrm{NPHIlog}}{\mathrm{NPHIclay}}\right) * \left(\frac{\mathrm{NPHIlog} - \mathrm{NPHIclean}}{\mathrm{NPHIclay} - \mathrm{NPHIclean}}\right)}$$

Where:

Vsh(N) = Vsh from Neutron log

NPHI log = Neutron log reading of formation

NPHI clean = Neutron log reading in front of a clean zone

NPHI clay = Neutron log reading in front of a shaly zone

3.4.2 Double Indicators

Hashmy and Alberty (1992) discussed lithology identification using double indicators crossplots (e.g., Neutron - density (N/D), Sonic- neutron (S/N), and Density-sonic (D/S). The double indicators are used to estimate Vsh by defining a clean line and a clay point in the cross plot (Interactive petrophysics, 2018). The distance between the shale point and the clean line is calculated to estimate the Vsh. An example of this method is shown in Figure 7 for the case of a Neutron/Density crossplot.

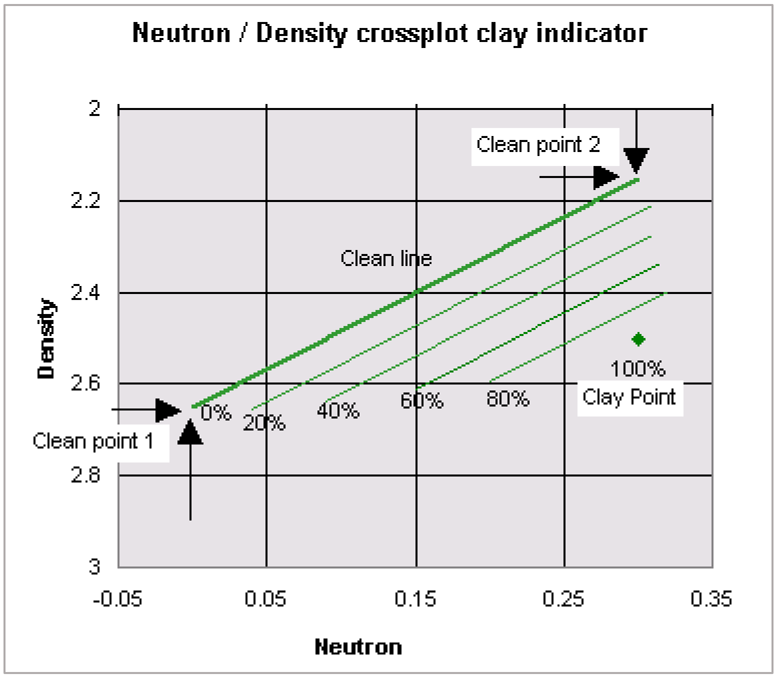


Figure 7: Example illustrating the use of neutron-density crossplot as Vsh indicator (Interactive petrophysics manual, 2018).

The double indicators equations are:

The Neutron-Density double indicator:

$$Vsh = \frac{(Dencl2 - Dencl1) * (Neu - Neucl1) - (Den - Dencl1) * (Neucl2 - Neucl1)}{(Dencl2 - Dencl1) * (Neuclay - Neucl1) - (Denclay - Dencl1) * (Neucl2 - Neucl1)}$$

where:

Neucl1, Dencl1, *Neucl2* and *Dencl2* are the neutron and density values for the two ends of the clean line.

The Neutron-Sonic indicator:

$$Vsh = \frac{(Neucl2 - Neucl1) * (Son - Soncl1) - (Neu - Soncl1) * (Soncl2 - Soncl1)}{(Neucl2 - Neucl1) * (Sonclay - Soncl1) - (Neuclay - Neucl1) * (Soncl2 - Soncl1)}$$

where:

Neucl1, Soncl1, *Neucl2* and *Soncl2* are the neutron and sonic values for the two ends of the clean line.

The Sonic-Density double indicator:

$$Vsh = \frac{(Dencl2 - Dencl1) * (Son - Soncl1) - (Den - Dencl1) * (Soncl2 - Soncl1)}{(Dencl2 - Dencl1) * (Sonclay - Soncl1) - (Denclay - Dencl1) * (Soncl2 - Soncl1)}$$

where:

Soncl1, *Soncl2, Dencl1 and Dencl2* are the sonic and density values for the two ends of the clean line.

3.5 Image analysis for shale volume estimations - A new approach

The main idea behind image processing for shale volume estimations is to extract grayscale pixel values from the core images that can reflect the visible Vsh in a form of curve. Then this curve could be used to validate the Vsh calculations from the well logs. The workflow initiates with the preparation of core images for processing. It involves image segmentation into discrete fragments and image extraction from standard geological report formats.

The image processing CV2 library was used to read and process the core images along with the NumPy and pandas Python libraries. The analyzer code transforms the input color images to grayscale images (Figure 8). The y-axis values are converted to an accurate depth interval of the core image. The best x-axis line with least missing data is chosen to extract the grayscale values of each pixel vertically along the core image. The extracted values are normalized in the range 0 (black) to 1 (white). Afterwards, the model is run, and the values of depth and the grayscale pixel are plotted as a curve using the matplotlib library (Figure 8, blue log). A filter is used to avoid errors related to color anomalies which are associated with fractures and sample plugs (Figure 8, red log). For more accuracy, human supervision and correction is needed.

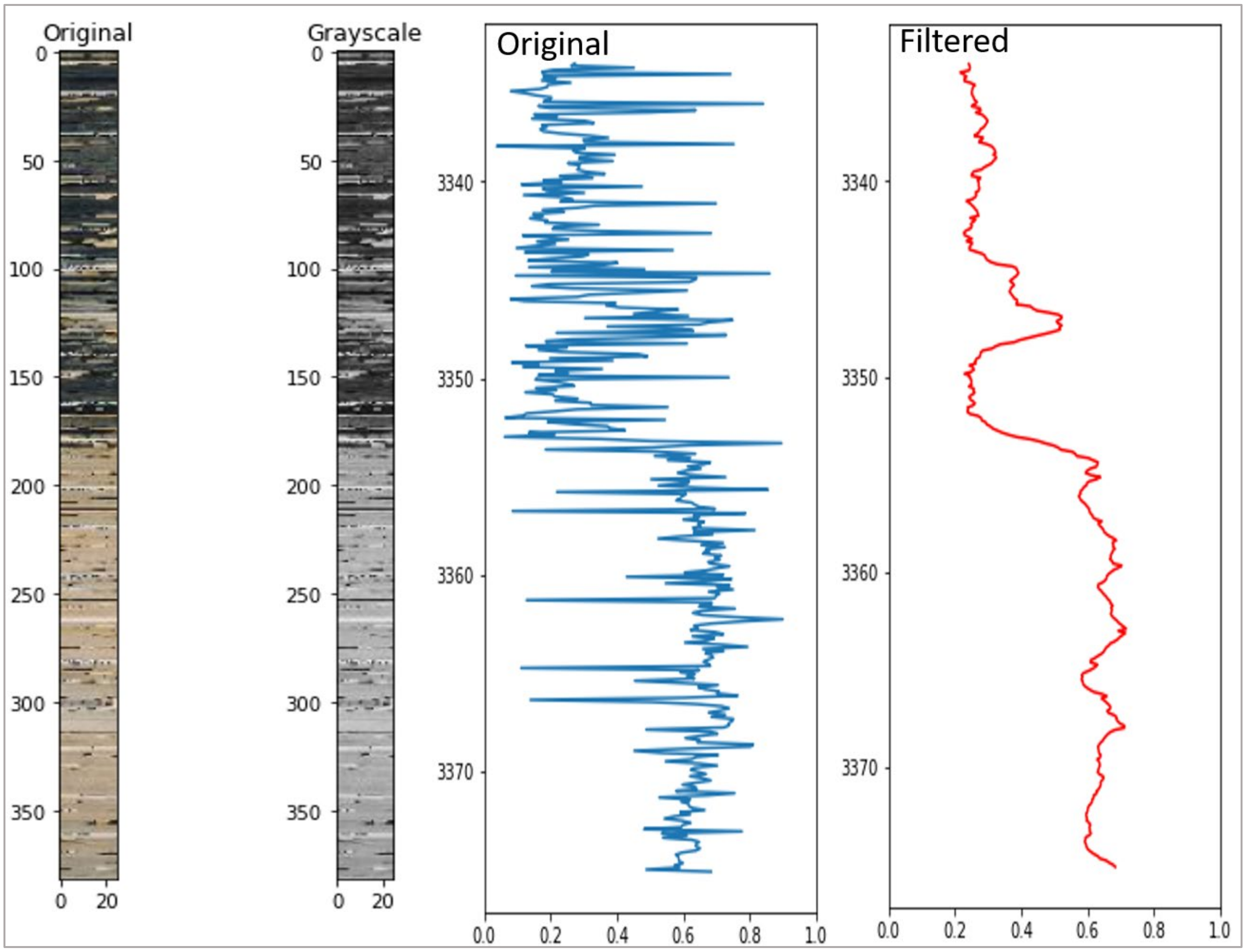


Figure 8: Extraction of core image grayscale pixel values. From left to right: original image, grayscale image, extracted, and filtered greyscale pixel values normalized from 0 to 1. The filtered curve is the Vsh estimate.

3.6 Machine learning approach

Machine learning (ML) is the process of creating computer algorithms that make predictions over time as result of experience and data (Mitchell, 1997). The ability to predict outcomes and make decisions is achieved by training ML algorithms with historical data. In comparison to the traditional methods of section 3.4, once the computer has learned, the estimations are faster and more accurate.

The ML algorithm to be used is determined by the type of problem to be solved, the type of input data, and the desired outputs. Models of supervised and unsupervised ML are the most common. For reasons that will be discussed in the following section, a supervised regression learning approach is used in this study.

3.6.1 Supervised machine learning

In supervised ML, the user provides the algorithm with sets of inputs and needed outputs, and the algorithm figures out a way to predict the needed output given an input (Müller and Guido, 2016) (Figure 9). It is also called a predictive model as it predicts features based on a given historical input data. The aim is to train the model with the given inputs and then test its ability to make accurate predictions using completely new data with the same properties as the training dataset.

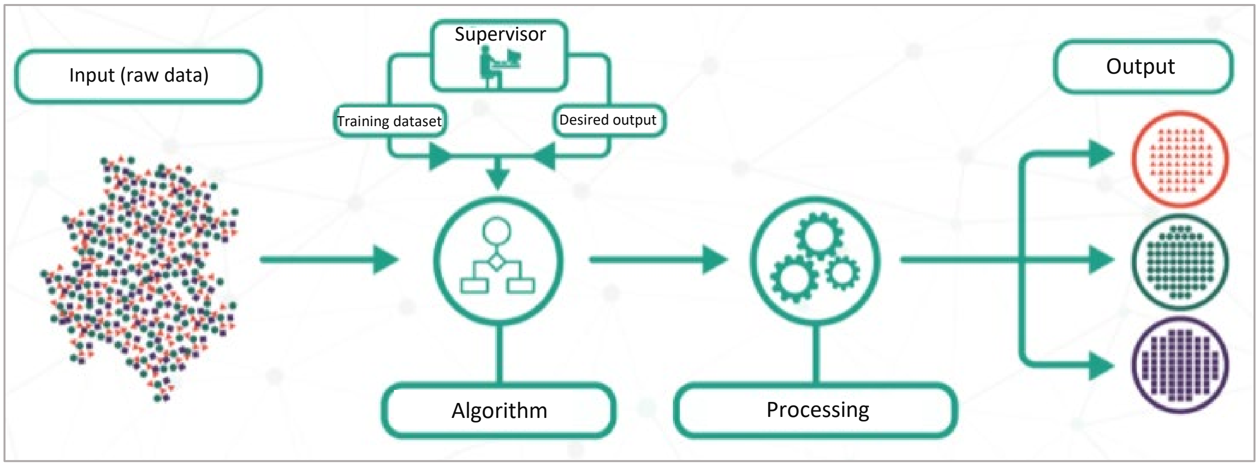


Figure 9: Supervised machine learning workflow (after bigdata-madesimple.com).

Classification and regression algorithms are examples of supervised learning, in which there is an input and an output, and the goal is to learn the mapping from input to output (Ethem, 2010). The key difference between regression and classification algorithms, according to Müller and Guido (2016), is that regression algorithms are used to predict a continuous number, while classification algorithms are used to predict a class label. Regression algorithms are used in this study since the aim is to predict Vsh using various logs, all of which are in the form of continuous values. Random forest (RF), XGBoost and K-nearest neighbor regression algorithms are chosen for reasons that will be discussed in part 4.5.3. The final model is built using the best of these algorithms.

The supervised learning dataset is made up of features and labels. The dataset is a collection of data elements that can be used for analytic and prediction aims. A feature is an input variable that is utilized to make predictions (e.g. well logs data in current study). A label is the desired prediction or target that is interpreted previously (Vsh from nonlinear GR in this study).

3.6.2 Scripting strategy

The Python programming language was used to build the ML models. The Python programs were run in the Jupyter Notebook interactive environment, while the Scikit learn library was used to implement various ML algorithms. These Python libraries can be installed using the operating system command prompt (terminal) window by typing “pip install <package name>”.

Scikit-learn is a very popular, open source, ML Python library. It provides easy access to various regression, classification, and clustering algorithms. It depends only on NumPy and SciPy (Hanke et al., 2009) which makes it flexible to use. Since NumPy can process multidimensional (n-dimensional) arrays and mathematical functions, it is used in scientific computing. In scikit-learn, the NumPy array is the basic structure for the input data and model parameters. The Pandas library is used for data analysis as it has the flexibility to deal with different data structures. The Matplotlib is used to build a variety of graphs and plots (histograms, scatter plots, bar charts, etc).

3.6.3 Machine learning algorithms

The research and development of algorithms that can learn from and predict data is a popular task in ML (Kohavi, 1998). Algorithms, simply put, are the engines of ML, since they construct the model from the ground up using input data sets. As previously mentioned, there are a variety of ML algorithms whose selection is based on the type of data and the problem. Regression tree models (Ensembles or Decision Trees models; including RF and XGBoost) and K-Nearest Neighbor (KNN) are used in this study.

Regression Ensembles (Decision tree) models are a type of nonlinear regression model that can be used to predict continuous values by segmenting a dataset into small groups. Müller and Guido

(2016) define ensemble ML methods as methods for combining multiple ML models to create more efficient models. Furthermore, data scaling is not needed. Regression trees (XGBoost and RF) have recently become one of the most effective and widely used ML algorithms (e.g. Biau, 2012; Chen and Guestrin, 2016).

The K-Nearest Neighbors (KNN) algorithm is a non-parametric approach for predicting new observations based on the K-nearest observations in the training set (Jiang et al, 2012). KNN is a ML algorithm that is fast, simple, efficient, and often produces satisfactory results. The algorithm is commonly used as a benchmark to which other algorithms are measured (Cover and Hart, 1967). The KNN method has been shown to be a reliable method for the field of geosciences (e.g., Wang et al., 2018).

3.6.3.1 Random Forest (RF)

RF is a very stable and efficient machine learning algorithm (Breiman, 2001). During the training stage, RF works by constructing multiple decision trees (Figure 10). Each tree is distinct from the others because of a set of random variables. Individual trees will perform reasonably well in terms of prediction, but they will almost certainly overfit some of the data. Overfitting occurs when a model learns so much information, including the noise in the training data, that it adversely affects the model output. The performance is ranked by combining the individual tree predictions, which reduces overfitting while increasing the predictive power of the model.

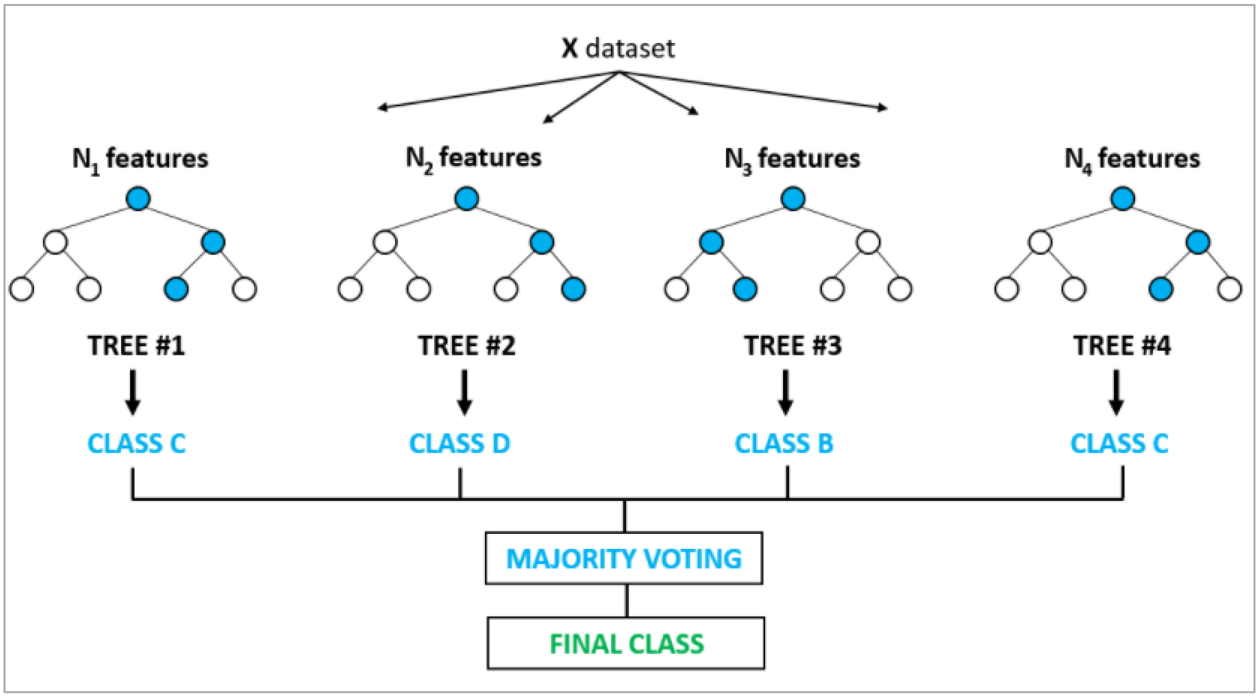


Figure 10: Illustration of an ensemble of decision trees (Kirasich et. al., 2018).

The first step in creating an RF model is deciding on the number of trees to use, which is known as the *n* estimators. A random bootstrap sample of *n* size is given to create the trees, it should be chosen at random with replacement from the training set. This will generate a new dataset from which a decision tree will be constructed. The decision tree has several nodes, and the algorithm selects a subset of the features at random for each node and then searches for the best possible test involving one of these features. When searching for the best split at each node, the max_features (the maximum number of features) parameter is used to consider the number of features. Thus, each node may decide based on different features.

3.6.3.2 Extreme Gradient Boost (XGBoost)

In 2016, Chen and Guestrin developed XGBoost, a highly scalable decision tree ensemble based on gradient boosting. Higher execution speeds, improved model consistency, cache optimization, and out-of-core computation have all been demonstrated for very large datasets using the XGBoost algorithm. Gradient boosting targets are like RF in that they incorporate multiple decision trees to provide a more efficient model. Gradient boosting differs from RF in that trees are built in sequence, with each tree attempting to correct the errors of the previous one. Figure 11 shows a simplified example of the XGBoost working mechanism.

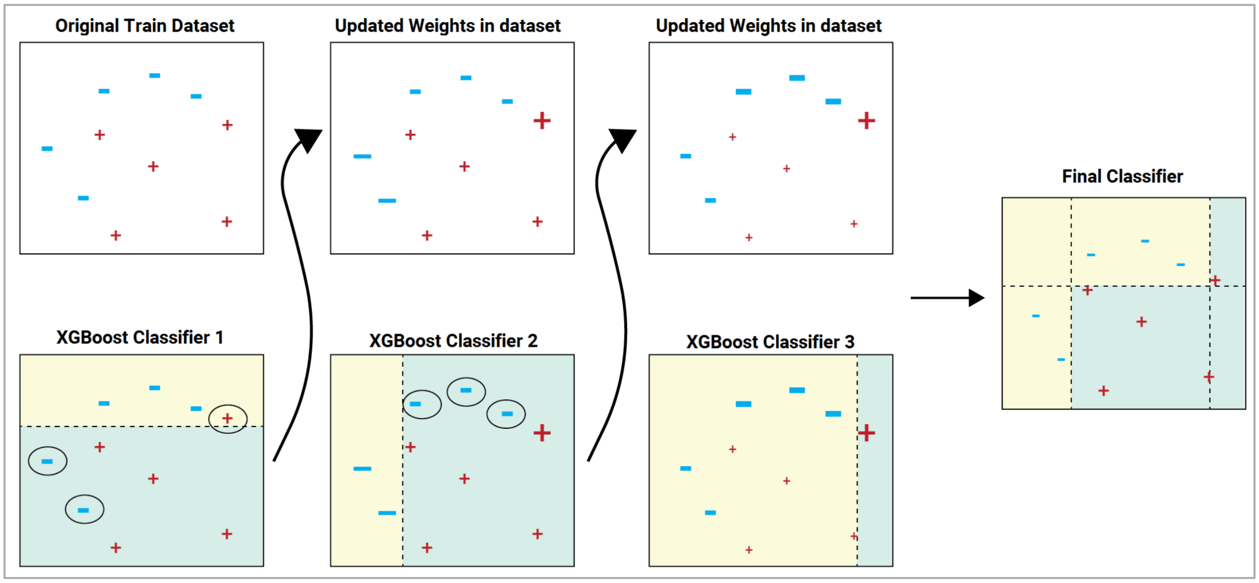


Figure 11: XGBoost working mechanism. In the yellow areas the classifier predicted hyphen, while in the blue areas it predicted plus. The circled points indicate incorrect predictions. The final classifier predicts all the data points correctly (after Quantinsti.com).

3.6.3.3 K-Nearest Neighbor (KNN)

The KNN algorithm is a supervised ML algorithm that can be used to solve classification and regression problems. The average of the K neighbors' responses is typically the expected value of the desired sample. The distance measurement and the number of neighbors are the two most important parameters for the KNN algorithm. KNN works by calculating the distances between a query and all the other input data features. Following that, the algorithm selects the defined number of examples (K) that are closest to the question and votes for the mean of its nearest neighbors' labels (Figure 12). KNN has a substantial disadvantage in that it becomes increasingly slower as the amount of data increases.

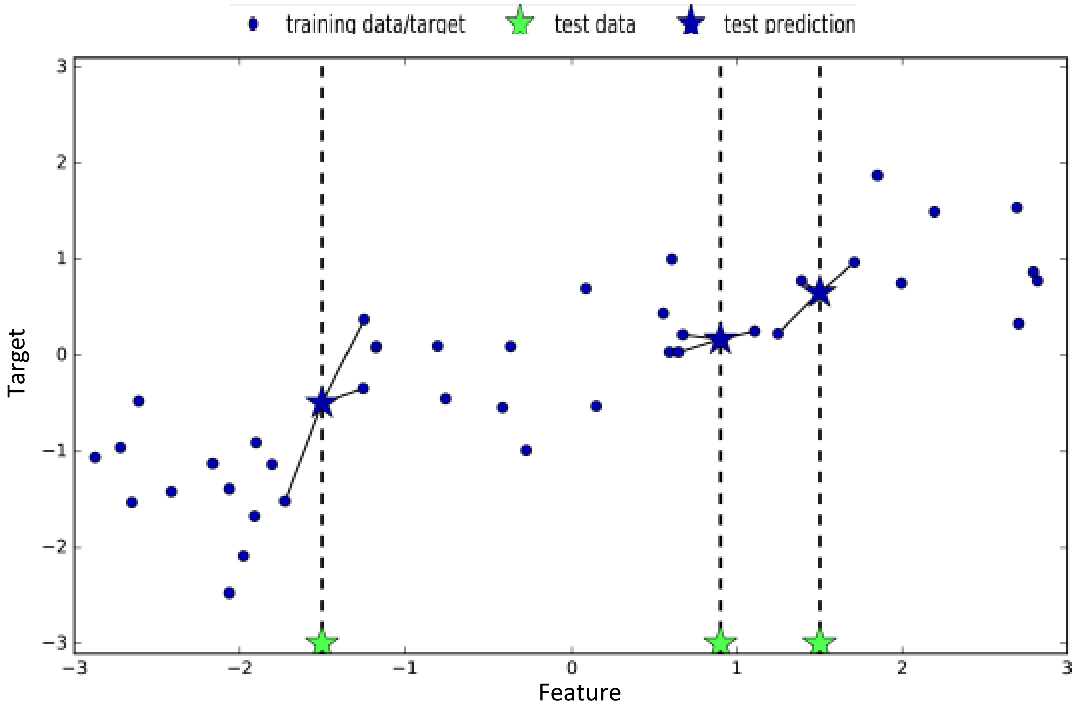


Figure 12: Example of predictions made by three-nearest neighbours (Müller and Guido, 2016).

3.6.4 Dataset construction

Since it is not possible to make predictions without input data, the training dataset is considered as the foundation of any ML algorithm. The dataset is usually in the form of a table, with each column representing a different feature and each row representing a data point or data sample. Good training sets are the key to get accurate predictions and so, it takes much human efforts to be ready before initiating the work with ML algorithms. Having good quality data reduces overfitting and leads to improved predictive models in supervised ML (Ying, 2019).

Basically, the data consists of the available well logs, location, categorical features (groups and formations), lithology index and the Vsh label from the petrophysical methods (Figure 13). Another idea was to use the greyscale method from core data to create a training dataset, but it is a major task beyond the scope of this thesis to analyze all available core images from the selected wells. Instead, the core data were used to validate the other training dataset where possible.

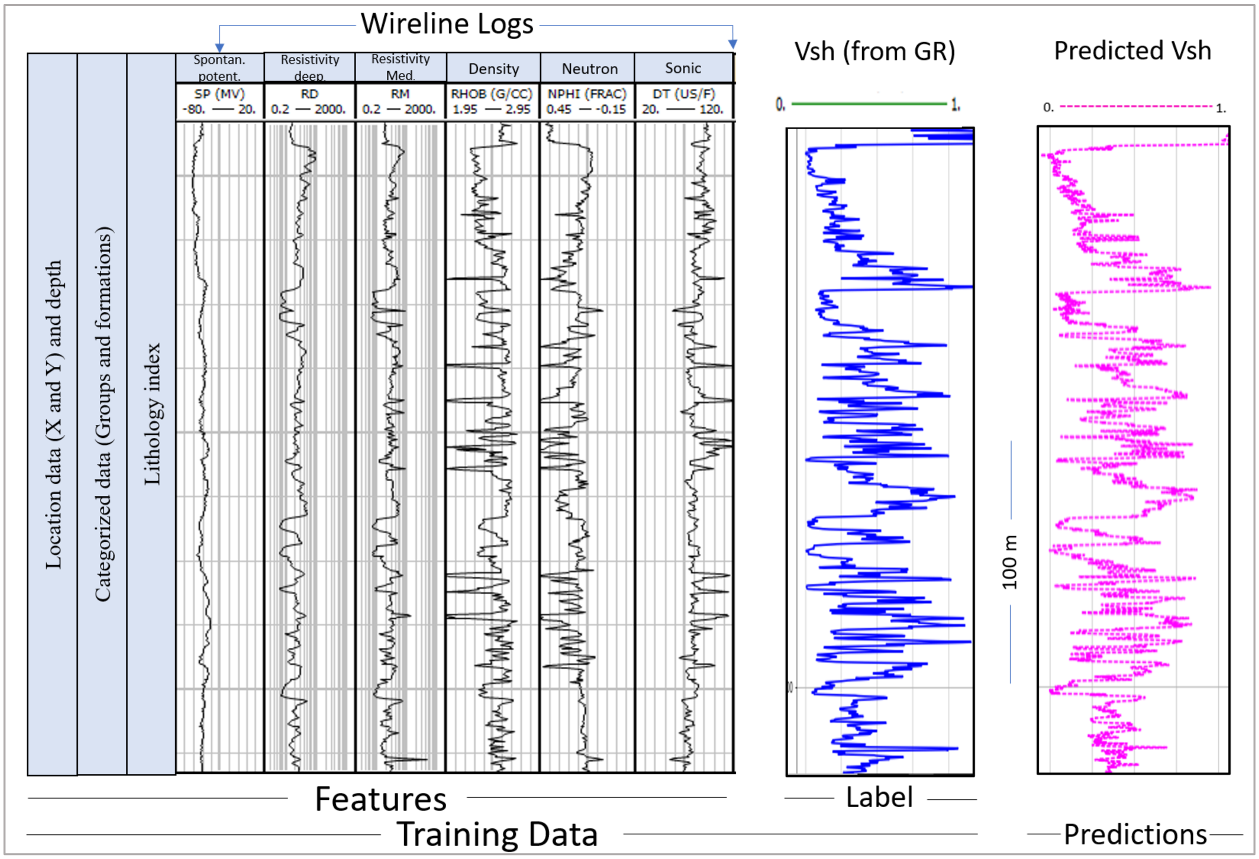


Figure 13: Machine learning model inputs and outputs overview. From the left, input features set, Vsh label from petrophysical analysis (blue curve), and predicted Vsh to the right (pink curve).

3.6.5 Models generation

Since the desired predicted feature or Vsh is a continuous value, the chosen approach for ML in this study is the supervised regression method, as previously discussed. This script was implemented in Python using the scikit-learn library. The overall workflow for creating the ML model is shown in Figure 14.

To train and evaluate the ML models, the dataset is divided into three sets: the training set, the validation set, and the testing set. The training set is used to train the model, the validation set is used to measure the model performance, and the test set is used to evaluate the real performance of the model on a dataset that the model has not seen before (also called blind test). Basically, any data contributing to one set must be removed from the other sets. More training data is beneficial since it allows the model to observe more examples. The training and validation sets are used to create the models and their hyperparameter tuning (see section 3.6.7). Every decision about the model and its hyperparameters is based on the validation set. The test set is then utilized to evaluate the model's real abilities.

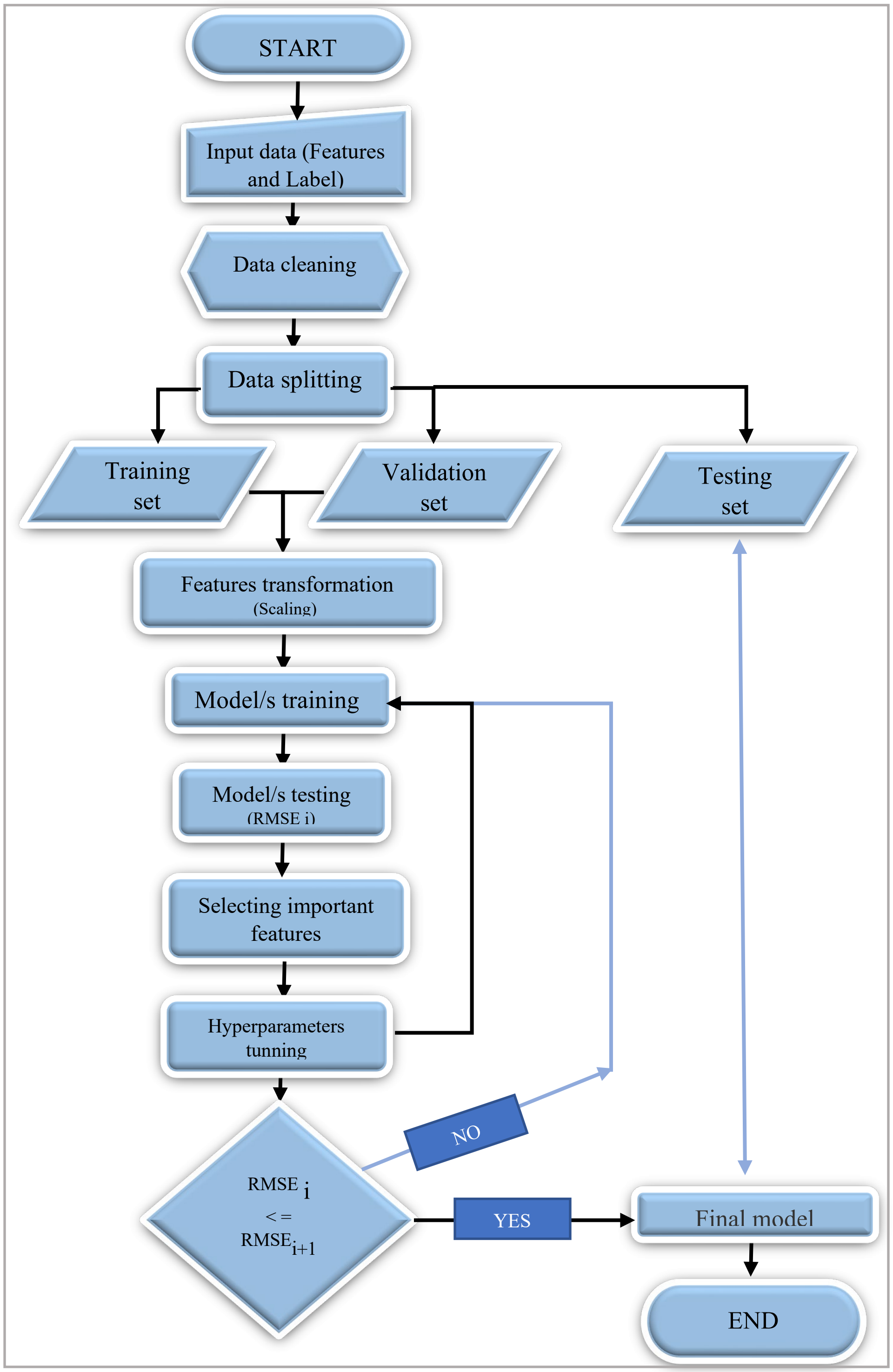


Figure 14: Flowchart depicting the steps involved in creating the ML model in this study.

In this study, the data from 55 wells is divided into two files. The first file contains data of 50 wells, for model train and validation purposes. The second file contains data of 5 wells, for testing purpose. The building of the ML models started by using the first (train and validation) file. For that, these data were split randomly into training and validation sets containing respectively 80% and 20% of the data (40 and 10 wells). Also, feature scaling was performed. This is a technique that normalizes the features' values into a standard range of values, such as 0 to 1. The machine learning models were then built on the training set with the desired algorithms.

3.6.6 Model performance evaluation

It is necessary to assess the ML model based on its performance; this enables understanding the model accuracy and the possibilities to improve it. Metrics are used to compare the trained model estimates with results from the testing data set, to assess the accuracy of the regression model. The main concept for finding the best model for a given data set is by estimating the prediction error of the supervised model (Harrell, 2001; Clarke et al, 2009). The optimal model is the one with the lowest prediction error.

It is important to use the right metric to measure accuracy while trying to improve the accuracy of a regression model. The Mean Absolute Error (MAE), Root Mean Square Error (RMSE), and Coefficient of Determination (R2) are examples of the most used statistical metrics. The RMSE is widely used as a standard metric for model errors in the geosciences (e.g., McKeen et al., 2005; Savage et al., 2013) and is used in this thesis.

$$RMSE = \sqrt{\frac{\sum_{i=1}^{N}(Predicted\ i - Actual\ i)^2}{N}}$$

Where:

N = Total number of observations.

Predicted $_i$ = The predicted value for the *i* observation.

Actual $_i$ = The observed value for the *i* observation

N = Total number of observations.

3.6.7 Model optimization

This is a crucial stage that can boost the model predictability. In the current study, it was done by applying two main steps to boost the model's accuracy, namely features selection and hyperparameter tuning.

3.6.7.1 Features importance and selection

The method of selecting the best subset of relevant features that best describes the relationship between independent variables and the target variable is known as feature selection. There may be features in the training data set that can be disabled without causing significant information loss (Bermingham et. al., 2015). With fewer features to deal with, processing is speed up, and model interpretability improves as overfitting is reduced.

One method is to display the correlation matrix. Pearson's correlation coefficient (Sedgwick, 2012) was used to create the correlation matrix in this study. The range of values for the correlation coefficient is -1 to 1, with 1 representing positive linear correlation, 0 representing no linear correlation, and -1 representing negative linear correlation.

More features may lead to a decline in the model accuracy, as they may create unnecessary noise. When a dependent variable and an independent variable have a high correlation value, it means that the independent variable is particularly important in determining the output. A high correlation between dependent and independent variables is desirable in general, whereas a strong correlation between two independent variables is undesirable.

Another option is to use one of the feature importance approaches, such as XGBoost's built-in feature importance, the permutation feature importance model inspection (Breiman, 2001), or any other technique. In this study, the Pearson’s correlation coefficient is used to assess features in general as it captures linear relationships between the input features and the label, while the permutation feature importance is used to discover which features are most relevant while developing the model.

3.6.7.2 Hyperparameters tuning

It is not enough to choose the correct algorithm. We must also tune the “hyperparameters” to find the best algorithm configuration for a dataset. Hyperparameters are parameters in ML algorithms that govern the learning process. Tuning is the process of determining which of a model's hyperparameters produces the most accurate predictions. The performance of ML algorithms varies greatly depending on the hyperparameters used (e.g. Hutter, 2014; Schratz et.al., 2019). Hyperparameter tuning can be done in a variety of ways, including Manual Search, Random Search, and Grid Search (Figure 15).

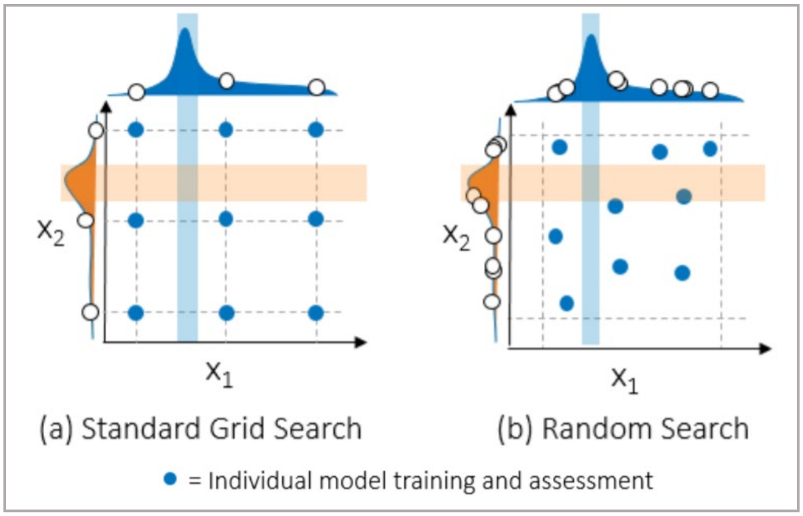


Figure 15: Hypothetical hyperparameter distributions (X1 and X2) with regard to a training objective (Koch et. al., 2017). The grid search algorithm searches for distinct hyperparameter combinations in order, whereas random search finds hyperparameter combinations at random.

Grid search was used to tune hyperparameters in this study. Even though the Grid search algorithm requires a lot of computing power and is time consuming, it proved to be an effective tuning method. Even though other search algorithms may have more appealing characteristics, grid search remains the most popular method due to its mathematical simplicity (Bergstra and Bengio, 2012). To overcome the overfitting constraint imposed by the standard Grid search, stratified cross-validation (CV) was used, in which samples are randomly divided into K-folds. The training data is subdivided into K subsets. The algorithm uses K-1 of the subsets as training data for each K fold, and then validates the model using the remaining subset by calculating the performance metric (such as RMSE for regression) (Figure 16). As a result of the CV, the algorithm computes the average of the performance metric from all the K-folds.

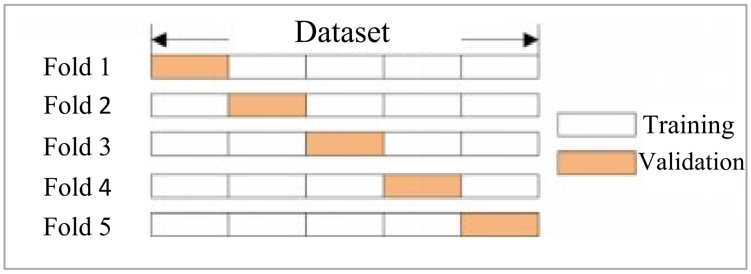


Figure 16: A schematic of five-fold cross-validation.

4. Results

The results of the Vsh estimations from all used methods are presented in this chapter. Starting with petrophysical evaluation and continuing with ML models, the results are displayed and represented using tables, and plots.

4.1 Petrophysical analysis

Vsh results from well logs are compared with the core grayscale curves and core visible shale descriptions for better calibration and evaluation. Figures 17 and 18 show the Vsh estimates from well logs in the Brent Group sandstones for three key wells in the dataset. Figure 17 includes the linear and non-linear Vsh estimates from the GR log, while Figure 18 includes the Vsh estimates from all logs and their combinations.

According to the Vsh results from the GR approaches, the nonlinear GR equations are more optimal as they produce more consistent Vsh values than those obtained with the linear equations (Figure 17). The Vsh results from the SP and neutron methods are generally higher than those from the non-linear GR method (Figure 18). Also, in certain cases some methods produce a very different Vsh value. For example, the SP method clearly overestimates Vsh in water zones (under the reservoir oil/water contact), which may be explained by low water salinity. An example of this is the Vsh (SP) log from well 34/10-21 in Figure 19. Overall, an increase of Vsh is the most common error in the Vsh estimates from these logs.

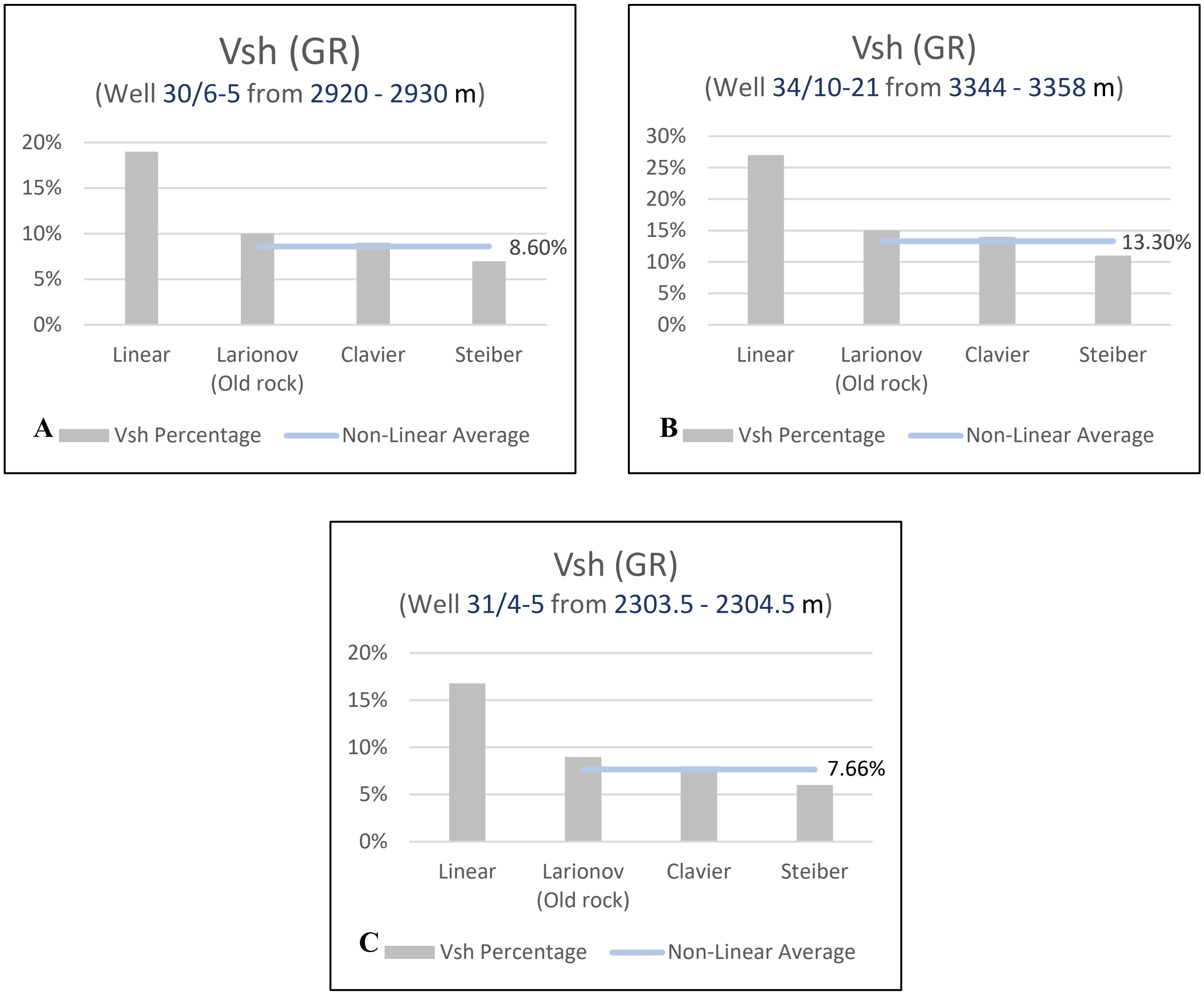


Figure 17: Examples of Vsh estimates in the Brent Group sandstones using the linear and non-linear GR equations. A) Vsh of well 30/6-5 from 2920 to 2930m depth, B) Vsh of well 34/10-21 from 3344 to 3358m depth, and C) Vsh of well 31/4-5 from 2303.5 to 2304.5m depth. Notice that the non-linear GR methods produce more consistent and lower Vsh values. Location of the wells is indicated in Figure 1.

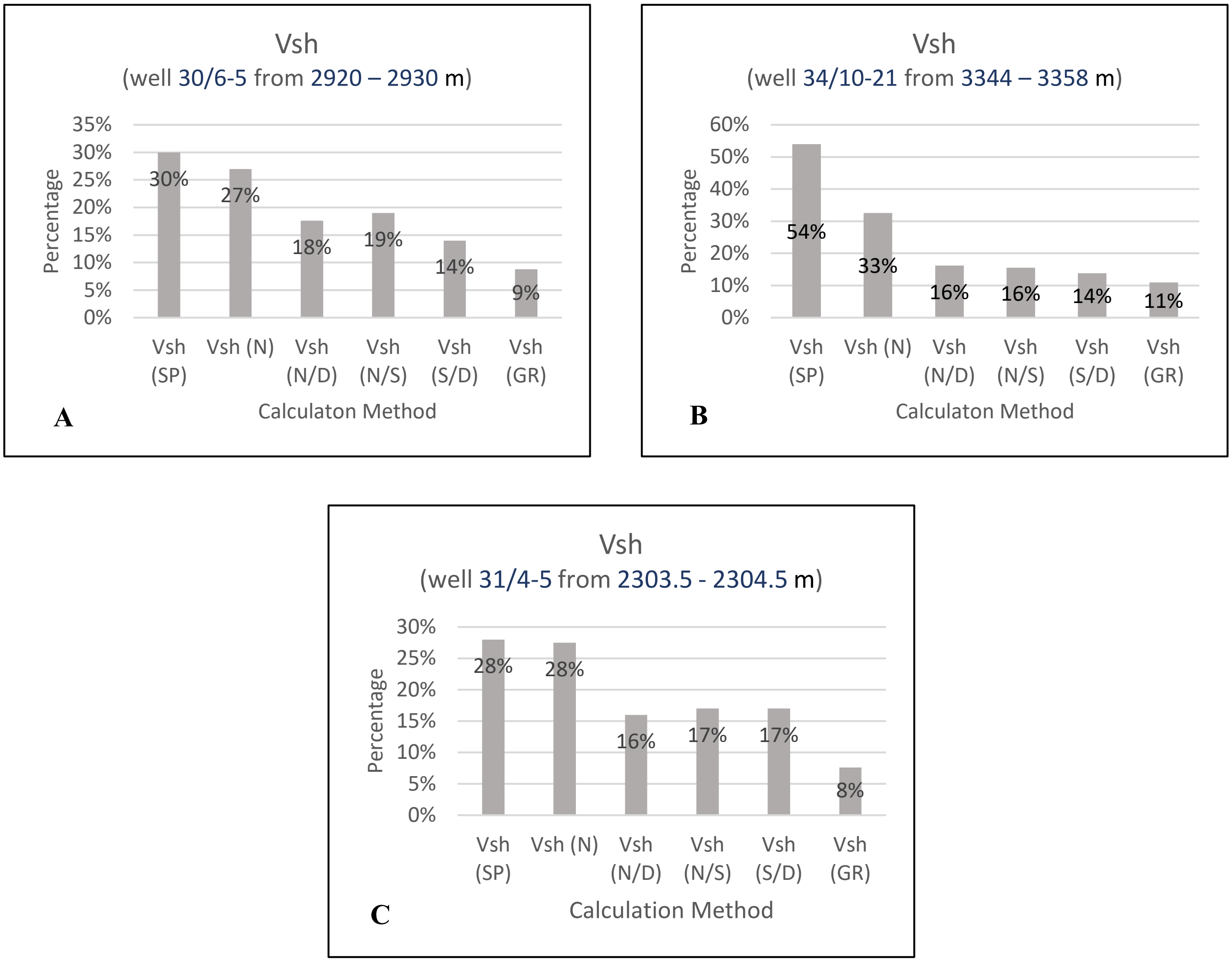


Figure 18: Examples of Vsh estimates in the Brent Group sandstones using different logs and their combinations: SP, neutron (N), neutron and density (N/D), neutron and sonic (N/S), sonic and density (S/D), and non-linear GR. A) Vsh of well 30/6-5 from 2920 to 2930m depth, B) Vsh of well 34/10-21 from 3344 to 3358m depth, and C) Vsh of well 31/4-5 from 2303.5 to 2304.5m depth. Location of the wells is indicated in Figure 1.

Figures 19 and 20 show the Vsh estimates from the core-image grayscale pixel values, in comparison to the Vsh estimates from the well logs, for two key wells in the dataset that are also included in Figures 17 and 18. The core-image grayscale pixel curves display good resolution and sharp boundaries, suggesting that they could reflect Vsh accurately. A general observation is that all Vsh estimations from the various petrophysical methods and grayscale pixel values have on average similar patterns with reasonable vertical resolution, but different absolute values (Figures 17 to 20).

Comparing the various Vsh petrophysical methods to the core grayscale pixel curves and core visible shale descriptions (Figures 19 and 20) shows that the Vsh results from the double indicators generally outperform the single indicators, except for the Vsh from the non-linear GR. Also, the double indicators and the non-linear GR methods show good correlation (Figure 18). Based on these observations, the average of the three nonlinear GR methods (Larionov old rock, Clavier, and Steiber) was used to create the Vsh label for ML and model training in the next phase (section 5.2), reducing any bias related to the GR selection method.

A reliability summary was carried out to show which of the other methods or well logs perform better in the Vsh estimation. To do this for the considered depth interval, the Vsh from the non-linear GR method was selected to be the reference as it is the most accurate estimation when compared to the core image Vsh. Table 2 shows the reliability of the Vsh estimation from the N, SP and RD logs in comparison to the non-linear GR log, for six key wells (two of them in Figures 17 to 20). In general, out of the six tested wells, the neutron and SP methods show less errors while resistivity is very inaccurate in most cases. This analysis does not consider the actual value of Vsh (which is unknown), but it focuses on the visual errors range with respect to the non-linear GR Vsh. Also, these results could be applied to the rest of the wells in case one needs to do more comprehensive studies.

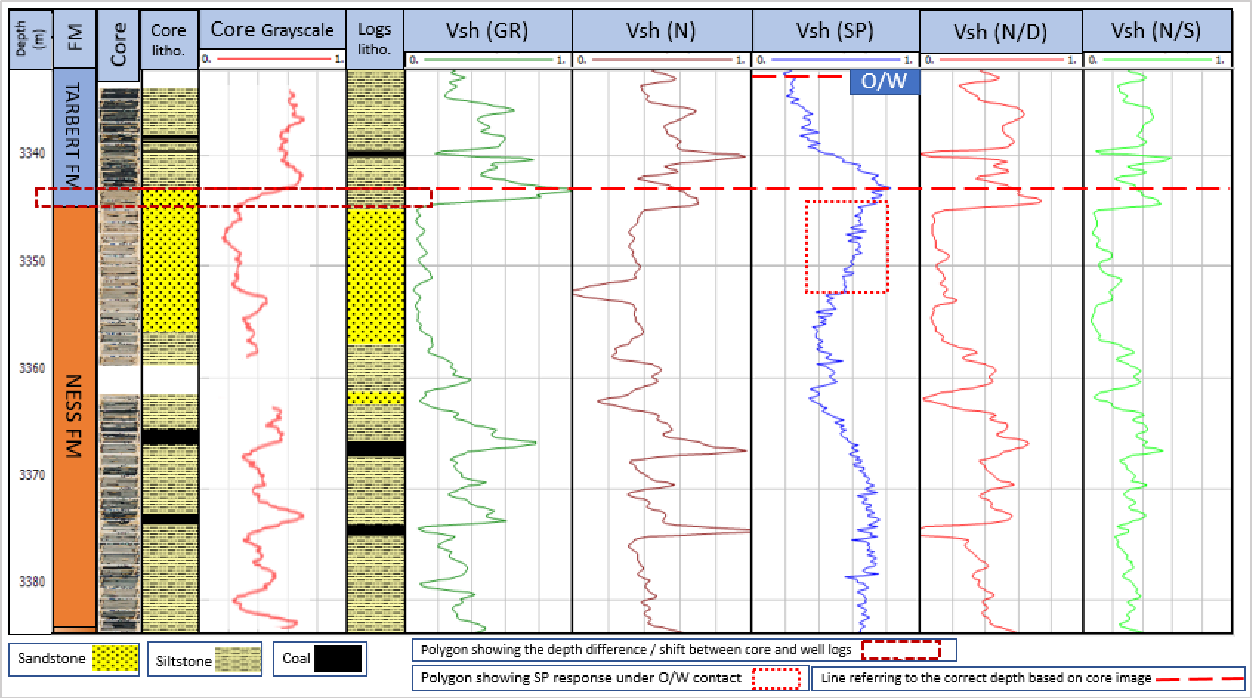


Figure 19: Vsh logs in well 34/10-21, including Vsh from the core-image grayscale pixel values, and Vsh from well logs. Vsh (GR) is Vsh from non-linear GR. Location of the well is indicated in Figure 1.

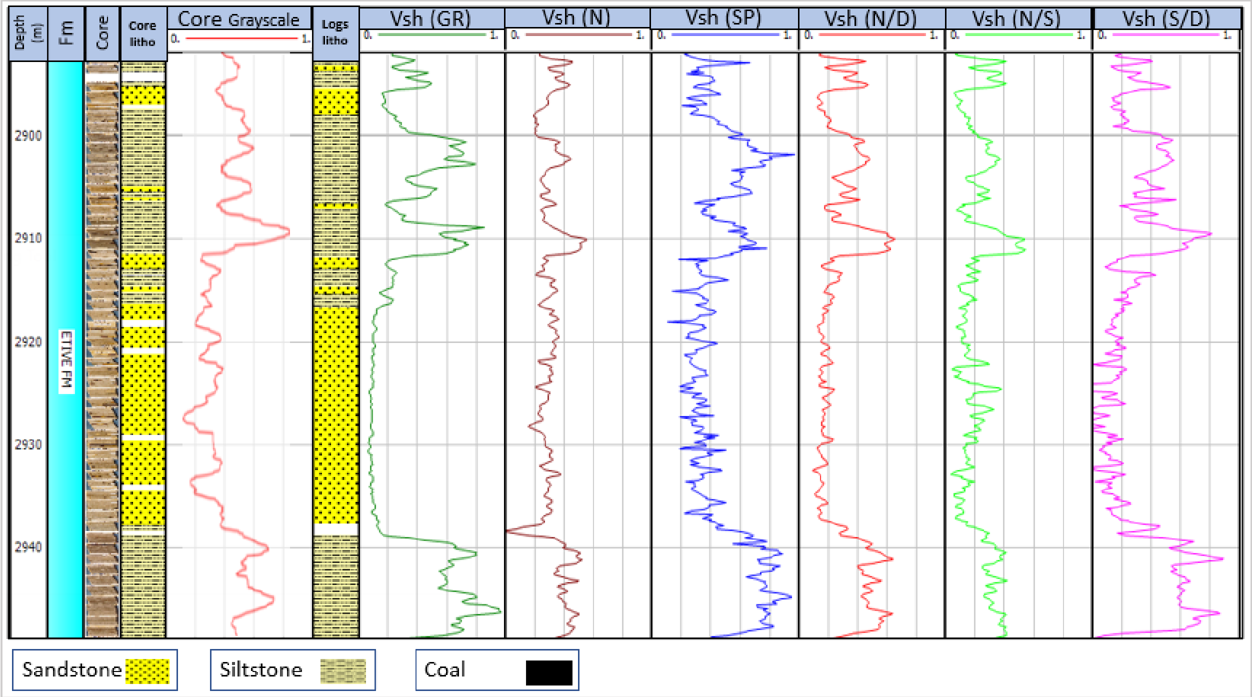


Figure 20: Vsh logs in well 30/6-5, including Vsh from the core-image grayscale pixel values, and Vsh from well logs. Vsh (GR) is Vsh from non-linear GR. Location of the well is indicated in Figure 1.

Table 2: Vsh single indicators and their *reliability in comparison the non-linear GR. 1 is good, 0 is not acceptable and '1' is acceptable with some errors. Locations of the wells are indicated in Figure 1.*

Well	Vsh (N)	Vsh (SP)	Vsh (RD)
30/3-3	1	1	0
30/6-5	'1'	'1'	0
31/2-8	1	1	0
31/2-19	1	'1'	0
33/5-2	1	1	'1'
34/10-21	1	1	'1'

4.2 Machine learning

4.2.1 Base line model

After assembling the dataset, the first step is to choose the optimal ML method for predicting Vsh. To accomplish this, the initial models of the previously described algorithms (XGBoost, RF, and KNN) are built using the well logs data. The models are trained on the training set. Then the performance of each trained model is evaluated based on the validation set RMSE score (Figure 21). The RMSE score is the error as compared to the labeled Vsh using non-linear GR. The model with the best validation set performance is chosen for further developing and testing. As the Vsh label was created using the GR logs, it has a high correlation (up to 100%) with these logs, which means that the prediction output would be biased by the GR. So, the ML models were trained without the GR logs.

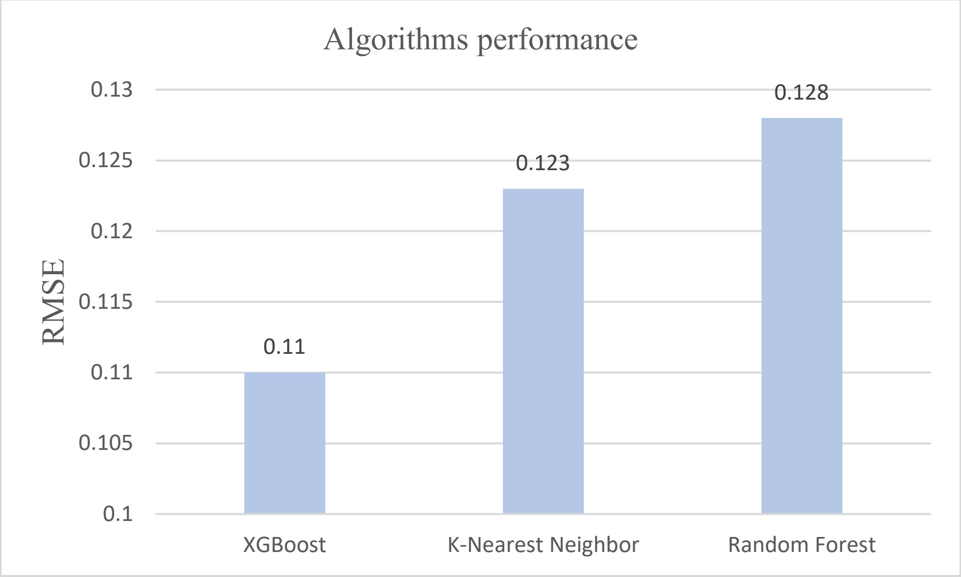


Figure 21: Performance index of the used ML algorithms in comparison to the Vsh label (non-linear GR).

As shown in Figure 21, all the three algorithms have approximately the same validation test error. However, the XGBoost model has the least error at RMSE = 0.11, and the highest performance in terms of computational resources and time to run the model compared to RF and KNN. Therefore, XGBoost algorithm was selected as the main approach to further develop the ML model.

4.2.2 XGBoost models

To get a better understanding of how ML models perform Vsh predictions based on feature inputs, three models were developed using the XGBoost algorithm based on different numbers and kinds of features. The three models' RMSEs are then compared (Figure 22), and the model with the least error is chosen for further development (section 4.3.3). The first model uses seven features, consisting of the basic well logs (Neutron (NPHI), density (ROHB), spontaneous potential (SP), compressional sonic (DTC), deep resistivity (RDEP), and medium resistivity (RMED)) and depth data. The second model uses nine features, including those from the first model as well as location data (X coordinate (X_LOC) and Y coordinate (Y_LOC)). The third model uses the same features as model 2 as well as categorical GROUP and FORMATION features (stratigraphic information). The models' RMSEs show that model 3 has the lowest errors in both the training and validation datasets, while model 1 has the highest errors (Figure 22). Model 3 is therefore taken forward.

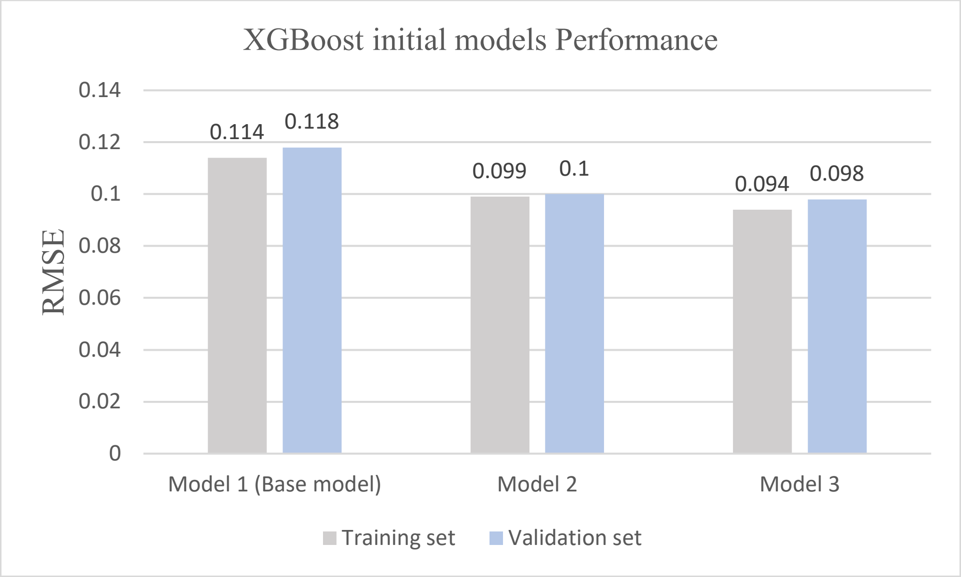


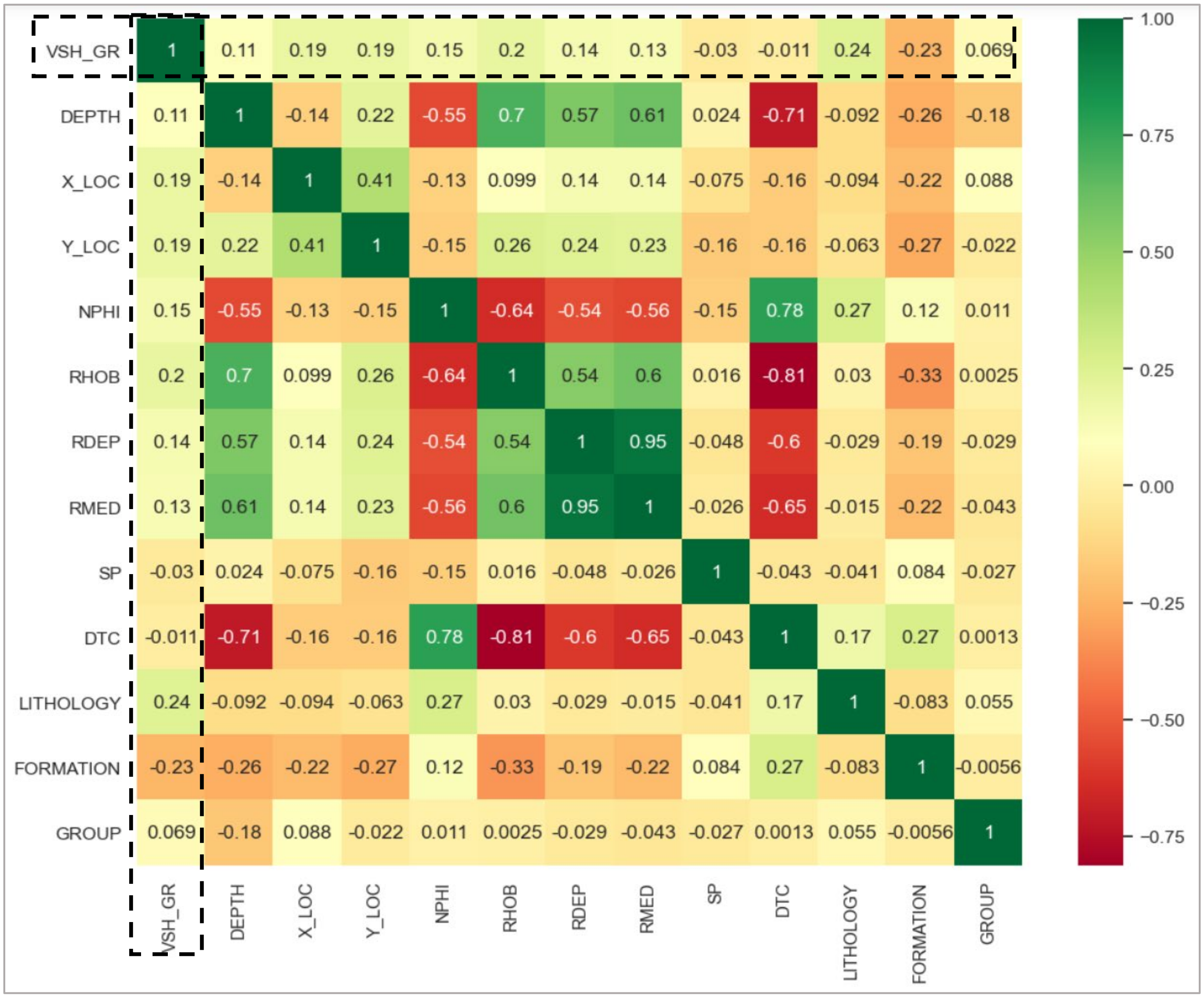
Figure 22: XGBoost initial model performance (in terms of RMSE) for three models with 9 (well logs, model 1), 11 (well logs plus location, model 2), and 13 (well logs, location and stratigraphic information, model 3) features, and the training and validation data.

4.2.2.1 XGBoost model optimization

As model 3 has the best performance (lowest RMSE), it was optimized in order to develop the final model.

Feature importance

A correlation matrix was computed to evaluate which features are the most relevant to the models (see section 3.6.7.1). The correlation between the dependent variable (VSH_GR, label) and the independent variables (features) shows that most of the features have a positive correlation with the label (VSH_GR) (Figure 23). However, SP, DTC, and GROUP show no linear correlation to the label. FORMATION is the only feature with negative correlation to the label with value of -0.23. The correlation between independent variables (features) reveals a strong correlation between DTC, DEPTH, NPHI, and RHOB. Also, there is a high correlation between RDEP and RMED with value of 0.95. Furthermore, when DEPTH, NPHI, and RHOB are correlated with DTC, the correlation value exceeds 0.70. This strong correlation between independent features indicates that it is preferable to remove DTC and RMED (or RDEP) from the training data before hyperparameter tuning, because it is regarded as a form of repetition that consumes time and computation.



	VSH_GR	DEPTH	X_LOC	Y_LOC	NPHI	RHOB	RDEP	RMED	SP	DTC	LITHOLOGY	FORMATION	GROUP
VSH_GR	1	0.11	0.19	0.19	0.15	0.2	0.14	0.13	-0.03	-0.011	0.24	-0.23	0.069
DEPTH	0.11	1	-0.14	0.22	-0.55	0.7	0.57	0.61	0.024	-0.71	-0.092	-0.26	-0.18
X_LOC	0.19	-0.14	1	0.41	-0.13	0.099	0.14	0.14	-0.075	-0.16	-0.094	-0.22	0.088
Y_LOC	0.19	0.22	0.41	1	-0.15	0.26	0.24	0.23	-0.16	-0.16	-0.063	-0.27	-0.022
NPHI	0.15	-0.55	-0.13	-0.15	1	-0.64	-0.54	-0.56	-0.15	0.78	0.27	0.12	0.011
RHOB	0.2	0.7	0.099	0.26	-0.64	1	0.54	0.6	0.016	-0.81	0.03	-0.33	0.0025
RDEP	0.14	0.57	0.14	0.24	-0.54	0.54	1	0.95	-0.048	-0.6	-0.029	-0.19	-0.029
RMED	0.13	0.61	0.14	0.23	-0.56	0.6	0.95	1	-0.026	-0.65	-0.015	-0.22	-0.043
SP	-0.03	0.024	-0.075	-0.16	-0.15	0.016	-0.048	-0.026	1	-0.043	-0.041	0.084	-0.027
DTC	-0.011	-0.71	-0.16	-0.16	0.78	-0.81	-0.6	-0.65	-0.043	1	0.17	0.27	0.0013
LITHOLOGY	0.24	-0.092	-0.094	-0.063	0.27	0.03	-0.029	-0.015	-0.041	0.17	1	-0.083	0.055
FORMATION	-0.23	-0.26	-0.22	-0.27	0.12	-0.33	-0.19	-0.22	0.084	0.27	-0.083	1	-0.0056
GROUP	0.069	-0.18	0.088	-0.022	0.011	0.0025	-0.029	-0.043	-0.027	0.0013	0.055	-0.0056	1

Figure 23: Correlation matrix between features and label (VSH_GR). Neutron (NPHI), density (ROHB), spontaneous potential (SP), compressional sonic (DTC), deep resistivity (RDEP), medium resistivity (RMED), X coordinate (X_LOC) and Y coordinate (Y_LOC).

The permutation features importance, model inspection technique, was applied to show the relationship between the features and the predictions. The features importance for model 3 (Figure 24) shows that X_LOC, DEPTH, NPHI and Y_LOC have the highest importance, while resistivity logs (RDEP and RMED) have the lowest importance. Based on this, RDEP and RMED can be removed from the training set without having a significant impact on the model. This could be done in the event of a lack of computational power, memory, or time constraints, but it was not necessary in the current study.

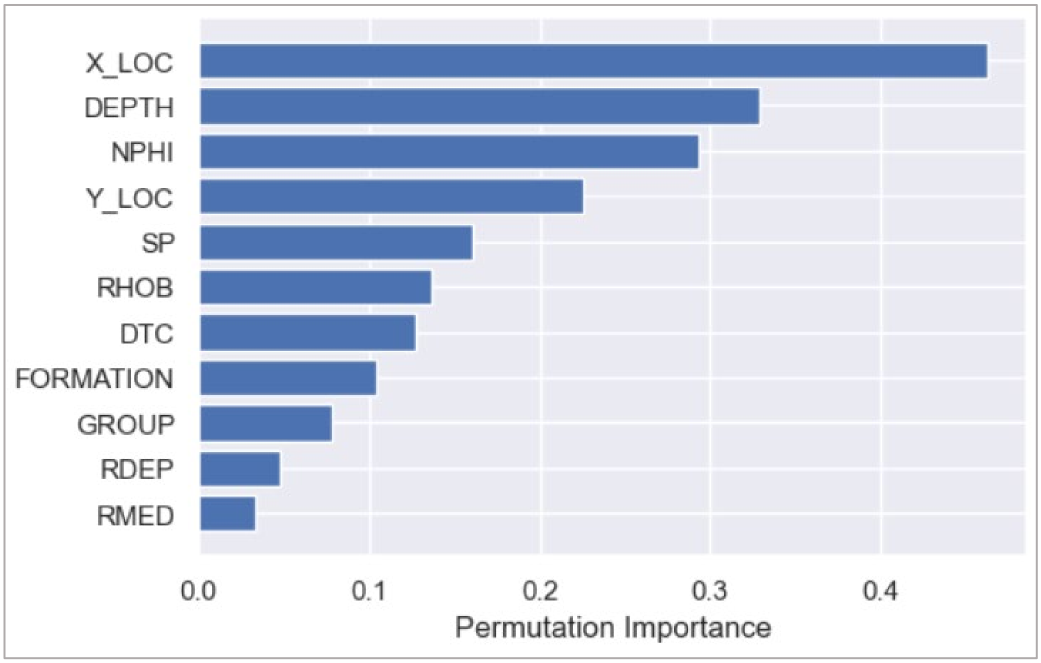


Figure 24: Features importance for Model 3.

Hyperparameters tunning

Hyperparameters are tunned using the grid search cross-validation technique (see section 3.6.7.2 and appendix 1). The tunned hyperparameters include:

- *n_estimators,* which is the number of gradient-boosted trees.
- *Learning rate (eta)*, which is a key hyperparameter that makes the model more robust by shrinking the weights on each step which prevent overfitting.
- *Max_depth,* which is a parameter used to prevent overfitting because a higher depth allows the model to learn relations very specific to a particular sample.
- *Subsample, which* specifies the proportion of observations that will be randomly sampled for each tree. Setting *Subsample* to 0.5, for example, means that the algorithm will randomly sample half of the training data before generating trees, preventing overfitting.
- *Colsample_bytree,* which is the subsample ratio of columns when constructing each tree. For each tree built, subsampling takes place once.
- *min_child_weight*, which specifies the minimum weighted total of all essential observations in a child. This parameter is utilized to keep over-fitting under control.

After hyperparameter tuning (see appendix 1), the performance of the optimized model, model 4, was measured for the different sets (Figure 25). The results show that the RMSE of model 4 (Figure 25) is lower than model 3 (Figure 22).

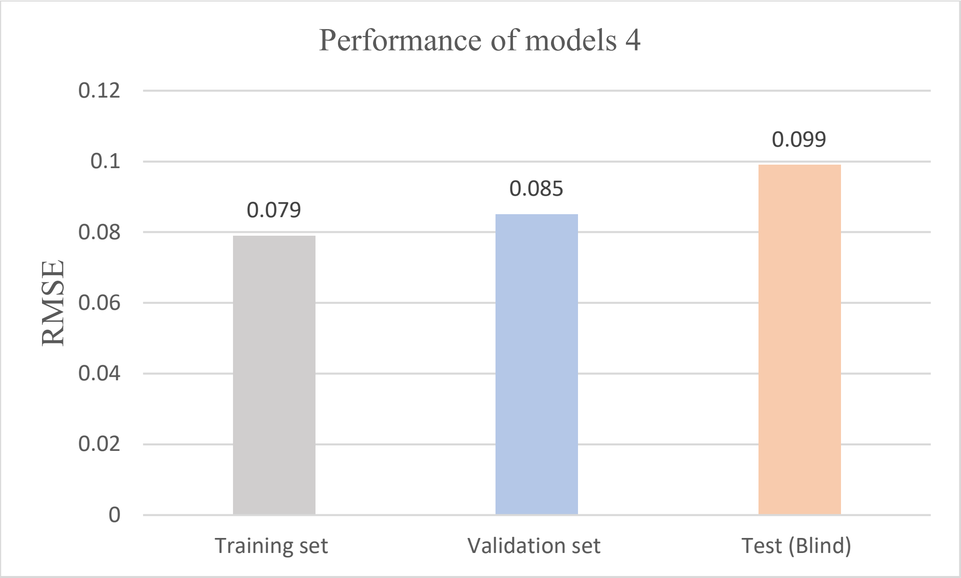


Figure 25: XGBoost model 4 and its performance indices as RMSE values for the training, validation, and test datasets.

4.2.2.2 Accuracy vs Lithology

To emphasize the precision of the validation and test steps, the RMSE of the Vsh predictions from model 4 (lowest error) was tested against lithologies (Figures 26 and 27). The validation results are displayed for the whole interval, while the test results are displayed for the Brent Group to highlight the reservoir blind predictions. Both the validation and test sets display good results (low RMSE) with respect to the different lithologies. Sandstone shows one of the best scores, which is very good since the main hydrocarbon bearing reservoir intervals in the study area consist of sandstones (Brent Group). In addition, Vsh predictions for three of the test wells in Figure 28 show high performance, as they closely match the labeled Vsh (Vsh_GR) and reflect the low RMSE values from Figure 27.

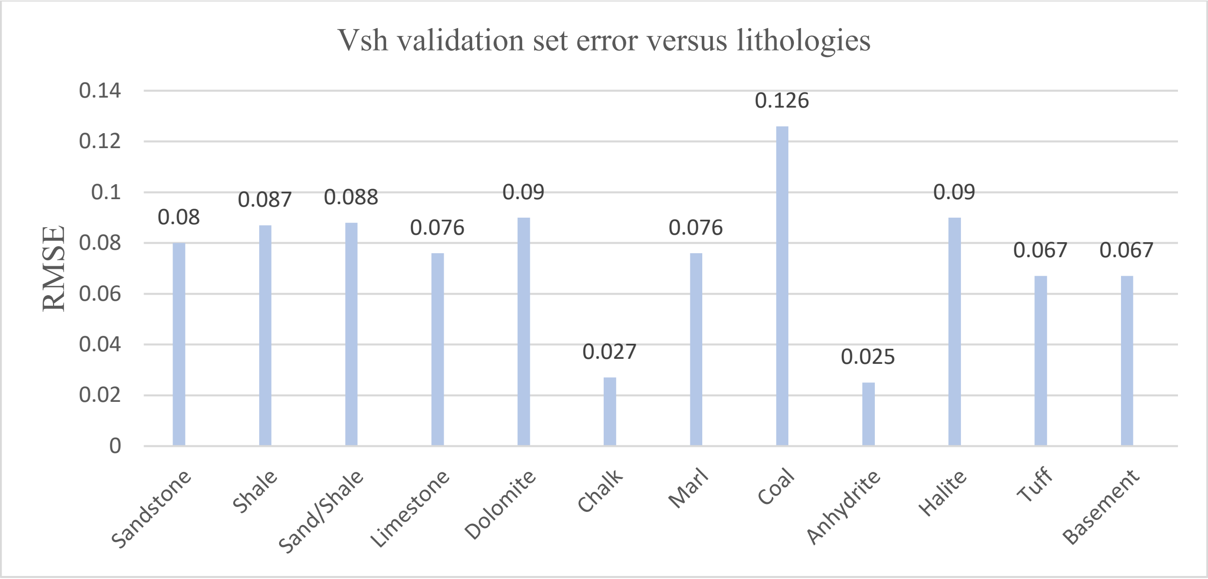


Figure 26: RMSE of validation set for model 4, different lithologies, and the whole borehole interval.

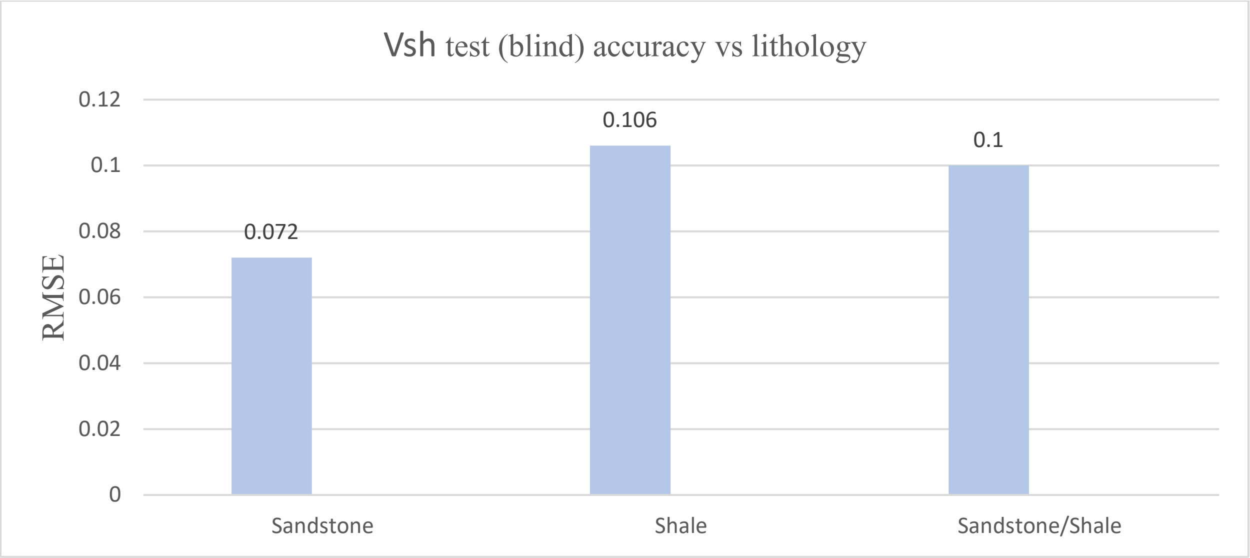


Figure 27: RMSE of test set for model 4, different lithologies, and the Brent Group.

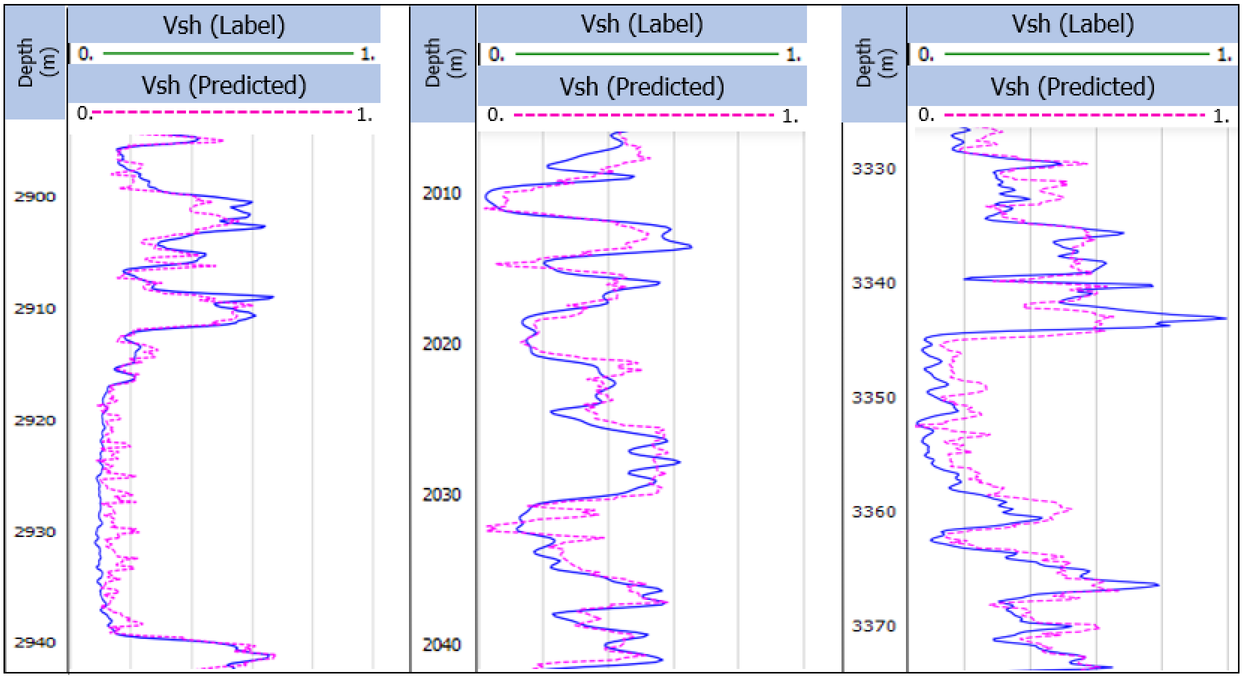


Figure 28: Blind Vsh predictions (dashed, pink line) for Brent Group using model 4 on three of the test wells. From left to right the wells are: 30/6-5, 31/6-5, and 34/10-21 (location in Figure 1). The Vsh (label, blue line) is the Vsh from the non-linear GR.

4.2.3 Model enhancements

4.2.3.1 Cluster-based models

As the correlation matrix and the features' importance (Figures 23 and 24) show that location data have significant importance, and the location map (Figure 1) of the wells show that they span a large geographic area, the prediction performance of model 3 was investigated using location clusters. Three wells location clusters are clearly visible in Figure 29. These clusters are used to develop three distinct models from model 3, i.e., with the same features that in this model.

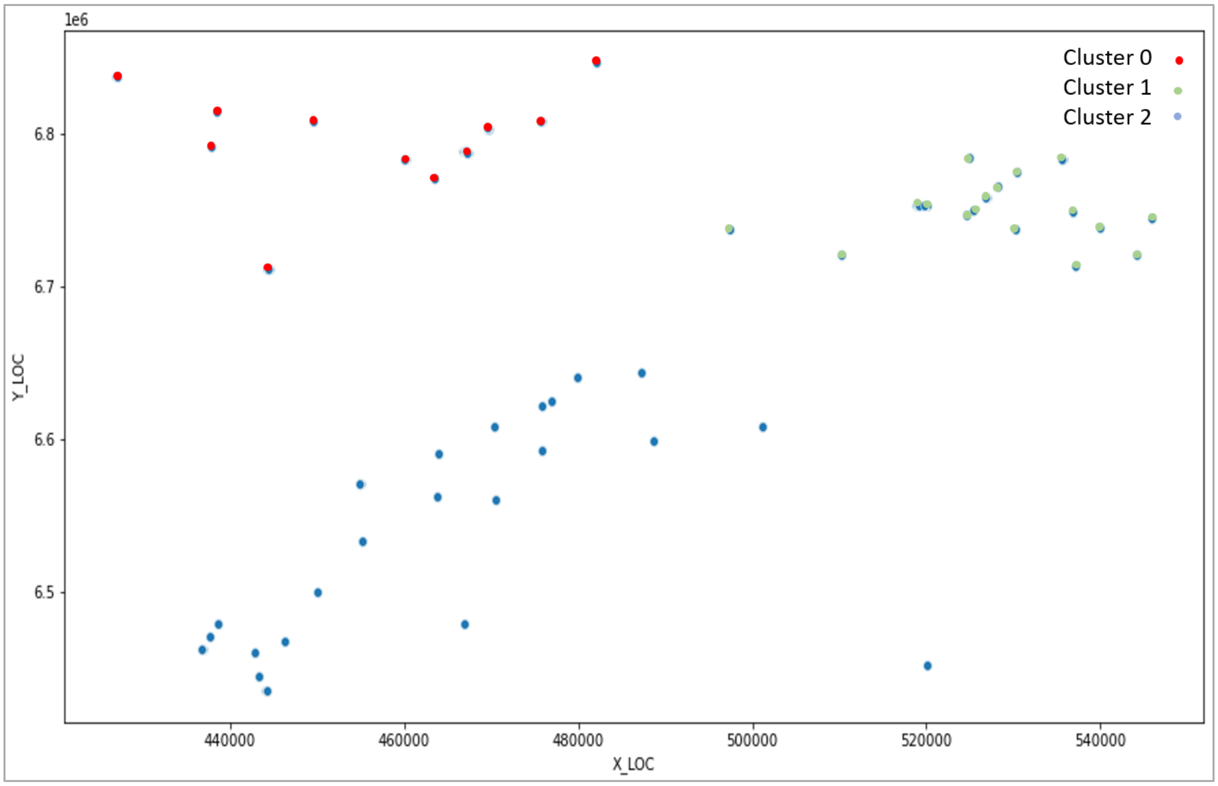


Figure 29: Clusters of wells locations. These are used to develop three distinct models from model 4.

The cluster-based initial models show lower RMSE error (Figure 30) than model 3 (Figure 22). This emphasizes the importance of clustering the wells by location.

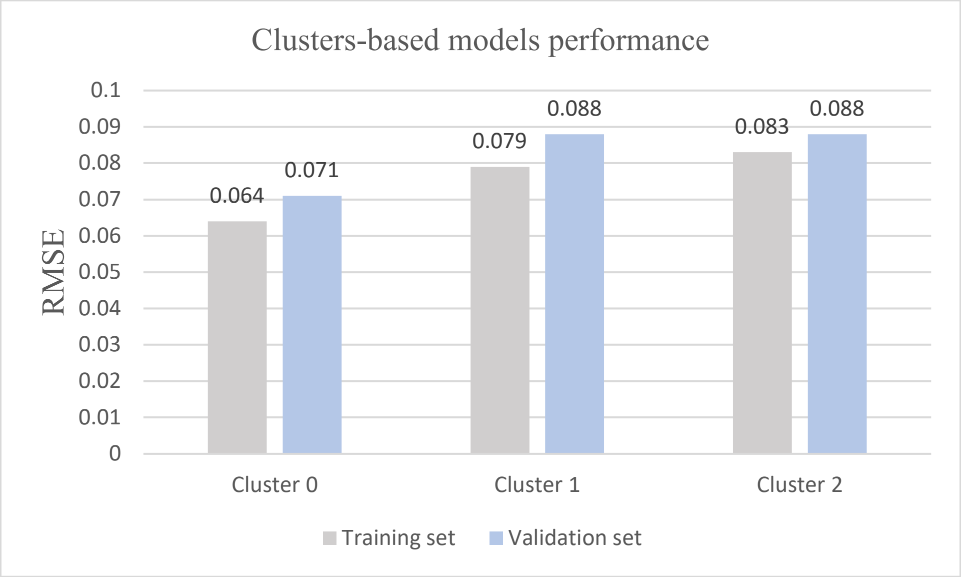


Figure 30: Performance of the clustered-based initial models defined by the location of the wells (Figure 29), for the training and validation datasets.

4.2.3.2 Lithology index feature

According to the correlation matrix (Figure 23), lithology indexes have a high correlation with the label, suggesting that they could improve the model accuracy. However, there is uncertainty about the interpretation of these lithology indexes (see section 5.2), which is why lithology as an input feature was not discussed in the early models. Figure 31 indicates that training the model with the lithology indexes included in the input features (model 5), reduces the error in comparison to model 4 (Figure 25) by a significant amount. For the validation set, the error of model 4 is 0.085 while that of model 5 is 0.078 (Figure 32).

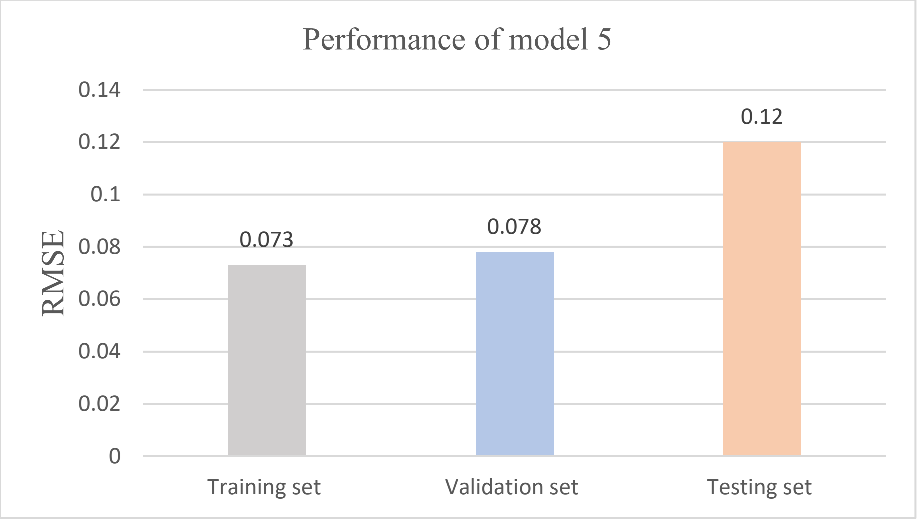


Figure 31: Model 5 performance indices as RMSE values for the training, validation, and test datasets.

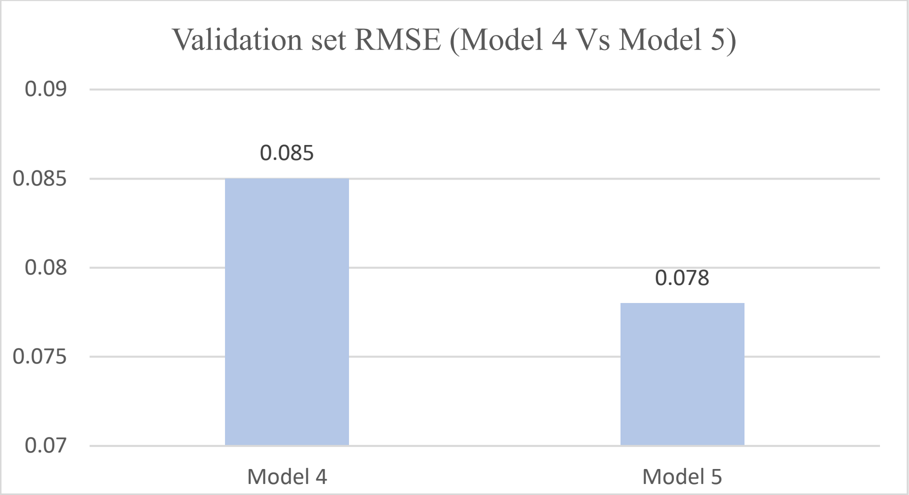


Figure 32: Performance index of model 4 compared to model 5 based on validation set. Model 5 includes lithology indexes.

4.2.3.3 Location-based model

As location features (X_LOC and Y_LOC) seem to have high importance, one intriguing idea is to train a model based on location and depth features only, and then evaluate the model's performance. Figure 33 indicates that the RMSE for an only location- and depth-based initial model (model 6) and for the training and validation datasets is acceptable in comparison to model 3 (Figure 22).

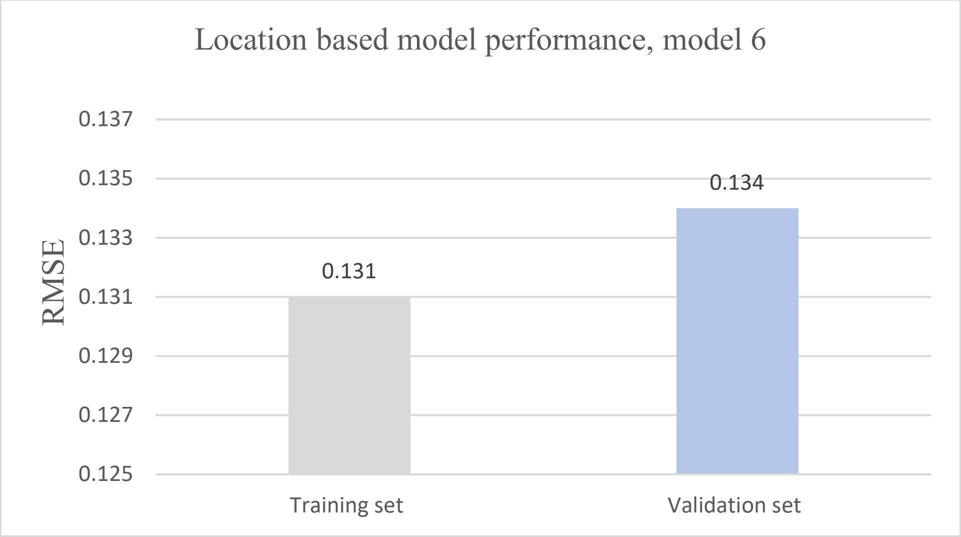


Figure 33: Performance index of the location-based model 6, for the training and validations sets.

5. Discussion

5.1 Core images and grayscale pixel curves

While challenging, using the cores images to generate numeric Vsh estimates proved to be very useful and allowed validation of the Vsh calculations using the classical methods. For future research, a large-scale dataset could be constructed from the core images and normalized to improve ML prediction of Vsh or other petrophysical parameters.

Core grayscale pixel curves could also be effective in other geological applications such as sequence stratigraphy and sedimentology. As these curves present boundaries sharply, they can be useful to identify sequence boundaries in the same way than GR logs. Also, the curves can be used to distinguish sedimentary environments based on color variation. For example, in Figure 19, clean sandstones have low grayscale values, while darker (grayish white) sandstones within the same zone have significantly higher grayscale values. Also, shifting the measured depth (MD) based on core depth is necessary to reduce its error. For instant, the core grayscale curves show that the log data are shifted down by an average of two meters in Figure 20.

During the curves' extraction process, some problems were discovered and resolved. The pixel values do not reflect the lithology itself but the color physical characteristics, so when there are faults, fractures, sample plugs, and oil stains, the grayscale values are incorrect. A filter was used to solve these issue besides manual data cleaning. The resolution of the grayscale curves could be increased by reducing the filtering effect during the extraction process to obtain more information from small scale color change.

5.2 Classical Vsh methods

Based on the petrophysical Vsh calculations, the results indicate that the Vsh from the nonlinear GR methods was the most optimal in comparison to the other methods, and it has the lowest errors compared to the core grayscale curves and core visible Vsh. The non-linear GR methods produce lower Vsh estimates than the linear GR method, which has been discussed in previous studies (e.g., David, 2015). An explanation for the differences between the linear and nonlinear GR methods, is that the nonlinear equations consider important variables like formation age and consolidation. Also, According to Rider (2002), there is no basis for assuming that the relationship between gamma ray value and Vsh is linear. This supports the choice of the nonlinear GR method (Figures 17 and 18) as the label for the ML phase.

5.3 Machine learning models

By performing the base line model (Figure 21), XGBoost shows the highest performance from the point of view of accuracy and speed, up to ten times faster than the other algorithms beside less memory use. According to the feature correlation matrix, 75% of the features used in this study have linear correlation with the label (Vsh_GR), while 25% do not have any correlation with the label. This explains the ML models performance in terms of RMSE. It also explains the relatively low RMSE values of the XGBoost, RF and KNN algorithms, as the dependent variable has good correlation with several of the independent variables.

The ML models based on different number of features, prove that including more features relevant to the label (i.e., with higher correlation to the label) could significantly decrease the model errors (Figure 22). A comparison of the RMSE results for the training and validation sets, in the XGBoost initial models with 7 features (model 1) and 11 features (model 3), shows that the RMSE of the training and validation sets decreases by 18% and 17%, respectively, in the model with more features (model 3) (Figure 34).

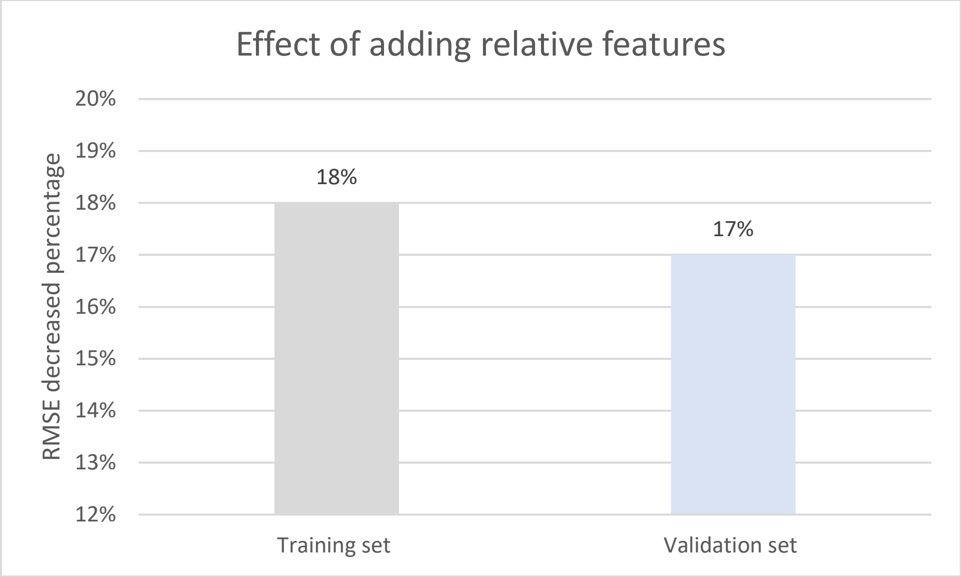


Figure 34: Features impact on models' performance. The RMSE in this figure measures the difference in error between a model with 7 features (model 1) and a model with 11 features (model 3), for the training and validation datasets.

The decrease in RMSE from model 1 (7 features) to model 2 (9 features) is greater than the decrease from model 2 to model 3 (11 features). This could be explained by the feature correlation matrix (Figure 23). In model 2, X LOC and Y LOC are included, and both have a positive correlation with the label (VSH_GR) of 0.19, indicating that they are very important. On the other hand, FORMATION and GROUP are the two extra features in model 3. FORMATION has a negative correlation with the label of -0.23, whereas GROUP has no correlation with the label, indicating that GROUP is not very important. To see if this is the case, model 3 was run without the GROUP, and it delivered similar RMSE values.

Tuning the hyperparameters was an essential step to optimize the ML model as it significantly decreased the validation set RMSE by 13 % from model 3 to model 4 (Figure 35). An important observation is that using the grid search cross-validation technique to search the best range of hyperparameters consumes a lot of computational power and time, but it is very effective in controlling the overfitting. Consequently, tuning the hyperparameters using grid search cross-validation is recommended. The difference in range of error (RMSE) between the validation and test sets in model 4 is around 15% (Figure 25), indicating acceptable overfitting effect.

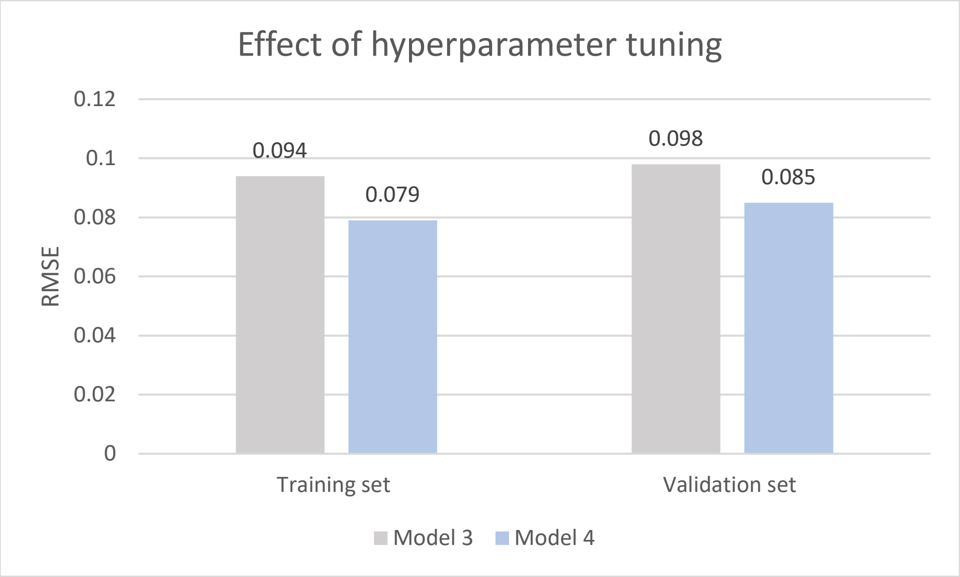


Figure 35: Effect of hyperparameter tuning on model performance (RMSE) and the training and validation datasets.

By studying the features importance after running model 3 (Figure 24), it was found that the most important well logs are NPHI and SP, and the least important logs are the resistivity ones (RDEP and RMED). This is consistent with the reliability analysis from the classical petrophysical part (Table 2). This means that the algorithm finds statistically logical relationships between the features, emphasizing the need for comprehending the problem and the desired output (label) to maximize the impact of ML. Thus, extra features highly related to the intended output should be created and used in the model training.

Another observation is that the SP log has high importance based on the permutation feature importance technique. However, based on the correlation matrix, SP has no correlation with the label. According to Saarela (2021), the model's most important features can vary depending on the used technique. Since the results of the petrophysical analysis render SP as “important”, this makes the permutation features importance model a more reliable technique than the correlation matrix. Thus, feature importance models should be preferred as the main tool to select features. This selection could be further supported with the correlation matrix.

As the research is motivated by applying new ideas to increase the methods’ value, the following three novel ideas are tested:

- Creating models based on the clustering of wells locations.
- Including lithology index in the input features.
- Building a location- and depth-based model using only the location (X_LOC, Y_LOC) and DEPTH features.

By creating three initial models based on wells locations clusters (Figure 29), the validation set's RMSE decreased by 27% in cluster 0 and 10% in clusters 1 and 2 (Figure 30) when compared to the RMSE findings of model 3 (Figure 22). This reflects the importance of location clustering in the study. It also highlights the importance of considering the geological setting while making predictions, as this could improve the model.

Another model (model 5) was created with the same features as model 4 plus the lithology index. Including the lithology index as a feature significantly decreases the RMSE of model 5 for the validation set by 8% compared to model 4 (Figure 36), which is explained by the high correlation of the lithology index with the Vsh label (Figure 23). However, there is a doubt regarding the lithology index. In the data, the lithology index is labelled as lithofacies, but it seems to be log facies (lithology interpretation from well logs), and in that case GR logs may have been used as input for the interpretation process. This means that the lithology index can produce biased results. To overcome this, it is recommended to interpret lithology without using the GR logs, which is possible. Another approach is to predict lithologies without utilizing GR as an input feature, then use the lithology predictions as input to the Vsh model. By applying these corrections, the model is expected to perform nearly the same, but it would be less biased and more reliable. This will make model 4 the best model for Vsh prediction.

Another observation is that, while the lithology indexes improved the RMSE of model 5 for the validation set, they were not good enough in the blind test where model 5 performs worse than model 4 (Figure 36). This could be due to one of two reasons; the first is that the lithology predictions or interpretations of the FORCE competition dataset are not good enough, which may be the primary reason why the lithology prediction of the FORCE competition was not very good. The second is that the model is overfit and requires further effort with model tuning.

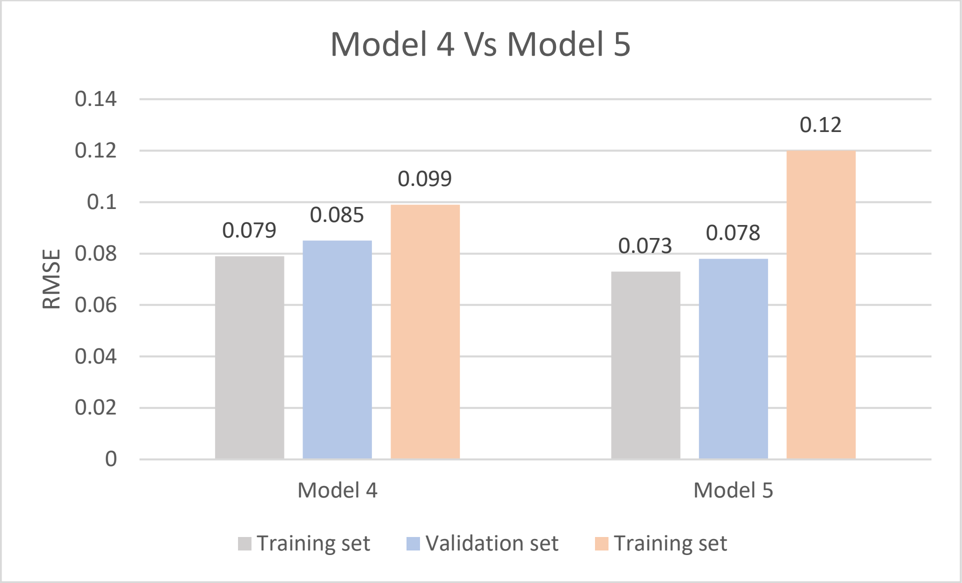


Figure 36: Performance index of model 4 compared to model 5.

The location- and depth-based initial model (model 6) performance on the validation set is acceptable from the point of view of RMSE, but still needs to be optimized (Figure 33). Nonetheless, this is a good start and could be considered as a base line model that needs further development. A good location- and depth-based model could allow Vsh prediction even before drilling the well. It is recommended to add more wells and test new features such as seismic attributes to improve the model's accuracy.

Figure 37 compares all the Vsh estimation approaches investigated in this study, including petrophysical techniques, core image grayscale curves, and ML Vsh predictions. It is evident from this figure that the ML method is effective and accurate, and the core grayscale method is promising.

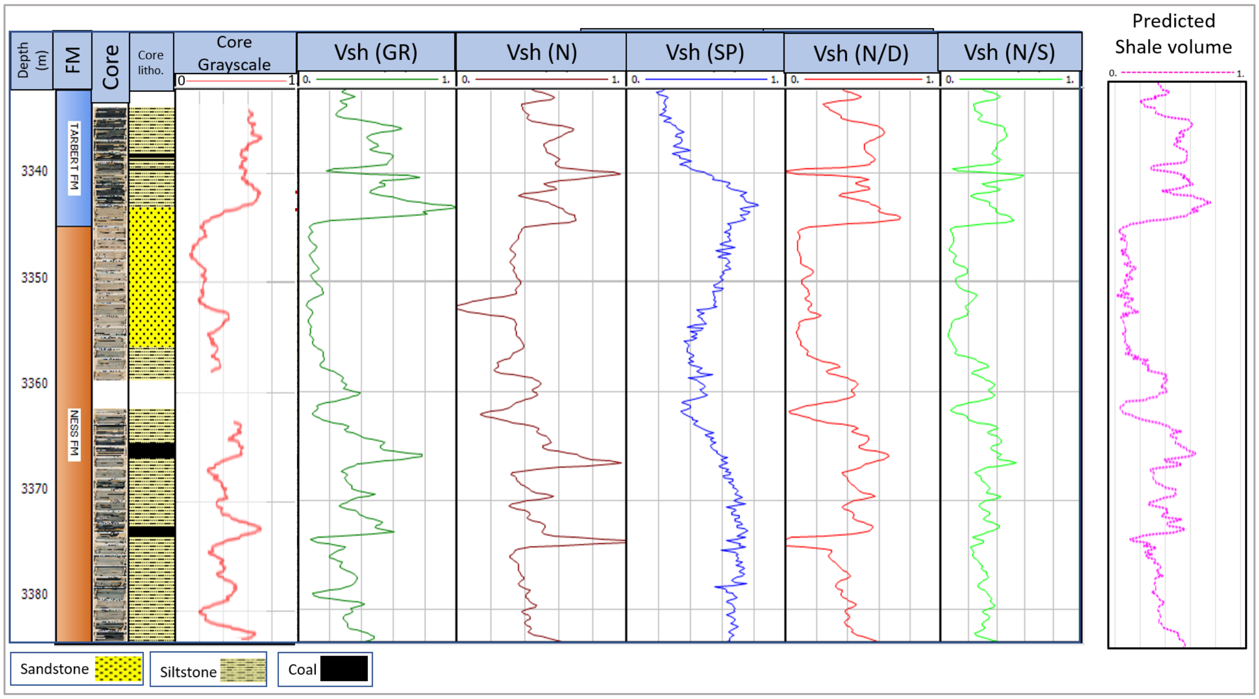


Figure 37: Different Vsh methods considered in this study. From left, core visible Vsh, core grayscale curve, Vsh from well logs and Vsh from ML model 4. Example from well 34/10-21 (location in Figure 1).

5.4 Future work

Here are some recommendations for future research. I considered some of these at the beginning of the project but unfortunately, there was not enough time to research these topics:

- Using machine learning to create a large-scale dataset from the core images (grayscale curves). By incorporating this dataset in model training, one could improve ML prediction of Vsh or other petrophysical parameters.
- Testing other ML algorithms such as Artificial Neural Networks (ANN).
- Applying Deep Learning and comparing it to the ML techniques.
- Improving the location- and depth-based model by adding more wells, seismic attributes, or velocity data to the training set.
- Using ML Vsh predictions in reservoir calculations such as net to gross (N/G) and reservoir effective porosity.
- Conducting a regional fault seal analysis for the Brent Group in the northern North Sea area using the ML Vsh predictions.

6. Conclusions

The focus of this study is the design of a fast, reliable and consistent Vsh ML models. This will save time and reduce bias in the analysis of large petrophysical datasets for regional exploration or the development of large fields. It will also improve the input to related geological applications such as reserve calculations and fault seal analysis. The study accomplished this by building a stable ML model that can predict Vsh while minimizing error. The work started by studying and validating normal petrophysical approaches for determining Vsh against core-image data, to choose the best petrophysical approach as label for training the ML algorithms. Then the ML models were built and tested to select the best model based on its performance (in terms of error).

Calculating Vsh using all the available well logs highlighted the error and reliability of the different petrophysical methods, and showed that Vsh derived from nonlinear GR is the best method. Also, the study provides an effective methodology for core images grayscale curves extraction. The idea of extracting core images grayscale pixel values proved to be an efficient Vsh estimator used for validation of the other methods. In future this can be used as label input to train ML models, besides the possibilities of using the grayscale values for other geological applications (e.g., sequence stratigraphy and ML model training).

Building and inspecting the ML models gave a better understanding of how ML models perform Vsh predictions based on different inputs. The XGBoost algorithm showed a high performance in predicting Vsh. Tunning the model hyperparameters and adding meaningful features such as lithology index, significantly increases the accuracy of the model. Also, making the model aware of the wells' location (e.g., clustering) can increase the model accuracy. Formation and group information did not seem to improve the Vsh predictions much. Finally, the Vsh predictive model RMSE based only on location and depth data was not as bad based on the validation set RMSE.

Since this model allows determining Vsh before drilling a well, it could be considered as a turning point in exploration, but it needs further developing.

References

Asquith, G. and Krygowski, D.A., 2004, Basic well log analysis: AAPG 2nd edition of Methods in Exploration Series, 244 p.

Badley, M. E., Price, J. D., Rambech Dahl, C., and Agdestein, T., 1988, The structural evolution of the northern Viking Graben and its bearing upon extensional modes of basin formation: Journal of the Geological Society, v. 145, no. 3, p. 455-472.

Bermingham, M. L., Pong-Wong, R., Spiliopoulou, A., Hayward, C., Rudan, I., Campbell, H., Wright, A. F., Wilson, J. F., Agakov, F., Navarro, P., Haley, C. S., 2015, Application of high-dimensional feature selection: evaluation for genomic prediction in man: Scientific Reports, v, 5.

Bergstra, J., and Bengio, Y., 2012, Random search for hyper-parameter optimization., Journal of machine learning research, v. 13, p. 281–305.

Biau, G., 2012, Analysis of a Random Forests Model: Journal of Machine Learning Research, v. 13, p. 1063–1095.

Bigdata-madesimple website, 2018, Machine learning explained: Understanding supervised, unsupervised, and reinforcement learning. Retrieved from: https://bigdata-madesimple.com/machine-learning-explained-understanding-supervised-unsupervised-and-reinforcement-learning/

Bjørlykke, K., 2015, Petroleum Geoscience - From Sedimentary Environments to Rock Physics: Springer Verlag, p. 662.

Bowen, J. M., 1975, The Brent oil field: WOODLAND, A. W., ed. Petroleum and the Continental Shelf of North-West Europe, v. 1: Geology: Applied Science Publishers, London, p. 353-362.

Breiman, L., 2001, Random forests: Machine Learning, v. 45, p. 5-32.

Brennand, T. P., Van Hoorn, B., and James, K. H., 1990, Historical review of North Sea exploration: GLENNIE, K. W. (ed.) Introduction to the Petroleum Geology of the North Sea: Blackwell, Oxford, p. 1-33.

Chen., T, and Guestrin, C., 2016, XGBoost: A Scalable Tree Boosting System: Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, New York, USA, p. 785–794.

Christiansson, P., Faleide, J. I., and Berge, A. M., 2000, Crustal structure in the northern North Sea: an integrated geophysical study: Geological Society, London, Special Publications, v. 167, no. 1, p. 15.

Clarke, B., Fokoue, E., and Zhang, H. H., 2009, Principles and Theory for Data Mining and Machine Learning: Springer New York.

Clavier, C. W. R. Hoyle, and D. Meunier, 1971, Quantitative interpretation of thermal neutron decay time logs: Part 1. Fundamentals and techniques: Journal of Petroleum Technology, v. 23, p. 743– 755.

Cover, T.M., and Hart, P.E., 1967, Nearest neighbor pattern classification: IEEE Transactions on Information Theory, v. 13, no. 1, p. 21–27.

David, S. O., Rodolfo, S. B., Jonathan, S. O., Pasquel, Oliver, and Duarry A., 2015, A Universal Equation to Calculate Shale Volume for Shaly-Sands and Carbonate Reservoirs: SPE Latin American and Caribbean Petroleum Engineering Conference.

Ethem, A., 2010, Introduction to Machine Learning: MIT Press. P 584

Færseth, R. B., 1996, Interaction of Permo-Triassic and Jurassic extensional fault-blocks during the development of the northern North Sea: Journal of the Geological Society, v. 153, no. 6, p. 931-944.

Fossen, H., and Rotevatn, A., 2016, Fault linkage and relay structures in extensional settingsA review: Earth-Science Reviews, v. 154, no. C, p. 14-28.

Halland, E.K., Mujezinovic, J. and Riis, F., 2014, CO2 Storage Atlas: Norwegian Continental Shelf: Norwegian Petroleum Directorate. npd.no/en/Publications/Reports/Compiled-CO2-atlas.

Hanke, M., Halchenko, Y. O., Sederberg, P. B., Hanson, S. J., Haxby, J. V., and Pollmann, S., 2009, PyMVPA: a python toolbox for multivariate pattern analysis of FMRI data: Neuroinformatics v.7, p. 37–53.

Harrell, F.E., 2001, Regression Modeling Strategies: Springer New York, v. 18, p. 87–103.

Hashmy, K., Alberty, M., 1992, Development Geology Reference Manual: Difficult Lithologies Wireline Methods, p.186-191.

Helland-Hansen, W., Ashton, M., Lomo, L., and Steel, R., 1992, Advance and retreat of the Brent delta: Recent contributions to the depositional model: Geological Society, London, Special Publications, v. 61, p. 109-127.

Hutter, F., Hoos, H., and Leyton-Brown, K., 2014, An efficient approach for assessing hyperparameter importance: Proceedings of the 31st International Conference on Machine Learning, PMLR v. 32, no. 1, p. 754-762.

Interactive petrophysics (IP) Help, 2018, Clay volume analysis: Lioyd's Register.

Jiang, S., Pang, G., Wu, M., and Kuang, L., 2012, An improved K-nearest-neighbor algorithm for text categorization: Expert Systems with Applications, v. 39, p. 1503-1509

Jones, T. G. J., Hughes, T. L., and Tomkins P., 1989, The ion content and mineralogy of a North Sea Cretaceous shale formation: Clay Minerals, v. 24, P. 393–410.

Kamel, M. H., and Mabrouk, W. M., 2003, Estimation of shale volume using a combination of the three porosity logs: J. Petrol. Sci. Eng., v. 40, p.145–157.

Kirasich, K., Smith, T., and Sadler, B., 2018, Random Forest vs logistic regression: binary classification for heterogeneous datasets: SMU Data Science Review, v. 1.

Koch, p., Wujek, B., Golovidov O., and Gardner, S., 2017, Automated hyperparameter tuning for effective machine learning: SAS Institute Inc.

Kohavi, R. and Provost, F., 1998, Glossary of terms. Machine Learning—Special Issue on Applications of Machine Learning and the Knowledge Discovery Process: Machine Learning, v. 30, p. 271-274.

Larionov, V. V., 1969, Borehole Radiometry: Moscow, U.S.S.R., Nedra.

Mabrouk, W. M., and Kamel, M. H., 2011, Shale volume determination using sonic, density and neutron data: Exploration Geophysics, v. 42, no. 2, p. 155-158.

McKeen, S. A., Wilczak, J., Grell, G., Djalalova, I., Peckham, S., Hsie, E., Gong, W., Bouchet, V., Menard, S., Moffet, R., McHenry, J., McQueen, J., Tang, Y., Carmichael, G. R., Pagowski, M., Chan, A., Dye, T., Frost, G., Lee, P., and Mathur, R., 2005, Assessment of an ensemble of seven real time ozone forecasts over eastern North America during the summer of 2004: J. Geophysical Research, p. 16.

Mitchell, T., 1997, Machine Learning: McGraw-Hill Higher Education, New York., p 432.

Morton, A. C., Haszeldine, R. S., Giles, M. R. and Brown, S., 1992, Geology of the Brent Group: Geological Society Special Publication no. 61, p. 506 pp.

Müller, A., and Guido, S., 2016, Introduction to Machine Learning with Python: O'Reilly.

Naeini, E. Z., Green, S., and Rauch-Davies, M., 2019, An integrated deep learning solution for petrophysics, pore pressure, and geomechanics property prediction: Lead. Edge v. 38, p. 53-59.

Odinsen, T., Christiansson, P., Gabrielsen, R. H., Faleide, J. I., and Berge, A. M., 2000, The geometries and deep structure of the northern North Sea rift system: Geological Society, London, Special Publications, v. 167, no. 1, p. 41-57.

Picard, M. D., 1953, A Misnomer as Used in Uinta Basin, Utah1: AAPG Bulletin, v. 37, no. 5, p. 1075–1077.

Poupon, A., and Gaymard, R., 1970, The evaluation of clay content from logs: SPWLA 11th Annual Logging Symposium, Los Angeles.

Quantinsti website, 2020, Introduction to XGBoost in Python. Retrieved from: https://blog.quantinsti.com/xgboost-python/

Rider, M., 2002, The geological interpretation of well logs: Rider-French consulting Ltd., Sutherland, Scotland, p. 280.

Ryseth, A., 2000, Differential subsidence in the Ness Formation (Bajocian), Oseberg area, northern North Sea: facies variation, accommodation space development and sequence stratigraphy in a deltaic distributary system.: Norsk Geologisk Tidsskrift, v. 80, p. 9-26.

Saarela, M., Jauhiainen, S., 2021, Comparison of feature importance measures as explanations for classification models: SN Appl. Sci. v. 3, p. 272.

Savage, N. H., Agnew, P., Davis, L. S., Ordóñez, C., Thorpe, R., Johnson, C. E., O'Connor, F. M., and Dalvi, M., 2013, Air quality modelling using the Met Office Unified Model (AQUM OS24-26): model description and initial evaluation: Geosci. Model Dev., v. 6, p. 353-372.

Schratz, P., Muenchow, J., Iturritxa, E., Richter, J., Brenning, A., 2019, Hyperparameter tuning and performance assessment of statistical and machine-learning algorithms using spatial data: Ecological Modelling, v. 406, p. 109-120.

Sedgwick, P., 2012, Pearson's correlation coefficient: BMJ, p. 2.

Shaw, D. B., and Weaver, C. E., 1965, The mineralogical composition of shales.: J. Sedimentary Petrology, v.35, P. 213-222.

Soto, R. B., Duarry, A., Cinta, M., and Freddy, R., 2010, The Correct Shale Volume Characterization Increases Hydrocarbon Reserve: Case Study of Cretaceous Formation, Lake Maracaibo, Venezuela: SPE international Peru, p. 9.

Steiber, S. J., 1970, Pulsed neutron capture log evaluation in the Louisiana Gulf Coast: Society of Petroleum Engineers, 45th Annual Meeting, paper SPE- 2961.

Syed, F., I., AlShamsi, A., Dahaghi, A., K., Neghabhan, S., 2020, Machine learning techniques to model geomechanics and petrophysical properties of shale reservoirs – A systematic literature review: Petroleum.

Vollset, J., and Doré, A.G., 1984, A revised Triassic and Jurassic lithostratigraphic nomenclature for the Norwegian North Sea, NPD-Bulletin 3, p. 53.

Wang, X., Yang, S., Zhao, Y., and Wang, Y., 2018, Lithology identification using an optimized KNN clustering method based on entropy-weighed cosine distance in Mesozoic strata of Gaoqing field, Jiyang depression: Journal of Petroleum Science and Engineering, v. 166 , p. 157-174.

Western Atlas International, Inc., 1995, Log Interpretation Charts, Houston, Texas, Western Atlas, 300 p.

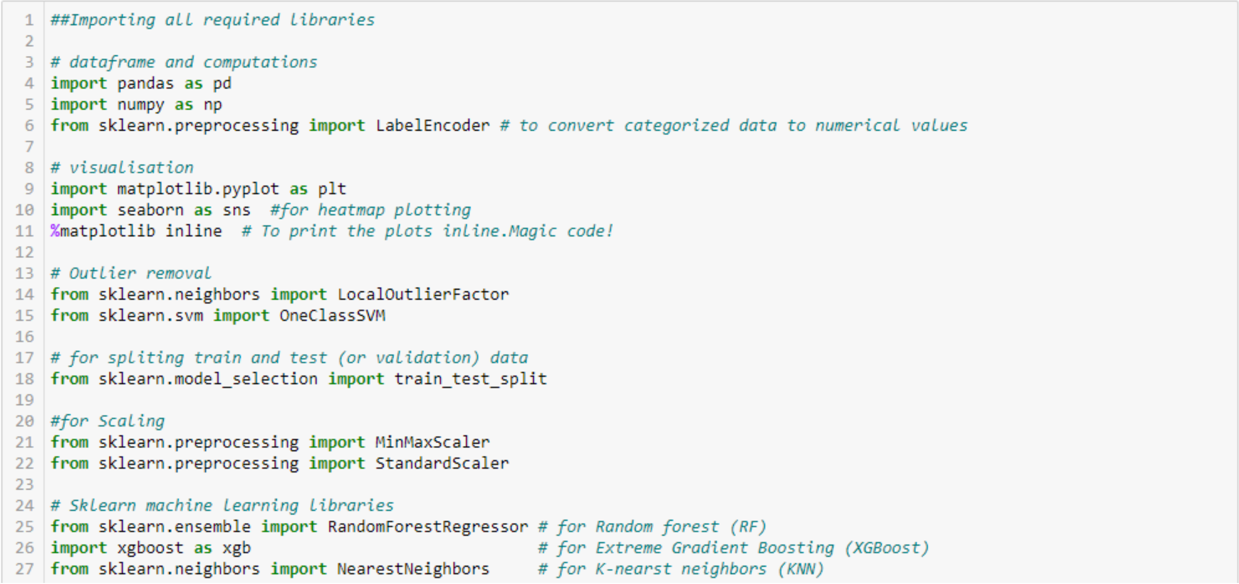
Worthington, P.E., 1985, The Evolution of Shaly Sand Concepts in Reservoir Evaluation: The Log Analyst, v. 26, no. 1, p. 23-40.

Yielding, G., and Roberts, A., 1992, Footwall uplift during normal faulting - implications for structural geometries in the North Sea, in Larsen, R. M., Brekke, H., Larsen, B. T., and Talleraas, E., eds., Structural and Tectonic Modelling and its Application to Petroleum Geology, v. 1: Elsevier Amsterdam, p. 289-304.

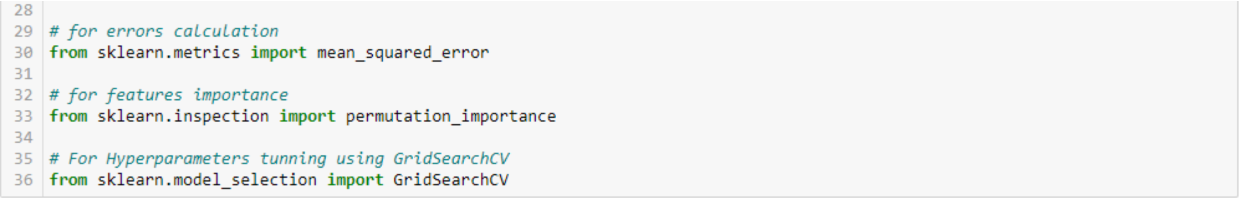
Ying, X., 2019, An overview of overfitting and its solutions: Journal of Physics, conference series, v. 1168, no. 2.

Appendix

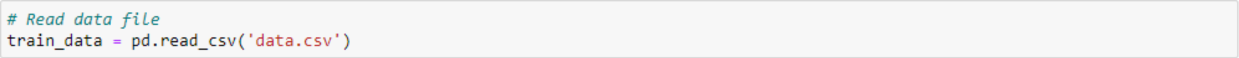
Appendix 1: Shale volume prediction input code in Python



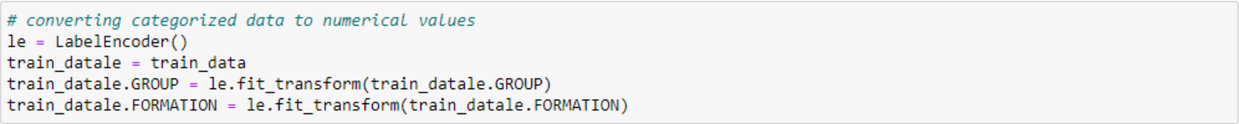
```
##Importing all required libraries

# dataframe and computations
import pandas as pd
import numpy as np
from sklearn.preprocessing import LabelEncoder # to convert categorized data to numerical values

# visualisation
import matplotlib.pyplot as plt
import seaborn as sns  #for heatmap plotting
%matplotlib inline  # To print the plots inline.Magic code!

# Outlier removal
from sklearn.neighbors import LocalOutlierFactor
from sklearn.svm import OneClassSVM

# for spliting train and test (or validation) data
from sklearn.model_selection import train_test_split

#for Scaling
from sklearn.preprocessing import MinMaxScaler
from sklearn.preprocessing import StandardScaler

# Sklearn machine learning libraries
from sklearn.ensemble import RandomForestRegressor # for Random forest (RF)
import xgboost as xgb                              # for Extreme Gradient Boosting (XGBoost)
from sklearn.neighbors import NearestNeighbors     # for K-nearst neighbors (KNN)
```



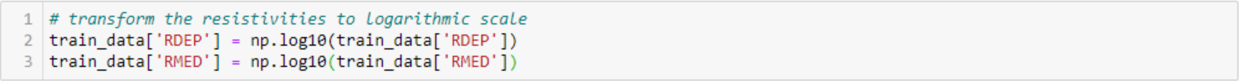
```

# for errors calculation
from sklearn.metrics import mean_squared_error

# for features importance
from sklearn.inspection import permutation_importance

# For Hyperparameters tunning using GridSearchCV
from sklearn.model_selection import GridSearchCV
```



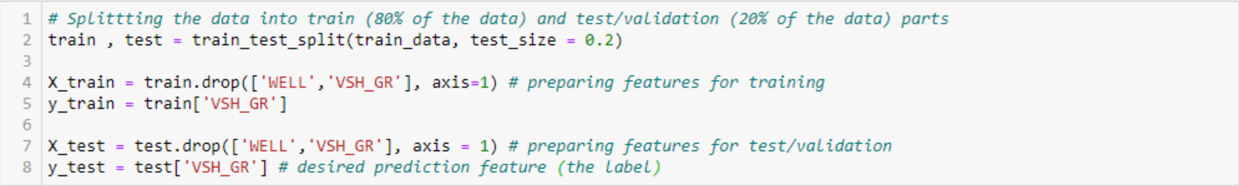
```
# Read data file
train_data = pd.read_csv('data.csv')
```



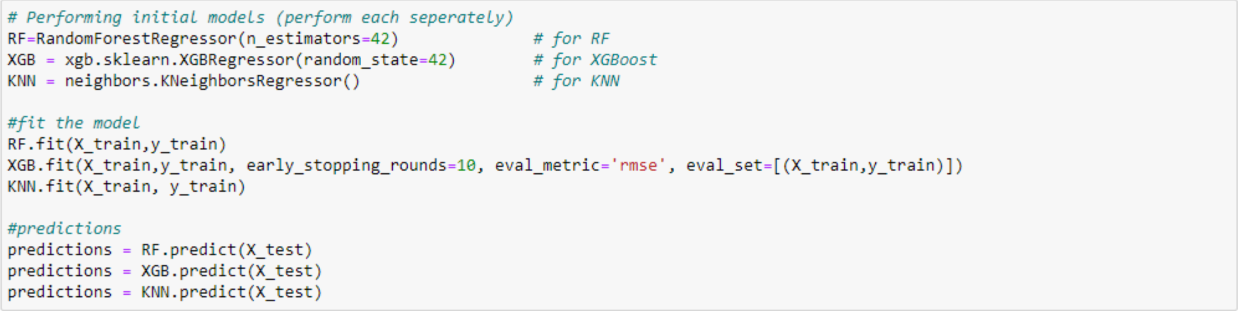
```
# converting categorized data to numerical values
le = LabelEncoder()
train_datale = train_data
train_datale.GROUP = le.fit_transform(train_datale.GROUP)
train_datale.FORMATION = le.fit_transform(train_datale.FORMATION)
```



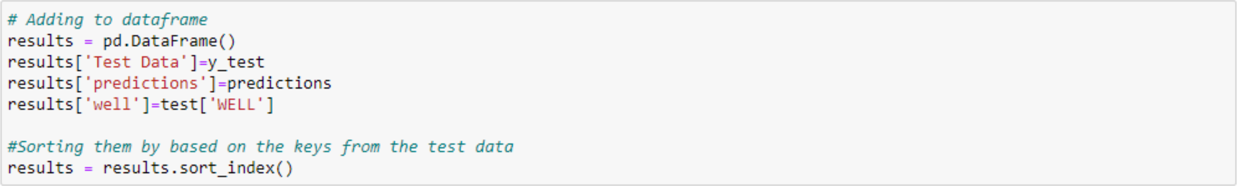
```
# transform the resistivities to logarithmic scale
train_data['RDEP'] = np.log10(train_data['RDEP'])
train_data['RMED'] = np.log10(train_data['RMED'])
```



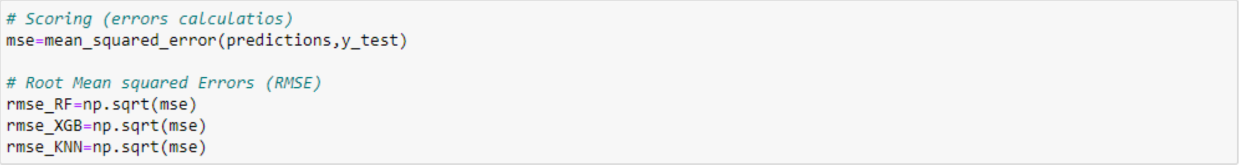
```
# Splittting the data into train (80% of the data) and test/validation (20% of the data) parts
train , test = train_test_split(train_data, test_size = 0.2)

X_train = train.drop(['WELL','VSH_GR'], axis=1) # preparing features for training
y_train = train['VSH_GR']

X_test = test.drop(['WELL','VSH_GR'], axis = 1) # preparing features for test/validation
y_test = test['VSH_GR'] # desired prediction feature (the label)
```



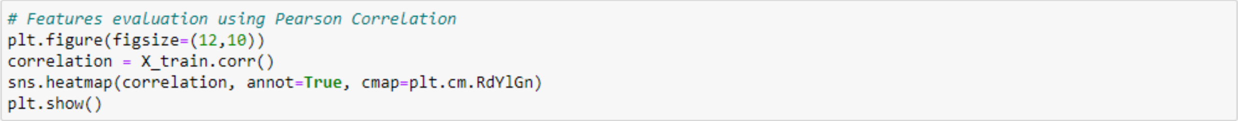
```
# Performing initial models (perform each seperately)
RF=RandomForestRegressor(n_estimators=42)              # for RF
XGB = xgb.sklearn.XGBRegressor(random_state=42)        # for XGBoost
KNN = neighbors.KNeighborsRegressor()                  # for KNN

#fit the model
RF.fit(X_train,y_train)
XGB.fit(X_train,y_train, early_stopping_rounds=10, eval_metric='rmse', eval_set=[(X_train,y_train)])
KNN.fit(X_train, y_train)

#predictions
predictions = RF.predict(X_test)
predictions = XGB.predict(X_test)
predictions = KNN.predict(X_test)
```



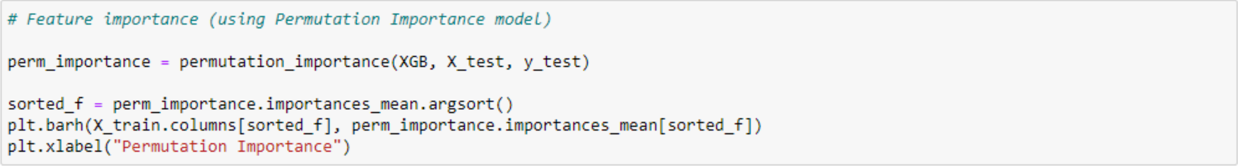
```
# Adding to dataframe
results = pd.DataFrame()
results['Test Data']=y_test
results['predictions']=predictions
results['well']=test['WELL']

#Sorting them by based on the keys from the test data
results = results.sort_index()
```



```
# Scoring (errors calculatios)
mse=mean_squared_error(predictions,y_test)

# Root Mean squared Errors (RMSE)
rmse_RF=np.sqrt(mse)
rmse_XGB=np.sqrt(mse)
rmse_KNN=np.sqrt(mse)
```

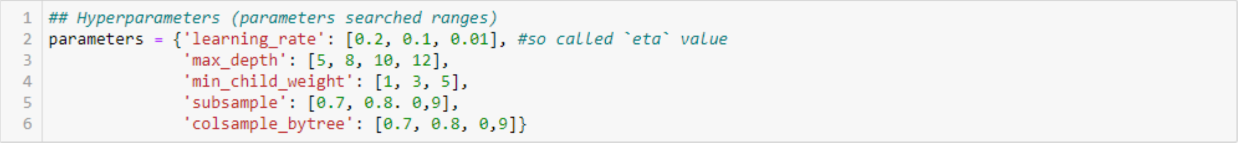


```
# Features evaluation using Pearson Correlation
plt.figure(figsize=(12,10))
correlation = X_train.corr()
sns.heatmap(correlation, annot=True, cmap=plt.cm.RdYlGn)
plt.show()
```

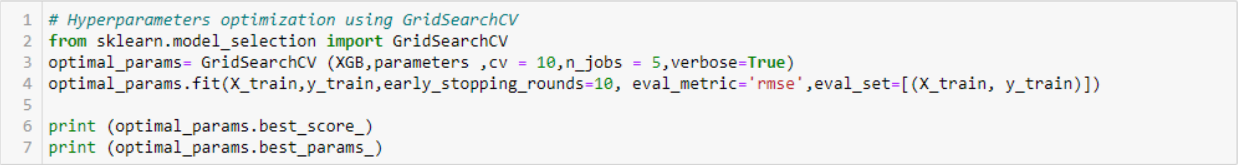


```
# Feature importance (using Permutation Importance model)

perm_importance = permutation_importance(XGB, X_test, y_test)

sorted_f = perm_importance.importances_mean.argsort()
plt.barh(X_train.columns[sorted_f], perm_importance.importances_mean[sorted_f])
plt.xlabel("Permutation Importance")
```

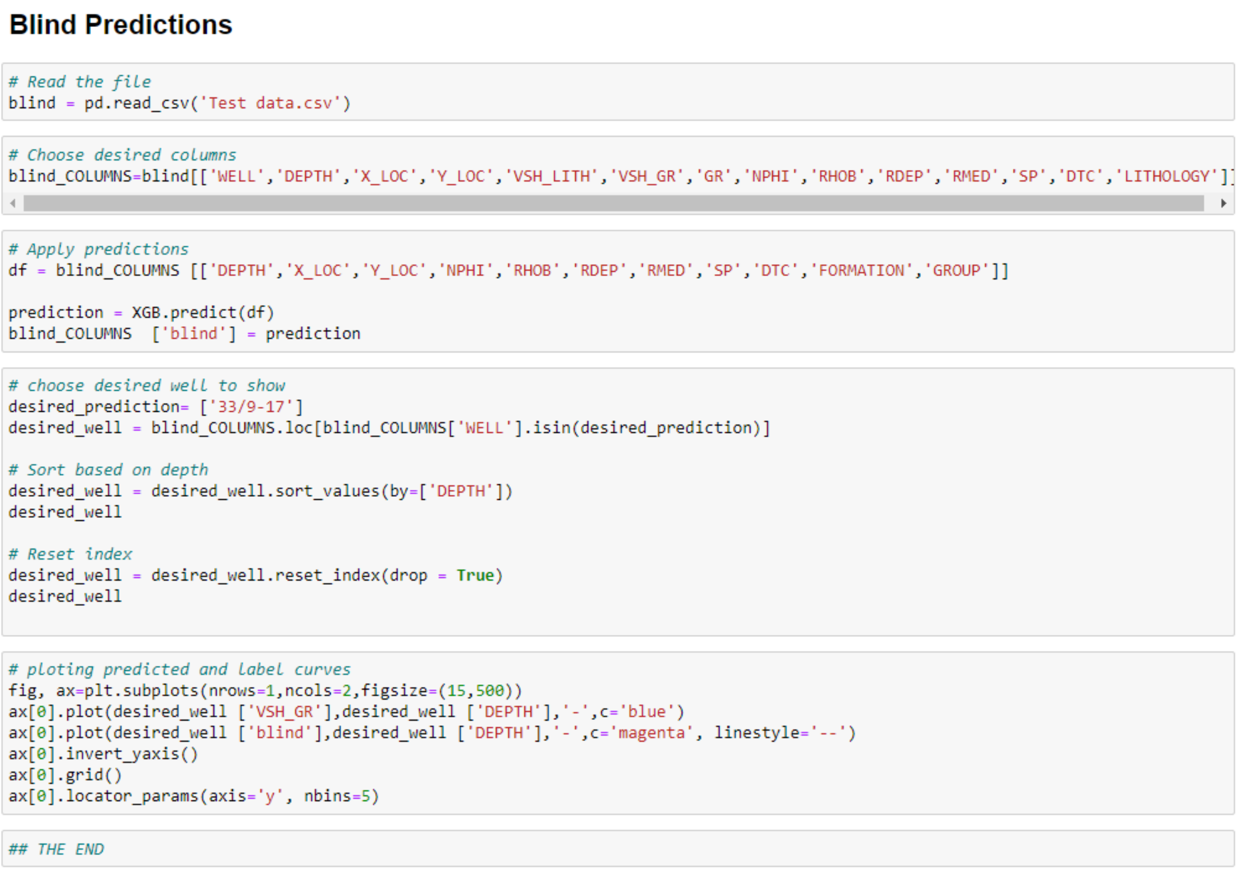
Hyperparameters tunning



```
## Hyperparameters (parameters searched ranges)
parameters = {'learning_rate': [0.2, 0.1, 0.01], #so called `eta` value
              'max_depth': [5, 8, 10, 12],
              'min_child_weight': [1, 3, 5],
              'subsample': [0.7, 0.8. 0,9],
              'colsample_bytree': [0.7, 0.8, 0,9]}
```



```
# Hyperparameters optimization using GridSearchCV
from sklearn.model_selection import GridSearchCV
optimal_params= GridSearchCV (XGB,parameters ,cv = 10,n_jobs = 5,verbose=True)
optimal_params.fit(X_train,y_train,early_stopping_rounds=10, eval_metric='rmse',eval_set=[(X_train, y_train)])

print (optimal_params.best_score_)
print (optimal_params.best_params_)
```



Blind Predictions

```
# Read the file
blind = pd.read_csv('Test data.csv')
```

```
# Choose desired columns
blind_COLUMNS=blind[['WELL','DEPTH','X_LOC','Y_LOC','VSH_LITH','VSH_GR','GR','NPHI','RHOB','RDEP','RMED','SP','DTC','LITHOLOGY']]
```

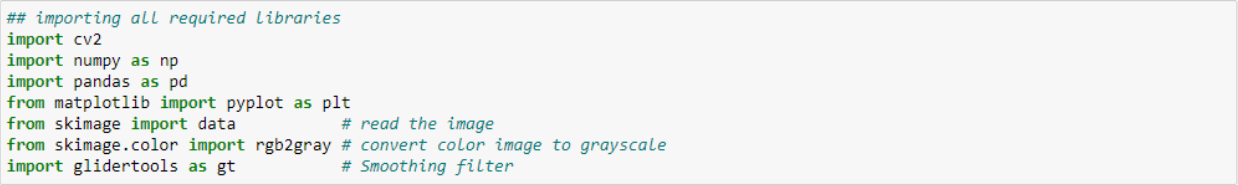
```
# Apply predictions
df = blind_COLUMNS [['DEPTH','X_LOC','Y_LOC','NPHI','RHOB','RDEP','RMED','SP','DTC','FORMATION','GROUP']]

prediction = XGB.predict(df)
blind_COLUMNS  ['blind'] = prediction
```

```
# choose desired well to show
desired_prediction= ['33/9-17']
desired_well = blind_COLUMNS.loc[blind_COLUMNS['WELL'].isin(desired_prediction)]

# Sort based on depth
desired_well = desired_well.sort_values(by=['DEPTH'])
desired_well

# Reset index
desired_well = desired_well.reset_index(drop = True)
desired_well
```

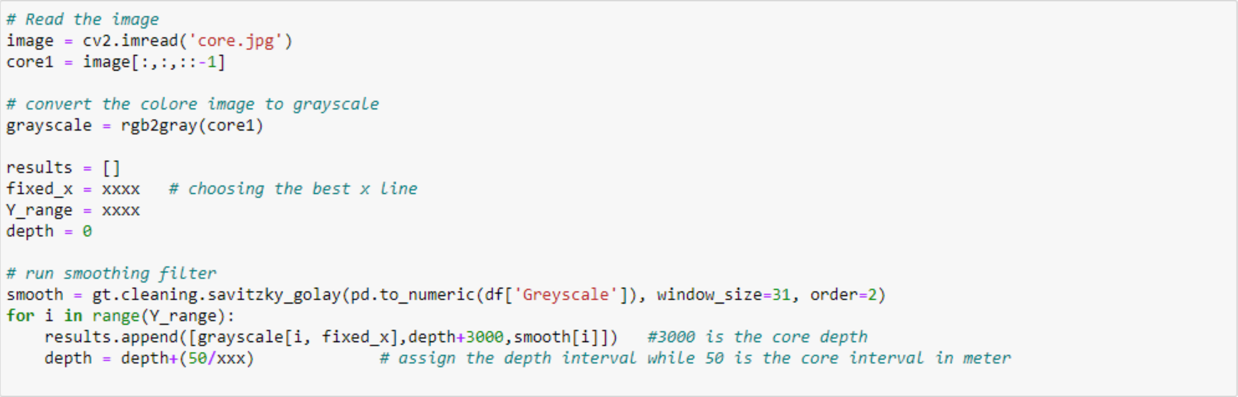
```
# ploting predicted and label curves
fig, ax=plt.subplots(nrows=1,ncols=2,figsize=(15,500))
ax[0].plot(desired_well ['VSH_GR'],desired_well ['DEPTH'],'-',c='blue')
ax[0].plot(desired_well ['blind'],desired_well ['DEPTH'],'-',c='magenta', linestyle='--')
ax[0].invert_yaxis()
ax[0].grid()
ax[0].locator_params(axis='y', nbins=5)
```

```
## THE END
```

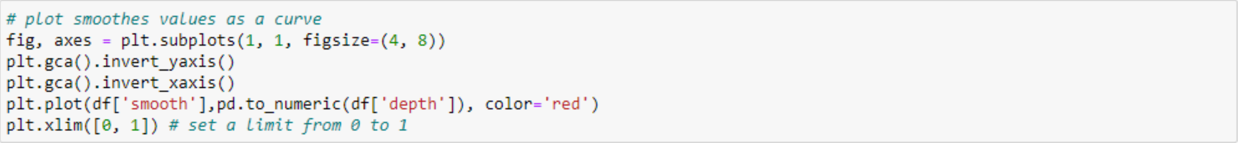
Appendix 2: Core images grayscale pixel values / curves extraction input code in Python



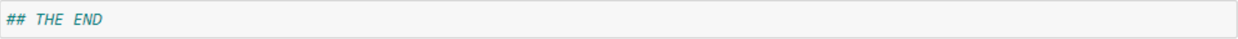
```
## importing all required libraries
import cv2
import numpy as np
import pandas as pd
from matplotlib import pyplot as plt
from skimage import data           # read the image
from skimage.color import rgb2gray # convert color image to grayscale
import glidertools as gt           # Smoothing filter
```



```
# Read the image
image = cv2.imread('core.jpg')
core1 = image[:,:,::-1]

# convert the colore image to grayscale
grayscale = rgb2gray(core1)

results = []
fixed_x = xxxx   # choosing the best x line
Y_range = xxxx
depth = 0

# run smoothing filter
smooth = gt.cleaning.savitzky_golay(pd.to_numeric(df['Greyscale']), window_size=31, order=2)
for i in range(Y_range):
    results.append([grayscale[i, fixed_x],depth+3000,smooth[i]])   #3000 is the core depth
    depth = depth+(50/xxx)             # assign the depth interval while 50 is the core interval in meter
```



```
# plot smoothes values as a curve
fig, axes = plt.subplots(1, 1, figsize=(4, 8))
plt.gca().invert_yaxis()
plt.gca().invert_xaxis()
plt.plot(df['smooth'],pd.to_numeric(df['depth']), color='red')
plt.xlim([0, 1]) # set a limit from 0 to 1
```



```
## THE END
```