**ANL488 FINAL PROJECT REPORT**

Evaluating Clustering Algorithms for Prediction of Rock Type

for Oil and Gas Applications

using United Kingdom Core Data



**Submitted by**

Tan Wen Xuan

**SCHOOL OF BUSINESS**

**Singapore University of Social Sciences**

**Presented to Singapore University of Social Sciences**

**in partial fulfillment of the requirements for the**

**Degree of Bachelor of Science**

**In Business Analytics**

2022

# **Abstract**

In the oil and gas industry, rock typing is important when it comes to finding optimal locations to drill in wells for optimizing extraction. Rock typing studies mostly rely on rock typing indices for rock typing and in newer studies, the usage of rock typing indices was combined with machine learning. However, for rock typing in the oil and gas industry, solely using unsupervised machine learning algorithms for rock typing as well as comparing the algorithms against a traditional method of rock typing has not been widely studied which this study explores. This study compares the performance of 5 unsupervised machine learning models against our version of the traditional rock typing methods. The data used for model training is the UK core data with 2000 data points for modeling efficiency, containing the porosity and permeability dataset features. The own method was derived using the IMLR clustering technique and is used as a benchmark for comparison. The unsupervised machine learning algorithms used are KMeans, Self-Organising Map (SOM) + KMeans, Density-Based Spectral Clustering of Application with Noise (DBSCAN), Balanced Iterative Reducing and Clustering using Hierarchies (BIRCH) and Gaussian Mixture Models (GMM). Optimization by changing parameters is attempted to boost the performance of all models. The models were compared using 3 different approaches. The first approach is by comparing the porosity (x-axis) vs Log-scaled permeability (y-axis) scatterplot showing the cluster distribution as well as the tables showing cluster averages of each method’s porosity and permeability averages. The second approach is by performing permeability prediction for all the models and plotting the residual plots for comparison. The last approach is by comparing the metric values of the unsupervised machine learning models and the metrics chosen are the Silhouette Coefficient, Calinski-Harabaz Index, and Davis Bouldin Index. From the 3 approaches for evaluating the performance of the models, apart from the benchmark method, KMeans and BIRCH have the best performance on the dataset and the results performance is comparable to the benchmark method.

**TABLE OF CONTENTS**

[**Abstract** 1](#_Toc118665877)

[**Chapter 1: Introduction** 4](#_Toc118665878)

[**Chapter 2: Literature Review** 7](#_Toc118665879)

[**Chapter 2.1: Traditional Rock Typing** 7](#_Toc118665880)

[**Chapter 2.2: Rock Typing utilizing machine learning** 11](#_Toc118665881)

[**Chapter 2.3: Rock typing – machine learning approach** 13](#_Toc118665882)

[**Chapter 2.4: Comparative study of clustering algorithms** 13](#_Toc118665883)

[**Chapter 2.5: Literature Conclusion** 14](#_Toc118665884)

[**Chapter 3: Data Understanding and Initial Preparation** 15](#_Toc118665885)

[**Chapter 3.1: Datasets Overview** 15](#_Toc118665886)

[**Chapter 3.2: Datasets Preparation** 16](#_Toc118665887)

[**Chapter 3.2.1: UK** 17](#_Toc118665888)

[**Chapter 3.3: Data Understanding** 18](#_Toc118665889)

[**Chapter 3.3.1: UK** 18](#_Toc118665890)

[**Chapter 3.4: Data Preparation** 21](#_Toc118665891)

[**Chapter 4: Modelling and Evaluation** 22](#_Toc118665892)

[**Chapter 4.1: Methods** 22](#_Toc118665893)

[**Chapter 4.1.1: Benchmark Method** 26](#_Toc118665894)

[**Chapter 4.1.2: K-Means** 28](#_Toc118665895)

[**Chapter 4.1.3: SOM+K-Means** 29](#_Toc118665896)

[**Chapter 4.1.4: Density-Based Spatial Clustering of Application with Noise (DBSCAN)** 30](#_Toc118665897)

[**Chapter 4.1.5: Balanced Iterative Reducing and Clustering using Hierarchies (BIRCH)** 32](#_Toc118665898)

[**Chapter 4.1.6: Gaussian Mixture Models (GMMs)** 33](#_Toc118665899)

[**Chapter 4.2: Results and Discussion** 34](#_Toc118665900)

[**Chapter 4.2.1: Comparing Porosity vs Log Scaled Porosity Plots and Cluster Averages** 35](#_Toc118665901)

[**Chapter 4.2.2: Comparing Residual Models** 38](#_Toc118665902)

[**Chapter 4.2.3: Comparing Unsupervised Machine Learning Algorithm Using Metrics** 40](#_Toc118665903)

[**Chapter 5: Conclusion and Future Works** 41](#_Toc118665904)

[**References** 44](#_Toc118665905)

# **Chapter 1: Introduction**

Rock typing is the characterization of reservoir rocks into similar groups that are petrophysical similar (Hollis et. al., 2020). Much research has been done on rock typing reservoir rocks of reservoirs for determining optimal rock types for drilling. Rock Typing studies typically consist of the application of indices like FZI or FZI\* onto data that uses additional data alongside core data like well logs data. For example, the study by Mirzaei-Paiaman et. al. (2020) presented a new framework (GHE) and in their study, the data used was using the routine core analysis-RCAL data.

One of the main sources of oil and gas is from drilling wells in sedimentary basins which are essentially layers of rock that hold hydrocarbons and crude oil in tiny spaces between the sedimentary rocks (*U.S. Energy Information Administration*, n.d.). Sedimentary basins are areas of the earth’s crust that have been underlaid under multiple thick layers of sedimentary rocks like sandstone or carbonates, whereby the thickness of the rock exceeds a certain threshold (*Sedimentary Basin*, n.d.). Although there is not a clear number for that thickness, many geologists would accept a 100 km length and width of 10 km to be a suitable threshold for an area with sediments to be a sedimentary basin (*Sedimentary Basin*, n.d.).

Oil and gas are then produced by drilling a well in the sedimentary basins. The primary recovery process is as follows, it would firstly be the fitting of tubes and casing of pipes for the transportation of hydrocarbons (Dutta, 2022). Once the well is tapped, the hydrocarbons are extracted and transferred to the surface for processing (Dutta, 2022). From there, filtering of the mixture of water, oil, and gas to their separate components is next, and then finally the oil and gas are treated and purified after removing the water (Dutta, 2022).

Before the production phase of oil and gas, when the well is first drilled, rock samples are extracted to determine the flow capacities and storage. In the current context, flow capacity refers to permeability which is a characteristic that describes the flow capacity of rock while storage refers to porosity, which is the characteristic describing the storage amount of the rock that holds the oil/gas (Geomore, n.d.). The two are important characteristics that are used for determining rock types.

For determining optimal drilling spots, we must first rock type to find clusters that are medium-high in porosity and permeability. Rock typing is the assignment of reservoir rocks to distinct geological facies (Tavakoli, 2018). Rock typing would give us knowledge of the different rock types in the well, aiding in the planning of drilling based on “optimum” rock type through the generalization of rock type results using the samples.

In the area of rock typing, the studies mostly use rock typing indices on a variety of data to rock type. There have been studies where machine learning has been applied in their methodology to rock type while using indices. Some studies even explore rock typing using machine learning without indices. However, such studies mostly use supervised machine learning and a mixture of data which may be costly to acquire. Comparing unsupervised learning methods and the traditional method on only core data is something that has not been researched extensively and hence there is value in focusing on the area.

In this project, we would be using different unsupervised machine learning clustering methods to perform rock typing on a publicly available dataset from the UK region and then evaluate which model is the best. The algorithms selected are K-Means, Self-Organizing Map with K-Means, DBSCAN, BIRCH, and Gaussian Mixture models. We would also do a traditional method of rock typing using the FZI Index with our spin to the method. The project objective would be to evaluate which method would have the best performance in rock typing on core data. The data that were available for this project were core data from different regions which include Africa, Australia, Brazil, Myanmar, Pakistan, Papua New Guinea (PNG), China, the North Sea, the Philippines, the UK, and the USA. Data preparation was performed on all datasets but only the UK dataset had been used for training due to time limitations. With UK data, we will be performing rock typing using multiple clustering models and a traditional rock typing method and after that evaluation of models will be done.

The proposal report format will be as follows: Introduction, Literature Review, Data Understanding, and Preparation, Modelling and Evaluation, and Conclusion and Future Works.

# **Chapter 2: Literature Review**

## **Chapter 2.1: Traditional Rock Typing**

Previous and prevalent approaches to rock typing revolve around the usage of rock typing indices on core, well logs, and other data (Special Core Analysis Laboratory (SCAL  
)). Common indices that are used are the FZI, FZI\*, and FZIM (Mohammadian, 2022). In most of the literature available on rock typing, rock typing indices are used in their methodology alongside different methods each literature proposed. The following are some examples of studies using rock typing indices as part of their approach for rock typing.

Faramarzi-Palanga and Mirzaei-Paiaman (2021) used rock type indices to determine flow for dynamic rock typing whereas, in dynamic rock typing, fluid flow traits are critical for dynamic rock typing. They used the FZI, FZI\*, FZIM, and MFZI indices which are slight variations of one another on SCAL data (Faramarzi-Palanga & Mirzaei-Paiaman, 2021). The performance of the indices was then compared using the (True Effective Mobility) TEM technique, which the TEM technique is a criterion for determining dynamic rock types (Faramarzi-Palanga & Mirzaei-Paiaman, 2021). The results of the study showed that the FZI\* performance was superior to the other three indices for both water and oil saturation and that FZI was better performing than MFZI and FZIM (Faramarzi-Palanga & Mirzaei-Paiaman, 2021). The study concluded that the additional parameters of the different indices can be complex, but it does not necessarily equate to superior performance or even improve applicability (Faramarzi-Palanga & Mirzaei-Paiaman, 2021). The formulas applied are as follows where k represents permeability and represents porosity:

**FZI**

**FZI\***

**FZIM**

Where m is the cementation factor

**MFZI**

Where is irreducible water saturation

Shahat et. al. (2021) proposed a new rock typing approach using well-log data. In their study, they proposed the Resistivity Zone Index (RZI) used on well logs data for enhancement in rock typing. The log data was from 21 wells with over 1000 core samples. The formula is as follows:

**RFZI**

Where:

is resistivity index

is true formation resistivity

is formation resistivity when water saturation is 100%

is formation water resistivity

is surface grain volume

k is permeability

is shape factor

The performance of the RZI approach was compared to the FZI technique using the same data. The conclusion was that the technique had high accuracy like the FZI technique with the values between 0.84 to 0.97 for the eight electrical flow units (EFU) attained by grouping RZI values that are similar within a certain range which the concept is like the HFU from the FZI method.

Soleymanzadeh et. al. (2019) explores changes to rock typing using the FZI\*, FZI, and Winland method when pressure is varied. The FZI\* and FZI methods have been covered in previous studies, the Winland method is as follows:

**Winland**

where:

is uncorrected air permeability

is porosity

r35 is the pore throat radius with 35% mercury saturation

The r35 can be used to differentiate rock types where rock samples with r35 similarity can be considered a rock type. The permeability and porosity values of the core samples were measured for five various pressure and the three indices were applied using those values. The goal of the study was to study how pressure affects rock typing and with the application of indices being the method of rock typing.

In the above-mentioned studies mentioned, we can see that the rock typing studies included rock typing indices as part of their methodology for their studies. The indices require different parameters to be available for the calculation of the index for rock typing. This is true for many rock typing studies previously done. This is a limitation due to differences between datasets like core and well data or even between two different wells. For a given index, it may not apply to different datasets if a particular required parameter is not available. The inflexibility of applying a rock typing approach to a different dataset is a disadvantage for rock trying.

## **Chapter 2.2: Rock Typing utilizing machine learning**

Rock typing studies in recent years have explored rock typing with machine learning models, both supervised and unsupervised algorithms. It is not very well explored, and those studies mostly focus on supervised models. In the oil and gas industry, machine learning research is more focused on anomaly detection, production, energy efficiency, risk, and failure detection. There are relatively fewer studies of applications of machine learning for rock typing on the core. However, there are still some studies that integrate machine learning onto traditional rock typing methods for the exploration of improvement in methodologies. The usage of machine learning can bring about high accuracy of rock typing whiles bringing flexibility as algorithms can easily be used for different data as compared to rock typing using rock type indices. The following studies show the use of machine learning for rock typing.

In Mohammadian et. al. (2022), rock typing was done using a modified rock typing index (FZIM\*) together with supervised and unsupervised machine learning algorithms. The data used was the core data consisting of four properties which are permeability, porosity, connate water saturation, and pore throat radius with 35% mercury injection (Mohammadian et. al., 2022). It accounts for connate water saturation and port throat radius with 35% mercury injection, unlike the normal FZI which only depends on porosity and permeability. The FZM\* is a variant of the FZI index. In the study, the data was used to train a supervised algorithm, the Extreme Gradient Boosting (XGB) with FZI index values which were calculated from the data set as a target to get the FZIM\* as the output. The FZIM\* was then used with k-means clustering to determine the ideal number of rock types based on the dataset. The study talks about four deployments and one of which was on rock typing. For the rock typing deployment, the performance of the method was compared to two other approaches of rock typing using indices, the FZI and FZI\*. The proposed method of combining machine learning for rock typing in this study had a better performance than the other two methods that were used to compare. From the study, we can see the inclusion of machine learning for rock typing in this study has indeed brought about better rock typing accuracy. However, the limitation of this approach is still the complexity of using indices-based approaches due to the harsh requirements of data needed.

Zakyan et. al. (2022) proposed an enhanced method of clay typing in their study. They performed rock typing with log data using an artificial neural network (ANN) algorithm as part of the proposed enhanced method of clay typing. The artificial neural network was used in their approach integrated with other non-machine learning methods. The result of their study was the development of the enhanced clay typing approach. A limitation of this study is that the method can only be used for log data.

In both the studies above, machine learning has been integrated and applied with other methods that require certain data to be available for their approaches. The limitation of needing certain data makes rock typing for those methods complex as the methods would only apply to specific kinds of data. Machine learning models in the studies were utilized to bring improvements to rock typing altogether.

Many rock typing studies use rock typing indices for rock typing, which can be considered the traditional and prevalent approach in rock typing. In some newer studies, machine learning has been integrated with rock typing indices for improvements to rock typing. The use of indices is very complex due to the many variations of indices like the FZI, FZI\*, FZIM, etc., and the restrictions on the type of data needed. Rock typing solely on machine learning has not been widely studied and they may have good performance of accuracy even if rock typing indices were not used. The approach would be more flexible in terms of data requirements, and this may be beneficial and more efficient for rock typing.

## **Chapter 2.3: Rock typing – machine learning approach**

In Mohamed et. al. (2019), rock typing was done using supervised and unsupervised machine learning algorithms. The well-logs data was split to train and test set before training of machine learning models. A total of five algorithms were used in the study for comparison and evaluation of each method for rock typing. The methods used are KNN, SVM, Random Forest, Neural Network, and K-means, three supervised and two unsupervised algorithms. Grid search was applied for the optimization of parameters for the models and evaluations are based on precision, F1, recall, and accuracy. The result of the study was that the supervised algorithms which are KNN, SVM, and Random Forest had better performance than the neural network and lastly, the unsupervised algorithm k-means had the poorest results, where all the algorithms had the hyperparameters optimized.

In the study, more supervised algorithms were used than unsupervised algorithms. The advantage of using machine learning to rock type without the need for rock typing indices is that the models can easily be applied to any dataset. For supervised algorithms, the rock type must be known before being able to train such models which can be a challenge as the rock type may not necessarily be available. Unsupervised learning however does not require the dataset to know the rock type. Although the unsupervised method had poorer performance, they do not need rock type to be known before training and thus have greater flexibility and applicability.

## **Chapter 2.4: Comparative study of clustering algorithms**

Rodriguez et. al. (2019) compared 9 popular clustering algorithms in their study with a systematic approach using artificial datasets with varied properties, like the number of classes and number of features. Comparisons of model performances were done through the usage of three methodologies, which are (1) performances of models with default parameters (2) one-dimension analysis, and (3) multi-dimension analysis. The first method is training the models with no adjustment to their parameters. The second method is training models with one parameter varied and the last method is a simultaneous adjustment to all parameters. The result of the study was the comparison of the performance of all 9 clustering algorithms of the three methodologies.

The similarity between this study and our own would be the application of multiple clustering algorithms on a set of data. In this study, they took on three approaches to evaluate the nine algorithms’ performance in each approach. For our study, our focus would be on optimizing the performance of each algorithm while variating algorithm parameters, which was the direction of this study. We will not however be using different approaches to compare performance on separate approaches.

## **Chapter 2.5: Literature Conclusion**

As seen from the literature reviews, rock typing mostly has been using rock typing indices to rock type. In several of the studies, machine learning has been introduced as part of their methodology to enhance rock typing. Rock typing with indices is limited due to the requirements of data for different indices which rock typing using machine learning can solve. In a study mentioned above, rock typing was done using five machine learning algorithms. In that study, the supervised algorithms had better performances than unsupervised algorithms. Supervised algorithms require the dataset to be labeled before training. In the context of rock typing, it would mean that the dataset would have to have the rock types for the data to be able to use supervised machine learning. This can be inefficient and impractical when it comes to rock typing due to the many differences in different data and types of data.

The study of rock typing using clustering approaches has not been studied enough and should be explored further as unsupervised machine learning does not require datasets to know the rock types of the data. In our study, we will be evaluating different clustering algorithms on core data for different regions.

# **Chapter 3: Data Understanding and Initial Preparation**

## **Chapter 3.1: Datasets Overview**

In our study, we are using a publicly available UK core dataset to train on unsupervised algorithms and a traditional rock typing method and then compare. The data understanding and initial preparation included the core data from 11 regions in preparation for possible use for training but only the UK dataset was used in the end.

The datasets will have to first be prepared as most of the data either comes in separate files with many columns or multiple sheets. We merged the separate sheets and files whenever applicable and did an initial selection of columns based on the total number of empty values and the meaning of each column. We then remove columns that are redundant and would most likely not be useful. This step would combine the many separate files and sheets whichever is applicable until we have eleven separate datasets representing the eleven regions. We then did exploratory data analysis on the datasets to study and followed up with cleaning and preparation of the dataset.

Exploratory data analysis, cleaning, and secondary preparation will take place concurrently to ensure the data is ready for modeling. While modeling, secondary preparation is constantly revised to suit the changing requirements.

## **Chapter 3.2: Datasets Preparation**

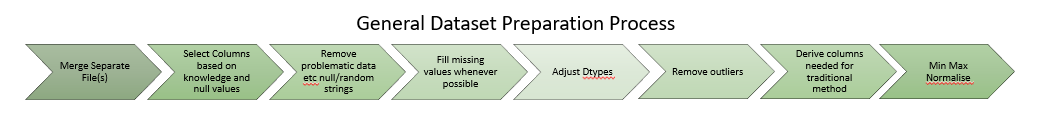
A total of eleven regions’ core data is prepared and retrieved from the following sources: [(Data Underground, n.d.), (GeoProvider, n.d.), (UK National Data Repository, n.d.)]. The regions are Africa, Australia, Brazil, Myanmar, the North Sea, Pakistan, Papua New Guinea (PNG), the UK, the USA, China, and the Philippines. The modeling is trained with the UK core dataset prepared in this step of the study.

Before data understanding, merging of the dataset and selection of appropriate columns must be done due to the number of files there are and, in some datasets, the number of columns there are. Some of which, these files number about 2000 files, and some datasets have over 200 columns. Moreover, the majority of the data column and rows in most files is empty and is not useful. Hence it is important to refine the dataset before moving on to data understanding and cleaning.

The steps for the 11 datasets are mostly similar with slight differences depending on the number of files, columns, and data structure each of the files for each region contains. The general data preparation process is shown below.

**Fig 1**

General Dataset Preparation Process Flowchart



In the preliminary preparation, the focus is on the merging of separate files and sheets as well as the selection of initial selection columns based on knowledge and the number of null values each column had.

A brief walk-through of the preliminary preparation will be done using the UK region. For the rest of the regions, the process is similar.

### **Chapter 3.2.1: UK**

For the UK dataset, there were close to two thousand files that needs to be merged. The format of the column was similar in the separate files which were confirmed through random sampling. To merge the separate files into a single dataset, we need to shift subfolder files all into a single location before running a code to merge the data. For the UK dataset, the process was the most tedious as it has multiple files with all sorts of issues, the other region’s dataset preparation was much more simplified.

For the UK dataset preparation, the two thousand files were all in their subfolders. The first step was to shift all the files with the file extension of .xls and .xlsx into a single location via a code to extract out from the individual subfolders. Then the .xls files were then placed in one location and .xlsx in another for merging. The .xls files were merged into a single file and the .xlsx files were merged into a single file separately. The corrupted file was removed using trial and error to remove corrupted files when merging both file types. Both the resulting file was saved as a .csv file. The two files were then merged into a single .csv file. At this point, the initial columns were selected by manually looking at the column headers as well as using a code to observe the total null value for each header.

Overall, the steps for preparing the data for the eleven regions vary from one another but the process is mainly the merging of files or sheets and the initial selection of columns. Following the data preparation, we can then move on to data understanding.

## **Chapter 3.3: Data Understanding**

For data understanding, we would be performing Exploratory Data Analysis (EDA) on each dataset before and after the data cleaning and preparation.

The key functions and methods used in our EDA are the head() method to get an overview of the dataset, the shape function to see how many rows and columns there are, info() to see the non-null count, describe() to show data statistics, isnull().sum().to\_string() to see number null values for each column, and plotting of graphs. The initial insight will let us determine which columns to keep based on null values. The null values are then cleaned in data cleaning. We also utilize the dtypes function to observe the data types of our dataset DataFrames. For our study, all columns are intended to be float type, however, some columns would show up as an object type due to strings present in the columns or errors in data input like a semi-colon added to a number. These would also be dealt with when cleaning the data.

For each dataset, the EDA process would be similar, but the data cleaning and preparation vary due to the differences between datasets. The process of EDA performed on the UK dataset will be shared below to show the general procedure. The process below will be similar for the remaining 10 regions.

### **Chapter 3.3.1: UK**

The dataset is first loaded into a DataFrame using the pd.read\_csv code. The head() method is then used on the DataFrame to have an overview of the top five rows of the DataFrame. This is followed by a shape function which tells us the number of rows and columns there are in the DataFrame. This, however, includes rows that may be empty in all columns. We then applied the isnull().sum() to see the number of null values in each column. The dtype function was applied to see the column dtype to see if they needed to be changed.

At this point, an iterative process was executed to make a list of string so that they can be removed with np.nan, and numbers that were represented in string form was also changed to numbers. The null values were then dropped with dropna(how=’all’) where if a particular row is empty for all columns, the row will be dropped. The isnull().sum() is executed again to see the remaining null values per column. From here, a couple of dropna with different settings were executed to reduce overall null values. The data type was then changed with the pd.to\_numeric applied to all columns of the UK dataframe.

The info() method is then used to show information on the columns including labels, number of columns, number of non-null rows in each column, and data type. Next is the .describe(), which shows the statistical data of the columns like mean, min, and max. Finally, we use .corr() to see the correlation between columns.

After these steps, we have a basic idea of the dataset before proceeding to further prepare the data. After which, the shape function is executed to see the final number of rows and columns.

The data have been normalized using MinMaxScaler() so that all column values would be of a similar range. At this point, we plotted charts to visualize the data. The charts are a scatter plot of porosity vs permeability, and two boxplots of porosity and permeability. The goal of the plots is to observe the data distribution.

**Fig 2**

Scatterplot Porosity vs Permeability of the Full Normalized Dataset

Chart, scatter chart

Description automatically generated

**Fig 3**

Porosity Boxplot of the Full Normalized Dataset

Chart, box and whisker chart

Description automatically generated

**Fig 4**

Permeability Boxplot of the Full Normalized Dataset

A picture containing diagram

Description automatically generated

The steps applied for the other regions are like the ones described above and may have slight variations.

## **Chapter 3.4: Data Preparation**

Data cleaning and preparation for the study mainly revolves around dealing with missing values, inappropriate data types, removal of outliers, and scaling the range of the columns to be comparable with one another.

The process for each dataset is different from one another depending on the data available and the proportion of missing values to available data. The methods used include dropna(), KNNImputer(), MinMaxScaler(), .replace(), to\_numeric(), and deriving porosity using grain density and bulk density. dropna() is for the removal of rows with null values, parameters were changed to adjust the removal of row conditions. KNNImputer() was used to fill in missing data using the KNN based on nearest neighbors. replace() was used for replacing strings with nulls. to\_numeric() was used to convert the column into float type. The porosity is calculated from grain density and bulk density with the formula of which was taken from (Richard, 1954). Outliers were treated by removing data that are 3 standard deviations away from each column’s standard deviation and mean and null values are checked using the isnull().sum() method. The methods and processes for each dataset are different depending on the data. All datasets are normalized to a range of 1 – 0. Outliers are also treated for all datasets.

For the UK dataset, the required columns for the traditional rock typing were also derived here after the min-max normalization. For the UK data, an un-normalized version of the data was prepared alongside the normalized one for the plotting of charts. Using the normalized data, additional columns are derived from the porosity and permeability columns. The columns are Reservoir Quality Index (RQI), Porosity Index, Flow Zone Indicator (FZI), Log RQI, and Log Porosity Index. This would result in some rows having infinity values due to calculation errors the rows containing infinity values were then removed. After a reset index on both sets of data, the normalized data were sorted by FZI ascending and the index was used to reorder the normalized dataset to match the order of both datasets. A reset index is performed on both datasets before saving them into 2 different CSV for modeling. The output from the data preparation here is 2 separate UK datasets, one being un-normalized with only the original values of porosity and permeability and the other dataset being min-max normalized with the additional columns calculated. Both dataset rows are matching one another for plotting purposes. The unnormalized data is for chart plotting visualization while the normalized data is used for training the models. After the data is cleaned, the Exploratory Data Analysis is continued as previously mentioned.

# **Chapter 4: Modelling and Evaluation**

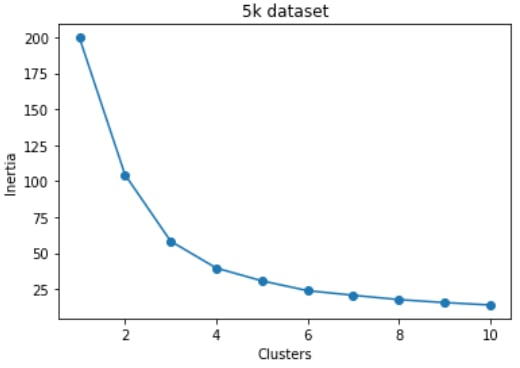
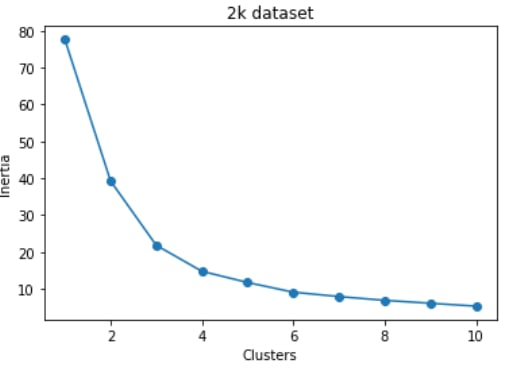
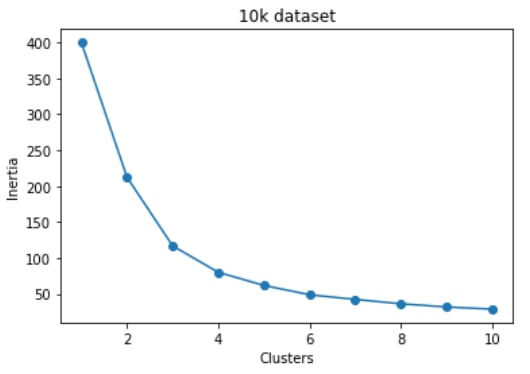
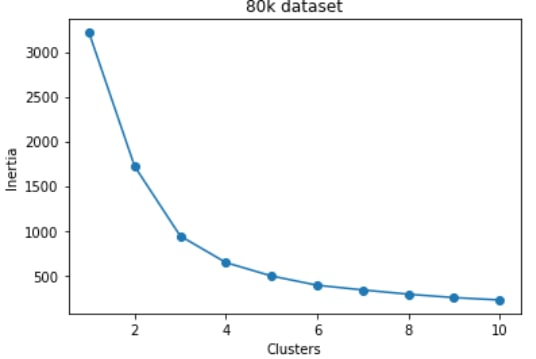
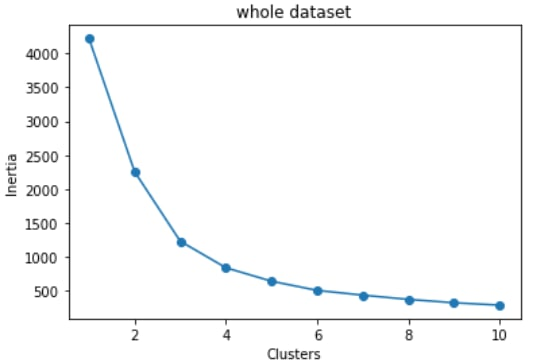
## **Chapter 4.1: Methods**

In this project, we trained several models using the normalized UK core dataset that was compiled and for visualization post-training of the model, the cluster results are tagged back to the unnormalized data for plotting of charts.

As our UK core dataset that was cleaned and prepared consists of over 100k rows, we sample tested different sizes of the dataset using the KMeans algorithm. We decided to test a total of 6 different sizes of the UK core dataset with the sizes being 2000, 5000, 10000, 40000, 80000, and the full dataset at 104832 rows. The 6 sample datasets are derived from the normalized UK core dataset with the same random state. We ran the 6 different sample datasets on K-Means with the initialization parameter set to k-means++ for cluster ranges of 1 to 10 with the same random state. An Elbow plot was used to determine the optimal cluster, which for all the datasets, the optimal cluster is 4 clusters as seen in the fig below.

**Fig 5**

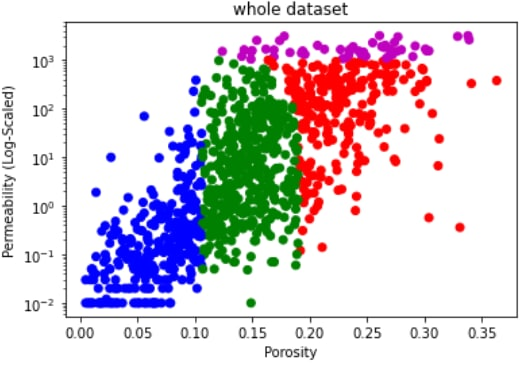
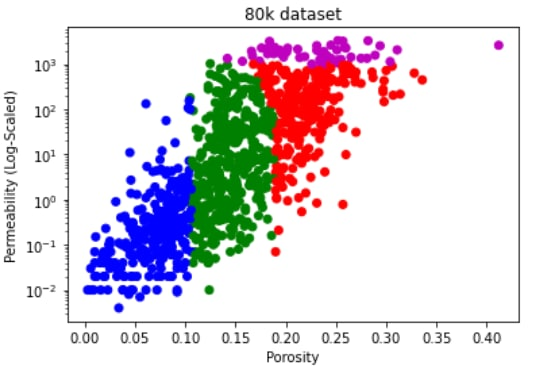
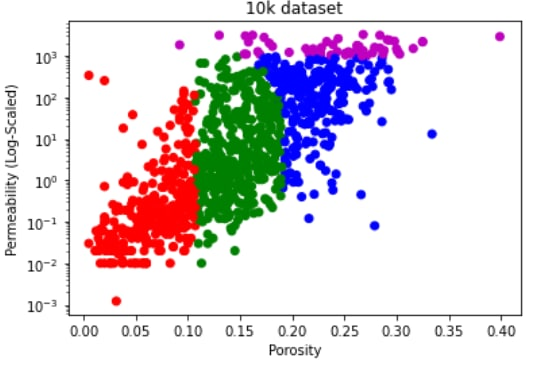
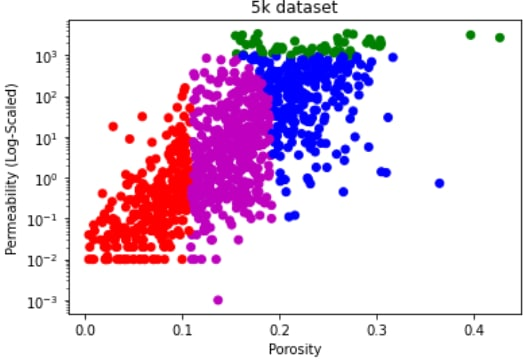
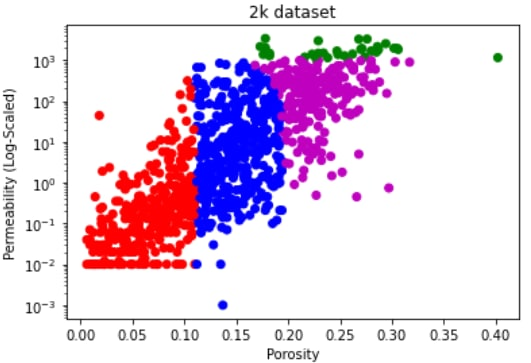
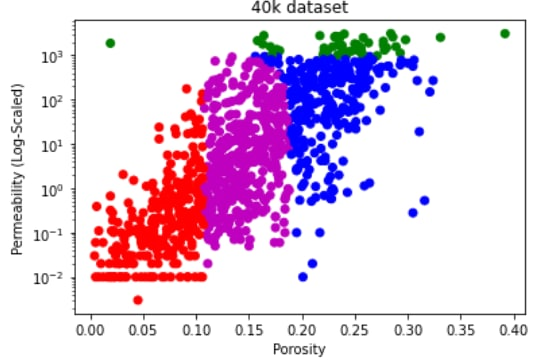
Inertia Plots of KMeans for Different Sample Sizes of the Un-normalized Dataset



For each of the sample datasets, K- Means was run with the cluster input equating to 4 which is their optimal cluster, and the labels are tagged back to unnormalized data. 6 charts of the x-axis being porosity and the y-axis being log-scaled Permeability are then plotted using 1000 random samples with the same random state for consistency. The result is that the output was similar for all 6 datasets and hence we chose to use the dataset with 2000 data for computational efficiency. Both the normalized dataset and the unnormalized one will be used to derive smaller datasets of 2000 rows with both samplings at the same random state. The training and plotting of charts will be using the 2000 rows of datasets. Below shows the scatterplot of the 6 datasets, which shows a high similarity between all 6 charts.

**Fig 6**

Porosity vs Log Scaled Permeability Plots of KMeans for Different Sample Sizes of the Un-Normalized Dataset



We first rock type using the traditional rock typing method derived from previous studies but with our implementation of the method. The implementation that we can up with was by referencing the iterative multi-linear regression clustering technique and the analysis by least square regression method (Khalid et al., 2019; Abdulelah et al., 2018). In Khalid et al. (2019), the iterative multi-linear regression technique was one of the more accurate yet tedious techniques for rock typing. We tried applying the method, but the method failed due to not being able to observe a clear separation of data points of a Log RQI (y-axis) vs Log Porosity (x-axis) scatter plot where points that lie on the same line with slope = 1 would mean that they have similar hydraulic flow unit. This would indicate they would likely be the same rock type. Below shows the Log RQI (y-axis) vs Log Porosity (x-axis) of our UK normalized dataset where the points do not have clear separation for us to estimate the HFUs. This highlights the weakness of the traditional method of rock typing, which is that it fails when data points are too close together or when there are too many data points. The Fig below shows the normalized dataset

**Fig 7**

Log Porosity Index vs Log RQI Scatterplot of the Normalized Training Dataset

Chart, scatter chart

Description automatically generated

Our approach to this was to come up with our method regarding this method as well as the analysis by least square regression method mentioned in another study (Abdulelah et al., 2018). This method would then act as our benchmark for comparison against the unsupervised model’s performance.

The unsupervised learning algorithms that will be trained in the study are the K-means, Self-Organizing Map (SOM) + K-means, DBSCAN, BIRCH, and Gaussian Mixture Models (GMM) algorithms. Different types of clustering algorithms are considered for the same sets of data. The models proposed are considered top algorithms for unsupervised learning algorithms to observe how well they rock type with adjusted parameters. The parameter required for the different unsupervised algorithms were changed and tested for different values to optimize the result when evaluating with the Silhouette Coefficient, Calinski-Harabaz Index, and the Davis-Bouldin Index metrics. The parameters that gave a better score overall were chosen so that the model performances are compared when the algorithms are all optimized iteratively.

For model evaluation, we will have several comparisons to observe how well the algorithms perform for rock typing. First, will be the unsupervised algorithms against the benchmark, which is our implementation of the traditional method. Next will be a comparison between the unsupervised algorithms using the Silhouette Coefficient, Calinski-Harabaz Index, and the Davis-Bouldin Index. Lastly, we will predict permeability using each method and compare their residual models.

### **Chapter 4.1.1: Benchmark Method**

The first model built is our implementation of the traditional method as previously mentioned. The iterative multi-linear regression (IMLR) clustering technique in a previous study was attempted first due to it being one of the more accurate methods for rock typing despite it being more tedious (Khalid et al., 2019).

The other approaches mentioned in the study for rock typing are a histogram analysis with log FZI in the x-axis and a normal probability plot with cumulative distribution function with log FZI in the x-axis. Both methods require rough guessing of the range which is more inaccurate than the IMLR.

For the histogram analysis method, the rock types are determined by choosing the ranges of the log FZI based on the closeness of the separate bars. This can be difficult to determine as data might not be so distinctly clustered between certain ranges and may be spread out.

For the normal probability plot method, the number of clusters is determined by the number of distinct straight lines that lie on the plot. For such a plot, identifying clear distinct lines with accurate ranges can be challenging and highly inaccurate like the histogram method.

The IMLR approach is the third approach mentioned in the study and the most accurate approach for identifying HFU. First will be plotting the log porosity index against log RQI, then guessing where to plot straight lines with slope = 1 takes place via observation of the plot with the intercept being the mean FZI. The number of straight lines, and HFUs depends on own judgment and the plot. The core samples are then allocated to the nearest line to them. The intercept of the HFUs is recalculated using least square regression. The old and new intercepts are compared and if the difference is not acceptable, the straight lines must be replotted via trial and error. This approach compared to the other two is more accurate as we can better define the ranges of FZI.

However, applying the IMLR approach to our dataset was a failure as we are not able to estimate the number of HFU, and is difficult to guess the straight-line locations. Our approach to this was to develop our implementation.

For our method to work, our dataset needed to be sorted by increasing FZI, which has been done in the data preparation phase. The process follows the diagram below.

**Fig 8**

Process of Benchmark Method

Diagram

Description automatically generated

### **Chapter 4.1.2: K-Means**

K-Means was chosen as it is one of the most widely used clustering methods. It is a centroid-based clustering method, which is a simple yet efficient algorithm. For K-Means, an initial number of clusters must be inputted where random starting points will be initialized (Sharma, 2022). Those points would act as cluster centroids and the data points would assign to the nearest cluster and the center point will shift to be the average of all the data points in that cluster (Sharma, 2022). The process repeats until all data points belong to a cluster (Sharma, 2022). An important step is that the cluster needs to be chosen beforehand and this was done by fitting K-Means with multiple K-clusters and compare with an inertia plot. The process applied for our K-Means model training is as follows.

**Fig 9**

Process of KMeans Method

Diagram

Description automatically generated

### **Chapter 4.1.3: SOM+K-Means**

A Self-Organizing Map (SOM) is an artificial neural network that uses competitive learning to weight adjust weights in neurons (Ralhan, 2018). We built the SOM using the MiniSOM library (Vettigli, 2018). We initialized a large SOM grid of 10X10 with default sigma and learning rates before applying K-Means to get a smaller number of clusters. The optimal cluster was decided with the inertia plot. Below is a simple flow of the process.

**Fig 10**

Process of the SOM + KMeans Method

Timeline

Description automatically generated

### **Chapter 4.1.4: Density-Based Spatial Clustering of Application with Noise (DBSCAN)**

DBSCAN is a clustering algorithm that belongs to the density-based clustering family (Kumar, 2021). It clusters the data points together based on a certain defined maximum distance between two points called epsilon (Kumar, 2021). Those that are sufficiently close together belong to a single cluster and those points that don’t have any data points near the defined epsilon are considered outliers (Kumar, 2021). In our case, we chose the outlier points to be their cluster instead of noise due to the points coming from a certain area. There are two parameters required for the DBSCAN algorithm, epsilon, and min samples, where epsilon is the max distance between points allowed to be considered as part of a cluster, and min samples are the minimum points that are required for a formation of a cluster (Kumar, 2021). The nearest neighbor of 4 for all data points was used to find the average distance between each point from its 4 nearest neighbors (Kumar, 2021). The average distance of the points was plotted in ascending order and the point of the greatest bend is used as the epsilon value (Kumar, 2021). The kneed package was used to identify that point.

**Fig 11**

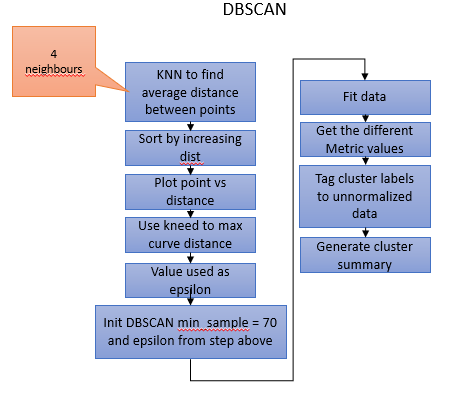
Elbow Plot with Kneed to Find Knee Point for DBSCAN Epsilon

A picture containing chart

Description automatically generated

**Fig 12**

Process of DBSCAN



### **Chapter 4.1.5: Balanced Iterative Reducing and Clustering using Hierarchies (BIRCH)**

The BIRCH method is a hierarchical clustering method that generates a small summary of the large dataset to cluster with as much information retained as possible which is referred to as a CF Tree (Verma, 2021). The three parameters needed are threshold, branching factor, and n \_cluster (Verma, 2021). The threshold represents the number of data points leaf node sub-clusters can have in a CF tree, the branching factor represents how many sub-clusters can a leaf node have and the n\_clusters represent how many clusters we require (Verma, 2021).

**Fig 13**

Process of BIRCH

Diagram

Description automatically generated

### **Chapter 4.1.6: Gaussian Mixture Models (GMMs)**

GMMs assume that the data consists of several Gaussian distributions and for each data, the points are assigned to a cluster based on probabilistic methods (Singh, 2022). For example, a data point that lies between two clusters has a different probability of being in either one using the GMMs (Singh, 2022). It assigns the data to the cluster it most likely belongs to. GMMs work like K-Means but the clusters can be non-circular, unlike K-Means (Singh, 2022). The initialization is selected as K-Means and the number of components is determined using BIC Criterion (Singh, 2022).

**Fig 14**

GMM BIC Plot

Chart, line chart

Description automatically generated

**Fig 15**

Process of GMM

Diagram

Description automatically generated

## **Chapter 4.2: Results and Discussion**

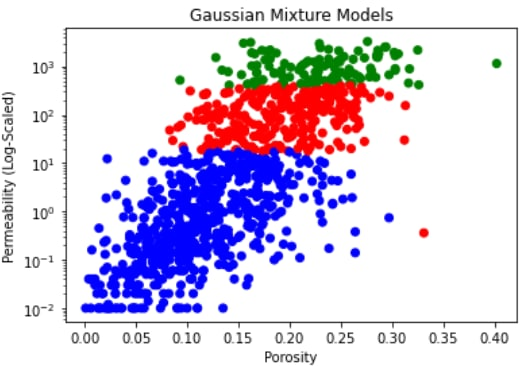
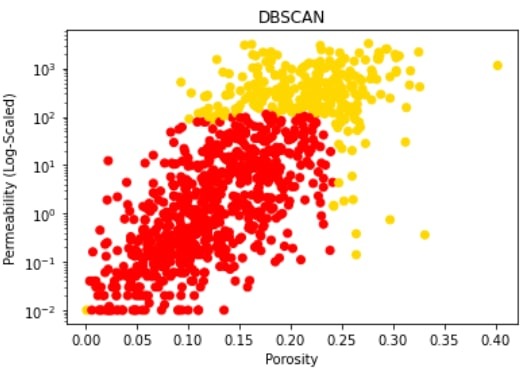
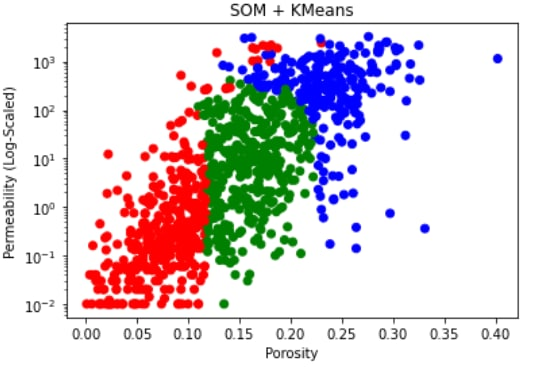
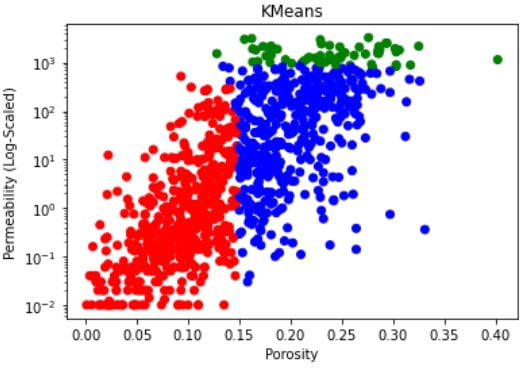
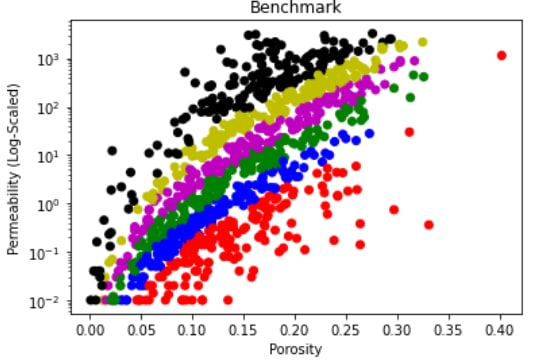
We applied several methods of evaluation to observe the overall performance of all our clustering methods. For our first method of model evaluation, the clustering methods were compared against the benchmark method using the porosity (x-axis) vs log scaled permeability (y-axis) scatter plot with the clusters shown as well as comparing cluster averages each method to see if any of the 6 clustering methods produce similar results. For our second method of model evaluation, using a sample dataset of 50 samples of the 2000 sample dataset used for training, we performed permeability predictions using our clustering algorithms and plotted residual models. The residual models were then compared against one another to observe how scattered each plot is. For the third method of model evaluation, we compare the unsupervised algorithms using the Silhouette Coefficient, Calinski-Harabaz Index, and the Davis-Bouldin Index.

### **Chapter 4.2.1: Comparing Porosity vs Log Scaled Porosity Plots and Cluster Averages**

For each of the 6 methods, after the model was trained using the normalized dataset, the cluster labels were retrieved and tagged back to the unnormalized dataset to have a visualization of the clusters on the normal value of the columns. Each cluster was tagged to a specific color and a color list was passed into the scatterplot to show the clusters. For clearer visualization, only 1000 sample of the 2000-row dataset was used for visualization. A porosity vs log scaled permeability scatterplot was plotted for all 6 methods porosity being the x-axis and permeability the y-axis but shown log scaled. After plotting the scatterplots, a group by operation on the ‘clusters’ column is performed on the unnormalize dataset and the mean of the columns is attained by applying the mean() function. All values were rounded off to 3 decimal places. This would show the mean porosity and permeability of each cluster for each of the clustering methods which would then be compared to one another. Ideally, the unsupervised algorithms results would resemble the one attained by the own implementation of the traditional method. However, the results displayed below shows that the outputs from the 5 clustering algorithms do not match the benchmark method which is expected as how each of the method works is very different from the benchmark method.

**Fig 16**

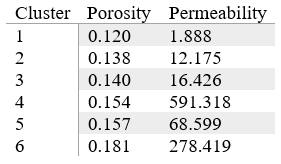
Porosity vs Log-Scaled Permeability Scatter Plot of All 6 Models



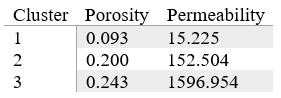
**Table 1**

Cluster Porosity and Permeabiliy Averages of All 6 Models

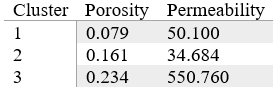
Benchmark



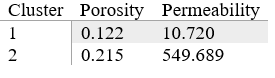
KMeans



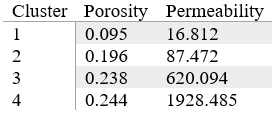
SOM + KMeans



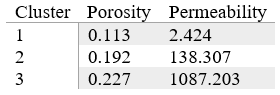
DBSCAN



BIRCH



Gaussian Mixture Models



Comparing the clusters generated from the 5 unsupervised machine learning methods against the benchmark in Fig 16, none of the scatterplots resembles the output of the benchmark method. Looking at each method’s cluster averages in Table 1, there is little similarity between the values for all the methods. Based on Table 1, BIRCH and KMeans are the most similar with the average porosity and permeability of the KMeans cluster 1 being like the BIRCH cluster 1 and the KMeans cluster 3 is to the BIRCH cluster 4.

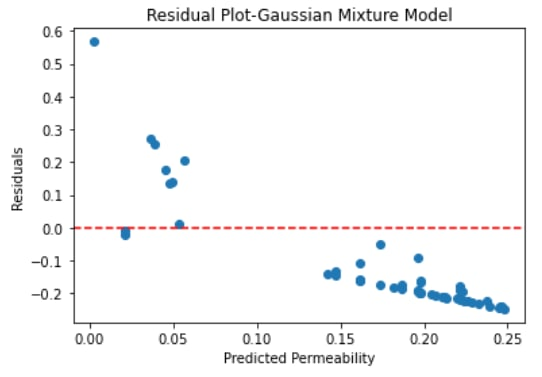
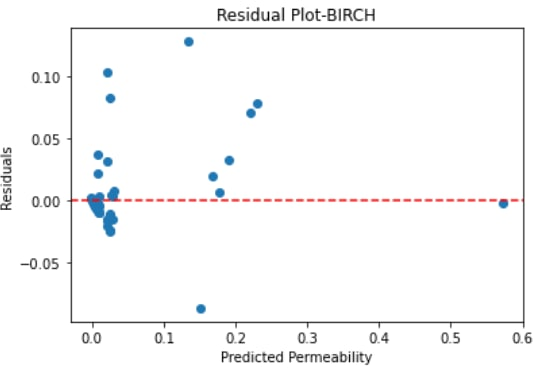
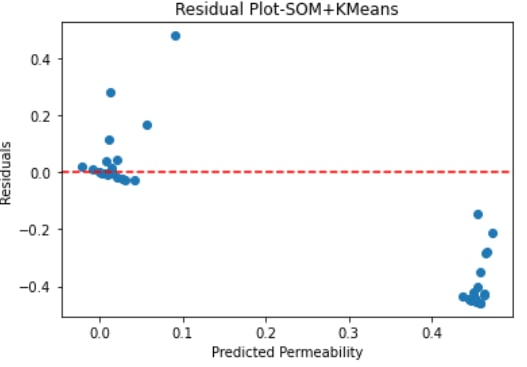
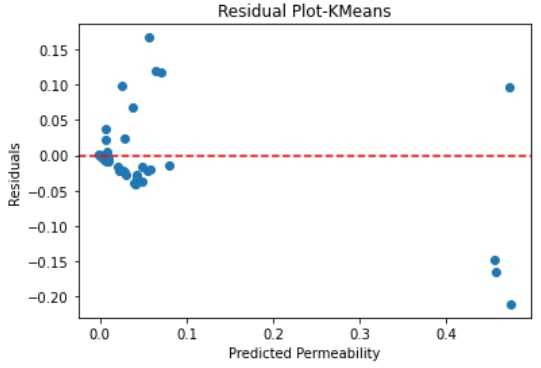
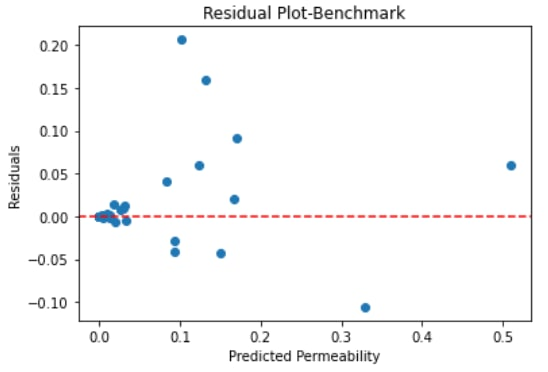
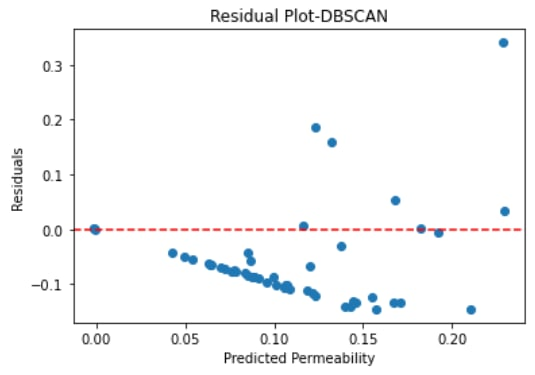
### **Chapter 4.2.2: Comparing Residual Models**

For this approach of evaluation, we did permeability prediction, then calculate the residuals to get the residual models for each of the 6 methods. Permeability prediction is one of the applications that are related to rock typing and has been used in the field for a long time. For this evaluation, the purpose was to compare the residual models for all 6 methods that are obtained by permeability prediction which can show us the sample data deviates from each of the models. This can give us some insight into the performance of each model for rock typing.

For the benchmark method, using the sample sets’ porosity, the log porosity index values were calculated and used to predict the log RQI by using the linear regression prediction depending on the FZI of the sample which affects which linear regression model to be used for prediction as our benchmark method when training the model, splits the dataset into equal parts depending on the iteration number. As the FZI is already sorted in the data preparation stage, the linear regression fitted HFU is based on a specific range of non-overlapping FZI values. The sample data points FZI is compared against the ranges for the 6 FZI ranges that were determined when training the model and the corresponding linear regression would be used to predict log RQI of the sample data point. The predicted RQI can then be worked out from the predicted RQI and the predicted permeability can then be calculated by performing some calculations. The residuals are then obtained by having the actual sample permeabilities minus the predicted permeabilities. A scatterplot with predicted permeability on the x-axis and residual on the y-axis with a dashed horizontal line at point 0 is plotted. This process is repeated for all 6 methods. Some of the methods have a simple approach to calculating the predicted permeability by simply using the predict function like the KMeans algorithm using the sci-kit learn KMeans. However, most of the methods do not have such prebuilt functions. What was done is very similar to the process of getting the predicted permeability and residuals for the benchmark method. In short, each cluster is fitted with a linear regression that will be used for permeability prediction for the samples. The sample’s predicted cluster is first obtained for the sample set data points. The samples’ predicted permeability is then obtained by predicting using the different cluster’s linear regression prediction method. The residuals can then be derived, and the residual model plotted. The residual models for all 6 models are then plotted and will be shown below.

Fig 17

Residual Plots for All 6 Models’ Permeability Prediction



Comparing the residual plots of the 6 clustering methods in Fig 17, the KMeans and BIRCH models had a small spread of residuals around residual = 0 like the benchmark method. SOM+KMeans, GMM, and DBSCAN have a rather wide spread of residual around residual = 0 which indicates that those models may be a poor fit for the data. The benchmark method has a small spread of residuals around the residual = 0 point which suggest the model is a good fit for the data. This is expected as it is a method adapted from the actual industry practices for rock typing. The comparison shows that Benchmark, KMeans, and BIRCH models are quite a good fit for the data. This may suggest the applicability of KMeans and BIRCH for rock typing over traditional rock typing methods, but this is not accounting for different data distributions which may cause results to differ when the same models are built on a different set of data.

### **Chapter 4.2.3: Comparing Unsupervised Machine Learning Algorithm Using Metrics**

This method of evaluation is comparing the unsupervised machine learning algorithms using metrics. The metrics chosen are the Silhouette Coefficient, the Calinski-Harabaz score, and the Davis-Bouldin Index. This evaluation excludes the benchmark as there was no straightforward way to for applying these metrics to the model. However, an implementation is possible to derive the metrics values for the benchmark method which was not touched on in this study. These metrics evaluate the uniqueness of clusters. For the Silhouette Coefficient and the Calinski-Harabaz index, the higher values are better. For the David-Bouldin Index, the lower values are better. The required inputs needed are the dataset and labels, of which the unsupervised algorithms used have implementations retrieve the labels. Below is an overview of all 3 metrics on the 5 unsupervised machine learning algorithms with values rounded to 3 decimal points.

**Table 2**

Summary of the 5 Clustering Models’ Metrics

|  |  |  |  |
| --- | --- | --- | --- |
| Algorithm | Silhouette Coefficient | Calinski-Harabaz Index | Davis-Bouldin Index |
| KMeans | 0.529 | 2556.740 | 0.642 |
| SOM+KMeans | 0.412 | 1505.616 | 0.859 |
| DBSCAN | 0.404 | 1180.614 | 1.041 |
| BIRCH | 0.474 | 2170.326 | 0.839 |
| GMMs | 0.263 | 1090.622 | 1.032 |

Based on Table 2, looking on an individual level for the silhouette coefficient, KMeans have the highest score, for the Calinski-Harabaz Index, KMeans and BIRCH have high scores and for the Davis-Bouldin Index, KMeans and BIRCH the lowest relative scores. From an overall viewpoint, both KMeans and BIRCH have good scorings based on the 3 metrics and have been shown to outperform the other algorithms.

# **Chapter 5: Conclusion and Future Works**

In this study, we performed rock typing using our implementation of the traditional rock typing method as well as 5 other unsupervised machine learning algorithms, namely KMeans, SOM+KMeans, DBSCAN, BIRCH, and GMM. The scaled-down dataset used for training consists of 2000 samples of the UK core data with porosity and permeability data which is the petrophysical properties that are typically used for rock typing in the field. The own implementation of the traditional method is derived by closely studying several studies. It acts as a benchmark model for us to compare the other models. We then compared all 6 trained models with optimized parameters using 3 different approaches.

When comparing the cluster distributions in the porosity vs log-scaled permeability scatterplots and the cluster averages of the 6 models, we observe that none of the 5 models' cluster distributions and cluster averages resemble the benchmark method results. KMeans and BIRCH cluster distribution and cluster averages are close to one another.

The 2nd approach to the evaluation was to perform permeability prediction using a sample set of 50 samples and plot residual models for comparison. The model that has less spread from the residual = 0 point would mean that the model fits the data best. Based on the comparison of the residual models, KMeans and BIRCH models have a similar residual spread to the benchmark model with the other models showing poor performance when used for permeability prediction.

The 3rd approach compares the Silhouette Coefficient, Calinski-Harabaz Index, and the Davis-Bouldin Index metrics for the 5 unsupervised machine learning algorithms. Overall, KMeans and BIRCH are the best-performing algorithms based on 3 metrics.

Overall, when comparing the 5 unsupervised machine learning models to the benchmark model, KMeans and BIRCH show to be able to cluster the rock types the best and the performance is comparable to the performance of the benchmark model. Although the clustering algorithm models cluster distribution is not like the benchmark models’, it is still able to be used as a method of rock typing. Using unsupervised machine learning algorithms would bring about time savings and efficiency while not compromising the quality and accuracy of the clustering of rock types. However, this is not to say to solely replace the expertise of experts, data is only useful depending on who is using it. Experts will still need to take over after training the models to make sense of the results. This would bring us to progress in the oil and gas rock typing area.

The study only used the limited dataset of UK core data. There is room for improvement by including more algorithms, using more datasets to account for variability, and benchmarking on multiple traditional methods of rock typing.

(8544)

# **References**

Abdulelah, H., Mahmood, S., & Hamada, G. (2018). Hydraulic flow units for Reservoir Characterization: A successful application on Arab-D carbonate. *IOP Conference Series: Materials Science and Engineering*, *380*, 012020. https://doi.org/10.1088/1757-899x/380/1/012020

Data Underground (n.d.). [Data sets]. https://dataunderground.org/

Dutta, S. (2022, April 19). *Oil and gas production - an introduction*. Oil & Gas IQ. Retrieved from https://www.oilandgasiq.com/oil-and-gas-production-and-operations/articles/oil-and-gas-production

Faramarzi-Palanga, M., & Mirzaer-Paiaman, A. (2021). Investigating dynamic rock quality in two-phase flow systems using TEM-function: A comparative study of different rock typing indices. *Petroleum Research, 6*(1), 16-25. http://doi.org.suss.remotexs.co/10.1016/j.ptlrs.2020.08.001

Geomore. (n.d.). *Porosity and Permeability*. Oil on my shoes – the original petroleum geology site! Retrieved August 14, 2022, from https://www.geomore.com/porosity-and-permeability-2/#:~:text=Porosity%20consists%20of%20the%20tiny,to%20flow%20through%20the%20rock

GeoPovider (n.d.) North\_Sea [Data sets]. https://www.geoprovider.no/

Hollis, C., Vahrenkamp, V., Tull, C., MookerJee, A., Taberner, C., & Huang, Y. (2010). Pore system characterization in heterogeneous carbonates: An alternative approach to widely-used rock-typing methodologies. *Marine and Petroleum Geology, 27*(4) 772-793. https://doi.org/10.1016/j.marpetgeo.2009.12.002

Khalid, M., Desouky, S. E.-D., Rashed, M., Shazly, T., & Sediek, K. (2019). Application of hydraulic flow units’ approach for improving reservoir characterization and predicting permeability. *Journal of Petroleum Exploration and Production Technology*, *10*(2), 467–479. https://doi.org/10.1007/s13202-019-00758-7

Kumar, B. V. (2021, December 15). *Tutorial for DBSCAN clustering in python sklearn*. MLK - Machine Learning Knowledge. Retrieved November 6, 2022, from https://machinelearningknowledge.ai/tutorial-for-dbscan-clustering-in-python-sklearn/

Mirzaei-Paiaman, A., Asadolahpour, S. R., Saboorian-Jooybari, H., Chen, Z., & Ostadhassan, M. (2020). A new framework for selection of representative samples for special core analysis. *Petroleum Research, 5*(3), 210–226. https://doi.org/10.1016/j.ptlrs.2020.06.003

Mohamed, I. M., Mohamed, S., Mazher, I., & Chester, P. (2019). *Formation Lithology Classification: Insights into Machine Learning Methods* [Paper Presentation].Paper presented at the SPE Annual Technical Conference and Exhibition, Calgary, Alberta, Canada, September 2019*.* https://doi.org/10.2118/196096-MS

Mohammadian, E., Kheirollahi, M., Liu, B., Ostadhassan, M., & Sabet, M. (2022). A case study of petrophysical rock typing and permeability prediction using machine learning in a heterogenous carbonate reservoir in Iran. *Scientific Reports*, *12*(1), 4505–4505. https://doi.org/10.1038/s41598-022-08575-5

Ralhan, A. (2018, September 17). *Self organizing maps*. Medium. Retrieved from https://medium.com/@abhinavr8/self-organizing-maps-ff5853a118d4

Richards, L. A. (1954). Diagnosis and Improvement of. *Saline and Alkali Soils. Handbook*, *60*.

Rodriguez, M. Z., Comin, C. H., Casanova, D., Bruno, O. M., Amancio, D. R., Costa, L. da F., Rodrigues, F. A., & Kestler, H. A. (2019). Clustering algorithms: A comparative approach. *PloS One, 14*(1), e0210236–e0210236. https://doi.org/10.1371/journal.pone.0210236

ScienceDirect. (n.d.). *Sedimentary Basin*. Sedimentary Basin - an overview | ScienceDirect Topics. Retrieved from https://www.sciencedirect.com/topics/earth-and-planetary-sciences/sedimentary-basin

Shahat, J. S., Mohamed, I. B., Mahamed, S. E., & Attia, M. A. (2021). Resistivity zone index: A new approach in rock typing to enhance reservoir characterization using well log data. *Energy Reports, 7*, 711-723. http://doi.org.suss.remotexs.co/10.1016/j.egyr.2021.01.026

Sharma, P. (2022, June 15). *K means clustering: K means clustering algorithm in Python*. Analytics Vidhya. Retrieved from https://www.analyticsvidhya.com/blog/2019/08/comprehensive-guide-k-means-clustering/

Singh, A. (2022, June 15). *Gaussian mixture models: Clustering algorithm python*. Analytics Vidhya. Retrieved from https://www.analyticsvidhya.com/blog/2019/10/gaussian-mixture-models-clustering/

Soleymanzadeh, A., Parvin, S., & Kord, S. (2019). Effect of overburden pressure on determination of reservoir rock types using RQI/FZI, FZI\* and Winland methods in carbonate rocks. *Petroleum Science, 16*(6), 1403-1416 https://doi.org/10.1007/s12182-019-0332-8

Tavakoli, V. (2018). Rock Typing. *Geological Core Analysis*, 85-99. https://doi.org/10.1007/978-3-319-78027-6\_6

UK National Data Repository (n.d.). UK [Data sets]. https://ndr.nstauthority.co.uk/

U.S. Energy Information Administration. (n.d.). *Oil and petroleum products explained*. Oil and petroleum products explained - U.S. Energy Information Administration (EIA). Retrieved from https://www.eia.gov/energyexplained/oil-and-petroleum-products/

Verma, Y. (2021, November 11). *Guide to birch clustering algorithm(with python codes)*. Analytics India Magazine. Retrieved from https://analyticsindiamag.com/guide-to-birch-clustering-algorithmwith-python-codes/

Vettigli, G. (2018) MiniSom: minimalistic and Numpy-based implementation of the Self Organizing Map. https://github.com/JustGlowing/minisom/

Zakyan, Permadi, A. K., Pratama, E. A., & Naufaliansyah, M. A. (2022). An Improved Method of Clay-Induced Rock Typing Derived from Log Data in Modelling Low Salinity Water Injection: A Case Study on an Oil Field in Indonesia. *Energies (Basel)*, *15*(10), 3749. https://doi.org/10.3390/en15103749