**Title: Evaluating Different Clustering Algorithms for Prediction of Rock Types**

**Autho**rs: Munish Kumara,b and Tan Wen Xuanb

**Affiliations:** aERCE Singapore, 48 Tras Street, #03-01, Singapore 078987

bSingapore University of Social Sciences, School of Business,463 Clementi Rd, Singapore 599494

Corresponding author. Email: mkumar@erce.energy

**Abstract:** Rock typing has multitudes of uses, from optimizing drilling spots , to determining perforation zones, evaluating in-place volumes using static and dynamic models, to better understanding complex flow properties that take place in oil and gas reservoirs. Studies on rock typing have mainly utilized porosity and permeability measurements converted to rock typing indices, with newer studies introducing elements of supervised machine learning as well. However, a comparison of different unsupervised machine learning algorithms for rock typing applications has not been widely researched. This paper aims to perform a comparative study of the performance and outputs from 5 unique unsupervised machine learning models, which we will benchmark against a modified iterative, multi-linear, regression (IMLR) rock typing technique. This study will be conducted on a core data comprising 2000 unique data points from the United Kingdom (UK). The 5 unsupervised machine learning model are the 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 (GMMs). The study’s results showed that KMeans and BIRCH rock typing performance are the best aside from the modified method when applied on the core dataset.

**One-Sentence Summary:** This paper aims to explore the performance of rock typing using different unsupervised machine learning algorithms against a modified IMLR rock typing method.

**Keywords (minimum 6):** Oil and Gas, Rock Typing, Unsupervised Machine Learning, KMeans, Density-Based Spectral Clustering of Application with Noise (DBSCAN), Self-Organising Map (SOM), Balanced Iterative Reducing and Clustering using Hierarchies (BIRCH), Gaussian Mixture Models (GMMs).

# 1.0 Introduction

Rock typing is based on the simple principle that not all reservoir rocks are created equal. Rocks that are similarly deposited, buried under similar conditions, and undergo similar diagenetic processes will share a set of unique characteristics (Archie, 1950) . In work by Rushing, there are identified 3 different classes of rock typing – depositional, petrographic and hydraulic (Rushing, 2008). Of the 3, hydraulic rock typing (RT) seeks to characterize reservoir rocks into similar groups based on petrophysical properties which (a) quantifies the ability of fluids to flow and (b) determines the availability of pore volume for storage (Hollis et. al., 2020). The physical properties that govern fluid storage and flow are fundamentally linked to the dimension (size), geometry (shape, surface roughness, tortuosity) and distribution of the pore and pore-throat network. Such characterisation is routinely performed using routine core analysis porosity and permeability measurements (Corbett, 2004), but it can also be performed with mercury injection capillary pressure (MICP) as well (Tang, 2020). These unique porosity-permeability relationships and flow characteristics are integrated across multiple scales (nm to km), so that large scale geologic features can be adequately represented from small scale petrology and petrophysical properties. Often, RT at the core scale is done manually, with experienced litho-stratigraphers and experts in core analysis involved in the process of subdividing the data into families, based on depositional and geological principles.

In certain instances, the unique relationships from porosity and permeability are not readily discernable. Assuming stratigraphic continuous rock units have similar reservoir properties, and assuming a capillary tube model as defined by Kozeny-Carman, the concept of flow units/ flow zone index (FZI, mm) was developed to honor the geology while deconvolving the relationship between these 2 variables (Amaefule, Mehmet, Djebbar, Kersey, & Dare, October 1993). Of particular importance is the mean hydraulic radius of pore throats, Rmh, which is used to evaluate the reservoir quality index (RQI, mm), and pore volume to grain volume ratio (also known as the normalized porosity index or fz), defined as:

|  |  |  |
| --- | --- | --- |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

where k is permeability (mD) and f is porosity (V/V). For rocks with similar RQI, a graph of (log) FZI vs (log) fz will fall on a straight line with a slope of -1. Rock samples with similar but not identical RQI values will cluster around, but not on this line. Rock samples with significantly different RQI values will fall on other parallel lines with even slope but different

intercepts. The intercept of these lines at fz = 1 defines the mean pore throat size of each linear group of sample points. Numerous authors have derived combinations of these 2 properties to generate variations of FZI (e.g. FZI\*, FZIM and MFZI) either from core data or applied data from well-logs to generate Resistivity Zone Index (RZI) (see Appendix 1 for details).

# Using Machine Learning for RT applications

While RT is one way in which reservoir rocks are classified, the advent of faster computing coupled with advances in memory has encouraged geoscientist to attempt new data driven means to attempt rock classification. Some of the newer rock typing work now utilizes machine learning (ML) to complement or even supplant existing methods as the expectation is that such methods would prove more accurate at anomaly detection and correlative evaluation . Most of the work has however been via the utilization of supervised machine learning which requires expert knowledge to firstly prepare the training sets into fundamental rock types that can be later be applied to the testing data sets. What has been less researched is how unsupervised machine learning algorithms would perform in similar rock classification tasks.

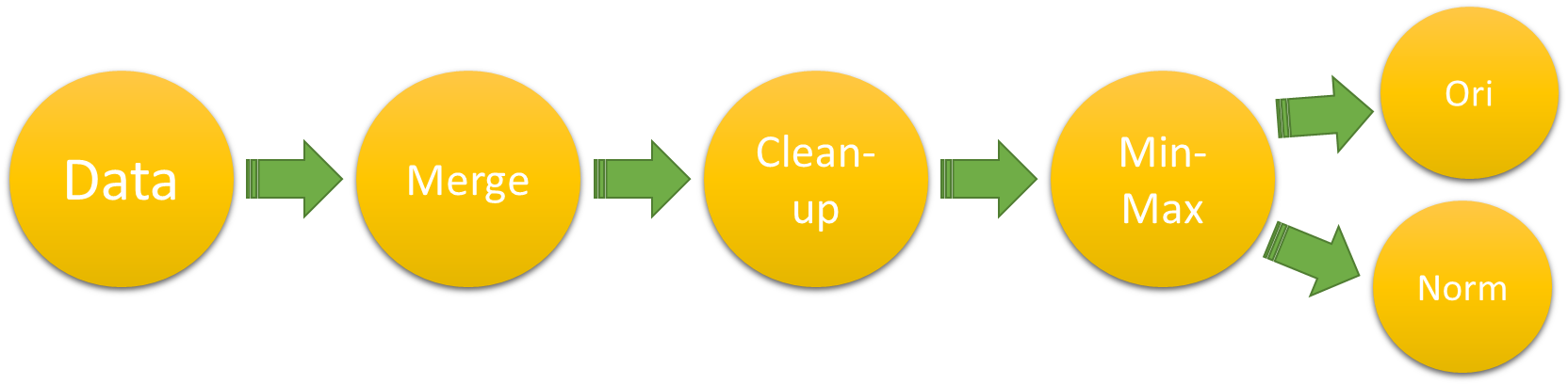
Mohammadian et al (Mohammadian, Kheirollahi, Ostadhassan, & Sabet, 2022) approached the task of rock typing using both supervised and unsupervised algorithms. He started with a data set of 128 carbonate core samples, and firstly applied supervised machine learning (Extreme Gradient Boosting or XGB) to generate an output in the form of a FZIM\*. The second step of this process involved using the FZIM\* output as input into a K-Means algorithm, ultimately resulting in 4 unique RT. A few things to note about this work is that his initial data set was small, he utilized other input variables besides porosity and permeability, like connate water saturation and data from MICP, and he required multiple ML methods to perform the classification. Mohamed et. al. (Mohamed, Mohamed, Mazher, & Pieprzica , 2019) similarly performed rock typing using unsupervised and supervised ML algorithms, with supervised ML classifiers such as Support Vector Machine (SVM), K-Nearest Neighbors (KNN), Random Forest – Decision Tree and one neural network approach. They only used one unsupervised ML method, K-means. Input variables were in the form of log data from 8 wells, but with facies already manually classified prior. They found that there were inherent differences between what was manually classified and what was classified computationally and stated that the unsupervised learning algorithm was the worst performing. Zakyan et. al. (Zakyan, Permadi , & Pratama, 2022) proposed a method which RT log data with artificial neural network (ANN) as an enhanced clay typing methodology. In both the work by (Mohamed, Mohamed, Mazher, & Pieprzica , 2019) and (Zakyan, Permadi , & Pratama, 2022), they data was purely lob based and did not consider core data.

# Scope & Methods

Given the dearth of information regarding unsupervised ML methods for core classification, we were interested in comparing the performance of 5 different unsupervised ML algorithms and benchmark it against the more “conventional” RT method based on the iterative, multi-linear, regression (IMLR) technique. We wanted to use a data source that was heterogeneous and would encompass a wide range of values, while also having sufficient density of data to warrant a ML technique. We decided that we would limit ourselves to one geographic locale (UK North Sea (Data Underground, n.d.), (GeoProvider, n.d.), (UK National Data Repository, n.d.)) and generate a “pseudo-data set” comprising of samples across the multiple sampled wells.

The data was available in an initially unstructured format. Input data files had to filtered, collated and refined before it can be further pre-processed. For instance, there were close to 2000 individual files which needed downloading and compilation. The format of the data present within each file was confirmed through random sampling and was not homogeneous. The file types themselves were not consistent, with ~30% of the files in ASCII format, 25% in comma separated variable format and the remaining in various excel vintages. There were also corrupt files present which needed to sieve out. We observed that some data files had as many as 250 columns, of which a majority were NULL or empty. We found that the most complete of the columns was only the porosity and permeability data, the very minimum of the data types needed for RT. Units were also inconsistent and had to be fixed across the different data files. Doing all this manually would have been extremely time consuming and tedious, with an estimate of >3000 man-hours spent sorting and standardizing the data. Repetitive tasks like this would also likely result in human errors, so we opted for a computational solution which automatically merged the individual data files into a single “mega-merge” (MM) file.

Following the generation of this MM file, we performed a “clean-up” by removing data which was of the NULL data type, alphanumeric or non-physical data (negative numbers or zero division errors). What was remaining was computationally corrected for, using statistical methods where empty data types were filled in with (mean/P10/P50?) values and outliers were removed to prevent data skew. The data is min-max normalized (Kumar, Swaminathan, Rusli, & Thomas-Hy, 2022) before being split into 2 separate csv files, one containing the unnormalized data file and the final input data file containing the normalized data. We schematically illustrate this process in Figure 1.



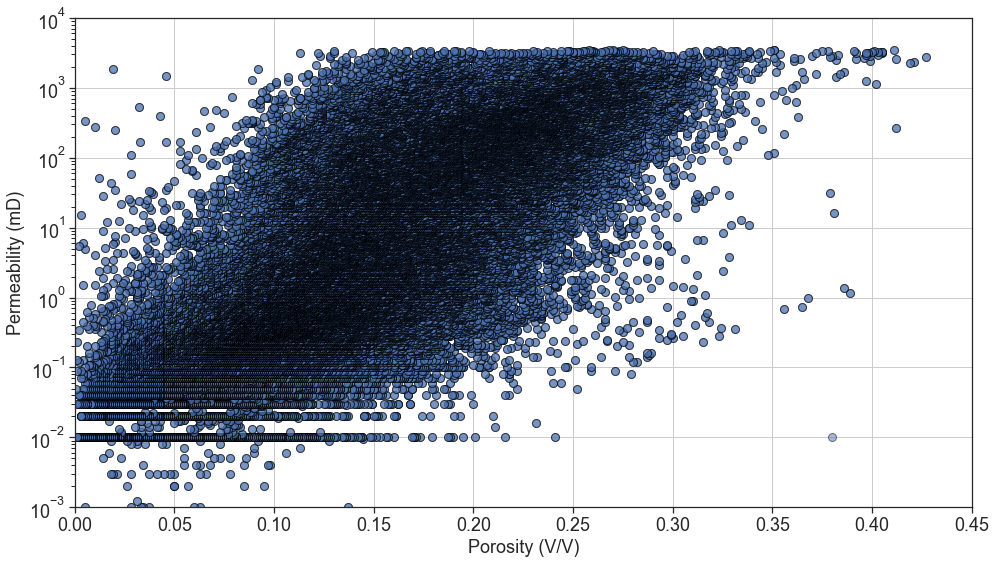
**Figure 1:** Generic data preparation flowchart

## Exploratory Data Analytics and Downsampling

Our final data set was made up of >100,000 unique data points. General histograms of the porosity and permeability of this data set is given in Figure 2. A cross-plot of the data is shown in Figure 3, and a table of descriptive statistics is given in Table 1.

|  |  |
| --- | --- |
|  |  |

**Figure 2:** Histograms depicting (Left) Porosity and (Right) Permeability of the core database



**Figure 3:** Histograms depicting (Left) Porosity and (Right) Permeability of the MM core database

**Table 1:** Descriptive Statistics for the MM Core Database

|  |  |  |
| --- | --- | --- |
| Statistics | Porosity (V/V) | Permeability (mD) |
| mean | 0.149581 | 176.162276 |
| std | 0.066884 | 434.395962 |
| min | 0.000500 | 0.001000 |
| 25% | 0.102000 | 0.370000 |
| 50% | 0.148000 | 5.800000 |
| 75% | 0.198000 | 113.000000 |
| max | 0.427000 | 3430.000000 |

While the data set above satisfies our requirement for sample size and heterogeneity, we are faced with a challenge in that the data itself is too dense to properly perform core classification using a clustering process, which would be the logical next step, if based purely on porosity and permeability. As stated earlier, RT would be manually done, with the rock being separated based on core description and facies observation. However, for data sets like this, this would not be possible. For optimal clustering using computational methods, the individual clusters would have be distinct and disparate enough so that distance based methods (which a lot of clustering algorithms are based on) would work reasonably well. Samples that were repeats or which had values close to one another did not necessarily add value to partitioning out of the individual clusters. Another issue with very large data sets like this is the rapidity of the analysis; for large data sets, running multiple clustering algorithms would be time consuming and computationally expensive.

We therefore attempted to downsample the data, being careful to optimize run time, while ensuring that no critical data was sacrificed in this process. Our approach was to use apply a relatively simple K-Means algorithms which would run quickly and which needed

Using this input data, we developed a workflow using 2000 randomly selected data points which tested 5 different unsupervised ML algorithms, for computational, interpretation and visualization ease.

## 2.0 ML Algorithm Choices

The choice of unsupervised ML algorithms 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 (GMMs). These algorithms were chosen because XXXXXX

Chart, scatter chart

Description automatically generated

Figure 3: Porosity Permeability scatterplot of the UK dataset

## 2.5 Selection of dataset sizing for modelling

Chart, line chart

Description automatically generated

Fig 6: K-Means Inertia Plot for the 2k, 40k and whole dataset normalized data

Each of the 3 datasets was run using K-Means with the optimal cluster derived from inertia chart, the labels obtained from the model training were tagged back to the corresponding unnormalized data. Charts were then plotted to visualize the cluster output for the 3 datasets using the 1000 random samples with same random state. The x-axis is porosity and y-axis are log-scaled permeability. The result shown in Fig 7 shows similarity of clustering results and hence the 2000 rows dataset is selected for computational efficiency. The reduction in data improves model training time and not overly affecting the results to a large extent. Both the normalized data and the un-normalized one is used to derive the smaller sets of 2000 rows with both sampling at the same random state.

Chart, scatter chart

Description automatically generated Chart, scatter chart

Description automatically generated

Chart, scatter chart

Description automatically generated

Fig 7: Porosity vs Log-Scaled Permeability Scatter Plots of K-Means for Different Sample Sizes of the Un-Normalized Dataset

## 2.6 Traditional Rock Typing and Failure of its Application

We attempted to rock type using the traditional rock typing method by referencing two techniques used, the iterative multi-linear regression (IMLR) clustering technique and the analysis by least square regression method (Khalid et al., 2019; Abdulelah et al., 2018) and using either technique. In Khalid et al. (2019). The IMLR technique was one of the more accurate rock typing techniques. We failed to apply the techniques but failed as data points do not have clear separation of a Log RQI (y-axis) and Log Porosity Index (x-axis) scatterplot where points that lie on the same line with slope being 1 will indicate the points having similar hydraulic flow units. Which this indicates that they are likely the same rock type. In Fig 8 shows a Log RQI and Log Porosity Index scatterplot where there are no clear lines of slope = 1 to segregate the datapoints by.

Chart, scatter chart

Description automatically generated

Fig 8: Log Porosity Index vs Log RQI Scatterplot of the Normalized Training Dataset

## 2.7 Modified Rock Typing Method

We came up with a modified version that is with reference to the two techniques previously mentioned. This method is used as a benchmark to compare the results from the machine learning algorithms with. The normalized and unnormalized dataset is first sorted by ascending FZI which is accomplished in the data preparation stage. The process flow can be seen in fig 9. The method is adapted from the IMLR and the least square regression method and adjusted to become the modified rock typing method used for benchmarking. Firstly, the log porosity index (y-axis) and log RQI (x-axis) scatterplot is plotted. Then from range of 1 till 10, the dataset is split into parts based on the loop number. Each mini dataset is used to train, fit, and predict linear regression and the average r-squared value is recorded. A r-squared vs number of clusters plot is plotted, and the optimal cluster is chosen. The method repeats using the optimal cluster number and the cluster labels are tagged back to the unnormalized data for visualization and label summaries are generated.

Diagram

Description automatically generated

Fig 9: Process of the Modified Rock Typing Method

## 2.8 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 in Fig 10.

Diagram

Description automatically generated

Fig 10: Process of K-Means Method

## 2.9 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.

Diagram

Description automatically generated

Fig 11: Process of SOM+K-Means Method

* + 1. **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 a 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. The knee is the point of the greatest bend to be used as the epsilon value for DBSCAN. Fig 12 shows the knee point of the dataset to and Fig 13 shows the process flow of DBSCAN.

A picture containing chart

Description automatically generated

Fig 12: Elbow Plot with Knee Point Identified by Kneed for Epsilon Value

Diagram

Description automatically generated

Fig 13: Process of DBSCAN Method

## 2.10 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).

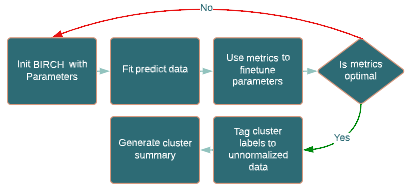


Fig 14: Process of BIRCH Method

## 2.11 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 Bayesian Information Criterion (BIC) (Singh, 2022). BIC is a model selection criterion where lower BIC means the model has lower penalty terms and thus the model is better (Datalab, 2019). The penalty term relates to number of parameters (Datalab, 2019). Fig 15 shows the optimal number components using the BIC plot and Fig 16 shows the process flow for GMMs method.

Chart, line chart

Description automatically generated

Fig 15: BIC Plot for Optimal Number of Components

Diagram

Description automatically generated

Fig 16: Process of Gaussian Mixture Model Method

# Results

We used 3 methods of evaluation to compare the performances of all the 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

## 3.1 Comparing Porosity vs Log Scaled Permeability Plots and Cluster Averages

For all the methods, training was done using the normalized dataset while visualization was done on the unnormalized dataset with the cluster labels taken from trained model. Each cluster was assigned a color and the list was passed into the scatterplot to observe the clusters. Only 1000 sample points was used for plotting for clearer visualization. Porosity vs log scaled permeability was plotted for all the models and with porosity being the x-axis and y-axis being permeability that was displayed log scaled.

The average values of the columns for each cluster of each method were obtained from the unnormalized data. The average table for each method was then compared against one another to find any resemblance in the clusters. This method would not be accurate as values would be greatly differing due to the scale and differences in cluster groupings.

The results displayed below for the scatterplots show that the 5 cluster methods cluster grouping on the scatterplot do not resemble the modified method. As for the cluster average table comparisons, the cluster averages of each method do not resemble one another.

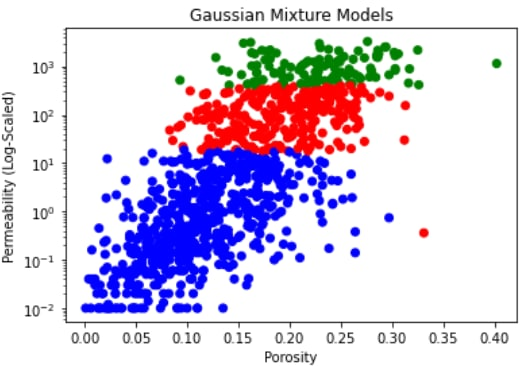
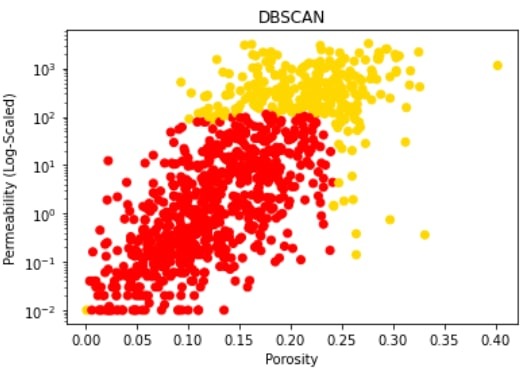
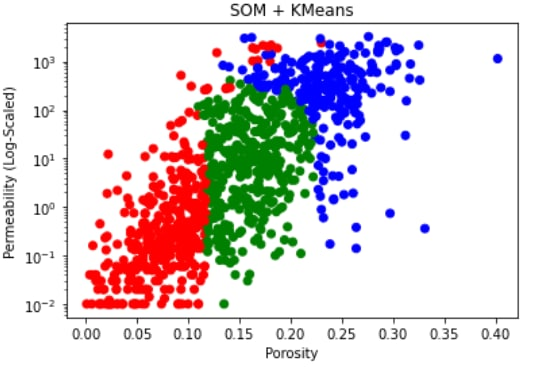
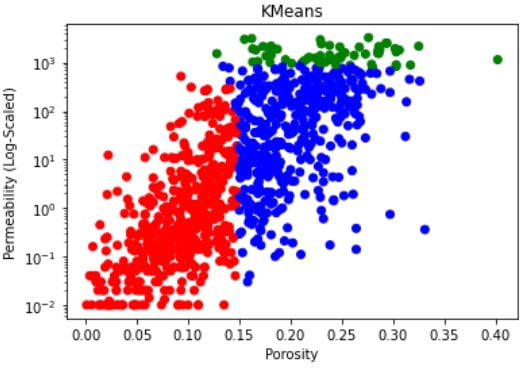
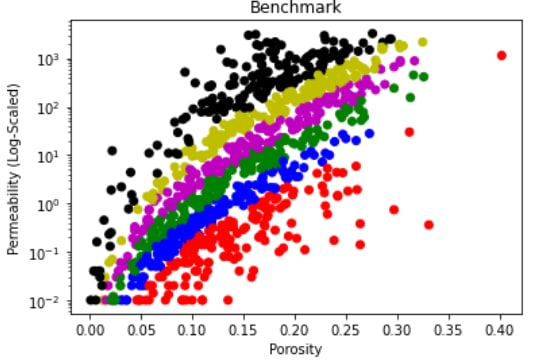
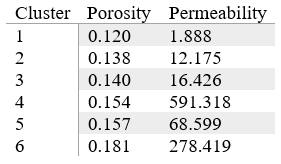
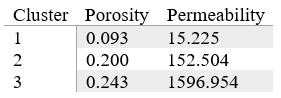


Fig 17: Porosity vs Log Scaled Permeability Scatterplot Cluster Distributions

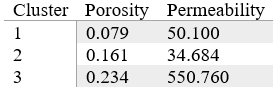
Benchmark



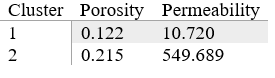
KMeans



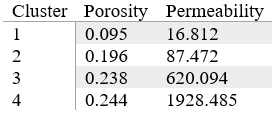
SOM + KMeans



DBSCAN



BIRCH



Gaussian Mixture Models

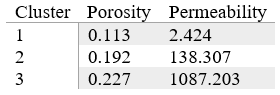


Table 1: Cluster Porosity and Permeability Averages

The cluster porosity and permeability averages of the 5 clustering algorithms comparing to the benchmark method do not look similar. The benchmark method has a total of 6 clusters whilst the 5 algorithms have 4 or less clusters which may indicate that it may be sufficient to just have a smaller number of rock types rather than the amount produced by the benchmark method. Based on Table 1 BIRCH and KMeans ae the most similar with the average porosity and permeability of cluster 1s in both algorithms to be very similar in terms of value. Likewise for KMeans’ cluster 3 to BRICH’s cluster 4.

## 3.2 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.

[still using lr rather than power functions, difficulty in plotting them]

[pending new charts]

## 3.3 Comparing Unsupervised Machine Learning Algorithms Using Metrices

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.

|  |  |  |  |
| --- | --- | --- | --- |
| 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 |

Table 2: Summary of the 5 Clustering Models’ Metrices

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.

# 4.0 Limitations of Study and Conclusion

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.

**References**

Amaefule, J. O., Altunbay, M., Tiab, D., Kersey, D. G., & Keelan, D. K. (1993). *Enhanced Reservoir Hydraulic (Flow) Units and Predict Permeability in Uncored Intervals/Wells* [Paper Presentation]. Paper presented at the SPE Annual Technical Conference and Exhibition, Houston, Texas, October 1993. https://doi.org/10.2523/26436-ms

Datalab, A. (2019, January 16). *What is Bayesian Information Criterion (BIC)?* Medium. Retrieved from https://medium.com/@analyttica/what-is-bayesian-information-criterion-bic-b3396a894be6

Izadi, M. & Ghalambor, A. (2013). A new approach in permeability and hydraulic-flow-unit determination. *SPE Reserv. Eval. Eng.* **16**(03), 257–264.

Kolodzie, S. (1980). *Analysis of pore throat size and use of the Waxman–Smits equation to determine OOIP in Spindle Field, Colorado* [Paper Presentation]. Paper presented at the SPE Annual Technical Conference and Exhibition, Dallas, Texas, September 1980. https://doi.org/10.2118/9382-MS

Mirzaei-Paiaman, A., Ostadhassan, M., Rezaee, R., Saboorian-Jooybari, H. & Chen, Z. (2018). A new approach in petrophysical rock typing. *Journal of Petroleum Science and Engineering,* **166**, 445–464. https://doi.org/10.1016/j.petrol.2018.03.075

Nooruddin, H. A. & Hossain, M. E. (2011). Modified Kozeny–Carmen correlation for enhanced hydraulic flow unit characterization. *Journal of Petroleum Science and Engineering,* **80**(1), 107–115. https://doi.org/10.1016/j.petrol.2011.11.003

**Appendix 1 – Rock Typing Indices Equations**

|  |  |  |  |
| --- | --- | --- | --- |
| Authors | Indices | Equations | Equation number |
| (Amaefule et al, 1993) |  | Where:  k is permeability (mD)  f is porosity(V/V)  fz is normalized porosity(V/V)  RQI is Rock Quality Index (mm)  FZI is Flow Zone Index (mm) |  |
| (Mirzaei-Paiaman et al, 2018) | FZI\* | Where:  k -> permeability  -> porosity |  |
| (Nooruddin and Hossain, 2011) | FZIM | Where:  m -> cementation factor |  |
| (Izadi and Ghalambor, 2013) | MFZI | Where:  -> irreducible water saturation  -> Rock Quality Index |  |
| (Shahat et. al.,2021) | RFZI | Where:  -> resistivity index  -> true formation resistivity  -> formation resistivity when water saturation is 100%  -> formation water resistivity  -> surface grain volume  k -> permeability  -> shape factor |  |
| (Kolodzie, 1980) | Winland | Where:  -> uncorrected air permeability  -> porosity  r35 -> pore throat radius with 35% mercury saturation |  |