Journal of Petroleum Exploration and Production Technology
https://doi.org/10.1007/s13202-024-01789-5

ORIGINAL PAPER-PRODUCTION ENGINEERING

Automated real-time prediction of geological formation tops during drilling operations: an applied machine learning solution for the Norwegian Continental Shelf

Behzad Elahifar[1] · **Erfan Hosseini**[2]

Received: 17 December 2023 / Accepted: 11 March 2024

Abstract

Accurate prediction of geological formation tops is a crucial task for optimizing hydrocarbon exploration and production activities. This research investigates and conducts a comprehensive comparative analysis of several advanced machine learning approaches tailored for the critical application of geological formation top prediction within the complex Norwegian Continental Shelf (NCS) region. The study evaluates and benchmarks the performance of four prominent machine learning models: Support Vector Machine (SVM), K-Nearest Neighbors (KNN), Random Forest ensemble method, and Multi-Layer Perceptron (MLP) neural network. To facilitate a rigorous assessment, the models are extensively evaluated across two distinct datasets - a dedicated test dataset and a blind dataset independent for validation. The evaluation criteria revolve around quantifying the models' predictive accuracy in successfully classifying multiple geological formation top types. Additionally, the study employs the Density-Based Spatial Clustering of Applications with Noise (DBSCAN) algorithm as a baseline benchmarking technique to contextualize the relative performance of the machine learning models against a conventional clustering approach. Leveraging two model-agnostic feature importance analysis techniques - Permutation Feature Importance (PFI) and Shapley Additive exPlanations (SHAP), the investigation identifies and ranks the most influential input variables driving the predictive capabilities of the models. The comprehensive analysis unveils the MLP neural network model as the top-performing approach, achieving remarkable predictive accuracy with a perfect score of 0.99 on the blind validation dataset, surpassing the other machine learning techniques as well as the DBSCAN benchmark. However, the SVM model attains superior performance on the initial test dataset, with an accuracy of 0.99. Intriguingly, the PFI and SHAP analyses converge in consistently pinpointing depth (DEPT), revolution per minute (RPM), and Hook-load (HKLD) as the three most impactful parameters influencing model predictions across the different algorithms. These findings underscore the potential of sophisticated machine learning methodologies, particularly neural network-based models, to significantly enhance the accuracy of geological formation top prediction within the geologically complex NCS region. However, the study emphasizes the necessity for further extensive testing on larger datasets to validate the generalizability of the high performance observed. Overall, this research delivers an exhaustive comparative evaluation of state-of-the-art machine learning techniques, offering critical insights to guide the optimal selection, development, and real-world deployment of accurate and reliable predictive modeling strategies tailored for hydrocarbon exploration and reservoir characterization endeavors in the NCS.

✉ Behzad Elahifar
behzad.elahifar@ntnu.no

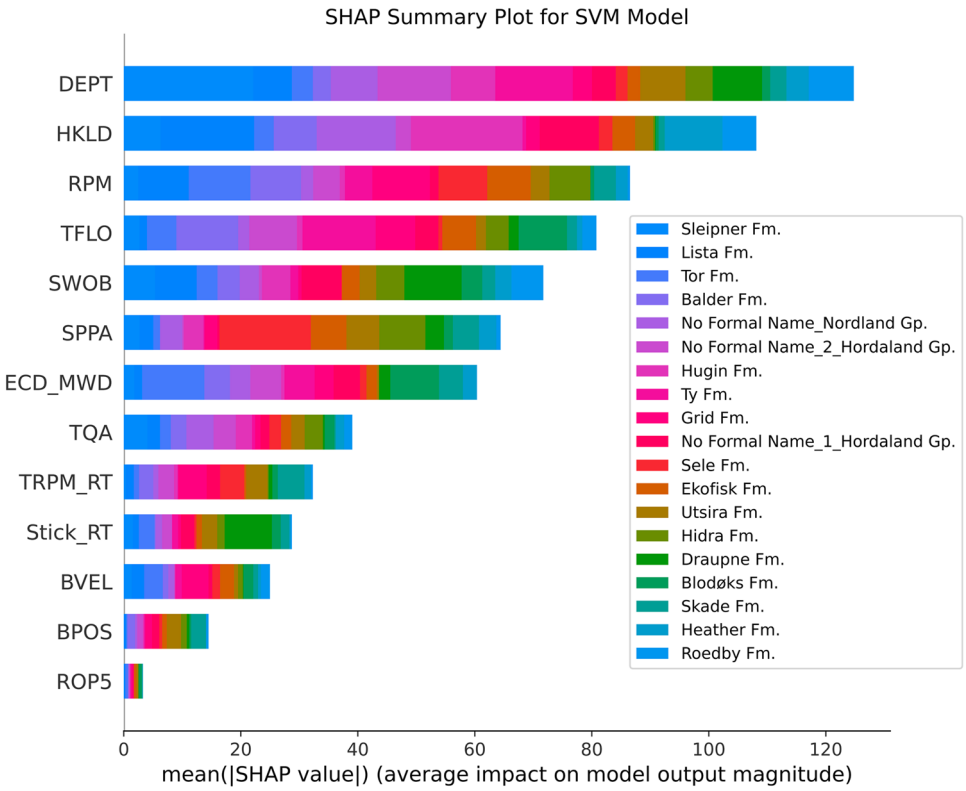
✉ Erfan Hosseini
e.hoseini@niocexp.ir

[1] Department of Geoscience and Petroleum, Norwegian University of Science and Technology (NTNU), Trondheim, Norway

[2] Exploration Directorate, National Iranian Oil Company (NIOC), Tehran, Iran

Published online: 08 April 2024

Graphical abstract



Keywords Geological formation tops prediction · Machine learning · Multi-layer perceptron · Support vector machine · K-nearest neighbors · Random forest · Density-based spatial clustering of applications with noise

Abbreviations

DEPT	Depth: bit depth in meters or measured depth (M)
ROP5	Rate of penetration; drilling progress in distance and time-averaged every 5 feet (M/HR)
BPOS	Traveling block position: the height at which the traveling block is located on the mast or derrick (M)
BVEL	Traveling block velocity: the velocity of movement of the block and the hoisting system (M/HR)
SWOB	Surface weight on bit: measurement of the weight exerted by the string on the bit and, therefore, on the formation (KKGF)
HKLD	Hook-load: measurement of the load on the hook by the working and drilling string (KKGF)
TQA	Torque: the torque exerted by TDS derived from string rotation in units of kilometers decanewtons (KMN)
RPM	Revolution per minute: measurement of the revolutions of the turbine contained in the BHA to energize downstream components (RPM)
TFLO	Total pump flow: the flow rate of drilling mud to well (LPM)
TRPM_RT	Bottom turbine revolutions: measurement of the revolutions of the turbine contained in the BHA to energize downstream components (RPM)
SPPA	Pump pressure; friction losses in the hydraulic system (BAR)
ECD_ARC	Equivalent circulating density: the density of the bottom-well fluid (SG)
Stick_RT	Stick and slip indicator; torsional vibration (RPM)
GFT	Geological formation top: the depth to the top of a geological formation (M)
AUC	Area Under the Curve
DBSCAN	Density-Based Spatial Clustering of Applications with Noise
DT	Decision tree

EDA	Exploratory data analysis
FPR	False Positive Rate
FN	False negative
FP	False positive
IF	Isolation Forest
KNN	K-Nearest Neighbors
LOF	Local Outlier Factor
NCS	Norwegian Continental Shelf
NaN	Non-numeric values
MLP	Multi-Layer Perceptron
MWD	Measurement while drilling
PFI	Permutation Feature Importance
RF	Random Forest
ROC	Receiver Operating Characteristic
SHAP	Shapley Additive exPlanations
SVM	Support Vector Machine
TPR	True Positive Rate
TN	True negative
TP	True positive

Introduction

The oil and gas industry generates vast amounts of data during operations, and applying machine learning algorithms to process this data has become increasingly important (Elkatatny 2018; Aniyom et al. 2022). Machine learning has been recognized as a promising tool in the oil and gas industry, with applications ranging from improving operational efficiency to predicting geological formations. Accurately predicting geological formation tops is an essential yet challenging task in hydrocarbon exploration and production activities (Mahmoud et al. 2020). Traditional manual interpretation of well logs to pick formation tops is labor intensive, prone to human subjectivity and errors, and unable to handle large datasets efficiently. This highlights the need for an automated and optimized approach. Applying machine learning for real-time prediction of geological formation tops using drilling data is a topic of significant interest in the oil and gas industry (Zhong et al. 2022). Several studies have explored different aspects of this topic, demonstrating the potential of machine learning to improve operational efficiency and reduce risks (Sircar et al. 2021; Losoya et al. 2021).

Al-AbdulJabbar et al. (2018) have developed a novel method for predicting formation tops in real-time that can replace more expensive techniques. Their method leverages drilling mechanics and rate of penetration data to identify lithological changes during drilling accurately. The authors gathered field data from two wells drilled with the same bit size and through the same formations. Data from Well A was used to train and test an artificial neural network (ANN) model (70% training, 30% testing), while data from Well B served as unseen test data. The optimized ANN model with one hidden layer and 20 neurons achieved high correlations of 0.94 and 0.98 on Wells A and B, respectively, demonstrating the method's ability to predict formation tops reliably. A key advantage of this approach is the real-time nature, as no log data processing or cuttings lag is required. By relying solely on low-cost, existing drilling data, formations can be accurately identified instantly without operational delays or expending resources on logs. This study highlights the potential for ANNs and drilling mechanics to enable rapid, precise, and inexpensive top detection during drilling. Mahmoud et al. (2021) developed artificial neural networks (ANN), adaptive neuro-fuzzy inference systems (ANFIS), and fuzzy neural network (FNN) models to predict lithology changes and formation tops during drilling operations. The models were trained on 3162 datasets across six input parameters. After optimization, the models were validated on 1356 datasets from a separate well. The ANN model achieved the highest accuracy, correctly predicting lithology distributions and formation tops for training and testing data over 98% of the time. Compared to ANFIS and FNN, the ANN model showed superior performance as a real-time predictive tool for lithology and formation changes during drilling. The study demonstrates the potential of ANN models to enable more informed decision-making and adjustments while drilling through multiple formations.

Vikara and Khanna (2022) developed an innovative framework to generate predictive models using various machine-learning classification algorithms. The goal was to identify specific stratigraphic units in the prolific Midland Basin of West Texas. After testing multiple algorithms, the random forest (RF) model achieved the highest prediction accuracy of 93% on holdout validation data. Notably, the RF model demonstrated exceptional performance in predicting major hydrocarbon-producing zones in the basin. This data-driven approach provides an accurate, cost-effective solution to complement traditional reservoir characterization methods across energy sector applications. Overall, the study by Vikara and Khanna establishes a robust framework leveraging machine learning for optimized subsurface analysis and resource identification. Ziadat et al. (2023) proposed a novel machine-learning approach for real-time detection of drilled formation tops and lithology types using only surface drilling data. They leveraged random forest and decision tree classifiers to develop highly accurate models predicting lithology from a dataset of five complex geological formations. Their methodology included rigorous data collection, preprocessing, exploratory analysis, feature engineering, model development, and hyperparameter tuning. Through comprehensive experiments, they demonstrated over 95% testing accuracy in lithology classification, even on intricate formation schemes. The study highlights the capability of machine learning techniques to enable real-time subsurface

lithology prediction solely from surface data. This could significantly enhance drilling efficiency and reduce costs by guiding the optimal, geology-specific selection of drilling parameters in real time without needing downhole measurements. The proposed data-driven methodology provides a broadly generalizable framework to unlock the full potential of surface data for real-time formation characterization.

Ibrahim et al. (2023) developed machine learning models from drilling data to accurately predict lithology and formation tops in real-time. They collected data from two wells in the Middle East and trained Gaussian naive Bayes (GNB), logistic regression (LR), and linear discriminant analysis (LDA) models. The GNB model demonstrated exceptional accuracy in predicting lithology, achieving near-perfect scores. The LR and LDA models also performed well, although LDA misclassified some carbonate/shale formations. During the new data validation, the models maintained high accuracies of 0.96, 0.95, and 0.92 for GNB, LR, and LDA, respectively. Their innovative modeling enables real-time rock type determination while drilling, allowing rapid geosteering decisions. Khalifa et al. (2023) have developed an innovative machine-learning approach for real-time lithology prediction during drilling operations. Using a dataset from the Volve field, they trained models to classify drilling data into three lithology classes—claystone, marl, and sandstone—with remarkable accuracy. Through careful preprocessing, including balancing the class distribution and reducing redundant features, they prepared an unbiased training set. Their best model achieved 95% overall testing accuracy and 98% average precision, demonstrating exceptional predictive performance. To enhance accessibility, they built GeoVision, an easy-to-use web application that allows drilling engineers to utilize the models on-site. This pioneering methodology and software tool enable more informed and rapid drilling decision-making, marking a significant step toward real-time geosteering. With rigorous methodology and testing, Khalifa et al. have set a high benchmark for lithology prediction from drilling data using machine learning. Their innovative integration of ML with drilling engineering promises to transform future drilling operations.

Challenges are involved despite the potential benefits of machine learning in the oil and gas industry. These include the need for high-quality data, the complexity of the algorithms, and the need for skilled personnel to interpret the results (Khalifah et al. 2020; Alsaihati et al. 2021). However, these challenges can be mitigated with continuous technological advancements and increasing adoption of machine learning. This study aims to develop a comprehensive machine-learning framework to accurately and reliably predict formation tops on the complex Norwegian Continental Shelf (NCS) using well-log data. Four main algorithms—support vector machines (SVM), random forest (RF), k-nearest neighbor (KNN), and multi-layer perceptron (MLP)—are implemented, optimized, and rigorously evaluated to determine the most suitable method for this problem. The methodology involves multiple stages. First, the dataset is preprocessed by handling missing values, outliers, and noisy data and applying techniques like normalization. Next, exploratory analysis uncovers patterns and relationships within the data. Optimal features are then extracted using statistical metrics and domain expertise. The cleaned dataset is split into training and test sets for model development and evaluation. The four machine learning algorithms are implemented with appropriate hyperparameters and configurations tailored to the formation top classification task. Models are optimized using techniques like grid search and cross-validation. Evaluation metrics such as accuracy, F1-score, precision, and recall quantify model performance. The best model is selected based on these metrics. Additional techniques like clustering using DBSCAN provide supplementary insights. This framework ensures accurate, reliable, optimized models while comprehensively understanding the data's underlying structure. By comparing multiple algorithms, their relative strengths and weaknesses are analyzed to identify the ideal approach for the NCS. The automated methodology overcomes human subjectivity and inefficiency. Accurate formation top picks have far-reaching impacts. They enhance subsurface geological models, minimize uncertainty, assist in assessing hydrocarbon potential, optimize drilling activities, and improve recovery strategies. This research enables geologists to focus on critical tasks rather than repetitive manual interpretation. The insights can inform data-driven decision-making to unlock value. In conclusion, this study develops an exhaustive machine learning approach for efficient, accurate, and automated formation top classification from well logs on the NCS. The results provide key insights into leveraging artificial intelligence to transform subjective processes into optimized, intelligent systems in the hydrocarbon industry. The methodology and findings significantly advance available techniques for subsurface characterization.

Approach and procedures

For this investigation, we gathered the dataset employed to predict the topography of geological formations on the Norwegian Continental Shelf (NCS) from a trustworthy data source. The dataset comprises a range of variables and features pertinent to predicting formation tops. These features play a vital role in discerning the characteristics and properties of various formations within the field. To guarantee the quality and reliability of the data, a sequence of preprocessing steps was undertaken. This encompassed addressing missing values, outliers, and other data quality issues. Furthermore, data normalization or standardization techniques were implemented to ensure uniformity

and comparability across the features. The exploratory data analysis (EDA) phase was crucial for comprehending the dataset and extracting insights into its characteristics. Summary statistics, encompassing measures like mean, median, mode, and standard deviation, were computed to offer a comprehensive overview of the data. These statistics aided in grasping central tendencies, dispersion, and variable distributions. Diverse data visualization techniques, such as histograms, heatmaps, box plots, and correlation matrices, were employed during EDA to scrutinize relationships and patterns within the dataset. These visualizations yielded valuable insights into variable distributions, potential outliers, and feature correlations.

During the exploratory data analysis (EDA) phase aimed at cleaning the dataset, the process involves the elimination or handling of data that is not valid, particularly those containing non-numeric values (NaN). It is crucial to identify invalid or missing data while analyzing the dataset. The approach removes such data from the dataset regarding non-numeric or NaN values. This step ensures the dataset's consistency and safeguards against invalid data's influence on subsequent analysis and modeling. Removing NaN or non-numeric data guarantees that the dataset comprises only valid numerical values, facilitating insightful analysis and accurate predictions. Nevertheless, it is crucial to emphasize that the decision to remove data should be executed judiciously and grounded in thoroughly comprehending the dataset. If the volume of NaN or non-numeric data is substantial or contains valuable information, alternative strategies like imputation techniques may be explored. Imputation involves filling in missing values with reasonable estimates and preserving data integrity. In summary, the EDA process in cleaning the dataset includes the option to eliminate data with non-numeric values (NaN) or non-numeric data. However, the decision-making process should consider the overall impact on the dataset. If warranted, alternative methods like imputation can be employed to maintain data completeness and enable more precise analysis and modeling.

Feature selection or extraction techniques were implemented to pinpoint the most pertinent features for real-time prediction of geological formation tops. This process included scrutinizing the relationship between parts and the target variable using statistical measures or domain knowledge. The identified features were subsequently utilized as inputs for machine learning models. The study incorporated various machine learning algorithms, such as multi-support vector machines (SVM), random forest, k-nearest neighbors (KNN), and multilayer perceptron (MLP). Each algorithm was instantiated with specific configurations and hyperparameters tailored to the real-time prediction of geological formation tops task. For example, the SVM algorithm employed a selected kernel and appropriate regularization parameters, while random forest had a designated number of trees and splitting criteria. Relevant evaluation metrics were utilized to gauge the performance of the models. Metrics, including accuracy, precision, recall, and F1-score, were computed to appraise the predictive prowess of each model in accurate real-time prediction of geological formation tops. These evaluation metrics furnished a quantitative gauge of the model's performance, enabling comparisons between machine-learning algorithms. The experimental setup encompassed a train-test split ratio, dividing the dataset into segments for model training and evaluation. A portion was allocated for training the models, while the remaining was for testing and assessing their performance. Cross-validation techniques might have been employed to validate the models' generalizability. Furthermore, statistical analyses could have been conducted to validate the results or compare the performance of various machine learning models. These analyses would offer additional insights into the findings' significance and the predictive models' reliability. The materials and methods outlined in this study were designed to guarantee the robustness, reproducibility, and validity of real-time prediction of geological formation top prediction models employing various machine learning algorithms. Integrating exploratory data analysis (EDA) techniques, data preprocessing procedures, diverse machine learning models, and evaluation metrics constituted a comprehensive framework, facilitating the attainment of precise and dependable predictions for geological formation tops in the NCS.

The flowchart depicted in Fig. 1 for "Modeling Geological Formation Tops Prediction in the NCS: A Machine Learning Approach using Multi SVM, Random Forest, KNN, MLP" offers a systematic guide for constructing and assessing predictive models. The process initiates with the importation of necessary libraries and the loading of the dataset. Subsequently, the dataset is divided into a training set and a blind set for model training and evaluation. Exploratory data analysis (EDA) is then executed to glean insights from the dataset, followed by multivariate data analysis to unveil intricate relationships. The workflow includes essential data preprocessing tasks, encompassing missing values, treatment of outliers, and resolution of data duplicates. Collinear independent variables are eliminated to prevent potential issues in model performance. Scaling and normalization techniques are then employed to ensure comparability among features. The definition of feature and output matrices for model training involves selecting specific inputs from the dataset. In this case, the chosen feature inputs are DEPT, ROP5, HKLD, SWOB, TQA, RPM, BPOS, BVEL, SPPA, TFLO, TRPM_RT, Stick_RT, ECD_MWD from the Measurement While Drilling (MWD) data. The target matrix is specified as GFT. This compilation forms the basis for training the model. The details of real-time measurement while drilling (MWD) records employed in this study are outlined in

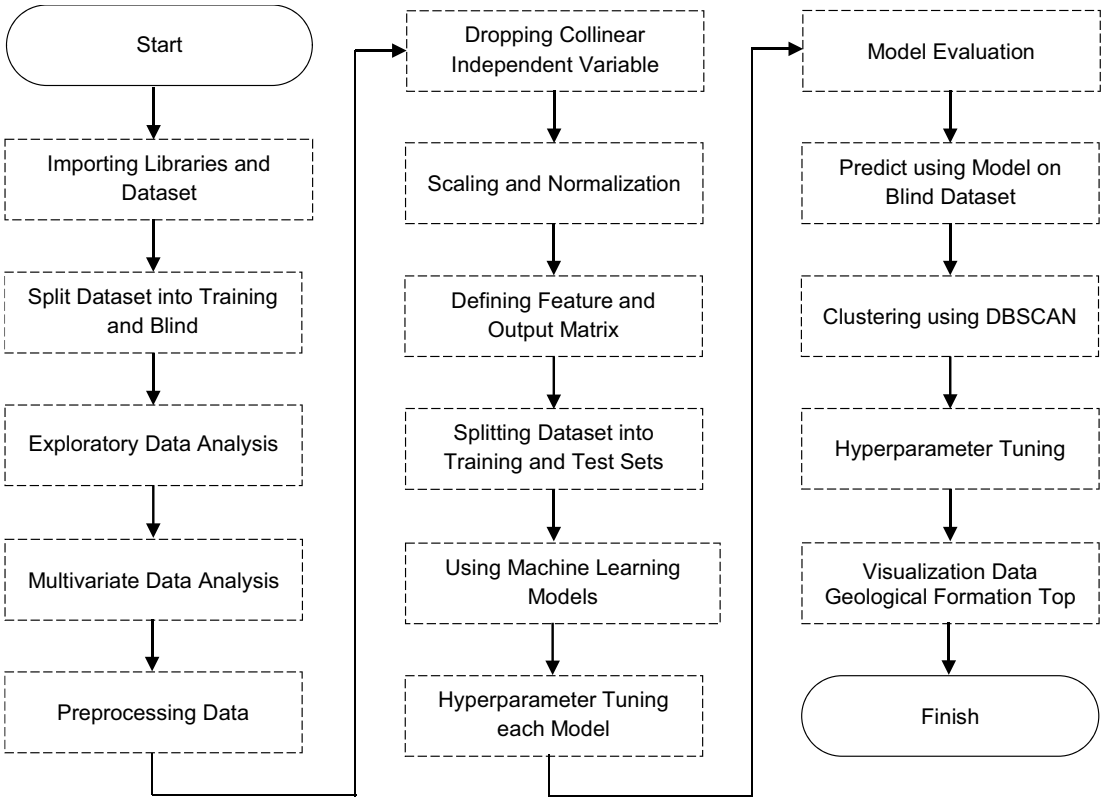


Fig. 1 Flowchart for modeling geological formation top prediction in the NCS

Table 1 Details of the dataset utilized in this model

Drilling data	Description
DEPT	Depth: bit depth in meters or measured depth (M)
ROP5	Rate of penetration; drilling progress in distance and time-averaged every 5 feet (M/HR)
BPOS	Traveling block position: the height at which the traveling block is located on the mast or derrick (M)
BVEL	Traveling block velocity: the velocity of movement of the block and the hoisting system (M/HR)
SWOB	Surface weight on bit: measurement of the weight exerted by the string on the bit and, therefore, on the formation (KKGF)
HKLD	Hook-load: measurement of the load on the hook by the working and drilling string (KKGF)
TQA	Torque: the torque exerted by TDS derived from string rotation in units of kilometers decanewtons (KMN)
RPM	Revolution per minute: measurement of the revolutions of the turbine contained in the BHA to energize downstream components (RPM)
TFLO	Total pump flow: the flow rate of drilling mud to well (LPM)
TRPM_RT	Bottom turbine revolutions: measurement of the revolutions of the turbine contained in the BHA to energize downstream components (RPM)
SPPA	Pump pressure; friction losses in the hydraulic system (BAR)
ECD_ARC	Equivalent circulating density: the density of the bottom-well fluid (SG)
Stick_RT	Stick and slip indicator; torsional vibration (RPM)
GFT	Geological formation top: the depth to the top of a geological formation (M)

Table 1. The dataset is divided into training and test sets, with a test size of 0.2 (20%), a training set of 0.8 (80%), and a random state set to zero for reproducibility. Various machine learning algorithms, including KNN, Random Forest, SVM, and MLP, are applied to construct predictive models. Hyperparameter tuning is executed to optimize the performance of each model. The model evaluation uses appropriate metrics, and the top-performing models

make predictions on the blind dataset. Furthermore, clustering analysis utilizing the DBSCAN algorithm might be incorporated to uncover inherent groupings within the data. Hyperparameter tuning could be employed to optimize the DBSCAN clustering process. Visualizations of the distribution of various formation tops are created to extract insights. Ultimately, the flowchart concludes the modeling and evaluation process. In summary, this systematic approach ensures the creation of precise formation top prediction models while attaining a thorough understanding of the dataset.

Dataset utilized in the study

The dataset employed in the modeling process comprises two distinct sets. The initial dataset serves as the training set, while the second is a blind set. The latter is utilized for validation and predicting the model trained on the initial dataset. The dataset used in this experiment encompasses real-time drilling data from two wells, denoted Wells A and B. Well A serves as the training dataset, while well B functions as the blind dataset for validation. The dataset comprises 14 measurement while drilling (MWD) parameters, including DEPT, ROP5, BPOS, BVEL, SWOB, HKLD, TQA, RPM, TFLO, TRPM_RT, SPPA, ECD_ARC, Stick_RT, and GFT. A detailed breakdown of these parameters is presented in Table 1. Also, Table 2 shows the descriptive statistics summarizing the characteristics of both input features and the target variable in Dataset A. Similarly, Table 3 provides descriptive statistics for the input features and target variable in Dataset B.

Preprocessing the dataset

The preprocessing phase in machine learning modeling, aimed at eliminating datasets containing NaN values or no data, involves a series of steps. Initially, the dataset is examined to identify the location and quantity of NaN values. Subsequently, the presence of NaN values is assessed, considering their pattern or impact on the data and the modeling objective. If NaN values are deemed insignificant or cannot be resolved through proper filling techniques, the subsequent step involves removing the rows or columns containing NaN values using the '.dropna()' method. Following the elimination, the dataset is reassessed to ensure that the quantity and distribution of the remaining data remain sufficient and representative. Evaluating the impact of NaN value elimination on class balance or target distribution within the model is crucial. Additionally, it is essential to verify the dataset index after the removal of NaN data. When making this decision, carefully considering the dataset's context and characteristics is vital, as eliminating NaN data can influence the quantity and representation of the available data.

Identifying outliers using automated methods

Outliers are anomalous points within a dataset. They are points that do not fit within the normal statistical distribution of the dataset and can occur for various reasons, such as sensor and measurement errors, poor data sampling techniques, and unexpected events. Within MWD logs, outliers can occur due to washed-out boreholes, tool and sensor issues, rare geological features, and issues in the data acquisition process. These outliers must be identified and investigated early in the workflow, as they can result in inaccurate predictions by machine learning models. Several

Table 2 Descriptive statistics for input and target characteristics in dataset A (training dataset)

	Count	Mean	std	Min	25%	50%	75%	Max
DEPT	82,592.0	2681.969386	1000.799976	224.7600	1818.061625	2939.83155	3606.2896	4089.8755
ROP5	82,592.0	23.798850	133.405723	1.1169	9.960000	14.74000	25.0476	2993.1134
HKLD	82,592.0	129.223147	10.377527	65.6028	121.390000	129.56000	137.2200	149.5600
SWOB	82,592.0	6.393388	5.122745	0.0000	3.075075	4.68640	8.7100	52.0333
TQA	82,592.0	14.312507	3.705972	0.0000	12.090000	14.86000	16.8000	34.2722
RPM	82,592.0	163.750630	73.287297	0.0000	119.000000	179.00000	235.0000	298.0000
BPOS	82,592.0	26.342694	12.586083	0.6052	15.758875	26.90000	37.1300	50.0000
BVEL	82,592.0	0.005476	0.010397	0.0000	0.000000	0.00270	0.0100	0.2800
SPPA	82,592.0	186.472985	38.183174	5.5223	154.800000	195.97145	216.6900	270.1300
TFLO	82,592.0	2706.819663	1012.302221	199.3929	1794.540000	2016.08000	3987.8572	4187.2499
TRPM_RT	82,592.0	2878.781166	547.679637	0.0000	2343.750000	2734.38000	3593.7500	4960.9400
Stick_RT	82,592.0	62.940830	74.653558	0.0000	15.000000	24.00000	84.0000	381.0000
ECD_MWD	82,592.0	1.388246	0.170141	0.0100	1.400000	1.42000	1.4600	15.7900
GFT	82,592.0	10.525220	5.055632	0.0000	7.000000	13.00000	15.0000	17.0000

Table 3 Descriptive statistics for input and target characteristics in dataset B (blind dataset)

	Count	Mean	std	Min	25%	50%	75%	Max
DEPT	48,323.0	2473.357415	902.311946	217.8902	1863.07900	2716.4544	3193.6215	3792.1993
ROP5	48,323.0	26.111682	17.122276	0.7822	11.56750	25.1238	39.6359	157.6270
HKLD	48,323.0	115.194651	6.837686	61.2415	111.19590	115.3996	119.9716	132.5865
SWOB	48,323.0	5.744478	3.393747	0.0000	3.20070	5.2920	7.9933	45.4156
TQA	48,323.0	18.208309	7.066991	0.0010	14.84250	19.0013	23.2835	35.2354
RPM	48,323.0	118.601701	51.262144	0.0000	79.00000	130.0000	150.0000	263.0000
BPOS	48,323.0	22.360620	12.341883	0.9826	11.94970	21.9808	33.3951	49.3230
BVEL	48,323.0	0.009642	0.013409	0.0000	0.00280	0.0070	0.0112	0.1938
SPPA	48,323.0	191.120540	41.452504	2.5690	170.81070	206.8920	214.6618	279.1792
TFLO	48,323.0	2941.754206	714.668730	686.7976	2104.70230	3456.1430	3500.4523	3987.8570
TRPM_RT	48,323.0	2922.623247	295.603948	312.5000	2890.62500	2968.7500	3125.0000	4960.9380
Stick_RT	48,323.0	70.586822	88.454134	0.0000	12.00000	27.0000	87.0000	381.0000
ECD_MWD	48,323.0	1.437650	1.253808	0.0090	1.37595	1.4523	1.5549	42.5220
GFT	48,323.0	10.510192	5.757096	0.0000	5.00000	12.0000	13.0000	23.0000

Table 4 Outlier elimination results for isolation forest (IF), one class SVM (SVM), and local outlier factor (LOF) methods on training dataset (Dataset A) and Blind Dataset (Dataset B)

	Outlier method	Isolation forest (IF)	One class SVM (SVM)	Local outlier factor (LOF)
Dataset A	Anomalous values	24,778	24,777	21,886
	Non-anomalous values	57,814	57,815	60,706
	Total values	82,592		
Dataset B	Anomalous values	14,494	14,494	13,013
	Non-anomalous values	33,829	33,829	35,310
	Total values	48,323		

unsupervised machine-learning methods can be used to identify anomalies/outliers within a dataset. In this study, we will look at three common ways: Isolation Forest (IF), One Class SVM (SVM), and Local Outlier Factor (LOF). Table 4 provides a detailed summary of the outcomes obtained from applying three distinct outlier elimination methods—Isolation Forest (IF), One Class SVM (SVM), and Local Outlier Factor (LOF)—to two datasets: the training dataset (Dataset A) and the blind dataset (Dataset B). Table 4 illustrates the number of abnormal and non-anomalous values identified by each method for both datasets. For Dataset A, the total values for each technique, including the sum of anomalous and non-anomalous instances, are displayed. Similarly, for Dataset B, the corresponding totals are presented. This comprehensive overview enables a comparative analysis of the effectiveness of the outlier elimination methods across the two datasets, offering insights into their performance in identifying and handling anomalous values in different contexts. The performance of each of the models using Surface weight on bit -Torque cross-plots is shown in Fig. 2.

The IF method provides a better result, followed by SVM and LOF. The first two methods remove most outliers on the plot's right-hand side. Figure 3 presents a comprehensive analysis of the training dataset before and after outlier removal using the isolation forest (IF) method. Subfigure (a) displays the boxplot of the training data before outlier removal, providing insights into the distribution and potential presence of outliers. Subfigure (b) showcases the boxplot after successfully applying the IF outlier removal method, highlighting the impact on the dataset's distribution. Removing outliers contributes to a more refined representation of the training data. Subfigure (c) supplements the analysis by presenting a count of the outliers released, categorized by geological formation top names and outlier detection method. This breakdown provides a detailed understanding of the specific formations affected and the effectiveness of the IF method in successfully eliminating outliers from the dataset. The improved data quality resulting from the IF outlier removal enhances subsequent analyses' and modeling efforts' robustness and reliability.

Impacts of outlier removal on detection, range determination, and confidence Outlier removal can have significant impacts that require careful consideration when developing machine learning models. Here is a summary of the outlier removal effects on detection, range determination, and confidence:

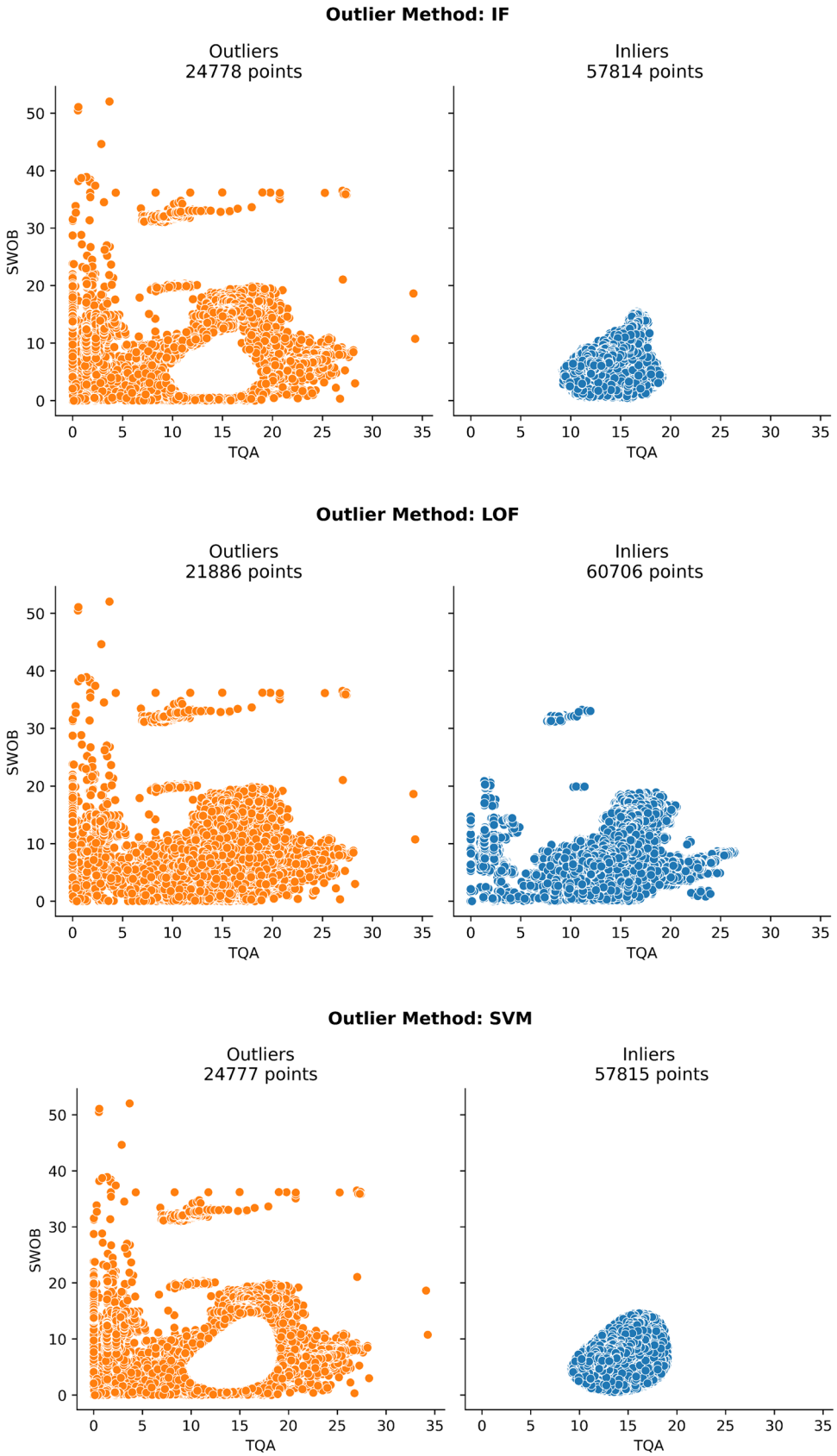


Fig. 2 Outlier detection for training dataset using Isolation Forest (IF), local outlier factor (LOF), and one-class SVM (SVM) methods

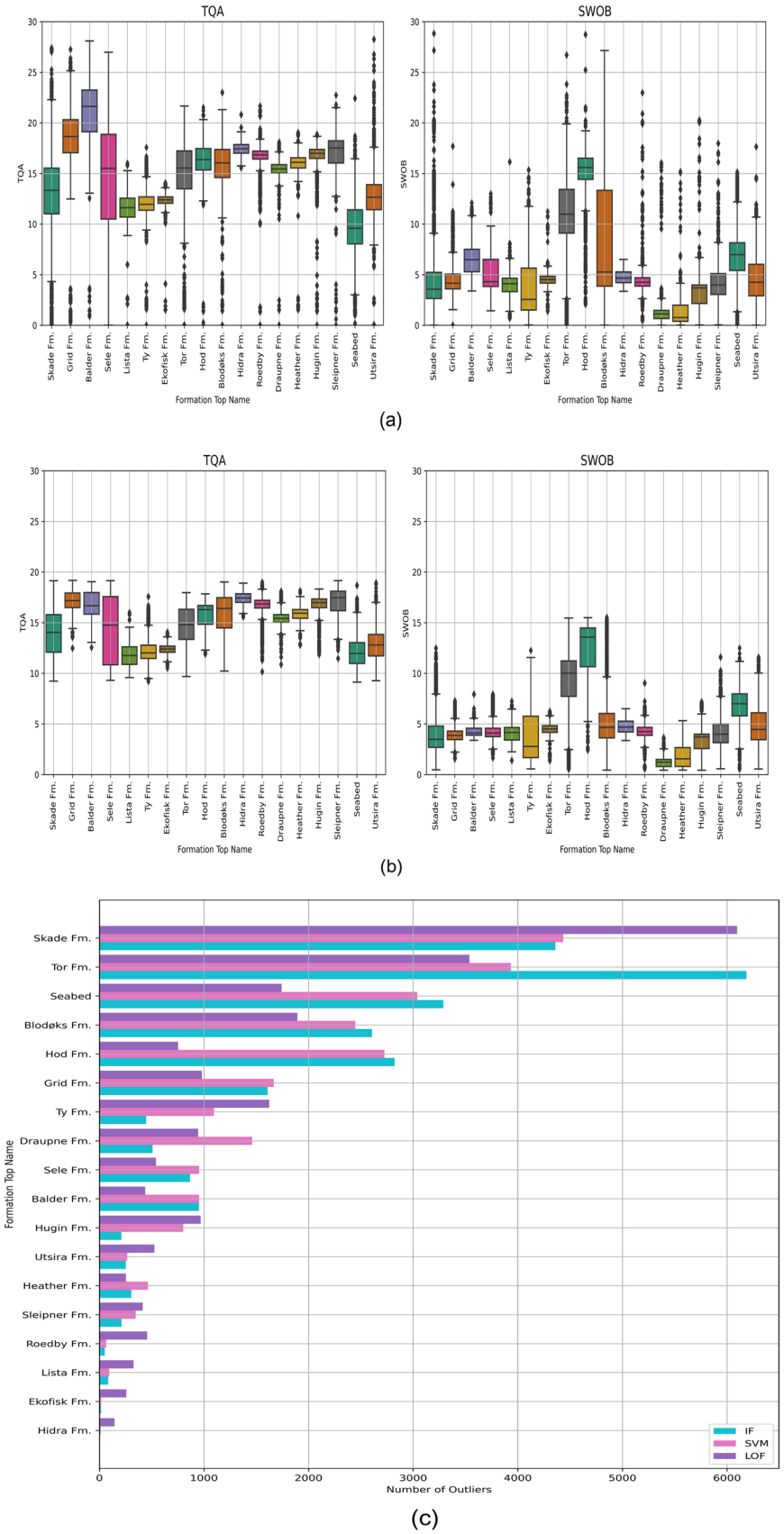


Fig. 3 Training data boxplot before outlier removal (**a**), training data boxplot after IF outlier removal (**b**), removed outliers' counts by formation top name and outlier detection method (**c**)

- Three outlier removal methods were tested—Isolation Forest (IF), One Class SVM (SVM), and Local Outlier Factor (LOF). They were applied on both the training dataset (Dataset A) and the blind validation dataset (Dataset B).
- For the training set (Dataset A), IF removed 24,778 outliers, SVM removed 24,777 outliers, and LOF removed 21,886 outliers. So IF and SVM detected a similar number of outliers, more than LOF.
- For the blind dataset (Dataset B), IF removed 14,494 outliers, SVM removed 14,494 outliers, and LOF removed 13,013 outliers. Again, IF and SVM detected nearly the same number of outliers, while LOF found fewer outliers.
- Visual analysis of SWOB vs TQA plots before and after IF outlier removal (Figs. 2 and 3) shows that applying IF helps eliminate most outliers, resulting in a cleaner distribution. This was the motivation to select IF for outlier elimination.
- By removing outliers, the data distribution becomes less skewed and more refined. This enhances the robustness and reliability of subsequent analysis and modeling using the "cleaned" dataset after IF outlier removal.

In summary, Isolation Forest (IF) was found to perform well in identifying and eliminating outliers from both the training and blind datasets. Applying IF outlier removal led to improved data quality and distributions for further analysis. This highlights the importance of detecting and handling outliers prior to applying machine learning algorithms. Therefore, utilizing multiple detection techniques, validating across datasets, inspecting distribution shifts visually, and incorporating domain expertise improves confidence that suitable outliers were identified and handled.

Encoding of geological formation tops

Encoding the categorical target variable, representing geological formation tops, is essential for modeling. The original formation names have been mapped to numerical values for computational efficiency. Table 5 presents the geological formation dictionary used for encoding:

In the dataset, each instance of the geological formation top is represented by the corresponding formation number. This encoding allows for seamless integration of the target variable into machine learning models, ensuring compatibility with various algorithms. Using numerical representations enhances computational efficiency and aids in interpreting model predictions. This encoding scheme will be employed throughout the subsequent modeling and analysis phases, providing a standardized and efficient representation of geological formation tops.

Table 5 Geological formation top dictionary

Formation top name	Formation top number
Balder Fm	Class 1
Blodøks Fm	Class 2
Draupne Fm	Class 3
Ekofisk Fm	Class 4
Grid Fm	Class 5
Heather Fm	Class 6
Hidra Fm	Class 7
Hugin Fm	Class 8
Lista Fm	Class 9
No Formal Name_1_Hordaland Gp	Class 10
No Formal Name_2_Hordaland Gp	Class 11
No Formal Name_Nordland Gp	Class 12
Roedby Fm	Class 13
Sele Fm	Class 14
Skade Fm	Class 15
Sleipner Fm	Class 16
Tor Fm	Class 17
Ty Fm	Class 18
Utsira Fm	Class 19

Feature selection

Feature selection in machine learning modeling often involves utilizing heatmaps to examine the correlation among independent variables. The initial step entails computing the correlation matrix for the independent variables within the dataset. Subsequently, the correlation results are presented visually through a heatmap (see Fig. 4), where bright colors represent high correlation levels and dark colors indicate low correlation levels. At this juncture, pairs of variables exhibiting substantial correlation, typically surpassing a predetermined threshold, are identified. Such a high correlation signifies redundancy in information. Following this, the task is to determine which variables should be eliminated within the identified pairs. This decision-making process considers factors such as domain relevance, model interpretation, and the contribution of variables to the overall model. Variables deemed more crucial or informative are retained, while those with lower or less significant contributions are excluded.

Referring to Fig. 4, the attributes exhibiting the most substantial absolute correlations include 'HKLD,' 'DEPT,' 'TQA,' and 'ECD_MWD.' These features demonstrate strong correlations with our target variable, GFT. After removing the variables, the machine learning model is retrained using the selected feature subset, and its performance is evaluated. It should be emphasized that heatmaps and correlations only provide an initial overview, and additional steps such as

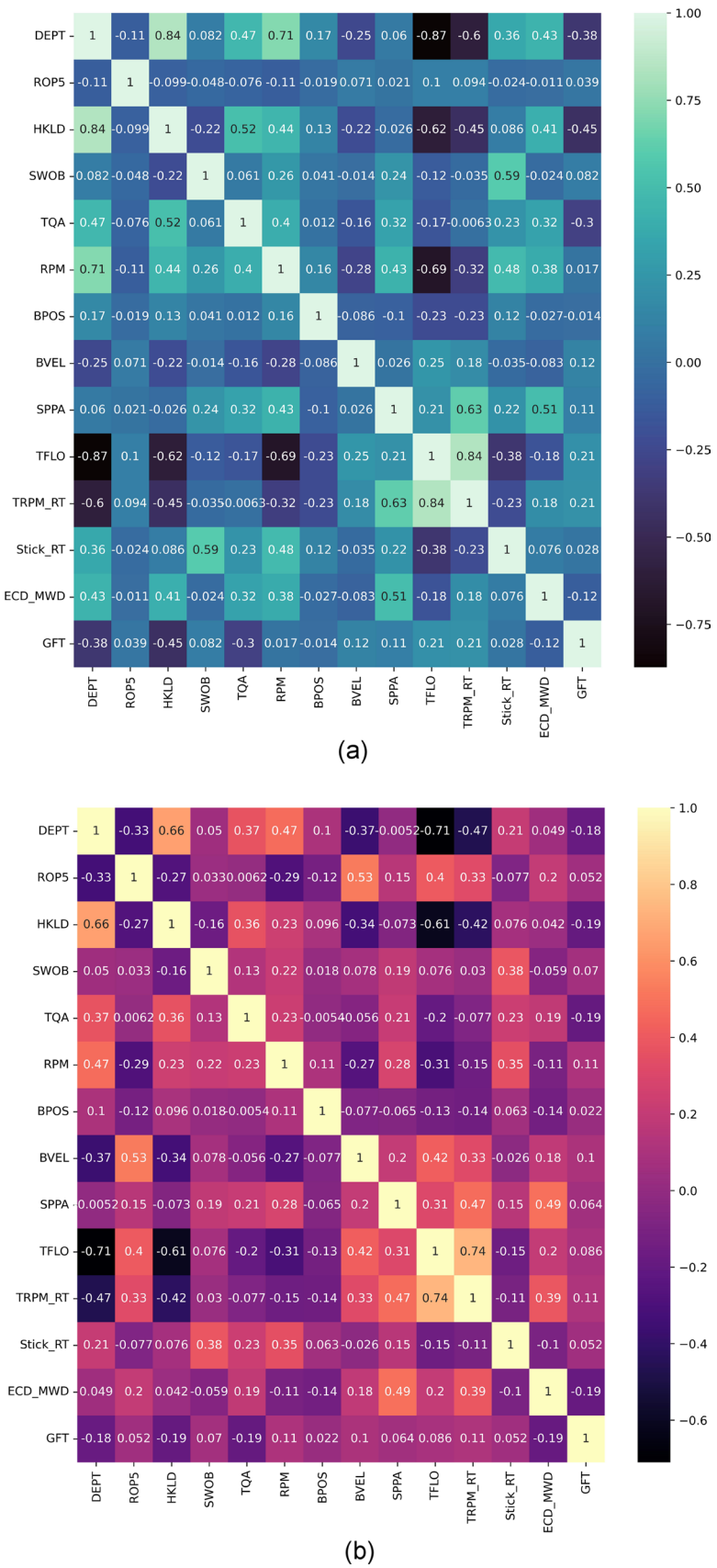
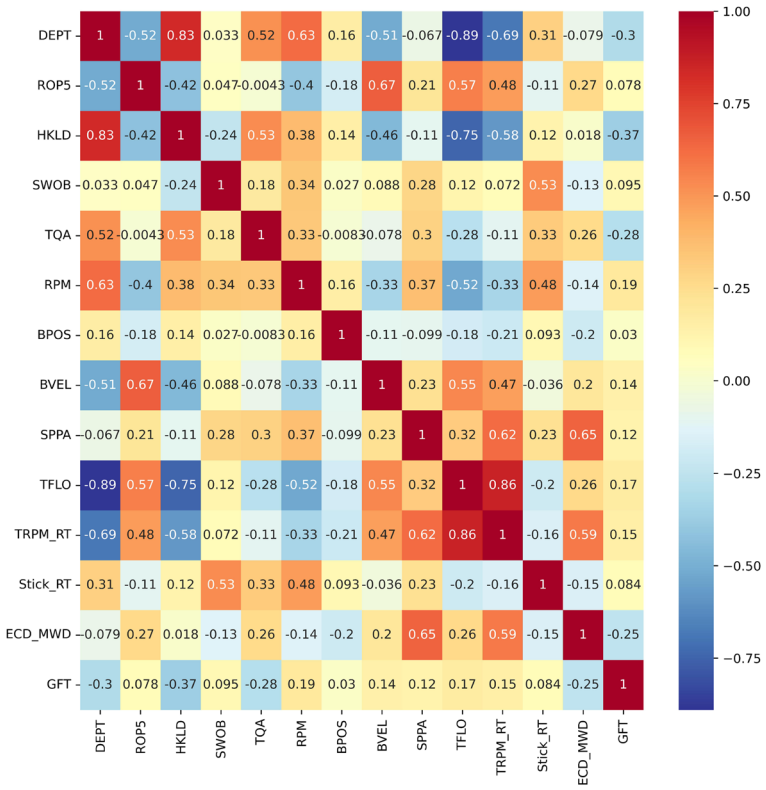


Fig. 4 Heatmap illustrating the associations between input variables and targets using different correlation measures: **a** Pearson correlation, **b** Kendall correlation, **c** Spearman correlation

Fig. 4 (continued)



(c)

statistical testing or other feature selection methods may be required to validate the feature selection decisions made. In this modeling process, the variables used as input are DEPT, ROP5, HKLD, SWOB, TQA, RPM, BPOS, BVEL, SPPA, TFLO, TRPM_RT, Stick_RT and ECD_MWD. While the variable set as output or target is GFT. More details about separate features and target matrix are shown in Fig. 5.

The stacked bar chart in Fig. 5 shows the correlation coefficients between different geological features/formations and the target variable. The features are sorted from highest to lowest mean correlation on the x-axis. The stacked bar chart indicates that Hook-load has the strongest positive correlation with the target variable, with an average correlation coefficient of around 0.8. This suggests that a Hook-load could serve as a key predictive feature for the target. In addition, bit depth, torque, and Equal circulating density display moderately strong positive correlations on average, with coefficients ranging from 0.6 to 0.7. These three formations also appear to have good predictive relationships with the target that could be utilized. On the other hand, Bottom turbine revolutions and Total pump flow demonstrate weaker but still positive correlations around 0.4–0.5, meaning their signals may retain some useful information. However, the remaining features exhibit low or negative average correlations, like traveling block velocity, RPM, etc. These formations likely have minimal or no predictive relationship with the target. In summary, the analysis indicates that hook load, bit depth, torque, and equivalent circulating density should be the primary features focused on for modeling. At the same time, Bottom turbine revolutions and Total pump flow may provide secondary signals, and the remaining components can likely be excluded from the predictive modeling. A multivariate perspective on the relationships between the selected input features and the target geological formation tops is provided through the pair-plot visualization in Fig. 6. Each subplot depicts the two-dimensional distribution between a feature pair, with points colored by formation top class. Distinct clustering by class is observed for variables including Bit depth, Equivalent circulating density, Hook-load, and Torque, indicating their efficacy in discriminating between formation tops. Based on the diagonal patterns, strong positive correlations are evident between bit depth, equivalent circulating density, bit depth, hook load, and equivalent circulating density and hook load. RPM exhibits significant overlap between multiple classes, implying limited differentiation capability. The pair-plot enables an assessment of the relevance and interrelationships of the selected features in predicting the target formation tops. These insights guide appropriate algorithm selection and parameter tuning to maximize classification performance.

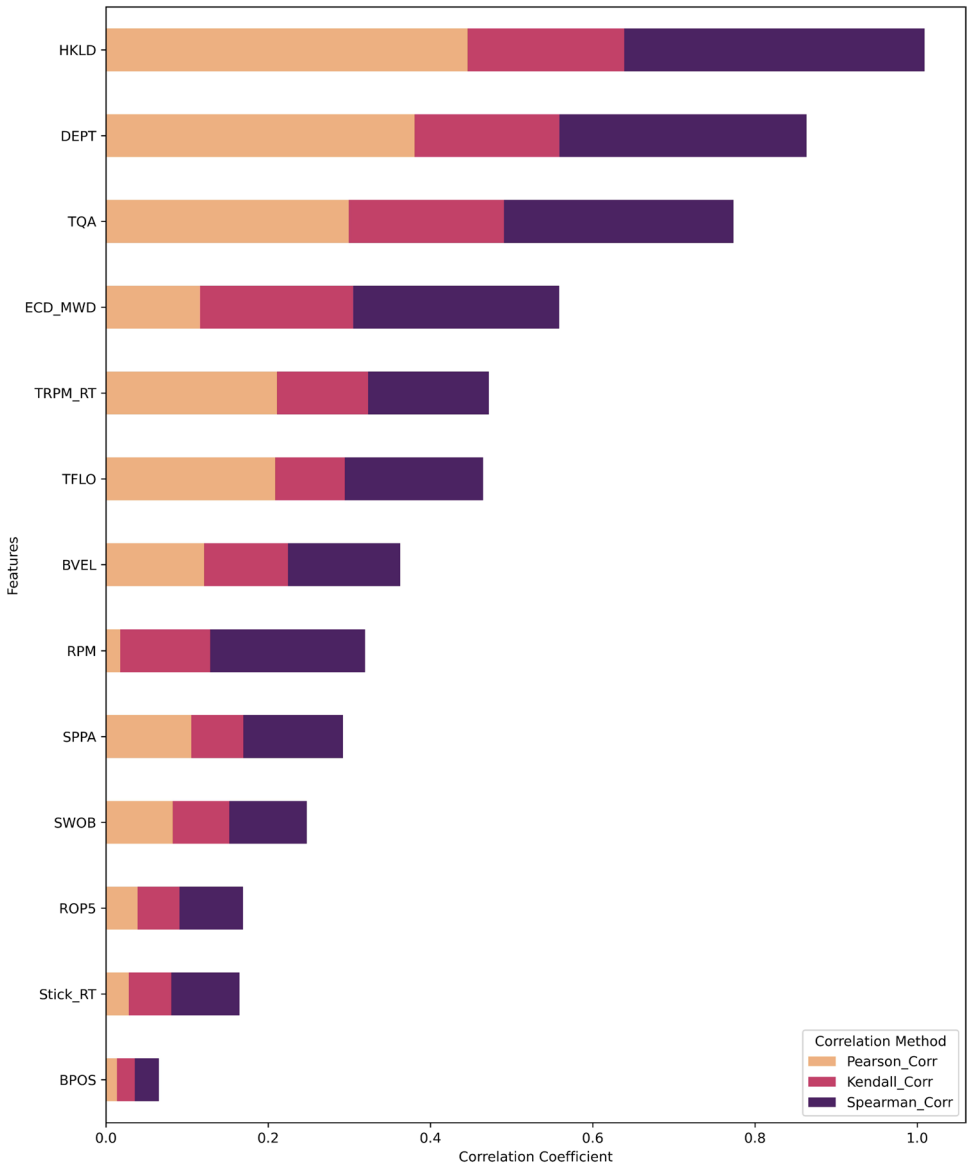


Fig. 5 Stacked bar chart of correlation coefficients for different features (sorted by mean correlation)

Several insights can be extracted from Fig. 6:

- There is a clear separation between many of the formation tops based on the variable pairs. For example, the Hod and Lista formations (light green and purple points) separate from other tops on axes like DEPT vs SWOB.
- Some formations demonstrate more compact, concentrated clusters (e.g., Sele in red), while others show greater dispersion (e.g., Tor in dark green). This indicates heterogeneity within formations.
- DEPT, ECD_MWD, HKLD, and TQA display distinct trends and clustering by formation top, suggesting they provide good discrimination. Others, like RPM, show a high overlap between classes.
- Based on the diagonal clusters, strong positive correlations are visible between DEPT and ECD_MWD, DEPT and HKLD, and ECD_MWD and HKLD. This is expected due to the direct relationships between these features.
- Looking at the DEPT vs. ECD_MWD subplot, we see that points belonging to the Hod formation (light green) cluster in the upper left region while points from the Lista formation (purple) fall in the lower right area. This indicates that the Hod formation tends to have higher depth (DEPT) and lower equivalent circulating density (ECD_MWD) than the Lista formation.
- In the DEPT vs. SWOB plot, the cloud of points corresponding to the Sele formation (red) occupies a narrow range of low surface weight on bit (SWOB) values across depths, suggesting consistent rock strength. In contrast, the Balder formation points (black) are more dispersed in SWOB for a given depth, implying greater heterogeneity.

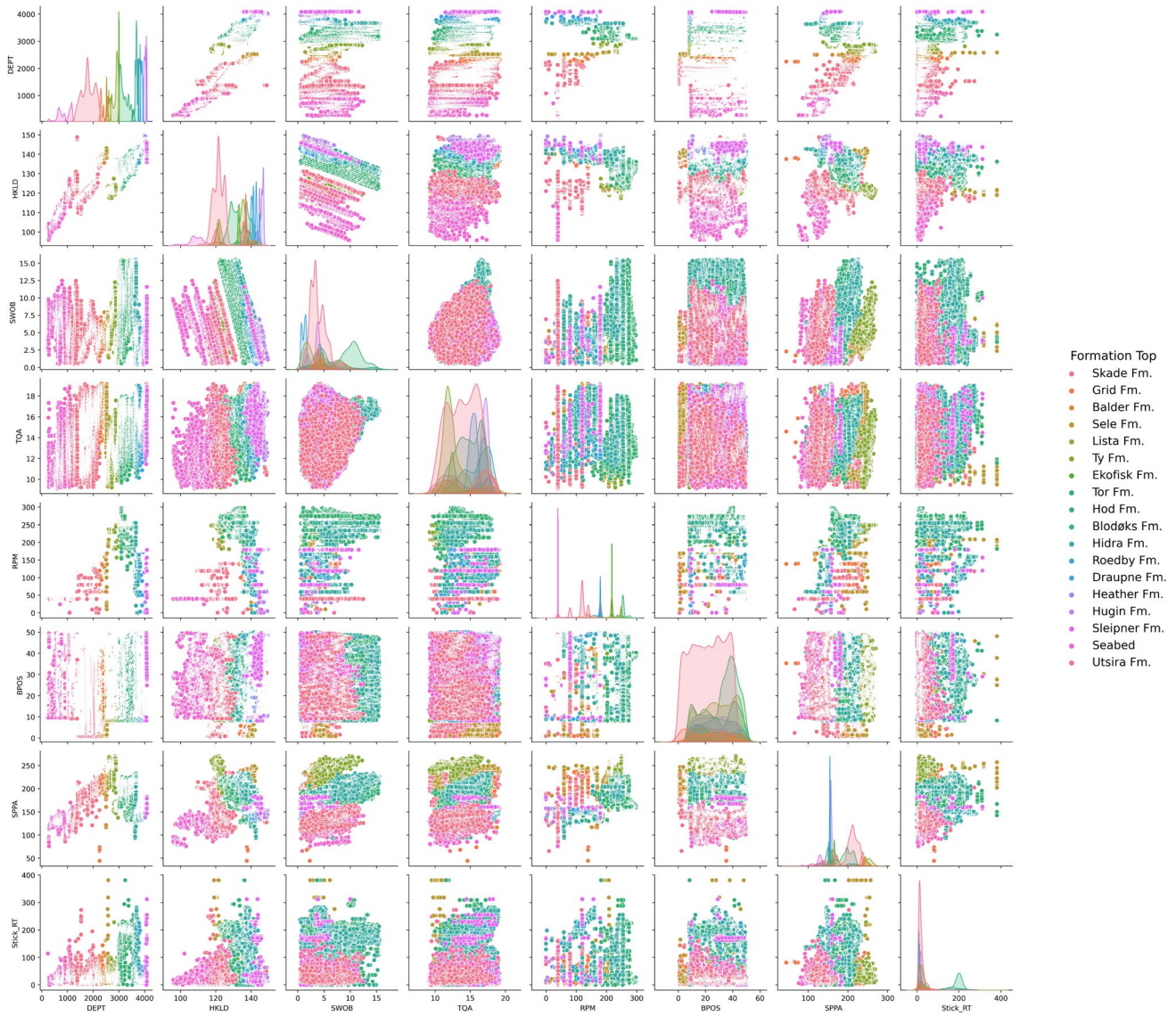


Fig. 6 Pair-plot results of data distribution per geological formation top

- For TQA vs. RPM, we see extensive overlap between many formations at lower torque (TQA) values. But at higher torque, formations like Ty (orange) and Draupne (blue) separate from others, likely due to differences in rock properties that impact drilling torque requirements.
- The HKLD vs. SPPA plot exhibits distinct clustering by formation top but also some overlap between formations with similar compressive strengths that dictate the hook-load (HKLD) and pump pressure (SPPA) relationship during drilling.
- Comparing SWOB vs. HKLD and SWOB vs. SPPA, we see diagonal patterns indicating a positive correlation between these features. Higher weight on the bit leads to higher hook load and pump pressure.

Based on a visual examination of the pair plot in Fig. 6, there appear to be some linear relationships between certain parameters:

- o Depth (DEPT) and Equivalent Circulating Density (ECD_MWD) show a strong positive linear correlation, as evidenced by the diagonal elongated cluster in their subplot. As depth increases, ECD also increases.
- o Similarly, Depth (DEPT) and Hookload (HKLD) display a positive linear relationship, with increasing depth associated with higher hookload values.
- o Hookload (HKLD) and Pump Pressure (SPPA) also seem to have an approximate positive linear association, though the relationship looks weaker.

In contrast, many of the other parameter combinations do not demonstrate strong linear relationships:

- The variables Surface Weight on Bit (SWOB) and Depth (DEPT) do not appear to have a clear linear correlation, with substantial scatter in their subplot.
- Revolution per Minute (RPM) vs Torque (TQA) also does not show a definitive linear trend, with overlapping classes spanning a wide range of values.
- Other pairs, like DEPT vs. SWOB, TQA vs. SPPA, etc., lack distinct linear patterns.

In addressing the inherent challenges posed by the presence or absence of linear relationships between parameters in machine learning, this study employs a diverse set of models, including support vector machines (SVM), random forests (RF), k-nearest neighbors (KNN), and multilayer perceptron (MLP). The intricate balance between linear and nonlinear relationships in the data is effectively managed by leveraging these diverse algorithms. The risk of overfitting, particularly prevalent in strong linear associations, is mitigated by the ensemble nature of Random Forests, preventing fixation on specific patterns. Similarly, support vector machines and multilayer perceptron models excel in handling complex, nonlinear relationships. By incorporating k-nearest neighbors, the models collectively reduce underfitting concerns, ensuring a nuanced capture of underlying data intricacies. This comprehensive approach not only addresses the challenges associated with linear and nonlinear patterns but also results in the highest accuracy in prediction across various scenarios.

Splitting training dataset into training and test set

We are utilizing the train_test_split function from Sklearn. model_selection, we partition the dataset into training and test sets, with the test set comprising only 20% of the data. During this partitioning, the random_state parameter is either turned off or set to 0 to ensure consistency and prevent variations in model outcomes.

Dataset standardization

Standardizing a dataset within the context of machine learning modeling holds significant importance. This process, facilitated by the StandardScaler library, involves converting the variables in the dataset to a consistent scale, thereby eliminating potential scale-related discrepancies that might impact model performance. Such differences can adversely affect algorithms sensitive to scale variations, such as those relying on distance or gradient-based methods. Utilizing StandardScaler ensures that the variables in the dataset are transformed to possess a mean of zero and a standard deviation of one, ensuring uniform scaling across all variables. This standardization facilitates faster algorithm convergence and enhances overall model stability. Moreover, it simplifies model interpretation by providing clear interpretations for variable coefficients. Standardization also mitigates outliers' influence on large-scale variables, improving model stability and accuracy.

Classification algorithms and parameter tuning

Optimal tuning of parameters holds a crucial role in achieving high-accuracy results when employing support vector machines (SVM), random forests (RF), and k-nearest neighbors (KNN). Each classifier entails distinct tuning steps and parameters. A range of values was systematically tested for each classifier to determine the optimal parameters, and the parameters resulting in the highest overall classification accuracy were identified. In this study, the classified results obtained under the optimal parameters for each classifier were utilized to assess and compare the performance of the classifiers.

Support vector machine (SVM)

In land cover classification studies, as highlighted by Knorn et al. (2009) and Shi and Yang (2015), the radial basis function (RBF) kernel of the support vector machine (SVM) classifier is commonly employed due to its demonstrated good performance. Accordingly, we utilized the RBF kernel to implement the SVM algorithm. Two crucial parameters must be set when applying the SVM classifier with the RBF kernel: the optimal parameters of cost (C) and the kernel width parameter (γ) (Qian et al. 2015; Ballanti et al. 2016). The C parameter determines the permissible level of misclassification for non-separable training data, allowing for the adjustment of the rigidity of the training data (Li et al. 2014). On the other hand, the kernel width parameter (γ) influences the smoothness of the shape of the class-dividing hyperplane (Melgani and Bruzzone 2004). Larger values of C may lead to an overfitting model (Ghosh and Joshi 2014), while an increase in the γ value affects the shape of the class-dividing hyperplane, potentially impacting classification accuracy results (Huang et al. 2002). In line with the approach outlined by Li et al. (2014) and validated for our dataset through pretesting, this study explored three values of C (1, 5, 10) to identify the optimal parameters for the SVM classifier.

Random forest (RF)

To implement random forest (RF), it is necessary to configure two parameters: the number of trees (ntree) and the number of features considered in each split (mtry). Numerous

studies have indicated that satisfactory outcomes can be attained using default parameters (Duro et al. 2012). However, as highlighted by Liaw and Wiener (2002), a higher number of trees can yield a more stable result in variable importance. Additionally, Breiman (2001) mentioned that exceeding the required number of trees might be unnecessary, but it does not adversely affect the model. Moreover, Feng et al. (2015) suggested that RF could achieve accurate results with ntree = 200. Regarding the mtry parameter, many studies opt for the default value mtry = $\sqrt{p}$, where p is the number of predictor variables (Duro et al. 2012). However, in this study, to identify the optimal RF model for classification, a range of values for both parameters was systematically tested and evaluated: ntree = 100, 200, 500, and 1000; mtry = 1 to 10 with a step size of 1.

K-nearest neighbor (KNN)

The KNN approach is a nonparametric method that originated in the early 1970s for statistical applications (Franco Lopez et al. 2001). The fundamental principle behind KNN is locating a group of k samples in the calibration dataset closest to unknown samples, typically determined based on distance functions. From these k samples, the label (class) of unknown samples is determined by calculating the average of the response variables, representing the class attributes of the k-nearest neighbors (Akbulut et al. 2017; Wei et al. 2017). Consequently, the value of k plays a crucial role in the performance of KNN, serving as the key tuning parameter (Qian et al. 2015). The parameter k was determined through a bootstrap procedure. This study explored k values ranging from 1 to 20 to identify the optimal k value for all training sample sets.

Results

Comparative analysis of machine learning models

Various machine learning algorithm models are available when predicting geological formation tops or lithology in hydrocarbon reservoir exploration and production. The multilayer perceptron (MLP) proves beneficial in cases of high complexity or nonlinear relationships between input features and output targets. Random forest is well-suited for data with independent features or intricate interactions, offering class probability estimates and insights into significant features. K-nearest neighbor (KNN) presents an easily implementable and effective option for nonlinear scenarios, though it may be inefficient for large datasets or those with unbalanced class distributions. Support vector machine (SVM) is apt for high dimensionality and clear class separation datasets. It is important to note that no single algorithm model universally attains the highest or best accuracy in this context. Performance hinges on data characteristics, problem complexity, and appropriate parameter settings. Thus, it is advisable to undertake experiments and cross-validation to assess the relative performance of each model within the specific parameters of the given scenario.

The KNN classifier

In the KNN classifier, the algorithm classifies an object based on the class attributes of its k-nearest neighbors. Therefore, the k value is a crucial tuning parameter for the KNN algorithm. In this study, we conducted tests with k values (3–12) for nearest neighbors and explored two weight options (uniform and distance), as illustrated in Table 6. The optimal parameter selection for the KNN classifier involves using training datasets to assess the performance with different k values. Based on the conducted tuning experiments, the parameter configuration yielding the best results consists of setting the k to 3, with the weight parameter set to distance. Table 6 displays the outcomes of a hyperparameter tuning experiment using the k-nearest neighbors (KNN) algorithm through GridSearchCV. The investigation explored the impact of two key hyperparameters: the number of neighbors (param_n_neighbors) and the weight function (param_weights) employed in the KNN algorithm. Different configurations were tested in the "param_n_neighbors" column, representing the number of neighbors considered during predictions. The "param_weights" column indicates the weight function used in the KNN algorithm, with "uniform" suggesting an equal contribution from all neighbors and "distance" implying that closer neighbors have more influence on predictions.

The "mean_test_score" column represents the average score of the model on the test data for a given set of hyperparameters. Higher scores indicate superior model performance. The "std_test_score" column denotes the standard deviation of test scores across folds or splits,

Table 6 Optimal hyperparameter configurations for K-Nearest Neighbors classifier: a GridSearchCV analysis

param_n_neighbors	param_weights	mean_test_score	std_test_score
3	Distance	0.992986	0.001009
6	Distance	0.992027	0.000902
3	Uniform	0.991724	0.001116
9	Distance	0.991277	0.000729
12	Distance	0.990286	0.000728
6	Uniform	0.988337	0.000977
9	Uniform	0.986324	0.000991
12	Uniform	0.984247	0.001487

providing insight into the model's consistency. The findings reveal that the most successful configuration involves three neighbors with distance weighting, achieving a mean test score of approximately 99.30%. Arrangements with 6 and 9 neighbors closely follow, both using distance weighting. Notably, distance-weighted configurations consistently outperform those with uniform weighting. The results suggest that a smaller number of neighbors, particularly 3, coupled with distance weighting, leads to optimal performance for this KNN classifier and dataset. Standard deviations in the "std_test_score" column are relatively low, indicating consistent model performance across different folds. These insights are crucial for configuring KNN models in similar contexts, emphasizing the significance of tuning the number of neighbors and the weighting scheme for enhanced classification accuracy.

Figure 7 shows a receiver operating characteristic (ROC) curve for a multi-class classification problem. The ROC curve is a useful tool for evaluating the performance of a classifier, as it shows the trade-off between true positive rate (TPR) and false positive rate (FPR) at different classification thresholds. The ROC curve in the figure

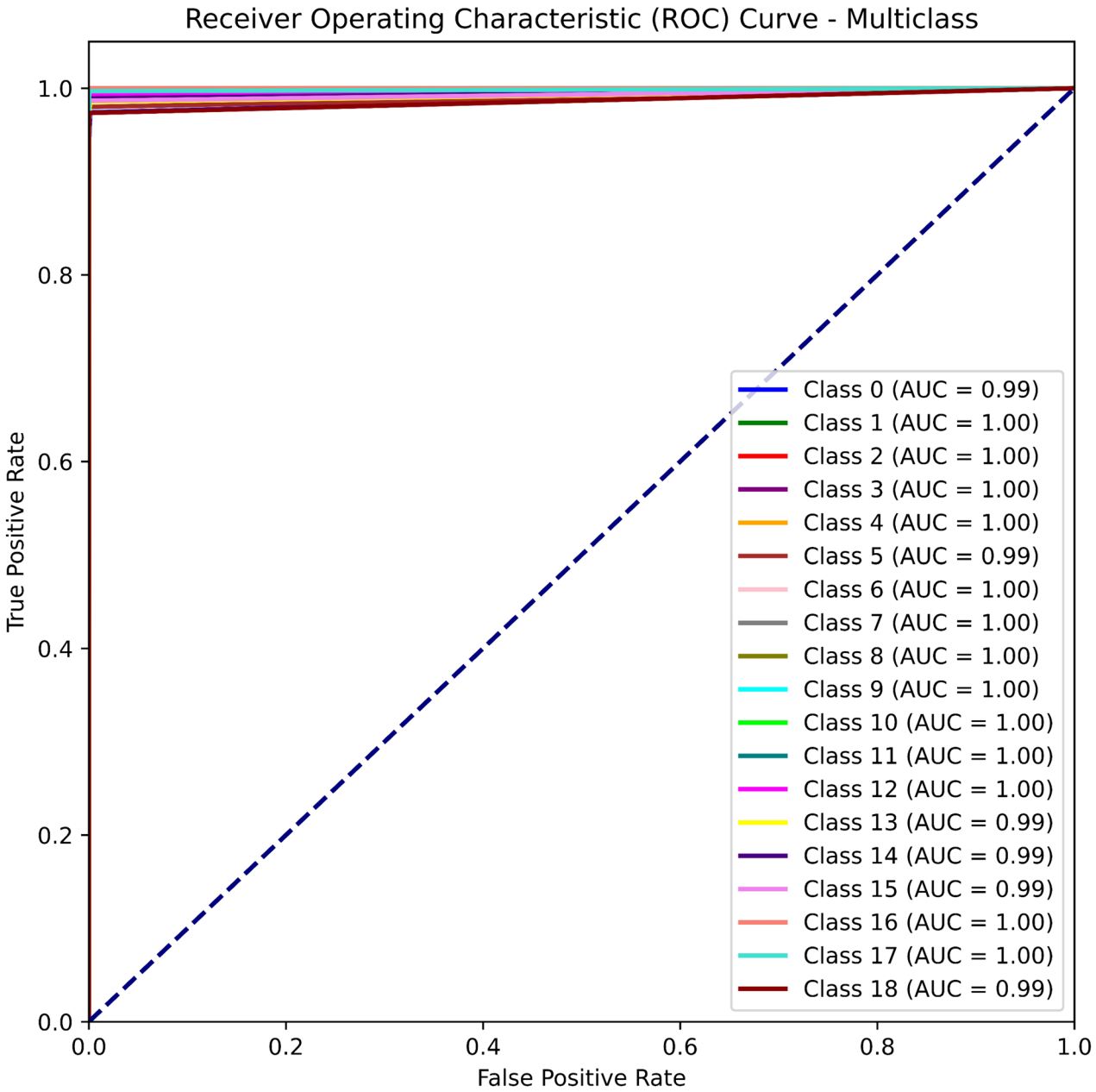


Fig. 7 Multiclass ROC curve for 19 classes with KNN model

shows that the classifier performs very well for all classes, with AUC values of 0.99 or higher for all classes except for class 13 and class 14, which have AUC values of 0.99. This means that the classifier can accurately identify positive examples (i.e., examples belonging to the target class) while minimizing the number of false positives (i.e., models incorrectly classified as belonging to the target class).

One way to interpret the ROC curve is to imagine that you have a classifier that is used to detect disease. The TPR is the proportion of diseased patients correctly identified by the classifier. At the same time, the FPR is the proportion of non-diseased patients incorrectly identified by the classifier as diseased. A high TPR means the classifier is good at identifying diseased patients, while a low FPR implies that the classifier avoids false positives. In Fig. 7, we can see that the ROC curve for each class is close to the top-left corner of the graph. This means the classifier can achieve a high TPR while maintaining a low FPR. For example, class 0 has an AUC of 0.99, which means that the classifier can correctly identify 99% of diseased patients while only misclassifying 1% of non-diseased patients as diseased. Overall, the ROC curve in the figure shows that the classifier is a very good performance. It can accurately identify positive examples while minimizing the number of false positives, which is an important goal for many classification tasks.

The diagonal elements of the confusion matrix represent the number of correct predictions for each class. The off-diagonal elements represent the number of incorrect predictions. For example, the entry in row 0, column 1, represents the number of examples incorrectly predicted as class 1, even though they belonged to class 0. The accuracy of the KNN model in the figure is 99.9%, which means that it correctly predicted the class of 99.9% of the examples in the test set. However, it is important to note that the accuracy score can be misleading for multi-class classification problems, especially when the classes are imbalanced. For example, if there are very few examples of class 10 in the test set, the KNN model can achieve a high accuracy score by simply predicting class 0 for all examples. A better way to evaluate the performance of the KNN model is to look at the precision, recall, and F1 score for each class. These metrics are more robust to class imbalance and provide a more complete figure of the model's performance (Figs. 8 and 9).

In summary, the presented model exhibits exceptional performance across all classes, demonstrating high precision, recall, and F1-Score. Notably, it distinguishes certain classes, such as 2, 11, 12, and 17, as evidenced by their

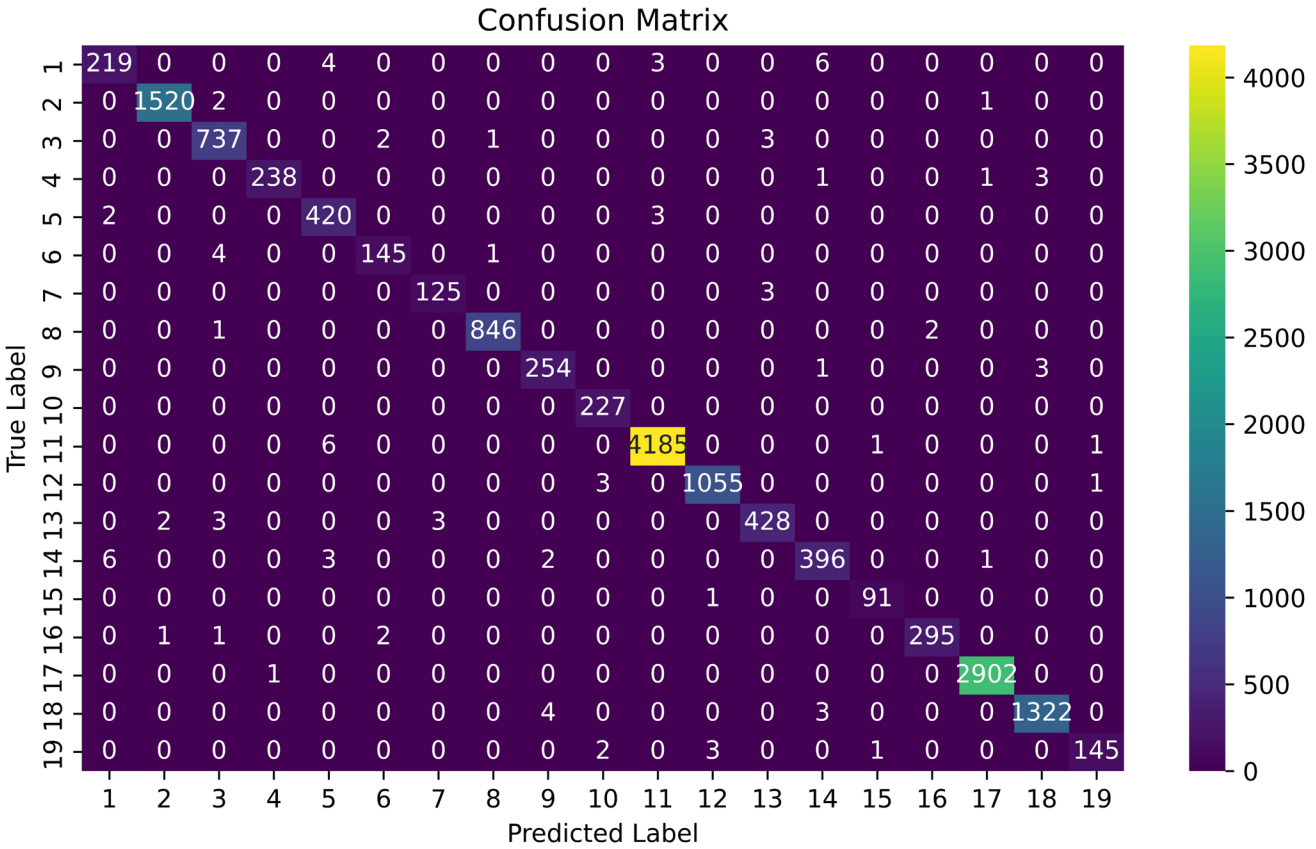


Fig. 8 Confusion matrix for K-Nearest Neighbors classification

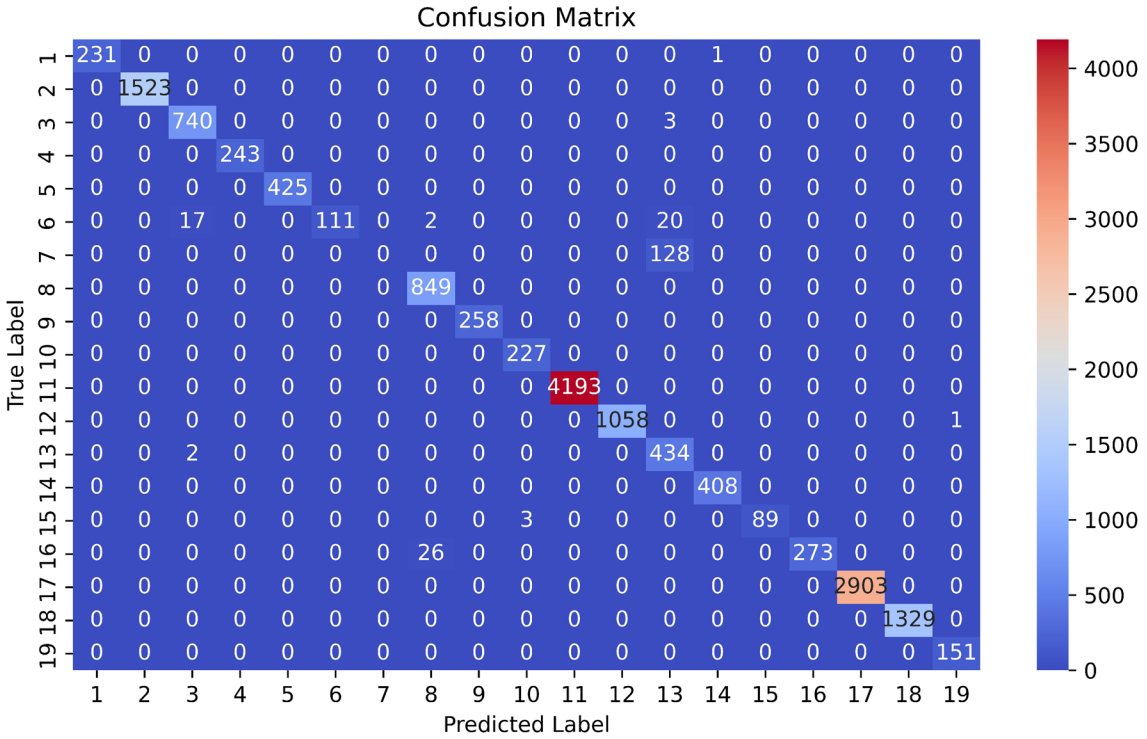


Fig. 9 Confusion matrix for random forest classifier

Table 7 KNN classification report: high-performance metrics across diverse classes

	Precision	Recall	f1-score	Support
1	0.96	0.94	0.95	232
2	1.00	1.00	1.00	1523
3	0.99	0.99	0.99	743
4	1.00	0.98	0.99	243
5	0.97	0.99	0.98	425
6	0.97	0.97	0.97	150
7	0.98	0.98	0.98	128
8	1.00	1.00	1.00	849
9	0.98	0.98	0.98	258
10	0.98	1.00	0.99	227
11	1.00	1.00	1.00	4193
12	1.00	1.00	1.00	1059
13	0.99	0.98	0.98	436
14	0.97	0.97	0.97	408
15	0.98	0.99	0.98	92
16	0.99	0.99	0.99	299
17	1.00	1.00	1.00	2903
18	1.00	0.99	1.00	1329
19	0.99	0.96	0.97	151
Accuracy			0.99	15,648
Macro avg	0.99	0.98	0.98	15,648
Weighted avg	0.99	0.99	0.99	15,648

perfect precision, recall, and F1-Score (Table 7). It is important to consider the specific requirements and characteristics of the classification problem when interpreting these performance metrics.

The random forest classifier

As highlighted in "Random forest (RF)" section, two key parameters, namely tree and mtry, significantly impact the performance of Random Forest (RF) classifiers. In this study, we engaged in hyperparameter tuning, exploring various parameters to achieve optimal accuracy. The hyperparameter tuning involved adjusting (n_estimator, max_depth, and max_feature), each with a specified range, as depicted in Table 8. Following the tuning process, the most effective parameters were identified as {'max_depth': 6, 'max_features': 6, 'n_estimators': 100}.

The configuration {'max_depth': 6, 'max_features': 6, 'n_estimators': 100} stands out as the best-performing, achieving a mean test score of 0.985 and ranking first. This suggests that a deeper tree ('max_depth': 6), using more features ('max_features': 6), and a moderate number of trees ('n_estimators': 100) contribute to higher accuracy. Higher standard deviations in some configurations, such as {'max_depth': 3, 'max_features': 1, 'n_estimators': 124}, indicate more variability in performance. The 'n_estimators' parameter varies between 100 and 145, with no clear

Table 8 Random forest hyperparameter tuning results

Params	mean_test_score	std_test_score	rank_test_score
{'max_depth': 6, 'max_features': 6, 'n_estimators': 100}	0.984741972	0.001879929	1
{'max_depth': 3, 'max_features': 1, 'n_estimators': 124}	0.768892794	0.011657568	9
{'max_depth': 6, 'max_features': 3, 'n_estimators': 104}	0.974820259	0.00153778	3
{'max_depth': 5, 'max_features': 2, 'n_estimators': 107}	0.915258028	0.005175264	6
{'max_depth': 5, 'max_features': 3, 'n_estimators': 112}	0.928726634	0.006858747	5
{'max_depth': 4, 'max_features': 5, 'n_estimators': 105}	0.882664962	0.001043287	7
{'max_depth': 6, 'max_features': 3, 'n_estimators': 145}	0.975139799	0.001706175	2
{'max_depth': 5, 'max_features': 4, 'n_estimators': 141}	0.944032593	0.006297064	4
{'max_depth': 3, 'max_features': 1, 'n_estimators': 104}	0.760760505	0.014307074	10
{'max_depth': 3, 'max_features': 6, 'n_estimators': 115}	0.813612398	0.011918066	8

Table 9 Classification report for random forest classifier

	Precision	Recall	f1-score	Support
1	1.00	1.00	1.00	232
2	1.00	1.00	1.00	1523
3	0.97	1.00	0.99	743
4	1.00	1.00	1.00	243
5	1.00	1.00	1.00	425
6	1.00	0.74	0.85	150
7	0.00	0.00	0.00	128
8	0.97	1.00	0.98	849
9	1.00	1.00	1.00	258
10	0.99	1.00	0.99	227
11	1.00	1.00	1.00	4193
12	1.00	1.00	1.00	1059
13	0.74	1.00	0.85	436
14	1.00	1.00	1.00	408
15	1.00	0.97	0.98	92
16	1.00	0.91	0.95	299
17	1.00	1.00	1.00	2903
18	1.00	1.00	1.00	1329
19	0.99	1.00	1.00	151
Accuracy			0.99	15,648
Macro avg	0.93	0.93	0.93	15,648
Weighted avg	0.98	0.99	0.98	15,648

pattern indicating an optimal number of trees. The hyperparameter tuning process identified a configuration that maximizes mean test scores, shedding light on effective hyperparameter values for this Random Forest model. Further validation on a separate test set is recommended for robust performance assessment (Table 9).

Based on the classification report, the random forest classifier performs well overall, with a weighted average F1 score of 0.98 and an accuracy of 0.99.

- Precision and recall scores are strong (mostly at or very close to 1.00) for most classes, indicating the model reliably predicts those classes and does not make many errors.
- Classes 7 and 13 have lower precision, meaning the model makes incorrect predictions on those classes when it predicts them. But recall is still high, meaning it correctly finds most examples of those classes.
- Class 6 and 15 have lower recall, so the model struggles to identify some examples of those classes (about 26% of class 6 missed, 3% of class 15). Precision is still high when it does predict those classes.
- The model seems robust to a class imbalance with both macro-average and weighted-average F1 scores at 0.93 and 0.98. Performance across minority classes does not drop off.

In summary, I generally performed extremely strongly, with just a few classes representing opportunities for improvement. The focus could be distinguishing classes 6, 7, 13, and 15. But generally an excellent, well-balanced classifier.

Figure 10 shows a decision tree visualization from a random forest classifier. The tree has a maximum depth of 2, meaning no node in the tree is more than two levels deep. The tree leaves are labeled with the class to which most of the training examples that reach that leaf belong. The tree is used to classify a set of models by starting at the root node and asking a question about one of the features. The answer to the question determines which child node of the root node to go to. This process is repeated at each child node until a leaf node is reached. The class label of the leaf node is then used to classify the example. The "TRPM RT 0.516" at the top of the tree is the value of the "TRPM_RT" feature used to split the data at the root node. The "gini = 0.863" is the Gini impurity of the root node. The Gini impurity measures how well the data is split at a

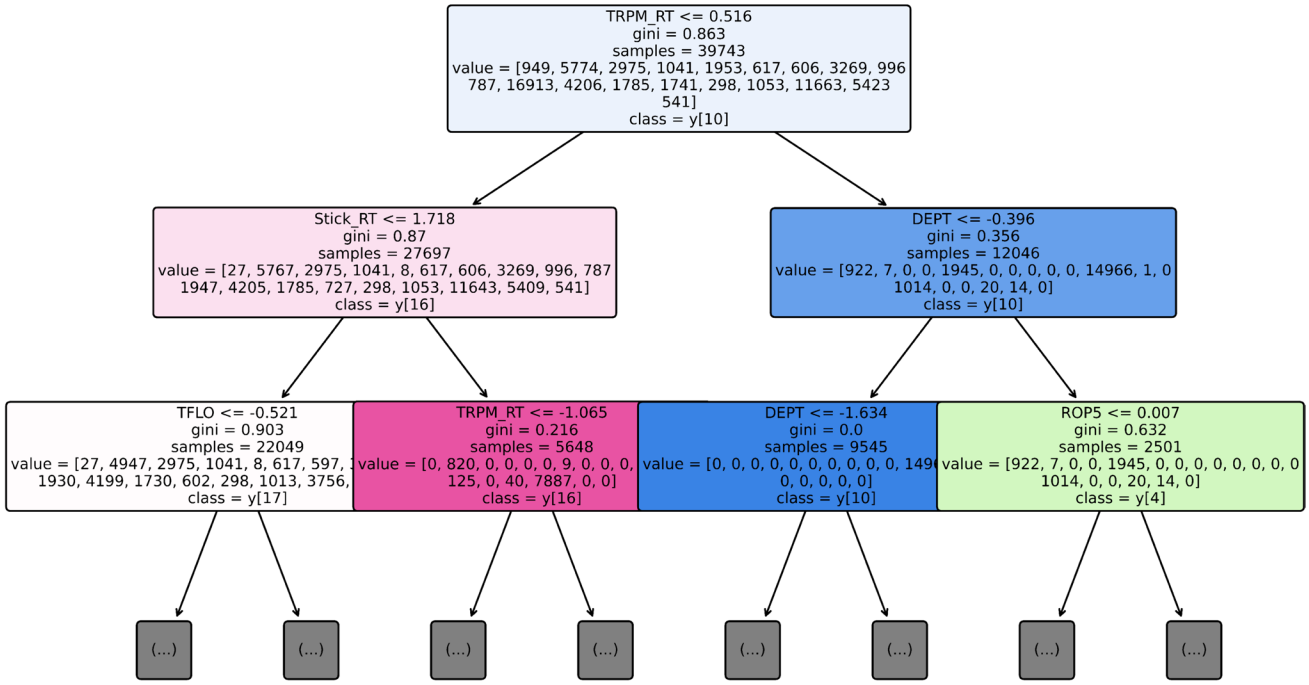


Fig. 10 Random forest classifier—decision tree visualization (max depth = 2)

node. A lower Gini impurity means that the data is purer at that node and that it is easier to classify the examples that reach it. The "samples 39,743" value at the root node is the number of training examples that get that node. The "value [949, 5774, 2975, 1041, 1953, 617, 606, 3269, 996, 787, 16,913, 4206, 1785, 1741, 298, 1053, 11,663, 5423, 5411]" is the number of examples in each class that reach the root node. The "class = y[10]" value at the root node is the majority class of the examples that match the root node. This Figure is a good example of how decision trees can be used to classify data. The tree is relatively simple but can still achieve accuracy on many tasks.

The support vector machine classifier

Similarly, in the case of SVM, our research involves hyperparameter tuning with several key parameters. The parameters subject to tuning include (C, kernel, and gamma). For the C parameter, we explore a range of values (1, 5, 10), while the kernel parameter is adjusted between Linear and RBF. Lastly, the gamma parameter is varied with options of 'scale' and 'auto,' with further details in Table 10. Following the tuning process, the optimal parameter configuration identified is {'C': 10, 'gamma': 'scale,' 'kernel': 'rbf'}. Table 10 summarizes the results of hyperparameter tuning for an SVM (Support Vector Machine) algorithm using a

Table 10 **SVM** hyperparameter tuning results

param_C	param_kernel	param_gamma	mean_test_score	std_test_score
10	rbf	Scale	0.992603	0.000969
10	rbf	Auto	0.992603	0.000990
5	rbf	Scale	0.988784	0.001492
5	rbf	Auto	0.988784	0.001492
10	Linear	Scale	0.987059	0.001316
10	Linear	Auto	0.987059	0.001316
5	Linear	Scale	0.984359	0.001679
5	Linear	Auto	0.984359	0.001679
1	Linear	Scale	0.975555	0.001901
1	Linear	Auto	0.975555	0.001901
1	rbf	Scale	0.973798	0.001572
1	rbf	Auto	0.973798	0.001595

Table 11 SVM classification report (best estimator)

	Precision	Recall	f1-score	Support
1	0.97	0.99	0.98	232
2	0.99	1.00	1.00	1523
3	1.00	0.99	0.99	743
4	0.92	0.99	0.95	243
5	1.00	1.00	1.00	425
6	0.90	0.99	0.95	150
7	0.94	0.99	0.97	128
8	0.98	0.98	0.98	849
9	0.98	1.00	0.99	258
10	1.00	1.00	1.00	227
11	1.0	1.00	1.00	4193
12	1.00	1.00	1.00	1059
13	0.99	0.96	0.97	436
14	0.99	0.98	0.98	408
15	0.98	0.99	0.98	92
16	1.00	0.95	0.97	299
17	1.00	1.00	1.00	2903
18	1.00	0.98	0.99	1329
19	0.99	0.99	0.99	151
Accuracy			0.99	15,648
Macro avg	0.98	0.99	0.98	15,648
Weighted avg	0.99	0.99	0.99	15,648

Grid Search approach. This table provides information about different combinations of hyperparameters and their corresponding mean test scores and standard deviations. The best-performing model has an 'rbf' kernel, 'scale' gamma, and a C value of 10, achieving a mean test score of approximately 99.26% with a standard deviation of around 0.00097. The table also provides insights into the performance of different combinations of hyperparameters, allowing you to compare linear and radial basis function kernels, different gamma values, and various levels of regularization (C values). This table can guide you in selecting the optimal hyperparameters for your SVM model. You may choose the hyperparameter combination that maximizes the mean test score while considering the standard deviation to ensure stability across different cross-validation folds (Table 11).

Figure 11 shows the confusion matrix for the SVM model's predictions on the test set. The confusion matrix visualization shows the model is very accurate, with most predictions along the diagonal indicating correct classification. Most predictions fall on the diagonal, indicating precise type. The classification report further backs this up. With ~ 15 k test samples, the SVM achieves 99% accuracy. Precision, recall, and F1-score for each class are also very high-most are above 95%, and many are at 100%. This indicates the model is very good at correctly predicting each class. The strong diagonal confusion matrix shows the model skillfully discriminates between the different classes. The

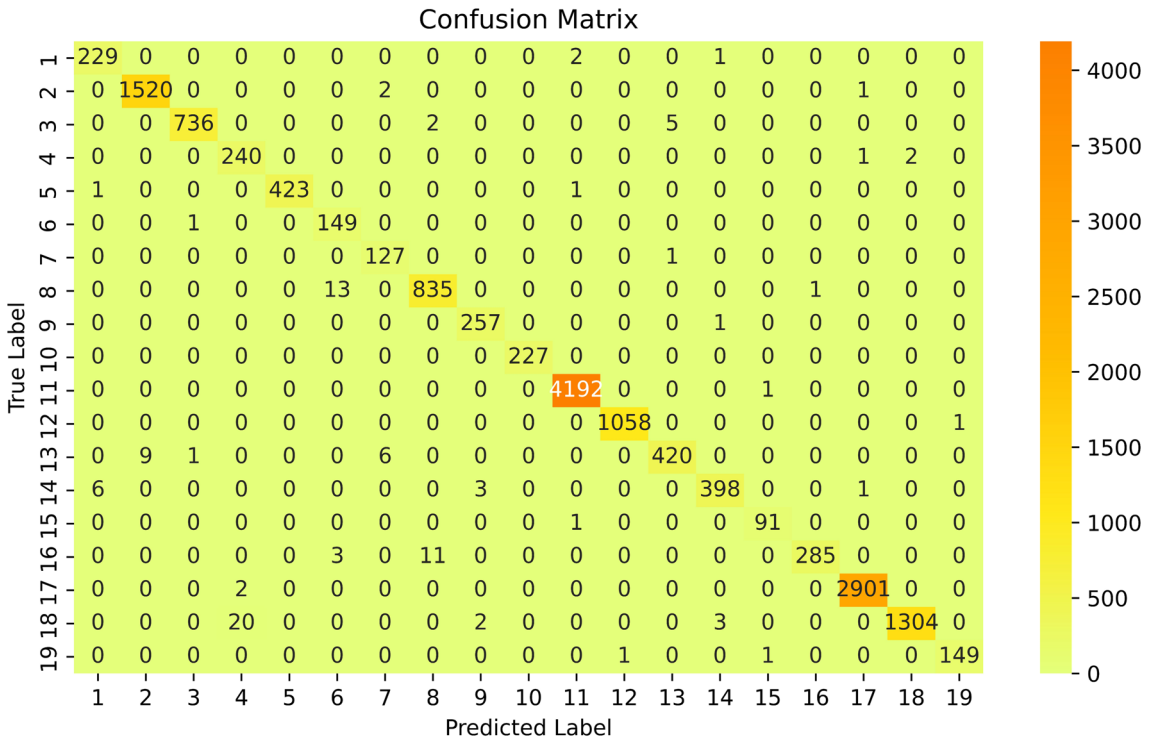


Fig. 11 SVM confusion matrix

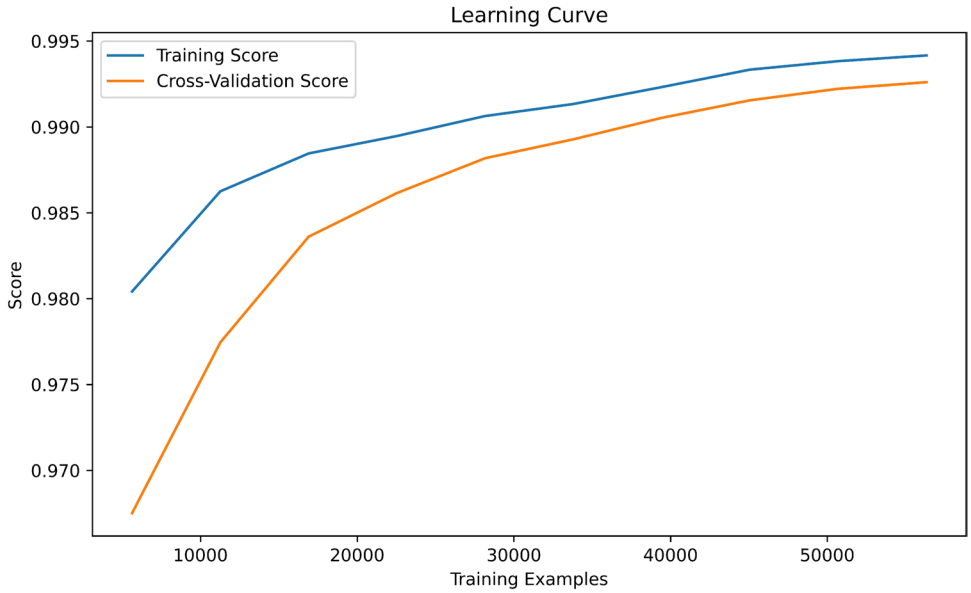


Fig. 12 Support vector machine learning curve

class-wise prediction reliability offered in the classification report is also very high. This analysis validates the SVM's exemplary performance on this dataset.

Figure 12 appears to show the learning curve of a machine learning model. The training score measures how well the model performs on the trained data. The cross-validation score measures how well the model acts on data it was not trained on. This is an important metric because it helps ensure that the model is not simply overfitting the training data. The graph shows that the training score increases as the number of training examples increases. This is expected, as the model is learning from the data. The cross-validation score is also growing but at a slower rate. This suggests that the model is starting to overfit the training data. The ideal situation is for the training and cross-validation scores to be as close together as possible. This would indicate that the model is learning from the data without overfitting. The gap between the training and cross-validation scores is relatively small. This suggests that the model is doing a good job of learning from the data without overfitting. The graph shows that the model is learning from the data and not overfitting the training data. However, there is still room for improvement.

Multilayer perceptron (MLP) classifier

The multilayer perceptron (MLP) is a class of artificial neural networks that utilizes backpropagation for training. MLPs contain multiple layers of nodes, including an input layer, one or more hidden layers, and an output layer. In this study, the MLP classifier was implemented using the Keras deep learning library in Python. For the MLP model, a sequential model was defined with an input layer size equal to the number of features, two hidden layers with 32 and 16 nodes, respectively, and an output layer with a size equal to the number of target classes (19 class types). The hidden layers used the Rectified Linear Unit (ReLU) activation function, and the output layer used softmax activation to generate probabilistic predictions. The MLP model was trained for 50 epochs using the Adam optimizer and categorical cross-entropy loss function. To prevent overfitting, L2 regularization was applied to the weights. The model achieved a training accuracy 1 and was evaluated on the test set data. Key strengths of MLPs include the ability to model complex nonlinear relationships, adaptability to various problems, and use of backpropagation to learn deep feature representations. However, challenges include longer training times, the need for parameter tuning, and susceptibility to overfitting. Overall, the MLP model performed comparable to the other classifiers on this dataset for predicting geological formation tops. Table 12 shows different MLP architectures experimented with by varying the number of layers, nodes per layer, activation functions, and regularization techniques. A 3-layer network with 64 and 32 nodes and ReLU or Softmax activation produced the best results, with Adam optimizer and Categorical Cross entropy loss function helping prevent overfitting. The tuning process helped determine the optimal structure and hyperparameters for the MLP model on this dataset.

In Table 12, three experiments with varying MLP architectures were tested, each characterized by specific hyperparameter settings. Experiment 1 reveals a moderate level of accuracy at 0.95. Given the chosen architecture and hyperparameters, the model performs reasonably well. It serves as a baseline for comparison with other experiments. Experiment 2 exhibits a significant improvement in accuracy, reaching

Table 12 MLP architectures tested

Experiment	1	2	3
Learning rate	0.001	0.01	0.0001
Epochs	10	15	8
Batch size	32	64	16
Hidden layers	1	2	3
Neurons per layer	64	128,64	64, 32
Activation function	ReLU, Softmax	ReLU, Softmax	ReLU, Softmax
Optimizer	Adam	Adam	Adam
Loss function	Categorical Crossentropy	Categorical Crossentropy	Categorical Crossentropy
Accuracy	0.95	0.98	1

0.98. The increased number of epochs, hidden layers, and neurons per layer likely contributed to enhanced model performance. This suggests that a more complex architecture can capture intricate patterns in the data, leading to improved accuracy. Experiment 3 demonstrates a perfect accuracy of 1. While achieving 100% accuracy may suggest overfitting to the training data, using a lower learning rate, fewer epochs, and a more intricate architecture might have contributed to this result. It is essential to assess the model's generalization performance on unseen data to ensure its effectiveness in real-world scenarios. In summary, these experiments showcase the impact of hyperparameter tuning on MLP model performance. Experiment 2, with a learning rate of 0.01 and a more complex architecture, stands out as the top performer, achieving the highest accuracy of 0.98. The choice of hyperparameters significantly influences the model's ability to learn and generalize from the training data (Fig. 13).

The multilayer perceptron (MLP) model demonstrates very strong performance in multi-class classification of handwritten digits, as evidenced by the confusion matrix and classification report. The model achieves an overall

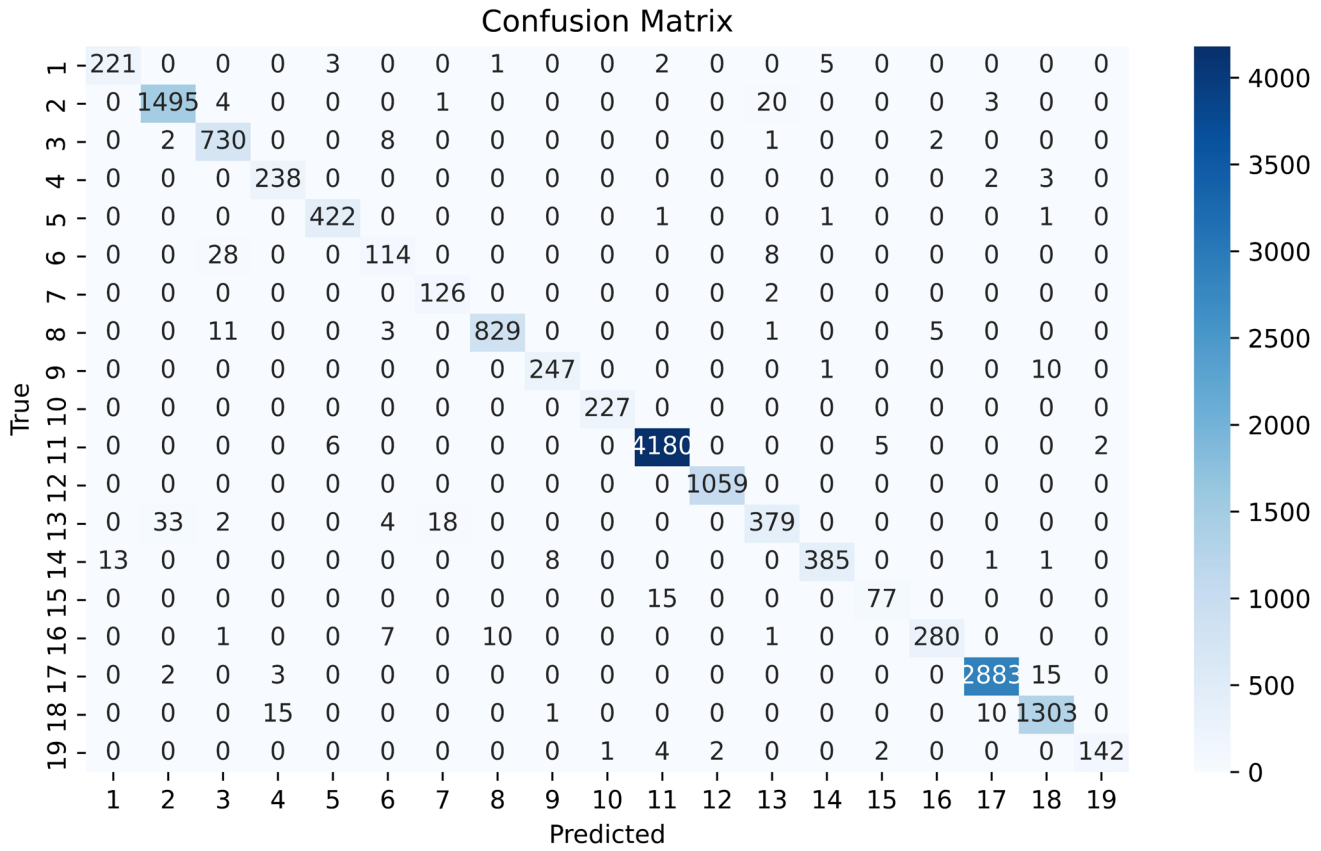


Fig. 13 Predictive accuracy—MLP confusion matrix

accuracy of 0.98 on the test set, correctly classifying 98% of the 15,648 test samples. This high accuracy indicates reliable predictive capabilities across the 20 different digit classes. Precision, recall, and F1 scores are mostly in the excellent 0.94–0.99 range across categories, signaling that the model predicts each true type with high precision and retrieves most samples within each category (high recall), leading to robust F1 balances. The macro-averaged precision, recall, and F1 scores are 0.96, 0.95, and 0.95, respectively, confirming reliable differentiation, particularly for low-volume classes in the test dataset. Weighted averages approaching the accuracy score further establish that most errors derive from smaller classes. While classes 6 and 15 exhibit comparatively lower scores, suggesting some difficulty differentiating these specific digits, this does not substantially impact overall performance given the much greater distribution of other classes in the dataset. In summary, precise class-wise metrics nearing or exceeding 0.95 and 98% test accuracy underscore MLP's proficiency in distinguishing handwritten digits based on the provided test set evaluation data and labels. The model achieves excellent inter-class differentiation to classify 98% of all samples successfully.

Figure 14 shows the side-by-side visualization of weight heatmaps for each layer in the neural network model. Each subfigure illustrates the spatial distribution of weights in the corresponding layer, providing insights into the learned features and relationships. The color intensity represents the magnitude of the importance, with warmer colors indicating higher values. These weight heatmaps offer a glimpse into the intricate patterns and structures captured by the neural network during the training process.

Figure 15 shows a model's performance over time, with an accuracy plot on top and a loss plot on the bottom.

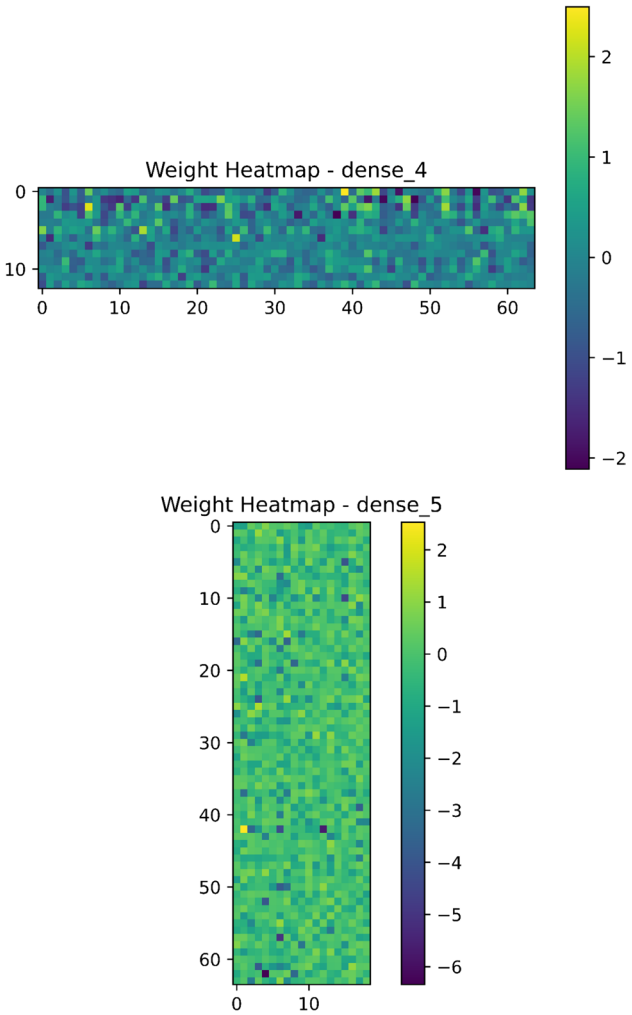


Fig. 14 Weight heatmaps of neural network layers: visual representation of the learned weights in each layer of the model

- Accuracy plot analysis:
  - The training accuracy starts around 0.6 and reaches near 1.0 after around 15 epochs. This indicates the model is learning from the training data.
  - The validation accuracy starts near 0.6 as well. It reaches a peak of about 0.8 by around eight epochs, then levels off and decreases slightly after that.
  - The gap between training and validation accuracy grows over time. This suggests overfitting is occurring, where the model performs better on training data than new validation data.
- Loss plot analysis:
  - The training loss decreases rapidly in the first five epochs, indicating the model is optimizing and learning quickly.
  - After five epochs, training loss continues decreasing but at a slower pace. This signals the model is still improving but at a slower rate.
  - Validation loss mirrors training loss early on, decreasing rapidly rather than more slowly. But after 5–8 epochs, validation loss levels off and stops improving much.

This model exhibits signs of overfitting, performing better on training data than validation data over time. The best weights are likely at 5–8 epochs when validation accuracy peaks and validation loss still decreases. To improve the model, regularization techniques could be added to reduce overfitting. But after ~ 8 epochs, performance on new data does not seem to improve much or begins degrading, suggesting training should likely be stopped by then.

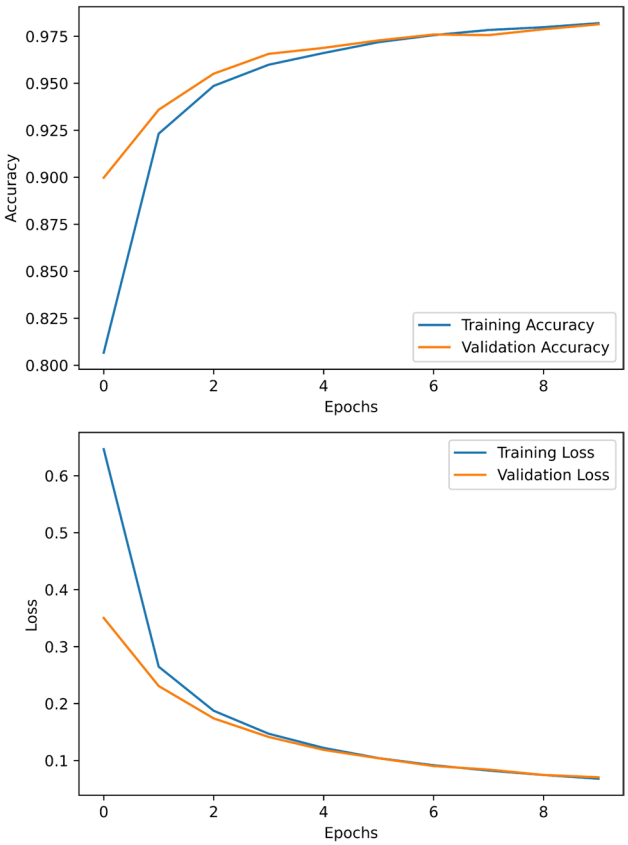


Fig. 15 Model performance over time. the top subplot is the accuracy plot showing training and validation performance, and the Bottom is the loss plot showing training and validation performance

High Accuracy in Machine Learning Models

Throughout the modeling process employing various machine learning algorithms, including Support Vector Machines (SVM), Random Forest, K-Nearest Neighbors (KNN), and Multilayer Perceptron (MLP), notable differences in accuracy scores were observed. Specifically, when applied to the training dataset, the Multilayer Perceptron (MLP) algorithm demonstrated a commendable accuracy score of 1. Also, SVM, Random Forest, and K-Nearest Neighbors (KNN) algorithms exhibited slightly lower accuracy score 0.99. The modeling procedures utilized the training dataset detailed in section "For this investigation, we gathered the", Approach and Procedures. These findings suggest that, in predicting geological formation tops in hydrocarbon reservoir exploration and production, the MLP algorithm excelled in accuracy compared to other algorithms. However, selecting the most suitable algorithm involves considering additional factors such as computational efficiency, interpretability of results, and application-specific requirements. Further assessment and testing are imperative to validate the reliability and applicability of the chosen model. There is a considerable disparity between these training dataset results and the accuracy scores obtained when predicting the blind dataset, falling within the range of 0.91–0.99. Several factors may contribute to this decrease in model performance on the blind dataset. Overfitting is one potential factor where the model becomes excessively tailored to the training data and struggles to generalize to unseen data. Overfitting may arise from a model's complexity or insufficient training data relative to the problem's intricacy. Furthermore, an imbalance in sample numbers or class distribution in the blind dataset might lead the model to favor predictions for the majority class. Other influences, such as inadequate parameter tuning, suboptimal data quality, or other sources of error, can also impact the accuracy of the blind dataset. Therefore, a comprehensive analysis is essential to pinpoint the reasons behind the performance decline on the blind dataset, and strategies like improved parameter tuning, regularization techniques, appropriate data preprocessing methods, or additional data collection may be applied to enhance the model's ability to accurately predicting formation tops or rock types in real-world scenarios. Refer to Table 13 for a detailed comparison of accuracy scores across machine learning models.

Table 13 Comparative accuracy scores of machine learning models

Chosen model	Evaluation of models on test sets	Evaluation of models on blind sets
K-Nearest Neighbors (KNN)	0.99	0.93
Random forest (RF)	0.99	0.91
Support vector machines (SVM)	0.99	0.95
Multilayer perceptron (MLP)	1	0.99

Table 13 presents predictions generated by a model trained on the training datasets, as outlined in the evaluation on test sets section. In contrast, the blind sets represent outcomes obtained when making predictions using data that has not been previously used or encountered. Figure 16 compares the predicted formation tops from four models (KNN, SVM, MLP, and RF) to the actual formation tops. Figure 16 shows that all four models could accurately predict the formation tops. The MLP model had the smallest average error, followed by the SVM, KNN, and RF models.

Feature importance analysis

To gain a deeper understanding of which parameters are most influential in driving the predictions of our formation tops models, we employed two model-agnostic feature importance analysis techniques—Permutation Feature Importance (PFI) and Shapley Additive exPlanations (SHAP). Permutation Feature Importance evaluates

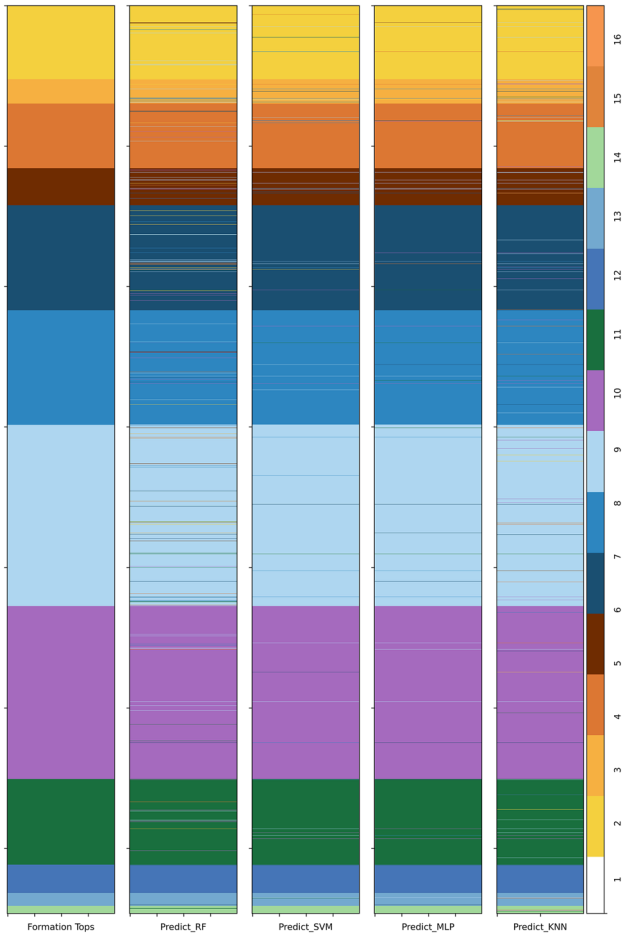


Fig. 16 Predicted vs actual geological formation tops for all models

the predictive power of each feature by calculating the decrease in the model's evaluation metric (such as accuracy) when the feature's values are randomly shuffled. This breaks the original association between the feature and the target variable. A large decrease in the evaluation metric after shuffling indicates higher feature importance. On the other hand, SHAP explains the model's predictions by computing Shapley values for each feature. These Shapley values represent the feature's contribution in "pushing" the forecast away from the expected value. Features with large absolute Shapley values are considered highly impactful on the prediction. We supplemented the analysis with visualizations of the importance of each model's SHAP and PFI features, as shown in Figs. 17, 18, 19, 20, 21, 22, 23, 24, 25, 26 and 27 below. By leveraging both PFI and SHAP for our KNN, Random Forest, SVM, and MLP models, we obtained the following insights:

o **KNN**

o According to PFI, HKLD, SWOB, SPPA, DEPT, and RPM, they have emerged as the most important features. This implies that these two features strongly influence determining the nearest neighbors for a data point, which drives the KNN model's formation tops prediction.

p The SHAP analysis also identified DEPT as a top contributor. Additionally, RPM, ECD_MWD, SPPA, and HKLD greatly impacted predictions. The agreement between PFI and SHAP provides confidence about the significant role of DEPT, RPM, SPPA, and HKLD in the KNN model.

The resulting SHAP summary plot represents the combined SHAP values across all classes. Each feature's importance and impact on the model's output are visualized. The *y*-axis of the plot typically represents the features, and the *x*-axis represents the average magnitude of the SHAP values. Each dot in the plot corresponds to a specific instance in your dataset.

The SHAP plot above reveals that DEPT had the highest SHAP values, indicating their strong influence on predictions in the KNN model. RPM, ECD_MWD, SPPA, and HKLD also had a substantial impact. This aligns with the PFI analysis. Conversely, features like BVEL, Stick_RT, SWOB, TFLO, BPOS, and TQA have lower SHAP values, indicating their influence is less pronounced. The plot also shows that the SHAP values are generally positive for higher feature values, meaning that increasing the importance of these features tends to increase the model's output. However, a few exceptions exist, such as BVEL, where expanding the value decreases the model's production.

The PFI bar chart shows the significant decrease in KNN model accuracy when ROP5 and TFLO were used, demonstrating their low permutation importance. This validates the findings from the SHAP analysis.

Here is a summary of the key points of the KNN model analysis:

- According to both PFI and SHAP analyses, DEPT, RPM, HKLD, SPPA, and SWOB were the most important features for predicting geological formation tops. These features had the strongest influence on determining nearest neighbors and model predictions.
- SHAP analysis also identified ECD_MWD as having a high impact. There was an agreement between PFI and SHAP on the significant roles of DEPT, RPM, SPPA, and HKLD.
- The SHAP summary plot visualized each feature's impact. DEPT had the highest SHAP values, followed by RPM, ECD_MWD, SPPA, and HKLD. Features like BVEL, Stick_RT, SWOB, and TFLO had less impact.

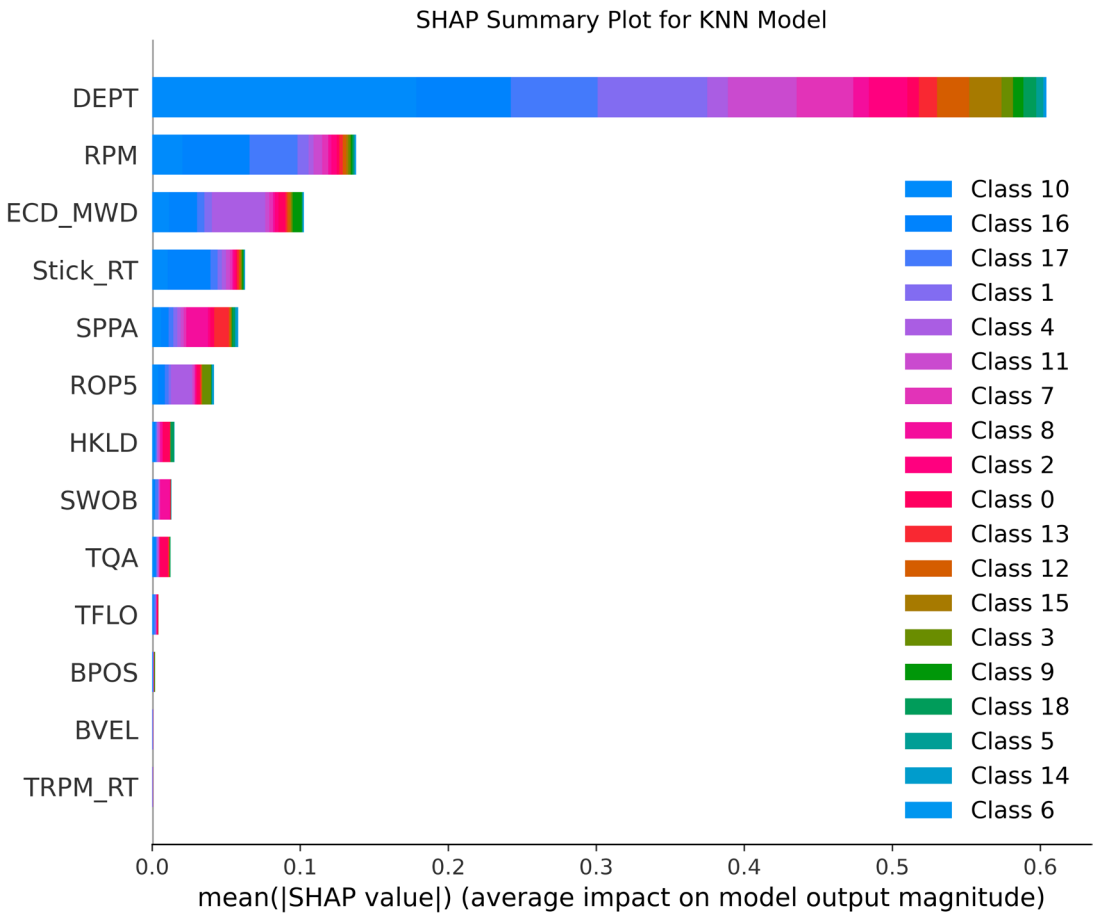


Fig. 17 SHAP feature importance plot for the KNN model

- Higher feature values tended to increase model output, with some exceptions like BVEL.
- The PFI analysis validated the SHAP findings—ROP5 and TFLO produced significant decreases in model accuracy when permuted, demonstrating their low importance.

In summary, DEPT, RPM, HKLD, SPPA, and ECD_MWD emerged as the most influential features in the KNN model for predicting geological formation tops. The PFI and SHAP analysis techniques aligned on the key variables driving the model.

- Random Forest

  – DEPT was flagged as a highly impactful feature by PFI. This can be attributed to its likely involvement in key decision tree splits within the Random Forest ensemble.

  – SHAP analysis concurred on the importance of DEPT. It also highlighted RPM as an additional key driver of predictions, aligning with the PFI results.

The SHAP plot highlights DEPT and RPM as highly influential features in the Random Forest model based on their SHAP values. TFLO also had a noticeable impact.

The PFI chart reveals the substantial Random Forest model accuracy dropped when ROP5, TQA, and BVEL were used, confirming their low permutation importance. BVEL also showed a significant accuracy decrease, agreeing with the SHAP results.

Figure 22 shows a graph showing the effect of various parameters on model output for prediction for Sleipner formation top. The chart has two axes: the x-axis shows the feature value, and the y-axis shows the SHAP value (impact on model output). The SHAP value measures how much each feature contributes to the model's production. The graph shows that the parameters with the highest impact on the model output are DEPT, RPM, ECD_MWD, and TFLO. These parameters have SHAP

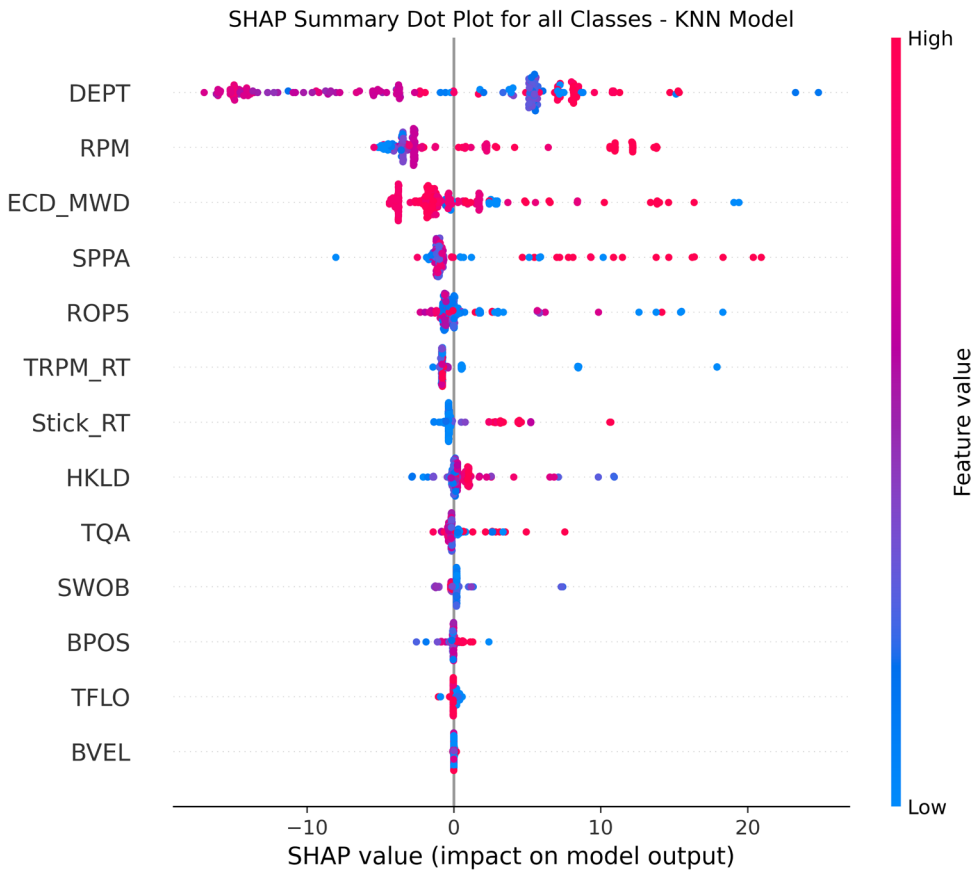


Fig. 18 SHAP summary dot plot for all classes—visualizing the impact of features on predictions for geological formation tops using SHAP values

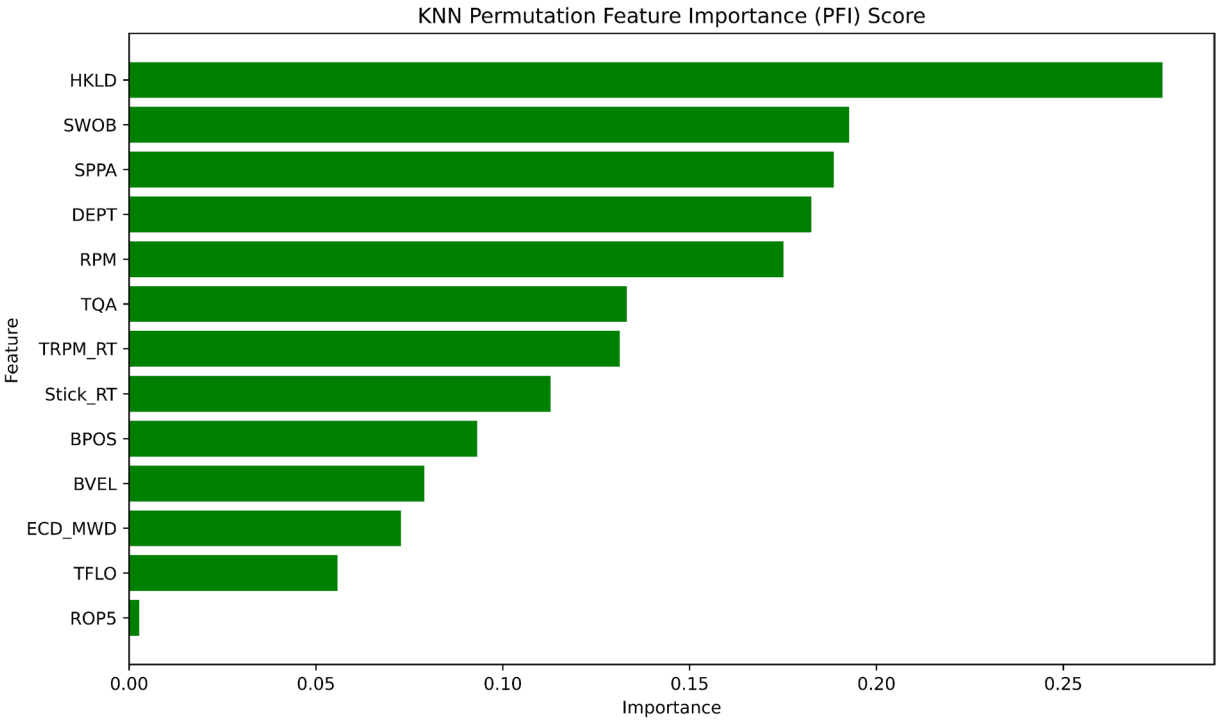
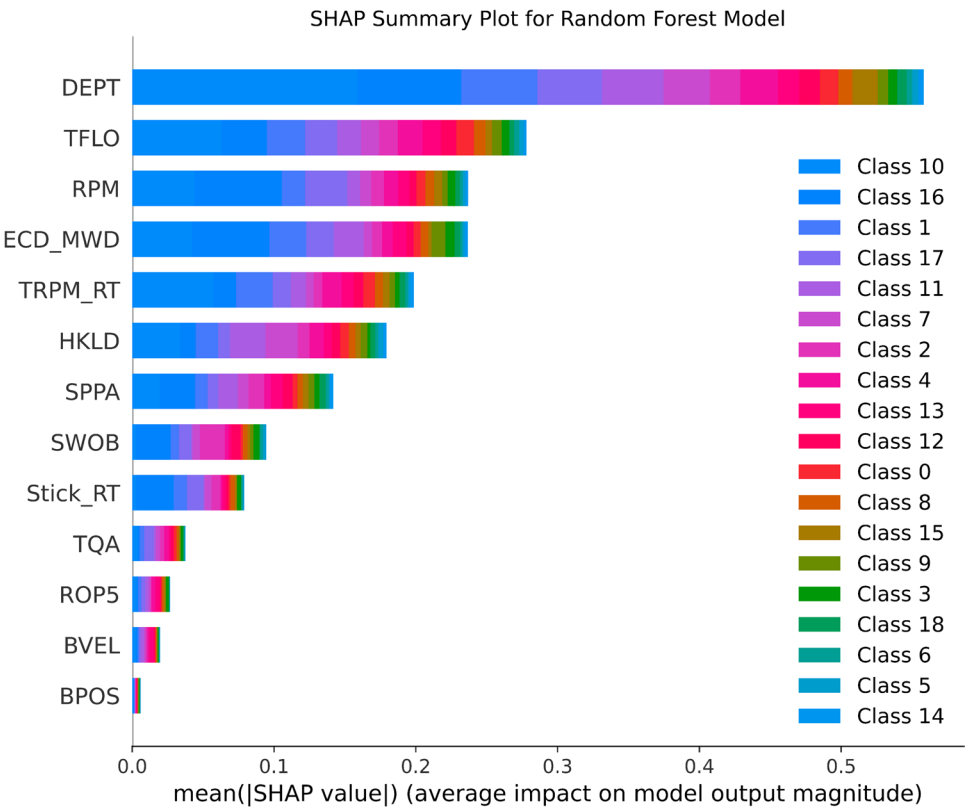


Fig. 19 PFI (permutation feature importance) bar chart for KNN



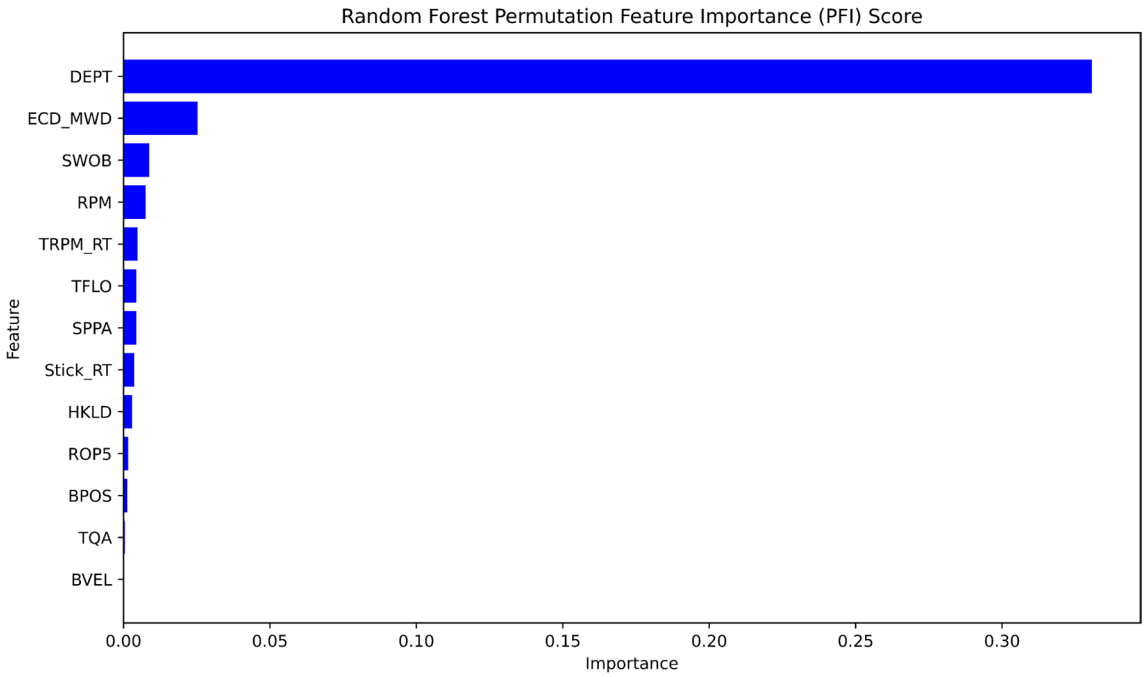


Fig. 20 SHAP feature importance plot for Random Forest

Fig. 21 PFI (permutation feature importance) bar chart for Random Forest

values that are greater than 0.2. The parameters with the lowest impact on the model output are Stick_RT, TRPM_RT, and BVEL. These parameters have SHAP values that are less than −0.2. The graph also shows that the relationship between the feature and SHAP values is not always linear. For example, the SHAP value

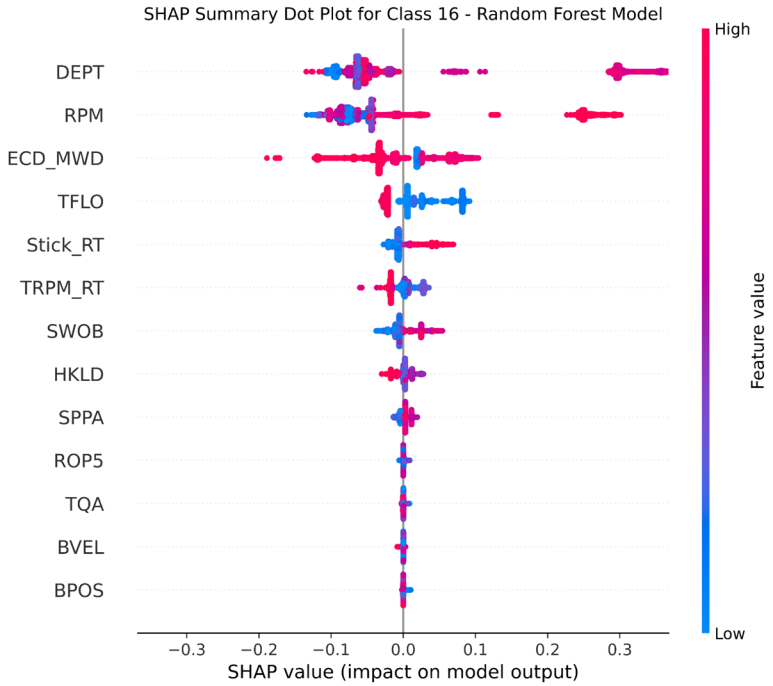


Fig. 22 SHAP summary dot plot for class 16 (Sleipner Fm.)—visualizing the impact of features on predictions for Sleipner formation top using SHAP values

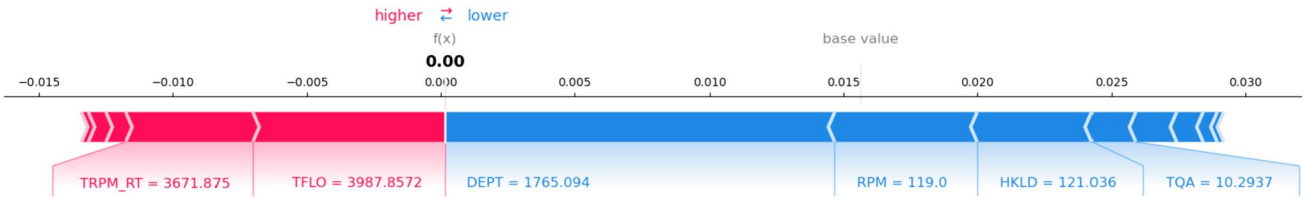


Fig. 23 SHAP force plot for random forest model explanation

for DEPT decreases as the feature value increases, while the SHAP value for RPM increases as the feature value increases. Overall, the graph provides a useful visualization of the impact of various parameters on model output. This information can be used to identify the model's most important parameters and understand how the model is making predictions.

This plot illustrates the contributions of individual features to the model prediction. Positive SHAP values push the model prediction above the expected value, while negative SHAP values pull it below. Components with larger absolute SHAP values have a greater impact on the model output for this specific instance. Figure 23 is a Force plot showing the SHAP values for different features: DEPT, RPM, HKLD, TQA, TRPM_RT, and TFLO. The graph shows that DEPT and RPM have the highest impact on the model output, followed by HKLD and TQA. The model's prediction accuracy increases by increasing parameters DEPT, RPM, HKLD, and TQA. Also, reducing the TFLO and TRPM_RT parameters increases the model's prediction accuracy.

Here is a summary of key points about the Random Forest model analysis:

- Permutation Feature Importance (PFI) identified DEPT as the most impactful feature, with RPM also being highly important. This aligns with the SHAP analysis.
- SHAP values highlighted DEPT and RPM as the main drivers of model predictions. TFLO also had a noticeable impact.
- The SHAP Summary Dot Plot visualized the impact of different features on predictions for the Sleipner formation top. DEPT, RPM, ECD_MWD, and TFLO had the highest SHAP values, indicating they are most important for the model.
- The SHAP Force Plot illustrated how individual features contribute to the model's predictions. It showed that DEPT and RPM had the greatest influence, followed by HKLD and TQA. Increasing DEPT, RPM, HKLD, and TQA increases prediction accuracy, while decreasing TFLO and TRPM_RT also improves accuracy.

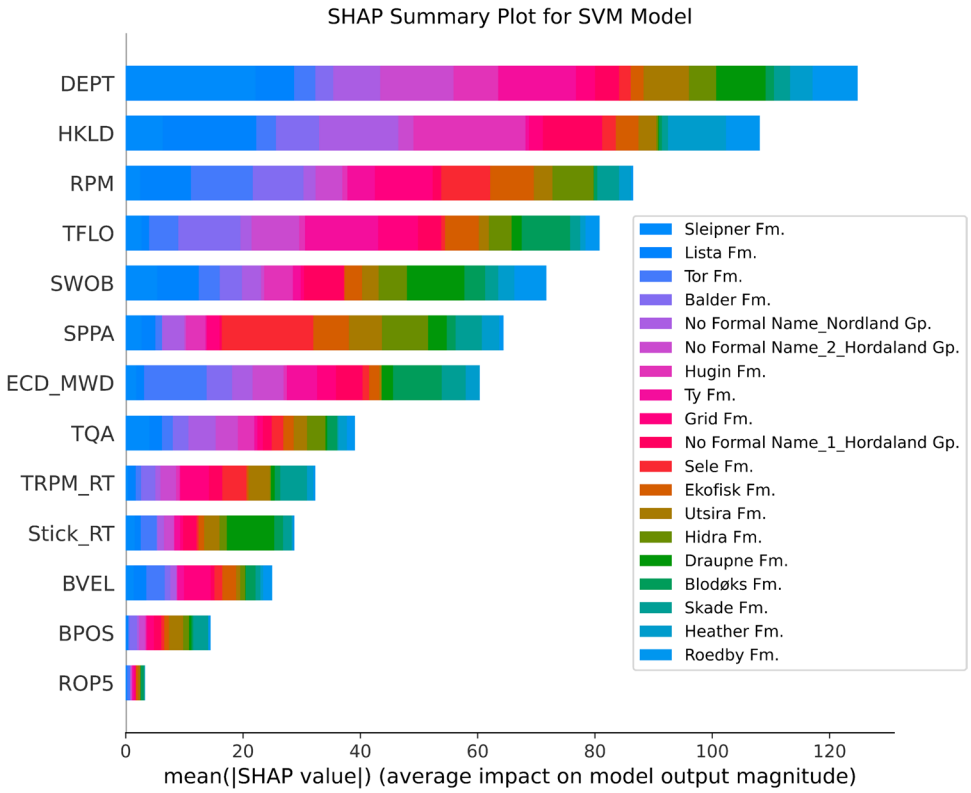


Fig. 24 SHAP feature importance plot for SVM

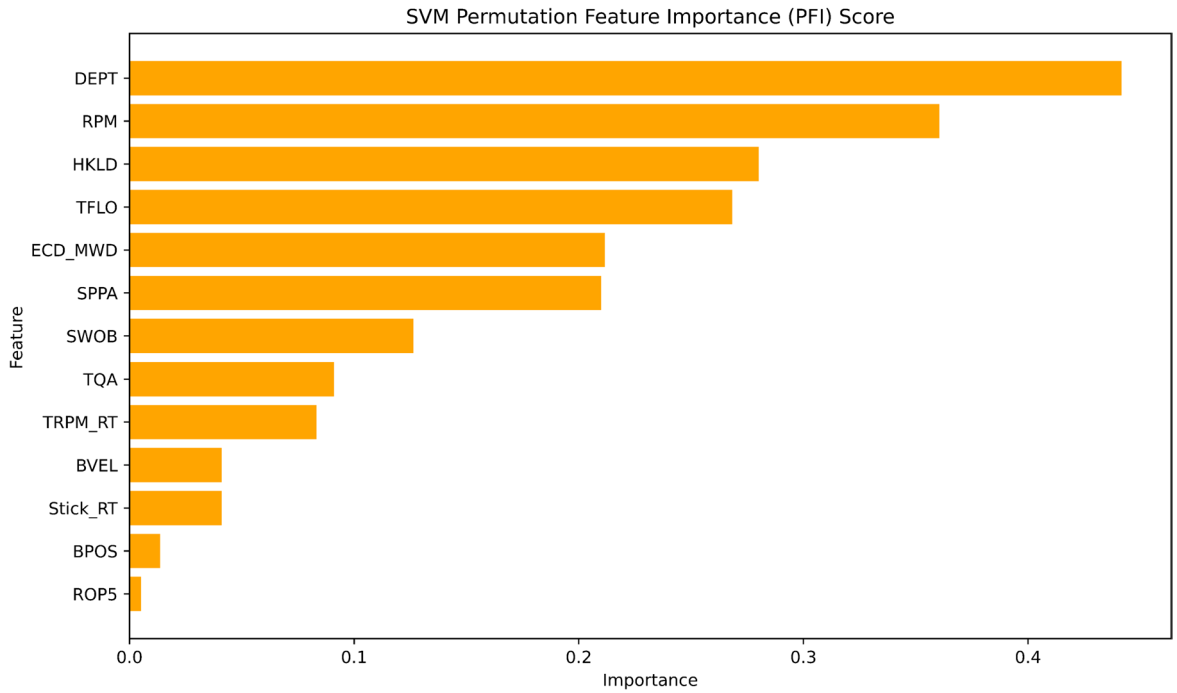


Fig. 25 PFI (permutation feature importance) bar chart for SVM

In summary, both PFI and SHAP analyses consistently pointed to DEPT and RPM as the most impactful features in the Random Forest model. The visual plots provided further confirmation and granular details on how the parts affect the predictions. This information can guide feature engineering efforts to improve model performance.

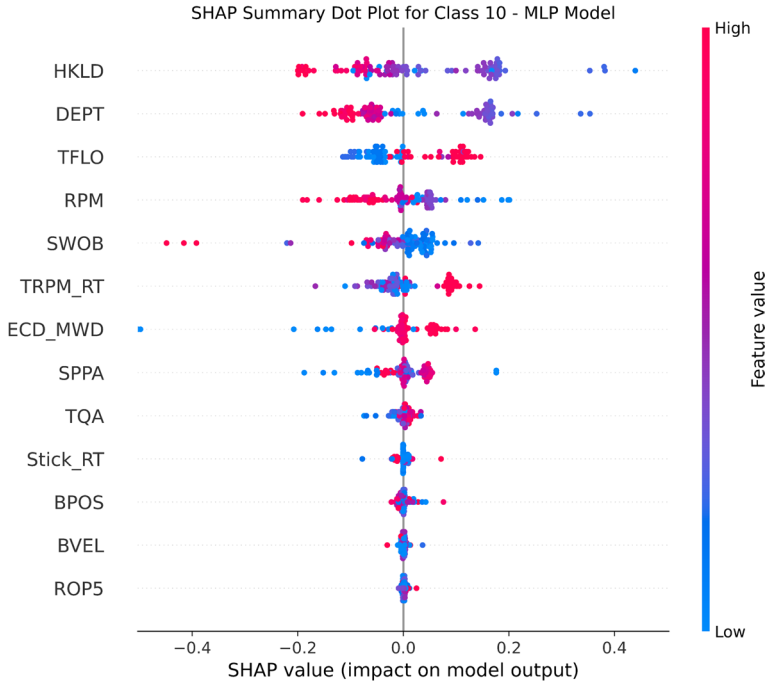


Fig. 26 SHAP summary dot plot for class 10 (No Formal Name_1_Hordaland Gp.)- Visualizing the impact of features on predictions for No Formal Name_1_Hordaland Gp. Formation top using SHAP values

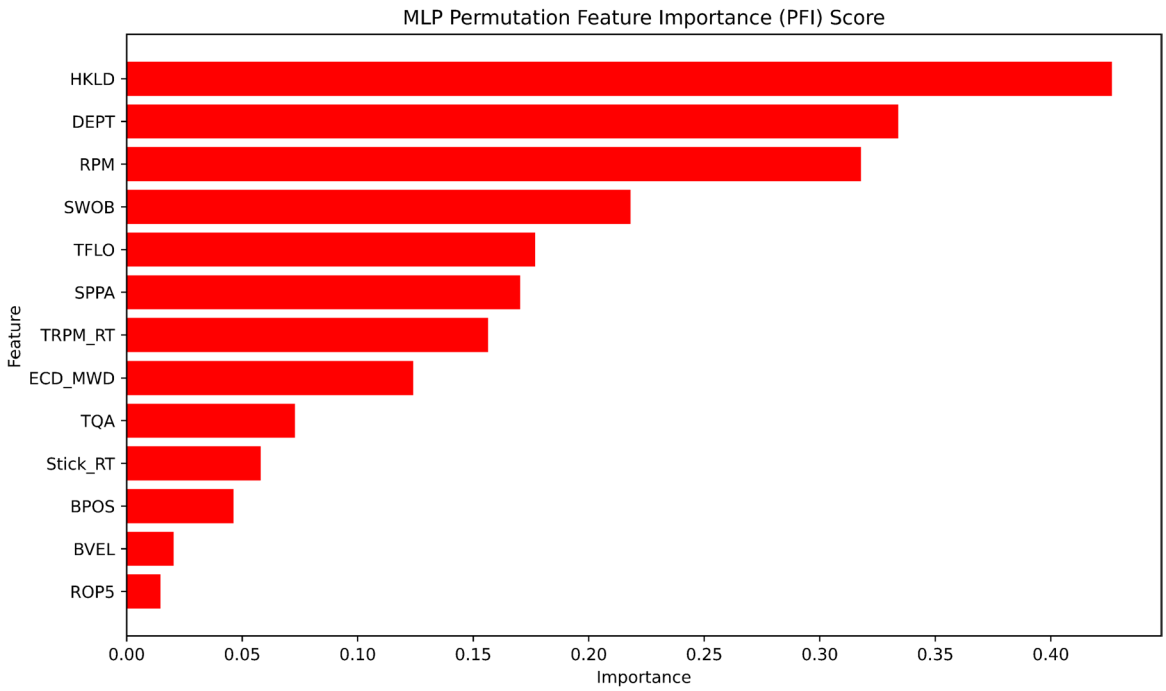


Fig. 27 PFI (permutation feature importance) bar chart for MLP

- **SVM**
  - DEPT, HKLD, and SWOB emerged as the most relevant parameters by permutation importance. This indicates their role in orienting the SVM model's maximum margin hyperplanes.
  - SHAP feature importance also pointed to DEPT, HKLD, and RPM as key features. TFLO was also identified as impactful, agreeing with the PFI findings.

The SVM SHAP plot above indicates DEPT, HKLD, and RPM as key parameters based on their high SHAP values.

The influence of TFLO is also observable. The parameters with the highest average impact for each class are:

- Sleipner Fm.: DEPT
- Lista Fm.: HKLD
- Tor Fm.: RPM
- Balder Fm.: TFLO
- No Formal Name_Nordland Gp.: SPPA
- No Formal Name_2_Hordaland Gp.: ECD_MWD
- Hugin Fm.: TQA
- Ty Fm.: TRPM RT
- Grid Fm.: Stick_RT
- No Formal Name_1_Hordaland Gp.: BVEL
- Sele Fm.: BPOS
- Ekofisk Fm.: ROP5

These parameters are the most important for predicting the target class for each formation. For example, DEPT is the most important parameter for predicting the target class for Sleipner Fm., while HKLD is the most important parameter for predicting the target class for Lista Fm.

The PFI chart shows the considerable declines in SVM accuracy when ROP5 and BVEL were used, validating their low permutation importance. DEPT and HKLD also exhibited a noticeable extent.

Here is a summary of the key points about the SVM model analysis:

- The permutation importance analysis identified DEPT, HKLD, and SWOB as the most relevant parameters for orienting the SVM model's maximum margin hyperplanes.
- SHAP feature importance also highlighted DEPT, HKLD, and RPM as key features, along with TFLO. This aligns with the permutation importance findings.
- The SHAP plot shows that DEPT, HKLD, and RPM have a high impact based on their SHAP values. TFLO's influence is also observable.
- The parameters with the highest average impact for predicting each formation class are:
  - Sleipner Fm: DEPT
  - Lista Fm: HKLD
  - Tor Fm: RPM
  - Balder Fm: TFLO
  - No Formal Name_Nordland Gp: SPPA

The permutation importance chart validated the low importance of ROP5 and BVEL based on the decline in accuracy when they were used. DEPT and HKLD also showed noticeable importance.

In summary, DEPT, HKLD, RPM, and TFLO emerge as key parameters in the SVM analysis, having an important role in prediction and model orientation. The permutation and SHAP methods validate one another in identifying influential features.

- **MLP**
  - PFI showed DEPT, HKLD, and RPM as central features in the neural network-based MLP model.
  - SHAP analysis corroborated the high importance of HKLD, DEPT, and RPM in influencing predictions.

The above SHAP plot shows that some features, such as HKLD, DEPT, and RPM, impact the model output more than others, such as ECD_MWD and TFLO. The scatter plot also shows that some features positively affect the model output (e.g., HKLD and DEPT), while others negatively impact (e.g., RPM and BVEL). Also, the parts BVEL, ROP5, and TQA have a negative SHAP value, which decreases the model output.

The PFI chart displayed the substantial drops in MLP accuracy when BVEL and ROP5 were used, highlighting their insignificance by permutation importance.

Here is a summary of the key points of the MLP model analysis:

- PFI and SHAP analysis showed that HKLD, DEPT, and RPM were the most important features influencing the MLP model's predictions. They had the greatest positive impact.
- BVEL, ROP5, and TQA were found to have a negative impact, decreasing the model output. The PFI chart showed substantial drops in MLP accuracy when BVEL and ROP5 were permuted, indicating they were insignificant for the model.
- The SHAP summary plot visually depicts the impact of different features on the model's predictions. HKLD, DEPT, and RPM had high positive SHAP values, significantly increasing the projection. BVEL had a strong negative impact.
- The PFI feature importance chart ranks the features by their importance, with the components causing the biggest drop in accuracy when removed as being the most important. This aligns with the high significance of HKLD, DEPT, and RPM in the SHAP and PFI analyses.

In summary, HKLD, DEPT, and RPM are the most significant drivers of the MLP model, while BVEL and ROP5 provide little predictive value. The SHAP and PFI techniques produced consistent results about the main factors influencing the model. In summary, DEPT, RPM, and HKLD consistently emerged as highly influential parameters across all four machine learning models by both PFI and SHAP

analyses. This highlights their overarching significance as inputs for reliable geological formation tops prediction. The strong agreement between the PFI and SHAP methods reinforces the validity of this feature importance analysis. These actionable insights can direct efforts toward feature engineering and selection to enhance model performance.

The key findings across the models are summarized below:

- KNN Model

The PFI and SHAP analyses aligned in identifying DEPT, RPM, HKLD, SPPA, and SWOB as the most important features. They had the strongest influence in determining the nearest neighbors and driving the model's predictions. SHAP analysis also highlighted ECD_MWD as having a high impact. Higher feature values generally tended to increase model output, except BVEL. The PFI analysis further validated the low significance of ROP5 and TFLO based on permutations.

- Random Forest

PFI and SHAP identified DEPT and RPM as the most impactful features influencing the Random Forest model's predictions. TFLO also had a noticeable impact. The SHAP plot visualized the effect of different parts on forecasts for the Sleipner formation top. Increasing DEPT, RPM, HKLD, and TQA values improved prediction accuracy, while decreasing TFLO and TRPM_RT also helped accuracy.

- SVM

Permutation importance analysis identified DEPT, HKLD, and SWOB as most relevant for orienting the SVM model's maximum margin hyperplanes. SHAP feature importance highlighted DEPT, HKLD, RPM, and TFLO as key drivers. The parameters most influential in predicting each target formation were also identified. Further, the low significance of ROP5 and BVEL was validated.

- MLP

The analyses showed HKLD, DEPT, and RPM as the most important features influencing the neural network-based MLP model's predictions. On the other hand, BVEL, ROP5, and TQA were found to have a negative impact, decreasing model output. Based on substantial accuracy declines, the PFI chart validated the low permutation importance of BVEL and ROP5.

In summary, DEPT, RPM, and HKLD consistently emerged as highly influential variables across the machine learning models, with some model-specific variations—the PFI and SHAP methods aligned in identifying impactful features for each model.

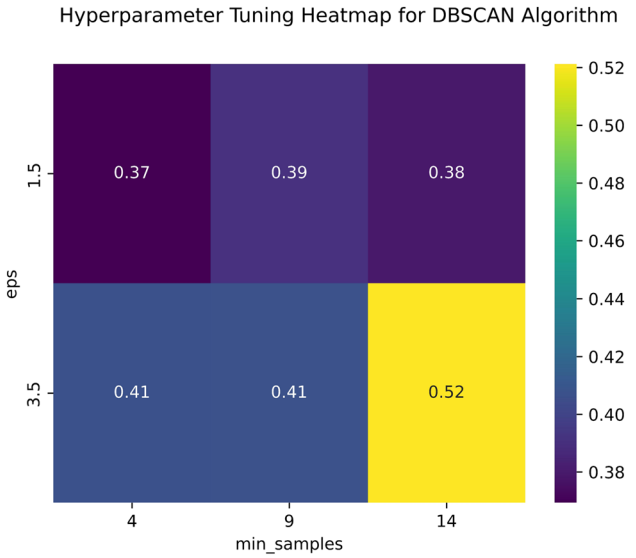


Fig. 28 Silhouette scores for different combinations of eps and min_samples

Clustering using DBSCAN

DBSCAN (Density-Based Spatial Clustering of Applications with Noise) is an algorithm designed to identify clusters in data based on density, making it particularly effective when the number of sets is unknown or when groups exhibit varying shapes and sizes. The steps involved in employing DBSCAN are as follows. Firstly, determine parameters such as epsilon (ε), defining the maximum distance between two data points to consider them neighbors, and minPts, specifying the minimum number of data points in a neighborhood required to be considered a core point. Subsequently, calculate the distances between each pair of data points in the dataset using an appropriate distance metric. Identify core points that meet the minPts criterion, indicating they have sufficient neighbors and are likely to belong to clusters. Connect all core points that are mutually reachable to form the same cluster. Data points that are not part of any core point and lack enough neighbors to create a cluster are deemed noise. DBSCAN excels at handling groups with diverse shapes and sizes while detecting noise in data. However, parameter sensitivity may affect its performance, especially in high-dimensional spaces. Thus, carefully selecting parameters and data preprocessing is crucial before applying DBSCAN. After clustering without hyperparameter tuning, the silhouette score obtained is 0.87, signifying good clustering results for geological formation tops. Yet, there remains a curiosity to explore the impact of hyperparameter tuning on these results.

Hyperparameter tuning for DBSCAN

Hyperparameter tuning is a pivotal step in optimizing the performance of machine learning models, including clustering algorithms like DBSCAN, as illustrated in Fig. 28. For DBSCAN, the two key hyperparameters under consideration are epsilon (ε) and the minimum number of samples (minPts). Epsilon determines the maximum distance between two points for them to be considered neighbors, while minPts specifies the minimum number of samples required to form a core point. Tuning these hyperparameters enables fine-tuning DBSCAN behavior, resulting in improved clustering outcomes. The choice of epsilon impacts cluster size and density; smaller epsilon values lead to denser clusters, whereas larger epsilon values result in larger and more sparse clusters. Adjusting minPts affects the minimum density required for a point to be considered a core point, with higher minPts values imposing more stringent density requirements, yielding smaller and denser clusters. Hyperparameter tuning facilitates the identification of the optimal combination of epsilon and minPts that aligns with underlying patterns and structures in the data. This process addresses the trade-off between overfitting (finding excessive small clusters) and underfitting (forming few or no meaningful clusters). The parameter values that yield the best clustering results can be identified through iterative adjustments and evaluation of clustering performance metrics, such as silhouette score or within-cluster sum of squares. Optimal hyperparameter tuning enhances the accuracy and effectiveness of clustering in machine learning modeling, revealing hidden patterns, meaningful data groups, and insights into underlying structures. Achieving the right balance between epsilon and minPts results in more accurate and robust clustering outcomes, which are valuable in applications like customer segmentation, anomaly detection, and recommendation systems. Post hyperparameter tuning, the silhouette score increased to 0.89, improving the pre-tuning score. Table 14 displays silhouette score values before and after hyperparameter tuning. At the same time, Fig. 29 depicts the distribution of data per geological formation top class through a box-and-whisker plot for true labels and clustering model results.

Table 14 The result of the hyperparameter tuning using the DBSCAN algorithm

Eps	min_samples	silhouette_score	n_cluster	n_noise
3.5	4	0.41	3	19
3.5	9	0.41	4	11
3.5	14	0.52	4	4
1.5	4	0.37	5	10
1.5	9	0.39	6	5
1.5	14	0.38	6	4

Based on the hyperparameter tuning results for the DBSCAN algorithm, Table 14 provides information on different combinations of hyperparameters (eps and min_samples) and their corresponding silhouette scores, number of clusters (n_cluster), and number of noise points (n_noise). The silhouette score is a metric that measures how well-defined the clusters are, with higher values indicating better-defined groups.

Here are some key observations and findings that we can conclude from Table 14:

1. Best Performing Configuration:
   - The configuration with eps = 3.5 and min_samples = 14 achieved the highest silhouette score of 0.521215.
   - This configuration resulted in 4 clusters (n_cluster) with 4 noise points (n_noise).
2. Effect of Eps and Min_samples:
   - Lower values of min_samples (e.g., 4) resulted in more clusters and noise points.
   - Smaller values of eps (e.g., 1.5) also led to more clusters and noise points, but with varying silhouette scores.
3. Trade-off between silhouette score and number of clusters:
   - Higher silhouette scores were generally associated with fewer clusters, suggesting a trade-off between cluster quality and quantity.
   - The configuration with eps = 3.5 and min_samples = 14 achieved a good balance between a high silhouette score and a reasonable number of clusters.
4. Sensitivity to hyperparameter values:
   - The algorithm's performance was sensitive to changes in both eps and min_samples, highlighting the importance of careful hyperparameter tuning.
5. Variability in results:
   - Different hyperparameter configurations led to a range of silhouette scores, indicating variability in the algorithm's ability to define meaningful clusters under different settings.

Box whisker plots depicting the distribution of drilling parameters across clusters. Each box represents a cluster, with variables such as DEPT, ROP5, HKLD, SWOB, TQA, RPM, BPOS, BVEL, SPPA, TFLO, TRPM_RT, and Stick_RT analyzed for variations among clusters (Table 15).

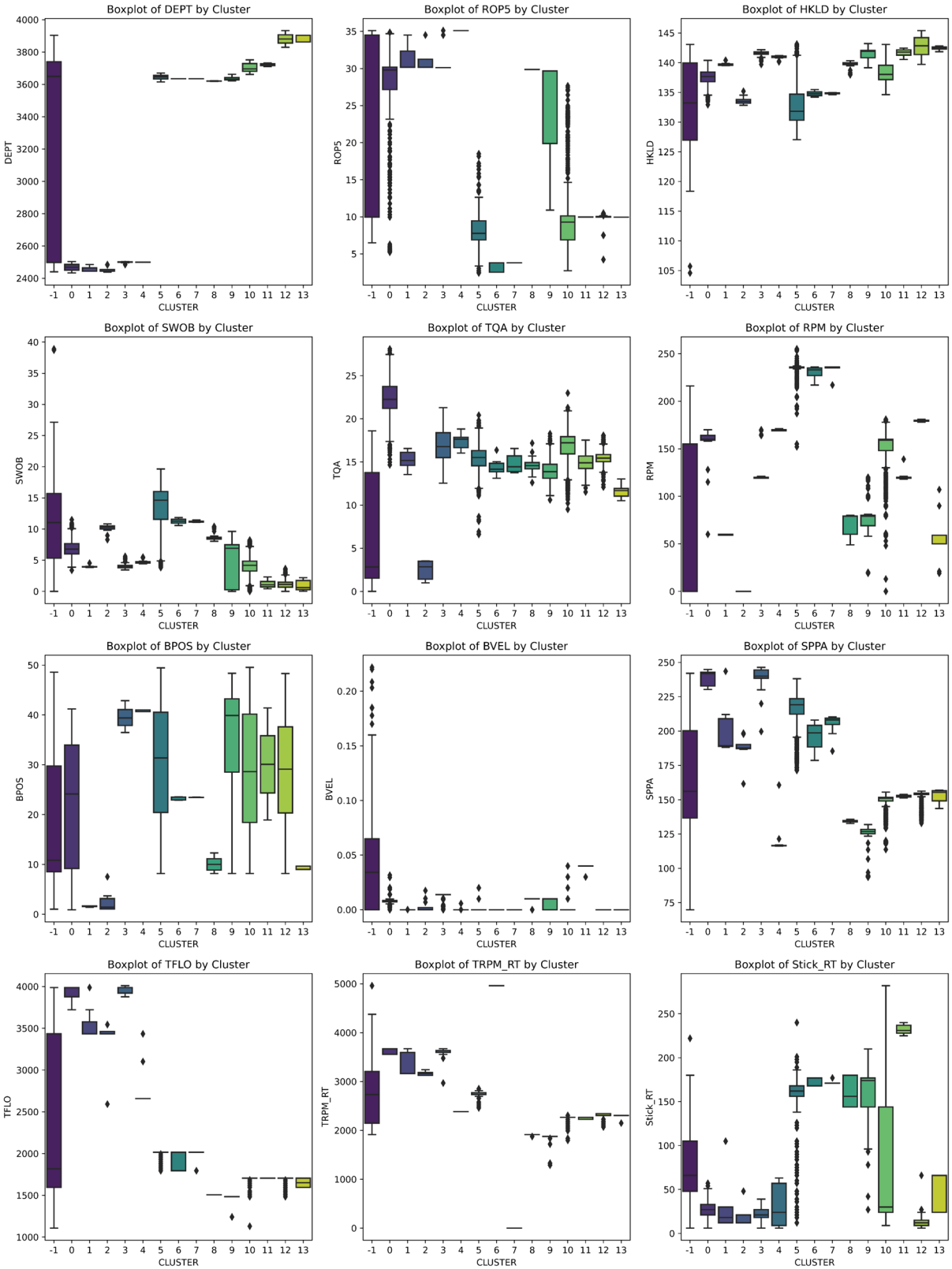


Fig. 29 Box Whisker Plots of Various Parameters Grouped by Cluster

Table 15 Silhouette score comparison for DBSCAN

DBSCAN clustering	Silhouette score
Before hyperparameter tuning	0.40
After hyperparameter tuning	0.52

Formatting of mathematical components

Confusion matrix

In machine learning, the confusion matrix is a fundamental metric (31), ubiquitously employed in diverse fields like computer vision, natural language processing (NLP), acoustics, and various scientific and engineering applications (Moazzeni and Haffar 2015; Oloso et al. 2017). This matrix is a crucial tool for evaluating model performance by juxtaposing predicted and actual values. A cross table meticulously documents the occurrences between the two raters, representing the authentic and indicated classifications. It delineates the percentages of four distinct classification outcomes: true positive (TP), false positive (FP), true negative (TN), and false negative (FN).

Precision, a key metric, denotes the percentage of instances our model predicts as Positive and are indeed Positive. This metric gauges the model's reliability when it indicates a positive outcome.

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FT}} \tag{1}$$

Recall is a metric that gauges the model's predictive accuracy specifically for the positive class. It quantifies the model's ability to identify all dataset-positive instances.

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}} \tag{2}$$

Accuracy assesses the extent to which the model correctly predicts outcomes across the entire dataset, with values ranging between 0 and 1. The complement of accuracy, representing the proportion of incorrect predictions, is called the Misclassification Rate.

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}} \tag{3}$$

The F1-Score evaluates the model's classification performance based on the information derived from the confusion matrix. In multi-class cases, it should encompass all the individual classes.

$$\text{F1Score} = \left(\frac{2}{\text{Precision}^{-1} + \text{Recall}^{-1}}\right) = 2 \times \left(\frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}\right) \tag{4}$$

Random forest algorithm

Building on the integration of bagging with DT-based learners in Random Forest (RF), the RF algorithm introduces an additional layer to the Decision Trees (DT) training process by incorporating the selection of random attributes (Abdelgawad et al. 2019a). RF is lauded for its simplicity, ease of implementation, low computational cost, and robust performance across various practical tasks. RF extends the conventional DT method by amalgamating multiple DTs to enhance prediction accuracy. A DT, being a typical single classifier, necessitates the creation of a model based on training data for classification purposes. Once the DT model is established, it is applied to classify unknown sample data. To mitigate the risk of overfitting, the pruning process is employed to trim certain subtrees or leaf nodes within the DT model. This process simplifies the model, thereby preventing overfitting. The ID3 algorithm uses information gain as the feature evaluation criterion during DT node splitting. The feature with the highest information gain is chosen as the test attribute, and the information gain calculation is grounded in information entropy. In the context of the ID3 algorithm, which generates child nodes recursively from the root node, the process unfolds from top to bottom until a leaf node is reached. The selected feature evaluation criteria play a pivotal role in this process, guiding the creation of child nodes. The information gain is determined based on information entropy, where X is considered a discrete random variable with finite values, and its probability distribution is delineated.

$$P\left(X = X_i\right) = p_i, i = 1, 2, , \ldots, n, \tag{5}$$

The entropy of the random variable X is defined as follows:

$$H(X) = -\sum_{i=1}^{n} p_i \log p_i \tag{6}$$

$$\lim_{n\to\infty} PE^{*} = P_{xy}\left(P\Theta(k(X,\Theta) = Y) - maxP\Theta\left(k_{j\neq Y}(X,\Theta) = J\right) < 0\right) \tag{7}$$

where nn is the number of trees in the forest.

Support vector machine

A support vector machine (SVM) constructs a hyperplane or set of hyperplanes in a high or infinite-dimensional space, applicable for classification, regression, or other related objectives. The concept revolves around achieving a robust separation through a hyperplane that maintains the maximum distance to any class's nearest training data points,

known as the functional margin. Generally, a larger margin corresponds to a lower generalization error in the classifier. The equation below illustrates the decision function for a linearly separable problem, highlighting three samples situated on the margin boundaries, referred to as "support vectors." The primal problem solved by support vector classification (SVC) is as follows:

$$min_{w,b,\zeta} \frac{1}{2} w^T w + C \sum_{i=1}^{n} \zeta_i \tag{8}$$

The intuitive objective is to maximize the margin (achieved by minimizing $\|w\|^2 = w^{\mathrm{T}} w$), all the while introducing a penalty when a sample is misclassified or falls within the margin boundary. Ideally, we aim for the value of $y_i\left(w^T \varnothing\left(x_i\right) + b\right)$ to be ≥ 1 for all samples, signifying a perfect prediction. However, real-world problems are often not perfectly separable with a hyperplane, so we allow some samples to be at a distance ζ_{i} from their correct margin boundary. The penalty term C regulates the strength of this penalty and, consequently, serves as an inverse regularization parameter (Patidar et al. 2023).

K-nearest neighbors algorithm

For this classifier type, having a training set that is not excessively small and possesses a discerning distance is crucial. K-nearest neighbors (KNN) demonstrates effective performance in solving multi-class problems simultaneously. Determining an optimal value for the parameter K is pivotal for achieving the best classifier performance. This optimal value of K typically hovers around $N^{1/2}$, where N represents the size of the dataset (Abdelgawad et al. 2019b).

Discussion

Interpretation of result analysis

Support vector machine

The support vector machine (SVM) model exhibited a commendable accuracy of 0.99 on the training dataset, showcasing its proficiency in accurately predicting formation tops. However, its performance on the blind dataset diminished to 0.95, pointing toward difficulties in generalizing the model to unfamiliar data. SVM's strengths lie in distinguishing between different classes within intricate feature spaces. Conversely, its weaknesses encompass prolonged training times, particularly on large datasets, and the necessity for meticulous parameter selection.

K-nearest neighbor

The k-nearest neighbors (KNN) model demonstrated an impressive accuracy of 0.99 on the training dataset. Nevertheless, the accuracy dropped to 0.93 when applied to the blind dataset. KNN's notable strengths lie in its straightforward concept, implementation, and capability to handle nonlinear data patterns. However, it is susceptible to the impact of outlier data, and the computational demands escalate when making predictions on extensive datasets.

Random forest

The random forest model attained an accuracy of 0.99 on the training dataset and 0.91 on the blind dataset. Its capabilities include effectively handling data with nonlinear and complex features, and it excels at mitigating overfitting by employing an ensemble of decision trees. However, drawbacks include lengthier training times than alternative machine learning models and reduced interpretability.

Multi-layer perceptron

The multilayer perceptron (MLP) model obtained an accuracy of 1 on the training dataset and 0.99 on the blind dataset, showcasing its capability to model complex relationships between features with hidden layers. MLP is versatile in addressing various problem types. However, it demands meticulous parameter selection, entails extended training times, particularly on large datasets, and is susceptible to overfitting if not appropriately regularized.

Cluster determination method using DBSCAN

DBSCAN, a density-based clustering method, discerns dense and noisy clusters by evaluating neighborhood density in the feature space. The silhouette score of 0.52 achieved by DBSCAN indicates its proficiency in grouping data into dense clusters. Interpreting geological formation tops through DBSCAN offers insights into spatial distribution patterns and interconnections between formation tops. DBSCAN's strengths encompass its ability to identify intricate clusters without a predetermined cluster count. However, it is sensitive to distance and density parameters and may generate irrelevant clusters if not appropriately configured.

Advantages and disadvantages of each model

Machine learning models have distinct advantages and disadvantages, influencing their suitability for different tasks. The following Table outlines the key characteristics of several models (Table 16).

Table 16 Advantages and disadvantages of each model

Model	Advantages	Disadvantages
Support vector machine	Effective in separating different classes in complex feature spaces Robust against overfitting by using a margin function	Long training times on large datasets Requires proper parameter selection for optimal results
K-Nearest Neighbor	Simple in concept and implementation Can handle nonlinear data	Susceptible to the influence of outlier data Increased computation when predicting large datasets
Random Forest	Can handle data with nonlinear and complex features Not prone to overfitting due to the ensemble of decision trees	Longer training times compared to other machine learning models Lower interpretability
Multilayer Perceptron	Can model complex relationships with hidden layers Flexible in modeling various types of problems	Requires proper parameter selection Long training times on large datasets, prone to overfitting if not properly regularized

Comparison of machine learning model performance with DBSCAN

Geological formation tops prediction accuracy

When assessing the accuracy of each model in comparison to DBSCAN, it becomes evident that machine learning models consistently exhibit higher accuracy on the blind dataset than DBSCAN. This observation implies that machine learning models possess a superior capability to discern and learn existing patterns in geological formation tops data in the NCS, surpassing the performance of clustering methods such as DBSCAN. Several factors contribute to the performance disparities between machine learning models and DBSCAN. Machine learning models excel in extracting intricate patterns and demonstrating better generalization to unknown data. In contrast, DBSCAN's performance is more sensitive to parameter settings, particularly distance and data density. This comparison underscores the potential of machine learning models to provide more accurate predictions in the context of geological formation tops, emphasizing their effectiveness in capturing complex relationships within the data compared to clustering techniques like DBSCAN.

Efficiency and computational speed

An examination of the computational time of each model in contrast to DBSCAN reveals a nuanced picture. Machine learning models exhibit varying training times, while DBSCAN typically demonstrates faster cluster production. However, the scalability of machine learning models for larger datasets is a crucial consideration. Machine learning models leverage advantages such as parameter flexibility and the ability to handle extensive datasets through algorithm optimization and computational parallelism.

Conclusion on the relative performance of model-machine learning and DBSCAN in formation tops prediction

Upon comprehensive comparison, it becomes evident that machine learning models, including SVM, KNN, Random Forest, and MLP, consistently outperform DBSCAN regarding accuracy on the blind dataset. Nevertheless, it is essential to acknowledge that DBSCAN, as a clustering method, provides valuable insights into the spatial distribution patterns of geological formation tops. Therefore, selecting the most suitable model for predicting formation tops in the NCS should consider the accuracy, computational efficiency, and the necessity for spatial pattern interpretation. The overarching conclusion underscores the importance of a balanced evaluation to meet the specific requirements of the formation tops prediction task.

Conclusion on the suitability of the best model in geological formation tops prediction in the NCS

The modeling results for SVM, KNN, Random Forest, and MLP revealed varying performance on both the training and blind datasets. MLP emerged as the top performer on the blind dataset, achieving a perfect accuracy score of 0.99. In contrast, SVM, KNN, and Random Forest exhibited lower accuracy levels on the blind dataset. The comparison and interpretation of results highlight the superior potential of machine learning models in formation tops prediction within the NCS compared to DBSCAN. However, determining the best model necessitates comprehensively considering factors such as accuracy, computational efficiency, and the interpretability of spatial patterns. In this study, MLP demonstrated the highest accuracy on the blind dataset. The implications of this research extend significantly to the oil and gas industry, offering valuable insights for developing reservoir management strategies and more informed decision-making. Accurate formation top prediction is crucial in identifying potential reservoir zones, understanding rock

properties, and optimizing the development of oil and gas fields. Although MLP exhibited superior performance, further assessment and testing on larger datasets are imperative to validate the reliability and generalizability of machine learning models. Moreover, SVM, KNN, and Random Forest showed comparable performance in formation top prediction within the NCS. Therefore, when recommending the best machine learning model, considerations should encompass industry requirements, computational capabilities, and the interpretability of results. This research represents a significant step toward enhancing the efficacy of geological formation top prediction in the NCS and lays the foundation for more advanced applications in the NCS.

Conclusions

This study explored and compared several machine learning models, including Support Vector Machine (SVM), K-Nearest Neighbors (KNN), Random Forest, and Multi-Layer Perceptron (MLP), for predicting geological formation tops in the Norwegian Continental Shelf (NCS). The models were evaluated based on their accuracy on both a test dataset and a blind dataset.

The key findings are:

- The MLP model demonstrated the highest accuracy on the blind dataset with a perfect score 0.99. In contrast, SVM, KNN, and Random Forest exhibited lower accuracy levels of 0.95, 0.93, and 0.91, respectively, on the blind set.
- On the test dataset, MLP achieved the highest accuracy of 1, followed by Random Forest (0.99), SVM (0.99), and MLP (0.99).
- The superiority of MLP on the blind set indicates its stronger capability to generalize to new unseen data compared to the other models. This highlights the potential of neural network-based approaches for formation tops prediction.
- All the machine learning models consistently outperformed the DBSCAN clustering algorithm regarding predictive accuracy on the blind dataset. This emphasizes their effectiveness in discerning intricate patterns in the geological formation top data.
- The feature importance analyses using SHAP and PFI revealed DEPT, RPM, and HKLD as the most influential parameters across models. This provides direction for feature engineering efforts.

In conclusion, while the MLP model achieved the highest accuracy on the blind dataset, confirming its reliability necessitates additional testing on more extensive datasets. Moreover, factors such as computational efficiency and interpretability must also guide the selection of the most appropriate model. The research underscores the promise of machine learning for enhanced formation tops prediction within the NCS. Further work can build on these findings to develop more robust and generalizable models for practical applications.

Funding The research, authorship, and publication of this article were not supported by external funding.

Declarations

Conflict of interest The authors have not declared a conflict of interest.

References

Abdelgawad K, Elkatatny S, Moussa T, Mahmoud M, Patil S (2019a) Real time determination of rheological properties of spud drilling fluids using a hybrid artificial intelligence technique. J Energy Resour Technol. https://doi.org/10.1115/1.4042233

Abdelgawad KZ, Elzenary M, Elkatatny S, Mahmoud M, Abdulraheem A, Patil S (2019b) New approach to evaluate the equivalent circulating density (ECD) using artificial intelligence techniques. J Petrol Explor Prod Technol 9:1569–1578. https://doi.org/10.1007/s13202-018-0572-y

Akbulut Y, Sengur A, Guo Y, Smarandache F (2017) NS-k-NN: neutrosophic set-based k-nearest neighbors classifier. Symmetry 9:179. https://doi.org/10.3390/sym9090179

Al-AbdulJabbar A, Elkatatny S, Mahmoud M, Abdulraheem A (2018) Predicting formation tops while drilling using artificial intelligence. SPE Kingdom of Saudi arabia annual technical symposium and exhibition, SPE-192345-MS. https://doi.org/10.2118/192345-MS.

Alsaihati A, Elkatatny S, Mahmoud AA, Abdulraheem A (2021) Use of machine learning and data analytics to detect downhole abnormalities while drilling horizontal wells, with real case study. J Energy Resour Technol 143(4):043201. https://doi.org/10.1115/1.4048070

Aniyom E, Chikwe A, Odo J (2022) Hybridization of optimized supervised machine learning algorithms for effective lithology. SPE Nigeria annual international conference and exhibition. SPE-212019-MS. https://doi.org/10.2118/212019-MS.

Breiman L (2001) Random forests. Mach Learn 45(1):5–32. https://doi.org/10.1023/A:1010933404324

Duro DC, Franklin SE, Dubé MG (2012) A comparison of pixel-based and object-based image analysis with selected machine learning algorithms for the classification of agricultural landscapes using

SPOT-5 HRG imagery. Remote Sens Environ 118:259–272. https://doi.org/10.1016/j.rse.2011.11.020
Elkatatny S (2018) New approach to optimize the rate of penetration using artificial neural network. Arab J Sci Eng 43:6297–6304. https://doi.org/10.1007/s13369-017-3022-0
Feng Q, Liu J, Gong J (2015) UAV remote sensing for urban vegetation mapping using random forest and texture analysis. Remote Sens 7:1074–1094. https://doi.org/10.3390/rs70101074
Franco Lopez H, Ek AR, Bauer ME (2001) Estimation and mapping of forest stand density, volume and cover type using the k-Nearest Neighbors method. Remote Sens Environ 77:251–274. https://doi.org/10.1016/S0034-4257(01)00209-7
Ghosh A, Joshi PK (2014) A comparison of selected classification algorithms for mapping bamboo patches in lower Gangetic plains using very high-resolution WorldView 2 imagery. Int J Appl Earth Obs Geoinf 26:298–311. https://doi.org/10.1016/j.jag.2013.08.011
Huang C, Davis LS, Townshend JRG (2002) An assessment of support vector machines for land cover classification. Int J Remote Sens 23:725–749. https://doi.org/10.1080/01431160110040323
Ibrahim AF, Ahmed A, Elkatatny S (2023) Applications of different classification machine learning techniques to predict formation tops and lithology while drilling. ACS Omega 8(45):42152–42163. https://doi.org/10.1021/acsomega.3c03725
Khalifa H, Tomomewo OS, Ndulue UF, Berrehal BE (2023) Machine learning-based real-time prediction of formation lithology and tops using drilling parameters with a Web App integration. Eng 4(3):2443–2467. https://doi.org/10.3390/eng4030139
Khalifah HA, Glover PWJ, Lorinczi P (2020) Permeability prediction and diagenesis in tight carbonates using machine learning techniques. Mar Petrol Geol 112:104096. https://doi.org/10.1016/j.marpetgeo.2019.104096
Knorn J, Rabe A, Radeloff VC, Kuemmerle T, Kozak J, Hostert P (2009) Land cover mapping of large areas using chain classification of neighboring Landsat satellite images. Remote Sens Environ 113:957–964. https://doi.org/10.1016/j.rse.2009.01.010
Li C, Wang J, Wang L, Hu L, Gong P (2014) Comparison of classification algorithms and training sample sizes in urban land classification with Landsat Thematic Mapper imagery. Remote Sens 6:964–983. https://doi.org/10.3390/rs6020964
Losoya ZE, Vishnumolakala N, Noynaert SF, Medina-Cetina Z, Bukkapatnam S, Gildin E (2021) Automatic identification of rock formation type while drilling using machine learning based data-driven models. IADC/SPE Asia Pacific drilling technology conference. SPE-201020-MS. https://doi.org/10.2118/201020-MS.
Mahmoud AA, Elkatatny S, Al-Shehri D (2020) Application of machine learning in evaluation of the static Young's modulus for sandstone formations. Sustainability 12(5):1880. https://doi.org/10.3390/su12051880
Mahmoud AA, Elkatatny S, Al-AbdulJabbar A (2021) Application of machine learning models for real-time prediction of the formation lithology and tops from the drilling parameters. J Petrol Sci Eng 203:108574. https://doi.org/10.1016/j.petrol.2021.108574
Melgani F, Bruzzone L (2004) Classification of hyperspectral remote sensing images with support vector machines. IEEE Trans Geosci Remote Sens 42:1778–1790. https://doi.org/10.1109/TGRS.2004.831865
Moazzeni A, Haffar MA (2015) Artificial intelligence for lithology identification through real-time drilling data. J Earth Sci Clim Change 6(3):265. https://doi.org/10.4172/2157-7617.1000265
Oloso MA, Hassan MG, Bader-El-Den MB, Buick JM (2017) Hybrid functional networks for oil reservoir PVT characterisation. Expert Syst Appl 87:363–369. https://doi.org/10.1016/j.eswa.2017.06.014
Patidar AK, Singh S, Anand S (2023) Subsurface lithology classification using well log data, an application of supervised machine learning. Machine Intelligence and Data Science Applications, Algorithms for Intelligent Systems. Springer, Singapore. https://doi.org/10.1007/978-981-99-1620-7_18
Qian Y, Zhou W, Yan J, Li W, Han L (2015) Comparing machine learning classifiers for object-based land cover classification using very high-resolution imagery. Remote Sens 7:153–168. https://doi.org/10.3390/rs70100153
Shi D, Yang X (2015) Support vector machines for land cover mapping from remote sensor imagery. In: Li J, Yang X (eds) Monitoring and modeling of global changes: a geomatics perspective. Springer Remote Sensing/Photogrammetry. Springer, Dordrecht. https://doi.org/10.1007/978-94-017-9813-6_13
Sircar A, Yadav K, Rayavarapu K, Bist N, Oza H (2021) Application of machine learning and artificial intelligence in oil and gas industry. Petrol Res 6(4):379–391. https://doi.org/10.1016/j.ptlrs.2021.05.009
Vikara D, Khanna V (2022) Machine learning classification approach for formation delineation at the basin-scale. Petrol Res 7(2):165–176. https://doi.org/10.1016/j.ptlrs.2021.09.004
Wei C, Huang J, Mansaray LR, Li Z, Liu W, Han J (2017) Estimation and mapping of winter oilseed rape LAI from high spatial resolution satellite data based on a hybrid method. Remote Sens 9:488. https://doi.org/10.3390/rs9050488
Zhong R, Salehi C, Johnson R (2022) Machine learning for drilling applications: a review. J Natural Gas Sci Eng 108:104807. https://doi.org/10.1016/j.jngse.2022.104807
Ziadat W, Gamal H, Elkatatny S (2023) Real-time machine learning application for formation tops and lithology prediction. In: Offshore technology conference, OTC-32447-MS. https://doi.org/10.4043/32447-MS