Bollettino di Geofisica Teorica ed Applicata Vol. 60, n. 1, pp. 69-80; March 2019
DOI 10.4430/bgta0256

Comparison of different Machine Learning algorithms for lithofacies classification from well logs

P. Dell'Aversana

Eni S.p.A. Upstream and Technical Services, San Donato Milanese (MI), Italy

(Received: 19 February 2018; accepted: 23 October 2018)

ABSTRACT Machine Learning algorithms can support the work of lithofacies classification using well logs. A wide range of automatic classifiers is available for that purpose. In order to investigate about the accuracy and the effectiveness of different methods, I compare six supervised learning algorithms. Using multiple data sets of composite logs, I discuss the entire workflow applied to two wells. The workflow includes the following main steps: 1) statistical data analysis; 2) training of six classification algorithms; 3) quantitative evaluation of the performance of each individual algorithm; 4) simultaneous lithofacies classification using all the six algorithms; 5) results comparison and reporting. Using cross-validation tests and confusion matrices, I perform a preliminary ranking of the six classifiers. Although the different algorithms show different performances, all the methods produce mutually consistent classification results. Consequently, I set a comprehensive workflow that includes all the classifiers working in parallel in the same Machine Learning framework. I show through tests on real data that this "systemic approach" allows efficient training of many algorithms, easy comparison of the results, and robust classification of multiple well data. This methodology is particularly useful when quick lithofacies classification/prediction is required for making real-time decisions, such as in case of well-site geological operations.

Key words: Machine Learning, supervised methods, lithofacies classification, well logs.

1. Introduction

Machine Learning (ML) is the subfield of Artificial Intelligence (AI) that gives computers the ability to learn without being explicitly programmed (Samuel, 1959). Statistical (or mathematical) techniques are applied for retrieving a model from observed data, rather than codifying a specific set of instructions that define the model for that data (Bishop, 2006). Over the past two decades, a multitude of AI and ML methods have been applied in many sectors, such as medical, social and financial disciplines [for an extended discussion about theory and applications of AI techniques, see Russell and Norvig (2009)].

The number of applications of ML has been growing over the past 10-15 years in geosciences too, including geophysics (Aminzadeh and de Groot, 2006; Lary *et al*., 2016). Common examples of applications are seismic facies recognition (Zhao *et al*., 2015; Hall, 2016; Zhang and Zhan, 2017) and automatic interpretation of seismic data (Barnes and Laughlin, 2002).

ML has been recently used for supporting the "manual" interpretation of well logs for lithofacies classification. In supervised learning methods, one or more automatic classifiers are trained on a labelled subset of data logs; then, the learnt classification rules are applied to the entire unlabelled data set. This approach allows speeding up significantly the entire interpretation workflow. Of course, similar to other fields of applications, also in log interpretation, ML should be intended as a computation tool augmenting human skills rather than replacing them.

Considering the wide range of ML methods, it is difficult to select in advance the optimal algorithm for solving any specific classification problem. For instance, Bestagini *et al*. (2017) used gradient boosting classifier that demonstrated to be effective for working also with relatively small training data sets and with few features.

Less recently, Bohling and Dubois (2003) applied neural network and Markov chain techniques to prediction of lithofacies from well logs.

The choice to apply one ML approach or another depends on many factors, such as the quality and the size of the data, the number of labelled samples forming the training data set, the uncertainties on the data, the availability of prior information, and so forth. More simply, the ML workflow often depends on the availability of specific libraries and/or on the personal experience/ preference in using some type of algorithm. In this paper, I discuss the problem of automatic classification of lithofacies from composite well logs comparing a number (necessarily limited) of different supervised ML methods. My objective is to provide a possible tutorial workflow through a comparative approach, highlighting benefits and limitations of the different methods here considered. Furthermore, I will describe, shortly, the possible practical implications of a comprehensive ML framework in the field of operation geology.

2. Test data

I used the data of two wells labelled in the following as Well A and Well B drilled in a complex geological setting. This is characterised by narrow and elongated fault compartments with thin stacked reservoir sandstones. The hydrocarbon field has been explored by extensive multidisciplinary geophysical surveys and by several wells penetrating hydrocarbon-bearing sands of Triassic ages.

For each well, I used almost 21,000 instances, including the following types of log: sonic, Rdep (resistivity), DEN (density), NEU (neutron logs), PEF (photoelectric absorption), GR (gamma ray) and SP (spontaneous potentials).

As an example, Fig. 1 shows the resistivity log of Well A, where the oil and gas bearing layers clearly appear with high resistivity values. Fig. 2 shows some examples of normalized cross plots of the other six logs of Well A.

The lithofacies taxonomy includes six main classes, as showed in Table 1, with the respective colours used in the following images. The first three facies correspond to "Prevalent Shale", "Interbedded Sandstones/Shale" and "Interbedded Sandstones/Siltstone". The remaining three facies correspond to Sandstones partially filled by hydrocarbon with variable saturation.

The main challenge of this automatic classification test is that the above lithofacies are partially overlapped in the feature space. Consequently, distinguishing one class from another can be difficult for both a human interpreter and an automatic classifier. This appears clearly in

Table 1 - Lithofacies classes and colour legend.

Class 1: Prevalent Shale
Class 2: Interbedded Sandstones/Shale
Class 3: Interbedded Sandstones/Siltstone
Class 4: Medium Saturation Hydrocarbon
Class 5: Low Saturation Hydrocarbon
Class 6: High Saturation Hydrocarbon

Fig. 3 showing, as examples, the statistical distribution (probability density distribution curves) of sonic and spontaneous potentials (SP) log measurements included in the labelled data set. Also the distributions of the other logs (here not displayed) show similar overlap between the classes. Consequently, each individual type of measurement alone is not sufficient to classify the data. Instead, appropriate classification is possible combining all the logs.

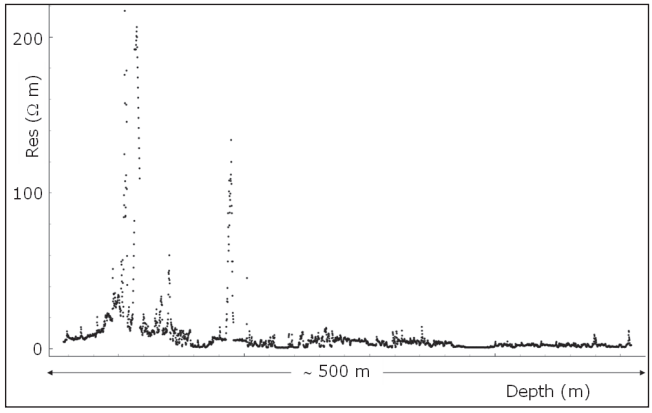

Fig. 1 - Resistivity log of Well A. The depth is a "relative depth" with respect to the first sample.

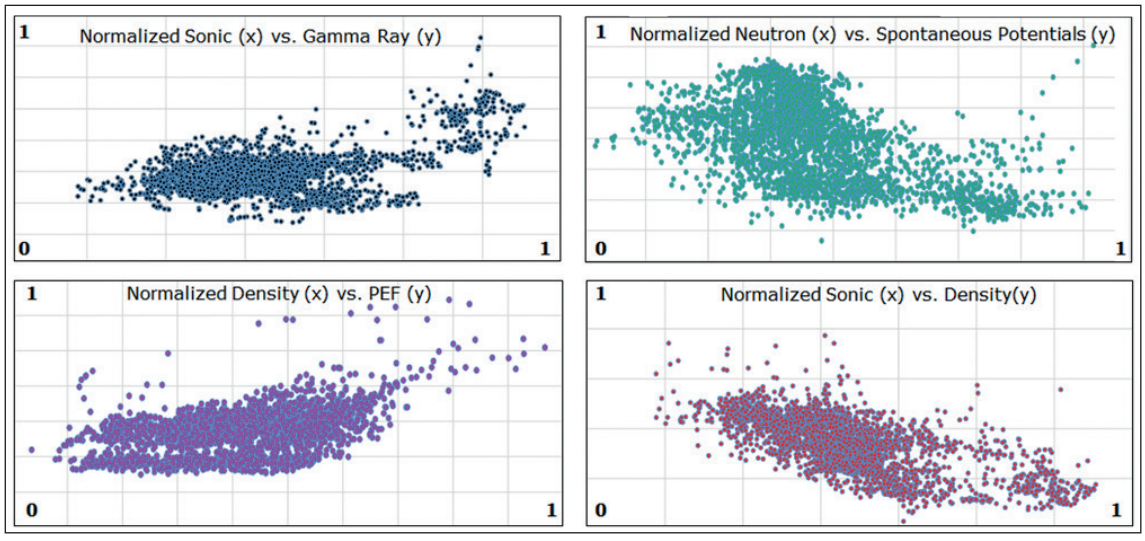

Fig. 2 - Normalised cross-plots of composite logs of Well A.

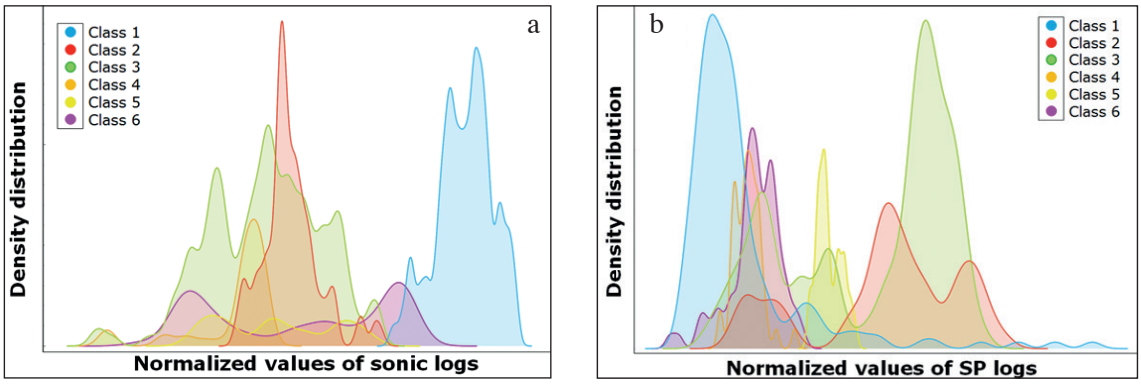

Fig. 3 - Probability density distribution of Well A: a) normalized values of sonic logs; b) normalized values of SP logs.

3. Learning algorithms

As anticipated in the previous section, all the composite log instances were collected in the same multi-feature matrix that was used as input for the automatic classification. For that purpose, I applied six supervised learners including CN2 Rule Induction, Naïve Bayes, Support Vector Machine, Decision Tree, Random Forest, and Adaptive Boosting. I used a suite of open source Python libraries that I modified and adapted for the specific purposes of my workflow. The following is just a brief and qualitative description of the algorithms that I used. For additional details about all these algorithms and their translation into Python codes, see for instance Raschka and Mirjalili (2017).

The CN2 Rule Induction algorithm is a classification technique designed for the efficient induction of simple, comprehensible rules of form "if condition, then predict class". It works properly even in presence of significant noise.

The Naïve Bayes classifier is based on a Bayesian approach. A probabilistic classifier estimates conditional probabilities of the dependent variable from training data and uses them for classification of new data instances. An important benefit of this algorithm is that it is fast for discrete features; however, it is less efficient for continuous features.

Support vector machine (SVM) is a learning technique that splits the attribute space with a hyper-plane, trying to maximize the margin between the instances of different classes or class values.

The Decision Tree algorithm splits the data into nodes by class purity. In other words, this technique separates the data into two or more homogeneous sets (or sub-populations) based on the most significant features in input variables. It is a precursor to Random Forest.

Random Forest is an "ensemble learning" method that uses a set of Decision Trees. Each Tree is developed from a sample extracted from the training data. When developing individual Trees, an arbitrary subset of attributes is drawn (hence the term "Random"). The best attribute for the split is selected from that arbitrary subset. The final model is based on the "majority vote" from individually developed Trees in the Forest.

Like Random Forest, Adaptive Boosting is made up of multiple classifiers and whose output is the combined result of output of those algorithms. Its objective is to create a strong classifier as linear combination of "weak" classifiers.

4. Workflow

The following are the main steps of the classification workflow.

1) Feature engineering represented the first step through which I evaluated the relevance and the sensitivity of each individual attribute (type of well log). Table 2 shows a ranking of the different data logs used for the lithofacies classification of Well A.

Table 2 - Features' ranking using different types of index (for Well A).

Log type	Inf. Gain	Gain Ratio	Gini	ANOVA	Chi2	RelieF	FCBF
SP	1,368	0,684	0,322	336,024	177,233	0,233	0,597
Resistivity	1,185	0,592	0,275	132,591	161,933	0,138	0,517
PEF	1,014	0,507	0,224	192,882	153,2	0,183	0,442
Density	0,971	0,486	0,198	106,271	134,6	0,15	0,424
Sonic	0,962	0,481	0,211	148,226	135,733	0,169	0,42
NEU	0,857	0,429	0,213	95,277	108,3	0,153	0,374
GR	0,814	0,407	0,198	188,759	96,8	0,186	0,355

The following is a synthetic explanation of the indexes used in Table 2.

Information Gain tells us how important a given attribute of the feature-vectors is. For instance, it is fundamental for deciding the ranking of attributes in the nodes of a Decision Tree. It is based on the decrease in "information entropy" after a data set is split on an attribute.

Gain Ratio is a ratio of the Information Gain and the attribute's intrinsic information. It reduces a bias towards multi-valued attributes by taking the number and size of branches into account when choosing an attribute.

Gini index is a measure of statistical dispersion. It is the inequality among values of a frequency distribution.

ANOVA index is the difference between average values of the feature in different classes.

Chi2 represents the dependence between the feature and the class as measure by the chi-square statistic.

RelieF is the ability of an attribute to distinguish between classes on similar data instances.

FCBF (Fast Correlation Based Filter) is the "entropy-based measure", which also identifies redundancy due to pairwise correlations between features.

2) The second step of the workflow consisted in setting the labelled data set for the training phase. I used the available CPI information and all the available geological information in the studied area for training the learner algorithms for classifying the data of Well A. The training data set consists of about 8% of the total well logs, for 1440 instances in total. It is properly balanced in terms of class distribution.
3) The training data set was useful also for comparing the "theoretical" classification performances of the various learners through cross-validation tests, as explained in the next section.
4) After having trained all the six classification algorithms, I applied them to the unlabelled data of Well A, in order to perform full automatic classification.

5) Finally, I performed a further test aimed at classifying the logs of another well (here indicated as Well B). It is located in the same exploration region, relatively far from the previous well. Well B crossed similar geological formations but with different depth distribution with respect to Well A. In the next sections, I am going to discuss the entire workflow in detail.

5. Training and selecting the learning algorithms

In the practice of ML, we can use many different algorithms such as predictors, classifiers, and clustering methods. They will work more or less effectively depending on many variables, such as the type and the quality of the data, the size of the training data set, the type of classification/prediction/clustering problems and so forth. A good approach for selecting the learning algorithm(s) is to test the generalisation power of different methods and, finally, to select the ones showing the best performance. One criterion for selecting the algorithm(s) is going through "cross-validation tests". For applying that method, we need a labelled data set. Cross-validation tests work on these subsets of data, by further partitioning the labelled data into complementary subsets. First, I perform the analysis of the various learners on one subset (called the "training subset"), and then I validate their generalization power on the other subset (called the "validation subset or testing subset").

I used several approaches including the "K-fold", "Random sampling" and "Leave one out" methods. In the first case, the original sample is randomly partitioned into K equal sized subsamples. Of the K subsamples, a single subsample is retained as the validation data for testing the model. The remaining K-1 subsamples are used as training data. I tested various numbers of folds, ranging from 2 to 10, and comparing the results. Table 3 is an example of evaluation results for K=5. The Random sampling method randomly splits the dataset into training and validation data. For each such split, the model is fit to the training data. Finally, the predictive accuracy is estimated using the validation sub data set. Leave-p-out cross-validation uses p observations as the validation set and the remaining observations as the training set. Leave-one-out cross-validation (LOOCV) is a particular case of leave-p-out cross-validation with p=1.

In Table 3, AUC represents the degree or the measure of "separability". It tells how much a certain model is capable of distinguishing between classes. Higher the AUC, better the model is at predicting classes. For instance, in medical applications, higher the AUC, better the model is at distinguishing between patients with disease and no disease. Classification accuracy (CA) is the proportion of correctly classified examples. F1 is a weighted harmonic mean of precision and recall. Precision is the proportion of true positives among instances classified as positive. Recall is the proportion of true positives among all positive instances in the data.

Confusion matrix is an additional technique to verify the performance of each classification algorithm used in the cross-validation tests. Each row of the confusion matrix represents the instances in a predicted class while each column represents the instances in an actual class. Thus, we can estimate the effectiveness of each algorithm in generalizing the classification results (obtained on the training subset) by verifying the percentage cases properly classified (on the validation subset).

Fig. 4 shows two examples of confusion matrix obtained through the Random Forest and Adaptive Boosting classifiers applied to Well A data. We can evaluate the "theoretical prediction

Table 3 - Evaluation results and comparison of the performance of the various learners.

	AUC	CA	F1	PRECISION	RECALL
METHOD					
Tree	0.971	0.943	0.943	0.944	0.943
Random Forest	1.000	0.990	0.990	0.990	0.990
CN2 Inducer	0.955	0.867	0.868	0.869	0.867
AdaBoost	0.979	0.965	0.965	0.965	0.965
SVM	0.988	0.981	0.980	0.981	0.981
Naive Bayes	0.998	0.956	0.956	0.958	0.956

capability" of the various learners just comparing their respective confusion matrix. The above quoted expression "theoretical prediction capability" means the effectiveness of the algorithm to generalise the classification rules that it learnt from the training phase.

This prediction effectiveness is quantified on the principal diagonal of each matrix, where the percentage of predicted vs. actual values is indicated for each class. Instead, looking at the other values above and below the principal diagonal, we have the percentage of wrong classifications.

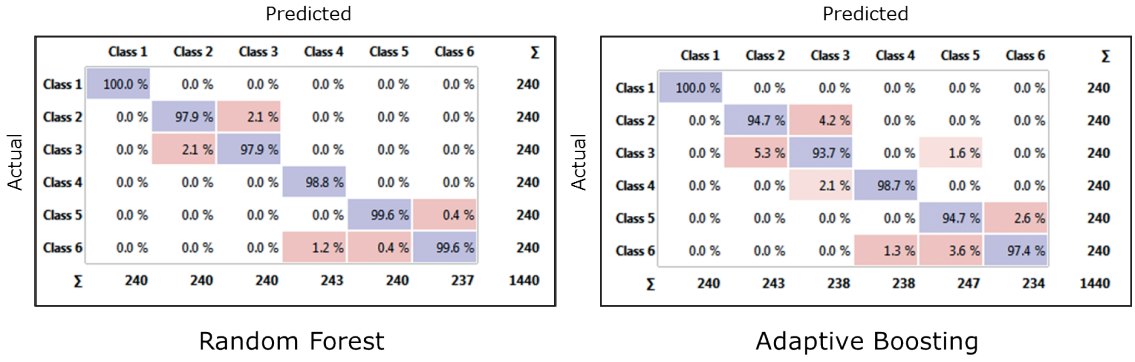

Fig. 4 - Confusion matrix for Random Forest (left) and Adaptive Boosting (right) classifier.

6. Classification of data of Well A

Although different performances emerged from the cross-validation tests (see for instance Table 3), I used all the six algorithms for automatic classification of the lithofacies of Well A. The reason is that, in the tutorial examples discussed in this paper, the data set is relatively small. Consequently, several ML algorithms can run simultaneously on a standard PC without requiring excessive computation time. Of course, in case of "Big Data", the cross-validation tests are useful for selecting the optimal algorithm(s), and for using only one or two methods for the final classification or prediction task. This can be the case, for instance, if we desire to classify seismic facies using an industrial 3D data set.

As examples, Fig. 5 shows the lithofacies classifications obtained with all the methods, using a different colour for each class, projected on the resistivity logs of Well A. Similar classification results have been plotted on the other logs, but they are not shown.

The classification results obtained with the various methods and shown in Fig. 5 are generally comparable, especially in the left half of the log (corresponding with the upper part of the well).

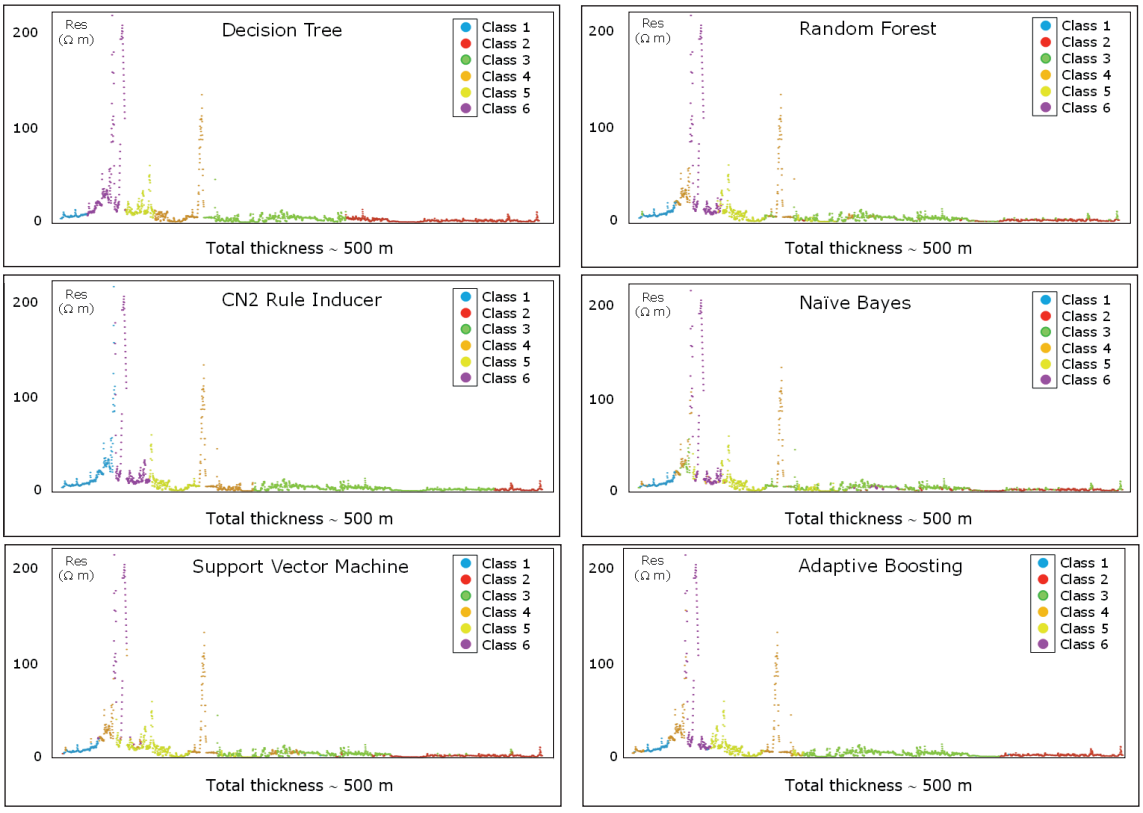

Fig. 5 - Lithofacies classification with all methods projected on resistivity logs (Well A).

This is encouraging because it means that the different classification algorithms tend to produce consistent results, although with some differences.

Going from top to bottom (from left to right in each individual panel of the figure) we can observe that there is a prevalent-shale formation (Class 1) sealing stacked sandy hydrocarbon reservoir with different saturation (Classes 4, 5, and 6). Then, the sedimentary sequence continues with a sequence of interbedded sandstones, shale and siltstones (Classes 2 and 3).

Significant differences appear for the various classification methods in the classification of Classes 2 and 3. This is understandable from a geological point of view, because both classes show similar sedimentary properties. In fact, they are largely overlapped in the feature space (see for instance Fig. 3a).

7. Tuning the hyper-parameters

An important part of the classification step consisted in tuning the hyper-parameters to predict the "unseen data". Indeed, in the practice of ML, there are two types of parameters: those that are learned from the training data and the specific parameters of a learning algorithm. These are commonly optimized separately. The latter are the tuning parameters, also called hyper-parameters, of a model. For instance, these can be the regularization parameter for an algorithm of Logistic Regression or the depth parameter of a Decision Tree. There are several approaches for tuning

the hyper-parameters. One of these is via "Grid search". This is a brute-force exhaustive search method where we specify a list of values for different hyper-parameters; finally, we evaluate the model performance for each combination of those with the final aim to obtain the optimal set. An alternative approach to sample different parameter combinations is Randomized Search. Using that approach, we can draw random parameter combinations from sampling distributions and then we compare the different performances. For instance, I performed many tests for setting the number of trees in the Random Forest method, ranging from 5 to 50. I compared for each test the evaluation results using the same index list showed in Table 3. Finally, I set 20 as the optimal value for that hyper-parameter.

I applied the same approach for tuning the hyper-parameters for the other classifiers. First, I applied a trial-and-error approach, just looking at the various precision indexes in correspondence of each trial. However, that procedure can be optimized and automatised. For that purpose, I used an algorithm from a Python library (GridSearchCV, from Scikit-learn). Describing the details of this algorithm is out of the scope of this paper. However, this approach and the correspondent code are discussed in detail by Raschka and Mirjalili (2017, p. 186). Just to provide an example, using this library, I tuned the key hyper-parameters for the SVM method, such as the type of Kernel, the regression cost, the numerical tolerance, the iteration limit and so forth.

8. A challenging classification test

I tried to classify the lithofacies drilled by another well (Well B) located in the same region, using the same learners trained on the Well A. The Well B drilled analogous geological formations as the well A, but with differences in the sedimentary sequence. In fact, it is located beyond a big fault system that separates the reservoir zone in various compartmented blocks. Furthermore, this well shows complex hydrocarbon distribution in a stacked reservoir formed by several thin layers with variable saturation. Consequently, this further application can represent a sort of "blind test" for verifying how efficiently the learners trained on one log data set can be generalised to multiple wells in the same geological context, even in presence of complex structural elements.

In case of wells drilled within a short distance range and in comparable sedimentary sequences, it would be reasonable to use the learners trained on the data of one well for classifying the data of a near well. In other words, in case of near wells, we can initially assume that the learners that worked properly on one well-data set will work properly on a similar data set. Instead, if there are significant lateral variations from one well to another, especially if there are faults between the two drilling locations, this type of generalization can generate classification artifacts and mistakes. This is the case of this new test on Well B.

In order to use the previous classification results without introducing artifacts in the new classification, a possible approach is to combine the training data set of both wells. Of course, the benefits of this approach increase with the number of wells and with the size of the labelled data set. The intuitive idea is that combining labelled data of a large number of wells allows obtaining a robust training data set for classifying unseen data of other wells in the same geological context.

I applied that strategy for classify the data of Well B. I mixed the training data set of Well A with a small percentage (5%) of labelled data of Well B. The classification results are encouraging. In fact, they are consistent with the expected sedimentary sequence crossed by Well B (based on CPI

and on other geological/geophysical information). As an example, Fig. 6 shows the classification results obtained with the Adaptive Boosting learner and plotted on the resistivity log. Instead, Fig. 7 shows the distribution of the sonic log feature (normalized). This last figure can be compared with Fig. 3a. It shows that the Well A and the Well B have qualitatively comparable class distribution concerning the sonic logs, but they also have significant differences. The same happens for the other logs (here not shown). Both analogies and differences justify the use of a "mixed training data set" for classifying the data of Well B.

Furthermore, we can take additional benefits from the previous classification work done on Well A. In fact, we can use the same hyper-parameters calibrated in the first test for optimizing the learners in the second test. The idea is that using the work already done on Well A for tuning the classifiers hyper-parameters can make the classification workflow more efficient on other wells.

Table 4 includes the main evaluation indexes as quantitative measures of the performance of the different learners using the "mixed training data set" (including data of both Wells A and B). That table has been obtained after performing a K-fold cross-validation test, for K=5. It can be compared with the equivalent Table 3 obtained for Well A. All the values of the various indexes for Well B indicate general good performance of the classifiers, excluding the SVM and CN2 Rule Inducer methods that show slightly lower values for CA, F1, Precision and Recall (AUC Index is still high).

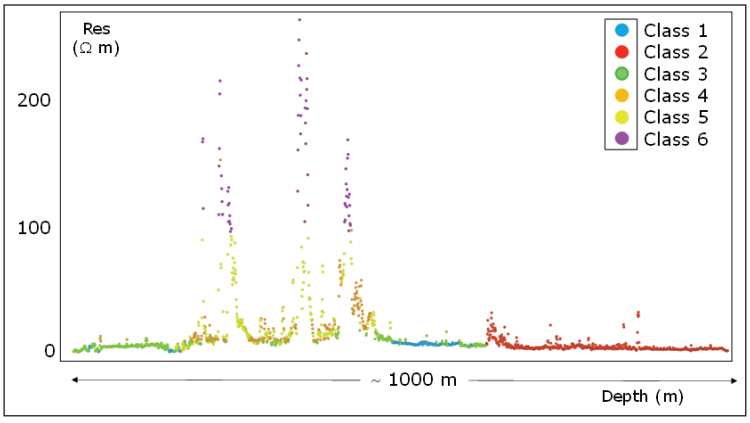

Fig. 6 - Adaptive Boosting classification results for Well B.

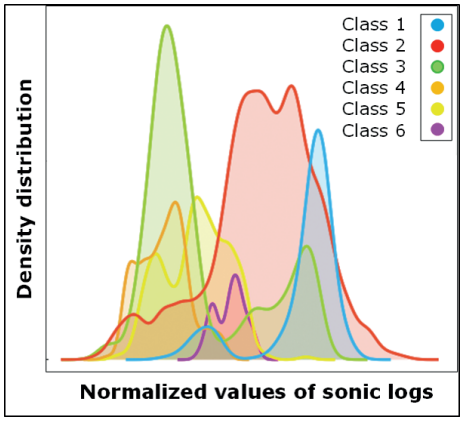

Fig. 7 - Probability density distribution of the normalized sonic logs of Well B.

Table 4 - Evaluation results of the performance of the various learners for Well B.

	AUC	CA	F1	PRECISION	RECALL
METHOD					
Tree	0.928	0.848	0.947	0.848	0.848
Random Forest	0.992	0.917	0.918	0.919	0.917
CN2 Inducer	0.909	0.767	0.768	0.769	0.767
AdaBoost	0.914	0.859	0.859	0.859	0.859
SVM	0.944	0.744	0.744	0.744	0.744
Naive Bayes	0.928	0.959	0.959	0.959	0.959

9. Practical implications and further possible applications

The approach described in this paper can be extremely useful when we need to classify the lithofacies of many wells located in the same geological context. In fact, in this way, the work of log analysis and formation evaluation can be accelerated significantly.

This efficient classification workflow can support well-site geological operations. Well-site geologists perform key operations, like identification of critical strata combining core samples, rock-cutting data, well logs, VSP, surface geophysical data and any other data useful for making operative decisions. Often, these decisions must be taken quickly and even in real time. For instance, operation geologists must decide when specific tests should be carried out and, ultimately, when to stop drilling. In order to support decisions through a multidisciplinary approach, the feature matrix (Fig. 8) used as input for the ML workflow can be populated with many different types of instances complementary to well logs.

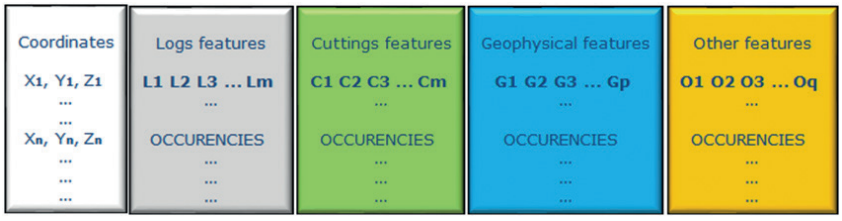

Fig. 8 - Conceptual scheme of the input matrix for the multidisciplinary ML workflow.

For instance, we can combine, in the same matrix, information from composite logs and well cuttings, chemical and mineralogical analyses. Furthermore, after upscaling the same matrix, we can include in it also geophysical information coming from VSP, electromagnetic cross-hole, and so forth. This type of "hybrid" matrix will feed up the ML workflow for many possible purposes. For instance, if we have the possibility to calibrate the matrix instances with the data of other wells in the same area, we can use the ML workflow for predictive purposes during the ongoing drilling operations. Prediction of overpressures or of other hazards is just an example among many possible applications.

10. Conclusions

ML can support the interpretation work of log analysis in the phase of lithofacies classification/interpretation. I compared the performance of six different supervised classifiers. In the tests

here discussed, all the algorithms produced consistent results. However, ensemble algorithms like Random Forest and Adaptive Boosting seem to provide slightly more reliable classifications/predictions than Naïve Bayes, Decision Tree, CN2 Rule Induction. SVM method demonstrated good performance too. Cross-validation tests and the geological meaning of the classification results seem to support this conclusion.

I remark that using the entire set of six algorithms simultaneously for classifying two or more wells does not require any special computation resource. In fact, automatic classification using different types of algorithms is extremely fast for data sets of the order of 50,000-100,000 instances. In the test described in this paper, I measured a computation time of about 5 seconds for running the entire process of ML. It includes training, cross-validation tests, confusion matrix calculation, lithofacies classification, results' plotting, reporting, and saving in the database, using simultaneously six different algorithms. For these tests, I used a standard PC (System characteristics: Dual core Intel processor, 2.5 GHz, RAM 12.0 GB, Windows 10, 64 bit). The most demanding part of the workflow, and the most delicate, is training the algorithms and setting properly all the hyper-parameters of each algorithm. This part of the job requires time, accuracy, knowledge of the data, knowledge of the algorithms, experience, and geological background.

In summary, the approach described in this paper makes the process of log interpretation and lithofacies classification much more efficient than performing the standard manual interpretation for each individual well. Many applications are possible in the field of operation/well-site geology, including drilling optimization and overpressure prediction. However, human supervision and interaction are fundamental, not only in the training phase and for setting the parameters, but also for checking the reliability of the results. In other words, the ML approach should be used as an automatic tool for supporting and enhancing human skills, rather than replacing them. In this sense, we can consider this approach as a cooperative Human-ML methodology.

REFERENCES

Aminzadeh F. and de Groot P.; 2006: *Neural networks and other soft computing techniques with applications in the oil industry*. EAGE Publications, Houten, the Netherlands, vol. 129, 161 pp.

Barnes A.E. and Laughlin K.J.; 2002: *Investigation of methods for unsupervised classification of seismic data*. In: Expanded Abstracts, SEG Technical Program, Salt Lake City, UT, USA, pp. 2221-2224, doi:10.1190/1.1817152.

Bestagini P., Lipari V. and Tubaro S.; 2017: *A machine learning approach to facies classification using well logs*. In: Expanded Abstracts, SEG Technical Program, Houston, TX, USA, pp. 2137-2142, doi:10.1190/segam2017-17729805.1.

Bishop C.M.; 2006: *Pattern recognition and machine learning*. Springer, New York, NY, USA, 758 pp.

Bohling G.C. and Dubois M.; 2003: *An integrated application of neural network and Markov chain techniques to prediction of lithofacies from well logs*. Kansas Geological Survey Open File Report, Lawrence, KS, USA, Technical Report 50, 6 pp.

Hall B.; 2016: *Facies classification using machine learning*. The Leading Edge, **35**, 906-909.

Lary D.J., Alavi A.H., Gandom A.H. and Walker A.L.; 2016: *Machine learning in geosciences and remote sensing*. Geosci. Front., **7**, 3-10.

Raschka S. and Mirjalili V.; 2017: *Python machine learning: machine learning and deep learning with python, scikit-learn, and tensorflow, 2nd edition*. Packt Publishing Ltd., Birmingham, UK, 622 pp.

Russell S. and Norvig P.; 2009: *Artificial intelligence: a modern approach, 3rd edition*. Prentice Hall, Upper Saddle River, NJ, USA, 1152 pp.

Samuel A.L.; 1959: *Some studies in machine learning using the game of checkers*. IBM J. Res. Dev., **3**, 210-229.

Zhang L. and Zhan C.; 2017: *Machine Learning in rock facies classification: an application of XGBoost*. In: Proc. International Geophysical Conference, Qingdao, China, pp. 1371-1374.

Zhao T., Jayaram V., Roy A. and Marfurt K.J.; 2015: *A comparison of classification techniques for seismic facies recognition*. Interpretation, **3**, SAE29-SAE58.

Corresponding author: Paolo Dell'Aversana
Eni S.p.A. Upstream and Technical Services
Via Emilia 1, 20097 San Donato Milanese (MI), Italy
Phone: +39 02 52063217; e-mail: Paolo.Dell'Aversana@eni.com