The Pitfalls and Insights of Log Facies Classification for a Machine Learning Contest

Marcelo Guarido, David J. Emery, Marie Macquet, Daniel O. Trad, and Kristopher Innanen

ABSTRACT

FORCE: Machine Predicted Lithology was a classification contest using well logs from the Norwegian coast of the North Sea. While lithology is the general physical characteristics of rocks our Machine Learning approaches concentrated on the petrology or composition of rocks sample by sample. We used different solutions for the provided data set, and created workflows that clean and complete the data. An additional problem was that the training data was not balanced with 1 class making up 62% of the training data and 7 of the classes less than 4%. We built two different models, one for balanced predictions using a gradient boosting algorithm, and another focusing on the common classes using a model that stacks gradient boosting and random forest probability predictions.

The primary Machine Learning pitfall was balance the petrophysical analysis with the lithofacies associated training classes. The FORCE label training classes also contain a mixture of lithofacies within each class and thus a high degree of mineralogy variation or crosstalk in the confusion matrix. A second pitfall was how the Machine Learning contest was scored used a penalty matrix metric that did not compensate for the imbalance of the input data. The first of our approaches had a great balanced accuracy score of 0.561, but with poor score for the contest metric, scoring -1.35. The second model scored -0.58 on the contest metric, with a trade-off on the balanced accuracy score, which reduced to 0.41.

INTRODUCTION

Lithofacies classification is an indirect field to determine the subsurface rocks types from well logs (Wadleigh and Ward, 1984; Crampin, 2008). Geological lithofacies commonly measure over meters with characteristics rock physical, chemical and biological features that distinguishes it from adjacent rocks. Mineral lithology or petrophysical analysis uses logs to provide similar information as the more expensive core (more precise, but rare) and generally provides information every foot or 30 cm. Interbedding of mineralogy is common in lithofacies as an exampling Marlstone have alternating mixture of carbonate rich mudstones, carbonates, silts and clay layers.

One solution for time optimization is the use of *machine learning algorithms*. There are different lines of research which try a wide number of methodologies. Bestagini et al. (2017); Zhang and Zhan (2017); Caté et al. (2017) use ensemble classifiers (such as *random forests*) as an optimization tool. Another commonly used algorithm is the *support-vector machines*, or *SVM*, which optimize the classification boundaries by computing the support vectors (Caté et al., 2017; Alexsandro et al., 2017; Wrona et al., 2018). *Deep-learning* algorithms, such as the *Artificial Neural Networks* (ANN, or just NN), where successfully applied by (Silva et al., 2014). Guarido (2019) used feature engineering and stacked different algorithms to create a more robust classifier.

For this paper we choose to use four methods from four different Machine learning classifiers (Hastie et al., 2001). The first, Logistic Regression has the advantage it creates output that can easiest be understood from the features (logs). Logistic regression unlike linear regression, that builds algebraic relationships from the input logs, works more on dichotomous relationships (yes/no). Correlation between features are not expected to be an issue but detrending factors such as porosity changes with depth and log normalizing are expected to be important. One of the our major concern with Logistic Regression is it does not handle missing data well nor mixed mineralogy within each class. This method was used for evaluation but not used for any of the final submission for the contest.

Naïve Bayes is a probabilistic classifier known to deal with missing values and small training sets. The method work by building a probability of each class from each feature (log) independently and uses a Bayesian summation to produce a final likelihood for each class. The method provides, along with probability of each class, an estimate of log importance. Regrettably Naïve Bayes like Logistic Regression suffer from multicollinearity between the input logs.

Random Forest, the third method we investigated, builds a set of decision trees from a random set of input logs. This methods is good at preventing overfitting and dealing with uneven data sets with missing variables. The output from each decision tree is summed to produce a final class likelihood using the mode or median.

Gradient Boosting is an ensemble method which unlike Random Forest combines the results at each step instead of at the end. This method build a series of decision trees each solving for the residual error of the previous training tree. Unlike our other three methods, Gradient Boosting can handle non-linear interaction between the features and the classes. The method does not work well in the present of noise nor missing data.

For this project, we used the data set from the FORCE: Machine Predicted Lithology contest and choose to use a more mineralogical/petrological approach than log signature lithofacies classification. The data set showed to be challenging, with a large scale of imbalance between the classes and unusual patterns; such as wells which using different units of measurement, large number of missing data, a considerable number of outliers, and multiple mineralogy within each class (sandstone class having a mixture of feldspathic, plagioclase, quartz minerology, and fluvial, deltaic, foreshore, mid-shore, turbidite facies). In the end, we created a workflow that minimizes most of the issues cited above, as well a classification method that stacks different algorithms. However, for the contest score, we realized that the penalty matrix brought some pitfalls to our strategy and a revised approach was used in the end.

THE DATA

As part of the contest, 108 well logs from the western coast of Norway where provide, as shown in Figure 1. There were 98 wells for the training set (blue), where they provide lithofacies classification, and 10 wells for testing (red) without lithofacies class that could be used for a temporary leaderboard. The winner of the contest was determined by running contestants' models over another set of well logs that were not provide nor we had access.

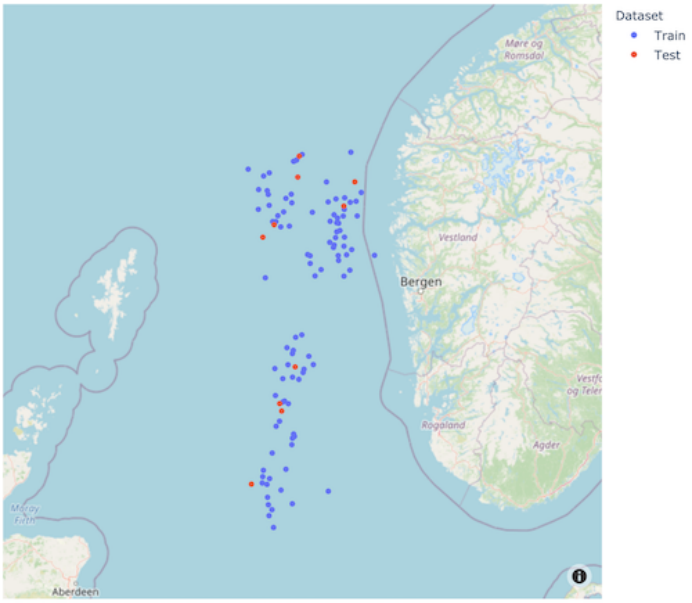

FIG. 1. Wells location for training (blue) and test (red) data sets

A wide range of information for each well was available for the contest, and an example of them is show in Figure 2. The data contains the following metadata columns:

- WELL: Well Name
- DEPTH_MD: Measured Depth
- X_LOC: UTM X coordinate
- Y_LOC: UTM Y coordinate
- Z_LOC: Depth
- GROUP: NPD lithostratigraphy group
- FORMATION: NPD lithostratigraphy formation

Several log curves are available, and they are listed bellow:

- BS: Bit Size
- CALI: Caliper
- RHOB: Bulk Density
- DRHO: Density Correction Log
- GR: Raw gamma data
- SGR: Spectral Gamma Ray
- RSHA: Shallow Resistivity
- RMED: Medium Resistivity
- RDEP: Deep Resistivity
- RXO: Flushed Zone Resistivity

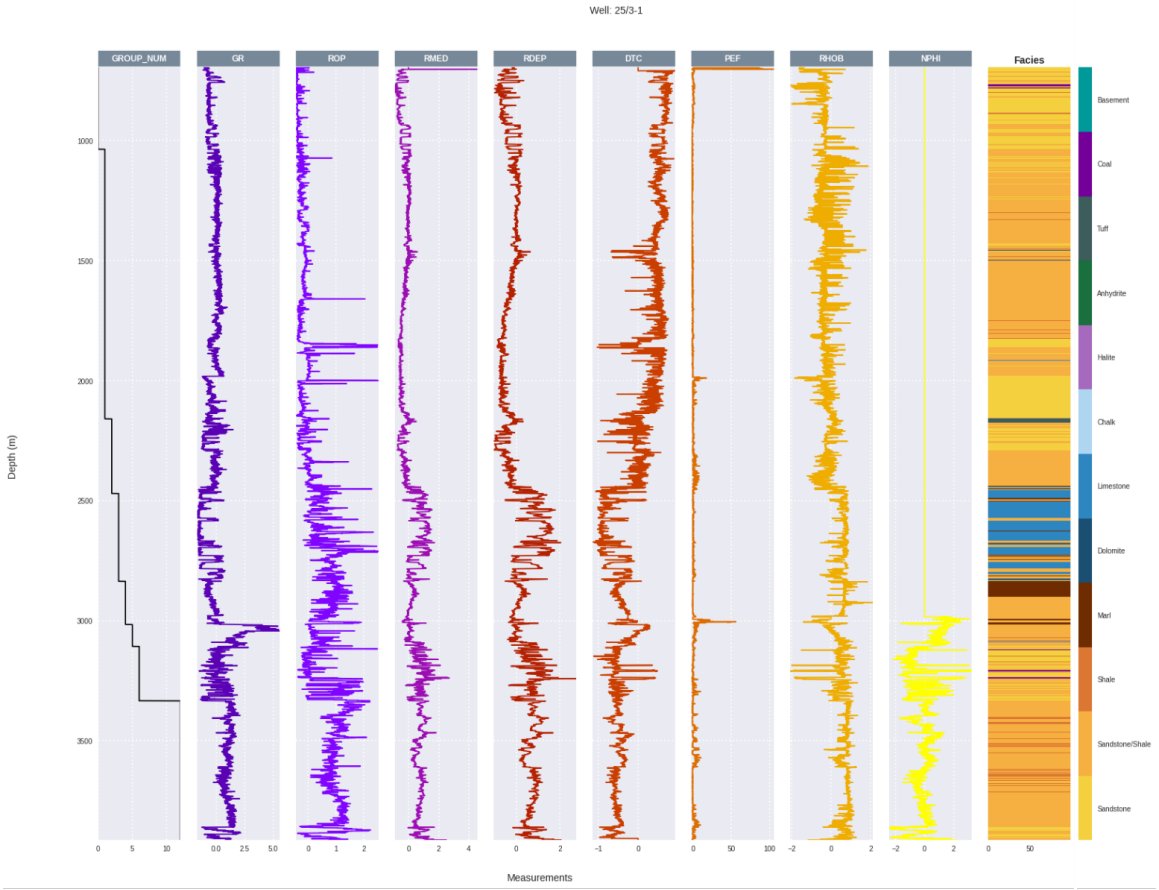

FIG. 2. An example of the logs and facies from the training data.

- RMIC: Micro Resistivity
- NPHI: Neutron Porosity
- PEF: Photoelectric Absorption Factor
- DTS: Sonic (Sheer Slowness)
- DTC: Sonic (Compressional Slowness)
- SP: Self Potential Log
- ROP: Rate of Penetration
- ROPA: Average Rate of Penetration
- MUDWEIGHT: Weight of Drilling Mud

And for the training set, the following interpretation logs were provided:

- FORCE_2020_LITHOFACIES_LITHOLOGY: lithology class label
- FORCE_2020_LITHOFACIES_CONFIDENCE: confidence in lithology interpretation (1: high, 2: medium, 3: low)

Due to the large number of missing data and reduced petrophysical importance, not all of the logs listed above were use for the contest. Figure 3 shows the percentage coverage for each of the logs and metadata. An exhaustive treatment was required to get the maximum insights from the logs.

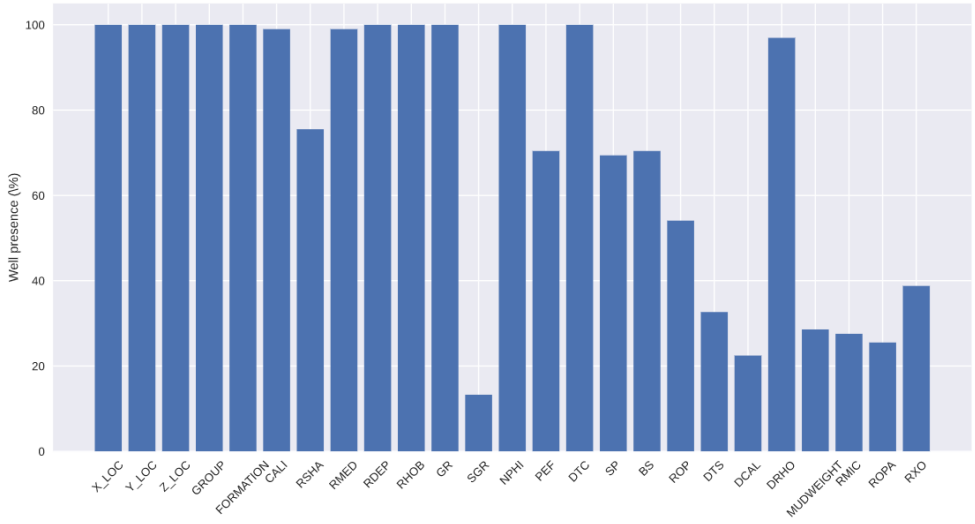

FIG. 3. Percentage of coverage for each of the logs and metadata.

The goal of the contest was, by using all the logs and information listed above, to correctly classify the interpreted lithofacies. There were 12 classes in the total (Table 1), along with a class confidence for rows in the data set. A first observation is the existence of the class *Sandstone/Shale* as well as a classes *Sandstone* and *Shale*: this mixing of mineral lithology can be a tricky "rocky type" to classify. From the beginning we were expecting that our model which focused on mineralogy to get confused during the classification of some classes, which actually happened (we will get into it soon).

Table 1. Class number for each rock type in the contest.

Class	Lithology
30000	Sandstone
65030	Sandstone/Shale
65000	Shale
80000	Marl
74000	Dolomite
70000	Limestone
70032	Chalk
88000	Halite
86000	Anhydrite
99000	Tuff
90000	Coal
93000	Basement

Another observation related to the classes is their distribution within the data set (Figure 4). Most of the samples (around 62%) was shale. The other classes rarely surpass the 10%

mark (only sandstone and sandstone/shale achieve this feature), and most of the classes are statistically under sampled for our approach. At this point, we need to decide what strategy we want to follow: focus on the classification of the most common classes (shale, sandstone, sandstone/shale, and limestone)? Or try a mode balanced classification by weighting the least frequent classes, with the cost of reducing the precision of the classification for the more common classes? We will discuss these options later, showing how this decision strongly affects the contest score.

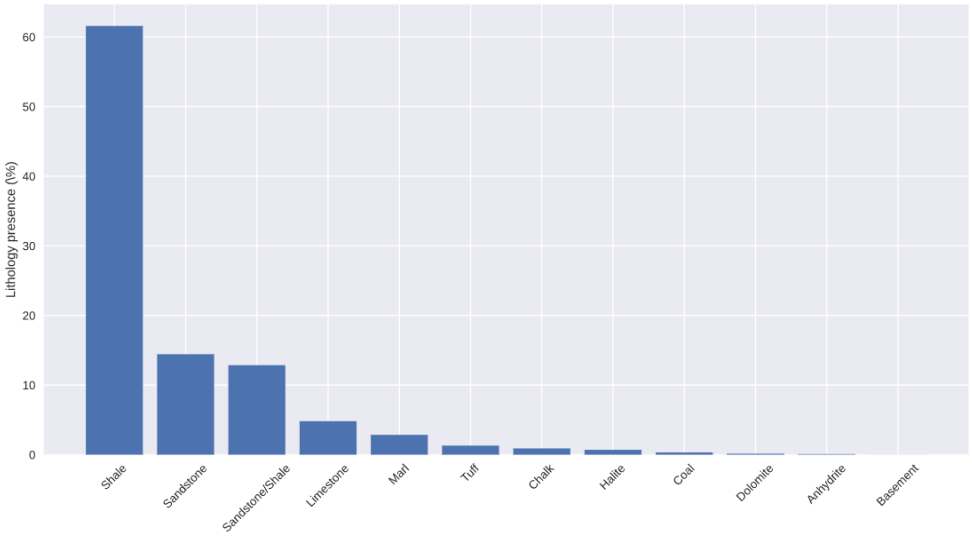

FIG. 4. Class distribution of the data.

WORKFLOW

Contest's data provided was a challenging one, with a large amount of missing data, imbalanced classes, data gaps, variation in units, and bad data points. Figure 5 shows the workflow created to treat and model the data for predictions. The first step was *Data Cleaning*, on which columns of the original data were edited following different criteria: the first was to remove column with missing data larger than 50% and low importance in estimating mineralogy. The second was to select logs that we were most interested to keep for our predictions, we choose to removed the columns *FORMATION* and *RSHA* so the Machine Learning solutions being test would focus more on the petrophysical analysis.

For the *Data Treatment* step, we fixed any possible scaling problems by normalizing each log per well by subtracting its median and divided by its standard deviation. Also, the resistivity logs (*RMED* and *RDEP*) were converted to $\log_{10}$ scale.

The *Data Imputation* step was one of the most complicated ones. Initially, we imputed the data only by replacing the missing values using the median of each column (not separating per well). That is not the most geological solution, but it can work to concentrate the Machine Learning solution on the remaining real data. Later, the imputation strategy was replaced by a chained method (van Buuren and Groothuis-Oudshoorn, 2011) on which the columns with the least amount of missing data are completed with a linear regression algorithm from the complete columns, for the next log be completed with the new set full logs, until all the columns are recovered.

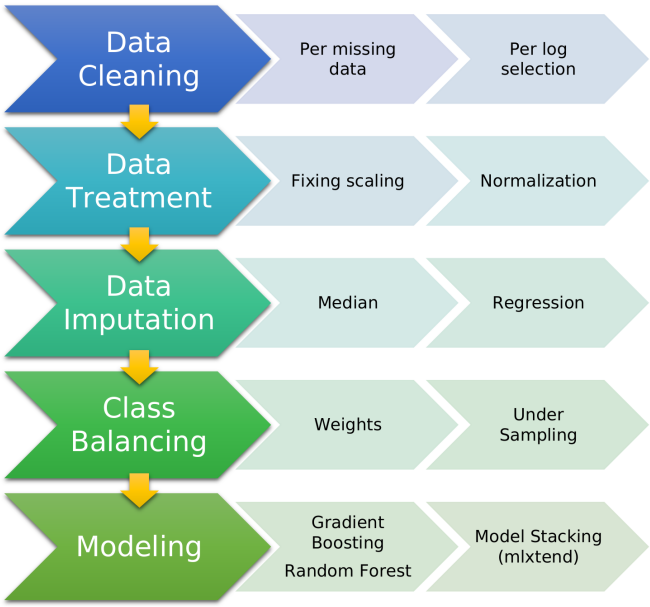

FIG. 5. The project's workflow.

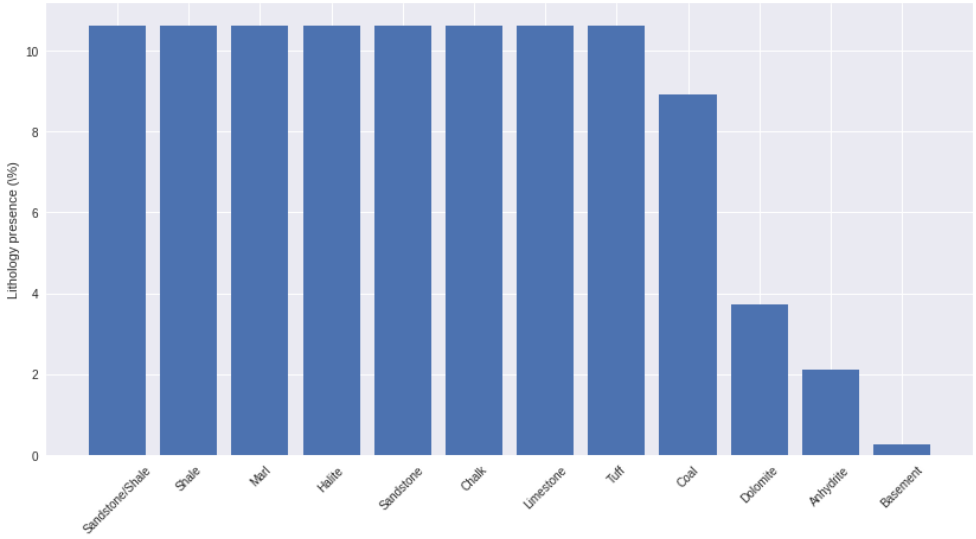

FIG. 6. Classes distribution after under-sampling.

The classes (lithofacies) are highly imbalanced (Figure 4), and there are different ways to work with imbalanced data (He and Garcia, 2009), that can be under-sampling (Yen and Lee, 2009), over-sampling (Han et al., 2005), and weight the data (Liu et al., 2007). Initially, we work to under-sampling the data randomly using the *python* package **imblearn** (Lemaître et al., 2017) so the most frequent classes have the same counting as *tuffstone* samples (Figure 6). However, it reduced considerably the number of rows in the data set (a reduction of around 90%), with the potential lost of information in the process. Weighting the classes was then the chosen methodology and the weight w_y for a class y is calculated using the equation 1:

$$w_y = \frac{N_{samples}}{N_{classes} N_y} \tag{1}$$

where $N_{samples}$ is the total number of the samples in the data, $N_{classes}$ is the number of classes, and N_y is the number of samples for the class y. During the modeling, classification algorithms use those weights to classify low frequent classes with higher precision.

The last step of the workflow was *Modeling*. This is a classification problem, and we tested the most powerful and/or robust methods: *Logistic Regression*, *Naïve Bayes*, *Random Forest*, and *Gradient Boosting* (Hastie et al., 2001). In the end, different combinations of models, weighted or not, were stacked using a vote system with the package **mlxtend** (Raschka, 2018).

DISCUSSION: WHAT WAS THE BEST STRATEGY FOR THE CONTEST?

In this section, we will discuss the difference of the strategies chosen: initially trying to retrieve a more balanced model for all the classes, then changing to give the models more freedom to classify without weights, focusing on the most frequent classes and generally produced the better contest score.

Contest's Metric

The organizers of the contest decided to use a scoring method that penalizes wrong lithfacies classification based on a *penalty matrix* (Figure 7).

label \ prediction	Sandstone	Sandstone/Shale	Shale	Marl	Dolomite	Limestone	Chalk	Halite	Anhydrite	Tuff	Coal	Crystalline Basement
Sandstone	0	2	3.5	3	3.75	3.5	3.5	4	4	2.5	3.875	3.25
Sandstone/Shale	2	0	2.375	2.75	4	3.75	3.75	3.875	4	3	3.75	3
Shale	3.5	2.375	0	2	3.5	3.5	3.75	4	4	2.75	3.25	3
Marl	3	2.75	2	0	2.5	2	2.25	4	4	3.375	3.75	3.25
Dolomite	3.75	4	3.5	2.5	0	2.625	2.875	3.75	3.25	3	4	3.625
Limestone	3.5	3.75	3.5	2	2.625	0	1.375	4	3.75	3.5	4	3.625
Chalk	3.5	3.75	3.75	2.25	2.875	1.375	0	4	3.75	3.125	4	3.75
Halite	4	3.875	4	4	3.75	4	4	0	2.75	3.75	3.75	4
Anhydrite	4	4	4	4	3.25	3.75	3.75	2.75	0	4	4	3.875
Tuff	2.5	3	2.75	3.375	3	3.5	3.125	3.75	4	0	2.5	3.25
Coal	3.875	3.75	3.25	3.75	4	4	4	3.75	4	2.5	0	4
Crystalline Basement	3.25	3	3	3.25	3.625	3.625	3.75	4	3.875	3.25	4	0

FIG. 7. Penalty matrix.

Scoring S is calculated by summing all the weighted misclassification $\mathbf{A}_{\hat{y}_i y_i}$ for a prediction y_i compared to the true label $\hat{y}$, and then divide by the N number of observations in the data set (equation 2):

$$S = -\frac{1}{N}\sum_{i=0}^{N}\mathbf{A}_{\hat{y}_i y_i} \tag{2}$$

This scoring method is not weighted for imbalanced classes. The best the model performs, closer the score is to 0.

Results: Balanced Models

Our first strategy was to work harder on the less frequent classes. For that, we undersampled the data and also computed the class weights for the remaining data. For the analysis, we separate 20 wells from the 98 training wells for validation, leaving 78 to train the models. All the metrics and analysis presented are the results obtained on our validation set.

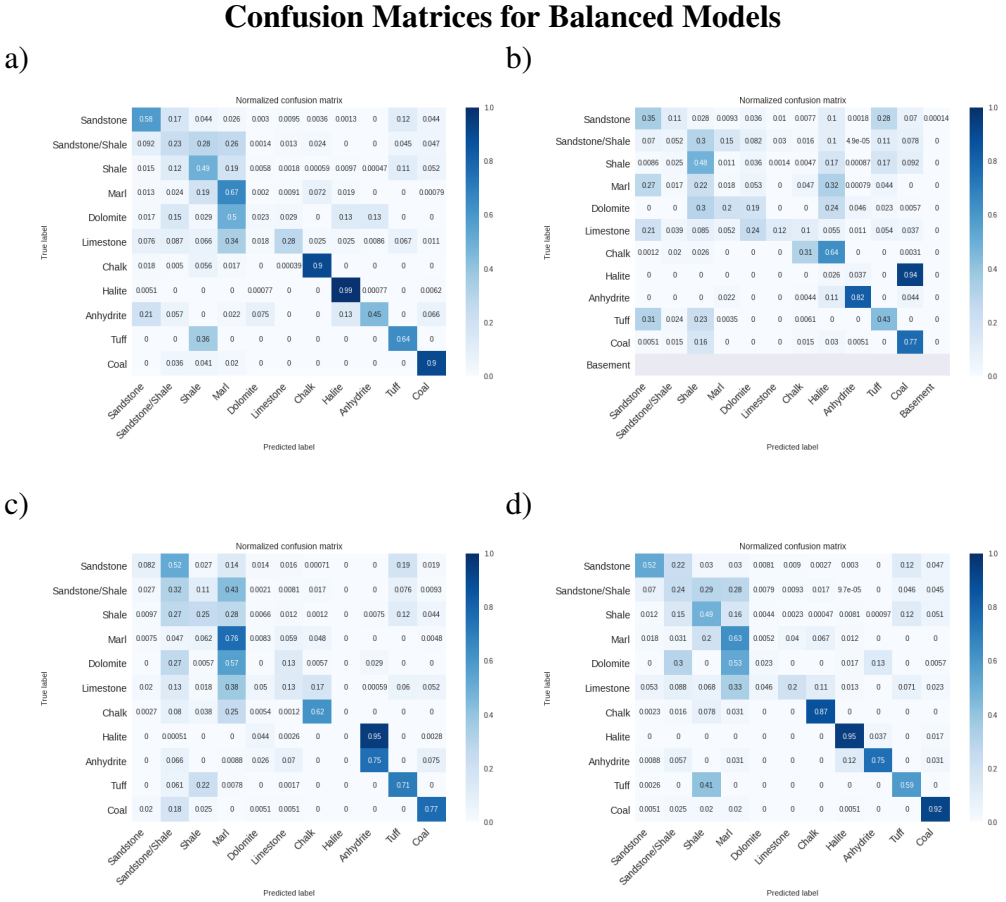

FIG. 8. Confusion matrix for the a) gradient boosting, b) logistic regression, c) naïve bayes, and d) stacked classifiers.

Three different algorithms were tested: *gradient boosting*, *logistic regression*, and *naïve*

bayes. Each one assumes its own strategy for classification, but the gradient boosting is the only non-linear one. All the algorithms were set to run for imbalanced classes (under-sampling and class weights) and, in the end, we stack the models by a "soft" voting system (using each model's probability outputs). Figure 8 shows the normalized confusion matrix for all the models: a) gradient boosting, b) logistic regression, c) naïve bayes, and d) stacked classifiers. The gradient boosting alone was the one with the best *balanced accuracy* (accuracy weight by class frequency), scoring 0.561, while the stacked model had a score of 0.56. Those are good scores for a balanced accuracy, as we have 12 classes (the balanced random guess would be $= 1/12 = 0.08$). And we can also note that the logistic regression and the naïve bayes models are actually did not aid the predictions when stacked. This comes to the assumption of linear relationship between the logs and lithofacies, what may not be the case. It is interesting to see that the best model (gradient boosting) did a great job predicting the less frequent classes, like chalk, halite, anhydrite, tuff, and coal, with also a good mention for the marlstone. However, it came with the cost of poorer classification for the most frequent classes (sandstone, shale, and sandstone/shale). Dolomite's classification was the trickiest one, and none of our models could learn a pattern to identify it.

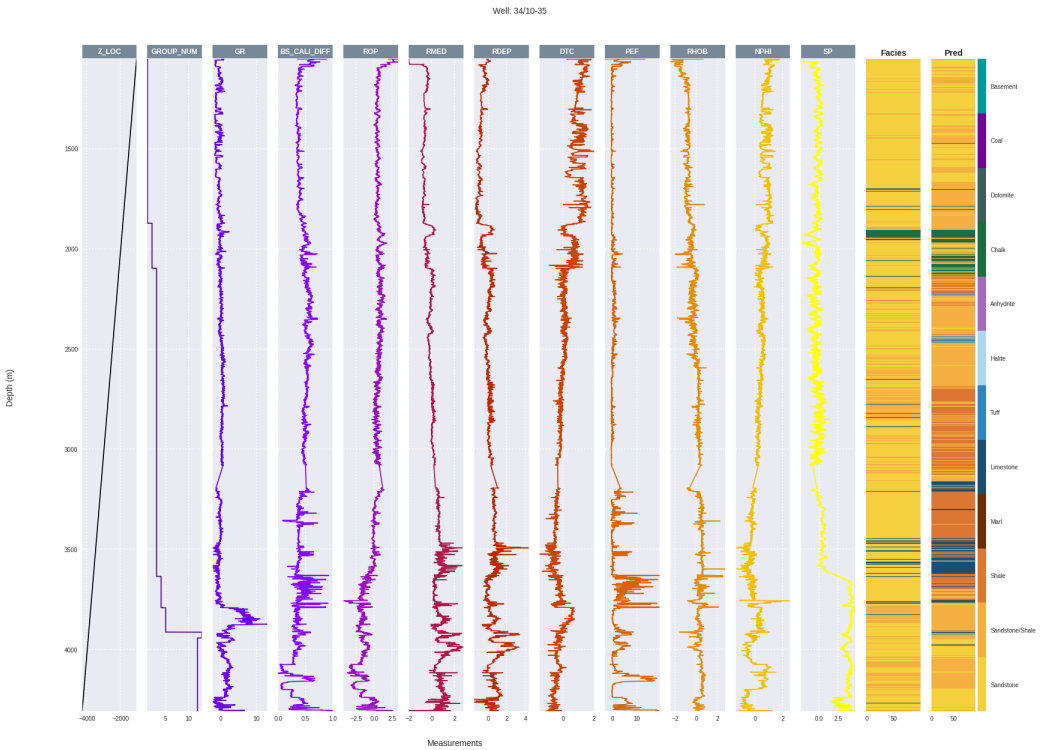

FIG. 9. Predictions with balanced models.

Figure 9 shows the prediction over one of the validation wells. Predicted lithofacies is promising, but with a large number of misclassification between the most frequent classes. But overall the model is catching the changes in lithology.

Even tough the best model (according to the balanced accuracy) seems to be stable and robust, it performed poorly on the contest metric (equation 2), with a score of -1.35. As a comparison, the imbalanced logistic regression (Figure 10a) predicts all the samples as

shale, the dominate lithofacies, resulting in a accuracy of 0.08 or equivalent to a random guess but a contest score of -0.96. This means that the contest's metric does not take in consideration class imbalance, only the absolute number of correct classified labels.

Results: Focusing on Contest's Metric

As the imbalanced models did a poor job on the contest's metric, the strategy was changed to create a combine models that used both balance and imbalanced models.

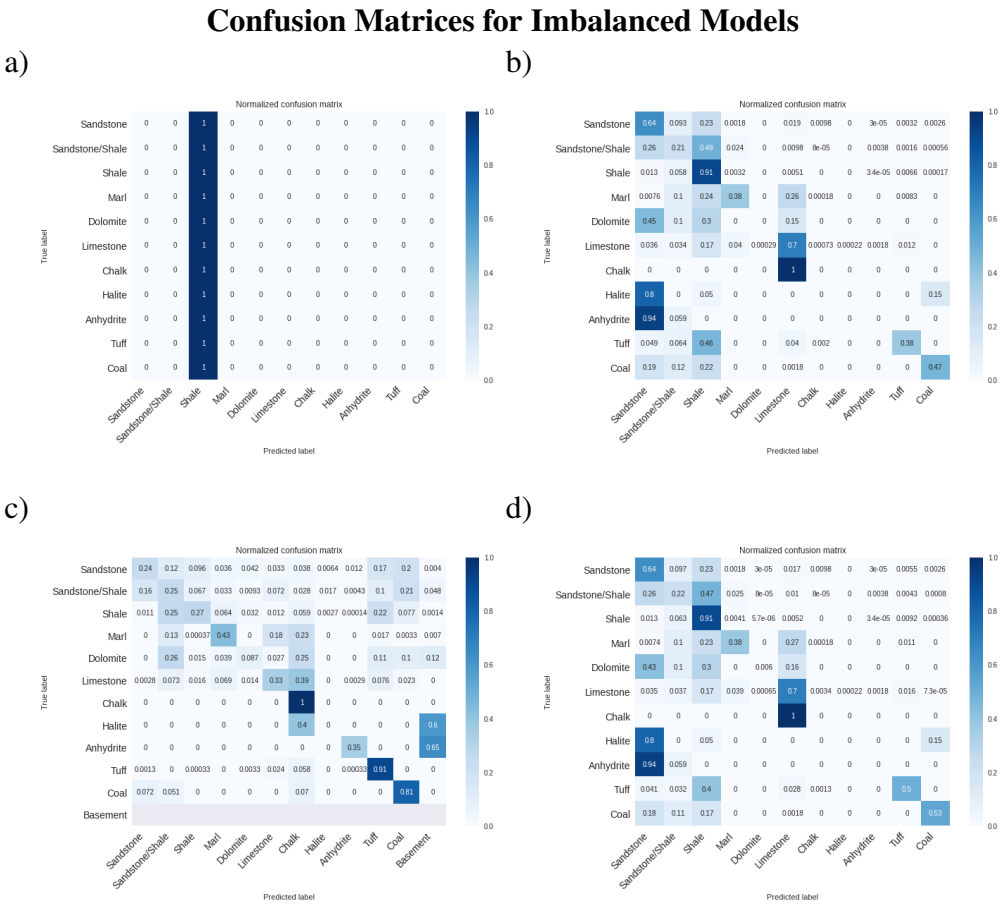

FIG. 10. Confusion matrix for the a) logistic regression (not used for stacking), b) gradient boosting, c) random forest, and d) stacked classifiers.

Figure 10 shows the confusion matrix for the a) logistic regression (not used for stacking), b) gradient boosting, c) random forest, and d) stacked classifiers Note that the logistic regression in this part was just used as a baseline classifier, as it only predicts *shale*. Naïve Bayes was dropped and replaced by a *random forest* algorithm, and it was balanced (by class weights) and was used as a support classifier for the gradient boosting during the model stacking. The gradient boosting model was imbalanced, focusing to classify more correctly the most frequent classes, such as shale, sandstone, and limestone. The class *sandstone/shale* is a common class but difficult to be correctly classified, as it tends to get confused with the *sandstone* and *shale* classes (which is not surprising as this is a combination of the other 2 classes).

Table 2. Models performance.

Model	Balanced Accuracy	Contest Metric
1. Gradient Boosting (balanced)	0.56	-1.35
2. Gradient Boosting	0.42	-0.59
3. Random Trees (balanced)	0.40	-2.00
4. Naïve Bayes	0.40	-1.86
5. Logistic Regression (balanced)	0.32	-2.17
6. Logistic Regression	0.08	-0.96
7. Stacked Models (balanced)	0.56	-1.38
8. Stacked Models	0.41	-0.58

Table 2 contains the *balanced accuracy* and *contest* metrics for all the tested models. It became clear that the contest metric, by using the equation 2 and the penalty matrix of Figure 7, focus on the correct classification of the most common classes. The best contest score was achieved by stacking the imbalanced gradient boosting (3) and the balanced random forest (5) by a voting system (4), scoring -0.58. However the balanced accuracy dropped significantly to 0.41, and from the Figure 10d, the model did a poor job on classifying the least frequent classes, in particular the halite and anhydrite, that were almost always classified as sandstone. Note that the stacked models was mainly controlled by the gradient boosting, as it probably contain higher probabilities to classify all the classes, even if misclassifying.

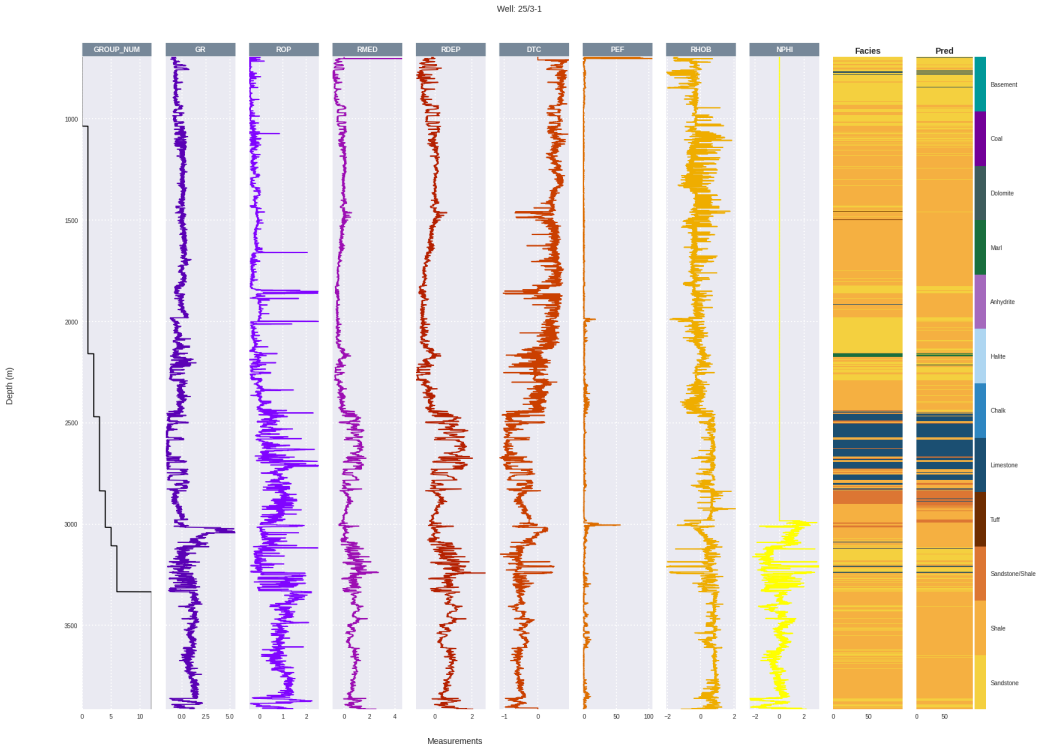

FIG. 11. Predictions with imbalanced models.

Figure 11 shows the new predictions for the mostly imbalanced stacked model. It looks visually better then the predictions of Figure 9, as the dominant colours are related to the most frequent classes.

At this point, to chose a more appropriate classification strategy, we need to understand what are the goals for the contest and/or a possible client. On the case of the contest, there was a strong bias to classifying the most common classes correctly, and the winner is the one who does a better job predicting the classes *shale*, *sandstone*, and *sandstone/shale* (our imbalanced model struggled on the predictions of the last class). Usually, identifying reservoir lithofacies and not shale are more important in classification, but was the most import in determine a score for the contest. Sandstone and limestone have great importance during reservoir characterization (where the imbalanced model worked well), as for salt rocks like halite and anhydrite (identified better with the balanced model).

CONCLUSIONS

We presented a workflow for lithofacies classification from well logs that allows different outputs depending on the focus of the research: one output that focus on an overall balanced classification, great to identify rare occurrences, and another output focused on the most common labels, scoring better in the contest. The workflow starts with the data cleaning and ends up with the modeling and prediction of the lithofacies from the well logs.

During the execution of this project, an import lesson was learned related petrophysical analysis to the goal of the research. Initially, the focus was to create a model that provided a balanced classification (by weighting a class according to its frequency), focusing on the *balanced accuracy* metric. A stable and robust model was generated (balanced gradient boosting), scoring 0.561 on the balanced accuracy metric, but scored poorly on the contest metric (-1.35). At that point it was realized that the contest metric penalizes the absolute values of the misclassification, and not a balanced or normalized one. This means that to optimize the contest score, the most common classes have a higher importance to be correctly classified, while correctly classifying the rarer classes reflects on small improvements in the score (important for untie models that already classified the common classes properly). With this realization, the focus was changed and an imbalanced model (a stack of an imbalanced gradient boosting and a balanced random forest) was generated, which worked poorly on rarer classes (like halite and anhydrite), but did a good job on the common classes (shale, sandstone, and limestone), improving the contest score to -0.58, with the cost of reducing the balanced accuracy to 0.41.

Of course the perfect scenario is to classify all the lithofacies with 100% accuracy, but it is unlikely to happen when we are dealing with imbalanced classes and mixed mineralogy in lithofacies. In the end, the question is: what are the most important classes of a facies classification project? That is a decision that needs to be taken by the "owners" of the data (a client or a contest organizer), as the solution varies and may be specific for a selected block of facies.

ACKNOWLEDGEMENTS

The authors thank the sponsors of CREWES for continued support. This work was funded by CREWES industrial sponsors and NSERC (Natural Science and Engineering Research Council of Canada) through the grant CRDPJ 461179-13 and CRDPJ 543578-19, and the financial support from Canada First Research Excellence Fund. We thank Soane Mota dos Santos for the suggestions, tips and productive discussions.

REFERENCES

Alexsandro, G. C., da P. Carlos, A. C., and Geraldo, G. N., 2017, Facies classification in well logs of the Namorado oilfield using Support Vector Machine algorithm, 15th International Congress of the Brazilian Geophysical Society; EXPOGEF, Rio de Janeiro, Brazil, 31 July-3 August 2017, 1853–1858.

Bestagini, P., Lipari, V., and Tubaro, S., 2017, A machine learning approach to facies classification using well logs, SEG Technical Program Expanded Abstracts 2017, 2137–2142.

Caté, A., Perozzi, L., Gloaguen, E., and Blouin, M., 2017, Machine learning as a tool for geologists: The Leading Edge, **36**, No. 3, 215–219.

Crampin, T., 2008, Well log facies classification for improved regional exploration: Exploration Geophysics, **39**, No. 2, 115–123.

Guarido, M., 2019, Machine learning strategies to perform facies classification: GeoConvention Expanded Abstract 2019.

Han, H., Wang, W.-Y., and Mao, B.-H., 2005, Borderline-smote: A new over-sampling method in imbalanced data sets learning, *in* Huang, D.-S., Zhang, X.-P., and Huang, G.-B., Eds., Advances in Intelligent Computing, Springer Berlin Heidelberg, Berlin, Heidelberg, 878–887.

Hastie, T., Tibshirani, R., and Friedman, J., 2001, The elements of statistical learning - data mining, inference, and prediction: Springer, second edn.

He, H., and Garcia, E. A., 2009, Learning from imbalanced data: IEEE Transactions on Knowledge and Data Engineering, **21**, No. 9, 1263–1284.

Lemaître, G., Nogueira, F., and Aridas, C. K., 2017, Imbalanced-learn: A python toolbox to tackle the curse of imbalanced datasets in machine learning: Journal of Machine Learning Research, **18**, No. 17, 1–5. URL `http://jmlr.org/papers/v18/16-365.html`

Liu, Y., Loh, H. T., Kamal, Y.-T., and Tor, S. B., 2007, Handling of Imbalanced Data in Text Classification: Category-Based Term Weights, Springer London, London, 171–192.

Raschka, S., 2018, Mlxtend: Providing machine learning and data science utilities and extensions to python's scientific computing stack: The Journal of Open Source Software, **3**, No. 24.

Silva, A., Neto, I. L., Carrasquilla, A., Misságia, R., Ceia, M., and Archilha, N., 2014, Neural network computing for lithology prediction of carbonate- siliciclastic rocks using elastic, mineralogical and petrographic properties, 13th International Congress of the Brazilian Geophysical Society & EXPOGEF, Rio de Janeiro, Brazil, 26–29 August 2013, 1055–1058.

van Buuren, S., and Groothuis-Oudshoorn, K., 2011, mice: Multivariate imputation by chained equations in r: Journal of Statistical Software, Articles, **45**, No. 3, 1–67.

Wadleigh, R. F., and Ward, J. A., 1984, Carbonate-anhydrite facies determination in the Paradox Basin by quantitative seismic stratigraphy, SEG Technical Program Expanded Abstracts 1984, 497–500.

Wrona, T., Pan, I., Gawthorpe, R. L., and Fossen, H., 2018, Seismic facies analysis using machine-learning: GEOPHYSICS, **0**, No. ja, 1–34.

Yen, S.-J., and Lee, Y.-S., 2009, Cluster-based under-sampling approaches for imbalanced data distributions: Expert Systems with Applications, **36**, No. 3, Part 1, 5718 – 5727.

Zhang, L., and Zhan, C., 2017, Machine Learning in Rock Facies Classification: An Application of XGBoost, International Geophysical Conference, Qingdao, China, 17-20 April 2017, 1371–1374.