The Pitfalls and Insights of Log Facies Classification using Machine Learning

David J. Emery, Marcelo Guarido, and Daniel O. Trad

ABSTRACT

The use of Machine Learning solutions has become a trend in Geophysics with a number of associations such as SEG, AAPG, FORCE, and the SPE running a number of competitions focused on geosciences. This paper deals with observation on Machine Learning solutions on the FORCE: Machine Predicted Lithology which was a classification contest using well logs from the Norwegian coast of the North Sea. The primary Machine Learning pitfalls were: ill-conditioning of the lithofacies class, uneven sampling of the class, finding the correct features engineering, need to impute missing values and eases of overfitting. The methods investigated for the contest were Logistic Regression, Naive Bayes, Random Forest, and Gradient Boosting. Following the contest, we investigated: improving the feature engineering by detrending, the role impute had on the results, and redefining the lithofacies class. Regrettably, the Machine Learning methods investigated predominately focused on mineralogy resulting in a poor classification of mixed lithofacies such as Marlstone, Coquina, Interbed Sandstone-Shale unit, etc., and confused units with similar mineralogy such as Chalk being classified as Limestone. XGBoost after feature engineering along with including the geological framework (X ,Y & formations), gave workable results and a reasonable score for the contest with the advantage of not needing imputation of missing data.

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 general parameters over meters with characteristics physical, chemical and biological features that distinguish from adjacent rocks. The more precise method of determining lithofacies is through the use of core but this method is relatively expensive and as such relatively rare. Generally, petrophysical analysis uses logs are the primary source for this analysis which can be aided by side-wall cores and chip samples. Well logs sample formation mineralogical/chemical properties over a foot (1/3 meter) and involve a time consuming processes of petrophysical analysis that combined with both offsetting information and knowledge of the geological setting assigns a lithofacies. One solution for time optimization is the use of *machine learning algorithms* to determine the lithofacies and hopefully improve the accuracy by removing the subjective step of petrophysical analysis.

There are different lines of Machine Learning 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 the contest (Guarido et al., 2020) we choose to use four methods from four different Machine learning classifiers original proposed by (Hastie et al., 2001). The first, Logistic Regression has the advantage to create outputs 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). One of our major concerns 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 submissions for the contest.

Naïve Bayes is a probabilistic classifier known to deal with missing values and small training sets. The method works 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 method 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 builds 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 XGBoost implication of this method doesn't require imputing of missing data and has significant speed improvement.

For this project, we supplemented the data set from the FORCE: Machine Predicted Lithology contest and choose to use a more mineralogical/petrological approach by reclassifying the lithofacies in mineral sub-classes. The data set remained to be challenging, with a large degree of unbalance between the classes, under-sampling, and a significant number of outliers.

THE DATA

As part of the contest, 108 well logs from the western coast of Norway were provided, as shown in Figure 1. There were 98 wells for the training set (blue), where they provided lithofacies classification, and 10 wells for testing (red) without lithofacies classes that were used for the temporary leaderboard. Another 10 wells were held back for evaluation of the contest. All 128 were made available following the completion of the contest under a Norwegian open data license including the ExploCrowd lithofacies.

The information provide for each well (Figure 2) contains the following metadata columns:

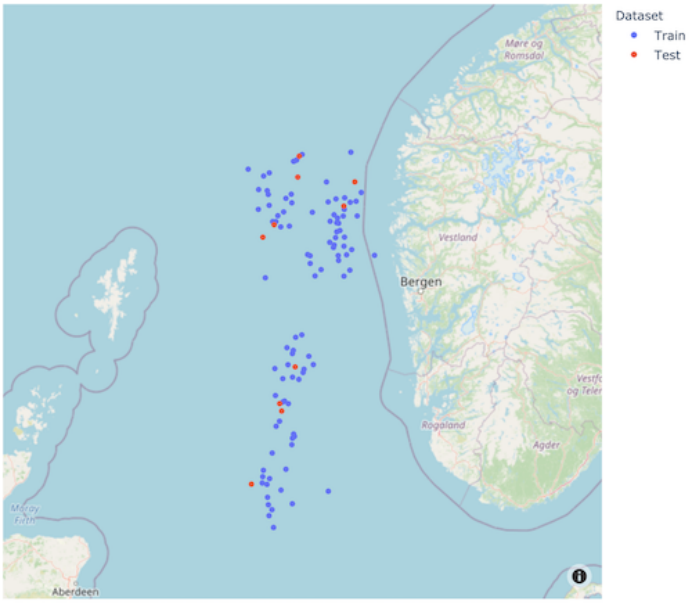

FIG. 1. Wells location for training (blue) and test (red) data sets

- 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

The primary petrophysical curves were:

- GR: Raw gamma data
- RHOB: Bulk Density
- RSHA: Shallow Resistivity
- RMED: Medium Resistivity
- RDEP: Deep Resistivity
- NPHI: Neutron Porosity
- PEF: Photoelectric Absorption Factor
- DTC: Sonic (Compressional Slowness)
- SP: Self Potential Log
- SGR: Spectral Gamma Ray
- DTS: Sonic (Sheer Slowness)

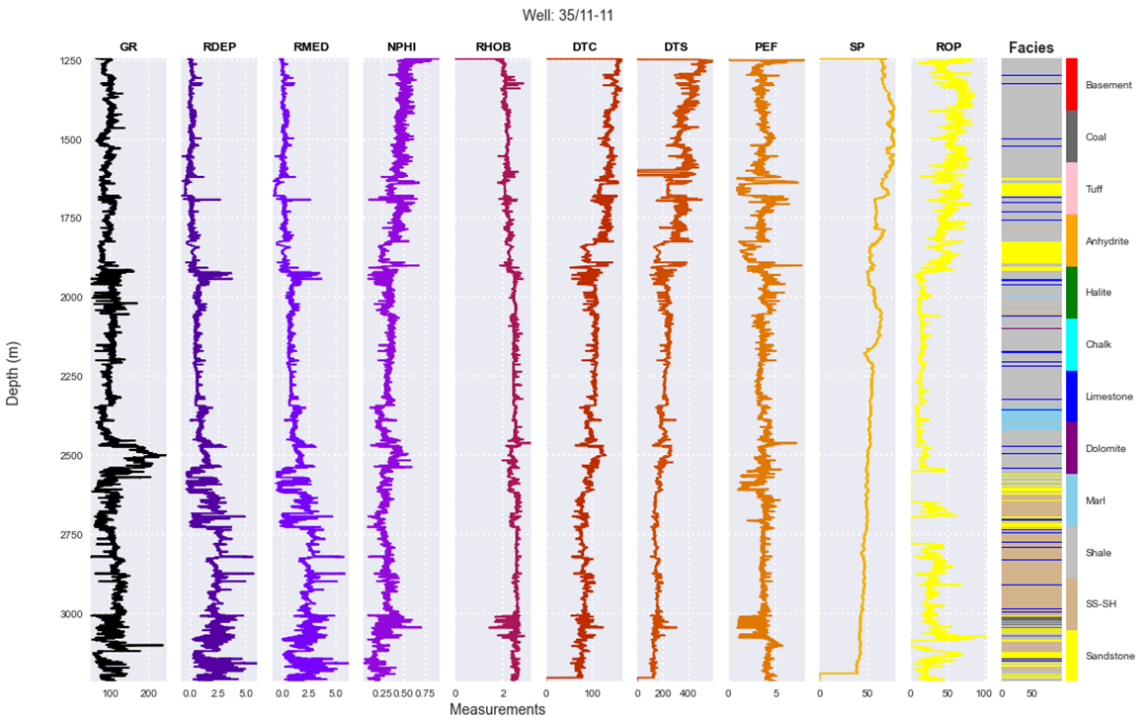

FIG. 2. An example of the logs and facies from the training data.

Additional provided curves were :

- BS: Bit Size
- CALI: Caliper
- DRHO: Density Correction Log
- RXO: Flushed Zone Resistivity
- RMIC: Micro Resistivity
- ROP: Rate of Penetration
- ROPA: Average Rate of Penetration
- MUDWEIGHT: Weight of Drilling Mud

And for the train petrophysical interpretation and confidence:

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

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.

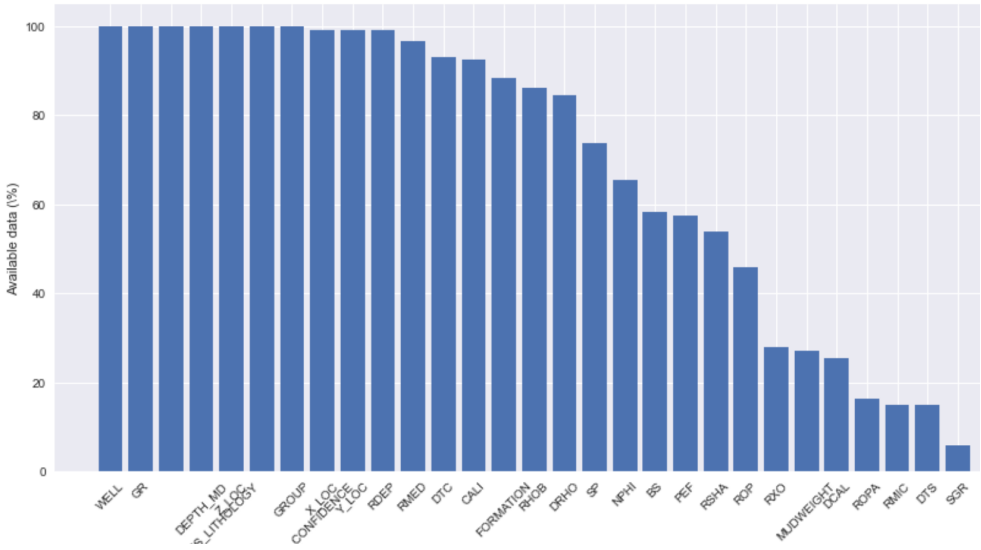

FIG. 3. Percentage of coverage for each of the logs and metadata.

The goal of the contest was, by using all the logs and information listed above, to correctly classify the interpreted 12 provided lithofacies. Most of the samples (around 89%) were siliciclastic with shale (62%), Sandstone (14%) and Sandstone-Shale (13%), carbonates made up 8.6% (Limestone, Marlstone, Chalk & Dolomite), and the other classes are rarely making up only 2.4% (Halite, Anhydrite, Coal, Tuff & Basement). Dealing with the unbalanced nature of the classes was a significant factor of our contest enter (Guarido et al., 2020). Taking advantage of the siliciclastic dominates to remove porosity variation with depth (Emery et al., 2020) was shown to improve the analysis.

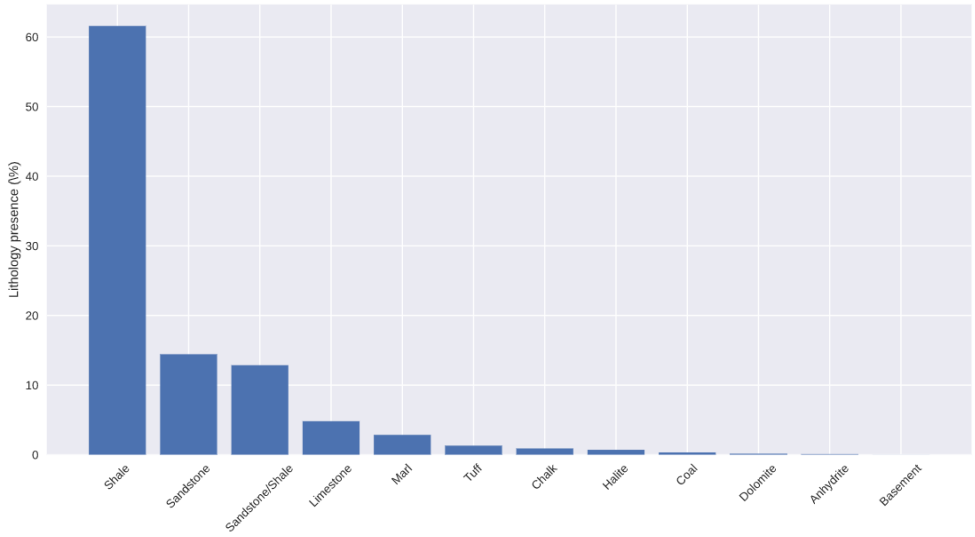

FIG. 4. Class distribution of the data.

FEATURE ENGINEERING

The first step was *Data Cleaning*, on which columns of the original data were edited following different criteria: remove column of low importance in estimating mineralogy,

drop logs with poor coverage (SGR, DTS), edit for variation in units, and remove bad data points.

Dominating our input data was also variation in porosity both lithological but particularly by compaction with burial. As the data was 89% siliciclastic and dominantly shale, with minimal erosion, we choose to estimate a single global trend and correct for the variation in water depth. The dichotomous nature for our selected machine learning methods doesn't necessarily require scaling as part of the *Data Treatment* step, estimating the depth porosity trend does require the resistivity logs *RMED* and *RDEP* to be converted to $\log_{10}$ scale and logs *DTC* and *DTS* into *Vp* and *Vs*.

Our goal was to estimate mineralogy and infer lithology, selecting curves that represent the mineral matrix was paramount but we also provide curves that hopefully would aid in compensating for local porosity and fluid type variation. The resistivity cross-over was provided as a feature (*RMED* minus *RDEP*). The *SP* was corrected individually for each well by subtracting a medium trend and standardizing the variance. To provide a hole conditioning curve a filter cake log was created from the *CALI* minus *BS*. As the *BS* coverage was less than the *CALI* we also estimated a *BS* from the *CALI* prior to creating the filter cake log.

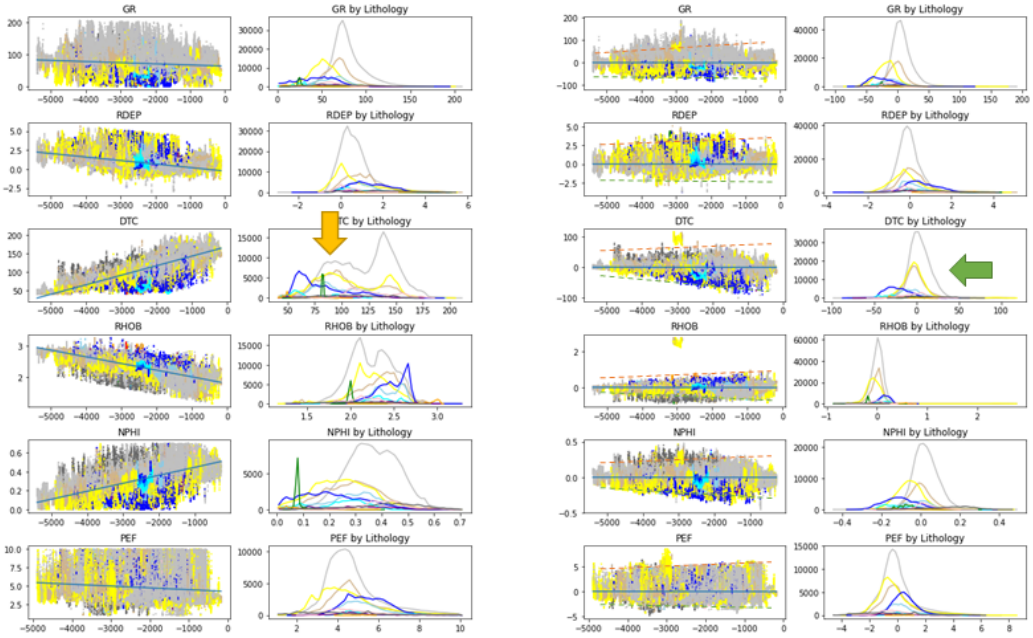

FIG. 5. Input and after feature engineering; display on the left is the coloured by lithoclass with a histogram on the right histogram. After trend removal the Bayesian statistics significantly improved resulting in a better separation of lithology

After feature engineering, the statistical variation was considerably reduced for all the logs (Figure 5) but significant overlap remains between the lithoclasses. Of the remaining curves (DRHO, RXO, RMIC, ROP, ROPA & MUDWEIGHT) only the ROP curve underwent correction. The ROP curve has a significant variation between wells along with substantial outliers, and a correction was therefore done individually for each well using a percentile normalization coupled with outlier removal. The question would be why didn't we do the petrophysical curve correction well-by-well? The reality was a significant num-

ber of wells end after penetrating Halite or Basement causing a substantial J or L shaped curve, which led to significant errors in the porosity trend estimation.

The final curves created by feature engineering were from the GROUP and FORMATION provided by the Norwegian Petroleum Directorate (NPD). The lithostratigraphy included in the training data was a subset of the possible NPD mnemonics and instead of building a dictionary from the NPD website, we simply numbered the GROUP and FORMATION by order encountered in the file and then multiplied the GROUP by 100 and add the two fields together (AGENUM). We did create a dictionary for the contest from the known train set and ignored any GROUP or FORMATION in the evaluation set that wasn't in the train data.

IMPUTATION & BALANCING

Of all the Machine Learning methods we investigated, only XGBOOST does not require imputing the missing curve values. This *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 correct 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.

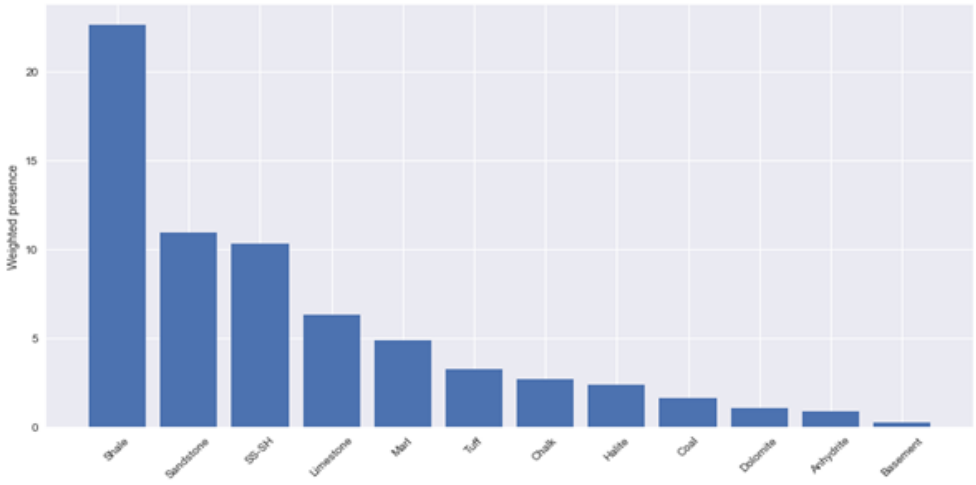

FIG. 6. Classes distribution after partial balancing for a n=0.3.

The classes (lithofacies) are highly unbalanced (Figure 4), and there are different ways to work with unbalanced 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. However, it reduced considerably the number of rows in the data set (a reduction

of around 90%), with the potential loss of information in the process. Weighting the classes (Figure 6) was then chosen as a better methodology and the weight w_y for a class y is calculated using the equation 1:

$$w_y = (\frac{N_{samples}}{N_{classes} N_y})^n \tag{1}$$

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. This weight was then scaled by n from 0 for unbalance to 1 for fully balanced. During the modeling, classification use those weights scaled by 1/2 or 1/3 helped balance the low frequent classes without affecting the overall precision.

INPUT CLASS CONDITIONING

For the contest we didn't deal with modifying any of the input class and tested the power and/or robustness of: *Logistic Regression*, *Naïve Bayes*, *Random Forest*, and *Gradient Boosting* (Guarido et al., 2020). In the end, different combinations of models, weighted or not, were stacked using a vote system with the package **mlxtend** (Raschka, 2018).

After the contest we attempted to correct some of the ill-condition nature of the train classes, first by sub-dividing the Shale fraction into 4 parts (silty-Shale, calcareous-Shale, Shale & organic-rich Shale) using the XGBoost results from the unbalanced (Shale dominate) and the fully-balanced (maximum weight for minor classes) where we used the confusion matrix to reclassify: Sandstone into Clean/Dirty; Sandstone-Shale into Silty-SS, Sandy-Silt, Siltstone; Marlstone into mud-stone, Marl, Marly-limestone; and Limestone into clean & dirty.

DISCUSSION: WHAT WAS THE BEST?

XGBoost easily outperformed the other models as the need to impute missing values appears to be one of the critical factors. The results also were highly dependent on the strategies chosen to balance the classes. Increasing the weight resulted in all cases in an improved balance accuracy but a decreased F1 and contest scores. The organizers of the contest use a scoring method that penalizes wrong lithofacies classification based on a *penalty matrix* styled after an F1 score but reduces the penalty for similar mineralogy (confusing Chalk for Limestone) and increases it for significant errors (Halite for Quartz).

Contest results: Balanced Models

Our first strategy was to focus on the less frequent classes. For that, we both tested under-sampled the data and using computed the class weights. 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.

Three different algorithms were tested: *gradient boosting*, *logistic regression*, and *naïve bayes*. Each one assumes its own strategy for classification, but gradient boosting is the

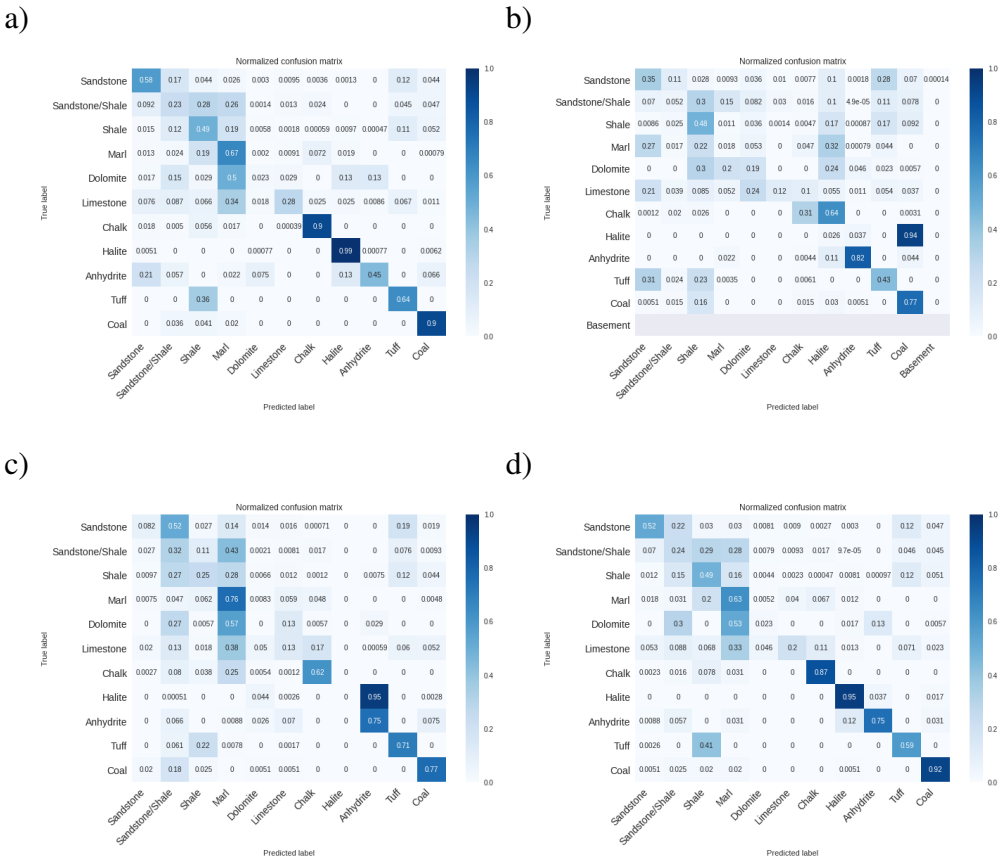

FIG. 7. Confusion matrix for the a) gradient boosting, b) logistic regression, c) naïve bayes, and d) stacked classifiers.

only non-linear one. All the algorithms were set to run for unbalanced classes (undersampling and class weights) and, in the end, we stack the models by a "soft" voting system (using each model's probability outputs). Figure 7 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 weighted 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 actually did not aid the predictions when stacked. This comes to the assumption of a linear relationship between the logs and lithofacies, which 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.

Figure 8 shows the prediction over one of the validation wells. Predicted lithofacies are promising, but with an increased proportion of misclassification of the most frequent classes when increasing weight. But overall, the model is catching the changes in lithol-

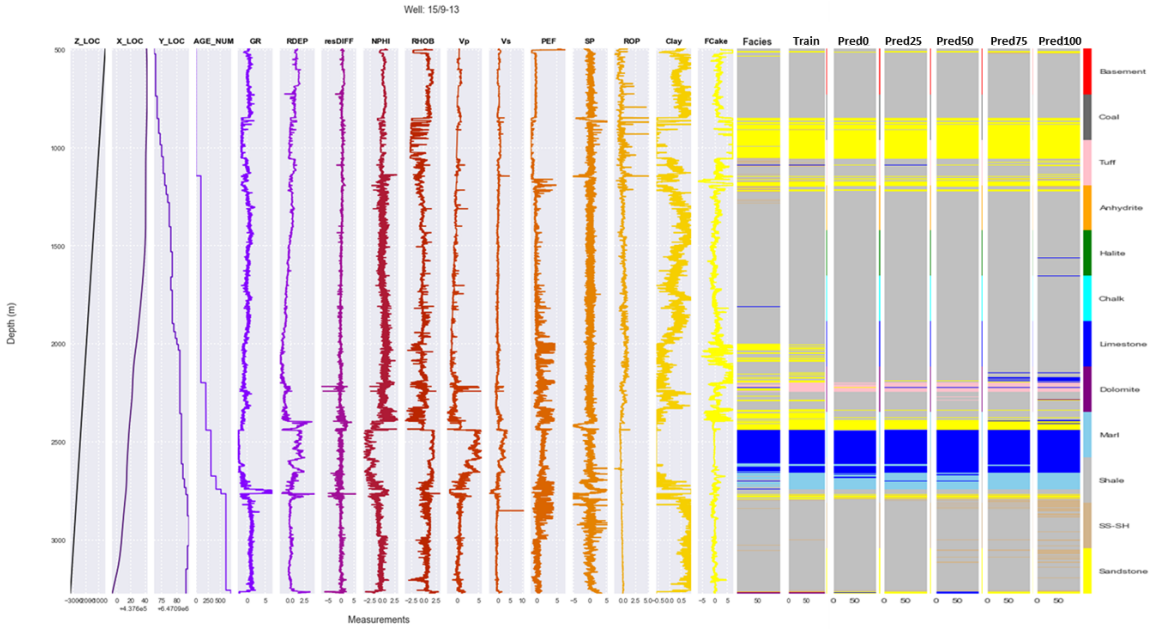

FIG. 8. Predictions with different weights

ogy. According to the balanced accuracy the best model with the highest weights but also performed poorly on the contest metric with a score of -1.35. As a comparison predicts all the samples as *shale* would result in an accuracy of 0.08 but a contest score of -0.96.

Table 1. 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. Shale Only	0.08	-0.96
7. Stacked Models (balanced)	0.56	-1.38
8. Stacked Models	0.41	-0.58

Table 1 contains the *balanced accuracy* and *contest* metrics for all the tested models done for the contest. It became clear that the contest metric, focus on the correct classification of the most common classes. The best contest score was achieved by stacking the unbalanced 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, 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.

Ongoing work

After the contest, we performed the compensating for the depth trends, using a semi-balanced approach, and sub-class of the mixed mineralogy class. The majority of this work was done using just XGBoost (Figure 9) and, as expected, partial balance produced the best result for the contest metric. On the other hand, some surprises were: how successful XGBoost was at fitting the raw data, the negative effect imputing had, and how dependent our contest performance had been on including the Group and Formation information.

For this analysis the validation was done by performing 4 runs, each with a different well held out and all 4 validation runs summed to check performance. Our work indicates that having a geological framework was significantly useful but this also means that our model may not generalize well to other geologies. The performance of XGBoost using the raw unedited data (all curves) was basically equivalent to the feature engineering when we limited the analysis to just the petrophysical logs. Imputing through linear regression had a significant reduction in performance.

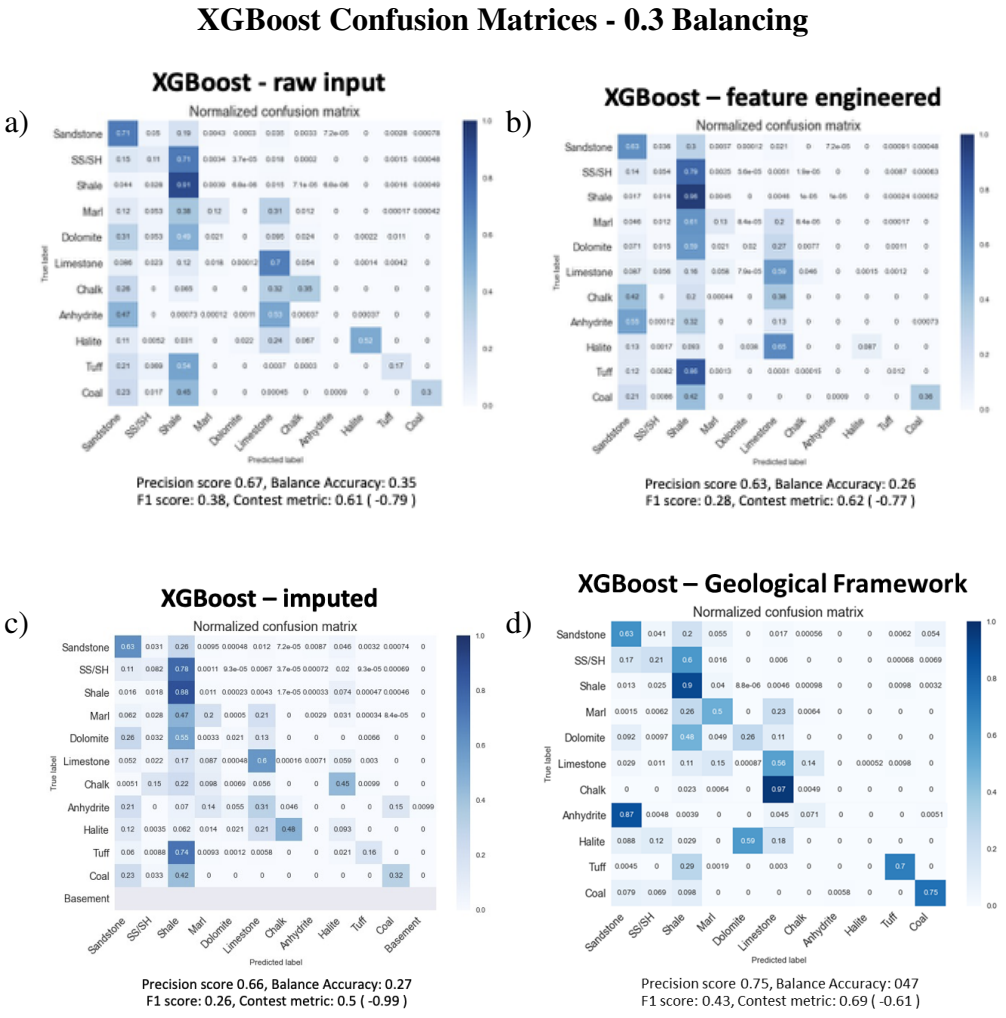

FIG. 9. Confusion matrix for a) raw data, b) feature engineered, c) imputed features, and d) geological framework.

Evaluation of the feature importance (Figure 10) for the solution using the raw data and the engineered curves, indicates how easily a good solution can be achieved from over-fitting. While the XGBoost on the unedited input the first four were not dynastic logs, for the feature engineer solution, 5 of the top 6 (GR, NPHI, DTC, RMED& RHOB) make petrophysical sense.

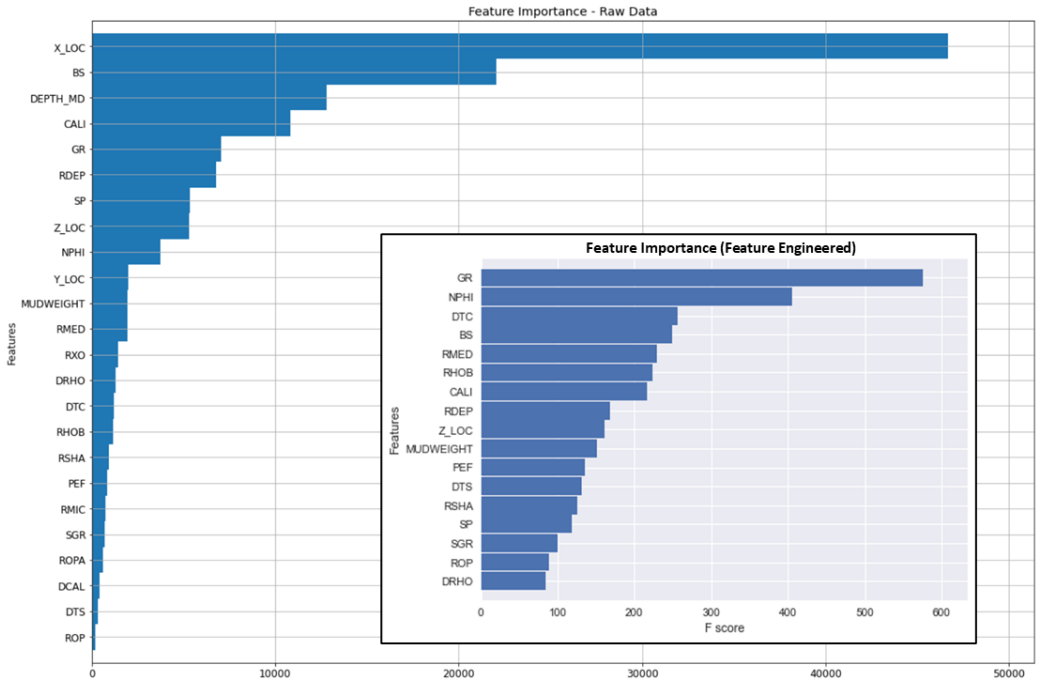

FIG. 10. Feature Importance - Raw with inlay for Feature Engineered curves

To deal with the ill-conditioning of the input lithoclasses, we created a more generalized sub-class model using the confusion matrix for the unbalanced (favoured Shale) and the fully balanced. For the XGBoost solution we only used the Petrophysical curves (GR, RDEP, RESdiff, NPHI, Vp, VpVs, RHOB, SP, & PEF) and we created 7 new sub-class by this process: Calc-Sandstone, Silty-Sandstone, Silty-Shale, Calc-Shale, Dirty-Limestone, Tight-Limestone & Dirty-Coal.

Figure 11 is the output from re-combining the sub-class back into the original litho-classes. General performance has improved particularly on the more minor class but additional work will be required.

Finding a good machine learning solution to Petrophysical analysis is highly probable, but finding the right approach for data preparation, mineralogy determination, and finally, lithology classification will take time. Our present work would indicate that having a geological framework should be significant when applied in a regional setting. A single-point estimator may be limited to determining mineralogy and another approach required for lithology or facies classification.

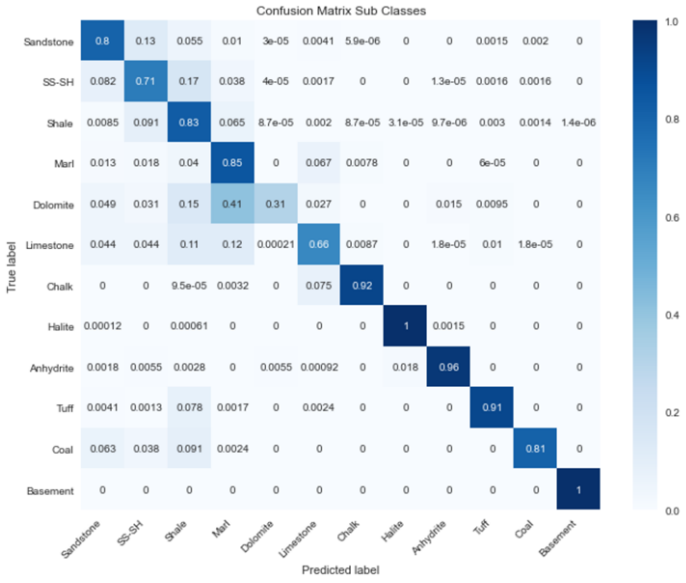

FIG. 11. Confusion Matrix - Using Sub-Classes

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 focuses on balanced classification, great to identify rare occurrences, another focused on the more common labels, scoring better in the contest, and another focusing on the geological framework. The workflow starts with the data cleaning and ends up with the modeling and prediction of the lithofacies from the well logs.

The methods investigated for the contest were Logistic Regression, Naive Bayes, Random Forest, and Gradient Boosting. Following the contest, we investigated: improving the feature engineering by detrending, the role impute had on the results, and redefining the lithofacies class. Regrettably, the Machine Learning methods investigated predominately focused on the mineralogy, resulting in a poor classification of mixed lithofacies such as Marlstone, Coquina, Interbed Sandstone-Shale unit, etc., and confused units with similar mineralogy such as Chalk being classified as Limestone.

Improvements came when using XGBoost after feature engineering along with including the geological framework (X ,Y & formations), gave workable results and a reasonable score for the contest with the advantage of not needing imputation of missing data.

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.

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.

Emery, D. J., Guarido, M., Trad, D., and Innanen, K., 2020, Lessons learned, pitfalls and feature engineering for force 2020: Log facies classification using machine learning.: GeoConvention Expanded Abstract 2020.

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

Guarido, M., Emery, D. J., Macquet, M., Trad, D., and Innanen, K., 2020, The pitfalls and insights of log facies classification for a machine learning contest: CREWES Research Report, 32.

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.