FORCE 2020 Lithology Prediction technical retrospective

The FORCE 2020 Lithology Prediction ML competition is over. A total of 329 teams signed up for the competition and 148 teams submitted predictions on the open test dataset to enter the competition leaderboard. At the end of the competition the top 30 teams in the leaderboard were invited to submit their pre-trained models for scoring on a hidden dataset.

Not your everyday ML competition

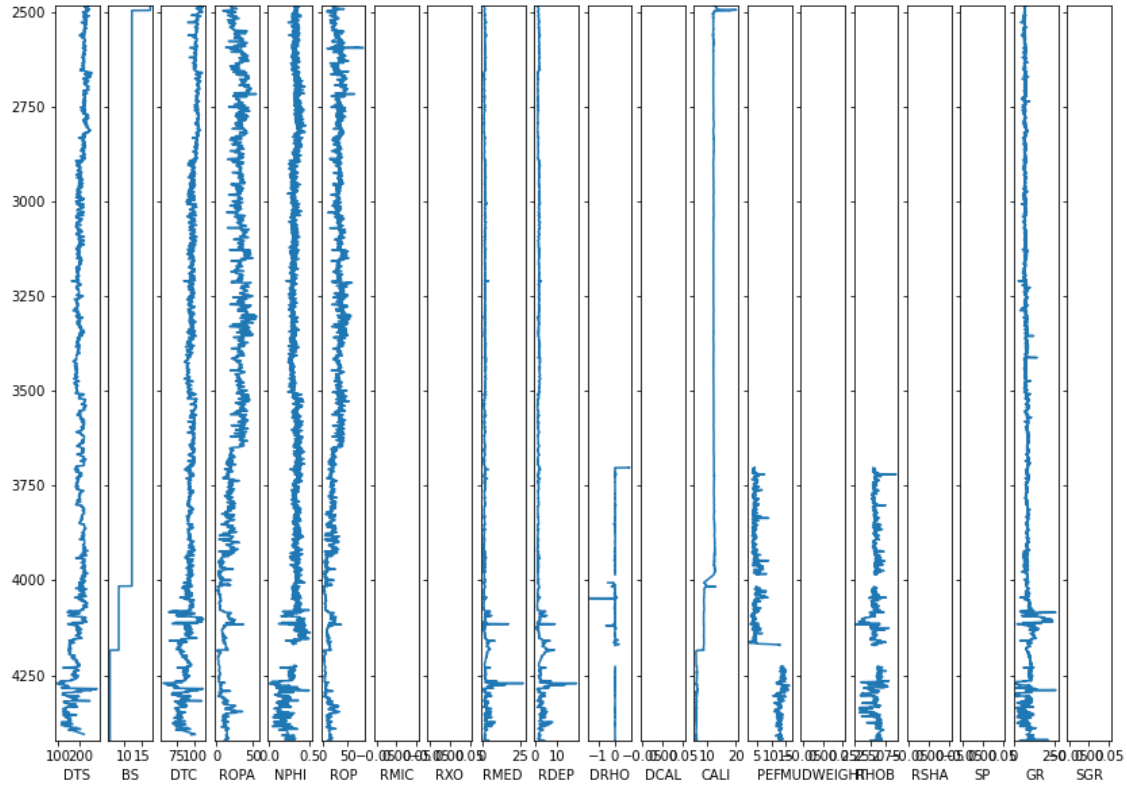
The FORCE 2020 Lithology Prediction competition was not your run-of-the-mill Kaggle competition. Several complicating factors were introduced to make the task more realistic, and ultimately make the always elusive leap from competition to practical use for well interpretation more feasible.

Firstly, instead of scoring the classification using metrics like accuracy of F1-score, errors were not punished equally. For example, mistaking Shale for Marl was considered significantly better than mistaking Shale for Anhydrite. This geologically motivated classification metric was encoded in the penalty matrix as shown below.

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

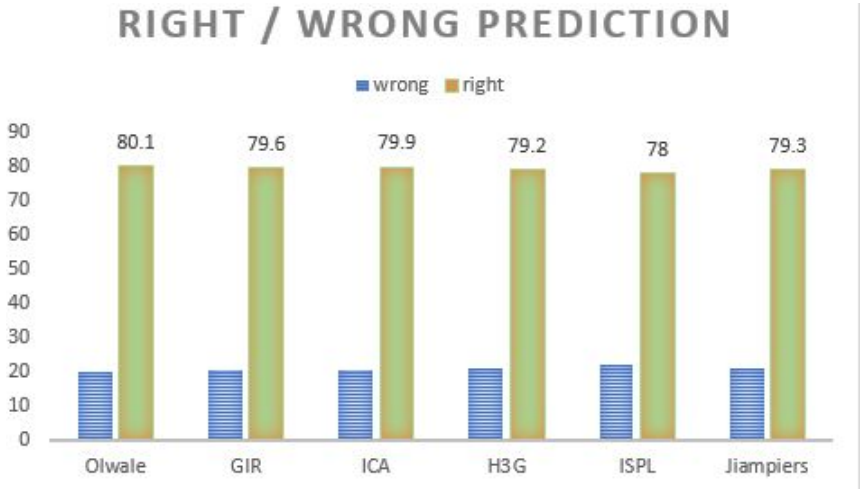
There is always a danger in introducing custom and more complicated metrics of evaluation in a machine learning competition. When complexity goes up, so does the potential for abuse and unintended optimization. However, by introducing such a metric you can hope to reduce the often problematic difference between the solution of the optimization problem and what a domain expert may think is a good interpretation of a well log.

Secondly, like in real well logs, the availability and coverage of curves varied greatly from well to well. Only a small number of curves were guaranteed to be available for the training and test data. For the blind test data used in final scoring, the teams did not know the availability of curves beyond what was guaranteed. This immediately complicates the application of machine learning to the problem. Leveraging varying feature availability in an interpretation task is one of the things humans often do better and more naturally than computers. Also, real well logs have missing curves in sections or in the whole well. Curating an artificially clean dataset for this competition will have made the leap to practical application difficult, if not impossible.



How did the top teams solve it?

With the competition being over, it is worth looking at how the top teams approached these challenges. After inviting the top 30 teams in the leaderboard to submit their code for final scoring, the top three teams were Olawale, GIR and Lab.ICA. The competition was extremely close, as shown in the top six teams' accuracy on the blind test data



The number of wells in the blind final test data was so small (10) that it is highly likely there are other high-performing and interesting solutions out there. We will keep adding team codes to the repository, and at some point there may be a more comprehensive review, but for now let's have a look at how some of these top teams handled the complexities of the FORCE Lithology Prediction challenge.

OLAWALE

Olawale uses a 10-fold stratified cross validation technique around the Kaggle favorite XGBoost algorithm (https://xgboost.readthedocs.io/en/latest/) to predict the lithology. The XGBoost method was also used by the GIR team, but Olawale employs more customization. XGBoost is a gradient boosting algorithm, i.e. it trains an ensemble of weak classifiers and turns it into a classifier. The cross validation scheme likely helped Olawale avoid overfitting to the leaderboard open test dataset. The effort he put into preventing overfitting seems to have paid off, as he is one of the few that saw a significant improvement in scoring on the blind test data compared to the open leaderboard score.

Olawale drops some of the rare curves such as 'SGR', 'DTS', 'RXO', 'ROPA', before running a suite of feature engineering steps including windowing and gradients. Interestingly, the winner of the competition made no effort to impute or estimate missing curve values.

Olawale did not explicitly use the custom scoring matrix when training the model and instead relied on a standard multiclass log-loss for the optimizer. The custom scoring was likely only used in model selection, hyperparameter optimization, and when choosing which curves to ignore.

GIR TEAM

The GIR team uses a standard implementation of the Kaggle favorite XGBoost algorithm to carry out the classification. This is the same method used by Olawale.The magic sauce of the GIR team is hence not in its choice of classifier, but in the imputation of missing curves and feature augmentation. The GIR team uses physical understanding of the curves to pick specifically which other curves are likely to indicate missing values. For example,

They augment every curve with non-local information, in this case gradients. Moreover, they engineer polynomial features of the original features.

The GIR team did not include the custom scoring matrix in the optimization or training of the model, and likely only used it for model selection.

ICA TEAM

The ICA team uses the RandomForest algorithm, another common boosted trees method that is conceptually not too different from XGBoost used by the top two teams. The ICA team uses a 5-fold stratified cross validation technique, similar to what Olawale employed.

In the feature engineering step the ICA team adds both normalized GR and RHOB, as well as the gradient of a number of curves. They include the formation as a feature. No attempt is made to impute missing curves, and these are simply filled with median values.

The ICA team also did not include the custom scoring matrix in training the model, and presumably only used it for model selection.

Some (final) thoughts

Combining boosted trees with common sense feature augmentation and preprocessing seems to have won the day. It is interesting that none of the top teams explicitly used the custom scoring metric in the optimization of their model parameters. It seems that using standard implementations of time proven methods in combination with powerful measures to prevent overfitting may be more effective than introducing custom loss functions adhering to the geologically motivated penalty matrix.

This will not be the last word written on the FORCE Lithology Prediction challenge. The biggest legacy of this competition is the dataset of 118 curated and consistently interpreted well logs, which is now in the public domain. With at least 148 teams having completed the challenge I am curious what creative solutions are out there. We will keep adding team codes and links to other repositories on the official competition github (https://github.com/bolgebrygg/Force-2020-Machine-Learning-competition) as they come in!