Capstone_final_report

February 14, 2021

1 2020 FORCE ML Competition: Lithology prediction using Grouped models

by: Rafael Pinto

2 Introduction

FORCE is a cooperating forum for improved exploration and improved oil and gas recovery conducted by oil and gas companies and Norway authorities. In 2020, this institution, in collaboration with its sponsors, organized the 2020 FORCE ML Competition. Two independent challenges were created:

1. Lithology prediction
2. Mapping faults on seismic data

In this work, I focus on the Lithology prediction challenge. I explore the winning model (Olawale's), propose an update to the feature enhancing functions in this solution, and perform tests on the model implementation decisions to understand their effect on the model score using the open data set. Finally, I proposed a model-building strategy using the geologic Groups and compare it to the reference models.

3 Problem identification

All rocks have defining properties that can be measured with, in the case of subsurface rocks, sophisticated apparatuses, or, as they are known in the industry, downhole tools. These measurements are collected when a well is drilled, but the corresponding type of rock or lithology is unknown. Geologists and petrophysicists evaluate these data to assign a lithology class to a set of measurements based on physical models and experience.

The lithology classification process is laborious and not scalable. It can take 2-3 days for experienced petrophysicists to evaluate a single well, depending on the available data's quantity and quality. Also, most of the time the evaluation process is carried on a well by well or one well at a time basis, which forfeits the use of the spatial information in the analysis.

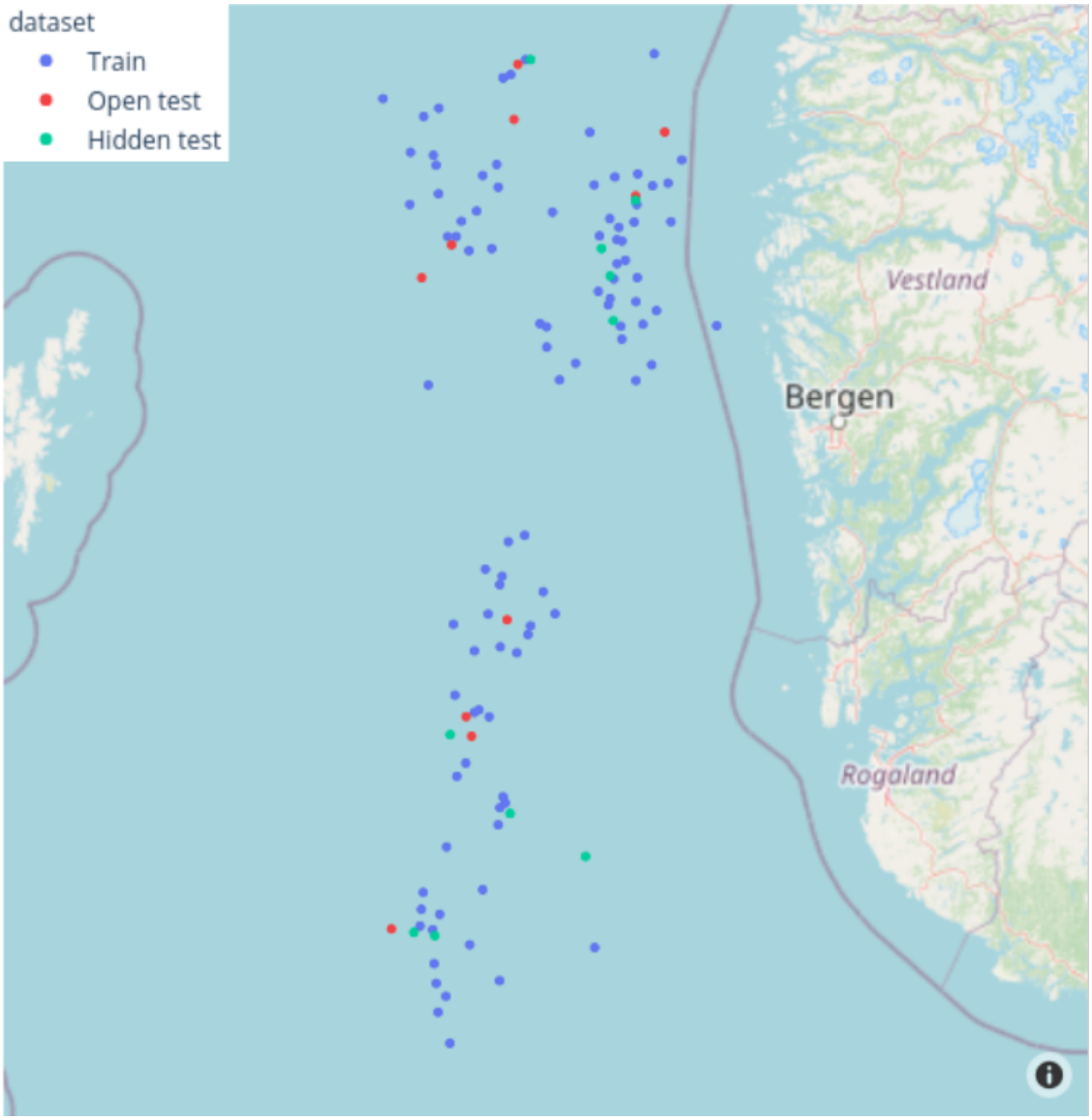
As a result, there is an increased need to perform this process in an automated fashion, to assist in the lithology classification when done by hand, by building a starting best guess, and to process volumes of wells at once, e.g., on basin-scale studies with hundreds or thousands of wells.

3.1 Data description

The 2020 FORCE ML contest provides a nice set of well log data from the North Sea. There are more than 20 distinct well logs, albeit not all the logs are present in all the wells, and the intersection where all well logs have non-missing values per well is rare. These observations also define typical field data. During the competition, only two datasets were available to participants:

1. Train set: 98 wells
2. Open test set: 10 wells

After the competition closed, the withheld data set (hidden) used to perform the final model ranking was released to the public. It consists of 10 wells from the same area. The map view of these wells is presented below, modified from Hall (2020).



3.2 Sucess criteria

The competition organizers provided a success metric S, which depends on a misclassification penalty matrix. There are 12 possible lithology classes. The idea is to punish geologically unreasonable results harder than geologically plausible errors. The scoring function is defined as follows:

$$S = -\frac{1}{N}\sum_{i=0}^{N} A_{\hat{y}_i y_i}$$

where N is the number of samples, y_i is the prediction for sample i, $\hat{y}_i$ is the true target for sample i, and A is the penalty matrix.

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

Under this scoring function and penalty matrix, the perfect score is zero, and less than perfect models will have scores that is less than zero. My goal is to achieve a more than -0.52 score in the hidden data set, putting my work within the top 13 submissions in the competition.

4 Solution strategy

I wanted to learn from the competition winner, Olawale, so I decided to start from his work. The winning model is presented in a self-contained jupyter notebook, which facilitated the code review. This submission has four primary data preprocessing steps:

1. Drop uncommon columns: CONFIDENCE, SGR, DTS, RXO, ROPA. Except for CONFIDENCE, these columns have very high rates of missing values. The CONFIDENCE column is tied to the train set target, so it makes sense to drop it.
2. Label encode categorical columns.

3. Fill missing and infinite values with -999
4. Augment the features with shift and gradient functions (Bestagini, 2016).

4.1 Preprocessing observations

4.1.1 Label encoding

There is a lot of debate on how to encode labels for classification tasks properly. On the one hand, the label encoder is only recommended to be used on the target variable. However, it seems that the model type and implementation define this limitation. Like many things in machine learning, we can test different approaches and decide based on the results.

The winning submission performs label encoding with the pandas `.astype` and `cat.codes` DataFrame methods. A possible unintended consequence of this operation is that all missing values in the category colums will be replaced with -1 in the encoded columns. In my model, I used a label encoder for the categorical features, but unlike the winning submission, I created a couple of functions (`build_encoding_map` and `label_encode_columns`) to prevent the value -1 from being assigned to missing values.

Also, I didn't include the WELL column as a feature in my models, as I consider this column to be an identifier and not a property of the rocks.

4.1.2 Shift and gradient

Bestagini's functions are designed to work with numpy arrays. I wanted to understand how they work and enable them to operate directly on pandas series, so I rewrote them accordingly (notebook 4.0-rp-build-features-bestagini).

As a result, I learned that the original functions pad the resulting array with zeros, in the places where a missing value would have been introduced by either shifting the logs up or down, or by taking the gradient. The original functions return the indexes of the padded rows, which in the 2016 ml competition are used to drop these rows, but on the 2020 FORCE winning submission, these are not used effectively replacing the missing values resulting from these operations with zeros.

Also, the winning submission applies the gradient to all the columns, including category columns (FORMATION, GROUP, WELL). Since these are not ordinal categories, I decided not to take their gradient.

4.1.3 Missing values

With the observations described above, the winning submission has three types of missing value representation:

1. -1: For missing values in categorical columns.
2. 0: For missing values introduced by shifting or taking the gradient on the logs.
3. -999: For missing values in the raw features.

In my models, I treated all missing values equally, i.e., as `numpy.nan`.

5 Models

The winning submission split the train data into 10 folds using `StratifiedKFold`. The idea was to reduce the possibility of overfitting. For each fold, an XGBoost Classifier was fit using the train part of the set, while the test part of the split was used as the evaluation set (`eval_set`) to monitor the model's performance. The model was trained until the validation didn't improve in 100 rounds.

There are 10 models after the training is done. The prediction is run on each model, resulting in 10 probability matrices that are then averaged across models. We assign the prediction label at each row by finding the maximum probability in this lithology average probability matrix.

I created a model just like the one described above, but with the preprocessing differences explained in the previous section. Let's refer to it as model 7, as a reference to the associated notebooks (7.0 and 7.1). Unfortunately, this model performance was far worse than the winning submission score (-0.515), scoring -0.538 on the open data set. Future work should investigate the reason for this discrepancy, which I believe lies with the treatment of missing values or the WELL feature's inclusion.

Next, I wanted to understand if this score difference was significant, for which I created a simple random forest model with no data split and no hyperparameter tuning. This model score -0.574 on the open data set. In sum, the difference between the winning submission (-0.515) and a simple random forest model (-0.574) was 0.059, which suggests that there is not a lot of sensitivity in this scoring function.

I also tested the data split's impact on the score by creating a model just like 7, but with no `StratifiedKFold`. This model scored -0.539 on the open data, which is close to model 7 -0.538 score. So the data split doesn't seem to affect the score.

5.1 Grouped model

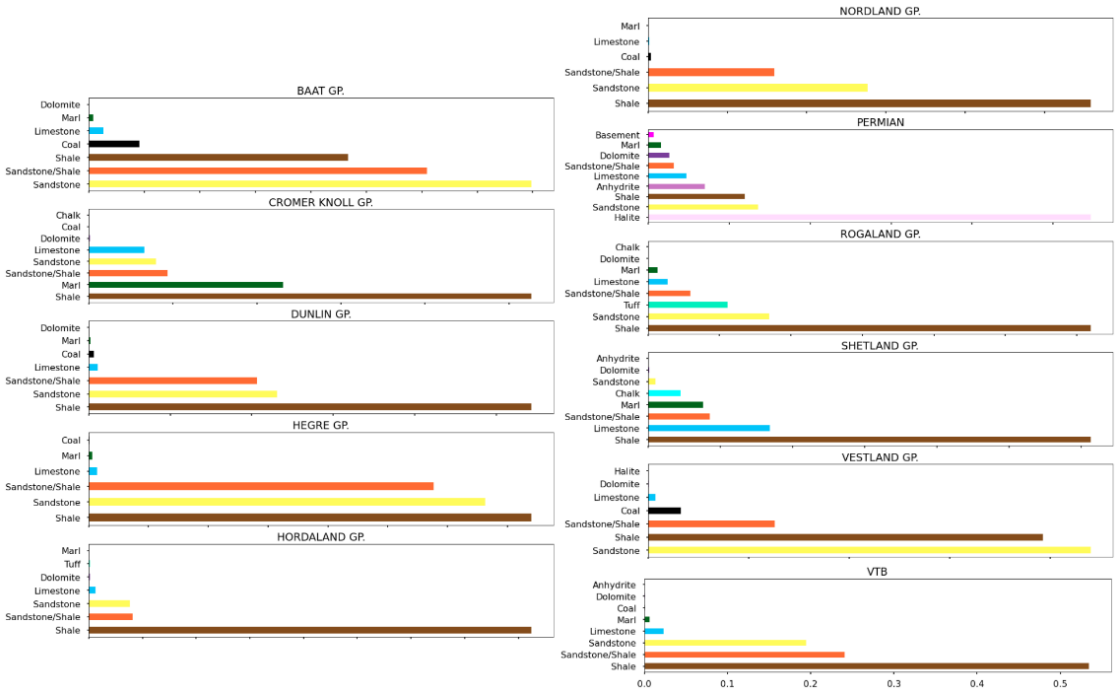
As a last attempt to improve the score of model 7 and inspired on the data split strategy from the winning submission, I devised a strategy to group the data by geologic GROUP, and then apply `StratifiedKFold` on each group, where K=5 instead of 10. Before doing this, I had to form super-groups for those groups with only a few wells per group (notebook 1.6-rp-eda-groups). I used the Lithostratigraphic Chart for the Norwegian North Sea to give some geologic context to these super-groups. Only two super-groups were needed:

1. VTB GP.: VIKING GP., BOKNFJORD GP. and TYNE GP.
2. PERMIAN GP.: ROTLIEGENDES GP. and ZECHSTEIN GP.

The downside of this approach is that it relies on the groups being available on each data set, which is the case for all three data sets in this competition. This narrows down the possible areas of application, i.e., this model will only work in areas where these groups exist, potentially only the Norwegian North Sea.

The vision behind this strategy has three founding ideas:

1. Most groups have only a handful of lithologies (1.6-rp-eda-groups), so we reduce the solution space in principle. The figure below shows the normalized lithology value count per group.



2. Some well logs trend with depth, making this a non-stationary problem. I tried de-trending some of the logs (1.3-rp-eda-gr-normalization and 1.5-rp-eda-rhob-detrend), but I couldn't come up with an easy way to do this. Alternatively, I thought that splitting the data into groups could alleviate this problem.

3. Not all well logs are present in all groups. If I could select the important logs that describe a given group's lithologies, the model might have a better chance of succeeding in the classification. The figure below shows the percent of non-missing samples for each feature per group.

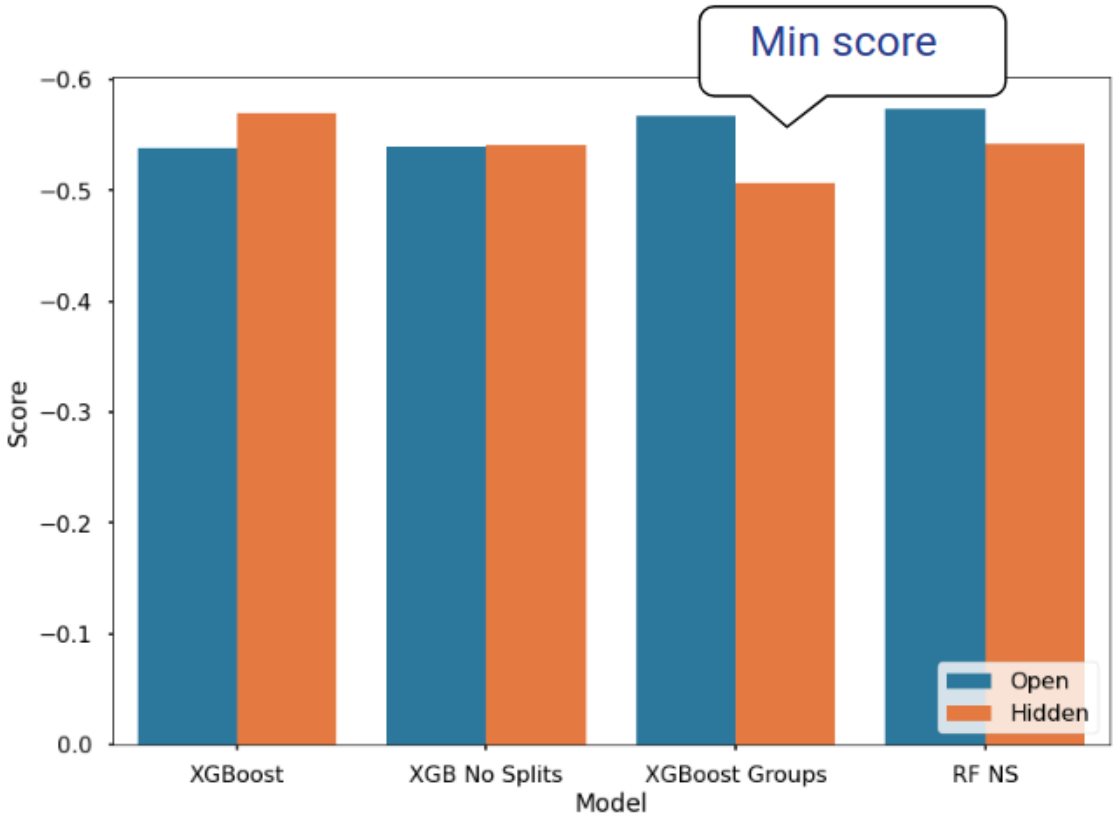
	BAAT GP.	CROMER KNOLL GP.	DUNLIN GP.	HEGRE GP.	HORDALAND GP.	NORDLAND GP.	PERMIAN	ROGALAND GP.	SHETLAND GP.	VESTLAND GP.	VTB
WELL	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000
DEPTH_MD	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000
X_LOC	1.000000	0.985168	1.000000	0.998778	0.998765	0.984133	0.694278	1.000000	0.995125	0.976643	0.997997
Y_LOC	1.000000	0.985168	1.000000	0.998778	0.998765	0.984133	0.694278	1.000000	0.995125	0.976643	0.997997
Z_LOC	1.000000	0.985168	1.000000	0.998778	0.998765	0.984133	0.694278	1.000000	0.995125	0.976643	0.997997
GROUP	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000
FORMATION	1.000000	1.000000	1.000000	1.000000	0.813641	0.272724	1.000000	1.000000	1.000000	1.000000	1.000000
CALI	0.995924	0.870260	0.970240	0.999712	0.908693	0.827240	0.999468	0.880768	0.949527	1.000000	0.981053
RSHA	0.748569	0.564163	0.634757	0.532308	0.501448	0.491712	0.364072	0.544390	0.432717	0.739776	0.671647
RMED	0.954024	0.975956	0.973767	0.998778	0.984152	0.933241	0.347305	0.985130	0.985886	0.974575	0.971275
RDEP	0.998772	0.985168	0.999496	0.998778	0.998765	0.984133	0.694278	1.000000	0.995125	0.976643	0.997000
RHOB	0.998102	0.963169	0.989881	0.999928	0.802487	0.523805	0.735196	0.917094	0.885245	0.999387	0.970380
GR	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000
SGR	0.195070	0.056881	0.134081	0.072235	0.022285	0.043529	0.000000	0.026117	0.025988	0.099403	0.138687
NPHI	0.998995	0.877733	0.989680	0.990656	0.434787	0.208673	0.741184	0.553424	0.682995	0.998660	0.962767
PEF	0.620216	0.593005	0.609649	0.555308	0.543310	0.423643	0.106387	0.597003	0.648833	0.757314	0.583816
DTC	0.984535	0.978192	0.940681	0.952275	0.922870	0.824827	0.982635	0.950646	0.929376	0.969176	0.971283
SP	0.707730	0.534002	0.749549	0.821965	0.830567	0.852668	0.303526	0.829958	0.599163	0.706655	0.724996
BS	0.521844	0.607798	0.620364	0.800331	0.528993	0.491452	0.682635	0.575138	0.599253	0.757505	0.668324
ROP	0.100969	0.592087	0.276601	0.454539	0.523010	0.614369	0.999601	0.416646	0.449980	0.752029	0.326035
DTS	0.186026	0.291036	0.134156	0.128944	0.041510	0.017625	0.063273	0.104938	0.265062	0.186705	0.286902
DCAL	0.041147	0.189469	0.267893	0.293538	0.328430	0.241501	0.210645	0.331095	0.196592	0.562031	0.151956
DRHO	0.966753	0.966858	0.971659	0.999928	0.758554	0.531106	0.736327	0.887983	0.876861	0.999464	0.967182
MUDWEIGHT	0.000000	0.252924	0.173162	0.293467	0.409527	0.398583	0.915902	0.303576	0.133467	0.529714	0.108788
RMIC	0.177902	0.313016	0.159088	0.095378	0.098562	0.000000	0.000000	0.188845	0.140398	0.222163	0.298308
ROPA	0.155905	0.380753	0.106193	0.137210	0.073367	0.059593	0.021490	0.086726	0.336417	0.123219	0.222968
RXO	0.353683	0.207798	0.325524	0.249048	0.250407	0.229895	0.105788	0.317589	0.236420	0.438773	0.383543
20_LITHOFACIES_LITHOLOGY	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000
0_LITHOFACIES_CONFIDENCE	0.999972	0.999560	0.999924	0.998850	0.999881	0.999776	0.999933	0.999932	0.999885	0.999847	0.999831
GROUPED	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000

We can see that the logs SGR, DTS, DCAL, RMIC, ROPA, and RXO have poor availability from the figure above. Also, only a few groups have sufficient ROP and MUDWEIGHT samples. I selected logs with more than 68% non-missing values in this model, except for the FORMATION log, which I included as a feature in all the groups.

5.2 Score

I applied the scoring function to the results of predicting the lithologies using the open and hidden data sets. The results are shown in the table and companion figure below.

Model	Notebook	Open score	Hidden Score
XGBoost Groups	9.0, 9.1, 9.2	-0.567	-0.506
XGBoost No Splits	10.0, 10.1, 10.2	-0.539	-0.541
Random Forest No Splits	12.0, 12.1, 12.2	-0.574	-0.542
XGBoost	7.0, 7.1, 7.2	-0.538	-0.570



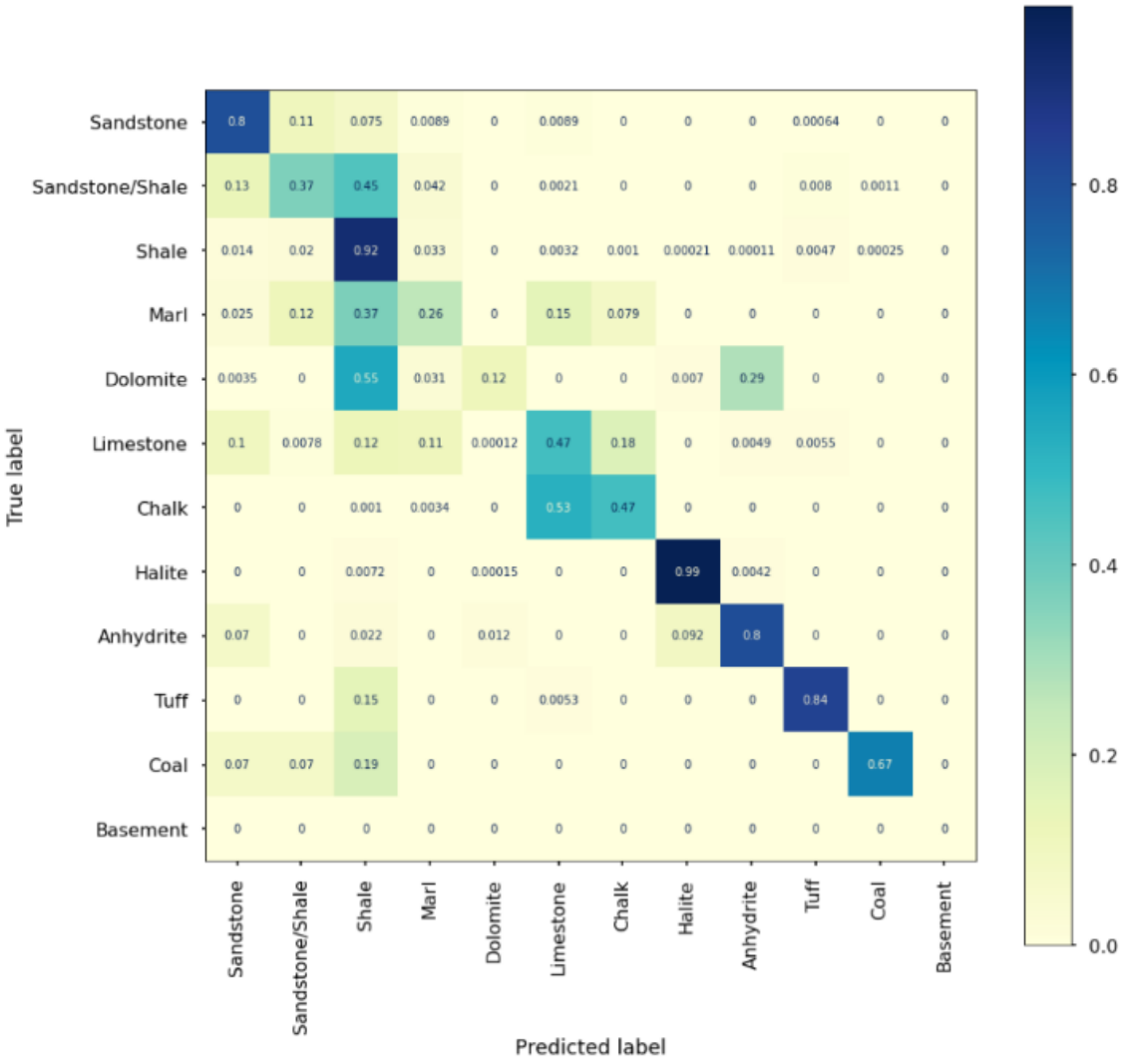
With this hidden score, the Grouped model would have ranked 5th in the final leaderboard.

-0.506

Final Score	My team name /personal name is	Score on hidden dataset	My current score XEEK leader board is	My current position XEEK leader board is
1	**Olawale Ibrahim**	-0.469	-0.5118	24
2	**GIR TEAM**	-0.4792	-0.5037	11
3	**Lab.ICA-Team / Smith A.**	-0.49536	-0.4943	6
4	H3G (Haoyuan Zhang, Harry Brandsen, Gregory Barrere, Helena Nandi Formentin)	-0.504489	-0.509	17
5	ISPL Team	-0.50835	-0.4885	2
6	Jiampiers C.	-0.50886	-0.5014	9
7	José Bermúdez	-0.509061	-0.5052	14
8	Bohdan Pavlyshenko	-0.51713	-0.5112	22
9	Jeremy Zhao	-0.51733	0.5264	31
10	Campbell Hutcheson	-0.52206	-0.505	13

Looking at the confusion matrix derived from the Grouped model applied on the hidden test data (9.3-rp-fit-predict-save-proba-grouped-hidden-score) we can draw the following observations:

1. Sandstone/Shales is easy to confuse with either Sandstone or Shale.
2. The model does very well at predicting Shale, Halite, and Anhydrite.
3. The model struggles to separate Shale from Marl, Dolomite, Tuff, and Coal.
4. The model struggles to separate Chalk from Limestone.
5. There were no Basement samples on the Hidden test set.



6 Future work

The Grouped model has a competitive score on the hidden test data, but it can be improved. The following steps should be considered next:

1. Build a category log comparison to visualize sample by sample where the misclassification occurs.
2. Explore the differences between my model 7 and the winning submission.
3. Tune the Grouped model hyperparameters.
4. Consider using a cost-sensitive approach to train the model (e.g., MetaCost).

7 References

Bestagini P. 2016. 2016 ml contest ispl

Bormann P., Aursand P., Dilib F., Dischington P., Manral S. 2020. FORCE Machine Learning Competition. https://github.com/bolgebrygg/Force-2020-Machine-Learning-competition

Hall, B. 2020. FORCE-2020-Lithology repository.