Obtaining a Complete Well Log Dataset Using Artificial Intelligence Tools

H. Uzkeda Apesteguia[1], O. Vidal Royo[2], A. Amilibia[2], F. O. Escosa[2]

[1]Departamento de Geodinámica, Estratigrafía y Paleontología, Universidad Complutense de Madrid
[2]Terractiva Consulting SL

Extended abstract

Introduction

Artificial intelligence (AI) tools have proven useful to predict well logs and petrophysical parameters in areas where datasets are either scarce or incomplete, which make them particularly suitable to optimize subsurface exploration of natural resources. One common issue found when working with well log data is the irregular abundance/availability of the parameters registered. This is especially relevant when working with datasets collected in different surveys that may span through years, even decades, or different companies using different standards. AI tools can be used to fill the gaps in the original database, resulting in a more complete, standardized one. In this work we present a workflow that consists of four main steps: 1) feature combination selection; 2) hyperparameter tuning; 3) performance assessment and best option choice; 4) blind testing.

We present an example in which we filled the missing density values in an incomplete dataset consisting of wells provided by the UK National Data Repository (NDR) of the Oil & Gas Authority (OGA), using the scikit-learn libraries implemented in Python (Python 3.8, ran in the Spyder software package).

Dataset

In this work we used free, public data kindly provided by the OGA through the NDR. Our dataset consisted of 44 wells from quadrants 15, 16, 22, 30 and 31, drilled by seven different companies between 1974 and 2017. The training/validation process was performed with the wells from quadrants 15, 16, 22 and 31, whereas the two wells from quadrant 30 were saved for the blind testing.

After compiling all the data and converting it into an AI-ready matrix of more than 800,000 rows we selected only those parameters recorded in, at least, 30% of the total number of observations, these attributes are: deep resistivity, density, gamma ray, medium resistivity, neutron porosity, shallow resistivity, and sonic DT.

Before starting with the regression process, we carried out a statistical analysis to determine whether some conversion, regularization or filtering might be needed to ensure consistency across all the wells employed. For instance, some density values had to be transformed from kg/m^3 to g/cm^3 and all the neutron porosity data was converted to V/V (fraction of one, instead of %). Additionally, we removed the 0.5-1.0 upper and lower percentiles to avoid issues related to extreme values and calculated the logarithms of the resistivities values.

Finally, we applied a min-max scaler to bring all the features to within a [1-10] range, an order of magnitude like that of the feature (density) to be predicted.

Regressor training

It consisted of the four stages stated above: feature combination selection, hyperparameter tuning, performance assessment and blind testing. During the feature combination selection phase, we tried to discover which of the 127 possible combinations (7x1 feature; 21x2 features; 35x3 features; 35x4 features; 21x5 features; 7x6 features; 1x7 features) worked better. To accelerate the process, we discarded some of them beforehand: those with only one input feature; those that, because of the data availability, were not present in, at least, 30% of the total observations; those combinations with two or more resistivities to avoid introducing redundant data as there is a direct relation between the different resistivities. In the end, 18 combinations were selected for trying the different methods and running the hyperparameter tuning.

Hyperparameter tuning

We chose three of the more commonly used AI methods: support vector regressor (SVR), random forests RF) and neural networks (multi-layer perceptron regressor, MLPR). For each of them we created three search grids varying some of the hyperparameters to find the best predictor for density in our dataset. 112 regressors (32 RF, 32 SVR, and 48 MLPR) were trained and validated with the 18 previously selected input feature combinations. We used k-fold cross validation, with each validation set formed by 4 wells. k oscillated between 5 and 7 depending on the feature combination and the corresponding data availability.

Performance assessment

To evaluate the performance of the different combinations and select the best one, we created predicted versus observed density plots for each run and extracted their slope and dispersion (R2) to calculate the average values for the different folds of each permutation. To compare the different options using only one parameter, we employed the harmonic mean of the slope and the R2. An approach similar to using the typical F1 score for classification problems that combines precision and recall at the same time. The harmonic mean helps to discard options with low precision and high accuracy, or, low accuracy and high precision, and to identify those options with a better balance between the two indicators. For the blind testing we selected two of the best possible combinations, given that their scores were very similar.

Blind testing

We applied the two theoretically best combination to two wells from quadrant 30 left out during the training/validation process. Their results should reflect the applicability of the regressors to other areas or datasets. As during the training/validation process, we generated predicted versus observed plots to calculate the slope, dispersion and pseudo-F1 score. When comparing to the validation, the estimated errors were slightly greater (increasing from about 2% to 4% normalized values) and the pseudo-F1 score lower (0.926 to 0.793). This is consequence of a higher dispersion, mostly derived from observations in well 30/13-7 at depths between 4,553.0 to 5,674.5 ft (~1388 to 1729 m). Out of this region the error is of around 3%, whereas within is almost 11%. This area of larger error corresponds to an intensely fractured succession where a rapid increment of the pore pressure exists. This could yield greater errors because two reasons: 1) such lithology/fracture characteristics were not found in the training/validation sets and, thus, the regressor does not know how to handle it; 2) the lithological properties might have hindered the logging, making it less reliable.

Despite the issue described above, the overall fitting was acceptable, making this regressor applicable with enough confidence.

Regressor application

Once, through the blind testing, we were certain of the quality of the trained regressor, we used it to fill the gaps in our original dataset, increasing the number of observations with density in a ~30%. The AI prediction provided density values for a wider range of depths, in contrast to the measured data that was restricted to the lower ~2000-4000 ft. We generated LAS files for each well including the predicted density and a newly created column identifying the value as either originally recorded or predicted with AI tools

Conclusions and recommendations

The workflow proposed in this study culminates with the attainment of two types of outcomes: more complete well log datasets and ready-to-use/pre-trained regressors.

In our example, with only the application of the best regressor, the observations with density increased a ~30%. This value might be further raised if less optimal regressors were used, at the cost of greater errors. In such cases, we recommend to progressively fill gaps starting with the best solution (highest pseudo-F1 score) and continuing with the less apt feature, method and hyperparameter combinations (lower pseudo-F1 scores). Although, commonly, ad-hoc (i.e., trained with the dataset where they will be used or a relatively similar one) regressors offers better results than ready-to-use/pre-trained regressors, the latter may be of great help when, for instance, training/validation is not possible (e.g., lack of enough data) or during initial stages to provide quick estimations, as the application is much faster than the whole process of training, validation and testing.

Another important point is, when exporting the AI-filled log file, to clearly state what data was initially recorded (measured in the well) and what were estimated using the AI methods.

Acknowledgements

We would like to acknowledge the OGA for providing the well data and the scikit-learn project for the AI tools employed. This work was partially funded by the National Research Agency (Agencia Estatal de Investigación, Ministerio de Ciencia, Innovación y Universidades of Spain) through their Torres Quevedo program. Also, we would like to thank Grant George Buffett for reviewing the English.