IP_Prediction3D

Machine learning well logs prediction using Seismic cubes

User Manual

IPLAB LLC

Table of Contents

1. Introduction

IP_Prediction3D IP_Seismic plug-in (version: 2019.1.0.2, release date: October 2019) can be used to predict well log property in the layer applying a set of seismic cubes and well logs in boreholes.

Well log property prediction in a layer applying a set of seismic cubes and well logs in boreholes by using machine learning algorithms (linear regression, nearest neighbor regression, classic neural network and Kolmogorov t-neural networks). Virtual cubes are the result of this plug-in application with average, standard deviation, P10, P50, P90 predictions according to several cross validations calculations.

There are two main calculation studies to perform this prediction:

1. Training stage – training stage for machine learning algorithms according to the array of training pairs – well log property and values of cubes attributes around the well log samples position. During this stage, all machine learning coefficients (neural network or other) will be estimated to minimize objective function.
2. Calculation stage – to calculate values of predictive parameter (well log) for each sample in the cube based on input cubes.

2. Start

Procedures -> Seismic Attributes-> Machine Learning cube prediction

3. Input parameters

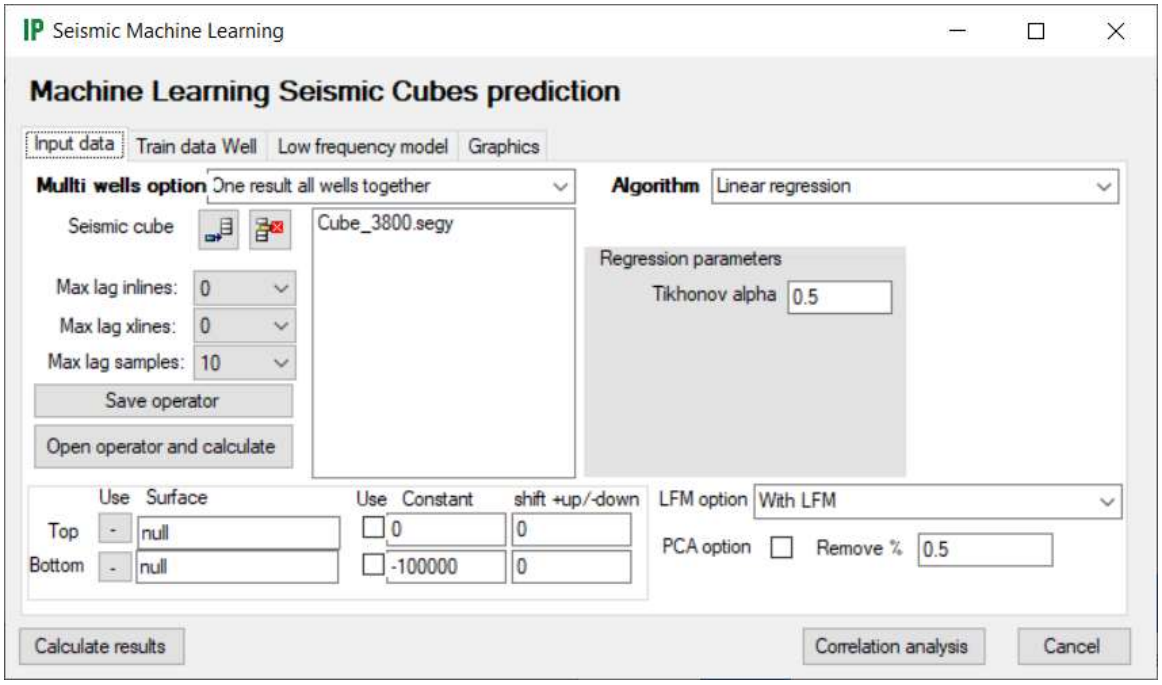
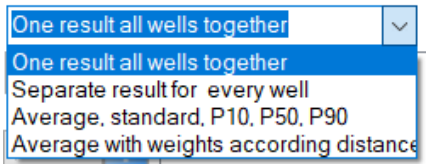


Figure 2: Input surface attributes data tab dialog view **IP_Prediction3D**

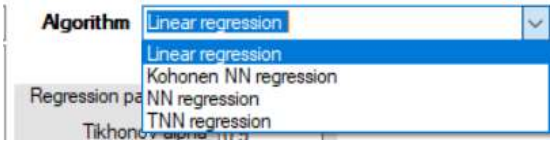
It is necessary to define the following parameters before calculations:

Multi wells option:



One result for all wells together – for all wells only one result – there is no cross validation.
Separate result for every well – it is necessary to investigate all wells separately.
Average, standard, P10, P50, P90 – separate prediction is used for each sample in cube and calculate average, standard, P10, P50, P90 – the recommended option
Average with weights according to the distance to wells – calculate weighting average according to the prediction from each well and using inverse distance weights to the wells. It is not necessary to apply a separate procedure for well matching to obtain the results.

Algorithms



Linear regression

With parameters:

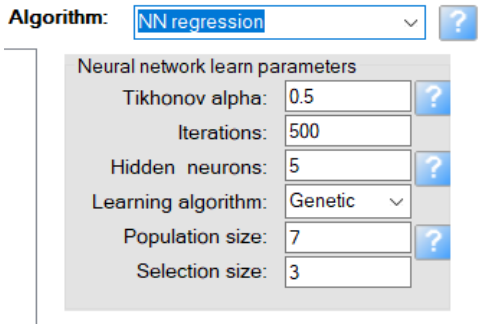
Tikhonov alpha: >0 and <1000 allow to avoid overlearning effect or instability for prediction. If alpha=0, then we can get a very good approximation of the training set, but the predictability can be very low and the results can be quite different for every realization. If alpha > 0, then training quality (correlation coefficient) will be less, if to use alpha=0 but predictability will be much higher (quality control with "blend well" test).

Kohonen NN regresion

With parameters:

Nearest neighbor number allow to define number of wells seismic traces to current trace with maximum similarity of the waveform

Neural network regression



Tikhonov alpha: 0 and <1000 allow to avoid overlearning effect or instability for the prediction.

Iteration: Maximum iterations in the course of learning stage

Hidden layers neurons: define structure of a neural network. If is necessary to use several hidden layers, comma is used: 10, 5, 3 mean 10 in the first, 5 in the second and 4 in the third hidden layers.

Learning algorithm:

Genetic – use genetic algorithm

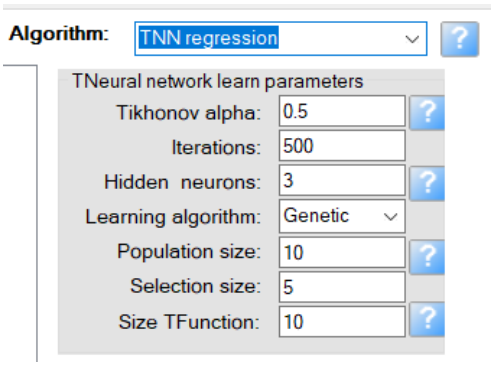
Gradient – use gradient algorithm

Hybrid – use genetic + gradient algorithms

Population size – is used only for genetic algorithm

Selection size – is used only for genetic algorithm

TNN regression



Tikhonov alpha: 0 and <1000 allow to avoid overlearning effect or instability for prediction.
Iteration: Maximum iterations during learning stage
Hidden layers neurons: define structure of neural network. If it is necessary to use several hidden layers, in this case comma is used: 10, 5, 3 mean 10 in the first, 5 in the second and 4 in the third hidden layers.
Learning algorithm:
Genetic – use genetic algorithm
Gradient – use gradient algorithm
Hybrid – use genetic + gradient algorithms
Population size – is used only for genetic algorithm
Selection size – is used only for genetic algorithm
Size TFunction – parameter for TNN >=5 and <=100

Seismic cubes: allow to select or unselect one or several cubes that will be used for calculations.

Radius inlines, Radius xlines, Radius samples: allow to define a moving window size around the well log points during this training stage and around seismic sample for calculation during calculation stage.

LFM option: allow to define low frequency option:

Without LFM - no low frequency model used.
With LFM - low frequency model will be creating from well logs to approximate stratigraphic depth from top to bottom surfaces included surfaces defined in **LFM surfaces Tab**
Only LFM - the result will be low frequency model.

The seismic cube can be limited by
Top surface or **constant**, both with **shifts**
Bottom surface or **constant**, both with shifts

To select well log necessary to use second tab with attributes

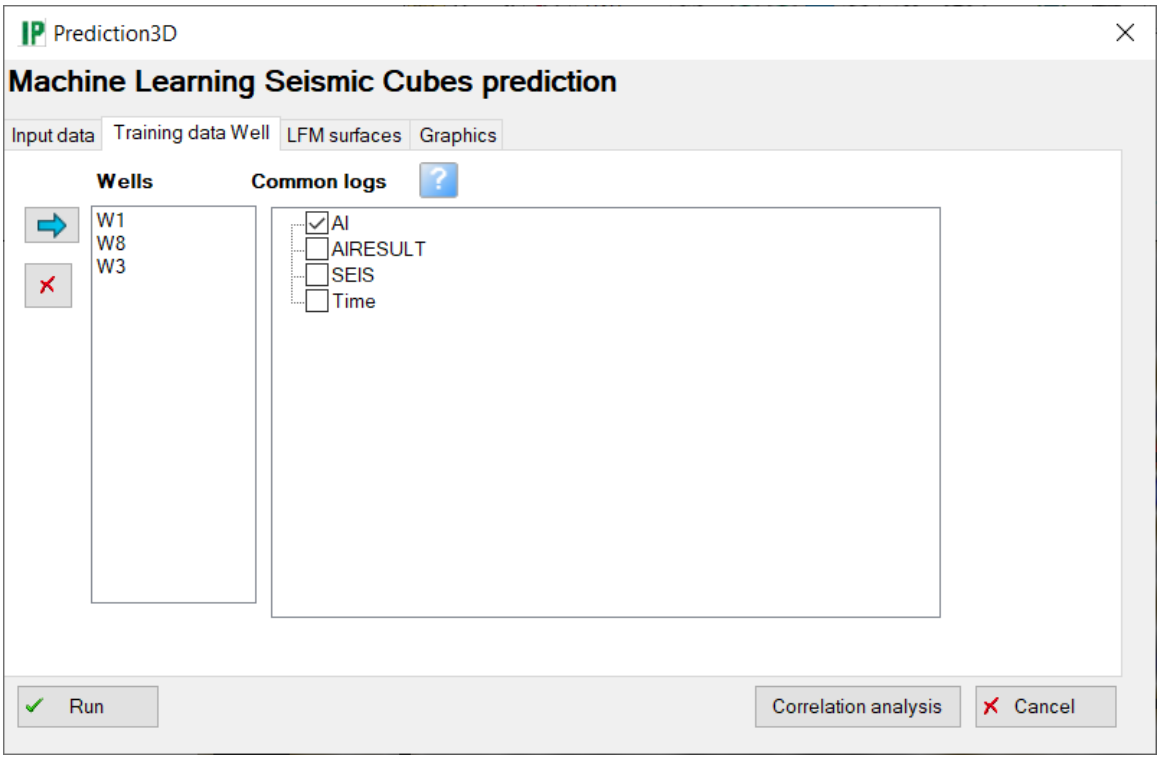


Figure 3: Input well log for training tab dialog

To use additional surfaces for low frequency model, need to define surfaces list in **LFM surfaces Tab** (see figure 4)

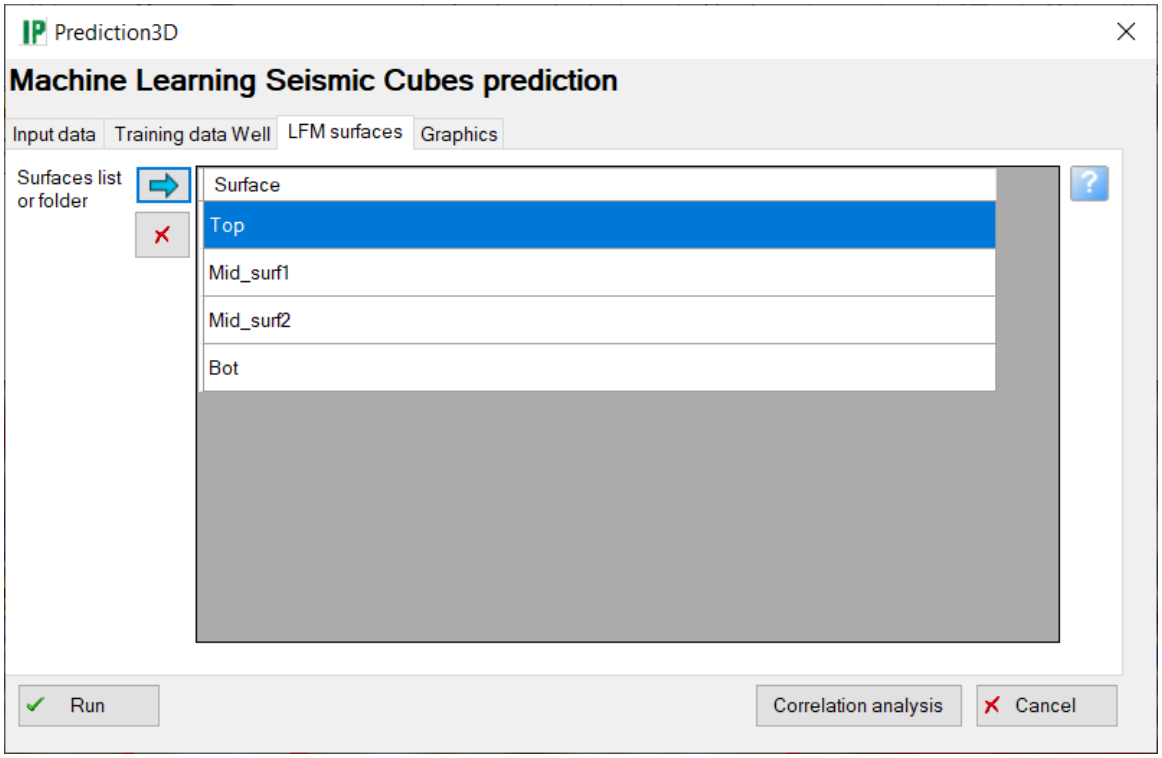


Figure 4: Surfaces defined to create low frequency model. It will be used only for LFM options – LFM Model and only LFM.

To perform correlation analysis, it is necessary to push "Correlation Analysis" bottom. After calculation you can see correlation table (see Figure 5) with cross correlation values and with Principal components (shift table to the right). Table is sortable for each column (click on 'column name'). Double click – sort from big to small. *The "Restore original sorting" button restores the original view.*

Figure 5: Correlation table results according to Correlation Analysis bottom

4. Calculation

To start prediction calculations, it is necessary to push "Calculate result" button (during training stage, error and correlation curves will be calculated in Graphics window (see Figure 6)

If to use ***Number realization*** >3, prediction results will be calculated several times according to this number (see Figure 7).

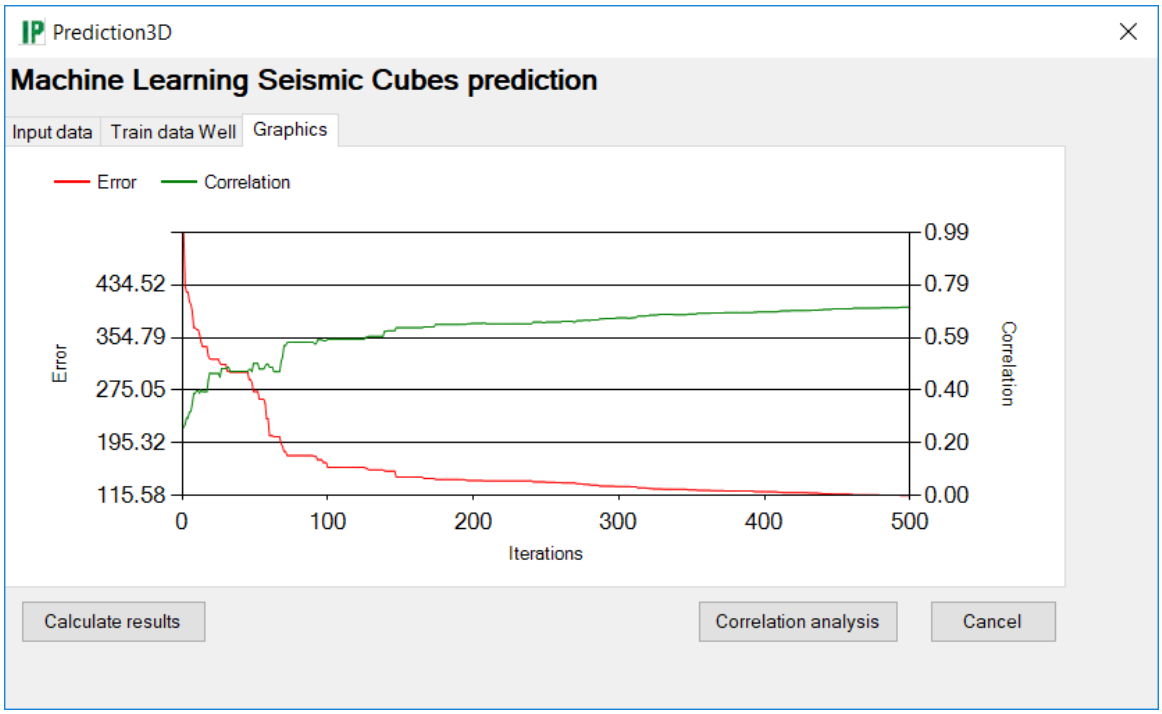


Figure 6: Output graphics window with error and correlation values during learning iterations (for number realization =1 and Cross validation =0)

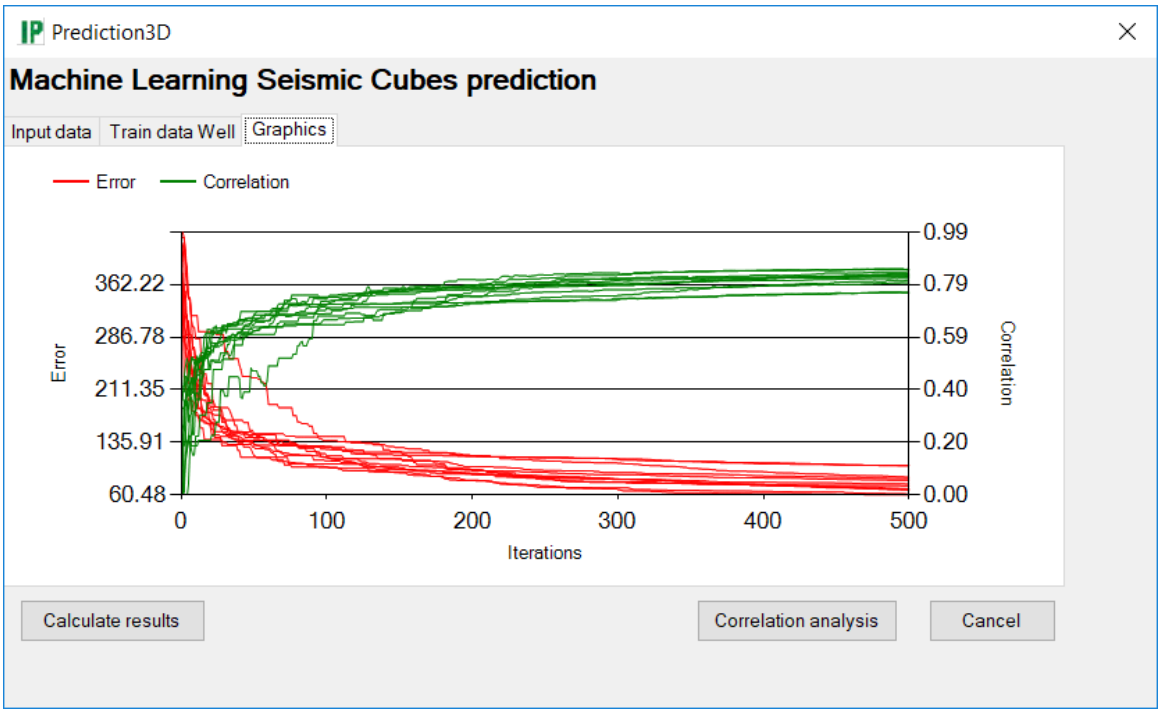


Figure 7: Output graphics window with error and correlation during learning iterations (10 wells)

5. Results

Predicted cubes will be added like virtual cubes according to ***Multi wells option.***

One result for all wells together – for all wells only one result – there is no cross validation.
Results is one virtual cube with name ***ML_<Algorithm>***

Separate result for every well – it is necessary to investigate all wells separately.
Results is several virtual cubes with names ***ML_<WellName>***

Average, standard, P10, P50, P90 – separate prediction is used for each sample in cube and calculate average, standard, P10, P50, P90 – the recommended option
Results is five virtual cubes with names ***ML_<Algorithm>_Avr, ML_<Algorithm>_Std, ML_<Algorithm>_P10, ML_<Algorithm>_P50, ML_<Algorithm>_P90***

Average with weights according to the distance to wells – calculate weighting average according to the prediction from each well and using inverse distance weights to the wells. It is not necessary to apply a separate procedure for well matching to obtain the results.
Results is two virtual cubes with names ***ML_<Algorithm>_Avr, ML_<Algorithm>_Std***

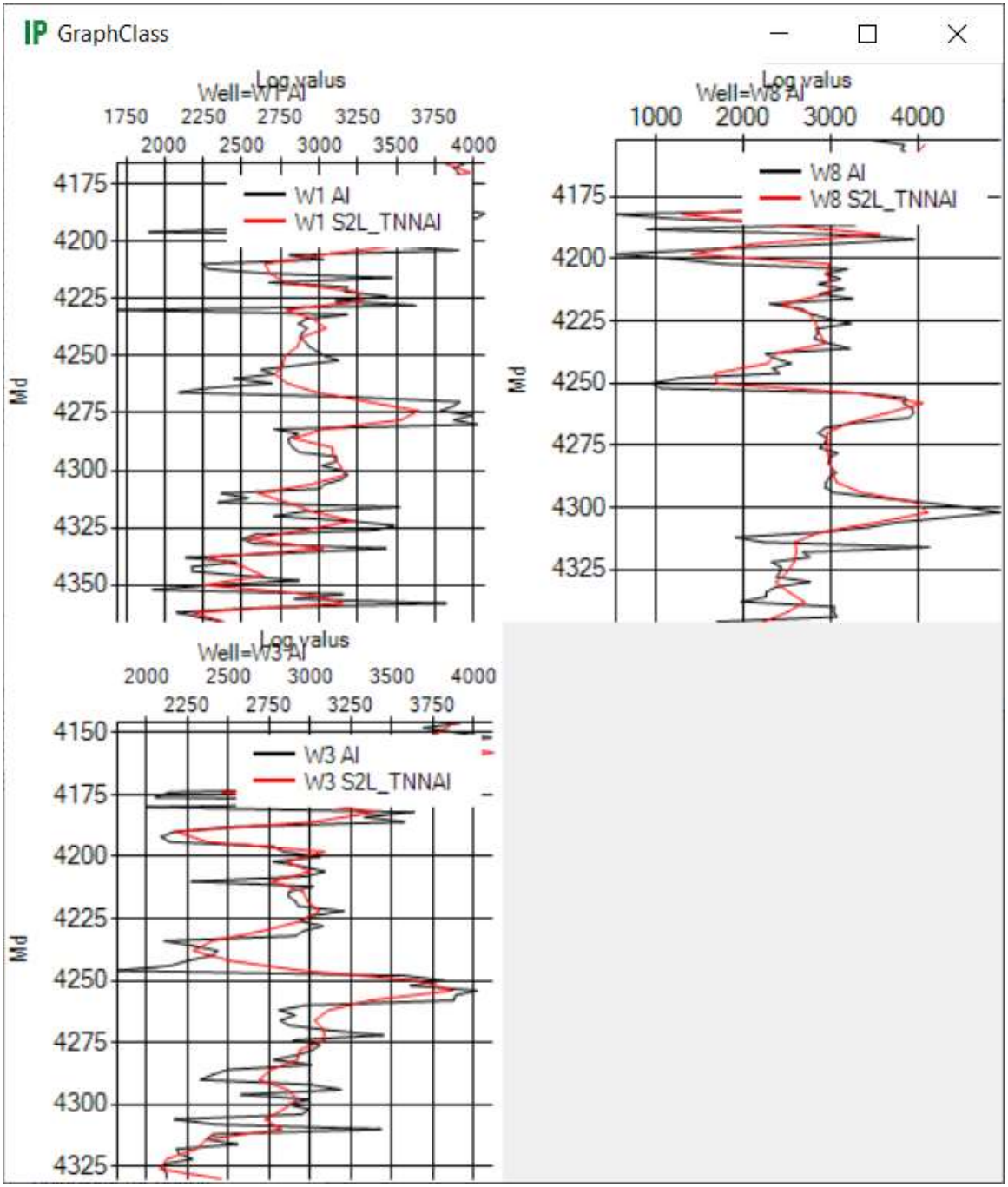


Figure 8: Well logs predicted and measured

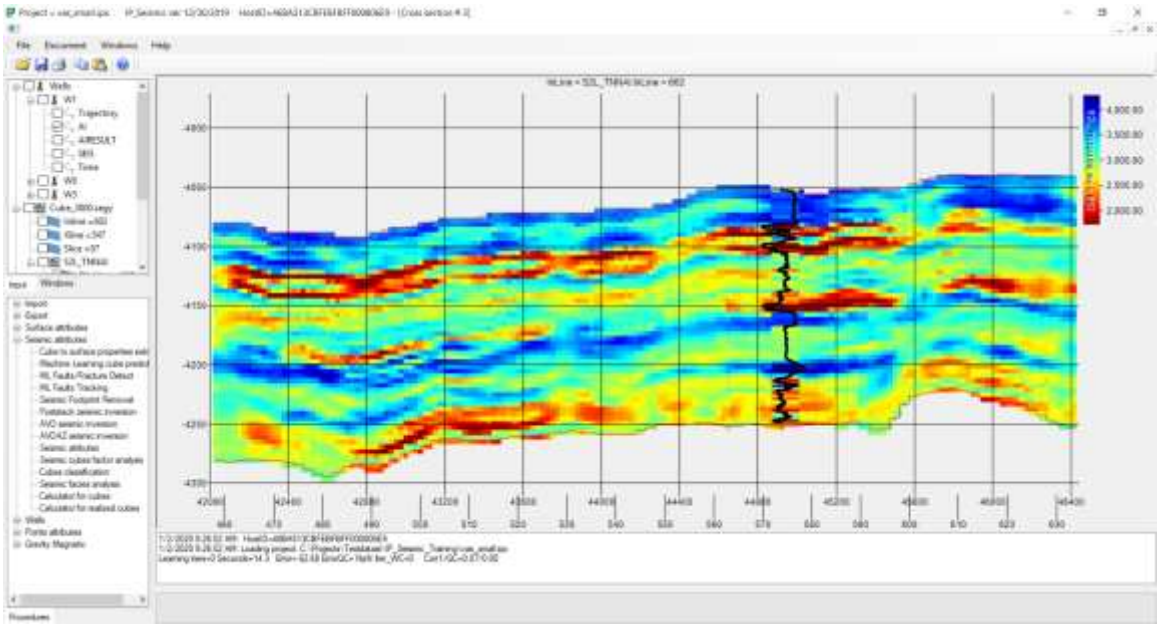


Figure 9: Virtual cube result