UNCONVENTIONAL RESOURCES TECHNOLOGY CONFERENCE
FUELED BY SPE • AAPG • SEG

URTeC: 2877021

Synthetic Well Log Generation Using Machine Learning Techniques

Oyewande Akinnikawe*, Stacey Lyne, Jon Roberts; Devon Energy Corp E&P.

Abstract

When using wireline log to characterize formation properties for an area we often run into incomplete datasets. One way to address this lack of data is to create synthetic curves to use in the analysis. This paper will cover workflows to generate synthetic photoelectric (PE) and unconfined compressive strength (UCS) logs. Modern well log data sets usually include PE logs which provides important information about the lithology of the formations the wellbore intersects. However, many legacy wells do not have PE logs that we need in order to understand the lithology of these formations. Similarly, it is important to obtain UCS data from mechanical failure tests done to core samples to determine the strength of the rock. Obtaining and testing core samples for the entire zones of interest is both expensive and time consuming. Synthetic well logs can be a reliable and cheaper alternative to predicting PE and UCS values rather than running a new set of logs or coring and testing the whole zone of interest.

In the first workflow, Synthetic PE logs were generated using wireline logs from over a hundred wells that included gamma ray, density, neutron, resistivity logs and volume of clay. The data was randomly partitioned into a 70:30 split for training and validation data set respectively. Model competition among a suite of machine learning algorithms such as Linear Regression, Artificial Neural Networks (ANNs), Decision Trees, Gradient Boosting and Random Forest was used to select the best algorithm based on the least average squared error (ASE) of the validation dataset.

In the second workflow, UCS data was generated using wireline logs and core rebound hammer data from fourteen wells including gamma ray, density, porosity, neutron, clay volume, kerogen volume, compressional slowness, shear slowness, Young's modulus (static and dynamic) and Poison's ratio. Variable clustering was used to remove collinearity, decrease variable redundancy, and choose the best variables for analysis. Cluster analysis was performed on the chosen variables to identify factors that differentiate data segments from the population. The data was randomly portioned into a 70:30 training and validation split and model competition amongst the suite of machine learning algorithms mentioned above was used to select the champion model based on the least ASE of the validation dataset.

Results show that neural networks and random forests generated the best prediction of UCS and synthetic PE logs compared to other machine learning algorithms used.

Introduction

When using wireline log to characterize formation properties for an area we often run into various problems with the data available. Often one of more curves is missing in some of the datasets we would like to use for the analysis. Regardless if the missing data is caused by old logs that did not include the measurement, tool failure, or looking outside the original area of interest. Going back to make these measurements is often infeasible after the original wireline logs, due to casing being present in the borehole or for economic reasons.

To address this problem several studies have been performed in which machine learning techniques were used to generate synthetic well logs. Some pertinent studies have use neural networks to predict reservoir properties from MWD and wireline logs, (Bhatt, 2002) predict density, resistivity and neutron using wells logs and location data, (Rolon *et al* 2005, Rolon *et al* 2009) create synthetic neutron logs using well log data, (Ghavami, 2011) create synthetic sonic logs from well log data, (Guan, 2012) and develop synthetic geomechanical logs (Eshkalak *et al.* 2013, 2014). These published studies show promising results with R^2 values between 0.7 and 0.95.

Additionally there has been research into the effects of data gaps. In a study, Lopes and Jorge (2017) generated random gaps with varying sizes from logs of a well with complete suite of bulk density, sonic, gamma ray, and neutron porosity log. Subsequently, they used well log data from over 1000 wells in the same region of the North Sea to predict the log values of the missing gaps they created in the single well using models such as Generalized Linear Models, Bayesian Regression, Random Sample Consensus, Random Forest, and Artificial Neural Network. Results shows that the performance of the algorithms depends on the gaps size and on the gap itself. Artificial neural networks perform better for larger gap sizes while the other models perform better for smaller gap sizes.

In this study, we perform a descriptive and predictive analysis on generating synthetic photoelectric (PE) and unconfined compressive strength (UCS) logs. The training data for the Synthetic PE log contained wireline logs from over a hundred wells that included gamma ray, density porosity, neutron porosity, deep resistivity, photo electric logs and volume of clay as input data. Similarly, UCS data was generated using wireline logs and core rebound hammer data from fourteen wells including gamma ray, density, porosity, neutron, clay volume, kerogen volume, formation tops, compressional slowness, shear slowness, Young's modulus (static and dynamic) and Poison's ratio. Variable clustering was used to remove collinearity, decrease variable redundancy, and choose the best variables for analysis. In both cases, the input data was randomly partitioned into a 70:30 split for training and validation data set respectively before passing it into machine learning algorithms such as Linear Regression, Artificial Neural Networks (ANNs), Decision Trees, Gradient Boosting and Random Forest. A champion algorithm is selected based on the least average squared error (ASE) of the validation dataset.

Methodology

In order to create synthetic PE and UCS logs from other wireline logs, the methodology used include five steps shown below. A detailed description of each step is explained afterwards:

1. Data Preparation
2. Data Transformation
3. Data Partitioning
4. Modeling
5. Model Validation

Data Preparation: Datasets stored in different folders or databases often have different formats and need to be merged. Also, data is often created with errors, missing values or other inaccuracies. Inconsistent, low quality data can generate errors that make analytics and data mining slow and unreliable. Excellent quality data are effortlessly processed and analyzed, leading to insights that helps organization make faster and better decisions. The process of correcting inaccuracies, performing verification and joining data sets constitutes a big part of the data preparation process. Data preparation is the first and most important step in creating a predictive model. Data preparation is the process of

gathering, cleaning, and consolidating data into one file or data table for analysis. Using prepared data leads to faster and better models.

Data Transformation: The input dataset contain attributes with varying scales where quantities such as Gamma Ray and Deep Resistivity have values ranging from 0 – 800 API and 0 – 2000 ohmm respectively. While other attributes like Density and Neutron Porosity have quantities that vary from -0.1 – 0.3. Many machine learning methods like the data attributes to have the same scale such as between 0 and 1 for the smallest and largest value for a given feature. Also, some machine learning algorithms like the data to be normally (Gaussian) distributed. Data transformation is the process of converting sets of data values from a source format to a format consistent with the model requirements. Some common data transformation used to process data before modeling include log, square root, square, and standard normal.

Data Partitioning: Data partitioning is the process of randomly separating your data into two or more segments. It is common to separate the data into two sets called the training and the validation (hold-out) data set. The training set is passed into the machine learning model to find a suitable model for the training data while the validation data which has not be seen by the machine learning model is used to measure how well the model predicts the output variable based on the comparing errors of both the training and validation data. The split ratio between the training and validation data is up to the analyst. In this study a 70:30 split for the training and validation data respectively is used.

Modeling: machine learning models are used to determine patterns in the relationship between the input variable and the target variable. These patterns are used to predict what the future target values will be when given a new set of input variables. The machine learning models used in this study are Linear Regression, Decision Trees, Gradient Boosting, Random Forest, and Artificial Neural Networks (ANNs). The training data is passed through each of the machine learning models and each of the models generate a corresponding model fit.

Linear Regression: A linear regression model assumes that there is a linear relationship between input variables (x) and one output variable (y). Linear models are simple and often provide an adequate and interpretable description of

$$f(X) = \beta_0 + \sum_{j=1}^{p} X_j \beta_j$$

how the inputs affect the output (Hastie et al, 2009). Linear regression has the form:

Here β_0 is the intercept term, β_j's are the unknown coefficients, and the variables X_j can be quantitative inputs, transformation of quantitative inputs such as log, square root or power, or interactions between input variables.

The most popular estimation method for linear regression is least squares method. Least square estimates often have low prediction accuracy because of their low bias but high variance. Also, large input variables leads to large number of effects which causes interpretation issues if all effects are used. A better model will be one that selects a smaller subset of the strongest effects and gives the optimum combination between bias and variance i.e. sacrifice bias but reduces the variance of the predicted values. The linear model selected is the stepwise selection model which starts with no effects and sequentially adds effects until a stay significance level or stop criteria is met. Least square regression is used to estimate the coefficients of the inputs that are retained.

Decision Trees: Decision trees can be used for regression or classification predictive modeling problems. The models are obtained by recursively separating the feature space and fitting a simple prediction model within each section. As a result, the separation can be represented graphically as a decision tree. To grow a regression tree, the algorithm needs to decide on the splitting variables and split points, and also what shape the tree should have. To find the best partition, the algorithm splits the parent node by exhaustively searching over all the input variable and split point that minimizes the total impurity of its child nodes. The process is applied recursively on the data in each child node. Splitting stops if the relative decrease in impurity is below a pre-specified threshold. Decision trees are not ideal for predictive

learning due to their low accuracy. Boosting decision trees improves their accuracy while maintain most of their desirable properties.

Gradient Boosting: Boosting is an ensemble method that combines the output of many "weak" classifiers to create a powerful "committee" strong classifier (Hastie et al, 2009). Gradient boosting does not require feature scaling, and are able to capture non-linearities and feature interactions. Gradient boosting iteratively trains a sequence of decision trees in order to minimize a loss function. This is done by building a model from the training data, then creating a second model that attempts to correct the errors from first model. With each iteration, gradient boosting further reduce this loss function on the training dataset. Models are added until the training set is predicted perfectly or a maximum number of models are added (Brownlee, 2016).

Random Forest: Another algorithm for learning ensembles of trees similar to gradient boosting but with a different training process is random forest. Random forests are one of the most successful machine learning models for classification and regression. They combine many decision trees in order to reduce the risk of overfitting. They do not require feature scaling, and are able to capture non-linearities and feature interactions. Random forests train a set of decision trees separately, so the training can be done in parallel. The algorithm injects randomness into the training process by subsampling the original dataset on each iteration to get a different training set (bootstrapping) and considering different random subsets of features to split on at each tree node. This makes each decision tree a bit different. To make a prediction on a new instance, a random forest must aggregate the predictions from its set of decision trees. For regression, the target variable is predicted to be the average of the tree predictions. Combining the predictions from each tree reduces the variance of the predictions, improving the performance on test data.

Artificial Neural Networks: Artificial Neural Networks (ANNs) were original developed by researchers who were trying to mimic the way the human brain functions. The researchers hoped to produce complex phenomena such as intelligence by combining many simple computing elements called neurons or nodes into a highly interconnected system. Neural networks are especially useful for prediction problems where no mathematical formula is known that relates inputs to outputs, prediction is more important than explanation, and there is a lot of training data. ANNs architecture consists of three layers. An input layer, an output layer, and any layer(s) between input and output layer called hidden layers. ANN is a two-stage regression or classification model. In the first stage called the forward pass, derived features are created from linear combinations of the weighted inputs from the input layer to the output layer through the hidden layers. An activation function such as sigmoid, ReLU, or hyperbolic tangent is applied to each neuron in the hidden layers to add non linearities to the network. An output function allows a final transformation of the output vector. For regression we typically choose the identity function while for classification we use the softmax function.

In the second stage called the backward pass, the error in the output layer is propagated backwards through the network and the weights of each neuron is adjusted. The two stages are repeated until the error in the network is minimized.

Model Validation: Once the training data has been passed through the various machine learning algorithms listed above and each algorithm has created a model to predict the target variables, the validation dataset is passed through each models and the error between the actual and predicted value is calculated. In selecting the best (champion) model

$$\frac{1}{n}\sum_{i=1}^{n}(Actual - Predicted)^2$$

we used a statistical estimate called the average squared error (ASE). The average squared error takes the square of the differences between the actual and predicted value of each observation, sums it and divides by the number of observation. It can be defined mathematically below:

The model with the lowest ASE was selected as the best model.

Case 1: Synthetic PE Generation

For the synthetic PE generation, a folder containing over 100 well log files in LAS format was required to be prepared. Each file had preselected variables necessary for the analysis such as unique well identifier (UWI), Depth, corrected gamma ray (GRC), Deep Resistivity (ILD), Compensated Neutron (PCNLS), Density Porosity (PDLS), PhotoElectric (PE) and Volume of Clay (VClay). All the LAS files were converted into a dataset of one CSV file and rows with missing data points were removed. A new column (DPNPD) was created for the difference between PCNLS and PDLS. Statistical analysis such as correlation and variable worth analysis were performed on the dataset to rank the variables in order of importance. Table 1 shows a statistical description such as the mean, standard deviation, min, median and max of the variable in the prepared dataset.

Table 1: Statistical Description of Synthetic PE Model Input Data

	DEPTH	DPNPD	GRC	PCNLS	PDLS	PE	ILD	VCLAY
count	309122.00	309122.00	309122.00	309122.00	309122.00	309122.00	309122.00	309122.00
mean	11273.16	0.00	44.60	0.04	0.04	3.70	763.83	0.05
std	2195.57	0.05	51.99	0.05	0.04	1.02	2081.95	0.09
min	7157.00	-0.60	1.99	-0.03	-0.18	0.57	0.27	0.00
25%	9544.50	-0.02	16.93	0.01	0.01	3.05	103.38	0.00
50%	11236.50	-0.00	27.23	0.02	0.03	3.58	278.41	0.02
75%	12698.50	0.02	51.73	0.04	0.05	4.22	670.30	0.06
max	16847.00	0.46	820.45	0.41	0.73	12.65	250564.16	0.99

It is also necessary to visualize the relationship between two input variables or an input variable and the target variable. This can be done by creating a matrix plot in which the numerical variables (input and output) are plotted on the rows and columns. Such a plot is called a scatter plot matrix or a pair plot. Fig 1 shows a scatter plot matrix of some input and target variable.

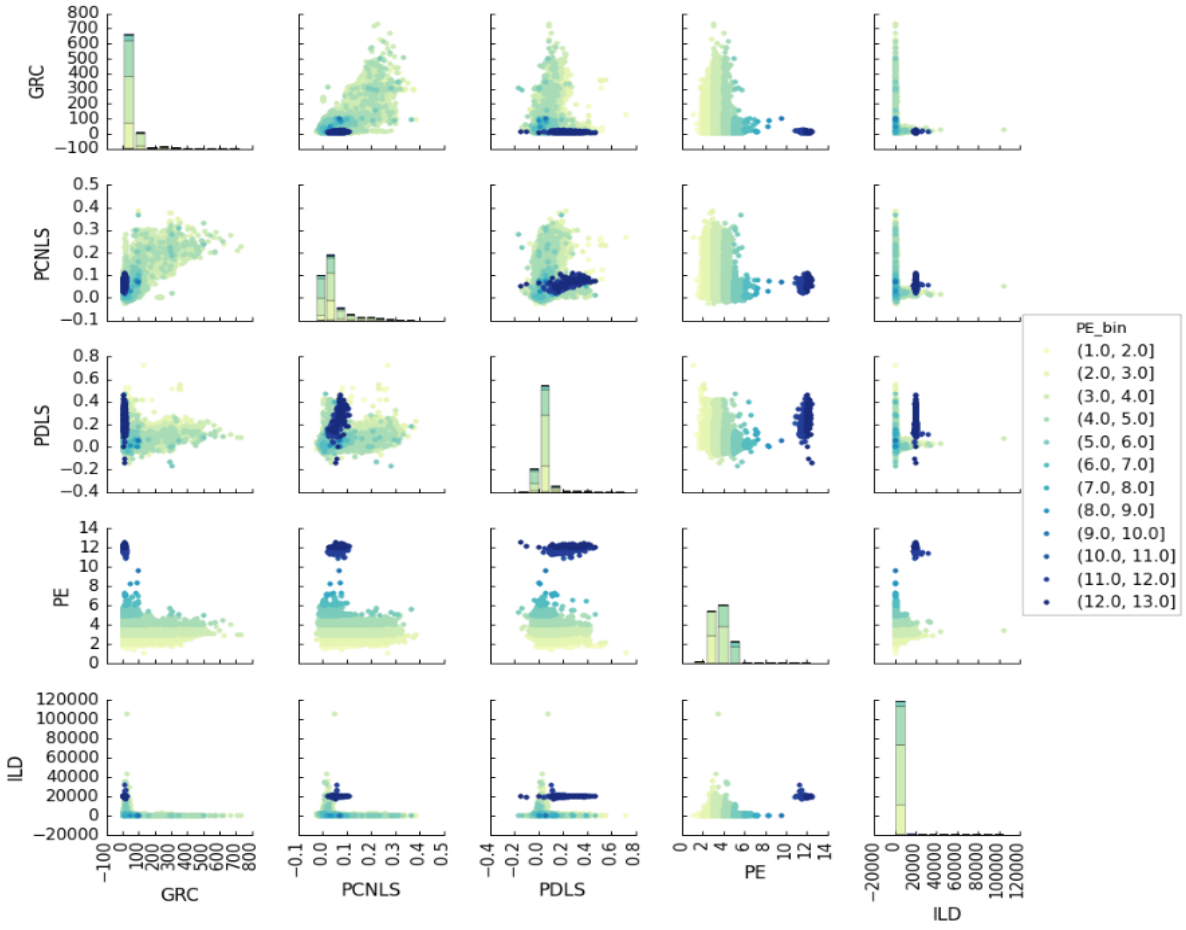

Figure 1: Pair plot for Synthetic PE

From the pair plot and the histogram (on the diagonals of the pair plot) it can be seen that some variables contained outliers that would negatively affect model prediction. Therefore, the outliers in the input variables were removed by performing a filter operation such that values of deep resistivity (ILD) ranges from 0 – 2000 ohmm, neutron porosity (PCNLS) and density porosity (PDLS) from -0.1 – 0.3 and photoelectric log from 1.7 – 5.1 b/e. Although the histogram for GRC is right skewed and might suggest some outliers, we did not perform a filter operation on GRC because we expected some high GRC values in this formation. Fig 2 shows the histogram of the input variables before and after applying the filter. It is necessary to determine which variables are most important in predicting the target variable. Fig 3 shows the variable worth plot of the input variables. The variable worth plot ranks input variables according to how best they can estimate the target variable using the Gini split worth statistics.

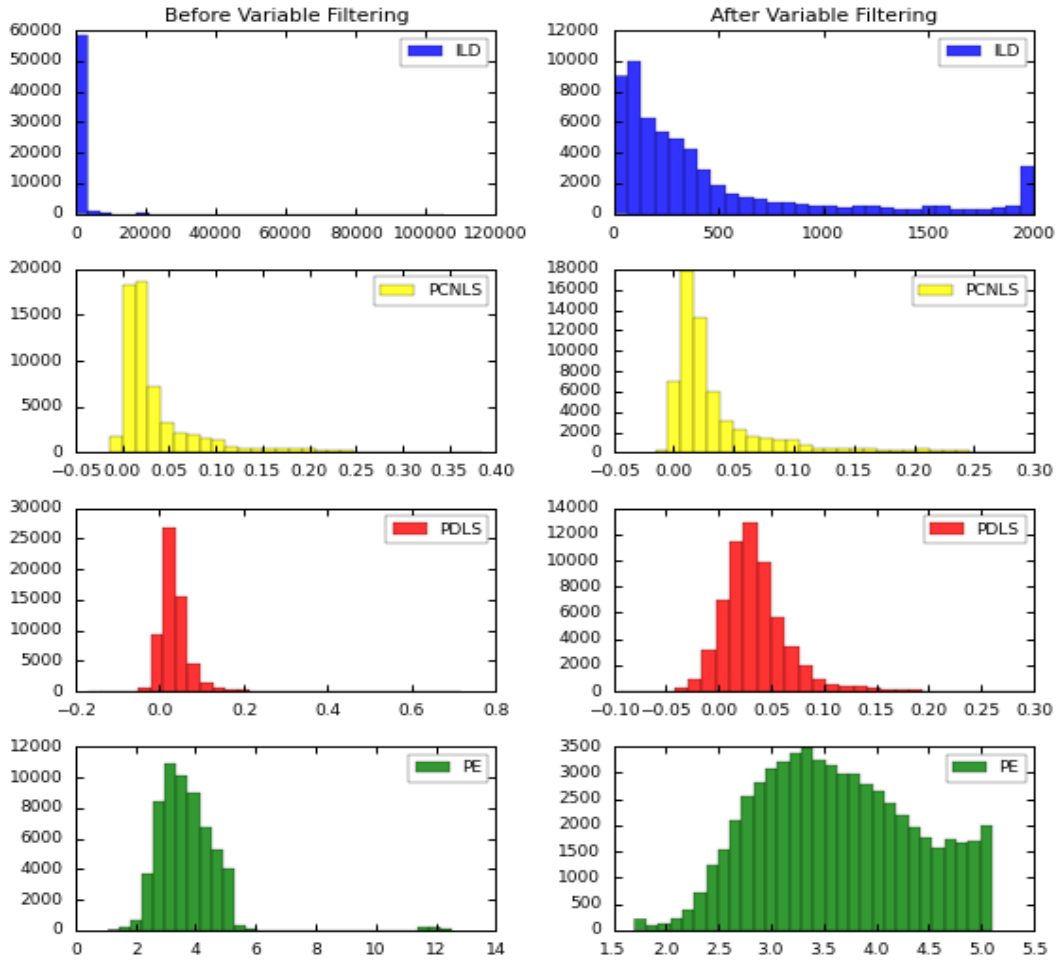

Figure 2: Variable Filtering

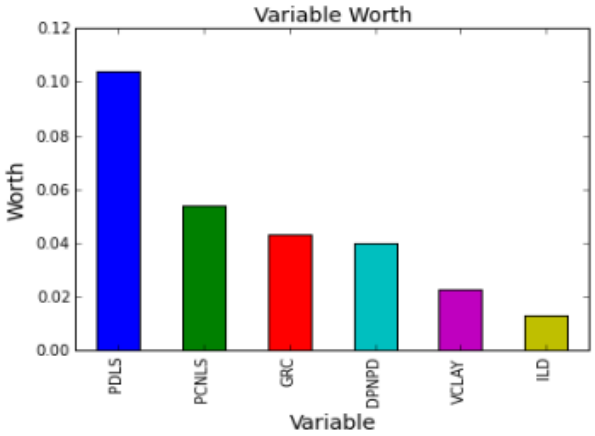

Figure 3: Variable worth plot of Target Variable (PE).

It is often necessary to find out if there is a correlation between any of the input variable with the target variable. Pearson's correlation is a measure of the linear correlation between the input variables and the target variable. Fig 4 shows a correlation plot of the input variables.

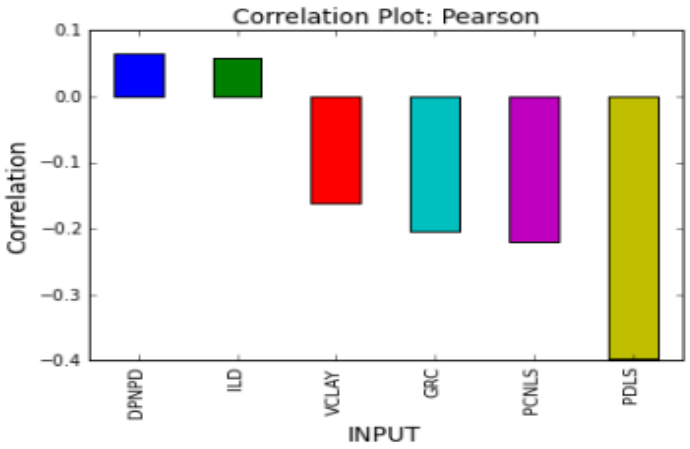

Figure 4: Correlation plot of input variables with target variable (PE)

Variable transformation is performed on the input variables such that the scales of the quantities for the input variables is within a reasonable range so that the machine learning models are not biased by the input dataset. Fig 5 shows the log transformation of GRC and ILD. The histogram for the gamma ray values have been transformed from right skewed to approximately normal while that for the deep resistivity is multi modal.

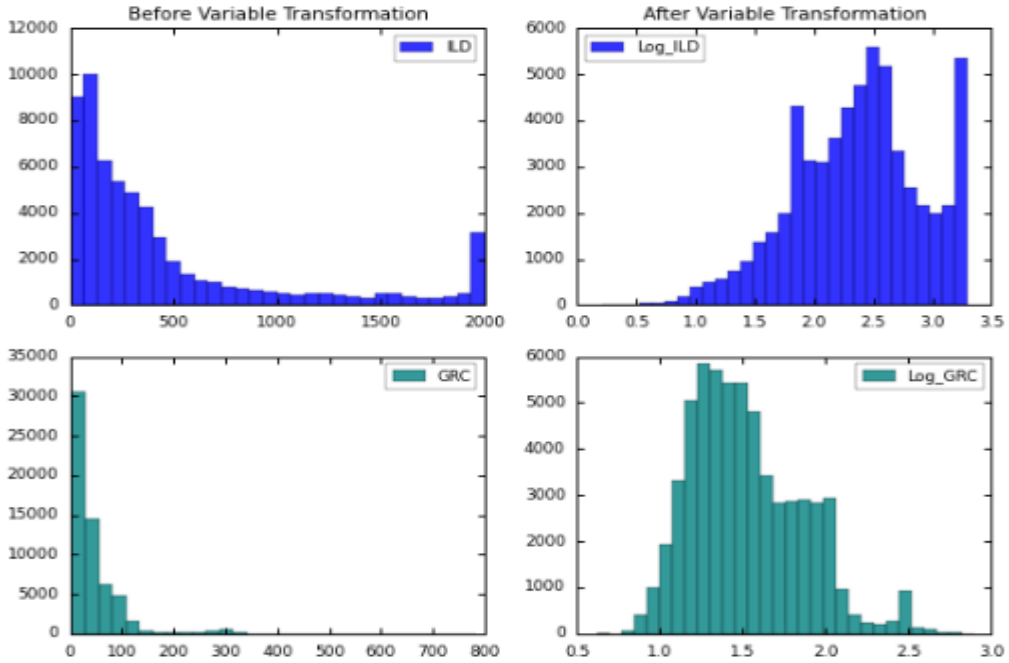

Figure 5: Variable transformation for deep resistivity (Top) and corrected gamma ray (Bottom)

The workflow used to model synthetic PE prediction is shown in Figure 6.

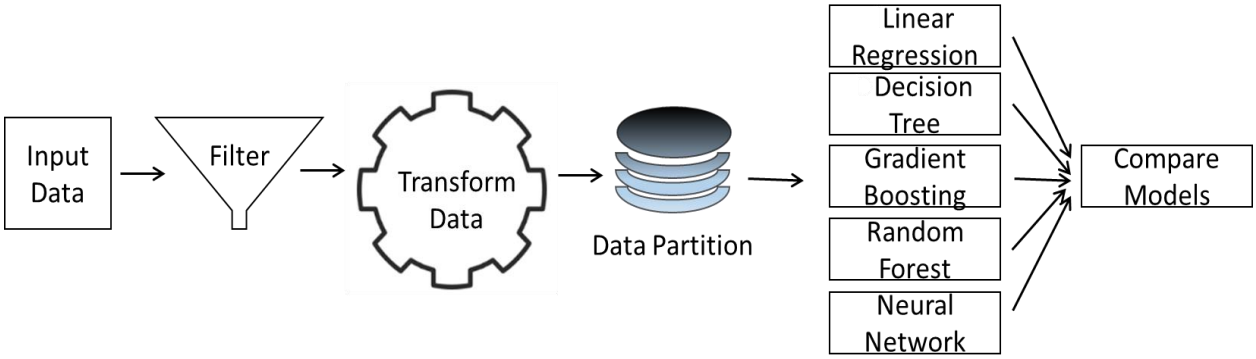

Figure 6: Workflow for synthetic PE Prediction

To select the champion model, the average squared error (ASE) of the validation data set of the competing models are compared and the one with the least ASE is selected. The comparison of the various model's ASE is shown in Table 2.

Table 2: Model Comparison for Synthetic PE prediction

Model Description	Training Average Squared Error	Validation Average Squared Error
Linear Regression	0.392	0.395
Decision Tree	0.330	0.349
Gradient Boosting	0.338	0.348
Random Forest	0.220	0.330
Neural Network	0.377	0.380

Case 2: Rock Strength Prediction

For the UCS curve generation the triple combo log data was run through the same quality control steps as the synthetic PE data. The log data for the UCS calculation included a dipole sonic in addition to the triple combo data. This data was also quality controlled, and used to calculate an isotropic poisons ratio and dynamic Young's modulus. The static Young's Modulus was then calculated using a correlation. Next a petrophysical analysis was performed on the data to generate component volumes.

In addition to the log data there was also core based measurements of UCS over the intervals where core data was available. There were two types of core data available, Triaxial core plug measurements, and rebound hammer data. We chose to use the rebound hammer, UCS correlation data for our target variable because there was a much larger and more statically significant number of test then was available with the Triaxial measurements. The rebound hammer measurements were then shifted to log depth for use in the analysis.

For the rock strength modeling, a file containing over 20 petrophysical parameters including density porosity (PDLS), effective porosity (PHIE), photoelectric (PE), corrected gamma ray (GRC), neutron porosity (PCNLS), and dynamic Young's Modulus (YME_DYN) from 14 wells in CSV format was provided. The target variable we want to model is

UCS, therefore the file was cleaned up by removing depths without measured UCS values, null values, and wells that didn't have parameters that were deemed necessary for predicting UCS. Statistical analysis such as correlation and variable worth analysis were performed on the dataset to rank the variables in order of importance. Table 3 shows a statistical description such as the mean, standard deviation, min, median and max of some of the variables in the cleaned dataset.

Table 3: Statistic description for Rock Strength Modeling

	PDLS	PHIE	PE	GRC	PCNLS	YME_DYN
count	4764.00	4764.00	4764.00	4764.00	4764.00	4764.00
mean	0.09	0.07	-155.70	84.78	0.13	6.24
std	0.04	0.03	366.11	29.43	0.05	1.71
min	-0.10	0.00	-999.25	13.43	0.01	3.15
25%	0.06	0.05	2.43	66.59	0.09	4.86
50%	0.09	0.07	2.83	84.78	0.12	6.05
75%	0.12	0.09	3.46	105.59	0.17	7.29
max	0.25	0.14	6.06	283.82	0.36	11.15

Large numbers of variables can complicate the task of establishing the relationships that might exist between the independent variables and the target variable in a model. Limiting the input variable helps reduce computing time, model interpretation and model stability. A variable clustering analysis was performed to remove collinearity, redundancy, and help reveal the underlying structure of the input variables in the data set. Variable clustering divides numeric variables into cluster of correlated variables. That is, the variables which provide the same kind of information belong to the same group. The variation proportion threshold was set to 95% and the variable with the highest variance explained in each group is selected as the independent variable to be used in the modeling exercise. Fig 7 shows a dendrogram of the input variables and the proportion of variance explained.

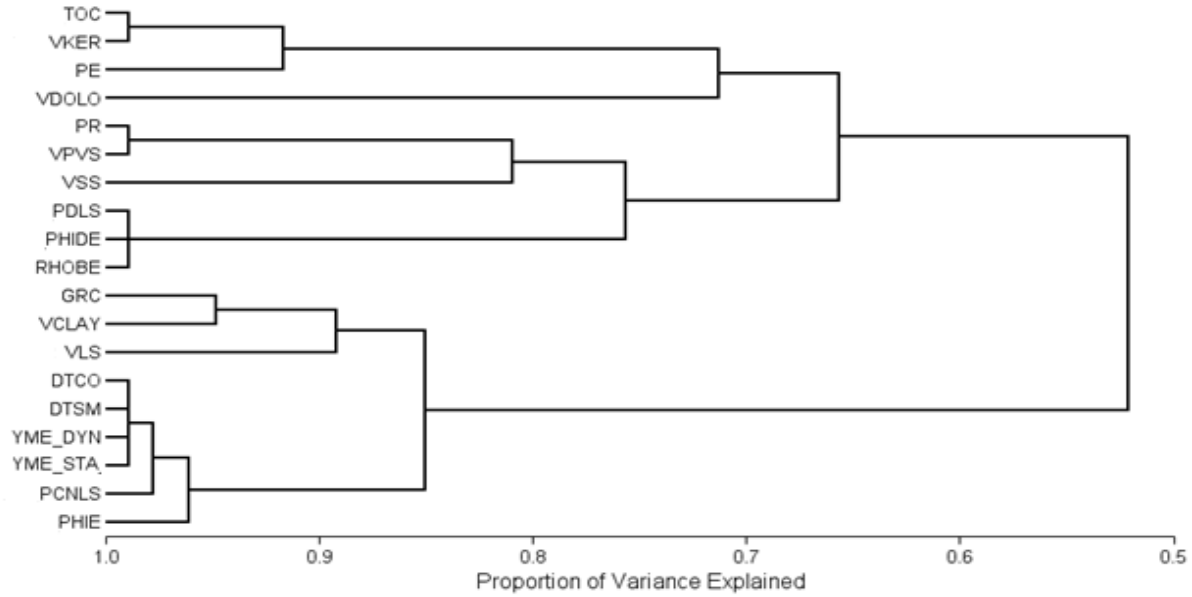

Figure 7: Dendrogram showing hierarchical clustering of rock strength prediction input variables

The 19 input variables shown on the vertical axis of the dendrogram in Figure 7 were divided into 12 clusters based on the 95% variable proportion threshold. A cluster network plot delineating the variable clusters is shown in Figure 8.

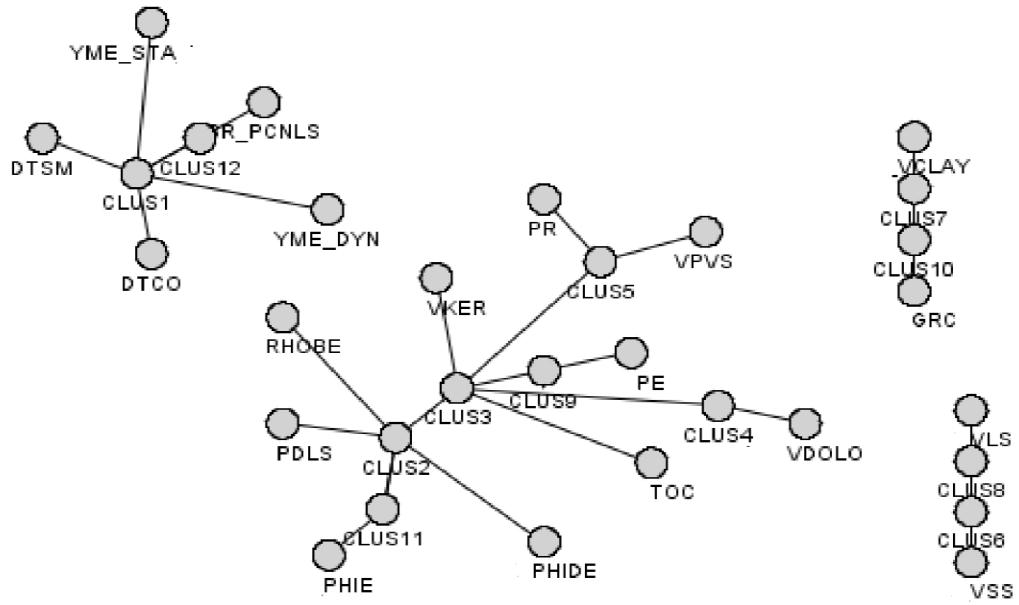

Figure 8: Cluster network plot for rock strength prediction

Clusters that are well separated have low R-squared values, and groups that have high R squared values with next cluster component are closer in the cluster network plot shown above.

After selecting the independent variables to be used for UCS prediction, a variable transformation was performed to scale the quantities of the input variables within a reasonable range (usually between 0 and 1) to enhance the performance of the machine learning models. Figure 9 shows the transformation performed on dynamic Young Modulus and the target UCS (measured). It should be noted that a back transform must be done after prediction if any transformation is performed on the target variable during modeling.

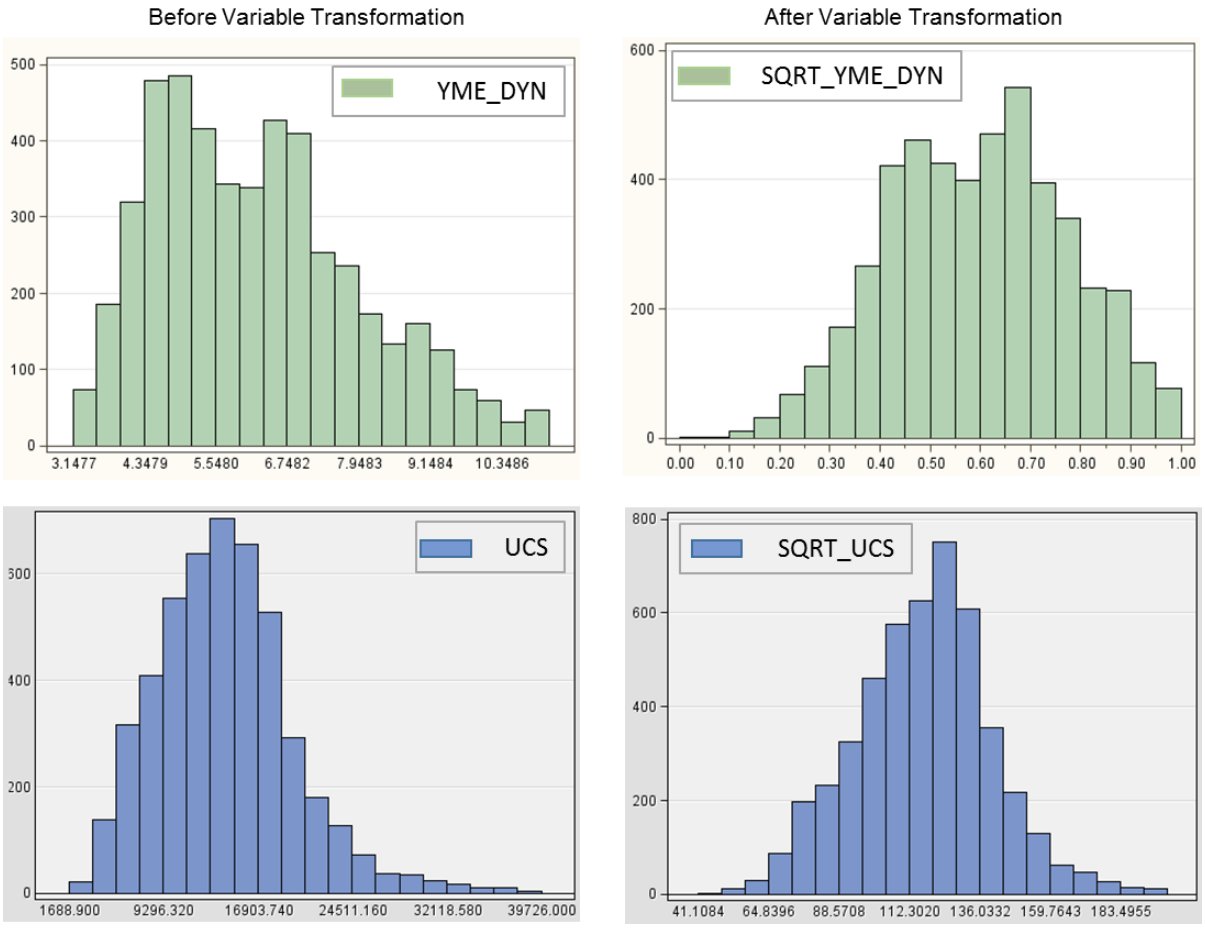

Figure 9: Variable transformation for dynamic Young's Modulus (Top) and Unconfined Compressive Strength (Bottom)

Due to rock type, it is expected that rock strength will vary from formation to formation. Once the input variables have been transformed to achieve normal distribution and standard scaling, a nominal variable (Formation) was added to the input variables and a cluster analysis was performed. Clustering splits the dataset into groups suggested by the data. The dataset in each cluster tend to be related to each other while those in different clusters are dissimilar. The goal here is to identify factors that differentiate data segments from the population. The workflow used for rock strength modeling is shown in Figure 10.

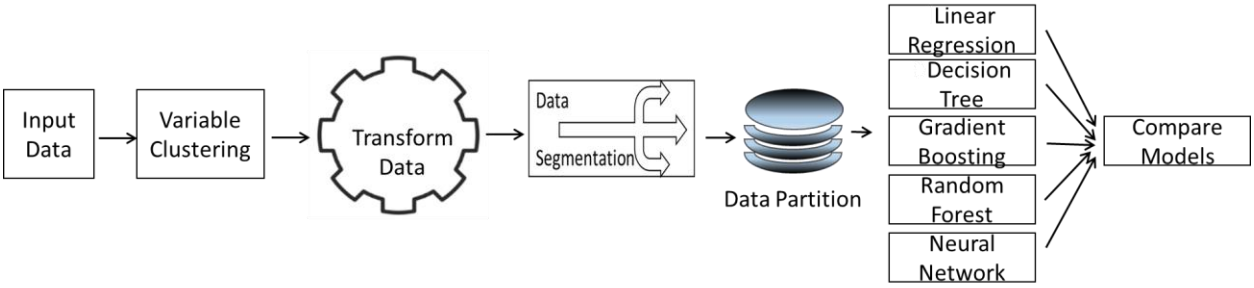

Figure 10: workflow for rock strength modeling

To select the champion model, the average squared error (ASE) of the validation data set of the competing models are compared and the one with the least ASE is selected. The comparison of the various model's ASE is shown in Table 4.

Table 4: Model comparison for UCS prediction

Model Description	Training Average Squared Error	Validation Average Squared Error
Linear Regression	323.4	332.9
Decision Tree	327.2	359.3
Gradient Boosting	341.9	356.5
Random Forest	290.8	338.1
Neural Network	309.2	320.2

Results and Discussion

Recall that the Training data set consists of 70% of the input data that was randomly chosen. While the validation data set contains the other 30% of the input data. The training data is used to create the model while the validation data is used to test the model. For synthetic PE prediction, it can be seen from Table 2 that the model with the least Training ASE is random forest with a value of 0.22. Also, the random forest model has the lowest validation ASE of 0.33. Therefore, the random forest model was chosen as the best model. Figure 11 shows the comparison between the wireline log measured PE and the random forest predicted PE. Overall there is a good overlay between the measured PE and the predicted PE.

For UCS (rock strength) prediction, it can be seen from Table 4 that random forest has the lowest training ASE value of 290.8 while neural network has the lowest validation ASE of 320.2. Therefore, the neural network model was chosen as the best model. It should be noted that the ASE values in Table 2 are much lower than those in Table 4 because the range of measurement of the target variable (PE) in Table 2 is between 1.7 – 5.1 b/e and the number of observation is about 290,000 while the range of measurement of the target variable (UCS) in Table 4 is between 1700 – 40000 psi and the number of observation is 4764. Several network configuration of the neural network was tried to see which one would give the lowest validation ASE. Ultimately the chosen network consist of 3 hidden layers with 10 neurons in the first hidden layer, 5 neurons in the second hidden layer and 3 neurons in the third hidden layer. Each neuron has a hyperbolic tangent activation function and the output (target) layer has an identity activation function.

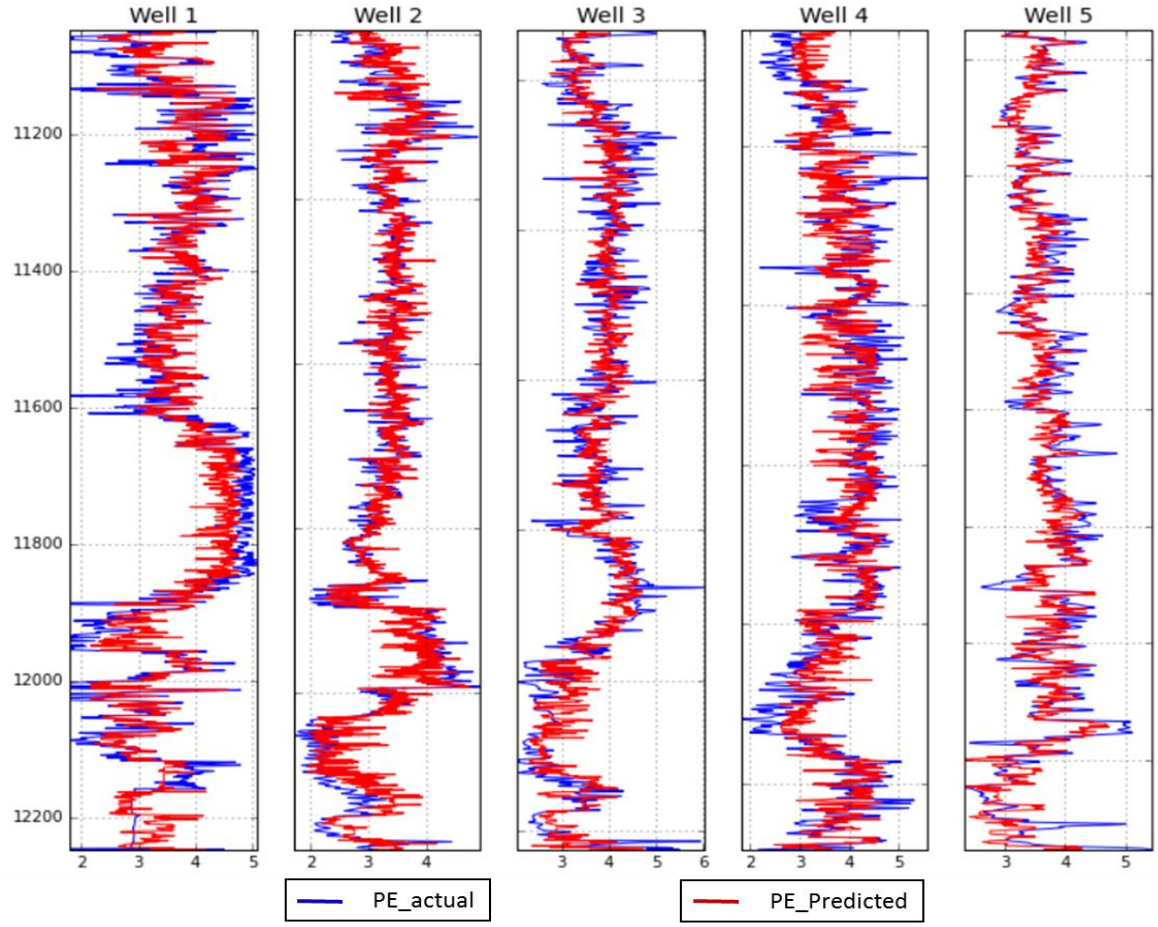

Figure 11: Comparison between actual PE (blue) and predicted PE (red)

Overall the results of the UCS correlation matched fairly well when compared with the rebound hammer data. When viewing the data in log view (Fig 12) it appears that the predicted UCS intercepts the mean value of the rebound hammer data, and is a pretty good fit considering the resolution differences of the measurements.

The predicted UCS from the neural network was also correlated very well with the log based UCS correlation that had been tuned for each formation in this area.

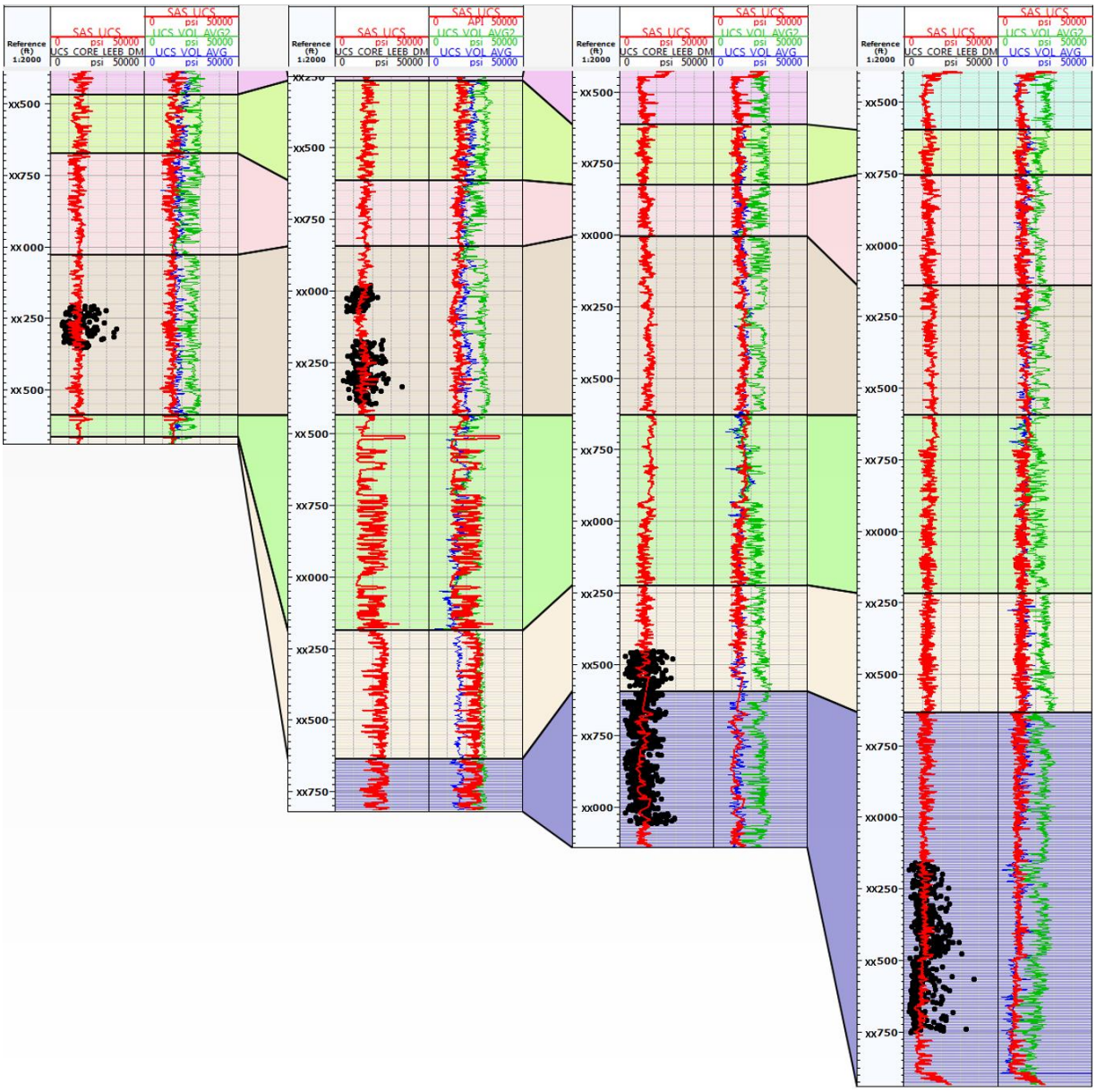

Figure 12: This is a log plot of 4 wells with track 1 containing the rebound hammer data (black) and the UCS created from the Neural network (Red). Track two contains the UCS from the neural network (Red), with two different log based UCS correlations (Blue and Green). The background colors present on all of the track represent different formations.

Summary and Conclusions

This study presents workflows for predicted log and core data from other log data. The first workflow consists of gathering, cleaning, and consolidating the input data, filtering the data to remove noisy PE measurement, transforming the input data to achieve standard normal distribution, partitioning the data into a training and validation dataset, creating a suite of models with the training data and selecting the best models based on the validation data with the least ASE.

The second workflow consists preparing the input data, performing variable clustering analysis to remove collinearity, remove redundant variables and reduce the amount of input variable. The data was transformed such that the input data had a reasonable scale near 0 and 1 and are normally distributed. Cluster analysis was performed to identify factors that differentiate data segments from the population and then the data was partitioned into training and validation data using a 70:30 split respectively. Similar to the first workflow, a suite of machine learning models were created and the best model was selected based on the validation ASE.

Overall the result from the first workflow showed that for this case study, the random forest model was the best model that can accurately predict the missing gaps in PE curves or an entire PE curve from well log data. The result from the second workflow shows that the neural network model gave a reasonable prediction of UCS data from well log data.

References

Bhatt, A. (2002). Reservoir properties from well logs using neural networks. PhD. Dissertation, Department of Petroleum Engineering and Applied Geophysics. Norwegian University of Science and Technology.

Brownlee, J (2016). Master Machine Learning Algorithms: Discover how They Work and Implement Them From Scratch. https://books.google.com/books?id=PdZBnQAACAAJ

Eshkalak, M.O., Mohaghebh, S.D., and Esmaili, S. (2013). Synthetic, Geomechanical Logs for Marcellus Shale. SPE paper 163690-MS presented at Digital Energy Conference and Exhibition, Woodland, Texas, 5 – 7 March, 2013. Doi: http://dx.doi.org/10.2118/163690-MS

Eshkalak, M.O., Mohaghegh, S.D., and Esmaili, S. (2014). Geomechanical Properties of Unconventional Shale Reservoirs. Journal of Petroleum Engineering, 2014. Doi: http://dx.doi.org/10.1155/2014/961641

Ghavami, F. (2011). Developing Synthetic Logs Using Artificial Neural Network: Application to Knox County in Kentucky. M.Sc. Thesis, Department of Petroleum and Natural Gas Engineering. West Virginia University.

Guan, S. (2012). Modeling Geomechanical Property Changes Using Well Logging and Pressure Data in a CO_2 Enhanced Oil Recovery Reservoir. M.Sc. Thesis, Department of Petroleum Engineering. Colorado School of Mines.

Hastie, T., Tibshirani, R., and Friedman, J. (2009). The Elements of Statistical Learning: data mining, inference, and prediction. 2nd ed. New York: Springer.

Lopes, R.L., and Jorge, A.M. (2017). Mind the Gap: Well Log Data Analysis. Proceedings of the Workshop on Data Mining for Oil and Gas, Houston, Texas, 29th April, 2017.

Rolon, L.F., Mohaghegh, S.D., Ameri, S., Gaskari, R., and McDaniel, B.A. (2005). Developing Synthetic Well Logs for the Upper Devonian Units in Southern Pennsylvania. SPE paper 98013-MS presented at SPE Eastern Regional Meeting, Morgantown, West Virginia, 14 16 September, 2005. Doi: http://dx.doi.org/10.2118/98013-MS

Rolon, L.F., Mohaghegh, S.D., Ameri, S., Gaskari, R., and McDaniel, B.A. (2009). Using Artificial Neural Networks to Generate Synthetic Well Logs. Journal of Natural Gas Science and Engineering 1 (2009) 118 – 133.