Geostatistics and Machine Learning Decision Tree



- Decision Tree
- Ensemble Methods

Introduction

Data Analytics

Inferential Methods

Predictive Methods

Advanced Methods

Conclusions

Michael Pyrcz, The University of Texas at Austin

Geostatistics and Machine Learning Decision Tree



Decision Tree

Introduction

Data Analytics

Inferential Methods

Predictive Methods

Advanced Methods

Conclusions

Michael Pyrcz, The University of Texas at Austin

Training and Testing



Training Phase

 The training subset of the data is applied to select the model parameters (fit the model) usually optimized to minimize the mean square error.

$$MSE = \frac{1}{n} \sum_{i=1}^{n} \left[\left(y_i - \hat{f}(x_1^i, ..., x_m^i) \right)^2 \right], for i = 1, ..., n_{train}$$

Testing Phase

- Apply the model to the testing data (data withheld from training)
- Optimize the model hyperparameters (e.g. complexity) to minimize mean square error with the testing data

$$MSE = \frac{1}{n} \sum_{i=1}^{n} \left[\left(y_i - \hat{f}(x_1^j, ..., x_m^j) \right)^2 \right], for i = 1, ..., n_{test}$$

Do not use all data to train or you will likely overfit to the data and not predict well with new data. Various methods, **k-fold cross validation** is common.

Decision Trees



Decision trees are used for supervised learning.

$$Y = f(X_1, \dots, X_m) + \epsilon$$

we are predicting a response, Y, from a set of features, X_1 , ..., X_m

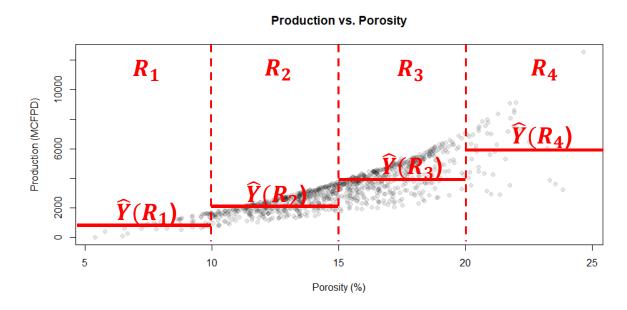
- May work with continuous Y for regression tree or categorical Y for classification tree.
- Why cover decision trees?
 - They are not the most powerful, cutting edge method in machine learning
 - But they are likely the most understandable, interpretable
 - Decision trees are expanded with random forests, bagging and boosting to be cutting edge.

"Let's learn first about a single tree and then we can comprehend the forest."

Decision Trees

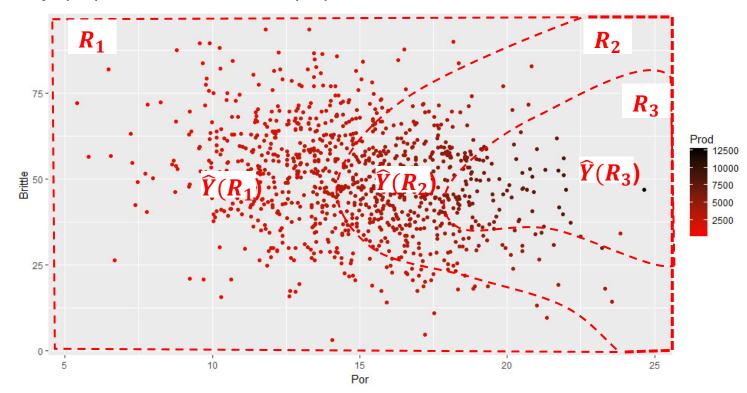


- The fundamental idea is to divide the predictor space, $X_1, ..., X_m$, into J mutually exclusive, exhaustive regions
 - mutually exclusive any combination of predictors only belongs to a single region, R_i
 - exhaustive all combinations of predictors belong a region, R_j , regions cover entire feature space (range of the variables being considered)
- For every observation in a region, R_j , we use the same prediction, $\hat{Y}(R_j)$
- For example predict production, \widehat{Y} , from porosity, X_1



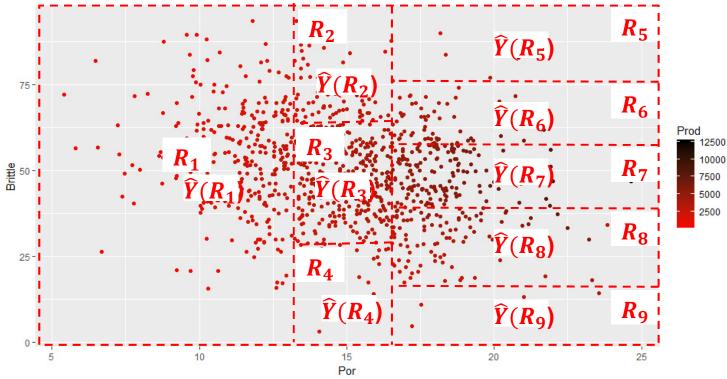


- How do we construct the Regions, $R_1, R_2, ..., R_J$?
 - They could be any shape!
 - Consider the 3 variable problem below.
- Prediction of unconventional well production (MCFPD) from porosity (%) and brittleness (%)





- How do we construct the Regions, $R_1, R_2, ..., R_I$?
 - They could be any shape!
 - Consider the trivariate (3 variable) problem below.
 - We decide to use high-dimensional rectangles or boxes ⇒simple interpretation / rules
 - » Hierarchical segmentation over the features very flexible, compact model!



Prediction of unconventional well production (MCFPD) from porosity (%) and brittleness (%)



How do we construct the Regions, $R_1, R_2, ..., R_I$?

– We want to minimize the Residual Sum of Squares:

$$RSS = \sum_{j=1}^{J} \sum_{i \in R_j} (y_i - \hat{y}_{R_j})^2$$

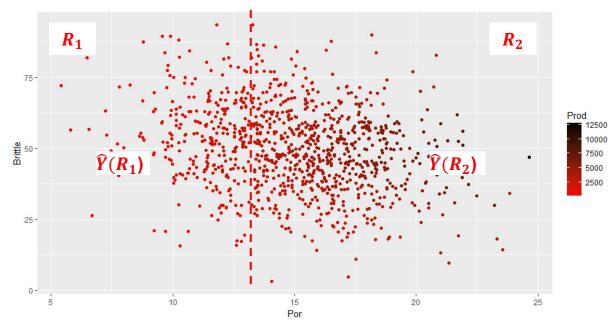
looping over *J* regions and data in each region, $i \in R_i$

- This is the sum of squares of all the data vs. the estimate in their region (the mean of the training data in the region)
- Hint: somehow we need to account for the cost of complexity
 - » We do this through cross validation and pruning



How do we construct the Regions, $R_{1}, R_{2}, ..., R_{J}$?

- Recursive, binary splitting
 - Greedy at each step the method selects the choice that minimizes RSS.
 There is no attempt to look ahead, jointly optimize over multiple choices
 - Top-down at the beginning all data belong to a single region, top of the tree, greedy selection of the single best split over any feature that best reduces the RSS



Prediction of unconventional well production (MCFPD) from porosity (%) and brittleness (%)



How do we construct the Regions, R_1, R_2, \dots, R_J ?

- Let's start with one region, R_1 with all the training data in it
 - We will place the region boundary based on a threshold, s, inside a this region, j, such that it minimize the RSS.
 - This requires search over all possible thresholds over all features within that region.
 - This is not computationally impossible (not a big space to search)

$$R_{1(m,s)} = \{X | X_m < s\} \text{ and } R_{2(m,s)} = \{X | X_m \ge s\}$$

- X_m are the features and s is the threshold for the segmentation into R_1 and R_2
- We segment such that we minimize the Residual Sum of Squares:

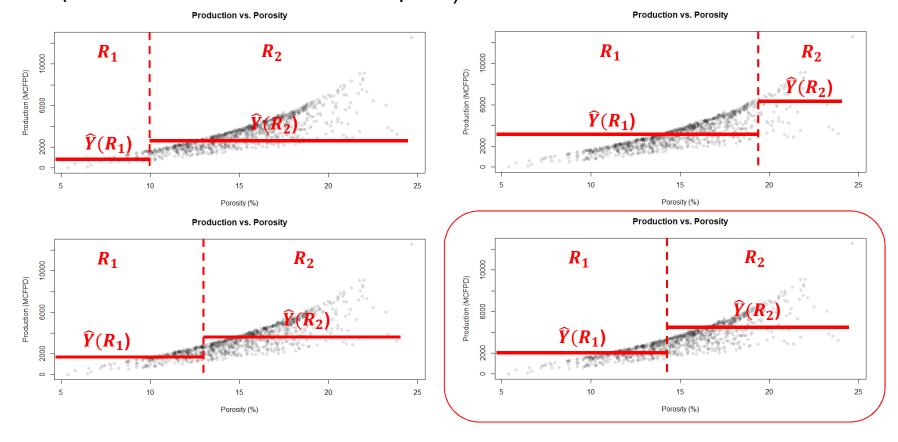
$$RSS = \sum_{i:x_i \in R_1(m,s)} (y_i - \hat{y}_{R_1})^2 + \sum_{i:x_i \in R_2(m,s)} (y_i - \hat{y}_{R_2})^2$$

 Then we just repeat for over the two region to find the next best segmentation.



Let's pause and go back to our initial univariate problem and make a tree by hand!

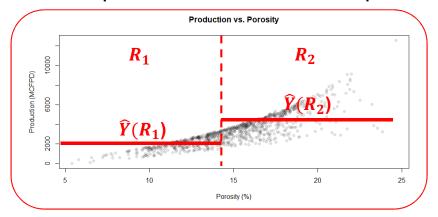
 Where should we split to minimize the error in a tree-based estimate (minimize the residual sum of square)?

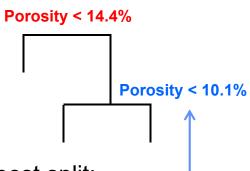




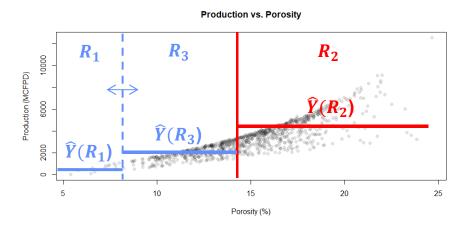
Let's pause and go back to our initial bivariate problem and make a tree by hand!

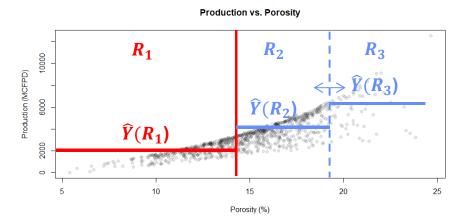
Found first split, now check for next split the maximizes accuracy





• Search over all regions and variables, to find the next best split:



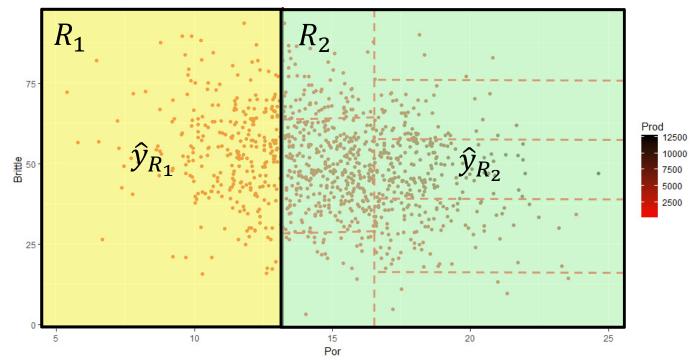




How do we construct the Regions, R_1, R_2, \dots, R_I ?

- The we continue sequentially segmenting region with threshold.
 - We will place the region boundaries based on a threshold, s, inside a previous

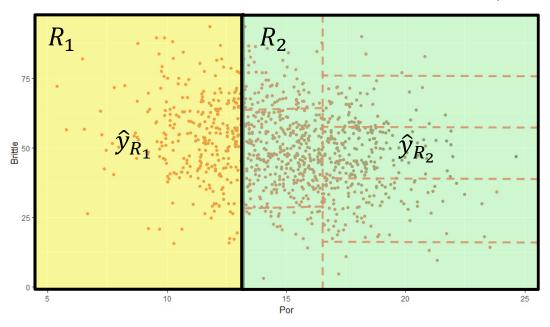
$$RSS = \sum_{i:x_i \in R_1} (y_i - \hat{y}_{R_1})^2 + \sum_{i:x_i \in R_2} (y_i - \hat{y}_{R_2})^2 + \dots + \sum_{i:x_i \in R_J} (y_i - \hat{y}_{R_J})^2$$

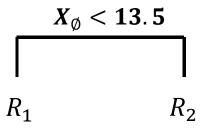


Prediction of unconventional well production (MCFPD) from porosity (%) and brittleness (%)



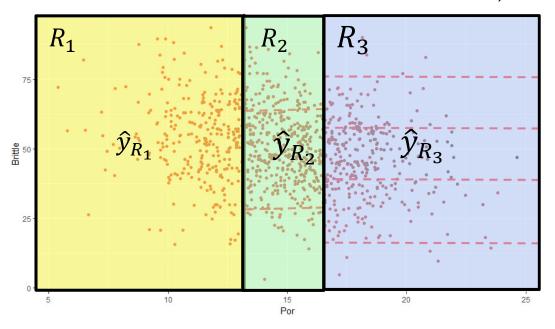
How do we construct the Regions, R_{1} , R_{2} , ..., R_{J} ?

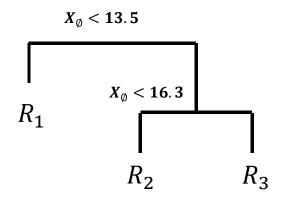






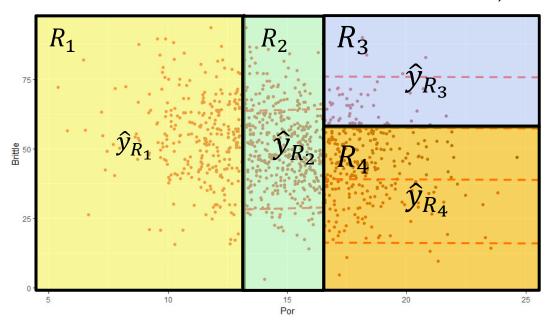
How do we construct the Regions, R_{1} , R_{2} , ..., R_{J} ?

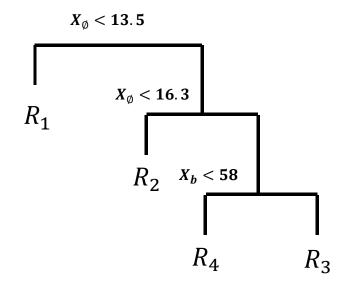






How do we construct the Regions, R_{1} , R_{2} , ..., R_{J} ?



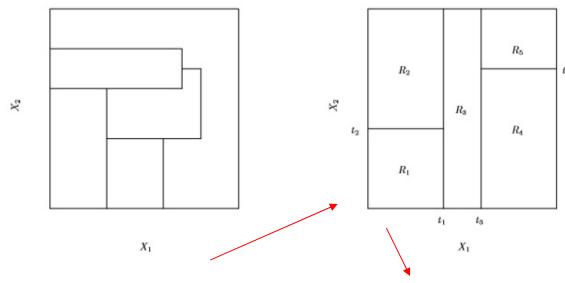


Decision Trees The Regions Not from recursive binary splitting

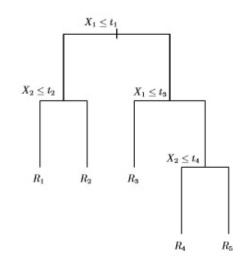
Segmented Feature Space

Example from James et al. (2017)

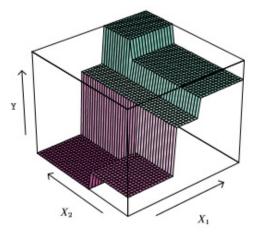
- Top-left 2D feature space partitioning that could not result from recursive binary splitting
- Top-right feature space partitioning, decision tree and estimation surface for feature space.



Decision Tree



Prediction Surface



Decision Trees Termination



When do we stop recursive binary splitting?

- We could continue until every training data value is in it's own region!
 - This would be overfit!
- The typical approach is to apply a minimum training data in each region criteria
 - The algorithm stops when all boxes have reached the minimum
- We could continue until we cannot not significantly reduce RSS
 - But the current split could lead to an even better split ⇒ short sighted

Decision Trees Pruning



Why do we want a less complicated tree?

- Decision trees, if allowed to grow very complicated are generally overfit.
- It is better to simplify the tree to a smaller tree with fewer splits
 - » lower model variance
 - » better interpretation
 - » with little added model bias
- Limiting tree growth with a high decrease in RSS hurdle is short sighted
- Best strategy is to build a large, complicated tree and then to prune the tree.
 - » We then select the sub tree to provides the lowest test error rate
 - » We cannot consider all possible sub trees (too vast of a solution space)

Decision Trees – Steps



Building a Regression Tree

- Apply recursive binary splitting to train / grow a large tree with training data, stop when each terminal node has fewer than a minimum number of data or insufficient RSS decrease.
- 2. Obtain the sequence of best subtrees as a function of complexity (number of terminal nodes) and RSS with training.
- 3. Use k-fold cross validation to chose the best complexity value. Divide the training observations into K folds. For each fold, k = 1, ... K:
 - a) Repeats steps from 1-2 on all training excluding those in k fold.
 - b) Evaluate the RSS on the left out data in the k fold.
- 4. Average the error for each α (K results over each fold) and select complexity (number of terminal nodes) that provides low enough RSS.

K-fold Cross Validation



Cross Validation

- Withhold subset of the data during model training
- Then testing the trained model with withheld subset dataset
- Must make sure cross validation is fair
- Training data set (used for training), Testing data set (withheld for testing)

K-fold Approach

- Select K, for example
- Break data set into K subsets
- Loop over K subsets:
 - use data outside the K part to predict inside the K subset
- Average to summarize the result

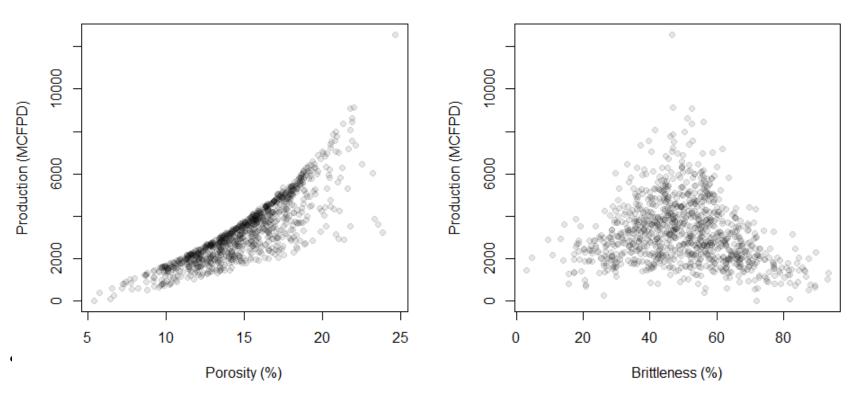


Let's use our Unconventional Multivariate Data

- We added in a production variable for prediction
- Both porosity and brittleness have interesting relationships with production

Production vs. Porosity

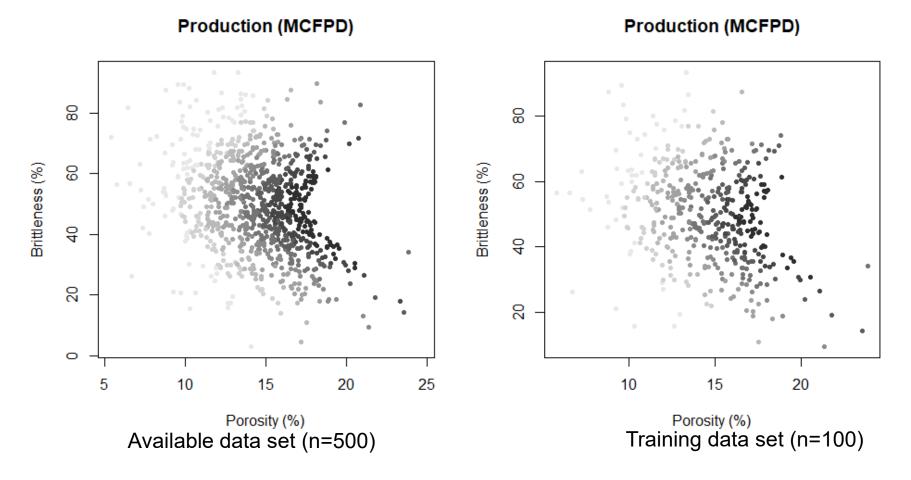
Production vs. Brittleness





Let's use our Unconventional Multivariate Data

 There is a complicated relationships between porosity, brittleness and production.





Build the initial reasonably complicated tree

- By using the default tree controls we get an 10 terminal node tree.
- We can use the summary command to:

```
Regression tree:
tree(formula = Prod ~ Por + Brittle, data = train, control = tree.control)
Number of terminal nodes: 10
Residual mean deviance: 302900 = 148400000 / 490
Distribution of residuals:
    Min. 1st Qu. Median Mean 3rd Qu. Max.
-2298.00 -303.50 57.16 0.00 327.50 3668.00
```

- check the complexity of the resulting tree (number of terminal nodes)
- check the summary statistics of the residuals and ensure that the model is not biased (mean = 0.0)
- residual mean deviance is the total residual deviance divided by (the number of observations – number of terminal nodes)
- for a regression trees the total residual deviance is the RSS, reminder:

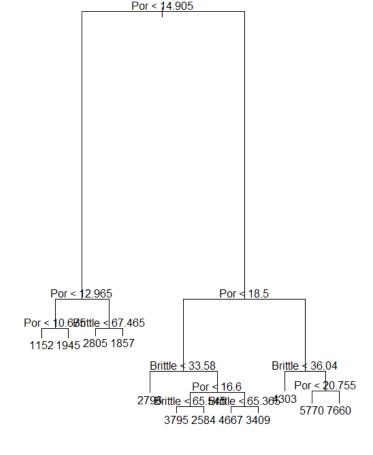
RSS =
$$\sum_{j=1}^{J} \sum_{i \in R_j} (y_i - \hat{y}_{R_j})^2$$



Build the initial reasonably complicated tree

Here's the tree:

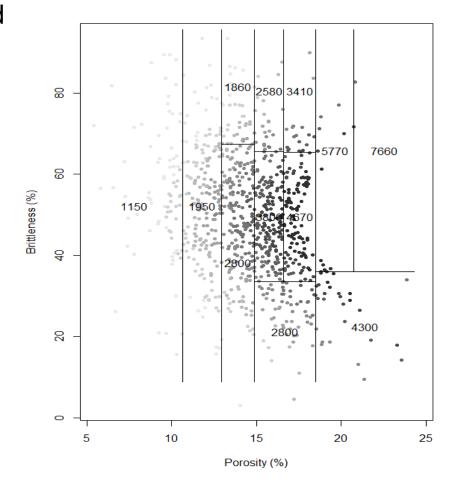
- first choice is porosity < or > 14.9%
- we get to the 3rd decision before brittleness is considered
- length of the branches is proportional to decrease in impurity
 - decress in RSS of the model for regression tree
 - a measure of node heterogeneity for classification trees





Build the initial reasonably complicated tree

 We can plot the original data and the binary recursive boundaries outlining the various regions and the mean values in each region used as the estimate.

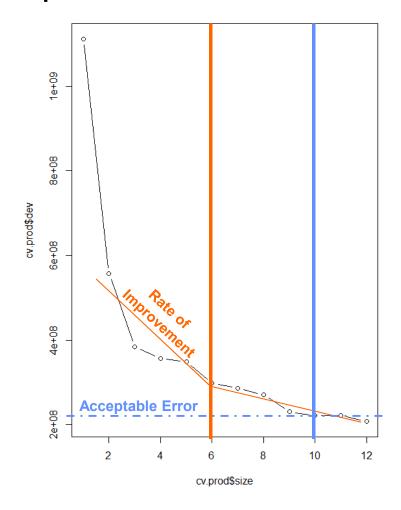




Build the initial reasonably complicated tree

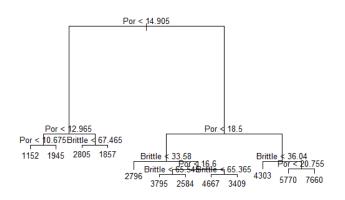
Then we perform k fold cross validation.

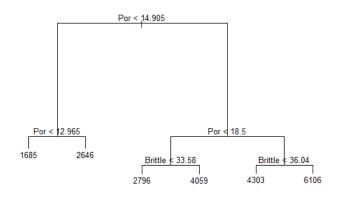
- Decrease tree complexity from 12 nodes (current model) to 1 node (uniform model)
- Calculate the RSS by averaging over k folds of the training data
- We can observed that each additional node improves the model
- Prune complexity based on:
 - Diminishing returns
 - Acceptable level of error

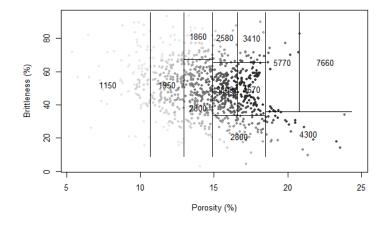


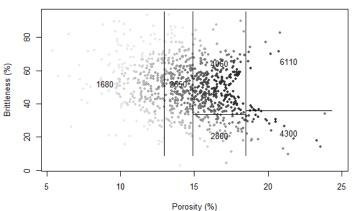


Original and pruned tree:



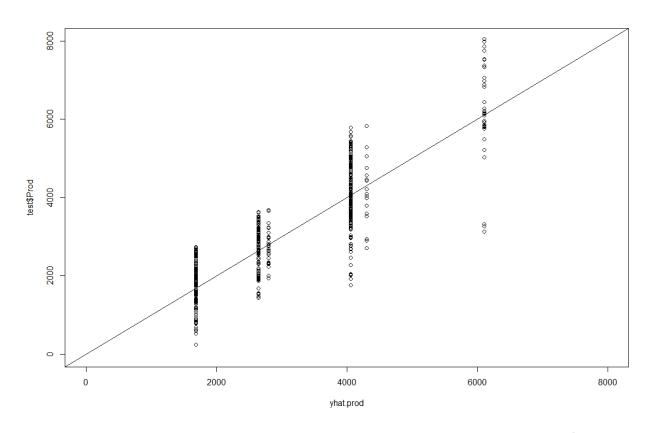








Cross validation with the testing data set



- Note: the binning due to estimation with the mean of only 6 regions
- We can calculate MSE to assess model accuracy

Decision Trees Demonstration in Python



Decision Tree Tutorial with Subsurface Demonstration in Python for Geoscientists and Geo-engineers Michael Pyrcz, University of Texas at Austin (@GeostatsGuy)



Decision Tree is one of the most simple, explainable and interpretable predictive models in machine learning; therefore, it is a great introduction to regression and classification with machine learning. In addition, the recursive binary segmentation is analogous to human decision making. Finally, understanding a decision tree is a prerequisite for more powerful bagging, random forest and boosting. This tutorial is in Jupyter with Markdown and a realistic unconventional dataset. There is enough documentation that any geoscientists or engineer should be able to try out machine learning.



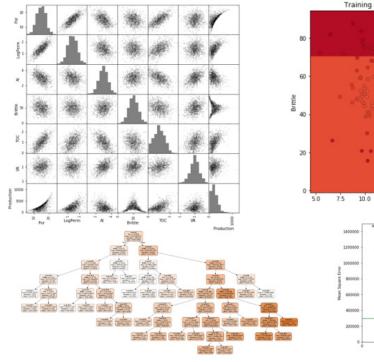
methods to explain, apply and interogate. In addition, understanding decision tree-based prediction is a prerequisite to more complicated and powerful methods such as random forest and tree-based bagging and boosting. For this demonstration we use a 1,000 well 7 variable unconventional dataset (file: unconv_MV.csv) ns/GeoDataSets. This dataset includes 6 predictors (features) and 1 response. We take this reality ariable datasset and only retain the three variables (2 credictors and 1 response) for a simple demonstration of the decision tree method. We break the data

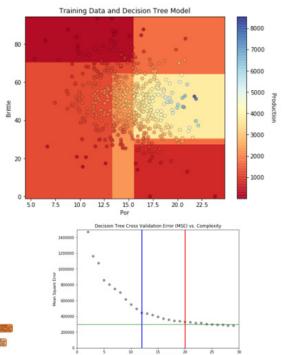
If make a couple of points about decision trees. For greater detail there are a lot of online resources on decision trees along with the book "An Int Statistical Learning" by James et al. (my favourite).

- c comprome precessors was a consumerative area. Community precessors as a registration are a registration to a function of leaf nodes in the tree). A inclamental issels to do notice feature space in the character, manually or custome registration (returned or leaf nodes in the tree). 4. estimate with the average of data in each region for configured precisions or the majority codingory for the data in each region for catego 5 septemb for helicular space with hierarchical hamps spitting that many be registered for a discission free. 6. apply a greedy method to find the sequential spitts for any feature that minimizes the residual sum of squares.
- Lef's build some decision trees together. You'll get a chance to see the trees and the divided feature space graphically

max	75%	50%	25%	min	std	mean	count	
24.65000	17.080000	14.98500	12.85750	5.400000	3.029634	14.950460	1000.0	Por
2.58000	1.680000	1.39000	1.13000	0.120000	0.405988	1.398880	1000.0	LogPerm
4.70000	3.360000	3.01000	2.57750	0.960000	0.577629	2.982610	1000.0	Al
93.47000	59.170000	49.68000	39.72250	-10.500000	15.077006	49.719480	1000.0	Brittle
2.71000	1.360000	0.99500	0.64000	-0.260000	0.504978	1.003810	1000.0	TOC
2.90000	2.172500	2.00000	1.81000	0.900000	0.308194	1.991170	1000.0	VR
12588.84413	3023.594214	1976.48782	1191.38958	2.713535	1484.258312	2247.295809	1000.0	Production







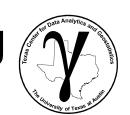
Decision Trees Comments



General Comments on Decision Trees

- Easy to explain
- Analog to human decision making
- Graphically displayed
- Continuous or categorical variables
- Lower predictive accuracy than other machine learning methods
- Model bias may be high

Geostatistics and Machine Learning Decision Tree



Ensemble Methods

Introduction

Data Analytics

Inferential Methods

Predictive Methods

Advanced Methods

Conclusions

Michael Pyrcz, The University of Texas at Austin

Bagging, Random Forest and Boosting

These are all methods to improve the prediction accuracy of trees

- Bagging (used with many types of models)
 - the use of bootstrap on the training dataset to get B training sets
 - train a tree on each data set
 - then use all models and average the result to get the prediction

$$\hat{f}_{bag}(x) = \frac{1}{B} \sum_{b=1}^{B} \hat{f}^{*b}(x)$$

- the trees are allowed to grow large
- 100s to 1,000s of trees (forest of mediocre estimates!)
- classification by majority vote
- out-of-bag data (about 1/3 for each tree) are used as a test data set!

Bagging, Random Forest and Boosting

These are all methods to improve the prediction accuracy of trees

- Random Forest
 - same as bagging, but we randomize selection of on about \sqrt{m} of the features!
 - prevents a single strong predictor form dominating the entire set of trees – forces diversity among the trees
 - decorrelating the trees

Bagging, Random Forest and Boosting

These are all methods to improve the prediction accuracy of trees

- Boosting (used with many types of models)
 - sequential modeling of a simple tree
 - build a tree, calculate residual
 - build a tree to model residual from 1st tree
 - build a tree to model the residual from 2nd tree
 - etc.

Geostatistics and Machine Learning Decision Tree



- Decision Tree
- Ensemble Methods

Introduction

Data Analytics

Inferential Methods

Predictive Methods

Advanced Methods

Conclusions

Michael Pyrcz, The University of Texas at Austin