Subsurface Data Analytics and Machine Learning
Predictive Models

Lecture outline . . .

- Prediction
- A Simple Parametric Model
- Decision Tree

Introduction

Data Analytics

Inferential Methods

Predictive Methods

Advanced Methods

Conclusions

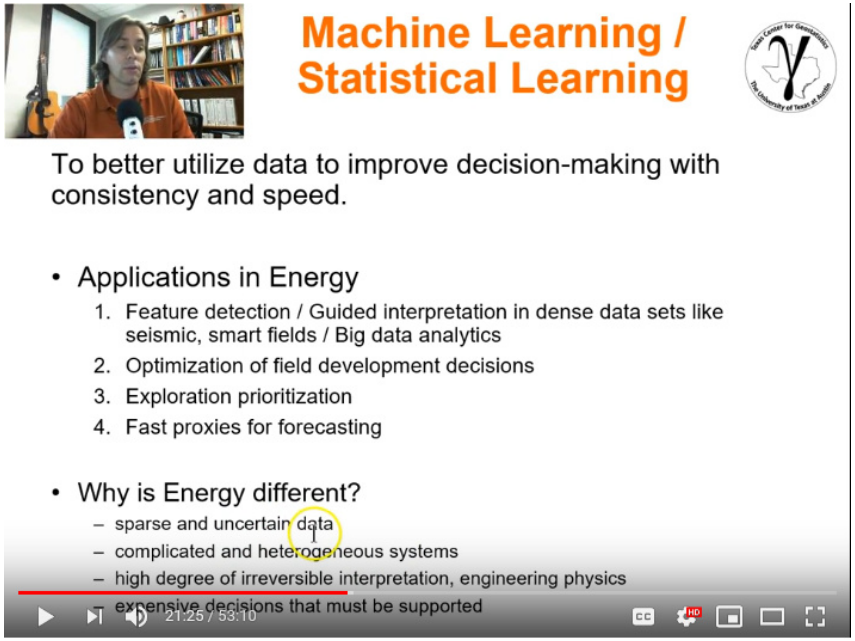
Instructor: Michael Pyrcz, the University of Texas at Austin

Subsurface Data Analytics and Machine Learning

Basic Prediction

Other Resources:

- **Statistical Learning, Dimensional Reduction and Decision Tree**



Instructor: Michael Pyrcz, the University of Texas at Austin

Goals of This Lecture

- We will introduce the idea of machine learning.
- Demonstrate concepts with a very simple machine!
- Expand to the easy to interpret decision tree.
- Next lecture we get much more advanced.

Subsurface Data Analytics and Machine Learning
Predictive Models

Lecture outline . . .

- Prediction

Introduction

Data Analytics

Inferential Methods

Predictive Methods

Advanced Methods

Conclusions

Instructor: Michael Pyrcz, the University of Texas at Austin

The Model

Predictors, Independent Variables, Features

- input variables
- for a model $Y = f(X_1, \ldots, X_m) + \epsilon$, these are the $X_1, \ldots, X_m$
- note ϵ is a random error term

Response, Dependent Variables

- output variable
- for a model $Y = f(X_1, \ldots, X_m)$, this is Y

Prediction

Estimating, $\hat{f}$, for the purpose of predicting $\hat{Y}$

- We are focused on getting the most accurate estimates, $\hat{Y}$
- We may not even understand what is happening between the X's!
- We are concerned about the relationships between X and Y

'Prediction is modeling the system to make estimates, forecasts.'

Some Definitions

- **Predictor Feature** – input variables for our model
- **Response Feature** – output variable for our model
- **Supervised Learning** – working with data with the response known, the data is labelled, to build a model to make predictions where the response is not known for the predictors.

Subsurface Data Analytics and Machine Learning
Predictive Models

Lecture outline . . .

- A Simple Parametric

Introduction

Data Analytics

Inferential Methods

Predictive Methods

Advanced Methods

Conclusions

Instructor: Michael Pyrcz, the University of Texas at Austin

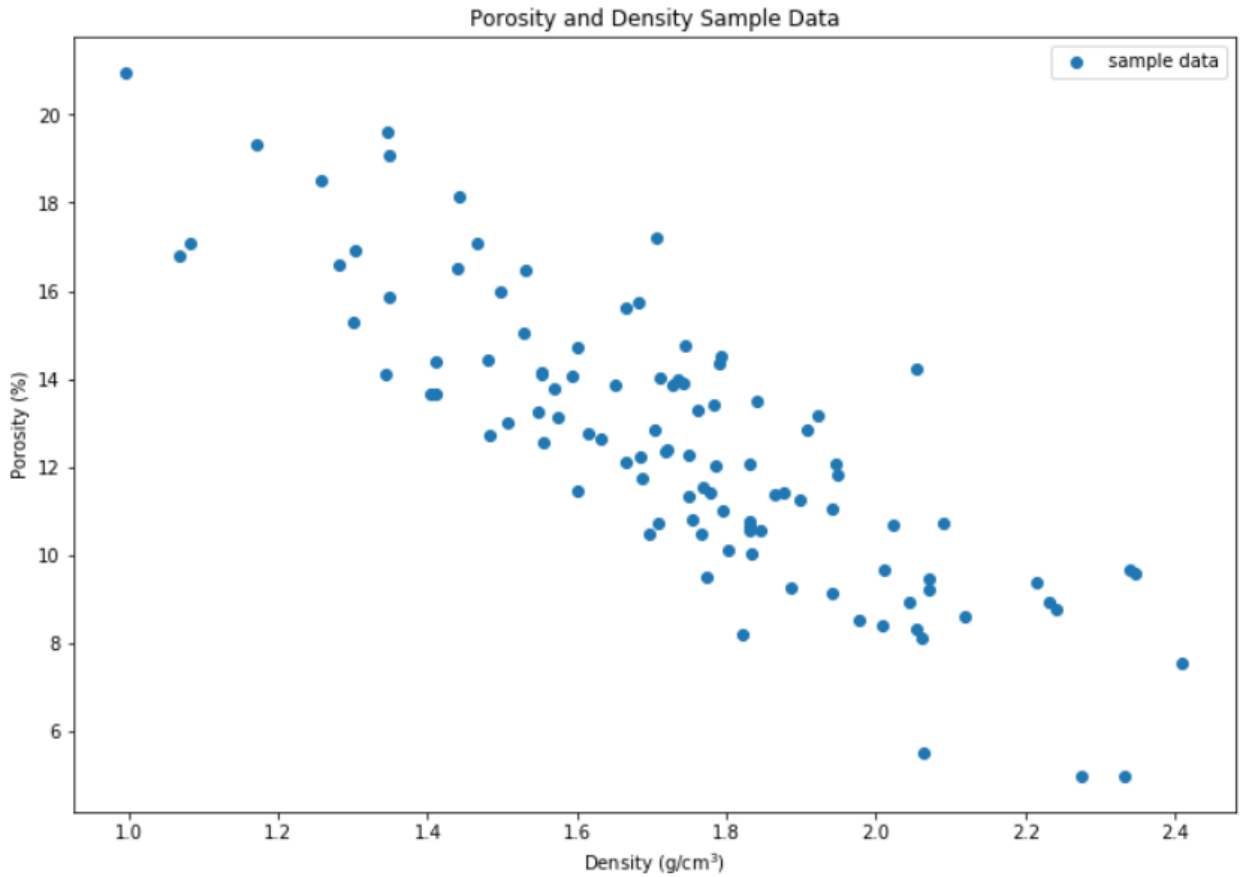
Statistical / Machine Learning for Prediction

What is Machine Learning?

- A mathematical / statistical model that learns from data, supported with expert knowledge
- Not explicitly told how to predict
- General method that may be applied to a range of problems

Our First Machine

- **Loaded up a simple porosity vs. density dataset in Python.**



Our First Machine

- Ran one line of Python and built a linear regression model

LinearRegression Model

Let's first calculate the linear regression model

```
slope, intercept, r_value, p_value, std_err = st.linregress(den,por)

print('The model parameters are, slope (b1) = ' + str(round(slope,2)) + ', and the intercept

```

```
The model parameters are, slope (b1) = -9.1, and the intercept (b0) = 28.35
```

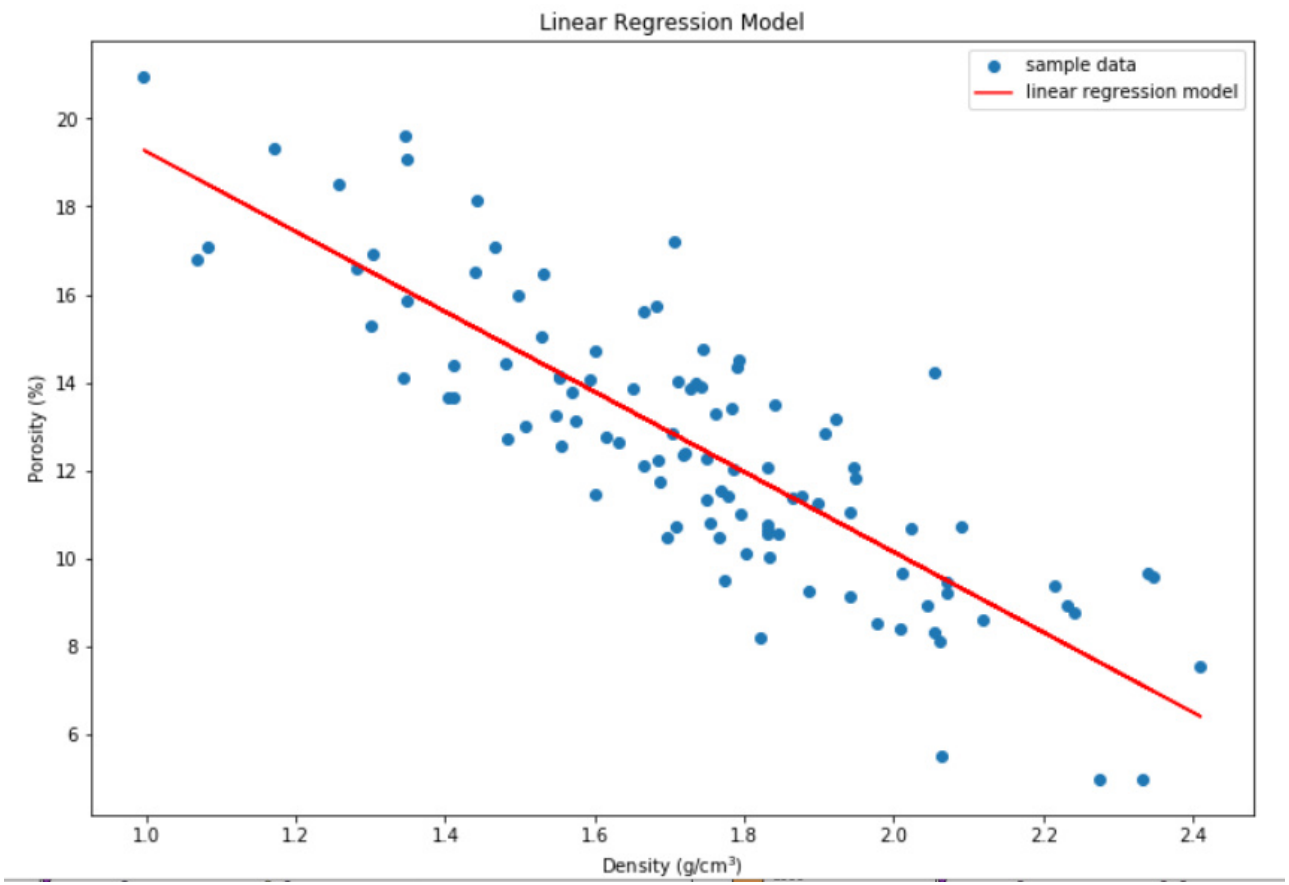
- The model is simply a line:

Response Feature → $\phi = b_1 \cdot \rho + b_0$ ← **Predictor Feature**

Our First Machine

Let's look at the model.

- If we change the data, the model would update. It learns!
- Nothing intimidating about linear regression!



Our First Machine

Model Parameters Set to Minimize Mismatch at With Training Data Locations

$$por = b_0 + b_1 \times density$$

- **Objective:**
  - **Find b_1 and b_0, fit a linear function, to:**
    - » **minimize Δy_i over all the data.**
    - » **Δy_i is prediction error**

$$\Delta y_i = y_i - y_{est}$$

data model

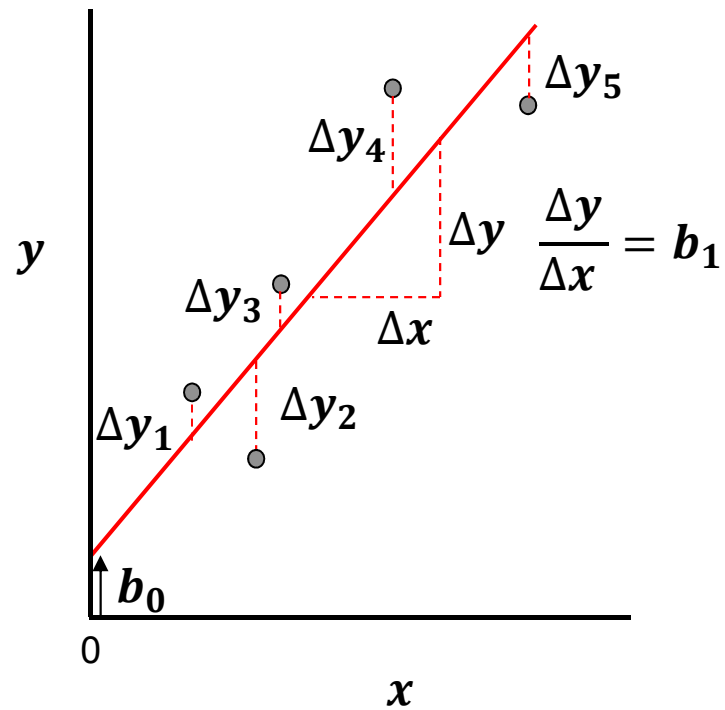
- **Minimize:**

Sum of Square Error

$$\sum_{i=1}^{n} (\Delta y_i)^2 = \sum_{i=1}^{n} (y_i - (b_0 - b_1 x))^2$$

Skipped derivation.

$$b_1 = \frac{\sum_{i=1}^{n} (x_i - \bar{x})(y_i - \bar{y})}{\sum_{i=1}^{n} (x_i - \bar{x})^2}, \quad b_0 = \bar{y} - b_1 \bar{x}$$



Our First Machine

The Model Includes Important Assumptions About The Data and the Model

- **Error-free:** predictor variables are error free, not random variables
- **Linearity:** response is linear combination of feature(s)
- **Constant Variance:** error in response is constant over predictor(s) value
- **Independence of Error:** error in response are uncorrelated with each other
- **No multicollinearity:** none of the features are redundant with other features

Our First Machine

The Model Can Be Tested for Significance and the Proportion of Variance Explained.

- r^2: strength of the model, proportion of variance explained by the model

Variance explained by the model

$$ssreg = \sum_{i=1}^{n} (\hat{y}_i - \bar{y})^2$$

Variance NOT explained by the model

$$ssresid = \sum_{i=1}^{n} (y_i - \hat{y}_i)^2$$

$$r^2 = \frac{ssreg}{ssreg + ssresid} = \frac{explained\ variation}{total\ variation}$$

- also note for bivariate case, $r^2 = (\rho)^2$, we can relate r^2 to the Pearson's correlation coefficient, ρ.

Our First Machine

We Can Calculate the Uncertainty in the Model

- Confidence interval for model parameters given the available training data

$$\widehat{b_1} \pm t_{(\alpha/2, n-2)} \times SE_{b_1} \qquad \widehat{b_1} \pm t_{\alpha/2, n-2} \times \left(\frac{\sqrt{n}\hat{\sigma}}{\sqrt{n-2}\sqrt{\sum (x_i - \bar{x})^2}} \right)$$

se1 in Excel

$$\widehat{b_0} \pm t_{(\alpha/2, n-2)} \times SE_{b_0} \qquad \widehat{b_0} \pm t_{\alpha/2, n-2} \times \left(\sqrt{\frac{\hat{\sigma}^2}{n-2}} \right)$$

seb in Excel

Our First Machine

Provides an Uncertainty Model for the Predictions

Recall prediction interval are concerned with uncertainty in the next observation

next sample

- We answer the question, given I know the porosity, x_{n+1}, what is the interval (e.g.) with 95% probability containing the true value permeability, y_{n+1}?

$$\hat{y}_{n+1} \pm t_{\alpha/2,n-2}\sqrt{MSE}\sqrt{1+\frac{1}{n}+\frac{(x_{n+1}-\bar{x})^2}{\sum(x_i-\bar{x})^2}}$$

model estimate

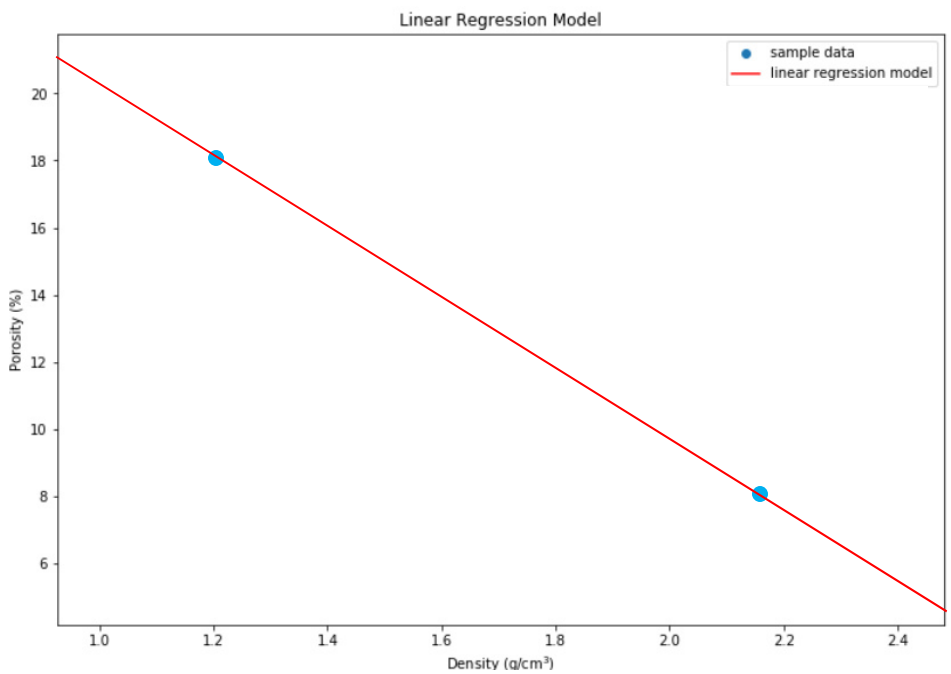
t-statistic

$$MSE = \sum_{i=1}^{n}\frac{(y_i-\hat{y}_i)^2}{n-2} = \sum_{i=1}^{n}\frac{(y_i-(b_0-b_1x))^2}{n-2}$$

standard error of our model estimate

Our First Machine

Would this be a fair model?



- Does the data support this model? We are **overfitting** the data!
- Is it safe to **extrapolate** with this model away from the data?

Our First Machine

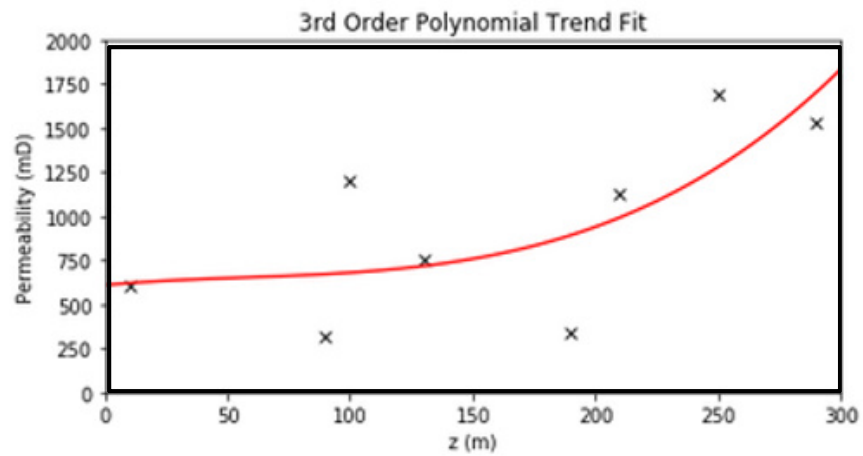
What did we learn from our simple machine?

1. Flexible to fit the data, learns from the data
2. Minimize error with the training data
3. Important assumptions about the data and model
4. Model can be tested for significance and the proportion of variance explained
5. Includes uncertainty in the model
6. Predict based on new data with uncertainty
7. Issues with overfit and extrapolation

Think of machine learning as advanced linear regression / line fitting to data!

Training Our Machine

Apply Training Data to Set the Model Parameters.



For example, the parameters of this 3rd order polynomial model.

b_3, b_2, b_1 and c

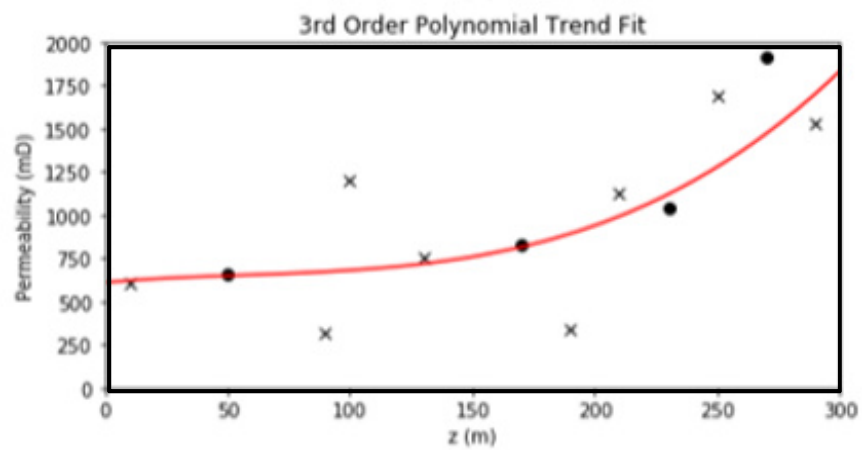
$$k = b_3 z^3 + b_2 z^2 + b_1 z + c$$

But not appropriate to determine level of complexity (hype parameter)

Hyperparameter of our model: 1st, 2nd, 3rd 4th … order polynomial?

Testing Our Machine

Apply Withheld Data to Test our Machine.



For example, the parameters of this 3rd order polynomial model.

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

In testing we use the parameters from training but we tune the hyper parameters.

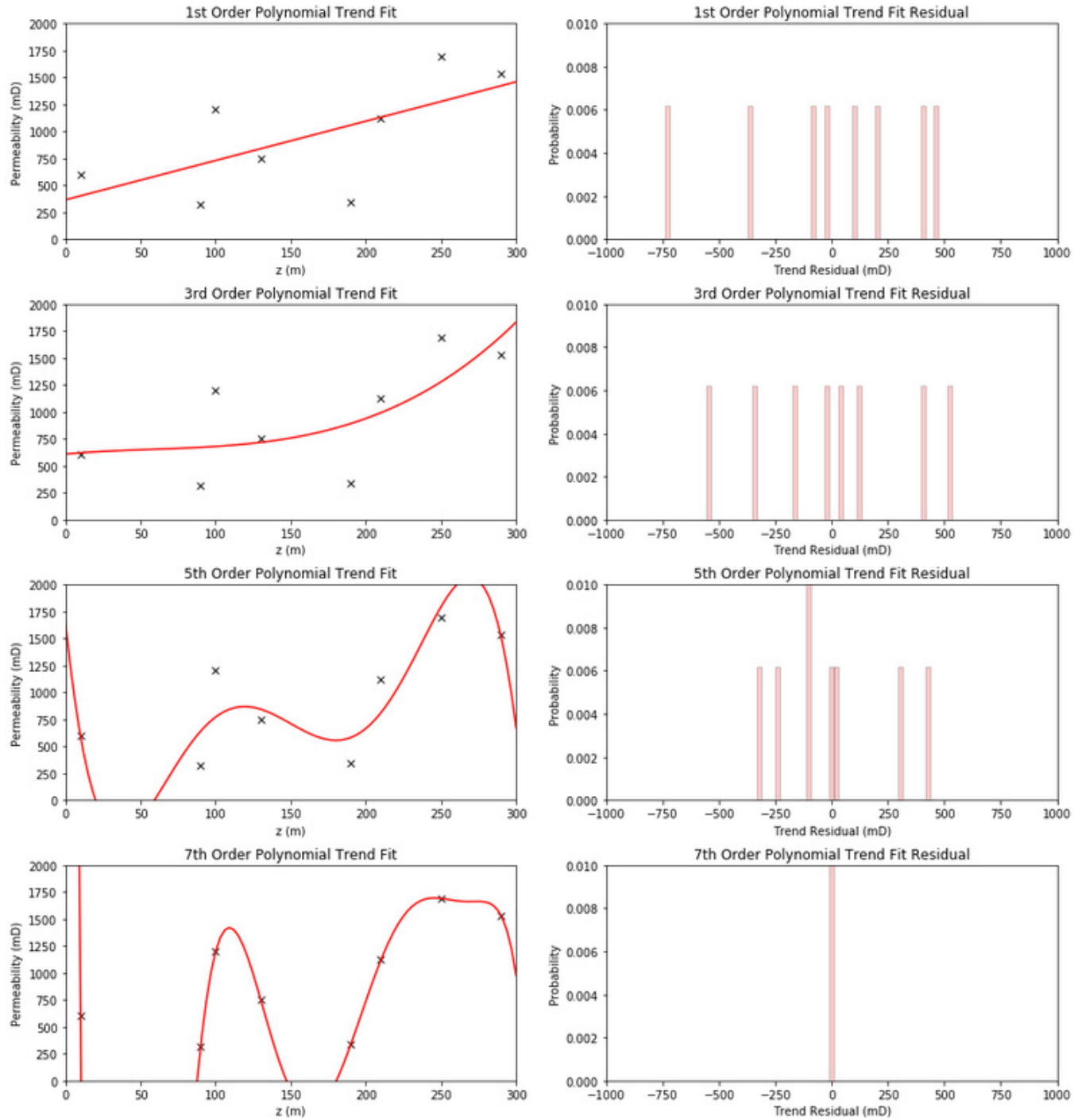
Hyperparameter of our model:
1st, 2nd, 3rd 4th … order polynomial?

Making Our Machine

What would happened if we just maximized fit to the data?

Very complicated model would be best.

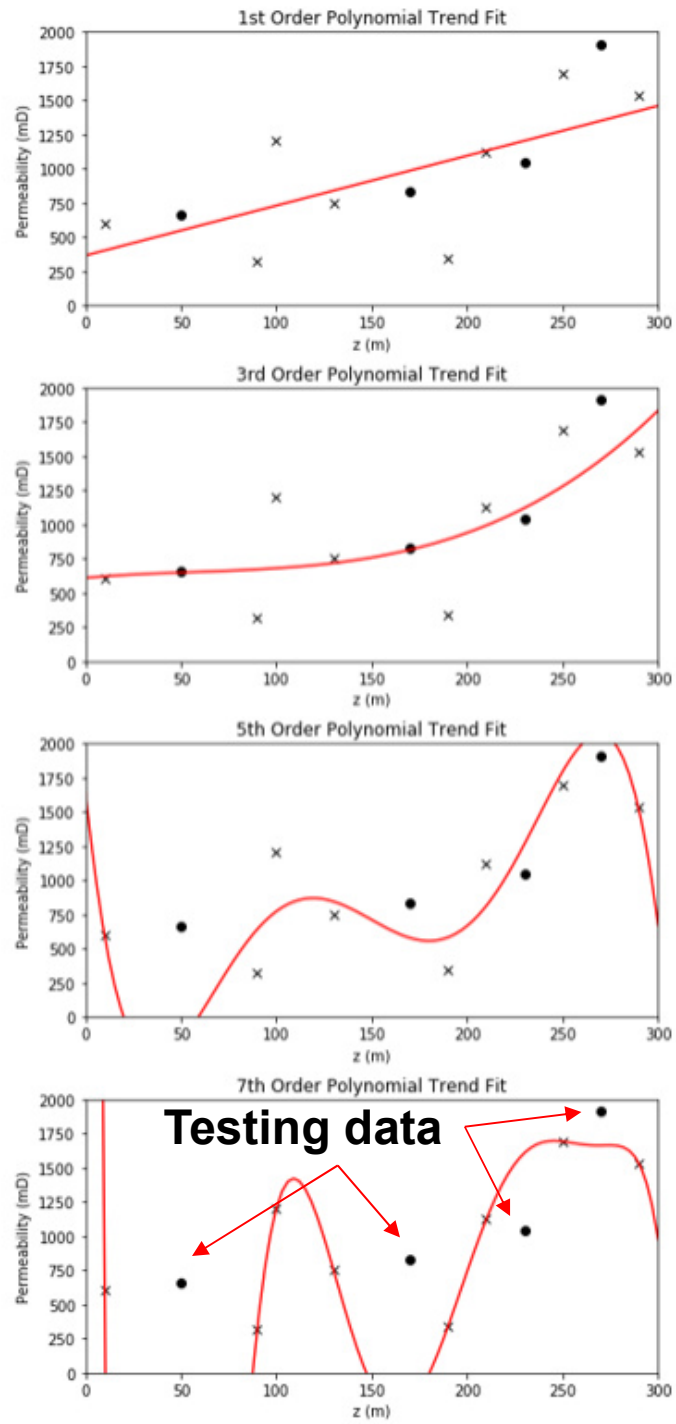
Perfectly fit the data.



Making Our Machine

The More Complicated Model Would be Overfit

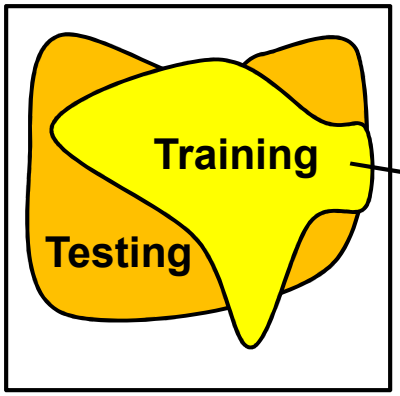
1. Have high accuracy at training data
2. Poor testing accuracy with new observations!
3. Very dangerous with extrapolation.
4. Low model bias, but **high model variance**.



Making Our Machine

The Training and Testing Workflow

Separate The Data



Train the Model

Train the Model Parameters Maximize Accuracy with Training Data

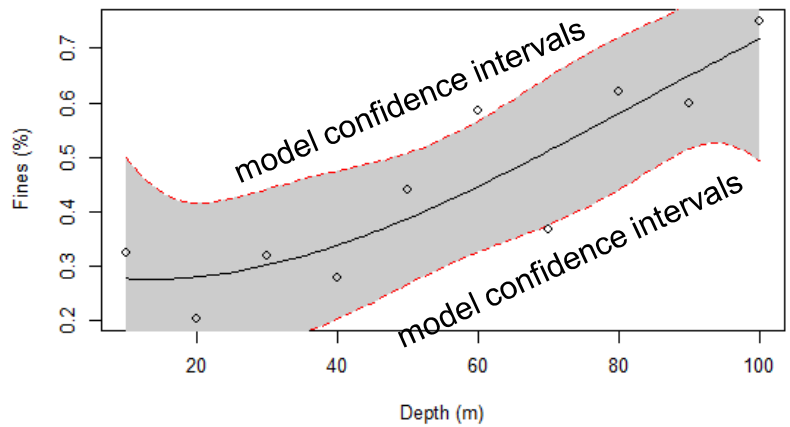
Tune the Model

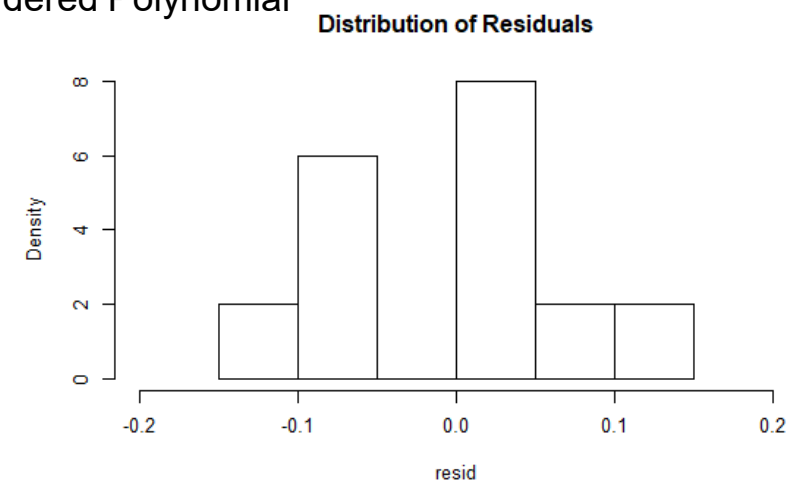
Test the Model Tune the Hyperparameters to Optimize the Complexity

We avoid the overfit problem.

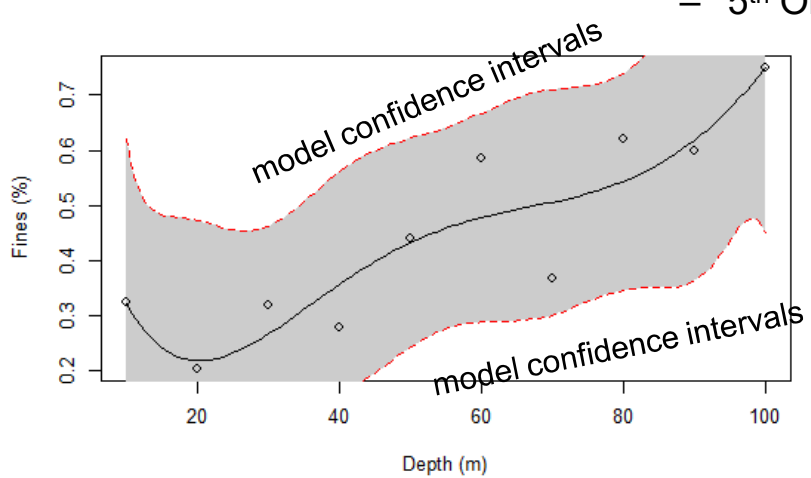
Overfitting

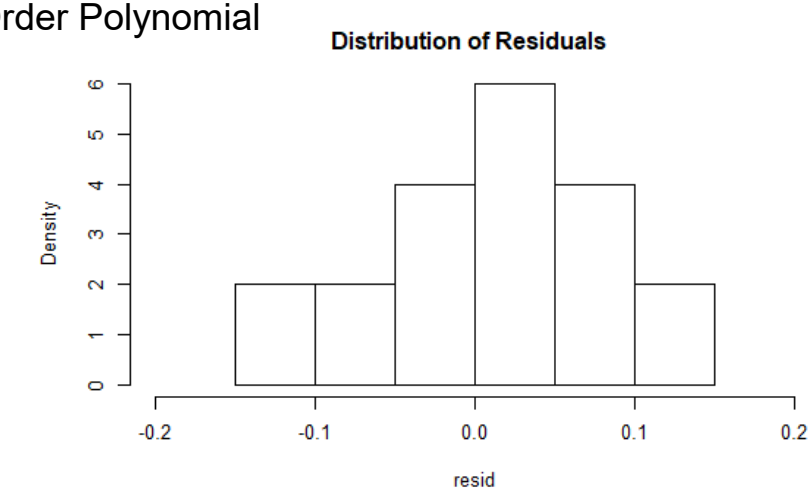
- Example of trend fits:
  - 3rd Ordered Polynomial





  - 5th Order Polynomial



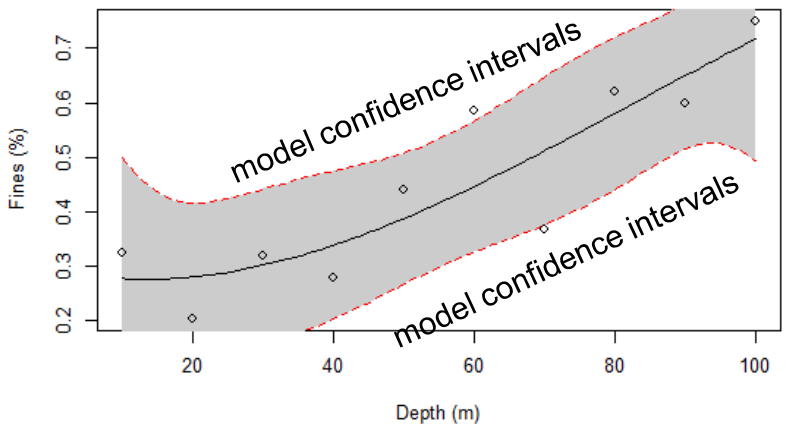


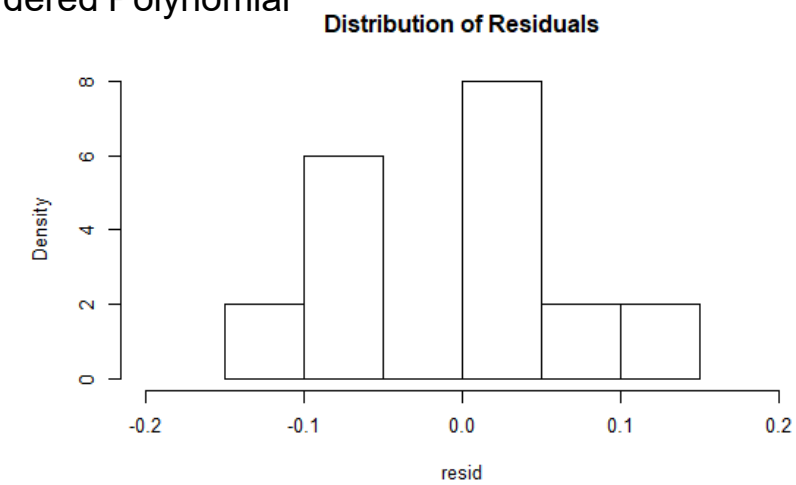
Overfit demonstration in R, code is here:
https://github.com/GeostatsGuy/geostatsr/blob/master/overfit.R

R code at Code/Overfit.R

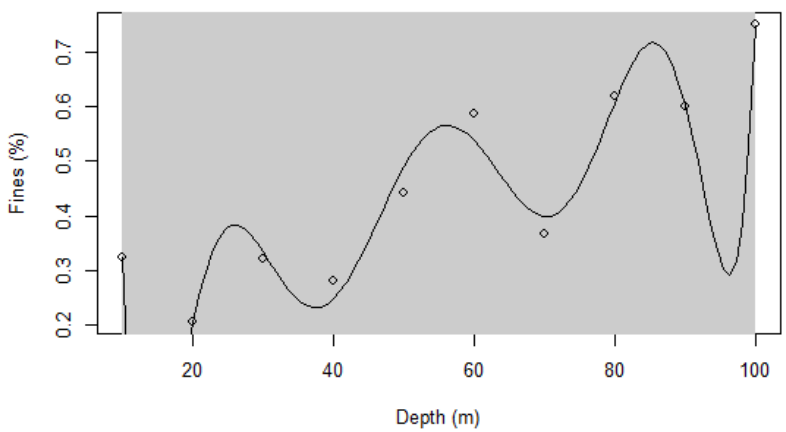
Overfitting

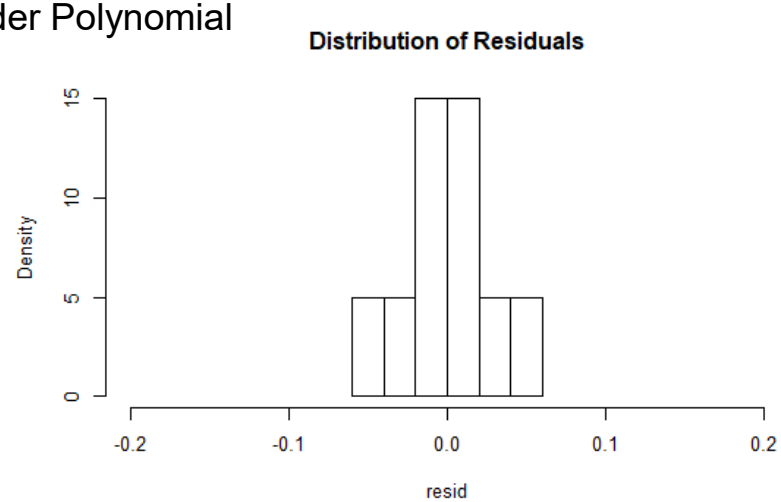
- Example of trend fits:
  - 3rd Ordered Polynomial





  - 8th Order Polynomial





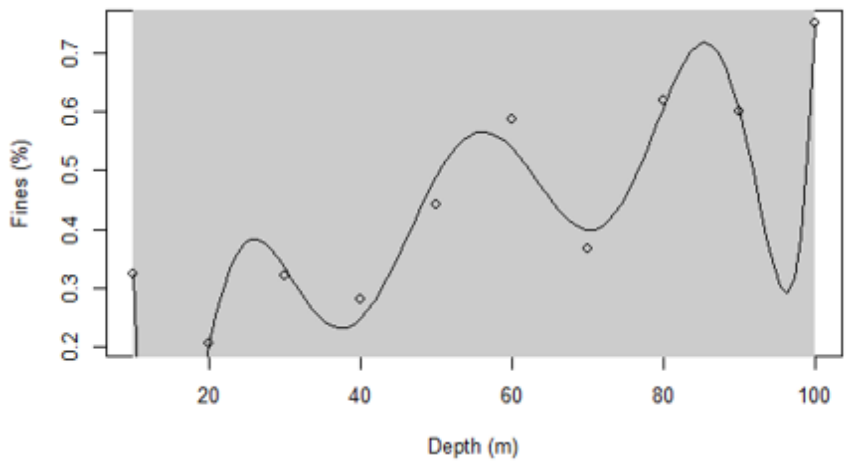
Overfit demonstration in R, code is here:
https://github.com/GeostatsGuy/geostatsr/blob/master/overfit.R

R code at Code/Overfit.R

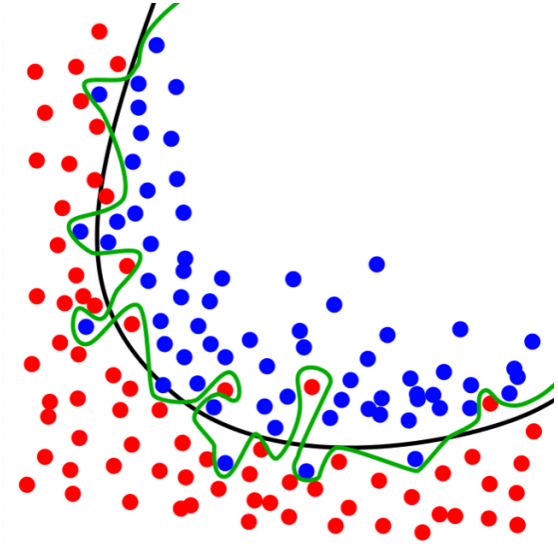
Definition of Overfitting

Overfit Model

- Overly complicated model to explain “idiosyncrasies” of the data, capturing data noise in the model
- Very high error away from the data / new data
- Very accurate at the data!



Overfit demonstration in R, code is here:
https://github.com/GeostatsGuy/geostatsr/blob/master/overfit.R



Overfit classification model example from:
https://en.wikipedia.org/wiki/Overfitting#/media/File:Overfitting.svg

Data Analytics and Geostatistics: Machine Learning

Lecture outline . . .

- Decision Tree

Introduction

Modeling Prerequisites

Spatial Estimation

Spatial Uncertainty

Multivariate, Spatial

Multivariate Analysis

Machine Learning

Novel Workflows

Conclusions

Instructor: Michael Pyrcz, the University of Texas at Austin

Decision Trees

- Decision trees are used for supervised learning.

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

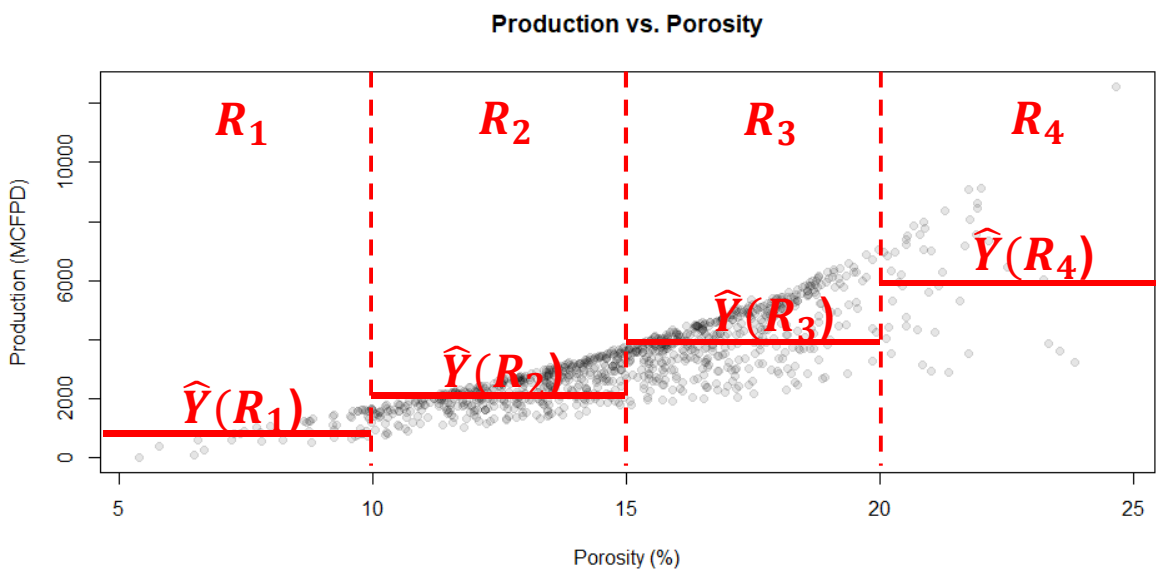
  we are predicting a response, Y, from a set of features, $X_1, \dots, X_m$
- May work with continuous Y for regression tree or categorical Y for classification tree.

- 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.

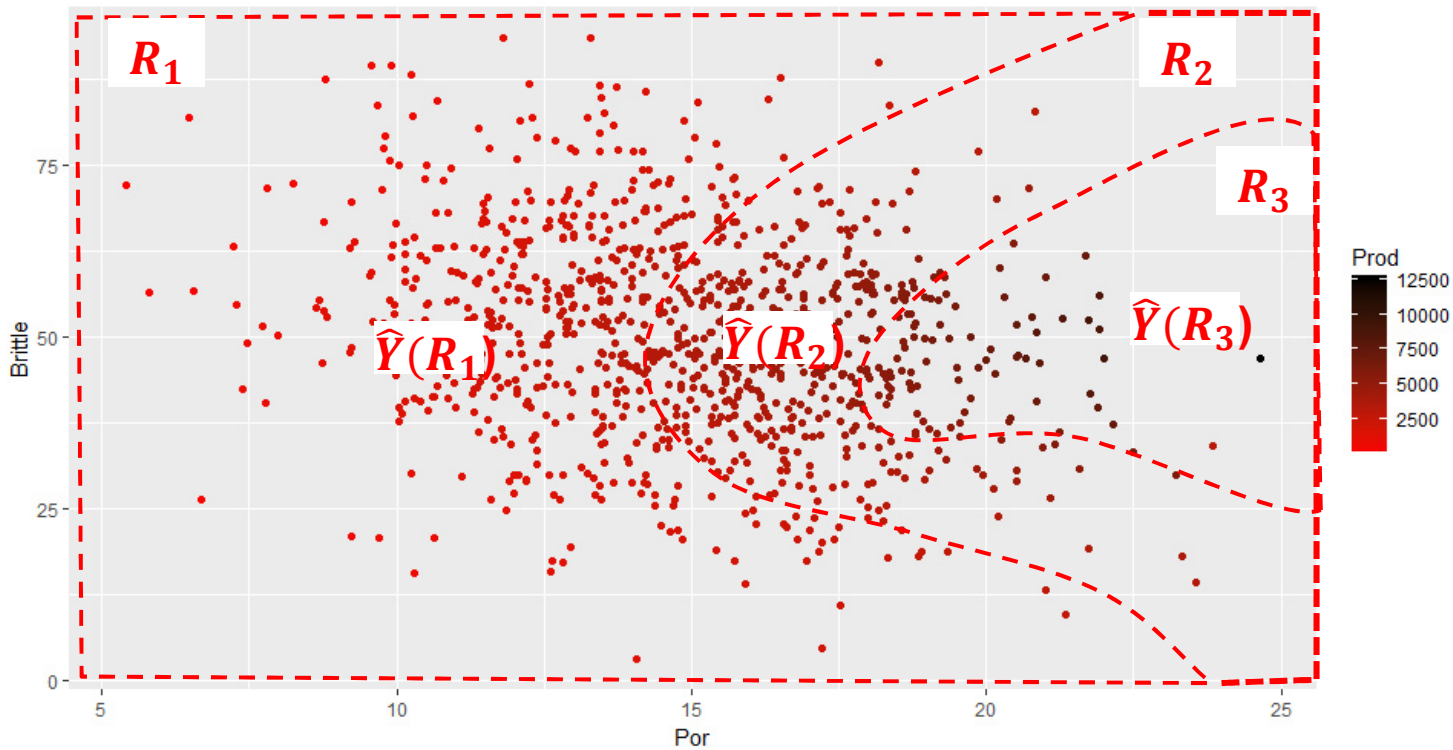
Decision Trees

- The fundamental idea is to divide the predictor space, $X_1, \ldots, X_m$, into J mutually exclusive, exhaustive regions
  - mutually exclusive – any combination of predictors only belongs to a single region, R_j
  - 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, $\hat{Y}$, from porosity, X_1



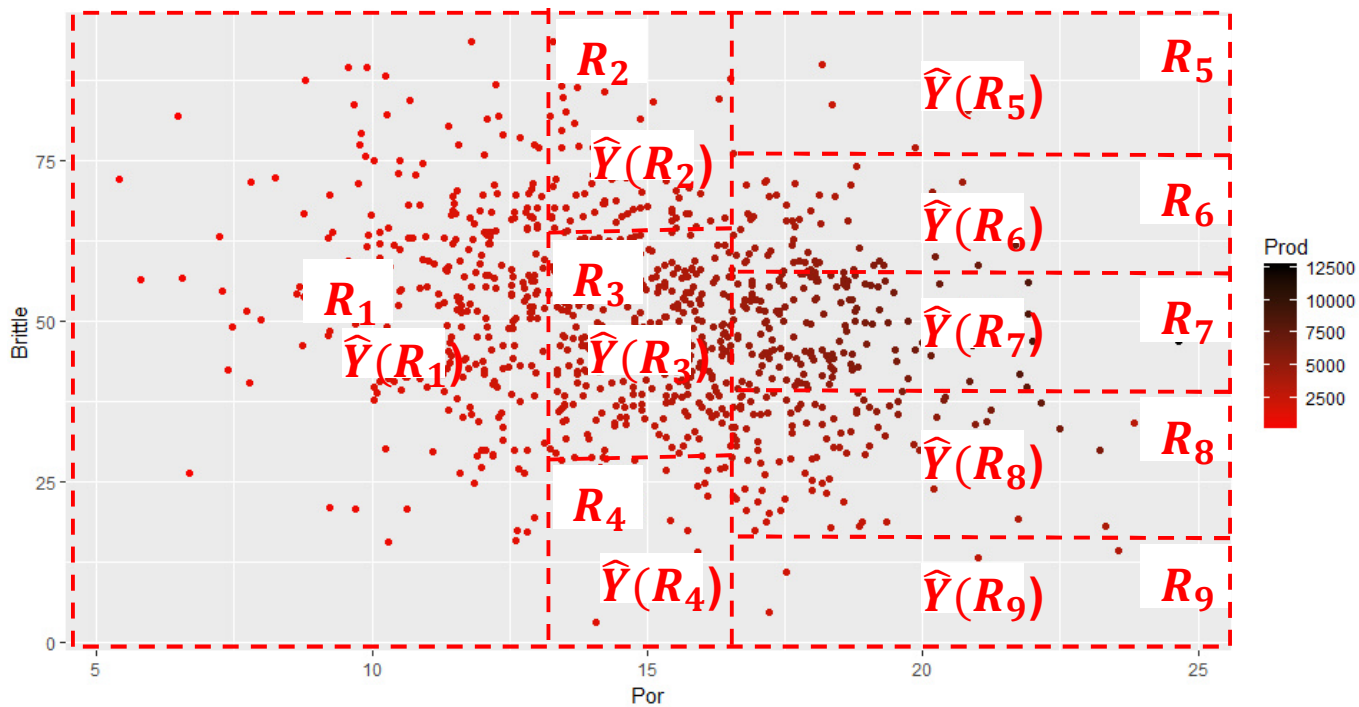
Decision Trees
The Regions

- How do we construct the Regions, $R_{1,}\, R_2\,, \ldots, R_J$?
  - They could be any shape!
  - Consider the 3 variable problem below.
- Prediction of unconventional well production (MCFPD) from porosity (%) and brittleness (%)



Decision Trees
The Regions

- How do we construct the Regions, $R_{1,}\, R_2\,, \ldots, R_J$?
  - They could be any shape!
  - 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 (%)

Decision Trees
The Regions

How do we construct the Regions, $R_{1,}\, R_2\,, \ldots, R_J$?

- We want to minimize the Residual Sum of Squares:

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

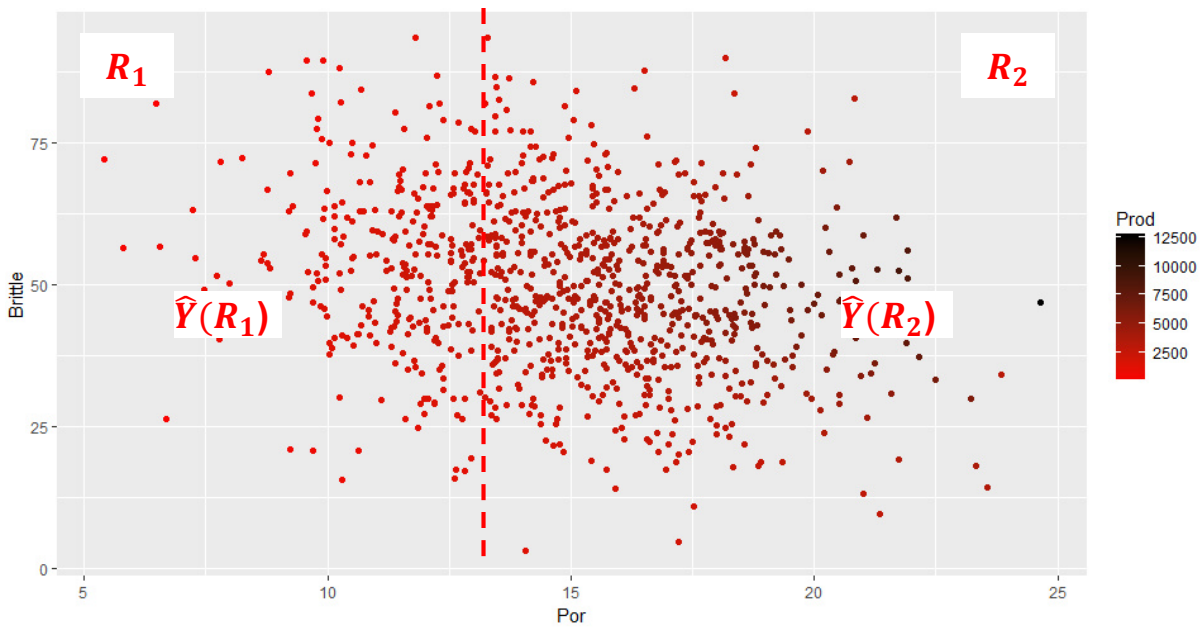
looping over J regions and data in each region, $i \in R_j$

- 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

Decision Trees
The Regions

How do we construct the Regions, $R_{1,}\, R_2\, , \ldots , R_J$?

- 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 (%)

Decision Trees
The Regions

How do we construct the Regions, $R_{1,}\, R_2\, , \ldots , R_J$?

- 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)

  - We segment such that we minimize the Residual Sum of Squares:

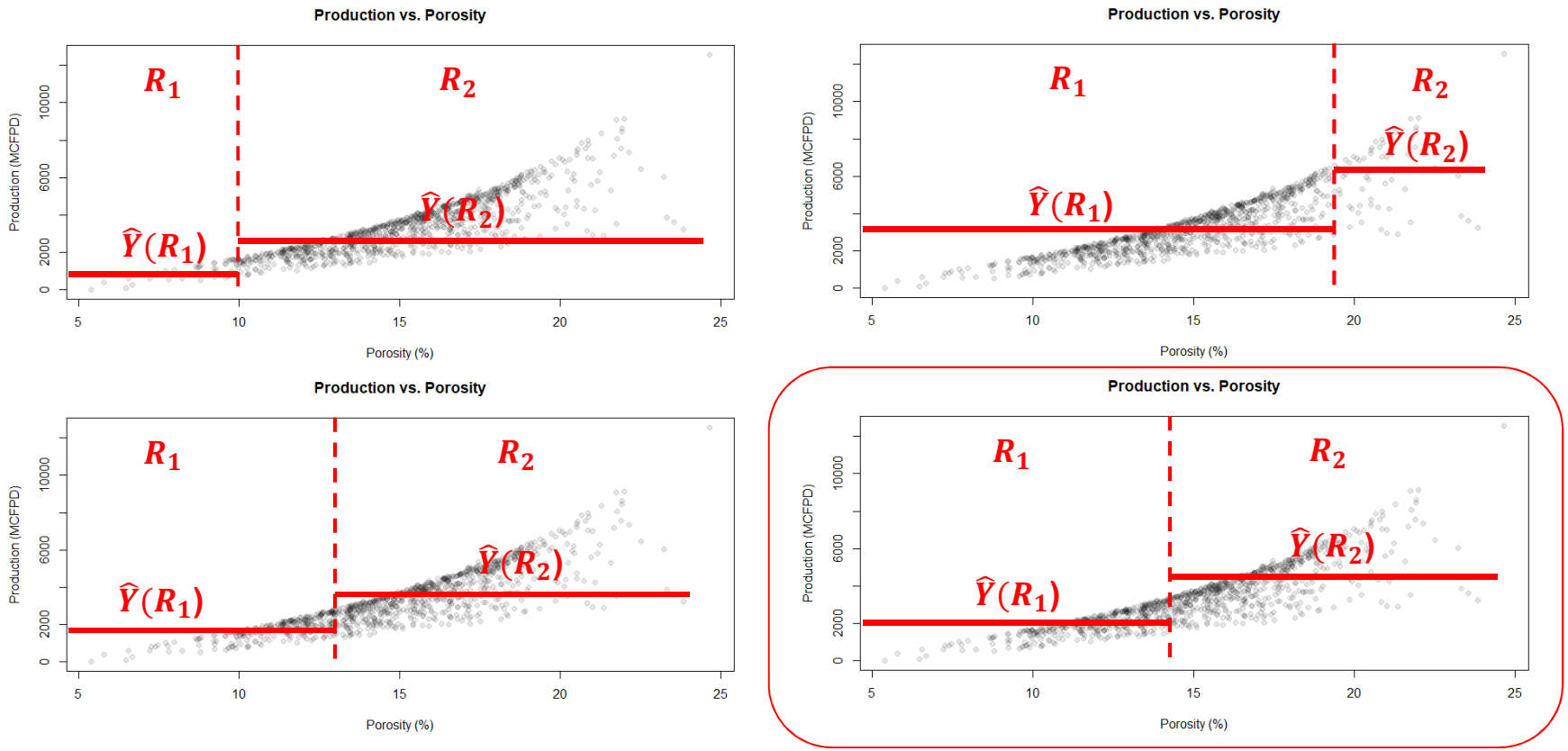
$$RSS = \sum_{i:x_i \in R_1(m,s)} (y_i - \hat{y}_{R_1})^2 + \sum_{i:x_i \in R_2(m,s)} (y_i - \hat{y}_{R_2})^2$$

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

Decision Trees
The Regions

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

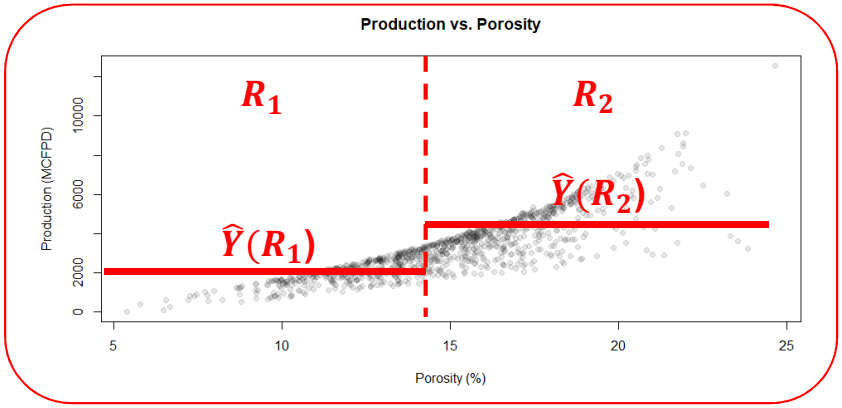
- Where should we split to minimize the error in a tree-based estimate (minimize the residual sum of square)?

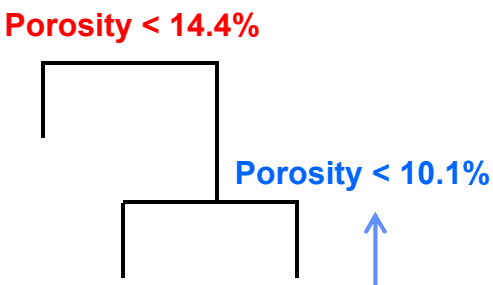


Decision Trees
The Regions

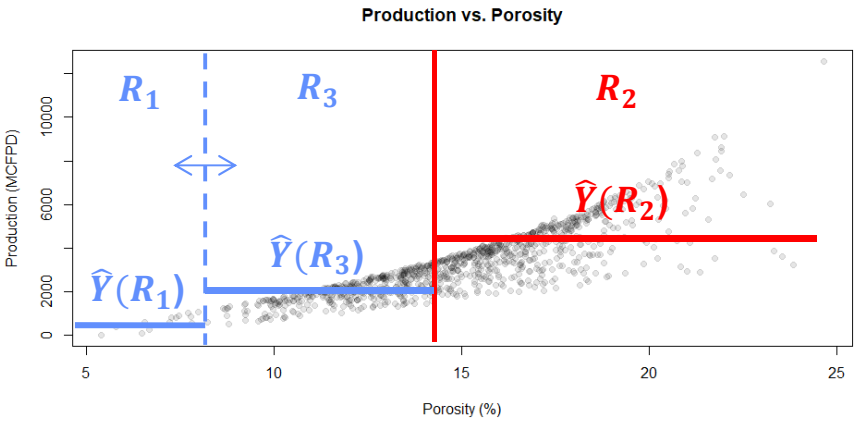
Let's pause and go back to our initial bivariate problem and make a tree by hand!

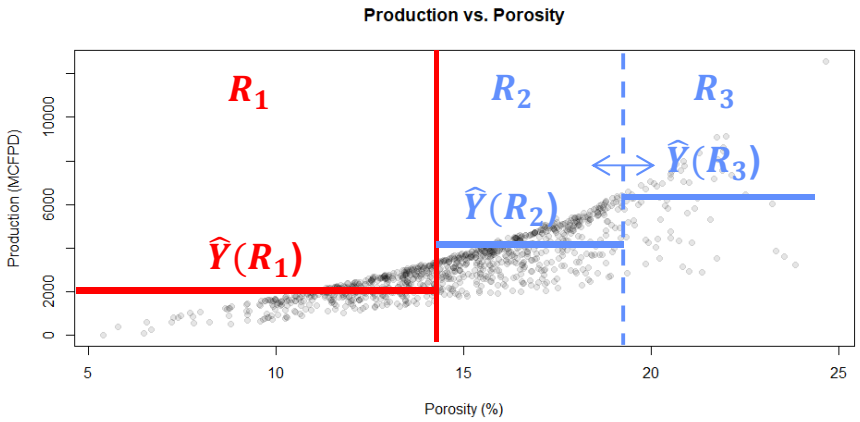
- Found first split, now check for next split the maximizes accuracy





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



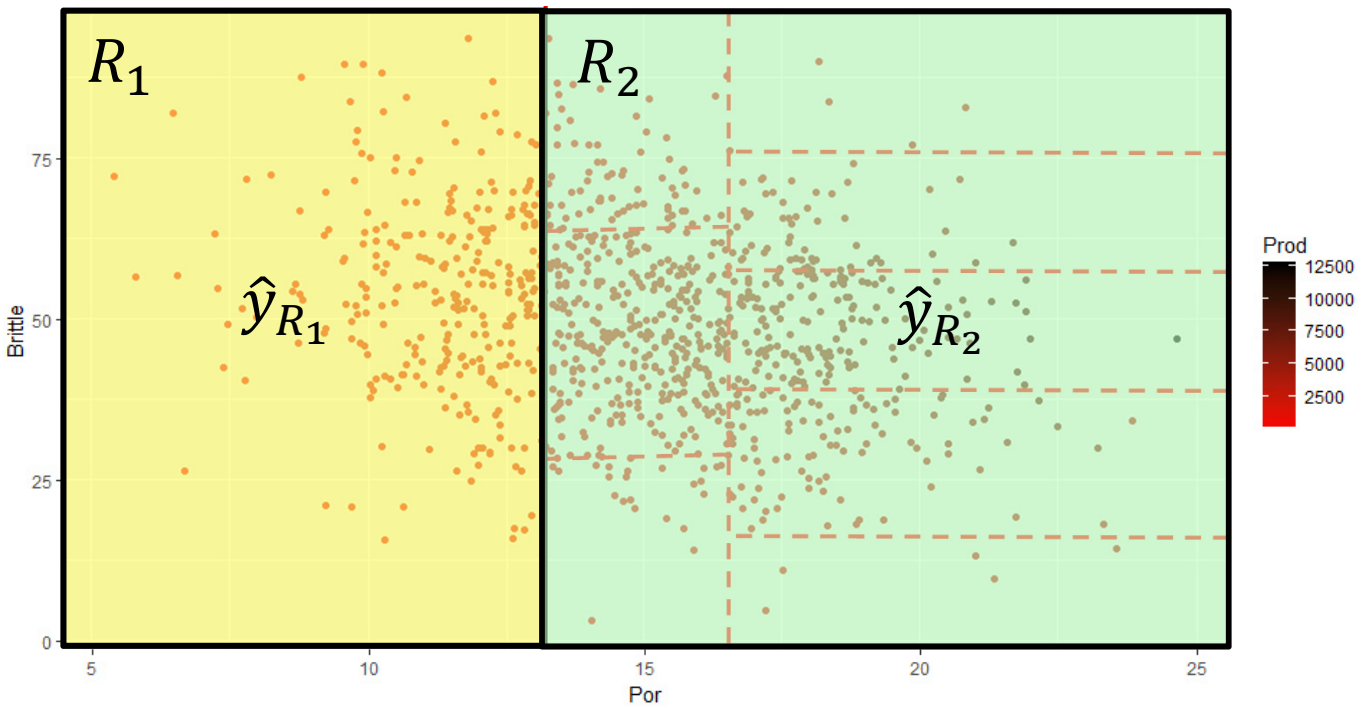


Decision Trees
The Regions

How do we construct the Regions, $R_{1,}\, R_2\,, \ldots, R_J$?

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

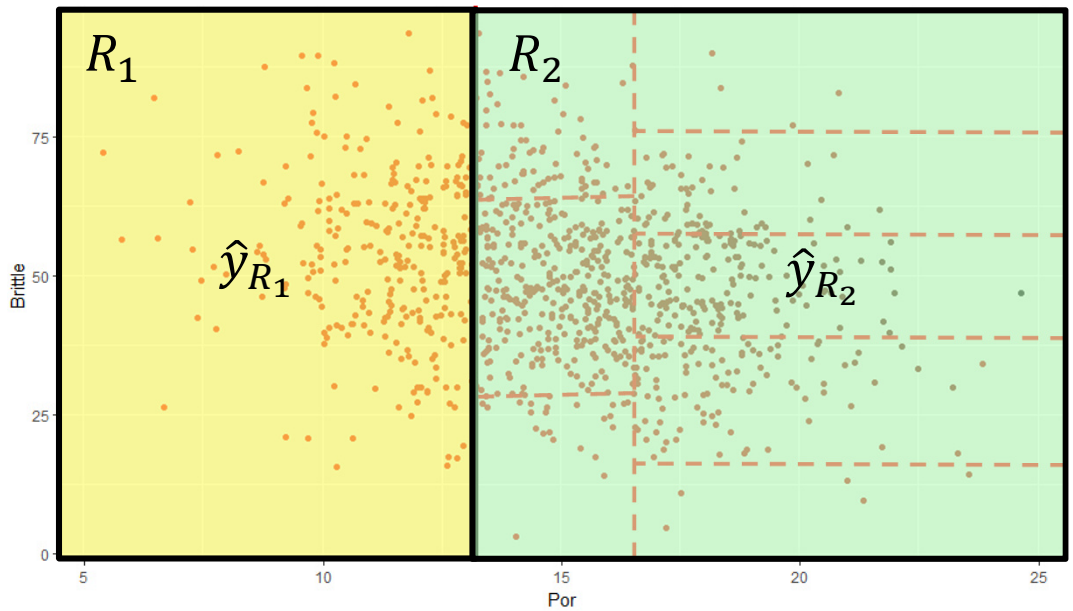
$$RSS = \sum_{i:x_i \in R_1} (y_i - \hat{y}_{R_1})^2 + \sum_{i:x_i \in R_2} (y_i - \hat{y}_{R_2})^2 + \cdots + \sum_{i:x_i \in R_J} (y_i - \hat{y}_{R_J})^2$$

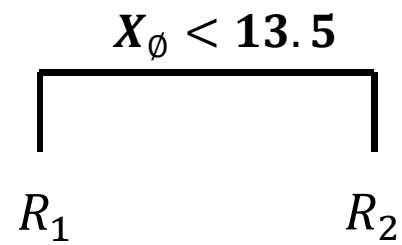


Prediction of unconventional well production (MCFPD) from porosity (%) and brittleness (%)

Decision Trees
The Regions

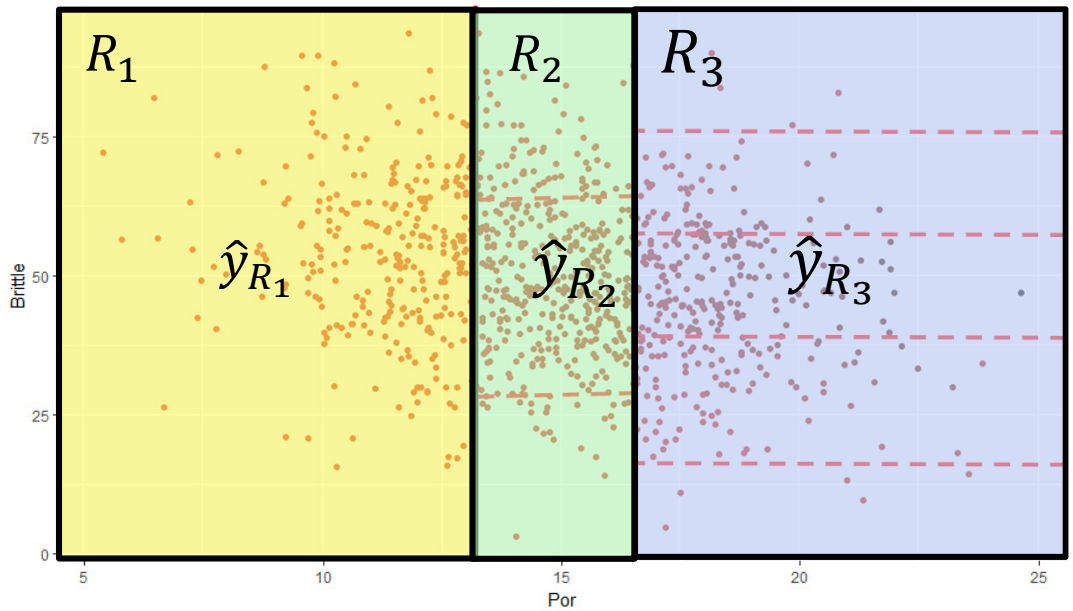
How do we construct the Regions, $R_{1,}\, R_2\,, \ldots, R_J$?

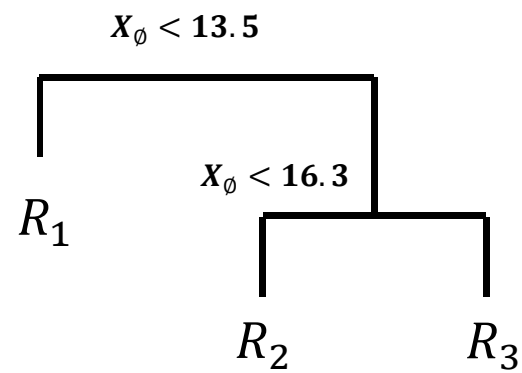




Decision Trees
The Regions

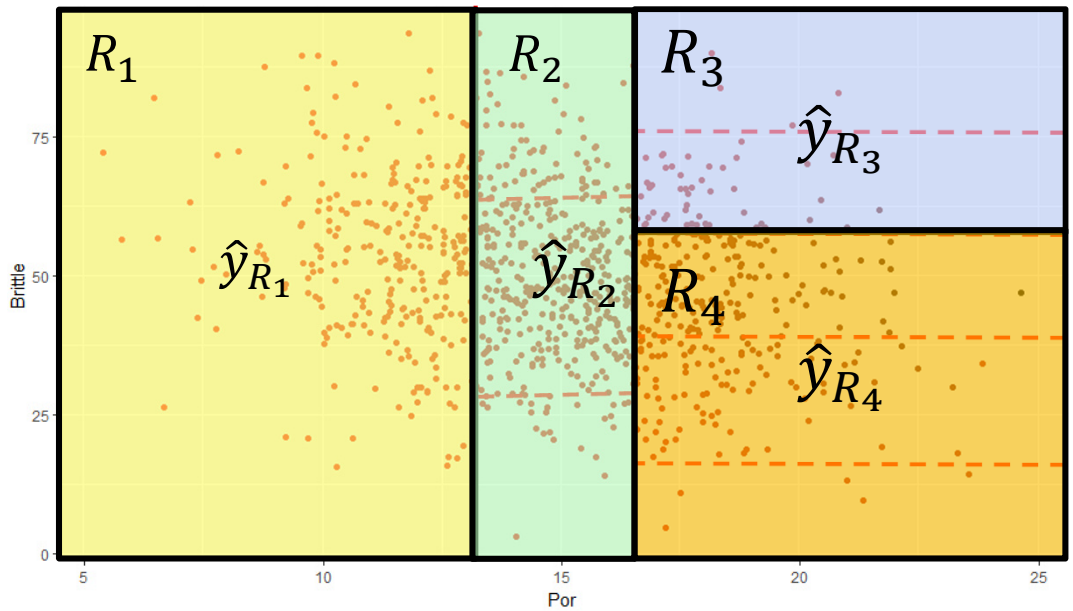
How do we construct the Regions, $R_{1,}\, R_2\, , \ldots , R_J$?

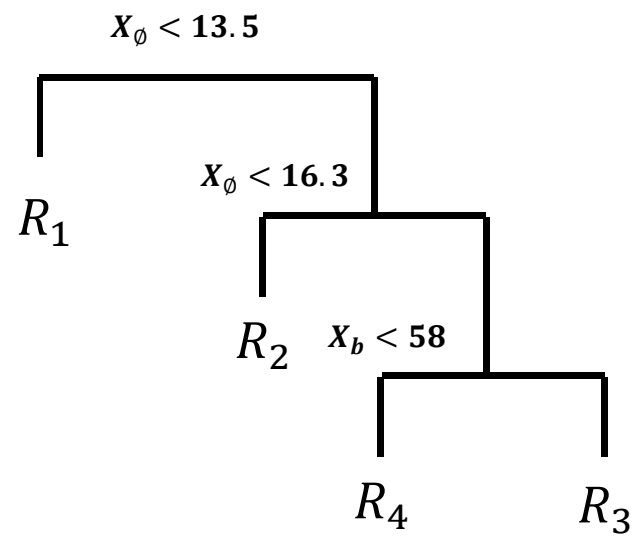




Decision Trees
The Regions

How do we construct the Regions, $R_{1,}\, R_2\, , \ldots , R_J$?



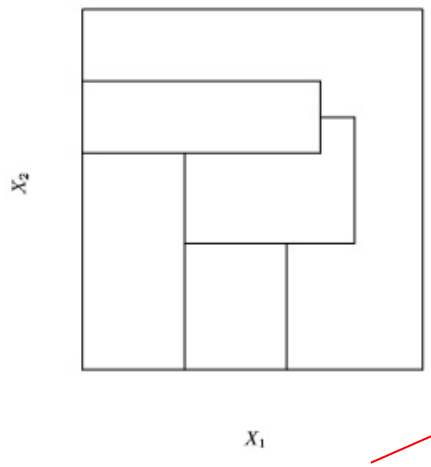


Decision Trees
The Regions

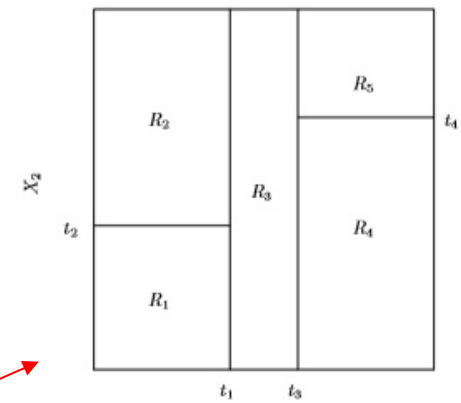
Example from James et al. (2017)

- 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.

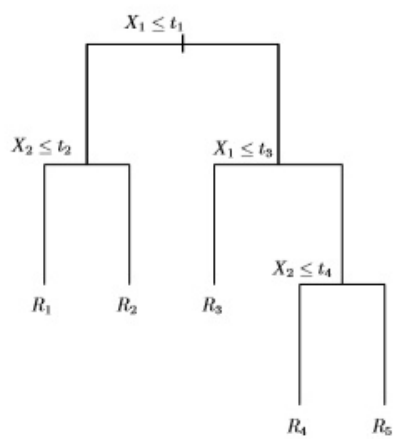
Not from recursive binary splitting



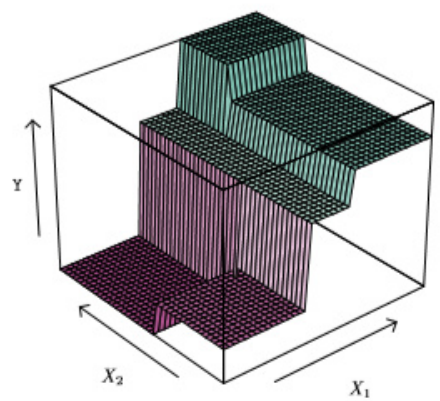
Segmented Feature Space



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 to only 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

1. Obtain the sequence of best subtrees as a function of complexity (number of terminal nodes) with training.

2. Use k-fold cross validation to chose the best complexity. Divide the training observations into K folds. For each fold, $k = 1, \ldots K$:
   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.

3. Use testing error vs. complexity to choose the best tree complexity.

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)

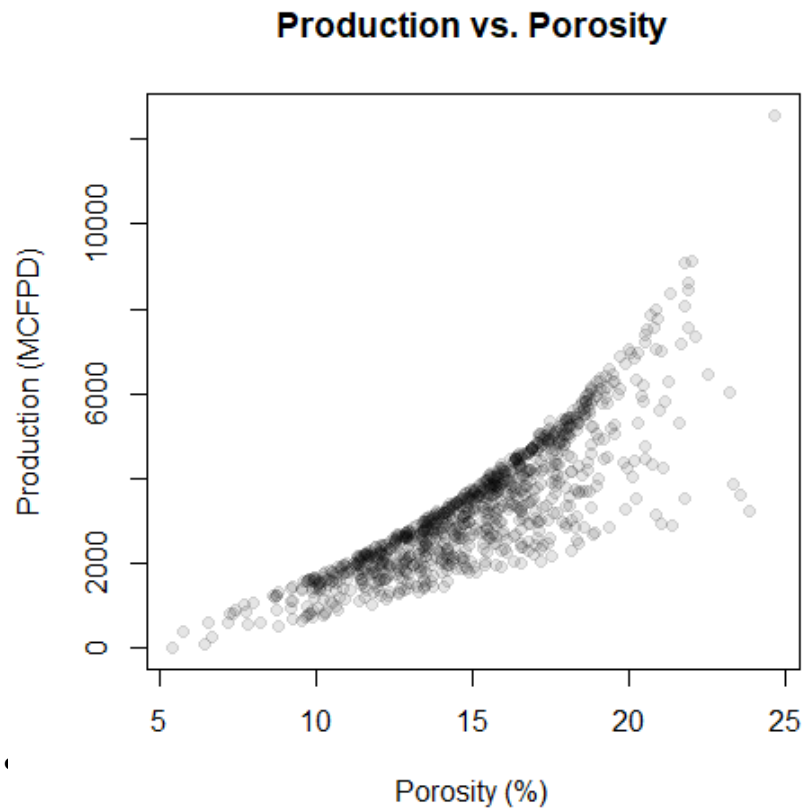
K-fold Approach

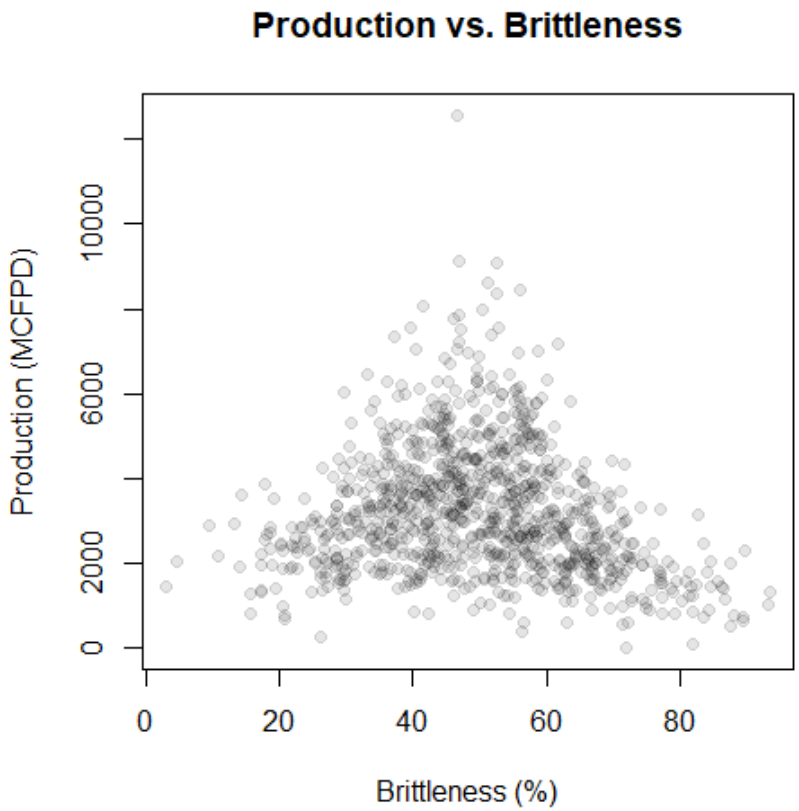
- Select K, for example
- Break data set into K subsets
- Loop over K subsets:
  - use data outside the K part to predict inside the K subset
- Average to summarize the result

Decision Trees Example

Let's use our Unconventional Multivariate Data

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

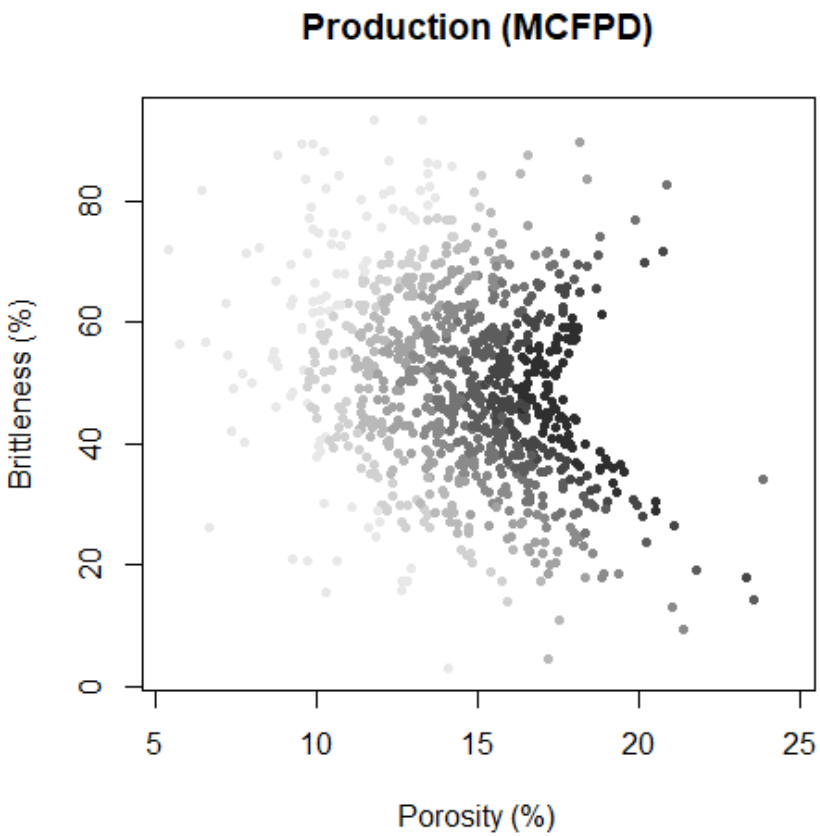




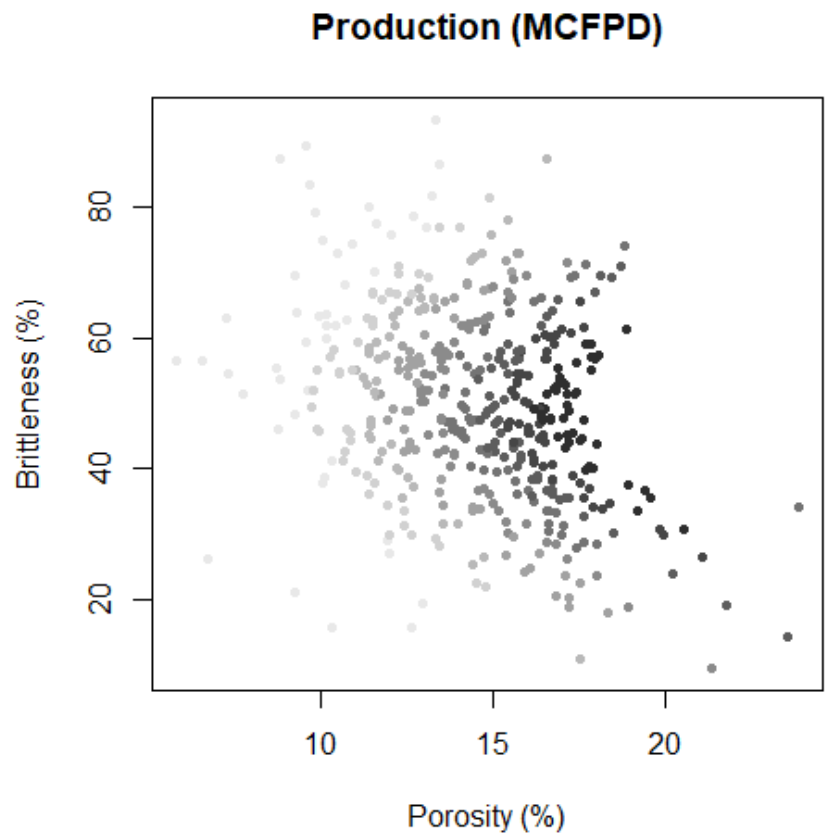
Decision Trees Example

Let's use our Unconventional Multivariate Data

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



Available data set (n=500)



Training data set (n=100)

Decision Trees Example

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:

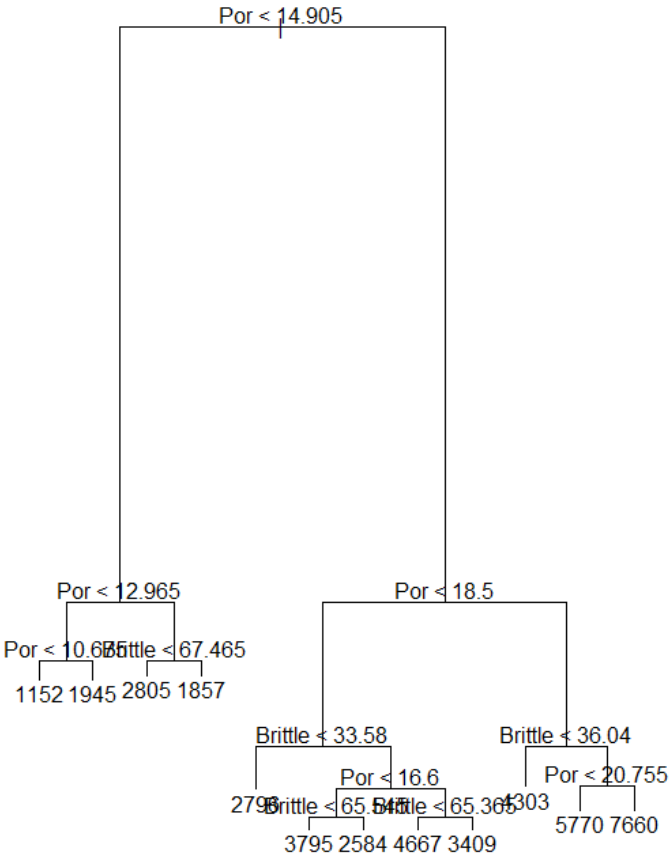
$$\mathrm{RSS} = \sum_{j=1}^{J} \sum_{i \in R_j} (y_i - \hat{y}_{R_j})^2$$

Decision Trees Example

Build the initial reasonably complicated tree

Here's the tree:

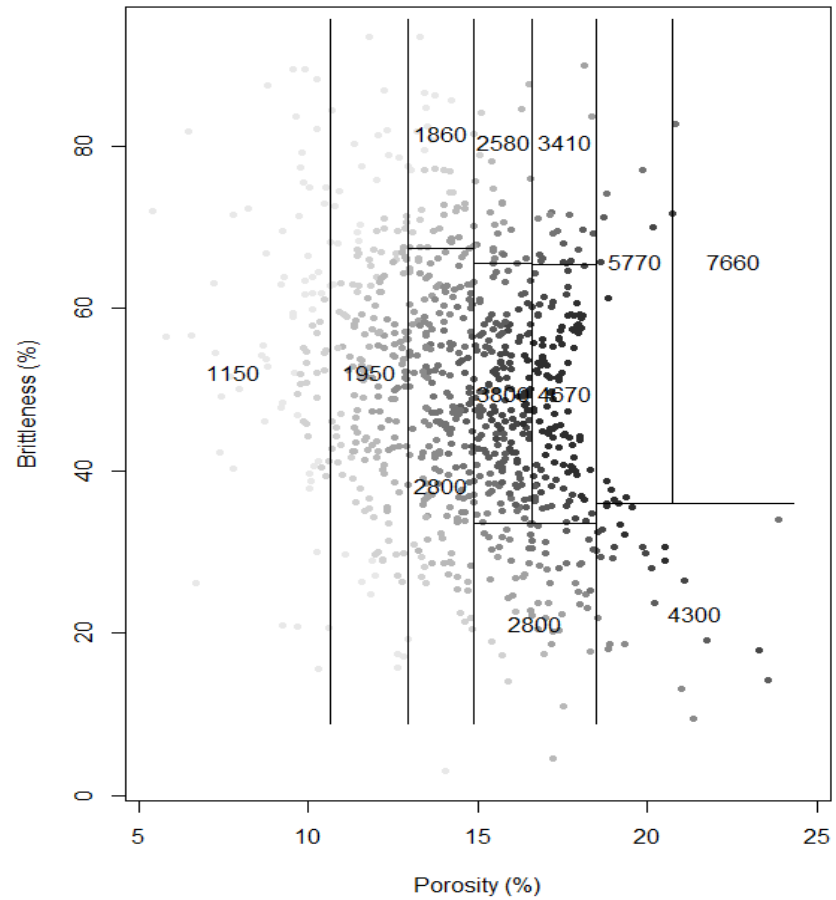
- first choice is porosity < or > 14.9%
- we get to the 3rd decision before brittleness is considered
- length of the branches is proportional to decrease in impurity
  - decreass in RSS of the model for regression tree
  - a measure of node heterogeneity for classification trees



Decision Trees Example

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.

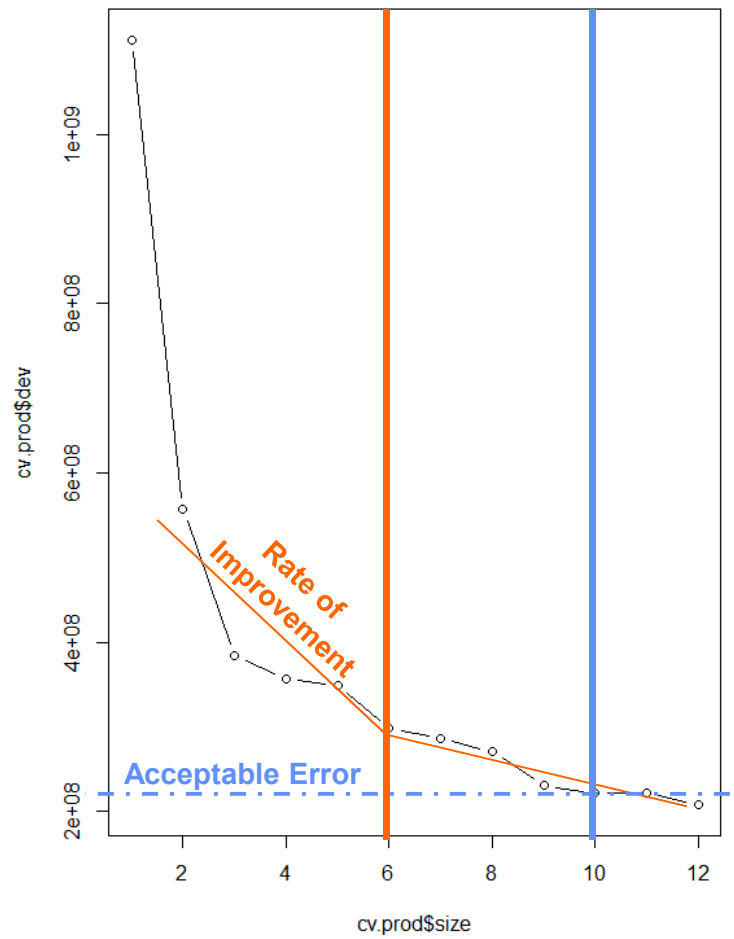


Decision Trees Example

Build the initial reasonably complicated tree

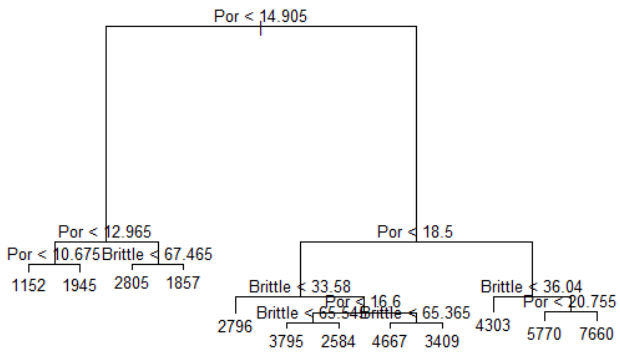
Then we perform k fold cross validation.

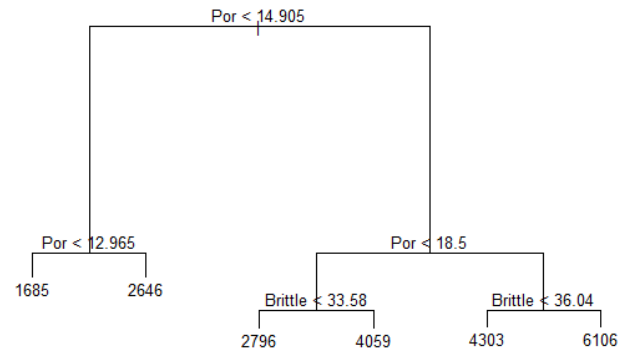
- Decrease tree complexity from 12 nodes (current model) to 1 node (uniform model)
- We can observed that each additional node improves the model
- Prune complexity based on:
  - Diminishing returns
  - Acceptable level of error

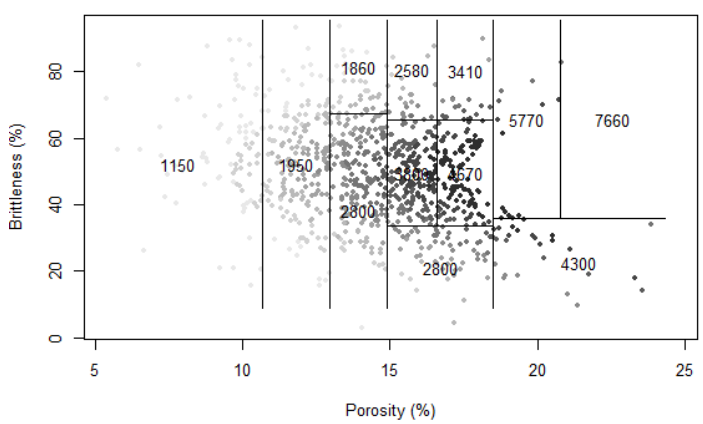


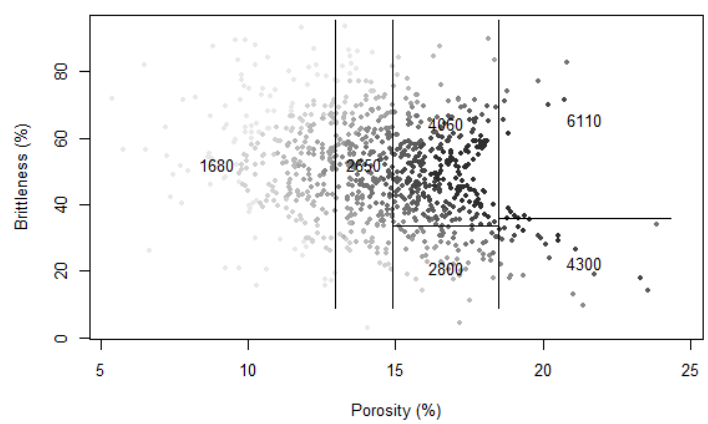
Decision Trees Example

Original and pruned tree:



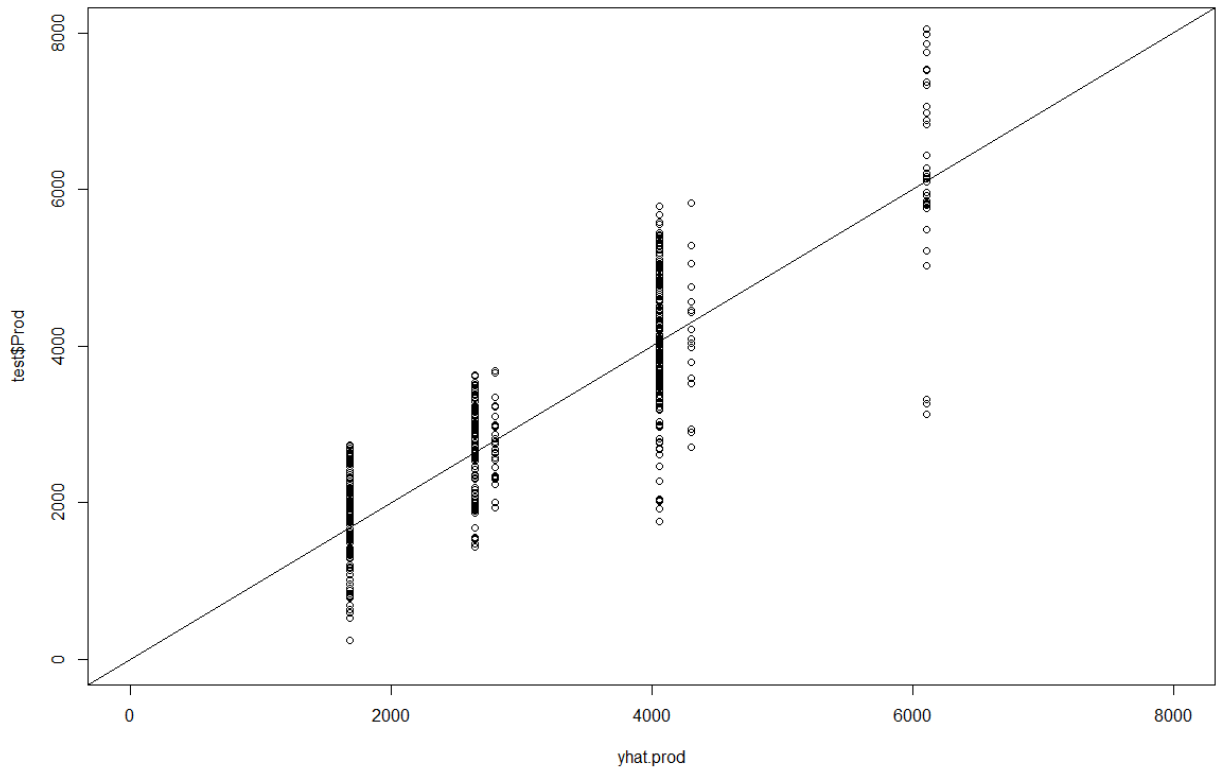






Decision Trees Example

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.

Decision Tree in Python for Engineers and Geoscientists ¶

Michael Pyrcz, Associate Professor, University of Texas at Austin

Contacts: Twitter:@GeostatsGuy | GitHub/GeostatsGuy | www.michaelpyrcz.com | Google Scholar | Book

This is a tutorial for / demonstration of building **decision trees in Python with scikit-learn**. Decision trees are one of the easiest machine learning, prediction methods to explain, apply and interrogate. 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) that is available on GitHub at https://github.com/GeostatsGuy/GeoDataSets. This dataset includes 6 predictors (features) and 1 response. We take this multivariate dataset and only retain the three variables (2 predictors and 1 response) for a simple demonstration of the decision tree method. We break the data set into 500 training data and 500 testing data. I used this tutorial in my Introduction to Geostatistics undergraduate class (PGE337 at UT Austin) as part of a first introduction to geostatistics and Python for the engineering undergraduate students. It is assumed that students have no previous Python, geostatistics nor machine learning experience; therefore, all steps of the code and workflow are explored and described. This tutorial is augmented with course notes in my class. The Python code and markdown was developed and tested in Jupyter.

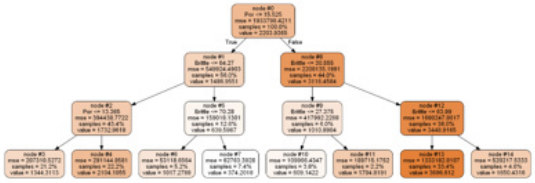
What is a decision tree?

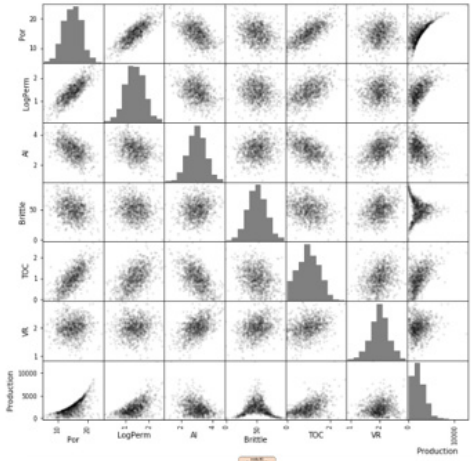
I'll make a couple of points about decision trees. For greater detail there are a lot of online resources on decision trees along with the book "An Introduction to Statistical Learning" by James et al. (my favourite).

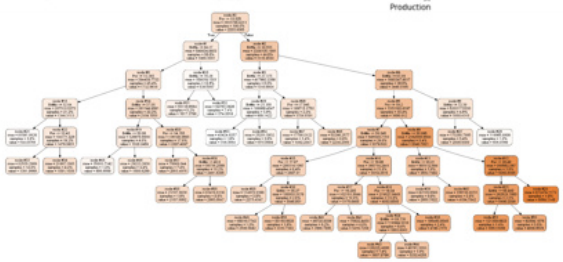
1. method for supervised learning
2. categorical prediction with a classification tree and continuous prediction with a regression tree
3. fundamental idea is to divide feature space into exhaustive, mutually exclusive regions (terminal or leaf nodes in the tree)
4. estimate with the average of data in each region for continuous prediction or the majority category for the data in each region for categorical prediction
5. segment the feature space with hierarchical, binary splitting that may be respresented as a decision tree
6. apply a greedy method to find the sequential splits for any feature that minimizes the residual sum of squares

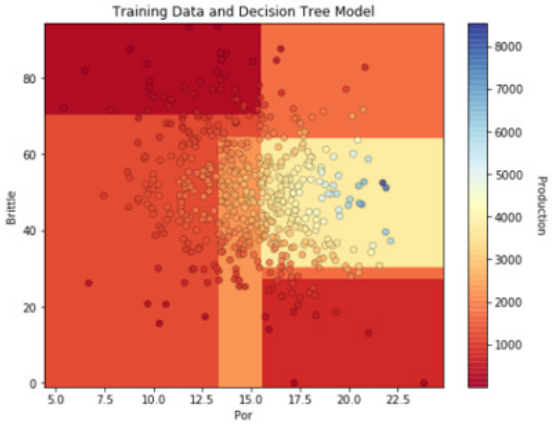
Let's build some decision trees together. You'll get a chance to see the trees and the divided feature space graphically.

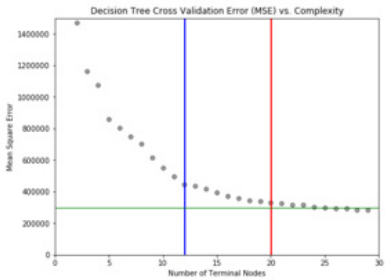
	count	mean	std	min	25%	50%	75%	max
Por	1000.0	14.950460	3.029634	5.400000	12.85750	14.98500	17.080000	24.65000
LogPerm	1000.0	1.398880	0.405986	0.120000	1.13000	1.39000	1.680000	2.58000
AI	1000.0	2.982610	0.577629	0.960000	2.57750	3.01000	3.360000	4.70000
Brittle	1000.0	49.719480	15.077006	-10.500000	39.72250	49.68000	59.170000	93.47000
TOC	1000.0	1.003810	0.504978	-0.260000	0.64000	0.99500	1.360000	2.71000
VR	1000.0	1.991170	0.308194	0.900000	1.81000	2.00000	2.172500	2.90000
Production	1000.0	2247.295809	1464.256312	2.713535	1191.36956	1976.48782	3023.594214	12568.64413











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

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.

Statistical Learning New Tools

Topic	Application to Subsurface Modeling
Principles of Statistical / Machine Learning	Garbage in, garbage out, importance of good data and domain expertise, training with data, testing to tune hyperparameters, simpler mode may be more accurate in testing. *Use training, testing and tuning to build models with good data.*
Overfit	Be careful about fitting very complicated models. *Is the complexity needed and supported by the data and interpretations.*
Decision Tree and Ensemble Tree Methods	Flexible, efficient non-parametric method to predict with multivariate datasets. *Use tree and ensemble tree methods for flexible and highly interpretable models.*

Subsurface Data Analytics and Machine Learning
Predictive Models

Lecture outline . . .

- Prediction
- A Simple Parametric Model
- Decision Tree

Introduction

Data Analytics

Inferential Methods

Predictive Methods

Advanced Methods

Conclusions

Instructor: Michael Pyrcz, the University of Texas at Austin