RANDOM FORESTS FOR BEGINNERS

Leverage the power of multiple alternative analyses, randomization strategies, and ensemble learning.

A publication of

Salford Systems and RandomForests

Salford Systems has been working with the world's leading data mining researchers at UC Berkeley and Stanford since 1990 to deliver best-of-breed machine learning and predictive analytics software and solutions. Our powerful, easy to learn and easy to use tools have been successfully deployed in all areas of data analytics. Applications number in the thousands and include on-line targeted marketing, internet and credit card fraud detection, text analytics, credit risk and insurance risk, large scale retail sales prediction, novel segmentation methods, biological and medical research, and manufacturing quality control.

FOLLOW US ON TWITTER
@SALFORDSYSTEMS

CONTENTS

CHAPTER 1

What are Random Forests?

Random Forests are one of the most powerful, fully automated, machine learning techniques. With almost no data preparation or modeling expertise, analysts can effortlessly obtain surprisingly effective models. “Random Forests” is an essential component in the modern data scientist’s toolkit and in this brief overview we touch on the essentials of this groundbreaking methodology.

Prerequisites

RandomForests are constructed from decision trees and it is thus recommended that the user of this brief guide be familiar with this fundamental machine learning technology.

If you are not familiar with decision trees we suggest that you visit our website to review some of our introductory materials on the CART (Classification and Regression Trees). You do not need to be a master of decision trees or CART to follow the current guide, but some basic understanding will help make the discussion here much more understandable.

You should also know more-or-less what a predictive model is and how data is typically organized for analysis with such models.

Leo Breiman

Random Forests was originally developed by UC Berkeley visionary Leo Breiman in a paper he published in 1999, building on a lifetime of influential contributions including the CART decision tree. As he continued to perfect Random Forests he worked with his long time collaborator and former Ph.D. student Adele Cutler to develop the final form of RandomForests including the sophisticated graphics permitting deeper data understanding.

Adele Cutler

Random Forests is a tool that leverages the power of many decision trees, judicious randomization, and ensemble learning to produce astonishingly accurate predictive models, insightful variable importance rankings, missing value imputations, novel segmentations, and laser-sharp reporting on a record-by-record basis for deep data understanding.

Preliminaries

- We start with a suitable collection of data including variables we would like to predict or understand and relevant predictors
- Random Forests can be used to predict continuous variables such as sales of a product on a web site or the loss predicted if an insured files a claim
- Random Forests can also be used to estimate the probability that a particular outcome occurs
- Outcomes can be “yes/no” events or one of several possibilities, such as which model of cell phone a customer will buy
- Could have many possible outcomes but typically multi-class problems have 8 or fewer outcomes

The Essentials

- Random Forests are collections of decision trees that together produce predictions and deep insights into the structure of data
- The core building block of a Random Forest is a CART® inspired decision tree.
- Leo Breiman's earliest version of the random forest was the "bagger"
- Imagine drawing a random sample from your main data base and building a decision tree on this random sample
- This "sample" typically would use half of the available data although it could be a different fraction of the master data base

More Essentials

- Now repeat the process. Draw a second different random sample and grow a second decision tree.
- The predictions made by this second decision tree will typically be different (at least a little) than those of the first tree.
- Continue generating more trees each built on a slightly different sample and generating at least slightly different predictions each time
- This process could be continued indefinitely but we typically grow 200 to 500 trees

Predictions

- Each of our trees will generate its own specific predictions for each record in the data base
- To combine all of these separate predictions we can use either averaging or voting
- For predicting an item such as sales volume we would average the predictions made by our trees
- To predict a classification outcome such as “click/no-click” we can collect counts of votes. How many trees voted “click” vs. how many “no-click” will determine prediction.
- For classification we can also produce a predicted probability of each possible outcome based on relative share of votes for each outcome

Weakness of Bagger

- The process just described is known as the “bagger”. There are many details we have omitted but we have presented the essentials.
- The bagger represented a distinct advance in machine learning when it was first introduced in 1994
- Breiman discovered that the bagger was a good machine learning method but not as accurate as he had hoped for.
- Analyzing the details of many models he came to the conclusion that the trees in the bagger were too similar to each other
- His repair was to find a way to make the trees dramatically more different

Putting Randomness into Random Forests

- Breiman's key new idea was to introduce randomness not just into the training samples but also into the actual tree growing as well
- In growing a decision tree we normally conduct exhaustive searches across all possible predictors to find the best possible partition of data in each node of the tree
- Suppose that instead of always picking the best splitter we picked the splitter at random
- This would guarantee that different trees would be quite dissimilar to each other

How Random Should a Random Forest Be?

- At one extreme, if we pick every splitter at random we obtain randomness everywhere in the tree
- This usually does not perform very well
- A less extreme method is to first select a subset of candidate predictors at random and then produce the split by selecting the best splitter actually available
- If we had 1,000 predictors we might select a random set of 30 in each node and then split using the best predictor among the 30 available instead of the best among the full 1,000

More on Random Splitting

- Beginners often assume that we select a random subset of predictors once at the start of the analysis and then grow the whole tree using this subset
- This is not how RandomForests work
- In RandomForests we select a new random subset of predictors in *each* *node* of a tree
- Completely different subset of predictors may be considered in different nodes
- If the tree grows large then by the end of the process a rather large number of predictors have had a chance to influence the tree

Controlling Degree of Randomness

- If we always search *all* predictors in every node of every tree we are building bagger models-- typically not so impressive in their performance
- Models will usually improve if we search fewer than all the variables in each node restricting attention to a random subset
- How many variables to consider is a key control and we need to experiment to learn the best value
- Breiman suggested starting with the square root of the number of available predictors
- Allowing just one variable to be searched in each node almost always yields inferior results but often allowing 2 or 3 instead can yield impressive results

How Many Predictors in Each Node?

N Predictors	sqrt	.5 sqrt	2*sqrt	ln2
100	10	5	20	6
1,000	31	15.5	62	9
10,000	100	50	200	13
100,000	316	158	632	16
1,000,000	1000	500	2000	19

In the table above we show some of the values Breiman and Cutler advised. They suggested four possible rules: square root of the total number of predictors, or one half or twice the square root, and log base 2.

We recommend experimenting with some other values. The value chosen remains in effect for the entire forest and remains the same for every node of every tree grown

Random Forests Predictions

- Predictions will be generated for a forest as we did for the bagger, that is, by averaging or voting
- If you want you can obtain the prediction made by each tree and save it to a database or spreadsheet
- You could then create your own custom weighted averages or make use of the variability of the individual tree predictions
- For example, a record that is predicted to show a sales volume in a relatively narrow range across all trees is less uncertain than one that has the same average prediction but a large variation in individual tree predictions
- It is easiest to just let RandomForests do the work for you and save final predictions

Out of Bag (OOB) Data

- If we sample from our available training data before growing a tree then we automatically have holdout data available (for that tree)
- In Random Forests this holdout data is known as “Out Of Bag” data
- There is no need to be concerned about the rationale for this terminology at this point
- Every tree we grow has a different holdout sample associated with it because every tree has a different training sample
- Alternatively, every record in the master data base will be “in bag” for some trees (used to train that tree) and “out of bag” for other trees (not used to grow the tree)

Testing and Evaluating

- Keeping track of for which trees a specific record was OOB allows us to easily and effectively evaluate forest performance
- Suppose that a given record was in-bag for 250 trees and out-of-bag for another 250 trees
- We could generate predictions for this specific record using just the OOB trees
- The result would give us an honest assessment of the reliability of the forest since the record was never used to generate any of the 250 trees
- Always having OOB data means that we can effectively work with relatively small counts of records

More Testing and Evaluation

- We can use the OOB idea for every record in the data
- Note that every record is evaluated on its own specific subset of OOB trees and typically no two records would share the identical pattern of in-bag versus out-of-bag trees
- We could always reserve some additional data as a traditional holdout sample but this is not necessary for Random Forests
- The idea of OOB testing is an essential component of Random Forest data analytics

Testing vs.. Scoring

- For model assessment using OOB data we use a subset of trees (the OOB trees) to make predictions for each record
- When forecasting or scoring new data we would make use of every tree in the forest as no tree would have been built using the new data
- Typically this means that scoring yields better performance than indicated by the internal OOB results
- The reason is that in scoring we can leverage the full forest and thus benefit from averaging the predictions of a much larger number of trees

Random Forests and Segmentation

- Another essential idea in Random Forests analysis is that of “proximity” or the closeness of one data record to another
- Consider two records of data selected from our data base. We would like to know how similar these records are to each other
- Drop the pair of records down each tree and note if the two end up in the same terminal node or not
- Count the number of times the records “match” and divide by the number of trees tested

Proximity Matrix

- We can count the number of matches found in this way for every pair of records in the data
- This produces a possibly very large matrix. A 1,000 record data base would produce a 1,000 x 1,000 matrix with 1 million elements
- Each entry in the matrix displays how close two data records are to each other
- Need to keep the size of this matrix in mind if we wish to leverage the insights into the data it provides
- To keep our measurements honest we can be selective in how we use the trees. Instead of using every tree for every pair of records we could use only trees in which one or both records are OOB
- This does not affect matrix size but affects the reliability of the proximity measures

Characteristics of the Proximity Matrix

- The Random Forests proximity matrix has some important advantages over traditional near neighbor measurements
- Random Forests naturally handle mixtures of continuous and categorical data.
- There is no need to come up with a measurement of nearness that applies to a specific variable. The forest works with all variables together to measure nearness or distance directly
- Missing values are also no problem as they are handled automatically in tree construction

Proximity Insights

- Breiman and Cutler made use of the proximity matrix in various ways
- One use is in the identification of “outliers”
- An outlier is a data value that is noticeably different than we would expect given all other relevant information
- As such an outlier would be distant from records that to which we would expect it to be close
- We would expect records that are “events” to be closer to other “events” than to “non-events”
- Records that do not have any appropriate near neighbors are natural candidates to be outliers
- RandomForests produces an “outlier score” for each record

Proximity Visualization

- Ideally we would like to plot the records in our data to reveal clusters and outliers
- Might also have a cluster of outliers which would be best detected visually
- In Random Forests we do this by plotting projections of the proximity matrix to a 3D approximation
- These graphs can suggest how many clusters appear naturally in the data (at least if there are only a few)
- We show such a graph later in these notes

Missing Values

- Classic Random Forests offer two approaches to missing values
- The simple method and default is to just fill in the missing values with overall sample means or most common values for categorical predictors
- In this approach, for example, all records with missing AGE would be filled in with the same average value
- While crude, the simple method works surprisingly well due to the enormous amount of randomization and averaging associated with any forest

Proximity and Missing Values

- The second “advanced” way to deal with missing values involves several repetitions of forest building
- We start with the simple method and produce the proximity matrix
- We then replace the simple imputations in the data with new imputations
- Instead of using unweighted averages to calculate the imputation we weight the data by proximity
- To impute a missing value for X for a specific record we essentially look at the good values of X among the records closest to the one with a missing value
- Each record of data could thus obtain a unique imputed value

Missing Imputation

- The advanced method is actually very common sense
- Suppose we are missing the age of a specific customer
- We use the forest to identify how close the record in question is to all other records
- Impute by producing a weighted average of the ages of other customers but with greatest weight on those customers most “like” the one needing imputation
- In the upcoming April 2014 version of SPM you can save these imputed values to a new data set

Variable Importance

- Random Forests include an innovative method to measure the relative importance of any predictor
- Method is based on measuring the damage that would be done to our predictive models if we lost access to true values of a given variable
- To simulate losing access to a predictor we randomly scramble its values in the data. That is, we move the value belonging to a specific row of data to another row
- We scramble just one predictor at a time and measure the consequential loss in predictive accuracy

Variable Importance Details

- If we scrambled the values of a variable just once and then measured the damage done to predictive performance we would be reliant on a single randomization
- In Random Forests we re-scramble the data anew for the predictor being tested in every tree in the forest
- We therefore free ourselves from dependence on the luck of single draw. If we re-scramble a predictor 500 times in front of 500 trees the results should be highly reliable

Variable Importance Issues

- If our data includes several alternative measures of the same concept then scrambling just one of these at a time might result in very little damage to model performance
- For example, if we have several credit risk scores, we might be fooled into thinking a single one of them is unimportant
- Repeating the scrambling test separately for each credit score could yield the conclusion that each considered separately is unimportant
- It may thus be important to eliminate this kind of redundancy in the predictors used before putting too much faith in the importance rankings

Variable Importance: Final Observation

- The data scrambling approach to measuring variable importance is based on the impact of losing access to information on model performance
- But a variable is not necessarily unimportant just because we can do well without it
- Need to be aware that a predictor, if available, will be used by models, but if not available, then substitute variables can be used instead
- The “gini” measure is based on the actual role of a predictor and offers an alternative importance assessment based on the role the predictor plays in the data

Bootstrap Resampling

- In our discussion so far we have suggested that the sampling technique underpinning RandomForests is the drawing of a random 50% of the available data for each tree
- This style of sampling is very easy to understand and is a reasonable way to develop a Random Forest
- Technically, RandomForests uses a somewhat more complicated method known as bootstrap resampling
- However, bootstrap sampling and random half sampling are similar enough that we do not need to delve into the details here
- Please consult our training materials for further technical details

The Technical Algorithm:

- Let the number of training cases be *N*, and the number of variables in the classifier be *M*.
- We are told the number *m* of input variables to be used to determine the decision at a node of the tree; *m* should be less and even much less than *M*.
- Choose a training set for this tree by choosing *n* times with replacement from all *N* available training cases (i.e., take a bootstrap sample). Use the rest of the cases to estimate the error of the tree, by predicting their classes (OOB data)
- For each node of the tree, randomly choose *m* variables on which to base the decision at that node. Calculate the best split based on these *m* variables in the training set.
- Each tree is fully grown and not pruned (as may be done in constructing a normal tree classifier).
- For prediction a new sample is pushed down the tree. It is assigned the label of the training sample in the terminal node it ends up in. This procedure is iterated over all trees in the ensemble, and the mode vote of all trees is reported as the random forest prediction.

CHAPTER 2

SUITED FOR WIDE DATA

Text analytics, online behavioral prediction, social network analysis, and biomedical research may all have access to tens of thousands of predictors. RandomForests can be ideal for analyzing such data efficiently.

Wide Data

- Wide data is data with a large number of available predictors numbering in the tens of thousands, hundreds of thousands, or even millions
- Wide data are often encountered in text mining where each word or phrase found in a corpus of documents is represented by a predictor in the data
- Wide data is also encountered in social network analysis, on-line behavioral modeling, chemistry, and many types of genetic research
- Statisticians often refer to data as “wide” if the number of predictors far exceeds the number of data records

RandomForests and Wide Data

- Suppose we have access to 100,000 predictors and we build a Random Forest searching over 317 randomly selected predictors at each node in each tree
- In any one node of any tree we reduce the workload of tree building by more than 99%
- Experience shows that such forests can be predictively accurate while also generating reliable predictor importance rankings
- RandomForests may be the ideal tool for analysis of wide data for the computational savings alone
- RandomForests is sometimes used as a predictor selection technique to radically reduce the number of predictors we ultimately need to consider

Wide Shallow Data

- In wide shallow data we are faced with a great many columns and relatively few rows of data
- Imagine a data base with 2,000 rows and 500,000 columns
- Here RandomForests can be effective not only in extracting the relevant predictors but also in clustering
- The proximity matrix will only be 2000 x 2000 regardless of the number of predictors

CHAPTER 3

STRENGTHS & WEAKNESSES OF RANDOM FORESTS

RandomForests has remarkably few controls to learn and is easily parallelizable. But the size of the model may far exceed the size of the data it is designed to analyze.

Random Forests: Few Controls to Learn & Set

- RandomForests has very few controls
- Most important: number of predictors to consider when splitting a node
- Number of trees to build
- For classification problems we obtain the best results if we grow each tree to its maximum possible size
- For predicting continuous targets might need to limit how small a terminal node may become effectively limiting the sizes of the trees

Easily Parallelizable

- Random Forests is an ensemble of independently built decision trees
- No tree in the ensemble depends in any way on any other tree
- Therefore, trees could be grown on different computers (just need to work with the same master data)
- Different trees could also be grown on different cores on the same computer
- Allows for ultra-fast analysis
- Scoring can also be parallelized in the same way

RandomForests Weaknesses

- RandomForests models perform best when the trees are grown to a very large size
- A crude rule of thumb is that if you have N training records you can expect to grow a tree with N/2 terminal nodes
- 1 million training records thus tend to generate trees with 500,000 terminal nodes
- 500 such trees yields 250 million terminal nodes and 500 million nodes in total
- Every node needs to be managed in a deployed model

Therefore…

Random Forests is well suited for the analysis of complex data structures embedded in datasets **containing potentially millions of columns but only a *moderate* number of rows.**

We recommend other tools such as TreeNet for mu larger data bases .

CHAPTER 4

SIMPLE EXAMPLE:

Boston Housing data predicting above average housing values

Boston Housing Data

506 Census Tracts in greater Boston Area with quality of life data and median housing value for each tract

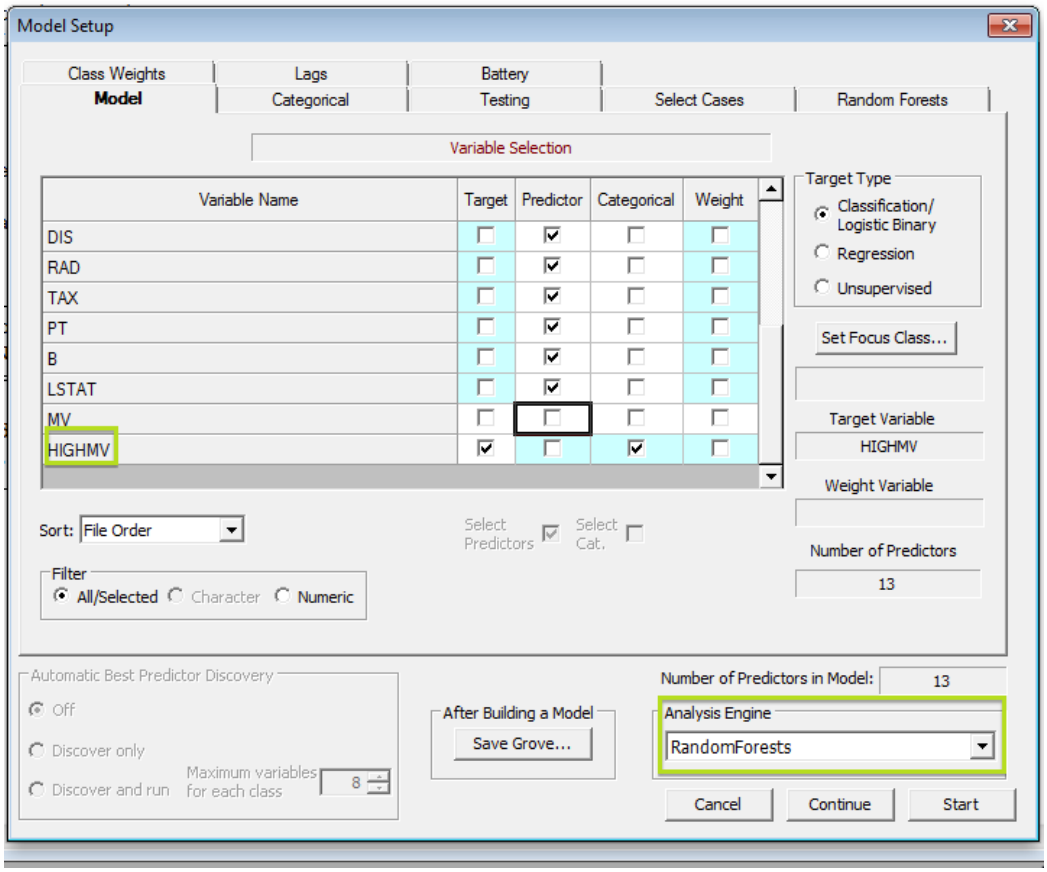
Usually the subject of regression analysis but here we create a binary indicator with tracts with median values above 23 coded as 1 and the remainder coded as 0

Predictors include FBI official crime statistics, socioeconomic level of residents, air pollution, distance from major employment centers, zoning for commercial or industrial use and a few others

We describe this data in detail in our training videos

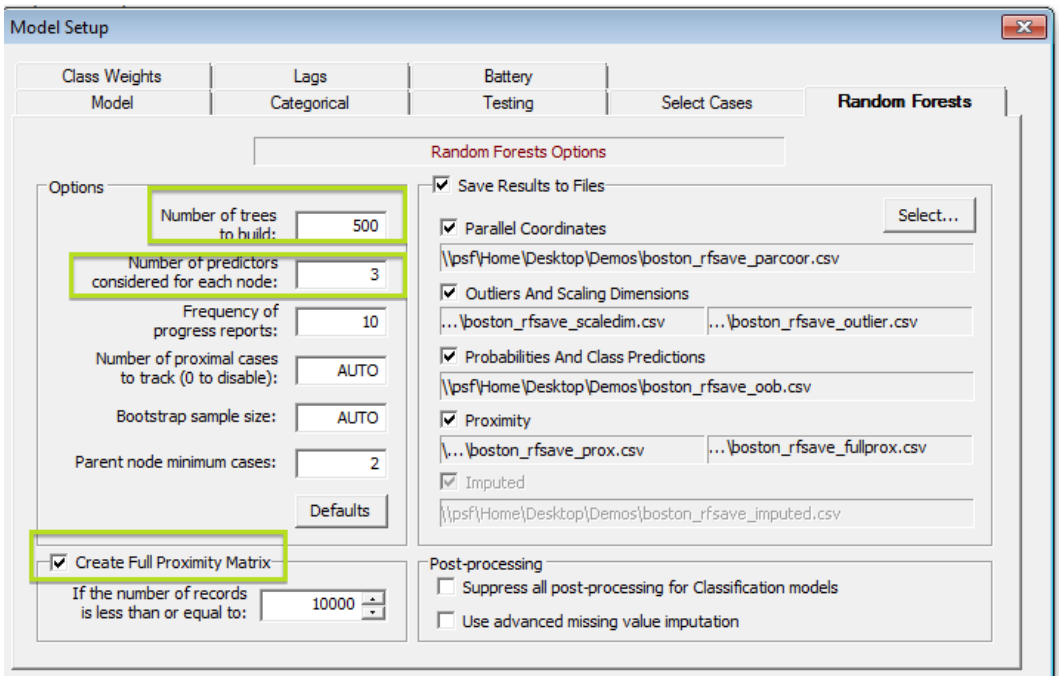
Run an RF model

In the screen shot below we set up the RF model selecting the target, the legal predictors and the analysis engine



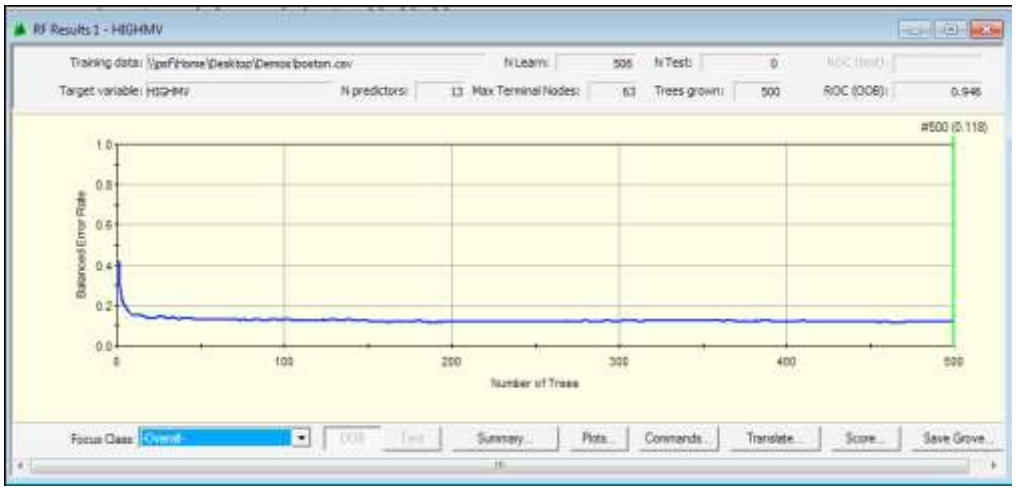
RF Controls

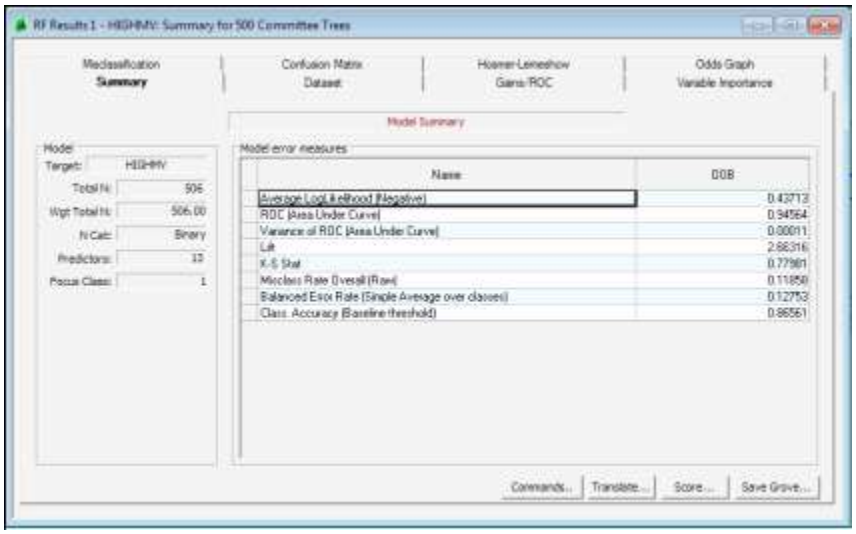
The essential controls govern the number of trees, the number of predictors to us in each node, and whether we will devote the potentially large computer resources to post-process the forest



Results Summary

Performance Overview (OOB) and access to many detailed reports and graphs





Confusion Matrix OOB

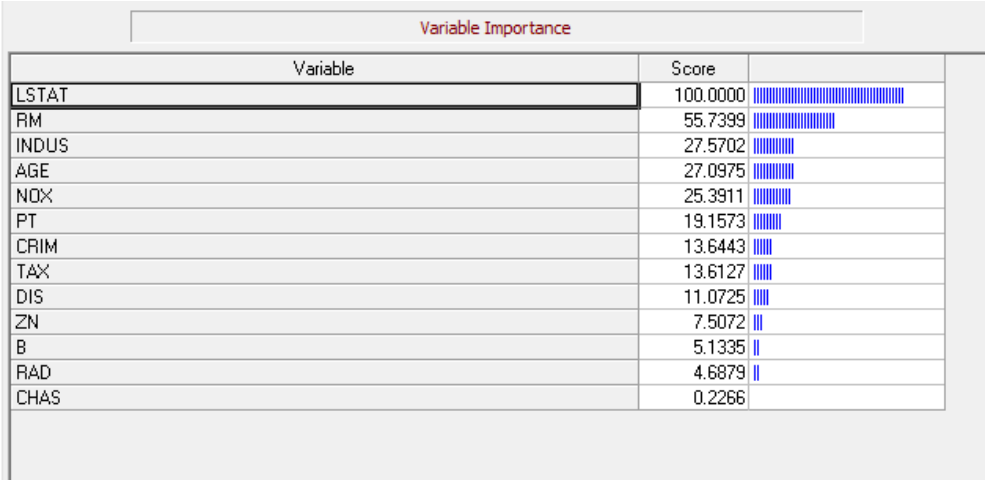
More performance measures

Actual Class	Total Class	Percent Correct	Predicted Classes	
			0 N = 286	1 N = 220
0	316.00	84.49%	**84.49**	15.51
1	190.00	90.00%	10.00	**90.00**
Total:	506.00			
Average:		87.25%		
Overall % Correct:		86.56%		
Specificity		84.49%		
Sensitivity/Recall		90.00%		
Precision		77.73%		
F1 statistic		83.41%		

500 trees with 3 predictors chosen at random for possible role as splitter in each node

Variable Importance

Size of the typical house and types of neighbors appear to be most important

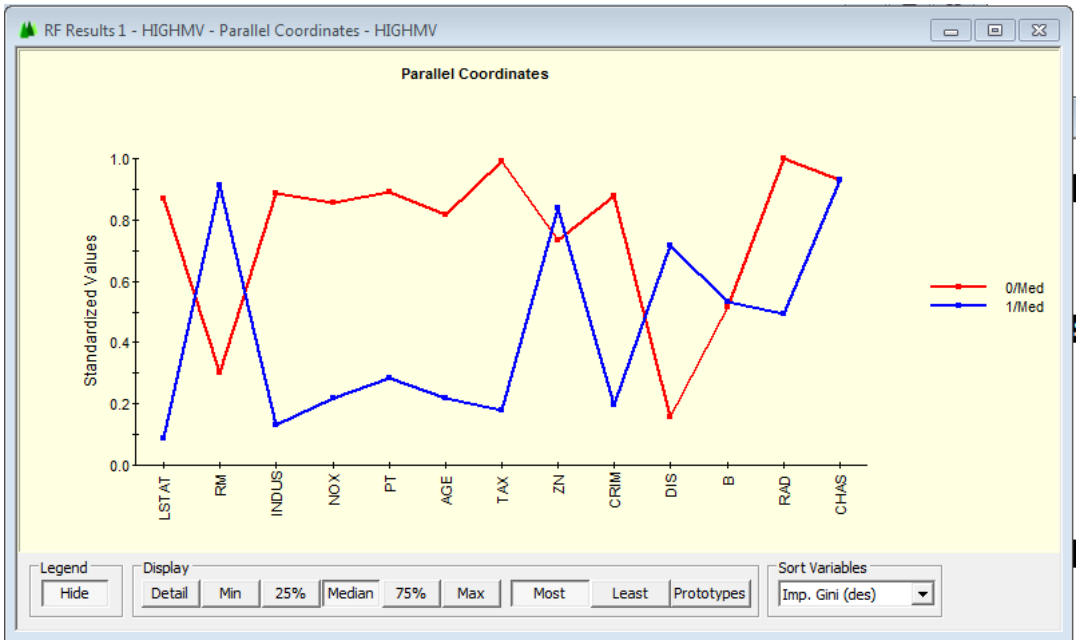


Variable Importance

Variable	Score
LSTAT	100.0000
RM	55.7399
INDUS	27.5702
AGE	27.0975
NOX	25.3911
PT	19.1573
CRIM	13.6443
TAX	13.6127
DIS	11.0725
ZN	7.5072
B	5.1335
RAD	4.6879
CHAS	0.2266

Most Likely Vs Not Likely Parallel Coordinate Plots

Contrasting the typical neighborhood with one of the highest probabilities of being above average versus a typical neighborhood at the other extreme



Here we take 25 neighborhoods from each end of the predicted probability (highest 25 and lowest 25) and plot average results

Parallel Coordinate Plots

All we are looking for here are large gaps between the blue (high value) and red (low value) lines

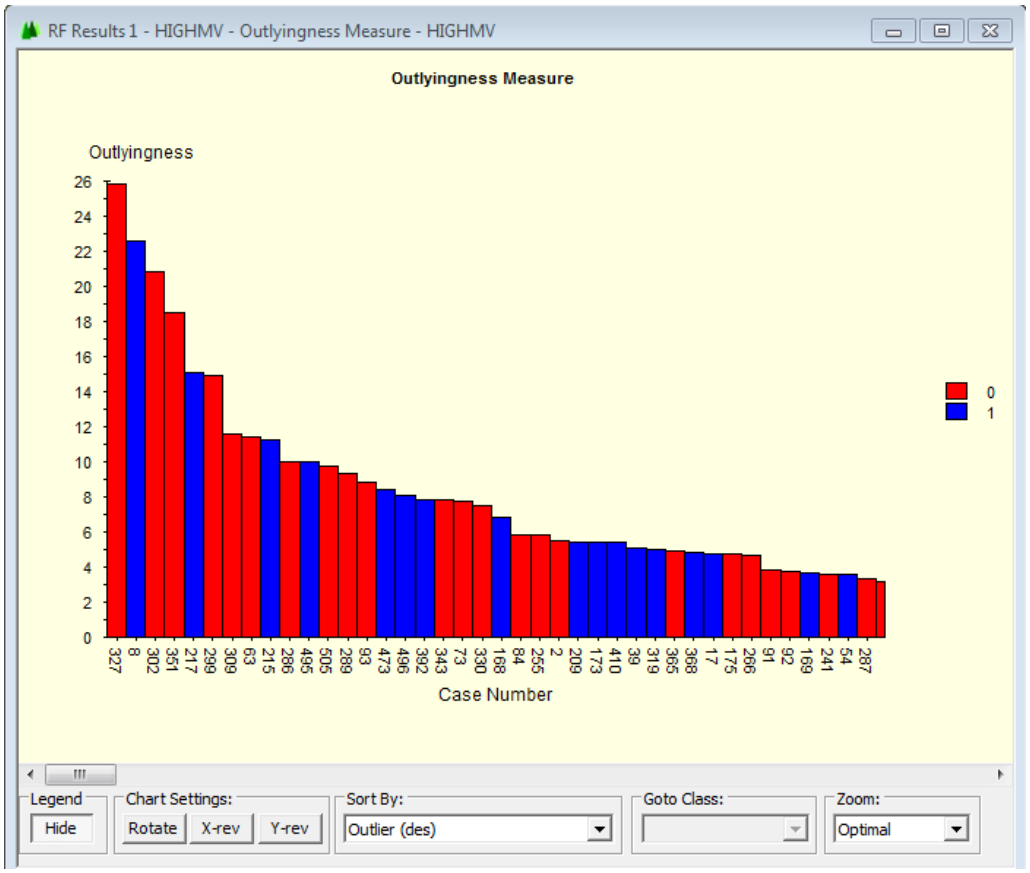
The blue line is positioned low for measures of undesirable characteristics while the red line is positioned high for those characteristics

These graphs serve to suggest the direction of the effect of any predictor

There are three variables which show essentially the same values between the two groups meaning that on their own they cannot serve to differentiate the groups

Outliers

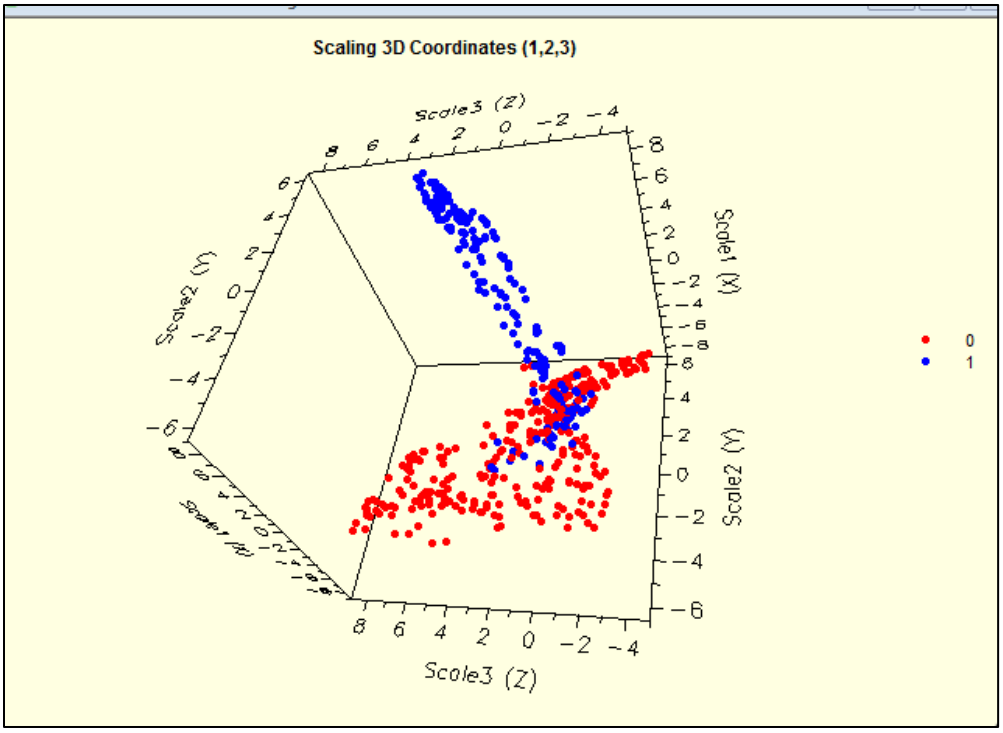
A score greater than 10 is considered worthy of attention and here we see the sorted list of scores with Record ID. 11 records look to be strange by this measure



Proximity & Clusters

The graph below plots all of the 506 data points to display relative distances using the RandomForests proximity measure

Points bunched together are very similar in this metric. Blue points are high value neighborhoods.



CHAPTER 5

REAL WORLD EXAMPLE

The Future of Alaska Project: Forecasting Alaska's Ecosystem in the 22nd Century

Analytics On a Grand Scale
Alaska over the next 100 years

- To assist long term planning related to Alaska's biological natural resources researchers at the University of Alaska, led by Professor Falk Huettman, have built models predicting the influence of climate change on many of Alaska's plants and animals
- An Associate Professor of Wildlife Ecology, Dr. Huettmann runs the EWHALE (Ecological Wildlife Habitat Data Analysis for the Land- and Seascape) Lab with the Institute of Arctic Biology, Biology and Wildlife Department at the University of Alaska-Fairbanks (UAF).

Challenge

Objective: forecast how climate change, human activities, natural disasters (floods, wildfires) and cataclysmic events (mass volcanic eruptions, melting polar ice cap) might affect Alaska's ecosystem over the next 100 years.

Analyzed data on more than 400 species of animals, thousands of plant species, and diverse landscape biomes (arctic tundra, coastal tundra plains, mountain and alpine areas), deciduous forests, arboreal forests, coastal rainforests, and the interior.

Some essential questions:

- Will melting ice create new shipping lanes?
- Will it be easier to exploit natural gas and oil fields or find new ones?
- How will a shrinking Arctic affect polar bears, migrating animals, commercial fishing, vegetation (will new arable land appear)?
- How will it affect global climate?

Dr. Huettmann concentrated on biomes and five key species that should be typical for all species. These included migratory caribou, water birds, invasive plant species and the Alaskan marmot. The latter was selected because climate warming and the melting ice in the upper Arctic are severely constricting the marmot's natural habitat, and it has no other place to go.

The solution

Dr. Huettmann elected to use Salford Systems' RandomForests predictive software

As Dr. Huettmann explains, "RandomForests is extremely well-suited to handle data based on GIS, spatial and temporal data. It delivers the high degree of accuracy and generalization required for our study, something other solutions couldn't achieve because of the immense size of the database and the statistical interactions involved. It achieves this with amazing speed. Also important, RandomForests software works in conjunction with languages such as Java, Python and with outputs from science programs such as GCMs (General Circulation Models that project global climate change) to provide a single and automated workflow."

Unlike more restrictive solutions, RandomForests predictive modeling software produces stronger statements, more advanced statistics and better generalizations. For predictive work, this is ideal and creates a new view and opportunities.

Dr. Huettmann is not only a wildlife ecologist; he is also a professor who includes students in many of his studies worldwide. As he explains it, “I only have students for a limited time, so when we use predictive modeling software, I want my students working, not struggling to use the software in the remotest corner of the world. The sophisticated GUI interface and software support Salford Systems uses in its predictive modeling software makes their programs very easy to use while delivering superior accuracy and generalization.”

"Our study opened up a completely new approach to modeling, predictions and forecasting projects that are relevant to the wellbeing of mankind," Dr. Huettmann states. "What we needed was a tool that could handle such incredible complexity. To give you some idea of the challenge, ordinary forecasting software provides accurate results using only a handful of predictors; however, with data mining and machine learning we can now use hundreds of variables. Using most of them and their interactions, we can get a much more accurate prediction."

The result

Dr. Huettmann's Future of Alaska report offers land managers, government agencies, communities, businesses, academics and non-profits a range of possible futures and scenarios, all based on substantial and transparent data.

The report provides a unique and useful way to evaluate how climate change and its contributors like carbon emissions, habitat change and consumption is impacting Alaska's ecosystems. And it will guide those concerned with making better management and sustainability decisions for oceans, wildlife, endangered species.

Customer Support

Professor Huettman also cites Salford Systems' customer support.

"They helped us install and set up the program to achieve progress," Dr. Huettmann concludes. "Someone was always available to answer questions, which is vitally important when working with students...and when the professor runs out of answers. Their customer support and developer team includes some of the most knowledgeable people I've ever had the privilege to work with."

CHAPTER 6

WHY SALFORD SYSTEMS

Why Salford Systems?

- With several commercial and open source implementations of Random Forests why should you go with Salford Systems?
- One compelling reason is that you will obtain *better results*
  - Greater accuracy
  - More reliable variable importance rankings
  - Built-in modeling automation
  - Built-in parallel processing

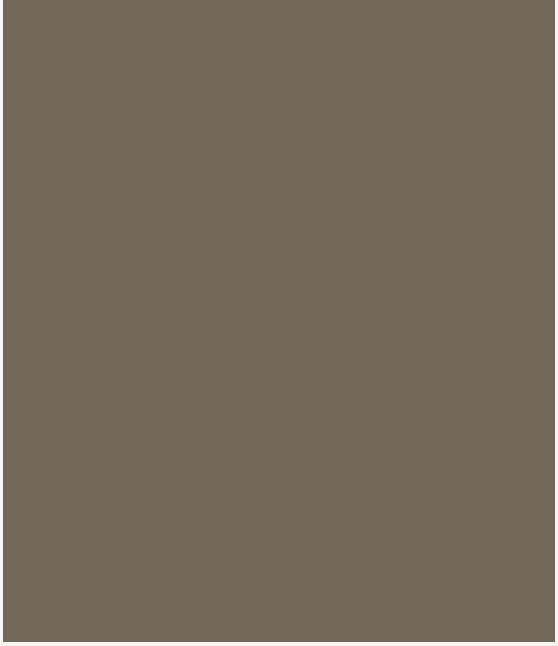
Unique to Salford

- Salford Systems jointly owns the Random Forests trademark and intellectual property
- Our implementation is based on source code Leo Breiman provided only to salford Systems
- Salford has continued working on RandomForests with co-creator Adele Cutler to refine and perfect the methodology
- Academic research confirms the superiority of Salford RandomForests

MEASURE THE ROI OF RANDOM FORESTS

Find out whether your investment in Random Forests is paying off. Sign up for Salford System's 30-day trial to access some insightful analytics.

>>> www.salford-systems/home/downloadspm



A publication of