Random Forest and CART for Body Fat Data

jdt

Fall 2023

Contents

Theory

Classification and Regression Trees

SAS HPSPLIT

The HPSPLIT procedure is a high-performance procedure that builds tree-based statistical models for classification and regression. The procedure produces classification trees, which model a categorical response, and regression trees, which model a continuous response. Both types of trees are referred to as decision trees because the model is expressed as a series of if-then statements.

The predictor variables for tree models can be categorical or continuous. The model is based on a partition of the predictor space into nonoverlapping segments, which correspond to the terminal nodes or leaves of the tree. The partitioning is done recursively, starting with the root node, which contains all the data, and ending with the terminal nodes. At each step of the recursion, the parent node is split into child nodes through

selection of a predictor variable and a split value that minimize the variability in the response across the child nodes.

Tree models are built from training data for which the response values are known, and these models are subsequently used to score (classify or predict) response values for new data. For classification trees, the most frequent response level of the training observations in a leaf is used to classify observations in that leaf. For regression trees, the average response of the training observations in a leaf is used to predict the response for observations in that leaf. The splitting rules that define the leaves provide the information that is needed to score new data.

The process of building a decision tree begins with growing a large, full tree. Various measures, such as the Gini index, entropy, and residual sum of squares, are used to assess candidate splits for each node. The full tree can overfit the training data, resulting in a model that does not adequately generalize to new data. To prevent overfitting, the full tree is pruned back to a smaller subtree that balances the goals of fitting training data and predicting new data. Two commonly applied approaches for finding the best subtree are cost-complexity pruning (Breiman et al. 1984) and C4.5 pruning (Quinlan 1993). For more information, see the section Building a Decision Tree.

SAS/STAT software provides many different methods of regression and classification. Compared with other methods, an advantage of tree models is that they are easy to interpret and visualize, especially when the tree is small. Tree-based methods scale well to large data, and they offer various methods of handling missing values, including surrogate splits.

However, tree models have limitations. Regression tree models fit response surfaces that are constant over rectangular regions of the predictor space, and so they often lack the flexibility needed to capture smooth relationships between the predictor variables and the response. Another limitation of tree models is that small changes in the data can lead to very different splits, and this undermines the interpretability of the model (Hastie, Tibshirani, and Friedman 2009; Kuhn and Johnson 2013).

Example - Baseball

You can alternatively fit a regression tree to predict the salaries of Major League Baseball players based on their performance measures from the previous season by using almost identical code. Regression trees are piecewise constant models that, for relatively small data sets such as this, provide succinct summaries of how the predictors determine the predictions. These models are usually easier to interpret than linear regression models. The Sashelp.Baseball data set contains salary and performance information for Major League Baseball players (excluding pitchers) who played at least one game in both the 1986 and 1987 seasons (Time Inc. 1987). The following statements create a regression tree model:

ods graphics on;

```
proc hpsplit data=sashelp.baseball seed=123;
   class league division;
   model logSalary = nAtBat nHits nHome nRuns nRBI nBB
   yrMajor crAtBat crHits crHome crRuns crRbi
   crBB league division nOuts nAssts nError;
   output out=hpsplout;
run;
```

By default, the tree is grown using the RSS criterion, and cost-complexity pruning with 10-fold cross validation is performed. The OUTPUT statement requests generation of the data set hpsplout, which contains the predicted salary from the tree model for each observation.

The following is from ChatGPT.

ChatGPT Explanation

CART (Classification and Regression Trees)

CART, an acronym for Classification and Regression Trees, is a decision tree learning technique that plays a pivotal role in the realm of machine learning and data mining. Developed by Breiman et al. in 1986, it provides a structured and intuitive way to represent decisions and decision-making processes.

The essence of CART lies in its bifurcating tree-like model, where each internal node denotes a decision criterion based on a feature, and each leaf node corresponds to an outcome. The process starts with the entire dataset at the root and then successively splits the data into subsets based on feature values. The ideal split at each stage is determined using measures like Gini impurity for classification tasks or mean squared error for regression tasks.

CART is versatile. In classification tasks, the goal is to correctly categorize data into discrete classes, while in regression, it aims to predict a continuous outcome variable. Its visual nature makes it easily interpretable, allowing users to understand the sequence and logic of decisions leading to a specific outcome.

Over the years, CART has laid the groundwork for various advanced algorithms like Random Forests and Gradient Boosted Trees. However, like all models, CART isn't without limitations. For instance, it can easily overfit data if not properly pruned. Despite its challenges, with the right pre-processing and tuning, CART remains a robust tool in a data scientist's arsenal.

Random Forest

The HPFOREST procedure is a high-performance procedure that creates a predictive model called a forest that consists of several decision trees. A predictive model defines a relationship between input variables and a target variable. The purpose of a predictive model is to predict a target value from inputs. The HPFOREST procedure trains the model; that is it creates the model using training data in which the target values are known. The model can then be applied to observations in which the target is unknown. If the predictions fit the new data well, the model is said to generalize well. Good generalization is the primary goal for predictive tasks. A predictive model might fit the training data well but generalize poorly.

A decision tree is a type of predictive model that has been developed independently in the statistics and artificial intelligence communities. The HPFOREST procedure creates a tree recursively. An input variable is chosen and used to create a rule to split the data into two segments. The process is then repeated in each segment, and then again in each new segment, and so on until some constraint is met. In the terminology of the tree metaphor, the segments are nodes, the original data set is the root node, and the final unpartitioned segments are leaves or terminal nodes. A node is an internal node if it is not a leaf. The data in a leaf determine the estimates of the value of the target variable. These estimates are subsequently applied to predict the target of a new observation assigned to the leaf.

The HPFOREST procedure creates decision trees that differ from each other in two ways. First, the training data for a tree is a sample, without replacement, from the original training data of the forest. Second, the

input variables considered for splitting a node are randomly selected from all available inputs. Among these variables, the HPFOREST procedure considers only a single variable when forming a splitting rule. The chosen variable is the one that is most associated with the target.

Example

This example compares the loss reduction variable importance measure on uncorrelated and correlated variables. The data have eight inputs that are generated from a standard normal distribution. The first four inputs are independent; the last four have a correlation of 0.9. The target Y is computed as

The following SAS statements create a SAS data set and run PROC HPFOREST:

```
data output;
   call streaminit(54321);
   do i=1 to 1000;
      x1 = rand('normal', 0, 1);
      x2 = rand('normal', 0, 1);
      x3 = rand('normal', 0, 1);
      x4 = rand('normal', 0, 1);
      output;
   end;
run;

data cov;
   input x5-x8;
   datalines;
   1 0.9 0.9 0.9
   0.9 1 0.9 0.9
   0.9 0.9 1 0.9
   0.9 0.9 0.9 1
run;

proc simnormal data=cov(type=cov)
   out = osim(drop=Rnum)
   numreal = 1000
   seed = 54321;
   var x5-x8;
run;

data output;
   merge output osim;
   y = x1 + x2 + 2*x3 + x5 + x6 + 2*x7;
run;

proc hpforest data=output vars_to_try=all;
   input x:/level=interval;
   target y/level=interval;
   ods select VariableImportance;
run;
```

Output shows the PROC HPFOREST variable importance table. The NRules column contains the number of splitting rules that use each variable. The next four columns are loss reduction measures of variable importance. The mean square error and the absolute error are computed with the training data. The OOB columns contain the same measures computed with out-of-bag data. In this example, the relative importance of any pair of variables is similar in every measure.

The following description is from ChatGPT.

ChatGPT explanation

Random Forest Procedure

Random Forest is an ensemble learning method renowned for its versatility and robustness in both classification and regression tasks. Conceived by Leo Breiman in 2001, the technique aggregates the predictions of multiple decision trees to produce a more accurate and stable result.

The procedure for building a Random Forest model encompasses the following steps:

- **Bootstrap Sampling**: For each tree in the forest, a bootstrap sample (a sample taken with replacement) is drawn from the original dataset. This means some data points may be sampled multiple times, while others may not be sampled at all.
- **Tree Building**: A decision tree is constructed using the bootstrap sample. However, unlike traditional decision tree algorithms, at each node, a random subset of features is selected for splitting. This injects diversity into the trees, ensuring that they aren't overly correlated.
- **Aggregation**: Once all trees have been built, the forest makes predictions by aggregating the results of individual trees. For classification tasks, a majority vote is taken, wherein the most common class prediction among all trees is chosen. For regression, the mean prediction of all trees is used.
- **Out-of-Bag (OOB) Error Estimation**: Since each tree is built using a bootstrap sample, some data points are left out during its construction. These left-out samples, termed "out-of-bag" samples, can be used to test the performance of the tree. By aggregating the OOB error of all trees, one can get a reasonable estimate of the forest's generalization error without needing a separate validation set.
- **Feature Importance**: Random Forest provides an inherent method to rank the importance of features. This is determined by the amount each feature decreases the overall model's impurity or error when it's used for splitting.

Random Forest is particularly admired for its ability to handle large datasets, tolerate missing values, and resist overfitting. However, its interpretability is less straightforward compared to a single decision tree. Despite this, its benefits, including high accuracy and the ability to assess feature importance, make it a staple in the machine learning domain.

Random Forest, as a popular ensemble method, has inspired and paved the way for various modifications and other ensemble methods. One of the most significant advancements in this domain is Gradient Boosting. Let's delve into it and some other main modifications:

1. Gradient Boosting:

Gradient Boosting is an iterative ensemble method that builds decision trees in a sequential manner. Unlike Random Forest, which builds trees independently, Gradient Boosting corrects the errors of the previous trees.

Key aspects include:

- **Error Correction**: Each subsequent tree aims to correct the residuals (errors) left by the combined ensemble of the preceding trees.
- **Weighted Updates**: During training, instances that are mispredicted by the current ensemble are given more weight, making them a focal point for the next tree.
- **Shrinkage**: Often, a learning rate (shrinkage) is applied to slow down the learning process, which can improve model performance by preventing overfitting.

2. Extremely Randomized Trees (Extra Trees):

Extra Trees is a modification of the basic Random Forest algorithm. The primary difference lies in the way splits are chosen:

While Random Forest selects the best split among a random subset of features, Extra Trees selects a random split for each feature and then picks the best among those. This added randomness often leads to a larger diversity in the forest, which can sometimes result in better generalization on certain datasets.

3. Random Forest with Feature Sampling:

In addition to sampling data points (bootstrap sampling), another modification involves random sampling of features for each tree. This brings about more diversity in trees and can help if some features are noisy or irrelevant.

4. Adaptive Boosting (AdaBoost):

Though not a direct modification of Random Forest, AdaBoost is another ensemble method inspired by the idea of boosting. AdaBoost focuses on training instances that were misclassified by the previous classifiers, giving them more weight. Unlike Gradient Boosting which fits to residuals, AdaBoost adjusts instance weights to improve subsequent classifier performance.

These are just a few of the many modifications and related ensemble techniques developed in machine learning. The key idea behind most of these advancements is to combine multiple weak learners to form a strong learner, aiming to improve accuracy, reduce overfitting, and enhance generalization across various data scenarios.

R

```r
  # clear the environment and set seed
rm(list = ls())
set.seed(123)
```

Read Body Fat Data

```r
bhouse = read.csv("bodyfat.csv")
summary(bhouse)
```

```
options("repos" = c(CRAN = "https://cran.rstudio.com"))
library(randomForest)
```

```
library(rpart)
library(rpart.plot)
```

Perform Regression

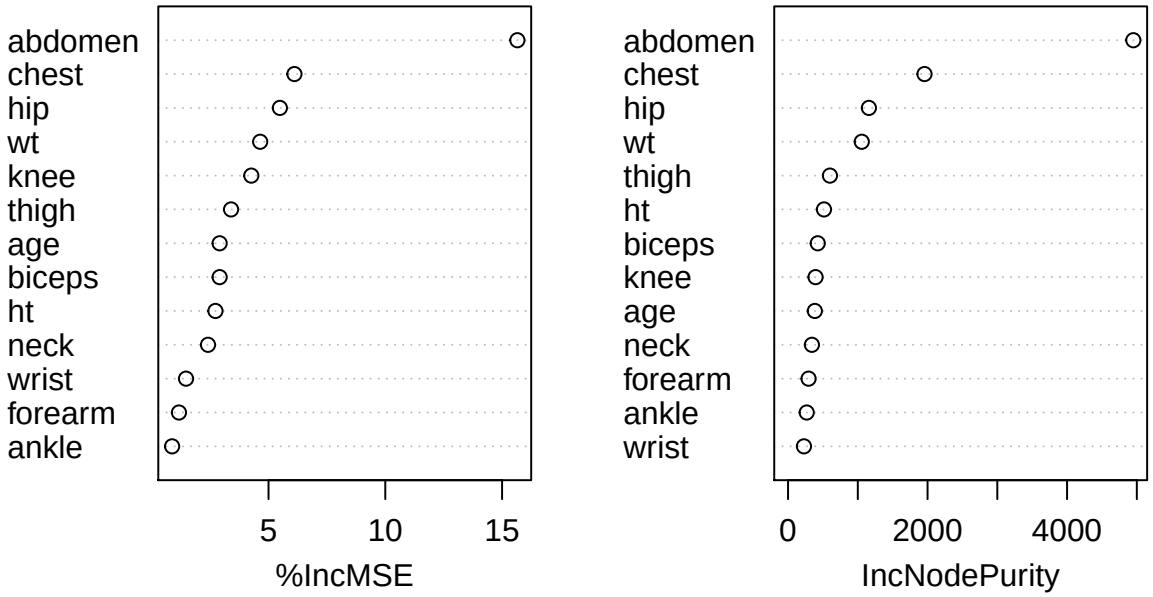
Random Forest

```
bfat.rf <- randomForest(per_fat ~ . - density,data=bfat,mtry=5,
              importance=TRUE, ntree=100,
              na.action=na.omit)

print(bfat.rf)
```

```
##
## Call:
##  randomForest(formula = per_fat ~ . - density,
    data = bfat, mtry = 5,
    importance = TRUE, ntree = 100, na.action = na.omit)
##                Type of random forest: regression
##                      Number of trees: 100
## No. of variables tried at each split: 5
##
##           Mean of squared residuals: 23.8103
##                     % Var explained: 66.84
```

```
varImpPlot(bfat.rf)
```

bfat.rf



CART

```
bfat.tr = rpart(per_fat ~ . - density, data=bfat)
# Output for Tree
print(bfat.tr)
```

```
## n= 180
##
## node), split, n, deviance, yval
##       * denotes terminal node
##
##  1) root 180 12925.60000 19.292780
##    2) abdomen< 91.9 91  3125.00600 13.328570
##      4) abdomen< 85.4 48  1049.39900 10.054170
##        8) neck>=36.35 16   166.87940  7.543750 *
##        9) neck< 36.35 32   731.26720 11.309380
##         18) biceps< 27.6 7    91.97714  6.657143 *
##         19) biceps>=27.6 25   445.36640 12.612000 *
##      5) abdomen>=85.4 43   986.47860 16.983720
##       10) ht>=72.875 8    94.19875 12.362500 *
##       11) ht< 72.875 35   682.38400 18.040000 *
##    3) abdomen>=91.9 89  3253.79300 25.391010
```

```
##       6) abdomen< 103 60  1197.41000 22.801670
##        12) abdomen< 98.1 38   630.40320 21.678950
##          24) neck>=38.95 17   183.85760 19.388240 *
##          25) neck< 38.95 21   285.12670 23.533330 *
##        13) abdomen>=98.1 22   436.37320 24.740910
##          26) wt>=196.875 9   119.74890 21.488890 *
##          27) wt< 196.875 13   155.54920 26.992310 *
##       7) abdomen>=103 29   821.79240 30.748280
##        14) abdomen< 112.3 21   243.64950 28.661900 *
##        15) abdomen>=112.3 8   246.77500 36.225000 *
```

```
plot(bfat.tr)
text(bfat.tr)
```

Perform Classification

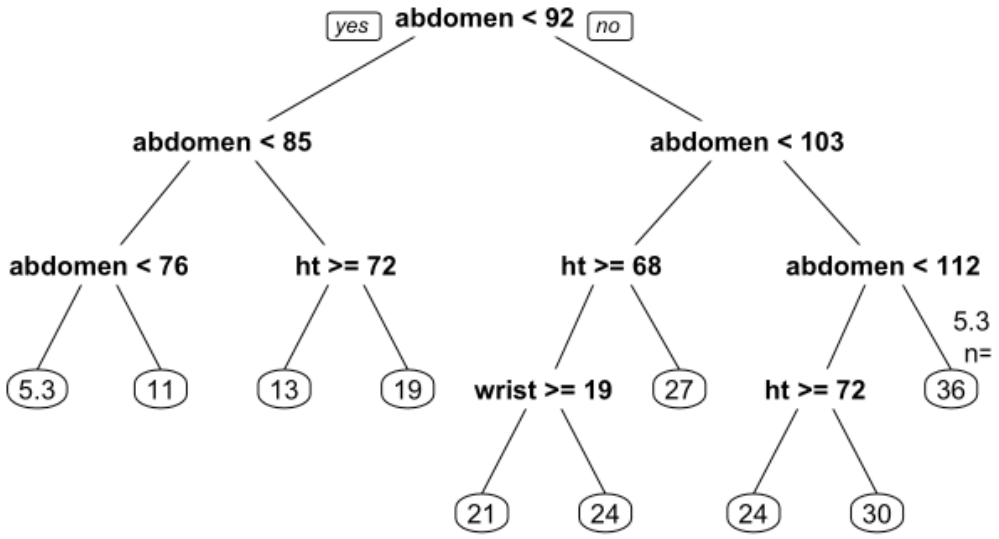
Random Forest

```
high_fat.rf <- randomForest(high_fat ~ .-density - per_fat,

data=bfat,mtry=5, importance=TRUE, ntree=100, proximity=TRUE)

print(high_fat.rf)
```

```
##
## Call:
##  randomForest(formula = high_fat ~ . - density -per_fat,
    data = bfat, mtry = 5, importance = TRUE,
    ntree = 100, proximity = TRUE)
##                Type of random forest: classification
##                      Number of trees: 100
## No. of variables tried at each split: 5
##
##         OOB estimate of  error rate: 18.89%
## Confusion matrix:
##       FALSE TRUE class.error
## FALSE   112   14   0.1111111
## TRUE     20   34   0.3703704
```

```
varImpPlot(high_fat.rf)
```

```
#plotting using rpart prp
library(rpart.plot)
prp(bfat.tr)
text(bfat.tr, use.n=T)
```



CART

```
high_fat.tr = rpart(high_fat ~ .-density - per_fat, data=bfat)

print(high_fat.tr)
```

```
## n= 180
##
## node), split, n, loss, yval, (yprob)
##       * denotes terminal node
##
##  1) root 180 54 FALSE (0.70000000 0.30000000)
##    2) abdomen< 98.1 129 14 FALSE (0.89147287 0.10852713)
##      4) abdomen< 91.3 87  2 FALSE (0.97701149 0.02298851) *
##      5) abdomen>=91.3 42 12 FALSE (0.71428571 0.28571429)
##       10) chest>=98.85 33  6 FALSE (0.81818182 0.18181818) *
##       11) chest< 98.85 9  3 TRUE (0.33333333 0.66666667) *
```

```
##     3) abdomen>=98.1 51 11 TRUE (0.21568627 0.78431373)
##       6) ht>=72.875 11  5 FALSE (0.54545455 0.45454545) *
##       7) ht< 72.875 40  5 TRUE (0.12500000 0.87500000) *
```

```r
plot(high_fat.tr)
text(high_fat.tr)
```

SAS

Code

```
options center nodate pagesize=100 ls=80;


title1 'Body Fat Data';
data bodyfat; set ldata.bodyfat;
high_fat = (per_fat > 24);
run;

title2 'Classification';

proc logistic data=bodyfat plots=roc;
class high_fat;
model high_fat = abdomen wt;
run;

proc hpsplit data=bodyfat seed=123;
   class high_fat;
   model high_fat =
      abdomen age ankle biceps chest forearm hip ht
      knee neck thigh wrist wt;
   grow entropy;
   prune costcomplexity;
run;


proc hpforest data=bodyfat maxtrees=100 inbagfraction=.3;
   input abdomen age ankle biceps chest forearm hip ht
        knee neck thigh wrist wt/level=interval;
   target high_fat/level=binary;
   ods output FitStatistics=fitstats(rename=(Ntrees=Trees));
run;

data fitstats;
   set fitstats;
   label Trees = 'Number of Trees';
   label MiscAll = 'Full Data';
   label Miscoob = 'OOB';
```

```
run;

proc sgplot data=fitstats;
   title "OOB vs Training";
   series x=Trees y=MiscAll;
   series x=Trees y=MiscOob/lineattrs=(pattern=shortdash thickness=2);
   yaxis label='Misclassification Rate';
run;
title;

title2 'Regression';

proc hpsplit data=bodyfat seed=123 ;
*   class high_fat;
   model per_fat =
      abdomen age ankle biceps chest forearm hip ht
      knee neck thigh wrist wt;
*   grow entropy;
*   prune costcomplexity;
  output out=hpsplout;
run;


title2 'Classification';
proc hpforest data=bodyfat maxtrees=100 inbagfraction=.3;
   input abdomen age ankle biceps chest forearm hip ht
        knee neck thigh wrist wt/level=interval;
   target per_fat/level=interval;
   ods output VariableImportance = variable;
   * FitStatistics=fitstats(rename=(Ntrees=Trees)) ;
run;

data fitstats;
   set fitstats;
   label Trees = 'Number of Trees';
   label MiscAll = 'Full Data';
   label Miscoob = 'OOB';
run;

proc sgplot data=fitstats;
   title "OOB vs Training";
   series x=Trees y=predall;
   series x=Trees y=predOob/lineattrs=(pattern=shortdash thickness=2);
   yaxis label='Average Squared Error';
run;
```

Output

Body Fat Data

Classification

The LOGISTIC Procedure

Model Information	
Data Set	WORK.BODYFAT
Response Variable	high_fat
Number of Response Levels	2
Model	binary logit
Optimization Technique	Fisher's scoring

Number of Observations Read	252
Number of Observations Used	252

Response Profile		
Ordered Value	high_fat	Total Frequency
1	0	176
2	1	76

Note	Probability modeled is high_fat=0.

Model Convergence Status
Convergence criterion (GCONV=1E−8) satisfied.

Model Fit Statistics		
Criterion	Intercept Only	Intercept and Covariates
AIC	310.550	154.629
SC	314.080	165.217
-2 Log L	308.550	148.629

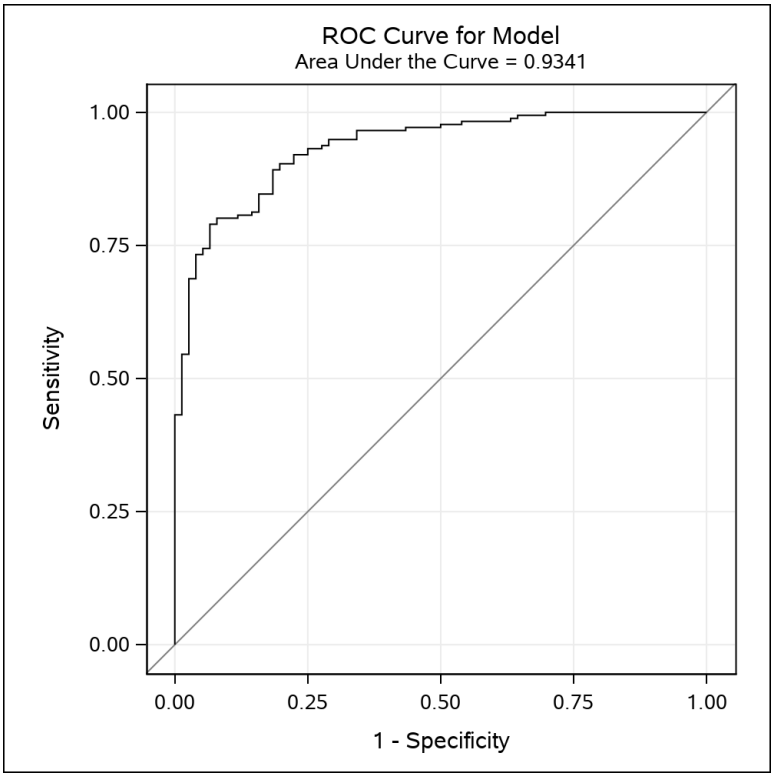
Testing Global Null Hypothesis: BETA=0			
Test	Chi-Square	DF	Pr > ChiSq
Likelihood Ratio	159.9217	2	<.0001
Score	118.1461	2	<.0001

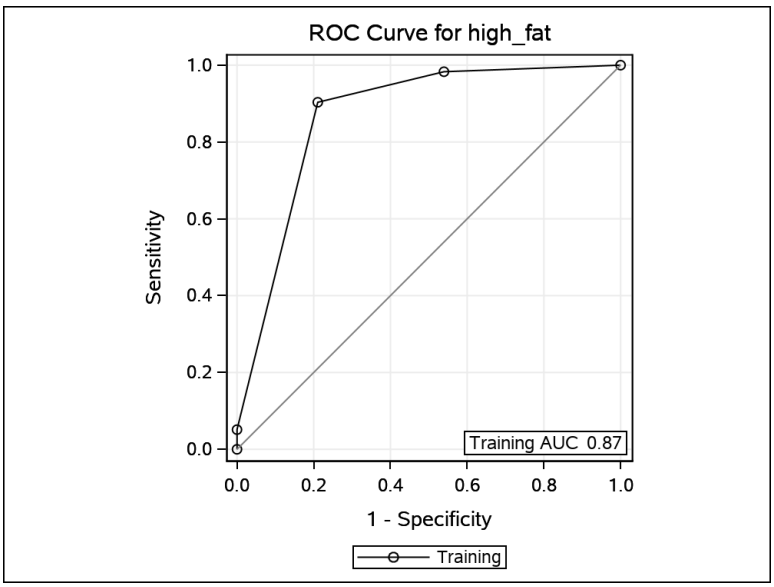
Testing Global Null Hypothesis: BETA=0			
Test	**Chi-Square**	**DF**	**Pr > ChiSq**
Wald	58.2659	2	<.0001

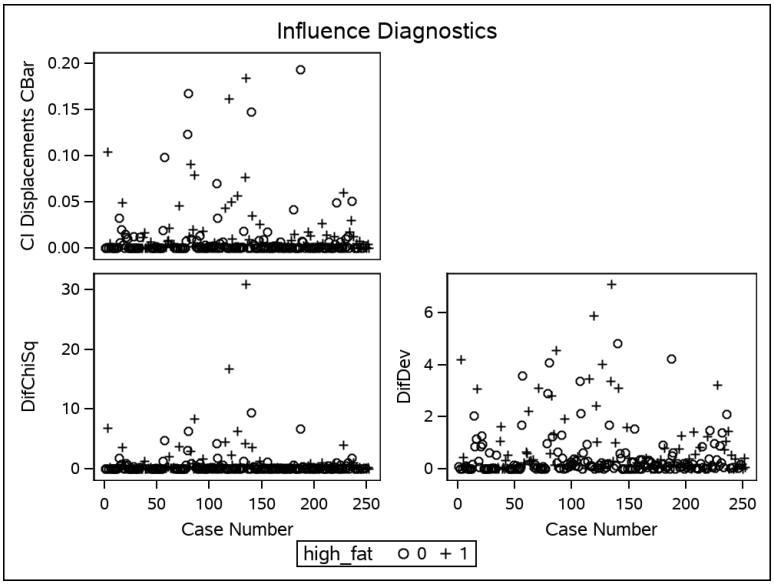
Analysis of Maximum Likelihood Estimates					
Parameter	**DF**	**Estimate**	**Standard Error**	**Wald Chi-Square**	**Pr > ChiSq**
Intercept	1	31.4737	4.0695	59.8161	<.0001
abdomen	1	−0.4814	0.0703	46.9196	<.0001
wt	1	0.0828	0.0191	18.8259	<.0001

Odds Ratio Estimates			
Effect	**Point Estimate**	**95% Wald Confidence Limits**	
abdomen	0.618	0.538	0.709
wt	1.086	1.046	1.128

Association of Predicted Probabilities and Observed Responses			
Percent Concordant	93.4	**Somers' D**	0.868
Percent Discordant	6.6	**Gamma**	0.868
Percent Tied	0.0	**Tau-a**	0.367
Pairs	13376	**c**	0.934


ROC Curve for Model
Area Under the Curve = 0.9341
Sensitivity
1 - Specificity
0.00
0.25
0.50
0.75
1.00


ROC Curve for high_fat
Sensitivity
1 - Specificity
Training AUC 0.87
Training
0.0
0.2
0.4
0.6
0.8
1.0



Body Fat Data

Classification

The HPSPLIT Procedure

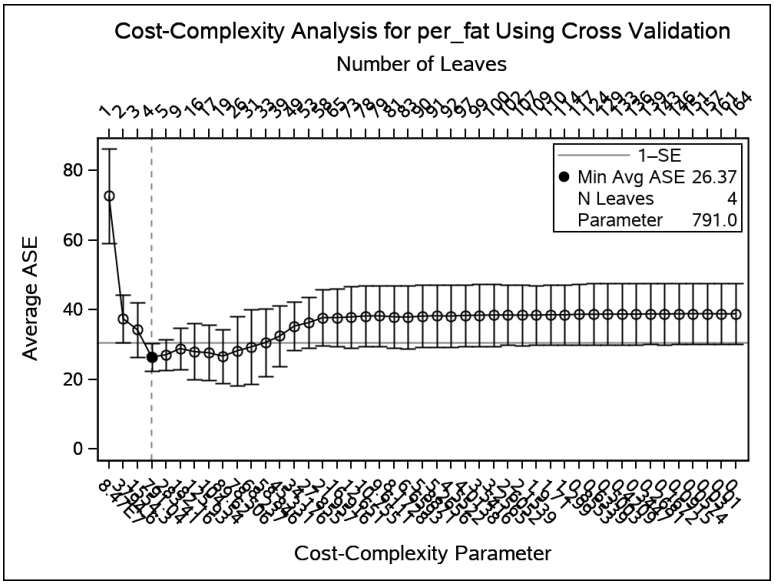
Performance Information	
Execution Mode	Single−Machine
Number of Threads	2

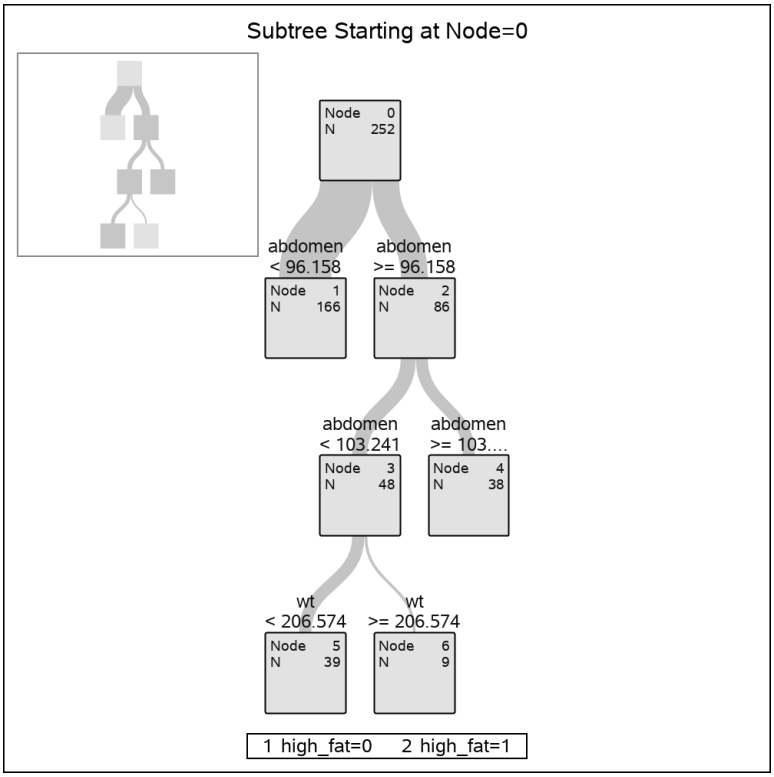
Data Access Information			
Data	**Engine**	**Role**	**Path**
WORK.BODYFAT	V9	Input	On Client

Model Information	
Split Criterion Used	Entropy
Pruning Method	Cost−Complexity
Subtree Evaluation Criterion	Cost−Complexity
Number of Branches	2
Maximum Tree Depth Requested	10
Maximum Tree Depth Achieved	10
Tree Depth	3

Model Information	
Number of Leaves Before Pruning	27
Number of Leaves After Pruning	4
Model Event Level	0

Number of Observations Read	252
Number of Observations Used	252




Subtree Starting at Node=0
Node 0
N 252
abdomen
< 96.158
abdomen
>= 96.158
Node 1
N 166
Node 2
N 86
abdomen
< 103.241
abdomen
>= 103....
Node 3
N 48
Node 4
N 38
wt
< 206.574
wt
>= 206.574
Node 5
N 39
Node 6
N 9
1 high_fat=0
2 high_fat=1

Body Fat Data

Classification

The HPSPLIT Procedure

Model-Based Confusion Matrix			
Actual	Predicted		Error Rate
	0	1	
0	159	17	0.0966
1	16	60	0.2105

Model-Based Fit Statistics for Selected Tree								
N Leaves	ASE	Mis- class	Sensitivity	Specificity	Entropy	Gini	RSS	AUC
4	0.1039	0.1310	0.9034	0.7895	0.5072	0.2079	52.3907	0.8673

Variable Importance			
Variable	Training		Count
	Relative	Importance	
abdomen	1.0000	6.9107	2
wt	0.3547	2.4515	1

Body Fat Data

Classification

The HPFOREST Procedure

Performance Information	
Execution Mode	Single−Machine
Number of Threads	2

Data Access Information			
Data	**Engine**	**Role**	**Path**
WORK.BODYFAT	V9	Input	On Client

Model Information		
Parameter	**Value**	
Variables to Try	4	(Default)
Maximum Trees	100	
Actual Trees	100	
Inbag Fraction	0.3	
Prune Fraction	0	(Default)
Prune Threshold	0.1	(Default)
Leaf Fraction	0.00001	(Default)
Leaf Size Setting	1	(Default)
Leaf Size Used	1	
Category Bins	30	(Default)
Interval Bins	100	
Minimum Category Size	5	(Default)
Node Size	100000	(Default)
Maximum Depth	20	(Default)
Alpha	1	(Default)
Exhaustive	5000	(Default)
Rows of Sequence to Skip	5	(Default)
Split Criterion	.	Gini
Preselection Method	.	BinnedSearch
Missing Value Handling	.	Valid value

Number of Observations	
Type	N
Number of Observations Read	252
Number of Observations Used	252

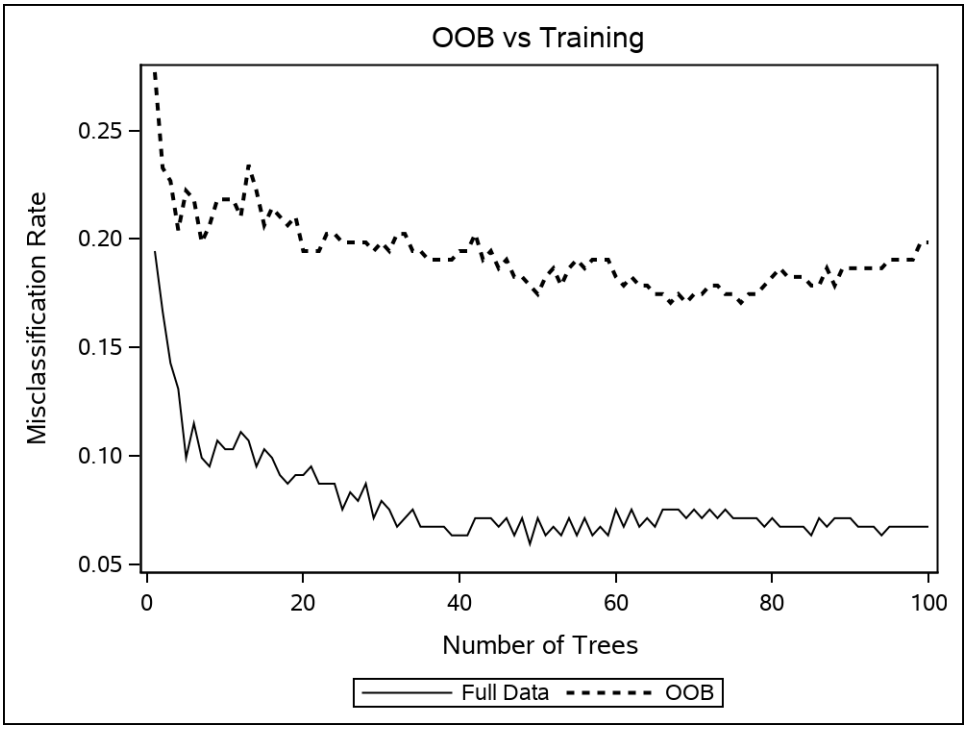
Baseline Fit Statistics	
Statistic	Value
Average Square Error	0.211
Misclassification Rate	0.302
Log Loss	0.612

Fit Statistics							
Trees	# Leaves	ASE (Train)	ASE (OOB)	MCR (Train)	MCR (OOB)	LLoss (Train)	LLoss (OOB)
1	13	0.1944	0.277	0.1944	0.277	4.477	6.374
2	25	0.1310	0.223	0.1667	0.233	1.890	4.460
3	38	0.1089	0.209	0.1429	0.227	0.954	3.616
4	49	0.0947	0.188	0.1310	0.204	0.528	2.638
5	65	0.0922	0.185	0.0992	0.222	0.369	2.297
6	79	0.0868	0.176	0.1151	0.218	0.279	1.956
7	91	0.0837	0.163	0.0992	0.198	0.272	1.438
8	104	0.0771	0.157	0.0952	0.206	0.253	1.261
9	116	0.0797	0.155	0.1071	0.218	0.260	1.095
10	129	0.0798	0.156	0.1032	0.218	0.261	1.102
11	141	0.0804	0.158	0.1032	0.218	0.264	1.105
12	154	0.0814	0.159	0.1111	0.210	0.266	1.109
13	169	0.0811	0.158	0.1071	0.234	0.265	1.024
14	181	0.0795	0.154	0.0952	0.222	0.261	1.012
15	196	0.0773	0.151	0.1032	0.206	0.256	1.005
16	211	0.0769	0.149	0.0992	0.214	0.257	0.923
17	224	0.0760	0.147	0.0913	0.210	0.256	0.918
18	242	0.0741	0.146	0.0873	0.206	0.254	0.680
19	253	0.0747	0.148	0.0913	0.210	0.257	0.685
20	269	0.0749	0.147	0.0913	0.194	0.259	0.606
21	282	0.0748	0.147	0.0952	0.194	0.258	0.607
22	299	0.0745	0.147	0.0873	0.194	0.259	0.449
23	315	0.0736	0.146	0.0873	0.202	0.258	0.447
24	326	0.0737	0.146	0.0873	0.202	0.258	0.450
25	340	0.0732	0.146	0.0754	0.198	0.257	0.448
26	357	0.0734	0.148	0.0833	0.198	0.259	0.454
27	372	0.0730	0.147	0.0794	0.198	0.258	0.452
28	385	0.0722	0.146	0.0873	0.198	0.257	0.449
29	400	0.0714	0.145	0.0714	0.194	0.255	0.446
30	414	0.0703	0.142	0.0794	0.198	0.252	0.439
31	433	0.0693	0.139	0.0754	0.194	0.250	0.429
32	448	0.0696	0.140	0.0675	0.202	0.251	0.432
33	464	0.0697	0.140	0.0714	0.202	0.252	0.431
34	481	0.0694	0.140	0.0754	0.194	0.251	0.432
35	497	0.0687	0.139	0.0675	0.194	0.250	0.430

Fit Statistics							
Trees	# Leaves	ASE (Train)	ASE (OOB)	MCR (Train)	MCR (OOB)	LLoss (Train)	LLoss (OOB)
36	513	0.0687	0.139	0.0675	0.190	0.251	0.432
37	526	0.0693	0.139	0.0675	0.190	0.252	0.434
38	536	0.0693	0.139	0.0675	0.190	0.252	0.433
39	550	0.0694	0.139	0.0635	0.190	0.252	0.430
40	563	0.0698	0.139	0.0635	0.194	0.253	0.431
41	579	0.0691	0.138	0.0635	0.194	0.251	0.429
42	593	0.0690	0.138	0.0714	0.202	0.251	0.426
43	607	0.0684	0.136	0.0714	0.190	0.250	0.422
44	622	0.0682	0.136	0.0714	0.194	0.249	0.420
45	638	0.0680	0.136	0.0675	0.187	0.250	0.420
46	656	0.0679	0.136	0.0714	0.190	0.250	0.421
47	670	0.0678	0.135	0.0635	0.183	0.250	0.420
48	686	0.0679	0.136	0.0714	0.183	0.250	0.420
49	698	0.0676	0.135	0.0595	0.179	0.248	0.418
50	713	0.0678	0.135	0.0714	0.175	0.249	0.420
51	725	0.0675	0.134	0.0635	0.183	0.248	0.417
52	739	0.0672	0.134	0.0675	0.187	0.247	0.417
53	754	0.0671	0.134	0.0635	0.179	0.247	0.417
54	770	0.0670	0.134	0.0714	0.187	0.247	0.415
55	781	0.0672	0.133	0.0635	0.190	0.247	0.414
56	794	0.0670	0.133	0.0714	0.187	0.247	0.414
57	810	0.0674	0.134	0.0635	0.190	0.248	0.416
58	827	0.0675	0.135	0.0675	0.190	0.248	0.417
59	843	0.0673	0.134	0.0635	0.190	0.247	0.416
60	857	0.0674	0.134	0.0754	0.183	0.248	0.417
61	870	0.0675	0.134	0.0675	0.179	0.248	0.417
62	881	0.0677	0.134	0.0754	0.183	0.248	0.418
63	895	0.0676	0.134	0.0675	0.179	0.248	0.418
64	915	0.0673	0.134	0.0714	0.179	0.248	0.419
65	926	0.0671	0.134	0.0675	0.175	0.247	0.418
66	939	0.0670	0.134	0.0754	0.175	0.247	0.418
67	951	0.0668	0.134	0.0754	0.171	0.247	0.418
68	960	0.0672	0.134	0.0754	0.175	0.247	0.418
69	975	0.0672	0.134	0.0714	0.171	0.247	0.417
70	989	0.0669	0.133	0.0754	0.175	0.246	0.415
71	1010	0.0666	0.133	0.0714	0.175	0.246	0.414
72	1021	0.0668	0.133	0.0754	0.179	0.246	0.413
73	1033	0.0666	0.133	0.0714	0.179	0.245	0.413
74	1049	0.0665	0.133	0.0754	0.175	0.245	0.413
75	1060	0.0664	0.133	0.0714	0.175	0.245	0.412
76	1076	0.0665	0.133	0.0714	0.171	0.245	0.412
77	1089	0.0669	0.133	0.0714	0.175	0.246	0.413
78	1106	0.0668	0.133	0.0714	0.175	0.246	0.414
79	1120	0.0667	0.133	0.0675	0.179	0.246	0.413
80	1138	0.0664	0.133	0.0714	0.183	0.246	0.412
81	1159	0.0664	0.133	0.0675	0.187	0.246	0.412
82	1174	0.0662	0.133	0.0675	0.183	0.245	0.411
83	1189	0.0663	0.133	0.0675	0.183	0.245	0.412
84	1203	0.0663	0.133	0.0675	0.183	0.246	0.413
85	1217	0.0665	0.134	0.0635	0.179	0.246	0.413
86	1232	0.0667	0.134	0.0714	0.179	0.247	0.415
87	1245	0.0669	0.135	0.0675	0.187	0.247	0.416
88	1258	0.0670	0.134	0.0714	0.179	0.247	0.415
89	1276	0.0668	0.134	0.0714	0.187	0.247	0.415

Fit Statistics							
Trees	# Leaves	ASE (Train)	ASE (OOB)	MCR (Train)	MCR (OOB)	LLoss (Train)	LLoss (OOB)
90	1289	0.0668	0.135	0.0714	0.187	0.247	0.415
91	1308	0.0667	0.135	0.0675	0.187	0.247	0.416
92	1323	0.0665	0.134	0.0675	0.187	0.247	0.415
93	1338	0.0665	0.135	0.0675	0.187	0.247	0.416
94	1351	0.0666	0.135	0.0635	0.187	0.247	0.416
95	1365	0.0664	0.134	0.0675	0.190	0.246	0.414
96	1381	0.0662	0.134	0.0675	0.190	0.246	0.414
97	1394	0.0662	0.134	0.0675	0.190	0.246	0.414
98	1407	0.0665	0.135	0.0675	0.190	0.246	0.415
99	1424	0.0663	0.134	0.0675	0.198	0.246	0.415
100	1438	0.0663	0.134	0.0675	0.198	0.246	0.414

Loss Reduction Variable Importance					
Variable	Number of Rules	Gini	OOB Gini	Margin	OOB Margin
chest	134	0.070497	0.01858	0.140994	0.089548
abdomen	38	0.030638	0.01751	0.061277	0.047259
hip	74	0.029034	−0.00257	0.058068	0.026199
age	54	0.019903	−0.00392	0.039806	0.014999
knee	126	0.046156	−0.00843	0.092311	0.038034
wt	150	0.039976	−0.01352	0.079952	0.023966
ankle	70	0.016698	−0.01622	0.033396	0.001128
biceps	113	0.035735	−0.01632	0.071471	0.018405
forearm	112	0.023035	−0.01880	0.046070	0.003853
thigh	101	0.025043	−0.02045	0.050086	0.006361
ht	104	0.029080	−0.02051	0.058160	0.009449
neck	117	0.024480	−0.02100	0.048959	0.003096
wrist	145	0.028398	−0.03017	0.056797	0.000557


OOB vs Training
Misclassification Rate
0.25
0.20
0.15
0.10
0.05
0
20
40
60
80
100
Number of Trees
Full Data
OOB

Regression

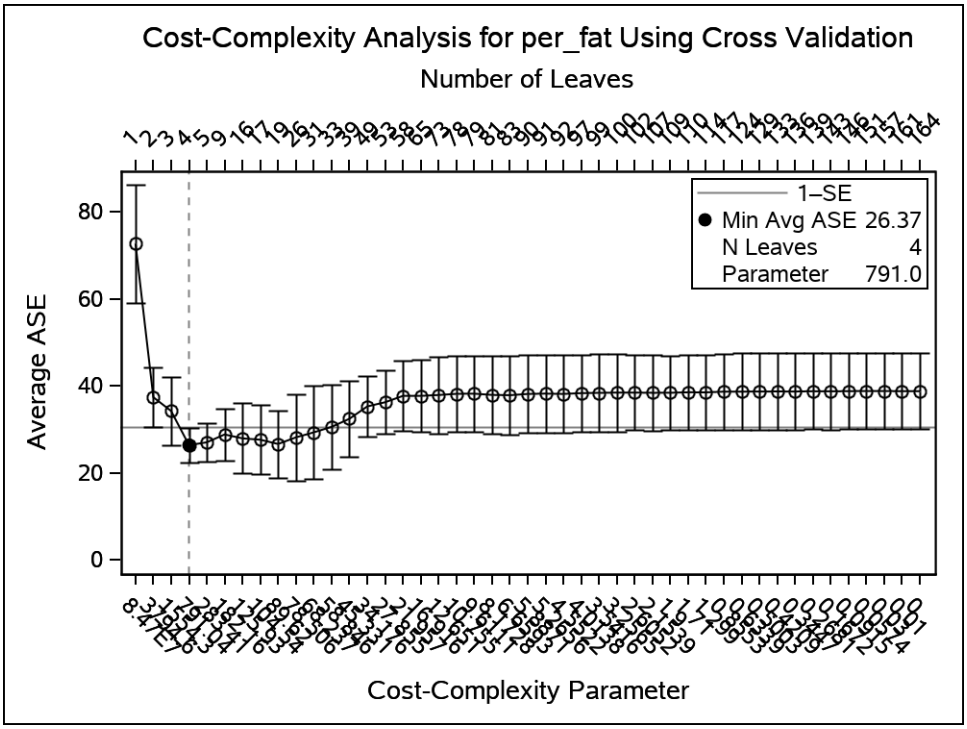
The HPSPLIT Procedure

Performance Information	
Execution Mode	Single−Machine
Number of Threads	2

Data Access Information			
Data	**Engine**	**Role**	**Path**
WORK.BODYFAT	V9	Input	On Client
WORK.HPSPLOUT	V9	Output	On Client

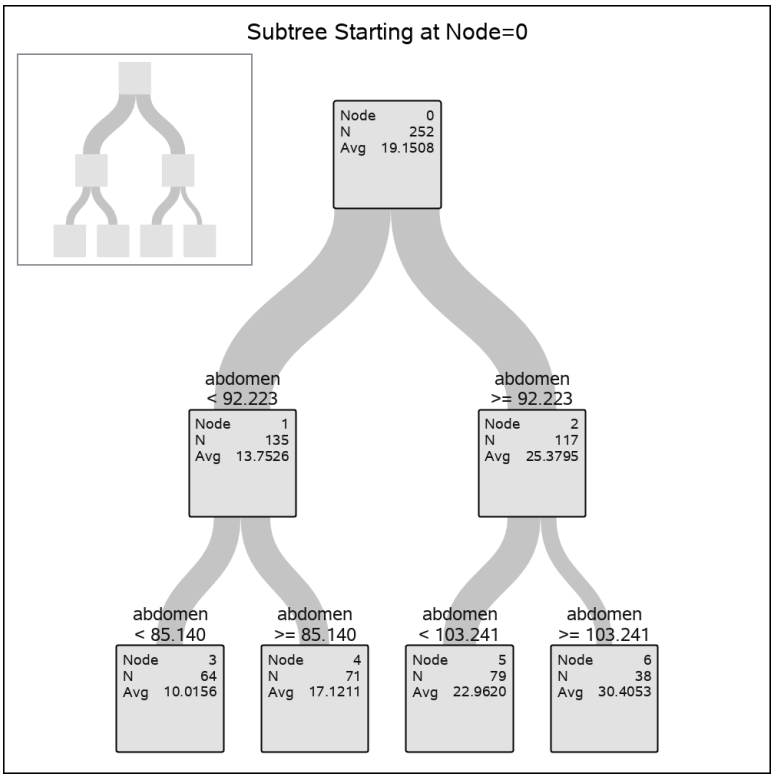
Model Information	
Split Criterion Used	Variance
Pruning Method	Cost−Complexity
Subtree Evaluation Criterion	Cost−Complexity
Number of Branches	2
Maximum Tree Depth Requested	10
Maximum Tree Depth Achieved	10
Tree Depth	2
Number of Leaves Before Pruning	169
Number of Leaves After Pruning	4

Number of Observations Read	252
Number of Observations Used	252


Cost-Complexity Analysis for per_fat Using Cross Validation
Number of Leaves
1–SE
Min Avg ASE 26.37
N Leaves 4
Parameter 791.0
Average ASE
0
20
40
60
80
Cost-Complexity Parameter

Regression

The HPSPLIT Procedure



Regression

The HPSPLIT Procedure

Model-Based Fit Statistics for Selected Tree		
N Leaves	ASE	RSS
4	23.7496	5984.9

Variable Importance			
Variable	Training		Count
	Relative	Importance	
abdomen	1.0000	107.7	3

Classification

The HPFOREST Procedure

Performance Information	
Execution Mode	Single−Machine
Number of Threads	2

Data Access Information			
Data	**Engine**	**Role**	**Path**
WORK.BODYFAT	V9	Input	On Client

Model Information		
Parameter	**Value**	
Variables to Try	4	(Default)
Maximum Trees	100	
Actual Trees	100	
Inbag Fraction	0.3	
Prune Fraction	0	(Default)
Prune Threshold	0.1	(Default)
Leaf Fraction	0.00001	(Default)
Leaf Size Setting	1	(Default)
Leaf Size Used	1	
Category Bins	30	(Default)
Interval Bins	100	
Minimum Category Size	5	(Default)
Node Size	100000	(Default)
Maximum Depth	20	(Default)
Alpha	1	(Default)
Exhaustive	5000	(Default)
Rows of Sequence to Skip	5	(Default)
Split Criterion	.	Variance
Preselection Method	.	BinnedSearch
Missing Value Handling	.	Valid value

Number of Observations	
Type	N
Number of Observations Read	252
Number of Observations Used	252

Baseline Fit Statistics	
Statistic	Value
Average Square Error	69.758

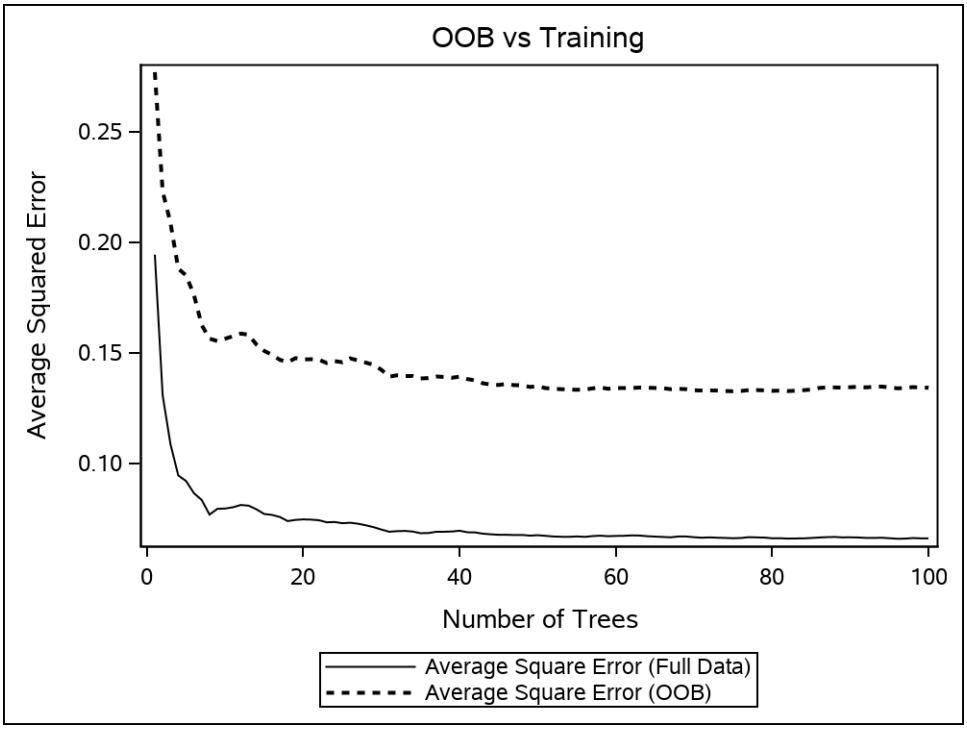
Fit Statistics			
# Trees	# Leaves	ASE (Train)	ASE (OOB)
1	75	44.0554	62.7229
2	149	25.3299	47.3117
3	223	21.8696	42.9064
4	298	21.6833	41.6902
5	373	19.4645	38.5677
6	447	18.4313	35.1656
7	521	18.6605	35.2504
8	596	17.9436	34.0874
9	667	17.6903	33.8045
10	740	17.2653	33.4603
11	814	16.9837	32.9796
12	887	16.2745	31.5819
13	961	16.0995	31.2615
14	1036	16.4530	31.2932
15	1110	16.1144	30.9546
16	1185	15.8478	30.3738
17	1260	15.6905	29.9065
18	1333	15.6553	29.9131
19	1408	15.5352	30.0728
20	1482	15.5500	29.9107
21	1556	15.6478	30.0178
22	1631	15.4337	29.7881
23	1704	15.4148	29.7264

Fit Statistics			
# Trees	# Leaves	ASE (Train)	ASE (OOB)
24	1779	15.5392	29.8202
25	1853	15.5268	30.0285
26	1928	15.7787	30.4601
27	2002	15.5951	30.1171
28	2076	15.3136	29.9284
29	2149	15.1620	29.6873
30	2223	15.0950	29.5590
31	2297	15.0422	29.3446
32	2372	14.9931	29.1881
33	2445	15.0336	29.2459
34	2520	14.9203	29.1273
35	2595	14.8388	29.0129
36	2670	14.9460	29.1552
37	2742	14.8873	29.0067
38	2817	14.8620	28.8959
39	2891	14.9365	29.0167
40	2964	14.8756	28.8771
41	3038	14.7495	28.6292
42	3112	14.7608	28.5720
43	3187	14.6516	28.3587
44	3262	14.5617	28.1798
45	3336	14.6104	28.3760
46	3411	14.5414	28.2971
47	3485	14.5887	28.4091
48	3560	14.6360	28.5514
49	3634	14.6678	28.6256
50	3708	14.6288	28.6642
51	3783	14.6269	28.7828
52	3858	14.6034	28.7416
53	3933	14.5678	28.7489
54	4006	14.5926	28.8127
55	4079	14.5733	28.7635

Fit Statistics			
# Trees	# Leaves	ASE (Train)	ASE (OOB)
56	4152	14.4424	28.5503
57	4225	14.4556	28.5354
58	4299	14.3550	28.4300
59	4371	14.3188	28.4266
60	4445	14.3636	28.5307
61	4519	14.3110	28.4819
62	4594	14.3949	28.5923
63	4668	14.3482	28.5132
64	4743	14.2635	28.3722
65	4817	14.2141	28.4033
66	4891	14.2650	28.4764
67	4965	14.2495	28.4364
68	5038	14.3294	28.5367
69	5112	14.2551	28.4090
70	5185	14.2289	28.3548
71	5259	14.2762	28.4055
72	5333	14.3479	28.5633
73	5407	14.2724	28.4601
74	5482	14.3244	28.5267
75	5555	14.3290	28.5755
76	5630	14.3595	28.6601
77	5703	14.3651	28.7229
78	5776	14.4597	28.8570
79	5851	14.4510	28.8064
80	5925	14.4130	28.7395
81	5999	14.4544	28.8020
82	6073	14.3661	28.6552
83	6146	14.3879	28.6827
84	6219	14.4058	28.6983
85	6293	14.4338	28.7022
86	6368	14.4613	28.6906
87	6442	14.4562	28.6641

Fit Statistics			
# Trees	# Leaves	ASE (Train)	ASE (OOB)
88	6515	14.5002	28.7346
89	6588	14.5060	28.7702
90	6663	14.5220	28.7897
91	6738	14.4920	28.7519
92	6811	14.5123	28.8025
93	6885	14.5434	28.9071
94	6959	14.5361	28.9151
95	7033	14.4996	28.8402
96	7108	14.5286	28.8821
97	7182	14.5136	28.8558
98	7255	14.5606	28.9603
99	7330	14.6279	29.0801
100	7405	14.6432	29.1072

Loss Reduction Variable Importance					
Variable	# of Rules	MSE	OOB MSE	AE	OOB AE
chest	376	15.72897	8.97173	1.132342	0.494677
abdomen	105	5.96134	5.75150	0.435924	0.352351
hip	496	7.10715	2.97986	0.612944	0.182800
wt	1389	8.13754	0.51918	0.879102	0.059398
knee	685	5.95566	0.44671	0.604846	0.052925
age	133	1.92889	−0.14963	0.187235	−0.010502
biceps	329	5.46501	−0.85974	0.474581	−0.056738
thigh	671	3.41096	−1.03755	0.449174	−0.052877
ht	642	4.29252	−1.30774	0.502844	−0.075708
neck	589	2.60546	−1.64790	0.368101	−0.092476
forearm	302	2.06023	−1.89392	0.275551	−0.103588
ankle	301	2.40876	−2.15517	0.319679	−0.089029
wrist	1287	2.84780	−2.99895	0.527503	−0.134661


OOB vs Training
Average Squared Error
0.25
0.20
0.15
0.10
0
20
40
60
80
100
Number of Trees
Average Square Error (Full Data)
Average Square Error (OOB)