SAS Enterprise Miner ®を使用した機械学習

辻　仁史

SAS Institute Japan 株式会社
ソリューションコンサルティング第1本部
エンタープライスアナリティクス推進グループ

Machine Learning With SAS® Enterprise Miner

Hitoshi Tsuji

SAS Institute Japan Ltd.

Enterprise Analytics Promotion Group
Solution Consulting Division1

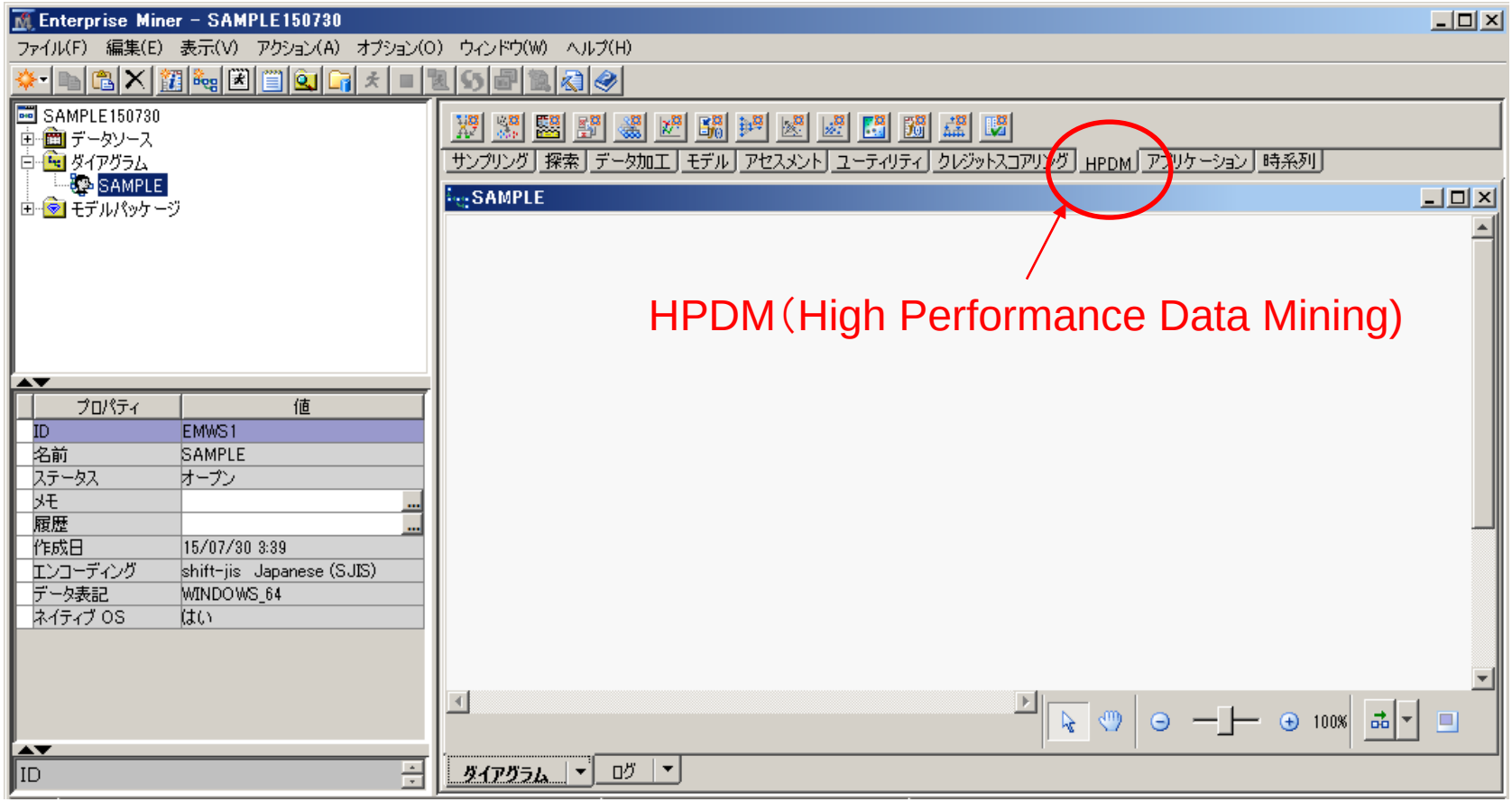
要旨:

SAS Enterprise Miner ®に実装されている機械学習について、使用例を交えてご説明します。

キーワード: Enterprise Miner　機械学習

はじめに：SAS Enterprise Miner ®とは

SAS の提供する大規模データマイニングツール



HPDMで提供するノード一覧

HP データ分割	学習と検証に使用するオブザベーションを識別する識別子変数を生成します。
HP クラスタ	K-Meansクラスタ。
HP探索	HP 環境で要約統計量を計算します。
HP変数選択	HP プロシジャを使用した変数選択。
HP補完	HP プロシジャを使用して、欠損値を補完します。
HP主成分	後続ノードの入力として使用する主成分を生成します。
HP変換	ハイパフォーマンスプロシジャを使用して、変換変数を作成します。
HPニューラル	HP 環境で ニューラルネットワークモデルを作成します。
HP回帰	ハイパフォーマンス線形回帰モデルとロジスティック回帰モデル。
HPツリー	HP 環境でツリーモデルを作成します。
HP BN分類子	Baysian Network（Naive Bayes）を使用した分類モデル。 14.1追加機能
HPフォレスト	HP 環境でランダム フォレストモデルを作成します。
HPGLM	HP 環境で一般化線形モデルを作成します。
HPSVM	HP 環境でサポートベクターマシン(SVM) モデルを作成します。

機械学習はチューニングが肝要

・機械学習は学習量を制御する何らかのチューニングパラメータを持っている。

・パラメータによって結果は大きく異なる。

・パラメータを適切に設定しないと、

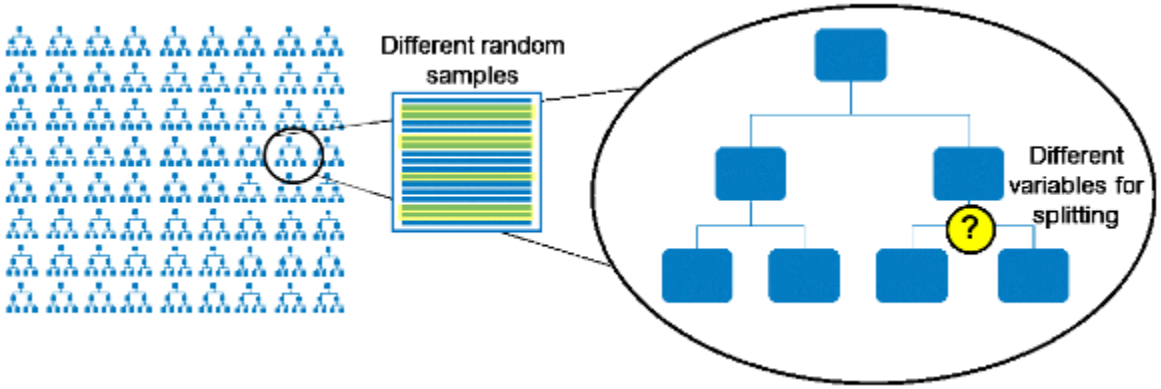
学習しすぎて応用が利かない。学習が足りずに誤差だらけ。

・適切な値は試行が必要

→ランダムフォレストを例にパラメータを考察する。

ランダムフォレストとは

1. 学習データからランダムにサンプリングをする（デフォルト：60%）
2. ツリー分割に使用する変数をランダムに選出する。（**デフォルト：すべて**）
3. ツリーの深さ (デフォルト：50) とノードに含まれる最小オブザベーション(デフォルト：1）. の深いツリーを作成する。
4. 学習データをリサンプリングし、ツリーを複数作る（**デフォルト100**）
5. 各ツリーの結果を平均化して予測値を算出する。



チューニングの方法

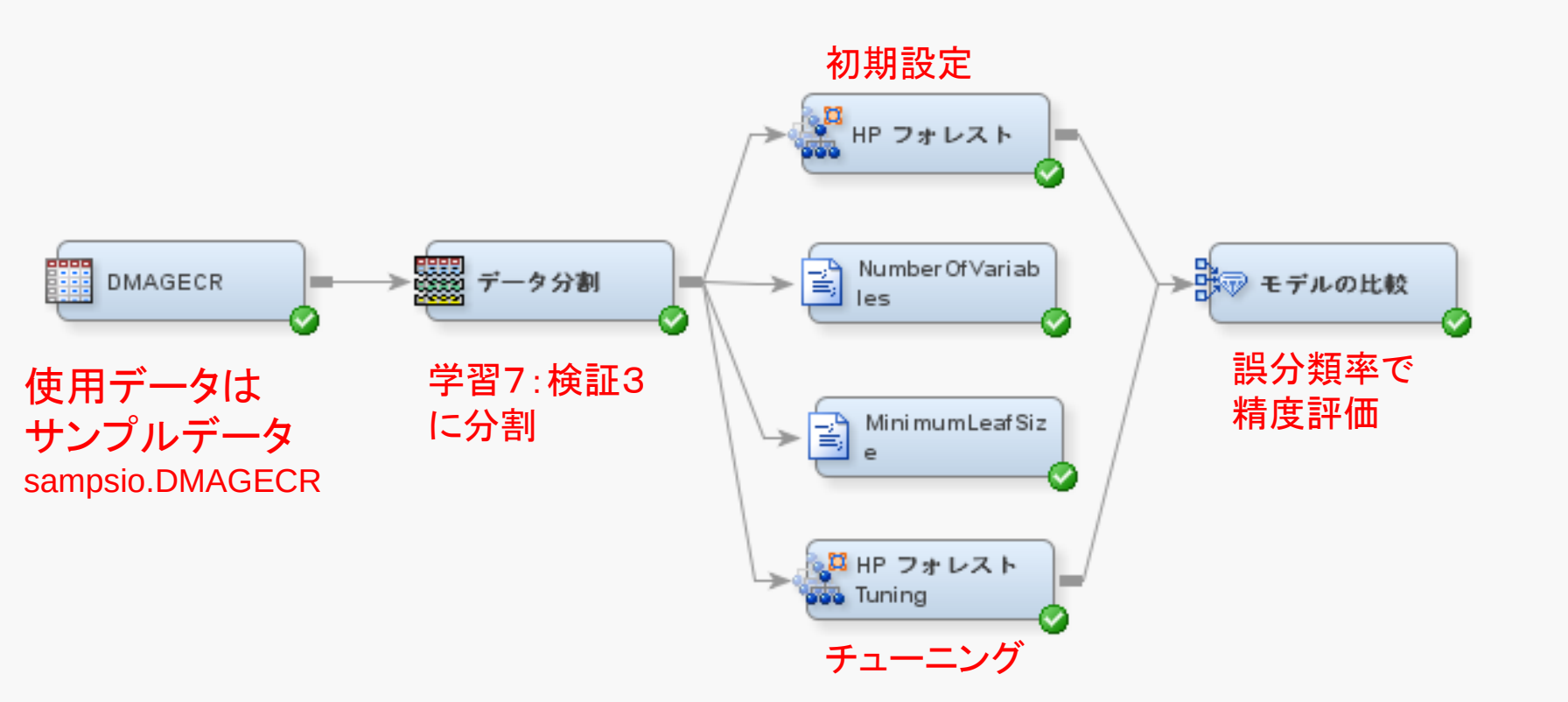
communities.sas.com に掲載されている

“Tip: Getting the Most from your Random Forest”

https://communities.sas.com/docs/DOC-11369

に則ってチューニングを行い、初期設定に対してどれぐらい改善されるか検証する。

検証方法


初期設定
HP フォレスト
DMAGECR
データ分割
Number Of Variables
モデルの比較
使用データは
サンプルデータ
sampsio.DMAGECR
学習7：検証3
に分割
Minimum Leaf Size
誤分類率で
精度評価
HP フォレスト Tuning
チューニング

チューニングパラメータは3つ

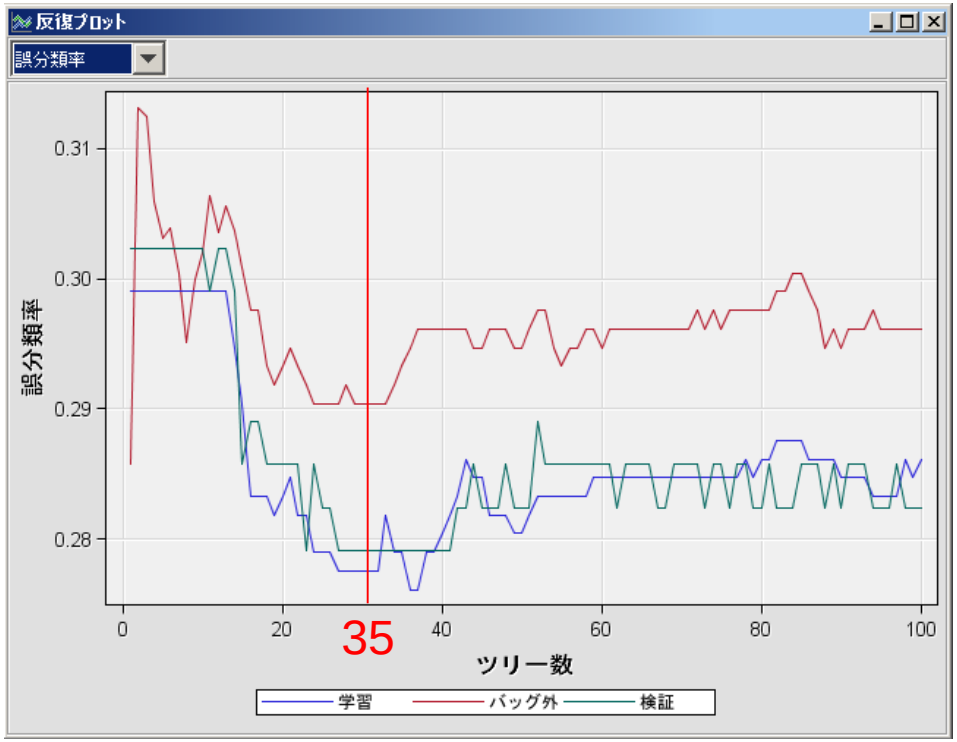
学習	
変数	
ツリーオプション	
最大ツリー数	100
シード	12345
サンプルの種類	比率
サンプルのオブザベーションの比率	0.6
サンプルのオブザベーション数	.
分割ルールオプション	
最大の深さ	50
欠損値	検索に使用する
検索での最小使用数	1
分割検索で対象となる変数の数	.
有意水準	0.05
分割検索における最大カテゴリ数	30
最小カテゴリサイズ	5
全数	5000
ノードオプション	
葉のサイズの決定法	デフォルト
ノードに含まれるオブザベーションの最小パーセント	1.0E-5
ノードに含まれるオブザベーションの最小数	1
分割サイズ	.

1. ツリーを最大いくつ作るか

2. 分割時に考慮する変数の数

3. ノードに含まれるオブザベーション

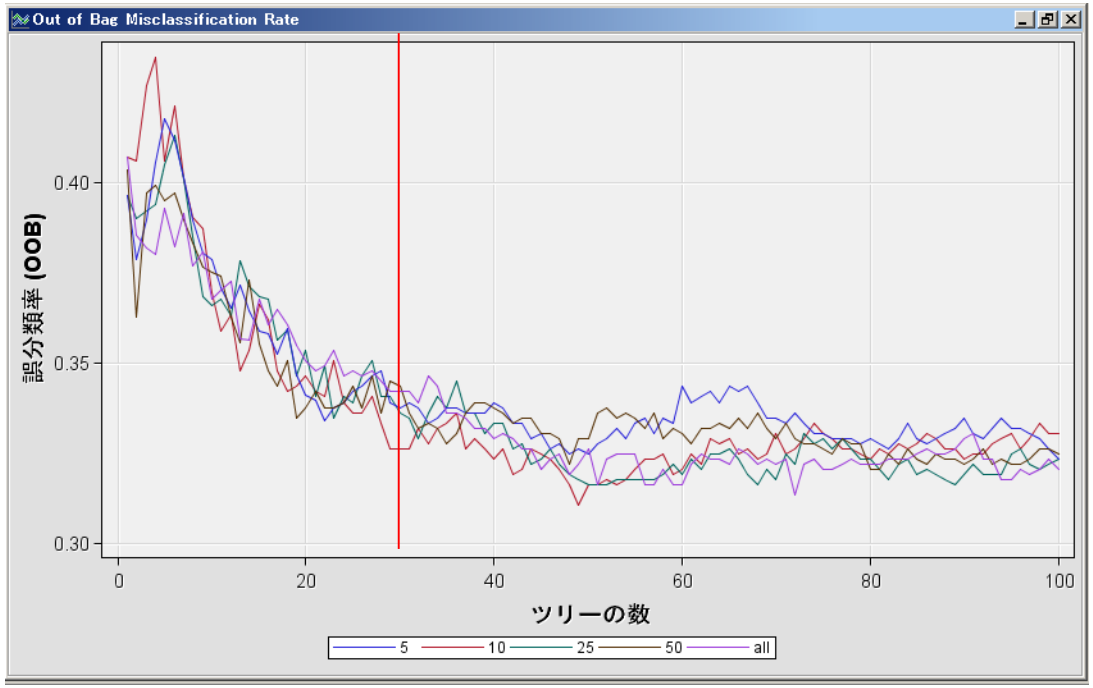
1. ツリーを最大いくつ作るか



HPフォレストノードを実行結果に
“反復プロット”が表示されます。
この画面より、ツリーの数はどの程度が
適切であるか読み取ります。

このデータでは、ツリーの数が多くなると
逆に誤分類率が高くなる傾向があるため、
学習、検証、バック外（OOB）の誤分類率
が最も少ない35をツリーの最大値とします。

2. 分割時に考慮する変数の数

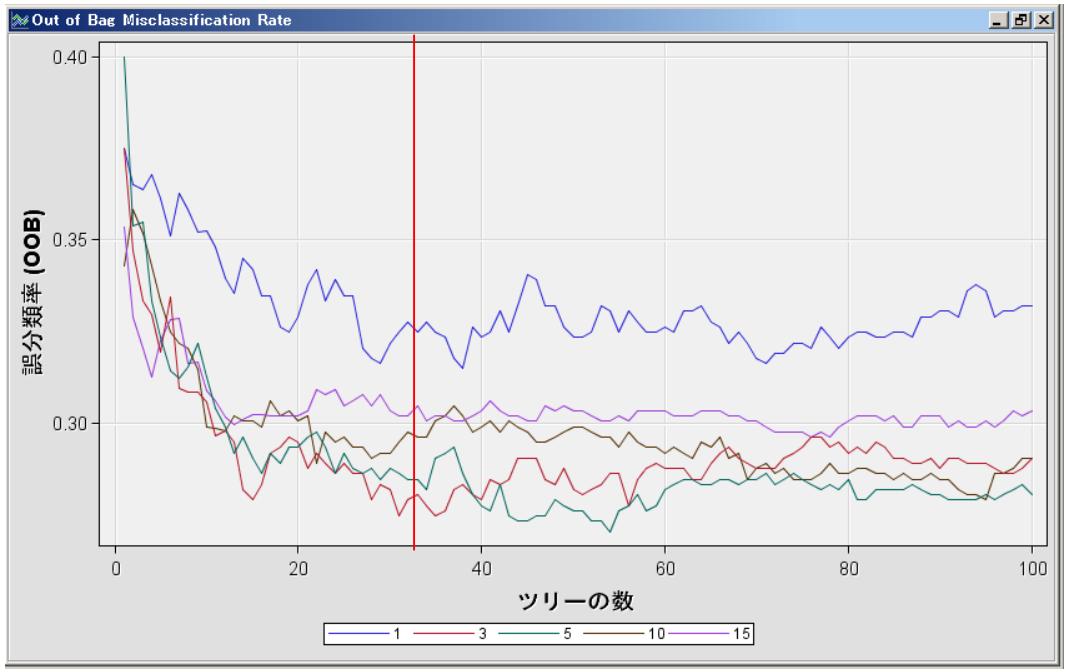


考慮する変数の数を
5、10, 25, 50、全て
の5パタンを試行するプログラムをSASコードノードで実行します。

このデータでは、ツリーの数が35あたりでは10変数が最も誤分類率が低くなっているので
10を考慮する変数数とします。

10

3. ノードに含まれるオブザベーション



ノードに含むオブザベーション（葉の最小数）を
1、3，5，10、15
の5パタンを試行するプログラムをSASコードノードで実行します。

このデータでは、ツリーの数が35あたりでは3オブザベーションが最も誤分類率が低くなっているので3を考慮する変数数とします。

パラメータを修正して再実行

学習	
変数	
ツリーオプション	
最大ツリー数	100
シード	12345
サンプルの種類	比率
サンプルのオブザベーションの比率	0.6
サンプルのオブザベーション数	.
分割ルールオプション	
最大の深さ	50
欠損値	検索に使用する
検索での最小使用数	1
分割検索で対象となる変数の数	.
有意水準	0.05
分割検索における最大カテゴリ数	30
最小カテゴリサイズ	5
全数	5000
ノードオプション	
葉のサイズの決定法	デフォルト
ノードに含まれるオブザベーションの最小パーセント	1.0E-5
ノードに含まれるオブザベーションの最小数	1
分割サイズ	.

学習	
変数	
ツリーオプション	
最大ツリー数	35
シード	12345
サンプルの種類	比率
サンプルのオブザベーションの比率	0.6
サンプルのオブザベーション数	.
分割ルールオプション	
最大の深さ	50
欠損値	検索に使用する
検索での最小使用数	1
分割検索で対象となる変数の数	10
有意水準	0.05
分割検索における最大カテゴリ数	30
最小カテゴリサイズ	5
全数	5000
ノードオプション	
葉のサイズの決定法	カウント
ノードに含まれるオブザベーションの最小パーセ	1.0E-5
ノードに含まれるオブザベーションの最小数	3
分割サイズ	.

実行結果とまとめ

選択モデル	先行ノード	選択基準: 検証: 誤分類率	学習: 誤分類率
Y	HPDMForest2	0.259136	0.234621
	HPDMForest	0.282392	0.286123

パラメータチューニングしたHPフォレストノードを実行した結果、学習、検証データともに誤分類率の改善が見られた。

検証データ:0.28→0.23
検証データ:0.28→0.25

SAS Enterprise Minerでは、データマイニング手法、機械学習手法が数多く実行できるが、
よりよい精度を得るには、
・どの手法を採用するか
・各手法のパラメータは最適値か
を検討し、予測モデルを決定する必要がある。
前者は“モデルの比較”ノードで判断できるが、パラメータ最適値については試行が必要であり、今回の方法は他の機械学習ノードでも適用できる方法なので、最適な予測モデル構築を目指す際にはぜひ試して頂きたい。

参考：分割時に考慮する変数の数プログラム

```
%macro hpforestStudy (nVarsList=10,maxTrees=200);

 %let nTries = %sysfunc(countw(&nVarsList.));
  /* Loop over all specified number of variables to try */
 %do i = 1 %to &nTries.;
  %let thisTry = %sysfunc(scan(&nVarsList.,&i));

  /* Run HP Forest for this number of variables */
  proc hpforest data=&em_import_data maxtrees=&maxTrees. vars_to_try=&thisTry.;
   input %EM_INTERVAL_INPUT /level=interval;
   target %EM_TARGET / level=binary;
   ods output fitstatistics=fitstats_vars&thisTry. ;
  run;
/* Append to the single cumulative fit statistics table */
   proc append base=fitStats data=fitstats_vars&thisTry.;
   run;
 %end;
%mend hpforestStudy;

%hpforestStudy(nVarsList=5 10 25 50 all,maxTrees=100);
/* Register the data set for use in the em_report reporting macro */
%em_register(type=Data,key=fitStats);
data &em_user_fitStats;
   set fitStats;
run;
%em_report(viewType=data,key=fitStats,autodisplay=y);
%em_report(viewType=lineplot,key=fitStats,x=nTrees,y=miscOOB,group=varsToTry,d
escription=Out of Bag Misclassification Rate,autodisplay=y);
```

参考：ノードに含まれるオブザベーションプログラム

```
%macro hpforestStudy (leafsizeList=5,maxTrees=200);

 %let nTries = %sysfunc(countw(&leafsizeList.));
 /* Loop over all specified number of variables to try */
 %do i = 1 %to &nTries.;
  %let thisTry = %sysfunc(scan(&leafsizeList.,&i));

   /* Run HP Forest for this number of variables */
   proc hpforest data=&em_import_data maxtrees=&maxTrees. leafsize=&thisTry.;
      input %EM_INTERVAL_INPUT /level=interval;
      target %EM_TARGET / level=binary;
      ods output fitstatistics=fitstats_vars&thisTry. ;
   run;
   /* Add the value of varsToTry for these fit stats */
   data fitstats_vars&thisTry.;
      length leafsize $ 8;
      set fitstats_vars&thisTry.;
      leafsize = "&thisTry.";
   run;

   /* Append to the single cumulative fit statistics table */
   proc append base=fitStats data=fitstats_vars&thisTry.;
   run;
  %end;
%mend hpforestStudy;

%hpforestStudy(leafsizeList=1 3 5 10 15,maxTrees=100);
/* Register the data set for use in the em_report reporting macro */
%em_register(type=Data,key=fitStats);
data &em_user_fitStats;
   set fitStats;
run;
%em_report(viewType=data,key=fitStats,autodisplay=y);
%em_report(viewType=lineplot,key=fitStats,x=nTrees,y=miscOOB,group=leafsize,description=Out of Bag Misclassification Rate,autodisplay=y);
```