

Paper 204-2019

Variable Selection using Random Forests in SAS

Denis Nyongesa, Kaiser Permanente Center for Health Research, Portland, Oregon, USA

ABSTRACT

Random forest is an increasingly used statistical method for classification and regression. It was introduced by Leo Breiman in 2001. A good prediction model begins with a great feature selection process. This paper proposes the ways of selecting important variables to be included in the model using random forest. The variables to be included in the model will be indexed or ranked according to the score of importance of each variable.

INTRODUCTION

The primary purpose of this paper is the use of random forests for variable selection. The variables to be considered for inclusion in a model can be ranked in order of their importance. The variable importance index (also known as gini index) based on random forests considers interaction between variables. This makes it a robust method to find important variables that can be used in a prediction model. This entails ranking of explanatory (independent) variables using the random forests score of importance.

Before delving into the subject of this paper, a review of random forests, variable importance and selection is helpful.

Random Forest

Breiman, L. (2001) defined a random forest as a classifier that consists a collection of tree-structured classifiers $\{h(x, \Theta_k), k = 1, \dots\}$ where the $\{\Theta_k\}$ are independent identically distributed random vectors and each tree casts a unit vote for the most popular class at input x .

For the k th tree, a random vector Θ_k is generated, which is independent of the past $\Theta_1 \dots \Theta_{k-1}$ random vectors.

Random forests are a combination of tree predictors such that each tree depends on the values of a random vector sampled independently and with the same distribution for all trees in the forest. The generalization error for forests converges almost surely to a limit as the number of trees in the forest becomes large.

The strength of the individual trees in the forest and the correlation between them determines the generalization error of a forest of tree classifiers (Breiman, 2001). Breiman, 2001 reiterates that using a random selection of features to split each node yields error rates that compare favorably to Adaboost (short for "Adaptive Boosting", proposed by Freund and Schapire in 1996, is a practical boosting algorithm focusing on classification problems. It aims to create a strong classifier by converting a set of weak classifiers) but are more robust with respect to noise.

The internal estimates measure the error, strength, correlation, and variable importance. The effect of increasing the number of features used in the splitting can also be shown by the corresponding changes in the internal estimates (i.e. error, strength and correlation). These ideas are as useful in regression as they are in classification.

The performance of random forests is related to the quality of each tree in the forest. The trees of the forest tend to have small correlation, if not uncorrelated. The correlation between trees is correlation of predictions on the out-of-bag (OOB) samples. The OOB sample refers to a set of those observations not used in building the current tree. The OOB sample is used to estimate the prediction error and evaluation of variable importance (Breiman 2001, Genuer R., et al 2010).

For this paper, the random forest performance will focus on both the training and OOB samples. This is necessary not only for comparison purposes (between training and OOB samples) but also to demonstrate how the model will perform on the validation sample (the sample not used to train the model) for generalization. They also happen to be among the default outputs created by most analytical software.

Variable Importance

Most statistical procedures for regression and classification measure variable importance indirectly by selecting variables using some criteria such as statistical significance, Schwarz Information Criterion (SIC) and Akaike's Information Criterion. The approach taken in RF is completely different. For each tree in the forest, there is a misclassification rate for the out-of-bag observations. To assess the importance of a specific predictor variable, the values of the variable are randomly permuted for the out-of-bag observations, and then the modified out-of-bag data are passed down the tree to get new predictions. The difference between the misclassification rate for the modified and original out-of-bag data, divided by the standard error, is a measure of the importance of the variable (Cutler et al. 2007).

For a set of p potential predictor variables and n observations in a sample. The measure of node impurity, given by the mean square error (MSE) for a node is; $I(\Omega) = \frac{1}{n_\Omega} \sum_{i \in \Omega} (y_i - \bar{y})^2$, where n_Ω is the number of individuals in the node Ω , y is the response variable, and \bar{y} is the mean of the corresponding elements of y . The best split is the one that maximizes the reduction in the node impurity given by, $\phi = I(\Omega) - I(\Omega_L) - I(\Omega_R)$, where $I(\Omega_L)$ = left child node of Ω and $I(\Omega_R)$ = right child node of Ω (Foulkes 2009).

Unlike the bagging method which grows each tree by considering all p predictors from the data set with ensembles that are highly correlated, the random forest method considers m predictors, $m < p$, from the total set of predictors. By randomly selecting a subset of predictors, the correlation of the trees in an ensemble is reduced, leading to a greater reduction in variance for the random forest model compared to simple bagging. Breiman (2001) proved that random forests do not overfit the data, even for a very large number of trees, an advantage over classification and regression trees (CART). Random forest decision boundaries tend to be axis-oriented due to the nature of the tree decision boundaries, but the ensemble voting allows for much more dynamic boundaries than sharp rectilinear edges.

The main parameters for the random forest method are the number of trees to grow, $ntree$, and the number of predictors to try per tree, $mtry$. These values can be chosen to minimize the estimated classification error using a cross-validation (if deemed necessary – RF have their in-built validation procedure). For classification problems, the default parameters are $ntree = 500$, and $mtry = \sqrt{p}$. These default values have been shown to have good results consistently across a variety of classification problems (Ahn and Moon 2010).

Selection of useful variables

There are many variable selection methods that incorporate the importance of the features. Rakotomanonjy (2003), proposes support vector machines (SVM) scores criterion based on weight vector or generalization error bounds sensitivity with respect to a variable. Poggi, et al (2006) suggests the selection of useful variables using a stepwise strategy that involves successive applications of the CART method.

Bruce, et al (2017) discusses the use of chi-square scores and p-values in selecting variables to include in logistic regression models. They demonstrated how PROC LOGISTIC/HPLOGISTIC/HPGENSELECT/GLMSELECT with SELECTION=SCORE/FORWARD (SELECT=SBC CHOOSE=SBC)/LASSO(CHOOSE=SBC) can be used to select the predictors or variables to be included in the model. The best predictors are ones in the model with highest score chi-square. The SELECTION=SCORE has a drawback; it does not support the CLASS statement. Converting a classification problem to a set of dummy variables is the suggested mitigation measure. In SELECTION=FORWARD (SELECT=SBC CHOOSE=SBC), a predictor is selected at each step with the most significant score chi-square. The assumption to this criterion is the significance level satisfies the threshold which the user sets via SLENTRY.

Hosmer, et al. (2013) suggests use of FORWARD, BACKWARD, and STEPWISE criteria in logistic regression as a variable selection as well as a data exploration method.

While some of the above techniques helps in obtaining the list of important variables to be included in the model to improve predictor performance, sophisticated wrapper or embedded methods and simpler variable ranking methods like correlation methods, may suffer from the curse of dimensionality, and multivariate methods may overfit the data in domains with large numbers of input variables (Guyon, et al, 2003).

This paper demonstrates how variable importance using Random Forest with focus on OOB error can be used to select the most useful variables. It is also important to note that the method assesses prediction performance of the variables in its selection criteria and is not prone to overfitting.

DATA

To demonstrate the objective of this paper, Sashelp.JunkMail data was used. The Sashelp.JunkMail data set comes from a study that classifies whether an email is junk email (1 = junk/spam, 0 = not junk/not spam) (Asuncion and Newman 2007). The data were collected in Hewlett-Packard labs and donated by George Forman. The data set contains 4,601 observations with 59 variables. The response variable (class) is a binary indicator of whether an email is considered spam or not. Out of the 4601 observations, 1,813 are classified as junk and 2,788 as not junk. There are 57 predictor variables that record the frequencies of some common words and characters and lengths of uninterrupted sequences of capital letters in emails. The other two are test and class. Test, a binary variable, was not used in this analysis.

The following step displays information about the data set Sashelp.junkmail:

```
title "Sashelp.junkmail : Classifying Email as Junk Or Not Junk";
proc contents data=sashelp.junkmail varnum;
ods select position;
run;
```

Table 1 below displays the information about the data set Sashelp.junkmail. It shows the first and last five variables, in the order they were created.

Table 1: First and last five Observations of the data set Sashelp.junkmail

Variables in Creation Order				
#	Variable	Type	Len	Label
1	Test	Num	8	0 - Training, 1 - Test
2	Make	Num	8	
3	Address	Num	8	
4	All	Num	8	
5	_3D	Num	8	3D
.				
.				
.				
55	Pound	Num	8	
56	CapAvg	Num	8	Capital Run Length Average
57	CapLong	Num	8	Capital Run Length Longest
58	CapTotal	Num	8	Capital Run Length Total
59	Class	Num	8	0 - Not Junk, 1 - Junk

First and last five variables created using the CONTENT procedure. The variables are in the order they were created.

To view the first ten observations of the Sashelp.junkmail dataset, use the following step:

```
title "The First Ten Observations Out of 4,601";
proc print data=sashelp.junkmail (obs=10) noobs;
run;
```

RANDOM FOREST – THE HIGH-PERFORMANCE PROCEDURE

The SAS Code below calls the random forest (High-Performance). Given that it is a classification problem, the target (response/dependent) variable is classified as nominal so that it is treated as a categorical variable. All the predictor/independent variables are continuous, thus level=interval. The High-Performance random forest procedure may have more than one input statement depending on the levels of the variables available to be included in the model.

Overview of the various options in SAS Enterprise Miner 14.3 High-Performance Procedures Documentation:

- **maxtrees** specifies the maximum number of trees.
- **vars_to-try** specifies the randomized number inputs to select at each node.
- **seed** sets the randomization seed for bootstrapping and feature selection.
- **trainfraction** specifies the fraction of the original observations used for bootstrapping each tree.
- **leafsize** indicates the minimum number of observations allowed in each branch.
- **alpha** specifies the p-value threshold a candidate variable must meet for a node to be split.
- **maxdepth** specifies the number of splitting rules for the nodes.
- **preselect** indicates the method of selecting a splitting feature

To predict an observation using RF, the HPFOREST procedure assigns the observation to a single leaf in each tree in the forest. Based on the tree that contains that leaf, the leaf makes a prediction. If the response variable is an interval (continuous variable), the prediction is equal to the average in that leaf. For a categorical (nominal) variable, the predicted class is the class with the largest posterior probability. For a tie, in case of one, the first class that occurs in the training data becomes the predicted class.

```

ods trace on;
proc hpforest data=sashelp.junkmail maxtrees=1000 vars_to_try=10 seed=1985
trainfraction=0.7 maxdepth=50 leafsize=6 alpha=0.5;
  target class /level=nominal;
  input Make Address All _3D Our Over Remove Internet Order Mail Receive
  Will People Report Addresses Free Business
  Email You Credit Your Font _000 Money HP HPL George _650 Lab Labs
  Telnet _857 Data _415 _85 Technology _1999
  Parts PM Direct CS Meeting Original Project RE Edu Table Conference
  Semicolon Paren Bracket Exclamation Dollar Pound CapAvg CapLong
  CapTotal / level = interval;
ods output FitStatistics = fit_at_runtime;
ods output VariableImportance = Variable_Importance;
ods output Baseline = Baseline;
run;
ods trace off;

```

Table 2 below shows first and last five observations for the fit statistics ranked by the misclassification error of the OOB sample obtained by running the above SAS code.

Table 2: HPFOREST procedure Fit Statistics

Fit Statistics							
Number of Trees	Number of Leaves	Average Square Error (Train)	Average Square Error (OOB)	Misclassification Rate (Train)	Misclassification Rate (OOB)	Log Loss (Train)	Log Loss (OOB)
1	147	0.0526	0.0737	0.0722	0.0978	0.291	0.657
2	298	0.0432	0.0714	0.0563	0.0916	0.160	0.616
3	450	0.0389	0.0690	0.0463	0.0895	0.136	0.596
4	607	0.0370	0.0675	0.0433	0.0862	0.134	0.551
5	763	0.0358	0.0657	0.0437	0.0831	0.132	0.482
.
.
.
996	148464	0.0292	0.0448	0.0313	0.0548	0.122	0.170
997	148611	0.0292	0.0448	0.0313	0.0548	0.122	0.170
998	148769	0.0292	0.0448	0.0313	0.0548	0.122	0.170
999	148929	0.0292	0.0448	0.0313	0.0550	0.122	0.170
1000	149061	0.0292	0.0448	0.0313	0.0550	0.122	0.170

First and last five observations obtained using the ODS OUTPUT FitStatistics statement.

Figures 1 and 2 shows the average square error (ASE) and the misclassification error (ME) plots for the training and OOB samples.

The ASE and ME from the training sample are slightly higher than those obtained from the OOB sample. This is no surprise; the difference manifested by this plot were expected.

The following SAS codes creates figures 1 and 2 respectively:

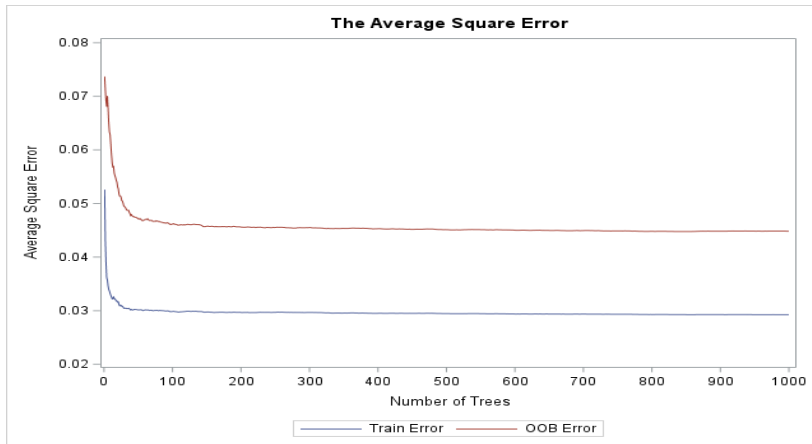
```

title "The Average Square Error";
proc sgplot data = fit_at_runtime;
  series x=NTrees y=PredAll/legendlabel='Train Error';
  series x=NTrees y=PredOOB/legendlabel='OOB Error';
  xaxis values=(0 to 1000 by 50);
  yaxis values=(0.02 to 0.08 by 0.01) label='Average Square Error';
run;

title "The Misclassification Error";
proc sgplot data = fit_at_runtime;
  series x=NTrees y=MiscAll/legendlabel='Train Misclassification Error';
  series x=NTrees y=MiscOOB/legendlabel='OOB Misclassification Error';
  xaxis values=(0 to 1000 by 50);
  yaxis values=(0.02 to 0.10 by 0.01) label='Misclassification Error';
run;

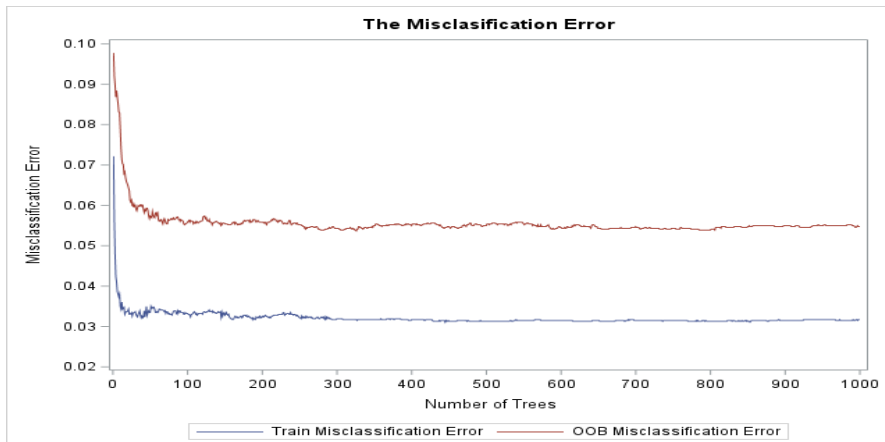
```

Figure 1. The Average Square Error (ASE)



Although 1000 trees were specified in the model, the minimum ASE seems to be attained at around 50 trees. Having many trees doesn't impact the model performance, it impacts the time of run. For example, specifying 50 trees had 1.21 and 1.89 seconds for real and cpu times respectively while it was 22.79 and 34.45 seconds for real and cpu times respectively when 1000 trees were specified. A clear visual picture of the behavior of ASE can be observed as trees increase.

Figure 2. The Misclassification Error



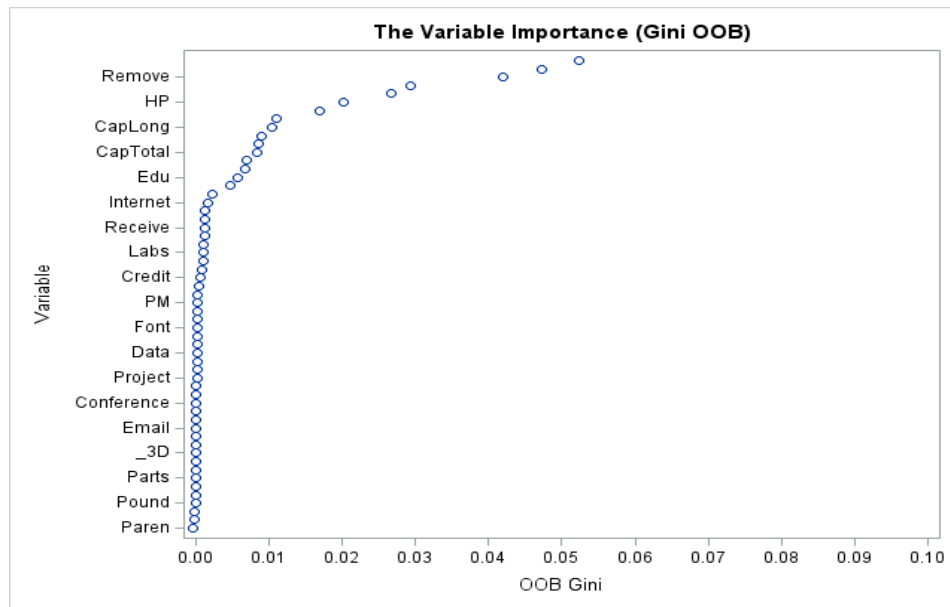
VARIABLE IMPORTANCE

The important variables, ranked in the order of importance (from most to least important) are: Dollar, Exclamation, Remove, Free, Your, HP, Money, _000, CapLong, Our, George, CapTotal, CapAvg, HPL, Edu, _1999, Business, Internet, You, RE, Receive, All, Meeting, Labs, _650, _85, Credit, Address, Over, PM, Order, Telnet, Font, Technology, Bracket, Data, Lab, Original, Project, _415, Semicolon, Conference, Addresses, CS, Email, Direct, _857, _3D, Table, Report, Parts, People, Make, Pound, Will, Mail, Paren.

Table 3. Loss Reduction Variable Importance

Variable	Number of Rules	Gini	OOB Gini	Margin	OOB Margin
Dollar	4529	0.054110	0.05164	0.108219	0.105466
Exclamation	9562	0.053506	0.04755	0.107011	0.100780
Remove	3376	0.042182	0.04192	0.084364	0.083885
Free	5269	0.032914	0.02976	0.065828	0.062424
Your	9137	0.031830	0.02626	0.063660	0.058150
.
.
.
Make	880	0.000355	-0.00002	0.000709	0.000346
Pound	777	0.000355	-0.00006	0.000709	0.000279
Will	5670	0.002439	-0.00015	0.004879	0.002206
Mail	2664	0.001126	-0.00019	0.002252	0.000940
Paren	7296	0.002628	-0.00047	0.005257	0.002140

Figure 3. The plot of Variables Ranked in the order of Importance



The top sixteen variables seem to be enough for the model. That is the point where the slope change is more noticeable. They are more likely to result in a parsimonious prediction model. The rest of the variables have about the same effect to the model and their inclusion will result into negligible effect to the predictive power of the model.

MODEL COMPARISON

For model comparison, I selected the top sixteen variables each obtained by RF variable importance and LOGISTIC regressions STEPWISE method and ran two separate RF models. The top sixteen variables ranked by score chi-square are: Your, Remove, _000, Dollar, Free, Exclamation, CapTotal, HP, Our, Business, Money, Meeting, RE, George, CapLong, and Internet. Twelve variables were common between the two methods of variable selection, only four were different.

Table 4. Summary Statistics of Average Square and Misclassification Errors obtained by Variable Selection by RF and STEPWISE techniques

	RF Model Performance (Variables selected using RF)		RF Model Performance (Variables selected using STEPWISE method)	
	Mean (std)	Median (Q1, Q3)	Mean (std)	Median (Q1, Q3)
Average Square Error (Train)	0.028(0.001)	0.027(0.027,0.027)	0.029(0.001)	0.029(0.029,0.029)
Average Square Error (OOB)	0.047(0.002)	0.047(0.047,0.047)	0.049(0.002)	0.048(0.048,0.049)
Misclassification Error (Train)	0.034(0.001)	0.033(0.033,0.034)	0.035(0.001)	0.035(0.034,0.035)
Misclassification Error (OOB)	0.061(0.002)	0.061(0.061,0.062)	0.062(0.002)	0.062(0.061,0.062)

It is apparent from table 4 above that variables selected by RF consistently performs better on both training and OOB data samples. With the current era of data abundance, where daily data generated is increasing exponentially, such techniques of selecting most important variables are greatly needed. I expect the performance differ by a wide margin for extremely large datasets with thousands of variables.

CONCLUSION

The random forests' variable importance is easy, and not only does it find the important variables, but also ranks them in the order of importance. It also provides an opportunity for one to decide whether to use gini (from training sample) or the out-of-bag gini values. The number of variables to be included in the final model depends on what the model is be used to predict and specific key variables needed (despite their ranking).

REFERENCES

Asuncion, A. and D.J. Newman. 2007. UCI Machine Learning Repository. Irvine, CA: University of California, School of Information and Computer Science.
<http://www.ics.uci.edu/~mllearn/MLRepository.html>

Breiman, L. (2001). "Random Forests." *Machine Learning* 45:5–32.

Foulkes, Andrea S. 2009. Applied Statistical Genetics with R: For Population-Based Association Studies. New York: Springer

Genuer, R., Poggi, J.M. and C. Tuleau-Malot. 2010. Variable selection using Random Forests. Pattern Recognition Letters, Elsevier, 2010, 31 (14), pp.2225-2236. fhal-00755489. Accessed July, 2019. <https://hal.archives-ouvertes.fr/hal-00755489/>

Guyon, I. and A. Elisseeff. 2003. An Introduction to Variable and Feature Selection. Journal of Machine Learning Research. 3, 1157-1182.

Hosmer D., Lemeshow S., and Sturdivant R. 2013. *Applied Logistic Regression*. 3rd Ed. New York. NY: John Wiley & Sons.

Nyongesa, D.B. 2016. Various Considerations on Performance Measures for a Classification of Ordinal Data." Accessed June, 2019. <https://pqdtopen.proquest.com/doc/1810990473.html?FMT=AI>

Poggi, J.M. and C. Tuleau, 2006. Classification supervisée en grande dimension. Application à l'agrément de conduite automobile. Revue de Statistique Appliquée. LIV (4), 39-58.

Rakotomamonjy, A., 2003. Variable selection using SVM-based criteria. Journal of Machine Learning Research. 3, 1357-1370.

SAS Institute Inc. "The Logistic Procedure." Accessed July, 2019. https://support.sas.com/documentation/cdl/en/statug/63033/HTML/default/viewer.htm#logistic_toc.htm

SAS Institute Inc. 2018. "Sashelp Data Sets." Accessed July, 2019. <https://support.sas.com/documentation/tools/sashelpug.pdf>

CONTACT INFORMATION

Your comments and questions are valued and encouraged. Contact the author at:

Denis B. Nyongesa
Kaiser Permanente Center for Health Research
denis.b.nyongesa@kpchr.org

SAS and all other SAS Institute Inc. product or service names are registered trademarks or trademarks of SAS Institute Inc. in the USA and other countries. ® indicates USA registration.

Other brand and product names are trademarks of their respective companies.

/*FULL SAS CODE*/

```
title "Sashelp.junkmail : Classifying Email as Junk Or Not Junk";
proc contents data=sashelp.junkmail varnum;
ods select position;
run;

title "The First Ten Observations Out of 4,601";
proc print data=sashelp.junkmail(obs=10) noobs;
run;

proc freq data=sashelp.junkmail;
tables class test test*class/missing;
run;

/*RANDOM FOREST(RF): initial model to use for variable selection*/
title "The Random Forest";
ods trace on;
```



```

proc hpforest data=sashelp.junkmail maxtrees=1000 vars_to_try=10 seed=1985
trainfraction=0.7 maxdepth=50 leafsize=6 alpha=0.5;
  target class /level=nominal;;
  input Make Address All _3D Our Over Remove Internet Order Mail Receive
Will People Report Addresses Free Business
      Email You Credit Your Font _000 Money HP HPL George _650 Lab Labs
Telnet _857 Data _415 _85 Technology _1999
      Parts PM Direct CS Meeting Original Project RE Edu Table Conference
Semicolon Paren Bracket Exclamation Dollar
      Pound CapAvg CapLong CapTotal / level = interval;
  ods output FitStatistics = fit_at_runtime;
  ods output VariableImportance = Variable_Importance;
  ods output Baseline = Baseline;
run;
ods trace off;

title;
proc print data=Variable_Importance (obs=10) ; run;
proc print data=fit_at_runtime (obs=10) label; run;

proc means data=fit_at_runtime maxdec=3 n nmiss mean std q1 median q3 qrange
min max range;
var PredAll PredOob MiscAll MiscOob;
run;

/*VARIABLE IMPORTANCE */
title "The Variable Importance (Gini OOB)";
proc sgplot data = Variable_Importance;
scatter X = GiniOOB Y = variable / legendlabel='GiniOOB';
xaxis values=(0.00 to 0.10 by 0.01);
yaxis type=discrete discreteorder= data reverse display=all
; /*fitpolicy=splitalways : to display all the tickmarks on y-axis*/
run;

/*TRAINING ERROR, MISCLASSIFICATION RATES, & LOG LOSSES BY NUMBER OF TREES*/
title "The Average Square Error";
proc sgplot data = fit_at_runtime;
series X = NTrees Y = PredAll / legendlabel='Train Error';
series X = NTrees Y = PredOOB / legendlabel='OOB Error';
xaxis values=(0 to 1000 by 50);
yaxis values=(0.02 to 0.08 by 0.01) label='Average Square Error';
run;

title "The Misclassification Error";
proc sgplot data = fit_at_runtime;
series X = NTrees Y = MiscAll / legendlabel='Train Misclassification Error';
series X = NTrees Y = MiscOOB / legendlabel='OOB Misclassification Error';
xaxis values=(0 to 1000 by 50);
yaxis values=(0.02 to 0.10 by 0.01) label='Misclassification Error';
run;

/*LOGISTIC REGRESSION: Model to use for variable selection*/
title 'Stepwise Regression on Junkmail Data';
proc logistic data=sashelp.junkmail desc outest=betas covout;
  class class(param=ref ref='1');
  model class(event='1') = Make Address All _3D Our Over Remove Internet
Order Mail Receive Will People Report Addresses Free Business

```

```

    Email You Credit Your Font _000 Money HP HPL George _650 Lab Labs
Telnet _857 Data _415 _85 Technology _1999
    Parts PM Direct CS Meeting Original Project RE Edu Table Conference
Semicolon Paren Bracket Exclamation Dollar
    Pound CapAvg CapLong CapTotal/ selection=stepwise
        slentry=0.1
        slstay=0.1
        pprob=0.5
        details
        lackfit;
    output out=pred p=phat lower=lcl upper=ucl
        predprob=(individual crossvalidate);
run;

/*FINAL MODELS (Top 16 variables)*/
/*RF Model 1 Variables selected using VARIABLE IMPORTANCE in RANDOM FOREST)*/
proc hpforest data=sashelp.junkmail maxtrees=1000 vars_to_try=10 seed=1985
trainfraction=0.7 maxdepth=50 leafsize=6 alpha=0.5;
    target class /level=nominal;;
    input Dollar Exclamation Remove Free Your HP Money _000 CapLong
        Our CapTotal George HPL CapAvg Edu _1999 / level =
interval;
    ods output FitStatistics = fit_at_runtime_RF;
    ods output VariableImportance = Variable_Importance_RF;
    ods output Baseline = Baseline_RF;
run;
ods trace off;

proc print data=Variable_Importance_RF (obs=20) ; run;
proc print data=fit_at_runtime_RF (obs=20); run;

proc means data=fit_at_runtime_RF maxdec=3 n nmiss mean std q1 median q3
qrange min max range;
var PredAll PredOob MiscAll MiscOob ;
run;

/*RF Model 2 (Variables selected using STEPWISE in LOGISTIC REGRESSION)*/
proc hpforest data=sashelp.junkmail maxtrees=1000 vars_to_try=10 seed=1985
trainfraction=0.7 maxdepth=50 leafsize=6 alpha=0.5;
    target class /level=nominal;;
    input Your Remove _000 Dollar Free Exclamation CapTotal HP Our Business
Money Meeting RE George CapLong Internet / level = interval;
    ods output FitStatistics = fit_at_runtime_LR;
    ods output VariableImportance = Variable_Importance_LR;
    ods output Baseline = Baseline_LR;
run;
ods trace off;

proc print data=Variable_Importance_LR (obs=20) ; run;
proc print data=fit_at_runtime_LR (obs=20); run;

proc means data=fit_at_runtime_LR maxdec=3 n nmiss mean std q1 median q3
qrange min max range;
var PredAll PredOob MiscAll MiscOob;
run;

/*RF Model 1: VARIABLE IMPORTANCE */

```

```

title "The Variable Importance (Gini OOB) RF model1";
proc sgplot data = Variable_Importance_RF;
scatter X = GiniOOB Y = variable / legendlabel='GiniOOB';
xaxis values=(0.00 to 0.10 by 0.01);
yaxis type=discrete discreteorder= data reverse display=all ;
run;

/*RF Model 1: TRAINING ERROR & MISCLASSIFICATION RATES BY NUMBER OF TREES*/
title "The Average Square Error RF model1";
proc sgplot data = fit_at_runtime_RF;
series X = NTrees Y = PredAll / legendlabel='Train Error';
series X = NTrees Y = PredOOB / legendlabel='OOB Error';
xaxis values=(0 to 1000 by 50);
yaxis values=(0.02 to 0.08 by 0.01) label='Average Square Error';
run;

title "The Misclassification Error RF model1";
proc sgplot data = fit_at_runtime_RF;
series X = NTrees Y = MiscAll / legendlabel='Train Misclassification Error';
series X = NTrees Y = MiscOOB / legendlabel='OOB Misclassification Error';
xaxis values=(0 to 1000 by 50);
yaxis values=(0.02 to 0.10 by 0.01) label='Misclassification Error';
run;

/*RF Model 2: VARIABLE IMPORTANCE */
title "The Variable Importance (Gini OOB) RF model2";
proc sgplot data = Variable_Importance_LR;
scatter X = GiniOOB Y = variable / legendlabel='GiniOOB';
xaxis values=(0.00 to 0.10 by 0.01);
yaxis type=discrete discreteorder= data reverse display=all ;
run;

/*RF Model 2: TRAINING ERROR & MISCLASSIFICATION RATES BY NUMBER OF TREES*/
title "The Average Square Error RF model2";
proc sgplot data = fit_at_runtime_LR;
series X = NTrees Y = PredAll / legendlabel='Train Error';
series X = NTrees Y = PredOOB / legendlabel='OOB Error';
xaxis values=(0 to 1000 by 50);
yaxis values=(0.02 to 0.08 by 0.01) label='Average Square Error';
run;

title "The Misclassification Error RF model2";
proc sgplot data = fit_at_runtime_LR;
series X = NTrees Y = MiscAll / legendlabel='Train Misclassification Error';
series X = NTrees Y = MiscOOB / legendlabel='OOB Misclassification Error';
xaxis values=(0 to 1000 by 50);
yaxis values=(0.02 to 0.10 by 0.01) label='Misclassification Error';
run;

```