Paper JM-04

JMP® Pro Bootstrap Forest

George J. Hurley, The Hershey Company, Hershey, PA

Abstract

JMP Pro includes a number of analytical features that are very powerful, including a technique called "Bootstrap Forest". The Bootstrap Forest uses many decision tree type classification models, based on data and variable subsets to determine an optimal model. Through this bootstrapping methodology, a superior model can typically be generated relative to typical decision tree partitioning methods. Generally speaking, most applications of classification that would use a typical decision tree can use the bootstrap forest method; hence there is wide applicability of this method across industries. This paper will focus on how to use JMP Pro to perform this analysis, as well as some potential applications of it. The paper is not intended to be a theoretical explanation of this method.

Sample Data

JMP Pro installs with a number of built in datasets. These should be available to anyone who is using JMP. To access these files, you go to the "Help" menu and select "Sample Data". Throughout this paper, various datasets from the sample data will be referenced. All examples in this paper will make use of JMP Pro.

Bootstrap Forest

The bootstrap forest method is available in JMP Pro. Bootstrap Forest is a method that creates many decision trees and in effect averages them to get a final predicted value. Each tree is created from its own random sample, with replacement. The method also limits the splitting criteria to a randomly selected sample of columns.[1]

Due to the nature of this methodology, in most instances where a decision tree is applicable, the bootstrap forest method is also an option. This author finds the method particularly useful for data mining and predictive modeling and will leverage these methods for examples.

Titanic Example: Part 1

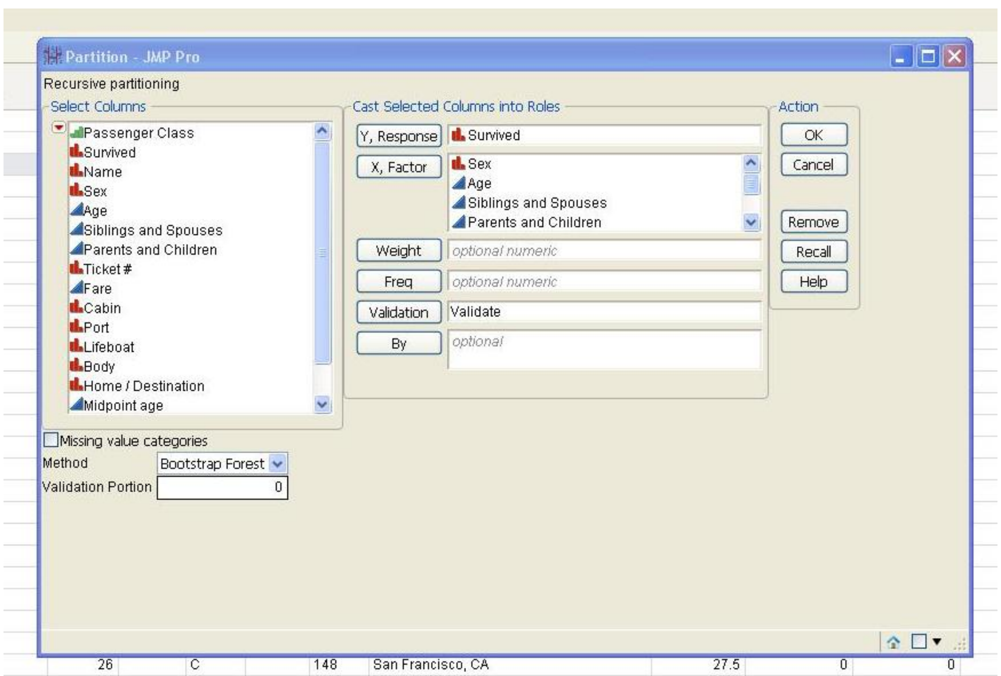
Here we look at using the bootstrap forest method to investigate drivers of survival for the passengers of the Titanic.

The Titanic data is accessed under “Help”, “Sample Data”, “Examples for Teaching”, “Titanic Passengers”. The dataset consists of one record per passenger who traveled on the ill-fated voyage of the Titanic, their demographic information, information on their home and destination, their passenger class, fare, cabin, and other information. A new nominal variable is also created here, called “On A Boat”, it represents if the passenger got on any lifeboat at all. If the variable “Lifeboat” is blank, then “On A Boat” is assigned 1; otherwise it is 0.

First, it is typically useful to create a validation dataset prior to running any of the partition methods. To do so, choose “Cols”, “New Column”. Typically, this column is named “Validate”, but does not have to be. Data can be initialized by selection “Random” under “Initialize Data”. A random indicator is used for this type of analysis. Hence it should be chosen. In JMP Pro, rows with 0 are used for model training, rows with 1 for validation, and rows with 2 for testing. Choosing the right percentage of each can be somewhat tricky and is dependent on dataset size; this author often begins with an allocation of 0.7 to training and 0.15 to each other classification.

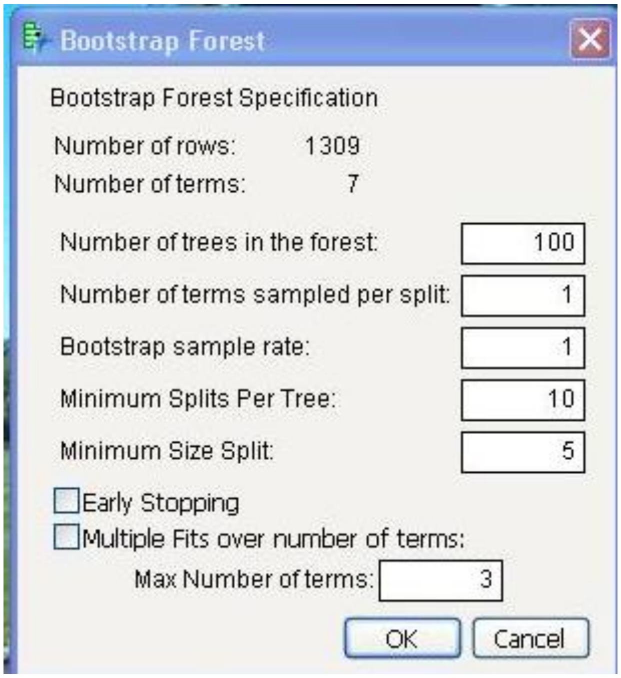
To run a bootstrap forest on this data, select “Analyze”, “Model”, “Partition” to bring up the partition menu (See Figure 1).

Figure 1: Screenshot of Partition Window Selecting Bootstrap Forest



Here, the variables “Sex”, “Age”, “Siblings and Spouses”, “Parents and Children”, “Fare”, “Port”, and “On A Boat” are chosen as the independent variables, with the response variable being “Survived”. For Validation, “Validate” is selected. Validation Portion is left at 0, because a validation column, “Validate” was chosen. Once this is selected, the bootstrap forest window will appear (See Figure 2).

Figure 2: Screenshot of Bootstrap Forest Menu



The bootstrap forest menu states the number of rows and terms (dependent/factor variables) chosen. Several options are then given. Per the JMP Pro 10.0.0 help file[1], they are:

Number of trees in the forest *is the number of trees to grow, and then average together.*

Number of terms sampled per split *is the number of columns to consider as splitting candidates at each split. For each split, a new random sample of columns is taken as the candidate set.*

Bootstrap sample rate *is the proportion of observations to sample (with replacement) for growing each tree. A new random sample is generated for each tree.*

Minimum Splits Per Tree *is the minimum number of splits for each tree.*

Minimum Size Split *is the minimum number of observations needed on a candidate split.*

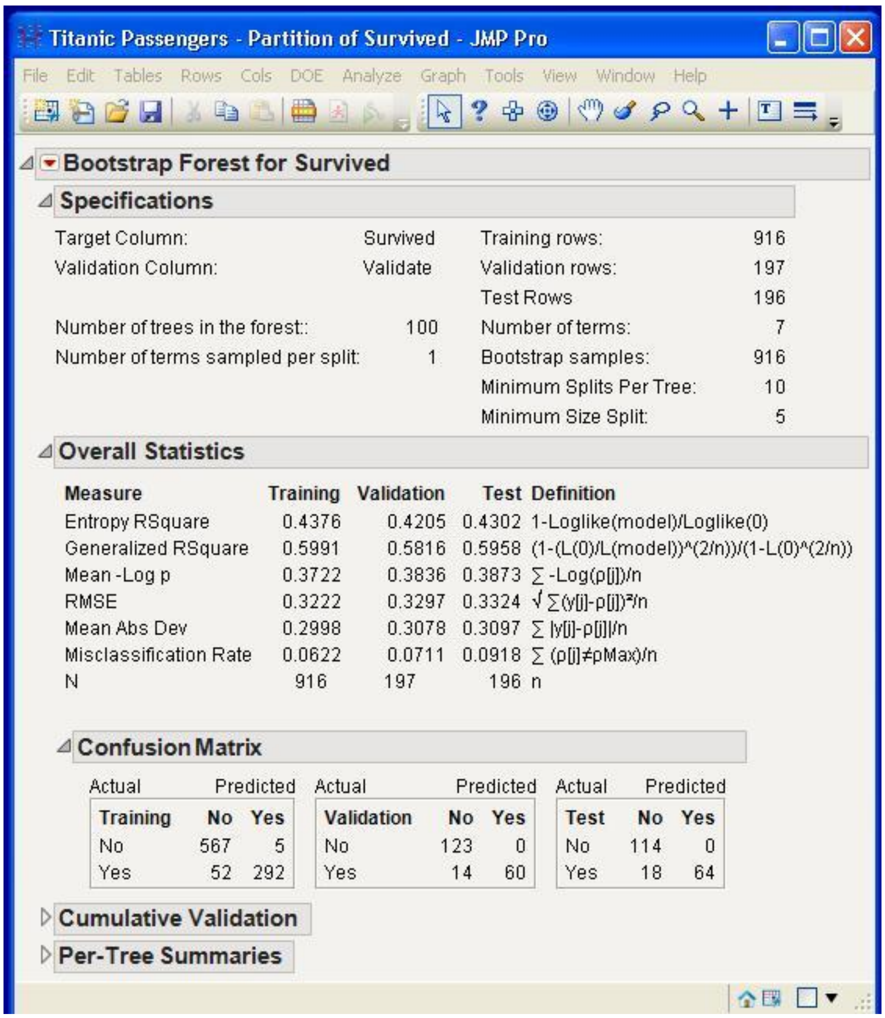
Early Stopping *is checked to perform early stopping. If checked, the process stops growing additional trees if adding more trees doesn't improve the validation statistic. If not checked, the process continues until the specified number of trees is reached. This option appears only if validation is used.*

Multiple Fits over number of terms *is checked to create a bootstrap forest for several values of Number of terms sampled per split. The lower value is specified above by the Number of terms samples per split option. The upper value is specified by the following option:*

Max Number of terms *is the maximum number of terms to consider for a split.*

The Titanic data, as seen above, defaults to 100 trees in the forest, 1 term sampled per split, with a sample size equal to the entire data set (note that it is not likely that the dataset sampled will match the underlying dataset because the sampling is done with replacement), a minimum of 10 splits per tree, and a minimum of 5 observations needed for the candidate split. Early Stopping and Multiple Fits are not selected. To run with the default settings, click "OK". Figure 3 shows results as presented by the bootstrap forest.

Figure 3: Screenshot of Bootstrap Forest Results

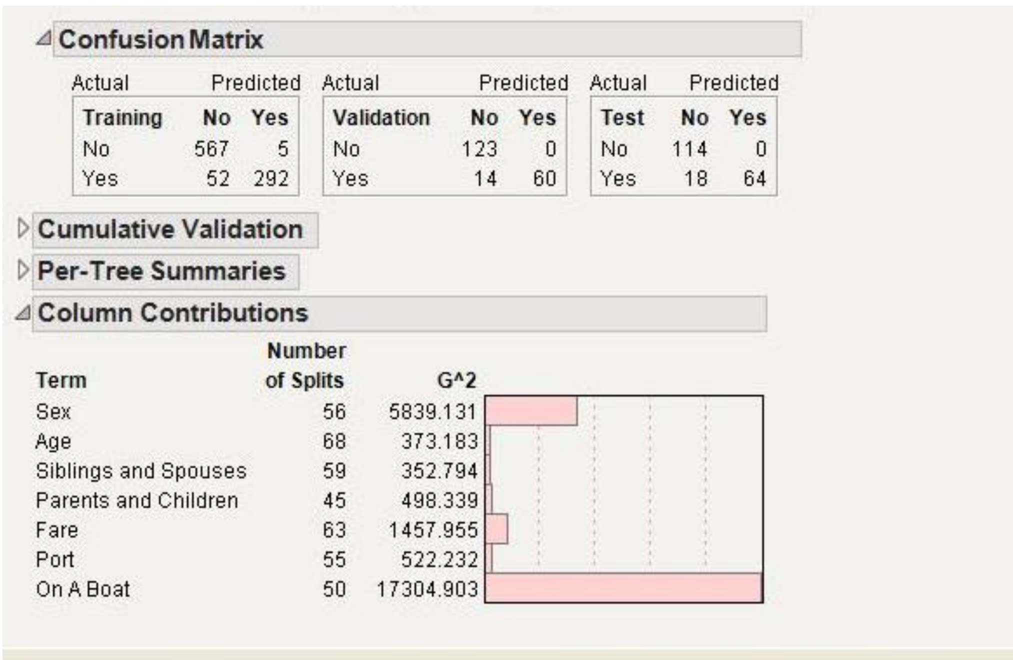


The results give a variety of statistics describing the model fit, such as Generalized R-Square, RMSE, and the Misclassification Rate.

What is immediately apparent from these results is that, based on the confusion matrix, the model does an extremely well at predicting survivors (as both the validation and test sets had every predicted survivor live). Furthermore, it does relatively well at predicting who died in the shipwreck.

If we go to the red arrow and click Column Contributions, a Column Contribution Summary will be presented. From Figure 4, it is clear that “On A Boat” had a large contribution. This, of course makes sense, and it can be hypothesized that those securing a seat on a lifeboat would have a higher propensity to survive than those left in the frigid waters.

Figure 4: Screenshot of Column Contributions



Confusion Matrix

Training (Actual)	Predicted No	Predicted Yes
No	567	5
Yes	52	292

Validation (Actual)	Predicted No	Predicted Yes
No	123	0
Yes	14	60

Test (Actual)	Predicted No	Predicted Yes
No	114	0
Yes	18	64

Cumulative Validation

Per-Tree Summaries

Column Contributions

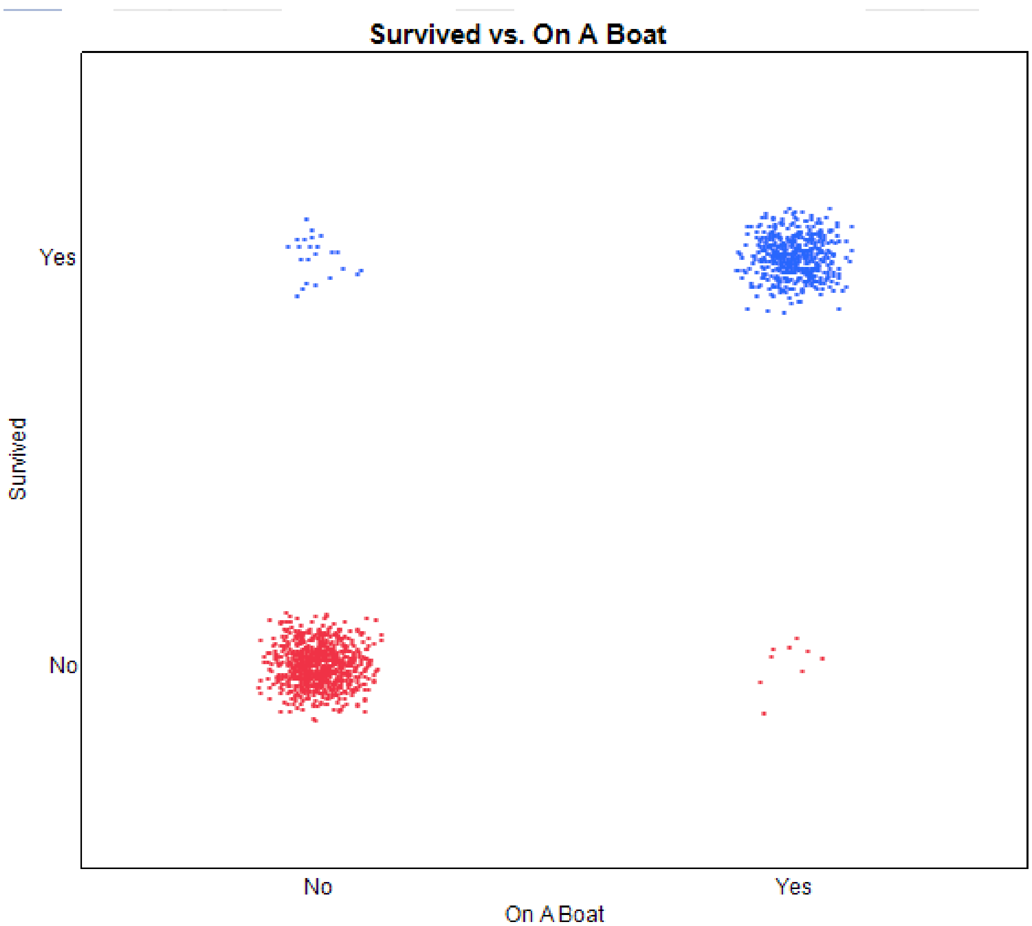
Term	Number of Splits	G^2
Sex	56	5839.131
Age	68	373.183
Siblings and Spouses	59	352.794
Parents and Children	45	498.339
Fare	63	1457.955
Port	55	522.232
On A Boat	50	17304.903

Based on the confusion matrix, we have built a useful predictive model using the bootstrap forest methodology. Prediction formulas and values can be saved by clicking the red triangle and choosing “Save Columns” and then “Save Predicteds” or “Save Prediction Formula”. This is especially useful if there are rows for which the condition is unknown and needs to be predicted based on information available. While in this case, we know with certainty the fates of each passenger, the analogous use would be to consider, for example, another boat, similar to the Titanic, launching. An individual may want to understand predict their propensity to survive if it should sink, based on their demographics, fare type, and life boat access. Further, before purchasing the ticket, they may want to consider what factors contribute to survival. Clearly, this is a contrived example (using non-confidential data), but it should be apparent that this methodology would be particular useful in business.

Titanic Example: Part 2

Since the lifeboat was such a large contributor to survival, yet many people did not have lifeboats, it is reasonable to consider if a reasonable model can be made to predict survival of those who waited for rescue in the water. However, a quick look at graph builder proves this will be a daunting task due to the small number of non-lifeboat survivors.

Figure 5: Survivors and Lifeboats



To look into this, rows for people not on a life boat can be shown and included using the data filter. The process for creating the bootstrap forest model is repeated, but without “On A Boat” as a predictive variable.

The results, however, predicted no survivors and hence, did not predict survival. A scaled up model with more trees and terms was created with the selections seen in Figure 6. Figure 7 shows that it also failed to classify the survivals.

This illustrates that this method, like most methods, has a difficult time predicting rare events.

Figure 6: Scaled Up Model Selection

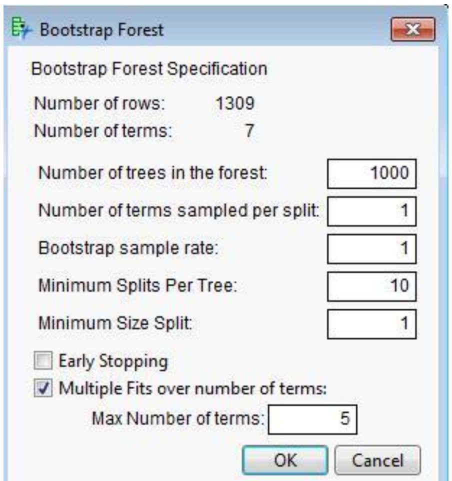
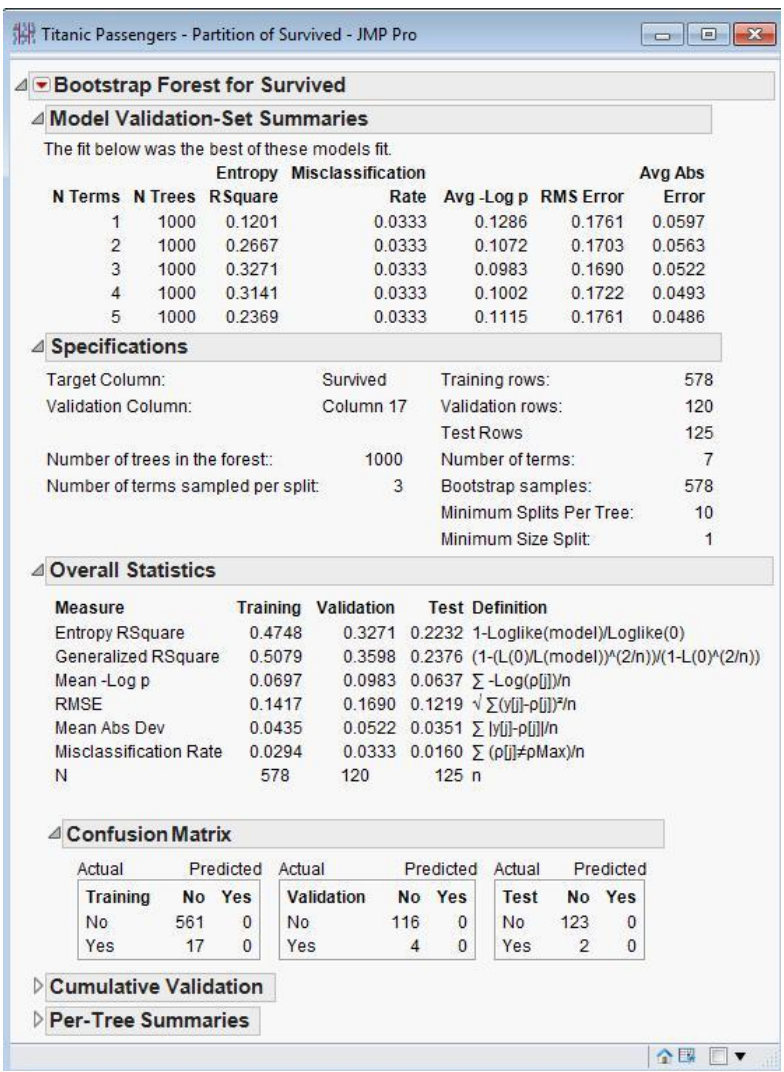


Figure 7: Scaled Up Model Results


Titanic Passengers - Partition of Survived - JMP Pro

Bootstrap Forest for Survived

Model Validation-Set Summaries

The fit below was the best of these models fit.

N Terms	N Trees	Entropy RSquare	Misclassification Rate	Avg -Log p	RMS Error	Avg Abs Error
1	1000	0.1201	0.0333	0.1286	0.1761	0.0597
2	1000	0.2667	0.0333	0.1072	0.1703	0.0563
3	1000	0.3271	0.0333	0.0983	0.1690	0.0522
4	1000	0.3141	0.0333	0.1002	0.1722	0.0493
5	1000	0.2369	0.0333	0.1115	0.1761	0.0486

Specifications

Target Column:	Survived	Training rows:	578
Validation Column:	Column 17	Validation rows:	120
		Test Rows	125
Number of trees in the forest::	1000	Number of terms:	7
Number of terms sampled per split:	3	Bootstrap samples:	578
		Minimum Splits Per Tree:	10
		Minimum Size Split:	1

Overall Statistics

Measure	Training	Validation	Test	Definition
Entropy RSquare	0.4748	0.3271	0.2232	1-Loglike(model)/Loglike(0)
Generalized RSquare	0.5079	0.3598	0.2376	(1-(L(0)/L(model))^(2/n))/(1-L(0)^(2/n))
Mean -Log p	0.0697	0.0983	0.0637	∑ -Log(p[j])/n
RMSE	0.1417	0.1690	0.1219	√ ∑(y[j]-p[j])²/n
Mean Abs Dev	0.0435	0.0522	0.0351	∑ |y[j]-p[j]|/n
Misclassification Rate	0.0294	0.0333	0.0160	∑ (p[j]≠pMax)/n
N	578	120	125	n

Confusion Matrix

Actual Training	Predicted No	Predicted Yes
No	561	0
Yes	17	0

Actual Validation	Predicted No	Predicted Yes
No	116	0
Yes	4	0

Actual Test	Predicted No	Predicted Yes
No	123	0
Yes	2	0

Cumulative Validation

Per-Tree Summaries

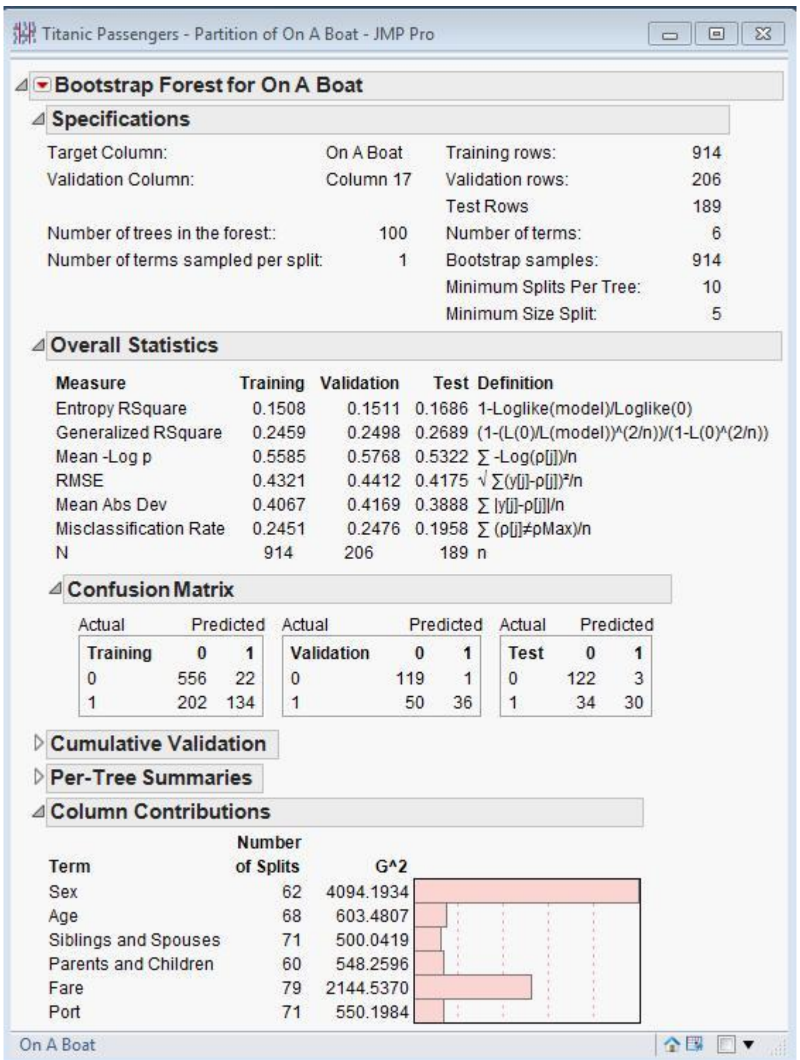
Titanic Example: Part 3

Finally, since lifeboats saved so many lives, it is worthwhile to ask to whom a lifeboat was provided. The bootstrap forest methodology can be used to consider this as well.

In this instance, the response becomes “On A Boat”. Using the default JMP settings, it is seen in Figure 8 that sex was a big driver of who got a life boat. Age interestingly didn’t have as high of a contribution as anticipated. However, Fare was also a large driver of who got a lifeboat.

Further, it can be seen that this model had a misclassification rate of 0.2451 and performed lower than hoped for in the validation and test conditions under the Confusion Matrix Section.

Figure 8: Lifeboat Prediction: Default Settings



Titanic Passengers - Partition of On A Boat - JMP Pro

Bootstrap Forest for On A Boat

Specifications

Target Column:	On A Boat	Training rows:	914
Validation Column:	Column 17	Validation rows:	206
		Test Rows	189
Number of trees in the forest:	100	Number of terms:	6
Number of terms sampled per split:	1	Bootstrap samples:	914
		Minimum Splits Per Tree:	10
		Minimum Size Split:	5

Overall Statistics

Measure	Training	Validation	Test	Definition
Entropy RSquare	0.1508	0.1511	0.1686	1-Loglike(model)/Loglike(0)
Generalized RSquare	0.2459	0.2498	0.2689	(1-(L(0)/L(model))^(2/n))/(1-L(0)^(2/n))
Mean -Log p	0.5585	0.5768	0.5322	∑ -Log(ρ[j])/n
RMSE	0.4321	0.4412	0.4175	√ ∑(y[j]-ρ[j])²/n
Mean Abs Dev	0.4067	0.4169	0.3888	∑ |y[j]-ρ[j]|/n
Misclassification Rate	0.2451	0.2476	0.1958	∑ (ρ[j]≠ρMax)/n
N	914	206	189	n

Confusion Matrix

Actual Training	Predicted 0	1
0	556	22
1	202	134

Actual Validation	Predicted 0	1
0	119	1
1	50	36

Actual Test	Predicted 0	1
0	122	3
1	34	30

Cumulative Validation

Per-Tree Summaries

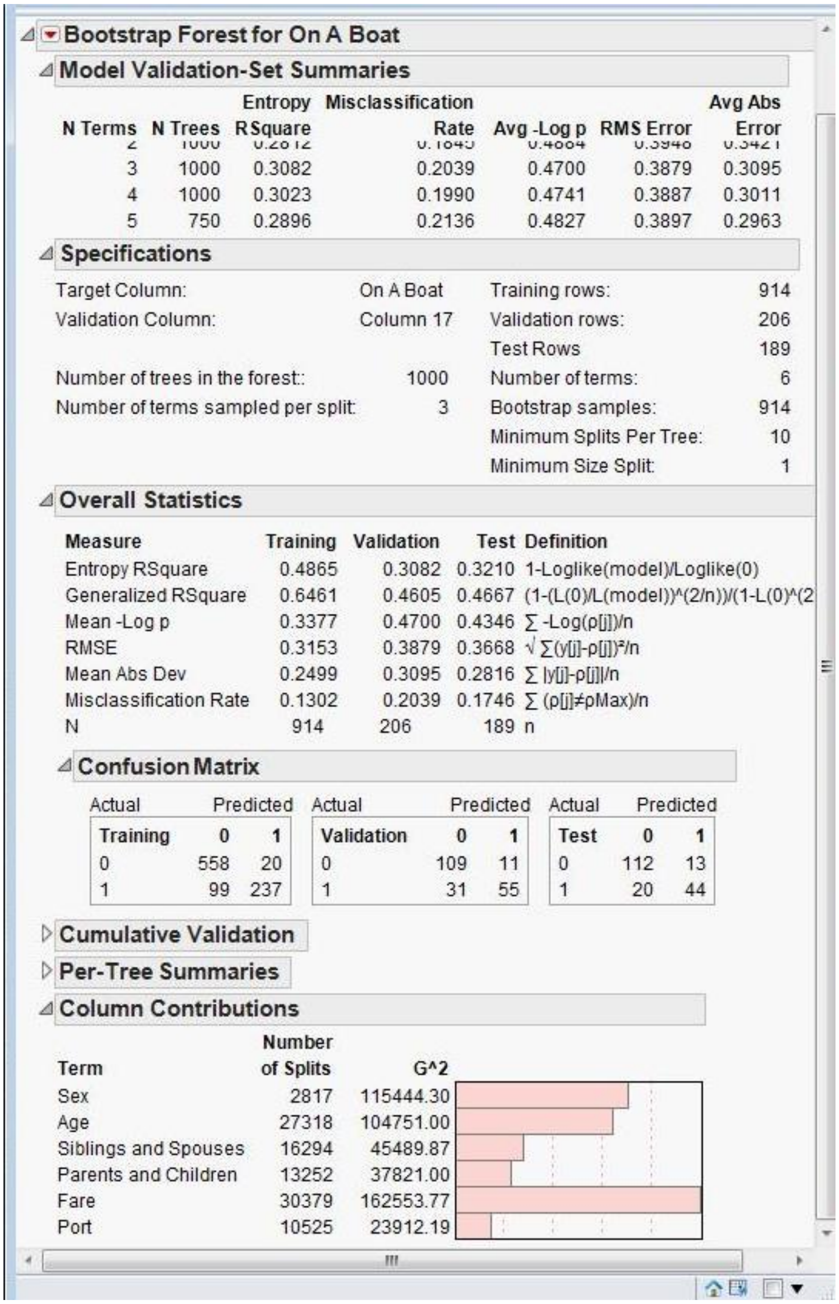
Column Contributions

Term	Number of Splits	G^2
Sex	62	4094.1934
Age	68	603.4807
Siblings and Spouses	71	500.0419
Parents and Children	60	548.2596
Fare	79	2144.5370
Port	71	550.1984

On A Boat

Wanting to improve predictive power, and considering that it is counterintuitive that age was not contributing more, the process was repeated using the same specifications shown in Figure 6. The result can be seen in Figure9.

Figure 9: Lifeboat Prediction: Scaled Up



Bootstrap Forest for On A Boat

Model Validation-Set Summaries

N Terms	N Trees	Entropy RSquare	Misclassification Rate	Avg -Log p	RMS Error	Avg Abs Error
2	1000	0.2812	0.1845	0.4884	0.3946	0.3421
3	1000	0.3082	0.2039	0.4700	0.3879	0.3095
4	1000	0.3023	0.1990	0.4741	0.3887	0.3011
5	750	0.2896	0.2136	0.4827	0.3897	0.2963

Specifications

Target Column:	On A Boat	Training rows:	914
Validation Column:	Column 17	Validation rows:	206
		Test Rows	189
Number of trees in the forest::	1000	Number of terms:	6
Number of terms sampled per split:	3	Bootstrap samples:	914
		Minimum Splits Per Tree:	10
		Minimum Size Split:	1

Overall Statistics

Measure	Training	Validation	Test	Definition
Entropy RSquare	0.4865	0.3082	0.3210	1-Loglike(model)/Loglike(0)
Generalized RSquare	0.6461	0.4605	0.4667	(1-(L(0)/L(model))^(2/n))/(1-L(0)^(2
Mean -Log p	0.3377	0.4700	0.4346	∑ -Log(ρ[j])/n
RMSE	0.3153	0.3879	0.3668	√ ∑(y[j]-ρ[j])²/n
Mean Abs Dev	0.2499	0.3095	0.2816	∑ |y[j]-ρ[j]|/n
Misclassification Rate	0.1302	0.2039	0.1746	∑ (ρ[j]≠ρMax)/n
N	914	206	189	n

Confusion Matrix

Actual Training	Predicted 0	Predicted 1
0	558	20
1	99	237

Actual Validation	Predicted 0	Predicted 1
0	109	11
1	31	55

Actual Test	Predicted 0	Predicted 1
0	112	13
1	20	44

Cumulative Validation

Per-Tree Summaries

Column Contributions

Term	Number of Splits	G^2
Sex	2817	115444.30
Age	27318	104751.00
Siblings and Spouses	16294	45489.87
Parents and Children	13252	37821.00
Fare	30379	162553.77
Port	10525	23912.19

From Figure 9, it is noted that the misclassification rate is now at 0.1302, which is a large improvement over the results using the default method. In the model chosen here, it is apparent that age is also a contributor, validating the expression, “Women and children first!” … but a thorough look at the results may suggest a revision to “Wealthy people, women, and children first!”

Conclusion

Through the use of bootstrap forest methodology, it has been shown that a dataset can be mined and predictive models built. This methodology is a very useful tool for performing these types of analyses, but can be computationally expensive, especially when performing analyses like the “Scaled Up” versions described in this paper, which were run on a very powerful machine. However, as computing power has become more available, methods like bootstrap forest are becoming more and more accessible to all analytics professionals.

References

[1] JMP Pro 10.0.0 Help

Contact Info

Please contact the author with any questions.

George J. Hurley
The Hershey Company
19 E Chocolate Ave.
Hershey, PA 17033
ghurley@hersheys.com

Trademark Citation