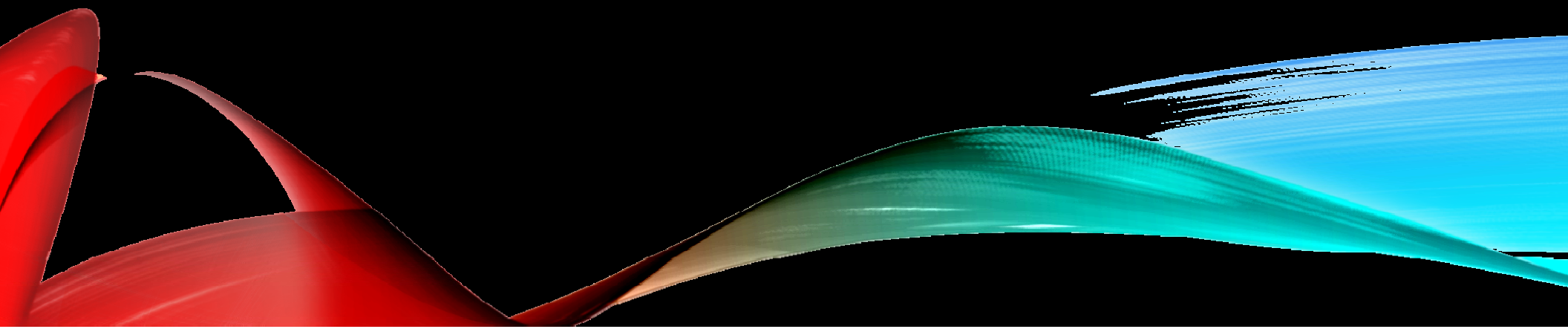
# ADVANCED DECISION TREES

JMP Partition Platform

# ADVANCED TREES



# Partition Modeling



| Y | X | Objective | Predictive Accuracy | Statistical Significance Measure | Model Fit |
|---|---|---|---|---|---|
| Categorical | Continuous or Categorical | Classification | Misclassification Rate \| Confusion Matrix | LogWorth | RMSE, MAD, ROC Curve |

| Y | X | Objective | Predictive Accuracy | Statistical Significance Measure | Model Fit |
|---|---|---|---|---|---|
| Continuous | Continuous or Categorical | Prediction | Rsquare, RMSE | LogWorth | RMSE, MAD, Lift Curve |

# Advantages and Disadvantages of Trees

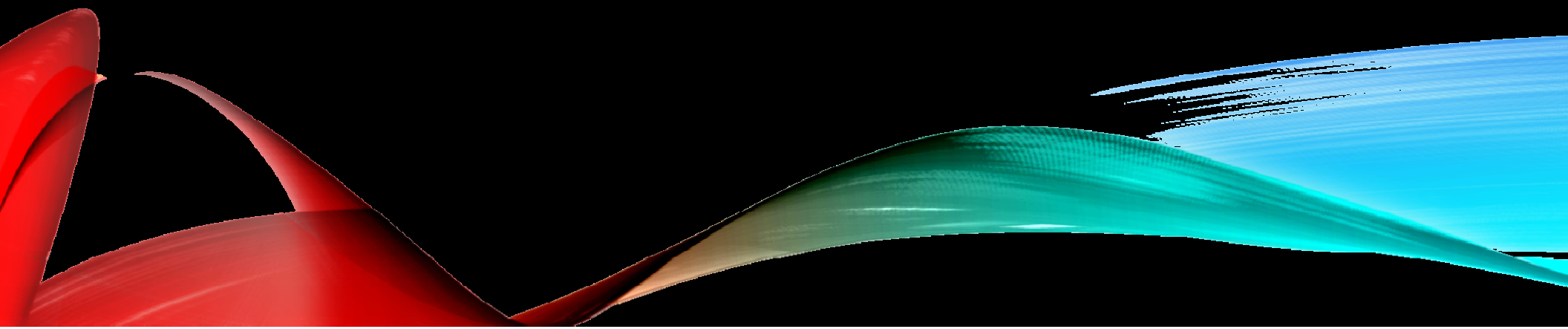
- ▲ Trees are very easy to explain to people. In fact, they are even easier to explain than linear regression!
- ▲ Some people believe that decision trees more closely mirror human decision-making than do the regression and classification approaches seen in previous chapters.
- ▲ Trees can be displayed graphically, and are easily interpreted even by a non-expert (especially if they are small).
- ▲ Trees can easily handle qualitative predictors without the need to create dummy variables.
- ▼ Unfortunately, trees generally do not have the same level of predictive accuracy as some of the other regression and classification approaches seen in this book.

However, by aggregating many decision trees, the predictive performance of trees can be substantially improved. We introduce these concepts next.



Stanford Online Course: Course Notes, Statistical Learning by Hastie and Tibshirani, Winter 2014 | Based on An Introduction to Statistical Learning by James, Witten, Hastie and Tibshirani, Springer (2013)

# BOOTSTRAP FOREST



# Bagging

- *Bootstrap aggregation*, or *bagging*, is a general-purpose procedure for reducing the variance of a statistical learning method; we introduce it here because it is particularly useful and frequently used in the context of decision trees.
- Recall that given a set of $n$ independent observations $Z_1, \ldots, Z_n$, each with variance $\sigma^2$, the variance of the mean $\bar{Z}$ of the observations is given by $\sigma^2/n$.
- In other words, *averaging a set of observations reduces variance*. Of course, this is not practical because we generally do not have access to multiple training sets.



Stanford Online Course: Course Notes, Statistical Learning by Hastie and Tibshirani, Winter 2014 | Based on An Introduction to Statistical Learning by James, Witten, Hastie and Tibshirani, Springer (2013)

# Bagging— continued

- Instead, we can bootstrap, by taking repeated samples from the (single) training data set.
- In this approach we generate $B$ different bootstrapped training data sets. We then train our method on the $b$th bootstrapped training set in order to get $\hat{f}^{*b}(x)$, the prediction at a point $x$. We then average all the predictions to obtain

$$\hat{f}_{\text{bag}}(x) = \frac{1}{B}\sum_{b=1}^{B}\hat{f}^{*b}(x).$$

This is called *bagging*.

Stanford Online Course: Course Notes, Statistical Learning by Hastie and Tibshirani, Winter 2014 | Based on An Introduction to Statistical Learning by James, Witten, Hastie and Tibshirani, Springer (2013)

# Bagging classification trees

- The above prescription applied to regression trees
- For classification trees: for each test observation, we record the class predicted by each of the $B$ trees, and take a *majority vote*: the overall prediction is the most commonly occurring class among the $B$ predictions.

Stanford Online Course: Course Notes, Statistical Learning by Hastie and Tibshirani, Winter 2014 | Based on An Introduction to Statistical Learning by James, Witten, Hastie and Tibshirani, Springer (2013)

# Random Forests

- *Random forests* provide an improvement over bagged trees by way of a small tweak that *decorrelates* the trees. This reduces the variance when we average the trees.
- As in bagging, we build a number of decision trees on bootstrapped training samples.
- But when building these decision trees, each time a split in a tree is considered, *a random selection of $m$ predictors* is chosen as split candidates from the full set of $p$ predictors. The split is allowed to use only one of those $m$ predictors.
- A fresh selection of $m$ predictors is taken at each split, and typically we choose $m \approx \sqrt{p}$ — that is, the number of predictors considered at each split is approximately equal to the square root of the total number of predictors (4 out of the 13 for the Heart data).

Stanford Online Course: Course Notes, Statistical Learning by Hastie and Tibshirani, Winter 2014 | Based on An Introduction to Statistical Learning by James, Witten, Hastie and Tibshirani, Springer (2013)


# The Two Most Important Algorithms in Predictive Modeling Today

*Jeremy Howard (Kaggle), Mike Bowles (Biomatica)*
1:30pm Tuesday, 02/28/2012
Data Science, Ballroom CD

When doing predictive modelling, there are two situations in which you might find yourself:

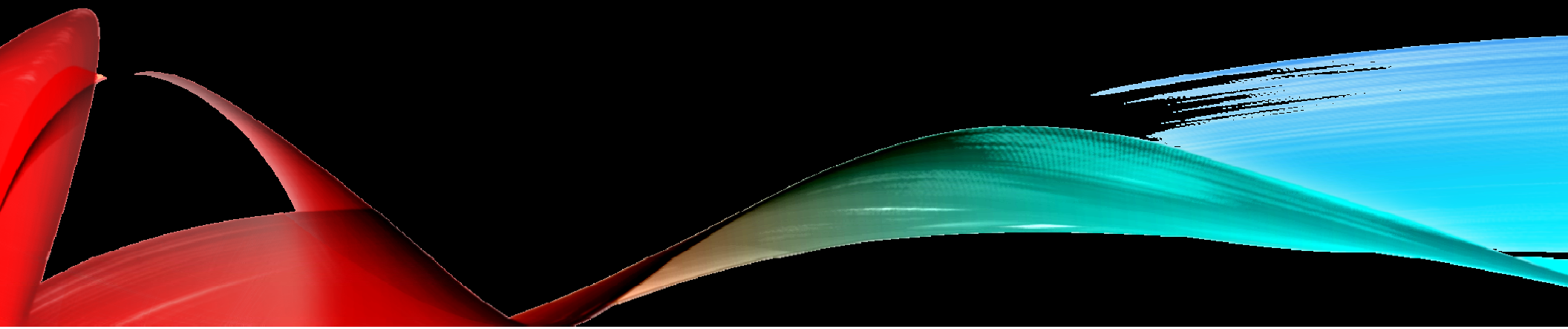
- You need to fit a well-defined parameterised model to your data, so you require a learning algorithm which can find those parameters on a large data set without over-fitting
- You just need a "black box" which can predict your dependent variable as accurately as possible, so you need a learning algorithm which can automatically identify the structure, interactions, and relationships in the data

For case (1), lasso and elastic-net regularized generalized linear models are a set of modern algorithms which meet all these needs. They are fast, work on huge data sets, and avoid over-fitting automatically. They are available in the "glmnet" package in R.

For case (2), ensembles of decision trees (often known as "Random Forests") have been the most successful general-purpose algorithm in modern times. For instance, most Kaggle competitions have at least one top entry that heavily uses this approach. This algorithm is very simple to understand, and is fast and easy to apply. It is available in the "randomForest" package in R.

Mike and Jeremy will explain in simple terms, using no complex math, how these algorithms work, and will also explain using numerous examples how to apply them using R. They will also provide advice on how to select from these algorithms, and will show how to prepare the data, and how to use the trained models in practice.

# TITANTIC DATA



"The sinking of the RMS Titanic is one of the most infamous shipwrecks in history. On April 15, 1912, during her maiden voyage, the Titanic sank after colliding with an iceberg, killing 1502 out of 2224 passengers and crew. This sensational tragedy shocked the international community and led to better safety regulations for ships.

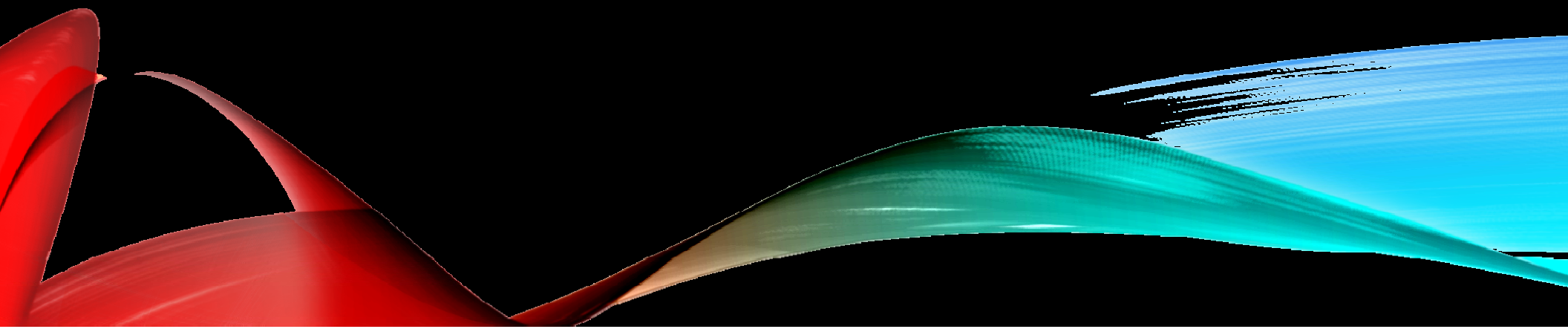
One of the reasons that the shipwreck led to such loss of life was that there were not enough lifeboats for the passengers and crew. Although there was some element of luck involved in surviving the sinking, some groups of people were more likely to survive than others."

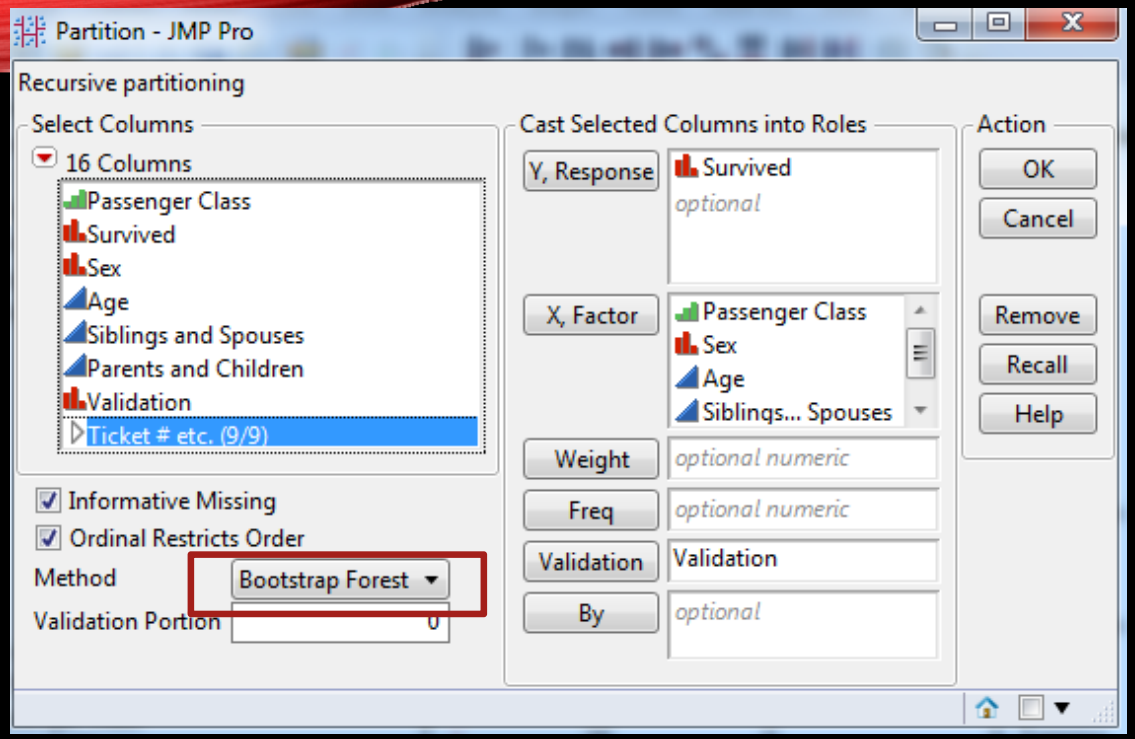
From a Kaggle competition: "Titanic: Machine Learning from Disaster", http://bit.ly/1f2crzi, data accessed 08/2014.

This data table describes the survival status of 1309 of the 1324 individual passengers on the Titanic. Information on the 899 crew members is not included.

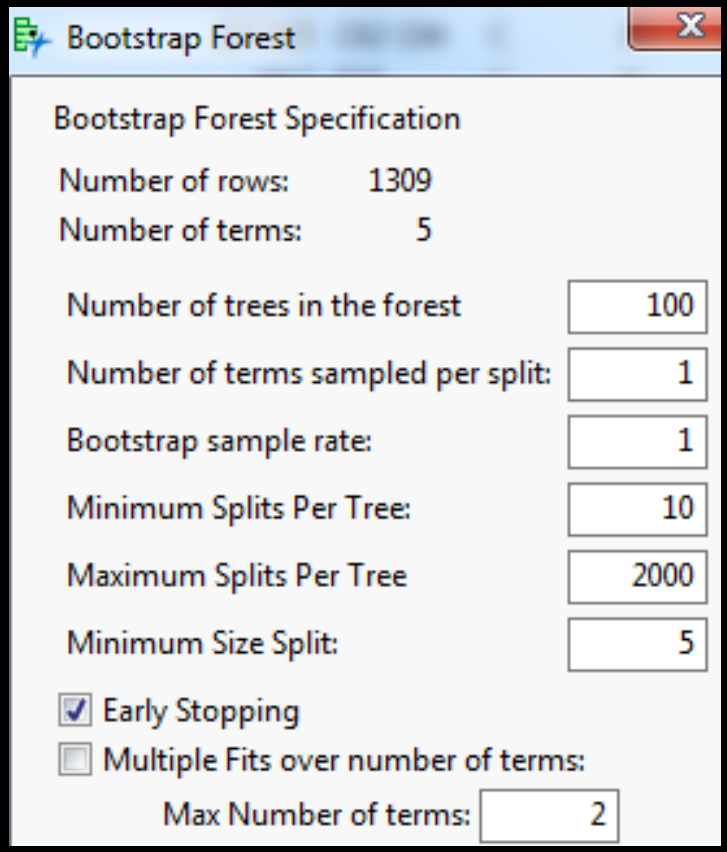
Name: Passenger Name
Survived: Yes or No
Passenger Class: 1, 2, or 3 corresponding to 1st, 2nd or 3rd class
Sex: Passenger sex
Age: Passenger age
Siblings and Spouses: The number of siblings and spouses aboard
Parents and Children: The number of parents and children aboard
Fare: The passenger fare
Port: Port of embarkment (C = Cherbourg; Q = Queenstown; S = Southampton)
Home/Destination: The home or final intended destination of the passenger

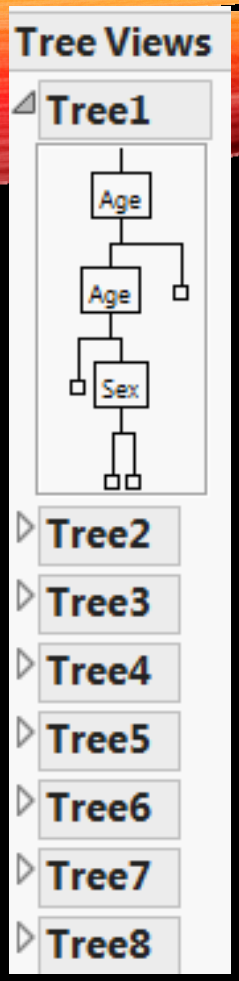
# BOOTSTRAP FOREST CATEGORICAL RESPONSE

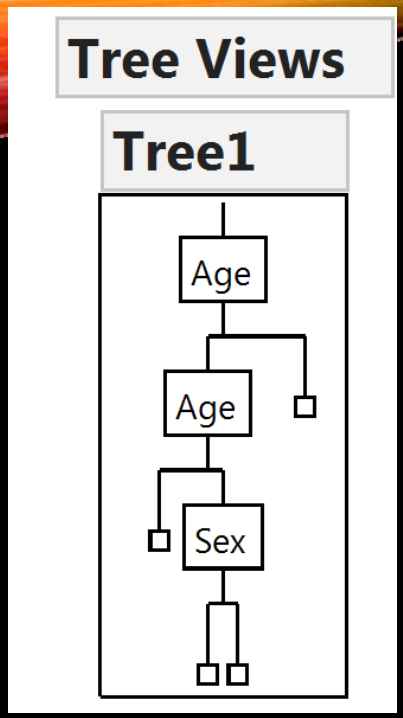


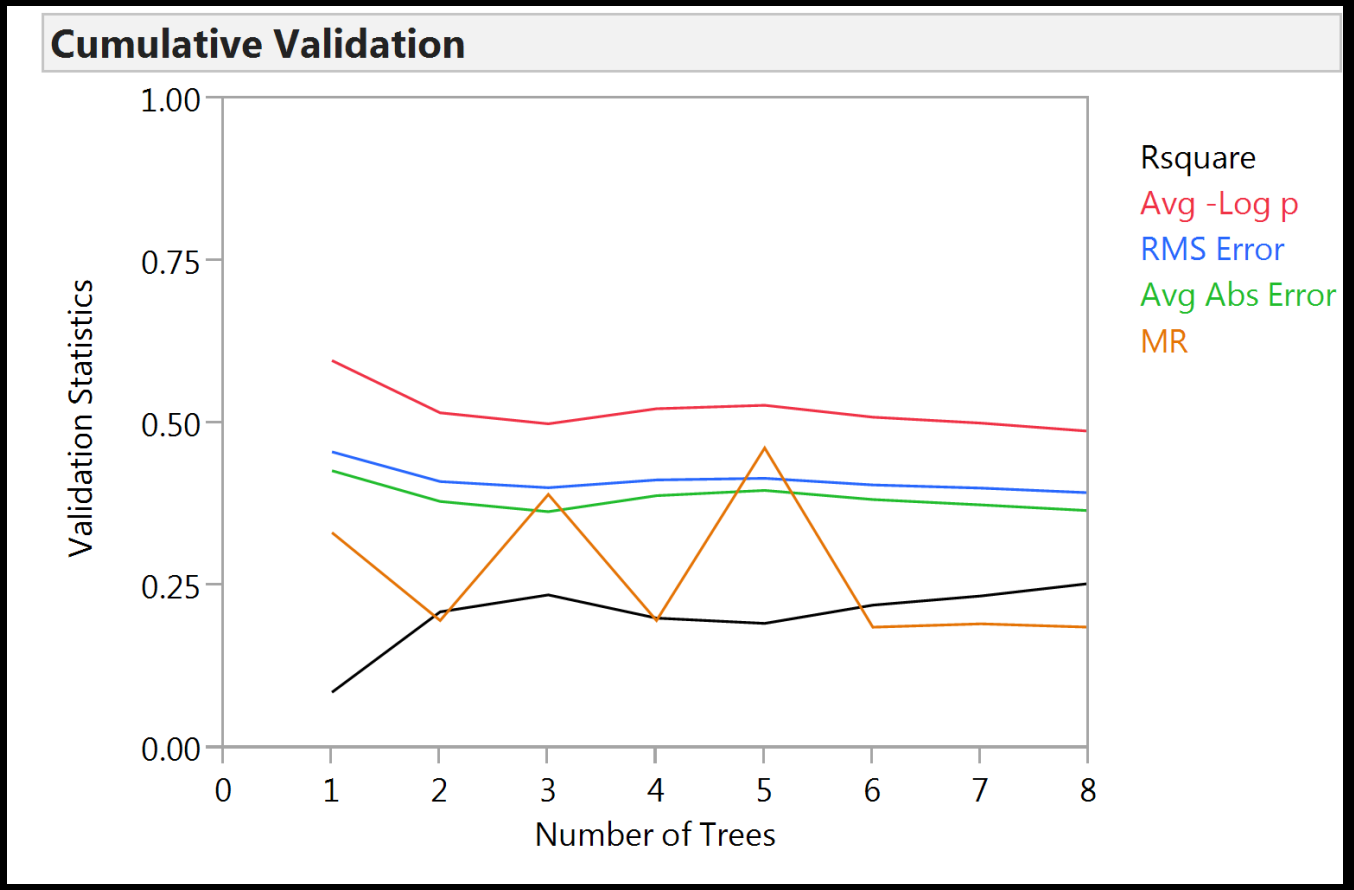



Use Defaults




Tree Views
Tree1
Age
Age
Sex
Tree2
Tree3
Tree4
Tree5
Tree6
Tree7
Tree8


Tree Views
Tree1
Age
Age
Sex


Cumulative Validation
1.00
0.75
0.50
0.25
0.00
Validation Statistics
0
1
2
3
4
5
6
7
8
Number of Trees
Rsquare
Avg -Log p
RMS Error
Avg Abs Error
MR

## Overall Statistics

| Measure | Training | Validation | Definition |
|---|---|---|---|
| Entropy RSquare | 0.2188 | 0.2527 | 1-Loglike(model)/Loglike(0) |
| Generalized RSquare | 0.3442 | 0.3854 | (1-(L(0)/L(model))^(2/n))/(1-L(0)^(2/n)) |
| Mean -Log p | 0.5231 | 0.4878 | $\sum$ -Log(ρ[j])/n |
| RMSE | 0.4116 | 0.3918 | $\sqrt{\sum(y[j]-\rho[j])^2/n}$ |
| Mean Abs Dev | 0.3803 | 0.3644 | $\sum \lvert y[j]-\rho[j]\rvert/n$ |
| Misclassification Rate | 0.2063 | 0.1858 | $\sum$ (ρ[j]≠ρMax)/n |
| N | 916 | 393 | n |

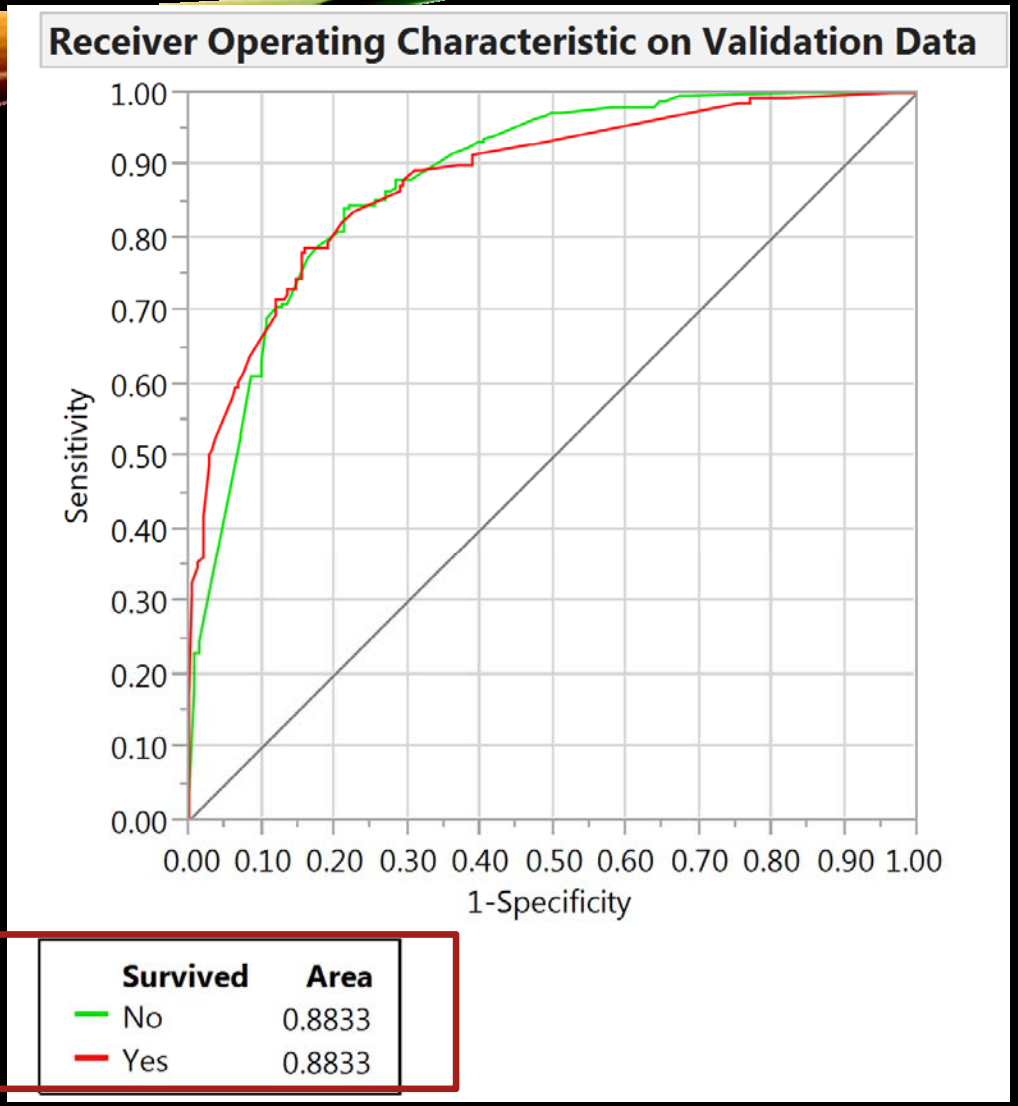
## Confusion Matrix

Actual Predicted

| Training | No | Yes |
|---|---|---|
| No | 482 | 75 |
| Yes | 114 | 245 |

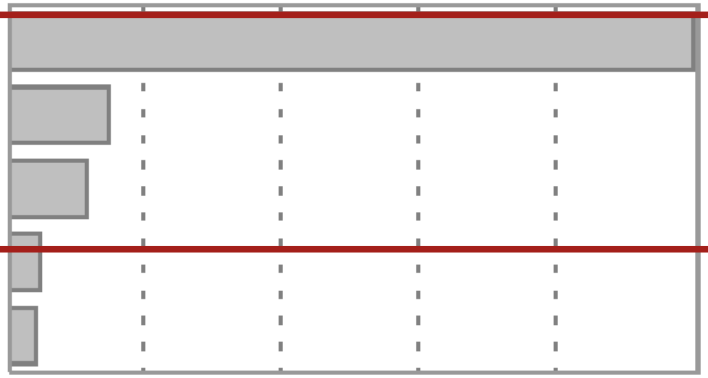
Actual Predicted

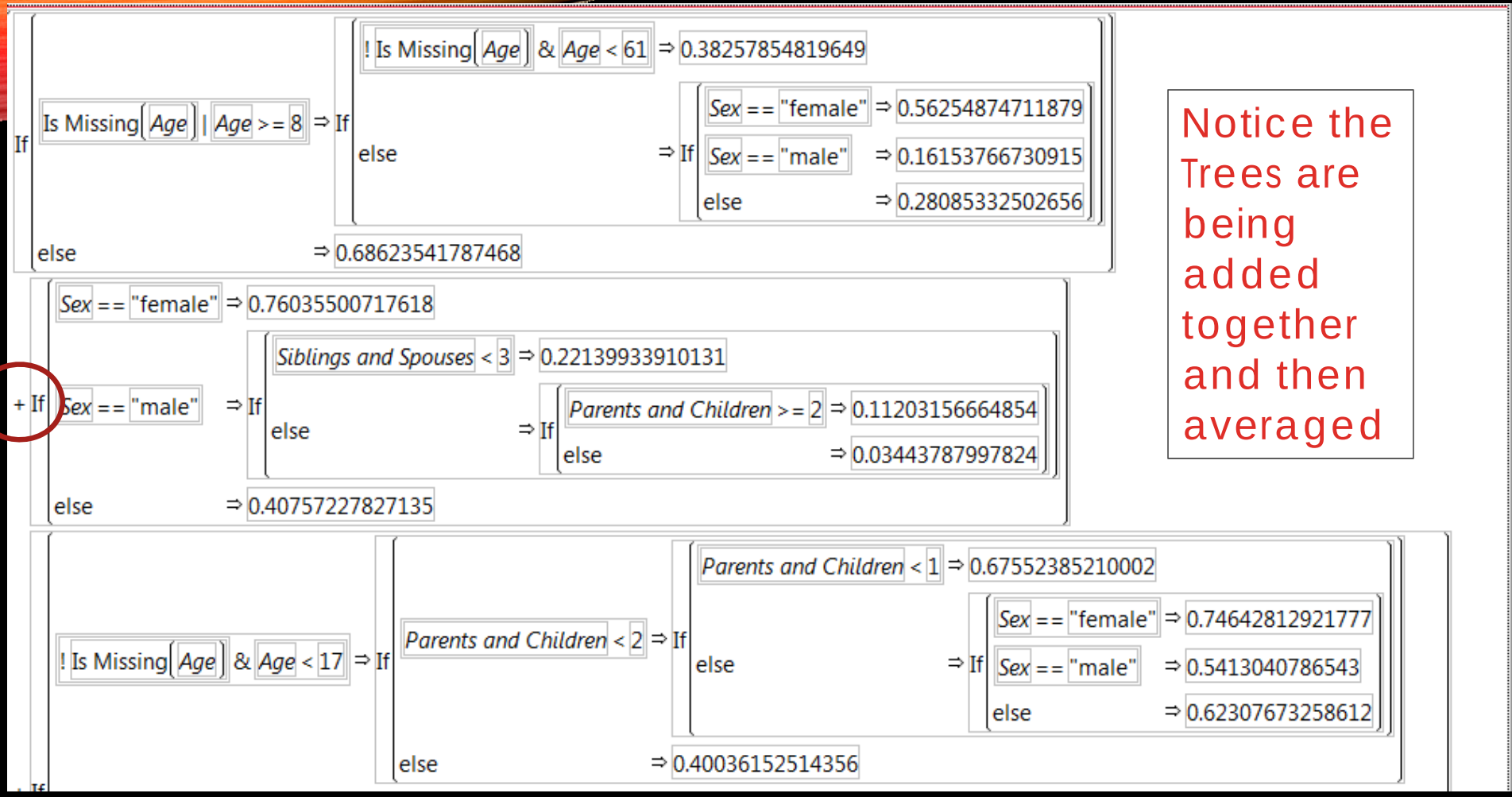
| Validation | No | Yes |
|---|---|---|
| No | 215 | 37 |
| Yes | 36 | 105 |


Receiver Operating Characteristic on Validation Data
Sensitivity
1.00
0.90
0.80
0.70
0.60
0.50
0.40
0.30
0.20
0.10
0.00
0.00 0.10 0.20 0.30 0.40 0.50 0.60 0.70 0.80 0.90 1.00
1-Specificity
Survived
Area
No
0.8833
Yes
0.8833

## Column Contributions

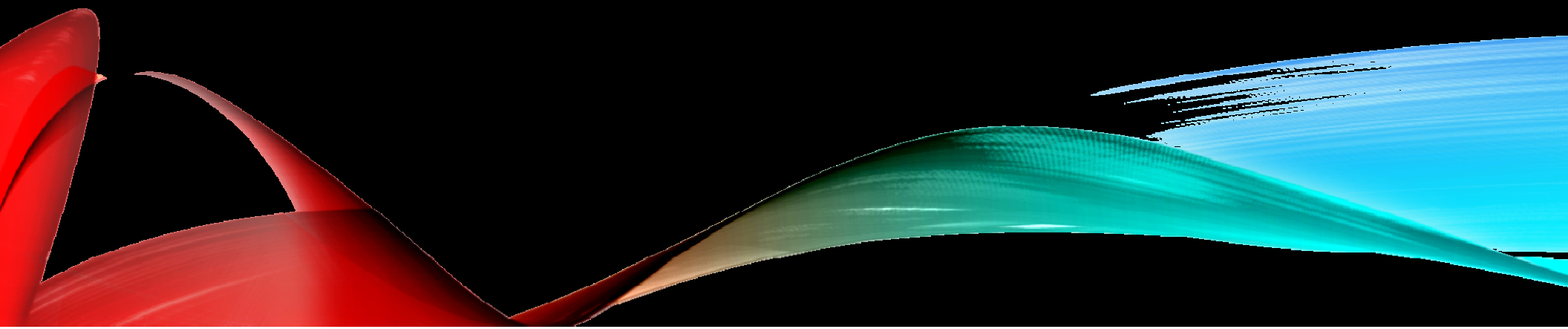
| Term | Number of Splits | G^2 | Portion |
|---|---|---|---|
| Sex | 7 | 747.889083 | 0.7371 |
| Passenger Class | 3 | 111.684997 | 0.1101 |
| Age | 8 | 86.7514698 | 0.0855 |
| Siblings and Spouses | 7 | 37.0069685 | 0.0365 |
| Parents and Children | 9 | 31.2527691 | 0.0308 |






Notice the Trees are being added together and then averaged

# BOOSTED TREE



# BOOSTED TREE

Boosting is the process of **building a large, additive decision tree** by fitting a sequence of smaller trees. **Each of the smaller trees is fit on the scaled residuals of the previous tree**. The trees are combined to form the larger final tree. The process can use validation to assess how many stages to fit, not to exceed the specified number of stages.

The tree at each stage is short, typically 1-5 splits. **After the initial tree, each stage fits the residuals from the previous stage**. The process continues until the specified number of stages is reached, or, if validation is used, until fitting an additional stage no longer improves the validation statistic. **The final prediction is the sum of the estimates for each terminal node over all the stages**.

If the response is categorical, the residuals fit at each stage are offsets of linear logits. The final prediction is a logistic transformation of the sum of the linear logits over all the stages. For categorical responses, only those with two levels are supported.

[JMP, Chapter 3, Specialized Models, Partition Models]

# Boosting

- Like bagging, boosting is a general approach that can be applied to many statistical learning methods for regression or classification. We only discuss boosting for decision trees.
- Recall that bagging involves creating multiple copies of the original training data set using the bootstrap, fitting a separate decision tree to each copy, and then combining all of the trees in order to create a single predictive model.
- Notably, each tree is built on a bootstrap data set, independent of the other trees.
- Boosting works in a similar way, except that the trees are grown *sequentially*: each tree is grown using information from previously grown trees.



Stanford Online Course: Course Notes, Statistical Learning by Hastie and Tibshirani, Winter 2014 | Based on An Introduction to Statistical Learning by James, Witten, Hastie and Tibshirani, Springer (2013)

# What is the idea behind this procedure?

- Unlike fitting a single large decision tree to the data, which amounts to *fitting the data hard* and potentially overfitting, the boosting approach instead *learns slowly*.
- Given the current model, we fit a decision tree to the residuals from the model. We then add this new decision tree into the fitted function in order to update the residuals.
- Each of these trees can be rather small, with just a few terminal nodes, determined by the parameter $d$ in the algorithm.
- By fitting small trees to the residuals, we slowly improve $\hat{f}$ in areas where it does not perform well. The shrinkage parameter $\lambda$ slows the process down even further, allowing more and different shaped trees to attack the residuals.

Stanford Online Course: Course Notes, Statistical Learning by Hastie and Tibshirani, Winter 2014 | Based on An Introduction to Statistical Learning by James, Witten, Hastie and Tibshirani, Springer (2013)

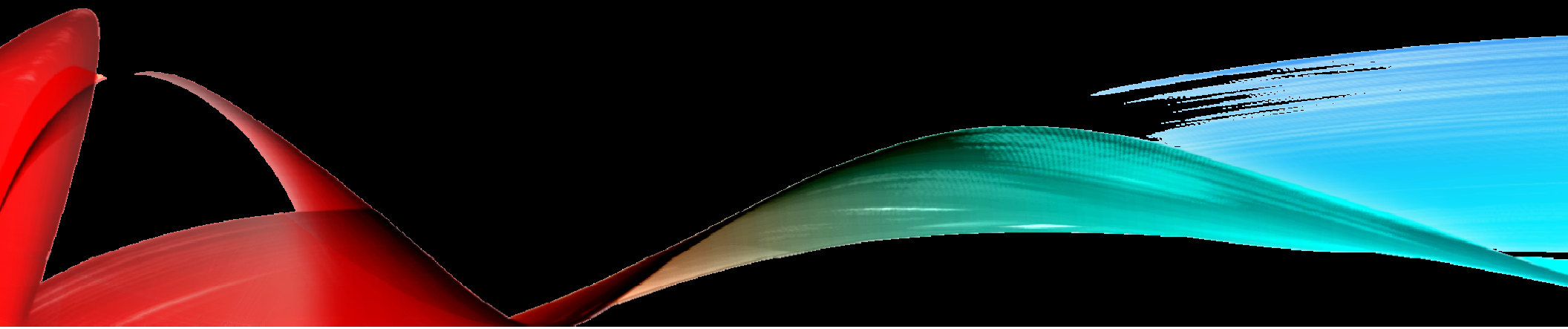
# Tuning parameters for boosting

1. The *number of trees* $B$. Unlike bagging and random forests, boosting can overfit if $B$ is too large, although this overfitting tends to occur slowly if at all. We use cross-validation to select $B$.
2. The *shrinkage parameter* $\lambda$, a small positive number. This controls the rate at which boosting learns. Typical values are 0.01 or 0.001, and the right choice can depend on the problem. Very small $\lambda$ can require using a very large value of $B$ in order to achieve good performance.
3. The *number of splits* $d$ in each tree, which controls the complexity of the boosted ensemble. Often $d = 1$ works well, in which case each tree is a *stump*, consisting of a single split and resulting in an additive model. More generally $d$ is the *interaction depth*, and controls the interaction order of the boosted model, since $d$ splits can involve at most $d$ variables.

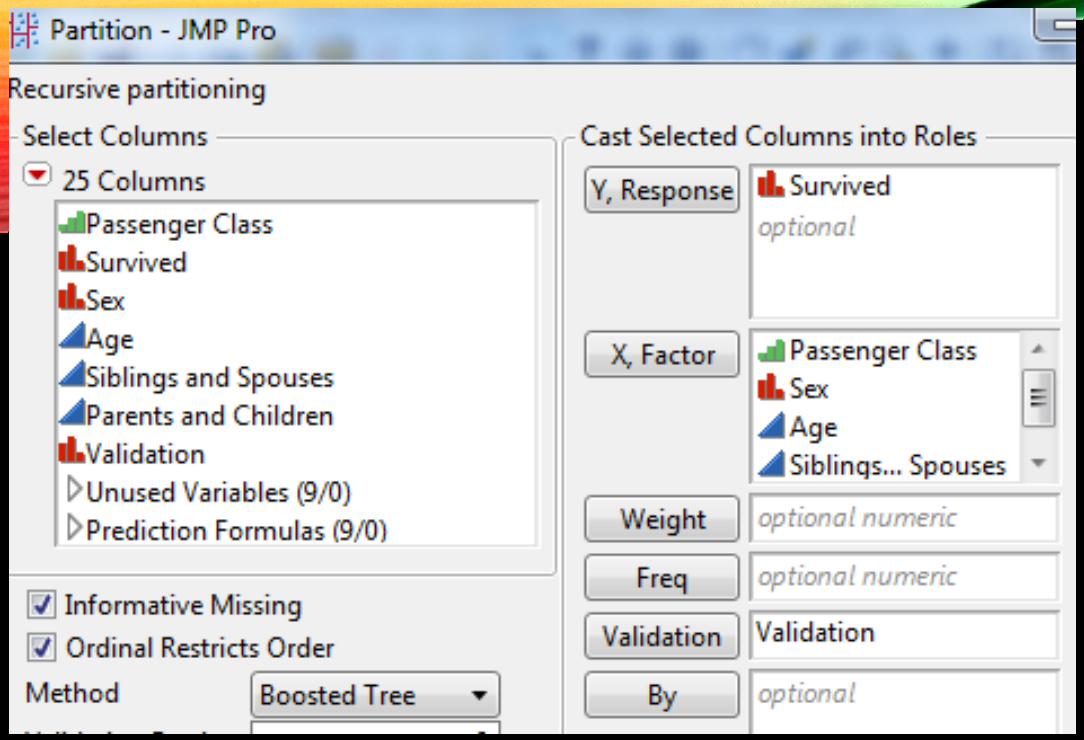


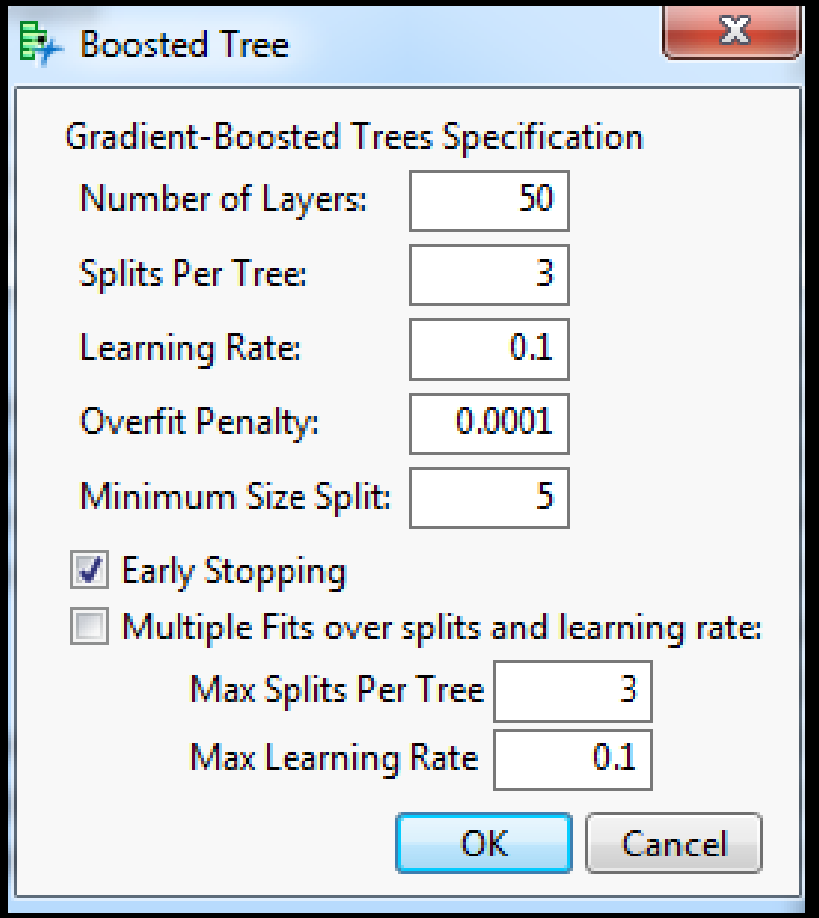
Stanford Online Course: Course Notes, Statistical Learning by Hastie and Tibshirani, Winter 2014 | Based on An Introduction to Statistical Learning by James, Witten, Hastie and Tibshirani, Springer (2013)

# BOOSTED TREE CATEGORICAL RESPONSE



Use Defaults



## Overall Statistics

| Measure | Training | Validation | Definition |
|---|---|---|---|
| Entropy RSquare | 0.3513 | 0.3788 | 1-Loglike(model)/Loglike(0) |
| Generalized RSquare | 0.5085 | 0.5352 | (1-(L(0)/L(model))^(2/n))/(1-L(0)^(2/n)) |
| Mean -Log p | 0.4344 | 0.4055 | $\sum$ -Log(ρ[j])/n |
| RMSE | 0.3716 | 0.3537 | $\sqrt{\sum(y[j]-\rho[j])^2/n}$ |
| Mean Abs Dev | 0.2881 | 0.2791 | $\sum \lvert y[j]-\rho[j]\rvert/n$ |
| Misclassification Rate | 0.1932 | 0.1756 | $\sum$ (ρ[j]≠ρMax)/n |
| N | 916 | 393 | n |

## Confusion Matrix

Actual Predicted

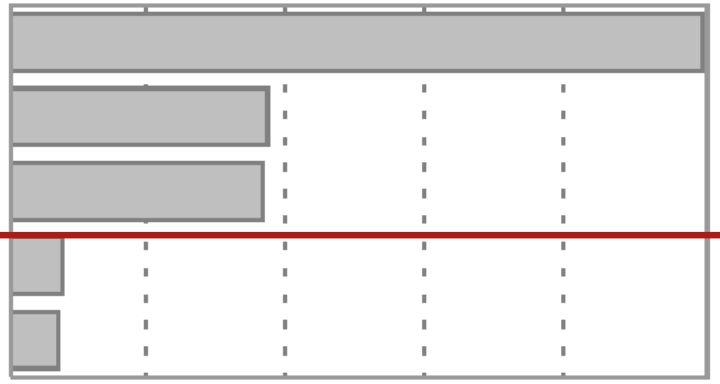
| Training | No | Yes |
|---|---|---|
| No | 509 | 48 |
| Yes | 129 | 230 |

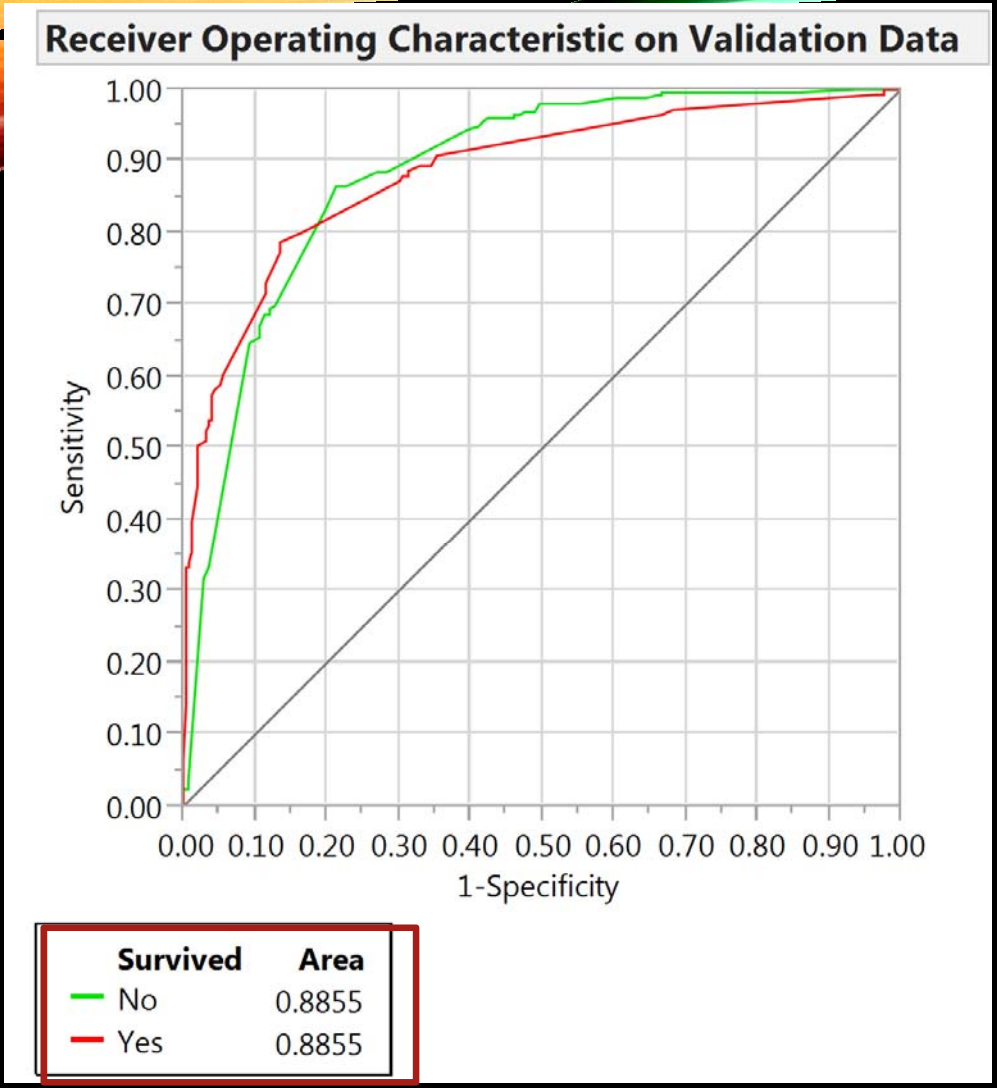
Actual Predicted

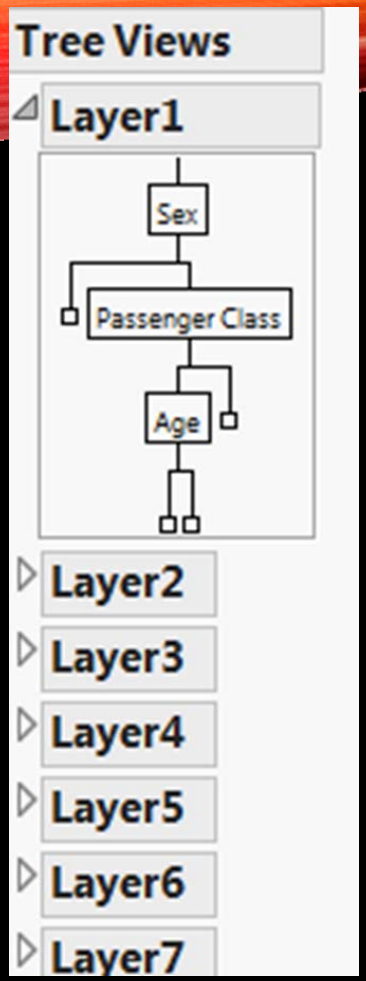
| Validation | No | Yes |
|---|---|---|
| No | 223 | 29 |
| Yes | 40 | 101 |

## Column Contributions

| Term | Number of Splits | G^2 | Portion |
|---|---|---|---|
| Sex | 57 | 54527.9454 | 0.5299 |
| Age | 23 | 20263.8193 | 0.1969 |
| Passenger Class | 39 | 19984.6722 | 0.1942 |
| Siblings and Spouses | 13 | 4208.00667 | 0.0409 |
| Parents and Children | 18 | 3922.45484 | 0.0381 |




Receiver Operating Characteristic on Validation Data
Sensitivity
1.00
0.90
0.80
0.70
0.60
0.50
0.40
0.30
0.20
0.10
0.00
0.00 0.10 0.20 0.30 0.40 0.50 0.60 0.70 0.80 0.90 1.00
1-Specificity
Survived
Area
No
0.8855
Yes
0.8855




Logist

-0.439242851439

+ If Sex == "male" ⇒ -0.0893205561555

+ If Sex == "female" ⇒ If Passenger Class == 1 | Passenger Class == 2 ⇒ If ! Is Missing Age & Age < 60 ⇒ 0.32458983863027

else ⇒ 0.21130867824872

Passenger Class == 3 ⇒ 0.03702503610437

else ⇒ 0

else ⇒ 0

+ If Sex == "male" ⇒ -0.0803885005401

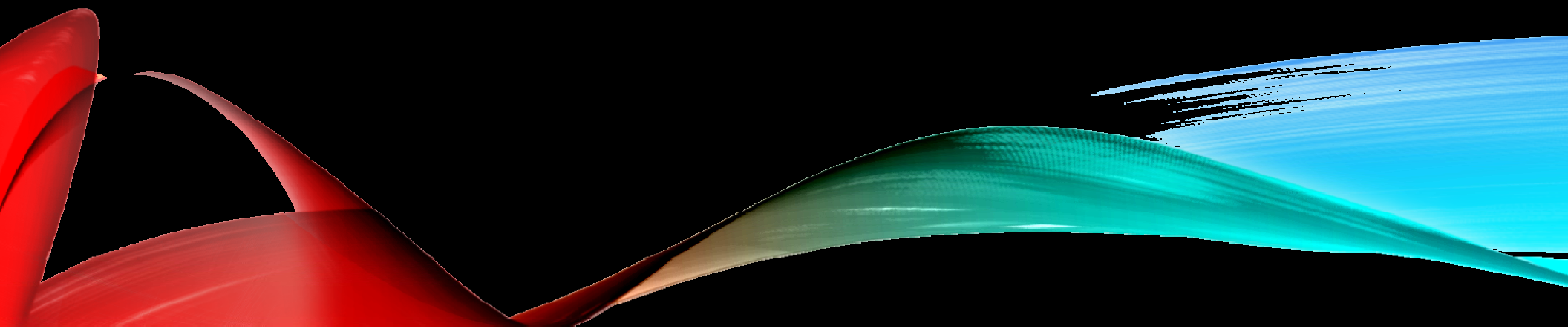
+ If Sex == "female" ⇒ If Passenger Class == 1 | Passenger Class == 2 ⇒ If ! Is Missing Age & Age < 60 ⇒ 0.29213084003711

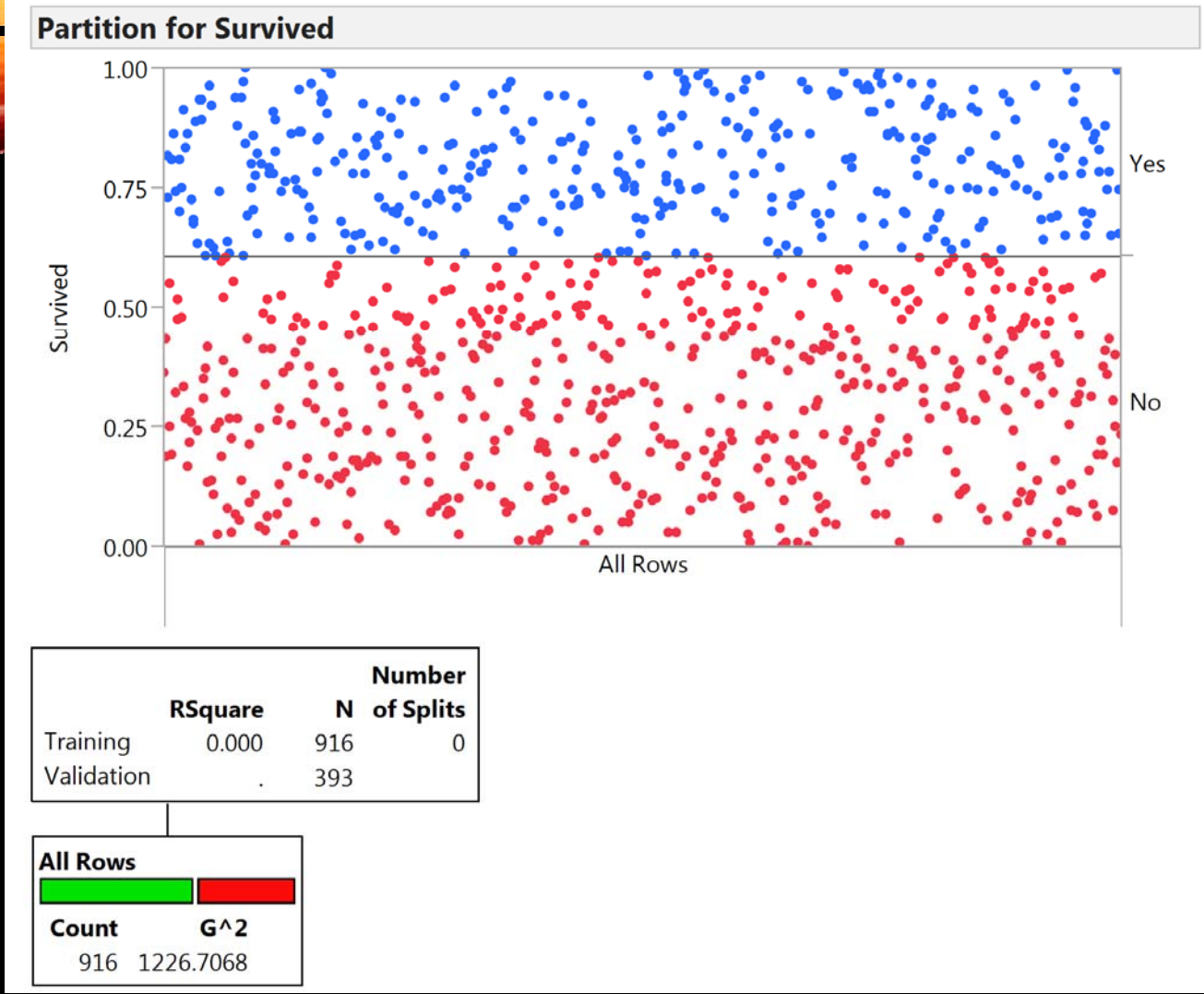
else ⇒ 0.19017781025151

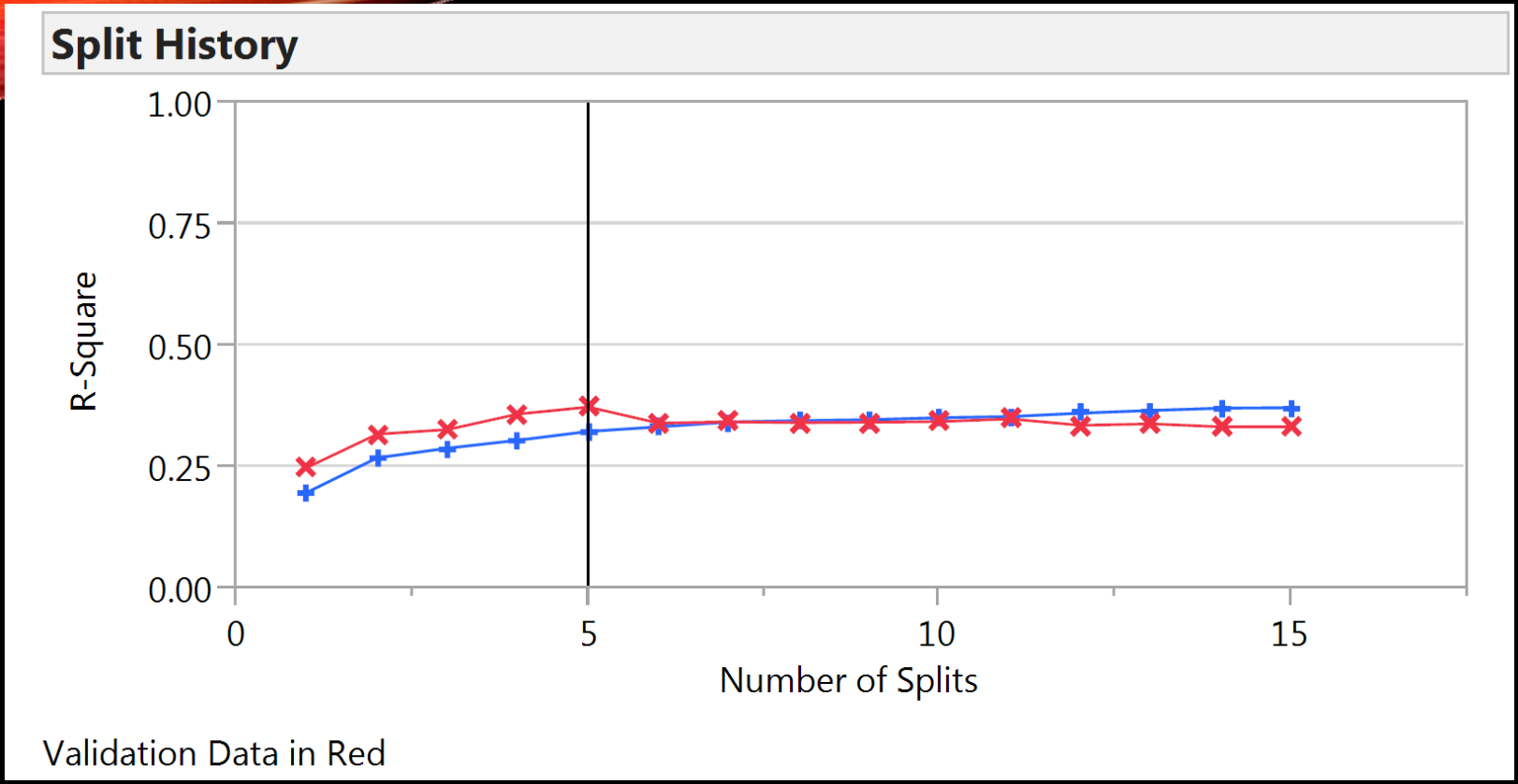
Passenger Class == 3 ⇒ 0.03332263313794

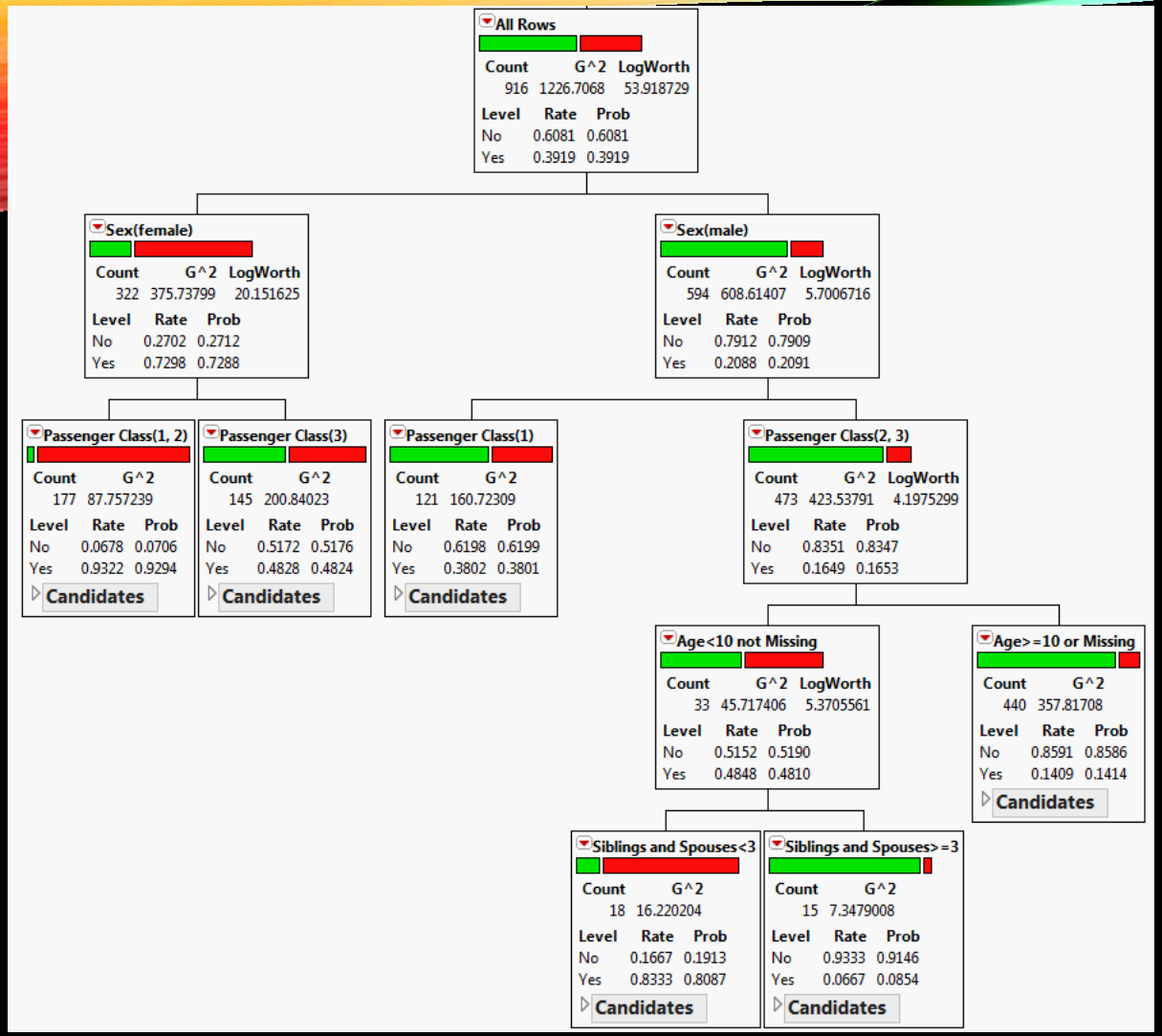
else ⇒ 0

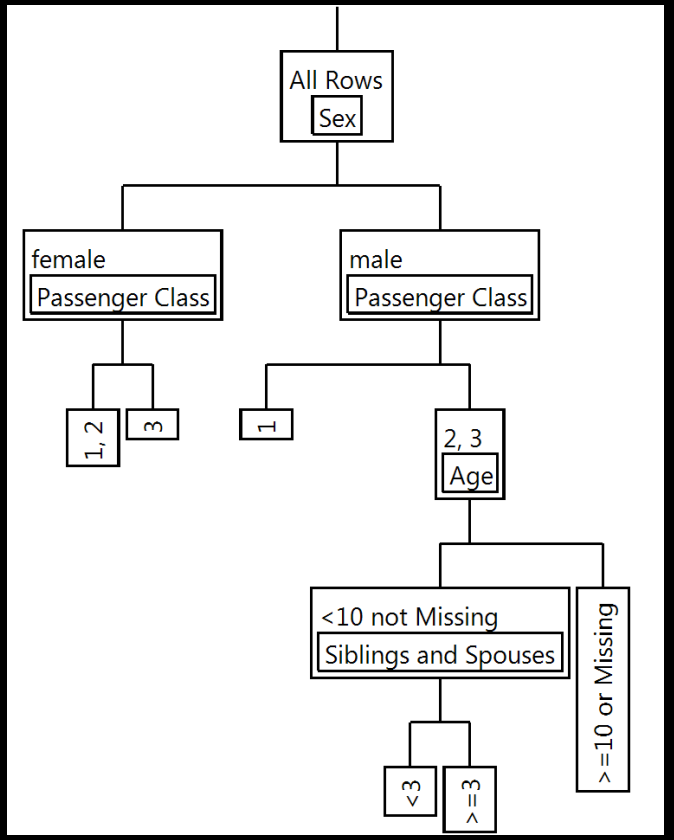
# DECISION TREE




Partition for Survived
1.00
0.75
0.50
0.25
0.00
Survived
Yes
No
All Rows
RSquare
N
Number of Splits
Training
0.000
916
0
Validation
.
393
All Rows
Count
G^2
916
1226.7068


Split History
1.00
0.75
0.50
0.25
0.00
R-Square
0
5
10
15
Number of Splits
Validation Data in Red


All Rows
Count G^2 LogWorth
916 1226.7068 53.918729
Level Rate Prob
No 0.6081 0.6081
Yes 0.3919 0.3919
Sex(female)
Count G^2 LogWorth
322 375.73799 20.151625
Level Rate Prob
No 0.2702 0.2712
Yes 0.7298 0.7288
Sex(male)
Count G^2 LogWorth
594 608.61407 5.7006716
Level Rate Prob
No 0.7912 0.7909
Yes 0.2088 0.2091
Passenger Class(1, 2)
Count G^2
177 87.757239
Level Rate Prob
No 0.0678 0.0706
Yes 0.9322 0.9294
Candidates
Passenger Class(3)
Count G^2
145 200.84023
Level Rate Prob
No 0.5172 0.5176
Yes 0.4828 0.4824
Candidates
Passenger Class(1)
Count G^2
121 160.72309
Level Rate Prob
No 0.6198 0.6199
Yes 0.3802 0.3801
Candidates
Passenger Class(2, 3)
Count G^2 LogWorth
473 423.53791 4.1975299
Level Rate Prob
No 0.8351 0.8347
Yes 0.1649 0.1653
Age<10 not Missing
Count G^2 LogWorth
33 45.717406 5.3705561
Level Rate Prob
No 0.5152 0.5190
Yes 0.4848 0.4810
Age>=10 or Missing
Count G^2
440 357.81708
Level Rate Prob
No 0.8591 0.8586
Yes 0.1409 0.1414
Candidates
Siblings and Spouses<3
Count G^2
18 16.220204
Level Rate Prob
No 0.1667 0.1913
Yes 0.8333 0.8087
Candidates
Siblings and Spouses>=3
Count G^2
15 7.3479008
Level Rate Prob
No 0.9333 0.9146
Yes 0.0667 0.0854
Candidates


All Rows
Sex
female
Passenger Class
male
Passenger Class
1, 2
3
1
2, 3
Age
<10 not Missing
Siblings and Spouses
>=10 or Missing
< 3
>= 3

## Fit Details

| Measure | Training | Validation | Definition |
|---|---|---|---|
| Entropy RSquare | 0.3227 | 0.3729 | 1-Loglike(model)/Loglike(0) |
| Generalized RSquare | 0.4755 | 0.5287 | (1-(L(0)/L(model))^(2/n))/(1-L(0)^(2/n)) |
| Mean -Log p | 0.4535 | 0.4093 | $\sum$ -Log(ρ[j])/n |
| RMSE | 0.3805 | 0.3560 | $\sqrt{\sum(y[j]-\rho[j])^2/n}$ |
| Mean Abs Dev | 0.2908 | 0.2782 | $\sum$ \|y[j]-ρ[j]\|/n |
| Misclassification Rate | 0.2118 | 0.1832 | $\sum$ (ρ[j]≠ρMax)/n |
| N | 916 | 393 | n |

## Confusion Matrix

Actual Predicted

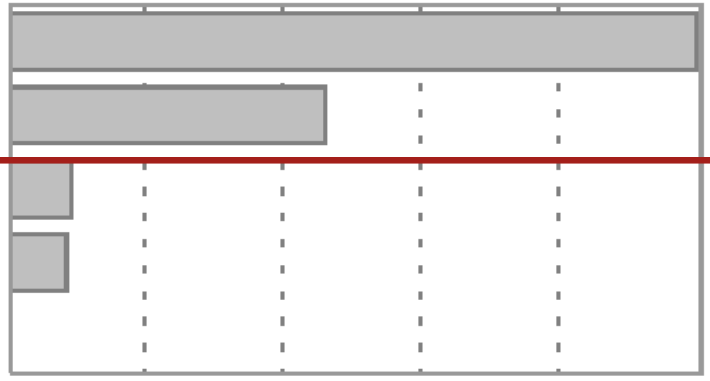
| Training | No | Yes |
|---|---|---|
| No | 542 | 15 |
| Yes | 179 | 180 |

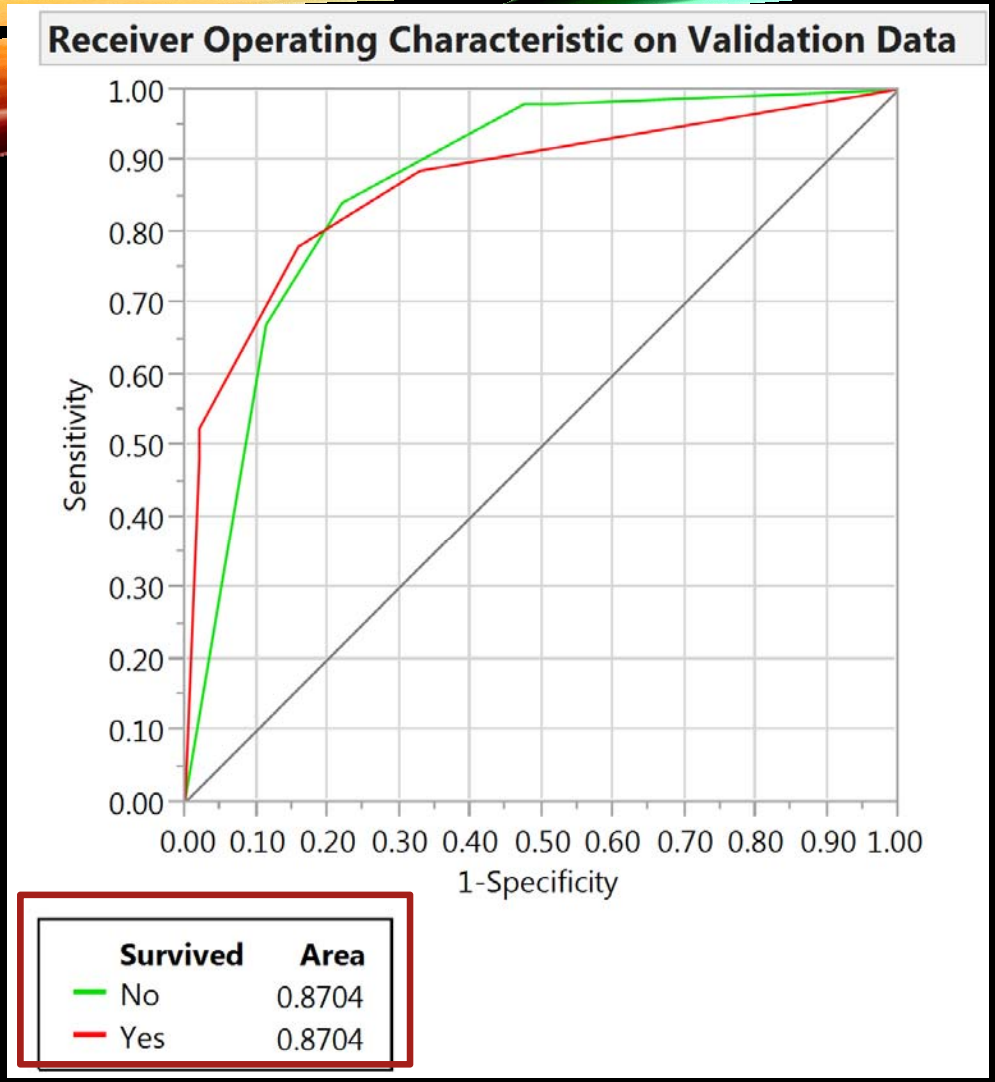
Actual Predicted

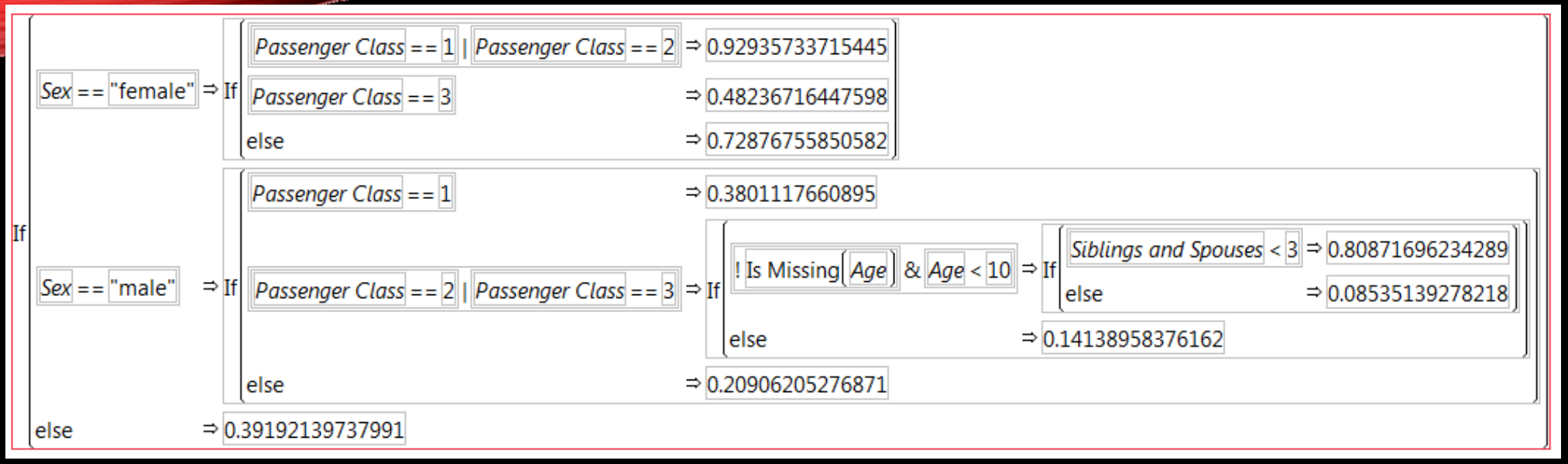
| Validation | No | Yes |
|---|---|---|
| No | 247 | 5 |
| Yes | 67 | 74 |

## Column Contributions

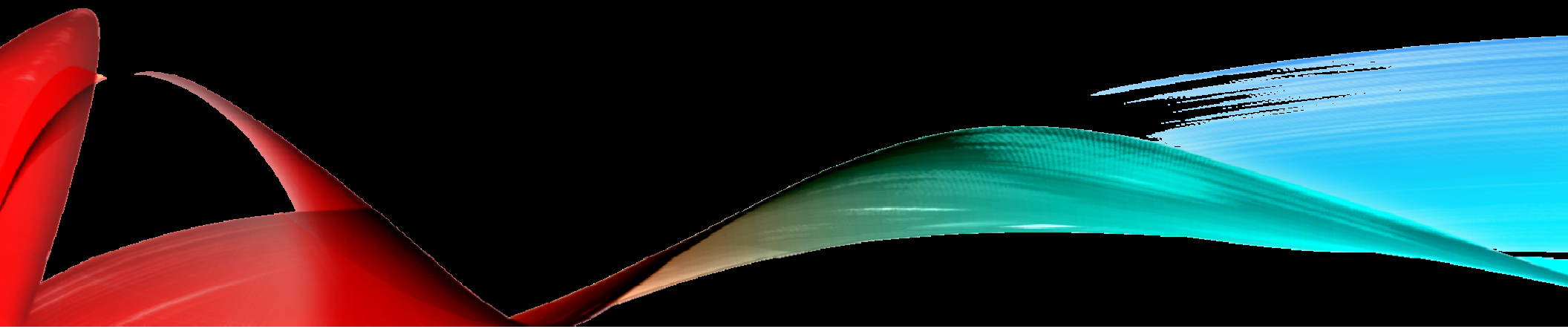
| Term | Number of Splits | G^2 | Portion |
|---|---|---|---|
| Sex | 1 | 242.354767 | 0.6120 |
| Passenger Class | 2 | 111.493595 | 0.2815 |
| Siblings and Spouses | 1 | 22.1493019 | 0.0559 |
| Age | 1 | 20.003416 | 0.0505 |
| Parents and Children | 0 | 0 | 0.0000 |

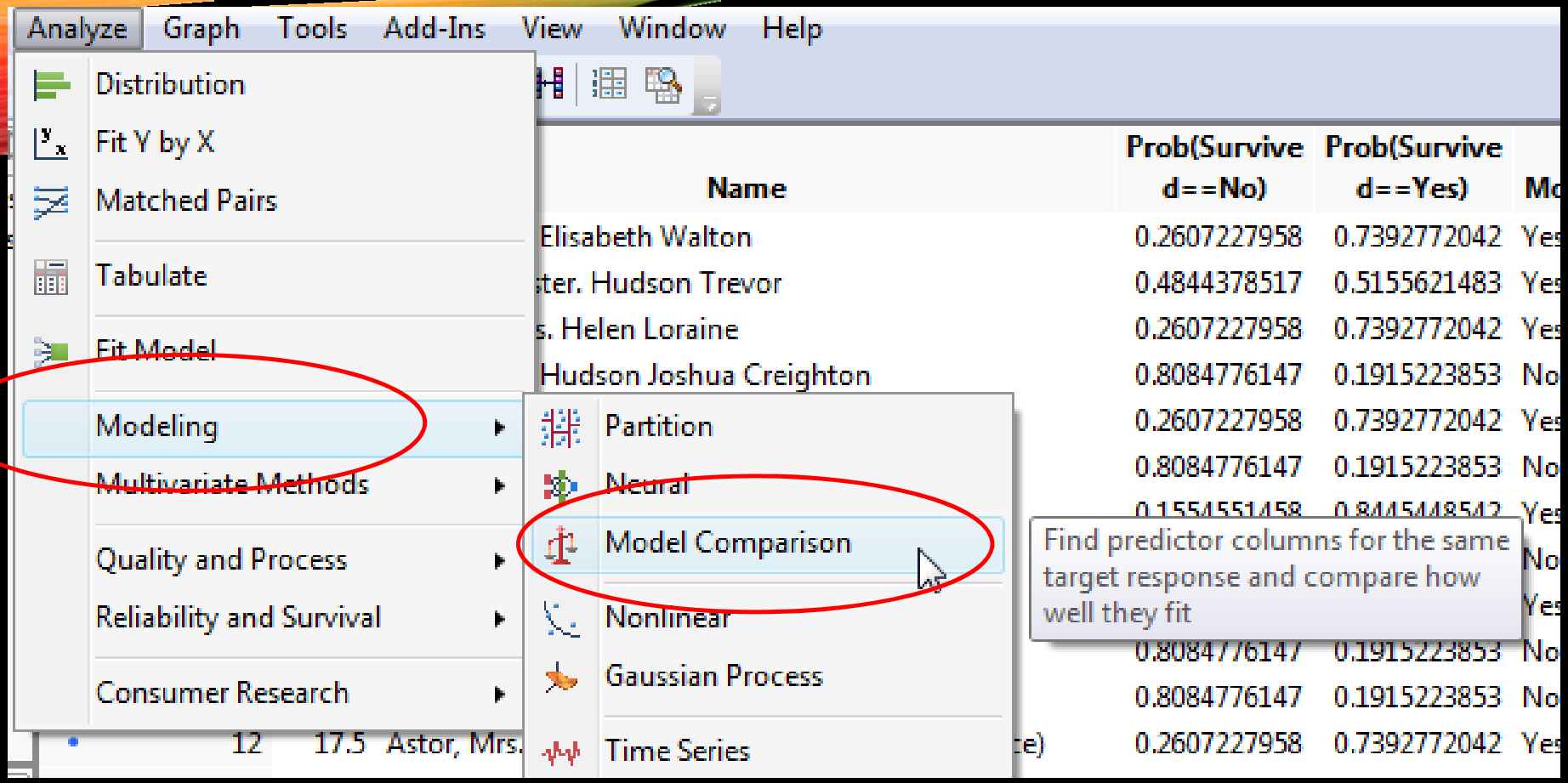


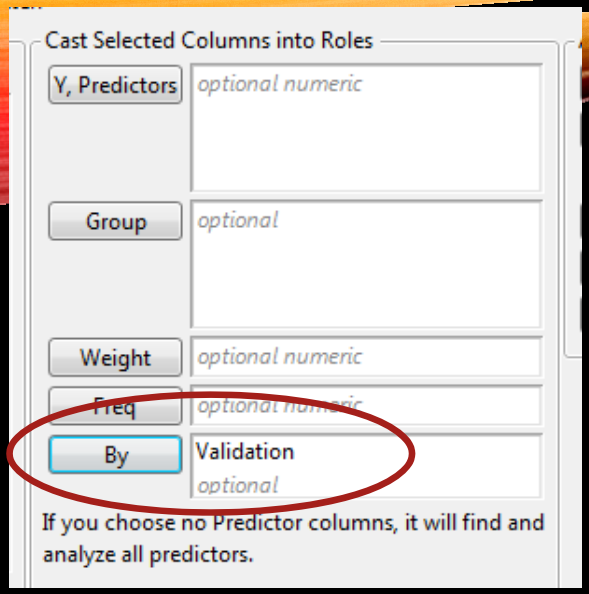

Receiver Operating Characteristic on Validation Data
1.00
0.90
0.80
0.70
0.60
0.50
0.40
0.30
0.20
0.10
0.00
Sensitivity
0.00 0.10 0.20 0.30 0.40 0.50 0.60 0.70 0.80 0.90 1.00
1-Specificity
Survived
Area
No
0.8704
Yes
0.8704


If
Sex == "female" ⇒ If
Passenger Class == 1 | Passenger Class == 2 ⇒ 0.92935733715445
Passenger Class == 3 ⇒ 0.48236716447598
else ⇒ 0.72876755850582
Sex == "male" ⇒ If
Passenger Class == 1 ⇒ 0.3801117660895
Passenger Class == 2 | Passenger Class == 3 ⇒ If
! Is Missing Age & Age < 10 ⇒ If
Siblings and Spouses < 3 ⇒ 0.80871696234289
else ⇒ 0.08535139278218
else ⇒ 0.14138958376162
else ⇒ 0.20906205276871
else ⇒ 0.39192139737991

# MODEL COMPARISON




Analyze
Graph
Tools
Add-Ins
View
Window
Help
Distribution
Fit Y by X
Matched Pairs
Tabulate
Fit Model
Modeling
Multivariate Methods
Quality and Process
Reliability and Survival
Consumer Research
Partition
Neural
Model Comparison
Nonlinear
Gaussian Process
Time Series
Find predictor columns for the same target response and compare how well they fit
Name
Prob(Survived==No)
Prob(Survived==Yes)
Elisabeth Walton 0.2607227958 0.7392772042
ter. Hudson Trevor 0.4844378517 0.5155621483
s. Helen Loraine 0.2607227958 0.7392772042
Hudson Joshua Creighton 0.8084776147 0.1915223853
0.2607227958 0.7392772042
0.8084776147 0.1915223853
0.8084776147 0.1915223853
0.8084776147 0.1915223853
12 17.5 Astor, Mrs. 0.2607227958 0.7392772042




## Model Comparison Validation=Training

### Predictors

### Measures of Fit for Survived

| Creator | .2 .4 .6 .8 | Entropy RSquare | Generalized RSquare | Mean -Log p | RMSE | Mean Abs Dev | Misclassification Rate | N |
|---|---|---|---|---|---|---|---|---|
| Bootstrap Forest | | 0.2188 | 0.3442 | 0.5231 | 0.4116 | 0.3803 | 0.2063 | 916 |
| Boosted Tree | | 0.3513 | 0.5085 | 0.4344 | 0.3716 | 0.2881 | 0.1932 | 916 |
| Partition | | 0.3227 | 0.4755 | 0.4535 | 0.3805 | 0.2908 | 0.2118 | 916 |

## Model Comparison Validation=Validation

### Predictors

### Measures of Fit for Survived

| Creator | .2 .4 .6 .8 | Entropy RSquare | Generalized RSquare | Mean -Log p | RMSE | Mean Abs Dev | Misclassification Rate | N |
|---|---|---|---|---|---|---|---|---|
| Bootstrap Forest | | 0.2527 | 0.3854 | 0.4878 | 0.3918 | 0.3644 | 0.1858 | 393 |
| Boosted Tree | | 0.3788 | 0.5352 | 0.4055 | 0.3537 | 0.2791 | 0.1756 | 393 |
| Partition | | 0.3729 | 0.5287 | 0.4093 | 0.3560 | 0.2782 | 0.1832 | 393 |

## Model Comparison Validation=Validation

### Predictors

**Target Column Predictors**

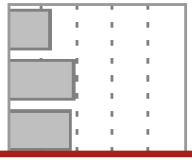
Survived

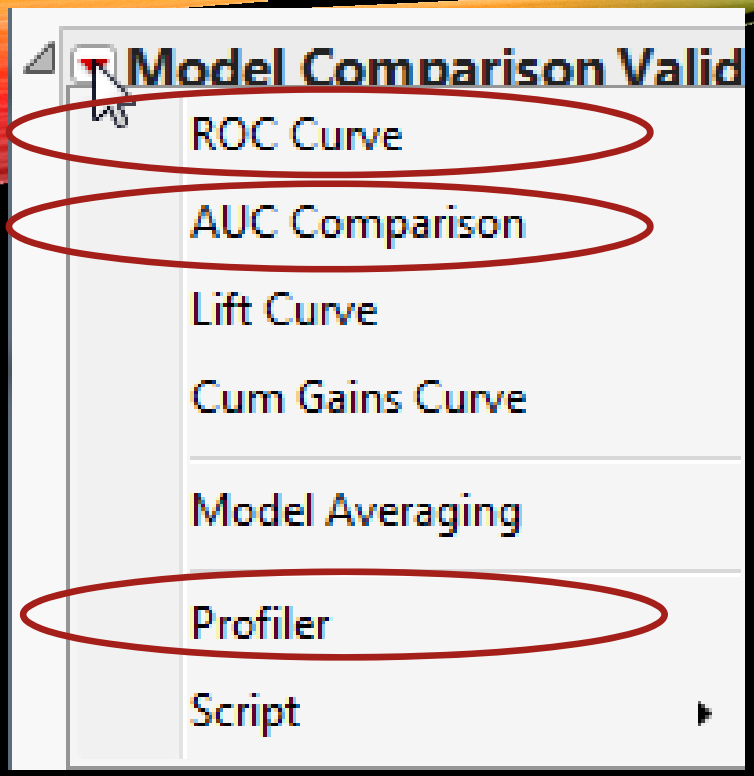
| Category | Probability Column | |
|---|---|---|
| No | Prob(Survived==No) | Bootstrap Forest |
| Yes | Prob(Survived==Yes) | |

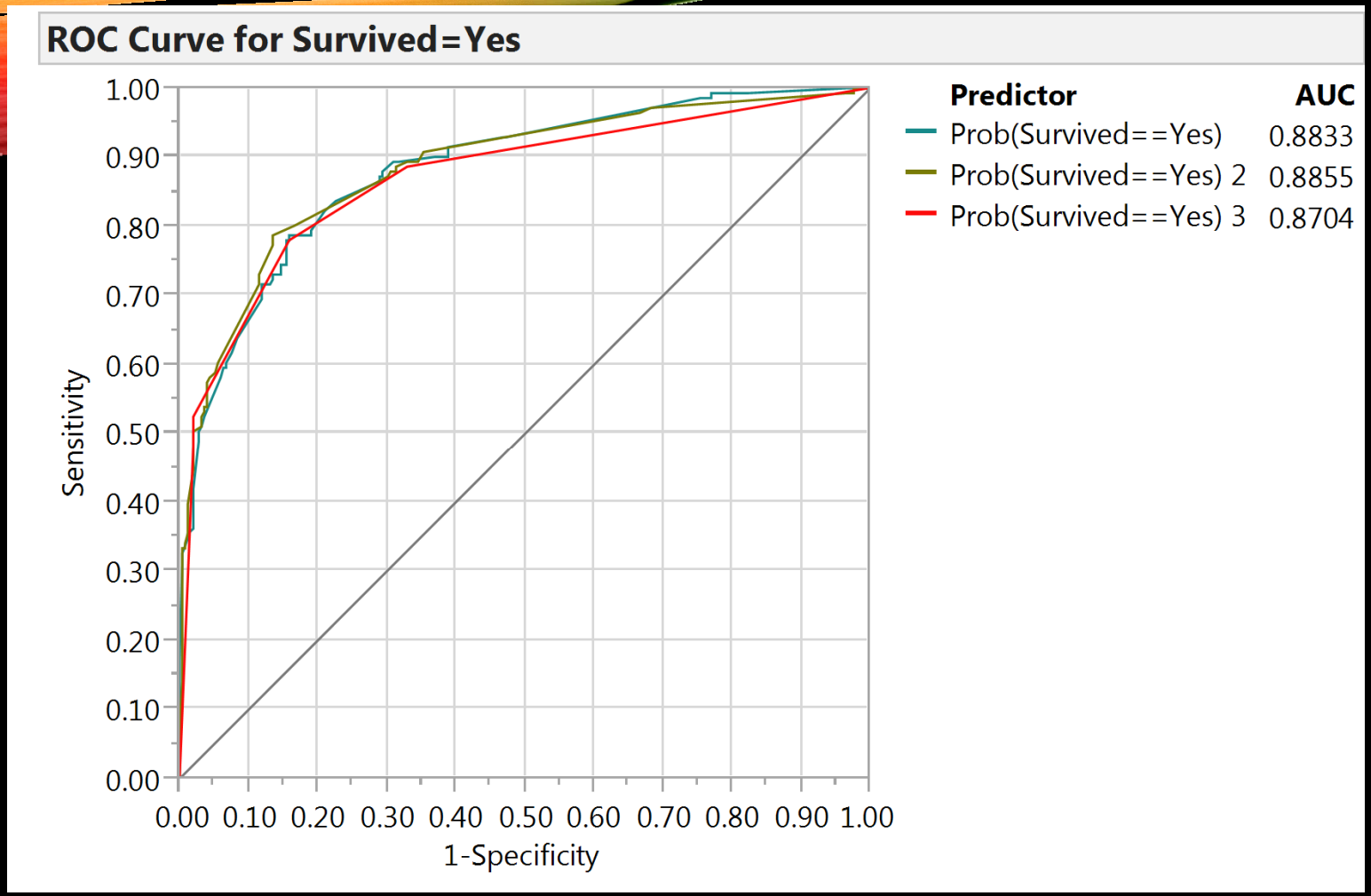
| Category | Probability Column | |
|---|---|---|
| No | Prob(Survived==No) 2 | Boosted Tree |
| Yes | Prob(Survived==Yes) 2 | |

| Category | Probability Column | |
|---|---|---|
| No | Prob(Survived==No) 3 | Partition |
| Yes | Prob(Survived==Yes) 3 | |

### Measures of Fit for Survived

| Creator | .2 .4 .6 .8 | Entropy RSquare | Generalized RSquare | Mean -Log p | RMSE | Mean Abs Dev | Misclassification Rate | N |
|---|---|---|---|---|---|---|---|---|
| Bootstrap Forest |  | 0.2527 | 0.3854 | 0.4878 | 0.3918 | 0.3644 | 0.1858 | 393 |
| Boosted Tree | | 0.3788 | 0.5352 | 0.4055 | 0.3537 | 0.2791 | 0.1756 | 393 |
| Partition | | 0.3729 | 0.5287 | 0.4093 | 0.3560 | 0.2782 | 0.1832 | 393 |


Model Comparison Valid
ROC Curve
AUC Comparison
Lift Curve
Cum Gains Curve
Model Averaging
Profiler
Script


ROC Curve for Survived=Yes
Sensitivity
1.00
0.90
0.80
0.70
0.60
0.50
0.40
0.30
0.20
0.10
0.00
0.00 0.10 0.20 0.30 0.40 0.50 0.60 0.70 0.80 0.90 1.00
1-Specificity
Predictor
AUC
Prob(Survived==Yes) 0.8833
Prob(Survived==Yes) 2 0.8855
Prob(Survived==Yes) 3 0.8704

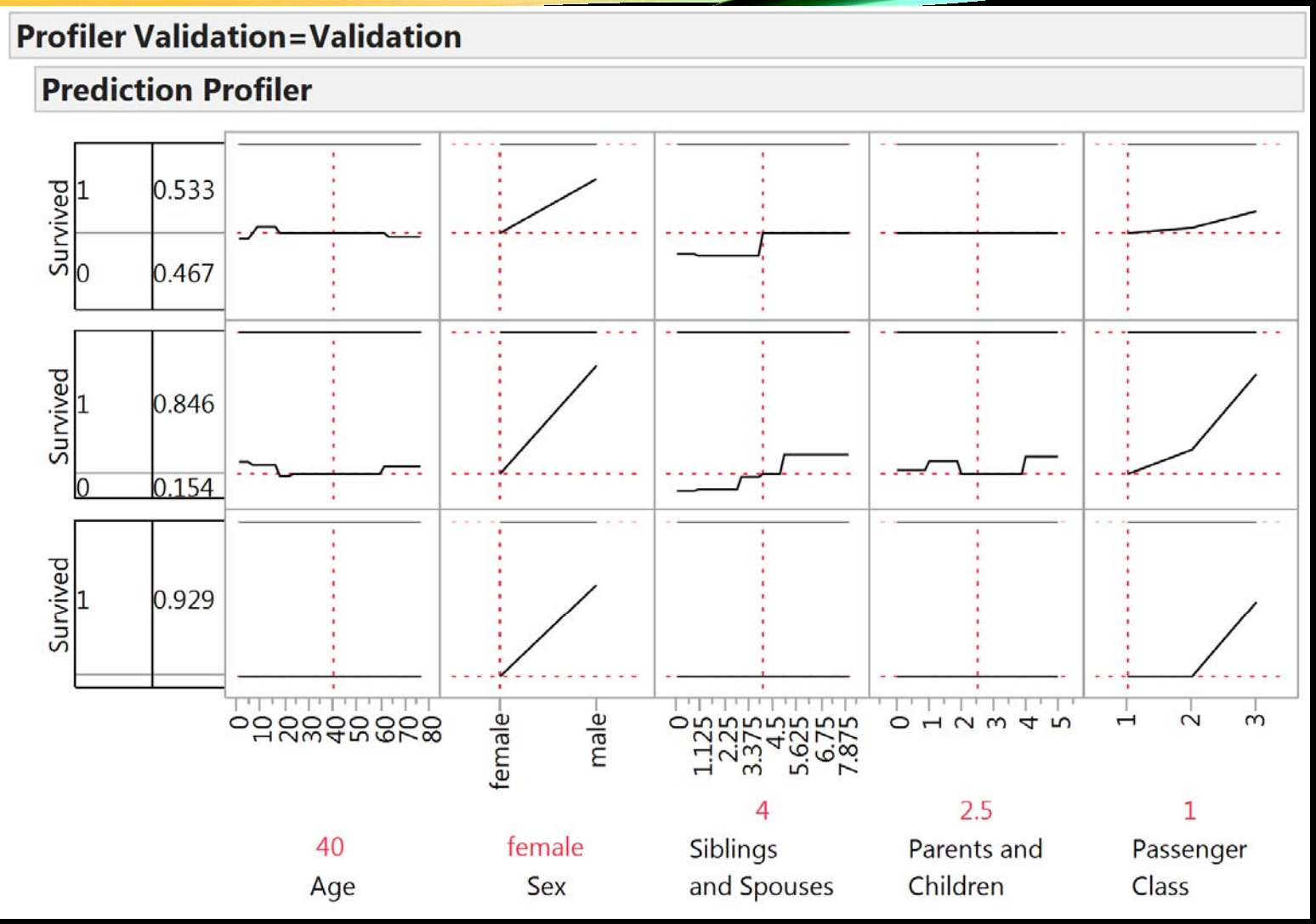
## AUC Comparison

### AUC Comparison for Survived=Yes

| Predictor | AUC | Std Error | Lower 95% | Upper 95% |
|---|---|---|---|---|
| Prob(Survived==Yes) | 0.8833 | 0.0178 | 0.8437 | 0.9139 |
| Prob(Survived==Yes) 2 | 0.8855 | 0.0182 | 0.8446 | 0.9167 |
| Prob(Survived==Yes) 3 | 0.8704 | 0.0192 | 0.8279 | 0.9037 |

| Predictor | vs. Predictor | AUC Difference | Std Error | Lower 95% | Upper 95% | ChiSquare | Prob>ChiSq |
|---|---|---|---|---|---|---|---|
| Prob(Survived==Yes) | Prob(Survived==Yes) 2 | -0.002 | 0.0064 | -0.015 | 0.0103 | 0.1176 | 0.7317 |
| Prob(Survived==Yes) | Prob(Survived==Yes) 3 | 0.0129 | 0.0076 | -0.002 | 0.0278 | 2.8789 | 0.0897 |
| Prob(Survived==Yes) 2 | Prob(Survived==Yes) 3 | 0.0151 | 0.0074 | 0.0007 | 0.0295 | 4.1977 | 0.0405* |

| Test | ChiSquare | DF | Prob>ChiSq |
|---|---|---|---|
| All AUCs equal | 4.45261 | 2 | 0.1079 |


Profiler Validation=Validation
Prediction Profiler
Survived
1
0
0.533
0.467
0.846
0.154
0.929
0 10 20 30 40 50 60 70 80
female male
0 1.125 2.25 3.375 4.5 5.625 6.75 7.875
0 1 2 3 4 5
1 2 3
40
Age
female
Sex
4
Siblings and Spouses
2.5
Parents and Children
1
Passenger Class