DECISION TREES

JMP Partition Platform

Predict

What will happen

model (continuous response)

- Multiple Regression
- Regression Trees
- Bootstrap Forest
- Boosted Tree
- Neural Network

model (categorical response)

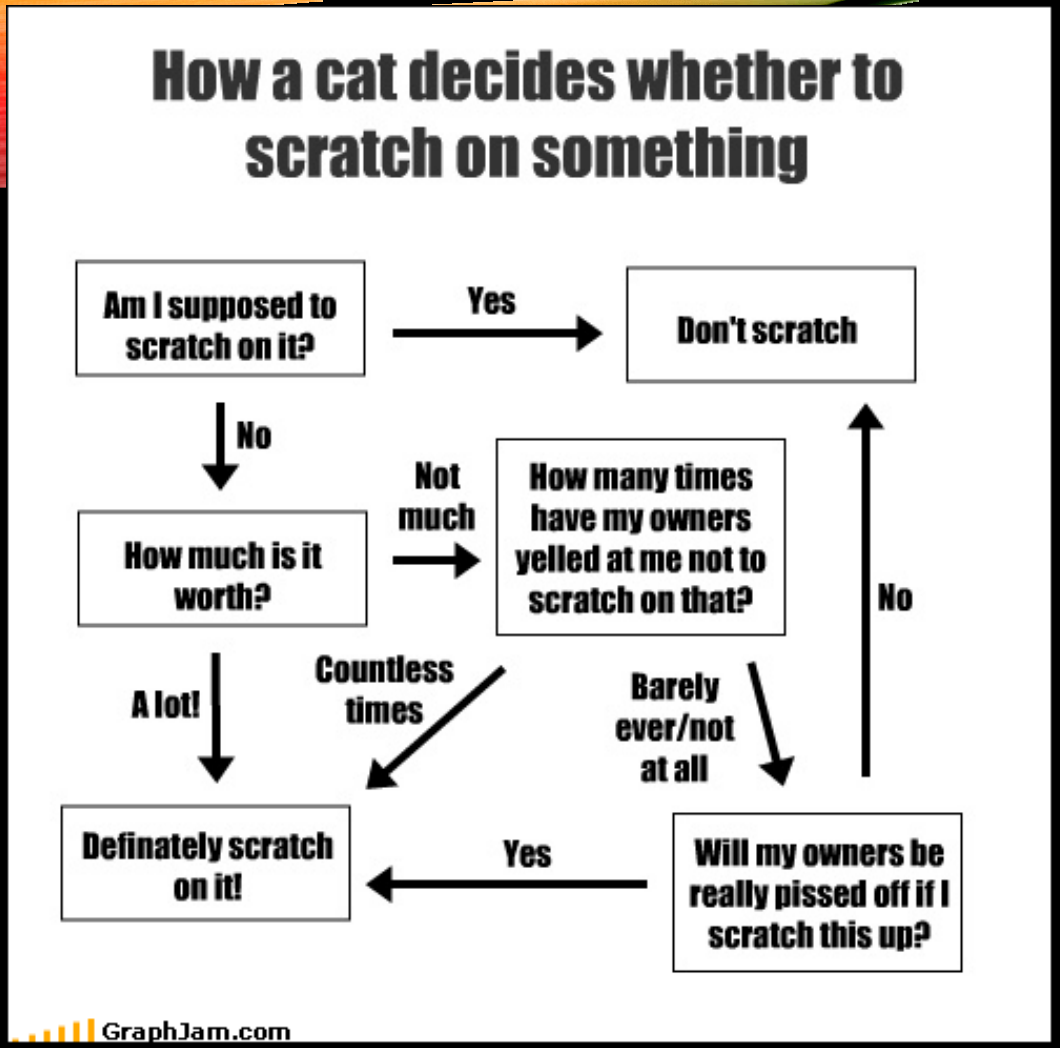
- Decision Trees
- Logistic Regression
- Discriminant Analysis
- Bootstrap Forest
- Boosted Tree
- Neural Network

Source: Jim Grayson, Ph.D. and Mia Stephens

TREE METHODS LEARNING GOALS

1. Be able to explain the usefulness of a decision tree
2. Be able to explain the logic of a decision tree
3. Be able to explain the differences between a classification tree and a regression tree
4. Be able to use JMP (analyze > modeling > partition) to construct a classification tree and use split history and pruning for selecting a best tree
5. Given a classification tree
    1. construct the If-Then prediction model
    2. interpret the predictors and their influence using Column Contribution
    3. explain the interpretation of G^2 and LogWorth
6. Be able to develop a predictive classification tree.
7. Be able to evaluate the predictive performance of a classification tree using the misclassification rate, Confusion Matrix and ROC curve
8. Be able to explain to a manager the insights from a classification tree
9. Be able to use Trees for problem insights and dimension reduction.

A CAT STORY



Funny Cats Compilation [Most See] Funny Cat Videos Ever ...
www.youtube.com/watch?v=tntOCGkgt98
Dec 30, 2013 - Uploaded by Forget Your Sadness
Get up to 40000 Instagram followers: http://imisland.us/instagram/
Funny Cats Compilation [Most See] Funny ...

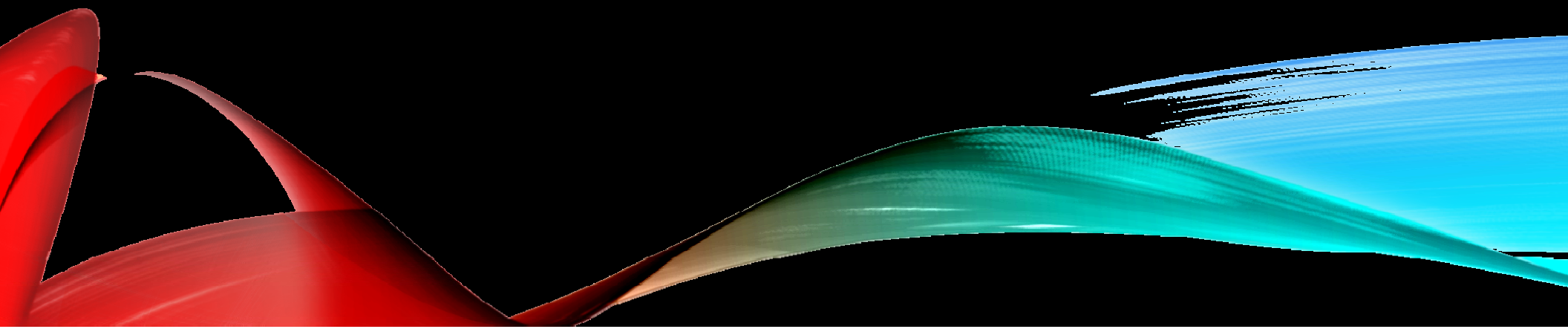
Best funny and cute cat videos compilation 2014 - YouTube
www.youtube.com/watch?v=p2H5YVfZVFw
Jan 13, 2014 - Uploaded by Tiger Productions
Awesome Halloween costumes for babies, children and adults: For babies: Puppy costume: ...

Funny Cats Videos 2014 - YouTube
www.youtube.com/watch?v=1s5bF6WCGqQ
Jan 31, 2014 - Uploaded by anacalderon1
cute cats| cute dogs | cats and dogs | cute cats and dogs | jealous cat | cute dogs and cats| cute | cats ...

Cat videos of cats - YouTube
www.youtube.com/watch?v=i_mKY2CQ9Kk
Jan 29, 2014 - Uploaded by anacalderon1
GOOGLE+ COMMUNITY Adventure of Cats and Dogs
https://plus.google.com/u/0/communities ...

Funny cat vines - Ultimate funny vines with cats ... - YouTube
www.youtube.com/watch?v=ZIOsu870j8E
Jun 29, 2014 - Uploaded by Funny Videos
CLICK HERE TO SUBSCRIBE http://goo.gl/XSdB11 Cat Vines | Best Cat Vines | Funny Cat Vines | Cat Vine ...

WHAT IS IT?



Decision trees are used for both exploratory data analysis and for predictive modeling. They are relatively easy to understand and explain, particularly to a non-technical audience.

Here are some typical question addressed by decision trees:

- Will a patient be readmitted to the hospital over the next 30 days?
- Will a new movie be a blockbuster? How much will it earn?
- Will a given candidate win an election?
- What is the probability that a flight will be delayed?
- Which variables, out of a large set of potential predictors, are most important in predicting housing prices?
- What is the risk that an applicant will default on a loan?

Excerpt from Chapter 6: Decision Trees from Building Better Models using JMP (Draft) by Grayson, Gardner and Stephens (SAS Press).

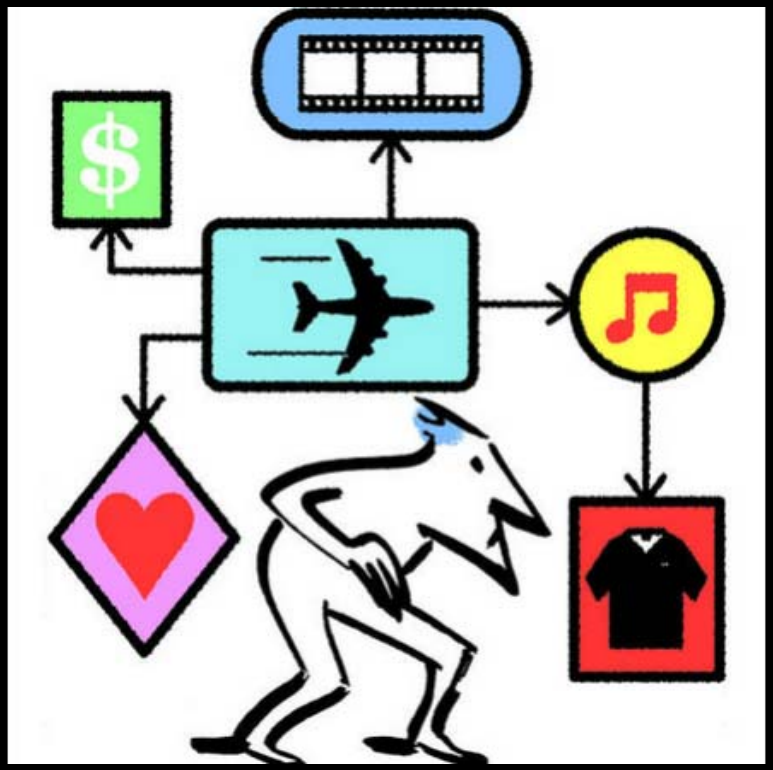
LIFE | IDEAS | MOVING TARGETS

Those Product Recommendations Just Don't Compute

Joe Queenan finds that the algorithms used by Netflix, Amazon and other companies to recommend products are hopelessly wrong

The Wall Street Journal. | Saturday/Sunday, June 20-21, 2015 | C11

By JOE QUEENAN
June 18, 2015 11:30 a.m. ET



Defective algorithms are making it increasingly difficult to function in everyday life. With data in hand about our past purchases and product ratings, many companies now use these mathematical formulas to determine our supposed likes and dislikes. They use algorithms to guess what films we'd like to watch, what music we'd like to download, what cities we'd like to visit and what gifts we'd like to give. The streaming-music service Spotify is just one gigantic algorithm.

But with shocking frequency, I find that the algorithms used by Netflix, Amazon, Expedia and other companies to predict my tastes are hopelessly wrong. A case in point: I like to watch violent movies in which things get blown up. I will watch anything starring Jason Statham, Bruce Willis, Chow Yun-Fat, the Rock, Arnie or Jet Li, but in a pinch I will also watch violent films starring Nicolas Cage, Jean-Claude Van Damme, Clive Owen, Jason Patric and even John Cusack.

The Wall Street Journal. | Saturday/Sunday, June 20-21, 2015 | C11

Why then, under "Top Picks for Joseph," did Netflix suggest that I watch "Grace and Frankie," a comedy series starring Jane Fonda and Lily Tomlin, about two women whose husbands leave them for each other? Why would Netflix suggest that I watch "Hector and the Search for Happiness," in which an unfulfilled shrink sets out on a trip around the world in search of true happiness? What was there about my recent viewing—"Death Wish," "Death Wish 2," "Killer Mermaid," "All Hell Broke Loose," "The Guillotines"—that would suggest I'd be interested in watching a lighthearted comedy about an unhappy psychiatrist?

Reviewing my recent viewing habits, I saw only one film out of the last 20 that was not absurdly violent or scary or twisted or stupid. That was "Shun Li and the Poet," a sweet little film about a Chinese bartender in Venice who develops an unusual friendship with a fisherman from Yugoslavia. But surely the *algorithmisti* at Netflix should have realized that watching that film was my wife's idea, not mine! Moreover, nothing in our collective viewing patterns would suggest that either my wife or I would ever watch a comedy series starring Jane Fonda and Lily Tomlin. In my house, we have standards.

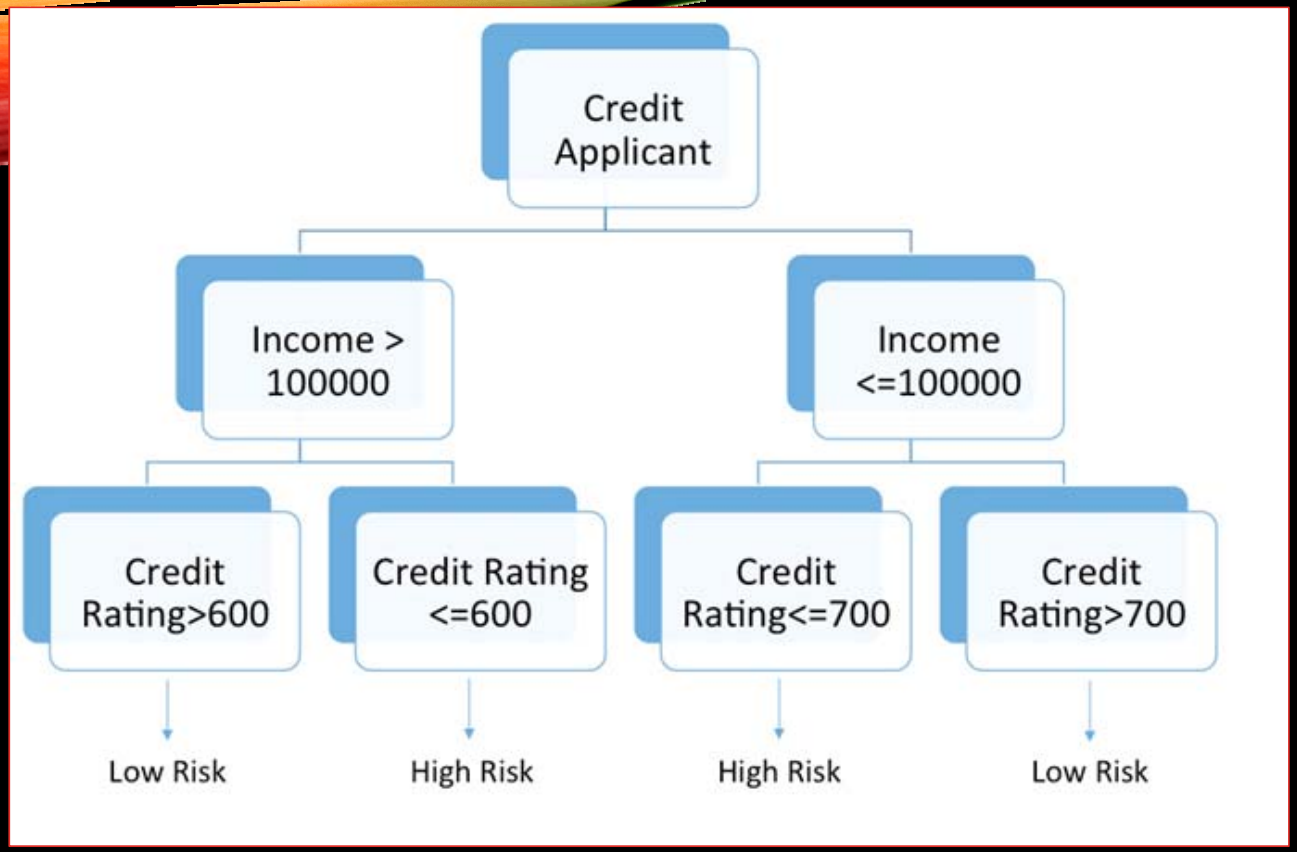
The Wall Street Journal. | Saturday/Sunday, June 20-21, 2015 | C11

A decision tree consists of a set of conditional rules, based on simple decision thresholds. The "tree" is essentially a series of nested if-then statements that lead to a classification or prediction.

As an example, conditional rules that could be used in a credit risk assessment might look like this:

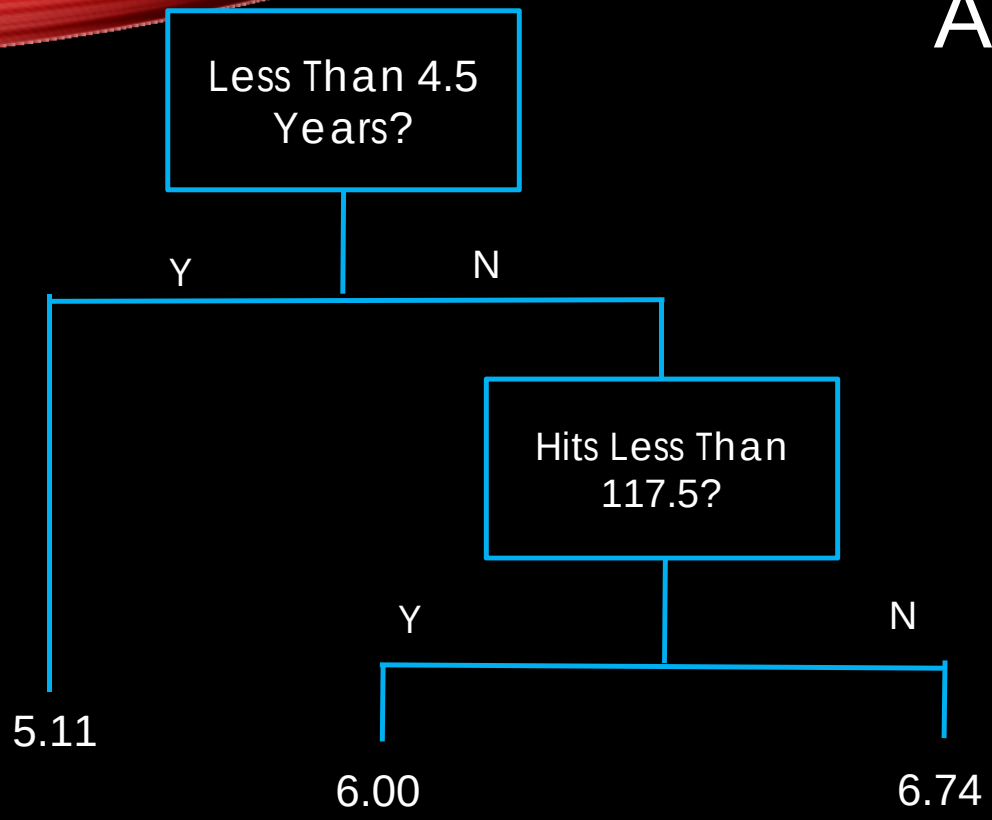
- If income is > 100,000 and credit rating is > 600 then credit risk is low, but if credit rating is < 600 then credit risk is high, and
- If income is <=100,000 and credit rating is < 700 then credit risk is high, but if credit rating is > 700 then credit risk is low.

Excerpt from Chapter 6: Decision Trees from Building Better Models using JMP (Draft) by Grayson, Gardner and Stephens (SAS Press).

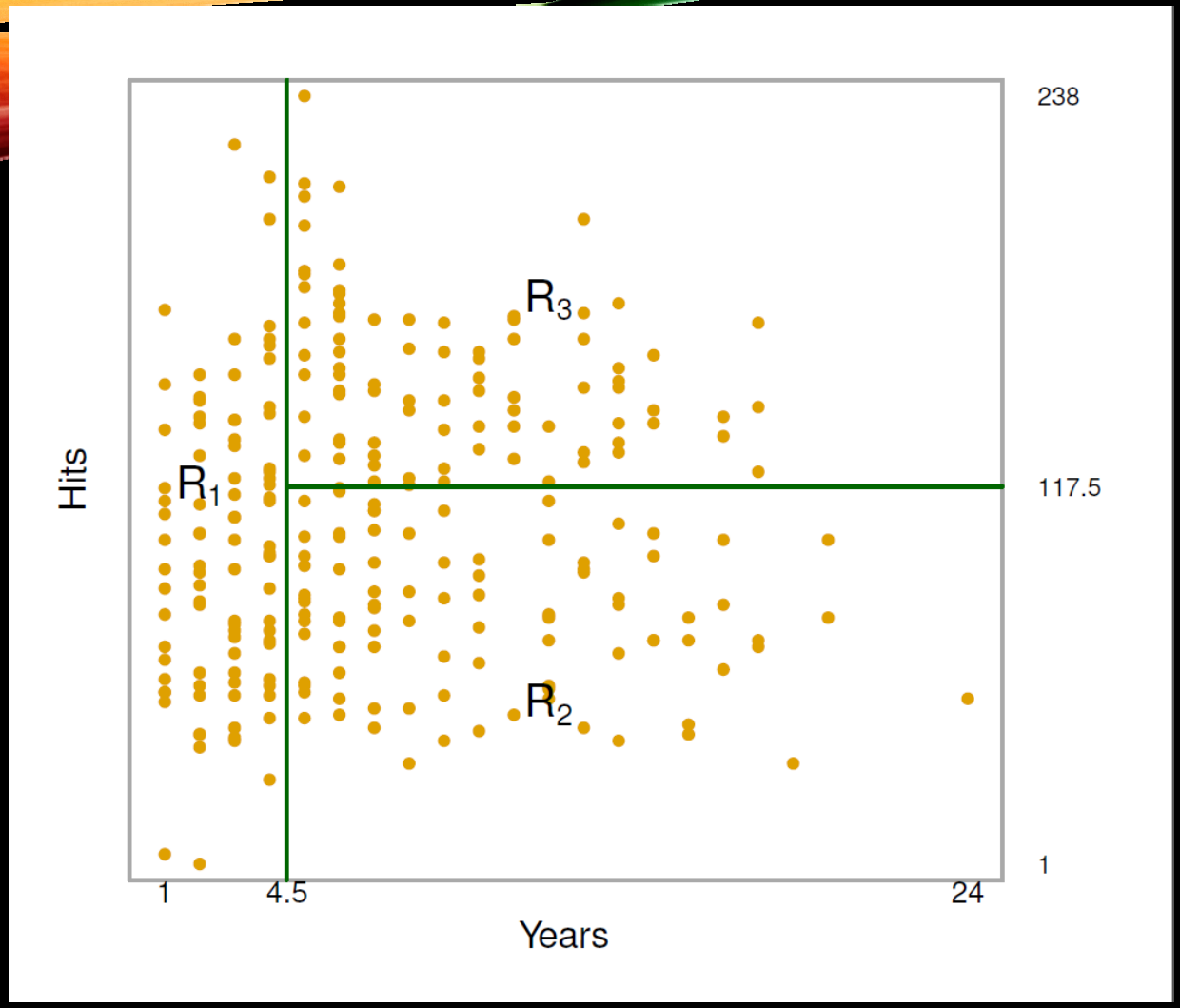


Excerpt from Chapter 6: Decision Trees from Building Better Models using JMP (Draft) by Grayson, Gardner and Stephens (SAS Press).

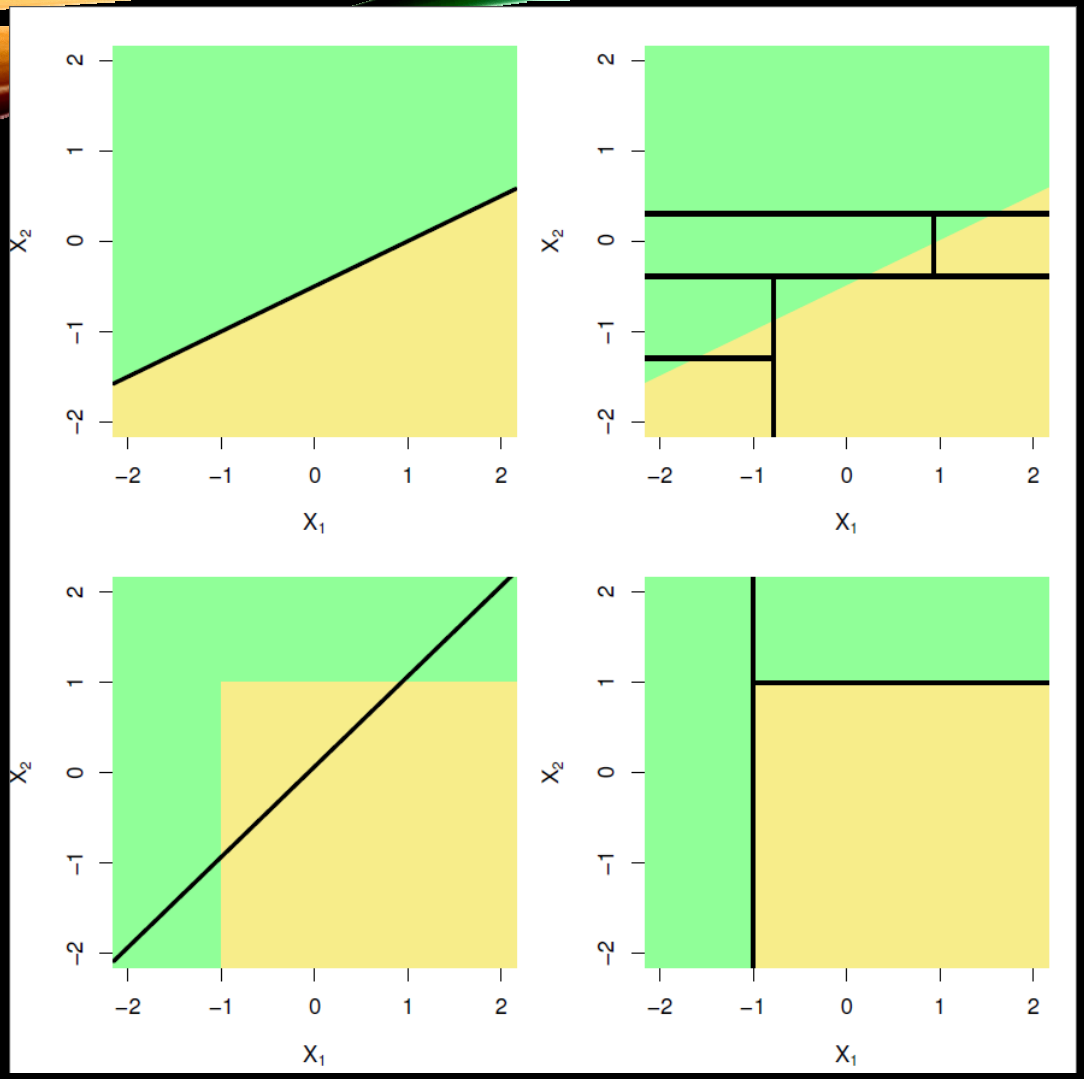
PREDICTING THE LOG SALARY OF A BASEBALL PLAYER



Chapter 8, Figure 8.1, An Introduction to Statistical Learning, James, et al

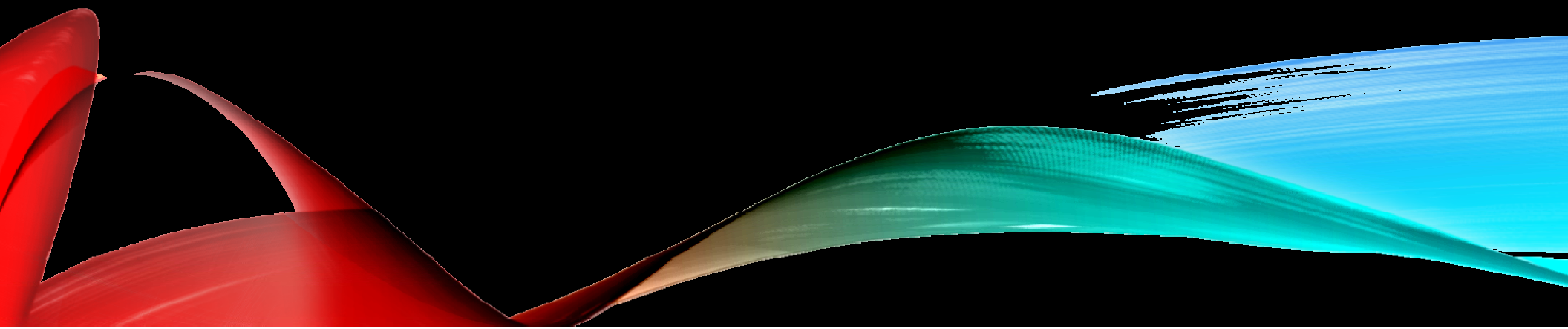


Chapter 8, Figure 8.2, An Introduction to Statistical Learning, James, et al



Chapter 8, Figure 8.7, An Introduction to Statistical Learning, James, et al

HOW IDENTIFIED?



Objective	Categorical Response	Continuous Response
Exploratory Analysis (understand data / structure)	Classification Trees	Regression Trees
Descriptive Analytics (identify important variables)	Classification Trees Logistic Regression	Regression Trees Multiple Linear Regression
Predictive Analytics (predict the future)	Classification Trees Logistic Regression Neural Networks	Regression Trees Multiple Linear Regression Neural Networks
Prescriptive Analytics (optimal settings)		Multiple Linear Regression

Decision Tree (Classification Tree) is an approach for predicting a categorical response variable (Y) with continuous or categorical predictor variables (X).

Y	X	Objective	Predictive Accuracy	Statistical Significance Measure	Model Fit (Error)	Operational Understanding
Categorical	Continuous or Categorical	Classification	Mis-classification Rate	Entropy RSquare	RMSE, MAD, ROC Curve	Column Contributions; Confusion Matrix

All performance in Predictive Analytics is based on the validation set and not the training set

WHY CHOOSE TREES (PARTITION PLATFORM)?

1. Categorical (Classification Tree) or Continuous (Regression Tree) Response Variable

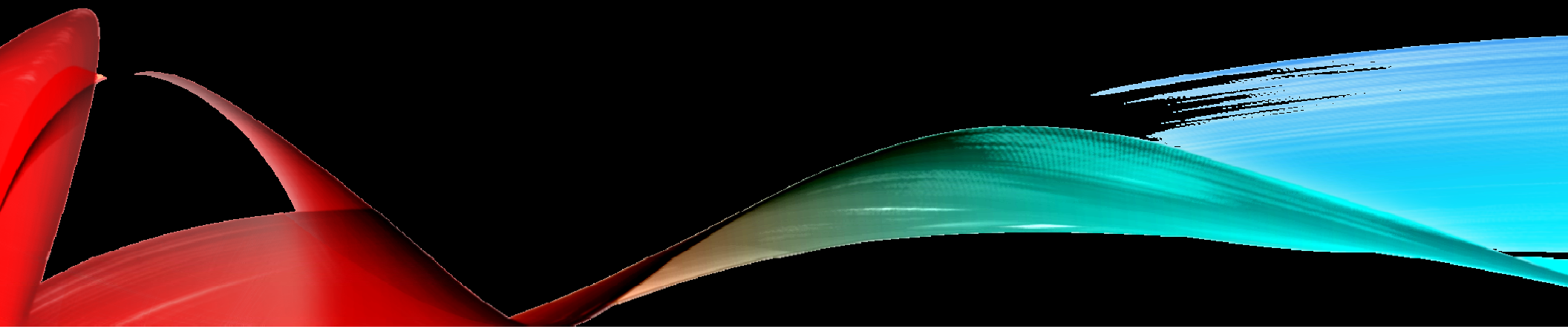
2. Can be used for classification or prediction

3. Easy to understand and explain

4. Intuitive

HOW TO DO IT

(mechanics of JMP)



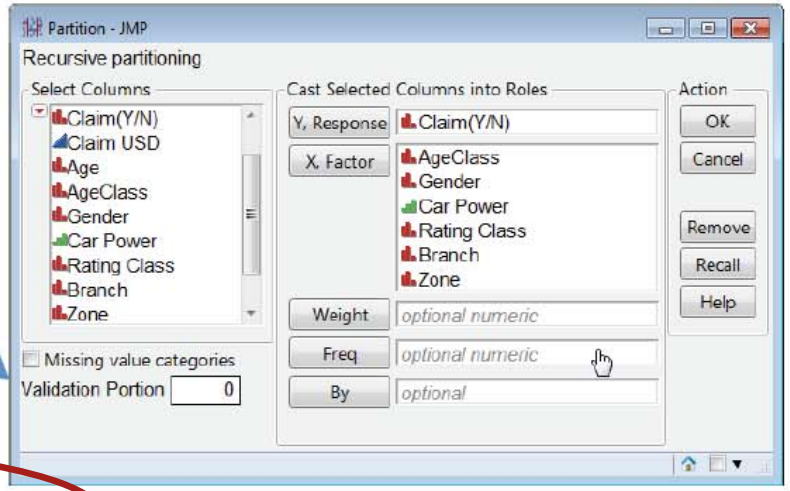
Classification Trees (Partition)

Use this data mining technique to predict a categorical (nominal or ordinal) response as a function of potential predictor variables using recursive partitioning.

Classification Trees

1. From an open table, select **Analyze > Modeling > Partition.**
2. Select a nominal or ordinal response variable from **Select Columns** and click **Y, Response**.
3. Select explanatory variables and click **X, Factor**.
4. If desired, enter the **Validation Portion** (a proportion, as shown) or select a validation column and click **Validation (JMP® Pro only)**.
5. In **JMP Pro only**, select the tree **Method**: **Decision Tree** (Default in JMP, shown), **Bootstrap Forest** or **Boosted Tree**.

Example: Auto Raw Data.jmp (Help > Sample Data)



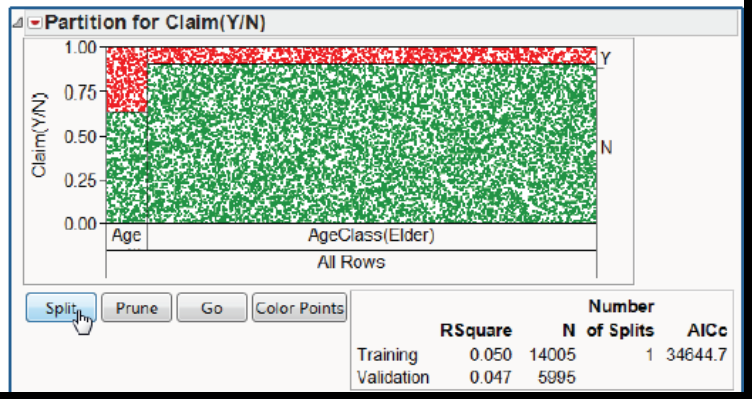
Classification Trees (Partition)

6. Click **OK.** JMP displays:

   - A graph, with horizontal lines drawn at the proportion of observations in each response level.
   - Statistics for the training and validation set(s). Note that results will vary if **Validation Portion** is used.
   - A summary of **All Rows**. Click on the **gray triangle** next to **Candidates** to view split statistics for each column.

7. Click the **Split** button. The original observations will be split into two nodes, or leaves (as shown).

Note: Click on the **top red triangle** and select **Display Options > Show Split Prob** to show **Rates** (proportion of observations) and **Probs** (predicted probabilities) for the response levels in each leaf.

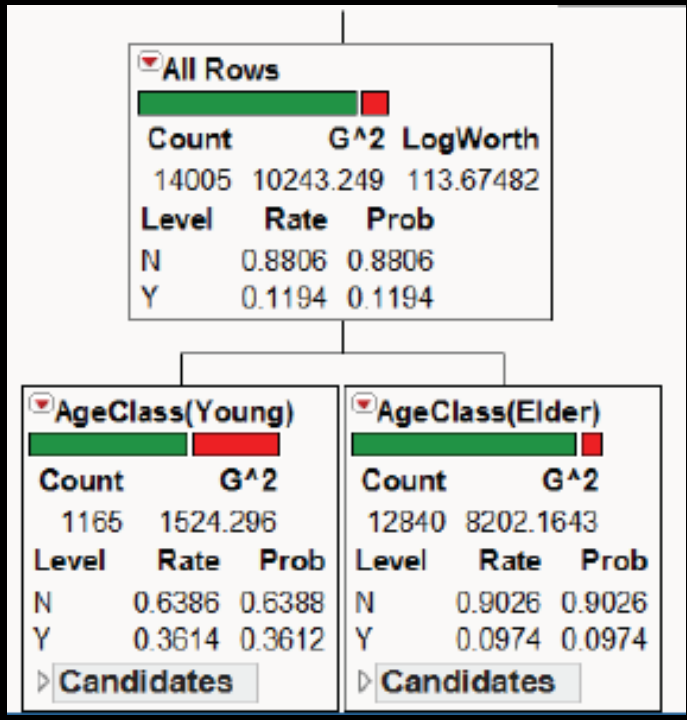


Classification Trees (Partition)

Interpretation (the response, in this example, is Claim (Y/N)):

- There are 1,157 rows in the left leaf, corresponding to AgeClass(Young). The response rate (predicted probability) for Claim(Y/N) = Y is 0.3604.
- There are 12,840 rows in the right leaf, corresponding to AgeClass(Elder). The response rate (predicted probability) for Claim(Y/N) = Y is 0.097.

8. Click **Split** to make an additional split. Click **Prune** to remove a split. If a validation portion or validation column is used, click **Go** to perform automatic splitting.



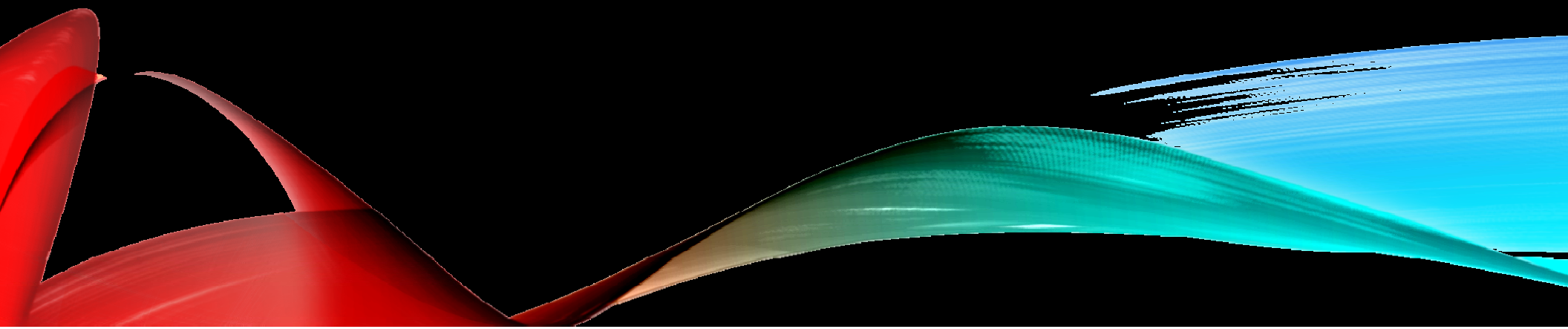
Classification Trees (Partition)

Notes:

For additional options, such as **Column Contributions**, **ROC** and **Lift Curves**, click the **top red triangle**. Other options, such as **Save Prediction Formula** and **Make SAS® DATA Step**, are available from the **top red triangle > Save Columns**. For split options for a particular node, click on the **red triangle for that node**.

For more information on fitting and evaluating classification trees, including **Validation**, **Bootstrap Forest** and **Boosted Trees**, search for "partition trees" in the JMP Help or in the ***Specialized Models*** book (under **Help > Books**).

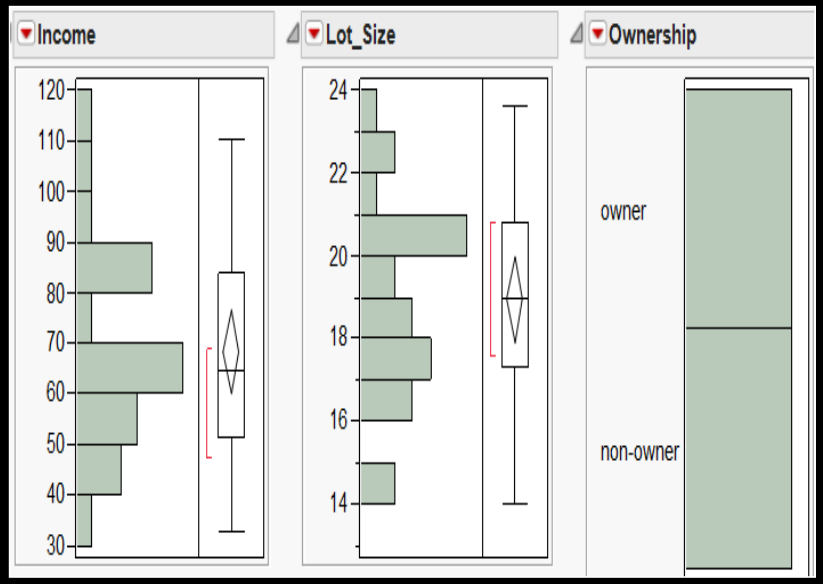
RIDING MOWER EXAMPLE



RIDING MOWERS

A riding mower manufacturer would like to find a way of classifying families in a city into those likely to purchase a riding mower and those not likely to buy one. A pilot random sample is undertaken of 12 owners and 12 non-owners in the city.

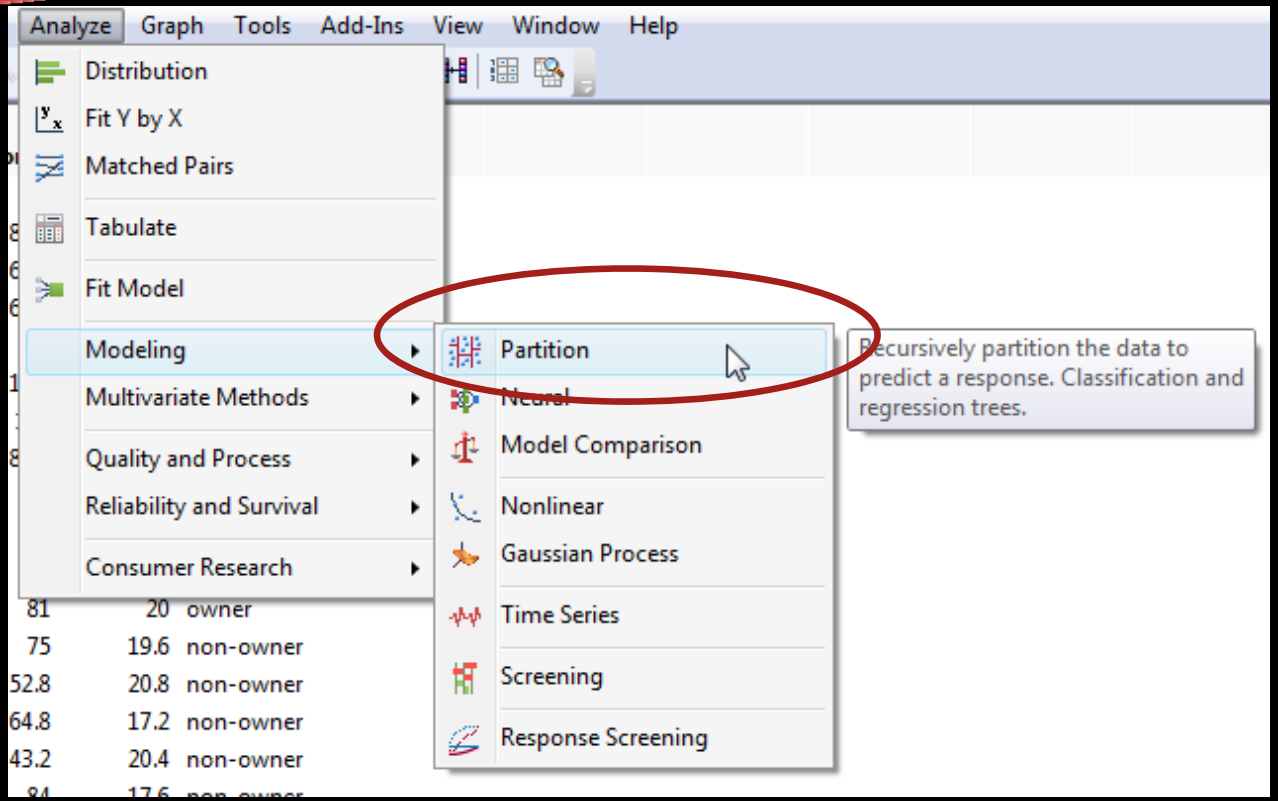
[source: Data Mining for Business Intelligence, 2e by Shmueli, Patel and Bruce, Wiley Publishing]

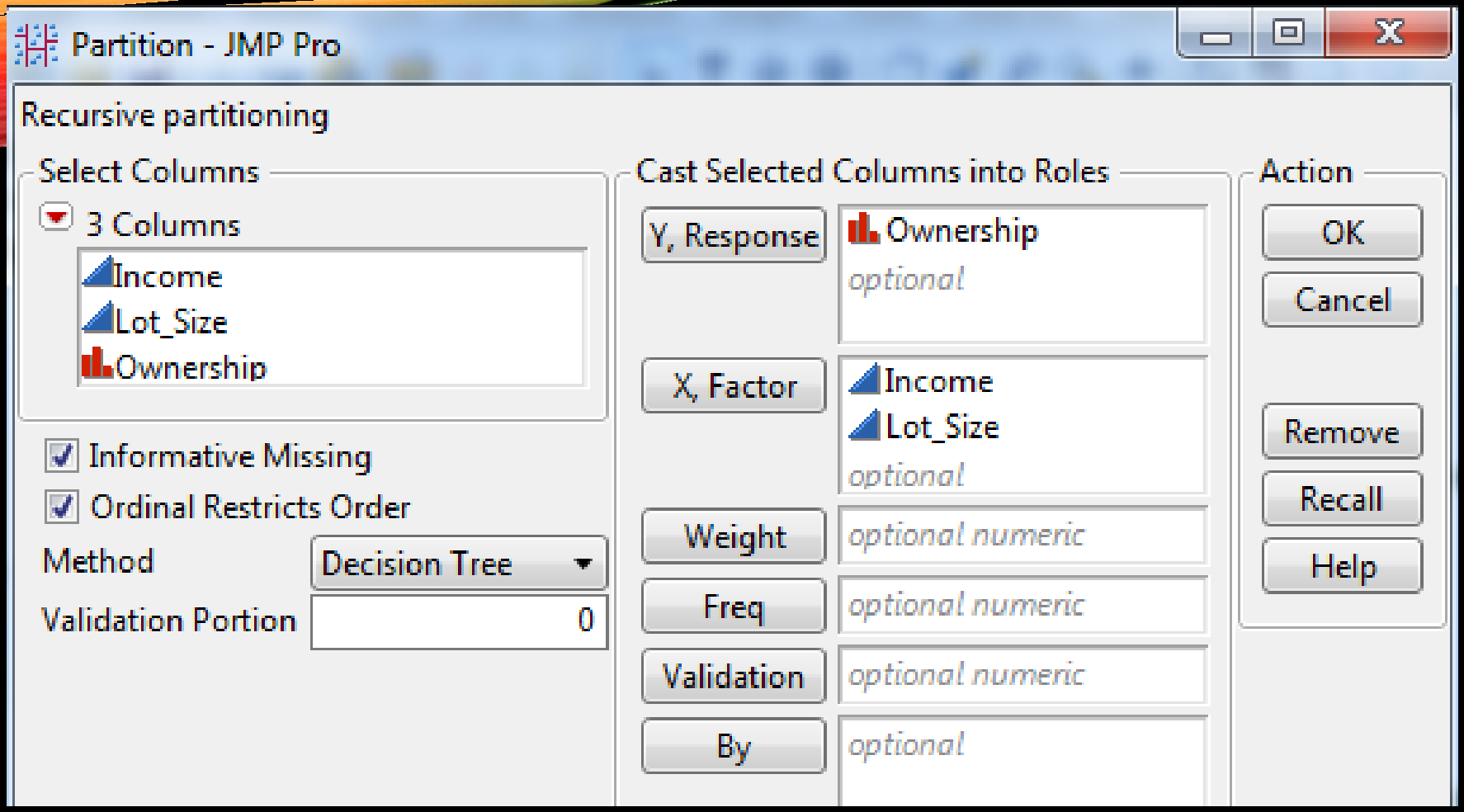


Income	Lot_Size	Ownership
60	18.4	owner
85.5	16.8	owner
64.8	21.6	owner
61.5	20.8	owner
87	23.6	owner
110.1	19.2	owner
108	17.6	owner
82.8	22.4	owner
69	20	owner
93	20.8	owner
51	22	owner
81	20	owner
75	19.6	non-owner
52.8	20.8	non-owner
64.8	17.2	non-owner
43.2	20.4	non-owner
84	17.6	non-owner
49.2	17.6	non-owner
59.4	16	non-owner
66	18.4	non-owner
47.4	16.4	non-owner
33	18.8	non-owner
51	14	non-owner
63	14.8	non-owner

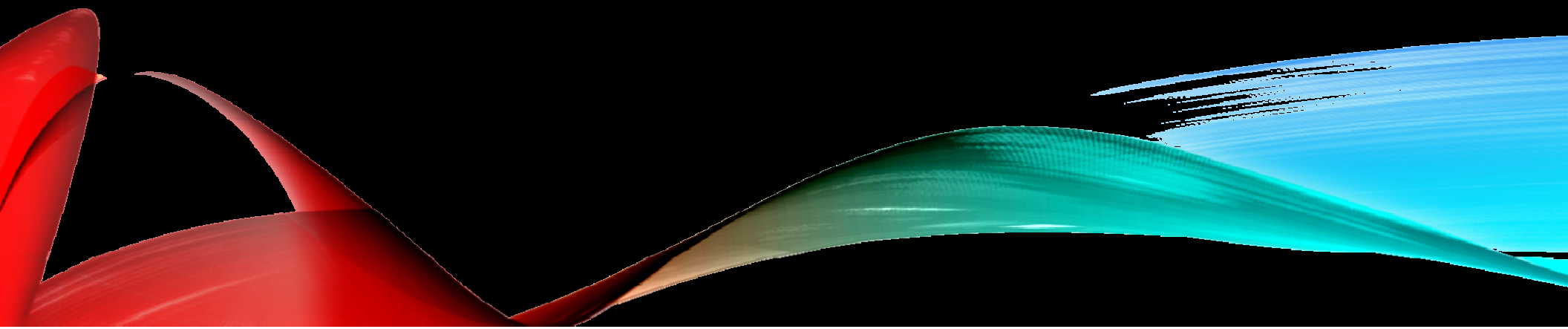
SELECTING THE PARTITION PLATFORM

Analyze > Modeling > Partition




Partition - JMP Pro
Recursive partitioning
Select Columns
3 Columns
Income
Lot_Size
Ownership
Informative Missing
Ordinal Restricts Order
Method
Decision Tree
Validation Portion
0
Cast Selected Columns into Roles
Y, Response
Ownership
optional
X, Factor
Income
Lot_Size
optional
Weight
optional numeric
Freq
optional numeric
Validation
optional numeric
By
optional
Action
OK
Cancel
Remove
Recall
Help

HOW DOES IT WORK?



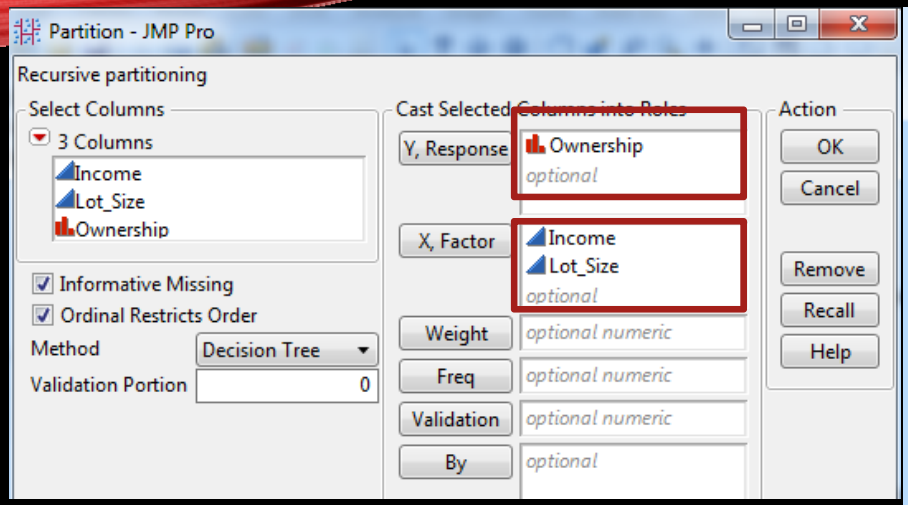
HOW DOES IT WORK?

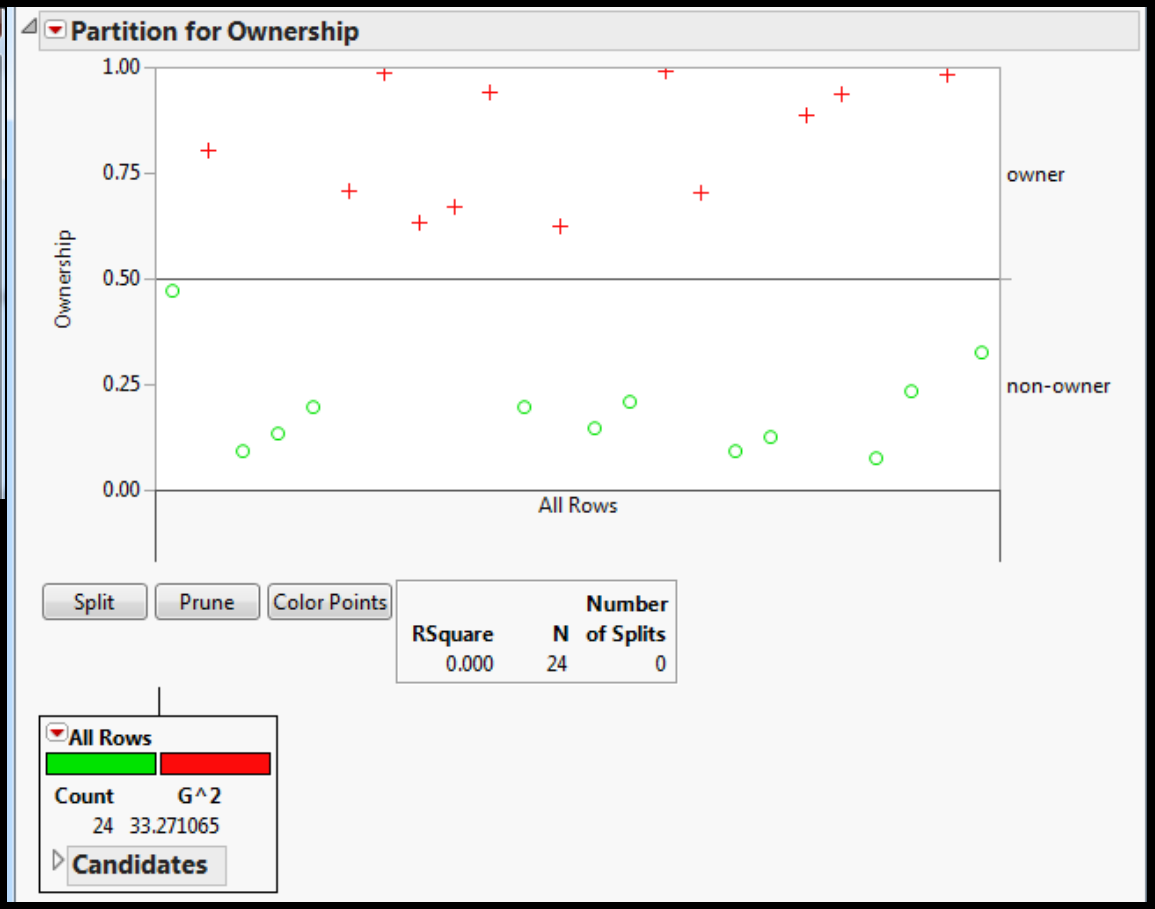
A top down, greedy algorithm known as a **recursive binary split**. Greedy because the algorithm seeks for the immediate "best" split, not the long term down the tree further best split.

Best split for continuous response -> the greatest reduction in RSS (residual sum of squares)

Best split for categorical response -> the smallest measure of node purity, meaning observations are predominantly from a single class

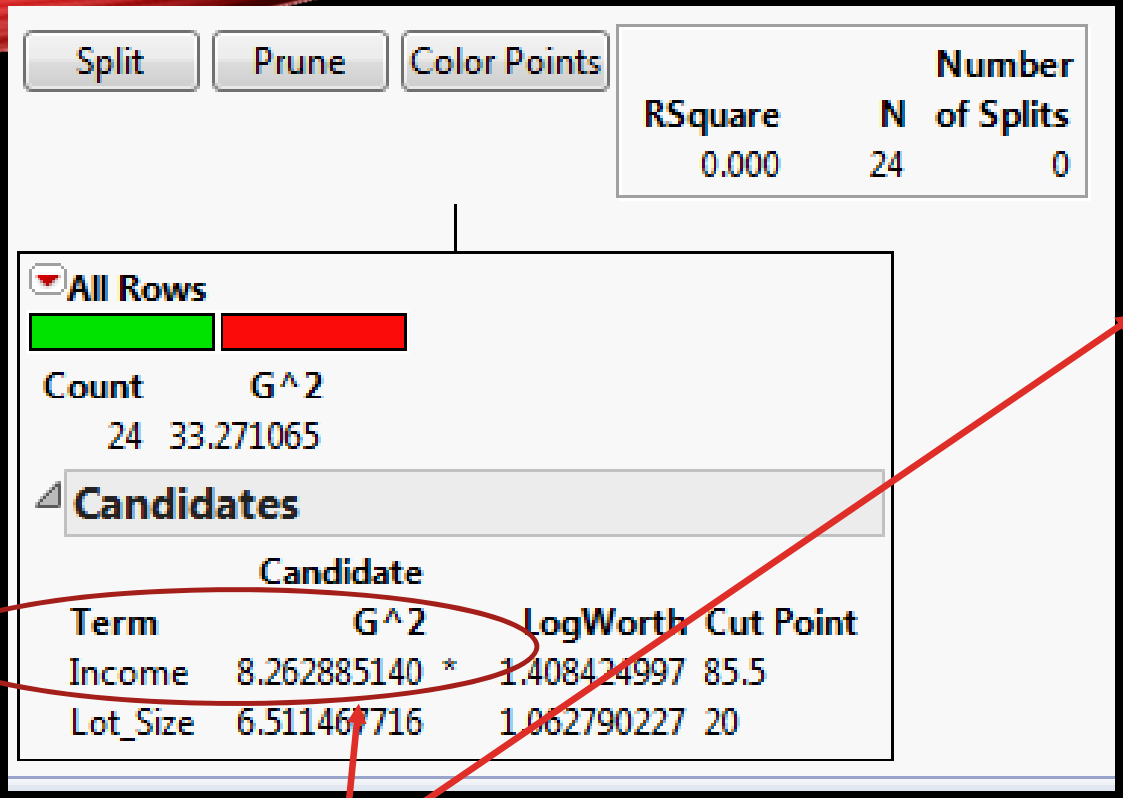
MODEL DIALOG AND STARTING SCREEN





Data source: "RidingMowers" from Data Mining for Business Intelligence, 2e, Shmueli, et al (Wiley)

Status – First Candidate Split



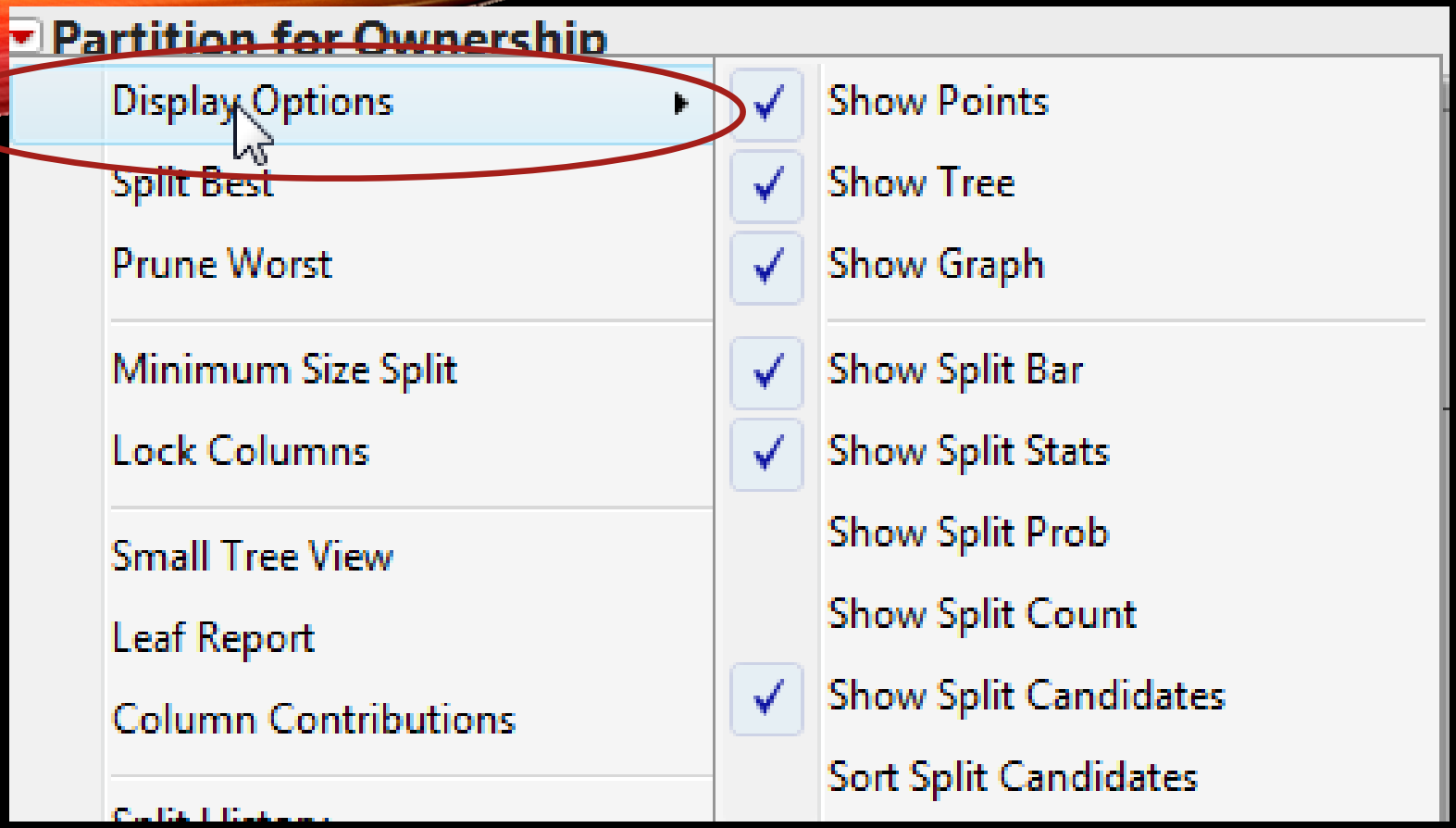
Gini index = most pure and is calculated as $2p_1p_2$ so that as either of the class probabilities is closer to zero the index is minimized; it is maximized if $p_1 = p_2$

Information Gain or gain(split) = info(prior to split) – info(after split)

Candidate G^2 = Likelihood ratio chi-square for the best split. Shown if the response is categorical.

LogWorth = The LogWorth statistic, defined as -log10(p-value). The optimal split is the one that maximizes the LogWorth.

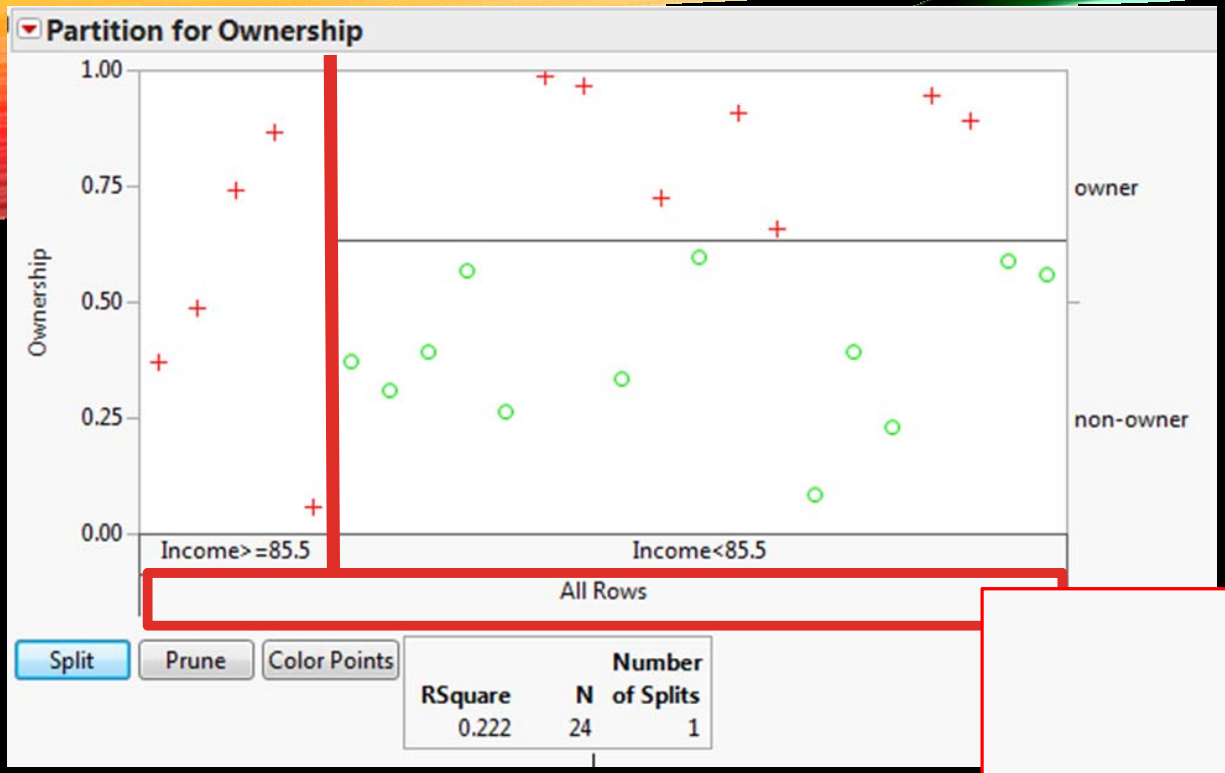
Want split with best information gain (higher candidate G^2)


Partition for Ownership
Display Options
Split Best
Prune Worst
Minimum Size Split
Lock Columns
Small Tree View
Leaf Report
Column Contributions
Show Points
Show Tree
Show Graph
Show Split Bar
Show Split Stats
Show Split Prob
Show Split Count
Show Split Candidates
Sort Split Candidates

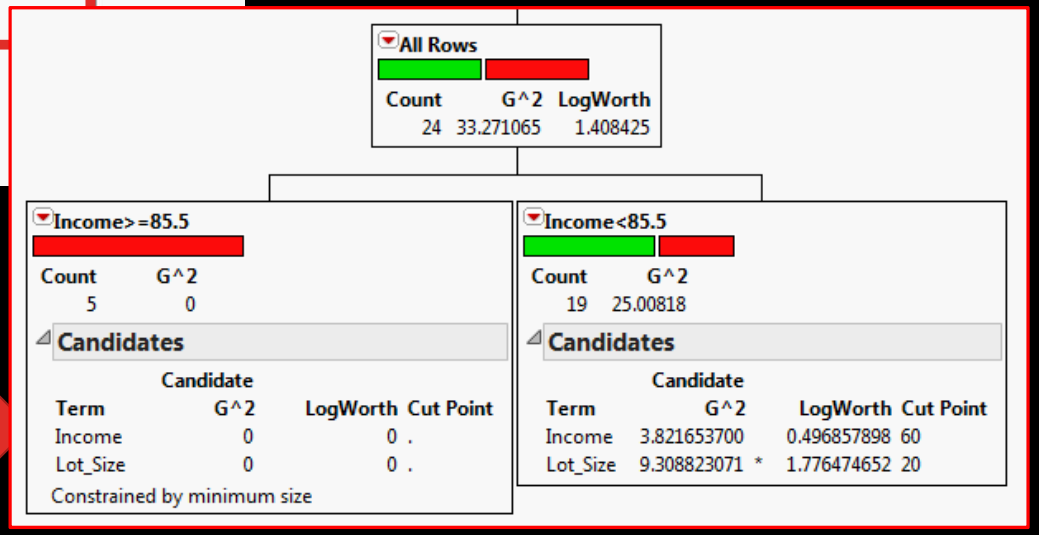
G^2									
n1	n2	p	1-p	G^2					
10	10	0.5	0.5	27.73	=2*(A3*(-LN(C3)) + B3*(-LN(D3)))				
10	5	0.667	0.333	19.1					
5	10	0.333	0.667	19.1					
1	10	0.091	0.909	6.702					

LogWorth			
p-value	LogWorth		
0.1	2.30259	=-LN(C11)	
0.05	2.99573		
0.01	4.60517		
0.005	5.29832		

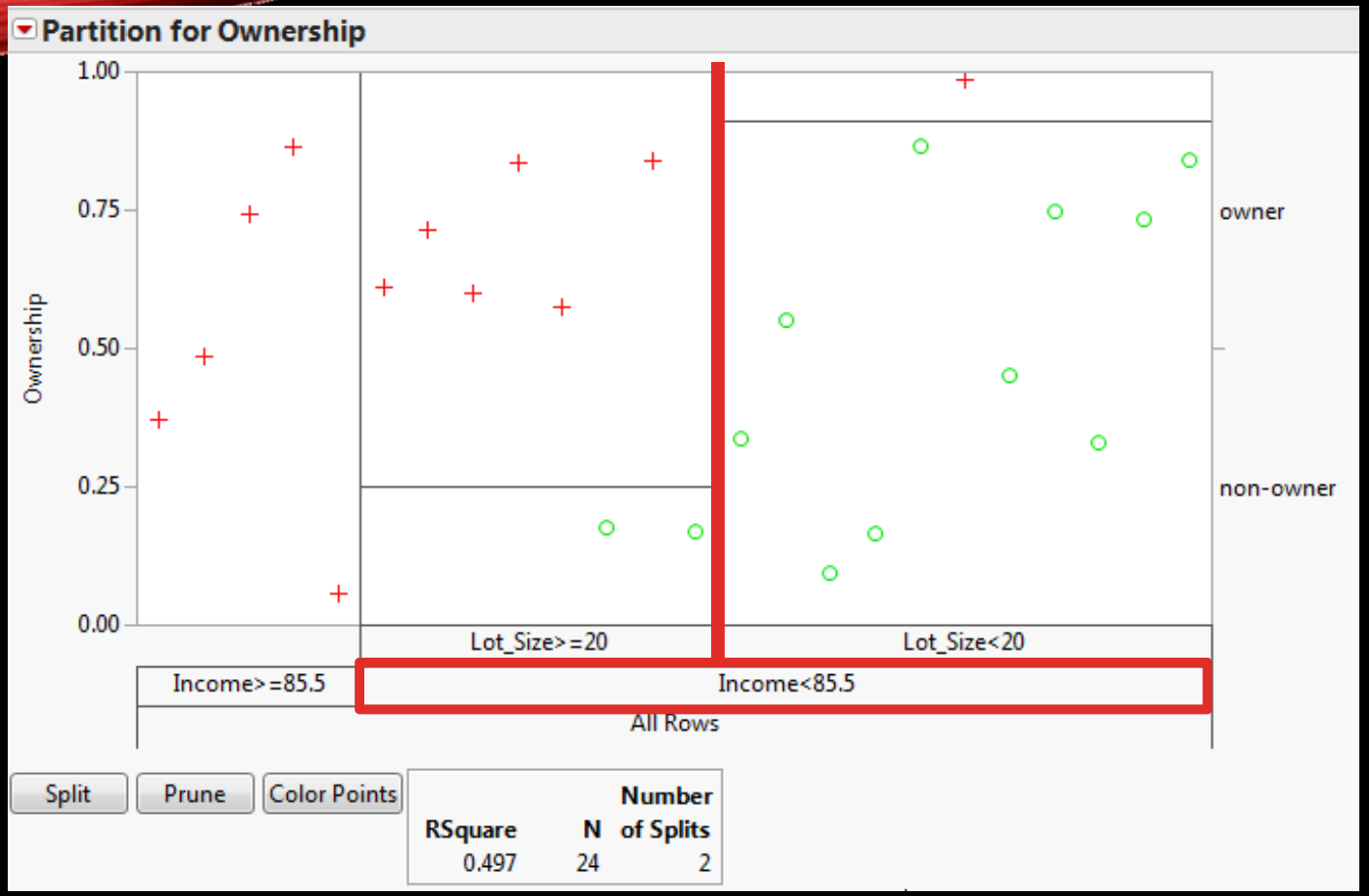
First Split



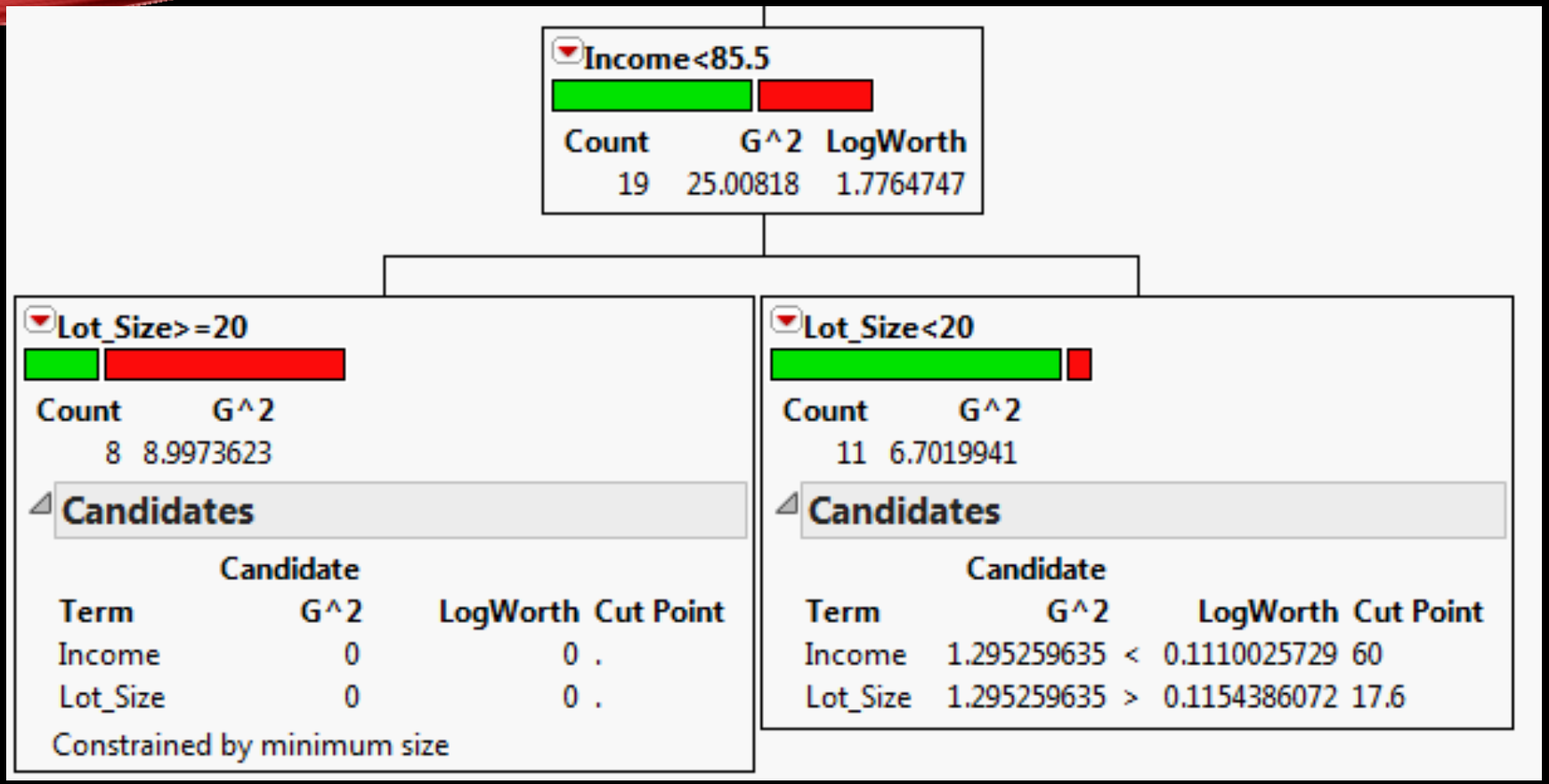
Second Split Candidates



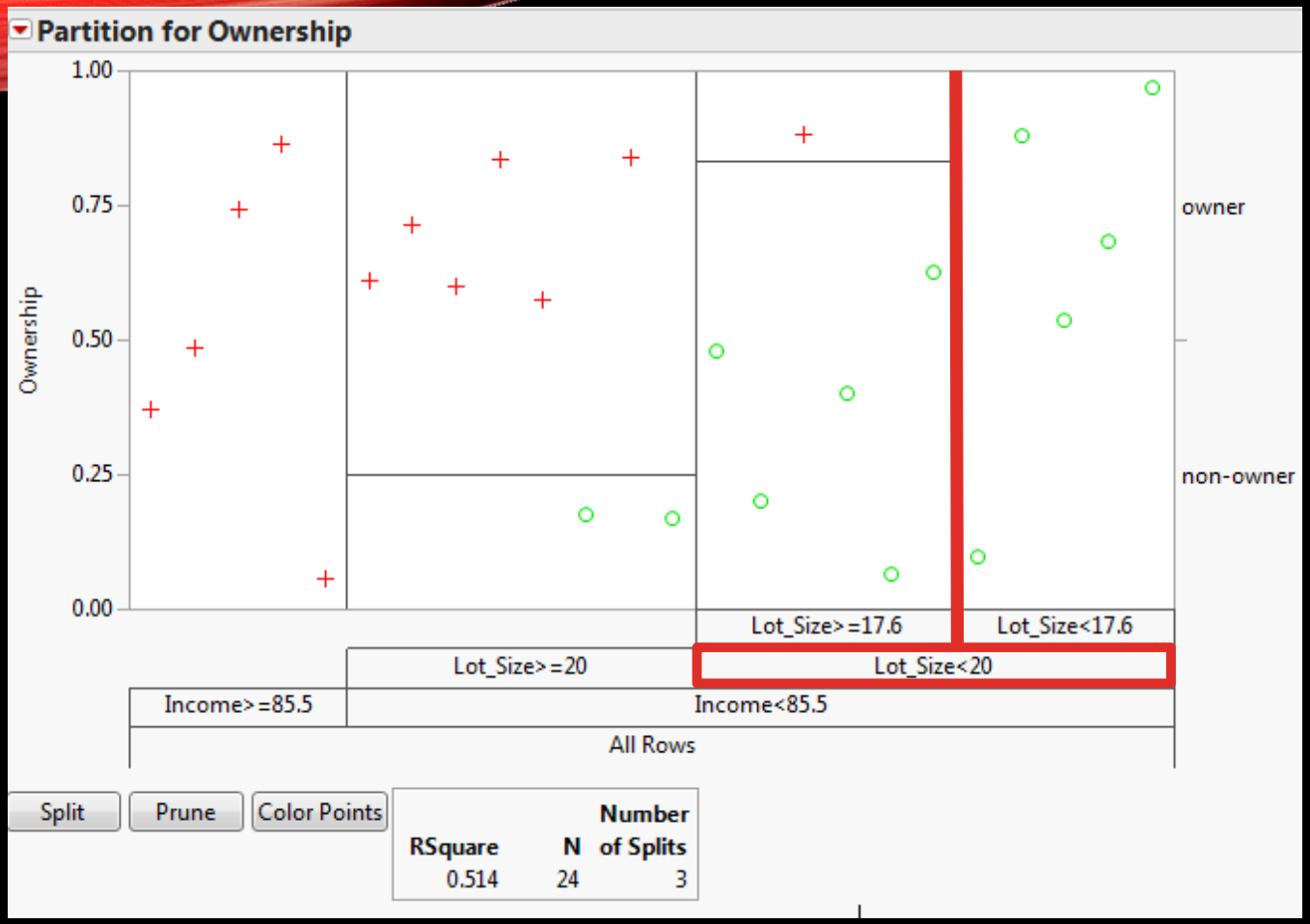
Second Split

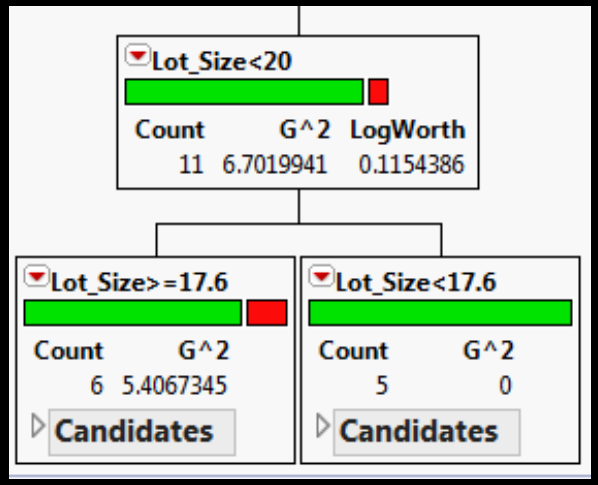


Second Split Candidates

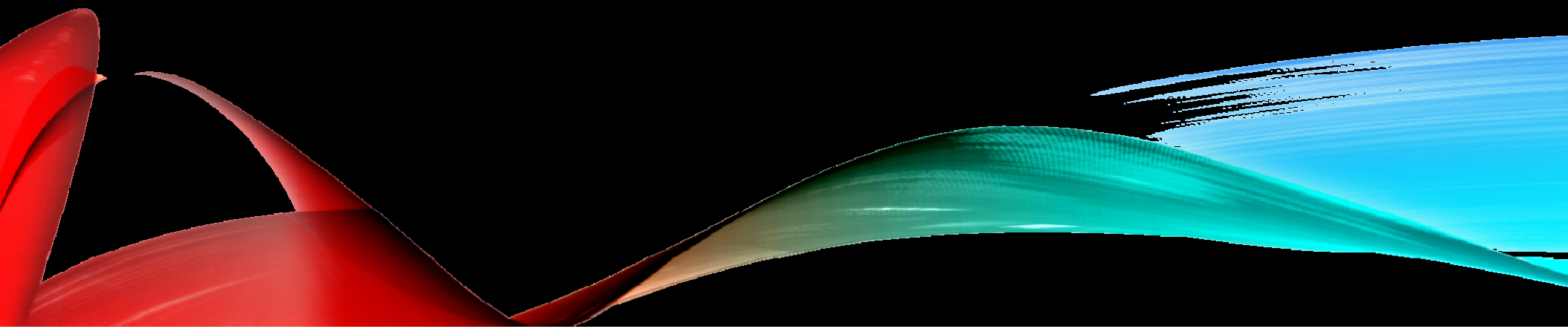


Third Split

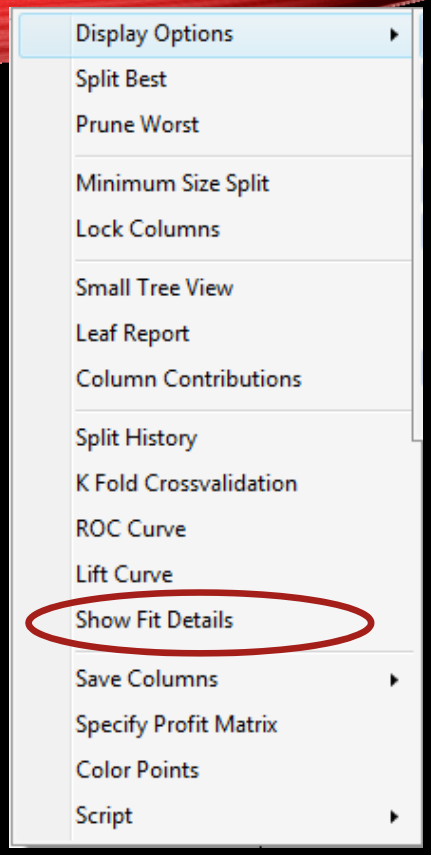

Partition for Ownership
1.00
0.75
0.50
0.25
0.00
Ownership
owner
non-owner
Lot_Size>=17.6
Lot_Size<17.6
Lot_Size>=20
Lot_Size<20
Income>=85.5
Income<85.5
All Rows
Split
Prune
Color Points
RSquare
N
Number of Splits
0.514
24
3


Lot_Size<20
Count
G^2
LogWorth
11
6.7019941
0.1154386
Lot_Size>=17.6
Count
G^2
6
5.4067345
Candidates
Lot_Size<17.6
Count
G^2
5
0
Candidates

HOW TO INTERPRET THE RESULTS



Misclassification Rate and Confusion Matrix



Fit Details

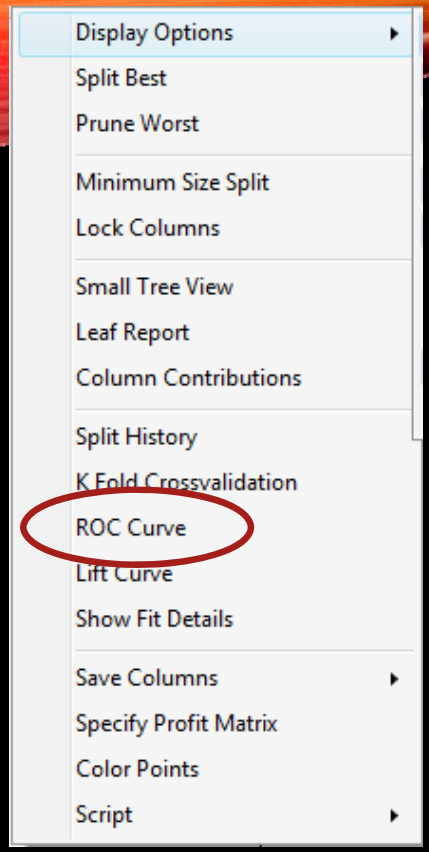
Measure	Training	Definition
Entropy RSquare	0.5145	1-Loglike(model)/Loglike(0)
Generalized RSquare	0.6799	(1-(L(0)/L(model))^(2/n))/(1-L(0)^(2/n))
Mean -Log p	0.3365	$\sum$ -Log(ρ[j])/n
RMSE	0.3171	$\sqrt{\sum(y[j]-\rho[j])^2/n}$
Mean Abs Dev	0.2391	$\sum \lvert y[j]-\rho[j]\rvert/n$
Misclassification Rate	0.1250	$\sum (\rho[j] \neq \rho Max)/n$
N	24	n

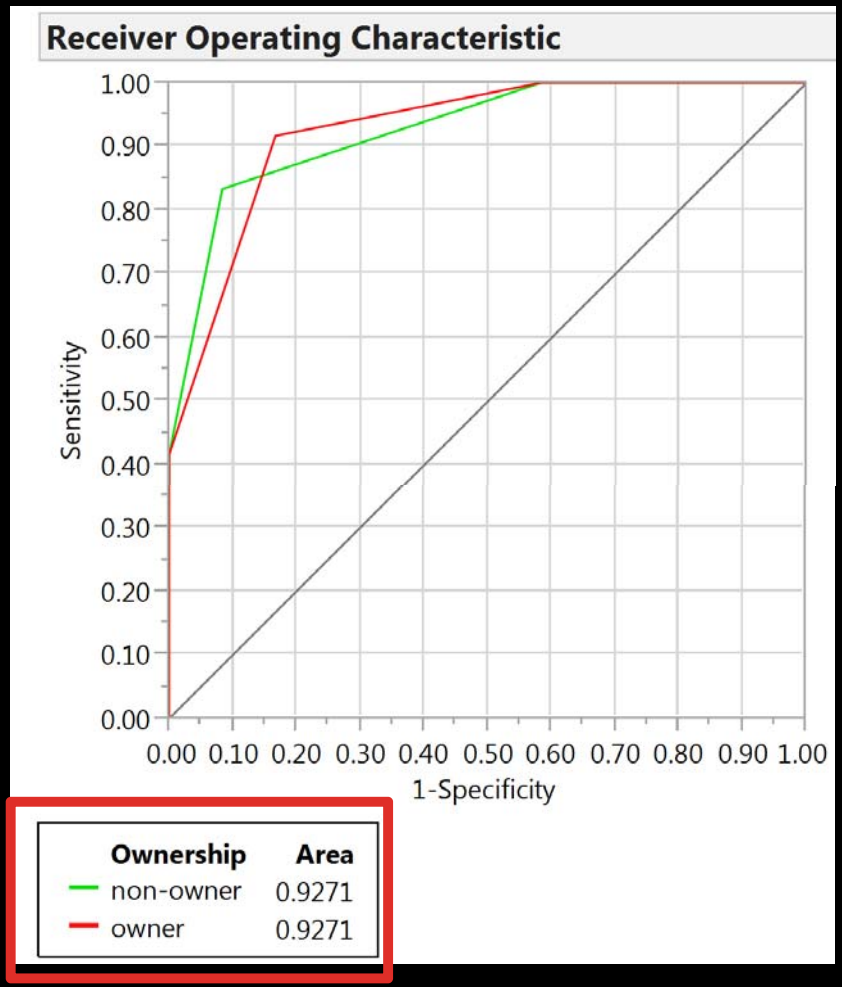
Confusion Matrix

Actual / Predicted

Training	non-owner	owner
non-owner	10	2
owner	1	11

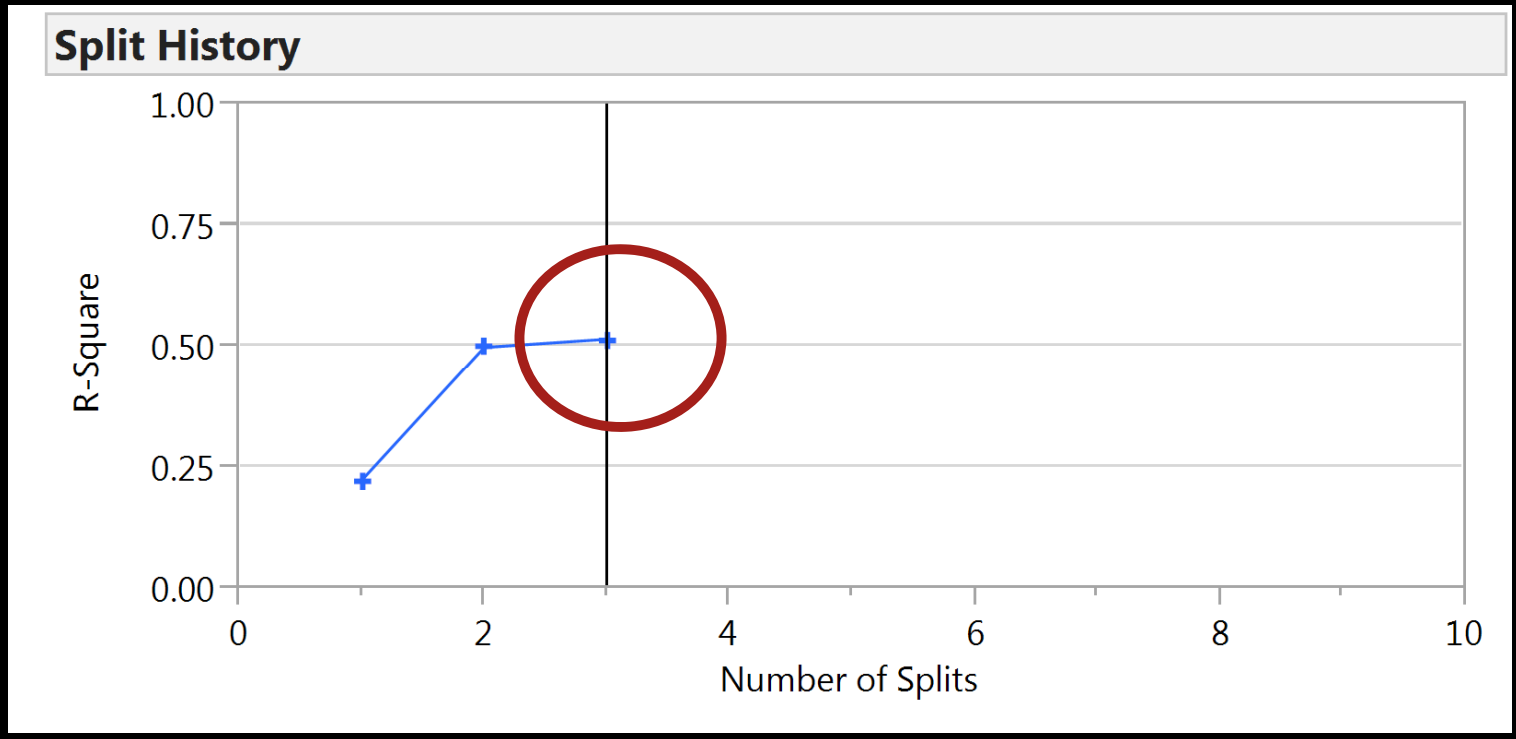
ROC CURVES


Display Options
Split Best
Prune Worst
Minimum Size Split
Lock Columns
Small Tree View
Leaf Report
Column Contributions
Split History
K Fold Crossvalidation
ROC Curve
Lift Curve
Show Fit Details
Save Columns
Specify Profit Matrix
Color Points
Script

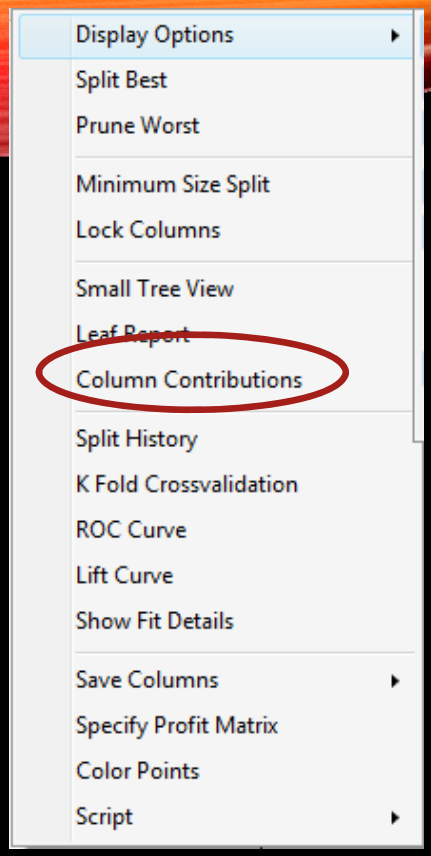

Receiver Operating Characteristic
Sensitivity
1.00
0.90
0.80
0.70
0.60
0.50
0.40
0.30
0.20
0.10
0.00
0.00 0.10 0.20 0.30 0.40 0.50 0.60 0.70 0.80 0.90 1.00
1-Specificity
Ownership
Area
non-owner
0.9271
owner
0.9271

SPLIT HISTORY

- Display Options
- Split Best
- Prune Worst
- Minimum Size Split
- Lock Columns
- Small Tree View
- Leaf Report
- Column Contributions
- Split History
- K Fold Crossvalidation
- ROC Curve
- Lift Curve
- Show Fit Details
- Save Columns
- Specify Profit Matrix
- Color Points
- Script



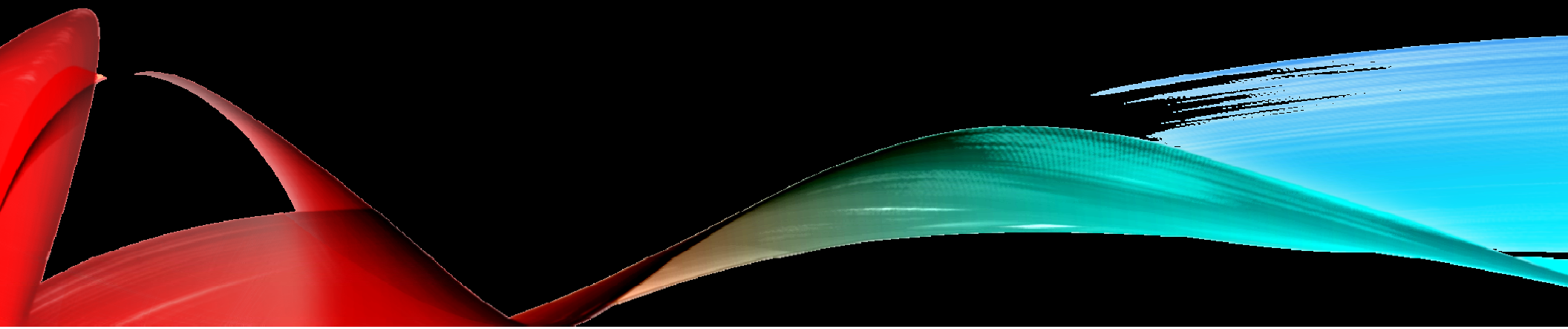
COLUMN CONTRIBUTION

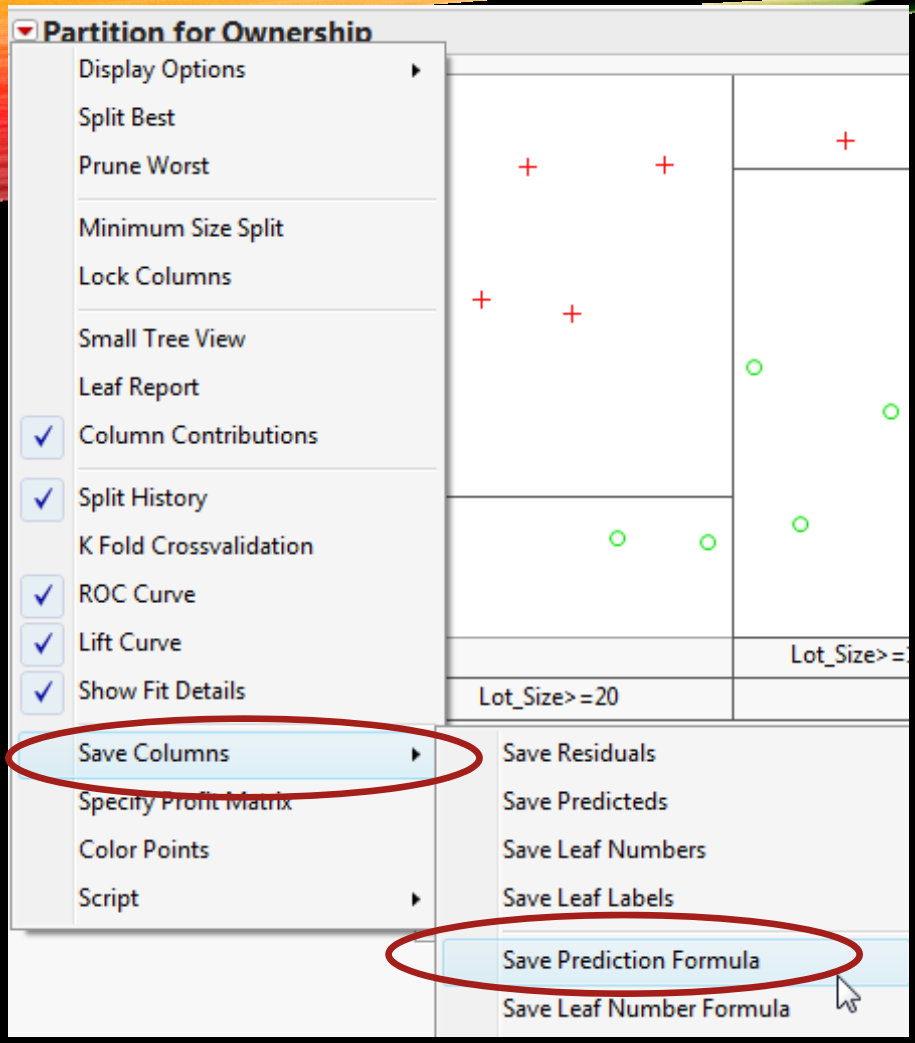


Column Contributions

Term	Number of Splits	G^2	Portion
Lot_Size	2	10.6040827	0.5620
Income	1	8.26288514	0.4380

HOW TO USE IT





Red Triangle > Save Columns > Save Prediction Formula

Prob(Ownership==non-owner)	Prob(Ownership==owner)	Most Likely Ownership
0.7926934524	0.2073065476	non-owner
0.0833333333	0.9166666667	owner
0.2791666667	0.7208333333	owner

Prob(Ownership==non-owner)

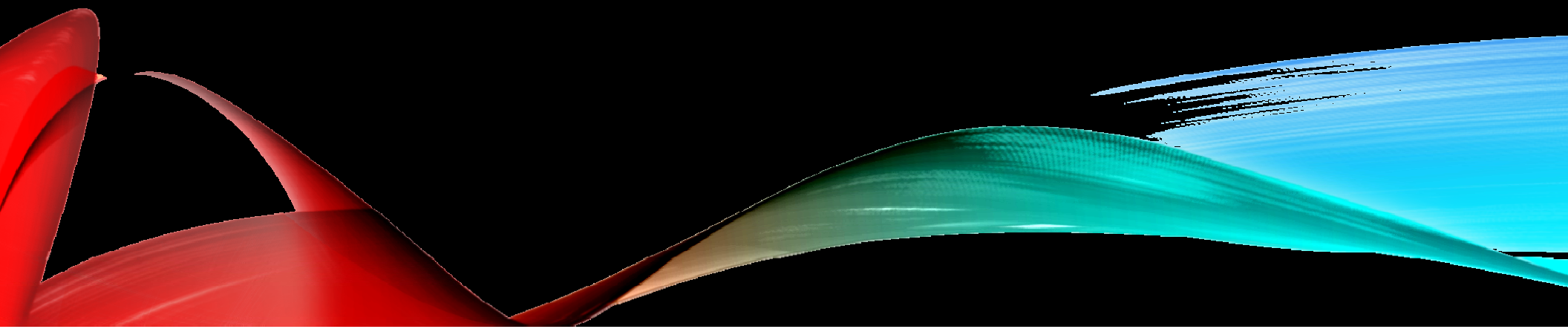
```
If  Income >= 85.5          ⇒ 0.08333333333333
    else                    ⇒ If  Lot_Size >= 20    ⇒ 0.27916666666667
                                  else              ⇒ If  Lot_Size >= 17.6  ⇒ 0.79269345238095
                                                        else              ⇒ 0.92480902777778
```

Prob(Ownership==owner)

```
If  Income >= 85.5          ⇒ 0.91666666666667
    else                    ⇒ If  Lot_Size >= 20    ⇒ 0.72083333333333
                                  else              ⇒ If  Lot_Size >= 17.6  ⇒ 0.20730654761905
                                                        else              ⇒ 0.07519097222222
```

```
IfMax  Prob(Ownership==non-owner)  ⇒ "non-owner"
       Prob(Ownership==owner)      ⇒ "owner"
       else                        ⇒ ""
```

HOW TO UNDERSTAND MANAGEMENT IMPLICATIONS



Relative Influence

Column Contributions

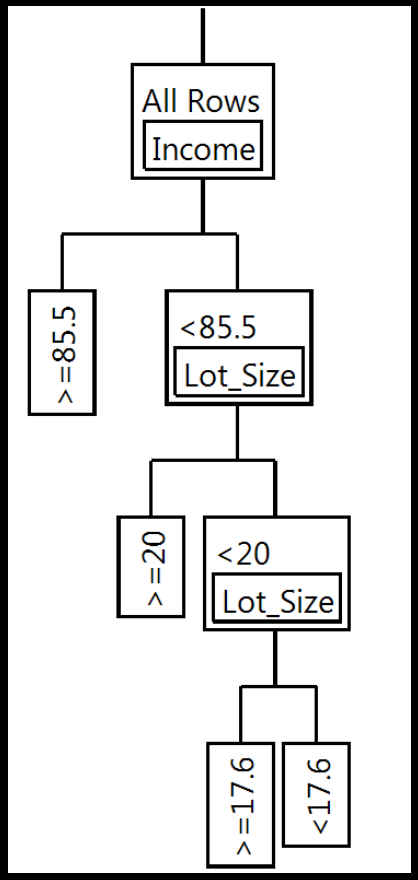
Term	Number of Splits	G^2	Portion
Lot_Size	2	10.6040827	0.5620
Income	1	8.26288514	0.4380

Prediction and Classification

Prob(Ownership==owner) If-Then Statements

```
If( Income >= 85.5 ⇒ 0.91666666666667,
    else ⇒ If( Lot_Size >= 20 ⇒ 0.72083333333333,
               else ⇒ If( Lot_Size >= 17.6 ⇒ 0.20730654761905,
                          else ⇒ 0.07519097222222 ) ) )
```

Tree Structure



Misclassification Cell Analysis

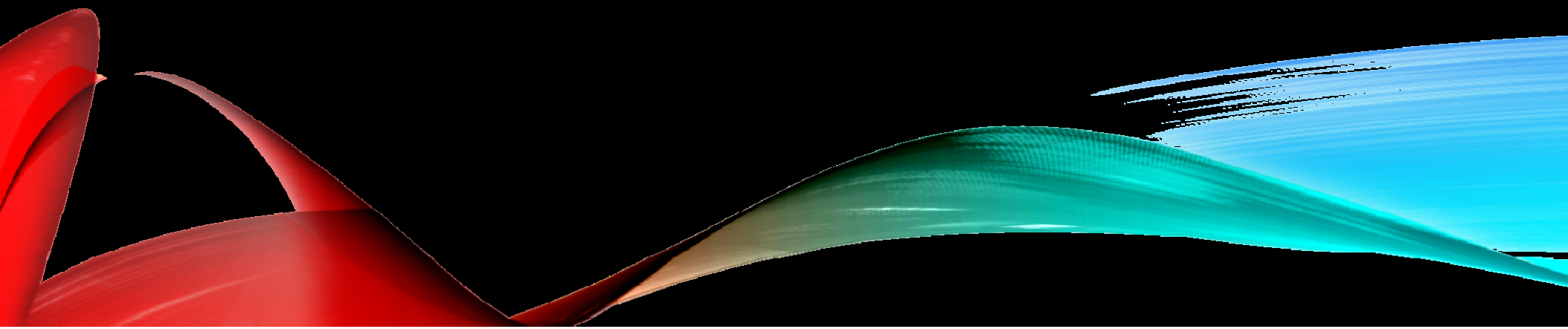
Confusion Matrix

Actual / Predicted

Training	non-owner	owner
non-owner	10	2
owner	1	11

PREDICTIVE MODELING EXAMPLE

Using a validation column



Default *Credit Card Default Data*

References: Games, G., Witten, D., Hastie, T., and Tibshirani, R. (2013) An Introduction to Statistical Learning with applications in R, www.StatLearning.com, Springer-Verlag, New York

Description

A simulated data set containing information on ten thousand customers. The aim here is to predict which customers will default on their credit card debt.

Usage

```
Default
```

Format

A data frame with 10000 observations on the following 4 variables.

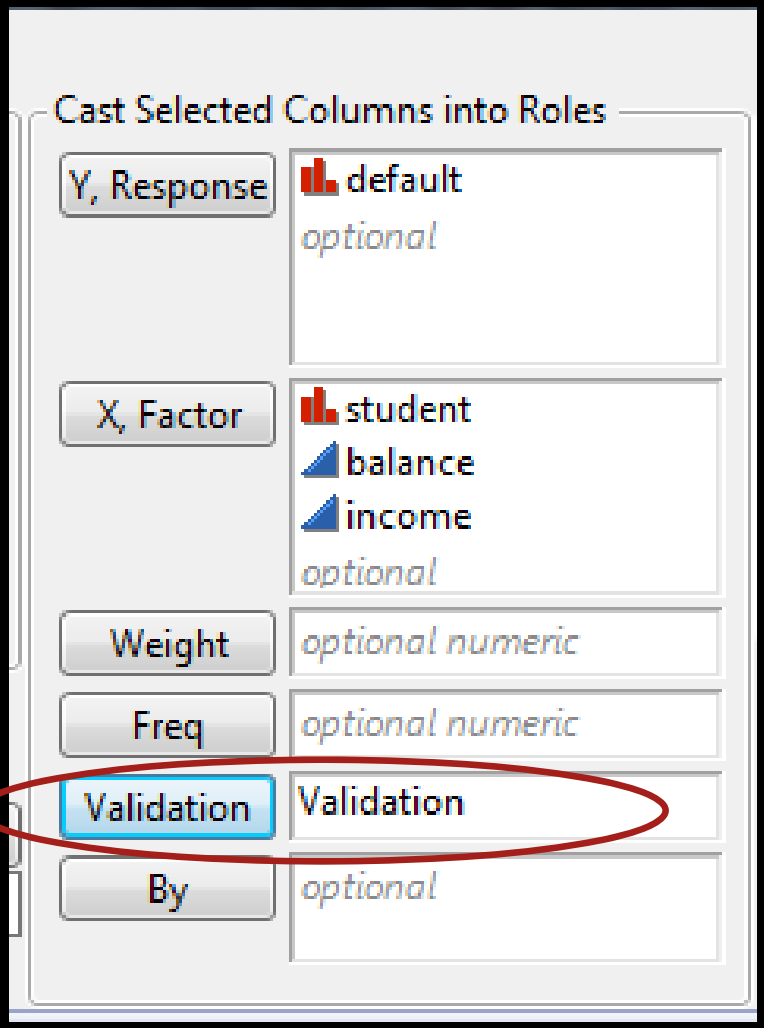
`default` A factor with levels `No` and `Yes` indicating whether the customer defaulted on their debt

`student` A factor with levels `No` and `Yes` indicating whether the customer is a student

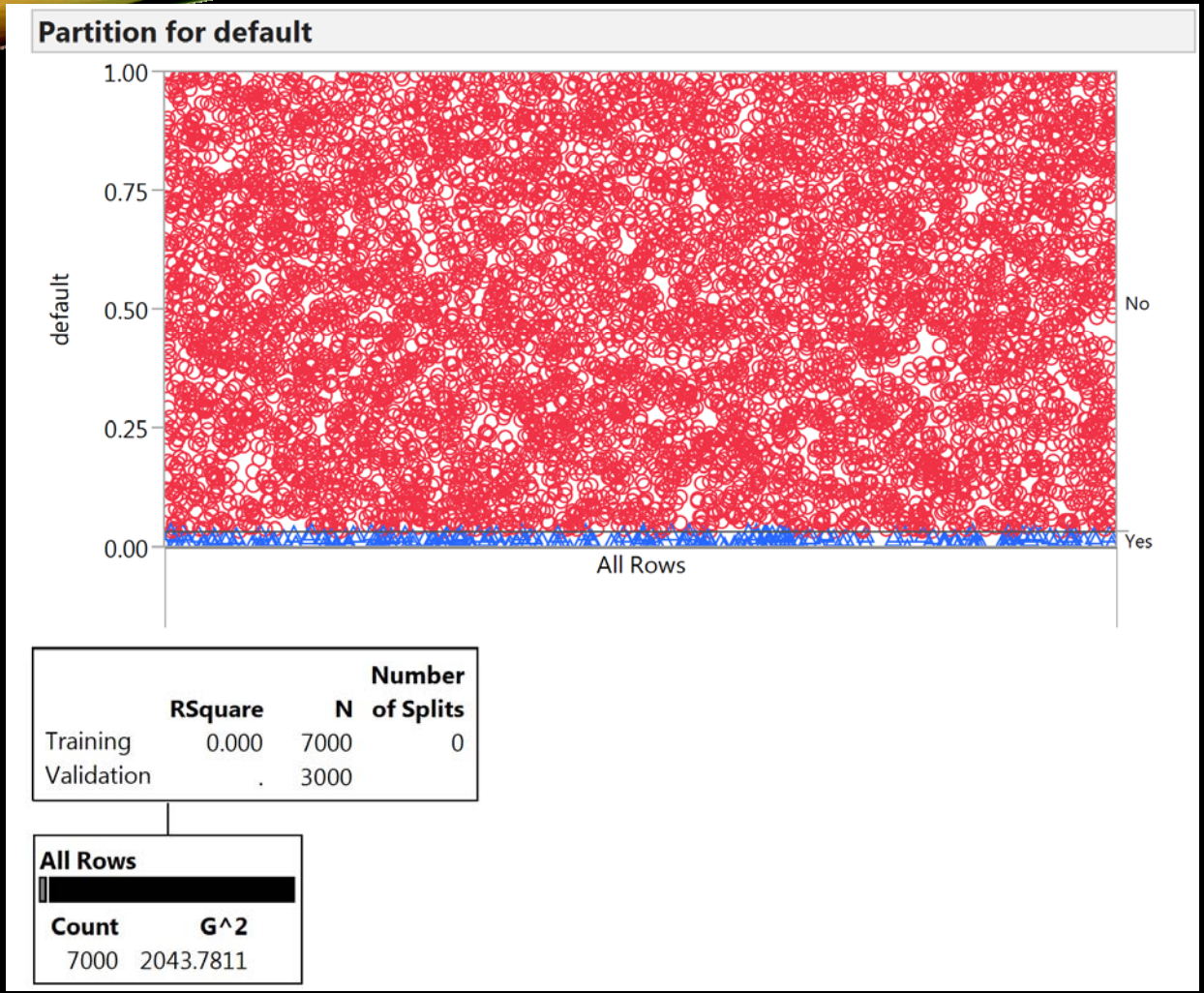
`balance` The average balance that the customer has remaining on their credit card after making their monthly payment

`income` Income of customer

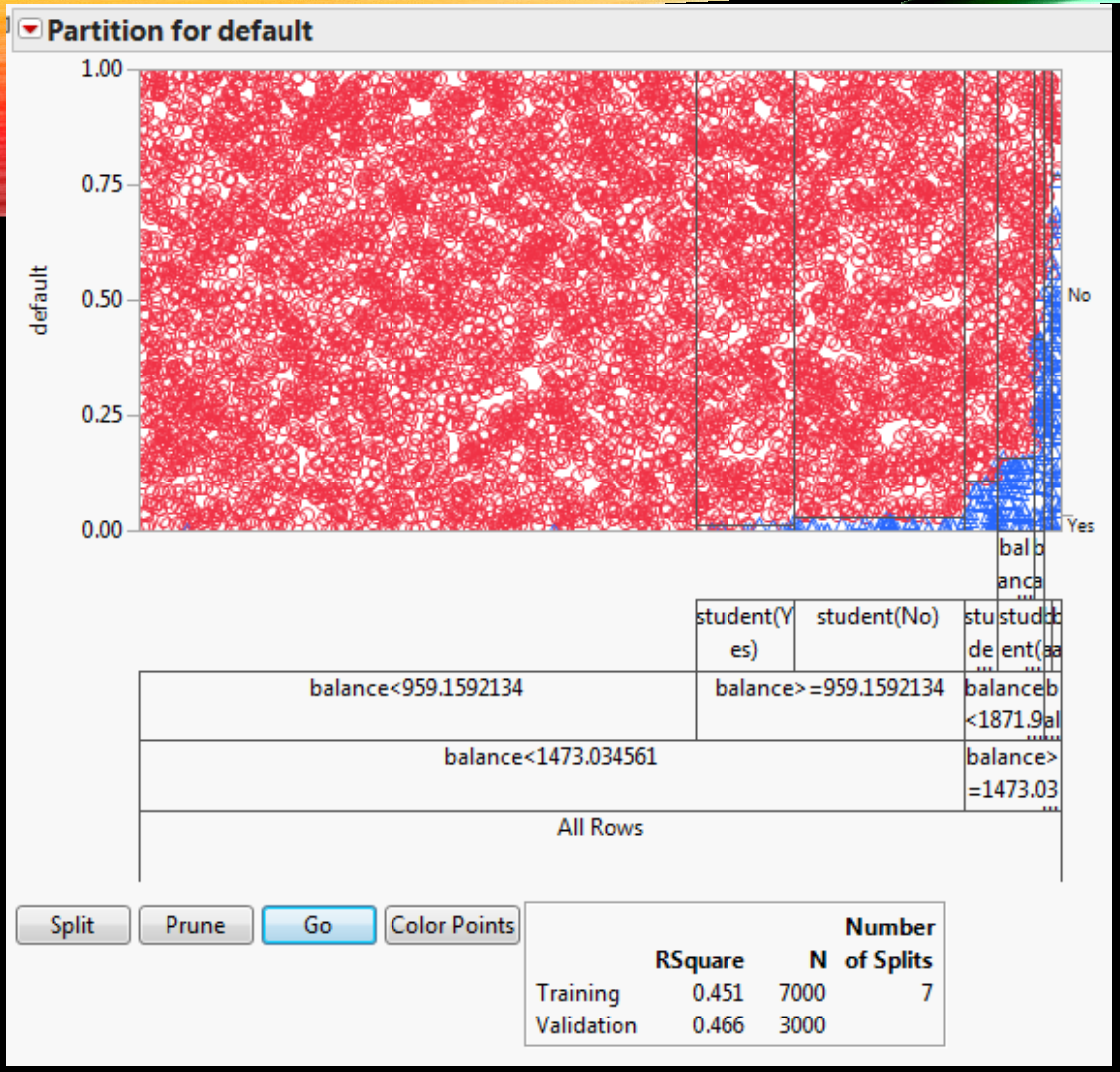
Create a validation variable associate it with the validation role in the dialog box

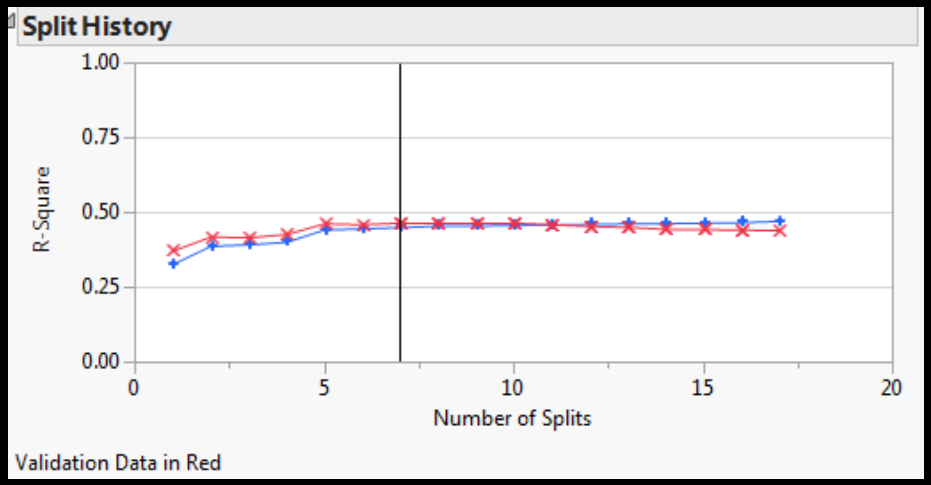


Because we have Validation, we either select “Go” for automatic selection or repeatedly select “Split” and then “prune” by examining split history

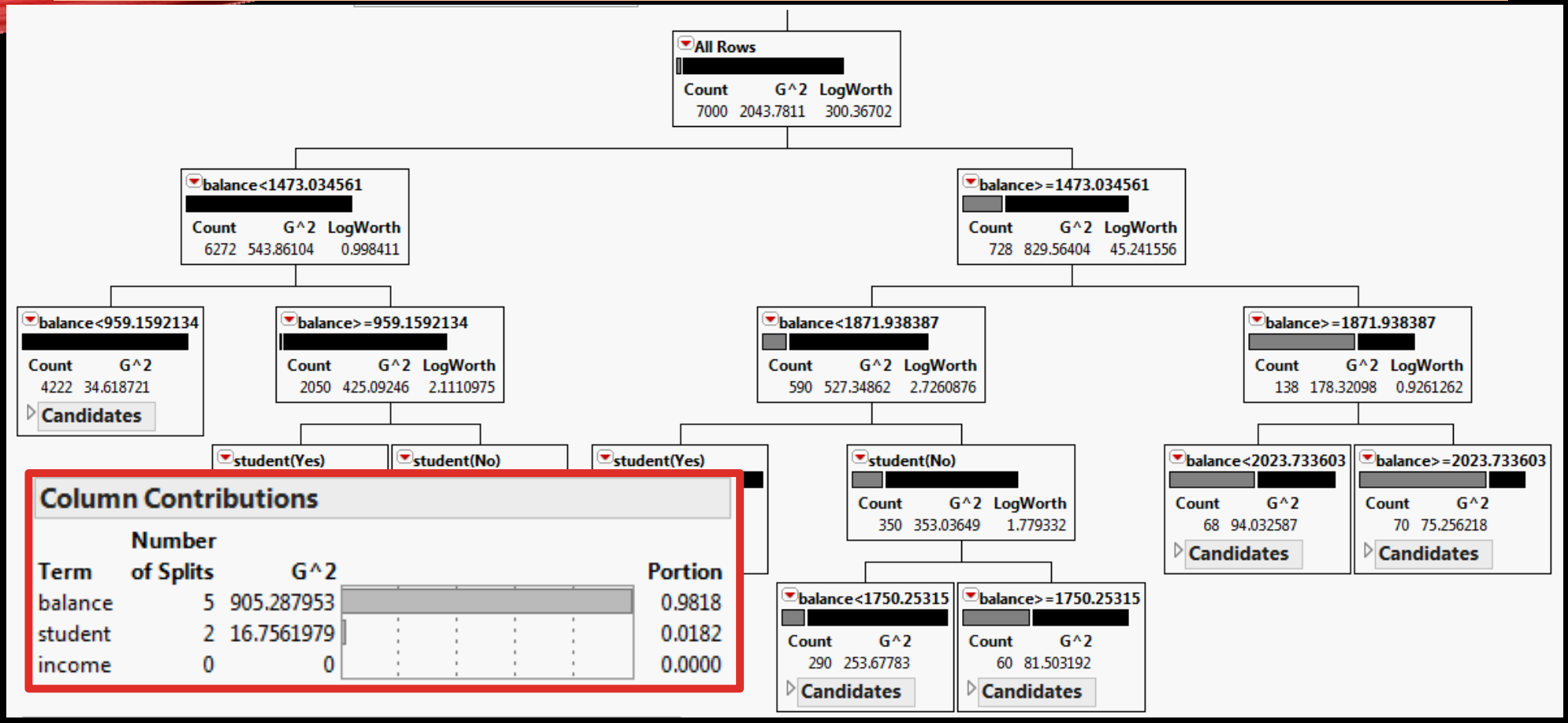


The automatic "pruning" process





Looking at the entire tree we can see that "balance" is the most significant column contributor



Fit Details

Measure	Training	Validation	Definition
Entropy RSquare	0.4511	0.4659	1-Loglike(model)/Loglike(0)
Generalized RSquare	0.4874	0.5023	(1-(L(0)/L(model))^(2/n))/(1-L(0)^(2/n))
Mean -Log p	0.0801	0.0781	$\sum$ -Log(p[j])/n
RMSE	0.1467	0.1449	$\sqrt{\sum(y[j]-p[j])^2/n}$
Mean Abs Dev	0.0431	0.0405	$\sum \lvert y[j]-p[j]\rvert/n$
Misclassification Rate	0.0273	0.0273	$\sum$ (p[j]≠pMax)/n
N	7000	3000	n

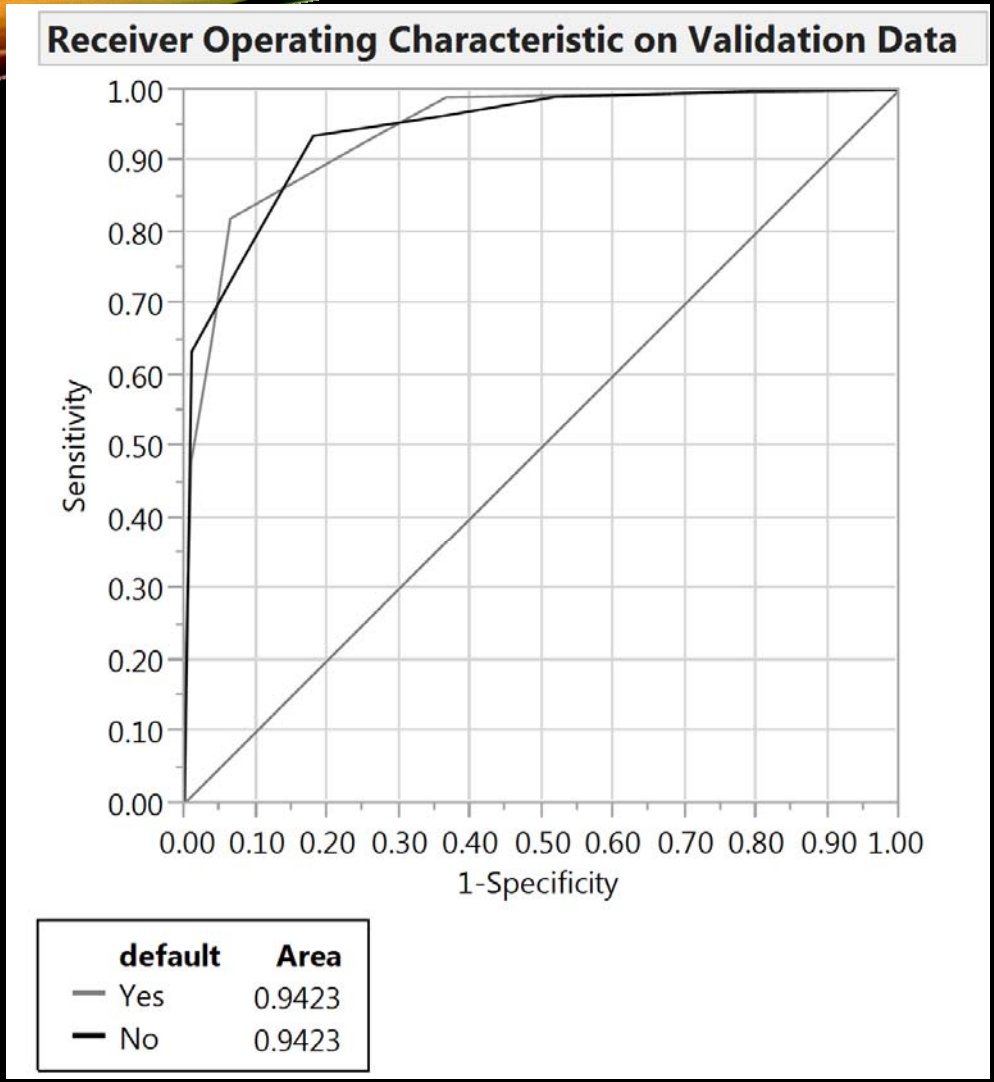
Confusion Matrix

Actual / Predicted

Training	Yes	No
Yes	90	143
No	48	6719

Actual / Predicted

Validation	Yes	No
Yes	39	61
No	21	2879

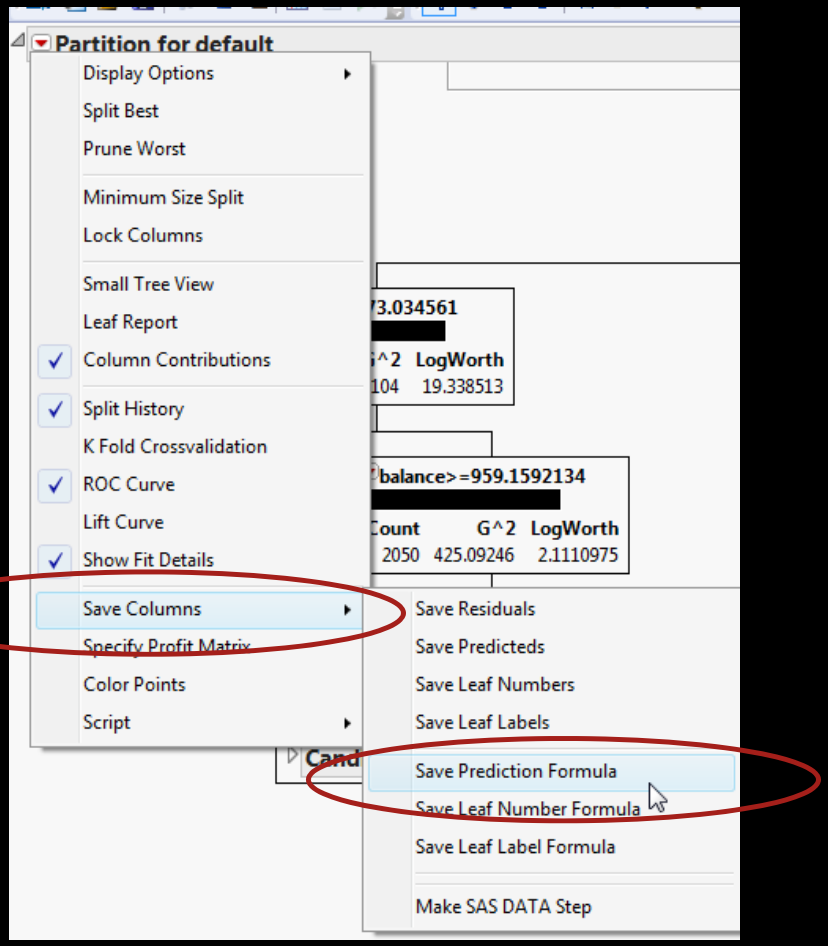

Receiver Operating Characteristic on Validation Data
Sensitivity
1.00
0.90
0.80
0.70
0.60
0.50
0.40
0.30
0.20
0.10
0.00
0.00 0.10 0.20 0.30 0.40 0.50 0.60 0.70 0.80 0.90 1.00
1-Specificity
default Area
Yes 0.9423
No 0.9423

Prob(default==Yes)

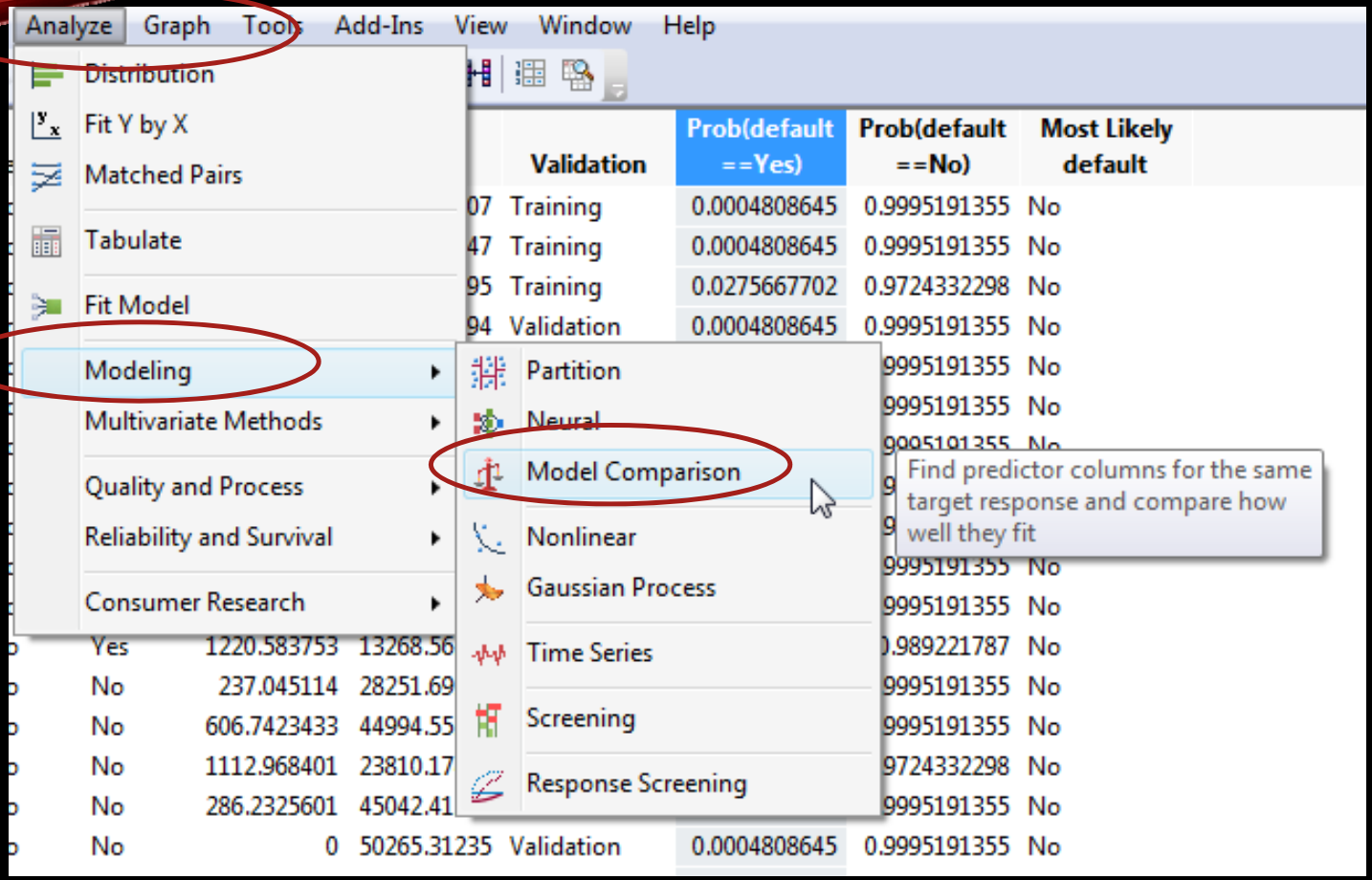
```
If
  balance < 1473.034561 ⇒ If
    balance < 959.1592134 ⇒ 0.00048086454533
    else ⇒ If
      student == "Yes" ⇒ 0.01077821297826
      student == "No" ⇒ 0.02756677021332
      else ⇒ 0.02146791368841
  else ⇒ If
    balance < 1871.938387 ⇒ If
      student == "Yes" ⇒ 0.10815964409653
      student == "No" ⇒ If
        balance < 1750.25315 ⇒ 0.15835076860238
        else ⇒ 0.41114874857855
      else ⇒ 0.16422269591591
    else ⇒ If
      balance < 2023.733603 ⇒ 0.52340348070535
      else ⇒ 0.76218084744605
```

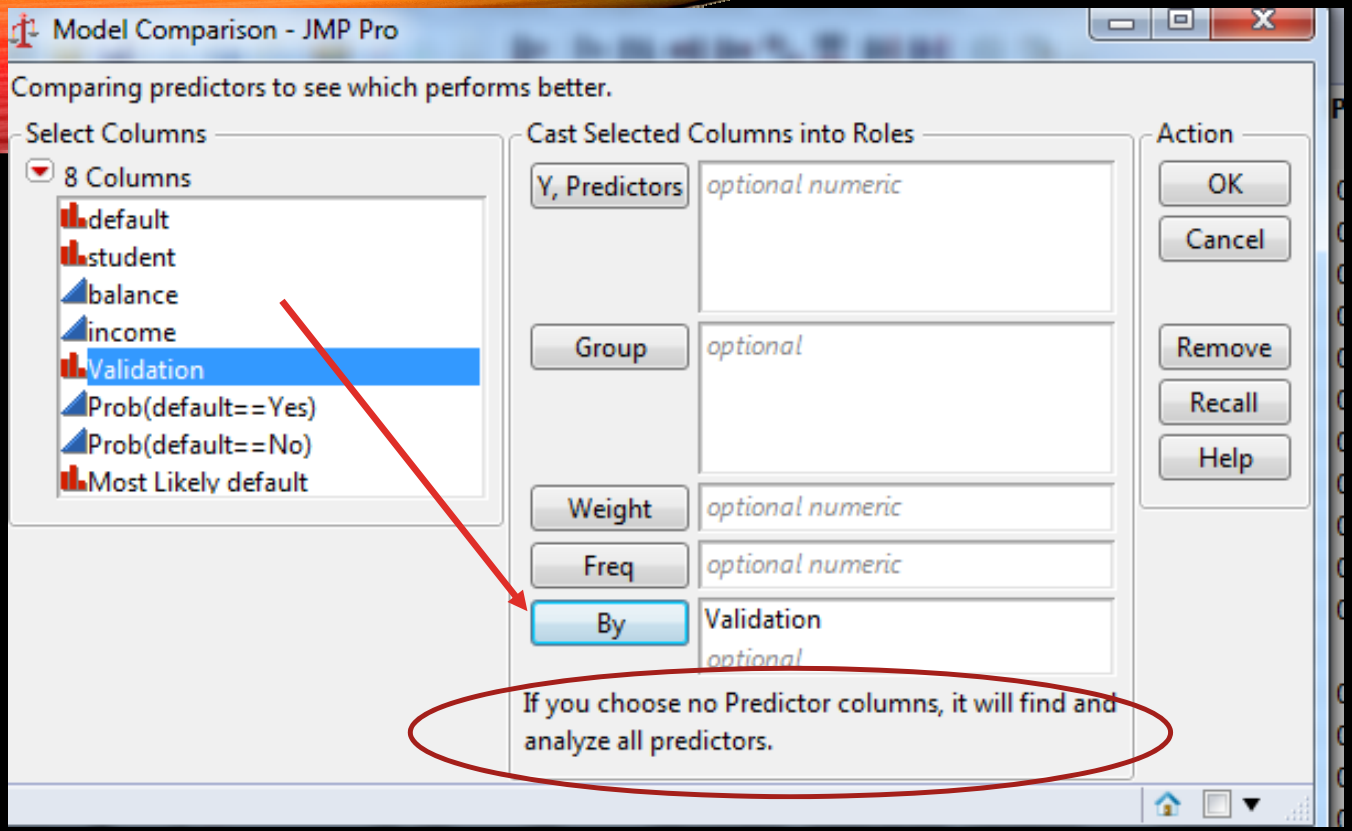
If we use the **Model Comparison** feature we can access the profiler and influential variables

First: Save Prediction Formula: Save Columns > Save Prediction Formula



Analyze > Modeling > Model Comparison




Model Comparison - JMP Pro
Comparing predictors to see which performs better.
Select Columns
8 Columns
default
student
balance
income
Validation
Prob(default==Yes)
Prob(default==No)
Most Likely default
Cast Selected Columns into Roles
Y, Predictors
optional numeric
Group
optional
Weight
optional numeric
Freq
optional numeric
By
Validation
optional
If you choose no Predictor columns, it will find and analyze all predictors.
Action
OK
Cancel
Remove
Recall
Help

Model Comparison Validation=Training

Model Comparison Validation=Validation

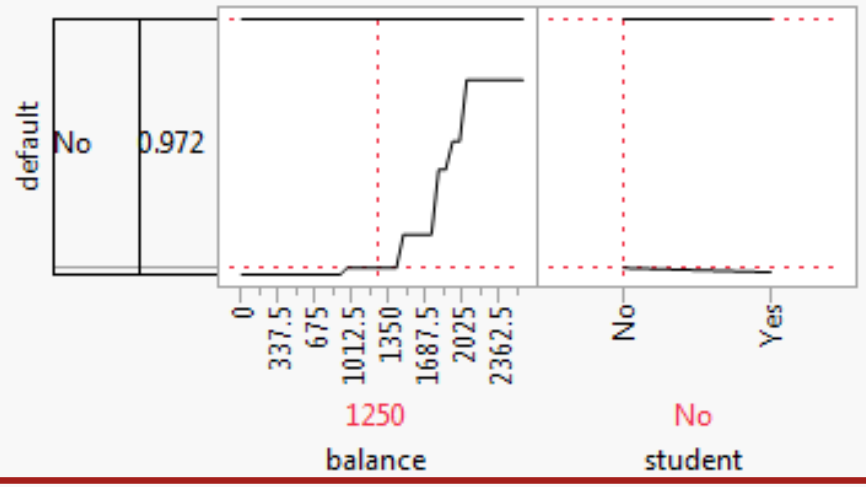
Predictors

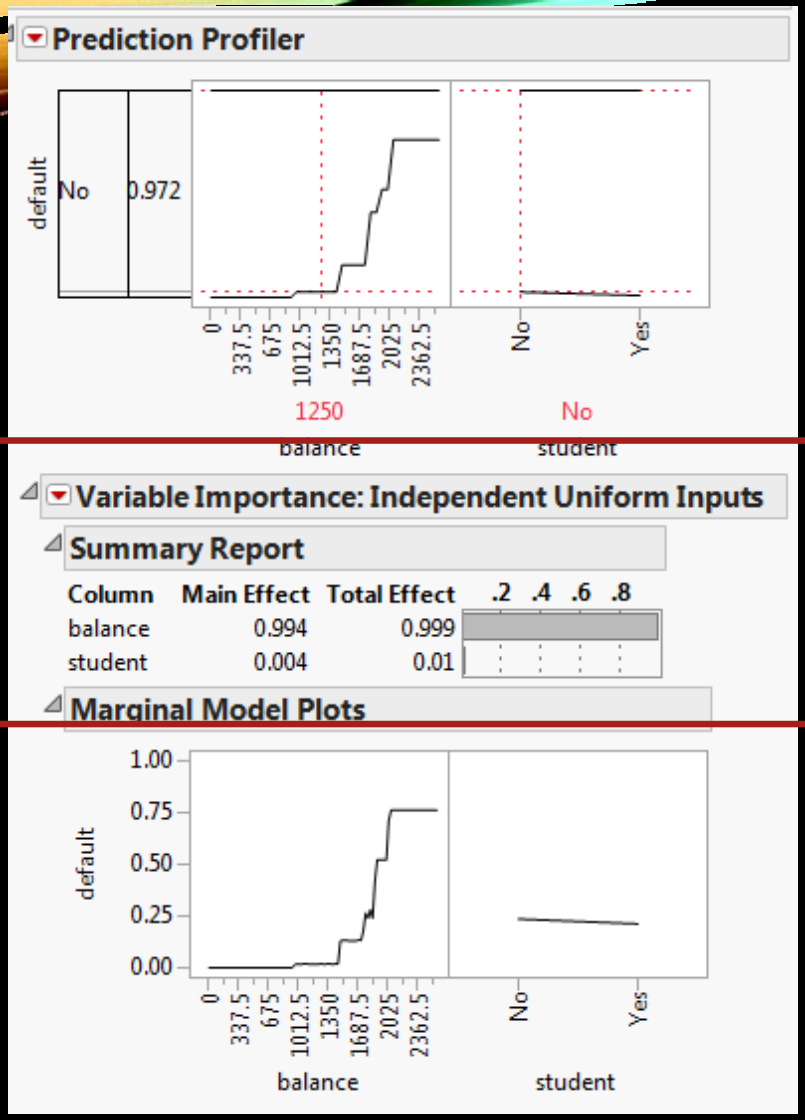
Measures of Fit for default

Creator	.2 .4 .6 .8	Entropy RSquare	Generalized RSquare	Mean -Log p	RMSE	Mean Abs Dev	Misclassification Rate	N
Partition		0.4659	0.5023	0.0781	0.1449	0.0405	0.0273	3000

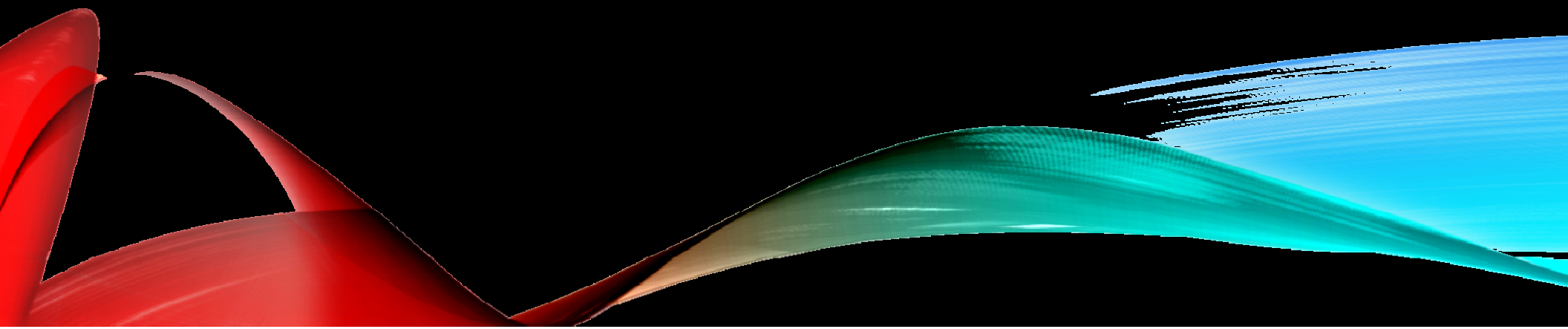
Profiler Validation=Validation

Prediction Profiler




Prediction Profiler
default
No
0.972
0
337.5
675
1012.5
1350
1687.5
2025
2362.5
No
Yes
1250
No
balance
student
Variable Importance: Independent Uniform Inputs
Summary Report
Column
Main Effect
Total Effect
.2
.4
.6
.8
balance
0.994
0.999
student
0.004
0.01
Marginal Model Plots
1.00
0.75
0.50
0.25
0.00
default
0
337.5
675
1012.5
1350
1687.5
2025
2362.5
No
Yes
balance
student

DECISION TREES: REGRESSION TREE



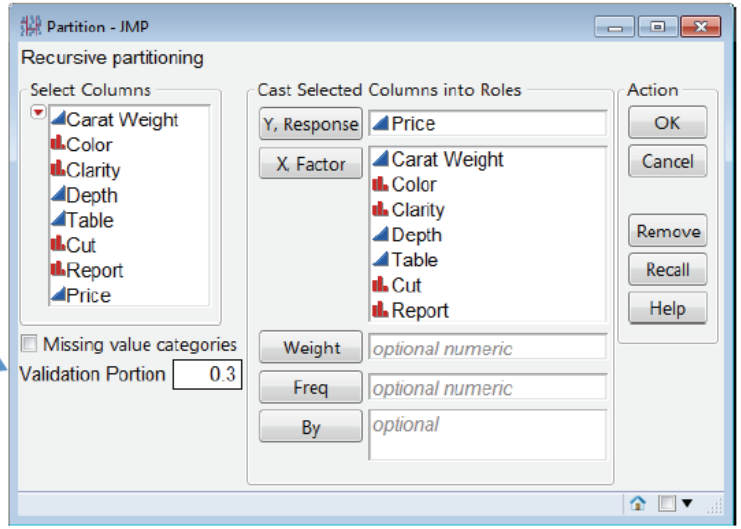
Regression Trees (Partition)

Use this data mining technique to predict a numeric (continuous) response as a function of potential predictor variables using recursive partitioning.

Regression Trees

1. From an open table, select **Analyze > Modeling > Partition.**
2. Select a continuous response variable from **Select Columns** and click **Y, Response**.
3. Select explanatory variables and click **X, Factor**.
4. If desired, enter the **Validation Portion** (a proportion, as shown) or select a validation column and click **Validation** (**JMP® Pro only**).
5. In **JMP Pro only**, select the tree **Method**: **Decision Tree** (Default in JMP, shown), **Bootstrap Forest** or **Boosted Tree**.

Example: Diamonds Data.jmp (Help > Sample Data)



Regression Trees (Partition)

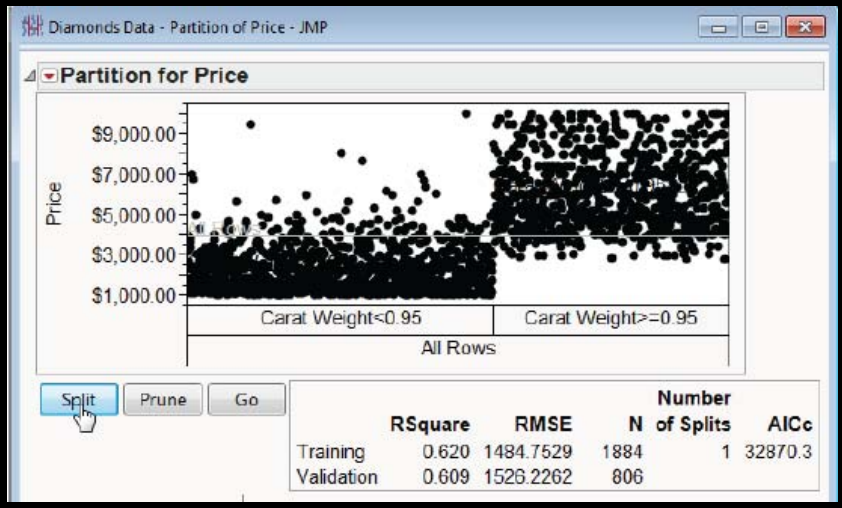
6. Click **OK.** JMP displays:

   - A graph with lines drawn at the overall mean response value.
   - Statistics for the training and validation set(s). Note that results will vary if Validation Portion is used.
   - A summary of **All Rows**. Click on the **gray triangle** next to **Candidates** to view split statistics for each column.

Regression Trees (Partition)

7. Click the **Split** button. The original observations will be split into two nodes, or leaves (as shown).

Note: In the graph, horizontal lines are drawn at the mean response within each leaf and vertical lines depict the leaf's relative size.



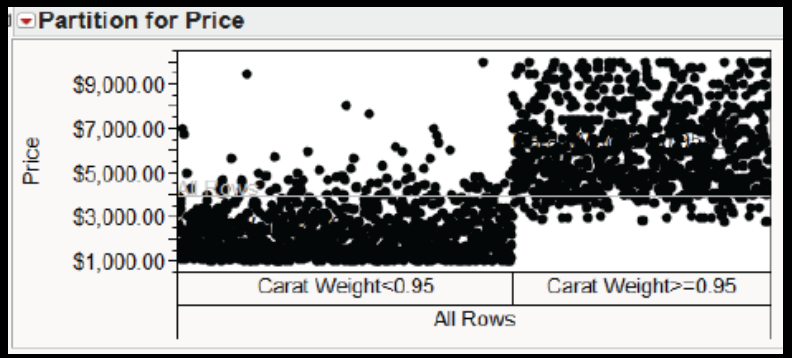
Note: In the graph, horizontal lines are drawn at the mean response within each leaf and vertical lines depict the leaf's relative size.

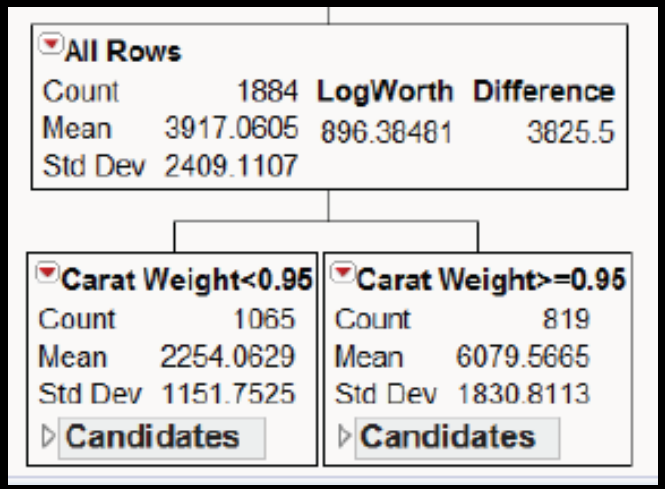
Regression Trees (Partition)

Note: In the graph, horizontal lines are drawn at the mean response within each leaf and vertical lines depict the leaf's relative size.

Interpretation (the response, in this example, is Diamond Price in $):

- In the left leaf there are 1,080 rows with Carat Weight < 0.95. The mean cost of these diamonds is $2,249.93.
- In the right leaf there are 825 rows with Carat Weight >= 0.95. The mean cost of these diamonds is $6,110.16.





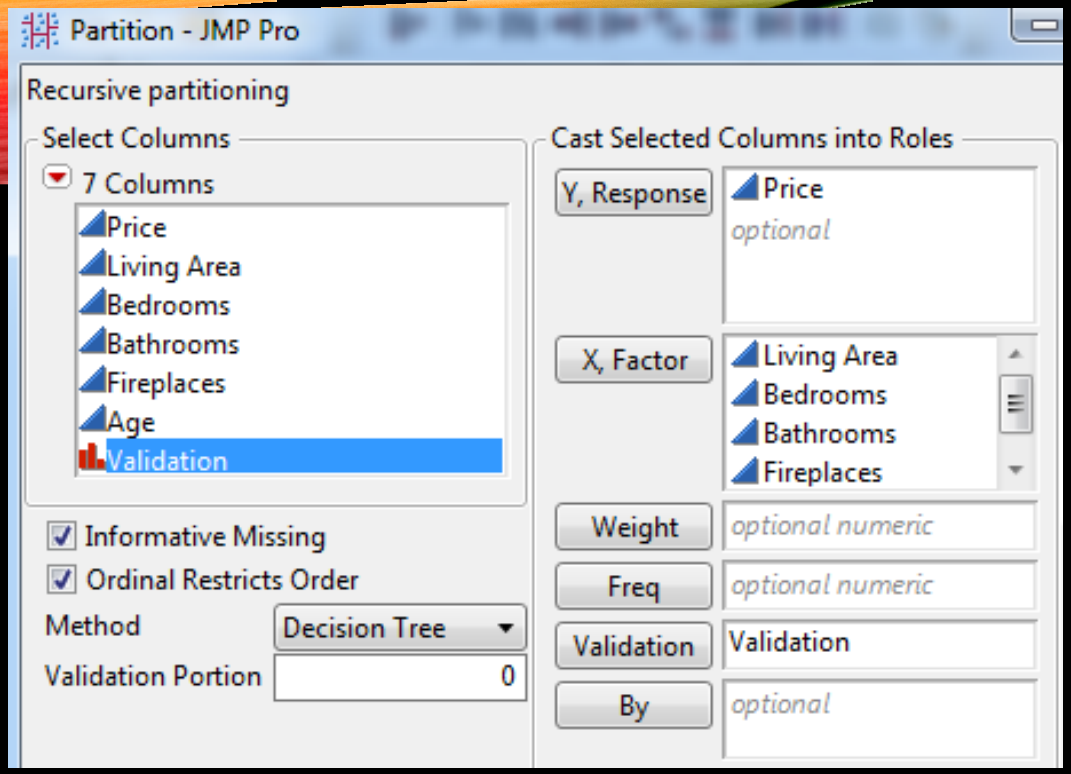
8. Click **Split** to make an additional split. Click **Prune** to remove a split. If a validation portion or validation column are used, click **Go** to perform automatic splitting.

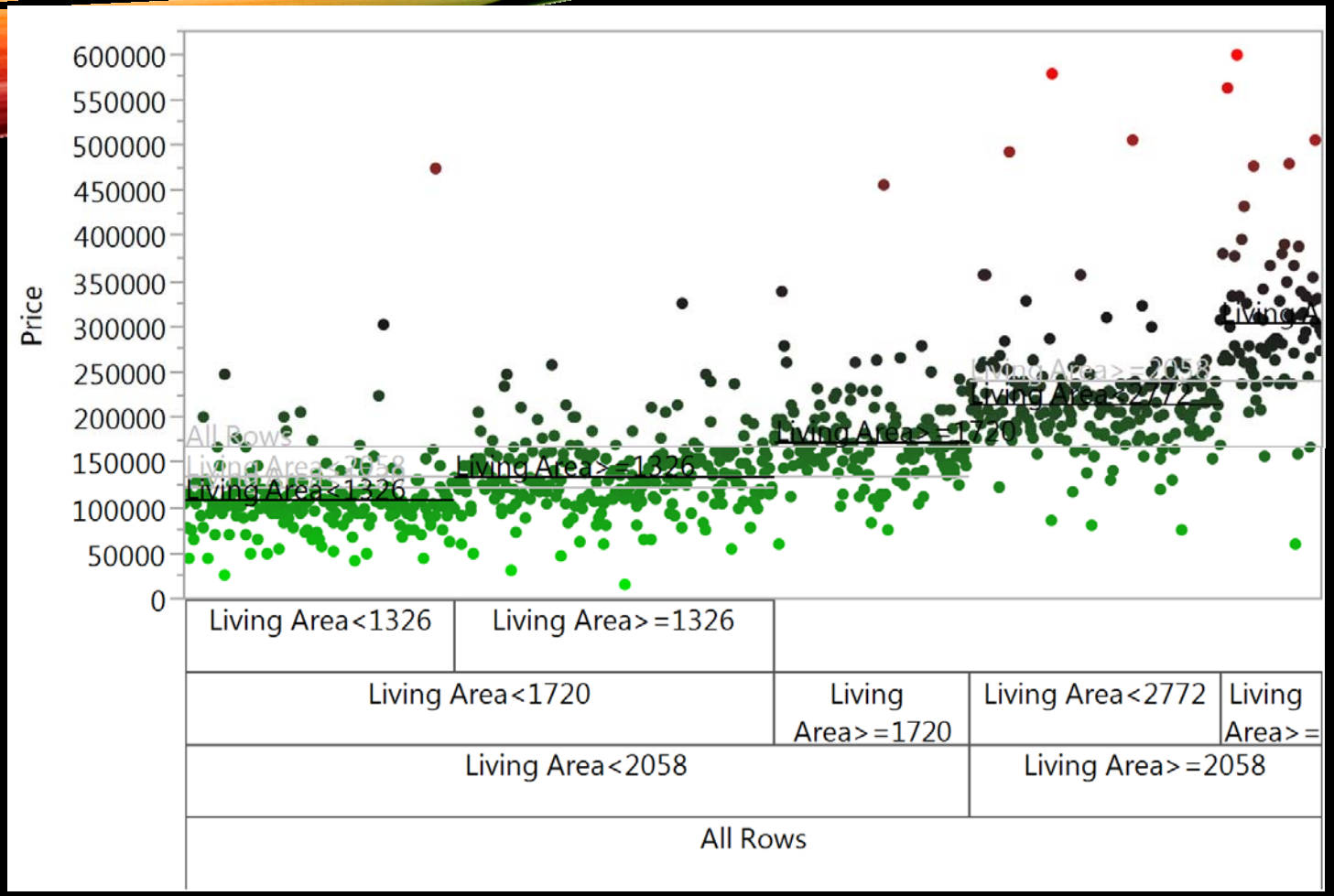
Regression Trees (Partition)

Notes:

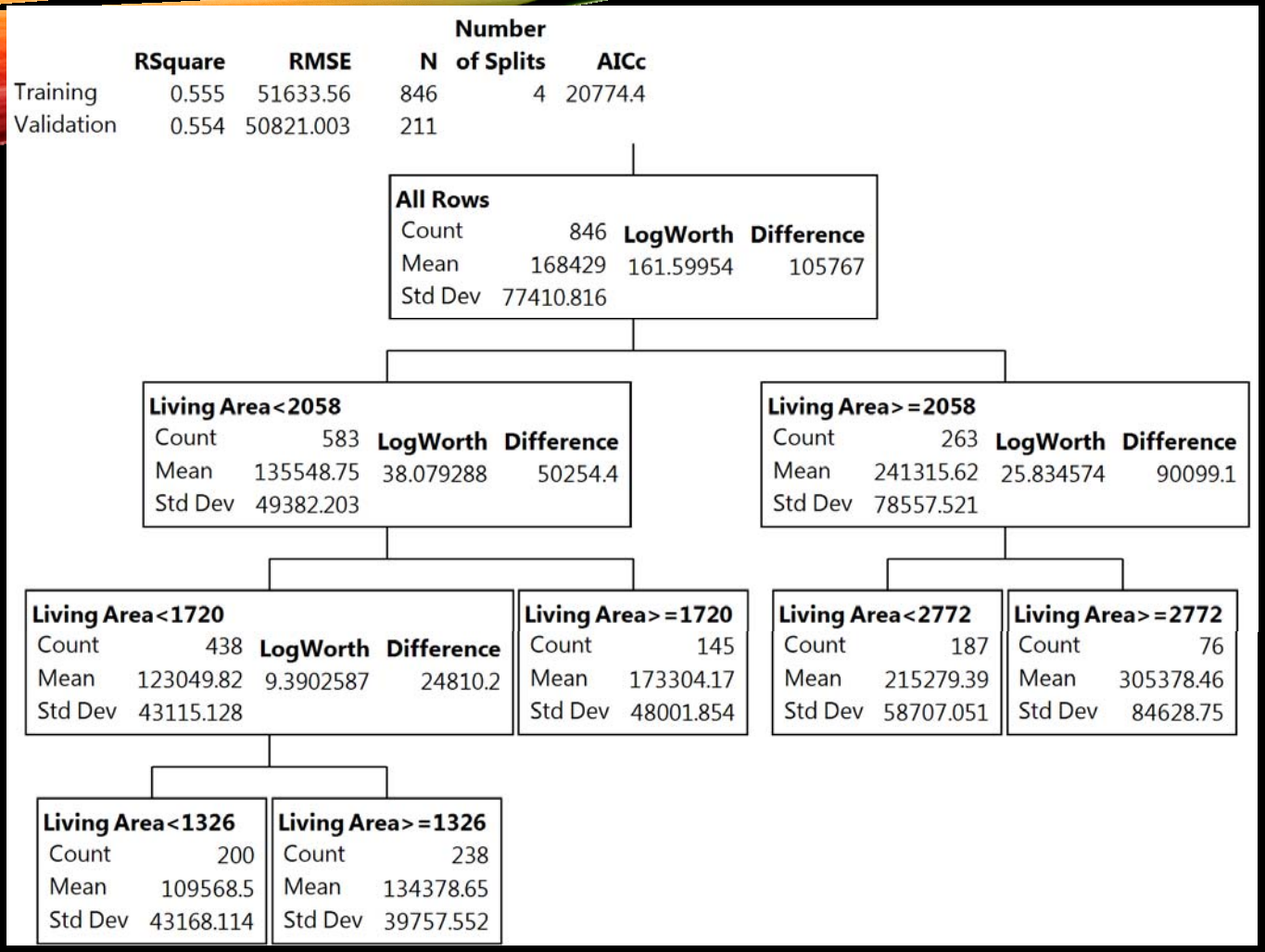
For additional options, such as **Leaf Report**, **Small Tree View**, **Column Contributions**, click the **top red triangle**. Other options, such as **Save Prediction formula** and **Make SAS® DATA Step**, are available from the **top red triangle > Save Columns**. For split options for a particular node, click on the **red triangle for that node**.

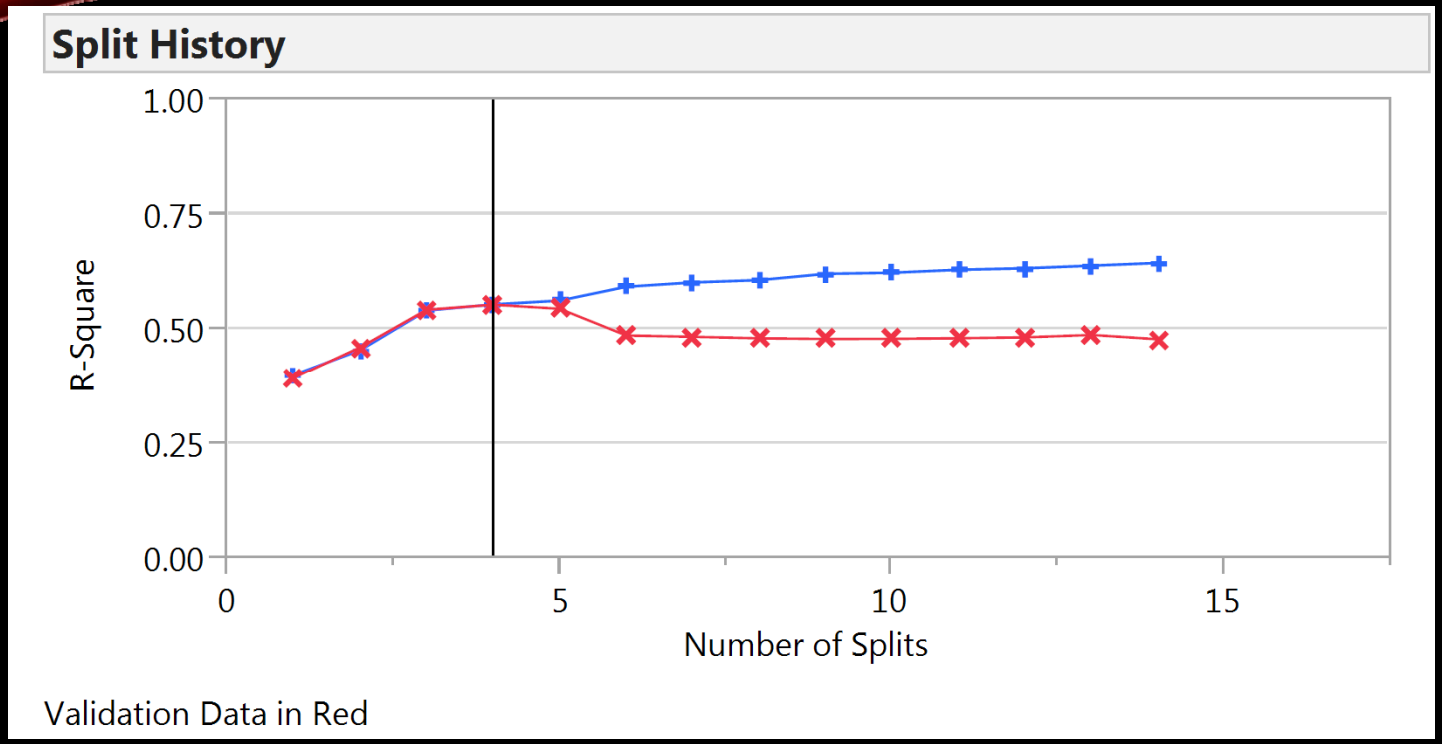
For more information on fitting and evaluating regression trees, including **Validation**, **Bootstrap Forest** and **Boosted Trees**, search for "partition trees" in the JMP Help or in the ***Specialized Models*** book (under **Help > Books**).

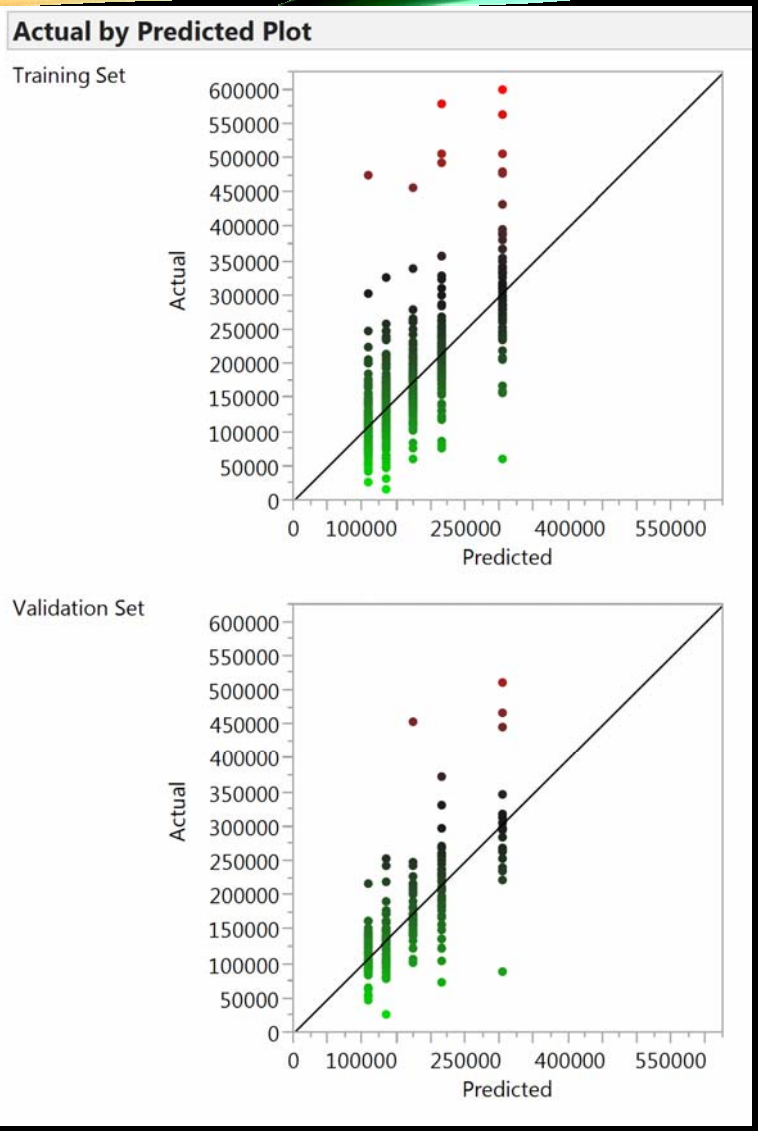

Partition - JMP Pro
Recursive partitioning
Select Columns
7 Columns
Price
Living Area
Bedrooms
Bathrooms
Fireplaces
Age
Validation
Informative Missing
Ordinal Restricts Order
Method
Decision Tree
Validation Portion
0
Cast Selected Columns into Roles
Y, Response
Price
optional
X, Factor
Living Area
Bedrooms
Bathrooms
Fireplaces
Weight
optional numeric
Freq
optional numeric
Validation
Validation
By
optional


Price
600000
550000
500000
450000
400000
350000
300000
250000
200000
150000
100000
50000
0
All Rows
Living Area<1326
Living Area>=1326
Living Area<1720
Living Area>=1720
Living Area<2772
Living Area<1326
Living Area<1720
Living Area>=1720
Living Area<2772
Living Area>=
Living Area<2058
Living Area>=2058
All Rows

	RSquare	RMSE	N	Number of Splits	AICc
Training	0.555	51633.56	846	4	20774.4
Validation	0.554	50821.003	211		

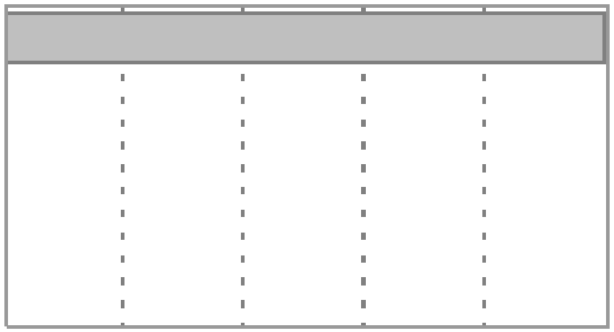


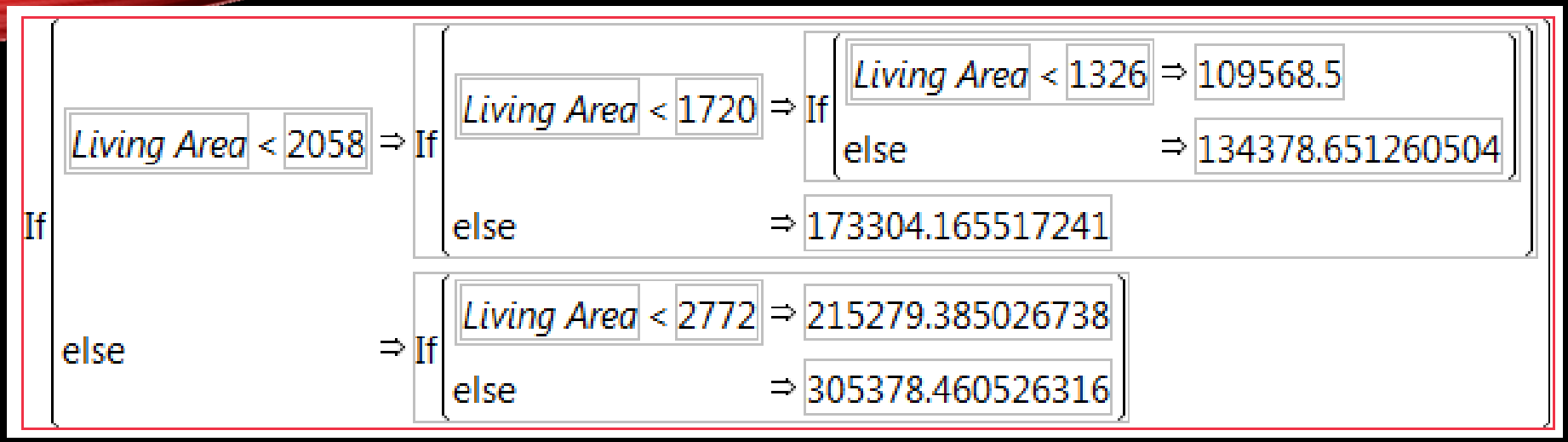

Split History
1.00
0.75
0.50
0.25
0.00
R-Square
0
5
10
15
Number of Splits
Validation Data in Red


Actual by Predicted Plot
Training Set
Actual
Predicted
0
50000
100000
150000
200000
250000
300000
350000
400000
450000
500000
550000
600000
Validation Set
Actual
Predicted

Column Contributions

Term	Number of Splits	SS	Portion
Living Area	4	2.8082e+12	1.0000
Bedrooms	0	0	0.0000
Bathrooms	0	0	0.0000
Fireplaces	0	0	0.0000
Age	0	0	0.0000




If
Living Area < 2058 ⇒ If
Living Area < 1720 ⇒ If
Living Area < 1326 ⇒ 109568.5
else ⇒ 134378.651260504
else ⇒ 173304.165517241
else ⇒ If
Living Area < 2772 ⇒ 215279.385026738
else ⇒ 305378.460526316