Classifying Customers Using IBM SPSS Modeler (v16)

Student Guide

Course Code: 0A0U5

Classifying Customers Using IBM SPSS Modeler (v16)
0A0U5
ERC: 1.0
Published August 2014

Contents

Course Overview

Classifying Customers Using IBM SPSS Modeler (v16) is an intermediate level course that provides an overview of how to use IBM SPSS Modeler to predict the category to which a customer belongs. Students will be exposed to rule induction models such as CHAID and C&R Tree. They will also be introduced to traditional statistical models and machine learning models. Business use case examples include: predict whether a customer switches to another provider/brand and whether a customer responds to a particular advertising campaign. Although this course focuses on classifying customers (including students, patients, employees, and so forth), the techniques can also be applied to business questions such as predicting breakdown of machine parts.

Intended Audience

IBM SPSS Modeler Analysts who have completed the Introduction to IBM SPSS Modeler and Data Mining course who want to become familiar with the modeling techniques used to classify customers in IBM SPSS Modeler. This includes data analysts and analytics business users.

Topics Covered

- Introduction to Classifying Customers
- Building Your Tree Interactively with CHAID
- Building Your Tree Interactively with C&R Tree and Quest
- Building Your Tree Directly
- Using Traditional Statistical Models
- Using Machine Learning Models

Course Prerequisites

Participants should have:

- Experience using IBM SPSS Modeler, including familiarity with the IBM SPSS Modeler environment, creating streams, importing data (Var. File node), basic data preparation (Type node, Derive node, Select node), reporting (Table node, Data Audit node), and creation of models.
- Introduction to IBM SPSS Modeler and Data Mining (v16)

Document Conventions

Conventions used in this guide follow Microsoft Windows application standards, where applicable. As well, the following conventions are observed:

Bold	Bold style is used in demo and workshop step-by-step solutions to indicate either: • actionable items (Point to **Sort**, and then click **Ascending**.) • text to type or keys to press (Type **Sales Report**, and then press **Enter**.) • UI elements that are the focus of attention (In the **Format** pane, click **Data**)
Italic	Used to reference book titles.
CAPITALIZATION	All file names, table names, column names, and folder names appear in this guide exactly as they appear in the application. To keep capitalization consistent with this guide, type text exactly as shown.
MODELER	Used to reference IBM SPSS Modeler version 16.

Workshops

Workshop Format

Workshops are designed to allow you to work according to your own pace. Content contained in a workshop is not fully scripted out to provide an additional challenge. Refer back to demonstrations if you need assistance with a particular task. The workshops are structured as follows:

The Business Question Section

This section presents a business-type question followed by a series of tasks. These tasks provide additional information to help guide you through the workshop. Within each task, there may be numbered questions relating to the task. Complete the tasks by using the skills you learned in the module. If you need more assistance, you can refer to the Task and Results section for more detailed instruction.

The Task and Results Section

This section provides a task based set of instructions that presents the question as a series of numbered tasks to be accomplished. The information in the tasks expands on the business case, providing more details on how to accomplish a task. Screen captures are also provided at the end of some tasks and at the end of the workshop to show the expected results.

Additional Training Resources

Bookmark Business Analytics Product Training http://www-01.ibm.com/software/analytics/training-and-certification/ for details on:

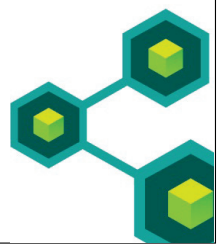
- instructor-led training in a classroom or online
- self-paced training that fits your needs and schedule
- comprehensive curricula and training paths that help you identify the courses that are right for you
- IBM Business Analytics Certification program
- other resources that will enhance your success with IBM Business Analytics Software

IBM Product Help

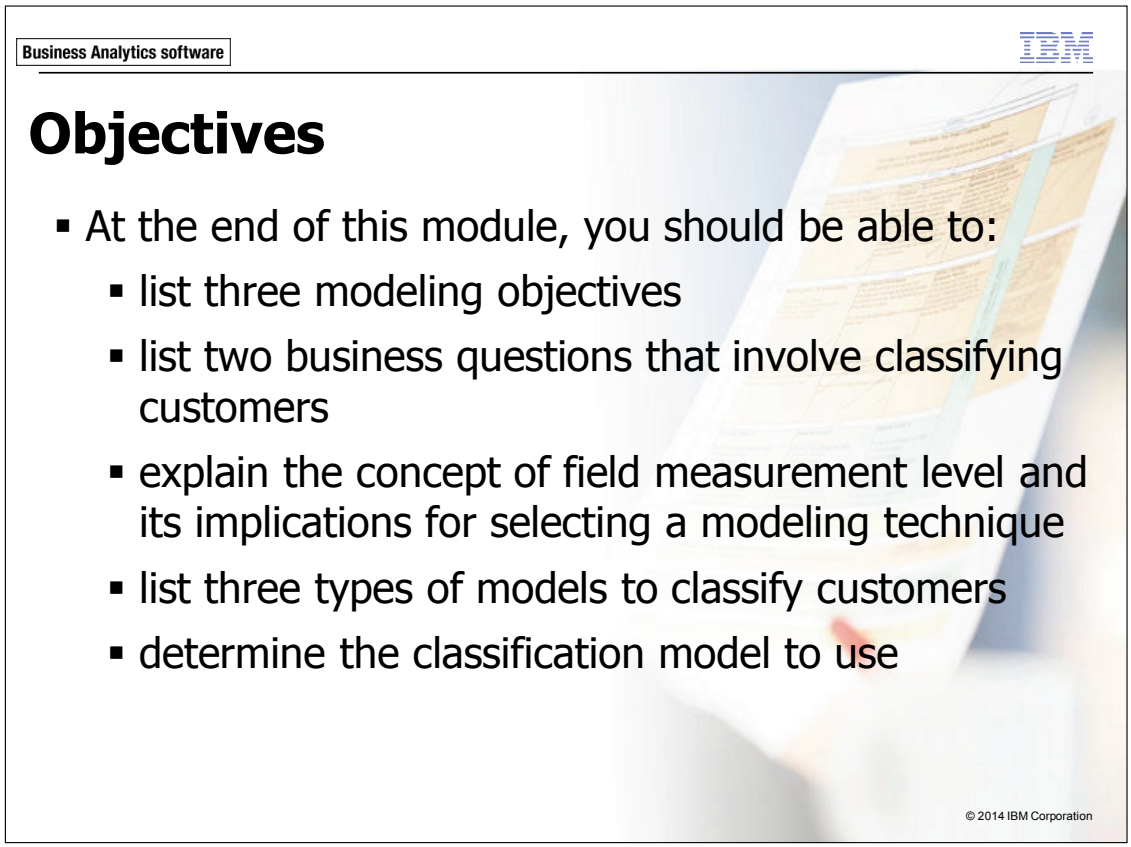
Help type	When to use	Location
Task-oriented	You are working in the product and you need specific task-oriented help.	*IBM Product* - Help link
Books for Printing (.pdf)	You want to use search engines to find information. You can then print out selected pages, a section, or the whole book. Use Step-by-Step online books (.pdf) if you want to know how to complete a task but prefer to read about it in a book. The Step-by-Step online books contain the same information as the online help, but the method of presentation is different.	Start/Programs/*IBM Product*/Documentation
IBM on the Web	You want to access any of the following:	
	• Training and Certification Web site	• http://www-01.ibm.com/software/analytics/training-and-certification/
	• Online support	• http://www-947.ibm.com/support/entry/portal/Overview/Software
	• IBM Web site	• http://www.ibm.com

Introduction to Classifying Customers

IBM SPSS Modeler (v16)



Business Analytics software



Before reviewing this module you should be familiar with:

- working with MODELER (streams, nodes, palettes)
- importing data (Var. File node)
- defining measurement levels, roles, blanks, and instantiating data (Type node)
- examining the data (Table node, Data Audit node)

Business Analytics software

IBM

Determining Your Modeling Objective

MODELING OBJECTIVE	DESCRIPTION
classification	predict a target, using one or more predictors
segmentation	cluster records, using one or more input fields
association	find associations between categories

MODELER provides you with many models. Although there are different taxonomies, these models can be classified into three main categories, according to the objective for modeling. This slide lists these modeling objectives.

Classification models have in common that a target field is predicted, using one or more predictors, or inputs as they are called in the Type dialog box. Applications include:

- Telco: Use service usage data to predict which customers are liable to transfer to another provider (a phenomenon known as churn).
- Banking: Use demographical information such as age, sex, income, and socio-economic status to predict which customers are at risk of not paying back a loan.

Segmentation models cluster records, based on one or more input fields. There is no target field. Important applications are:

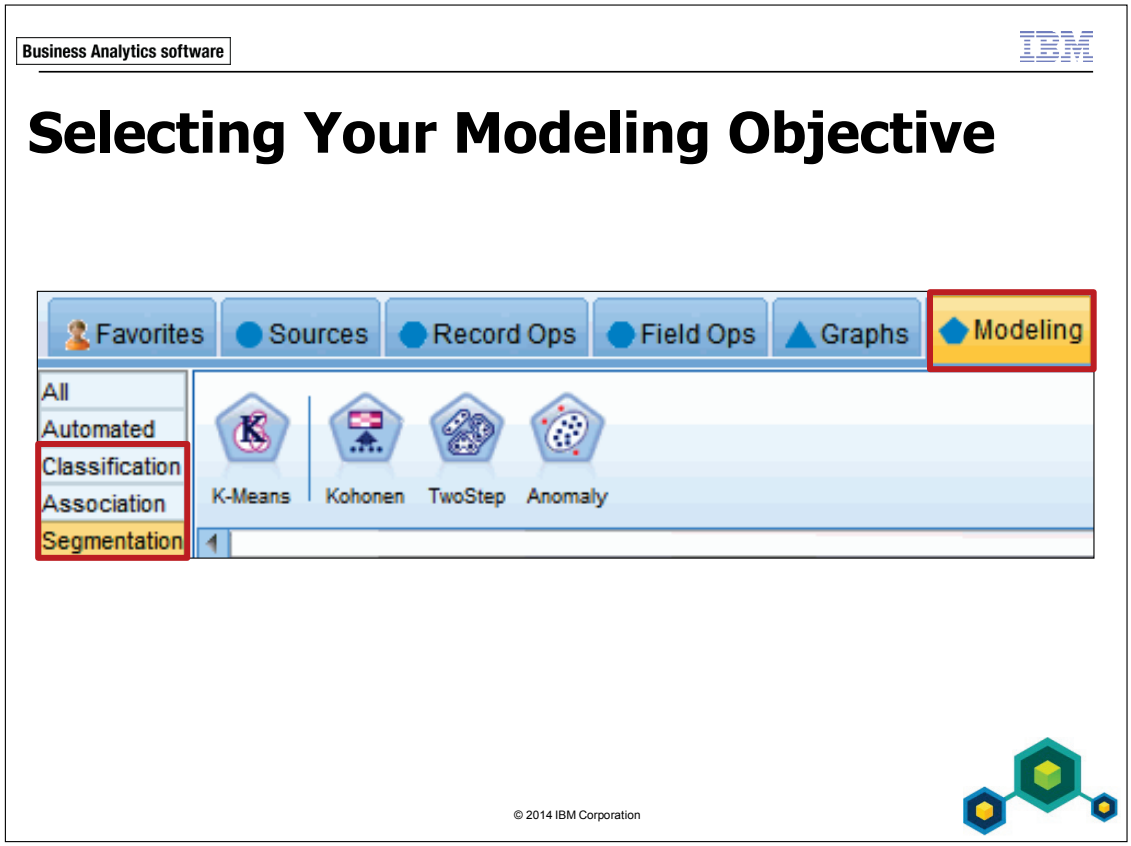
- Marketing: Create segments of bronze, silver and gold customers, based on fields such as recency, frequency and monetary value.
- Insurance: Cluster claims and look for unusual records within the groups. Also known as anomaly detection, for example to detect fraudulent claims.

Association models describe relationships between fields' categories. A typical example is found in market basket analysis, where one has a number of fields that flag product possession. An association model produces rules such as:

Rule #1: 60% of the customers have product A and B and 80% of these customers also have product C.

Rule #2: 40% of the customers have product C, D and G and 75% of these customers also have product A.

Association models typically have fields that have both role input and target. For example, product A is an input field for rule #1 and a target for rule #2.



The three modeling objectives are itemized in sub palettes when the Modeling palette is selected. Selecting a specific item selects all models for that task.

This slide shows the Modeling palette when the Segmentation item is selected. The four models that are presented provide methods to cluster records.

Note: To better select a model in demonstrations and workshops, select the appropriate item first. This restricts the number of models that are displayed in the Modeling palette so that you can find the model of interest easier.

Business Analytics software

IBM

Determining Measurement Level

ICON	MEASUREMENT LEVEL	EXAMPLES
	flag	churn (y/n)
	nominal	region (north /east/south/west)
	ordinal*	educational level (1 – 7)
	continuous	age
	typeless	customer_id

Categorical fields

* **ordinal predictors must have numeric storage to be handled as ordinal.**

Within the group of classification models, not every model can be used for a certain classification task. The measurement level of the target determines which models are appropriate. Thus, the concept of measurement level is crucial to selecting the appropriate model. Refer to the *Introduction to IBM SPSS Modeler and Data Mining (v16)* course for a presentation of measurement levels.

As an example, when you predict a categorical target using the CHAID node, then the CHAID node will be labeled with the name of the target. However, when you use the Linear node to predict the same categorical target, the Linear node will not be labeled with the name of the target. This means that Linear is not an appropriate modeling technique to predict a categorical target.

Note: The choice of a classification model does not depend on the measurement level of the predictors; it only depends on the measurement level of the target.

Business Analytics software

IBM

Classifying Customers

- Focus on classification models
- Focus on predicting a categorical target
  - specifically a flag target

The focus in this course is on classification models that enable you to predict a categorical target. To simplify the presentation, the focus will be on classification models for flag targets. The models that are reviewed in this course, however, also apply to nominal and ordinal targets, although those would need more elaboration.

Thus, predictive models are presented that classify customers into two groups. Examples include:

- Predict whether a customer is a good or bad risk for a loan.
- Predict whether a customer does or does not churn.
- Predict whether a customer responds to a mailing yes/no.
- Predict whether a customer is satisfied or not.
- Predict whether a claim is fraudulent or not.
- Predict whether a student fails or passes an exam.

Although this course focuses on classifying customers, classification models apply to other business questions also. For example, an industrial manufacturing company may monitor the operating condition of various pieces of machinery in order to predict breakdowns. Here classification applies to classifying machines or machine parts.

Thus, although the examples that are presented in this course only deal with classifying customers, the same models can be applied to any business question involving prediction. From a modeling perspective, all these questions come down to the same issue: to predict a flag target with predictors of any measurement level.

Business Analytics software

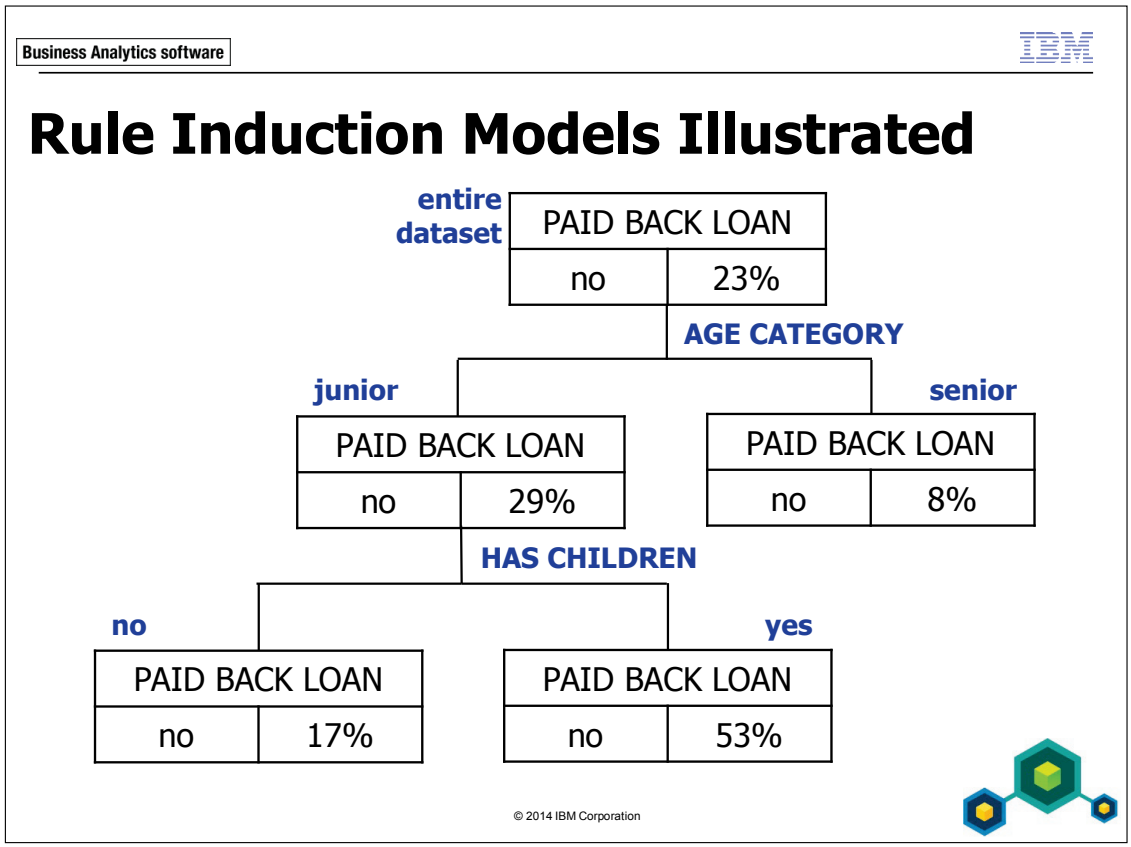
IBM

Determining Your Classification Model

TYPE OF CLASSIFICATION MODEL	DESCRIPTION
rule induction	list rules that describe distinct segments within the data in relation to the target
traditional statistical model	use statistical test to produce a set of equations
machine learning techniques	learn complex patterns

MODELER offers many classification models, for both categorical and continuous targets. Three classes of classification models can be distinguished:

- Rule induction models: These models derive a set of rules that describe distinct segments within the data in relation to the target. The model's output shows the reasoning for each rule and can therefore be used to understand the decision-making process that drives a particular outcome. Models that produce trees belong to this class of models and are presented in this course.
- Traditional statistical models: These models produce a set of equations and they make certain assumptions about the fields' distributions to run statistical tests.
- Machine learning models: These models are optimized to learn complex patterns. Machine learning models make no assumptions about the fields' distributions as traditional statistical techniques do. Machine learning models do not produce a set of rules as rule induction models do. Machine learning models are often referred to as black box models.

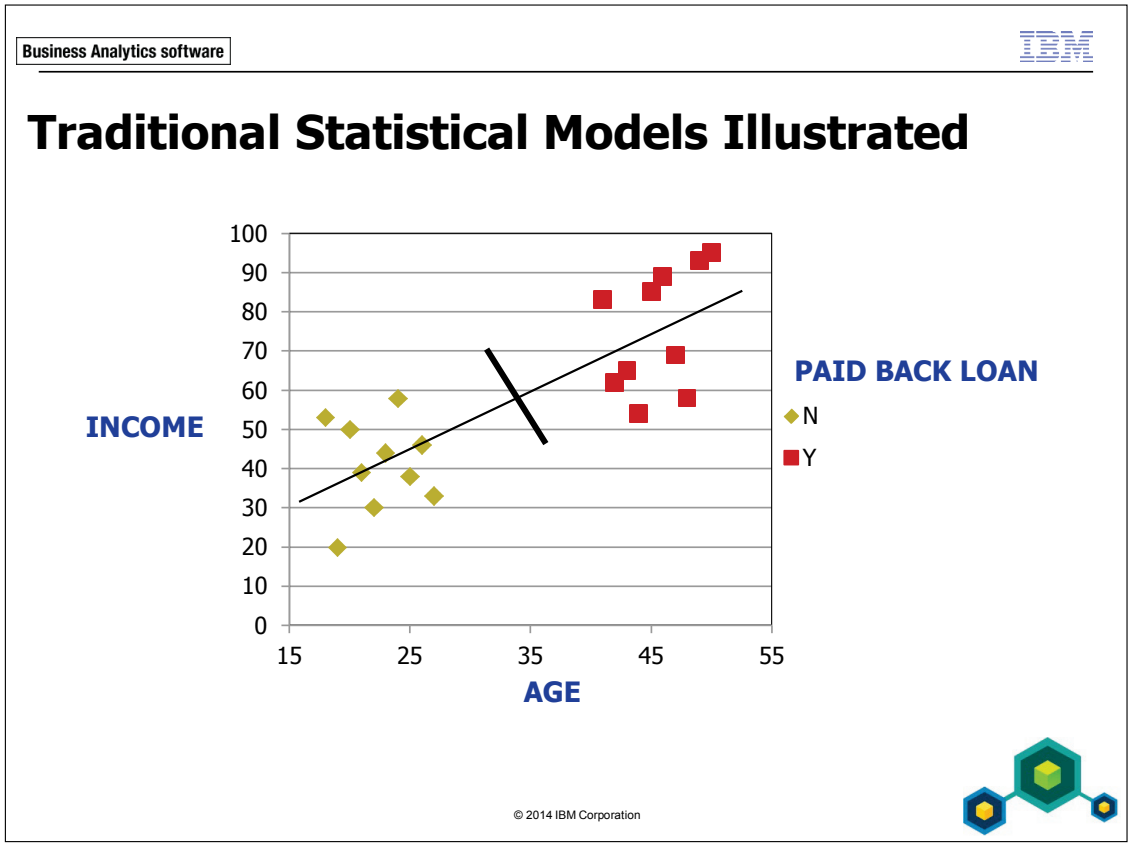


As an example, this slide presents an analysis of a bank that was interested in profiling groups that did not pay back a loan. The root node (top of the tree) shows that 23% of the customers were a risk to the bank because they did not pay back the loan. The root node is split on AGE CATEGORY, which means that AGE CATEGORY is related to PAID BACK LOAN. Juniors show a higher percentage of risk than seniors, but within the junior group there is a further difference between those without and with children. The rules are derived from the terminal nodes (the nodes that are not split):

- Juniors without children: 17% did not pay back the loan
- Juniors with children: 53% did not pay back the loan
- Seniors: 8% did not pay back the loan

Having this information in place, the bank could act upon it. For example, the bank could apply a different acceptance policy for new customers that are juniors with children.

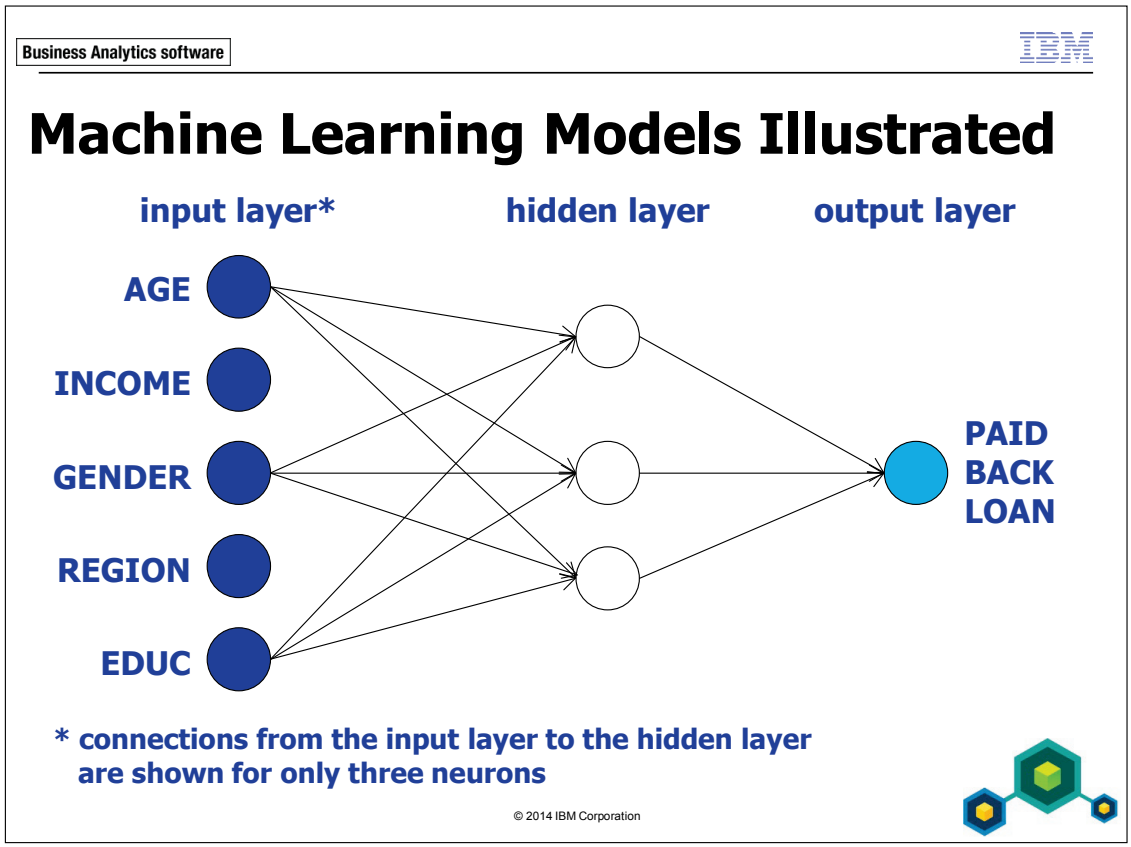
This course presents CHAID, C&R Tree, Quest and C5.0. These rule induction models present their results in a tree-like figure. The first three models enable you to build your tree interactively, so you can choose the fields on which the nodes are split, thus enabling you to apply your business knowledge.



This slide sketches the idea of a traditional statistical model, Discriminant. Discriminant will combine the predictors into a new field in such a way that the two categories of the target field will be separated maximally on this new field.

In this example, given the input fields AGE and INCOME, a new field is computed by an equation such as NEW FIELD = a * AGE + b ´INCOME, with weights a and b that will maximally separate the groups PAID BACK LOAN=N and PAID BACK LOAN=Y. This new field is represented in the plot by the straight line running from the lower left to the upper right. This new field will separate the PAID BACK LOAN categories Y/N and the midpoint (represented here by the short line segment) helps to classify a record to either group.

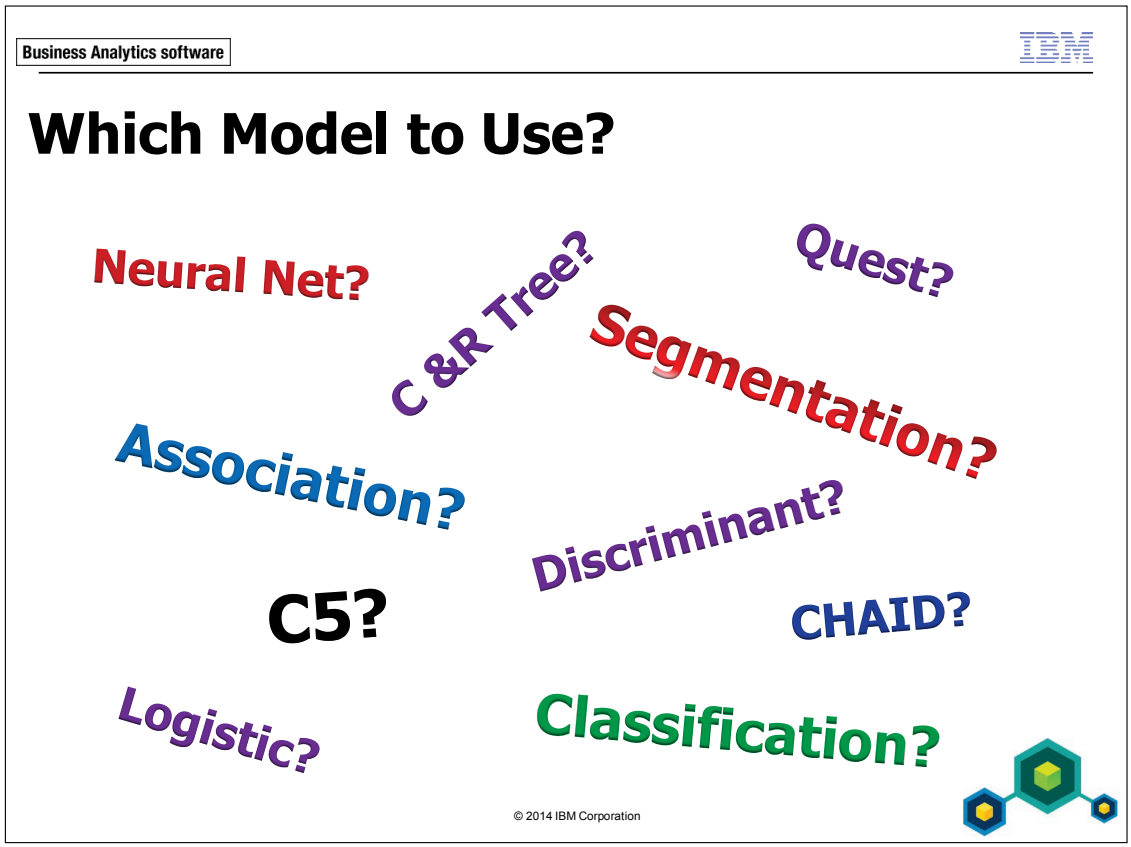
This course presents two traditional statistical models: Discriminant and Logistic.



As an example of a machine learning model, consider the Neural Net model, depicted here. The input layer represents the predictors, the output layer, and the target. There is an intermediate layer which transmits the inputs to the output. The layers are made up of neurons, thus mimicking the human brain. All neurons in one layer of the network are connected to all neurons within the next layer. The weights for the connections are computed in such a way that, given the input values, the output value is close, if not identical, to the actual value of the target.

Neural Net will produce a set of equations, but because there is a hidden layer the interpretation of the weights is not straightforward, as with traditional statistical models. The same goes for the other machine learning models.

This course presents the Neural Net model as one example of machine learning techniques.



Your modeling objective will determine which model you choose.

When a classification model is needed, a second decider is the business context. When you want to have explicit rules that align with marketing campaigns, you will prefer a rule induction model over a black box machine learning model. When the business context is such that the model itself is of no interest, but only that the model should have the highest accuracy in predicting the target, then each model is a candidate for the modeling task. Or you could even combine multiple models into one.

A third decider in the choice of a classification model is the measurement level of the target. Some models are only appropriate for categorical targets, others for continuous targets. It cannot be emphasized enough that setting correct measurement levels is a necessary condition for any analysis.

But even when one class of models is preferred over another, more models within that class are available. Models differ in how they handle missing values, how they handle categorical predictors or continuous predictors, and how they score data. Such differences are addressed in this course. In the end, however, it is always the business user, balancing all pros and cons, who decides which model should be used.

Apply Your Knowledge

Purpose:
Use the questions in this section to test your knowledge of the course material.

Question 1: A telecommunications firm uses transactional data such as number of outgoing calls, number of incoming calls, and number of text messages to groups its customers into "leaders" and "followers". This is an example of:

A. Classification

B. Segmentation

C. Association

Question 2: A retailer runs an analysis on what customers have in their shopping carts to find which product combinations are most popular. This is an example of:

A. Classification

B. Segmentation

C. Association

Question 3: An insurance company has historical data on claims with a fraction of claims flagged as fraudulent. The company discovered that fraud is related to number of claims within a one-year period, claim amount, and the gender and age of the policy holder. This is an example of:

A. Classification

B. Segmentation

C. Association

Question 4: Which of the following statements is the correct statement?

A. The role of a field in a model is set in the Action column in the Type node.

B. The measurement level of a field is set in the Values column in the Type node.

C. The measurement level of the target determines which classification model you can use.

D. The measurement levels of the predictors determine which classification model you can use.

Question 5: What is the measurement level of a field named SEASON that stores the values 1, 2, 3, and 4, representing spring, summer, fall, and winter respectively?

A. Flag

B. Nominal

C. Ordinal

D. Continuous

Question 6: Select all that apply.

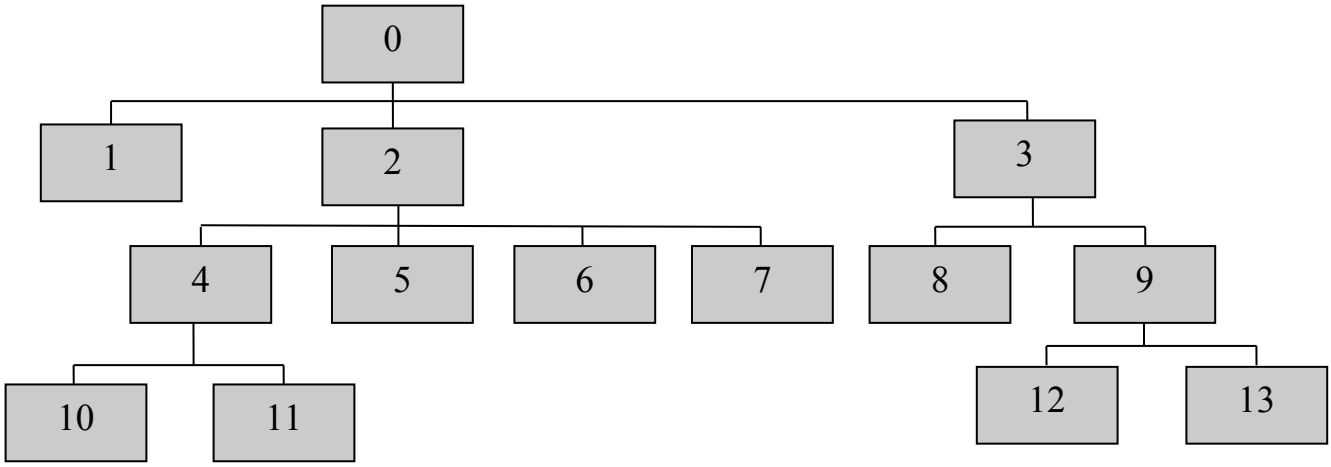
A. If the business context urges that the model must be explained in a set of rules, machine learning models are inappropriate.

B. A traditional statistical model produces a set of rules.

C. All classification models are appropriate for a flag target.

D. A number of classification models can be used to predict a flag target.

Question 7: A map of the tree below shows nodes only and includes no node details. This tree defines how many rules?



A. 1

B. 9

C. 13

D. You cannot tell

Answers to questions:

Your responses to the Apply Your Knowledge questions should appear as follows.

Answer 1: B. Grouping customers on transactional data is an example of segmentation.

Answer 2: C. Using shopping cart analysis to find popular product combinations is an example of association.

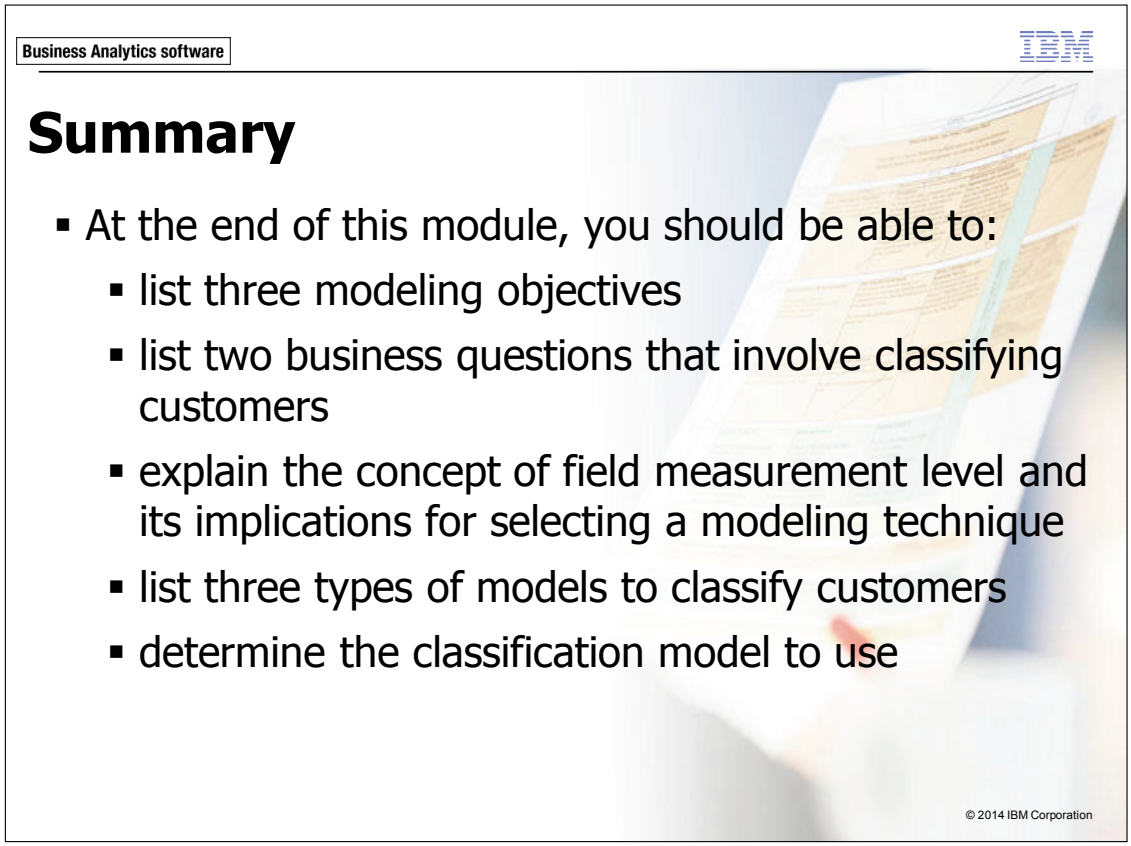
Answer 3: A. Using background information on claims and customers to classify a claim as fraudulent or not is an example of classification.

Answer 4: C. The measurement level of the target, not the measurement level of the predictors, determines which classification models can be used. The measurement level of a field is set in the Measurement column of the Type node and a field's role is set in the Role column in the Type node.

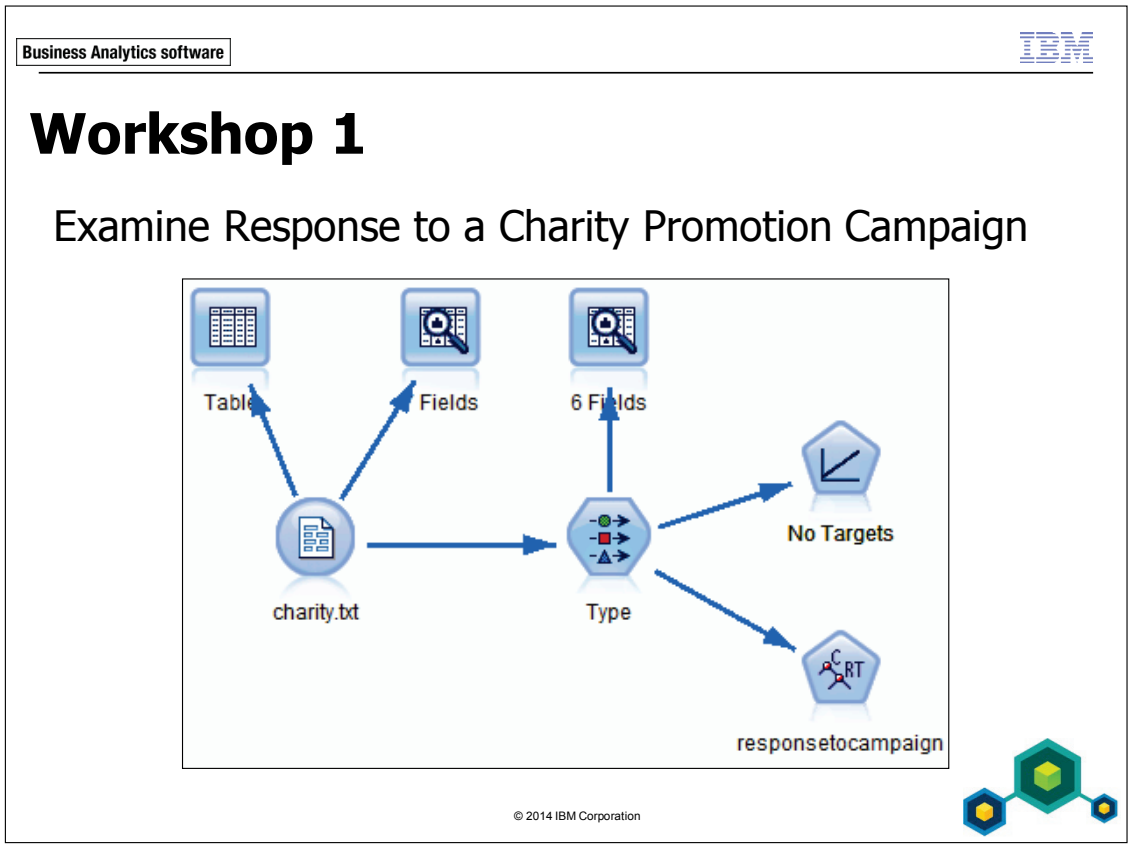
Answer 5: B. The measurement level of SEASON is nominal, although its values are integers in this example.

Answer 6: A, D. If the business context urges that the model must be explained in a set of rules, machine learning models are inappropriate. A number of different classification models can be used to predict a flag target. Traditional statistical models produce equations, not a rule set. Not all classification models can be used to predict a flag target.

Answer 7: B. The decision tree has nine terminal nodes, therefore will produce nine rules.



This module introduced you to the modeling capabilities of MODELER. You should now be able to explain how your modeling objective determines your choice of the model. Also, you should now be able to distinguish the three types of classification models. Classification models to predict a categorical target are the focus of this course.



The following (synthetic) file is used in this workshop:

- **charity.txt**: A text file that represents data from a charity organization. It contains information on individuals who were mailed a promotion. The information includes whether the individuals responded to the campaign, their spending behavior with the charity and basic demographics such as age, gender and demographic group. The file is located in **C:\Train\0A0U5**.

Before you begin with the workshop, ensure that:

- You have started MODELER (click **Cancel** when the splash screen appears when you start MODELER).
- You have set MODELER's working folder. (In MODELER, select **File\Set Directory**. When you are working in an IBM training environment, browse to the folder **C:\Train\0A0U5** and then click **Set** to select it. When you are not working in an IBM training environment, ask your trainer for the folder name.)

Workshop 1: Examine Response to a Charity Promotion Campaign

In this workshop you will explore the charity data and set the roles for the fields.

To do this, you must:

- Use a **Var. File** node (Sources palette) to import data from the text file **charity.txt** and examine the data with a **Table** node (Output palette) and a **Data Audit** node (Output palette).

  How many records are there in this dataset?

  What is the percentage of customers that responded positively to the campaign?

- Add a **Type** node (Field Ops palette) downstream from the **Var. File** node, edit the **Type** node and click the **Read Values** button to instantiate the data (so the fields' values are known to MODELER). Then, set the **Role** for **gender**, **age**, **mosaic bands**, **pre-campaign expenditure**, and **pre-campaign visits** to **Input**, and the **Role** for **response to campaign** to **Target** (the other fields have **Role None**).

- Add a **Data Audit** (Output palette) node downstream from the **Type** node and run this **Data Audit** node to re-examine the data.

  Is the Data Audit output different from the Data Audit output in the first task?

  Examine the graph for **gender**: Is there a difference between men and women with respect to response to the campaign?

- Add a **Linear** node (Modeling palette, Classification item) downstream from the **Type** node. Also, add a **C&R Tree** node (Modeling palette, Classification item) downstream from the **Type** node.

  Can you think of a reason why the C&R Tree node is labeled **responsetocampaign**, and the Linear node is labeled **No Targets**?

For more information about where to work and the workshop results, refer to the Task and Results section that follows. If you need more information to complete a task, refer to earlier demos for detailed steps.

Workshop 1: Tasks and Results

Task 1. Import and examine the data.

1. From the **Sources** palette, double-click the **Var. File** node to add it to the stream canvas.
2. Double-click the **Var. File** node to open its editor.

   From this point forward in this course, you will be instructed only to edit a node in order to perform this step.
3. In the **Var. File** dialog box, to the right of the **File** box, click **Browse**  (The Browse window should automatically open to the **C:\Train\0A0U5** folder).
4. Select **charity.txt** and then click **Open**.
5. Click **OK** to close the **Var. File** dialog box.

   From this point forward in this course, you will be instructed only to close a dialog box to accomplish what is detailed in this step.
6. From the **Output** palette, drag a **Table** node to the stream canvas and drop it to the right of the source node entitled **charity.txt**.
7. Right-click the **charity.txt** node, click **Connect**, and then click the **Table** node.

   From this point forward in this course, you will be instructed to add a node downstream from another node in order to accomplish what is detailed in steps 6 and 7.
8. Right-click the **Table** node, and then click **Run**.

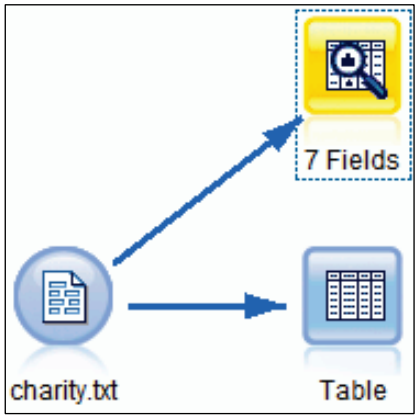
   From this point forward in this course, you will be instructed to run a node in order to accomplish what is detailed in this step.

   The title bar of the Table output window shows that there are 2,398 records in the dataset.
9. Click **OK** to close the **Table** output window.

   From this point forward in this course, you will be instructed to close the output window to accomplish what is detailed in this step.

10. From the **Output** palette, add a **Data Audit** node downstream from the **charity.txt** node.

    A section of the results appear as follows:



11. Run the **Data Audit** node and notice the **7 fields**.
12. In the **Data Audit** output window, scroll to the bottom, if necessary.
13. Double-click the cell that corresponds to the **response to campaign** row and the **Sample Graph** column.

    A section of the results appear as follows:

Value	Proportion	%	Count
No		68.68	1647
Yes		31.32	751

    Somewhat more than 31% of the customers responded to the campaign.

14. Close the **Distribution** window, and then close the **Data Audit** output window.

Task 2. Instantiate the data and set the fields' roles.

1. From the **Field Ops** palette, add a **Type** node downstream from the **charity.txt** node (reposition the **Table** and **Data Audit** nodes to make room for the **Type** node).
2. Edit the **Type** node, and then:

    - click the **Read Values**  button

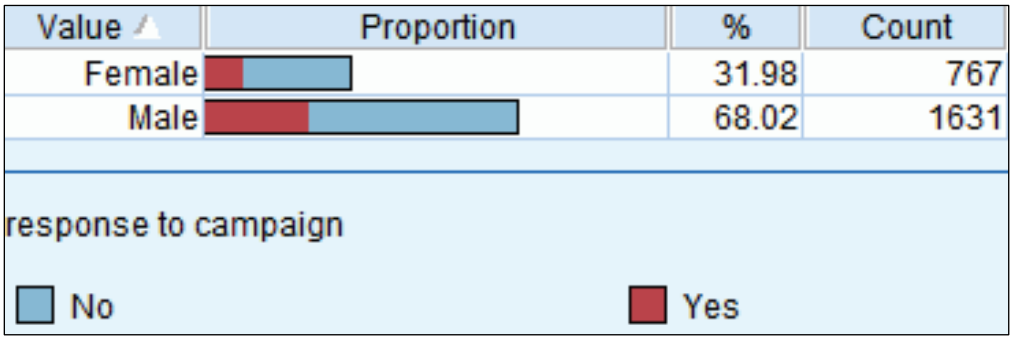
    The Values column is populated with values from the data. Also, the measurement level for customer_id is automatically set to Typeless and its role to None.

- ensure that the **Role** for **gender**, **age**, **mosaic bands**, **pre-campaign expenditure**, and **pre-campaign visits** is set to **Input**
- set the **Role** for **response to campaign** to **Target**
- ensure that the **Role** for other fields is **None**
- close the **Type** dialog box

Task 3. Run a Data Audit node when a target is defined.

1. From the **Output** palette, add a **Data Audit** node downstream from the **Type** node, and then run the **Data Audit** node.
2. In the **Data Audit** output window, double-click the **Sample Graph** for **gender**.

   A section of the results appear as follows:



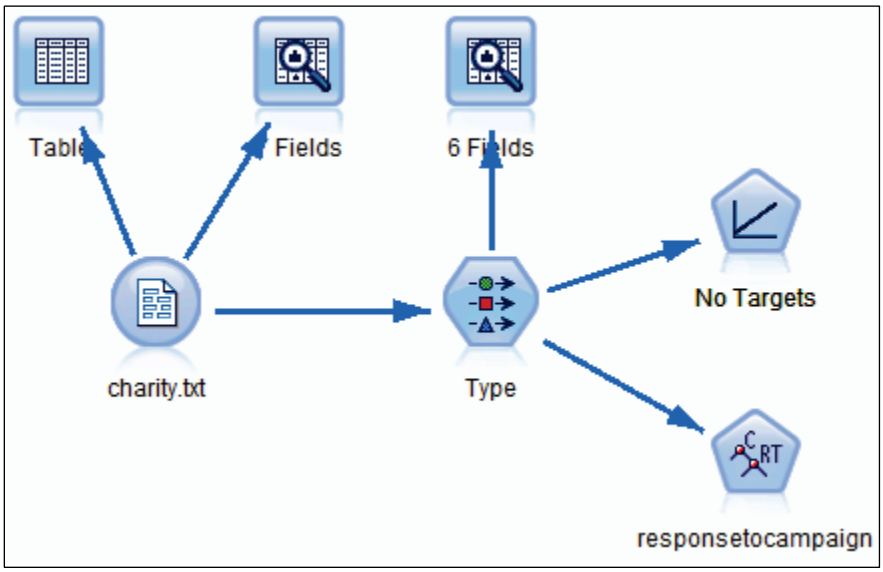
   The distribution is overlaid with the target, response to campaign.

   The distribution for gender shows that men have a higher tendency to respond to the campaign, although that is a little hard to tell from this distribution graph.
3. Close the **Distribution** window, and then close the **Data Audit** output window.

Task 4. Explore which models are relevant to predict response.

1. Click the **Modeling** palette, and then select the **Classification** item.
2. From the **Modeling** palette, add a **Linear** node downstream from the **Type** node.
3. From the **Modeling** palette, add a **C&R Tree** node downstream from the **Type** node.

   A section of the results appear as follows:

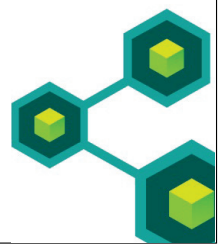


   Apparently, C&R Tree can be used to predict a flag target, while Linear is not an appropriate technique to predict a flag target.
4. Exit MODELER without saving anything.

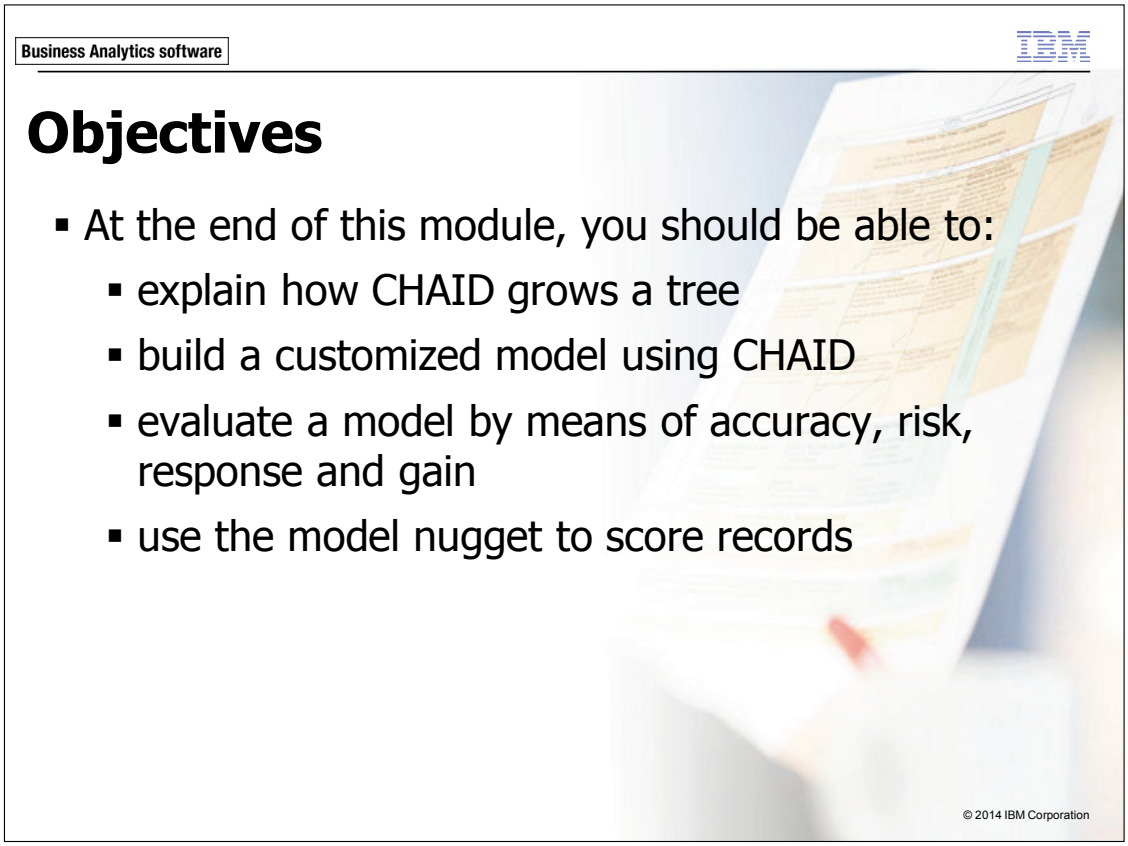
Note: The stream **workshop_introduction_to_classifying_customers_completed.str**, located in the **01-Introduction_to_Classifying_Customers\Solutions** sub folder, provides a solution to the workshop.

Building Your Tree Interactively with CHAID

IBM SPSS Modeler (v16)



Business Analytics software



Before reviewing this module you should be familiar with:

- working with MODELER (streams, nodes, palettes)
- importing data (Var. File node)
- defining measurement levels, roles, blanks, and instantiating data (Type node)
- examining the data (Table node, Data Audit node)
- the three modeling objectives
- the three types of classification models

Business Analytics software

IBM

Building Your Tree Interactively with CHAID

- Build a tree, guided by:
  - business knowledge
  - the CHAID algorithm

When you want to grow a tree you can generate the tree directly, or you can build the tree interactively. In the latter case you can take control of the process and apply your business knowledge, rather than being dependent on how an algorithm builds the model. When you build a tree interactively, this does not mean that the algorithm cannot be used. At any point, you are guided by the algorithm to make your choices.

Building a model interactively prevents overly complex rule sets from being generated and ensures that the model is practical enough to be applied to a business problem.

In this module you will interactively build your tree, guided by the CHAID algorithm. CHAID (CHi -square Automatic Interaction Detection) is one of the simplest and most popular methods to grow a tree, and thus makes a good starting point.

After the tree has been built, it needs to be evaluated to examine its fit. This will be addressed in the second section. Evaluations are presented here in the context of CHAID, but are generic measures and thus will be measures used throughout the course.

Business Analytics software

IBM

A Business Case: Finding High Risk Groups

sample data

ID	GENDER	AGE CATEGORY	HAS CHILDREN	PAID BACK LOAN
1	male	junior	yes	no
2	male	senior	no	yes
3	female	senior	yes	yes
...	...	...	...	...

results

PAID BACK LOAN	N	%
no	559	22.7%
yes	1896	77.3%

As an example, consider a bank that is confronted with a high incidence of loans that are not paid back. The bank needs to sort out which groups are at risk of not paying back the loan. If the bank has this information it can hopefully solve the issue; for example, by not admitting new customers with that profile, raising the premium for such groups, and so forth. All in all, the bank needs to build a predictive model to answer the business question.

To manage the amount of data, the bank has drawn a sample of customers from their database. The first records of this (synthetic) sample are shown here. The sample is comprised of 2,455 customers and includes demographic information and a field that flags whether the customer has paid back the loan. The results table shows that 559 (22.7%) customers in the sample did not pay back their loan, while 1,896 (77.3%) did.

A note on the word "predictive": Predictive models are built on historical data and then applied to future cases. Thus, words such as "predictive" and "prediction" refer to building models on historical data. In this example, historical data are used to "predict" which groups of customers are at risk.

Business Analytics software

IBM

Identifying Important Predictors

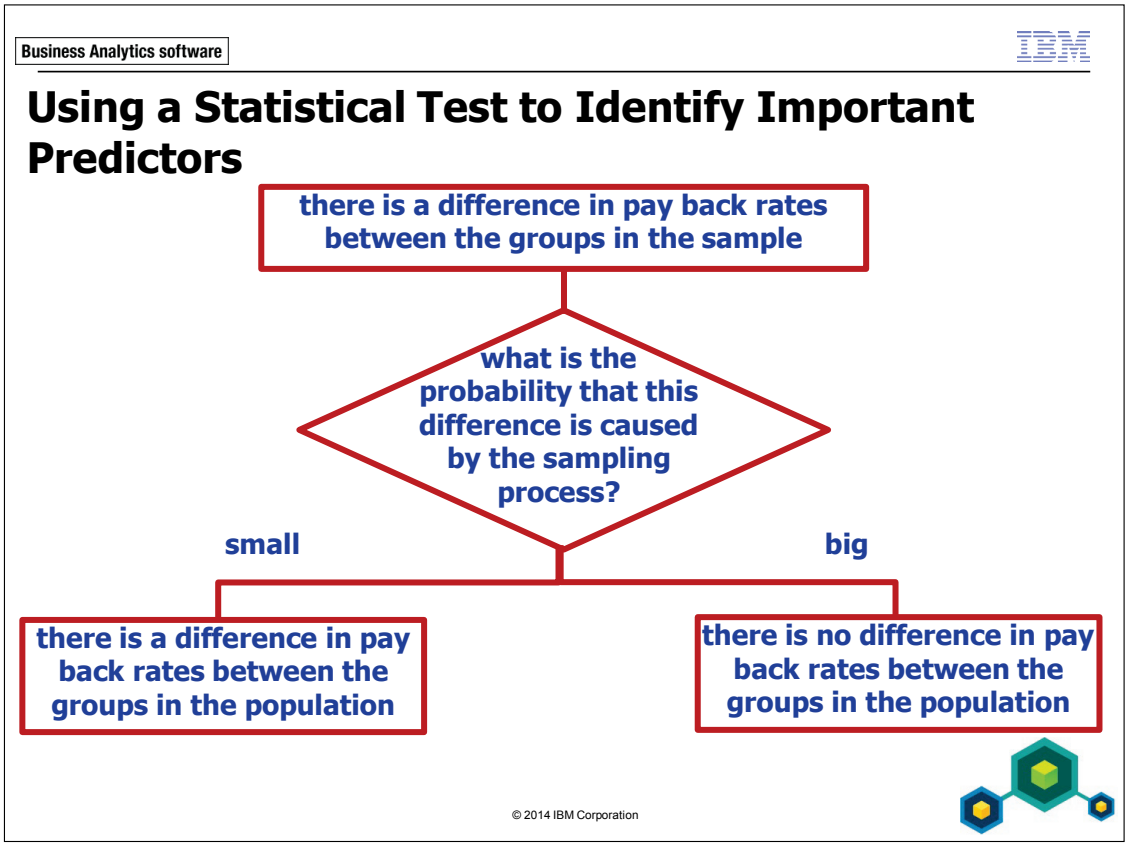
	GENDER	
PAID BACK LOAN	male	female
no	277	282
	22.6%	23.0%
yes	951	945
	77.4%	77.0%

	AGE CATEGORY	
PAID BACK LOAN	junior	senior
no	504	55
	29.1%	7.6%
yes	1230	666
	71%	92.4%

The question is whether the percentage that did not pay back the loan is the same for different groups.

This slide shows how PAID BACK LOAN relates to GENDER and AGE CATEGORY. Examining the table on the left, it appears that 22.6 % of the men did not pay back the loan, while the percentage is somewhat higher for women, 23.0%. Thus, pay back rates hardly differ and you may conclude that GENDER is not relevant for paying back loans.

In the table on the right, 29.1% of the juniors did not pay back their loan, against only 7.6% for seniors. Thus, it seems to matter if the customer is a junior or a senior when it comes to paying back the loan. Of course this is only an informal conclusion, because there is no way to tell when a difference is big enough to call it substantial. Given the percentages in this example it seems justified to conclude that GENDER is not related to PAID BACK LOAN, while AGE CATEGORY is. But what if the percentage pay back is 10% for one group, and 15% for another? What would then be your conclusion?



A statistical test can be used to assess the relevance of a predictor (GENDER, AGE CATEGORY) for the target field (PAID BACK LOAN).

To assess a predictor's relevance, the test has to take into account that the data is a sample. In a sample, there will always be a difference between groups in pay back rate, even if the groups do not differ in pay back rates in the population. In other words, you will always observe a difference in the sample because of the sampling process.

A statistical test computes the probability that the observed difference in the sample is caused by the sampling process. A probability ranges from 0 to 1, so when the probability is close to 1 it means that there is a big probability that the observed sample difference is caused by the sampling process. Hence, you cannot conclude that there is a difference in the population. If the probability is close to 0 then the probability is very small that the observed sample difference is caused by the sampling process, and the difference must be attributed to another reason; that is, you observe a difference in the sample because it reflects a difference in the population.

Business Analytics software

IBM

Setting the Threshold for the Statistical Test

- Rule of thumb:
  - probability >= 0.05
    - the observed sample difference can be attributed to the sampling process
    - conclusion: there is no difference in the population
  - probability < 0.05
    - the observed sample difference cannot be attributed to the sampling process
    - conclusion: there is a difference in the population

Having a statistical test in place that computes the probability that the difference is random (caused by the sampling process), the question then becomes: When is the probability small enough to conclude that the difference is significant? Typically, small probabilities values are considered to be probabilities less than 0.05. Thus, when the probability is smaller than 0.05 you may conclude that the observed sample difference cannot be attributed to the sampling process and that a sample difference reflects a population difference. When the probability is 0.05 or above you may not conclude that there is a difference between the groups in the population.

Setting the threshold to declare a difference significant to such a small value as 0.05 means that it is not too easily concluded that there is a difference. For example, if the probability equals 0.06 then the threshold of 0.05 will still lead to the conclusion that there is no difference. In this sense, setting the significance level to 0.05 is a conservative strategy. Only when the probability is smaller than 0.05, will appropriate action (which may involve huge investments) be taken.

Business Analytics software

IBM

Using the Chi-square Test to Identify Important Predictors

	GENDER	
PAID BACK LOAN	male	female
no	277	282
	22.6%	23.0%
yes	951	945
	77%	77%

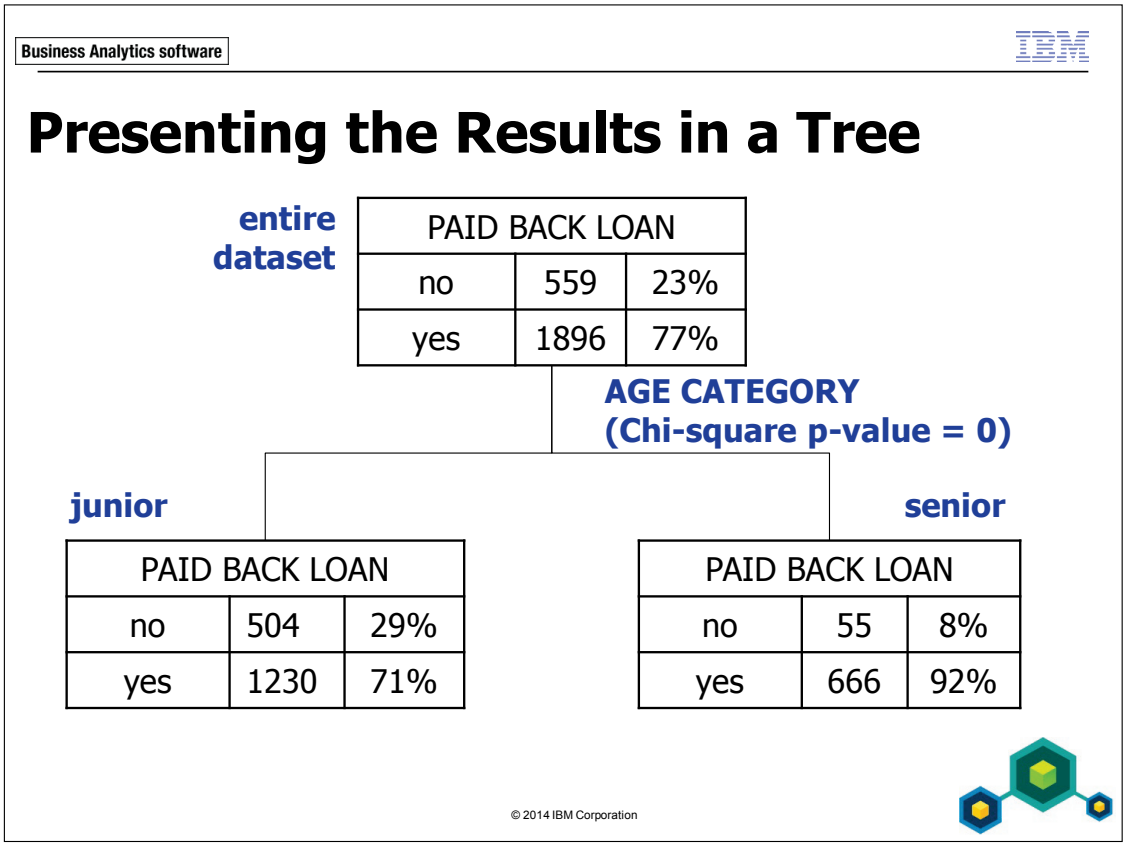
Chi-square value = 0.063
p-value = 0.801

	AGE CATEGORY	
PAID BACK LOAN	junior	senior
no	504	55
	29.1%	7.6%
yes	1230	666
	71%	92.4%

Chi-square value = 133.086
p-value = 0

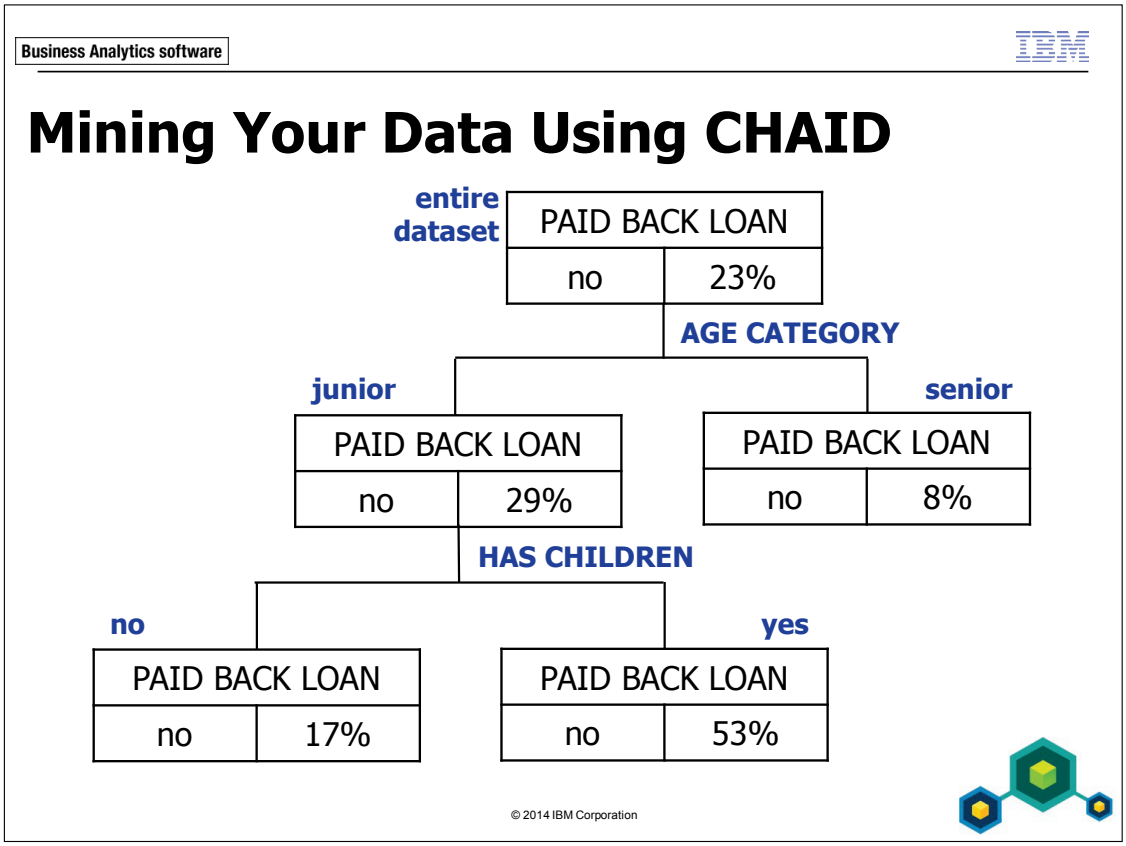
The statistical test that is used when two fields are categorical is the Chi-square test. The Chi-square test reworks the difference in percentages to a value, the Chi-square value, which in itself is only a technical detail. More importantly, the probability can be derived from this value (and from more technical details such as degrees of freedom; refer to a statistics textbook for more information about the Chi-square test).

This slide illustrates the Chi-square test. The Chi-square value for the table on the left is 0.063: close to 0. This reflects the small difference in percentages. This Chi-square value is merely a technical detail to compute the probability that the observed sample difference is caused by sampling. The probability for this table equals 0.801, which is greater than 0.05. Thus, you may conclude that there is no difference in pay back rate between men and women in the population, or, in other words, that GENDER is not related to PAID BACK LOAN. The Chi-square value for the table on the right is 133.086, with an associated probability of 0. Thus, there must be a difference in pay back rate between juniors and seniors, not only in the sample but also in the population from which the sample comes.



Rather than presenting the results in a table, the results can be presented in a tree-like figure.

This slide shows the overall distribution of PAID BACK LOAN in the root node. The root node is the parent node for two child nodes representing the age categories. The Chi-square is displayed for AGE CATEGORY, which is the so-called split field in the tree.



At this point you have found two groups, with one group (juniors) more at risk of not paying the loan than the other group (seniors).

You can mine your data to look for further differences in pay back rates, using the same Chi-square test. For example, you can select the group of juniors, and test whether pay back rates for juniors without children differ from pay back rates of juniors with children. Suppose that 17% of the juniors without children did not pay back their loan, while this percentage was 53% for juniors with children. Also, suppose that this difference was statistically significant (probability value less than 0.05). Your data mining effort then has produced three groups: juniors without children (17% did not pay back the loan), juniors with children: (53 %) and seniors (8%).

This slide summarizes the data mining results. Because the Chi-square test was used, it is an example of a CHAID analysis (CHi -square Automatic Interaction Detection).

Note: You can build such a tree directly, or interactively. In the latter case you can split on any field, not necessarily significant ones, letting your business knowledge prevail.

Business Analytics software

IBM

How CHAID Handles Categorical Predictors

	PREDICTOR X		
PAID BACK LOAN	1	2	3
no	254	250	55
	29.2%	28.9%	7.6%
yes	616	614	666
	70.8%	71.1%	92.4%

Chi-square value = 133.103
p-value = 0

Up to this point the presentation was limited to a flag predictor. What if the predictor has more than two categories? Fortunately, the Chi-square test is not limited to a flag predictor and can be used for any categorical predictor.

This slide shows an example. The probability value for the Chi-square test for this table is 0. This means that there are differences between the three groups in pay back rate. But now it needs further investigation where the differences between the categories lie.

It could be that all percentages differ or that two of the three percentages are much alike and differ from the third percentage. The latter seems to be the case here: category 1 and 2 do not differ much in their pay back rates, and the third category shows a much higher pay back rate. Thus, it looks as if the Chi-square test reflects the difference between the first two categories on one side and the third category on the other.

But again, this is only an informal assessment. What will you conclude if the percentages of not paying back were 27% and 32% for the first and second group, respectively? In short, how can you statistically underpin the conclusion that categories are alike or different?

Business Analytics software

IBM

How CHAID Merges Categories

	PREDICTOR X		
PAID BACK LOAN	1	2	3
no	254	250	55
	29.2%	28.9%	7.6%
yes	616	614	666
	70.8%	71.1%	92.4%

Chi-square value of sub table formed by categories 1 and 2 = 0.014

p-value = 0.905

	PREDICTOR X	
PAID BACK LOAN	1 & 2	3
no	504	55
	29.1%	7.6%
yes	1230	666
	70.9%	92.4%

1 and 2 merged

To assess if two categories are alike or differ, surprisingly enough, the Chi-square comes to rescue, now applied to the sub table of the two categories that you want to compare. This brings the situation back to that of using a flag predictor.

The Chi-square probability value will be computed for all sub tables, and the algorithm will determine the sub table that has the highest probability value. When this probability is greater than or equal to 0.05 it means the categories are alike, and the algorithm will merge the categories into one new category. This will create a new table, to which the same procedure is applied: sub tables of two categories are formed and the Chi-square is applied to examine if categories can be merged. This procedures repeats until no more categories can be merged. This slide shows an example of the procedure. Three sub tables, {1, 2}, {1, 3}, {2, 3}, are formed, and the Chi-square test is applied to each sub table. In this dataset, the sub table {1, 2} produces the highest probability value, which is above 0.05. Subsequently these two categories are merged, resulting in the table on the right. No categories can be merged in that table, so the algorithm stops.

Business Analytics software

How CHAID Merges Categories: Nominal vs. Ordinal Predictors

- Nominal predictors:
  - all categories tested for merge
- Ordinal predictors:
  - only adjacent categories tested for merge
- It is a business decision how you want to treat an ordinal predictor.

On the previous slide, all pairs of categories (1 versus 2, 1 versus 3, and 2 versus 3) were tested for a merge. Pairwise comparisons between all categories are meaningful when the predictor is nominal, but what if the predictor is ordinal? For example, suppose that you have age categories 1 (young), 2 (middle-aged) and 3 (older) and that the pay back rates for categories 1 and 3 are the same and differ from that of category 2, then do you want to merge categories 1 and 3?

Which categories are candidates for a merge is determined by the measurement level of the predictor. For nominal predictors all categories are compared and possibly merged. For ordinal predictors only adjacent categories are compared and possibly merged.

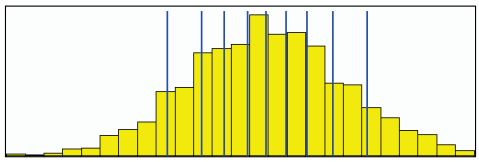
Which measurement level you choose for a predictor is a business decision. Even if the predictor is ordinal you can set its measurement level to nominal to allow the algorithm to find the best merge, and possibly find unexpected results. On the other hand, results may be more difficult to explain.

Business Analytics software

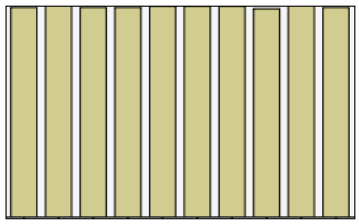
IBM

How CHAID Handles Continuous Predictors

- Bin the continuous predictor into deciles
- Treat as ordinal predictor



INCOME



INCOME binned into deciles

The Chi-square statistic is only applicable to the situation where both predictor and target are categorical. What then if the predictor is continuous? The answer to this question is that the algorithm will create a new field behind the scenes with 10 categories that contain approximately the same number of records. The binned field is treated as an ordinal predictor, and thus only adjacent categories are evaluated for a merge. Because categories can be merged you may have less than 10 categories in a tree rather than the 10 categories that were originally created. That the continuous predictor is binned into categories also implies that you will not be able to create a split at any value of the continuous predictor, but only at the values that define the bins.

Note: If you do not want to be dependent on how the algorithm bins the continuous predictor, you can use the Derive node or Binning node to create your own categories.

Note: When the target field is continuous then the Chi-square test no longer applies. In this situation another statistical test is needed, but the algorithm operates in the same way. Refer to the *Predicting Continuous Targets Using IBM SPSS Modeler (v16)* course for more information.

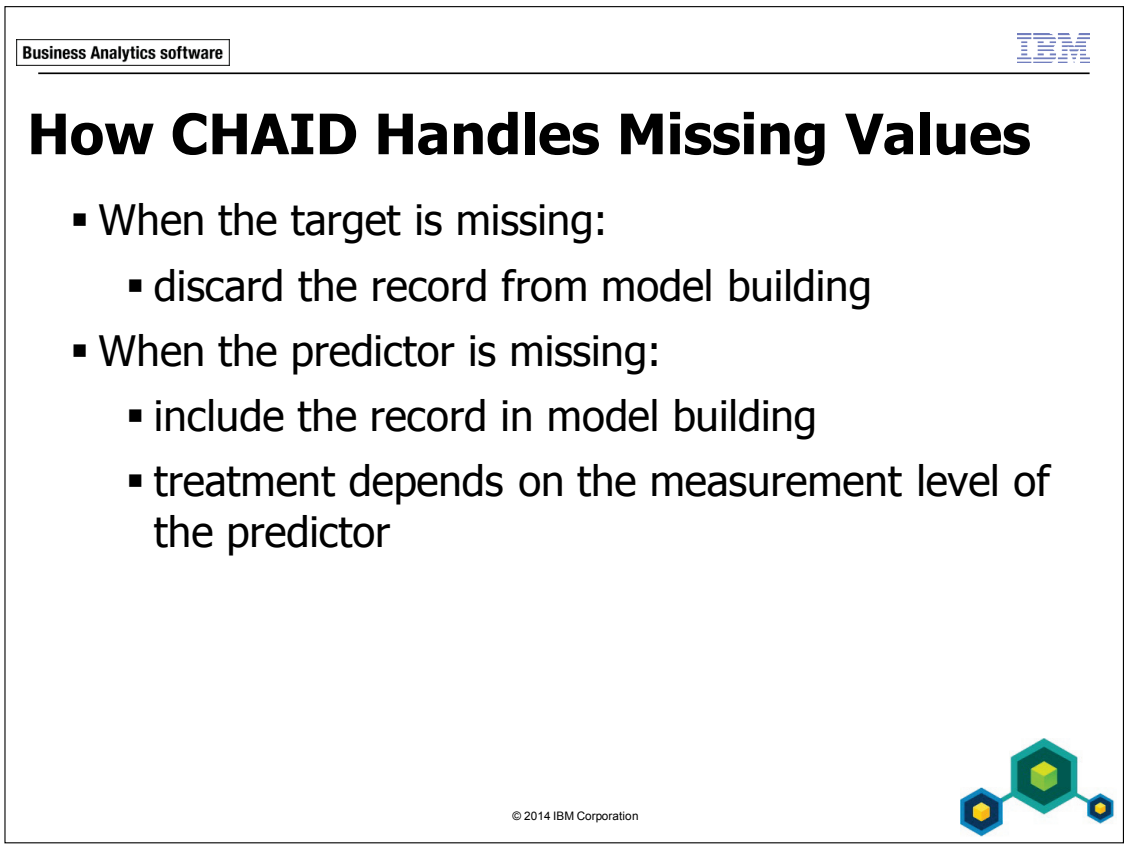
Technical note on the Chi-square probability.

A correction is made to the probability value, according to the number of pairwise comparisons that are made.

To protect against finding significant results too easily, the threshold specified for statistical significance (the so-called alpha level) is adjusted downward by dividing alpha by the number of comparisons performed. This is known as the Bonferroni adjustment. This is the reason why a split on a predictor in a tree will read "Adj. probability" rather than "probability".

Applying the Bonferroni adjustment is an option in CHAID that is on by default. You should normally leave this option turned on. In small samples or with only a few predictors you could turn it off to increase the power of the analysis (find splits, where no splits would have been found with Bonferroni adjustments).

The number of tests is determined by the number of categories, so that the Bonferroni adjustment increases with each additional category of a predictor. This means that predictor s with more categories are less likely to be selected than predictors with fewer categories.

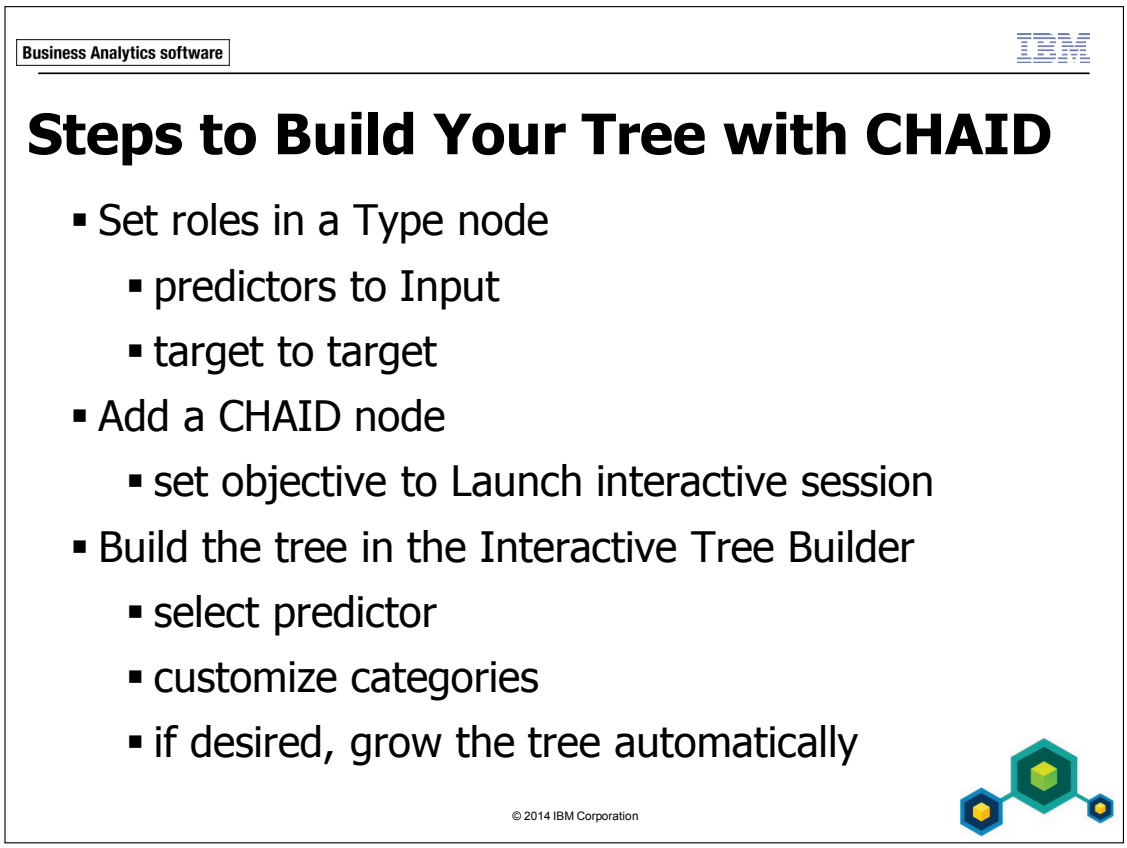


If the target is missing (user-defined blank or undefined ($null$)), the record is ignored in model building. Records with a missing value on a predictor are considered to be a separate category and are treated as any other category for that predictor.

Technical note on missing values on predictors.

How CHAID handles missing values on a predictor depends on the measurement level of the predictor:

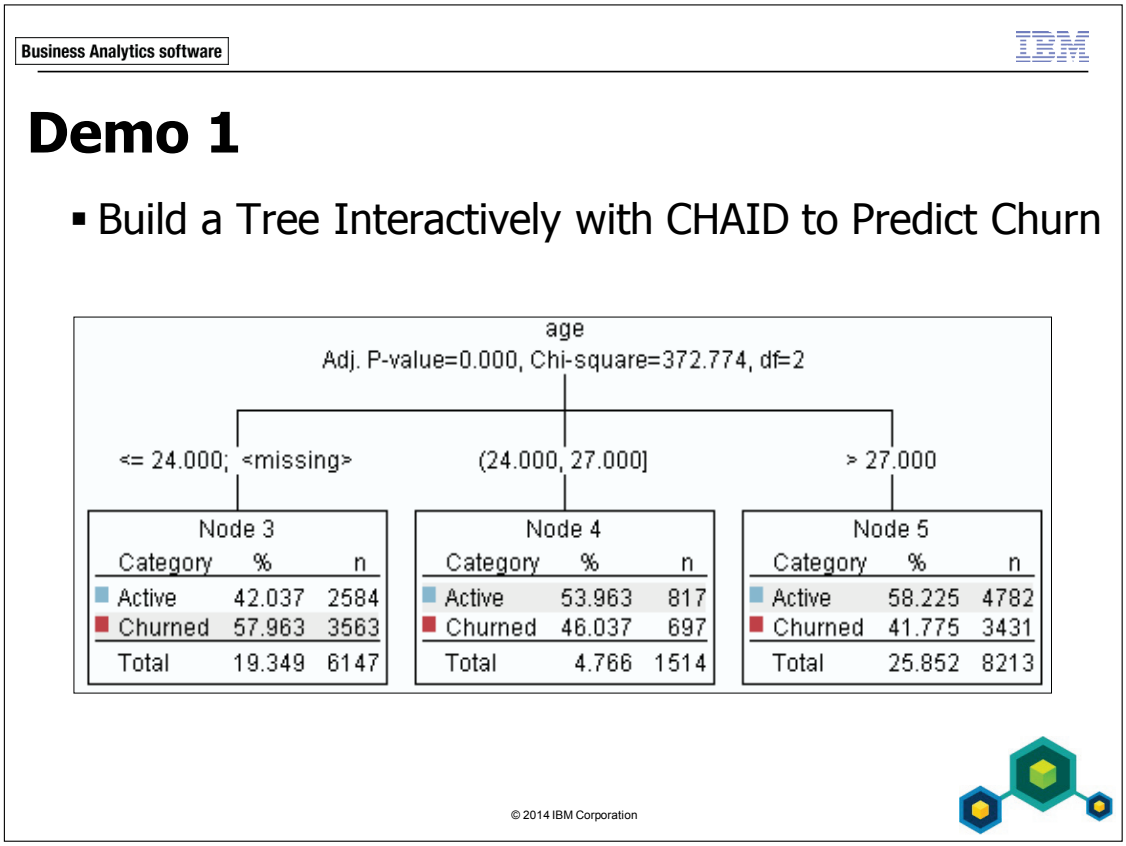
- Nominal predictor: Missing values become their own category and the missing category is treated the same as other categories in the analysis.
- Ordinal predictor: the algorithm first generates the best set of categories using all non-missing information. Then the algorithm identifies the category that is most similar to the missing category. Finally, two (adjusted) p-values are calculated: one for the set of categories formed by merging the missing category with its most similar category, and the other for the set of categories formed by adding the missing category as a separate category. The set of categories with the smallest p-value is used.



To build your tree using the CHAID algorithm, set the role for the fields in a Type node. Then add a CHAID node (Modeling palette, Classification item) downstream from the Type node.

By default, CHAID will grow a tree directly and will generate a model nugget for it. To build a tree interactively, set the objective to Launch interactive session on the Objective item on the Options tab in the CHAID dialog box.

The Interactive Tree Builder window will open when you execute the CHAID node, displaying the root of the tree. You can build stepwise from the root, select your predictor in each step and specify how categories should be merged or separated. The predictor that you select does not necessarily have to be significant to include it in the tree. Interactively building your tree means that your business knowledge is leading tree growth, not the CHAID algorithm. If desired, you can use the CHAID algorithm at any point to grow the tree automatically. You can always delete branches from the tree and rebuild the tree with other predictors. All in all, you can build the tree entirely according to your preferences.



The following (synthetic) file coming from a (fictitious) telecommunications firm is used to demonstrate how you build your tree interactively with CHAID:

- **telco x modeling data.txt**: Information on approximately 32,000 customers of the firm. The data includes demographics and calling minutes, as well as churn status. Churn status is stored in a field named churn. The values for the churn field can be either Active for current customers or Churned for churned customers. The file is located in **C:\Train\0A0U5**.

Before you begin the demo, ensure that:

- You have started MODELER (click **Cancel** when the splash screen appears when you start MODELER).
- You have set MODELER's working folder. (In MODELER, select **File\Set Directory**. When you are working in an IBM training environment, browse to the folder **C:\Train\0A0U5** and then click **Set** to select it. When you are not working in an IBM training environment, ask your trainer for the folder name.)

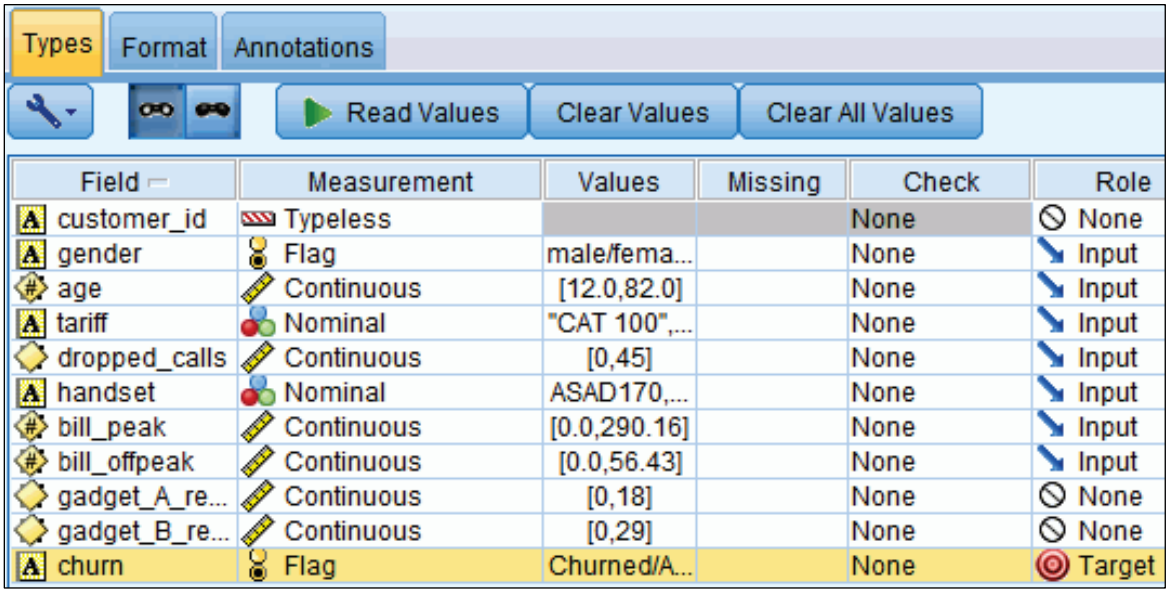
Demo 1: Build a Tree Interactively with CHAID to Predict Churn

Purpose:
You will build a customized tree to predict churn. Therefore you will grow the tree step-by-step, selecting predictors and merging or separating categories as you require.

Task 1. Import and instantiate the data.

1. From the **Sources** palette, double-click the **Var. File** node to add it to the stream canvas.
2. Edit the **Var. File** node, and then:
   - to the right of the **File** box, click **Browse**  (the Browse window should automatically open to the **C:\Train\0A0U5** folder**)**
   - select **telco x modeling data.txt** and then click **Open**
   - close the **Var. File** dialog box
3. From the **Field Ops** palette, add a **Type** node downstream from the **Var. File** node.
4. Edit the **Type** node, and then:
   - click **Read Values** to instantiate the data
   - ensure the **Role** for **gender**, **age**, **tariff**, **dropped_calls**, **handset**, **bill_peak**, and **bill_offpeak** to **Input**
   - set the **Role** for **churn** to **Target**
   - set the **Role** for the other fields to **None**

A section of the results appear as follows:



Field	Measurement	Values	Missing	Check	Role
customer_id	Typeless			None	None
gender	Flag	male/fema...		None	Input
age	Continuous	[12.0,82.0]		None	Input
tariff	Nominal	"CAT 100",...		None	Input
dropped_calls	Continuous	[0,45]		None	Input
handset	Nominal	ASAD170,...		None	Input
bill_peak	Continuous	[0.0,290.16]		None	Input
bill_offpeak	Continuous	[0.0,56.43]		None	Input
gadget_A_re...	Continuous	[0,18]		None	None
gadget_B_re...	Continuous	[0,29]		None	None
churn	Flag	Churned/A...		None	Target

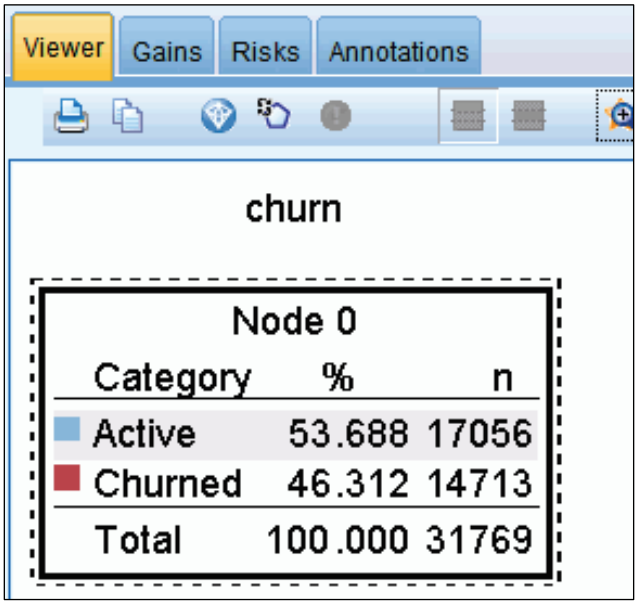
5. Close the **Type** dialog box.

Task 2. Add and configure the CHAID node.

1. Click the **Modeling** palette, select the **Classification** item, and then add a **CHAID** node downstream from the **Type** node.
2. Edit the **CHAID** node, and then:
   - click the **Build Options** tab
   - select the **Objective** item and, in the **Build a single tree** pane, select the **Launch interactive session** option
   - select the **Advanced** item, and then set the **Overfit prevention set(%)** to **0**

   Note: if an Overfit prevention set is in use, CHAID internally separates records into a training set on which the model is built, and a testing set, which is an independent set of records used to examine how the model performs on never seen data. In this course, no overfit prevention set will be used, but, if required, a Partition node will be used to examine the model on a testing set.
   - click **Run**

The Interactive Tree Builder window opens. A section of the results appear as follows:

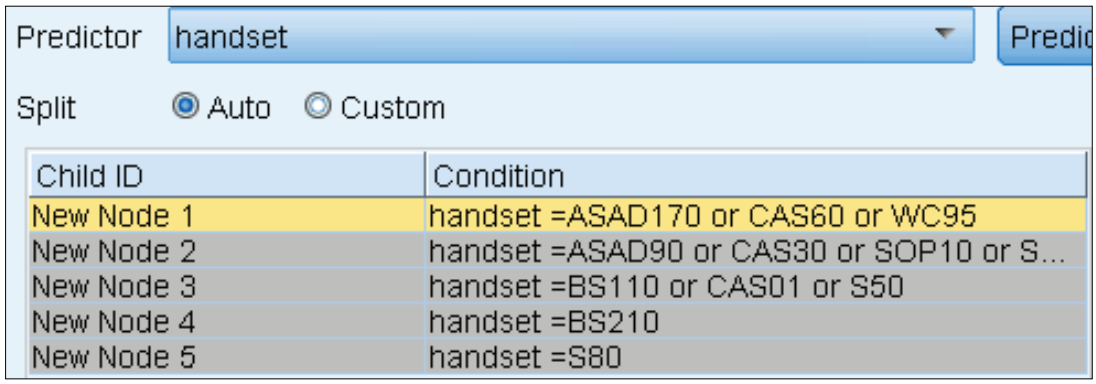


The Interactive Tree Builder window displays the percentage Active and Churned for all records, in the root node.

You will examine which predictor is the most important one, in terms of statistical significance.

3. From the **Tree** menu, click **Grow Branch with Custom Split** (alternatively, click the **Grow Branch with Custom Split** button).

The results appear as follows:



The handset field gives the statistically best split. Its categories were merged into five new nodes, by the stepwise process of merging categories.

You will examine the significance for the other predictors, and select a different predictor from handset to grow the tree with.

4. Click the **Predictors** Predictors... button.

The results appear as follows:

Select Predictor

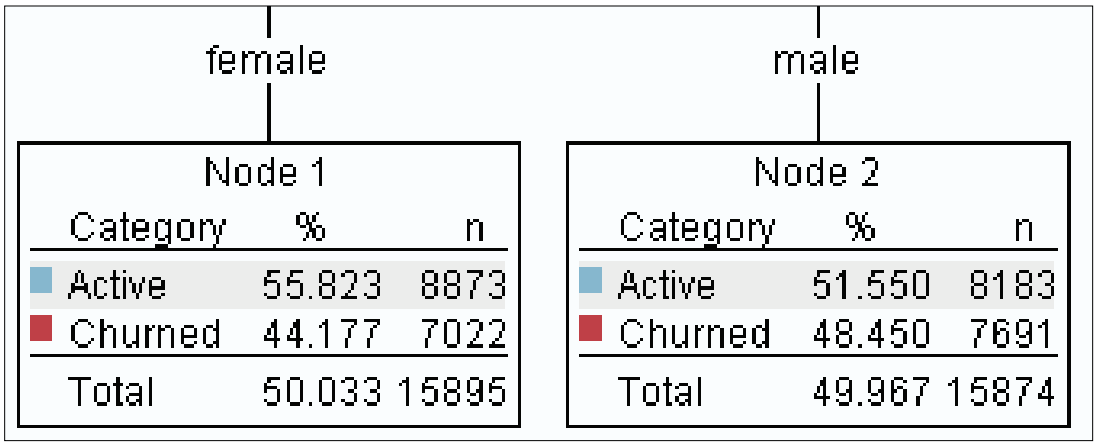
Predictor	Nodes	Statistic	DF	Adj. Prob.
handset	5	Chi-square=13938.944	4	0.000
dropped_calls	2	Chi-square=1462.862	1	0.000
age	3	Chi-square=830.375	2	0.000
bill_peak	4	Chi-square=658.309	3	0.000
tariff	4	Chi-square=501.399	3	0.000
bill_offpeak	5	Chi-square=166.624	4	0.000
gender	2	Chi-square=58.319	1	0.000

The Select Predictor window displays the number of nodes in which the field would be split, some technical details about the algorithm, such as the Chi-square statistic and the degrees of freedom (DF), and the adjusted probability for the split.

You will grow the tree with gender as split field.

5. Click **Cancel** to close the **Select Predictor** window.
6. In the **Define Split** window, from the **Predictor** list, click **gender**, and then click **Grow**.

A section of the results appear as follows:



Men show a higher percentage churn. Although the difference is moderate, it is highly significant because of the sample size of 31,769 records.

Note: You will notice a red symbol under the root node. This indicates that the node was grown with a custom split.

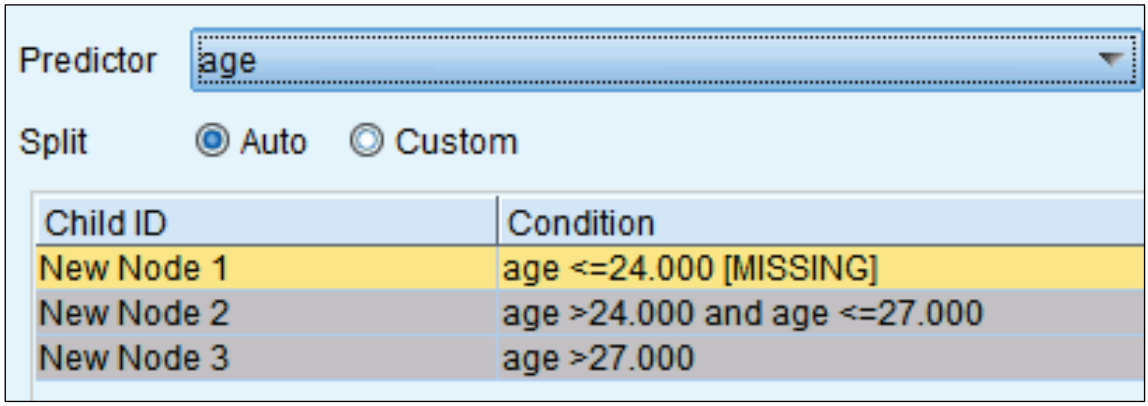
You will grow the tree for the male group, again using a custom split.

7. Click **Node 2** to select the **male** branch, and then select **Tree\Grow Branch with Custom Split** (alternatively, click the **Grow Branch with Custom Split**  button).

   You will further grow the male branch with age.

8. From the **Predictor** list, select **age**.

   The results appear as follows:

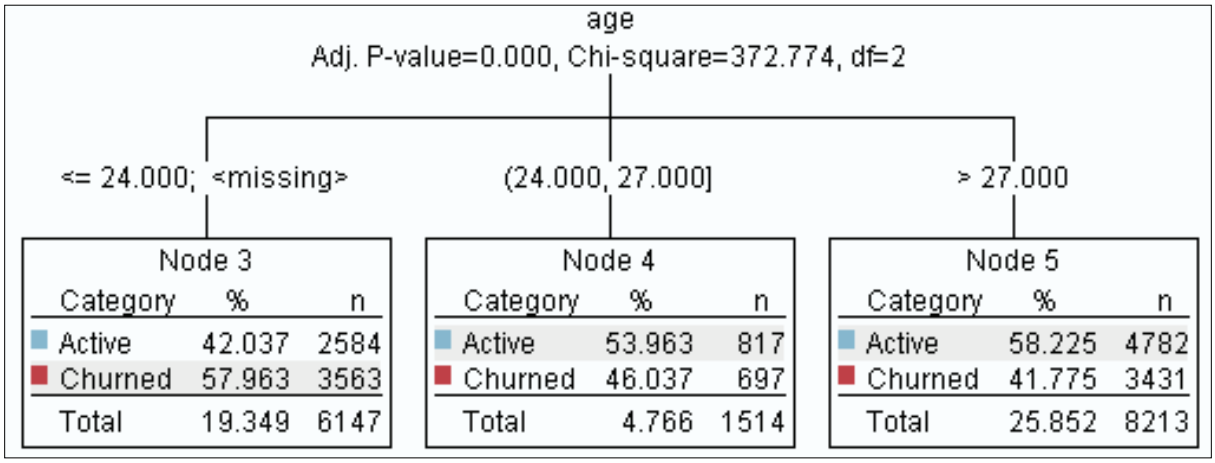


The age field is split into three age categories. Actually, the continuous field was binned into deciles first, and then adjacent categories that did not show a significant difference in percentage churn were merged.

Men with age missing are in the same node as age <= 24 year men. Apparently, men having a missing value on age and young men have a similar churn rate.

9. Click **Grow** to grow the tree with **age**, for the **male** node.

   A section of the results appear as follows:



   Notice the difference between the (and], for excluded and included values, respectively. For example, a 27 year old man is in Node 4.

   The female branch must be split on age also, into the same age categories, but you will assign the missing values to a separate category.

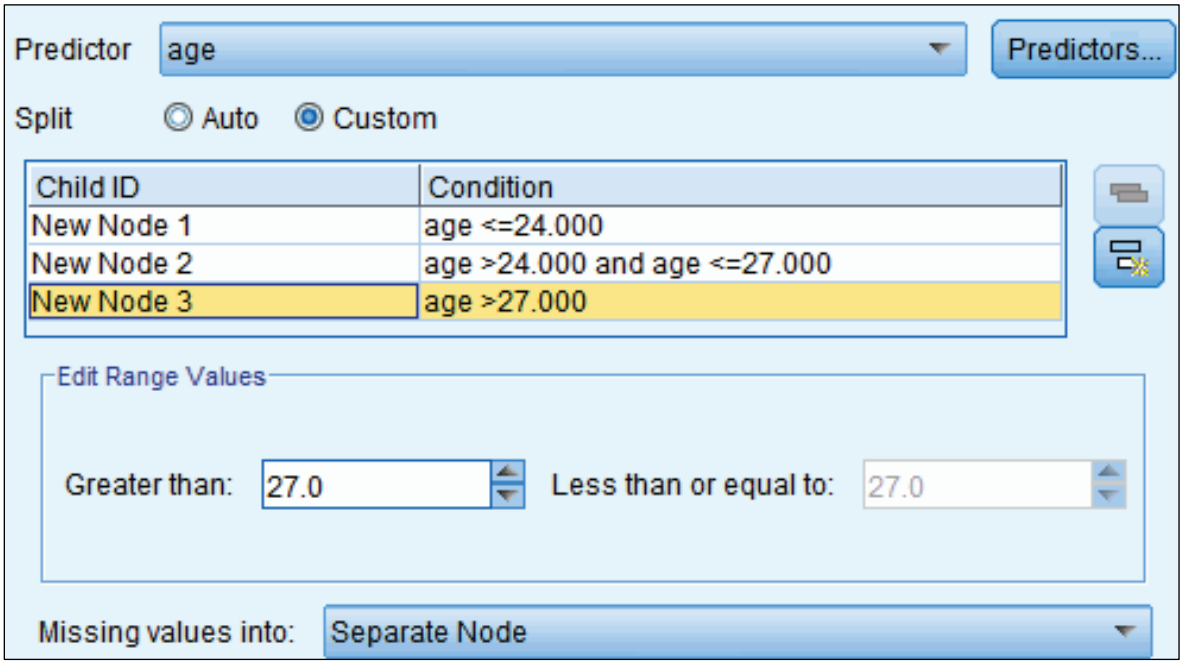
10. Click **Node 1** to select the **female** branch, select **Tree\Grow Branch with Custom Split** (or click the **Grow Branch with Custom Split** button), and then select **age** from the **Predictor** list.

11. For **Split**, select the **Custom** option, from the **Child ID** list, select **New Node 3** and **New Node 4** (use the Ctrl-key for a multiple selection), and then click the **Group values** button.

    New Nodes 3 and 4 are combined into New Node 3.

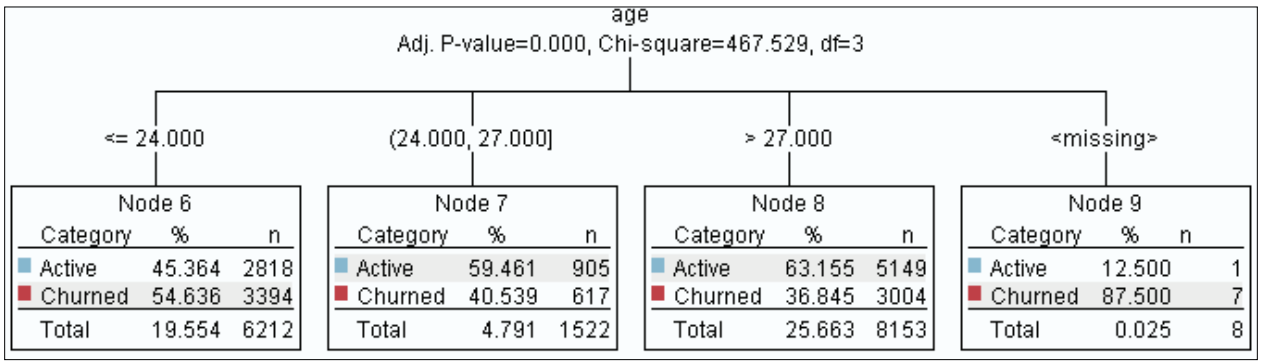
12. From the **Missing values into** list, select **Separate Node**.

    A section of the results appear as follows:



13. Click **Grow**.

    A section of the results appear as follows:



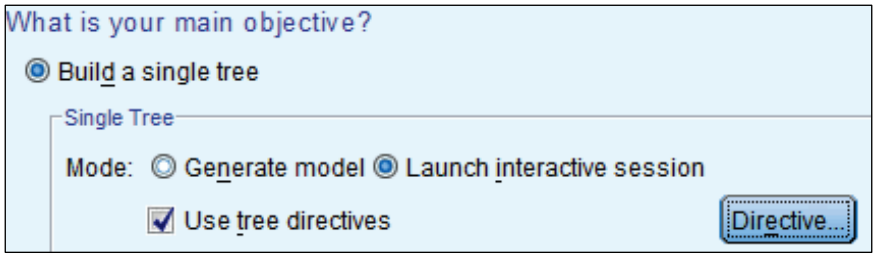
The highest churn rate for women is found in node 9, the missing category which only contains eight records.

Note: You can navigate to the portion of the tree that you wish to examine by using the tree map window. The **Show or hide the tree map window**  button toggles between the two modes.

You will start the next demo with the tree that you have created at this point. To preserve your results from the interactive tree-building session, you will save the directives used to generate the current tree. These directives are written back into the CHAID node. By re-running the CHAID node that stores the directives, you will regenerate the tree in its current state.

14. Select **File\Update directives**.
15. Select **File\Close** to close the **Interactive Tree Builder** window.
16. Edit the **CHAID** node, click the **Build Options** tab, and then select the **Objective** item.

    A section of the results appear as follows:

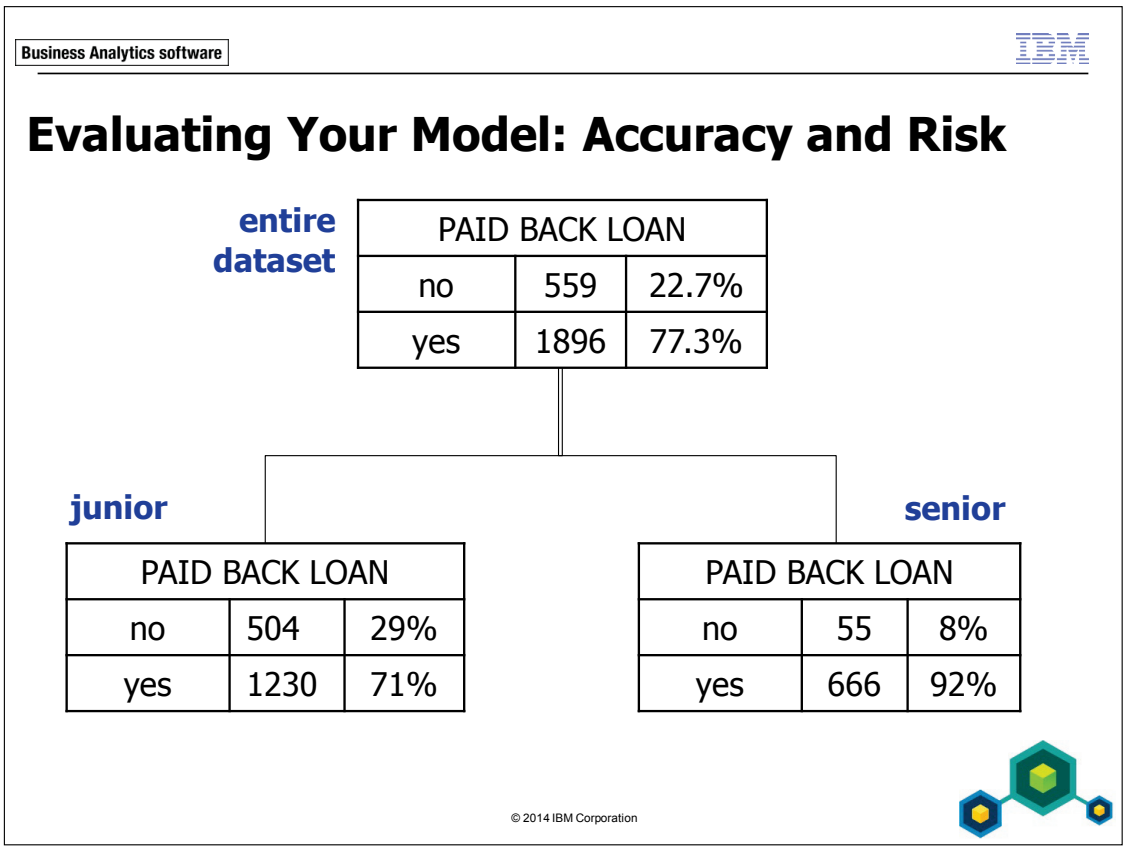


    The next time that you run the CHAID node the tree directives will generate the tree that you just built in the interactive session. The directives can be viewed when you click the **Directive** button.

17. Close the **CHAID** dialog box.

    Leave the stream open for the next demo.

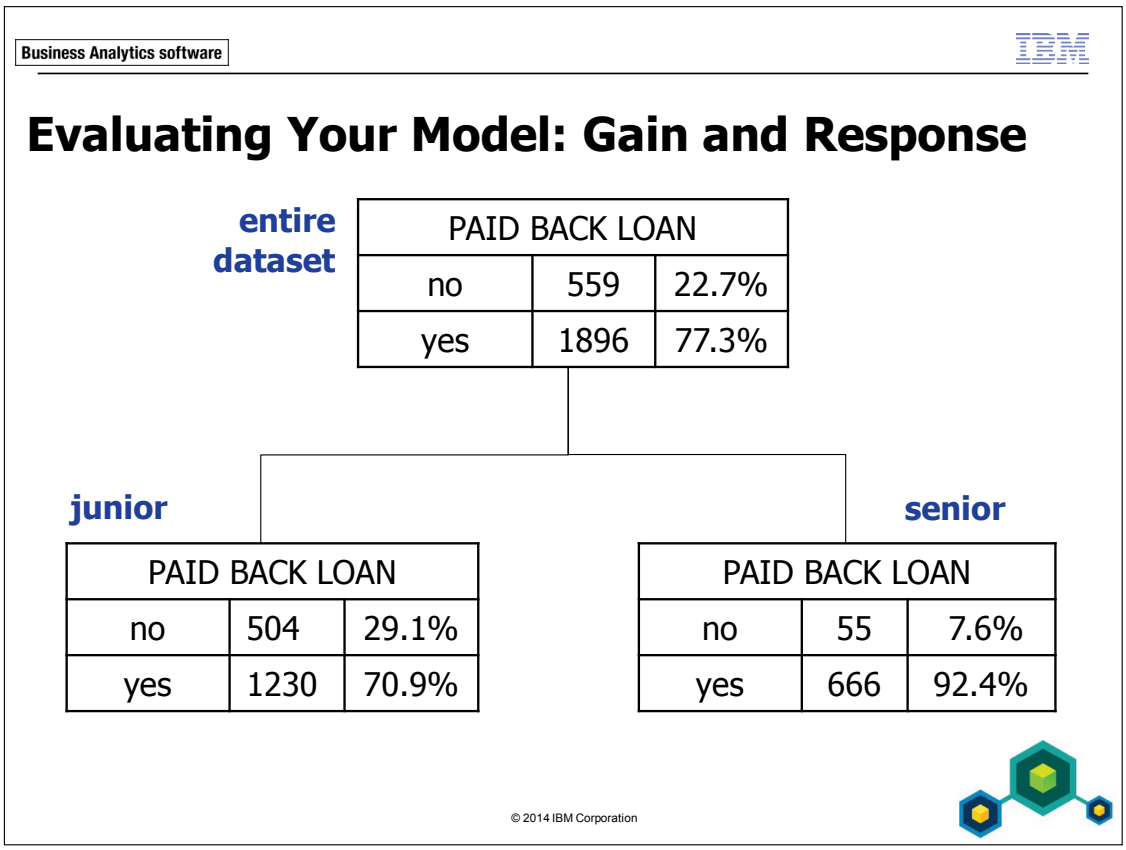
Results:
You have grown a tree interactively, building the tree according to your preferences.



The terminal nodes make up the model. A model predicts a value for a target, given the values on the predictors. In a tree, the predicted value is the value that occurs most often in the node to where the record belongs, because then you will have the smallest error in prediction.

This slide shows a tree that is comprised of only one flag predictor. The tree has two terminal nodes making up the model, the junior node and the senior node. When the value for PAID BACK LOAN must be predicted given that the person is junior, the prediction that leads to the smallest prediction error indicates that the person will pay back the loan. Likewise, when you know that the person is senior, you will predict that the person will pay back the loan. Your prediction is correct in (1230 + 666) / (504 + 1230 + 55 + 666), or (1896/2455) * 100 = 77.2% of all records. Your prediction is incorrect in (504 + 55) /2455 = 22.7% of all records.

The percentage of records that is predicted correctly is known as accuracy (in this example 77.2%). The percentage of misclassified records, reported as a proportion, is known as risk estimate (in this example 22.7 / 100 = .227).



Accuracy and its counterpart, risk, are two statistics that assess the fit of the entire tree. You can also examine the information that is conveyed in each terminal node. One statistic, the response percentage, is the number of hits in a terminal node as a percentage of number of records in that node. Another statistic, the gain percentage, is the percentage of hits in a terminal node in relation to the number of hits in the entire dataset.

The tree depicted on this slide has two terminal nodes. Suppose that the focus is on those that did not pay back the loan. Thus, not paying back the loan is defined as a hit.

In total there were 559 persons who did not pay back the loan, or 559 hits. For the junior group, the response percentage equals (504 / (504 + 1230)) = 29.1%. For the seniors it is 7.6%. The node of juniors includes 504 persons that did not pay back the loan, which is (504/559) * 100 = 90.2% of all hits. Thus, the gain percentage is 90.2% for the junior group. The node of seniors captures (55/559) * 100 = 9.8% of all hits, so this node's gain is 9.9%.

Business Analytics software

Gain and Response Illustrated: Tables

overview by node

NODE	NODE: N	NODE (%)	GAIN: N	GAIN (%)	RESPONSE (%)
junior	1734	70.6%	504	90.2%	29.1%
senior	721	29.4%	55	9.8%	7.6%

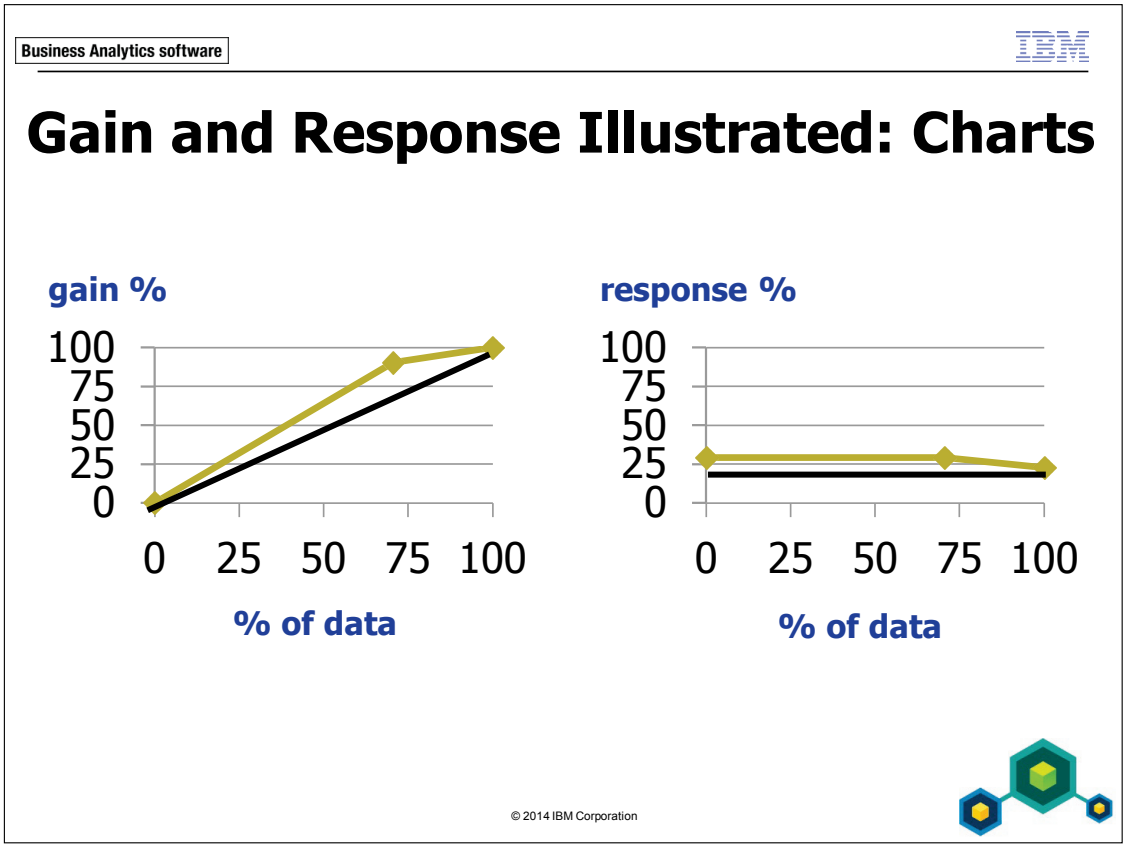
cumulative overview by node

NODE	NODE: N	NODE (%)	GAIN: N	GAIN (%)	RESPONSE (%)
junior	1734	70.6%	504	90.2%	29.1%
junior, senior	2455	100%	55	100%%	22.7%

The results can be summarized in various tables. The upper table depicted on this slide reports the gain and the response percentage for each node. Alternatively, you can request the cumulative statistics, as displayed in the lower table. The measures for the first group do not change, but the following row refers to both the junior and senior group (and brings Node % and Gain % to 100%).

Rather than report the cumulatives by node, you can the report the cumulatives by percentiles (steps of 1%), or deciles (steps of 10%), amongst others. For example, when you select the best 10%, or about 246 records from the 2,455 records, these records would all come from the junior group. In this group you expect a response percentage of 29.1, so you expect 0.291 * 246 = 72 hits, which is (72/559) * 100 = 12.8% of all hits, so the gain is 12.8%.

Some analysts are looking for a model that captures 80% of the hits in the best 20% of the data, but this may be a hard requirement to meet.



Gain and response can also be plotted in charts. This slide presents a gain chart on the left and a response chart on the right.

The horizontal axis in both charts represents the percentage of the records selected, the vertical axis the cumulative gain percentage of hits (gain chart) or cumulative response percentage (response chart).

In the gain chart, the diagonal line represents the base rate, the expected gain percentage if the model is predicting the target at random. For example, if 50% of all records are selected at random then it is expected that 50% of all hits are captured. The upper line displays the model results.

The advantage of the model is reflected in the degree to which the model-based line exceeds the base-rate line. If the model line is steep for early percentiles, relative to the base rate, then the hits tend to concentrate in those percentile groups of data. Typically you will look for a "kink" in the line where it either begins running parallel to the base rate line (representing nodes that are no better at identifying hits than random selection) or it tilts downward relative to the base rate line (representing nodes where one is less likely to find hits compared to random selection).

At the practical level, this would mean that many of the hits can be found within a small portion of the data. For example, this may imply that in a database marketing application only that portion of the customers should be mailed, which will save money without losing response.

The horizontal line in the response chart represents the overall response percentage. Again, the more the model line differs from the base rate, the better the model.

Business Analytics software

IBM

Scoring Records

- Select Generate\Generate Model in the Interactive Tree window
- Add the generated model nugget to your stream

fields added to your data

ID	AGE CATEGORY	PAID BACK LOAN	$R-PAID BACK LOAN	$RC-PAID BACK LOAN
1	junior	no	yes	0.71
2	senior	yes	yes	0.92
3	senior	yes	yes	0.92

When you find the quality of the model to be satisfactory, you can use the tree to score records by generating a model nugget for the tree and including the model nugget in your stream. The generated model nugget will add two fields to your dataset:

- The predicted category: the value that occurs most often in the terminal node to which the record belongs.
- The confidence: the confidence that you have that the predicted category is correct.

Continuing with the example of the tree with target PAID BACK LOAN, split on AGE CATEGORY, this means that the predicted category for a junior is yes. The confidence that you can have that this prediction is correct is 0.71. A senior is predicted to pay back his or her loan with a confidence of 0.92.

Rather than thinking of confidences, think of probabilities. The probability that a junior will pay back the loan is 0.71, the probability that a senior will pay back the loan is 0.92.

Business Analytics software

Scoring Records: Propensities

ID	X	PAID BACK LOAN	$R-PAID BACK LOAN	$RC-PAID BACK LOAN	$RRP-PAID BACK LOAN (for "yes")
					propensity
1	A	no	yes	0.71	0.71
2	B	yes	yes	0.92	0.92
3	B	yes	yes	0.92	0.92
4	**C**	**no**	**no**	**0.8**	**0.2**

The problem with confidence scores is that these apply to the category that was predicted. In cases where the prediction is the false category, a high confidence actually means a low likelihood for the true category.

To overcome this problem and to allow comparison across records, propensity scores, or propensities, are used. Propensities indicate the likelihood for the true value. Propensities are used for flag targets only.

The table on this slide gives an example for a hypothetical tree with a predictor X having three categories A, B and C. The target field is PAID BACK LOAN, with categories yes and no, with yes being the true value for PAID BACK LOAN.

- A: The predicted category is yes. The confidence equals 0.71, which is also the propensity because the predicted category is the true value.
- B: The predicted category is yes and the confidence and propensity equal 0.92.
- C: The predicted category is no, with a confidence of 0.8. The propensity is the probability for the true value, 1 - 0.8 = 0.2.

Business Analytics software

Demo 2

- Evaluate Your Tree to Predict Churn and Score Records

Table | Annotations

	churn	$R-churn	$RC-churn	$RRP-churn
1	Churned	Active	0.582	0.418
2	Churned	Active	0.540	0.460
3	Churned	Active	0.582	0.418
4	Churned	Active	0.582	0.418
5	Churned	Active	0.582	0.418
6	Churned	Active	0.582	0.418
7	Churned	Active	0.582	0.418
8	Churned	Active	0.540	0.460
9	Churned	Active	0.582	0.418
10	Churned	Active	0.582	0.418

This demo builds from the stream that you have created in the first demo.

Demo 2: Evaluate Your Tree to Predict Churn and Score Records

Purpose:
You will evaluate the tree that you have built interactively and, assuming that the fit is satisfactory, you will score records.

Task 1. Evaluate the tree in terms of gain and risk.

In this demo you will build from the stream that you created in the previous demo. If you do not have the end result of the previous demo, open **demo_building_your_tree_interactively_with_chaid_completed.str**, located in the **02-Building_Your_Tree_Interactively_with_CHAID\Solutions** sub folder.

1. Run the **CHAID** node named **churn**.

   The Interactive Tree Builder window opens, displaying the model as defined by the directives.

2. Click the **Gains** tab, and from the **Target category** 

list, click **Churned**.

   A section of the results appear as follows:

Tree Growing Set

Nodes	Node: n	Node (%)	Gain: n	Gain (%)	Response (%)	Index (%)
9	8.00	0.03	7.00	0.05	87.50	188.93
3	6147.00	19.35	3563.00	24.22	57.96	125.16
6	6212.00	19.55	3394.00	23.07	54.64	117.97
4	1514.00	4.77	697.00	4.74	46.04	99.41
5	8213.00	25.85	3431.00	23.32	41.78	90.20
7	1522.00	4.79	617.00	4.19	40.54	87.53
8	8153.00	25.66	3004.00	20.42	36.85	79.56

Node 9, with 8 records (0.03% of the total number of records), has 7 churners in it. The response percentage is (7/8) * 100 = 87.5%, which is 1.8893 times the overall response percentage of 46.31%, or an index of 188.93. It captures 7 of the 14,713 churners, a gain of (7/*14713) * 100, or 0.05%.

This node, although having the highest probability to churn is only a small segment of the data and numerically contributes very few at all. Thus, it is advised to walk through this table to identify the segments that contribute the most churners numerically, not just the ones that had the highest percentage of churners. Here, node 3 is the node that contributes most to the churners.

3. Click the **Cumulative**  button.

   A section of the results appear as follows:

Tree Growing Set

Nodes	Node: n	Node (%)	Gain: n	Gain (%)	Response (%)	Index (%)
9	8.00	0.03	7.00	0.05	87.50	188.93
3	6155.00	19.37	3570.00	24.26	58.00	125.24
6	12367.00	38.93	6964.00	47.33	56.31	121.59
4	13881.00	43.69	7661.00	52.07	55.19	119.17
5	22094.00	69.55	11092.00	75.39	50.20	108.40
7	23616.00	74.34	11709.00	79.58	49.58	107.06
8	31769.00	100.00	14713.00	100.00	46.31	100.00

   The top three nodes contain 38.93% of the data, with a response percentage of (6964/12367) *100 = 56.31%. These top three nodes capture (6964/14713) * 100 = 47.33% of all churners.

   Rather than having the cumulatives by node, you will examine the cumulatives by decile. This answers a question such as: What percentage of churners is captured when the top 10% is selected?

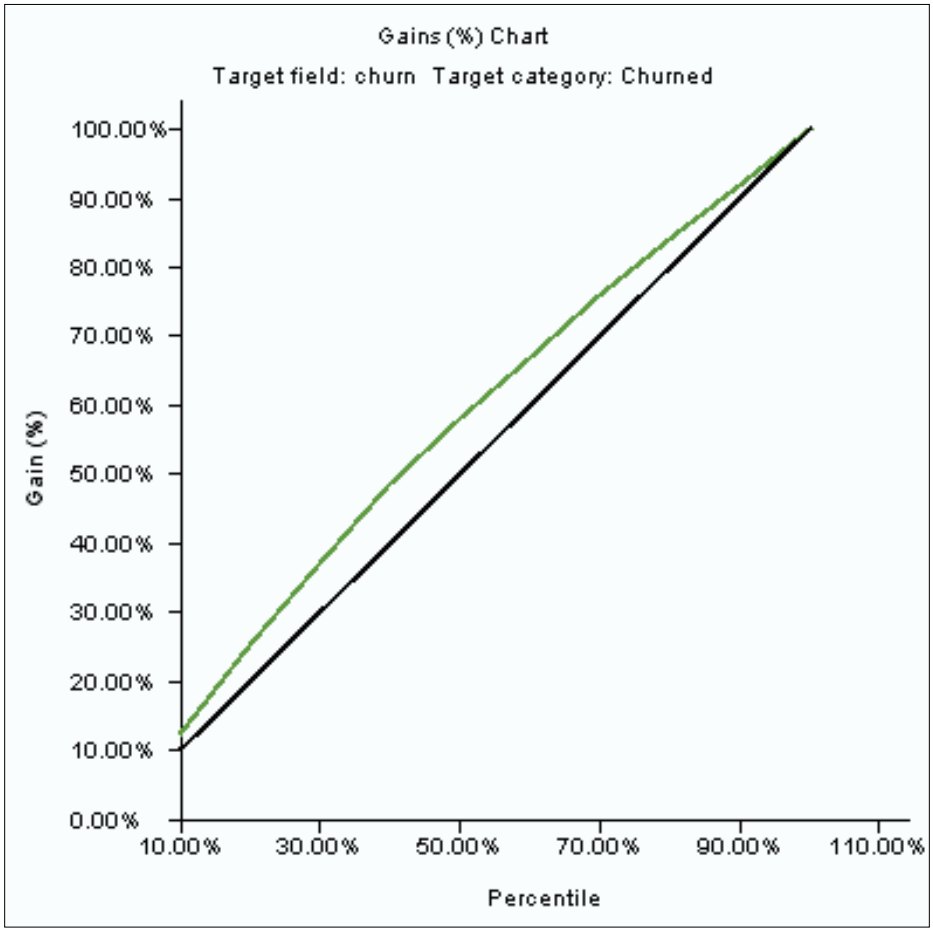
4. Click the **Quantiles**  button, and then from the list, select **Decile**.

   The top 10%, coming from nodes 9 and 3, includes 12.53% of the churners.

   You will also examine the results in a chart.

5. Click the **Chart** button.

A section of the results appear as follows:



The gain for the model does not deviate much from the random model, which could be expected because the model was built interactively and is only two levels deep.

You will examine the accuracy of the model to evaluate the model.

6. Click the **Risks** tab.

   The results appear as follows:

Misclassification Matrix

Risk Estimate

0.414

Standard Error

0.003

Predicted

Actual	Active	Churned	Total
Active	11653	5403	17056
Churned	7749	6964	14713
Total	19402	12367	31769

The model predicts 11,653 + 6,964 = 18,617 records correctly, which is 58.6% of the 31,769 records. Or, viewed as a Risk Estimate, the proportion of misclassified values equals 1 - 0.586 =.414.

The Standard Error requires special attention. Had a different sample been drawn, and the same tree been built, the risk estimate would differ from the specific value in this dataset (0.414).

To get an idea how much the Risk Estimate varies from sample to sample, all of the size of 31,769 records, the Standard Error is reported. The smaller the standard error, the less variation there will be from sample to sample in the Risk Estimate.

In other words, the smaller the Standard Error, the more certainty there is about the Risk Estimate. The Standard Error is 0.003 here. This means, that you can expect a risk estimate of, say, 0.412 in another sample, but that it is very unlikely that the Risk Estimate is 0.300 in another sample.

Task 2. Score records using the interactively built model.

Suppose that the fit is satisfactory and that you wish to score records with this model.

1. Click the **Viewer** tab.
2. Select **Generate\Generate Model**.
3. In the **Generate New Model** window, ensure that the **Create node on: Canvas** option is selected, and then click **OK**.

   A model nugget is generated and placed on the stream canvas.

4. Select **File\Close** to close the **Interactive Tree Builder** window.
5. On the stream canvas, drag the **model nugget** named **churn1** to a position to the right of the **Type** node and connect them so that the **model nugget** is downstream from the **Type** node.
6. Edit the **model nugget** and then click the **Settings** tab.
7. Enable the **Calculate raw propensity scores** option.
8. Click **Preview** and then, in the **Preview** window, scroll all the way to the right.

   A section of the results appear as follows:

Table | Annotations

	churn	$R-churn	$RC-churn	$RRP-churn
1	Churned	Active	0.582	0.418
2	Churned	Active	0.540	0.460
3	Churned	Active	0.582	0.418
4	Churned	Active	0.582	0.418
5	Churned	Active	0.582	0.418
6	Churned	Active	0.582	0.418
7	Churned	Active	0.582	0.418
8	Churned	Active	0.540	0.460
9	Churned	Active	0.582	0.418
10	Churned	Active	0.582	0.418

The $R-churn field stores the predicted category and the $RC-churn field stores the confidence for the predicted value. The $RRP-churn field stores the probability for the true value of the churn field, which is Churned (you can verify this in the Type node). Thus, the propensity stores the probability to churn. Record #2 and record #8 have the highest probability to churn.

Notice that the first 8 records, those included in this preview, are all predicted incorrectly: these customers actually churned, yet the value for the prediction, $R-churn, is Active.

9. Close the **Preview** output window.
10. Close the **model nugget**.
11. Close the stream without saving anything.

Results:
You used evaluation measures to examine the fit of the model and you used the generated model nugget to add propensities to the dataset.

Note: You will find the solution results in **demo_building_your_tree_interactively_with_chaid_completed.str**, located in the **02-Building_Your_Tree_Interactively_with_CHAID\Solutions** sub folder.

Apply Your Knowledge

Use the questions in this section to test your knowledge of the course material.

Question 1: Consider the following table that reflects a matrix of PAID BACK LOAN by whether or not the customer lives in a city. True or false: The Chi-square value equals 0.007, which means that there is a significant relationship between living in a city and paying back a loan. (Note: use a threshold of 0.05 to declare a result significant).

LIVES IN A CITY

PAID BACK LOAN		**No**	**Yes**	**Total**
No	**Count**	**5**	**4**	**9**
	Column %	**15.625**	**14.815**	**15.254**
Yes	**Count**	**27**	**23**	**50**
	Column %	**84.375**	**85.185**	**84.746**
Total	**Count**	**32**	**27**	**59**
	Column %	**100**	**100**	**100**
Cells contain cross-tabulation of fields (including missing values) **Chi-square = 0.007, df=1, probability = 0.931**				

A. True

B. False

Question 2: Consider the following table that reflects the matrix of data collected on a sample of customers. Select all statements that apply.

GENDER

PAID BACK LOAN		**female**	**Male**	**Total**
No	**Count**	**16620**	**169204**	**335409**
	Column %	**22.557**	**22.983**	**22.770**
Yes	**Count**	**57060**	**56700**	**113760**
	Column %	**77.443**	**77.017**	**77.230**
Total	**Count**	**73680**	**73620**	**147300**
	Column %	**100**	**100**	**100**
Cells contain cross-tabulation of fields (including missing values) **Chi-square = 3.798, df=1, probability = 0.051**				

A. There is a statistically significant relationship between gender and paying back a loan. (Note: use a threshold of 0.05 to declare a result significant).

B. Of the customers that did not pay back the loan, 22.557% are female.

C. Of the females, 22.557% did not pay back the loan.

D. The Chi-square value and the significance do not change when the row field is placed in the column, and the column field in the row.

Question 3: Select all that apply.

A. The Chi-square test can be used to select a predictor to split a node.

B. The Chi-square test can be used to see if categories can be merged.

C. The Chi-square test can be applied to a categorical target that has more than two categories.

D. The Chi-square is sensitive to sample size (the number of records in the dataset).

Question 4: Consider the following table. True or false: the column Y gives the propensity scores.

Name	Predicted Category	X	Y
Joe Jensen	**T**	**0.8**	**0.8**
Deb Wilson	**T**	**0.7**	**0.7**
Jim Salcedo	**F**	**0.9**	**0.1**

A. True

B. False

Question 5: For CHAID, which of the following is the correct statement?

A. In CHAID interactive mode, the user can only select a significant predictor to split a node on.

B. The CHAID algorithm cannot be used to automate tree growth in interactive mode.

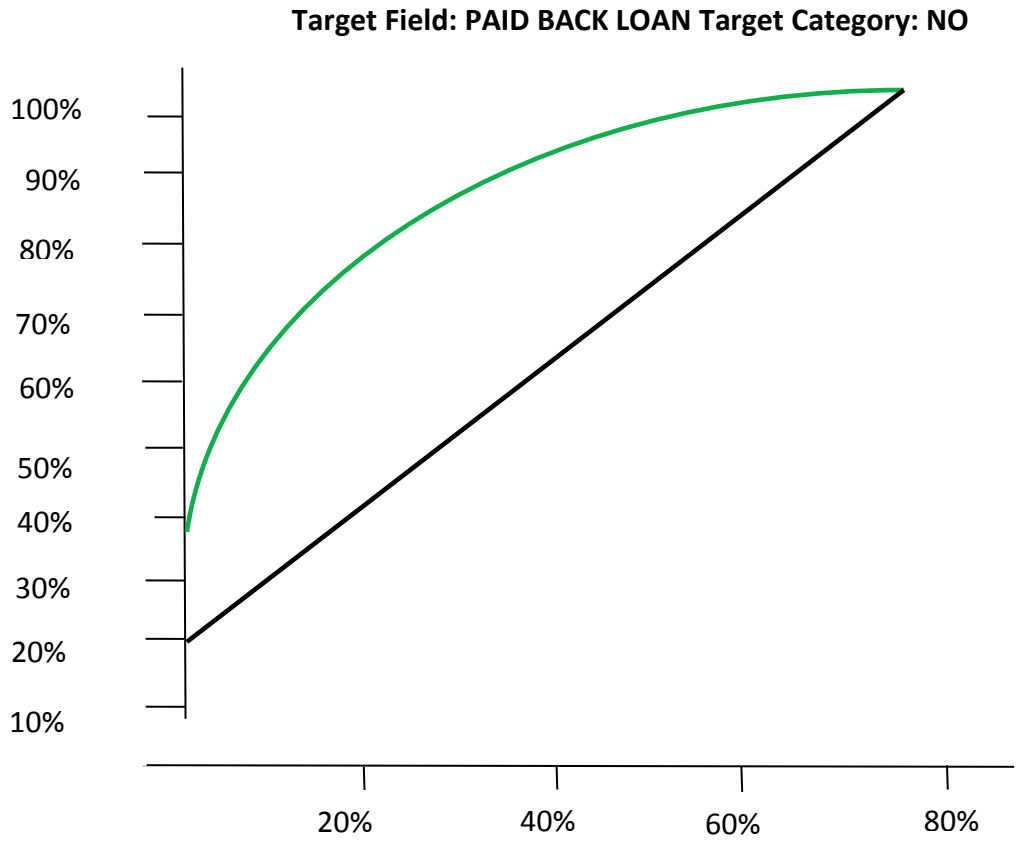
C. In CHAID interactive mode, you can select another statistic than the Chi-square statistic to grow your tree with.

D. None of the above statements are correct.

Question 6: The graph depicted below is an example of which type of chart?

A. Gain

B. Response



Question 7: Consider the table below, which gives the first 4 cumulative deciles of an analysis. True or false: The 10% records with the highest response percentages capture 5.79% of all hits.

A. True

B. False

Percentile	Percentile: n	Gain: n	Gain (%)	Response (%)
10.00	5500.00	318.00	24.43	5.79
20.00	11000.00	637.00	48.87	5.79
30.00	16500.00	820.00	62.93	4.97
40.00	22000.00	929.00	71.27	4.22

Question 8: Consider the table that follows. Which of the following is the correct statement?

A. The model always predicts N.

B. It is very likely that the Risk Estimate is 0.481 in another sample of this size.

C. The model is worthless because it always predicts no.

	Misclassification Matrix			
		Predicted		
Risk Estimate	Actual	N	Y	Total
0.024	N	53697	0	53697
Standard Error	Y	1303	0	1303
0.001	Total:	55000	0	55000

Answers to questions:

Your responses to the Apply Your Knowledge questions should appear as follows.

Answer 1: B. False. The Chi-square value equals 0.007, which is smaller than 0.05. However, what matters is the associated probability for this Chi-square value, and that probability is 0.931. This probability is greater than the threshold value of 0.05 which means that there is not a significant relationship between living in a city and a loan being paid back.

Answer 2: C, D. Using a threshold value for declaring a result significant of 0.05, there is not a statistically significant relationship between gender and loans paid because the probability value of 0.051 is greater than 0.05. The percentages in the table are column percentages, so they are based on GENDER. The Chi-square value and its probability do not change if the row field is placed in the column of the matrix and when the column field is placed in the row of the matrix.

Answer 3: A, B, C, and D. All statements are true.

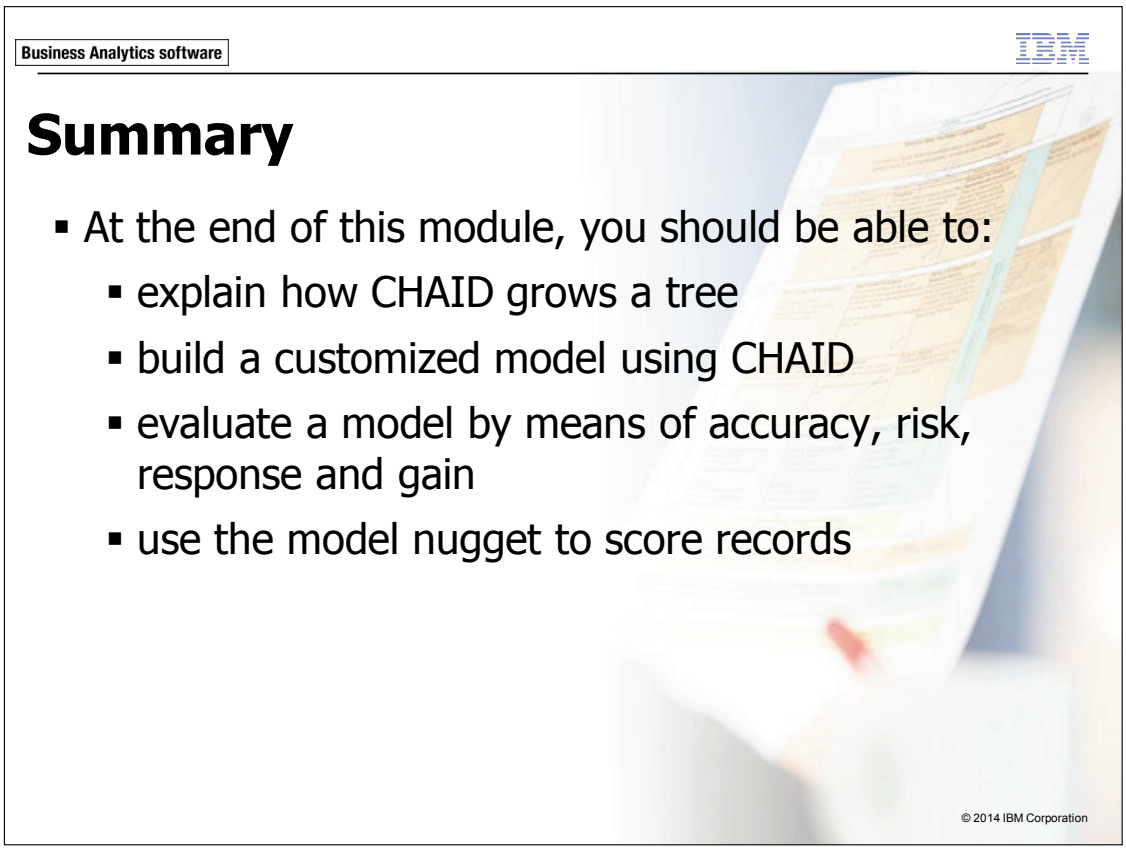
Answer 4: A. True. Propensities give the probability for the true value of the target, which is what the Y field stores.

Answer 5: D. None of the statements are correct. In CHAID interactive mode, the user is completely free to choose the predictor, including a non-significant one. Also, you can grow the tree at any point by using the CHAID algorithm. CHAID will only use the Chi-square test and you cannot select a different test.

Answer 6: A. The graph depicted is an example of a gain chart.

Answer 7: B. False. The best 10% records with the highest response percentage capture almost 24.43% of all hits (the Gain percentage).

Answer 8: A. The model always predicts N. This does necessarily mean that the model is useless. Therefore, you would have to examine the gain, because it could well be that the model identifies the hits in the first deciles.
It is very unlikely that the risk estimate in another sample is 0.481 given the standard error of 0.001.



In this module, details were provided on CHAID and commonly used evaluation measures. The key points to take away from this module are:

- CHAID is a popular model to use, because the statistical foundations are simple and the model (tree) provides insight.
- You can build the model (tree) interactively, giving, if required, business knowledge priority over statistics.
- To evaluate a model, use various evaluation measures. Using only one measure may not tell the whole story.
- The measures were discussed in the context of a CHAID model, but are generic measures to evaluate the fit of a model and can be used for any model that predicts a flag target.

In the end, what makes a model a "good enough" model, in terms of insight that it provides and performance in terms of evaluation measures, is a business decision.

Business Analytics software

IBM

Workshop 1

- Use CHAID Interactively to Predict Response to a Charity Promotion Campaign

Table | Annotations

	$R-response to campaign	$RC-response to campaign	$RRP-response to campaign
1	No	0.625	0.375
2	No	0.789	0.211
3	No	0.748	0.252
4	Yes	0.831	0.831
5	No	0.883	0.117

The following (synthetic) files are used in this workshop:

- **charity.txt**: A text file that represents data from a charity organization. It contains information on individuals who were mailed a promotion. The information includes whether the individuals responded to the campaign, their spending behavior with the charity and basic demographics such as age, gender and demographic group.
- **prospects for campaign.txt**: A text file that stores data for persons who have not been approached for the campaign, but are prospects to be approached for the campaign.

Both files are located in **C:\Train\0A0U5**.

Before you begin the workshop, ensure that:

- You have started MODELER (click **Cancel** when the splash screen appears when you start MODELER).
- You have set MODELER's working folder. (In MODELER, select **File\Set Directory**. When you are working in an IBM training environment, browse to the folder **C:\Train\0A0U5** and then click **Set** to select it. When you are not working in an IBM training environment, ask your trainer for the folder name.)

Workshop 1: Use CHAID Interactively to Predict Response to a Charity Promotion Campaign

In this workshop you will use CHAID interactively to build a model to predict the response to a promotion campaign. Once you have created a model, you will use the model to select prospects (not yet included in the campaign) that have a probability of at least 0.8 to respond positively to the campaign.

To predict the responses from the mail campaign, you must:

- Use a **Var. File** node (Sources palette) to import data from the text file **charity.txt** and examine the data with a **Data Audit** node.

  What is the percentage of customers responding positively to the campaign?

- Add a **Type** node downstream from the **Var. File** node. Configure the **Type** node so that **gender**, **age**, **mosaic bands**, **pre-campaign expenditure**, and **pre-campaign visits** are predictors for **response to campaign**. Also, instantiate the data (click the **Read Values** button).

- Add a **CHAID** node downstream from the **Type** node. Configure the **CHAID** node so that running the **CHAID** node will launch an interactive session (refer to the **Build Options** tab, **Objective** item) and all records will be used for model building (refer to the **Build Options** tab, **Advanced** item, **Overfit prevention set**).

- Run the **CHAID** node.

  What is the percentage that responded positively to the campaign?

- In the **Interactive Tree Builder** window, select **Tree\Grow Branch with Custom Split**.

  Which predictor is most important, in terms of statistical significance?

  Are all predictors significant (probability smaller than 0.05)?

- Split the root node into the following age categories:
  - age less than or equal to 41
  - age between 42 and 49
  - age greater than 49
- Split the nodes for the three age categories in the following way:
  - age less than or equal to 41: split into the three following pre-campaign expenditure categories:
    1. pre-campaign expenditure less than or equal to 92
    2. pre-campaign expenditure greater than 92, less than or equal to 103
    3. pre-campaign expenditure greater than 103
  - age between 42 and 49: same split as for the first age category (age up to 41)
  - age greater than 49: split on gender.

  Which of the eight terminal nodes has the highest response percentage?
- Report the gain by decile, both in a table and in a graph.

  What is the gain percentage for the best 20%?

  Note: Ensure that "Yes" is selected as the target category.
- What is the risk of misclassifying a record?

  The results of growing a customized tree are not impressive, so you will use the CHAID algorithm to build the tree.
- Remove the branch from the root node so that you have only the root node, and then grow the tree from the root node, using the CHAID algorithm.

  What is the gain percentage for the best 20% for this new tree?
- Generate a model nugget for the model that you just created.

- The charity organization has acquired a dataset with prospects. This dataset is called **prospects for campaign.txt**. Import the data from this dataset and use the model nugget to score this dataset.

  What is the probability to respond positively to the campaign for the first prospect?

Workshop 1: Tasks and Results

Task 1. Import and examine the data.

1. From the **Sources** palette, double-click the **Var. File** node to add it to the stream canvas.
2. Edit the **Var. File** node, and then:
   - to the right of the **File** box, click **Browse**  (the Browse window should automatically open to the **C:\Train\0A0U5 folder**)
   - select **charity.txt** and then click **Open**
   - close the **Var. File** dialog box
3. From the **Output** palette, add a **Data Audit** node downstream from the **Var. File** node, run the **Data Audit** node, and double-click the **Sample Graph** for the **response to campaign** field.

   The Distribution window shows that 31.32% responded positively to the campaign.
4. Close the **Distribution** window, and then close the **Data Audit** output window.

Task 2. Instantiate the data and set the roles for the fields.

1. From the **Field Ops** palette, add a **Type** node downstream from the **charity.txt** node.
2. Edit the **Type** node, and then:
   - click the **Read Values** 

     button

   The Values column is populated with values from the data.
   - set the **Role** for **gender**, **age**, **mosaic bands**, **pre-campaign expenditure**, and **pre-campaign visits** to **Input**
   - set the **Role** for **response to campaign** to **Target**
   - set the **Role** for the other fields to **None**
   - close the **Type** dialog box

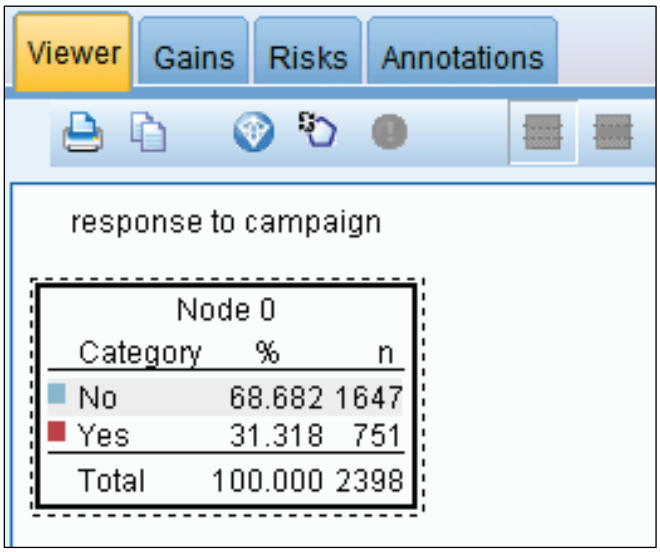
Task 3. Add and configure a CHAID node.

1. From the **Modeling** palette (**Classification** item), add a **CHAID** node downstream from the **Type** node.
2. Edit the **CHAID** node, and then:
   - click the **Build Options** tab
   - on the **Objective** item, select **Launch interactive session**
   - on the **Advanced** item, set **Overfit prevention set (%)** to **0**

Task 4. Run the CHAID node.

1. Click **Run**.

   The Interactive Tree Builder window opens, and the results appear as follows:



   About 31% responded positively to the campaign, which is in agreement with earlier findings.

Task 5. Select the predictors and split the root node.

1. Click **Tree\Grow Branch with Custom Split**, and then click the **Predictors** button.

   A section of the results appear as follows:

Select Predictor

Predictor	Nod...	Statistic	...	Adj. Prob.
pre-campaign visits	6	Chi-square=423.551	5	0.000
pre-campaign expenditure	4	Chi-square=246.641	3	0.000
age	4	Chi-square=46.834	3	0.000
gender	2	Chi-square=9.827	1	0.002
mosaic bands	2	Chi-square=23.153	1	0.003

   All predictors are significant, with pre-campaign visits as the most significant one.

Task 6. Select the predictors and split the root node.

1. In the **Select Predictor** window, click **age**, and then click **OK**.
2. In the **Define Split** window, next to **Split**, click **Custom**, select the first two nodes (use the Ctrl-key for a multiple selection), and then click the **Group value(s)** button.

   The results appear as follows:

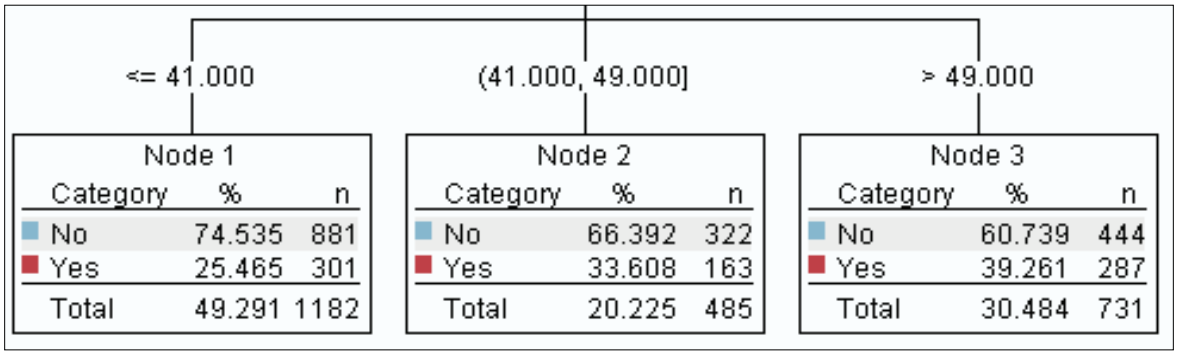
Predictor age

Split ◎ Auto ◉ Custom

Child ID	Condition
New Node 1	age <=41.000
New Node 2	age >41.000 and age <=49.000
New Node 3	age >49.000

3. Click **Grow**.

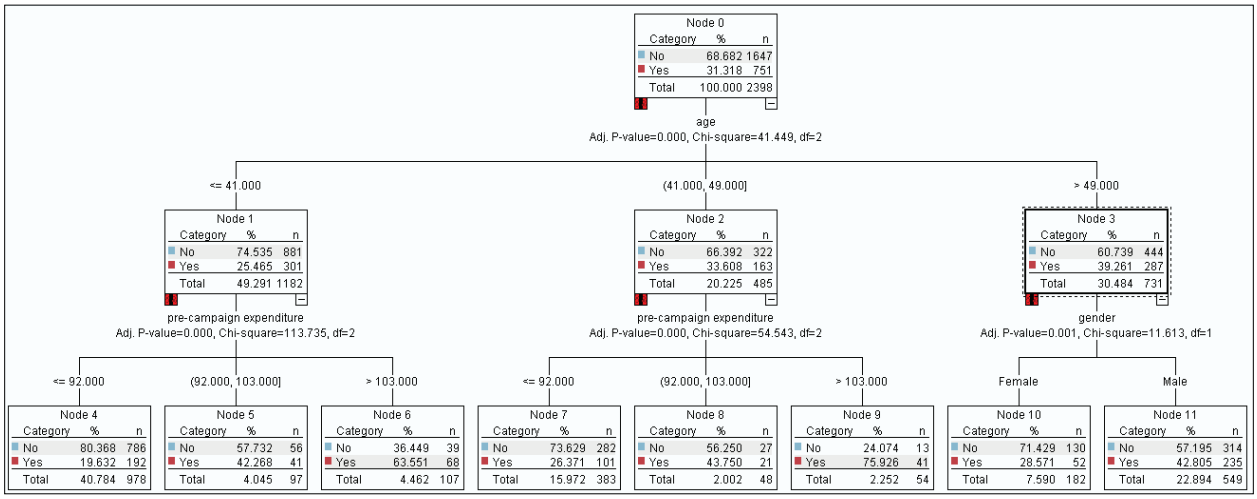
   The result appears as follows:



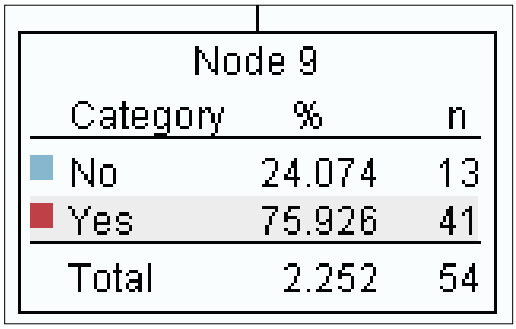
Task 7. Make further splits.

1. One at a time, select the nodes that represent age <=41, age between 42 and 49, and age > 49, and repeat the **Tree\Grow Tree with Custom Split**. Select the predictors and define categories as suggested (for the first two age categories: pre-campaign expenditure less than or equal to 92, pre-campaign expenditure greater than 92 and less than or equal to 103, pre-campaign expenditure greater than 103; for the third age category, split on gender).

   The result is a tree with a structure that appears as follows:



The group with age between 41 and 49 and pre-campaign expenditure greater than 103 represented by node #9 shows the highest response percentage:



Task 8. Examine the gain.

1. Click the **Gains** tab and from the **Target category** list, click **Yes**.
2. Click the **Cumulative** button, click the **Quantiles** button, and then select **Decile**.

   A section of the results appear as follows:

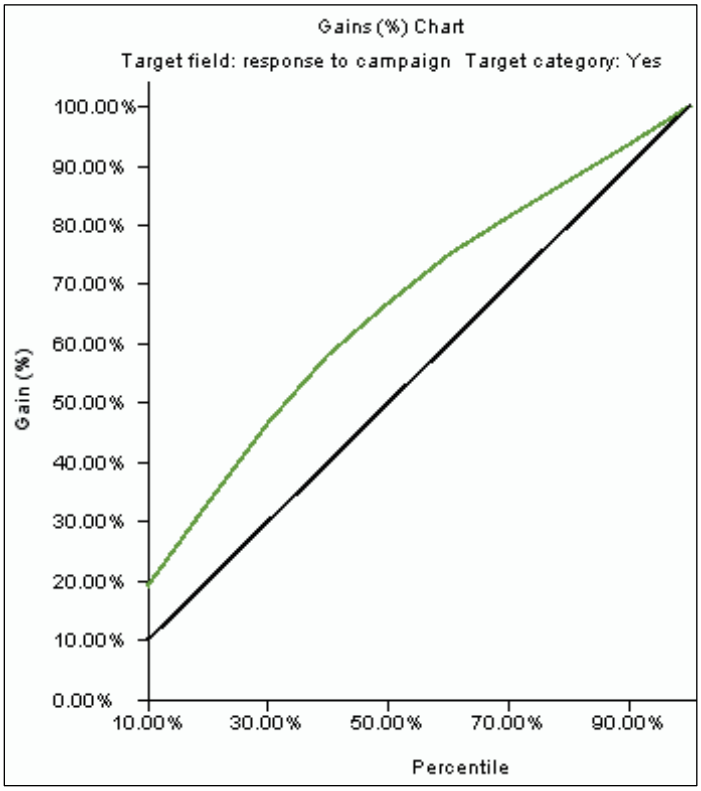
Tree Growing Set

Nodes	Percentile	Percentile: n	Gain: n	Gain (%)	Response (%)	Index (%)
9,6,8,11	10.00	240.00	143.00	19.08	59.70	190.61
11	20.00	480.00	246.00	32.76	51.25	163.65
11	30.00	719.00	348.00	46.38	48.44	154.68
11,5,10	40.00	959.00	436.00	58.02	45.43	145.07

The gain in the top 20% is 32.76%. Thus, the model captures 32.76% of the responders in 20% of the data.

3. Click the **Chart** button.

   A section of the results appear as follows:



   The gain is not very different from that of a random model.

Task 9. Examine the risk.

1. Click the **Risks** tab.

   A section of the results appear as follows:

Tree Growing Set

Misclassification Matrix

Risk Estimate		Predicted		
0.289	Actual	No	Yes	Total
	No	1595	52	1647
Standard Error	Yes	642	109	751
0.009	Total	2237	161	2398

   The probability to misclassify a record is 0.289. Given the fact that this is a custom model, this is not a bad result. However, the gain of the model is rather disappointing. Thus, in the next task a tree is grown by using the CHAID algorithm.

Task 10. Grow the tree from the root node.

1. Click the **Viewer** tab.
2. Right-click **node 0**, click **Remove Branch**, right-click **node 0** again and then click **Grow Tree**.
3. Click the **Gains** tab, and then select the **Yes** category of the target.
4. Click the **Cumulatives** button, click the **Quantiles** button, and ensure that **Decile** is selected.

   A section of the results appear as follows:

Tree Growing Set

Nodes	Percentile	Percentile: n	Gain: n	Gain (%)	Response	Index (%)
25,31,24	10.00	240.00	188.00	25.01	78.27	249.92
24,30,29,21	20.00	480.00	324.00	43.09	67.42	215.29
21,27	30.00	719.00	426.00	56.68	59.20	189.04
27,19,26	40.00	959.00	513.00	68.37	53.54	170.97
26,18	50.00	1199.00	568.00	75.59	47.35	151.18

   The best 20% captures 43.09% of the responders.

Task 11. Generate a model nugget.

1. Click the **Viewer** tab, and then select **Generate\Generate Model**. For **Create node on**, select the **Canvas** option, and then click **OK**.
2. Close the **Interactive Tree Builder** window.

Task 12. Score data with the model nugget.

1. Use a **Var. File** node to import data from **prospects for campaign.txt** (located in **C:\Train\0A0U5**; use default settings for the import).
2. Add the **model nugget** (generated in the previous task) downstream from the **Var. File** node.
3. Edit the **model nugget**, click the **Settings** tab, and then enable the **Calculate raw propensity scores** option.

4. Click the **Preview** button, and then scroll to the right.

   A section of the results appear as follows:

Table | Annotations

	$R-response to campaign	$RC-response to campaign	$RRP-response to campaign
1	No	0.625	0.375
2	No	0.789	0.211
3	No	0.748	0.252
4	Yes	0.831	0.831
5	No	0.883	0.117

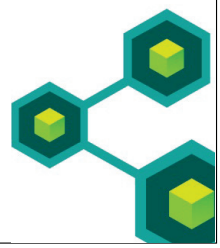
   The first prospect has a probability of 0.375 to respond. A good candidate for the campaign is record #4, with a probability of 0.831 to respond.

5. Close the **Preview** output window, and the close the **model nugget**.
6. Close MODELER without saving anything.

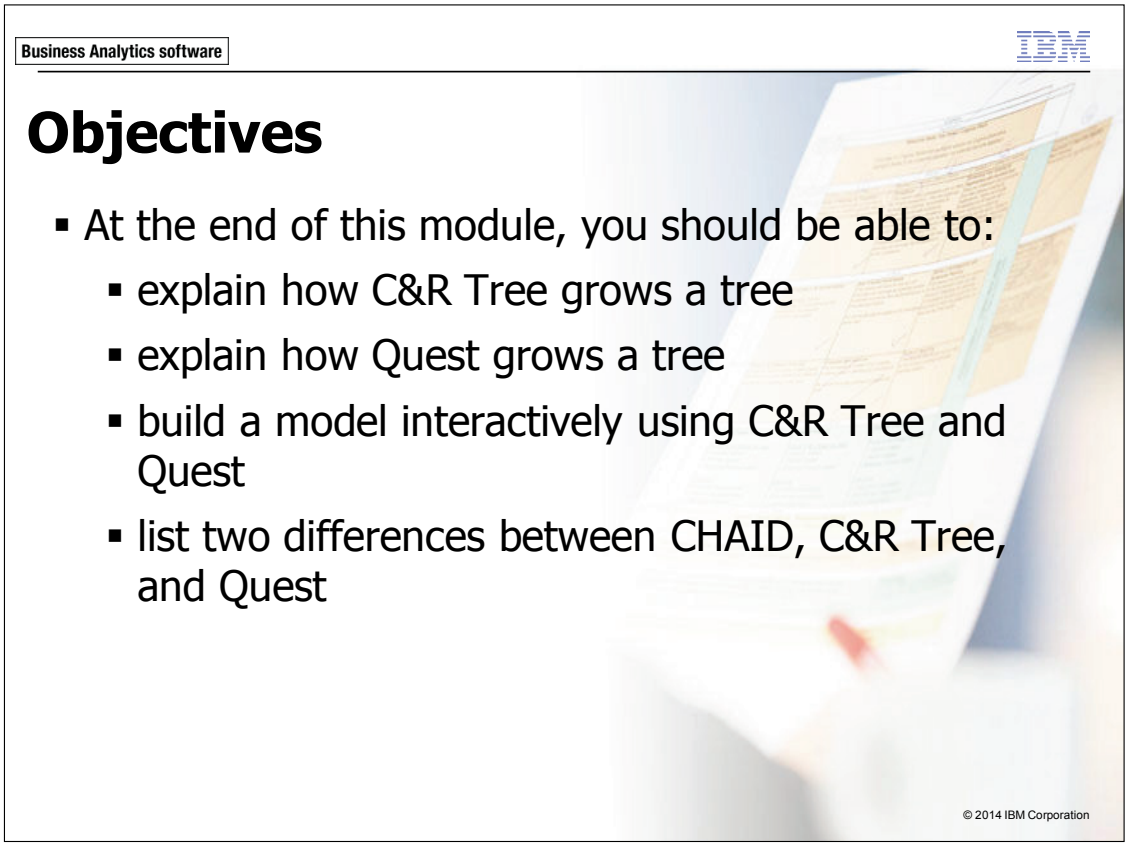
Note: The stream **workshop_building_your_tree_interactively_with_chaid_completed.str**, located in the **02-Building_Your_Tree_Interactively_with_CHAID\Solutions** sub folder, provides a solution to the workshop tasks.

Building Your Tree Interactively with C&R Tree and Quest

IBM SPSS Modeler (v16)



Business Analytics software



Before reviewing this module you should be familiar with:

- working with MODELER (streams, nodes, palettes)
- importing data (Var. File node)
- defining measurement levels, roles, blanks, and instantiating data (Type node)
- examining the data (Table node, Data Audit node)
- using CHAID in interactive mode
- assessing the quality of your model, using accuracy, risk estimate, gain and response measures
- using the model nugget to score data

Business Analytics software

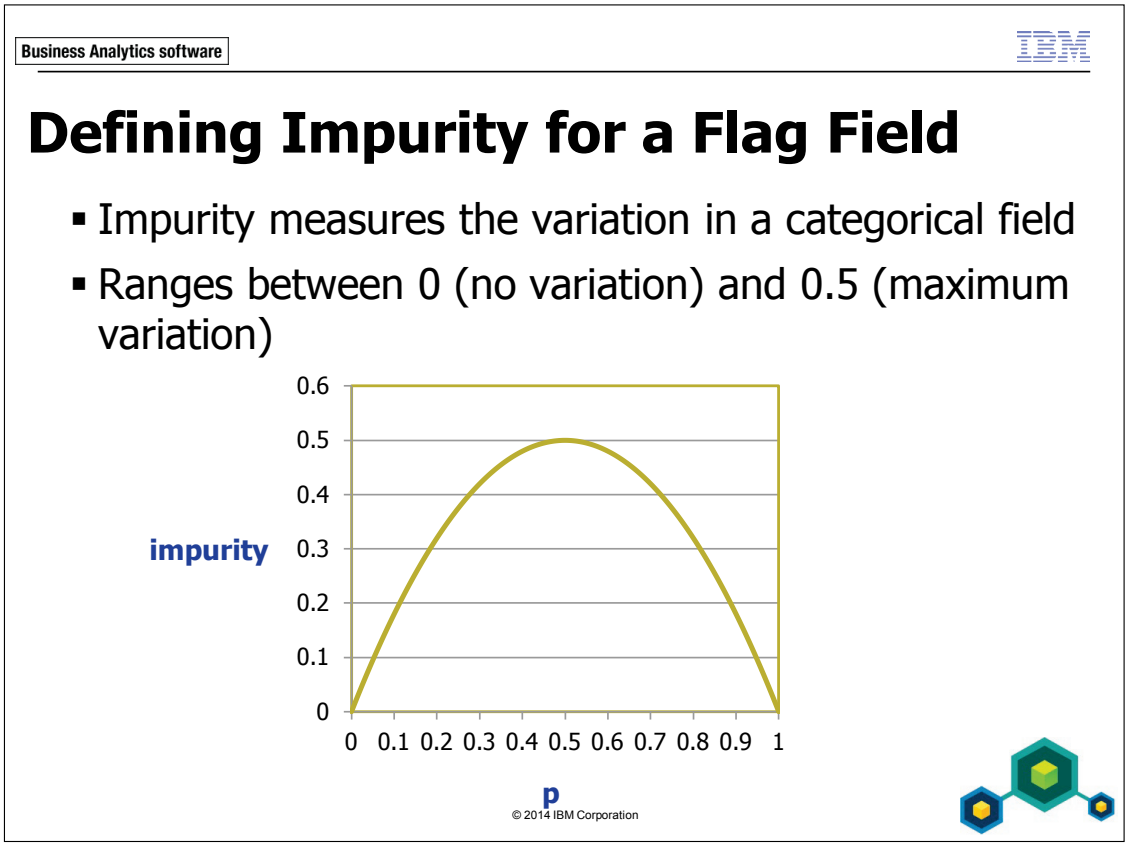
IBM

Growing the Tree with C&R Tree

- Same procedure as CHAID to grow the tree interactively:
  - select predictor
  - merge or split categories
- Uses the impurity statistic to grow the tree.

Using the Chi-square test is only one way to assess a relationship between two categorical fields. Alternatively, a criterion based on the so-called impurity can be used. This is the approach C&R Tree (Classification and Regression Tree) takes for growing trees.

The procedure to grow a tree interactively is the same, whether you use CHAID or C&R Tree. You can grow the tree of your own preference by selecting predictors and merging or splitting categories. The only difference is that the process of growing the tree is based on the impurity statistic rather than the Chi-square test. Thus, a tree grown with C&R Tree may be different from a tree grown with CHAID.



Key to C&R Tree is the concept of impurity. Impurity captures the degree to which records are spread out in the categories. For a flag field, impurity is defined as:

$$\text{impurity flag field} = 1 - p^2 - (1-p)^2$$

where p is the proportion of records in the true category of the flag field.

The figure on this slide depicts impurity as a function of the proportion true. The impurity is 0 when the proportion true is 0 or 1, and it reaches its maximum when the proportion true is 0.5 (in which case there are as many records in the false category as in the true category).

All in all, the more concentrated the records are in a single category, the lower the impurity, and the more balanced the categories are, the higher the impurity. Thus, impurity captures the dispersion in a categorical field.

Note: The impurity statistic presented here appears as Gini's impurity measure in the C&R Tree dialog box.

Business Analytics software

Impurity for a Flag Target Illustrated

sample data

ID	GENDER	AGE CATEGORY	HAS CHILDREN	PAID BACK LOAN
1	male	junior	yes	no
2	male	senior	no	yes
3	female	senior	yes	yes
...	...	...	...	...

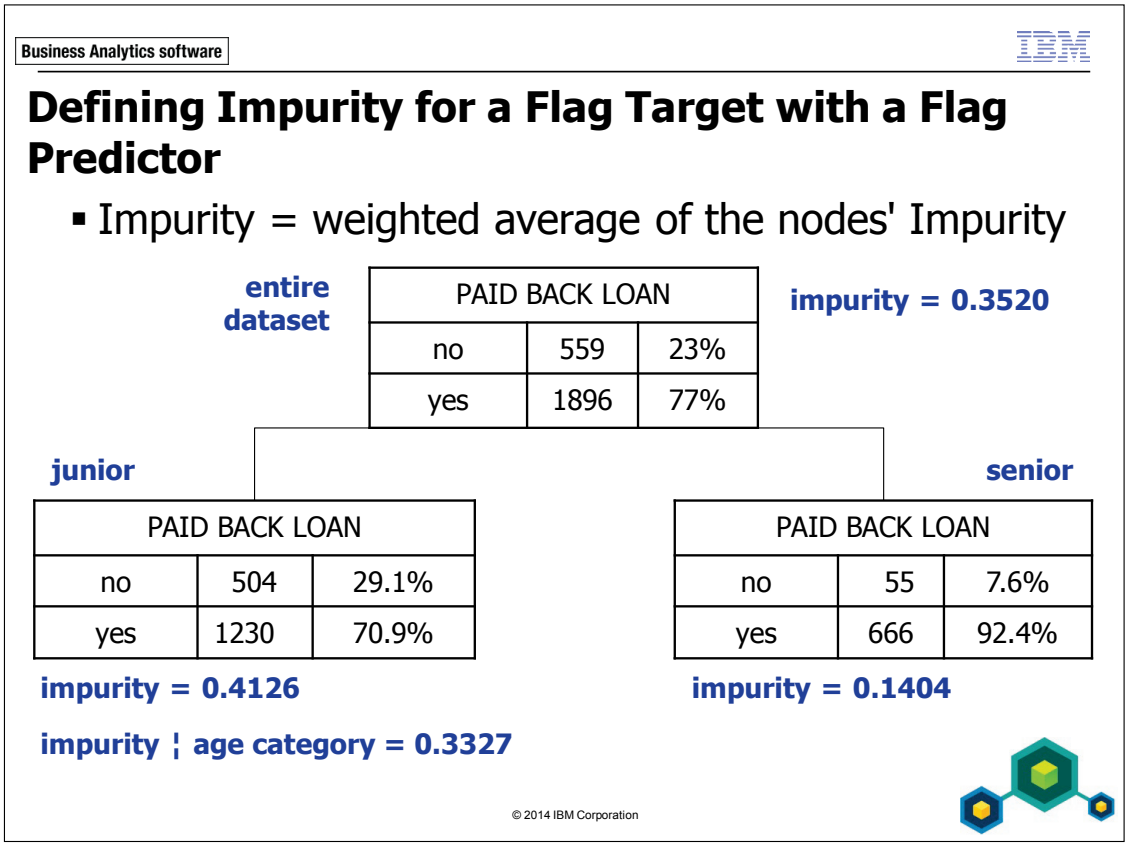
results

PAID BACK LOAN	N	%
no	559	22.8%
yes	1896	77.2%

impurity = 0.3520

This slide shows an example dataset, with a number of predictors and a target, PAID BACK LOAN. The impurity for the flag field PAID BACK LOAN equals $1 - 0.228^2 - 0.772^2 = 0.3520$.

To assess the importance of the predictors, the question is what the impurity in PAID BACK LOAN is when a predictor is taken into account.



When the node is split on a predictor, the impurity is computed as the weighted average of the impurity of the categories. In a formula, for a flag target Y and a flag predictor X:

$$\text{impurity for } Y \mid X = (n_1 * \text{impurity } Y \mid \text{cat } 1 + n_2 * \text{impurity } Y \mid \text{cat } 2) / (n_1 + n_2)$$

where:

impurity for $Y \mid X$	Impurity for Y, given X
impurity $Y \mid \text{cat}_i$	Impurity for Y, given category i of X
n_i	number of records in category i of X

In the example on this slide, the impurity for PAID BACK LOAN for the junior group (1734 records) is 0.4126, for the senior group (721 records) it is 0.1404. Thus, the impurity for PAID BACK LOAN given the age category is (1734 * 0.4126 + 721 * 0.1404) / 2455 = 0.3327.

Having defined the impurity for a flag target, given the information on the predictor, it is now possible to assess the importance of a predictor for the flag target.

Business Analytics software

IBM

Selecting a Predictor: Improvement

- Improvement = reduction in impurity
- Select the predictor that gives greatest improvement

PREDICTOR	IMPROVEMENT
AGE CATEGORY	0.019
HAS CHILDREN	0.004
GENDER	0.000

In the example dataset used here, the impurity for PAID BACK LOAN was 0.3520, and the impurity for PAID BACK LOAN when age category was taken into account was 0.3327. Thus, impurity is reduced by 0.3520 - 0.3327 = 0.019 by taken into account age category. The reduction in impurity when a predictor is taken into account is defined as improvement.

The improvement (reduction in impurity) statistic plays a key role in growing a tree. The best candidate to grow the tree, from C&R Tree's view, is the predictor that gives the greatest improvement. The table on this slide shows the improvement for three predictors. The best candidate to grow the tree with is AGE CATEGORY.

All in all, C&R Tree does not use statistical significance to grow the tree. A typical situation where results will differ from CHAID is when you have millions of records. Given the sensitivity of the Chi-square test to sample size, CHAID will show a high significance for a predictor, although the difference in percentages is very small. On the other hand, C&R Tree will show a very small improvement value for the predictor, so from this perspective the predictor is no candidate to grow the tree with.

Business Analytics software

IBM

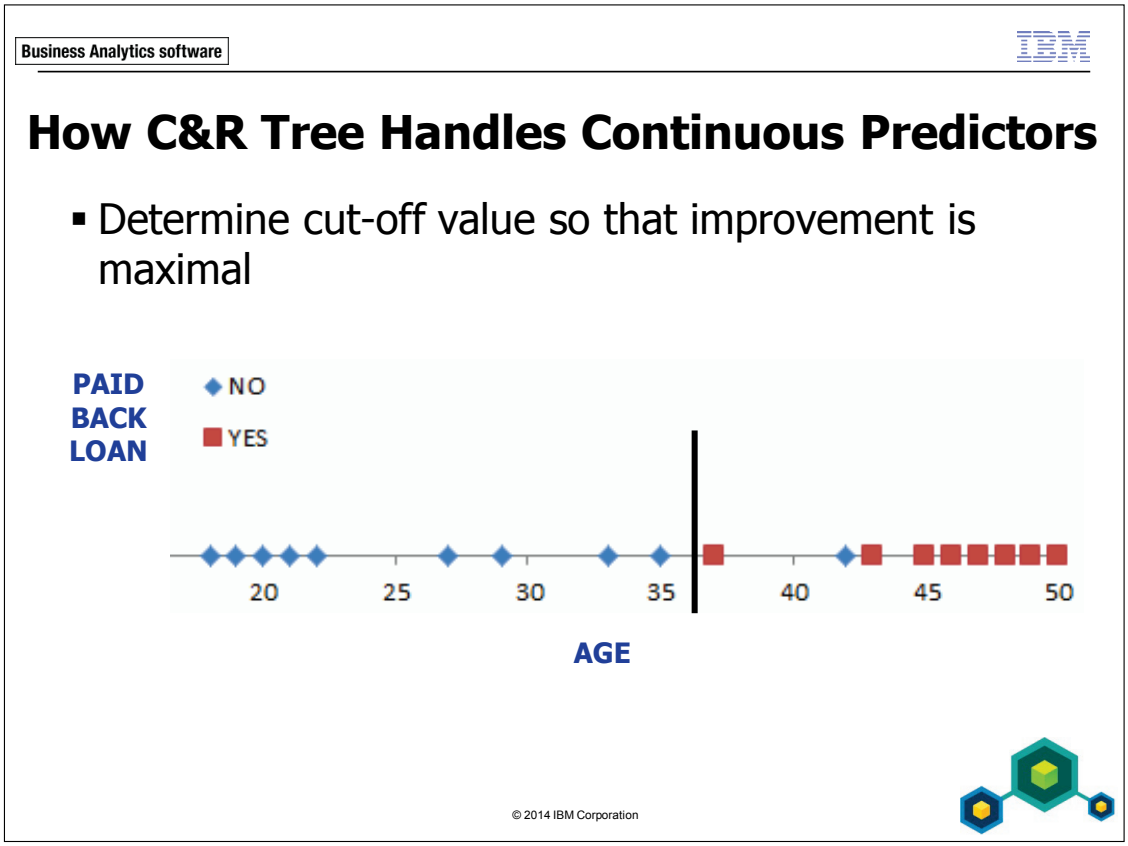
How C&R Tree Handles Categorical Predictors

- C&R Tree makes binary splits
- C&R Tree uses the improvement statistic to test categories for merge
- It depends on the measurement of the predictor which categories are tested for merge

C&R Tree will always perform a binary split: from a certain parent node there will be two child nodes. If the predictor is a flag field, there is no other option than a binary split, but what if the predictor has more than two categories? The approach C&R Tree takes depends, as with CHAID, on the measurement level of the predictor:

- Nominal predictors: All partitions of the categories into two groups are tried, and the split with the largest improvement is retained. For example, for a predictor with 3 categories 1, 2 and 3, the partitions that are evaluated are: {1, 2} versus {3}, {1, 3} versus {2}and {1}}.versus {2, 3}.
- Ordinal predictors: Partitions of the categories that retain the order of the categories are tried and the split that produces the largest improvement is recorded. Continuing the example, the partitions that are tested are {1, 2} versus {3}, and {1} versus {2, 3}.

Because C&R Tree makes binary splits, categories will be merged until two nodes remain.

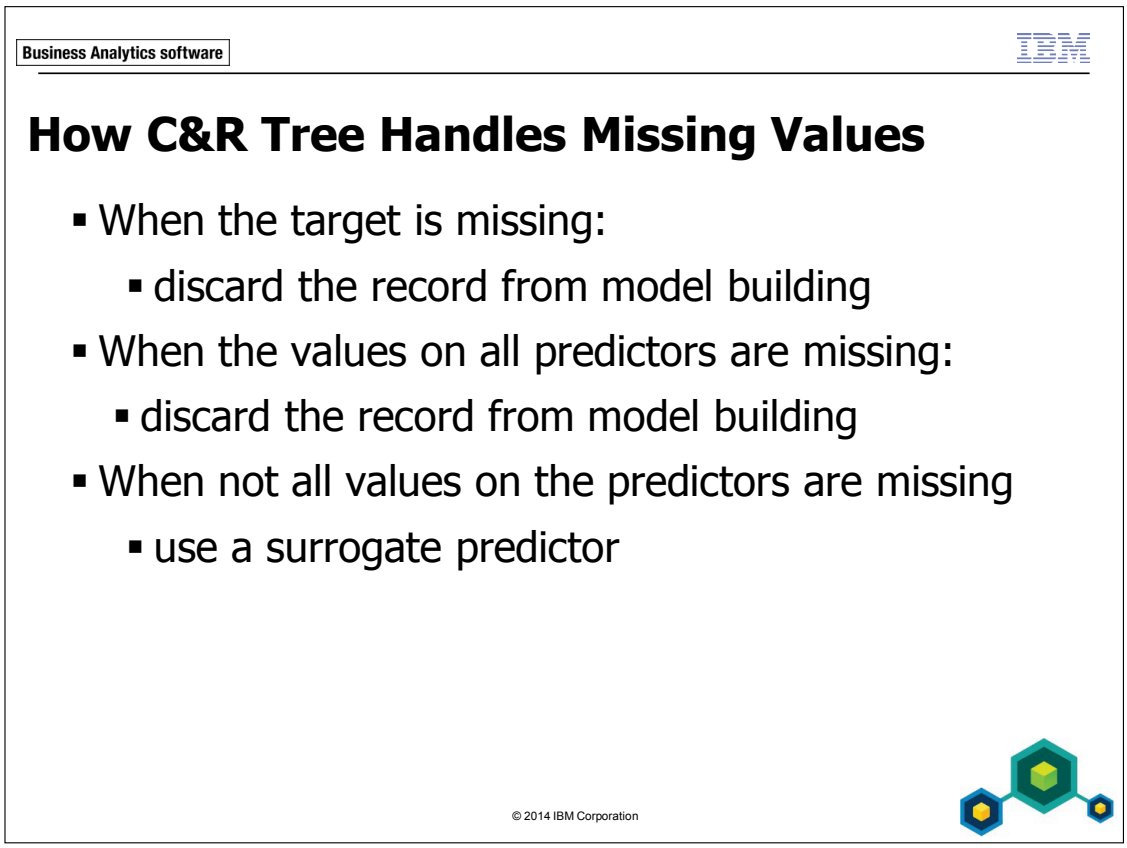


Impurity and improvement are only applicable when both the predictor and the target are categorical. A continuous predictor has to be reworked to a categorical one, so that impurity and improvement can be computed.

For a continuous field C&R Tree sorts the values from the smallest to the largest. It then computes the improvement for all possible cut points and retains the split that produced the maximum improvement.

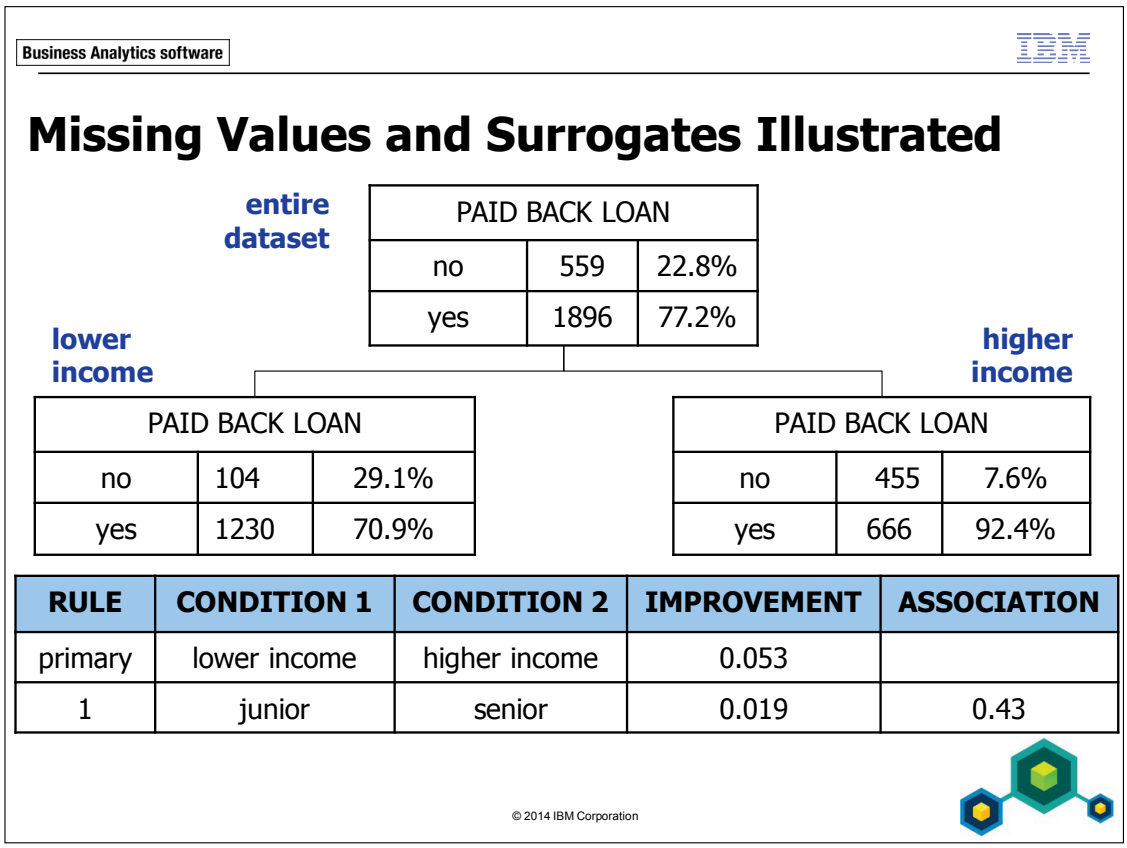
This slide shows an example. The straight vertical line at AGE 36 is the cut point yielding the highest improvement. The impurity for the group AGE< 36 will be 0 and the impurity for the AGE > 36 will slightly differ from 0 because of the one record that will be misclassified.

This approach is different from the approach that CHAID takes. Where CHAID creates a new ordinal field by binning the original continuous field, without any reference to the target field, C&R Tree finds the split in the predictor that gives a maximum improvement on the target. This is an advantage of C&R Tree compared to CHAID.



If the target value for a record is missing (user-defined blank or undefined ($null$), or if all the predictors are missing, the record is ignored in model building.

If the target is not missing and a predictor is missing but other predictors are not, another predictor that yields a split similar to the predictor that is missing is used as a substitute and its value is used to assign the record to one of the child nodes. Such a substitute predictor is called a surrogate.

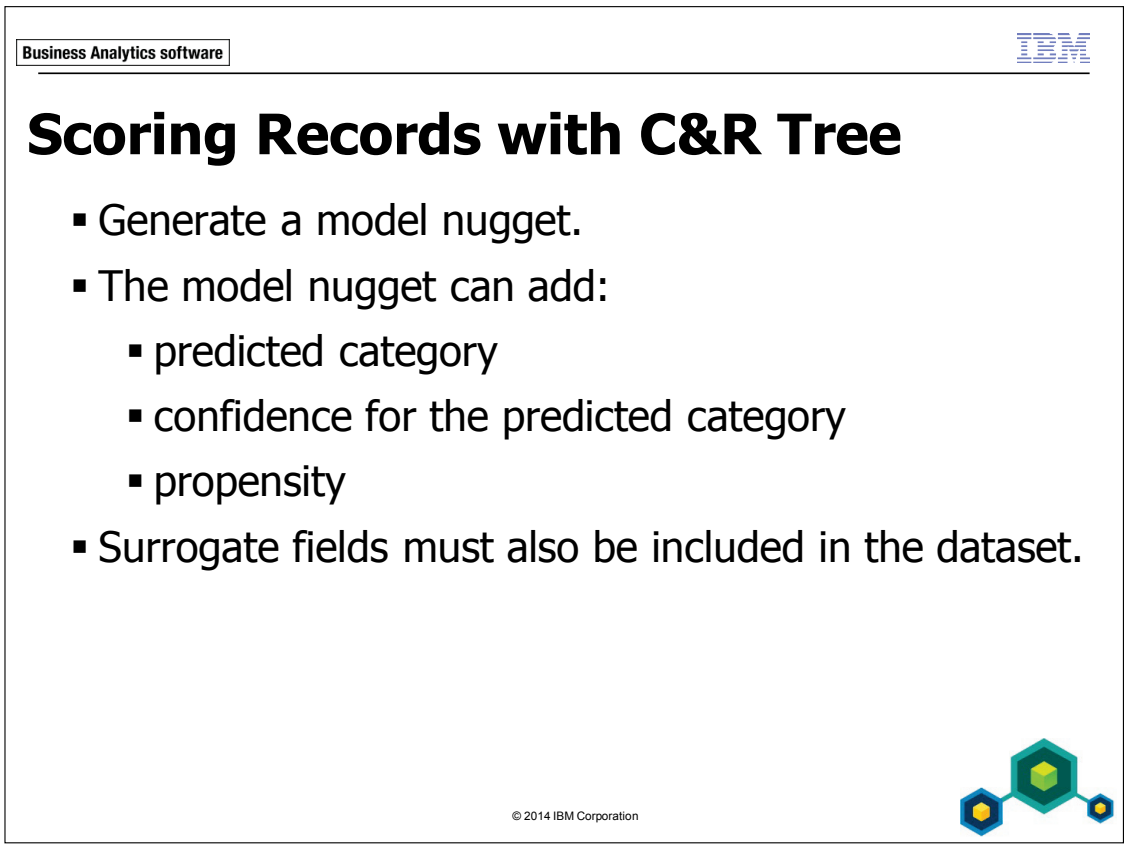


This slide presents an example of how C&R Tree uses a surrogate predictor when the predictor that shows the highest improvement has missing values.

The INCOME CATEGORY field yields the highest improvement and is used as primary split field. But for records where INCOME CATEGORY was missing a surrogate field was used: AGE CATEGORY. This field had the highest association with INCOME CATEGORY.

If a record misses INCOME CATEGORY, then C&R Tree will process AGE CATEGORY. If the record is junior, then C&R Tree will place it into the lower income category; a senior will be assigned to the higher income category (refer to the CONDITION columns). When AGE CATEGORY itself is missing the next best surrogate is used.

Again, this approach differs from CHAID, which treats missing values as just another category.



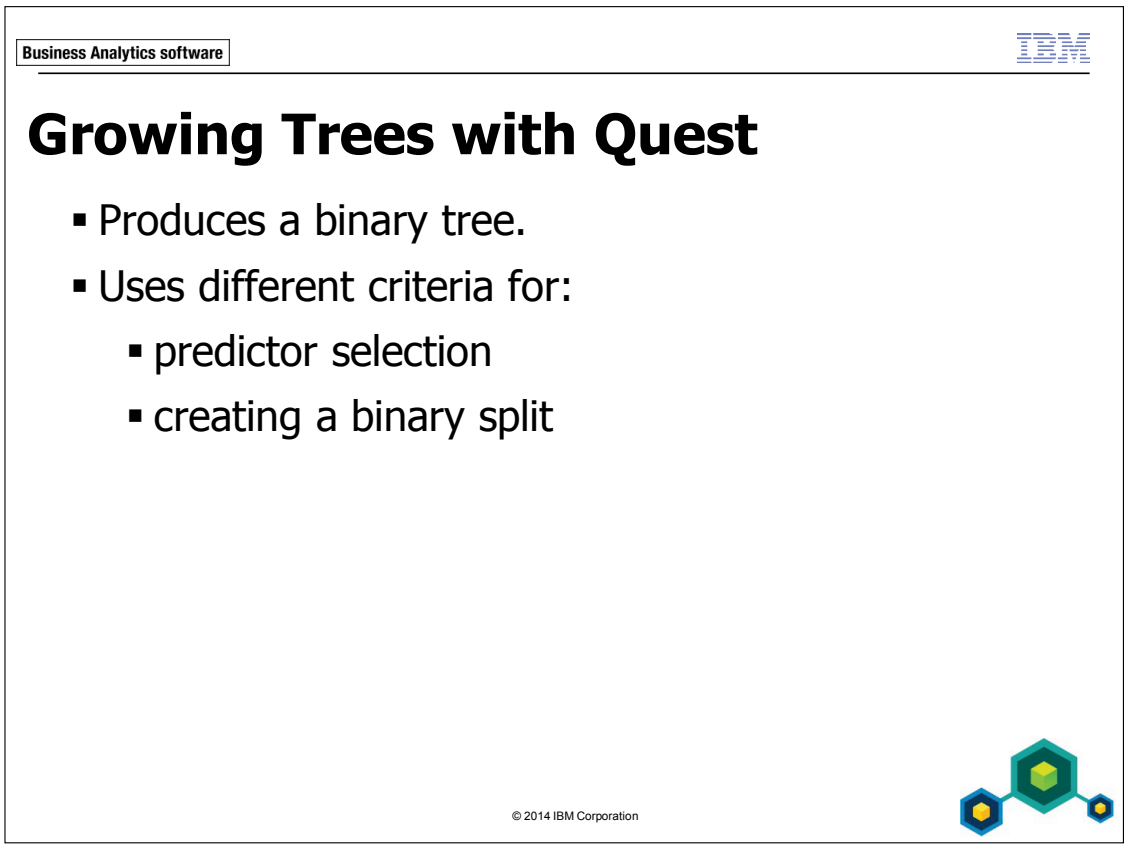
When a tree has been built, the tree can be used to score records by generating a model nugget for it and including the model nugget in the stream.

C&R Tree operates in the same way as CHAID does. It can add:

- The predicted category: The predicted category is the category for which the confidence is higher than 0.5.
- The confidence: The confidence for the predicted category.
- Propensity: The probability for the true value of the flag target.

Propensities will not be added by default, but you can enable the relevant option on the Settings tab in the model nugget.

When you use the model nugget to score records in datasets other than the one that you used for model building, ensure that not only the predictors that are included in the tree are in the dataset, but also the surrogates. If the surrogates are not included, it may not be possible to score your data because the surrogates are used behind the scenes when you have missing values.



Quest (Quick Unbiased Efficient Statistical Tree; Loh and Shih, 1997) is a tree method for nominal targets. It can include a predictor of any measurement level.

A major motivation in its development was to reduce the processing time required for large C&R Tree analyses. A second goal of Quest was to reduce the tendency found in some tree methods to favor predictors that allow more splits (continuous predictors or nominal predictors with many categories).

Similar to C&R Tree, Quest will perform binary splits. But unlike C&R Tree and like CHAID, it uses statistical tests to select predictors.

Quest also separates the issue of predictor selection and merging categories, applying different criteria to each. This contrasts with CHAID (the Chi-square test is used both for predictor selection and merging categories) and C&R Tree (improvement is used for both predictor selection and merging categories).

Quest is statistically more complex than CHAID and C&R Tree and in this course only basic concepts are presented. Refer to the *IBM SPSS Modeler Algorithms Guide* for more information.

Business Analytics software

IBM

How Quest Selects Predictors

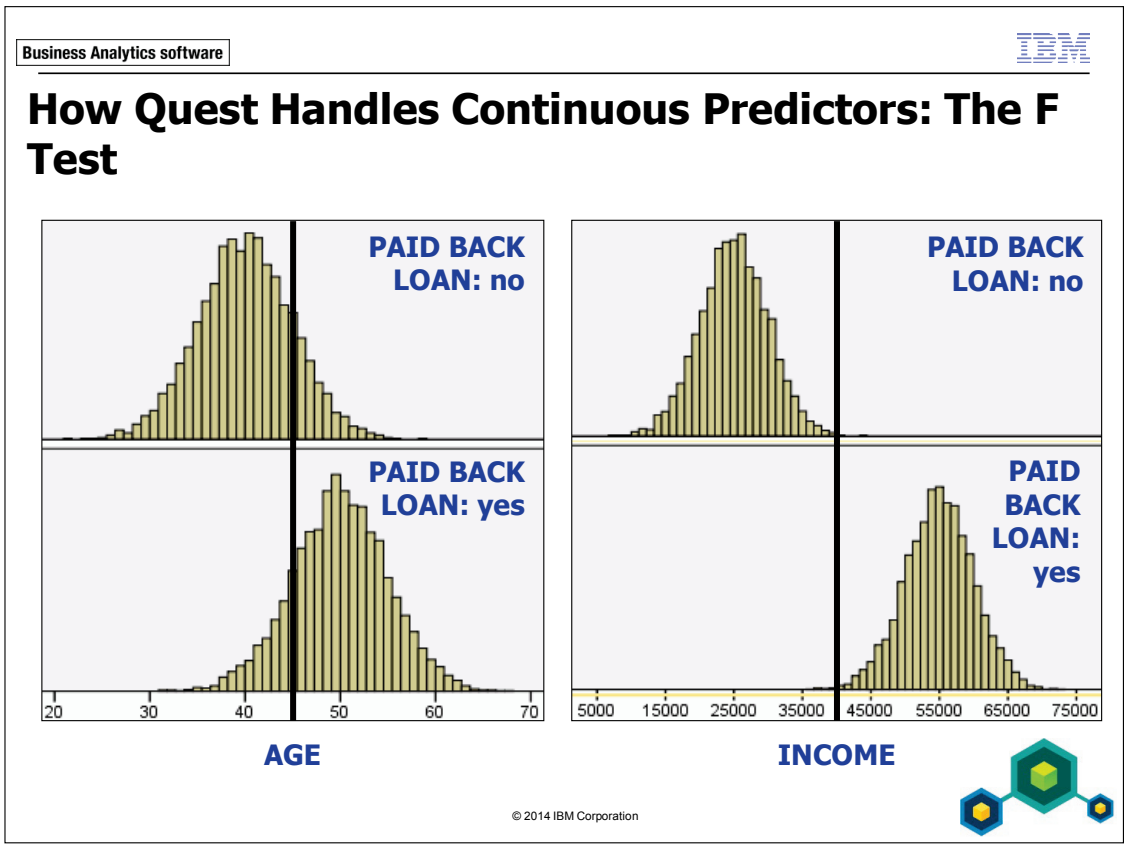
- Depends on the measurement level of the predictor:
  - Flag and nominal fields: Chi-square test
  - Ordinal and continuous fields: F test

PREDICTOR	STATISTIC	DF	PROBABILITY
INCOME	F=1224.59	1, 1742	0.000
REGION	Chi-square=24.62	3	0.000
GENDER	Chi-square= 2.81	1	0.094
EDUCATION	F=0.027	1, 1742	1.000

Quest uses significance tests to evaluate the predictors at a node. The measurement level of the predictors determines the test:

- Flag and nominal predictors: A Chi-square test is performed.
- Ordinal and continuous predictors: An F test is performed. This test considers the difference between the means for the two groups (as defined by the target).The table presented on this slide shows an example of the predictor selection dialog box, when you run Quest in interactive mode. The fields GENDER and REGION are categorical, so the results of the Chi-square test are displayed. The EDUCATION field is ordinal, INCOME is continuous. Thus, the F test was performed for these fields.

The Chi-square value, F value, and their associated degrees of freedom are merely technical details. The column of interest is entitled PROBABILITY. Here, INCOME and REGION are significant, while GENDER and EDUCATION are not (using a threshold of 0.05 to declare a result significant or not).

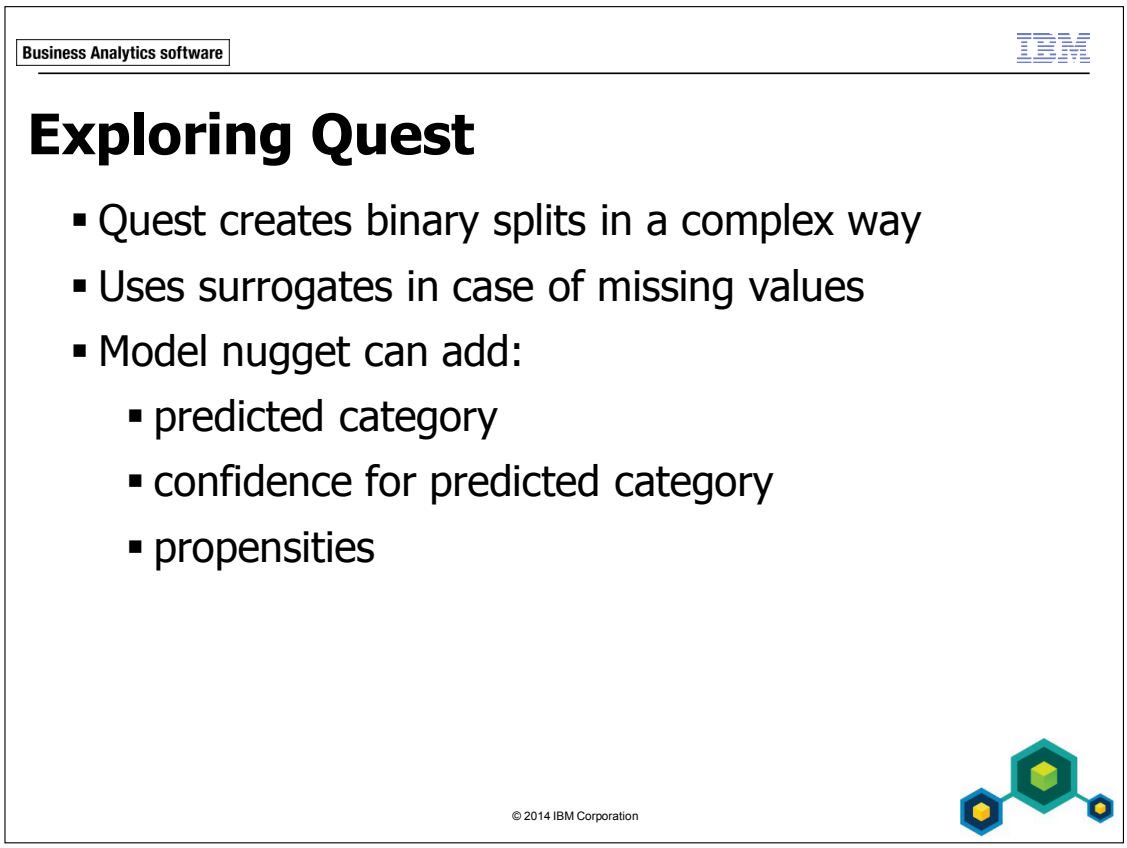


Where Quest uses a Chi-square test to assess the relationship between two categorical fields, an F test is used to examine the relationship between a categorical field and an ordinal or continuous field. The F test considers the difference between the means for the two groups (as defined by the target field), taking into account the distributions.

This slide presents an example for a target field PAID BACK LOAN and two predictors, AGE and INCOME. When you examine AGE, the two groups have different means, but the distributions overlap.

For INCOME, the two groups have different means, but the distributions do not overlap. The probability value for the F test for INCOME will be smaller than the probability for AGE, so Quest will prefer INCOME over AGE as a split field.

The super-imposed line can be used to classify records and comes close to what Quest does to determine the cut-off value for the binary split.



Quest uses significance tests to select the predictor. The specific test that is used depends on the measurement level of the predictor, as was explained on the previous two slides.

How Quest arrives at a binary split when the predictor is nominal, ordinal, or continuous is rather complex and beyond the scope of this course. The previous slide sketches the idea for a continuous predictor, but even in this case a description of the algorithm would need more technical details.

Quest handles missing values in the same way as C&R Tree. Records with missing values on predictors are included in model building and surrogate fields are used to stand in for a predictor with missing values. Refer to the presentation of surrogates in the C&R Tree section.

Quest scores records in the same way as C&R Tree and CHAID does. Propensities are not added by default, but can be requested on the Settings tab in the model nugget.

Business Analytics software

IBM

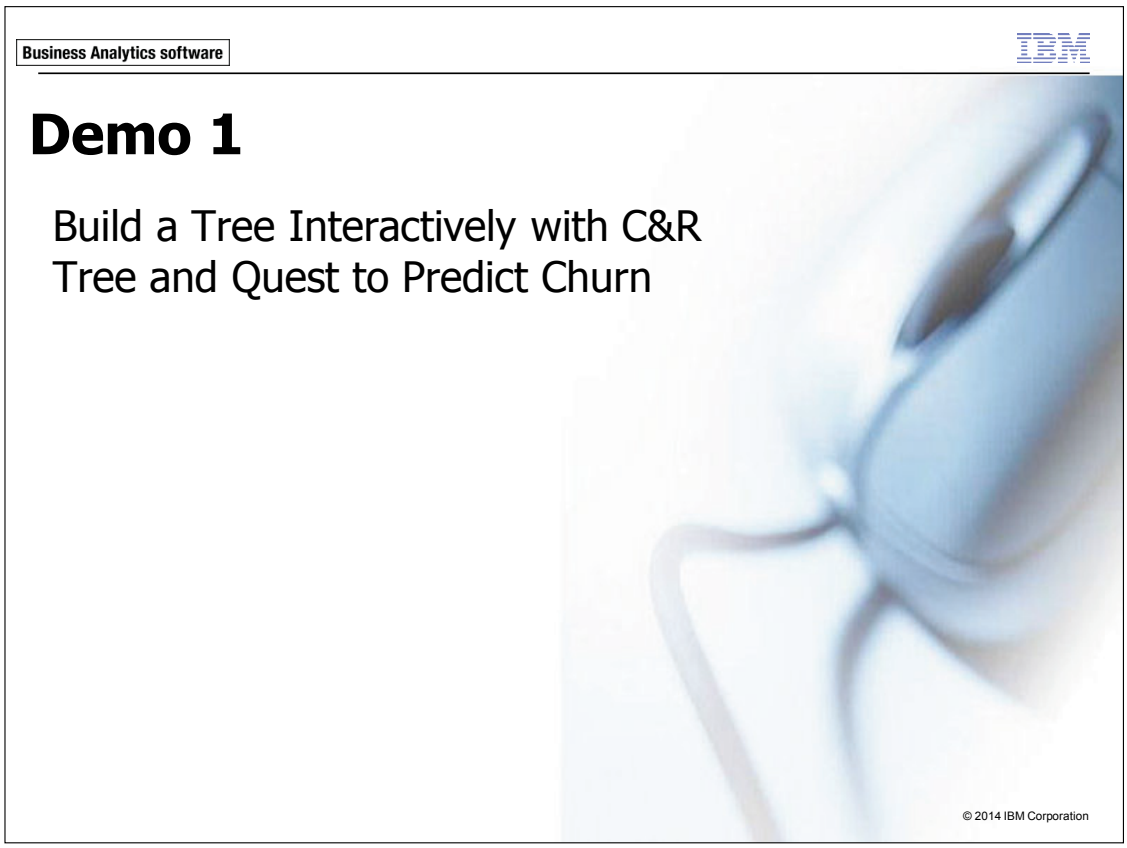
Determining the Tree Method to Use

CRITERION	CHAID	C&R Tree	Quest
type of tree	non-binary	binary	binary
statistic for predictor selection	significance (Chi-square test)	improvement	significance (Chi-square or F)
merging categories	based on Chi-square test	based on Improvement	rather complex
continuous predictor	bin into deciles, irrespective of target	optimal split wrt target	rather complex
missing values	treated as any other category	uses surrogates	uses surrogates

This slide lists the differences between the three tree methods. Given these differences you should not expect the techniques to produce identical results for the same data. What you can expect is that important predictors will be included in the resulting model built by any algorithm.

It is a matter of taste which model you prefer. Maybe one model has a somewhat higher accuracy and shows a better gain chart than another, at the expense of a more complex tree. Or maybe the results in one tree can be explained better to outsider than the results in another tree.

Also, you can combine the models into a single, so-called ensemble model. For example, you can build three trees, using each of the techniques, and then take the value that is predicted most often as the predicted value. Note: Although C&R Tree and Quest produce binary trees, the result could be the same as a CHAID tree. For example, suppose the CHAID splits a node into {1}, {2} and {3}. C&R Tree may split into {1, 2}, {3}, and then split {1, 2} into {1} and {2}. The end result will be the same as that of CHAID.



The following (synthetic) file coming from a (fictitious) telecommunications firm is used to demonstrate how you build your tree interactively with C&R Tree and Quest:

- **telco x modeling data.txt**: Data on approximately 32,000 customers of the firm. The data includes demographics, calling minutes, and product features, as well as a churn status. The values for the churn field can be either Active for current customers, or Churned, for churned customers. The file is located in **C:\Train\0A0U5**.

Before you begin the demo, ensure that:

- You have started MODELER (click **Cancel** when the splash screen appears when you start MODELER).
- You have set MODELER's working folder. (In MODELER, select **File\Set Directory**. When you are working in an IBM training environment, browse to the folder **C:\Train\0A0U5** and then click **Set** to select it. When you are not working in an IBM training environment, ask your trainer for the folder name.)

Demo 1: Build a Tree Interactively with C&R Tree and Quest to Predict Churn

Purpose:
You want to build a model interactively using the C&R Tree and Quest methods to predict whether or not customers will churn.

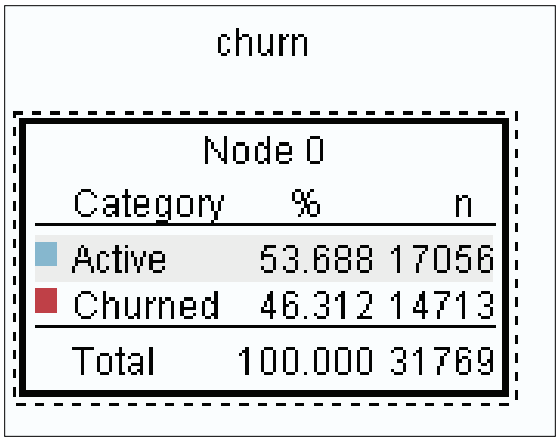
Task 1. Import and instantiate the data.

1. From the **Sources** palette, double-click the **Var. File** node to add it to the stream canvas.
2. Edit the **Var. file** node, and then:
   - in the **Var. File** dialog box, to the right of the **File** box, click **Browse**  (the Browse window should automatically open to the **C:\Train\0A0U5** folder)
   - select **telco x modeling data.txt** and then click **Open**
   - close the **Var. File** dialog box
3. From the **Field Ops** palette, add a **Type** node downstream from the **Var. File** node.
4. Edit the **Type** node, and then:
   - click **Read Values** to instantiate the data
   - ensure that the **Role** for **gender** to **bill_offpeak** is **Input**
   - select **churn** and set its **Role** to **Target**
   - set the role for the other fields to **None**
   - close the **Type** dialog box

Task 2. Add, configure and run the C&R Tree node.

1. From the **Modeling** palette (**Classification** item), add a **C&R Tree** node downstream from the **Type** node.
2. Edit the **C&R Tree** node, and then:
   - click the **Build Options** tab
   - click the **Objective** item, and in the **Build a single tree** pane, select the **Launch interactive session** option
   - click the **Advanced** item, and then set the **Overfit prevention set(%)** to **0** (you will use all records for model building)
   - click **Run**

   The Interactive Tree Builder window appears. A section of the results appear as follows:



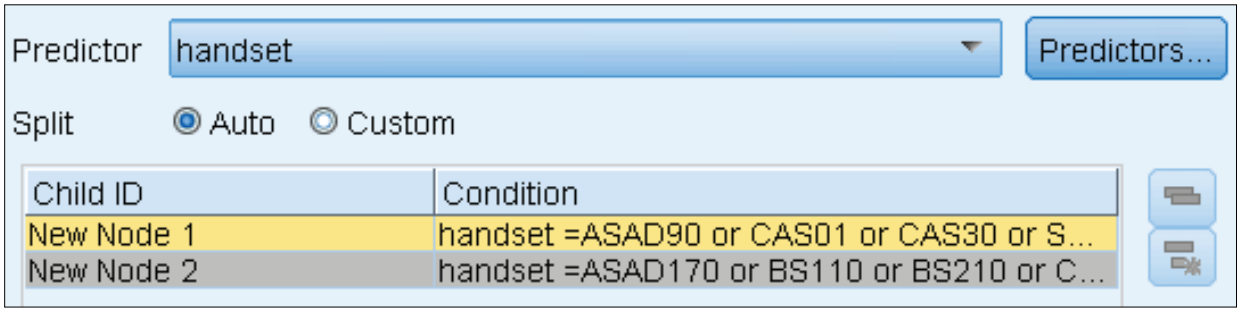
   The percentage of records that churned was 46.3% in this dataset.

   You will grow a tree interactively from the root node, selecting a predictor of your own preference.

Task 3. Examine the relevance of the predictors.

1. Select **Tree\Grow Branch with Custom Split** (alternatively, click the **Grow Branch with Custom Split**  button).

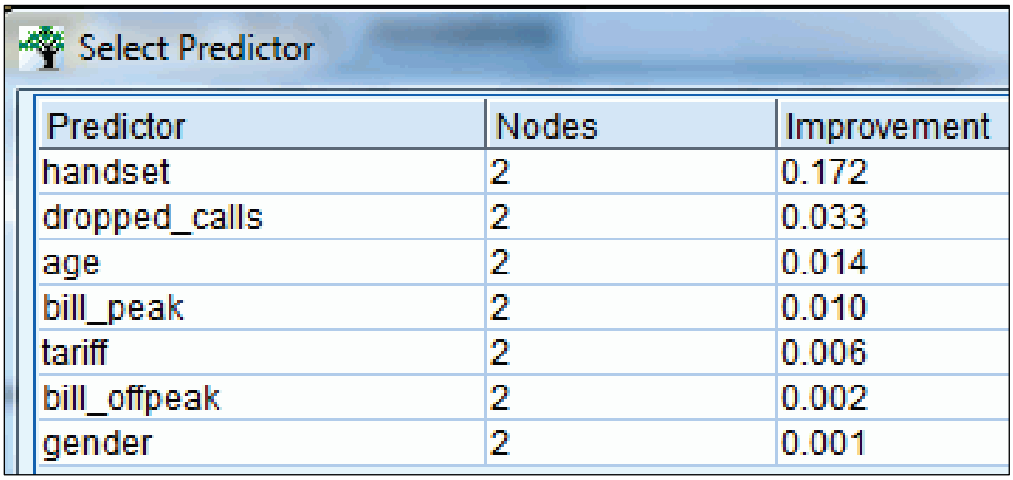
   The results appear as follows:



   The handset field is the first suggested predictor, which means that a split on handset yields the highest improvement. It uses a binary split of ASAD90 and so forth in one branch, and ASAD170 and so forth, in the other.

2. Click the **Predictors**  button.

   A section of the results appear as follows:

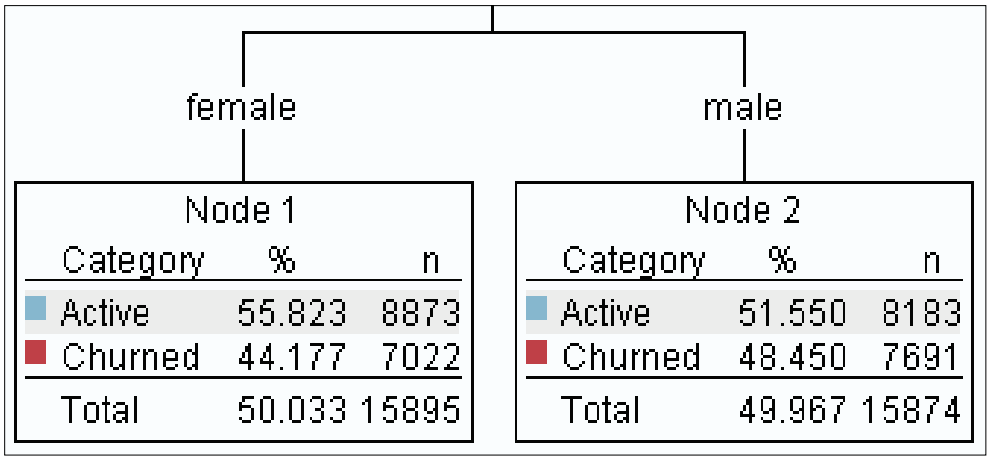


Predictor	Nodes	Improvement
handset	2	0.172
dropped_calls	2	0.033
age	2	0.014
bill_peak	2	0.010
tariff	2	0.006
bill_offpeak	2	0.002
gender	2	0.001

   For each predictor the number of nodes, or branches, in which the field would be split is listed. The number of nodes will always be two for C&R Tree.

   The handset field results in the largest improvement. Business needs may require you to grow the tree with a field that may actually result in the lowest improvement. That is the approach you will take here, and you will grow the tree from the root using gender as split field.

3. Click **Cancel** to close the **Select Predictor** window.
4. In the **Define Split** window, from the **Predictor** list, click **gender**, and then click **Grow**.
5. A section of the results appear as follows:



The tree reflects the requested split. Men show a higher churn percentage than women.

Suppose that you want to grow the tree for the male group using a custom split.

6. Click **Node 2** to select the **male** node, and then select **Tree\Grow Branch with Custom Split** (alternatively, click the **Grow Branch with Custom Split**  button).

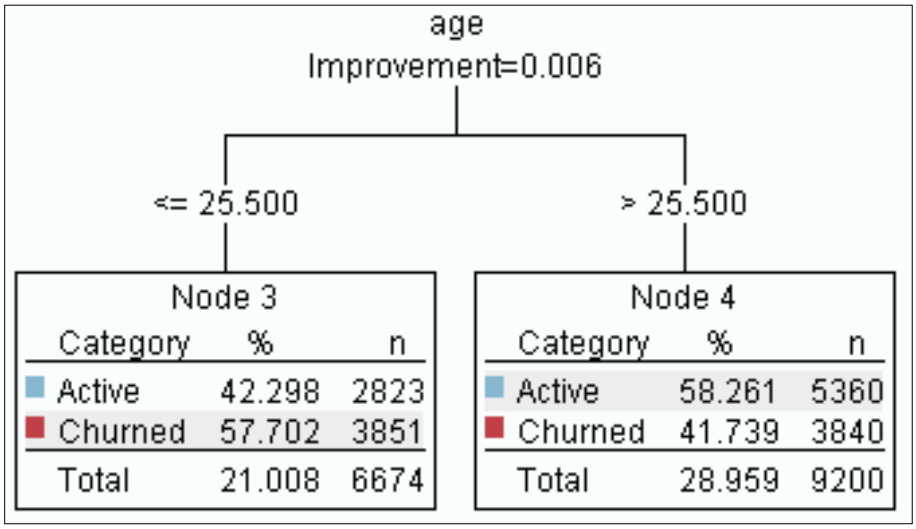
You will further grow the tree for males using age as the next split field.

7. From the **Predictor** list, select **age**.

The suggested split for age shows that the cut point that results in the largest improvement is at age 25.5, with 25.5 itself included in the first child node.

8. Click **Grow** to grow the tree with age, for males.

   A section of the results appear as follows:



   The category "age MISSING" that was present in the CHAID tree is not in the C&R Tree. That is because C&R Tree treats missing data differently from CHAID. CHAID treats the missing data just as another category, but C&R Tree uses surrogates.

   You will examine the surrogates that are used.

9. On the toolbar, click the **Show Split Information**  button.

   The bottom pane displays information about surrogates. The result appears as follows:

Rule	Condition 1	Condition 2	Improvement	Association
Primary	age <=25.500	age >25.500	0.006	.
1	handset =ASAD90	handset =ASAD170 or ...	0.033	0.049
2	dropped calls >14.500	dropped calls <=14.500	0.002	0.001
3	bill offpeak >45.128	bill offpeak <=45.128	0.000	0.001
4	bill peak >287.010	bill peak <=287.010	0.000	0.000

   If a record is missing age, the first candidate to replace age is handset, because of its association with age. If the record's handset equals ASAD90, the record is treated as if age was <= 25.500; as the Condition 1 column indicates. If the record's handset is other than ASAD90, the record will be treated as if age was greater than 25.5.

As with CHAID in interactive mode, you can grow the tree one level, grow a particular branch, remove branches, and so forth. You can also evaluate the model (the Gains tab), assess the misclassification rate (Risks tab), and generate a model nugget (on the Viewer tab, select Tree\Generate Model). These capabilities are not demonstrated for C&R Tree.

10. Close the **Interactive Tree Builder** window.

Task 4. Use Quest interactively.

1. From the **Modeling** palette (**Classification** item), add a **Quest** node downstream from the **Type** node.
2. Edit the **Quest** node, and then:
   - click the **Build Options** tab
   - select the **Objective** item, and in the **Build a single tree** pane, select the **Launch interactive session** option
   - select the **Advanced** item, and then set the **Overfit prevention set(%)** to **0** (all records will be used for model building)
   - click **Run**

   The Interactive Tree Builder window appears. You will grow a tree interactively from the root node.
3. Select **Tree\Grow Branch with Custom Split** (alternatively, click the **Grow Branch with Custom Split**  button).

4. Click the **Predictors** Predictors... button.

A section of the results appear as follows:

Select Predictor

Predictor	Nodes	Statistic	DF	Adj. Prob.
dropped_calls	2	F=818.259	1; 31767	0.000
age	2	F=498.351	1; 31752	0.000
handset	2	Chi-square=13941.022	11	0.000
bill_peak	2	F=109.080	1; 31767	0.000
bill_offpeak	2	F=95.804	1; 31767	0.000
tariff	2	Chi-square=502.385	4	0.000
gender	2	Chi-square=58.319	1	0.000

The test that is used, Chi-square or F, depends on the measurement level of the predictor.

The dropped_calls field has the lowest (adjusted) probability value, thus is the most significant predictor. Notice that dropped_calls was the second best field in the C&R Tree analysis. The dropped_calls field was also second best when CHAID was used. Thus, it matters whether:

- the field is binned into an ordinal field to which a Chi-square test is applied (as CHAID does)
- the field is treated as continuous for which the best cut-off value in terms of improvement is determined (as C&R Tree does), or
- whether an F test is applied (as Quest does)

All in all, CHAID, C&R Tree, and Quest have different angles from which they look at the data, so, in general, the results will be different.

As with CHAID and C&R Tree in interactive mode, you can grow the tree one level, grow a particular branch, remove branches, and so forth. You can also evaluate the model (the Gains tab), assess the misclassification rate (Risks tab), and generate a model nugget (on the Viewer tab, select Tree\Generate Model). These capabilities are not demonstrated for Quest.

5. Click **Cancel** to close the **Select Predictor** window, and then click **Cancel** to close the **Define Split** window.
6. Close the **Interactive Tree Builder** window.
7. Close the stream without saving anything.

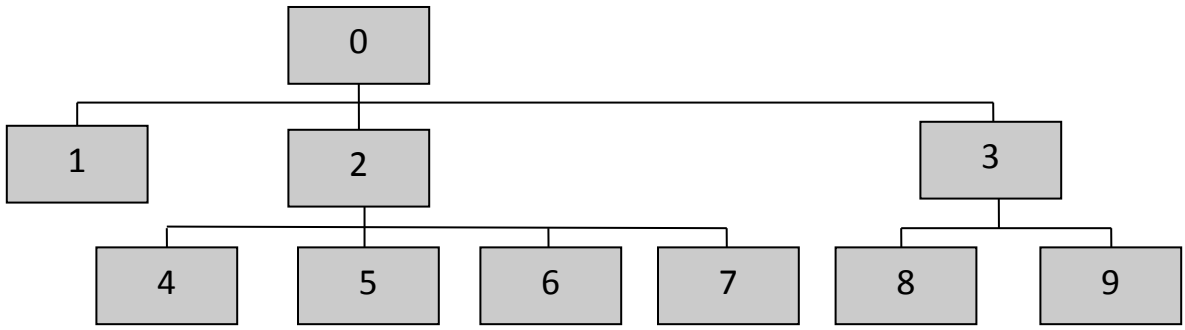
Results:
You built your tree model interactively, with C&R Tree and Quest to predict churn.

Note: You will find the solution results in **demo_building_your_tree_interactively_with_c&r_tree_and_quest_completed.str**, located in the **03-Building_Your_Tree_Interactively_with_C&R_Tree_and_Quest\Solutions** sub folder.

Apply Your Knowledge

Use the questions in this section to test your knowledge of the course material.

Question 1: Is the following statement true or false? This tree was built using C&R Tree.



A. True

B. False

Question 2: Is the following statement true or false? The Impurity statistic is a statistic that gives the probability that two categorical fields are related.

A. True

B. False

Question 3: Select all that apply.

A. Consider a flag target Y and a categorical predictor X with more than 2 categories. When splitting Y on X, the improvement treating X as a nominal field is always greater than or equal to the improvement when X is treated as an ordinal field.

B. For a flag field, the closer the probability for the true value is to 0.5, the smaller is the impurity.

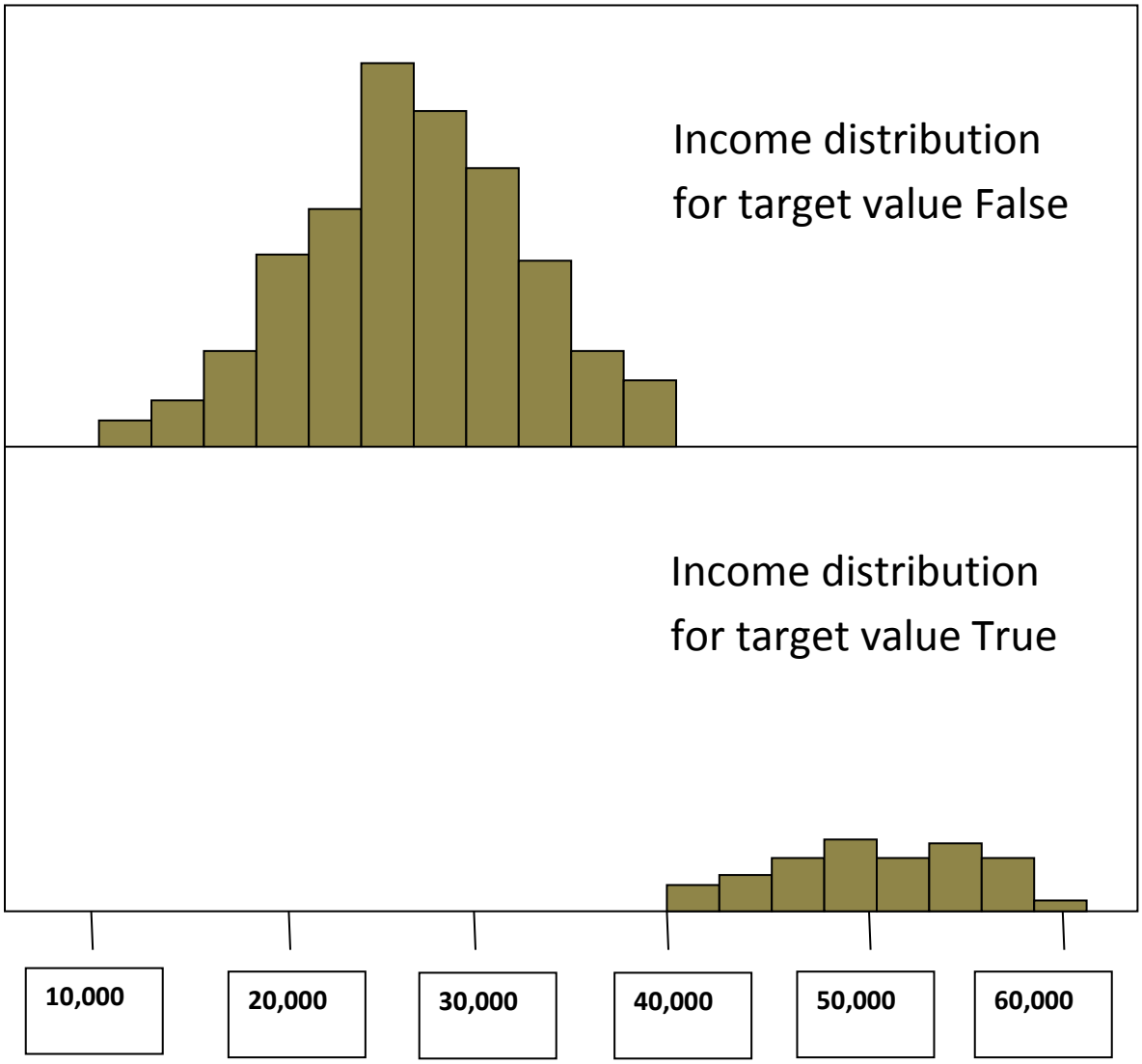
C. Consider two flag fields X and Y. For X, 80% has value T, 20% has value F. For Y, it is the other way around: 20% has value T, 80% has value F. The impurity for X and Y is the same.

Question 4: In the figure below, the income distribution is plotted for the true and false values for a flag target. When INCOME is used as a predictor, which of the following cut-off values for INCOME gives the largest improvement?

A. 20,000

B. 40,000

C. 60,000



Question 5: Select all that apply. A reason that CHAID and C&R Tree may produce different trees is:

A. CHAID uses significance, while C&R Tree uses improvement.

B. CHAID does not necessarily grow a binary tree, while C&R Tree does.

C. In CHAID, missing values are a separate category, while C&R Tree uses surrogates.

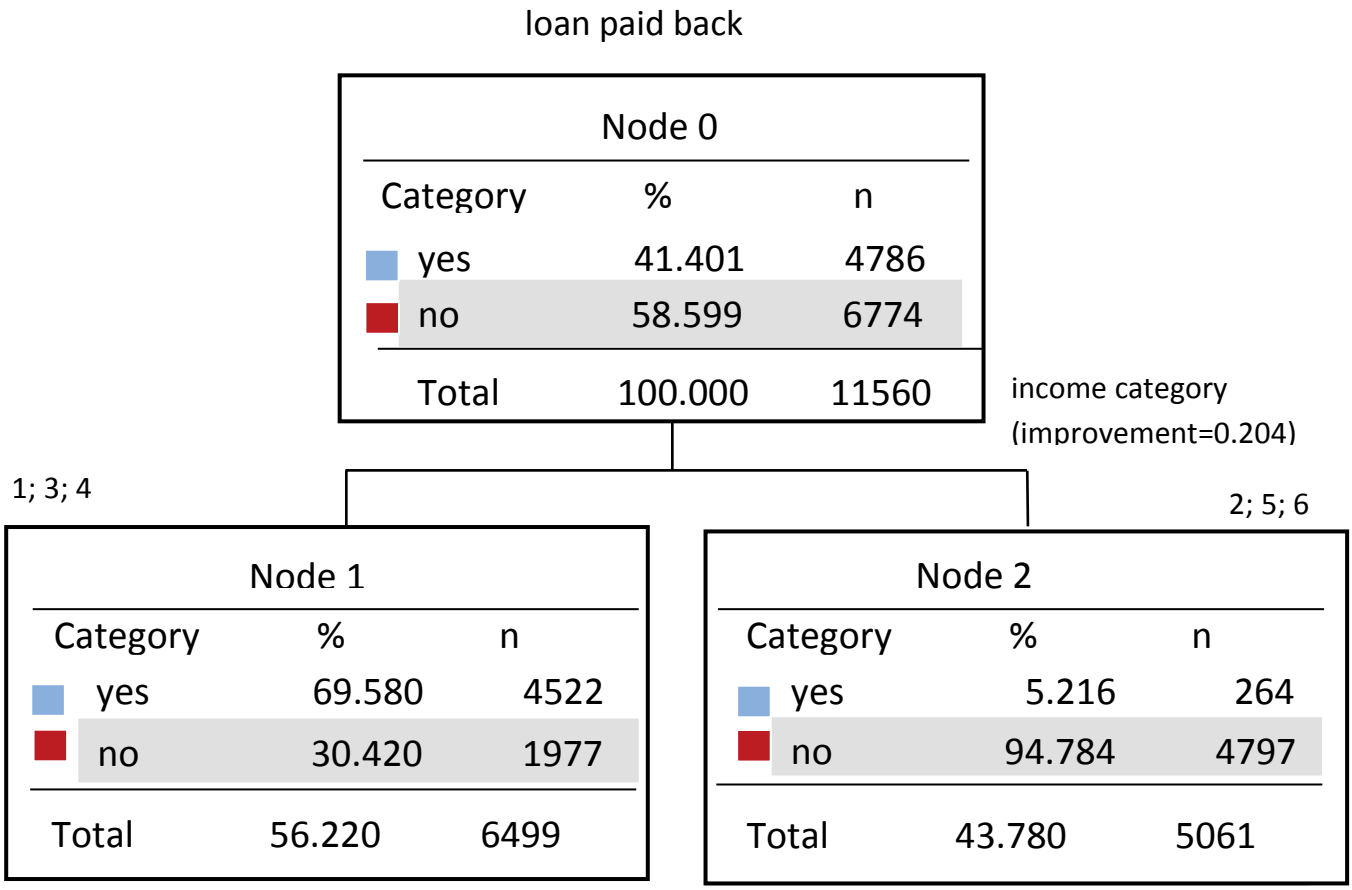
D. CHAID bins a continuous predictor in deciles, while C&R Tree finds the best cut-off value (the value that results in the highest improvement).

Question 6: Consider the below figure, a tree built with C&R Tree. Also included is information on surrogates. Which of the following statements are correct?

A. When the income category for a 27-year-old customer is missing, then this customer will be included in Node 1.

B. The measurement level of income category was nominal.

C. The impurity in the root node is 0.58599.



Rule	Condition 1	Condition 2
Primary	income category = 1 or 3 or 4	income category = 2 or 5 or 6
1	age > 24.500	age <= 24.500

Information about surrogates for split at income category

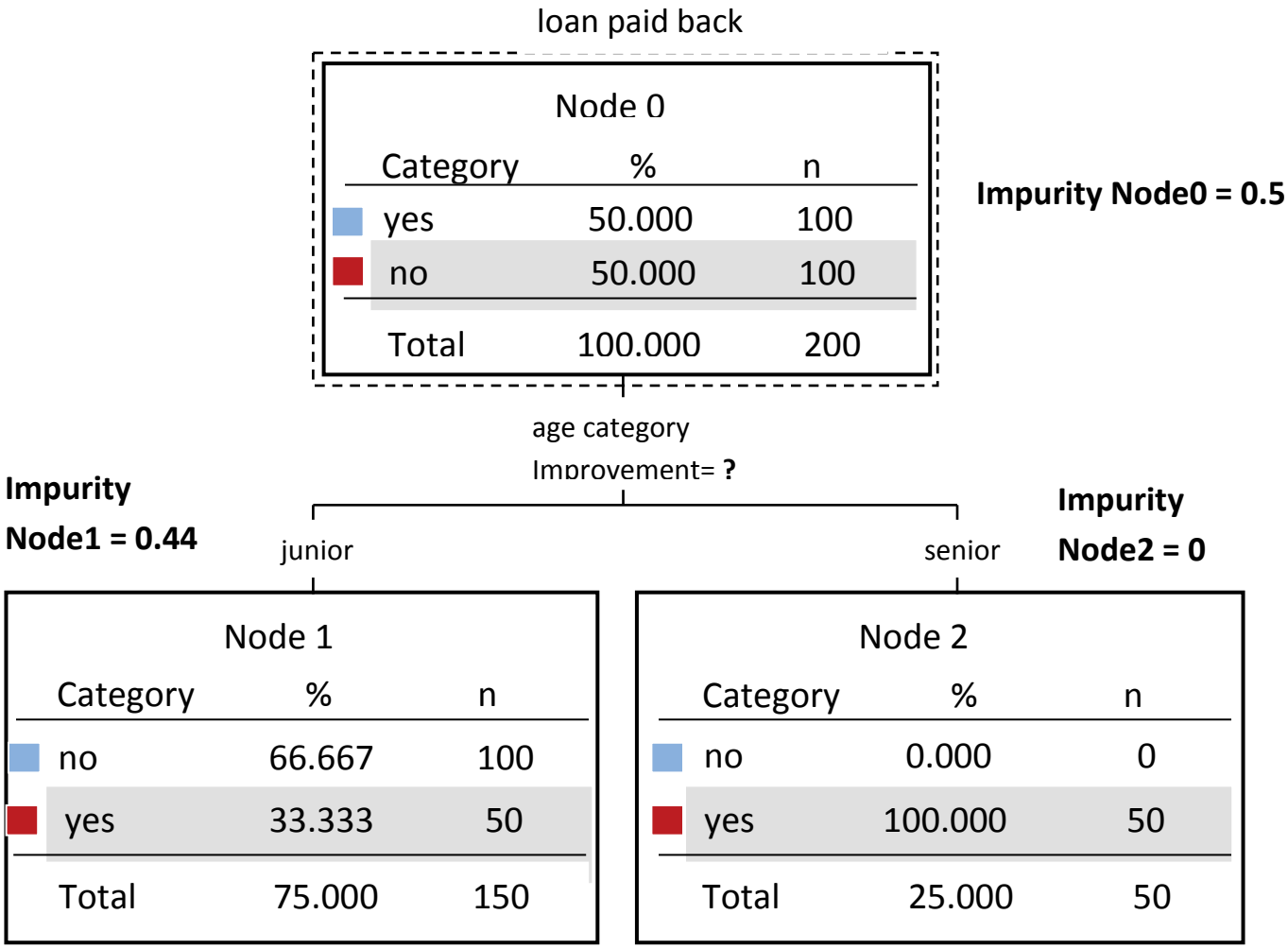
Question 7: Consider the below figure, a tree built with C&R Tree. The value for the improvement is:

A. 0

B. 0.11

C. 0.17

D. 0.33



Question 8: Select all that apply.

A. Quest always grows a binary tree.

B. Quest uses surrogate fields to classify a record with a missing value on a split field.

C. Quest uses the same test for an ordinal and continuous predictor to assess their relevance.

Question 9: When running interactive Quest, the following results appeared in the Select Predictor dialog. Select all the correct statements.

Predictor	Nodes	Statistic	DF	Adj. Prob.
AGE	2	F=224.636	1;1742	0.000
NUMBER OF CHILDREN	2	F=113.440	1;1742	0.000
MARITAL STATUS	2	Chi-square=70.542	2	0.000
INCOME1K	2	F=2.673	1;1742	0.511
GENDER	2	Chi-square=0.043	1	1.000

A. AGE is an ordinal or continuous field.

B. The number in the Nodes column is always 2.

C. Men and women do not differ significantly with respect to the target.

D. The column labeled DF is only a technical detail needed to compute probability values.

Answers to questions:

Answer 1: B. False. This is not a binary tree.

Answer 2: B. False. The impurity statistic captures the degree to which records are concentrated, without any reference to statistical significance.

Answer 3: A and C.

When splitting Y on a nominal field X, all combinations of the categories are tested for a merge, while for an ordinal field only combinations of adjacent categories are tested. Thus, the improvement when X is nominal will always be greater than or equal to the improvement then when you treat X as ordinal.

The impurity for two fields X and Y, X with 80% T and 20% F, Y with 20% T and 80% F is the same, because their distribution is the same (although in different categories).

For a flag field, the closer the probability for the true value is to 0.5, the higher the impurity (and not the smaller).

Answer 4: B. When income is used as a predictor, 40,000 yields the highest improvement. For income 40,000, the True and False categories will be perfectly separated: the target field is False for INCOME <= 40,000, and True for INCOME > 40,000. So, the impurity drops to 0 after splitting on INCOME =40,000.

Answer 5: A, B, C, and D. All four statements are valid reasons why CHAID and C&R Tree may produce a different tree.

Answer 6: A, B

When income category is missing, age is used as surrogate and a 27-year-old person will be handled as if he or she was in income category 1, 3, or 4.

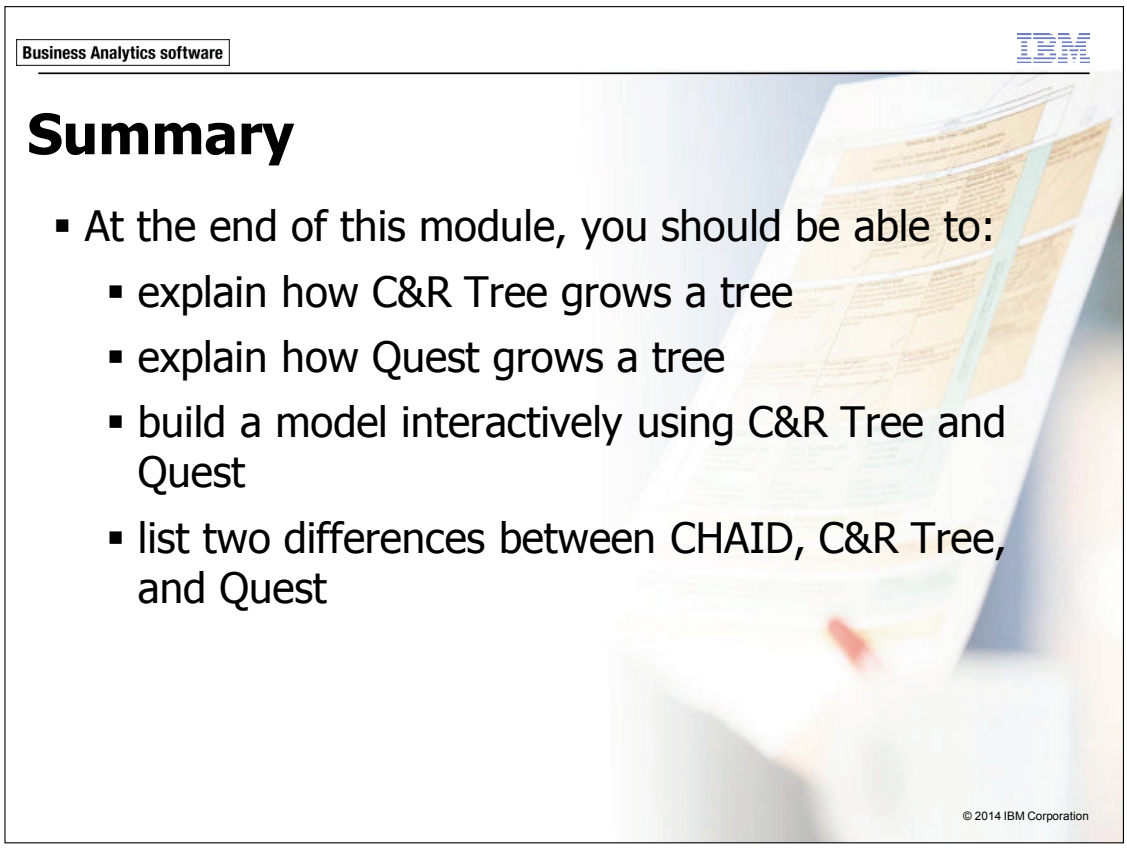
The income category was defined as nominal, because non-adjacent categories have been merged.

The impurity in the root node cannot be 0.58599, because impurity can never be greater than 0.5.

Answer 7: C. The impurity in Node 1, with 150 records, is 0.44. The impurity in Node 2, with 50 records, is 0. The impurity in loan paid back after the split in age category thus is (150 * 0.44 + 50 * 0) / 200 = 0.33. This means that the impurity is reduced by 0.50 - 0.33 = 0.17.

Answer 8: A, B, and C are all true.

Answer 9: A, B, C, and D. All four statements are true. AGE apparently is an ordinal or continuous field, because the F test is reported. The number in the Nodes column is always 2 when you use Quest. Men and women do not differ with respect to the target because the reported (adjusted) probability is 1. The column labeled DF is only a technical detail needed to compute probability values.

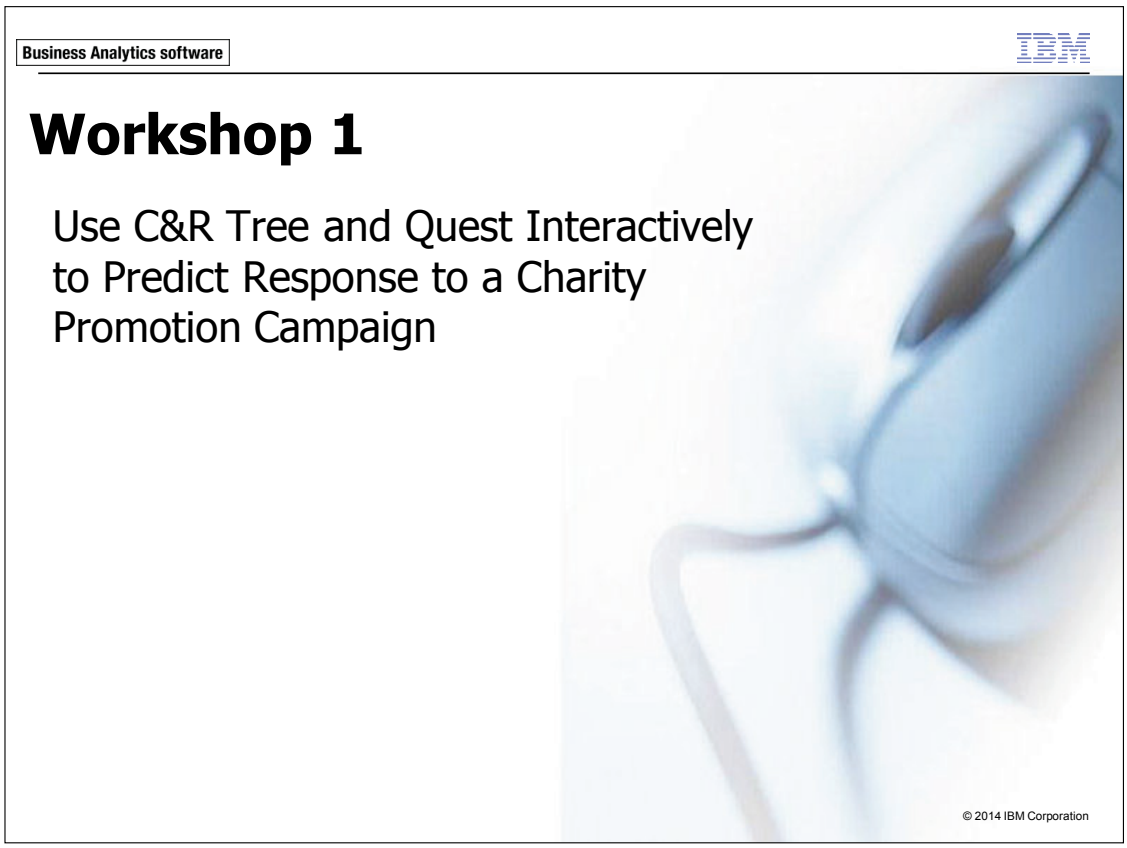


In this module, details were provided on two rule induction models: C&R Tree and Quest.

The key points to take away from this module are:

- The user can build the model interactively, giving, if desired, business knowledge higher priority than statistics.
- Each model has a different view on data, in terms of criteria to grow the tree, handling of categorical and continuous predictors, and handling of missing values. Therefore, you cannot expect the results to be the same.

In the end, it is a business decision which model you prefer.



The following (synthetic) file is used in this workshop:

- **charity.txt**: A text file that stores data from a charity organization. It contains information on individuals who were mailed a promotion. The information includes whether the individuals responded to the campaign, their spending behavior with the charity and basic demographics such as age, gender, and mosaic group. The file is located in **C:\Train\0A0U5**.

Before you begin the workshop, ensure that:

- You have started MODELER (click **Cancel** when the splash screen appears when you start MODELER).
- You have set MODELER's working folder. (In MODELER, select **File\Set Directory**. When you are working in an IBM training environment, browse to the folder **C:\Train\0A0U5** and then click **Set** to select it. When you are not working in an IBM training environment, ask your trainer for the folder name.)

Workshop 1: Use C&R Tree and Quest Interactively to Predict Response to a Charity Promotion Campaign

In this workshop you will use C&R Tree and Quest to interactively predict promotion campaign mail recipient response based on certain predictor fields. You will use charity.txt as the data source for this interactive evaluation. You will designate the following predictors: gender, age, mosaic brands, pre-campaign expenditure, and pre-campaign visits. Designate the response to campaign field as the target.

To do this, you must:

- Use a **Var. File** node to import data from **charity.txt**, and run a **Data Audit** to examine the data.

  What is the percentage of customers responding positively to the campaign?

- Add a **Type** node downstream from the **Var. File** node. Configure the **Type** node so that **response to campaign** will be predicted by **gender**, **age**, **mosaic bands**, **pre-campaign expenditure**, and **pre -campaign visits**.

- Add a **C&R Tree** node downstream from the **Type** node. Configure the **C&R Tree** node by setting the objective to launch an interactive session (**Build Options** tab, **Objective** item) and to use all records for model building (**Build Options** tab, **Advanced** item, set the **Overfit prevention set** to **0**).

- Run the **C&R Tree** node.

  Which predictor is most important, in terms of improvement? (Do not split the root node on this field.)

- Split the root node on **age** into the categories that are suggested by C&R Tree.

  Are these the same age categories as were suggested in the previous module by CHAID? If not, why not?

- Use the C&R Tree algorithm to further grow the tree automatically.

  What is the gain percentage for the best 20% customers?

  What is the risk of misclassifying a record when you use this tree?

  Next, you will run Quest interactively.

- Add a **Quest** node downstream from the **Type** node. Configure the **C&R Tree** node by setting the objective to launch an interactive session and to use all records for model building. Then, build a model similar to that of C&R Tree:
  - use **age** as first split field and at the same age value that you had in C&R Tree
  - use the Quest algorithm to further grow the tree automatically

  What is the gain percentage for the best 20% customers?

  What is the risk of misclassifying a record when you use this tree?

- Generate a model nugget for the current tree and use the model nugget to add propensities to the dataset.

  What is the probability to respond for the first record?

Workshop 1: Tasks and Results

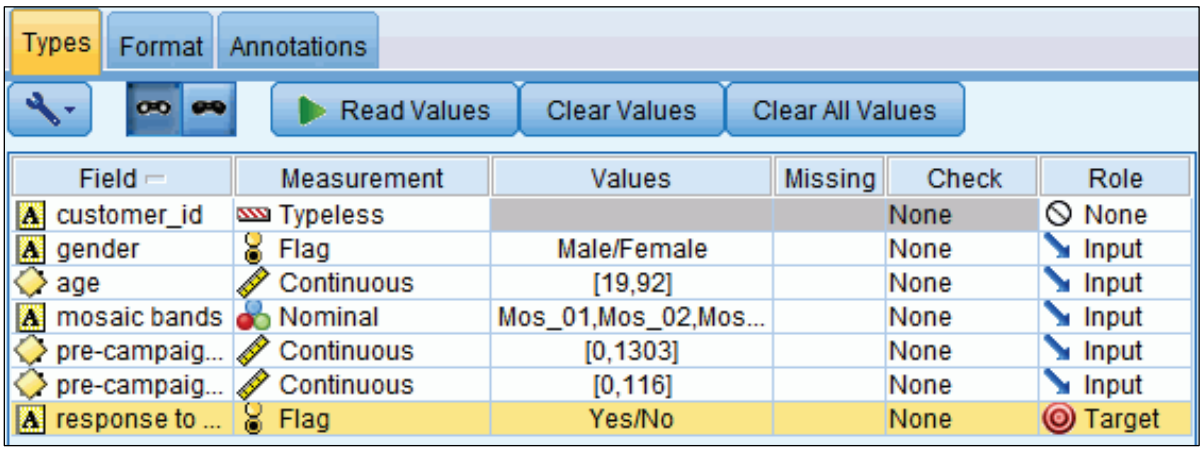
Task 1. Import and examine the data.

1. From the **Sources** palette, double-click the **Var. File** node to add it to the stream canvas.
2. Edit the **Var. File** node, and then:
   - to the right of the **File** box, click **Browse**  (the Browse window should automatically open to the **C:\Train\0A0U5 folder**)
   - select **charity.txt** and then click **Open**
   - close the **Var. File** dialog box
3. From the **Output** palette, add a **Data Audit** node downstream from the **Var. File** node, run the **Data Audit** node, and then double-click the **Sample Graph** for the **response to campaign** field.

   The Distribution output window shows that 31.32% responded to the campaign.
4. Close the **Distribution** output window, and then close the **Data Audit** output window.

Task 2. Set roles and instantiate the data.

1. From the **Field Ops** palette, add a **Type** node downstream from the **Var. File** node.
2. Edit the **Type** node, click **Read Values** and ensure that the **Measurement** and **Role** columns are set according to the following result:



Field	Measurement	Values	Missing	Check	Role
customer_id	Typeless			None	None
gender	Flag	Male/Female		None	Input
age	Continuous	[19,92]		None	Input
mosaic bands	Nominal	Mos_01,Mos_02,Mos...		None	Input
pre-campaig...	Continuous	[0,1303]		None	Input
pre-campaig...	Continuous	[0,116]		None	Input
response to ...	Flag	Yes/No		None	Target

3. Close the **Type** dialog box.

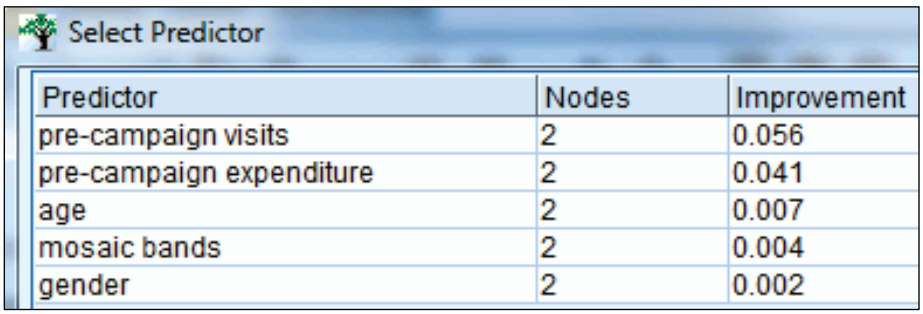
Task 3. Add and configure a C&R Tree node.

1. From the **Modeling** palette (**Classification** item), add a **C&R Tree** node downstream from the **Type** node.
2. Edit the **C&R Tree** node, and then:
   - click the **Build Options** tab
   - select the **Objective** item
   - select the **Launch interactive session** option
   - select the **Advanced** item
   - set **Overfit prevention set (%)** to **0**

Task 4. Run C&R Tree.

1. Click **Run** in the **C&R Tree** dialog box.
2. In the **Interactive Tree Builder** window, select **Tree\Grow Branch with Custom Split** and then click the **Predictors** button.

   A section of the results appear as follows:



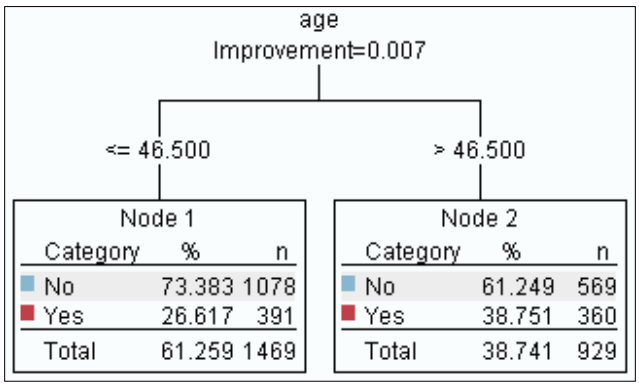
Predictor	Nodes	Improvement
pre-campaign visits	2	0.056
pre-campaign expenditure	2	0.041
age	2	0.007
mosaic bands	2	0.004
gender	2	0.002

   The pre-campaign visits field is listed first, because it yields the largest improvement. You will not use this field to grow the tree, but you will customize tree growth.
3. Click **Cancel** to close the **Select Predictor** window.

Task 5. Grow the tree with age.

1. In the **Define Split** window, from the **Predictor** list, click **age**, and then click **Grow**.

   A section of the results appear as follows:



   The categories differ from those that CHAID suggested in the previous module. C&R Tree finds the optimal split to maximize improvement, where CHAID binned age into deciles and then merged categories. Also, CHAID recommended four categories. But C&R Tree always performs a binary split.

Task 6. Grow a complete tree and examine gain and risk.

1. From the menu, click **Tree\Grow Tree**.

   The tree grows to a complete model.

2. Click the **Gains** tab, and in the **Target category list**, select **Yes**.

3. Click the **Cumulative** button, click the **Quantiles** button and then ensure that **Decile** is selected.

   A section of the results appear as follows:

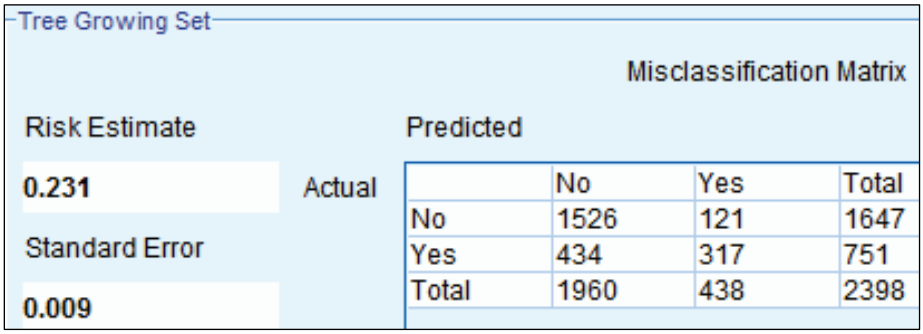
Tree Growing Set

Nodes	Percentile	Percentile: n	Gain: n	Gain (%)	Response (%)	Index (%)
21,14,46,48,38,22	10.00	240.00	202.00	26.90	84.19	268.81
22,47,37,18,36	20.00	480.00	336.00	44.72	69.97	223.43
36,34,45,27	30.00	719.00	432.00	57.54	60.11	191.92
27,16,44	40.00	959.00	517.00	68.77	53.86	171.97

   The top 20% records have a gain percentage of 44.72%.

4. Click the **Risks** tab.

   A section of the results appear as follows:



Tree Growing Set

Misclassification Matrix

Risk Estimate		Predicted		
0.231	Actual	No	Yes	Total
Standard Error	No	1526	121	1647
0.009	Yes	434	317	751
	Total	1960	438	2398

   There is a risk of 0.231 to misclassify a record when you use this tree as your model.

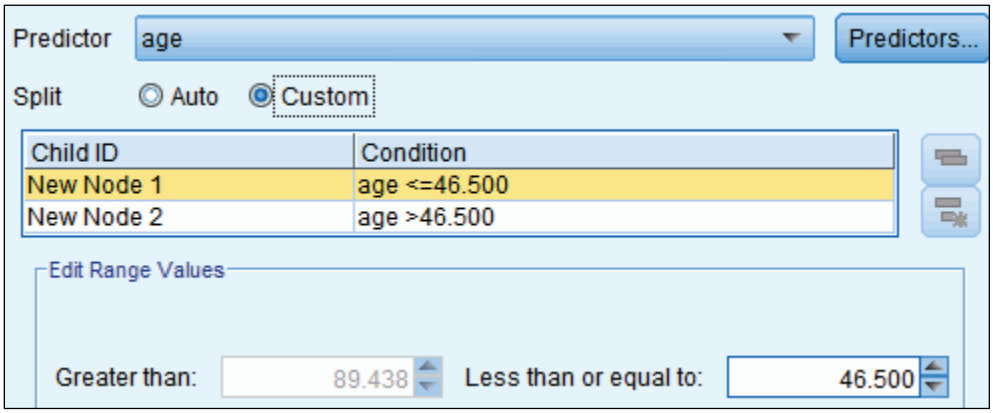
5. Close the **Interactive Tree Builder** window.

Task 7. Add, configure and run Quest.

1. From the **Modeling** palette, add a **Quest** node downstream from the **Type** node.
2. Edit the **Quest** node, and then:
   - click the **Build Options** tab
   - select the **Objective** item
   - select the **Launch interactive session** option
   - select the **Advanced** item
   - set **Overfit prevention set (%)** to **0**
   - click **Run**
3. In the **Interactive Tree Builder** window, select **Tree\Grow Branch with Custom Split** and in **Define Split** window, select **age**.

4. For **Split**, select the **Custom** option, and then in the **Edit Range Values** pane, for **Less than or equal to**, type **46.5** (which was the value that C&R Tree used for the split).

   A section of the results appear as follows:



5. Click **Grow**.
6. Select **Tree\Grow Tree**.
7. Click the **Gains** tab, and in the **Target category list**, select **Yes**.
8. Click the **Cumulative** button, click the **Quantiles** button and then ensure that **Decile** is selected.

   A section of the results appear as follows:

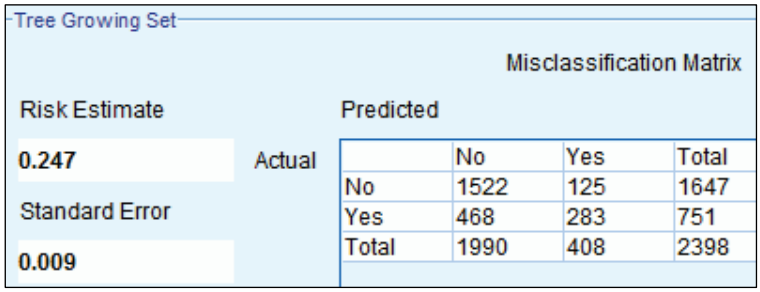
Tree Growing Set

Nodes	Percentile	Percentile: n	Gain: n	Gain (%)	Response (%)	Index (%)
14,10,13,12	10.00	240.00	182.00	24.19	75.71	241.75
12,9,18,20	20.00	480.00	315.00	41.93	65.61	209.49
20,16,24	30.00	719.00	409.00	54.53	56.95	181.85

The top 20% records have a gain percentage of 41.93%.

9. Click the **Risks** tab.

   A section of the results appear as follows:


Tree Growing Set

Misclassification Matrix

Risk Estimate

0.247

Standard Error

0.009

Predicted

Actual	No	Yes	Total
No	1522	125	1647
Yes	468	283	751
Total	1990	408	2398

   There is a risk of .247 to misclassify a record when you use this tree as your model.

Task 8. Score records.

1. Click the **Viewer** tab.
2. Select **Generate\Generate Model**, and then click **OK** in the dialog box that displays.
3. Close the **Interactive Tree Builder** window.
4. Add the generated **model nugget** downstream from the **Type** node.
5. Edit the **model nugget**, click the **Settings** tab, enable the **Calculate raw propensity scores** option, and then click **Preview**.
6. Scroll to the last fields in the **Preview** window.

   A section of the results appear as follows:

response to campaign	$R-response to campaign	$RC-response to campaign	$RRP-response to campaign
Yes	No	0.712	0.288
Yes	No	0.712	0.288
Yes	No	0.862	0.138
Yes	Yes	0.597	0.597

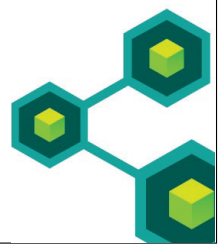
   The first record has a propensity of 0.288 to respond positively to the mailing.

7. Close the **Preview** window, and then close the **model nugget**.
8. Exit MODELER without saving anything.

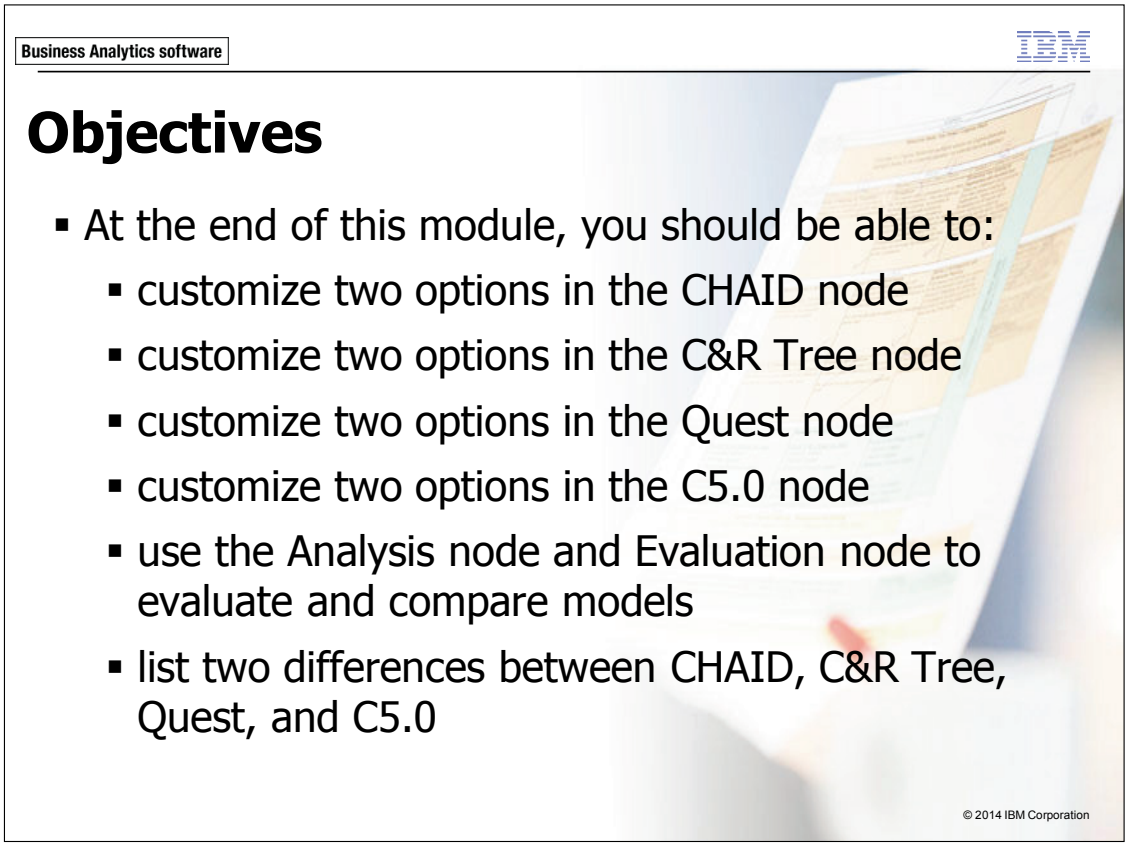
Note: The stream **workshop_building_your_tree_interactively_with_c&r_tree_and_quest_completed.str**, located in the **03-Building_Your_Tree_Interactively_with_C&R_Tree_and_Quest\Solutions** sub folder, provides a solution to the workshop tasks.

Building Your Tree Directly

IBM SPSS Modeler (v16)



Business Analytics software



Before reviewing this module, you should be familiar with the following topics:

- working with MODELER (streams, nodes, palettes)
- importing data (Var. File node)
- defining measurement levels, roles, blanks, and instantiating data (Type node)
- examining the data (Table node, Data Audit node)
- using CHAID, C&R Tree and Quest in interactive mode
- assessing the quality of your model, using accuracy, risk estimate, gain and response measures
- using the model nugget to score data

Business Analytics software

IBM

Building Your Tree Directly

- Set the options in the modeling node
- Run the modeling node
- Evaluate the model nugget that was generated
- Use the model to score records

CHAID, C&R Tree and Quest enable you to grow your tree interactively. Interactively building your model enables you to let your business knowledge prevail, guided by statistics, if required.

In this module these three algorithms are revisited, now focusing on the direct mode. The direct mode creates a model nugget right away.

When presenting these models, extra options will be discussed. Although discussed in this module, these options also apply when you work interactively. For example, when you select the Exhaustive CHAID method and run the CHAID node interactively, the Exhaustive CHAID algorithm will be used rather than the default algorithm.

This module will also introduce you to a fourth model named C5.0. C5.0 grows a tree, but is available only in direct mode.

You will not have the Interactive Tree Builder window to assess the gains and risk when you work in direct mode, but the Analysis node and the Evaluation node will enable you to evaluate and compare the models. You can then choose the best model to score records, or, if required, combine models into a single model.

Business Analytics software

IBM

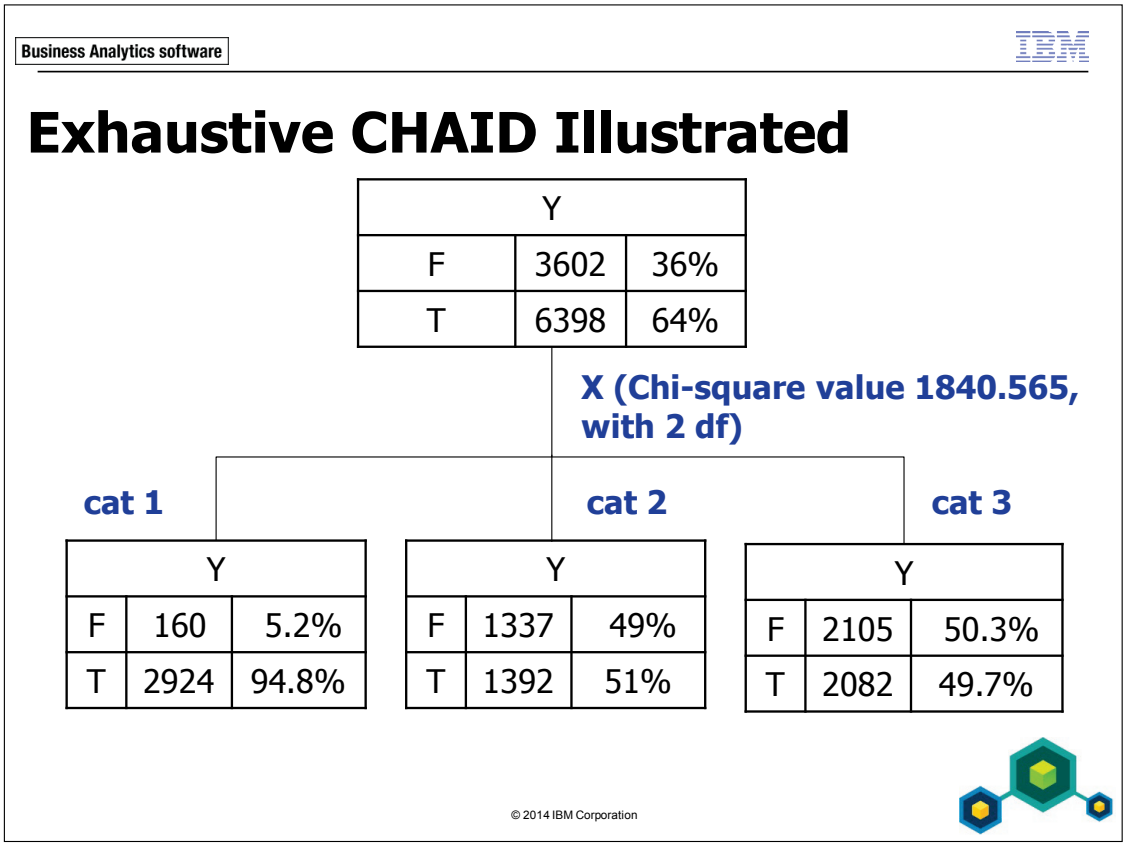
Choosing Your CHAID Tree Growing Algorithm

- CHAID:
  - merges categories until categories cannot be merged because they are all significantly different
- Exhaustive CHAID:
  - continues merging categories until two categories remain
  - records the significance for each merge step
  - selects the merge for the step that yielded the highest significance

Exhaustive CHAID (developed by Biggs, de Ville, and Suen, 1991) was developed to address some of the weaknesses of CHAID.

When merging categories for a predictor, Exhaustive CHAID continues merging the least significantly different categories until only two nodes remain, while CHAID stops when significantly different nodes remain. Exhaustive CHAID records the significance for each merge step and then selects the merge that yielded the highest significance (lowest probability value).

By carrying the merging operation further, Exhaustive CHAID examines additional split possibilities. Thus, for a given predictor, Exhaustive CHAID selects the split showing the greatest statistical significance from a larger (greater than or equal to) set of possible splits than does CHAID. Exhaustive CHAID requires more processing time but improves your chance of finding category splits that make the best predictions.



As an example, consider a target Y and a predictor A with three categories, depicted here. The Chi-square value is1840.565, with 2 degrees of freedom.

The regular CHAID method will merge categories 2 and 3, because these categories do not significantly differ from each other (the probability value for this sub table is 0.297). Thus, the end result of regular CHAID is a tree where X is split in a node that contains X = cat 1, and another node that contains X= cat 2, cat 3. The Chi-square value for this split is 1839.386, with 1 df. Exhaustive CHAID also will also merge categories 2 and 3, because it will always merge the least significant categories until only two nodes remain. But Exhaustive CHAID has recorded the probability value for all splits that were examined. And, the probability value for the original split on X with three categories (Chi-square value 1840.565 with 2 degrees of freedom) is smaller than the probability value for the split in two categories (Chi-square value 1839.386 with one degree of freedom), so Exhaustive CHAID will retain the original tree as split.

Business Analytics software

IBM

Using Boosting to Improve Accuracy

- Boosting builds models iteratively:
  - Misclassified records are more heavily weighted in next run
- Boosting combines the component models into one ensembled score.

CHAID supports boosting. Boosting tries to improve the accuracy of a model.

As the first step, a single tree is constructed. This tree will usually misclassify some records. These misclassified records are weighted more heavily in building a second tree, trying to get these records right. This will result in a different tree, but again some records will be misclassified and a third tree will focus on these records by giving them a heavier weight. This process continues for a pre-determined number of trials.

Each trial produces a tree. These individual models are referred to as component models. To arrive at the prediction when boosting is applied, you have various options in how you want to combine the predictions from the component models:

- Majority voting: The category that is predicted most often will be assigned.
- Highest probability: Selects the category that has the highest probability across all component models. Highest mean probability: Selects the category with the highest value when the category probabilities are averaged across the component models.

Although boosting can provide a more accurate prediction, it will take longer to train.

Business Analytics software

IBM

Using Bagging to Improve Generalizability

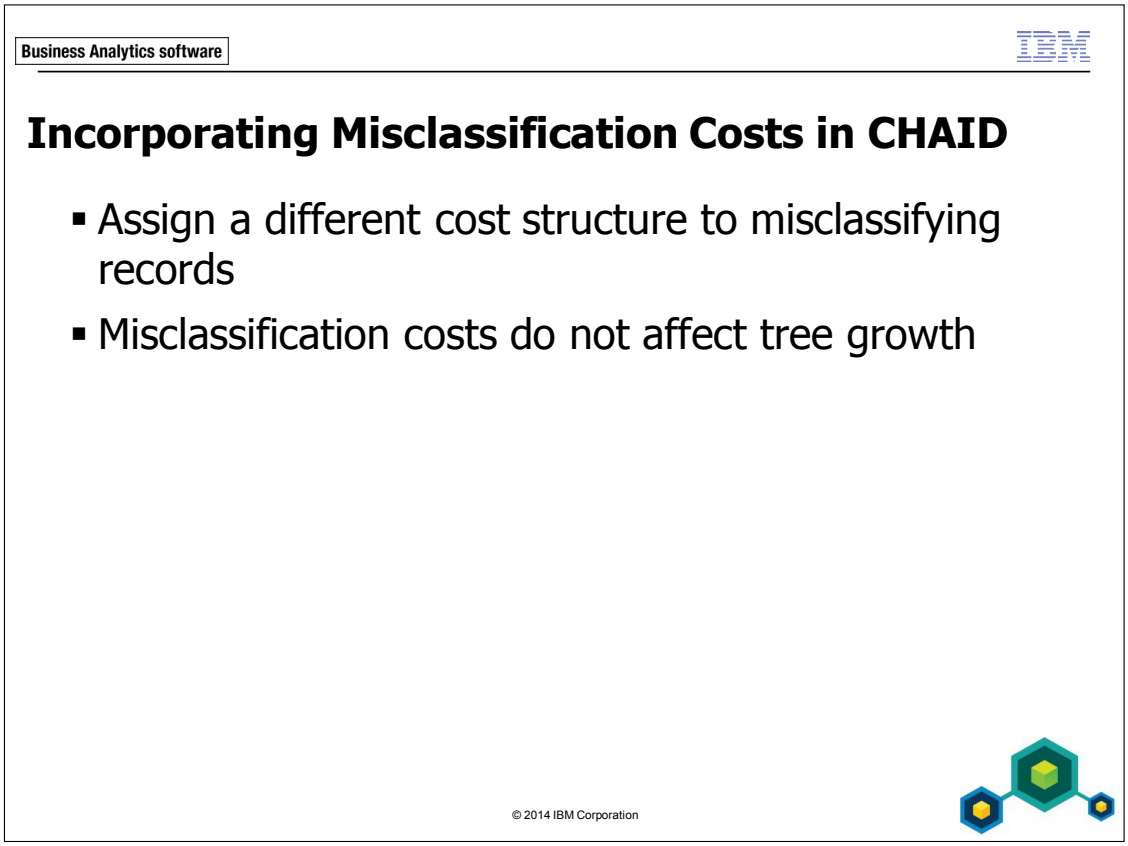
- Draw a pre-determined number of samples 8samples with replacement)
- Combine the component models into one ensembled score

Where boosting tries to improve accuracy, bagging (Bootstrap Aggregating) focuses on the stability or generalizability of the model. Similar to boosting, bagging creates component models, but it will try to obtain more reliable predictions.

Bagging builds multiple models by bootstrapping. In the bootstrapping procedure samples are drawn with replacement, of the same size as the dataset. A model is built on each sample and the predictions from the component models are combined into a single prediction, using the same ensemble methods as boosting does.

When you achieve a higher accuracy with boosting than with bagging, this means that a small portion of the data behaves differently with respect to the target. Boosting will pick up such patterns because it will give these records extra weights. Bagging, on the other hand, just samples from the same dataset and the anomalous records will always be a small portion in each sample, so bagging most likely will not pick up the patterns.

When experimenting with boosting and/or bagging, make sure that you validate the model again a testing dataset.



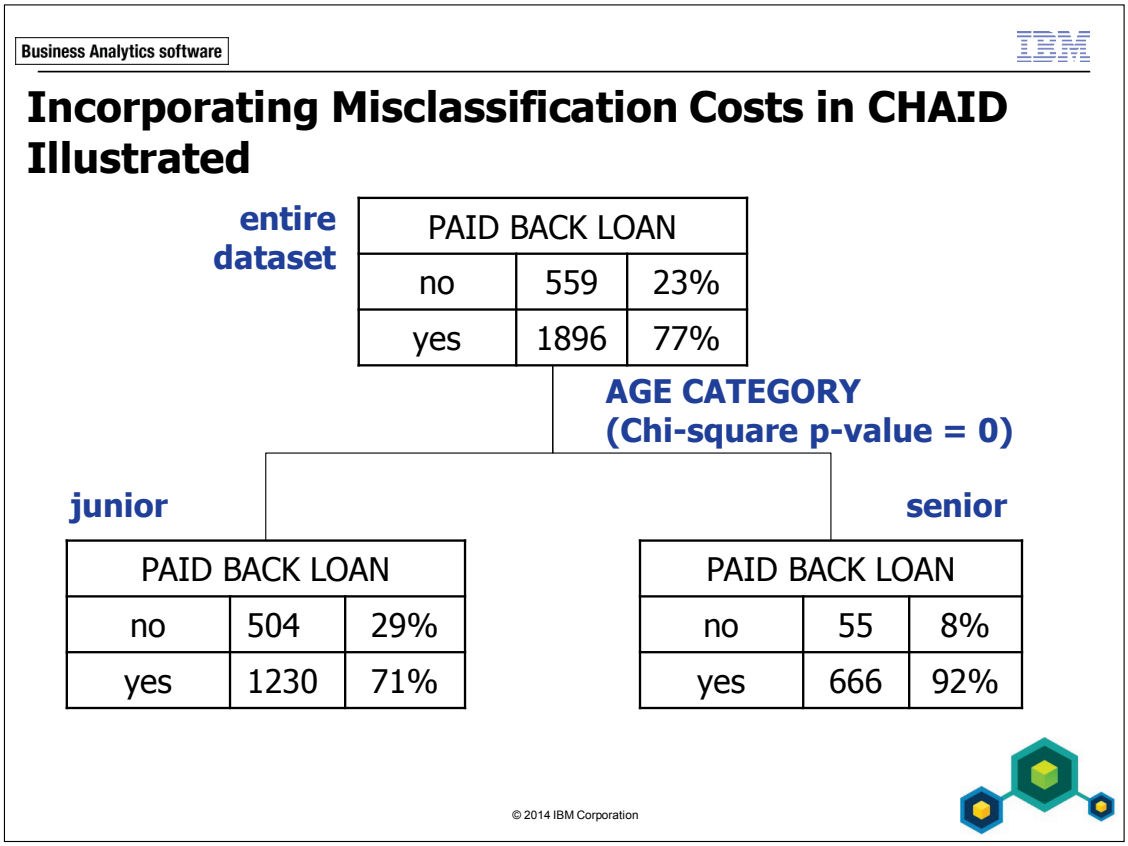
CHAID classifies all records in a terminal node into the node's most popular target category. This is true when all misclassification costs are equal. But it may be that the costs of misclassification are not the same for each category of the target.

For example, suppose that a patient is classified into one of two categories: free of disease and serious level of disease. Erroneously classifying a patient with a serious level of disease in the disease-free category is probably a more costly error than incorrectly assigning a disease-free patient to a serious level of disease.

CHAID can account for misclassification costs when classifying records. Misclassification costs do not influence CHAID tree growth. They are only reflected in assignment rules, which will impact accuracy and risk.

To use misclassification costs you should have a reasonable estimate of the costs involved.

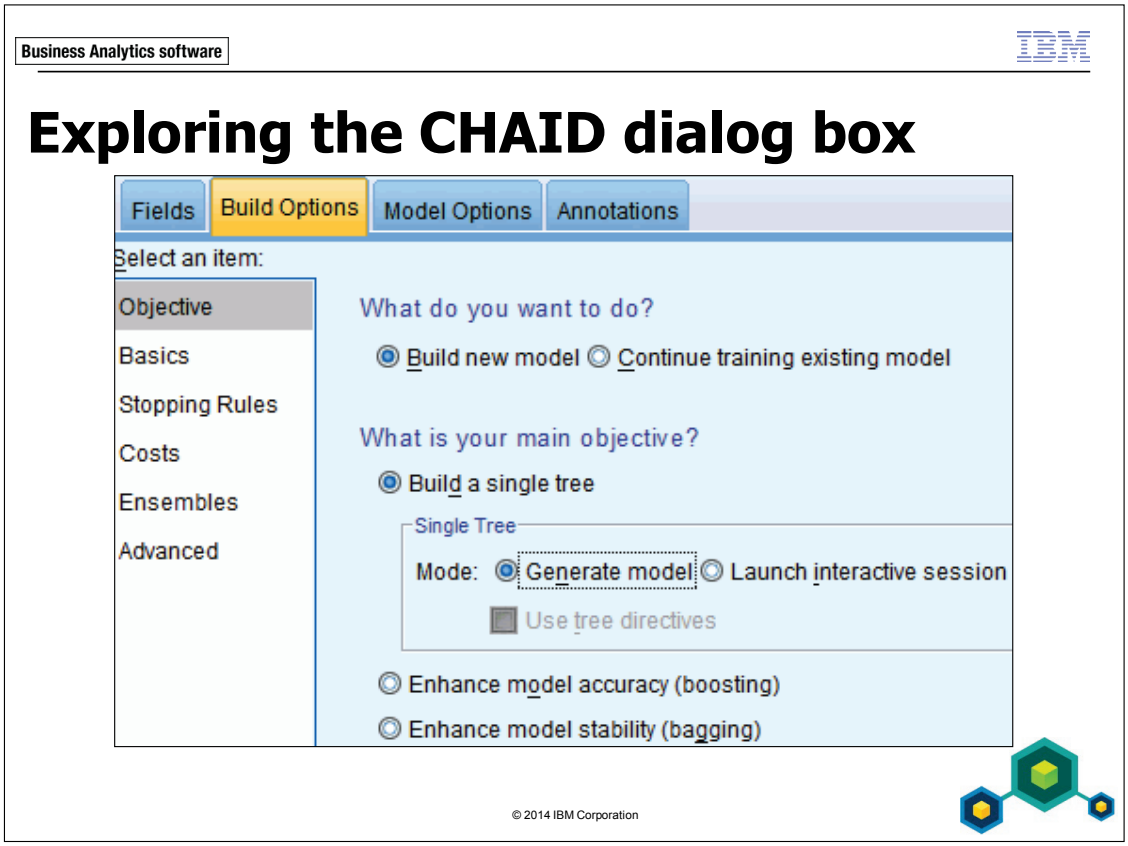
Note: The risk estimate can attain values greater than 1when you use misclassification costs.



As an illustration, consider the tree depicted on this slide.

When the misclassification costs are equal, the predicted value will be PAID BACK LOAN=yes, because that is the most frequent value for each age category. In this scenario, 504 juniors and 55 seniors will be incorrectly classified, bringing the misclassification costs to 559.

Now suppose that classifying PAID BACK LOAN=no incorrectly as PAID BACK=yes costs 10 times as much as misclassifying the PAID BACK LOAN=yes category as a no. For juniors the misclassifications costs then are 10 * 504 = 5040 when the PAID BACK LOAN=no category is predicted as yes, while classifying the PAID BACK LOAN=yes category as no would bring a cost of 1230. Thus, to minimize misclassification costs the best prediction is category PAID BACK LOAN=no for juniors. For seniors the best prediction is PAID BACK LOAN=yes. The total misclassification costs are 1230 + 550 = 1780. The risk estimate then is 1780 / 2455 = 0.725.



The Build Options tab enables you to customize model building:

- On the Objective item, select whether to build a single tree, or to use boosting or bagging.
- On the Basics item, select the tree growing algorithm, CHAID or Exhaustive CHAID. Also, determine the maximum tree depth (5 by default).
- On the Stopping Rules item, determine the minimum number of records in the parent node and in child nodes, either as an absolute number, or as a percentage.
- On the Costs item, set the misclassification costs.
- On the Ensembles, specify how predictions from the component models must be combined into a single prediction. Also, set the number of trials.
- On the Advanced item, set the significance levels for predictor selection and merging categories, amongst more advanced options.

Business Analytics software

Using C&R Tree to Directly Grow Your Tree: Pruning

- Trees grown with the original C&R Tree algorithm:
  - tended to be too large
  - did not replicate well
- Prune (*) trees to overcome these issues.

* prune a tree = cut branches in a fully grown tree.

The originators of C&R Tree found that when trees were grown, the trees tended to be too large and results did not replicate as well as desired. They found that if they allowed a tree to grow large but then pruned it, the result was smaller trees with better validation properties.

Breiman et al. (1984) recommend generating a large tree (perhaps with as few as five records or even one record per node) before pruning. For large datasets this is computationally intensive. However, the point is that they believe that a good solution is more likely when you grow a larger tree before pruning.

Tree growth followed by pruning became the foundation of the C&R Tree method. Reading the Breiman et al. book is recommended. Although proofs are derived, the authors present a number of examples and describe the experience that led them to make certain design choices (for example, cost-complexity pruning). This perspective is helpful and the recommendations they provide are valuable, especially for an inexperienced analyst.

Business Analytics software

IBM

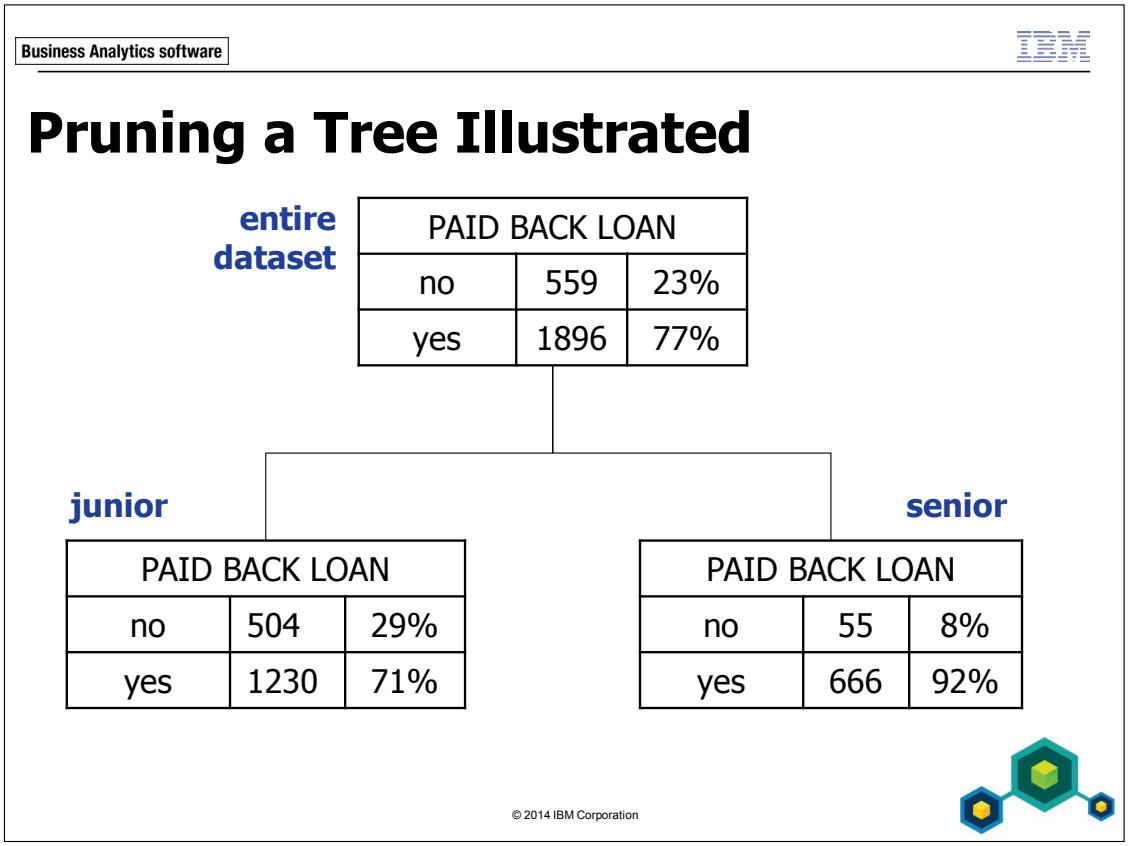
How C&R Tree Prunes Trees

- Uses:
  - risk estimate
  - standard error of the risk estimate
- Procedure:
  - grow a full tree
  - prune the tree back to a tree with a similar risk as the fully grown tree

Central to the process of pruning is the risk estimate and its standard error.

The risk estimate gives the probability to make an incorrect prediction. Taking the data as a sample, this statistic is an estimate of the unknown (population) risk estimate, so an error margin for the risk estimate has to be taken into account. This is the standard error. The standard error indicates the difference in risk estimate that you can expect from sample to sample.

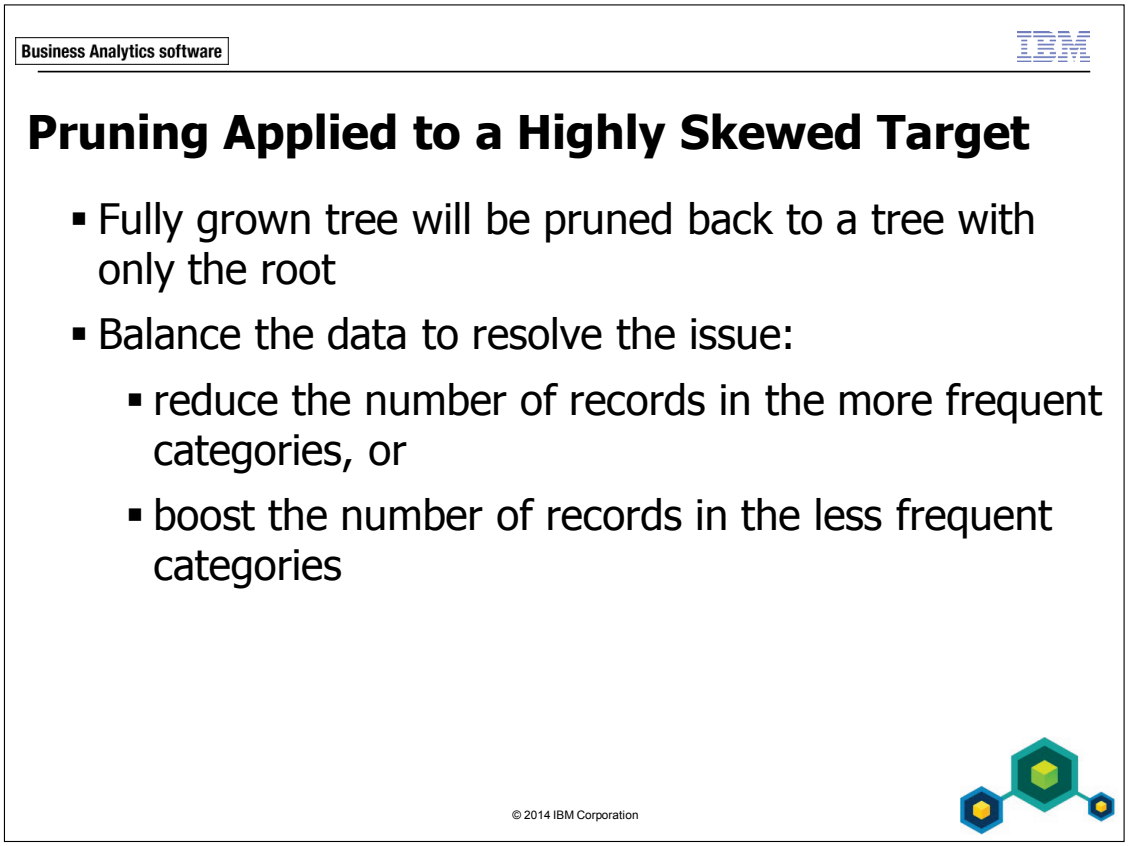
The idea of pruning is to fully grow a tree, and then prune this tree back to a tree that does not differ substantially in risk estimate from the fully grown tree. In C&R Tree, a pruned tree is not substantially different from the fully grown tree when the risk estimate of the pruned tree falls within the range of the risk estimate of the fully grown tree plus one standard error.



As an example, consider the tree that is depicted on this slide. Suppose that this is the fully grown tree. In this tree, the model always predicts yes. The accuracy for this model is (1896/2455) * 100 = 77.8% and the risk estimate equals 1- 0.778 = 0.222. The standard error is 0.008 (not shown here).

Now, consider the tree with only the root node. The predicted category is again yes. The risk estimate for this tree (with only the root node) is identical to that of the fully grown tree. Thus, the tree with only the root node is within the range of the risk estimate of the fully grown tree plus one standard error. This implies that the two trees are equally good from a perspective of risk. And because a parsimonious tree is preferred over a tree with many terminal nodes, the tree with the smallest number of terminal nodes wins.

In summary: although C&R Tree grows a full tree, a less complex tree can have a risk estimate that is not significantly differently from that of the fully grown tree, and thus the latter tree is preferred because it is more parsimonious and it has a higher generalizability.



When the distribution of the target is heavily skewed in favor of one of the categories, you may encounter problems when growing a tree using C&R Tree, because the risk estimate of the tree with only a root will not differ from the risk estimate of the fully grown tree.

One solution to overcome this problem is to balance the data, which will overweight the less frequent categories. This can be accomplished with the Balance node. The Balance node reduces the number of records in the more frequent categories, or boosts the number of records in the less frequent categories.

It is recommended to use the reduce option in preference to the boosting option. The latter duplicates records and thus magnifies problems and irregularities, as only a relatively few cases can be heavily weighted. However, when working with small datasets, reducing is often not feasible and boosting is the only sensible solution to imbalances within data.

Note: A highly skewed target does not pose any problem for CHAID, because CHAID does not prune a tree.

Business Analytics software

IBM

Exploring C&R Tree

- C&R Tree supports bagging and boosting.
- C&R Tree incorporates misclassification costs:
  - affects tree growth
  - affects classification
- C&R Tree can set prior probabilities.
- Presents a dialog box similar to CHAID
  - options relate to impurity (predictor selection), standard error (pruning)

C&R Tree incorporates misclassification costs when the tree is grown, so, contrary to CHAID, misclassification costs affect tree growth.

Also, prior probabilities for each of the categories of the target field can be assigned. Usually, custom priors are specified if the expected frequencies for a population are known.

C&R Tree supports boosting and bagging, to improve accuracy and generalizability, respectively. Refer to the CHAID section for a presentation of boosting and bagging.

The C&R Tree dialog box shares many items with the CHAID dialog box. In C&R Tree, however, the settings relate to pruning and to the impurity statistic. On the Basics item you can enable or disable the pruning option and when enabled you can set the parameter to assess the difference in risk. If you want to prune the tree more severely, set this parameter to a higher value than the default of 1.

As one of the options on the Advanced tab, consider setting the Minimum change in impurity to a smaller value (default 0.0001) when you get the message no tree will be built because stopping criteria have been met.

Business Analytics software

Exploring Quest

- Quest prunes tree.
- Quest supports bagging and boosting.
- Quest incorporates misclassification costs:
  - affects tree growth
  - affects classification
- Quest can set prior probabilities.
- Presents a dialog box similar to CHAID and C&R Tree
  - options relate to significance (predictor selection) and standard error (pruning)

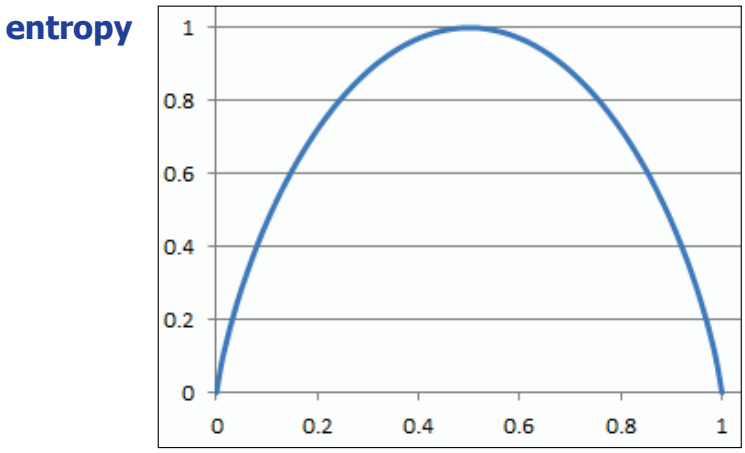
This slide summarizes the capabilities of Quest.

Business Analytics software

IBM

Using C5.0 to Grow Your Tree

- Uses entropy as statistic to grow the tree



Next to CHAID, C&R Tree and Quest, MODELER offers a fourth modeling node that grows a tree: C5.0. C5.0 does not use significance to grow a tree, but an information gain criterion, the so-called entropy. C5.0 differs from C&R Tree in that the tree built is not necessarily binary.

Entropy measures the dispersion in a categorical field. Entropy can be regarded as an alternative for the impurity statistic that is used in C&R Tree. Both impurity and entropy measure the dispersion in the target node and both create a tree that minimizes this dispersion. Formally, entropy for a flag field Y is defined as:

$$\text{Entropy } Y = - p * \text{Log}^2 (p) - (1-p) * \text{Log}^2(1-p)$$

For a flag field, entropy reaches its maximum value of 1 for p=0.5, and approaches 0 when p is close to 0 or 1. Thus, the larger the entropy, the more balanced the two categories of the flag field are.

Business Analytics software

IBM

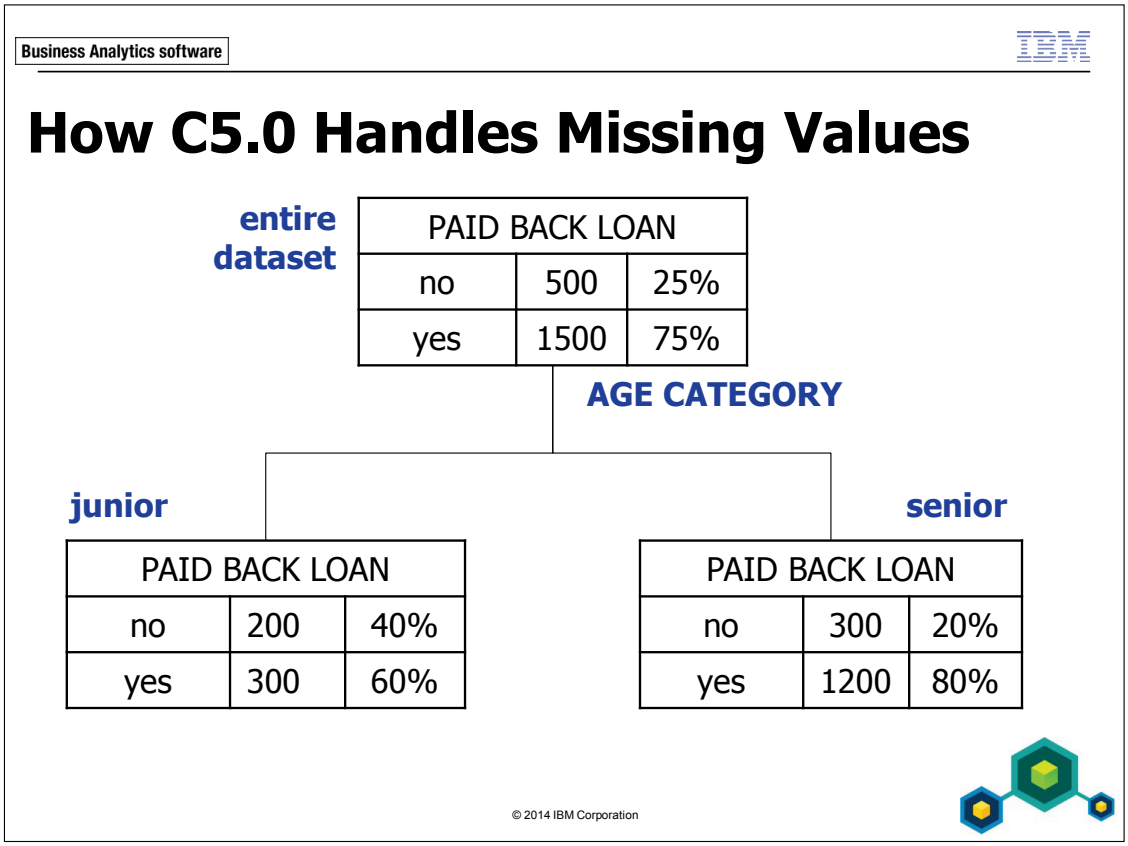
How C5.0 Grows a Tree

- For a flag target Y, and a categorical predictor X:
  - Information $Gain_X$ = Entropy Y – Entropy Y_X
- Adjust the Information Gain:
  - Gain $Ratio_X$= Information $Gain_X$ / Split $Info_X$

Similar to C&R Tree, C5.0 evaluates how much of the entropy can be reduced when a certain predictor is taken into account. In the formula that is depicted on this slide Entropy Y is the entropy for Y on itself and Entropy Y¦X is the entropy in Y given the information on X (computed as a weighted average of the entropies in Y for each category of X).

Although the Information Gain criterion gives good results, it has a flaw in that a categorical predictor with many values has an advantage over one with a few values. The Gain Ratio criterion rectifies this problem by dividing by a factor that depends on the number of categories of X.

The C5.0 algorithm will choose the field that maximizes the Gain Ratio. This maximization is subject to the constraint that the information gain must be at least as great as the average gain over all fields examined. This constraint avoids the instability of the Gain Ratio criterion, when the split is near trivial and the gain in information is thus small.

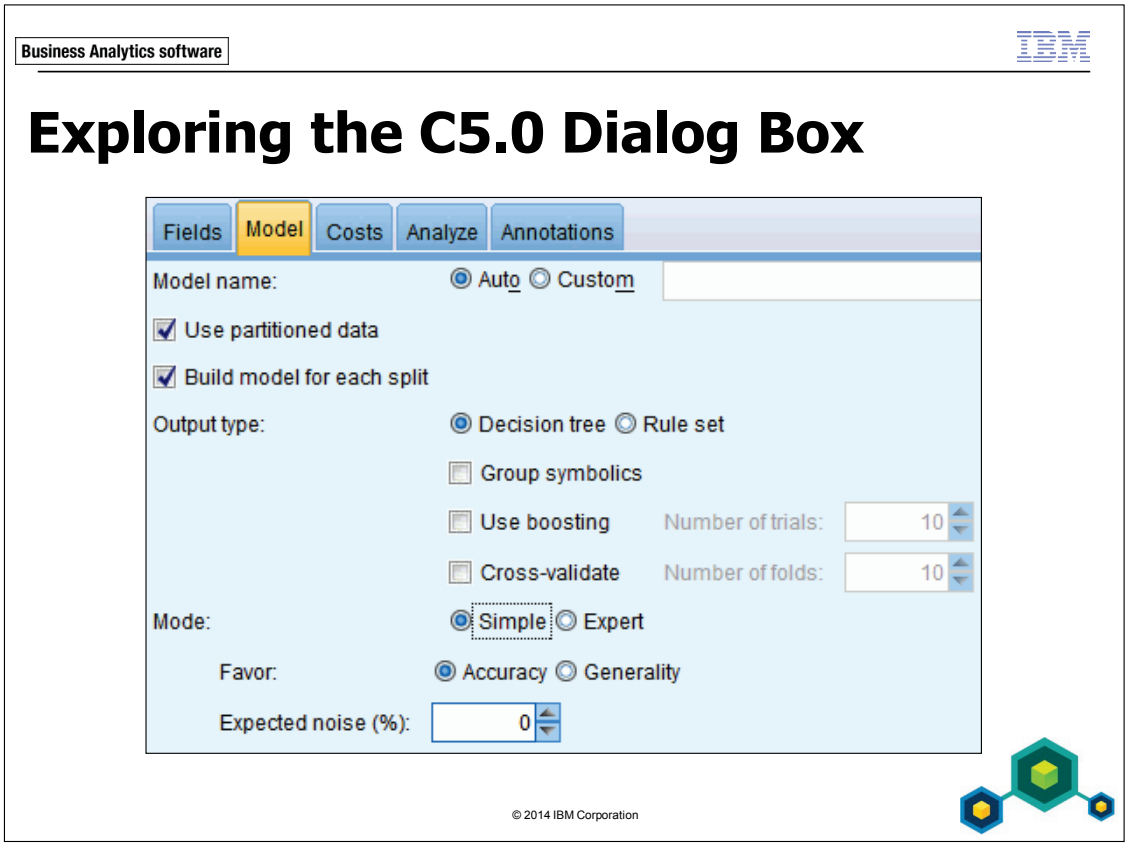


A record is discarded from model building when the target is missing or when all the predictors are missing. If only the predictor is missing then C5.0 splits a record in proportion to the distribution of the predictor field and passes a weighted portion of the record down each tree branch. This approach is known as fractionalization.

As an example, suppose that a field AGE CATEGORY has two categories, with 499 juniors and 1,497 seniors. For the sake of argument, let us say that this is a 25% - 75% distribution. Now suppose 4 records have their age category missing. Also, assume that all four records did pay back their loan.

C5.0 will fractionalize each record and assigns 25% of the record to the junior branch and 75% to the senior branch. Doing this for all four records, this will result in one record assigned to the junior group and three records assigned to the senior group. Also, the counts for the PAID BACK LOAN = yes will be updated accordingly.

Compared to using surrogates, as C&R Tree and Quest do, the advantage is that only the fields that are used in the tree need to be available in order to score new datasets.



Within C5.0, once the tree has been built, it is pruned back to create a more general (and less bushy) tree. The algorithm used to decide whether a branch should be pruned back toward the parent node is based on comparing the predicted error for the "sub-tree" (unpruned branches) with those for the "leaf" (or pruned node).

As in C&R Tree, the risk estimate and its standard error play a key role in this process. If the risk estimate of the leaf is within a certain standard error margin of the risk estimate of the sub-tree, the leaf will be preferred. The error margin is determined by the so-called pruning severity. Pruning severity ranges from 0 (no pruning) to 100 (maximum pruning). Higher values will lead to a less bushy tree.

A second phase of pruning (global pruning) is then applied by default. It prunes further based on the performance of the tree as a whole, rather than at the sub-tree level considered in the first stage of pruning.

One other consideration when building a general decision tree is that the terminal nodes within the tree are not too small in size. In Expert mode, you can set the minimum records per child branch, which specifies that at any split point in the tree, at least two sub-trees must cover at least a certain number of records. The default is two records but increasing this number can be useful for noisy datasets and will produce less bushy trees.

C5.0 can examine the usefulness of the predictors before starting to build the model. Predictors that are found to be irrelevant are then excluded from the model-building process. This option is called winnow attributes. This option can produce a model that uses fewer predictors yet maintains nearly the same accuracy, which can be an advantage in model deployment. Also, this option can be effective when there are many predictors and where predictors are statistically related.

C5.0 allows for boosting to achieve a higher accuracy. Cross-validation is useful when you have too few records to permit assessing accuracy on a separate testing set. Cross-validation partitions the data into N equal-sized subgroups and fits N models. Each model uses (N-1) of the subgroups for training, then applies the resulting model to the remaining subgroup and records the accuracy. Accuracy figures are pooled over the N holdout subgroups and this summary statistic estimates model accuracy applied to new data. Since N models are fit, cross-validation is more resource-intensive and it reports the accuracy statistic, but it does not present the N decision trees or rule sets.

Similar to C&R Tree, you can incorporate misclassification costs (refer to the Costs tab), and they will affect tree growth and classification.

Note: The code for C5.0 is licensed from RuleQuest Research Ltd Pty, and the algorithms are proprietary. Exact details on how C5.0 groups categories, determines cut-off values for continuous fields, and so forth are not found in the documentation that ships with MODELER. For more information on C5.0, refer to the RuleQuest website at http://www.rulequest.com/.

Business Analytics software

IBM

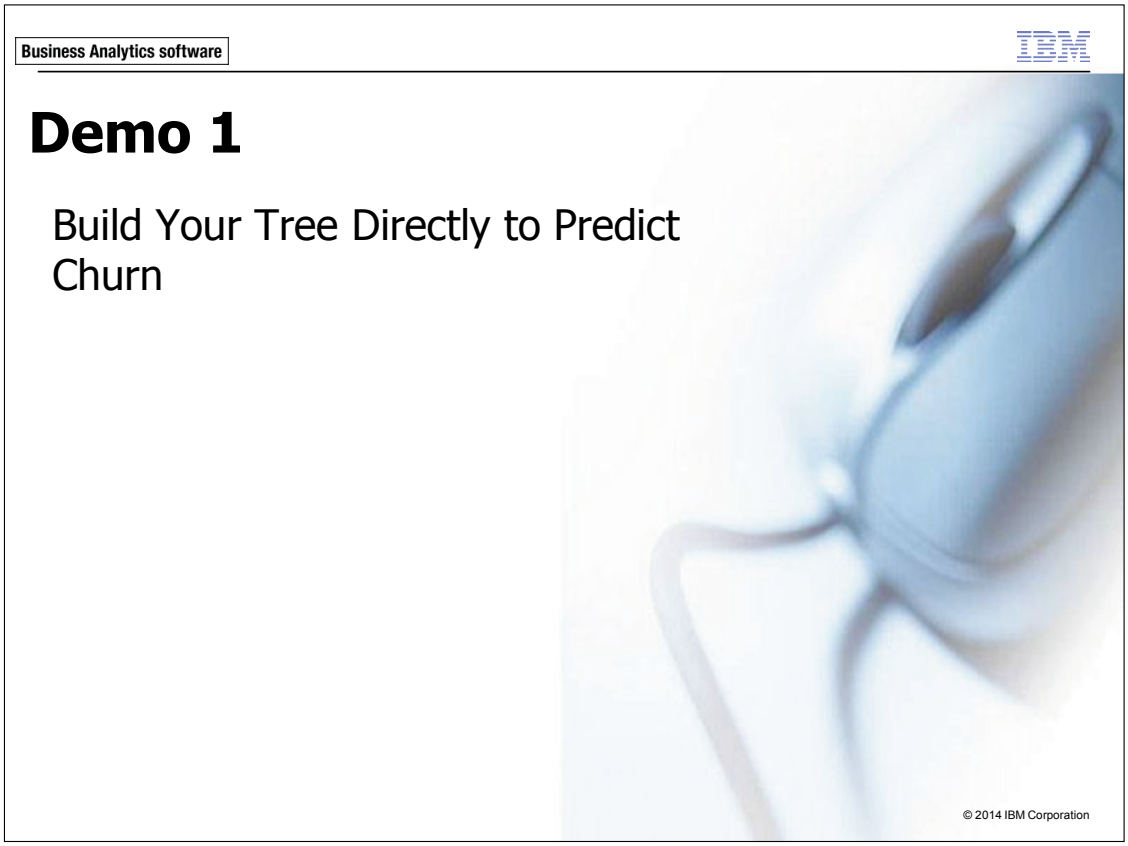
Determining the Tree Method to Use

CRITE-RION	CHAID	C & R Tree	Quest	C5.0
interactive	yes	yes	yes	no
binary tree	no	yes	yes	no
bagging	yes	yes	yes	no
pruning	no	yes	yes	yes
misclassi-fication costs	affect classifica-tion	tree growth; classification	tree growth; classification	tree growth; classification
missing values	separate category	uses surrogates	uses surrogates	fractionali-zation

This slide lists the differences between the various algorithms. Refer to the *Building Your Tree Interactively with C&R Tree and Quest* module for more differences.

There are also similarities. All methods support boosting. The model nuggets that are generated by the various models can all add the predicted category, confidence score, and propensities..

A binary split does not necessarily have to result in a different rule set from a non-binary tree. For example, suppose that CHAID splits on X, in categories {1} {2} and {3}. When C&R Tree splits X on {1, 2} and {3}, and then splits {1, 2} into {1} and {2} in the next level, you have the same end result as in CHAID. Thus, although the trees differ in structure, they may define the same terminal nodes. As noticed in the *Building Your Tree Interactively with C&R Tree and Quest* module, it is up to the business analyst, balancing all pros and cons, to choose the best model or to decide to combine multiple models into a single one.



The following (synthetic) file coming from a (fictitious) telecommunications firm is used to demonstrate how you build your tree directly using various methods: **telco x modeling data.txt**: Information on approximately 32,000 customers of the firm. The data includes demographics, calling minutes, product features, as well as a churn status. Churn status is stored in a field named churn. The values of the data for the churn can be either Active for current customers, or Churned, for churned customers. The file is located in **C:\Train\0A0U5**. Before you begin the demo, ensure that:

- You have started MODELER (click **Cancel** when the splash screen appears when you start MODELER).
- You have set MODELER's working folder. (In MODELER, select **File\Set Directory**. When you are working in an IBM training environment, browse to the folder **C:\Train\0A0U5** and then click **Set** to select it. When you are not working in an IBM training environment, ask your trainer for the folder name.)

Demo 1: Build Your Tree Directly to Predict Churn

Purpose:
You want to build and compare tree models directly to predict whether or not customers will churn. You will use the best model to score records.

Task 1. Import and instantiate the data.

1. From the **Sources** palette, double-click the **Var. File** node to add it to the stream canvas.
2. Edit the **Var. file** node, and then:
   - in the **Var. File** dialog box, to the right of the **File** box, click **Browse**  (the Browse window should automatically open to the **C:\Train\0A0U5** folder).
   - select **telco x modeling data.txt** and then click **Open**
   - close the **Var. File** dialog box
3. From the **Field Ops** palette, add a **Type** node downstream from the **Var. File** node.
4. Edit the **Type** node, and then:
   - click **Read Values** to instantiate the data
   - ensure that the **Role** for **gender** to **bill_offpeak** is **Input**
   - select **churn** and set its **Role** to **Target**
   - set the role for the other fields to **None**
   - close the **Type** dialog box

Task 2. Compare a default CHAID with an Exhaustive CHAID model.

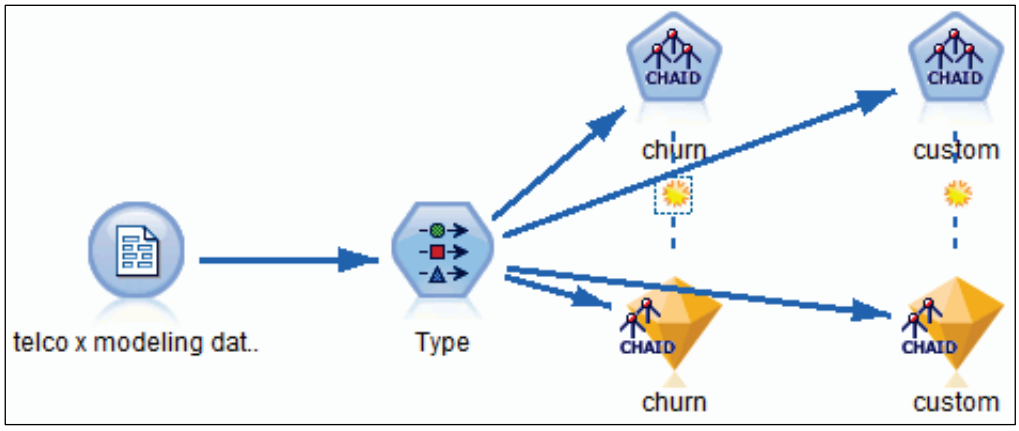
You will compare the accuracy and gain of a model where the regular CHAID method is applied to a model where Exhaustive CHAID is applied.

1. From the **Modeling** palette (**Classification** item), add a **CHAID** node downstream from the **Type** node.

   You will add a second CHAID node, select the Exhaustive CHAID method, and annotate the node to distinguish it from the first CHAID node.

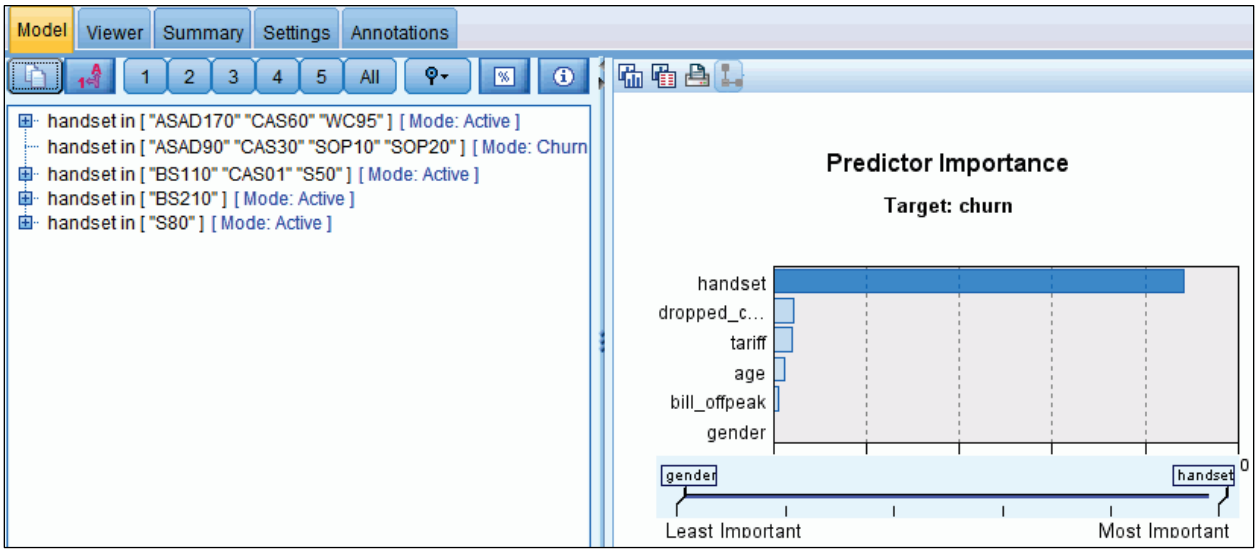
2. Add a second **CHAID** node downstream from the **Type** node.
3. Edit the second **CHAID** node, and then:
   - click the **Build Options** tab
   - click the **Basics** item, and from the **Tree growing algorithm** list, click **Exhaustive CHAID**
   - click the **Annotations** tab, click **Custom**, and in the **Name** box, type **custom**
   - close the **CHAID** dialog box
4. Select both **CHAID** nodes, right-click one of them, and then select **Run Selection**.

   The stream canvas appears as follows:



Two model nuggets are added to the stream canvas and connected to the respective CHAID node.

5. Edit the model nugget named **churn**, and then click the **Model** tab.
   A section of the results appears as follows:



The left pane lists the rules. You can expand all rules to examine the rule set. This will not be pursued in this course. Instead, the results are viewed in the format of a tree.

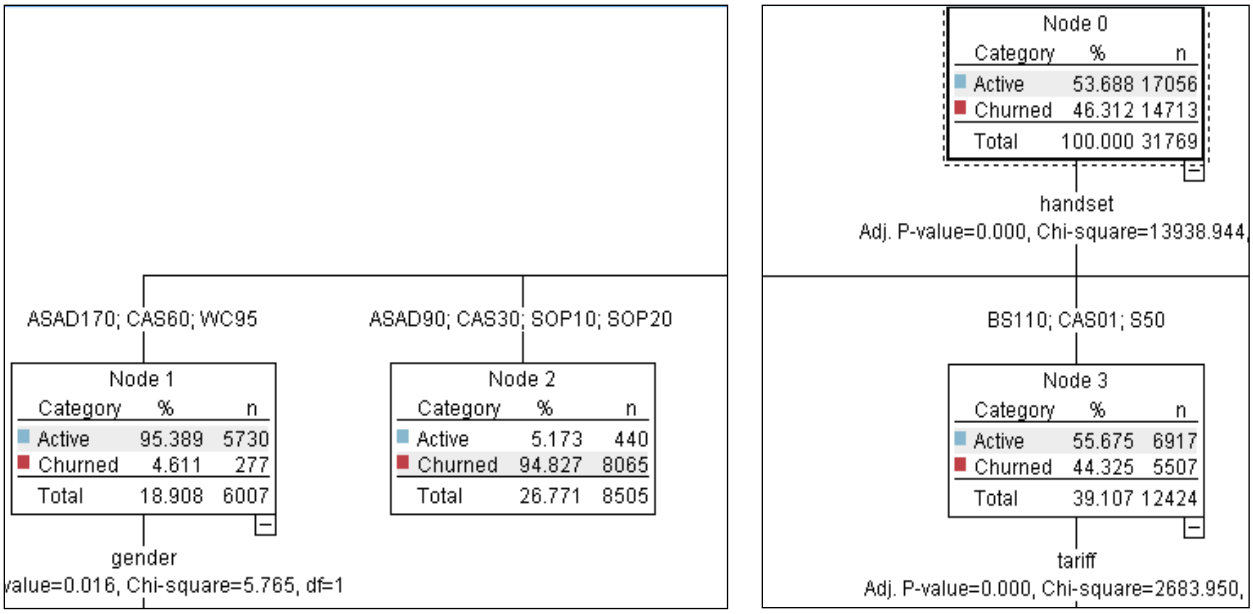
The right pane displays the Predictor Importance graph, which was a default option in the CHAID node. Predictor importance values sum to 1, so the relative importance of each field can be directly compared. Predictor importance does not relate to model accuracy.

You will examine the rules in the tree viewer.

6. Click the **Viewer** tab, and then click the **zoom** button.

   Scroll to the left so that the left branch of the tree from the root is visible.

   The results appear as follows:



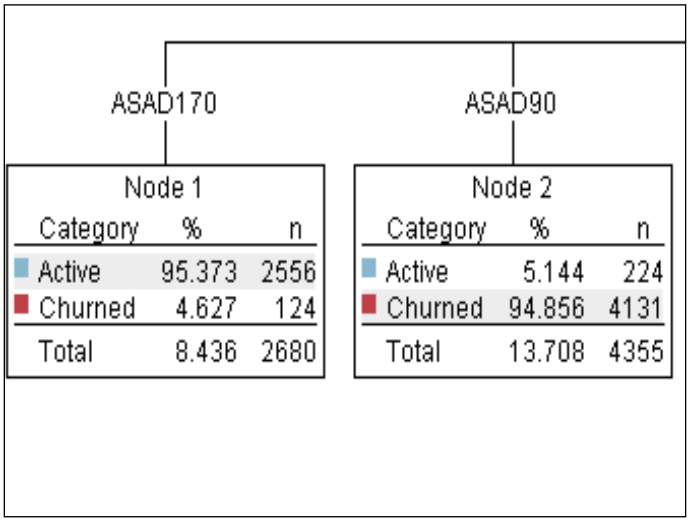
   Handsets, for example ASAD170 and CAS60, have been merged, because there is no significant difference between them in percentage churn.

7. Close the model nugget window named **churn**.

8. Edit the model nugget named **custom**, click the **Viewer** tab and then click the **zoom** button.

9. Scroll to the left so that the left branch of the tree is visible.

   The results appear as follows



   Handsets ASAD170 and ASAD90 are not merged. Apparently, keeping them as separate categories resulted in a higher significant value (lower probability value) for the handset by churn relationship than merging them.

10. Close the **custom** model nugget window.

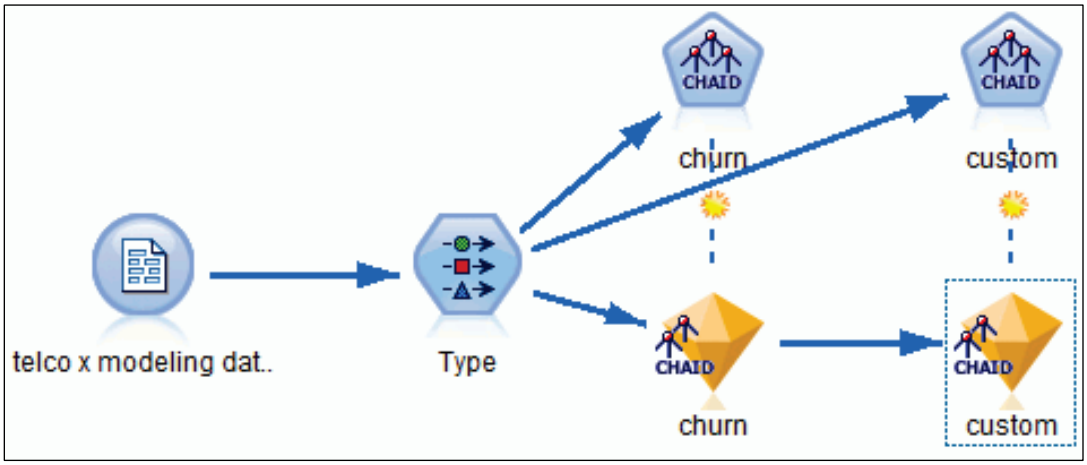
Task 3. Compare the fit of the two CHAID models.

You will compare the accuracy of the two models. Therefore, include both model nuggets in the same branch of the stream.

1. Right-click the **custom** model nugget and click **Disconnect** to disconnect it from the **Type** node.

2. Connect the **custom** model nugget so that it is immediately downstream from the model nugget named **churn**.

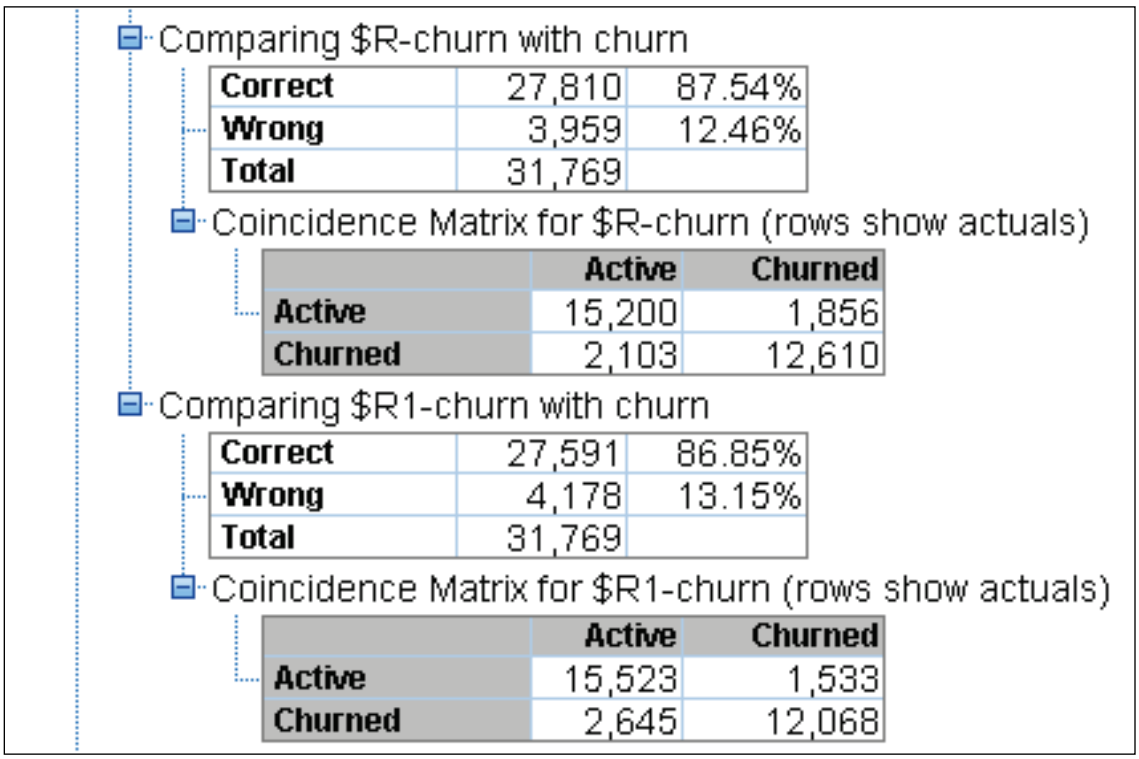
   A section of the results appears as follows:



3. From the **Output** palette, add an **Analysis** node downstream from the **custom** model nugget.
4. Edit the **Analysis** node, enable the **Coincidence matrices (for symbolic targets)** option, and then close the **Analysis** dialog box.
5. From the **Graphs** palette, add an **Evaluation** node downstream from the **custom** model nugget.

6. Right-click the **Analysis** node and then select **Run**.

   The results appear as follows:



Comparing $R-churn with churn

Correct	27,810	87.54%
Wrong	3,959	12.46%
Total	31,769	

Coincidence Matrix for $R-churn (rows show actuals)

	Active	**Churned**
Active	15,200	1,856
Churned	2,103	12,610

Comparing $R1-churn with churn

Correct	27,591	86.85%
Wrong	4,178	13.15%
Total	31,769	

Coincidence Matrix for $R1-churn (rows show actuals)

	Active	**Churned**
Active	15,523	1,533
Churned	2,645	12,068

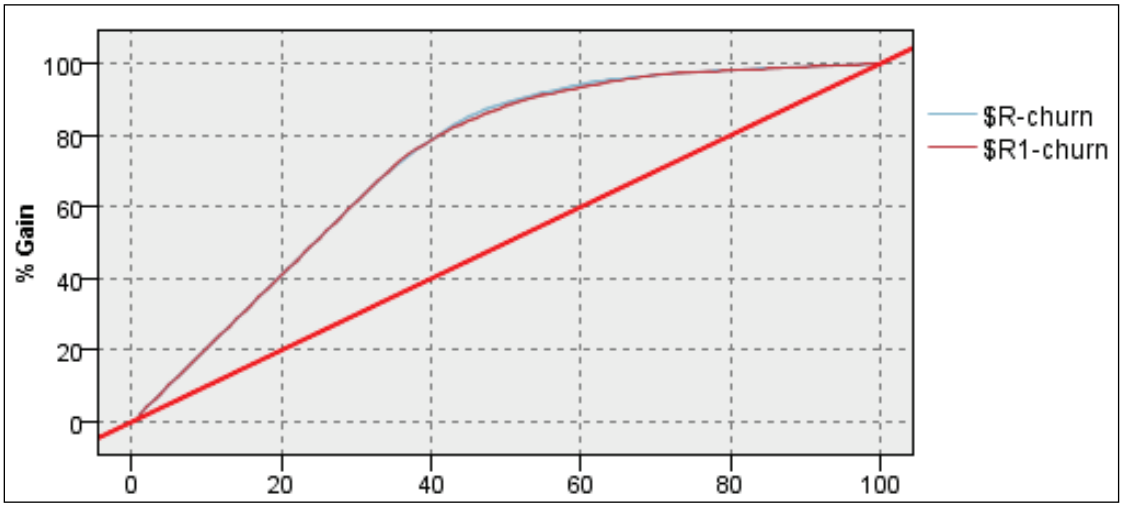
   The Analysis output shows that the accuracy for the default model, $R-Churn in the output, is slightly higher than the accuracy for the Exhaustive CHAID model, $R1 in the output. The default model also identifies more churners (12,610 versus 12,068).

   Note: the fields added by a CHAID model nugget begin with $R and are numbered corresponding to their order in the stream.

7. Close the **Analysis** output window.

8. Right-click the **Evaluation** node and then click **Run**.

   The result appears as follows:



   The gain charts are almost identical.

   All in all, the default model performs slightly better. In the end, it is a business decision to prefer one over the other. It may depend upon whether the business analyst wants to approach separate handset-customer groups.

9. Close the **Evaluation** output window.

   Leave the stream open for the next task.

Task 4. Compare two C&R Tree models.

You will compare the accuracy and gain of a model where the default C&R Tree method is applied to a C&R Tree model where misclassification costs are incorporated.

By default, C&R Tree uses an overfit prevention set. In this task, you will use all records for model building.

1. From the **Modeling** palette (**Classification** item), add a **C&R Tree** node downstream from the **Type** node, edit it, and then:

   - click the **Build Options** tab
   - click the **Advanced** item, and set the **Overfit prevention set (%)** to **0**
   - close the **C&R Tree** dialog box

2. Add a second **C&R Tree** node downstream from the **Type** node.

3. Edit the second **C&R Tree** node, and then:
   - click the **Build Options** tab
   - click the **Costs & Priors** item
   - enable the **Use misclassification costs** option
   - double-click the cell that corresponds to **Actual Churned**, **Predicted Active** to give it focus and allow editing, and type **2**
   - in the **Select an item** pane, click **Advanced**, and set the **Overfit prevention set (%)** to **0**
   - click the **Annotations** tab and name the node **custom**
   - close the **C&R Tree** dialog box
4. Select both **C&R Tree** nodes, right-click one of them, and then select **Run Selection**.
5. Edit the C&R Tree model nugget named **churn**, click the **Viewer** tab, and zoom in to the first split field, handset.

   The split is binary and the impurity is reduced by 0.172.
6. Close the model nugget window named **churn**.
7. Edit the C&R Tree model nugget named **custom**, select the **Viewer** tab, and examine the first layer of the tree.

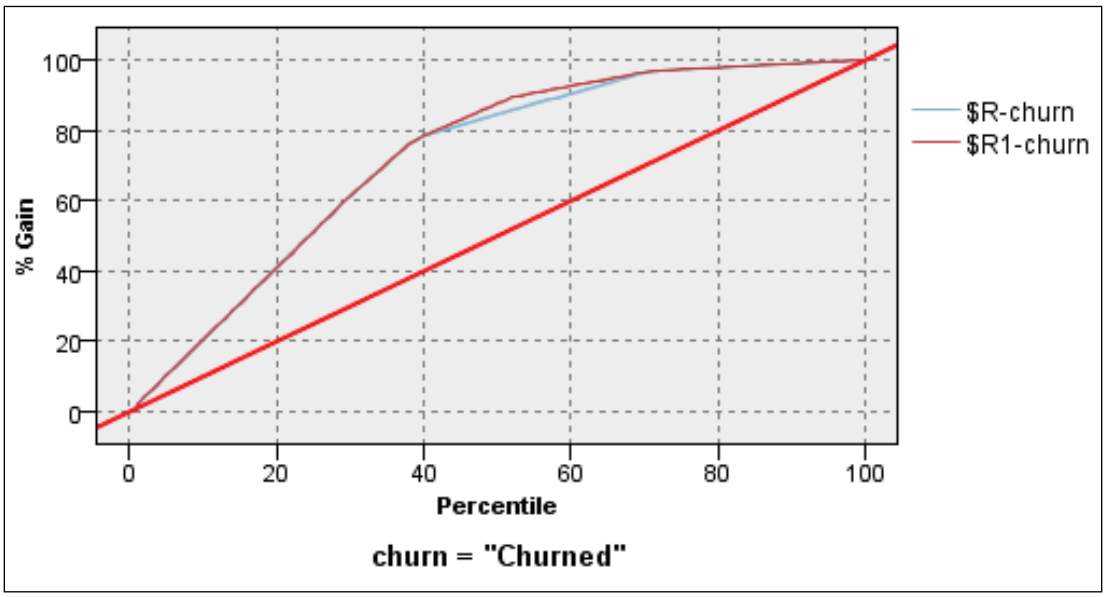
   The best split field is handset again, with the same assignment to the nodes as before. The predicted category for the root node has changed: given that misclassifying a churner as active costs twice as much as vice versa, for this distribution predicting category Churned produces the lowest overall misclassification costs.

   The improvement for the first split field has changed from 0.172 to 0.258, which demonstrates that incorporating misclassification costs in C&R Tree will result in a different tree.
8. Close the model nugget window named **custom**.
9. Disconnect the C&R Tree model nugget named **custom** from the **Type** node and connect it downstream from the C&R Tree model nugget named **churn**.

10. From the **Output** palette, add an **Analysis** node downstream from the C&R Tree model nugget named **custom**.
11. Edit the **Analysis** node, enable the **Coincidence matrices (for symbolic targets)** option, and then close the **Analysis** dialog box.
12. From the **Graphs** palette, add an **Evaluation** node downstream from the C&R Tree model nugget named **custom**.
13. Right-click the **Analysis** node and click **Run**.

    The Analysis output shows that the accuracy for the default model, $R-Churn, is higher than the accuracy for the model that incorporates misclassification costs. However, the latter model performs much better in identifying the churners, 13,172 compared to 11,488 in the default model. This shows the effect of putting a heavier penalty on misclassifying the actual churners.
14. Close the **Analysis** output window.
15. Right-click the **Evaluation** node and click **Run**.

    The result appears as follows:



The gain for the two models is almost identical, with the costs model performing slightly better in the 40%-60% range.

Examining the accuracy, identification of churners, and gain, the costs model is preferred. Of course, the acceptance of this model relies on how reasonable it is to put a penalty of 2 on misclassifying churners as active customers.

Note: the fields added by a C&R Tree model nugget begin with $R and are numbered corresponding to the order in the stream.

16. Close the **Evaluation** output window.

Leave the stream open for the next task.

Task 5. Compare two C5.0 models.

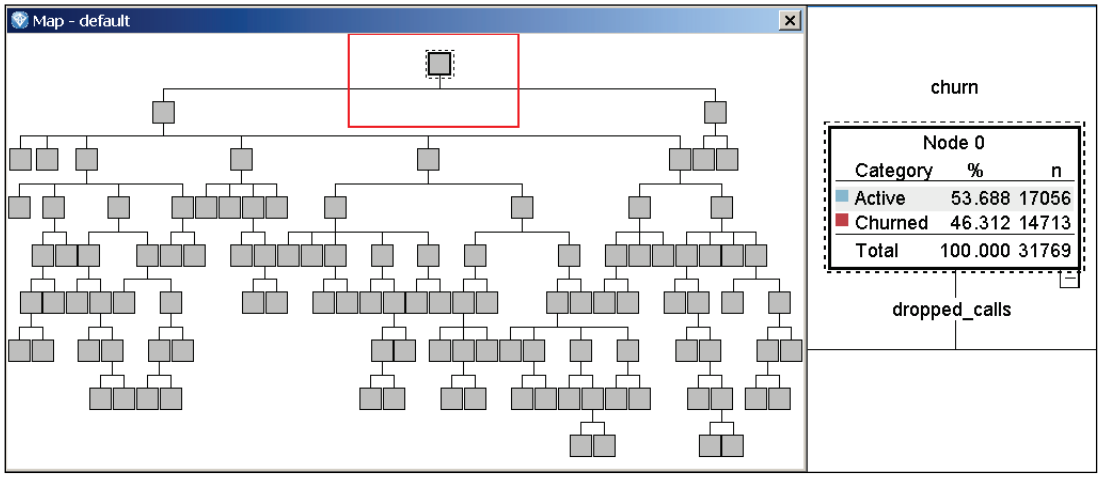
You will compare the accuracy and gain of a model where the default C5.0 method is applied to a C5.0 model where the tree is pruned more heavily.

1. From the **Modeling** palette (**Classification** item), add a **C5.0** node downstream from the **Type** node.
2. Add a second **C5.0** node downstream from the **Type** node.
3. Edit the second **C5.0** node, and then:
   - click the **Model** tab and for **Mode** select the **Expert** option
   - set **Pruning severity** to **85** and **Minimum records per child branch** to **50**
   - click the **Annotations** tab, select **Custom** and name the node **custom**
   - close the **C5.0** dialog box
4. Select both **C5.0** nodes, right-click one of them, and then select **Run Selection**
5. Edit the C5.0 model nugget named **churn**, and then click the **Viewer** tab.

   The Predictor Importance graph shows that handset is by far the most important predictor, followed by dropped_calls.
6. Click the **Viewer** tab, zoom in on **Node 0**, and then click the **Show or hide the tree map window**  button.

7. Arrange the tree map window and so that you can see the root node and the tree map at the same time.

   The result appears as follows:



   Examine the tree structure closely to get an idea of the default tree.

8. Close the model nugget window named **churn**, and then edit the model nugget named **custom**.
9. Click the **Viewer** tab, zoom in on **Node 0**, and then click the **Show or hide the tree map window** button.

   The tree map displays a tree that is less bushy than the default tree.

10. Close the **C5.0** model nugget window named **custom**.
11. Disconnect the **C5.0 model** nugget named **custom** from the **Type** node and connect it downstream from the C5.0 model nugget named **churn**.
12. Add an **Analysis** node downstream from the C5.0 model nugget named **custom**, edit it, enable the **Coincidence matrices (for symbolic targets)** option, and then close the **Analysis** dialog box.
13. Add an **Evaluation** node downstream from the C5.0 model nugget named **custom**.

14. Right-click the **Analysis** node and click **Run**.

    The Analysis output shows that the custom model with higher pruning severity is only 0.6% less accurate than the default model (90.19% versus 90.8%). Also, the number of churners identified by each of the models does not differ much.

15. Close the **Analysis** output window.
16. Right-click the **Evaluation** node and click **Run**.

    The chart shows that the gain for both models is almost identical. Most likely, you will prefer the most parsimonious model where heavier pruning is applied to the default model.

17. Close the **Evaluation** output window.

Task 6. Scoring records using the C5.0 custom model.

You want to add a field to your dataset that stores the propensities. C5.0 achieved the highest accuracy among the rule induction models discussed so far: where CHAID and C&R Tree reported an accuracy of about 87%, C5.0 is about 90% accurate. In general, C5.0 often gives good if not the best results.

Thus, to score records, you will use the C5.0 custom model. This model was the most C5.0 parsimonuous model, and had approximately the same accuracy as the C5.0 default model.

1. Edit the C5.0 model nugget named **custom**, and then click the **Settings** tab.
2. Enable the **Calculate raw propensity scores** option, and then click **Preview**.

3. Scroll all the way to the right of the Preview window.

   A section of the results appear as follows:

$CC-churn	$C1-churn	$CC1-churn	$CRP1-churn
0.948	Churned	0.948	0.948
0.948	Churned	0.948	0.948
0.948	Churned	0.948	0.948
0.948	Churned	0.948	0.948
0.948	Churned	0.948	0.948
0.948	Churned	0.948	0.948
0.948	Churned	0.948	0.948
0.948	Churned	0.948	0.948
0.948	Churned	0.948	0.948

   The **$CRP1-churn** field stores the propensities.

4. Close the **Preview** output window, and then close the model nugget named **custom**.
5. Close the stream without saving anything.

Results:
You have run CHAID, C&R Tree and C5.0 models, compared their accuracies and used the best model to score records.

Note: You will find the solution results in the file **demo_building_your_tree_directly_completed.str**, located in the **04-Building_Your_Tree_Directly\Solutions** sub folder.

Apply Your Knowledge

Use the questions in this section to test your knowledge of the course material.

Question 1: True or false: The predictor importance graph is related to the model's accuracy.

A. True

B. False

Question 2: Which of the following statements is the correct one?

A. Exhaustive CHAID will always grow a binary tree.

B. The accuracy of a tree grown with Exhaustive CHAID is always higher than the accuracy of a tree grown with regular CHAID.

C. Exhaustive CHAID takes less time to compute than regular CHAID.

D. Exhaustive CHAID provides a modification of CHAID that does a more thorough job of examining possible splits for each predictor.

Question 3: Select all that apply: For which methods do incorporating classification costs affect tree growth?

A. CHAID

B. C&R Tree

C. Quest

D. C5.0

Question 4: Select all that apply: Which methods grow a binary tree?

A. CHAID

B. C&R Tree

C. Quest

D. C5.0

Question 5: Select all that apply: Which methods support pruning?

A. CHAID

B. C&R Tree

C. Quest

D. C5.0

Question 6: Suppose that CHAID is run on a dataset of millions of records and a huge tree is produced. Now suppose that it is required that the tree is more parsimonious. What can you do to grow a less bushy tree?

A. Set the Significance level for split is to a higher value.

B. Set the Significance level for merge to a higher value.

C. Set the Maximum Tree Depth to a higher value.

D. Set the Minimum records in the parent branch to a higher value.

Question 7: What can you do to grow a less bushy tree when you use C&R Tree?

A. Set the Minimum change in impurity to a higher value.

B. Set the Maximum difference in risk (in standard errors) to a higher value.

C. Set the Maximum Tree Depth to a higher value.

D. Set the Minimum records in the parent branch to a higher value.

Question 8: What can you do to grow a less bushy tree when you use C5.0?

A. Set the Pruning severity to a higher level.

B. Set the Minimum records per child branch to a higher value.

C. Enable the Winnow attributes option.

Question 9: Select all that apply with respect to pruning a tree.

A. A pruned tree is less bushy than a tree to which no pruning is applied.

B. A tree does not replicate well when no pruning is applied.

C. Pruning may result in a tree with only the root node.

D. The methods that support pruning all use the Chi-square statistic for pruning.

Answers to questions:

Answer 1: B. False. The predictor importance graph is not related to the model's accuracy.

Answer 2: D. Exhaustive CHAID will not necessarily grow a binary tree. Also, there is no guarantee that the accuracy of a model that is built with Exhaustive CHAID is greater than the accuracy of a model that was built with the regular CHAID algorithm. Exhaustive CHAID takes more computer time, because it does a more thorough job of examining possible splits.

Answer 3: B, C, D. Incorporating classification costs affect tree growth for all algorithms, except for CHAID.

Answer 4: B, C. C&R Tree and Quest grow a binary tree.

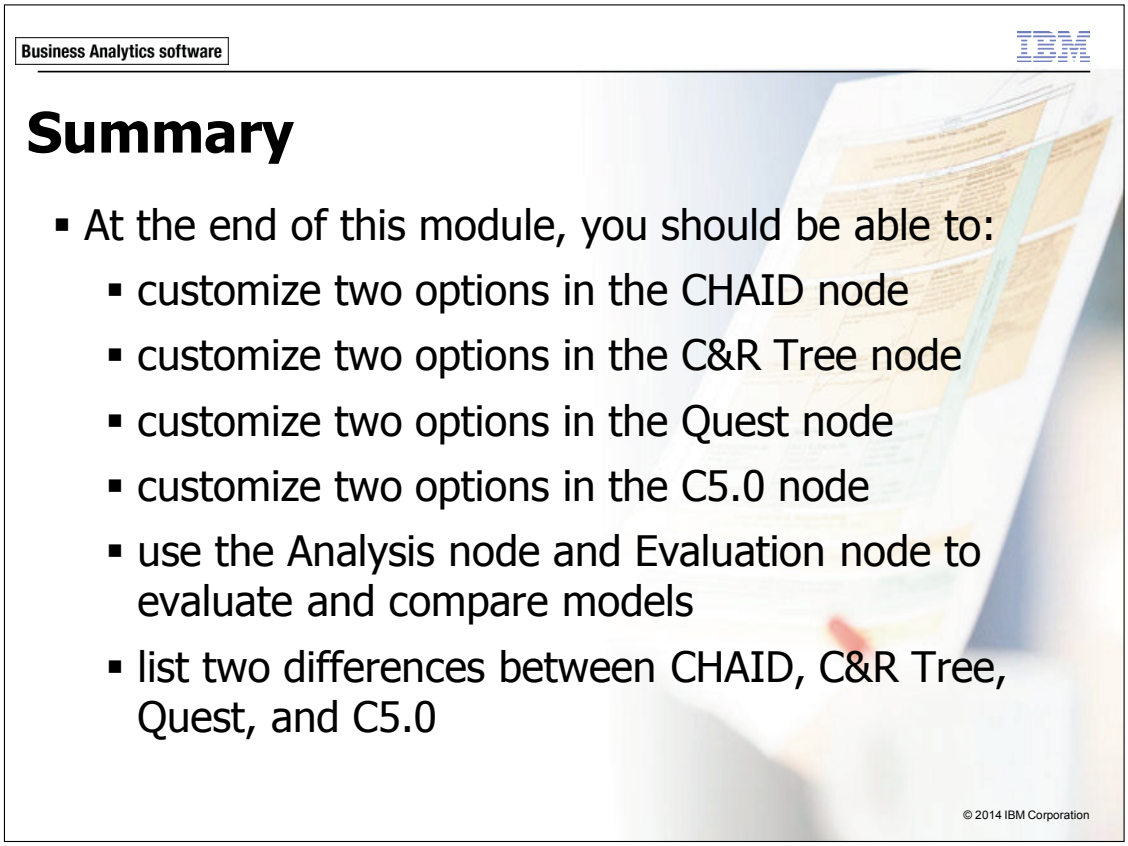
Answer 5: B, C, D. CHAID does not implement pruning. C&R Tree, Quest and C5.0 do.

Answer 6: D. Setting the significance level for split to a higher value results in more splits, so will not result in a less bushy tree. Setting the significance level for merge to a higher value means that fewer categories will be merged, so it will not result in a less bushy tree. Setting the tree depth to a higher value will result in even more levels, so certainly will not make a tree less bushy. Increasing the minimum number of records in parent nodes does result in a less bushy tree.

Answer 7: A, B, D. All settings will result in a less bushy tree, except incrementing tree depth.

Answer 8: A, B, C. All settings apply when you want the tree to be less bushy.

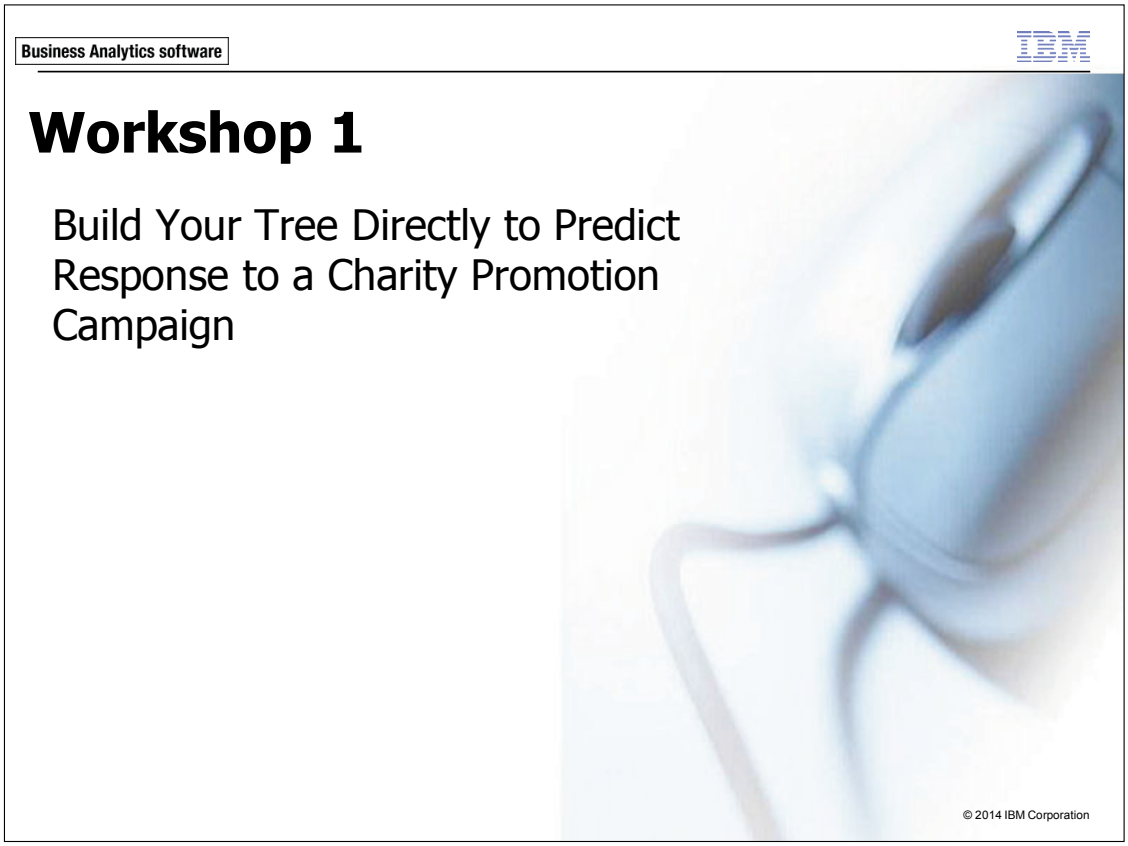
Answer 9: A, B, C. A pruned tree is less bushy. Also, pruning may result in a tree that has only a root node. In addition, a pruned tree replicates better. The Chi-square statistic is not used for pruning, but the risk estimate in combination with the standard error is used.



This module presented four modeling techniques, which directly produce a model (specifically, a tree).

All techniques build the tree step-by-step, which means that they all optimize predictor selection at each level of the tree, and not for the whole tree built. Thus, they all build trees that do not result in the highest accuracy *perse*.

If you want to compare multiple models, make sure that you partition the data, and evaluate the models on a testing dataset (a dataset that was not used for building the models).



The following (synthetic) file is used in this workshop:

- **charity.txt**: A text file that represents data from a charity organization. It contains information on individuals who were mailed a promotion. The information includes whether the individuals responded to the campaign, their spending behavior with the charity and basic demographics such as age, gender, and demographic group. The file is located in **C:\Train\0A0U5**.

Before you begin the workshop, ensure that:

- You have started MODELER (click **Cancel** when the splash screen appears when you start MODELER).
- You have set MODELER's working folder. (In MODELER, select **File\Set Directory**. When you are working in an IBM training environment, browse to the folder **C:\Train\0A0U5** and then click **Set** to select it. When you are not working in an IBM training environment, ask your trainer for the folder name.)

Workshop 1: Build Your Tree Directly to Predict Response to a Charity Promotion Campaign

In this workshop you will predict promotion campaign mail recipient response based on certain predictor fields using direct mode. You will use CHAID, C&RTree, Quest, and C5.0 and compare the models. The charity dataset will be the data source for this direct mode evaluation, but you will partition the dataset into a training and a testing dataset. Designate the response to campaign field as the target.

To do this, you must:

- Use a **Var. File** node to import data from **charity.txt**. Then, run a **Data Audit** to examine the data.

  What is the percentage of customers responding positively to the campaign?

- Partition the data into a Training partition size of 70%, and a Testing partition size of 30%. Set the seed value to **1234567**, so you can replicate results.
- Add a **Type** node downstream from the **Partition** node. Configure the **Type** node so that **response to campaign** will be predicted by **gender**, **age**, **mosaic bands**, **pre-campaign expenditure**, and **pre-campaign visits**.
- Add **CHAID**, **C&R Tree**, **Quest**, and **C5.0** nodes downstream from the **Type** node. To differentiate the nodes in analysis, annotate each node with its model name. Make sure that all records are used for model building (set the value for the Overfit prevention set to 0 for C&R Tree and Quest).
- Run all four nodes and compare the models with an **Analysis** node and an **Evaluation** node.
- Pick the best model based on the gains chart on the testing set, and evaluate boosting and bagging with respect to that model.

Workshop 1: Tasks and Results

Task 1. Import and examine the data.

1. From the **Sources** palette, double-click the **Var. File** node to add it to the stream canvas.
2. Edit the **Var. File** node, and then:
   - to the right of the **File** box, click **Browse**  (the Browse window should automatically open to the **C:\Train\0A0U5 folder**)
   - select **charity.txt** and then click **Open**
   - close the **Var. File** dialog box
3. From the **Output** palette, add a **Data Audit** node downstream from the **Var. File** node, run the **Data Audit** node, and double-click the **Sample Graph** for the **response to campaign** field.

   The Data Audit output shows that 31.32% responded to the campaign.
4. Close the **Distribution** output window, and then close the **Data Audit** output window.

Task 2. Partition the data into a training set and testing set.

1. From the **Field Ops** palette, add a **Partition** node downstream from the **Var. File** node, and set the **Training partition size** to **70** and the **Testing partition size** to **30**.
2. Ensure that the **Repeatable partition assignment** option is enabled, with seed value **1234567**.
3. Close the **Partition** dialog box.

Task 3. Instantiate the data and set the roles for the fields.

1. From the **Field Ops** palette, add a **Type** node downstream from the **Partition** node.
2. Edit the **Type** node, and then:
   - click the **Read Values** 

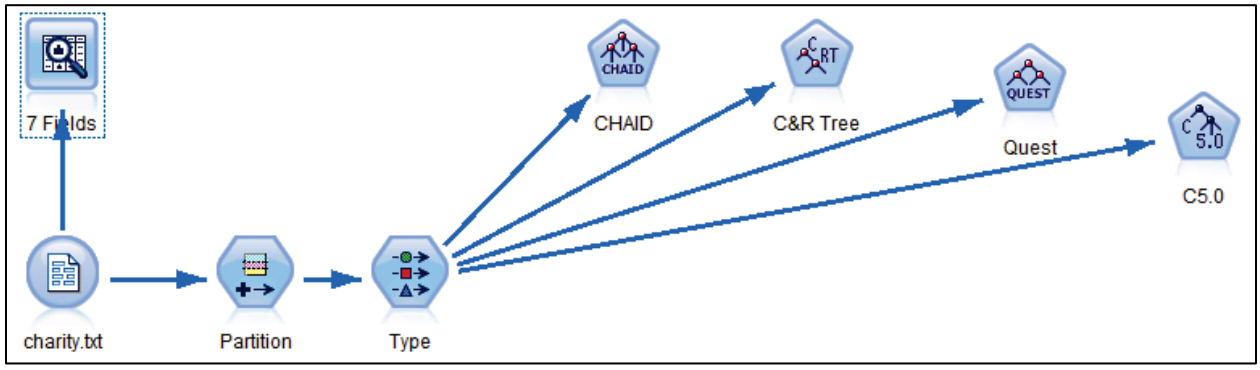
   button

   The Values column is populated with values from the data.
   - set the **Role** for **gender**, **age**, **mosaic bands**, **pre-campaign expenditure**, and **pre-campaign visits** to **Input**
   - set the **Role** for **response to campaign** to **Target**
   - set the **Role** for the other fields to **None**; except **Partition** which should be **Partition**
   - close the **Type** dialog box

Task 4. Add the modeling nodes to the stream.

1. From the **Modeling** palette (**Classification** item), add the **CHAID**, **C&R Tree**, **Quest**, and **C5.0** nodes downstream from the **Type** node, in that order.
2. One by one, edit the nodes and annotate the node with the model name.
3. One by one, edit the **C&R Tree** and **Quest** nodes, on the **Build Options** tab, **Advanced** item, set the **Overfit prevention set (%)** to **0.0**.

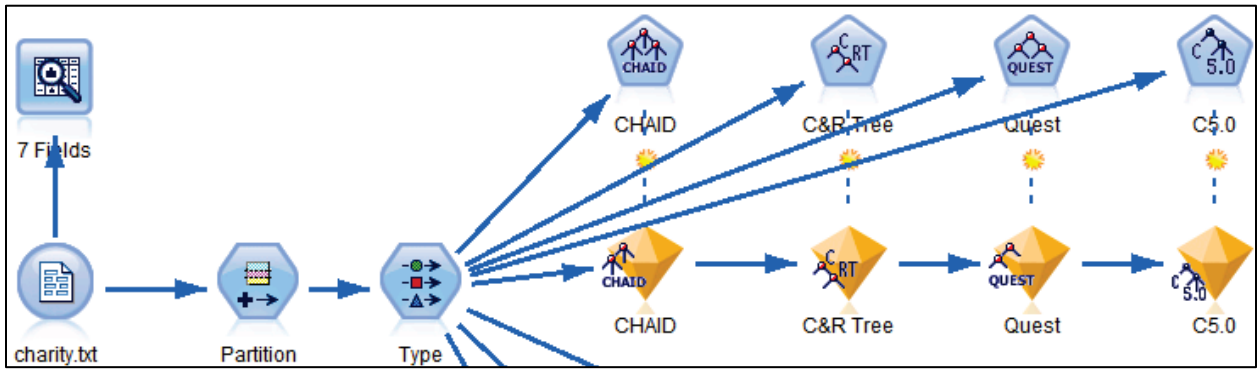
   The overall stream structure appears as follows:



Task 5. Create the analysis branch, and evaluate the models.

1. Run each of the modeling nodes and include all model nuggets in one branch.

   The overall stream structure appears as follows:



2. From the **Output** palette, add an **Analysis** node downstream from the **C5.0** model nugget.

3. Edit the **Analysis** node, enable the **Coincidence matrices (for symbolic targets)** option, and then run the **Analysis** node.

   A small section of the results appears as follows:

Comparing $R2-response to campaign with response to campaign

'Partition'	1_Training		2_Testing	
Correct	1,242	76.01%	564	73.82%
Wrong	392	23.99%	200	26.18%
Total	1,634		764	

Coincidence Matrix for $R2-response to campaign (rows show actuals)

'Partition' = 1_Training	No	Yes
No	1,028	105
Yes	287	214

'Partition' = 2_Testing	No	Yes
No	467	47
Yes	153	97

Comparing $C-response to campaign with response to campaign

'Partition'	1_Training		2_Testing	
Correct	1,304	79.8%	568	74.35%
Wrong	330	20.2%	196	25.65%
Total	1,634		764	

Coincidence Matrix for $C-response to campaign (rows show actuals)

'Partition' = 1_Training	No	Yes
No	1,079	54
Yes	276	225

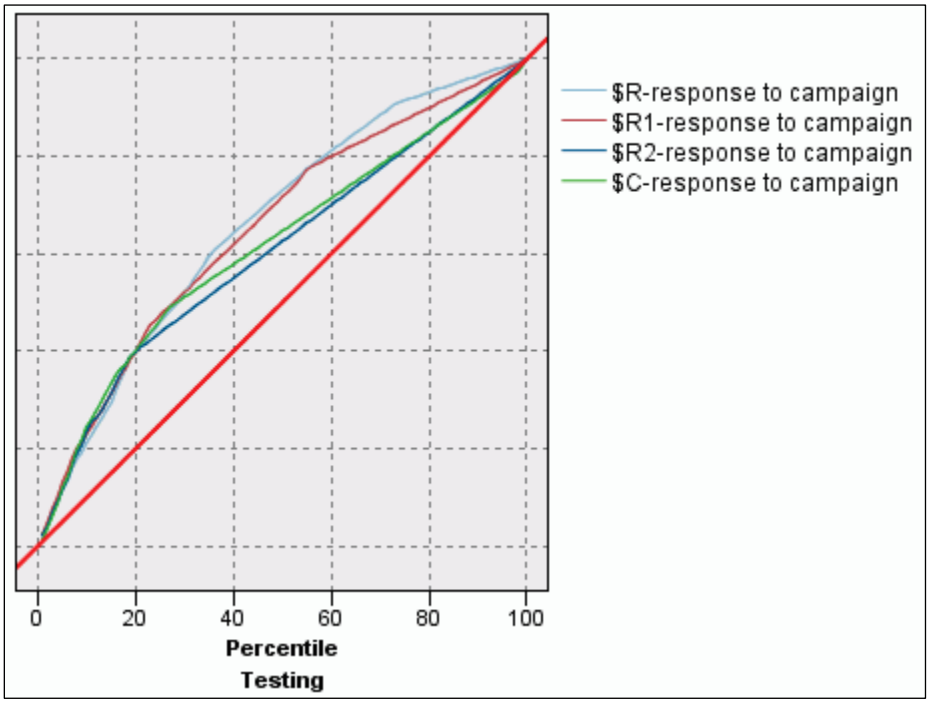
'Partition' = 2_Testing	No	Yes
No	480	34
Yes	162	88

   The Analysis output shows that the models do not differ much on the test dataset. C5.0 has the highest accuracy on the test data, but the drawback is that the number of hits identified by the model is smaller than that for the other models.

4. Close the **Analysis** output window.

5. From the **Graphs** palette, add an **Evaluation** node downstream from the **C5.0** model nugget, and then run the **Evaluation** node.

   A section of the results appear as follows:



   The gains chart show that the models do not differ in the first 20% of the testing dataset. Then the models diverge, with CHAID being the best. Recall that all the other models prune the tree by default.

   You will check out the boosting and bagging options using the CHAID model.

6. Close the **Evaluation** output window.

Task 6. Exploring boosting and bagging with CHAID.

1. Add two more **CHAID** nodes downstream from the **Type** node.
2. Edit one of the **CHAID** nodes, on the **Build Options** tab, **Objective** item, click **Enhance model accuracy (boosting)**, and then annotate the node with **CHAIDBoost**.
3. Click the **Ensembles** item to set the **Number of component models for boosting or bagging** to **15**.
4. Close the **CHAID** dialog box.

5. Edit the other **CHAID** node, on the **Build Options** tab, **Objective** item, select **Enhance model stability (bagging)**, and then annotate it with **CHAIDBag**. Click the **Ensembles** item to set the **Number of component models for boosting or bagging** to **15**.
6. Run both nodes and put the model nuggets in a single branch, **CHAIDBoost** first, followed by **CHAIDBag**.
7. From the **Output** palette, add an **Analysis** node, edit the node, enable the **Coincidence matrices (for symbolic targets)** option, and then run the **Analysis** node.

   A section of the results appear as follows:

Results for output field response to campaign

Individual Models

Comparing $R-response to campaign with response to campaign

'Partition'	1_Training		2_Testing	
Correct	1,288	78.82%	552	72.25%
Wrong	346	21.18%	212	27.75%
Total	1,634		764	

Coincidence Matrix for $R-response to campaign (rows show actuals)

'Partition' = 1_Training	No	Yes
No	1,034	99
Yes	247	254

'Partition' = 2_Testing	No	Yes
No	450	64
Yes	148	102

Comparing $R1-response to campaign with response to campaign

'Partition'	1_Training		2_Testing	
Correct	1,278	78.21%	568	74.35%
Wrong	356	21.79%	196	25.65%
Total	1,634		764	

Boosting gives a higher accuracy on the training dataset than bagging, which could be expected since boosting puts extra weight on incorrectly predicted records. However, bagging has the highest percentage correct on the testing dataset. This also could be expected, because bagging focuses on generalizability to another dataset, not accuracy on the training dataset.

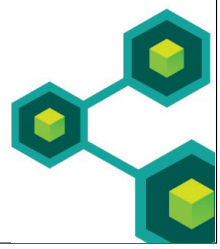
All in all, however, the results are not better than using the regular CHAID model, so that is still the preferred model.

8. Close the **Analysis** output window.
9. Exit MODELER without saving anything.

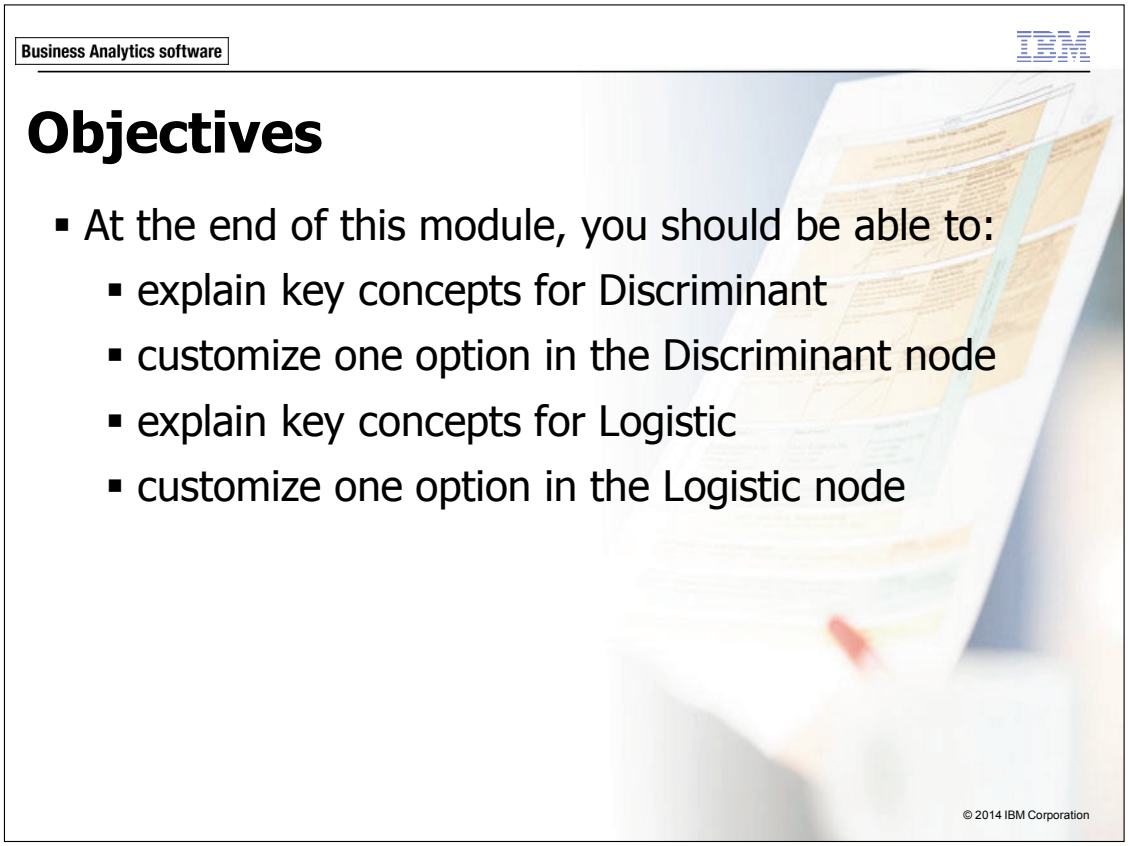
Note: The stream **workshop_building_your_tree_directly _completed.str**, located in the **04-Building_Your_Tree_Directly\Solutions** sub folder, provides a solution to the workshop tasks.

Using Traditional Statistical Models

IBM SPSS Modeler (v16)



Business Analytics software



Before reviewing this module, you should be familiar with the following topics:

- working with MODELER (streams, nodes, palettes)
- importing data (Var. File node)
- defining measurement levels, roles, blanks, and instantiating data (Type node)
- examining the data (Table node, Data Audit node)
- assessing the quality of your model, using accuracy, risk estimate, gain and response measures
- using the model nugget to score data

Business Analytics software

IBM

Examining the Discriminant Model

- Basic idea:
  - derive a new field, say D, that is a combination of the predictors and best separates the target categories
  - $D = b_1 * X_1 + b_2 * X_2 + ... + b_n * X_n$

Discriminant is a technique that is designed to characterize the relationship between a set of continuous predictors on one hand and a categorical target on the other.

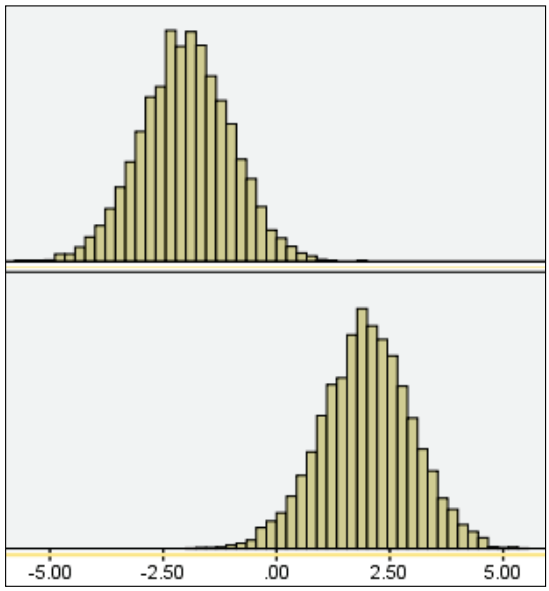
Discriminant computes a linear combination of the predictors that best characterizes the differences between the groups as defined by the target. Discriminant will find coefficients *a*, *b*, etc that will maximally separate, or discriminate, the two categories of the target. The new field that is computed is referred to as the discriminant score, labeled D in the formula.

In Discriminant, there is no interest in the formula itself, because the coefficients are only indirectly related to the target field (via the discriminant score field). Thus, the coefficients and discriminant scores themselves are taken for granted, and the focus is on how well the discriminant scores discriminate the two groups.

Business Analytics software

IBM

Discriminant Illustrated



Distribution for D, for group LOAN PAID BACK = no

Distribution for D, for group LOAN PAID BACK = yes

This slide illustrates Discriminant. A target LOAN PAID BACK is predicted, given AGE, INCOME, and HOUSEHOLD SIZE, all continuous fields.

Behind the scenes, Discriminant has computed coefficients a, b, and c, and D = a * AGE+ b * INCOME and c * HOUSEHOLD SIZE so that the groups LOAN PAID BACK = no and LOAN PAID BACK = yes are maximally separated on D.

Discriminant uses the discriminant score to compute the probability that the record will pay back the loan or not. However, it does not only take the discriminant score into account. It also takes into account what the probability is for a record to belong to either group, regardless of any analysis. For example, suppose that history shows that the probability to pay back a loan equals 0.99, and the analysis here computes a discriminant score of -1 for a record. Considering the plot alone, it is a little bit more likely that this record will not pay back the loan than that he will pay back. However, considering the fact that 99% of all people pay back their loan, the final outcome may be that he has a higher likelihood to pay back the loan than to not pay back the loan.

The next slide presents this idea more formally.

Business Analytics software

IBM

How Discriminant Computes Probabilities and Scores Records

- Use Bayes' Theorem:

  *posterior probability = prior probability * evidence from data*

- Predicted category is the category with the highest posterior probability.
- Discriminant model nugget can add:
  - predicted category
  - probability for predicted category
  - propensities

Discriminant uses Bayes' theorem to compute probabilities and to classify records. Bayes' theorem uses two probability estimates to arrive at the final probability:

- The prior probability: An estimate of the probability that a record belongs to a particular category when no information from the predictors is available. Prior probabilities are typically either determined by the number of records in each category of the target field, or by assuming that the prior probabilities are all equal. For example, if there are two groups, the prior probability of belonging to each group is 0.5.
- The conditional probability: The probability of obtaining a specific discriminant score given that a record belongs to a specific category. By assuming that the predictors are multivariate normal distributed, which implies that the discriminant scores are normally distributed, it is possible to compute this probability.

Using these two probabilities and Bayes' theorem, the so-called posterior probability is calculated, which is the probability for a category, given a specific discriminant score.

Roughly said, this is the prior probability for a category modified by the evidence from the data for that category. It is this probability that is used to assign a record to a category. That is, a record is assigned to the category with the highest posterior probability.

By default, before examining the data, Discriminant assumes a record is equally likely to belong to each category. If it is known that the sample proportions reflect the distribution of the target in the population then Discriminant can be instructed to make use of this information.

For example, if a target category is very rare in the sample and that reflects the situation in the population, Discriminant can make use of this fact in its prediction equation. If you do not account for this, equal prior probabilities for the categories will be assumed, and the prediction for the rare categories will be overestimated (higher probabilities will be assigned than they are in reality).

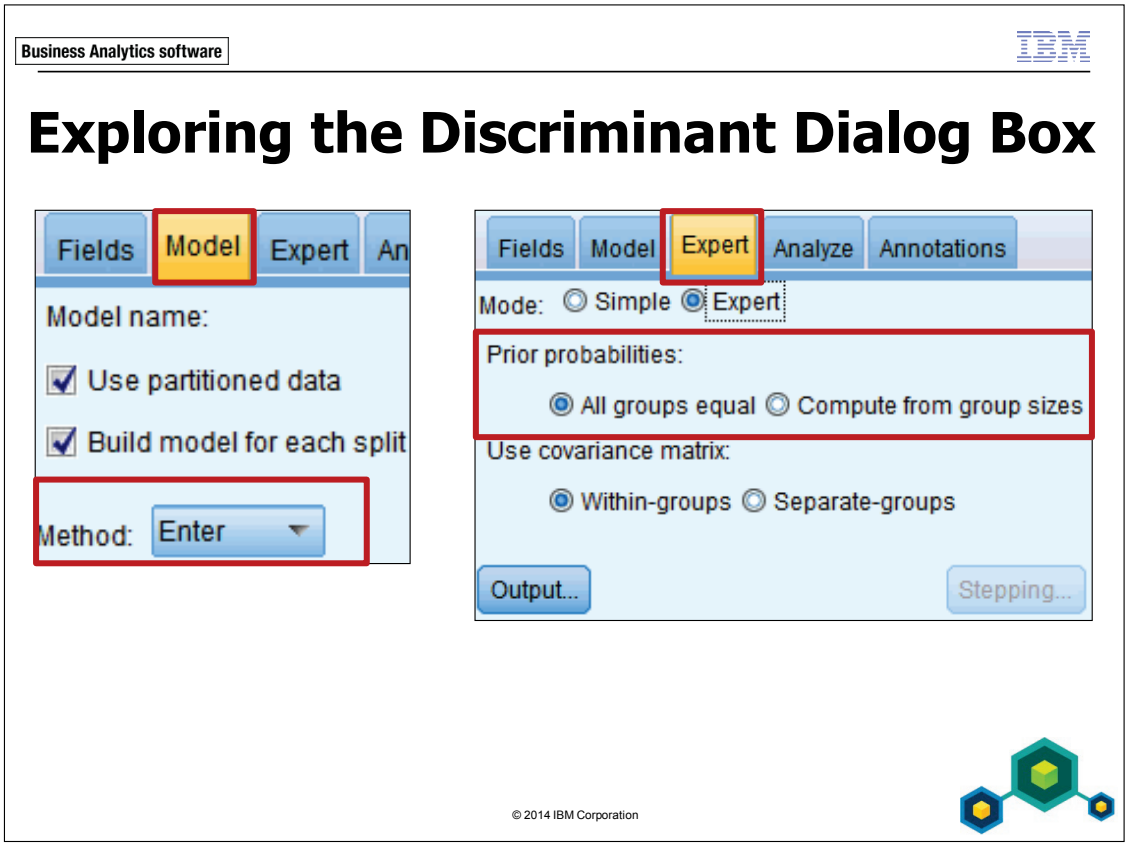
Business Analytics software

IBM

How Discriminant Handles Missing Values

- When the target is missing:
  - discard the record from model building
- When one or more values on predictors are missing:
  - discard the record from model building
- In scoring, records with a missing value on a predictor will not be scored.

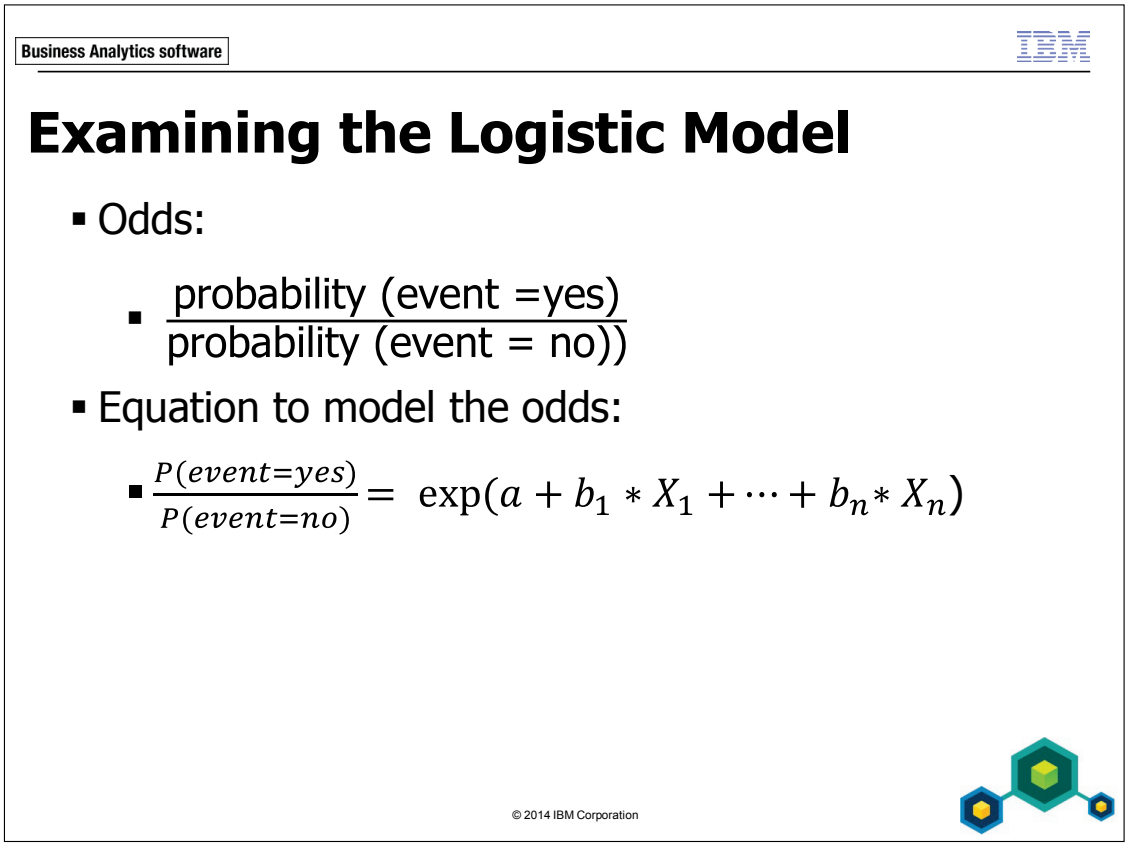
This slide summarizes how Discriminant handles missing values. Contrary to rule induction models such as CHAID, C&R Tree, Quest and C5.0, that all incorporate missing values on predictors one way or the other, Discriminant discards a record when the record has a missing value on any predictor, both for model building and scoring. The reason is simply that the equation cannot be computed when a value is missing on a predictor.



In the Discriminant dialog box, on the Model tab, select the method that determines which predictors will be included in the equation for the Discriminant score:

- Enter: All predictors will be included in the equation.
- Stepwise: Predictors are entered one-by-one in the equation until no significant predictor is available or until all predictors are consumed. The Stepwise method has a tendency to overfit the training data. Thus, when using this method, it is especially important to assess the validity of the model on a testing set.

On the Expert tab, in Expert mode, specify whether you want equal prior probabilities or prior probabilities that match the proportions of the target field's categories. When the target is highly skewed, and reflects the population distribution, it is recommended to compute the priors from the sample, to get realistic probabilities.



The logistic model is expressed in terms of a ratio: the probability that a particular event occurs (a customer churns, a customer accepts an offer, a claim is fraudulent, a customer does not pay back a loan, a student passes an exam, and so forth) versus the probability that the event does not occur. This ratio is known as odds.

For example, when the probability that a customer pays back a loan is 4/5, then the probability that that customer does not pay back the loan= 1- 4/5 = 1/5 (refer to the note below) and the odds are (4/5) / (1/5) = 4 for this customer. When the odds are 1, you know that the probability for the event to occur equals the probability that the event does not occur, and both probabilities are 0.5.

The odds are linked to the predictors by the equation depicted on this slide. In this formula, exp (…) is another way to write $e^{(\ldots)}$, where e is, approximately, the number 2.72. For example: exp (1) equals e^1 ~= 2.72.

Note: For a flag field, the probability that the event does not occur equals 1 - the probability that the event does occur.

Business Analytics software

IBM

Interpreting the Coefficients: One Predictor

- Suppose there is one predictor, X, with values 0 and 1
- Equation to model the odds:
  - $\frac{P(event=yes)}{P\ (event=no)} = \exp(a + b\ * X)$
    - a: intercept (also called constant)
    - b: the odds change by exp (b) when X increments by 1

For an interpretation of the coefficients, consider one predictor only, say X, with values 0 and 1. Replacing the values for X in the formula, you have:

X=0: $Odds\ (\frac{event=yes)}{event=no)}$=exp (a + b * 0) = exp (a)

X=1: $Odds\ (\frac{event=yes)}{event=no)}$=exp (a + b * 1) = exp (a + b)= exp (a) * exp (b)

Thus, the odds change by the factor exp (b), when X increments by 1. Also:

- b < 0 means that $e^b < 1$, so the odds for X=1 are smaller than the odds for X=0, or the probability that the event occurs is smaller for X=1 than for X=0
- b=0 means that $e^b=1$, so the odds for X=1 equal the odds for X=0
- b > 0 means that $e^b > 1$, so the odds for X=1 are greater than the odds for X=0, or the probability that the event occurs is greater for X=1 than for X=0

Thus, a negative, zero, or positive coefficient corresponds to a negative, zero, or positive effect of the predictor, which facilitates the interpretation.

Business Analytics software

IBM

Interpreting the Coefficients: More Predictors

- Equation to model the odds:
  - $\frac{P\,(event=yes)}{P\,(event=no)} = \exp(a + b_1 * X_1 + \ldots + b_n * X_n)$
- The odds change by a factor b_i when X_i increments by 1 (all other predictors fixed at certain values)
- b_i depends on the unit of measurement of X_i

When you have more predictors, the situation is basically the same as when you have one predictor. A coefficient, say b_i, now gives the factor with which the odds change when X_i increments by 1, all other things equal.

Notice that a coefficient depends on the unit of measurement of the predictor. For example, suppose that a predictor X_1 has an effect of b_1 on the odds and that X_1 is divided by 2 to arrive at a predictor X_2. Then when you use X_2 rather than X_1, the coefficient for X_2 will be $b_1$multiplied by 2. Hence, the factor with which the odds change when X_2 increments by 1 equals exp (b_1 * 2).

Business Analytics software

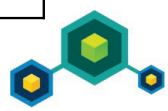
IBM

How Logistics Handles Categorical Predictors

- Create indicator fields for each category, except for the reference category

GENDER	G1
female	1
male	0

MARITAL STATUS	MS1	MS2
divorced	1	0
married	0	1
single	0	0



Flag, nominal, and ordinal predictors will be transformed behind the scenes into a number of so-called indicator fields. An indicator field is 1 for the category that it indicates and 0 for all other categories.

For a flag predictor, this is simply a recoding to the values 0 and 1 (so that you have the situation previously described: a continuous predictor with values 0 and 1). Nominal and ordinal fields will be handled the same way: for each category but the last MODELER will create an indicator field behind the scenes.

This slide shows an example. Suppose that you want to predict LOAN PAID BACK with MARITAL STATUS as predictor.

The logistic equation then is:

$$Odds\left(\frac{event=yes)}{event=no)}\right)=\exp\left(a + b_1 * I_1 + b_2 * I_2\right)$$

To see what this means, substitute the values for the marital status categories in the formula.

Divorced: $I_1=1$, $I_2=0$: $Odds\left(\frac{event=yes)}{event=no)}\right.$=exp (a + b_1 * 1 + b_2*0) = exp (a) * exp (b_1)

Married: $I_1=0$, $I_2=1$: $Odds\left(\frac{event=yes)}{event=no)}\right.$=exp (a + b_1 * 0 + b_2*1) = exp (a) * exp (b_2)

Single: $I_1=0$, $I_2=0$: $Odds\left(\frac{event=yes)}{event=no)}\right.$=exp (a + b_1 * 0 + b_2*1) = exp (a)

This shows that exp (a) gives the odds for the single group. The odds for the divorced group differ from the odds of the single group by a factor of exp (b_1).

The odds for the divorced group differ from the odds of the single group by a factor of exp (b_2). This clarifies that the single group serves as a so-called reference group for the other groups.

It is always with respect to the reference group that the coefficients for the indicator fields must be interpreted. If another category is taken as reference group, then the coefficients for the indicator fields will all change.

Business Analytics software

IBM

Interpreting the Coefficients Illustrated

- Target LOAN PAID BACK, with values YES and NO
- Predictors AGE (in years), GENDER, and MARITAL

PREDICTOR	B	SIGNIFICANCE B	EXP (B)
AGE	0.05	0.001	1.05
GENDER=FEMALE	0.3	0.090	1.35
GENDER=MALE	0	-	-
MARITAL=DIVORCED	2	0.000	7.39
MARITAL=MARRIED	1	0.002	2.72
MARITAL=SINGLE	0	-	-
INTERCEPT	-3	-	0.05

This slide shows an example of a table of coefficients. The interpretation is:

- AGE: When AGE increments by 1 year, the odds change by a factor of 1.05. Or, compare two persons with the first being 10 years older than the second, with identical gender and marital status. Then the odds for the ten year older person are $(1.05)^{10} = 1.65$ of the odds of the first person. For example, suppose a single man of age 20 has a 0.5 probability to pay back the loan, and a 0.5 probability to not pay back the loan, thus the odds of this person are 1. Now consider a 30-year -old single man. His odds are 1 * 1.65 = 1.65 to pay back the loan, which means a probability of 1.65/2.65 = 0.62 to pay back, and a 0.38 probability to not pay back. The coefficient for AGE appears to be significant, (probability value = 0.001).

- GENDER: This field is encoded behind the scenes as FEMALE=1, MALE=0. All other things (age, marital status) being equal, the odds for females are 1.35 * the odds for a man. However, this coefficient is not significant, so GENDER appears not to be relevant for LOAN PAID BACK. MARITAL: Two indicator fields are created behind the scenes with the last category, SINGLE, as the reference category. The exponentiated coefficient for MARITAL = DIVORCED equals 7.39. This means that the odds for a divorced person to pay back the loan are 7.39 the odds for a single person with the same age and gender. In the same manner, the odds for a married person are 2.72 times the odds of a single person with the same age and gender. Both coefficients are significant, which means that divorced and married people significantly differ from singles in their behavior when it comes to paying back a loan. Note: The significance that is reported concerns the hypothesis that the coefficient at hand is 0 in the population. As with the Chi-square test, significance values less than 0.05 lead to rejection of this hypothesis. In other words, the predictor is relevant for the target field. Significance values greater than or equal to 0.05 indicate that the coefficient can be 0 in the population, and thus that the predictor is not a relevant predictor for the target.

Business Analytics software

IBM

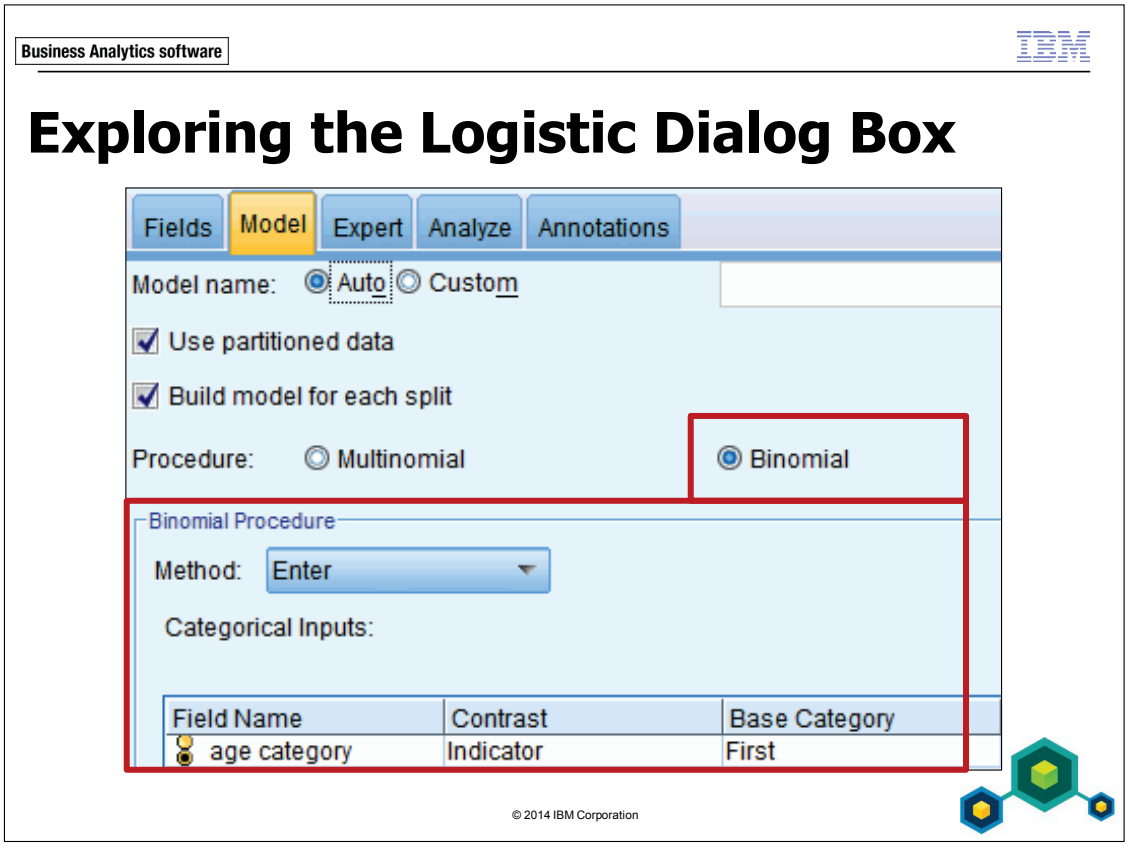
Exploring Logistic

- Missing values:
  - discard the record from model building, when the target or one of the predictors is a missing value
- Scoring:
  - predicted category
  - confidence for the predicted category
  - probability for each category
- Records with a missing on a predictor will not be scored

When the target for a record is a missing value, then the record will be excluded from model building. When a record has a missing value on one or more predictors, then the record will also be ignored in model building. The reason is the same as with Discriminant: the equation cannot be computed when a predictor is missing. This also holds in scoring records.

When you run the Logistic model, a model nugget will be automatically added to your stream. The model nugget will add the following fields:

- The predicted category: The category for which the confidence is higher than 0.5.
- The confidence: The confidence for the predicted category.
- The probability for each of the categories.



Select the Binomial option when you want to predict a flag target and Multinomial when your target has more than 2 categories. By default, all the fields that have role Input will be included in the equation. This is known as method Enter. Logistic offers two other stepwise methods:

- Forward selection: The initial model is the model with the intercept only and fields producing the highest improvement are added one by one, until the best candidate field does not produce a large-enough improvement in the model or until no more fields can be added because all fields are consumed.
- Backward selection: The initial model includes all of the predictors and predictors that contribute little to the model are removed one by one until no more fields can be removed without worsening the model, yielding the final model.

The problem with stepwise methods is that they find the subset of fields that maximize the improvement from step to step, but this is not the same as maximizing predictive accuracy. Notice that you can choose the reference category for categorical predictors.

Business Analytics software

Exploring Logistic Output

Dependent Variable Encoding

Original Value	Internal Value
no	0
yes	1

Categorical Variables Codings(a)

		Frequency	Parameter coding (1)	Parameter coding (2)
MARITAL	divorced	819	1.000	.000
	married	818	.000	1.000
	single	818	.000	.000
GENDER	female	1228	1.000	
	male	1227	.000	

a. This coding results in indicator coefficients.

This slide shows excerpts from advanced output when you have run Logistic.

The table entitled "Dependent Variable Encoding" displays how the original string values for the target are recoded into 0 and 1. The odds that are modeled by Logistic are always of the format $Odds\ (\frac{target=1)}{target=0)}$, so always check that the original values are correctly mapped to 0 and 1.

The table entitled "Categorical Variables Codings" displays how the values of the categorical predictors are recoded into 0 and 1.

Both encoding schemes are crucial for the interpretation of the coefficients, so always let your interpretation of the results be guided by these encoding schemes.

Business Analytics software

IBM

Determining the Traditional Statistical Model to Use

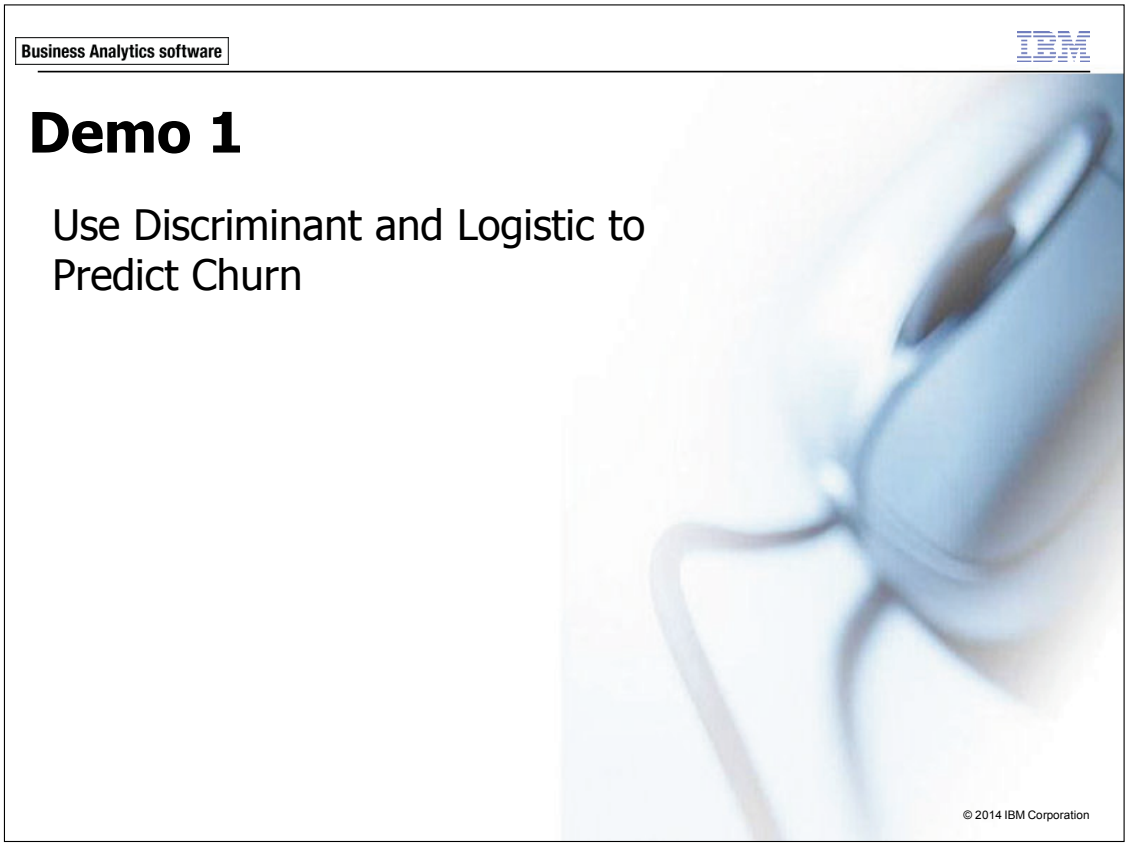
CRITERION	DISCRIMANT	LOGISTIC
is the equation of interest?	no	yes
categorical predictors	not accounted for	automatically accounted for
assumptions	multivatiate normality of predictors	no assumptions
prior probabilities	equal or matching sample proportions	irrelevant

This slide lists the differences between Discriminant and Logistic.

Compared to Discriminant, Logistic has a strong position, because:

- it can handle categorical predictors
- no assumption is made on the distribution of the predictors
- the probabilities are directly derived from the equation, without being dependent on priors (although the possibility to use priors is an advantage when sample proportions do not match population proportions).

Note: the assumption of a multivariate normal distribution of the predictors can be tested in Discriminant, by requesting Box' s M. However, in general this test is of limited use when huge amounts of data are involved (it will always be significant).



The following (synthetic) file coming from a (fictitious) telecommunications firm is used to demonstrate how you build your tree directly using various methods: **churn data.txt**: Information on approximately 18,000 customers of the firm. The data includes demographics, calling minutes, and product features, as well as a churn status. Churn status is stored in a field named CHURN. The values of the data for CHURN can be either NO for current customers, or YES, for churned customers. The file is located in **C:\Train\0A0U5**. Before you begin the demo, ensure that:

- You have started MODELER (click **Cancel** when the splash screen appears when you start MODELER).
- You have set MODELER's working folder. (In MODELER, select **File\Set Directory**. When you are working in an IBM training environment, browse to the folder **C:\Train\0A0U5** and then click **Set** to select it. When you are not working in an IBM training environment, ask your trainer for the folder name.)

Demo 1: Use Discriminant and Logistic to Predict Churn

Purpose:
You will use Discriminant and Logistic to predict whether or not customers will churn.

Task 1. Import and instantiate the data.

1. From the **Sources** palette, double-click the **Var. File** node to add it to the stream canvas.
2. Edit the **Var. file** node, and then:
   - in the **Var. File** dialog box, to the right of the **File** box, click **Browse**  (the Browse window should automatically open to the **C:\Train\0A0U5** folder)
   - select **churn data.txt** and then click **Open**
   - close the **Var. File** dialog box
3. From the **Output** palette, add a **Data Audit** node downstream from the **Var. File** node, and then run the **Data Audit** node.
4. In the **Data Audit** output window, scroll all the way down to **CHURN**, and then double-click the **SAMPLE GRAPH**.

   A section of the results appear as follows:

Value	Proportion	%	Count
No		95.78	17056
Yes		4.22	752

   About 4% has churned.
5. Close the **Distribution** window, and then close the **Data Audit** output window.

Task 2. Instantiate the data and set the roles.

1. From the **Field Ops** palette, add a **Type** node downstream from the **Var. File** node.
2. Edit the **Type** node, and then:
   - click **Read Values** to instantiate the data
   - ensure that the **Role** for **GENDER** to **BILL_OFFPEAK** is **Input**
   - select **CHURN** and set its **Role** to **Target**
   - set the role for the other fields to **None**
   - close the **Type** dialog box

Task 3. Run and compare two Discriminant models.

In this task you will compare the default Discriminant model (with equal prior probabilities) to a model where priors match the sample proportions.

1. From the **Modeling** palette (**Classification** item), add a **Discriminant** node downstream from the **Type** node.
2. Add a second **Discriminant** node downstream from the **Type** node.
3. Edit the second **Discriminant** node, and then:
   - click the **Expert** tab
   - next to **Mode**, click the **Expert** option button, and under **Prior possibilities**, select the **Compute from group sizes** option (by doing this, you are setting the prior probabilities to match sample proportions)
   - click the **Annotations** tab, click **Custom**, and in the **Name** box, type **custom**
   - close the **Discriminant** dialog box

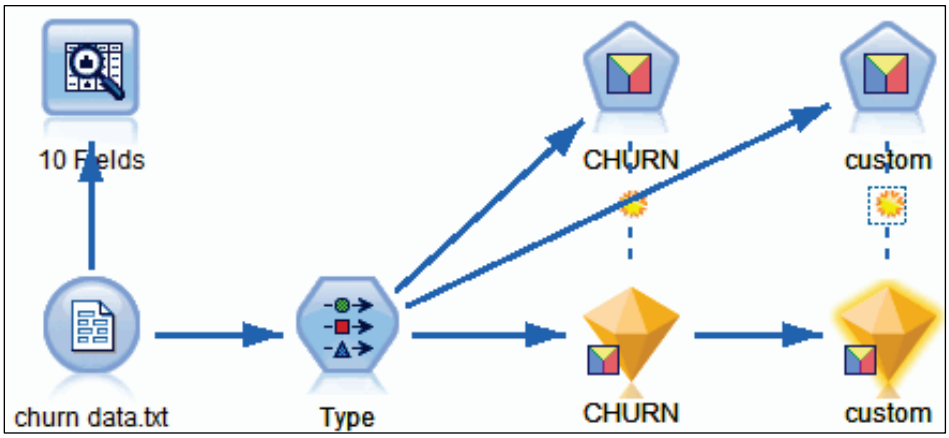
4. Select both **Discriminant** nodes, right-click one of them, and then select **Run Selection**.

   Two model nuggets are created on the stream canvas for sample priors and default. You will not examine the content of the model nuggets, because it is of no interest. The model nugget stores details on how the discriminant scores are computed, but in general these are of no use.

5. Right-click the model nugget named **custom** and click **Disconnect** to disconnect it from the **Type** node.

6. Connect the model nugget named **custom** so that it is immediately downstream from the model nugget named **CHURN**.

   A section of the stream appear as follows:



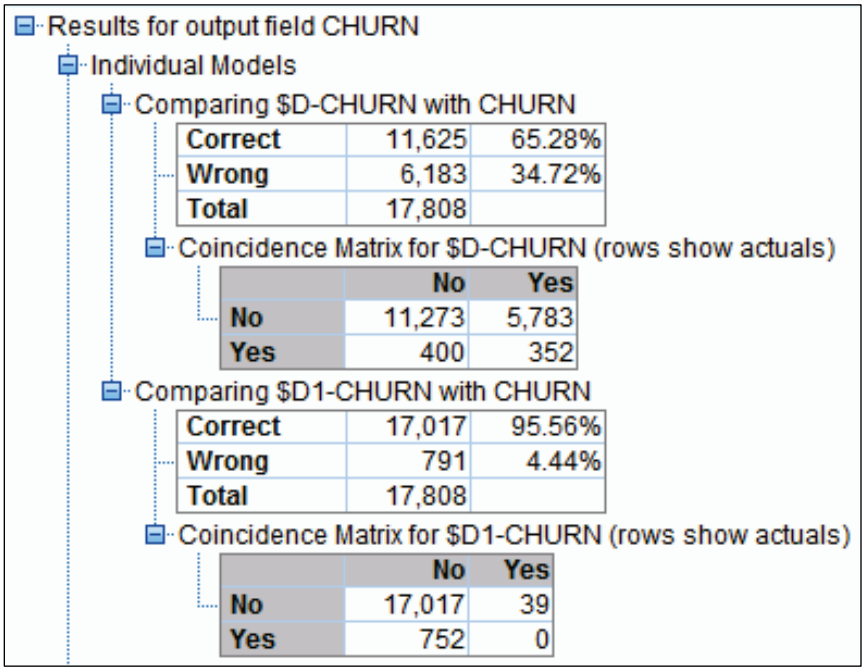
7. From the **Output** palette, add an **Analysis** node downstream from the model nugget named **custom**.

8. Edit the **Analysis** node, enable the **Coincidence matrices (for symbolic targets)** option, and then close the **Analysis** dialog box.

9. From the **Graphs** palette, add an **Evaluation** node downstream from the **custom** model nugget.

10. Right-click the **Analysis** node and click **Run**.

    The results appear as follows:


Results for output field CHURN

Individual Models

Comparing $D-CHURN with CHURN

Correct	11,625	65.28%
Wrong	6,183	34.72%
Total	17,808	

Coincidence Matrix for $D-CHURN (rows show actuals)

	No	**Yes**
No	11,273	5,783
Yes	400	352

Comparing $D1-CHURN with CHURN

Correct	17,017	95.56%
Wrong	791	4.44%
Total	17,808	

Coincidence Matrix for $D1-CHURN (rows show actuals)

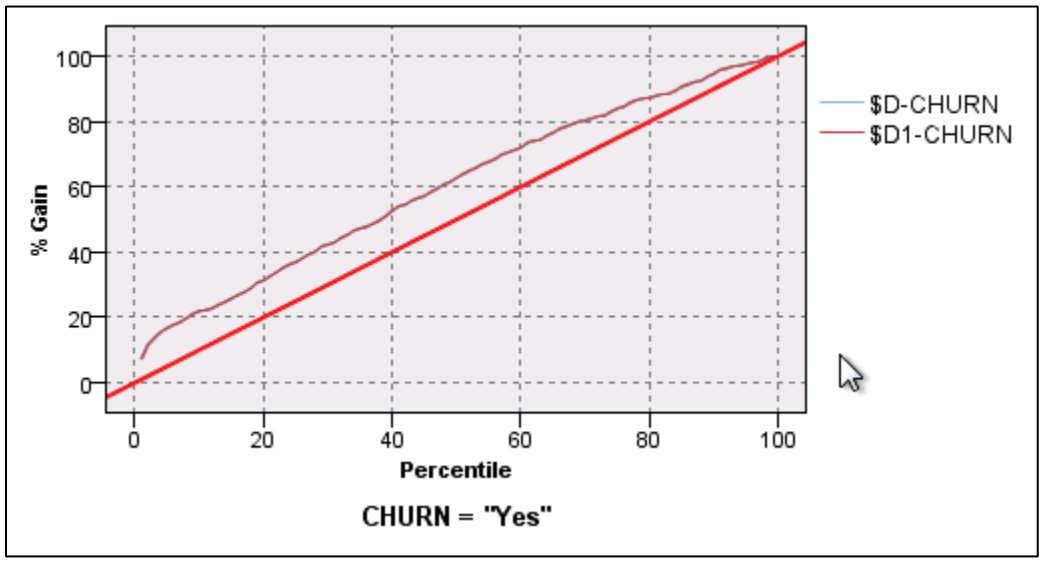
	No	**Yes**
No	17,017	39
Yes	752	0

    The default mode ($D) has a lower accuracy, but identifies the churners better. The model that matches sample proportions almost always predicts No, and does not capture any churner. The reason that almost no churner is identified is because of the prior probabilities that match sample proportions.

11. Close the **Analysis** output window.

12. Right-click the **Evaluation** node and click **Run**.

    The result appears as follows:



    The gain chart shows that both models do not differ when it comes to identifying churners using the ranked scores.

    If you have to choose one of these two models, then the model with adjusted prior probabilities is the preferred one, because the propensities coming out of this model are realistic (assuming that the sample proportions reflect the population proportions).

    Notice that the model is hardly better than using no model at all. This may not come as a surprise because one of the most important predictors, BRAND_HANDSET, is left out of the analysis since it is a categorical field.

13. Close the **Evaluation** graph window.

Task 4. Run and compare two Logistic models.

1. Edit the **Type** node that you have on the stream canvas, set the role for **GADGET_A_REVENUES** and **GADGET_B_REVENUES** to **Input**, and the close the **Type** dialog box.
2. From the **Modeling** palette (**Classification** item), add a **Logistic** node downstream from the **Type** node in the stream.

3. Edit the **Logistic** node, and then:
   - select the **Model** tab
   - for the **Procedure** option, click **Binomial** and ensure that in the **Binomial Procedure** pane, the **Method** is designated as **Enter**
   - close the **Logistic** dialog box
4. Add a second **Logistic** node downstream from the **Type** node.
5. Edit the second **Logistic** node, and then:
   - select the **Model** tab
   - for the **Procedure** option, click **Binomial** and ensure that in the **Binomial Procedure** pane, the **Method** is designated as **Forwards Stepwise**
   - click the **Annotations** tab, click **Custom**, and then type **custom** in the text box
   - close the **Logistic** dialog box
6. Select both **Logistic** nodes, right-click one of them, and then click **Run Selection**.
7. Edit the model nugget named **CHURN**, click the **Advanced** tab, and then scroll down to the **Dependent Variable Encoding** table.

   A section of the results appear as follows:

Dependent Variable Encoding

Original Value	Internal Value
No	0
Yes	1

   For the target, CHURN, Yes is recoded to 1, No to 0. So, the coefficients must be interpreted in odds: (probability CHURN = Yes) / (probability CHURN= No).

8. Scroll down to the **Categorical Variables Codings** table.

   A section of the results appear as follows:

Categorical Variables Codings(a)

		Frequency	Parameter coding		
			(1)	(2)	(3)
BRAND_HANDSET	BS	4125	1.000	.000	.000
	CAS	864	.000	1.000	.000
	NO	7072	.000	.000	1.000
	SA	5744	.000	.000	.000
GENDER	Female	9235	1.000		
	Male	8570	.000		

a. This coding results in indicator coefficients.

   The SA brand is the reference category for BRAND_HANDSET, Male is the reference category for GENDER.

   You will examine the coefficients in the equation. There are two Variables in the Equation tables. One is for Step 0, the initial solution, which is of no interest. The other is for Step 1, the final solution. This is the one you will examine.

9. Scroll down to the second **Variables in the Equation** table.

   Be sure to browse to the table for Step 1. A section of the results appears as follows:

Variables in the Equation

		B	S.E.	Wald	df	Sig.	Exp(B)
Step 1(a)	GENDER(1)	.162	.090	3.243	1	.072	1.176
	AGE	-.013	.004	13.970	1	.000	.987
	DROPPED_CALLS	.050	.010	23.739	1	.000	1.051
	BRAND_HANDSET			1626.751	3	.000	
	BRAND_HANDSET(1)	2.604	.274	90.472	1	.000	13.517
	BRAND_HANDSET(2)	5.915	.269	481.938	1	.000	370.625
	BRAND_HANDSET(3)	2.387	.274	75.840	1	.000	10.877
	BILL_PEAK	-.003	.001	7.031	1	.008	.997
	BILL_OFFPEAK	-.016	.005	8.444	1	.004	.984
	GADGET_A_REVENUES	.014	.010	1.821	1	.177	1.014
	GADGET_B_REVENUES	-.012	.005	4.751	1	.029	.988
	Constant	-5.369	.307	306.275	1	.000	.005

a. Variable(s) entered on step 1: GENDER, AGE, DROPPED_CALLS, BRAND_HANDSET, BILL_PEAK, BILL_OFFPEAK, GADGET_A_REVENUES, GADGET_B_REVENUES.

The exponentiated effect for GENDER (1) equals 1.176. The previous table showed that male is the reference category. Thus, the odds for women are 1.76 times the odds for men. This means that women, other things equal, have a higher probability to churn than men.

The exponentiated effect for age is 0.987. This means that the odds decrease by a factor .987 if age increments by one. Older people are less likely to churn than younger people, other things equal.

The odds for those with brands BS, CAS, or NO are higher than the odds for those with brand SA, with the odds for BRAND_HANDSET (2), CAS, being 370.625 times the odds for brand handset SA. Therefore customers with brand handset CAS have the highest probability to churn, other fields equal.

The GADGET_A_REVENUES field is not significant at the 0.05 level.

10. Close the output window, edit the model nugget named **custom**, and select the **Advanced** tab.
11. Scroll down to the second **Variables in the Equation** table.
12. Locate the **Step 5(e)** section of the table and examine the coefficients.

    The output shows that five predictors are included in the equation. Notice that both gadgets fields were excluded, not only GADGET_A_REVENUES. Apparently, given the information on the five predictors that were added GADGET_B_REVENUES does not add new information. Thus, it matters whether a field (GADGET_B_REVENUES) was set to play an equal role as other predictors (as in method Enter), or whether a field has to add new information to what is already known.
13. Close the **Logistic** output window.

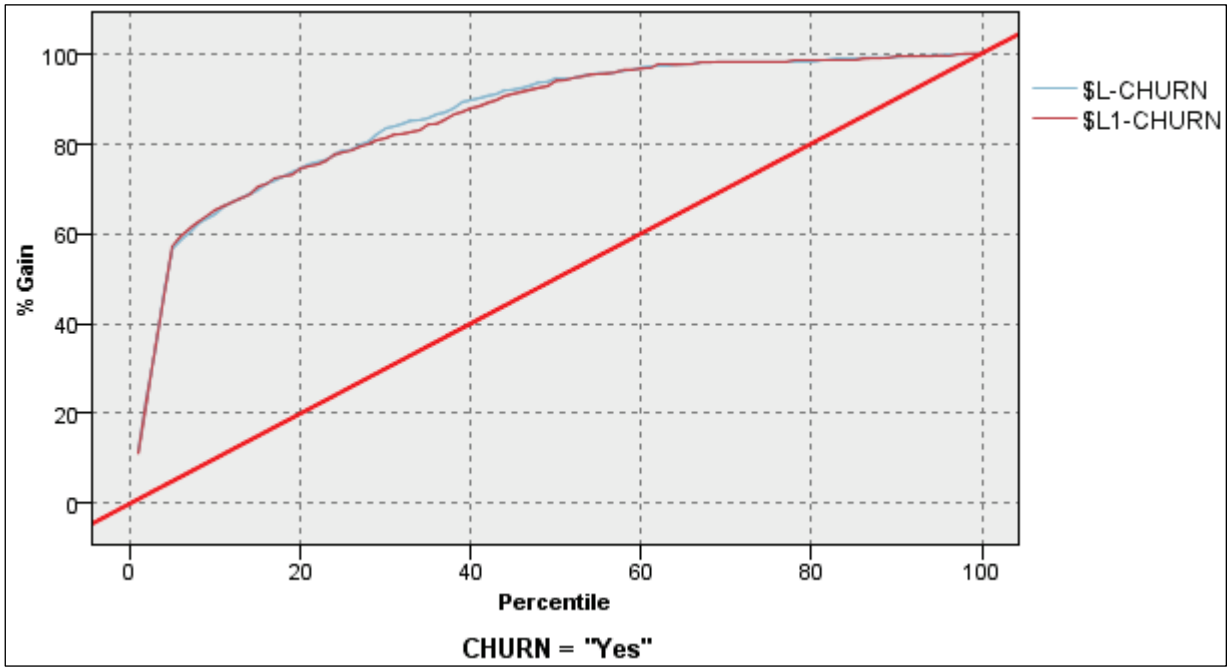
Task 5. Compare the two models in terms of accuracy and gains.

1. Disconnect the model nugget named **custom** from the **Type** node and connect it downstream from the **CHURN** model nugget.
2. From the **Output** palette, add an **Analysis** node downstream from the **custom** model nugget.
3. Edit the **Analysis** node, enable the **Coincidence matrices (for symbolic targets)** option and then close the **Analysis** dialog box.
4. From the **Graphs** palette, add an **Evaluation** node downstream from the model nugget entitled **custom**.
5. Right-click the **Analysis** node and click **Run**.

   The accuracy is about the same, and the numbers churned are comparable. This indicates that both models perform about equally well.
6. Close the **Analysis** output window.

7. Right-click the **Evaluation** node and then click **Run**.

   The result appears as follows:



   The difference between the two models is small. Given these findings, the more parsimonious model (the model with only the predictors that appeared to be significant in the stepwise procedure) is the preferred model.

   The Logistic model outperforms the Discriminant model in terms of accuracy and gain. The main reason is that Logistic takes the categorical predictor BRAND_HANDSET into account, while Discriminant did not do that.

   When you want to include BRAND_HANDSET in Discriminant, you could create indicator fields yourself for BRAND_HANDSET, using the Restructure node. However, using indicator fields in Discriminant will violate the assumption of the multivariate normal distribution of the predictors. This assumption does not apply to Logistic, and nominal predictors are of no concern. Thus, Logistic is a far more attractive model to use than Discriminant when you have categorical predictors.

8. Close the Evaluation window.
9. Close the stream without saving anything.

Results:
You predicted churn data with Discriminant and Logistic and compared the models.

Note: You will find the solution results in the file **demo_using_traditional_statistical_models_completed.str**, located in the **05-Using_Traditional_Statistical_Models\Solutions** sub folder.

Apply Your Knowledge

Use the questions in this section to test your knowledge of the course material.

Question 1: Suppose a flag target Y is predicted by X1 and X2 using Discriminant. Discriminant will compute a new field D and find coefficients a and b for the formula D = a * X1 + b * X2.
True or false: There are no other coefficients, say a* and b*, that separate the groups better than a and b do.

A. True

B. False

Question 2: True or false: It is meaningless to include a nominal field as predictor in Discriminant.

A. True

B. False

Question 3: What theorem does Discriminant use to classify records?

A. Bernstein's theorem

B. Euclid's theorem

C. Bayes' theorem

D. Descartes' Node bisector theorem

Question 4: An estimate of the probability that a record belongs to a particular category when no information from the predictors is available is known as:

A. Prior probability

B. Conditional probability

C. Posterior probability

Question 5: Given that the odds are 2 for a target field fraud, therefore (probability fraud = yes / probability fraud=no) = 2. What then is the probability fraud=yes?

A. 2.

B. 2/3.

C. 1/3.

D. You cannot tell.

Question 6: True or false: In scoring using the Logistic model nugget, when a record has a missing value on any of the predictors, that record will have an undefined ($null$) value for the predicted target field.

A. True

B. False

Question 7: Suppose the field REGION with values East, South, West, and North is a predictor in Logistic. Which one of the following statements is correct?

A. Logistic will discard the REGION field because it is a categorical predictor.

B. Logistic will create four indicator fields behind the scenes for REGION.

C. No indicator field will be created for North, because this category is the reference group.

D. None of the above statements is correct.

Answers to questions:

Answer 1: A. True. Given the scenario described, Discriminant will find the factors a and b in the formula D = a * X1 + b * X2, so that the new field D best separates the two groups.

Answer 2: B. True. Only continuous predictors are relevant in Discriminant.

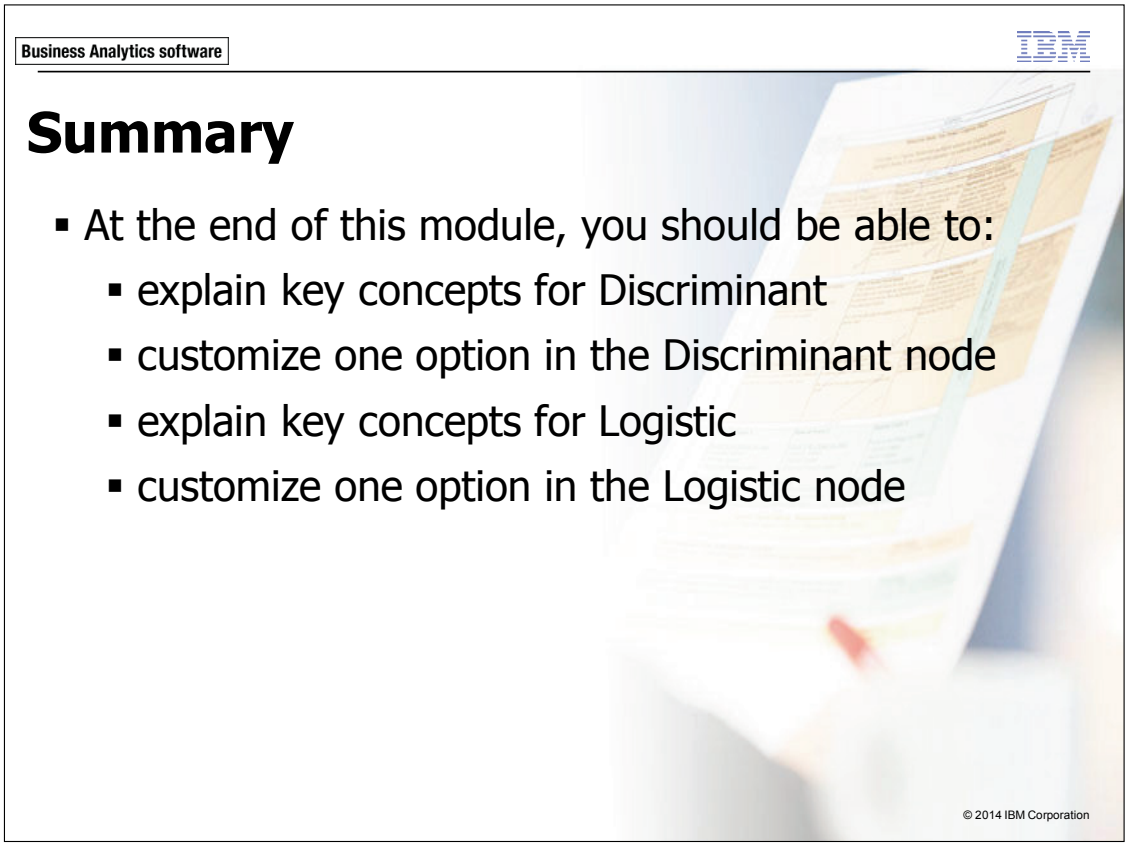
Answer 3: C. Discriminant uses Bayes' theorem to classify records.

Answer 4: A. An estimate of the probability that a record belongs to a particular category when no information from the predictors is available is known as prior probability.

Answer 5: B. The odds are 2, which is only possible (knowing that the probability (fraud=yes) and the probability (fraud=no) must sum to 1), if the ratio probability (fraud=yes) / probability (fraud=no) was (2/3) / (1/3).

Answer 6: A. True. In the Logistic model, a missing value on a predictor will result in a ($null$) value for the predicted target field.

Answer 7: D. None of the statements listed are correct. Logistic can handle categorical predictors, and three indicator fields will be created behind the scenes, with West as reference category.

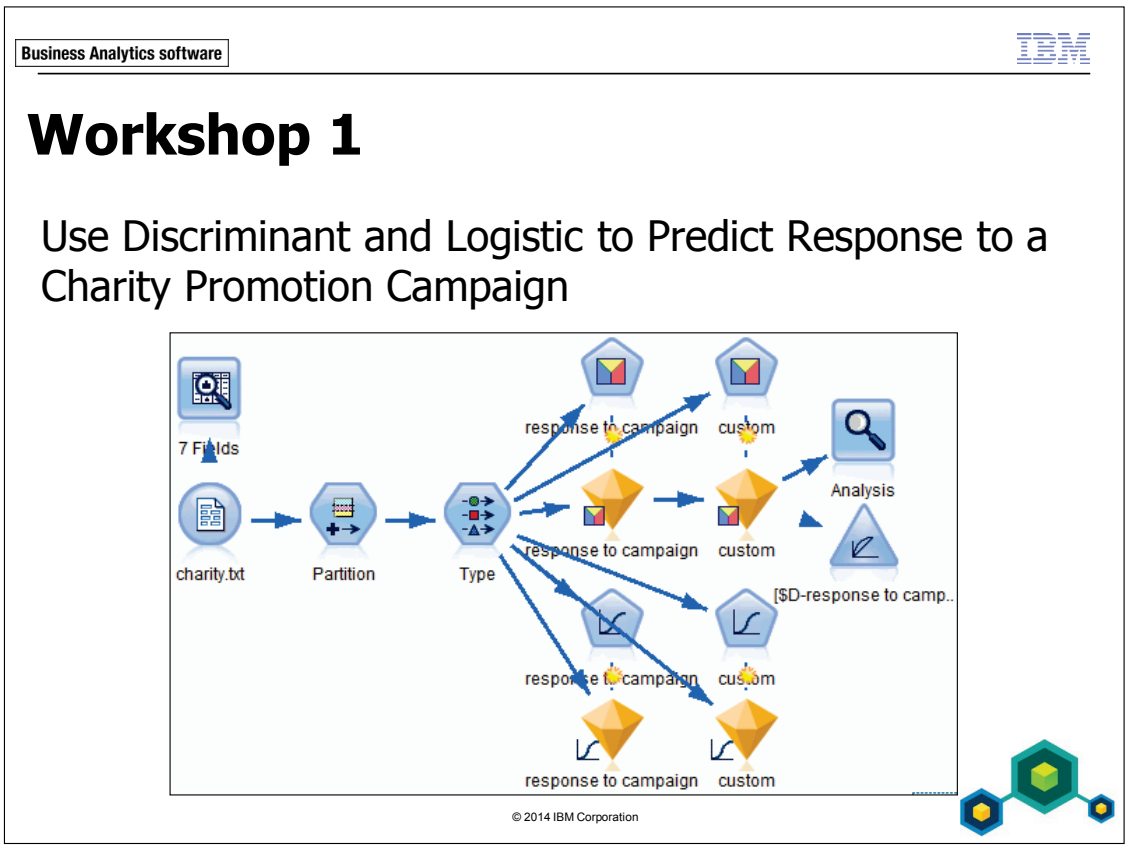


You were introduced to two models to predict a flag target, Discriminant and Logistic.

Both models produce a set of equations. In Discriminant this equation is of little importance and interpreting the coefficients was not pursued here. In Logistic, the coefficients have a useful interpretation when you are familiar with the concept of odds.

Both Discriminant and Logistic can be used when the target is not a flag field but nominal or ordinal. The ideas are the same as presented for a flag field, although there will be more than one equation.

For example, when you have a target with three categories, Discriminant will compute two discriminant score fields to separate the three groups, and Logistic will use two equations, with one category of the target as the reference category. The latter analysis is known as the multinomial model (the other option in the Logistic dialog box).



The following (synthetic) file is used in this workshop:

- **charity.txt**: A text file that represents data from a charity organization. It contains information on individuals who were mailed a promotion. The information includes whether the individuals responded to the campaign, their spending behavior with the charity and basic demographics such as age, gender and demographic group. The file is located in **C:\Train\0A0U5**.

Before you begin the workshop, ensure that:

- You have started MODELER (click **Cancel** when the splash screen appears when you start MODELER).
- You have set MODELER's working folder. (In MODELER, select **File\Set Directory**. When you are working in an IBM training environment, browse to the folder **C:\Train\0A0U5** and then click **Set** to select it. When you are not working in an IBM training environment, ask your trainer for the folder name.)

Workshop 1: Use Discriminant and Logistic to Predict Response to a Charity Promotion Campaign

In this workshop you will predict promotion campaign mail recipient response with Discriminant and Logistic. You will compare the fit of two Discriminant models, and interpret the results of two Logistic models.

To do this, you must:

- Use a **Var. File** node to import data from **charity.txt**, and run a **Data Audit** to examine the data.

  What is the percentage of customers responding positively to the campaign?

- Partition the data into a training set of size **70**, and a testing set of size **30**. Set the seed value to **1234567**, so that you can replicate results.

- Add a **Type** node downstream from the **Partition** node. Edit the **Type** node, instantiate the data, and configure the **Type** node so that **response to campaign** will be predicted by **gender**, **age**, **mosaic bands**, **pre-campaign expenditure**, and **pre -campaign visits**.

- Evaluate two **Discriminant** models: a default **Discriminant** model and a **Discriminant** model with prior probabilities equal to sample group sizes. To differentiate the nodes in the analysis, annotate the **Discriminant** node that matches prior probabilities with group sample sizes.

- Run two **Logistic** models: a default **Logistic** model and a **Logistic** model with the first mosaic band as the reference category. Ensure that you select the **Binomial** procedure for both models. To differentiate the nodes in analysis, annotate the Logistic node where the first mosaic group is the reference group.

  Examine the results for the default model. Which predictors are significant? Do older people have a higher probability to respond to the campaign than younger people? What is the interpretation of, say, the coefficient for mosaic bands (1)? Are the coefficients for the indicator fields for mosaic band significant? Is the coefficient for the entire field mosaic band significant?

Examine the model with the first group as reference. What is the interpretation of, say, the coefficient for mosaic bands (1)? Are the coefficients for the indicator fields for mosaic band significant? And is the coefficient for the entire field mosaic band significant?

Can you think of a reason why the results for the indicator fields are different in the two models?

Workshop 1: Tasks and Results

Task 1. Import and examine the data.

1. From the **Sources** palette, double-click the **Var. File** node to add it to the stream.
2. Edit the **Var. File** node, and then:
   - to the right of the **File** box, click **Browse**  (the Browse window should automatically open to the **C:\Train\0A0U5 folder**)
   - select **charity.txt** and then click **Open**
   - close the **Var. File** dialog box
3. From the **Output** palette, add a **Data Audit** node downstream from the **Var. File** node, run the **Data Audit** node, and double-click the **Sample Graph** for the **response to campaign** field.

   The Data Audit output shows that 31.32% responded to the campaign.
4. Close the **Distribution** window, and then close the **Data Audit** output window.

Task 2. Partition the data into a training set and testing set.

1. From the **Field Ops** palette, add a **Partition** node downstream from the **Var. File** node, set the **Training partition size** to **70%** and the **Testing partition size** to **30%**. Ensure that the **Repeatable partition assignment** option is enabled, with seed value **1234567**.

Task 3. Instantiate the data and set the roles for the fields.

1. From the **Field Ops** palette, add a **Type** node downstream from the **Partition** node.
2. Edit the **Type** node, and then:
   - click the **Read Values**  button

   The Values column is populated with values from the data.
   - set the **Role** for **gender**, **age**, **mosaic bands**, **pre-campaign expenditure**, and **pre-campaign visits** to **Input**
   - set the **Role** for **response to campaign** to **Target**
   - ensure that the **Role** for the **Partition** field is set to **Partition**
   - set the **Role** for all other fields to **None**
3. Close the **Type** dialog box.

Task 4. Add, configure, run, and evaluate two Discriminant models.

1. From the **Modeling** palette (**Classification** item), add a **Discriminant** node downstream from the **Type** node.
2. Add a second **Discriminant** node downstream from the **Type** node.
3. Edit the second **Discriminant** node downstream from the **Type** node, and then:
   - click the **Expert** tab
   - select the **Expert** option
   - for **Prior probabilities**, select **Compute from group sizes**
   - click the **Annotations** tab, select the **Custom** option, and type **custom**
   - close the **Discriminant** dialog box
4. Select the two **Discriminant** nodes, right-click one of them, and click **Run Selection**.
5. Disconnect the nugget named **custom** from the **Type** node and connect it downstream from the model nugget named **response to campaign**.

6. From the **Output** palette, add an **Analysis** node downstream from the model nugget named **custom**, edit it, enable the **Coincidence matrices (for symbolic targets)** option, and then close the **Analysis** node.
7. From the **Graphs** palette, add an **Evaluation** node downstream from the model nugget named **custom**.
8. Right-click the **Analysis** node and click **Run**.

   A section of the results appear as follows:

Results for output field response to campaign

Individual Models

Comparing $D-response to campaign with response to campaign

'Partition'	1_Training		2_Testing	
Correct	1,205	73.75%	547	71.6%
Wrong	429	26.25%	217	28.4%
Total	1,634		764	

Coincidence Matrix for $D-response to campaign (rows show actuals)

'Partition' = 1_Training	No	Yes
No	933	200
Yes	229	272

'Partition' = 2_Testing	No	Yes
No	416	98
Yes	119	131

Comparing $D1-response to campaign with response to campaign

'Partition'	1_Training		2_Testing	
Correct	1,203	73.62%	554	72.51%
Wrong	431	26.38%	210	27.49%
Total	1,634		764	

Coincidence Matrix for $D1-response to campaign (rows show actuals)

'Partition' = 1_Training	No	Yes
No	1,108	25
Yes	406	95

'Partition' = 2_Testing	No	Yes
No	500	14
Yes	196	54

The Analysis output shows that the custom model achieves a higher accuracy on the testing set.

The custom model does not identify as many responders as the default model does. This is because the default model uses a prior probability of 0.5 for each of the two categories response to mailing=No and response to mailing=Yes, while the custom model uses prior probabilites equal to the sample proportions, which will weigh the response to mailing=No heavier than the response to mailing=Yes category.

9. Close the **Analysis** window.
10. Right-click the **Evaluation** node and click **Run**.

    The chart shows that the gain for both models is almost identical.

    To score records, the custom model is preferred, because this model will not inflate the propensities (assuming that the sample proportion match the population proportions).
11. Close the **Evaluation** output window.

Task 5. Add, configure, run, and interpret two Logistic models.

1. From the **Modeling** palette (**Classification** item), add a **Logistic** node downstream from the **Type** node.
2. Edit the **Logistic** node, and then:
   - click the **Model** tab
   - for **Procedure**, select the **Binomial** option
   - close the **Logistic** dialog box
3. Add a second **Logistic** node downstream from the **Type** node.
4. Edit the second **Logistic** node, and then:
   - click the **Model** tab
   - for **Procedure**, select the **Binomial** option
   - below **Categorical inputs**, select **mosaic bands**, and for **Base Category**, select **First**
   - click the **Annotations** tab, select the **Custom** option, and type **custom**
   - close the **Logistic** dialog box
5. Select the two **Logistic** nodes, right-click one of them, and click **Run Selection**.

6. Edit the Logistic model nugget named **response to campaign**, click the **Advanced** tab, and scroll down to the **Variables in the Equation** table (the last table in the output).

   A section of the results appear as follows:

Variables in the Equation

		B	S.E.	Wald	df	Sig.	Exp(B)
Step 1(a)	gender(1)	-.131	.133	.963	1	.326	.878
	age	.016	.005	12.596	1	.000	1.017
	mosaic bands			22.778	11	.019	
	mosaic bands(1)	-22.251	40191.492	.000	1	1.000	.000
	mosaic bands(2)	-22.241	40191.492	.000	1	1.000	.000
	mosaic bands(3)	-22.377	40191.492	.000	1	1.000	.000
	mosaic bands(4)	-22.789	40191.492	.000	1	1.000	.000
	mosaic bands(5)	-22.606	40191.492	.000	1	1.000	.000
	mosaic bands(6)	-23.108	40191.492	.000	1	1.000	.000
	mosaic bands(7)	-22.291	40191.492	.000	1	1.000	.000
	mosaic bands(8)	-23.450	40191.492	.000	1	1.000	.000
	mosaic bands(9)	-22.489	40191.492	.000	1	1.000	.000
	mosaic bands(10)	-22.217	40191.492	.000	1	1.000	.000
	mosaic bands(11)	-21.907	40191.492	.000	1	1.000	.000
	pre-campaign expenditure	.001	.002	.404	1	.525	1.001
	pre-campaign visits	.263	.024	123.433	1	.000	1.301
	Constant	19.733	40191.492	.000	1	1.000	371427341.665

a. Variable(s) entered on step 1: gender, age, mosaic bands, pre-campaign expenditure, pre-campaign visits.

The age field and the pre-campaign visits field are significant at the 0.05 level. When age increments by 1 (year), the odds change by a factor of 1.017.

For example, the odds for a 40-year-old person equal the odds for a 30-year-old person * 1.017^{10} . All in all, age has a positive effect on response: older people will have a higher probability to respond than younger people.

All indicator fields for mosaic bands are not significant (significance is 1). Because the last category, mosaic band 12, was taken as reference category, this means that none of the other mosaic bands differ signifcantly in odds from the odds of mosaic band 12 (all other fields equal). However, the overall effect of mosaic band is significant at the 0.05 level, which means that there are differences in odds between the mosaic bands. To solve this puzzle you will examine the results of the second model.

7. Close the **Logistic** output window.
8. Edit the Logistic model nugget named **custom**, click the **Advanced** tab, and scroll to the **Categorical Variables Codings** table.

   A section of the results appear as follows:

Categorical Variables Codings

		Frequency	Parameter coding										
			(1)	(2)	(3)	(4)	(5)	(6)	(7)	(8)	(9)	(10)	(11)
mosaic bands	Mos_01	269	.000	.000	.000	.000	.000	.000	.000	.000	.000	.000	.000
	Mos_02	231	1.000	.000	.000	.000	.000	.000	.000	.000	.000	.000	.000
	Mos_03	325	.000	1.000	.000	.000	.000	.000	.000	.000	.000	.000	.000
	Mos_04	144	.000	.000	1.000	.000	.000	.000	.000	.000	.000	.000	.000
	Mos_05	37	.000	.000	.000	1.000	.000	.000	.000	.000	.000	.000	.000
	Mos_06	135	.000	.000	.000	.000	1.000	.000	.000	.000	.000	.000	.000
	Mos_07	151	.000	.000	.000	.000	.000	1.000	.000	.000	.000	.000	.000
	Mos_08	49	.000	.000	.000	.000	.000	.000	1.000	.000	.000	.000	.000
	Mos_09	87	.000	.000	.000	.000	.000	.000	.000	1.000	.000	.000	.000
	Mos_10	162	.000	.000	.000	.000	.000	.000	.000	.000	1.000	.000	.000
	Mos_11	43	.000	.000	.000	.000	.000	.000	.000	.000	.000	1.000	.000
	Mos_12	1	.000	.000	.000	.000	.000	.000	.000	.000	.000	.000	1.000

The first mosaic band is the reference category now. Notice that mosaic band 12, which was the reference category in the default model, only contains 1 record in the training dataset.

9. Scroll down to the **Variables in the Equation** table (the last table in the output).

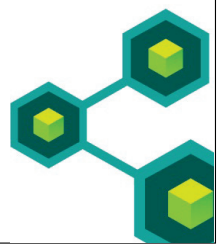
   Some indicator fields are significant. For example, indicator field mosaic band (5) has an exponentiated value of 0.425, with a significance of 0.03. Recalling the encoding for mosaic bands, this means that the odds for mosaic band 6 are 0.425 times the odds of mosaic band 1 (the reference category). Thus, those in mosaic band 6 have a smaller probability to respond, compared to those in mosaic band 1.

   Notice that the overall effect of mosaic bands did not change. Again, this tells you that there are differences between the mosaic bands with respect to response. Because the reference group in the default model was comprised of only one record, this significance was not reflected in the indicator fields. When the reference group has an adequate number of records in it, as in this analysis, differences between mosaic groups can be detected.
10. Close the **Logistic** output window.
11. Exit MODELER without saving anything.

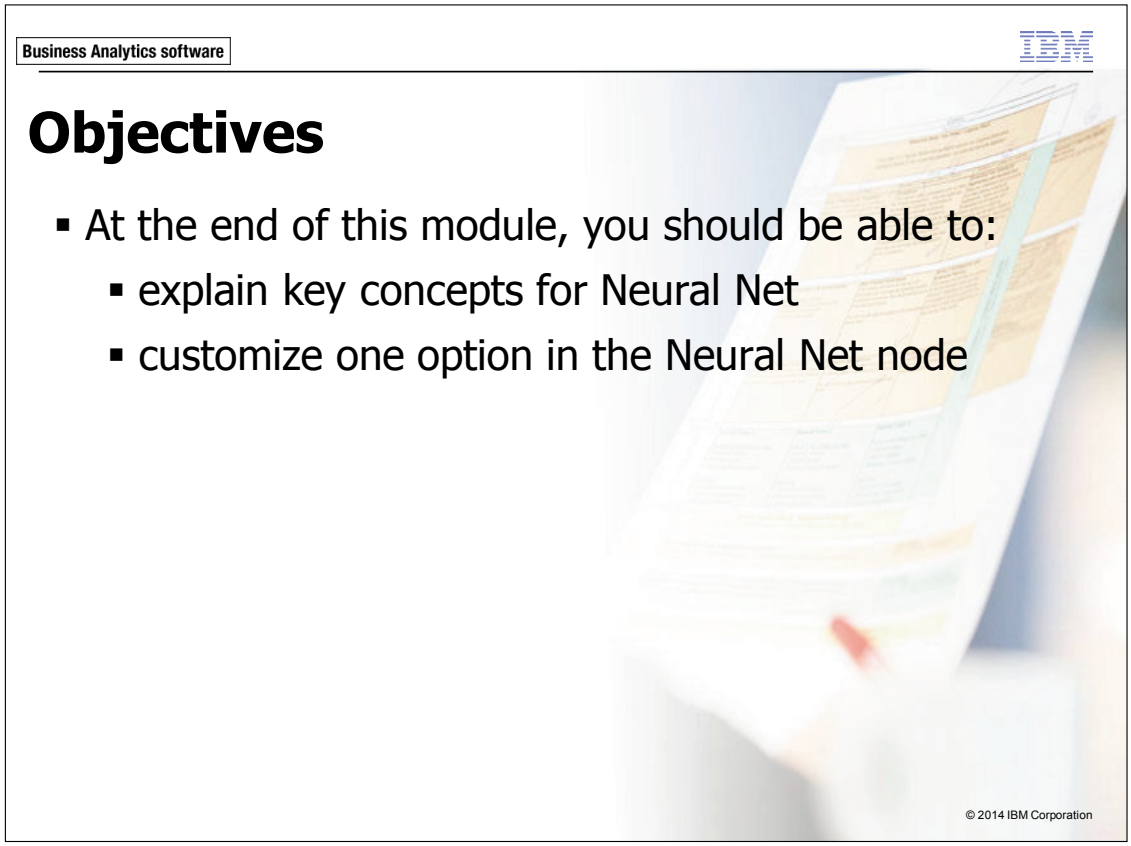
Note: The stream **workshop_using_traditional_statistical_models_completed.str**, located in the **05-Using_Traditional_Statistical_Models\Solutions** sub folder, provides a solution to the workshop tasks.

Using Machine Learning Models

IBM SPSS Modeler (v16)

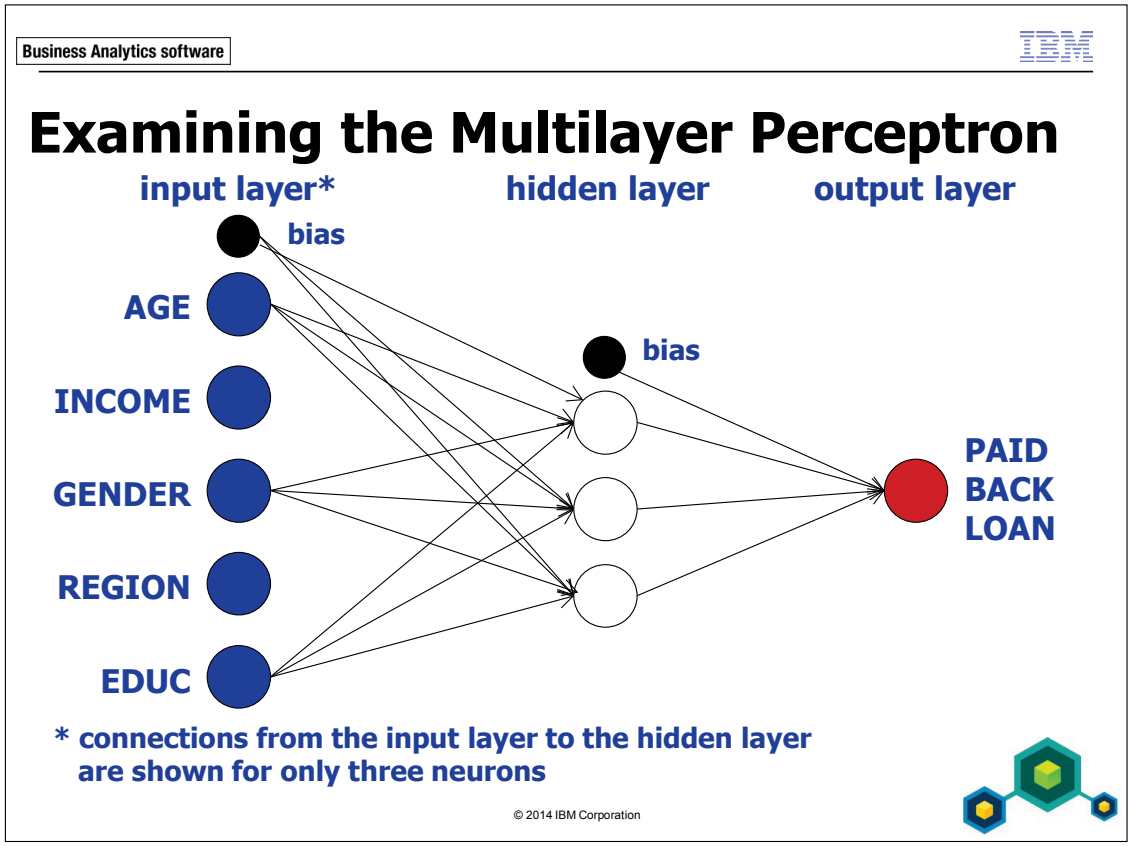


Business Analytics software



Before reviewing this module, you should be familiar with the following topics:

- working with MODELER (streams, nodes, palettes)
- importing data (Var. File node)
- defining measurement levels, roles, blanks, and instantiating data (Type node)
- examining the data (Table node, Data Audit node)
- assessing the quality of your model, using accuracy, risk estimate, gain and response measures
- using the model nugget to score data

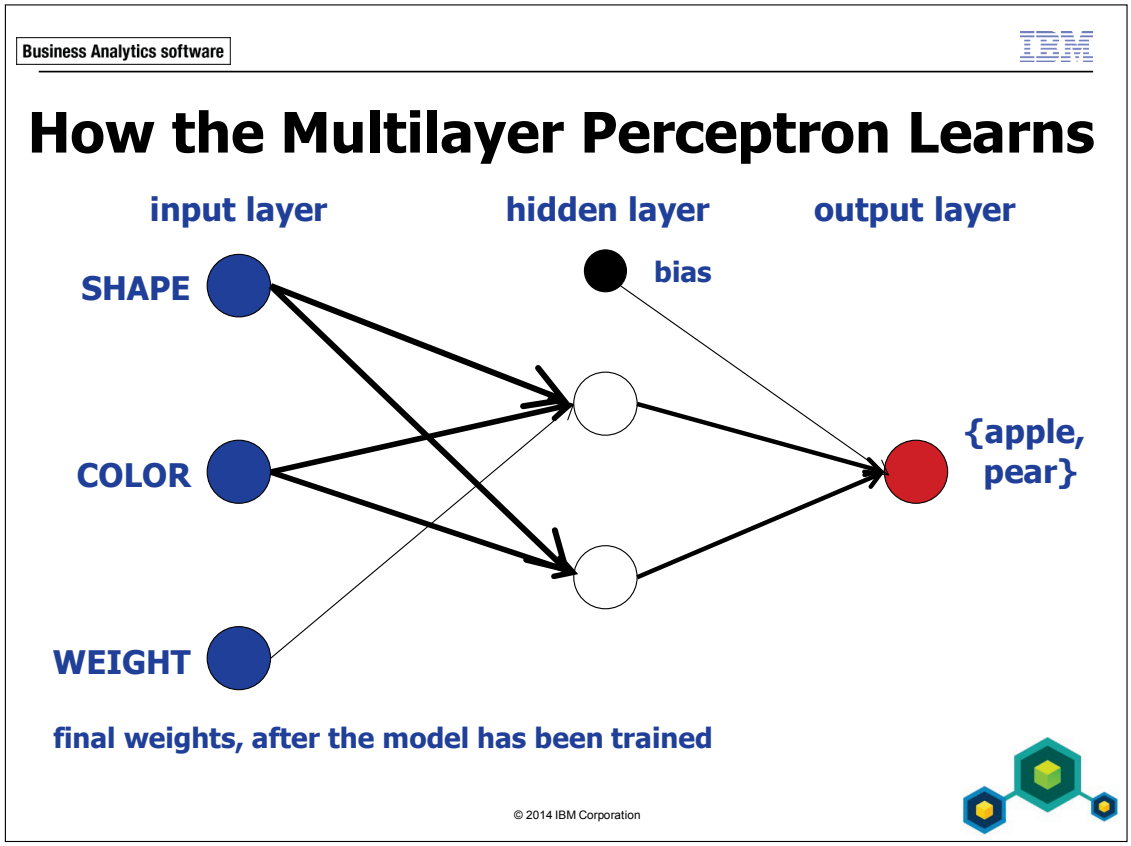


Historically, neural networks attempted to solve problems using methods based on how the brain operates. Today they are generally viewed as powerful modeling techniques.

Two neural net models are available in Modeler: the Multilayer Perceptron and the Radial Basis Function. Key concepts are reviewed in this module. Refer to the *Statistical Algorithms Guide for IBM SPSS Modeler 16* for information on the algorithms.

The Multilayer Perceptron consists of several processing units, the neurons, arranged in layers to create a network. The neurons in the input layer represent the predictors. The neuron in the output layer represents the target. Each neuron in the hidden layer receives an input based on a weighted combination of the values of the neurons in the previous layer. The neurons within the hidden layer are, in turn, combined to produce an output value, the prediction.

This predicted value is compared to the actual value of the target and the difference between the two values (the error) is fed back into the network (known as "back propagation"), which in turn is updated.

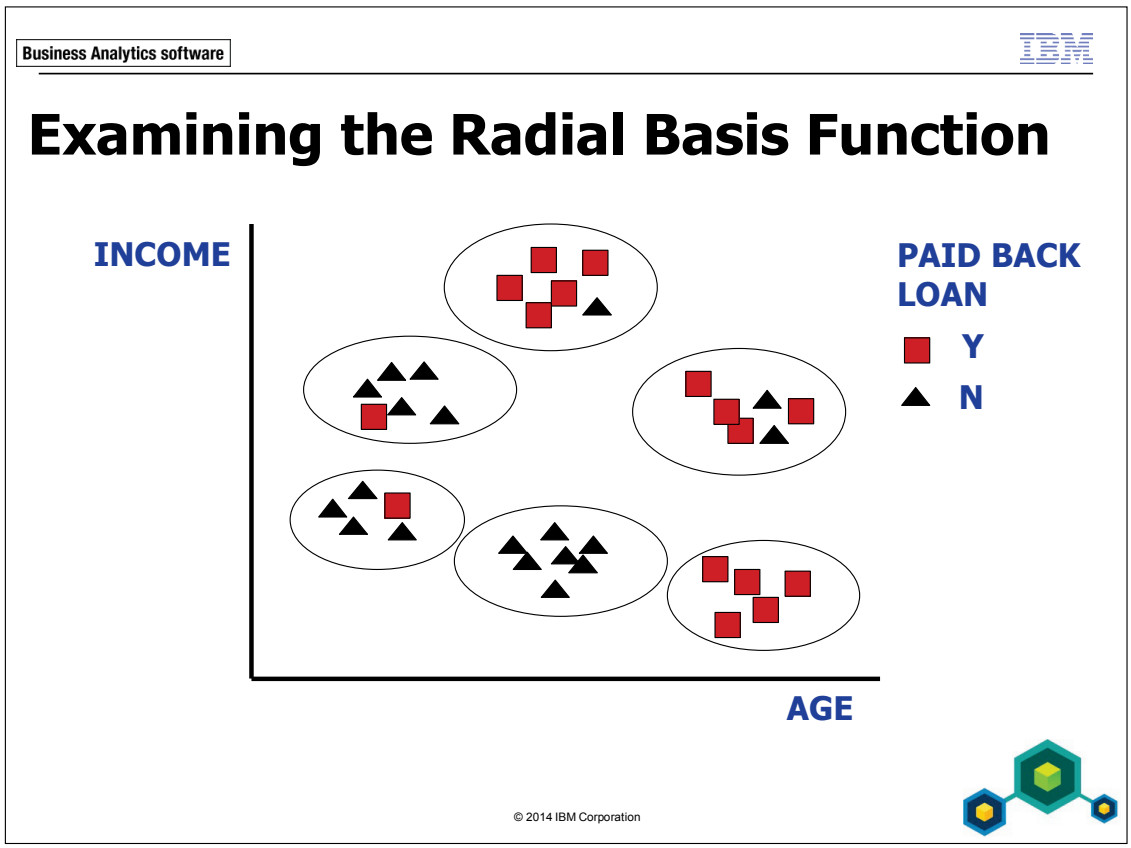


To illustrate this process, consider the example of a child learning the difference between an apple and a pear.

When shown the first example of a fruit, she may look at the fruit and decide that it is round, red in color and of a particular weight. Not knowing what an apple or a pear actually looks like, the child may decide to place equal importance on each of these factors. The importance is what a network refers to as weights.

At this stage the child is most likely to randomly choose either an apple or a pear for her prediction. On being told the correct response, the child will increase or decrease the relative importance of each of the factors to improve her decision (reduce the error).

In a similar fashion a Multilayer Perceptron network begins with random weights placed on each of the inputs. On being told the actual value of the target, the network adjusts these internal weights. In time, the child and the network will hopefully make correct predictions.



The Radial Basis Function (RBF) is a more recent type of network and is quicker to train than the Multilayer Perceptron.

The RBF can be thought of performing a type of clustering within the input space, encircling individual clusters of data by a number of so-called basis functions. If a data point falls within the region of activation of a particular basis function, then the neuron corresponding to that basis function responds most strongly.

The concept of the RBF is extremely simple; but the selection of the centers of each basis function is where difficulties arise.

Business Analytics software

How Neural Net Handles Categorical Predictors

- Create indicator fields for each category of a flag, nominal or ordinal field

MARITAL STATUS	MS1	MS2	MS3
divorced	1	0	0
married	0	1	0
single	0	0	1

A categorical field is recoded into a number of indicator fields. The value of the indicator field corresponding to the category that it indicates is set to 1, and the other indicator fields are set to 0.

The figure in this slide gives an example for a field marital with values divorced, married and single. The field marital is transformed into three new fields I1, I2, and I3. The I1 field is an indicator for the divorced category, I2 is an indicator for the married category, and I3 is an indicator for the single category.

Business Analytics software

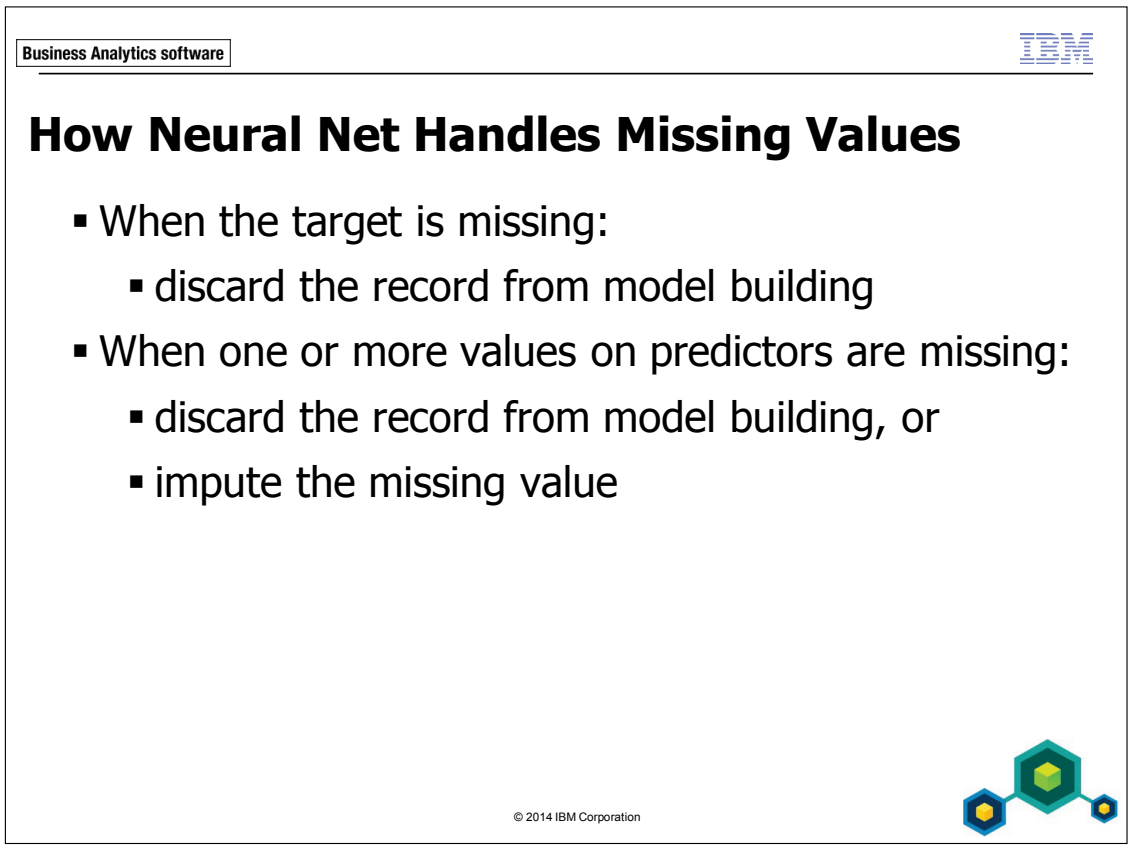
IBM

How Neural Net Handles Continuous Predictors

- Rescaled so values are in range [0, 1]
- Makes interpretation (even more) difficult

In most datasets there is a great deal of variability in the scale of continuous fields. For example, consider income and household size. If both of these fields are used in their natural scale as predictors, income is likely to be given much more weight in the model than household size, simply because the values (and therefore the differences between records) for the former are so much larger than for the latter.

To compensate for this effect of scale, continuous fields are rescaled to have values between 0 and 1. This is one of the reasons that it is difficult to interpret the coefficients in the network, because the fields are not in their original scale.



By default, records with a missing value (a user-defined blank or an undefined ($null$) value) on any of the predictors is discarded from the analysis. This is known as listwise deletion.

Alternatively, Neural Net can impute (replace) missing values on a predictor by substituting a value for it. The value that is imputed depends on the measurement level of the predictor:

- Categorical fields: The most frequently occurring category, or mode, is imputed.
- Continuous fields: The average of the minimum and the maximum observed value is imputed.

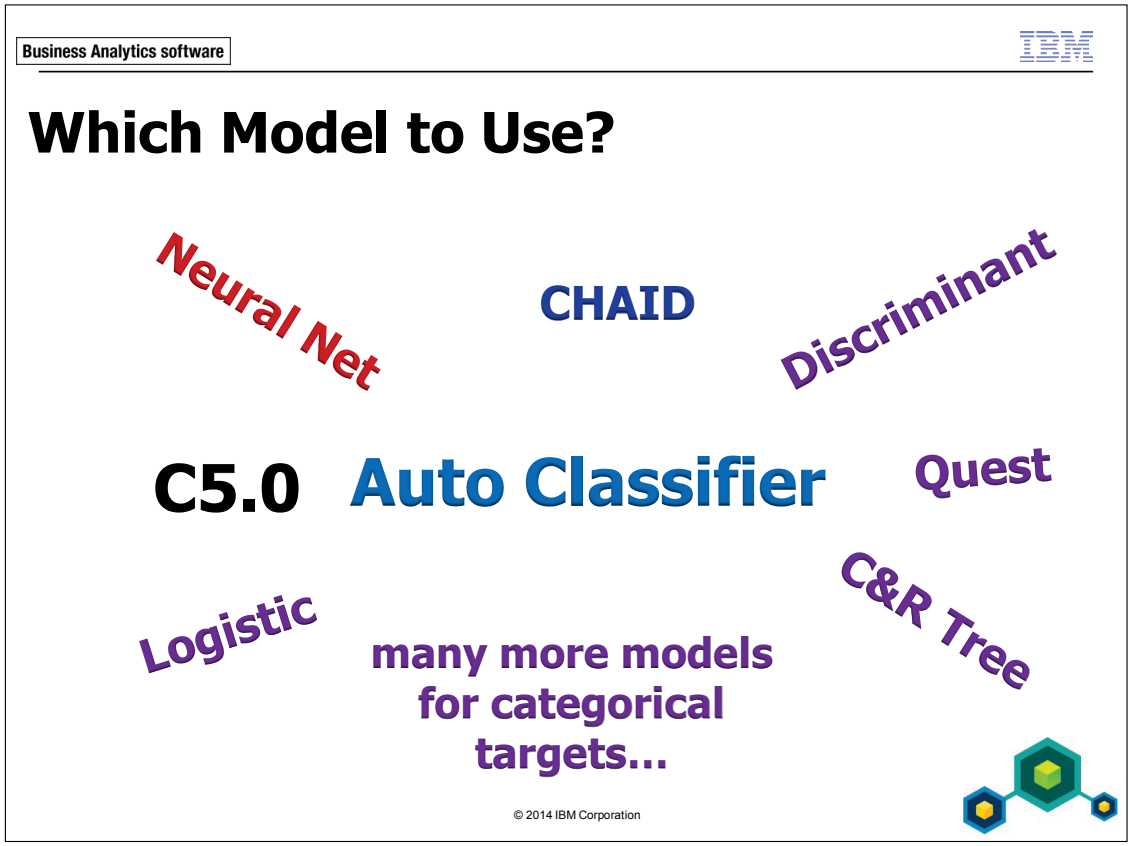
After imputation, the above described transformations for categorical and continuous fields are applied.

Business Analytics software

Exploring Neural Net

- Supports boosting and bagging
- The model nugget can add:
  - predicted category
  - confidence for the predicted category
  - propensities

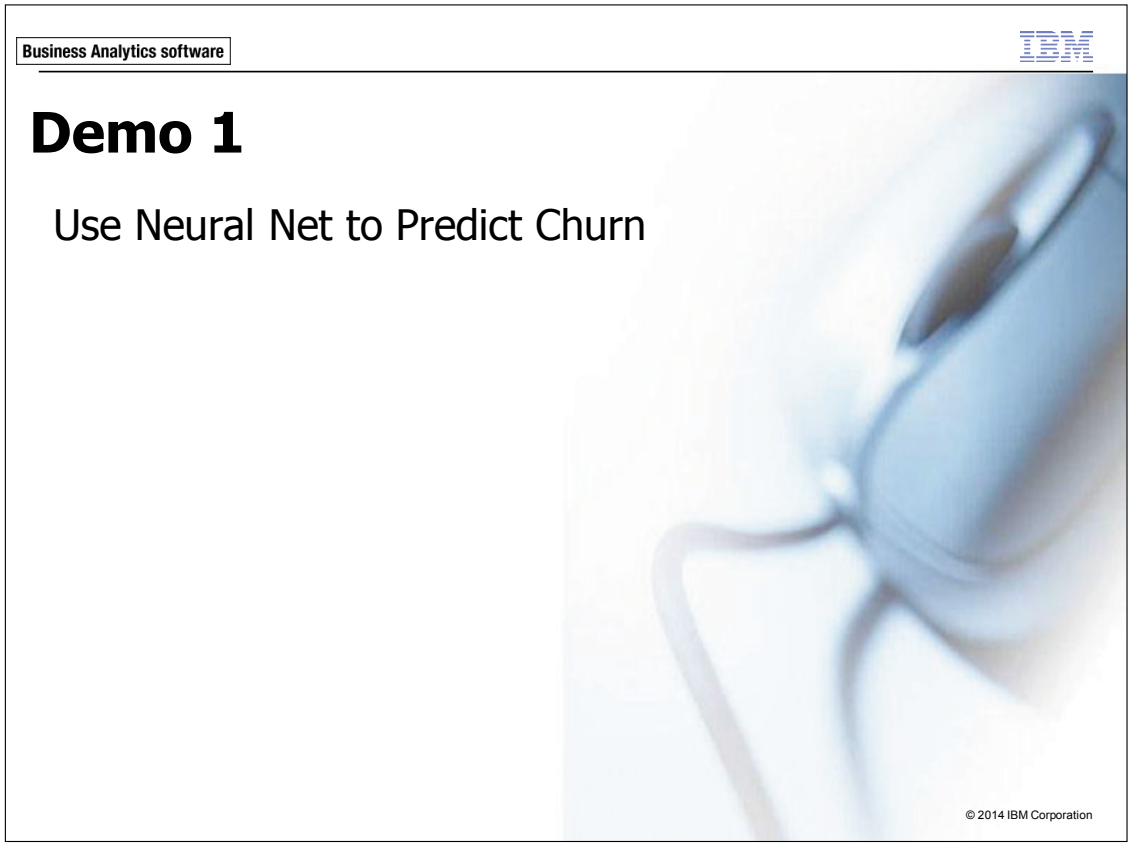
This slide summarizes the capabilities of Neural Net.



This course introduced you to a number of models to predict categorical targets, with business use cases focusing on classifying customers.

As pointed out throughout the course, it is a business decision which model to use. There is not one best model.

Also mentioned throughout the course is MODELER's capability to combine multiple models into a single one. If you plan to do so, consider using the Auto Classifier node (Modeling palette, Automated item). Refer to the *Automated Data Mining with IBM SPSS Modeler* course, or the *Advanced Predictive Modeling using IBM SPSS Modeler* course for more information.



The following (synthetic) file coming from a (fictitious) telecommunications firm is used to demonstrate how you use Neural Net: **telco x modeling data.txt**: Information on approximately 32,000 customers of the firm. The data includes demographics, calling minutes, and product features, as well as a churned status. Churn status is stored in a field named churn. The values for churn can be either Active for current customers, or Churned, for churned customers. The file is located in **C:\Train\0A0U5**. Before you begin with the demo, ensure that:

- You have started MODELER (click **Cancel** when the splash screen appears when you start MODELER).
- You have set MODELER's working folder. (In MODELER, select **File\Set Directory**. When you are working in an IBM training environment, browse to the folder **C:\Train\0A0U5** and then click **Set** to select it. When you are not working in an IBM training environment, ask your trainer for the folder name.)

Demo 1: Use Neural Net to Predict Churn

Purpose:
You are working as a data miner for a telecommunications firm. You have to identify customers who are likely to churn. Two models are built and evaluated for the Telco data: a default Neural Net model and a second Neural Net model with two hidden layers.

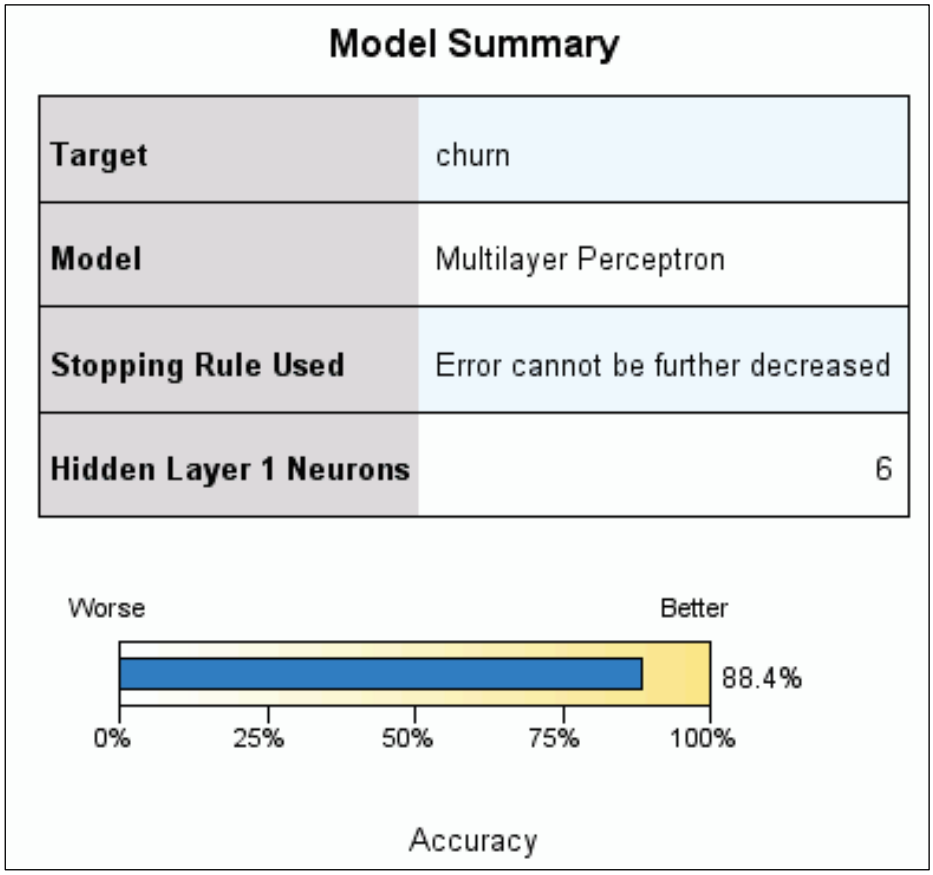
Task 1. Build a model using historical data.

1. Use a **Var. File** (**Sources** palette) to import data from **telco x modeling data.txt**.
2. Add a **Type** node downstream from the **Var. File** node.
3. Edit the **Type** node, and then:
   - click **Read Values** to instantiate the data
   - set the **Role** for **gender**, **age**, **tariff**, **dropped_calls**, **handset**, **bill_peak**, and **bill_offpeak** to **Input**
   - set the **Role** for **churn** to **Target**
   - set the **Role** for the other fields to **None**
   - close the **Type** dialog box
4. From the **Modeling** palette (**Classification** item), add a **Neural Net** node downstream from the **Type** node
5. Add a second **Neural Net** node downstream from the **Type** node.

6. Edit the second **Neural Net** node, and then:
   - click the **Build Options** tab
   - select the **Basics** item
   - select the **Customize number of units** option
   - set the number of units of the first layer to **5**
   - set the number of units of the second layer to **5**
   - click the **Annotations** tab
   - select the **Custom** option
   - type **custom**
   - close the **Neural Net** dialog box
7. Run both **Neural Net** nodes.
8. Edit the **model nugget** named **churn**.

9. Click the **Model Summary** item.

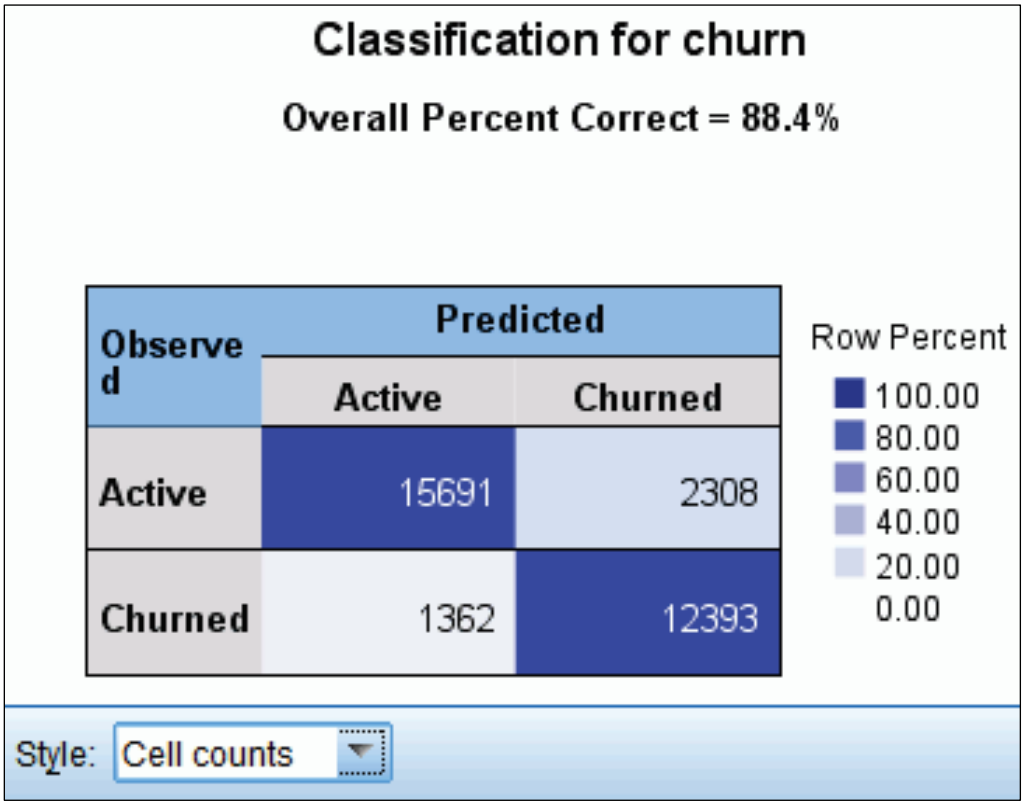
   A section of the results appear as follows:



   The accuracy (percentage of records that is classified correctly) is 88.4%. To view the details, you will select the classification item.

10. Click the **Classification** item, and for **Style** at the bottom in the output window, select **Cell counts**.

    A section of the results appear as follows:



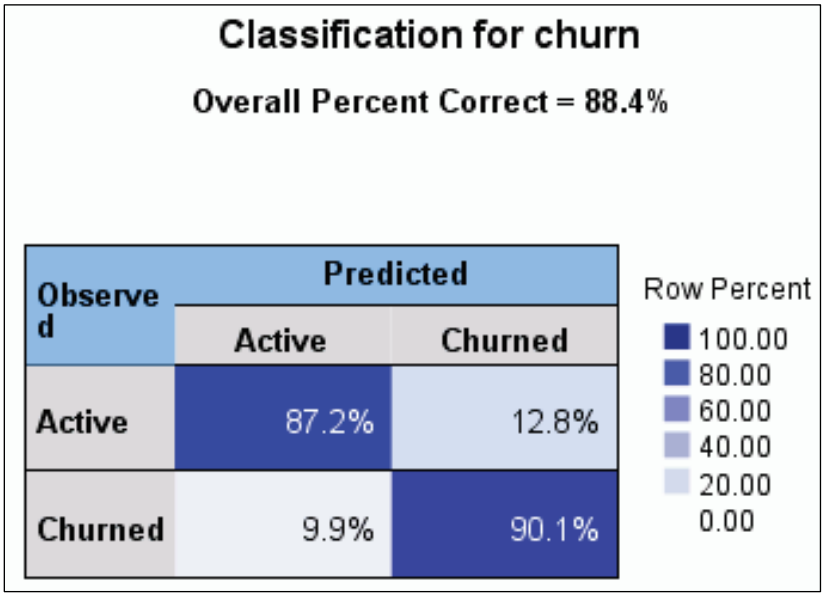
    In total, (15,691 + 12,393)/31,754 * 100 = 88.4% of the records were classified correctly, as was indicated by the model summary.

    The total number of records (31,754) is not the same as the number of records in the dataset (31,769), because there were 15 records that have a missing value on one or more predictors. These were ignored in model building. You can impute missing values to use the entire dataset. This feature is not demonstrated in this course.

    You will examine the percentages correct for each of the observed categories.

11. For **Style**, select **Row percents**.

    A section of the results appear as follows:

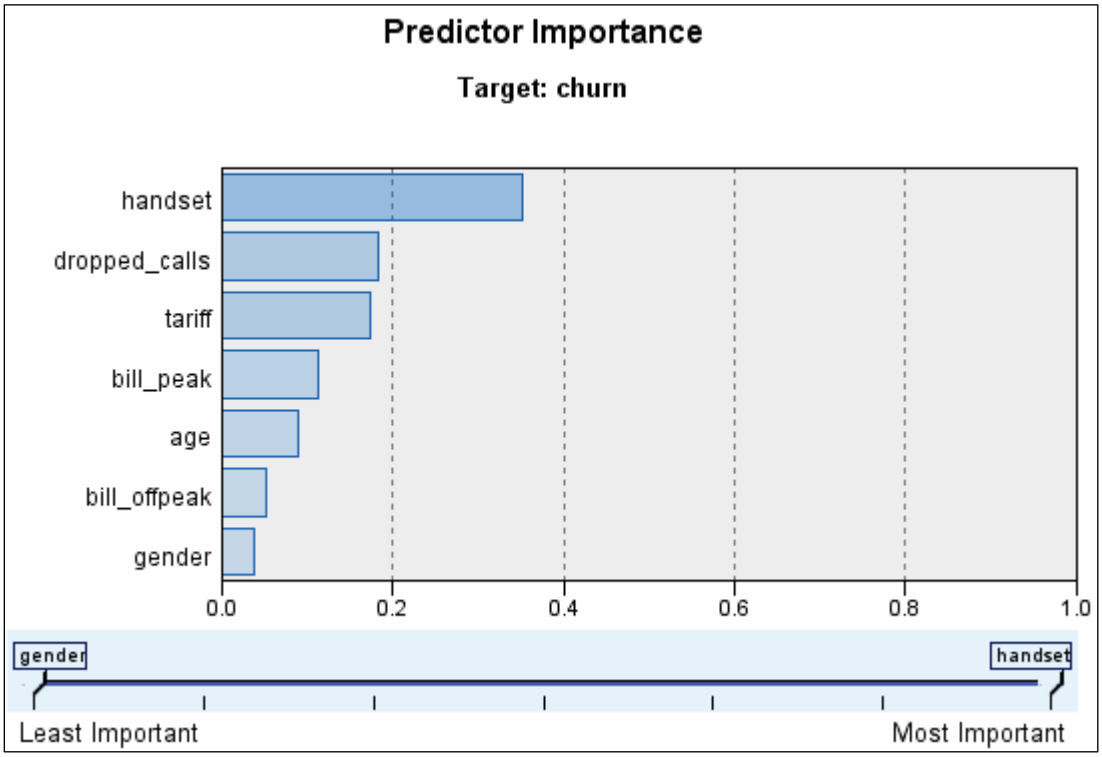


    Of the persons that actually churned%, 90.1% have been found by the model, which is an excellent result.

    You will examine which are the most important predictors.

12. Select the **Predictor Importance** item.

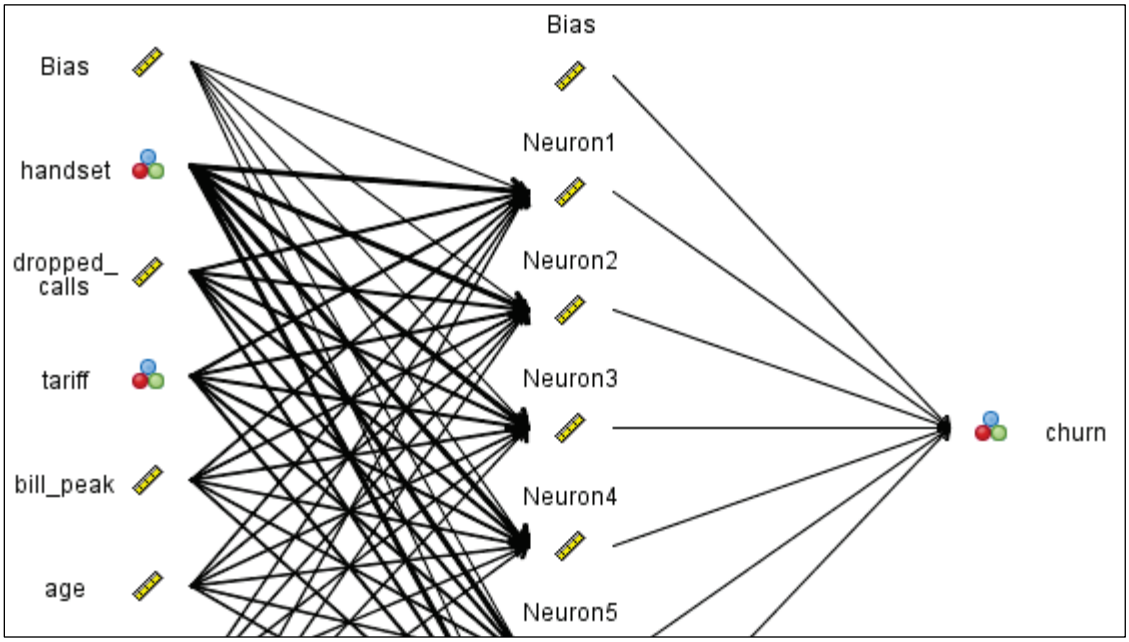
    A section of the results appear as follows:



Handset is by far the most important predictor. This will be reflected in the diagram of effects

13. Select the **Network** item, and ensure that **Effects** is selected for **Style**.

    A section of the results appear as follows:

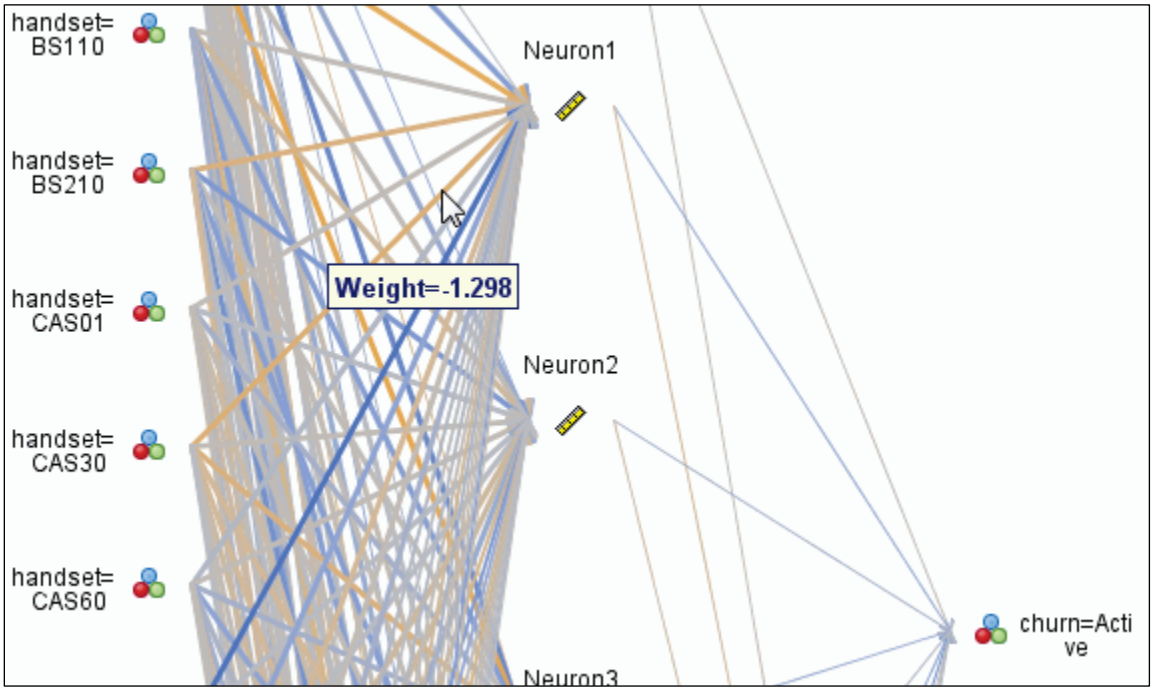


The hidden layer in the network is comprised of 6 neurons. The arrows originating from handset have the heaviest weight, confirming that handset is the most important predictor.

You will view the diagram of coefficients to examine how the handsets relate to the categories of the target field. Recall, that Neural Net creates indicator fields for each categorical predictor, so each handset will be present in the diagram of coefficients.

14. For **Style**, select **Coefficients**.

    A section of the results appear as follows:



    The legend explains that orange represents negative weights, and blue arrows positive weights.

    For the interpretation, examine handset CAS30. The indicator field for this handset has a negative weight to Neuron 1, and in its turn, Neuron 1 has a positive weight on churn=Active, and a negative weight on churn=Churned.

    This means that CAS30 has an overall negative effect on churn=Active and an overall positive effect on churn=Churned. In other words, compared to other handsets and with other fields being equal, CAS30 has a higher percentage of churners.

15. Close the **model nugget**.

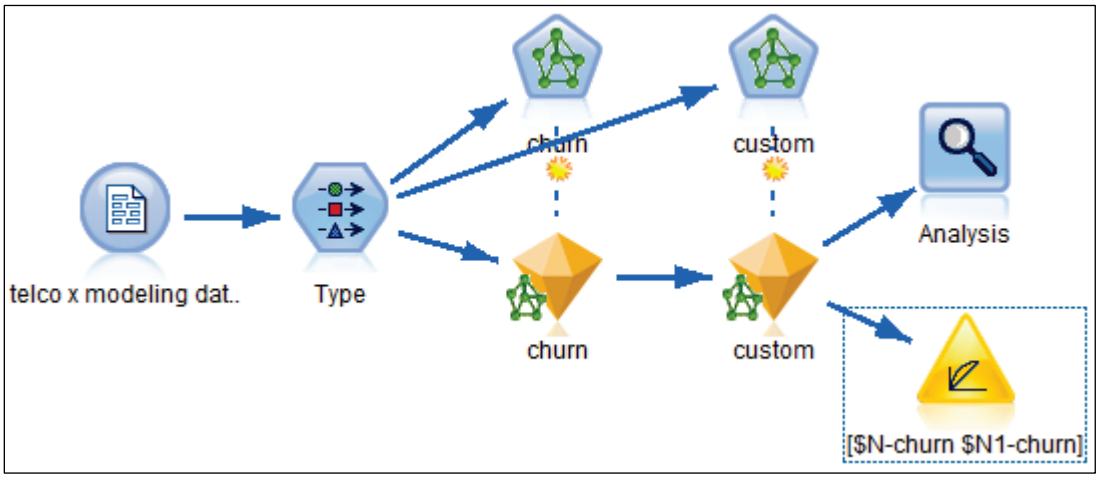
    You will not examine the output for the second model. Instead, you will evaluate both models with an Analysis node and an Evaluation graph.

16. Rearrange the nodes so that the Neural Net model nugget named **custom** is downstream from the model nugget named **churn**.

17. From the **Output** palette, add an **Analysis** node downstream from the model nugget named **custom**.

18. Edit the **Analysis** node, enable the **Coincidence matrices (for symbolic targets)** option, and then close the **Analysis** dialog box.
19. From the **Graphs** palette, add an **Evaluation** node downstream from the **model nugget** named **custom**.

    A section of the results appear as follows:



20. Run the **Analysis** node.

    A section of the results appear as follows:

Results for output field churn

Individual Models

Comparing $N-churn with churn

Correct	28,084	88.4%
Wrong	3,685	11.6%
Total	31,769	

Coincidence Matrix for $N-churn (rows show actuals)

	Active	**Churned**	**$null$**
Active	15,691	1,362	3
Churned	2,308	12,393	12

Comparing $N1-churn with churn

Correct	28,085	88.4%
Wrong	3,684	11.6%
Total	31,769	

Coincidence Matrix for $N1-churn (rows show actuals)

	Active	**Churned**	**$null$**
Active	15,648	1,405	3
Churned	2,264	12,437	12

The Analysis output shows that the results are approximately the same. The gain for both models is also the same.

It looks like the custom model, with two hidden layers and five neurons in each layer, overfits the data, although you would redo the analysis with a training and a testing set to conclude this with more certainty.

21. Close the **Analysis** output window.
22. Close the stream without saving anything.

Results:
You predicted churn with two Neural Net models, and compared the models with the usual evaluation measures.

Note: You will find the solution results in the file **demo_using_machine_learning_models_completed.str**, located in the **06-Using_Machine_Learning_Models\Solutions** sub folder.

Apply Your Knowledge

Use the questions in this section to test your knowledge of the course material.

Question 1: Is the following statement true or false? A Neural Net learns by examining individual records, generating a prediction for each record, and making adjustments to the weights whenever it makes an incorrect prediction. This process is repeated many times, and the network continues to improve its predictions until one or more of the stopping criteria have been met.

A. True

B. False

Question 2: Which of the following statements is the correct one?

A. Neural Net will group categories of a predictor when categories are not significantly different with respect to the target.

B. Neural Net cannot handle missing values on predictors.

C. Neural Net creates an indicator field behind the scenes for each category of a nominal predictor.

D. A Neural Net automatically discards predictors that do not contribute significantly to the accuracy of the predictions.

Question 3: Is the following statement true or false? When the Radial Basis Function is selected as the network model, the user can specify that the network has two hidden layers.

A. True

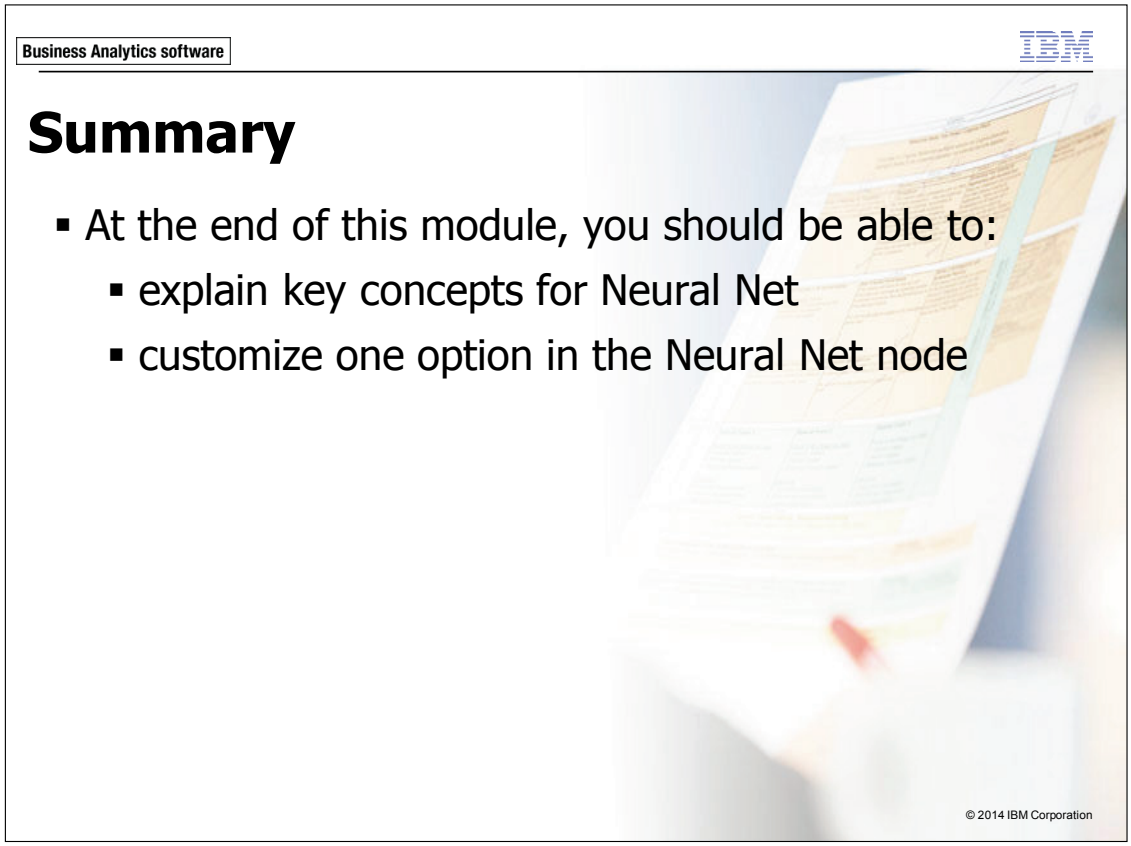
B. False

Answers to questions:

Answer 1: A. True. Weights are adjusted each time an incorrect prediction is made.

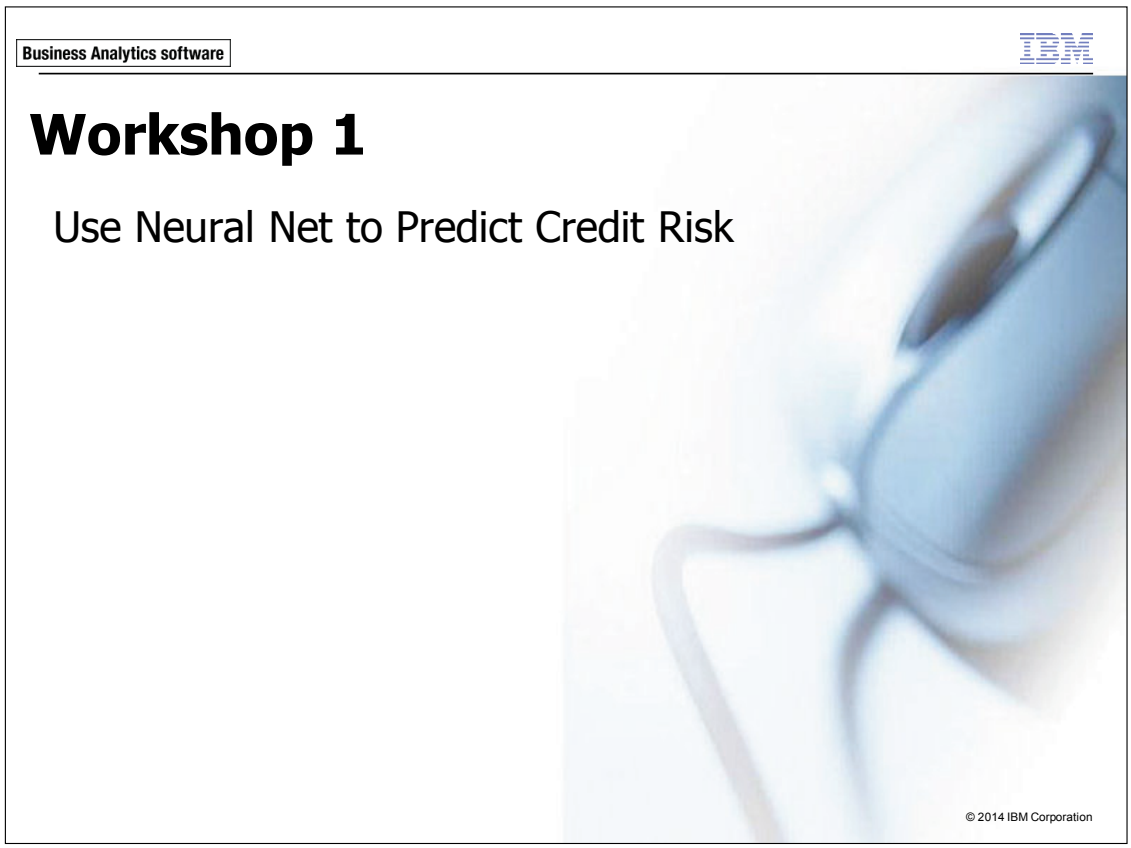
Answer 2: C. Neural Net creates indicator fields for categorical predictors. Neural Net can handle missing values by imputing missing values. Neural Net does not use statistical significance, so categories will not be grouped even if they show the same behavior with respect to the target, and no predictor-selection method is used (as, for example, Logistic does).

Answer 3: B. False. Hidden layers do not apply to the Radial Basis Function model.



In this module, Neural Net was presented as an example of a machine learning model.

This model was presented as a black box model, because even if you know all the values of the coefficients, there is no straightforward link from the predictors to the target. What also complicates interpretation is that a bias is added to each layer of the network. Finally, neurons in the hidden layer and in the output layer also have a so-called activation function which makes that a value is only output if it exceeds a certain threshold. For all these reasons, a Neural Net model does not provide an insight as rule induction tree techniques or traditional statistical techniques do. But apart from the exact computational details, the way that a Neural Net learns mimics the human brain, and as with the human brain, although it is yet unknown how it operates, the predictions can be infallible.



The following (synthetic) file is used in this workshop:

- **bank loan risk.txt**: A text file that represents data from a bank. The file is located in **C:\Train\0A0U5**.

Before you begin with the workshop, ensure that:

- You have started MODELER (click **Cancel** when the splash screen appears when you start MODELER).
- You have set MODELER's working folder. (In MODELER, select **File\Set Directory**. When you are working in an IBM training environment, browse to the folder **C:\Train\0A0U5** and then click **Set** to select it. When you are not working in an IBM training environment, ask your trainer for the folder name.)

Workshop 1: Use Neural Net to Predict Credit Risk

In this workshop you will predict credit risk with a number of Neural Net models. You will compare the fit of Neural Net models, and explain why some Neural Net models pick up the patterns in the data, while others do not.

To do this, you must:

- Use a **Var. File** node (Sources palette) to import data from the comma-separated text file **bank loan risk.txt** and the preview the data.
- Run a **Distribution** node (Graphs palette) on **PAID_BACK_LOAN**.

  What percentage did not pay back the loan?
- Add a **Plot** node (Graphs palette) downstream from the **Var. File** node. Edit the **Plot** node, for **X field** select **AGE**, for **Y field** select **INCOME**, and for **Panel** select **PAID_BACK_LOAN**. Then, run the **Plot** node.

  What ages and incomes are at risk of not paying back the loan?
- Add a **Type** node downstream from the **Var. File** node, edit the **Type** node, instantiate the data, set the **Role** for **AGE and INCOME** to **Input**, and then set the **Role** for **PAID_BACK_LOAN** to **Target** (all other fields have Role None).
- Run the following four Neural Net models by adding a **Neural Net** node downstream from the **Type** node for each model. Also, annotate the models as indicated:

  a) a **Neural Net** model with default settings (annotate the model with **default**)

  b) a **Neural Net** model, using **Bagging** (annotate the model with **bagging**)

  c) a **Neural Net** model, using **Boosting** (annotate the model with **boosting**)

  d) a **Neural Net** model, with **Radial Basis Function** as neural net model, with **100** units in the hidden layer (annotate the model with **RBF**)
- Use an **Analysis** node to evaluate the four models.

  Which are the best two models? Can you explain the results?

Workshop 1: Tasks and Results

Task 1. Import and examine the data.

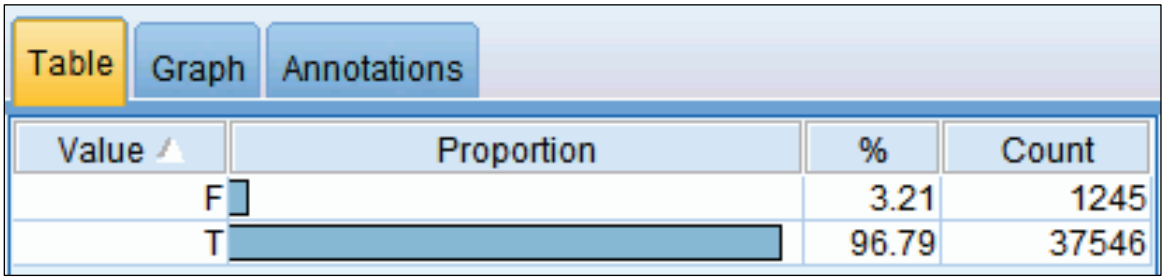
1. Use the **Var. File** node to import data from **bank loan risk.txt** (use default settings for the import).
2. Click the **Preview** button in the **Var. File** dialog box, or add and run a **Table** node downstream from the **Var. File** node to examine the data.

   The preview shows four fields: CUSTOMER_ID, AGE, INCOME, and PAID_BACK_LOAN.
3. Close the **Preview** window, and then close the **Var. File** dialog box.

Task 2. Examine the percentage that did not pay back the loan.

1. From the **Graphs** palette, in the **Field** dropdown, select **PAID_BANK_LOAN**, and then add a **Distribution** node downstream from the **Var. File** node.
2. Edit the **Distribution** node, select **PAID_BACK_LOAN**, and then click **Run**.

   A section of the results appear as follows:



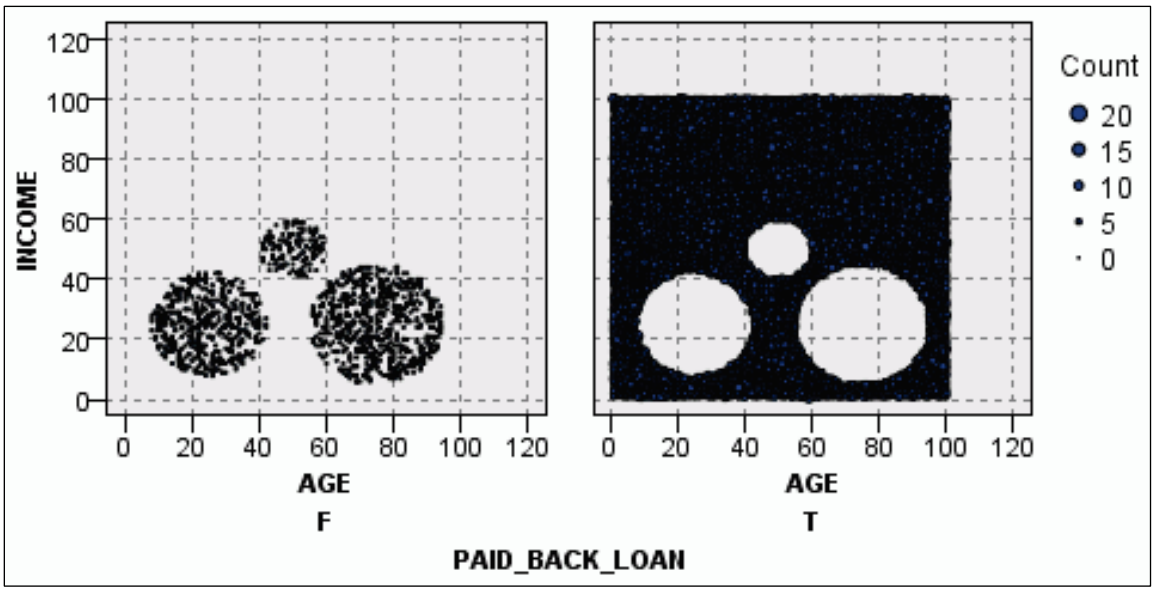
   Of the 38,791 customers, 3.21% did not pay back their loan.
3. Close the **Distribution** output window.

Task 3. Examine the data with a plot.

1. From the **Graphs** palette, add a **Plot** node downstream from the **Var. File** node.

2. Edit the **Plot** node, and then:
   - for **X field** select **AGE**
   - for **Y field** select **INCOME**
   - for **Panel** select **PAID_BACK_LOAN**
   - click **Run**

   A section of the results appear as follows:



   The figure on the left depicts the data for those that did not pay back the loan. Three clusters of records can be distinguished that did not pay back the loan. The biggest cluster centers on age 75 and income 20.

   The figure on the right depicts the data for those that did pay back the loan. There are no persons that paid back the loan for exactly those areas that are comprised of customers that did not pay back the loan.

   All in all, the pattern is clear, but the question is which neural network can detect this pattern.

3. Close the **Plot** output window.

Task 4. Set the roles for predictors and targets.

1. From the **Field Ops** palette, add a **Type** node downstream from the **Var. File** node.

2. Edit the **Type** node, and then:
   - click **Read Values**
   - set the role for **AGE** and **INCOME** to **Input**
   - set the role for **PAID_BACK_LOAN** to **Target**
   - set the role for the other fields to **None**
   - close the **Type** dialog box

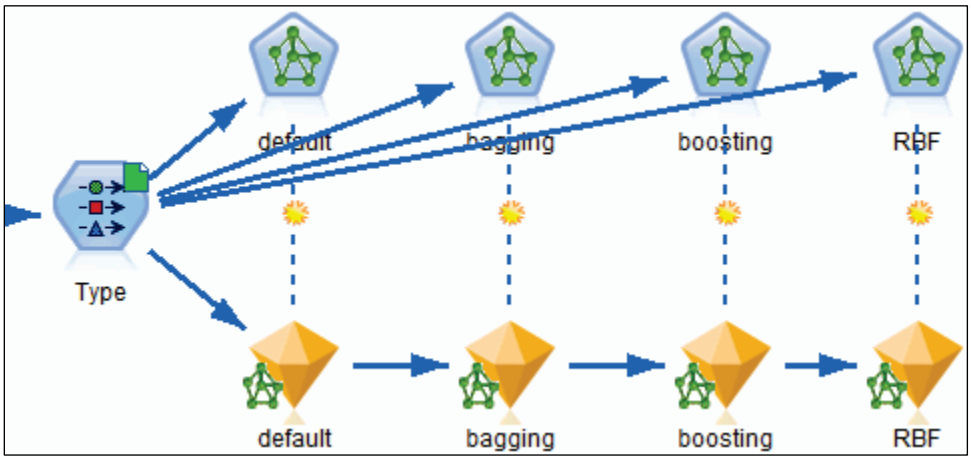
Task 5. Add, configure, and run Neural Net models.

1. From the **Modeling** palette (**Classification** item), add a **Neural Net** node downstream from the **Type** node.
2. Edit the **Neural Net** node, and then:
   - click the **Annotations** tab
   - enable the **Custom** option
   - type **default**
   - close the **Neural Net** dialog box
3. Add a second **Neural Net** node downstream from the **Type** node.
4. Edit the **Neural Net** node, and then:
   - click the **Build Options** tab
   - click the **Objectives** item
   - enable the **Enhance model stability (bagging)** option
   - click the **Annotations** tab
   - enable the **Custom** option
   - type **bagging**
   - close the **Neural Net** dialog box
5. Add a third **Neural Net** node downstream from the **Type** node.

6. Edit the **Neural Net** node, and then:
   - click the **Build Options** tab
   - click the **Objectives** item
   - enable the **Enhance model accuracy (boosting)** option
   - click the **Annotations** tab
   - enable the **Custom** option
   - type **boosting**
   - close the **Neural Net** dialog box
7. Add a fourth **Neural Net** node downstream from the **Type** node.
8. Edit the **Neural Net** node, and then:
   - click the **Build Options** tab
   - click the **Basics** item
   - for **Neural network** model, select **Radial Basis Function (RBF)**
   - enable the **Customize number of units** option
   - in the **Hidden layer 1** text box, type **100**
   - click the **Annotations** tab
   - enable the **Custom** option
   - type **RBF**
   - close the **Neural Net** dialog box
9. Select the four **Neural Net** nodes, right-click one of them, and then click the **Run Selection** button in the main menu.

Task 6. Evaluate the models.

1. Arrange the model nuggets as is depicted in the figure below:



2. From the **Output** palette, add an **Analysis** node downstream from the **model nugget** named **RBF**.
3. Edit the **Analysis** node, and then:
   - enable the **Coincidence matrices (for symbolic targets)** option
   - click **Run**

   The results for the first two models (the default model and the bagging model) appear as follows:

Results for output field PAID_BACK_LOAN

Individual Models

Comparing $N-PAID_BACK_LOAN with PAID_BACK_LOAN

Correct	37,546	96.79%
Wrong	1,245	3.21%
Total	38,791	

Coincidence Matrix for $N-PAID_BACK_LOAN (rows show actuals)

	T
F	1,245
T	37,546

Comparing $N1-PAID_BACK_LOAN with PAID_BACK_LOAN

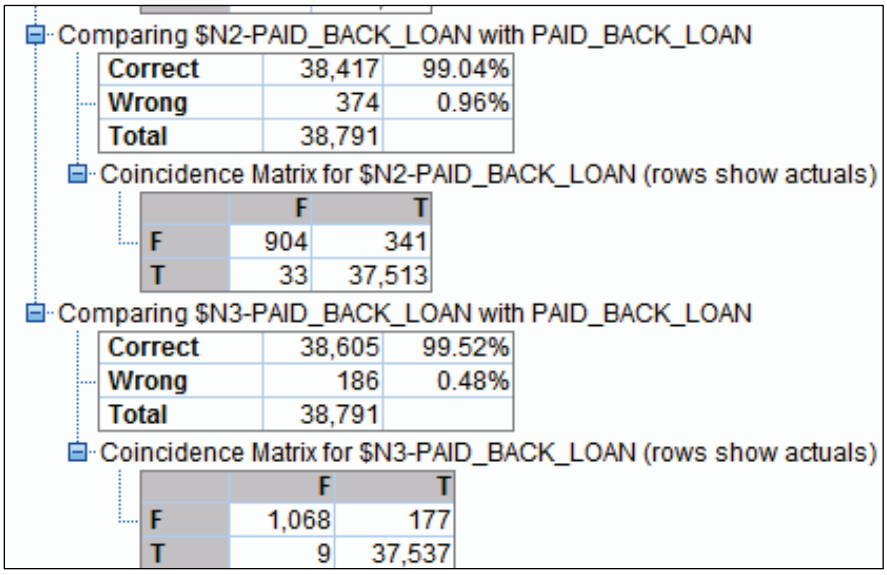
Correct	37,969	97.88%
Wrong	822	2.12%
Total	38,791	

Coincidence Matrix for $N1-PAID_BACK_LOAN (rows show actuals)

	F	T
F	423	822
T	0	37,546

The first model always predicts T. So, the default model did not pick up the pattern. Bagging, the second model, performs better, and identifies 423 F records.

The results for the last two models (the boosting model and the RBF model with 100 neurons) appear as follows:



Comparing $N2-PAID_BACK_LOAN with PAID_BACK_LOAN

Correct	38,417	99.04%
Wrong	374	0.96%
Total	38,791	

Coincidence Matrix for $N2-PAID_BACK_LOAN (rows show actuals)

	F	T
F	904	341
T	33	37,513

Comparing $N3-PAID_BACK_LOAN with PAID_BACK_LOAN

Correct	38,605	99.52%
Wrong	186	0.48%
Total	38,791	

Coincidence Matrix for $N3-PAID_BACK_LOAN (rows show actuals)

	F	T
F	1,068	177
T	9	37,537

Both models identify a large portion of the customers who did not pay back the loan. Boosting worked, because records that were predicted incorrectly weighted more heavily in the next trial.

The Radial Basis Function method worked, because of the 100 neurons that were used. Refer to the plot of AGE and INCOME, with LOAN_PAID_BACK as panel field. The Radial Basis Function scattered 100 neurons over the input space and some of these neurons could identify records that did not pay back the loan.

4. Close the **Analysis** output window.
5. Exit MODELER without saving anything.

Note: The stream **workshop_using_machine_learning_models_completed.str**, located in the **06-Using_Machine_LearningLearning_Models\Solutions** sub folder, provides a solution to the workshop tasks.

This information was developed for products and services offered in the U.S.A.

IBM may not offer the products, services, or features discussed in this document in other countries. Consult your local IBM representative for information on the products and services currently available in your area. Any reference to an IBM product, program, or service is not intended to state or imply that only that IBM product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any IBM intellectual property right may be used instead. However, it is the user's responsibility to evaluate and verify the operation of any non-IBM product, program, or service.

IBM may have patents or pending patent applications covering subject matter described in this document. The furnishing of this document does not grant you any license to these patents. You can send license inquiries, in writing, to:

IBM Director of Licensing
IBM Corporation
North Castle Drive
Armonk, NY 10504-1785
U.S.A.

The following paragraph does not apply to the United Kingdom or any other country where such provisions are inconsistent with local law: INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some states do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you.

This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time without notice.

Any references in this information to non-IBM Web sites are provided for convenience only and do not in any manner serve as an endorsement of those Web sites. The materials at those Web sites are not part of the materials for this IBM product and use of those Web sites is at your own risk.

IBM may use or distribute any of the information you supply in any way it believes appropriate without incurring any obligation to you.

Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

This information contains examples of data and reports used in daily business operations. To illustrate them as completely as possible, the examples include the names of individuals, companies, brands, and products. All of these names are fictitious and any similarity to the names and addresses used by an actual business enterprise is entirely coincidental.

If you are viewing this information softcopy, the photographs and color illustrations may not appear.