# Chapter 7: Introduction to decision trees

## Introduction

This lesson will include an introduction prior to speaking about decision tree algorithms and their common ground.

The mechanism by which decision trees work will be introduced before speaking about Altair Analytics Workbench decision tree capabilities prior to a demonstration followed by a summary.

Figure 1: Contents

A screenshot of a computer

Description automatically generated

Decision Trees are a popular technique that represent results visually a tree diagram. The dependent variable can be either categorical or continuous and Independent variables can be of any type.

Figure 2: Introduction

A diagram of a child

Description automatically generated

Decision Trees can be used for modelling, exploring an unfamiliar dataset and complimenting other

techniques to provide some insight and clarity on their operation.

Their popularity is not only due to the visual representation but also as a result of being able to supply independent variables of any types and volume. There are no assumptions associated with model building and output is easy to interpret.

Decision Trees operate by successively splitting data into subgroups to better identify and distinguish membership of the dependent variable. There will be Segments that have a high concentration of one category or the other and these lead to being able to create rules to better predict category membership.

Segments in the tree diagram are referred to as nodes. The root node is the entire dataset with the dependent variable distribution. Nodes can be parents or child and terminal nodes are nodes that do not undergo further spitting and the point at which a prediction takes places. Branches are the splits from one variable and subtrees are a subset of the overall tree.

## Decision tree algorithms

Decision Tree modelling evolved from cross-tabulations into techniques such as CHAID, CaRT and C4.5.

More variants of decision tree algorithms exist and all have varying operating mechanisms, however

they are all designed for the same purpose and common ground exists regardless of the algorithm

selected in that they all segment data by applying algorithms to determine splits and have options

to specify the tree depth along with other stopping rules or pruning.

Figure 3: Decision tree algorithms

A screenshot of a computer

Description automatically generated

## Decision tree mechanism

Decision trees, regardless of algorithm follow the same steps of categorising variables, them minimising

the number of categories, selecting the best predictor as a split and repeating the process until a stopping rule is encountered.

Figure 4: Decision tree mechanism

A diagram of a structure

Description automatically generated

As Decision trees uses cross tabulations in some form or other, all variables must be categorical. Continuous variables are converted to categorical equivalents and categorical variables are retained as

they are.

Many binning methods can be applied and varying bins created. For illustrative purposes, here, as can

be seen the variables *income* and *age* are continuous. These are converted to categorical equivalents by using 10 equal height bins as a starting point.

The variables *martial status* and *sex* are retained as they are as they are already categorical. The next step is to minimize the number of categories. This is achieved by using the dependent variable and pairs of bins.

Figure 5: Minimise

A screenshot of a computer

Description automatically generated

For example, to reduce the number of bins in the variable *age*, the first two bins are cross-tabulated with the dependent variable and their distributions compared. If they are determined to be similar, they

are merged.

The process continues by cross-tabulating the dependent variable with the combination of the first two bins and the third bin. Again, if they are determined to be similar regarding their dependent variable distribution then they are merged, otherwise, they are left separate. This process continues until a minimal set of bins has been arrived at.

Note that the minimization process for nominal variables is different compared to continuous variables in that, for nominal variables, for example *marital status*, as there is no order to the categories, any categories can be merge.

For continuous or ordinal variables, when converted to categorical, are ordered, for example, age bins might be less than or equal to 30, 31 to 41 and 42+, to main the order, bins that are non-contiguous can only be merged if the intervening bin can also be grouped with them.

At this point, the best splitter variable is selected and its bins are used as the first split in the tree. For

Illustrative purposes, *age* with three splits is selected.

Figure 6: Select the best predictor

A screenshot of a graph

Description automatically generated

At this point the entire process begins again with the left most split - this can be seen as a separate dataset. It has a root node and now must assess independent variables: categorising, minimising and selecting the best split.

This continues until a stopping rule has been met, which can be that no independent variables meet the criteria for inclusion, another way to say this is that no potential predictor meets the algorithm entry requirement whether that be by way of significance or a threshold value being reached.

Figure 7: Stopping rule

A diagram of a company

Description automatically generated

A second criteria is tree depth, this can be preset, for example, 5. If the tree reaches a depth of 5 levels, it will grow no further.

Bins size to split means that there must be at least a minimum number or percentage of observations in a node for It to be split further.

Bins size to create means that there must be at least a minimum number or percentage of observations in a node for It to be created

All of these options and more can be set via the Decision Tree block from the Model Training group and three decision tree methods are available: CART, c4.5 and BRT.

Figure 8: Methods available in ALTAIR

A screenshot of a computer

Description automatically generated

CART and C4.5 are well known industry decision tree algorithms. CART can handle any categorical or continuous Dependent variable and uses a *Gini* statistic for tree growth. C4.5 allows categorical dependent variables only and uses entropy variance for tree growth. Options for both are limited but do Include tree depth and minimum node size.

BRT - Binary Response Tree is ALTAIR implementation of a decision tree with options to choose from one of four growing algorithms: *Chi-square*, *Entropy Variance*, *Gini Variance* and *Information Value*.

Options are available to control tree depth, and minimum node creation size as well minimum size to split and a host of additional options that make it the go to choice when building decision trees.

Regardless of the method chosen, all avail of auto-grow and interactive growing methods to either automatically Select variable and find splits or manually select variables and split points.

## Demonstration

So let’s get onto a demonstration.

This demonstration uses the project created in a previous lesson and adds a new Workflow named *Decision\_Tree*. The data used in this lesson are contained in the file *Risk\_Def.csv*, this file is contained in the eLearning data folder and is dragged onto the Workflow canvas from the File Explorer view.

Figure 9: Risk\_def.csv

A screenshot of a computer

Description automatically generated

Opening the file with the data profiler it can be seen there are 20000 observations and 13 variables.

Viewing the data it can be seen there is an *ID* variable, some demographic and financial variables aswell as a variable named *DV*. This will be the focus of the model and used as the dependent variable.

Viewing its distribution from univariate charts it can be seen that it contains two values: *bad* and *good*, a

breakdown of approximately *24*% and *76%* respectively.

Figure 10: Chart of DV

A screenshot of a computer

Description automatically generated

The data come from a financial institution and the dependent variable records whether an account has been deemed to be good / bad. There is interest in building a model to identify bads.

Prior to getting on to any modelling it is a good idea to partition the data first. Here the partition block is dragged from the data preparation group to the *risk\_def* dataset and docked. Notice that two datasets are created immediately, this is due to the block being included in *Auto Run*.

Opening its configuration dialog, two partitions are created by default with 50% of observations randomly assigned to each.

More partitions can be added by clicking the add partition button, here, partition names are changed to development and testing with a 70:30 split and the random seed of 1000 is also used. The model will be created with the development partition and tested on the testing partition.

Clicking OK creates the partitions. Opening both with the Data Profiler it can be seen the development partition contains 14000 observations and the testing partition contains 6000 observations.

From the model training group, a Decision Tree block is dragged onto the *development* partition, docked and opened.

Figure 11: Adding and opening a Decision tree block

A screenshot of a computer

Description automatically generated

The first time the block is opened the preferences dialog pop ups automatically. There are three dialog pages: Variable selection, Growth Preferences and Binning Preferences.

From Variable selection, model variables are set. The dependent variable is selected as *DV*, the tree type automatically changes to classification. Altair Analytics Workbench decision trees can be applied to a dependent variable that is either categorical or continuous, if a continuous dependent variable is selected, the tree type would change to regression.

From the Target Category drop-down *bad* is selected. The model results will be identical regardless of the target category selected, the selection simply impacts the way tree results are presented.

A weight variable can be included, this can be any continuous dataset variable, and one is not included here. All variables bar *ID* are moved from the Unselected Independent Variables list to the Selected Independent Variables list.

All variables appear with an Entropy Variance statistic to guide variable inclusion. Notice the treatment dropdown. Variable type can be set as Nominal, Ordinal or Interval for numeric variables and nominal or ordinal for string variables.

Here, the variable *education\_num* is changed from nominal to interval as it contains the number of years of education complete.

Figure 12: Variable Selection

A screenshot of a computer

Description automatically generated

It is wise to ensure that variable type is set correctly and this facility provides the ability to adjust if necessary.

From Growth Preferences the algorithm can be chosen. Clicking the Growth algorithm drop-down it can be seen there are three to choose from C4.5, CART and BRT.

C4.5 provides options to set the maximum tree depth and minimum node size as well as providing options to merge categories and exclude missing values.

Pruning removes terminal nodes and subtrees that do not significantly improve predictive accuracy.

CART, similarly, has options to set the maximum tree depth, the minimum node size, exclude missings and apply pruning. It also provides a dropdown to select the citerion to determine splits, the default being Gini but Twoing is also available.

Options for c4.5 and cart are limited, selecting BRT - Binary regression tree, ALTAIR decision implementation, provides a range of options that make it the clear choice when building a decision tree.

Figure 13: BRT

A screenshot of a computer

Description automatically generated

For the most part, options available via C4.5 and CART are also present here, bar pruning methods.

Notice the criterion dropdown provides four options: Chi-squared, Entropy Variance, Gini Variance and Information Value, with the default being Information Value. The maximum depth can be set, if not selected, there is no maximum depth.

The options: Select minimum node size by ratio, Minimum node size (%) and Minimum node size relate

To the minimum node creation size, this can be set as a % of the overall dataset size or as a value.

Tree nodes will only be created if they reach a default threshold size, here .5% of the overall dataset size – with the dataset here contains 14000 observations, 700, a half a percent, is the minimum size a node must be, to be created.

The next three options are similar but relate to the size a tree node must be to be considered to be split further, currently this is set at 2% but can be changed.

Options selected by default and not present in C4.5 or CART are:

* Allow same variable split
* Open left and open right

The same variable split means that a variable may be split further, for example if a split was created with the variable and two nodes created *age*: <=30 and greater than 30, selecting this option means that either category may undergo further splitting, For example, the greater than 30 category might be split in two categories: 30 - 41 and 42+ in a subsequent split.

Open left and Open right ensure that values below a dataset minimum or above a dataset maximum for any variable will be accommodated. For example, If a model uses the variable *age* and its range is 20 - 89. Lets say the first bin for this variable is 20 - 30. This excludes all values below 20, using open left, the bin range changes to <=30, this is a useful option to include and here, selected by default.

Missing values are included by default in the tree but will always appear as a separate category, they can be excluded by selecting Exclude missings or can be forced to be a valid category by selecting merge missing bin and will therefore be considered for merging with other bins.

Monotonicity can be forced using weight of evidence and the number of bins can be set including the maximum number of optimal bins. Other options relate to the criterion chosen and thresholds for selecting and splitting variables.

At this point all options are accepted at their defaults. Bear in mind that *Help* is available from within the dialog by clicking the help icon. Selecting *Decision Tree Preferences* opens the help pages for all growth preference options.

Figure 14: Decision tree help

A screenshot of a computer

Description automatically generated

The third page relates to binning. ALTAIR decision trees provides automatic and interactive growing methods with the ability to force variables into a model. Given this, the selected algorithm can be used to choose variables and decide splits or, variables can be forced into the model and splits applied by way of ALTAIR binning.

Figure 15: Binning Preferences

A screenshot of a computer

Description automatically generated

This page contains two areas: Default bin count and Winsorrate and Optimal Binning. The Default bin count and Winsorrate relate to Equal Height, Equal Width and Winsorized binning, and the Optimal Binning area relates to optimal binning settings.

Setting can be changed, for example the default bin count can be increased to give greater granularity across splits. This page will be revisited when forcing variables into the model.

## Summary

This lesson included an introduction prior to speaking about decision tree algorithms and their common ground.

The mechanism by which decision trees operate was introduced before speaking about Altair Analytics Workbench decision tree capabilities which was followed by a demonstration