Weka Analyzer Visualizations Tutorial

# Summary

This tutorial will give you a hands on overview of using three custom built Weka Visualizations in Analyzer. You'll learn about three common data mining algorithms including classification, clustering and attribute selection. We'll also talk about how to use the models you build to score new customers. Finally, for developers, we'll go into the details on how the visualizations were built using the Visualization Framework and D3.

This tutorial was based on the Pentaho World 2015 presentation "Extending the Pentaho Platform with a Decision Tree Visualization" presented by Benny Chow, Pedro Vale and Steve Szabo.

# Business Background

Data mining algorithms can be difficult to use for business users and so the goal of the Weka Analyzer visualizations is to leverage the easy to use interface of Analyzer, coupled with the advanced algorithms in Weka to enable a business user to quickly model, explore and run data mining models on their own data.

In this tutorial, we will work with a marketing manager Ben as he tries to design the next marketing campaign at a big bank. He is going to analyze recent telemarketing campaign response data to see what types of customers actually ended up purchasing a product. He will combine his understanding of the business with three different data mining algorithms to build a new model to predict what types of new customers are likely to purchase. He will then use his new model to score new customers and target those likely to purchase in the next marketing campaign.

His data set includes the following important attributes about his customers:

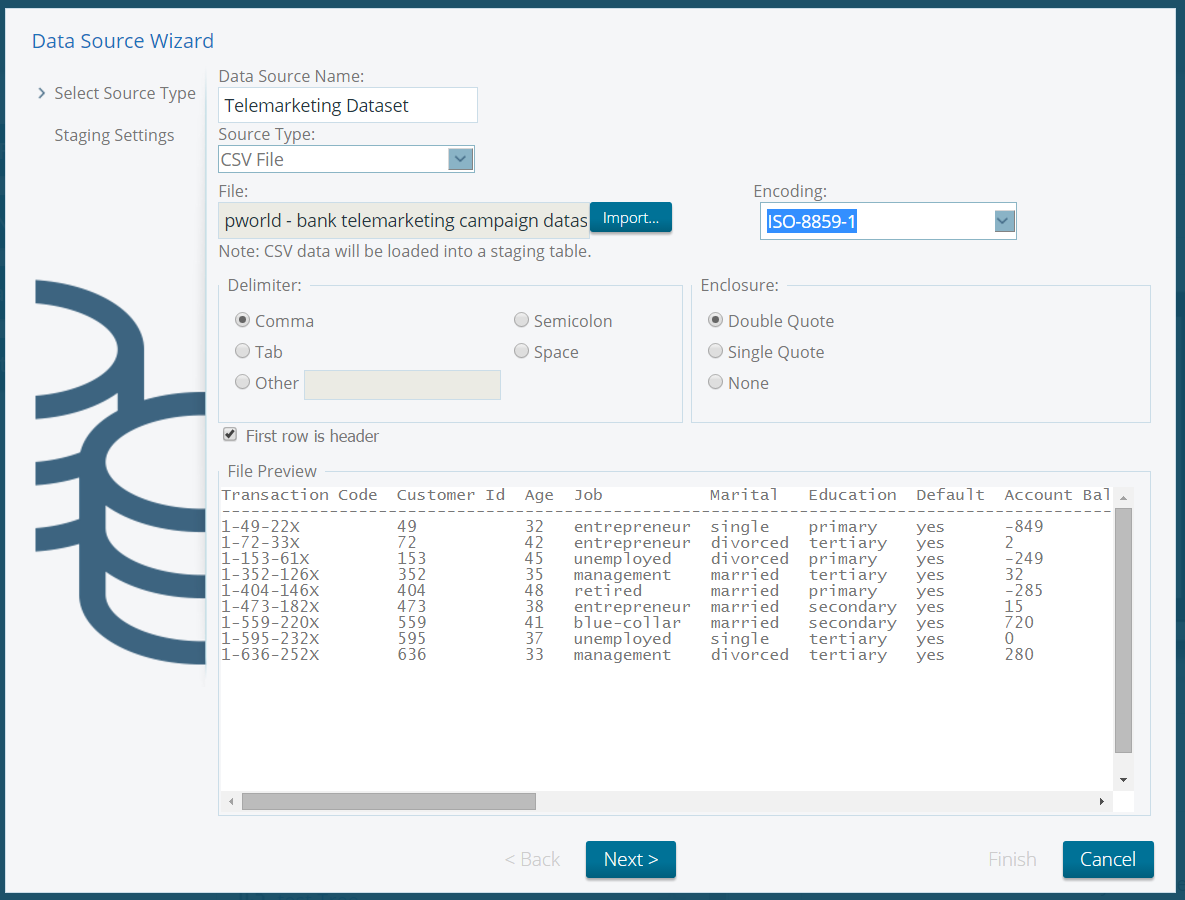
* Customer Age
* Job
* Marital Status
* Education Level
* Account Balance - Negative if they owe the bank money
* Housing - Whether they own a house
* Loan - Whether they have a loan with the bank
* Contact - How the user was contacted by the telemarketer
* Day/Month/Date - Date time of contact
* Call Duration - How long the call lasted
* Time Between Calls - Time between the current call and last call
* # of Calls
* Campaign Success - Outcome of the call. Interest level of customer in product
* Purchased Product - Whether the customer ultimately purchased the product

# Importing Data and Modeling

There are many ways to get data into Pentaho and the quick and dirty way is to upload a CSV file using the data source wizard. This wizard will inspect the file and create a single DB table with column types matching the CSV headers.

You can download the telemarketing dataset here:

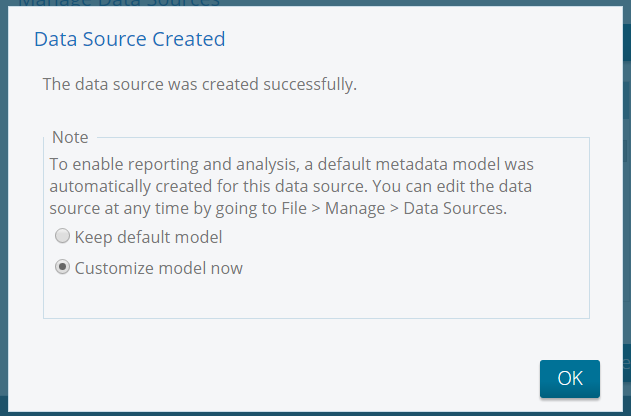
Fortunately, Ben has his dataset on his desktop and he loads it into Pentaho as shown below:



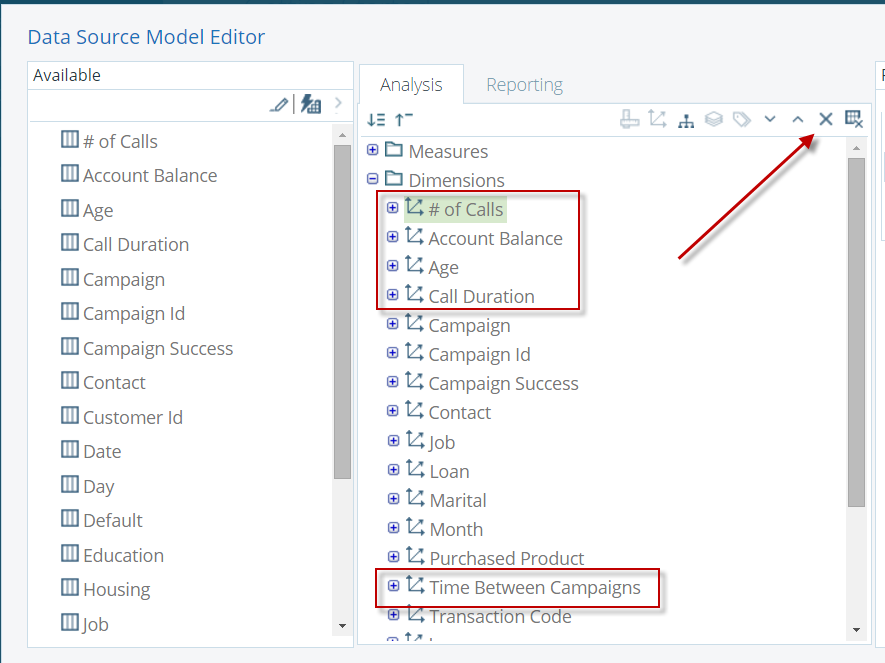
Take all defaults and click Next and then Finish. We'll explore this dataset later in Analyzer.

Datasource Wizard will expose every numeric column as both a single level hierarchy and a measure. For data mining purposes, it's better to leave numeric data as numeric and not turn them into strings. This allows the algorithms to find arbitrary split points such as Income between 21,588 and 53,328. So, let's clean this up by removing any numeric columns that have been added as levels.

Select Customize Model as shown below.



Select the following Dimensions and delete them from the model:



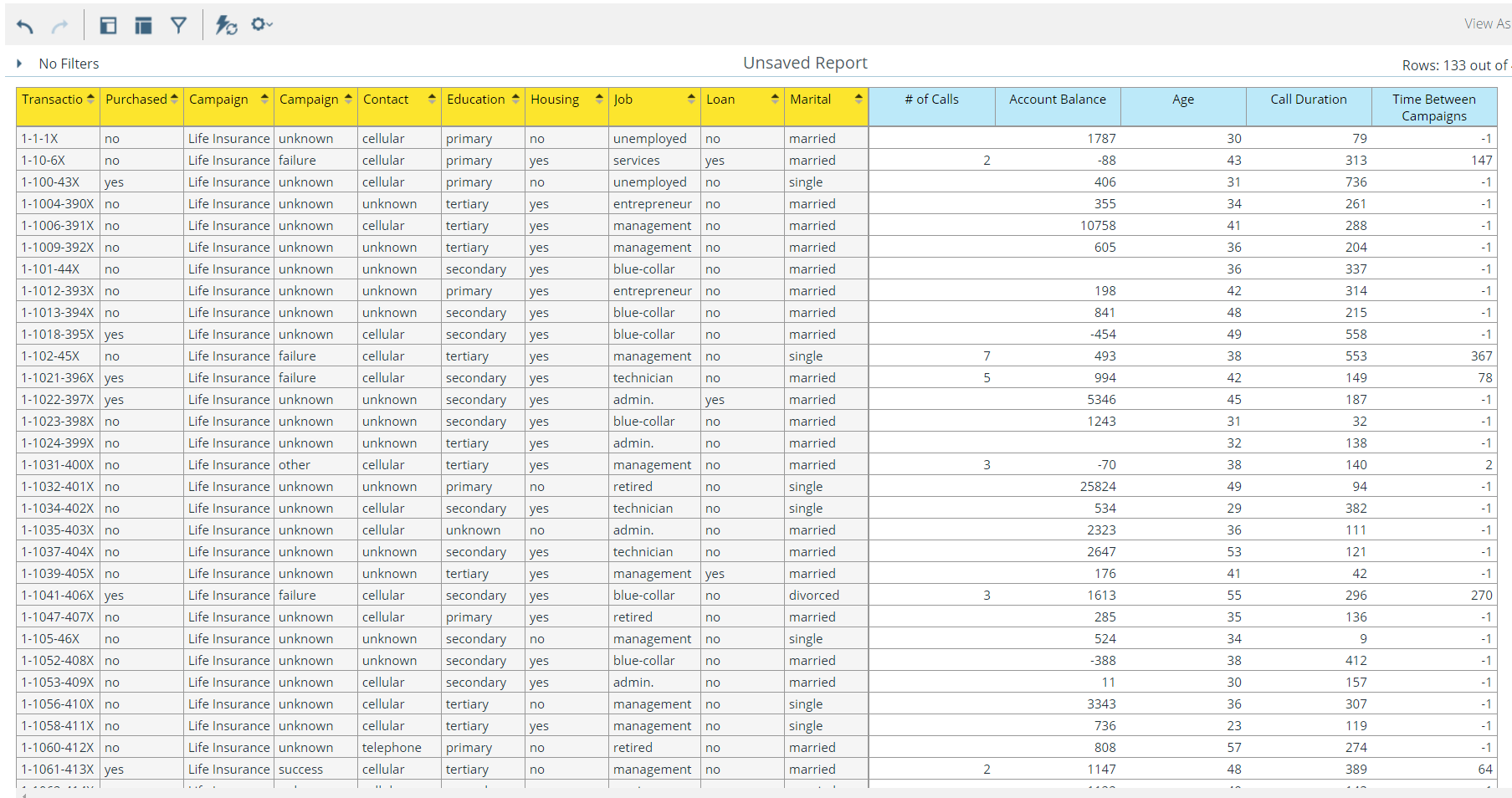
Now open your new data source in Analyzer...

# Classification: Decision Tree

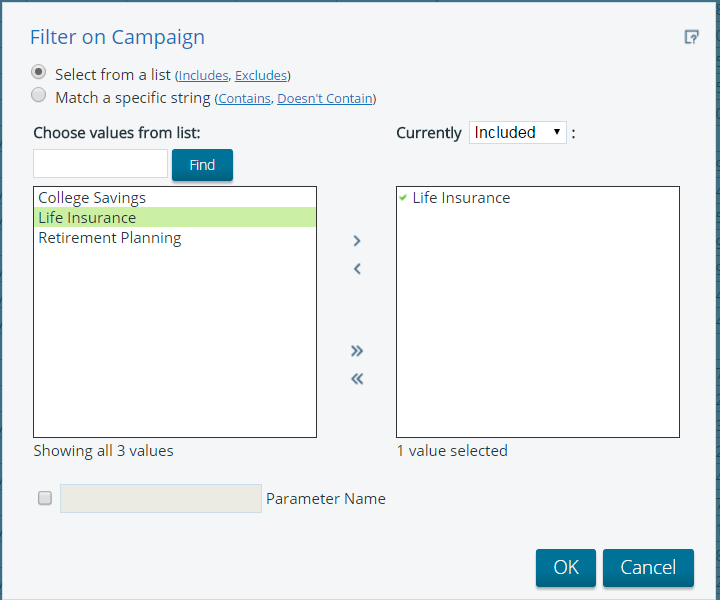
Decision Trees are a type of classification model that can predict categorical class labels (ex. "Purchased Product" class with labels "Yes" and "No") based on numeric or categorical inputs. They result in a white box model which is easy to understand and interpret because the branch rules make business sense.

The first thing Ben will need to do is to spot check his response data so he drags out all the fields that could go into the his model:

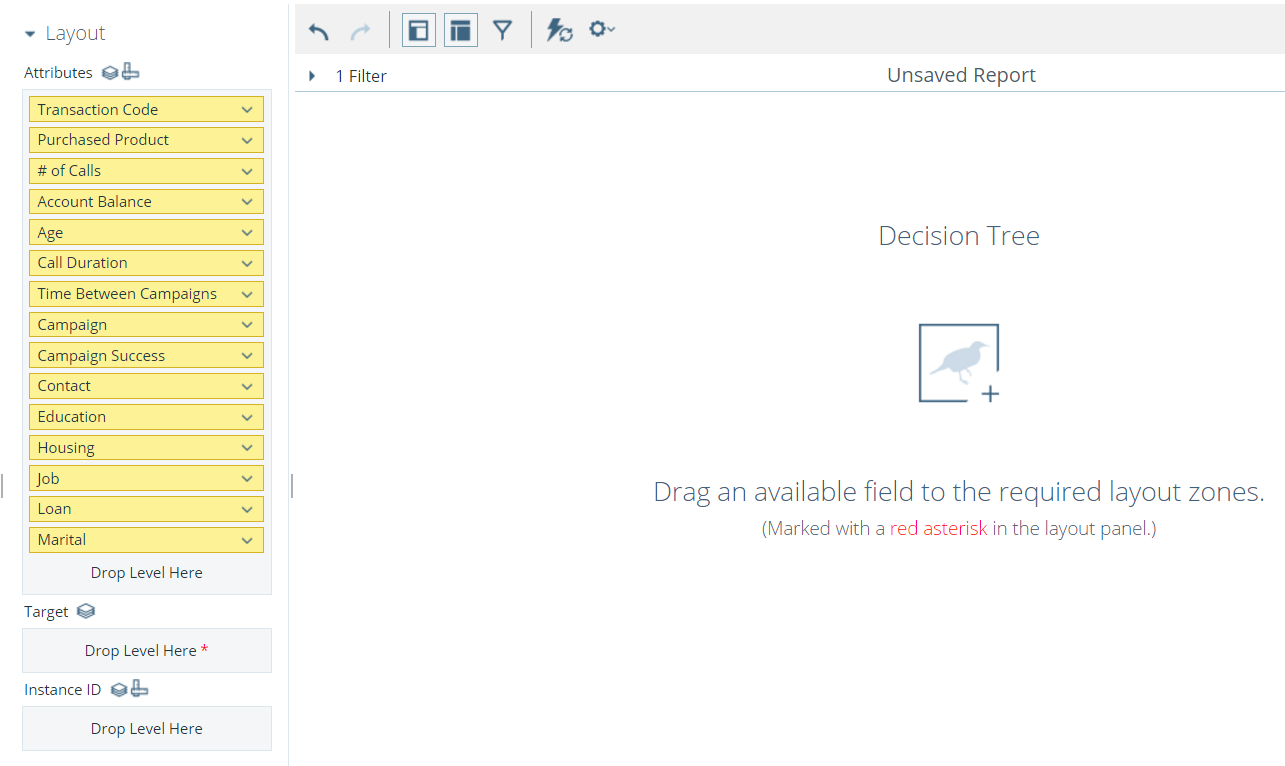
* Transaction Code - This unique identifies each customer response and outcome
* Purchased Product - Whether the customer purchased or not. Yes/No.
* Campaign
* Contact
* Education
* Housing
* Job
* Loan
* Marital
* # of Calls
* Account Balance
* Age
* Call Duration
* Time Between Campaigns



Yep... the data looks right. However, Ben remembers he needs to filter on just his campaigns so he creates a new filter on Campaign = Life Insurance which is easy to do in Analyzer:



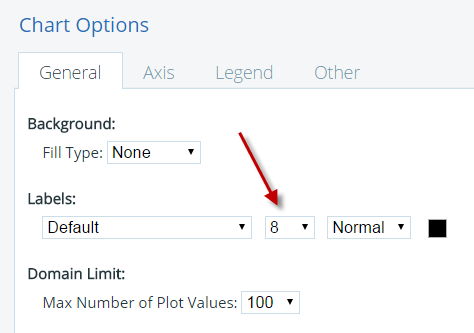
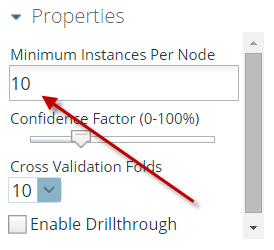
Now Ben opens the View As chart dropdown menu and selects Decision Tree as shown below:



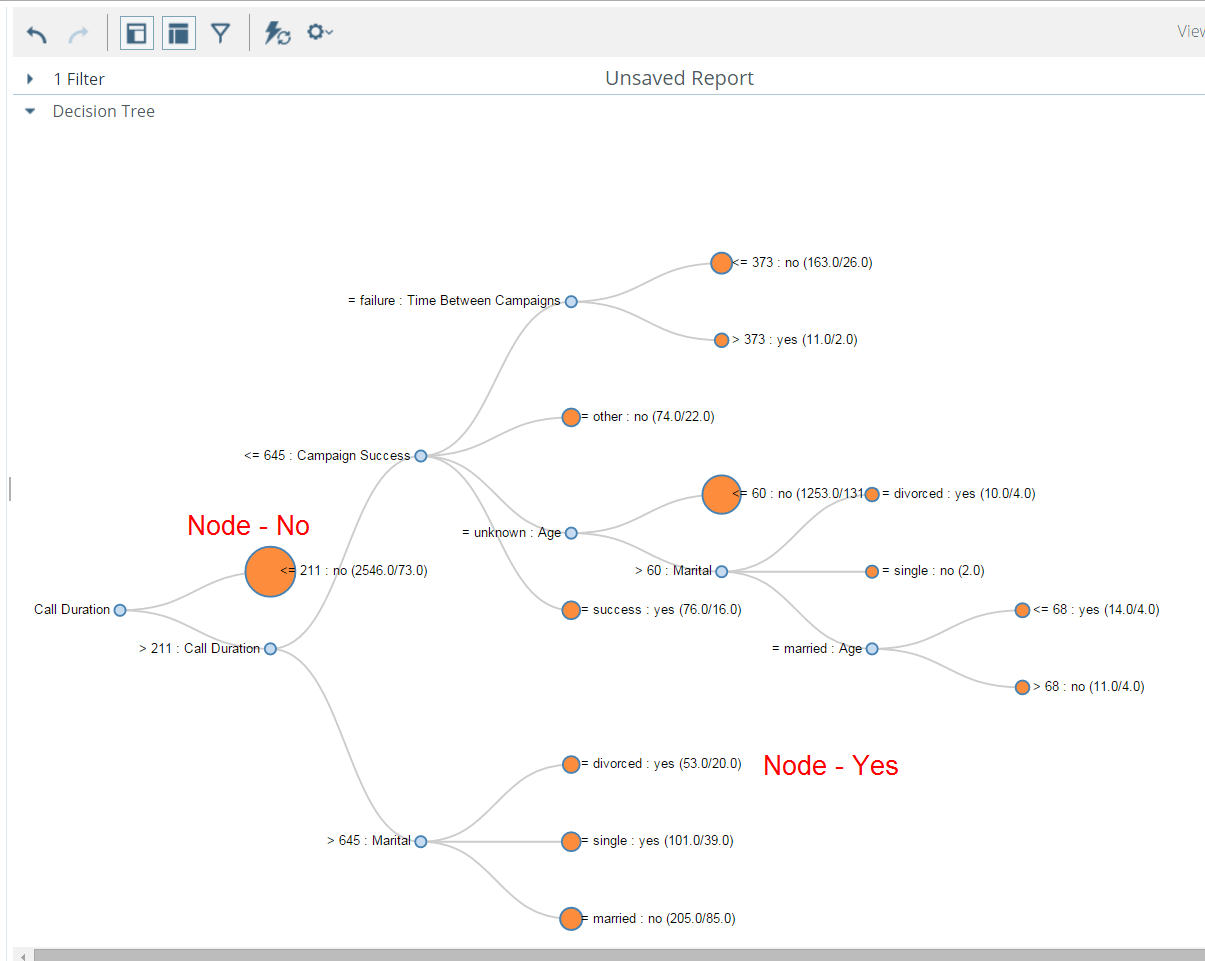
The layout panel for all of the Weka visualizations is the same and contains the following layout zones:

* Attributes - These are the independent variables that will go into the model and be typically used to predict the target.
* Target - This is the dependent variable which the model is predicting or testing against.
* Instance ID - This attribute is used to uniquely identify instances but will not be passed into the model. For example, Transaction Code is needed to uniquely identify each customer outcome but we don't need this code in the model. If we didn't include Transaction Code in Instance ID, then the instances would be aggregated up to the Attributes and Target fields.

Because Ben is trying predict which customers purchased a product, so he drags Purchased Product to Target and drags Transaction Code to Instance ID. In order to make the results easier to understand, Ben sets the Minimum Instances per Node to 10 and Chart Options Label size to 8.

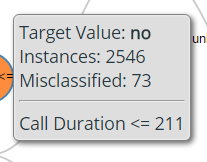


This results in the below decision tree:



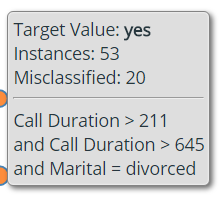
Let's take a look at what's happening here...

The "Node - No" is predicting that the customers in that node will NOT purchase a product:



On mouse over, Ben can see that if the telemarketing call duration was less than 211 seconds, then the predicted outcome was no product purchased. In his dataset, 2546 of the 4519 records had call duration <= 211 seconds. Of those 2546, only 73 were misclassified as not purchasing a product. Those 73 actually did end up purchasing and thus are false negative. From a marketing point of view, it is obvious that if you can't keep the customer on the phone, then you won't be able to generate interest in the product.

Now, Ben takes a look at the "Node - Yes".

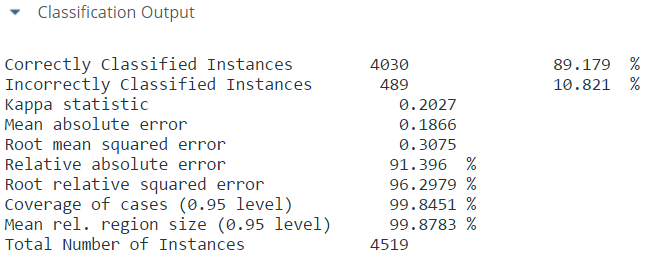


This node is predicting that these customers will purchase a product. It says that if Call Duration > 211 AND Call Duration > 645 seconds and Marital Status = Divorced than that customer will likely purchase. While 20 false negatives out of 53 total may seem inaccurate, if a typical campaign response rate is less than 1%, this node's rules coupled with other nodes that predict "Yes" may still improve the campaign response rate dramatically targeting the best customers.

At this point, Ben can utilize his domain knowledge and try out different model inputs. For example, he may think that "Call Duration" is too correlated with purchasing a product because a customer willing to talk to you inherently means that he is interested. In that case, Ben can drop Call Duration from the model and predict on other model inputs.

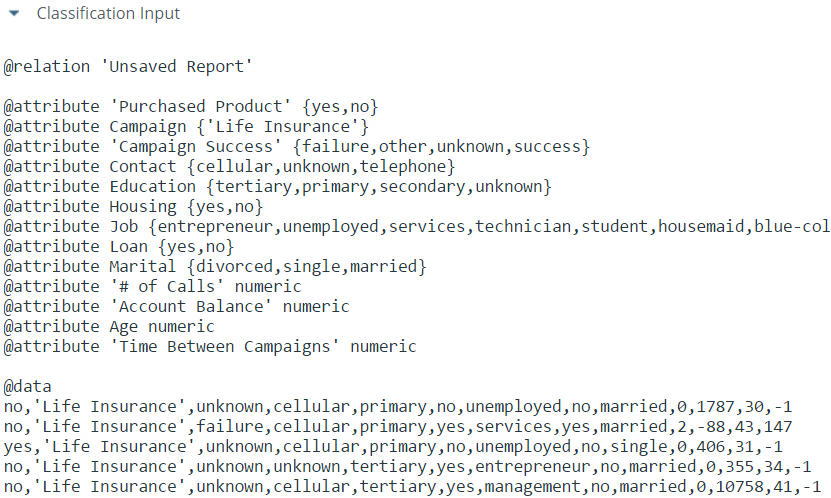
The decision tree visualization includes dropdown panels for various model outputs described below:

## Classification Output



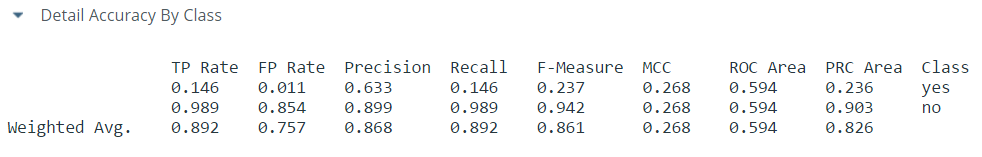
This measures the predictive performance of the classification model by running the model N fold times and averaging the results. Each fold consists of a 90-10 ratio of training data to test data. In Ben's model, 89.179% of the test data was correctly classified averaged across 10 different runs.

## Classification Input

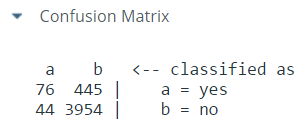


This is the raw instance data is fed into the Weka data mining algorithm.

## Detailed Accuracy by Class

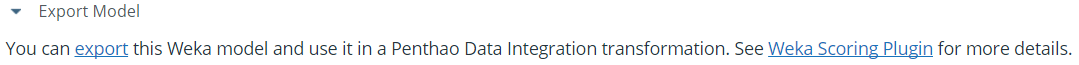


## Confusion Matrix



This matrix tells us that there were a total of 76+445=521 real Yes outcomes of which 76 were correctly predicted/classified as Yes (true positive) and 445 incorrectly classified as No (false negative).

## Export Model



This link allows Ben to export the model and use it to score new customers in a PDI transformation. More on this later...

# Clustering

# Attribute Selection

# Scoring in PDI

# Building a Custom Visualization

## Visualization Framework

## Visualization Metadata

## Code Details

### plugin.xml

### weka.js

### weka\_analyzer.js