SAS® GLOBAL FORUM 2020 #SASGF

Paper 4701-2020

Using Analytics to Predict Tax Recovery and Prioritized Audits and Investigations in Canada

With: SAS® Enterprise Miner™ and Enterprise Guide™

Jason A. Oliver, Senior Compliance Analyst, Canada Revenue Agency (CRA)

ABSTRACT

The Canada Revenue Agency (CRA) has made tremendous inroads in the last two years by leveraging the power of predictive analytics, notably by using web domain and E-Commerce data for corporate taxpayers. This session leverages the capabilities of SAS® Enterprise Guide® and SAS® Enterprise Miner™ in unearthing predictive patterns of interest with the clear objective of strengthening a feedback loop between tax risk assessment and the corresponding accrual of tax via audit. We examine powerful data learning techniques, as they apply to tax-based analytics, such as neural networks, decision trees, and regression analysis.

INTRODUCTION

This SAS® paper is intended to inform the reader of the methods and results that have come about as a result of the application of experimental predictive analytics for tax risk and recovery at the Canada Revenue Agency (CRA). This could conceivably be leveraged to other tax jurisdictions, or potentially in an even more abstract manner to certain sectors (including but not limited to insurance or banking). It is not intended to explain the intricacies of the CRA organizational structure (giving just a cursory overview of high-level functions), but rather to focus on the application of data science methods to our business need and show how SAS® has been a game-changer. We shall explore regression, neural networks, and decision trees, and what the logical next steps ought to be given our observations.

THE IMPETUS FOR PREDICTIVE MODELING AT THE CRA

The Canada Revenue Agency is Canada's national tax administration, which is responsible for the administration and enforcement of the ITA (Income Tax Act) and ETA (Excise Tax Act). Our compliance programs cover domestic and international matters, in accordance with tax treaties. Within our Compliance Programs Branch (CPB), I work with the **RITS** or *Risk Identification Tools Section*, under the *Business Intelligence and Data Division (BIDD)*. I am entrusted with **the RITS Analytics Lab.**

ON TRADITIONAL RISK ANALYSIS AT THE CRA

The **NRAS**, or *National Risk Assessment System,* includes the IRAS (International Risk Assessment System), which is entrusted with creating risk issue algorithms employed in support of the ITA/ETA. We have literally hundreds of algorithms that cover a wide range of scenarios, with some overlap in their logical construct. For the most part, these have proven valuable in assessing taxpayer risk – and thus referral to audit – but such traditional methods are not the be-all and end-all, which is where SAS® has been a valued catalyst.

The NRAS/IRAS algorithms are more traditional and heuristic audit and accounting formulas, such as benchmarking/ratios, and rule-based scenarios. Some of these might include:

- Low net income assessed relative to neighbourhood (postal code) or industry sector
- Salary & wage expenses high
- Gifts & charitable donations questionable
- Irregular property matters

It is important to keep in mind when using these scenarios, that “they must be compared to their own brethren”. In other words, one would not compare the filings of an oil and gas company to those of a bakery. When it comes to the *non-traditional* side of the equation, that is predictive analytics, we *may* make use of the NAICS (North American Industry Code Standard) class variable; but the more overarching concern is the “fourth dimension” (i.e. *time*). That is, we can only legitimately use predictive inputs (explanatory variables) where they logically precede the target (outcome) variable. I don’t know about you, but I don’t intend to be a pioneer in the hitherto undiscovered field of “quantum analytics”.

ON SUITABILITY OF PREDICTOR CANDIDATES

We can see from the staircase diagram below that if we wish to predict, for example, on a given range or threshold of TEBA (tax earned by audit) as our outcome variable, we can use virtually all of the variables higher up the “stairs”. But if we wish to predict on the Selection Reason Code (SRC) of a screened audit case – or the Priority Code – we can’t use any of the effects on the lower stairs as inputs.

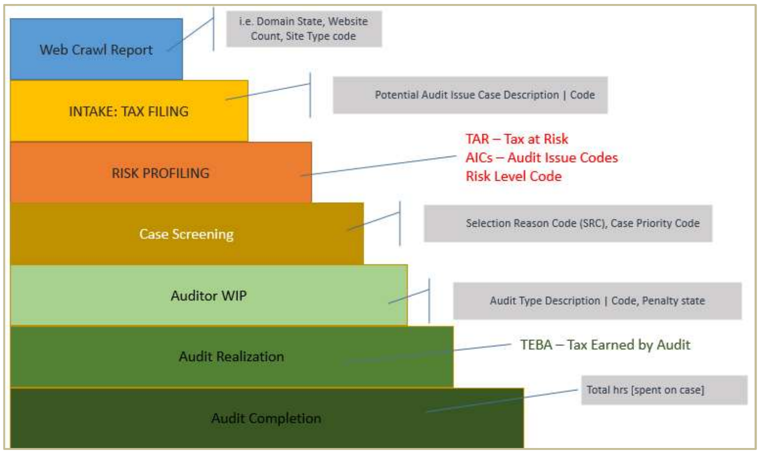

Web Crawl Report
i.e. Domain State, Website Count, Site Type code
INTAKE: TAX FILING
Potential Audit Issue Case Description | Code
RISK PROFILING
TAR – Tax at Risk
AICs – Audit Issue Codes
Risk Level Code
Case Screening
Selection Reason Code (SRC), Case Priority Code
Auditor WIP
Audit Type Description | Code, Penalty state
Audit Realization
TEBA – Tax Earned by Audit
Audit Completion
Total hrs [spent on case]

Figure 1. “Staircase” diagram of tax filing lifecycle stages & key inputs

One must also take note that the first step entailing web reconnaissance (“web crawl”) is more parallel to the remaining steps, as this is something done by our RITS analytics lab.

Its purpose is twofold:

1. To determine websites in an area of interest and verify owners (from WHOIS lookup) against the taxpayer database to determine if they are filing with the CRA;
2. For those where taxpayers are identified from our database, to derive predictive inputs such as domain state, count of website possessions, and site type.

As an aside for providing assurance, this process has been fully authorized by a PIA (Privacy Impact Assessment) and is used by other tax jurisdictions in Europe. But thinking more in our analytics context, this helps enrich our predictive capabilities beyond what the traditional algorithms on benchmarks and ratios could accomplish. For instance, we have a class variable called **DTC**, which is the ***Domain Turnover Code***. This is measured based on whether a site is domestic or foreign-hosted, whether it is masked or unmasked, and/or whether it has been sold (discontinued).

It is ultimately our hope that, through iterative experimentation and fine-tuning, we can use the power of SAS® solutions to enhance the workload prioritization system currently in place. Which brings us to our next consideration.

ON THE CURRENT LACK OF FEEDBACK LOOP

As you have seen from the staircase diagram, the flow of inputs just cascades downwards. In a utopian world, it might flow upwards too – but just like Newton's apple having an infinitesimally small chance of going upwards according to quantum physics, we could be waiting a very long time unless we intervene with a divine force (there I go with my quantum analogies again!) in the form of enhanced predictive modeling.

And therein lies the predicament – there is currently a lack of feedback from the audit functions back up to the risk profiling function. Oftentimes, an auditor will adjust the priority factors in a given case, which would otherwise obviate any risk-rated factors upstream. But such back-feeding of priority updates never takes place; consequently, we cannot use the risk level or tax-at-risk (TAR) from the profiling stage as inputs in linear models for predicting audit outcomes. (More on this later with images and output.)

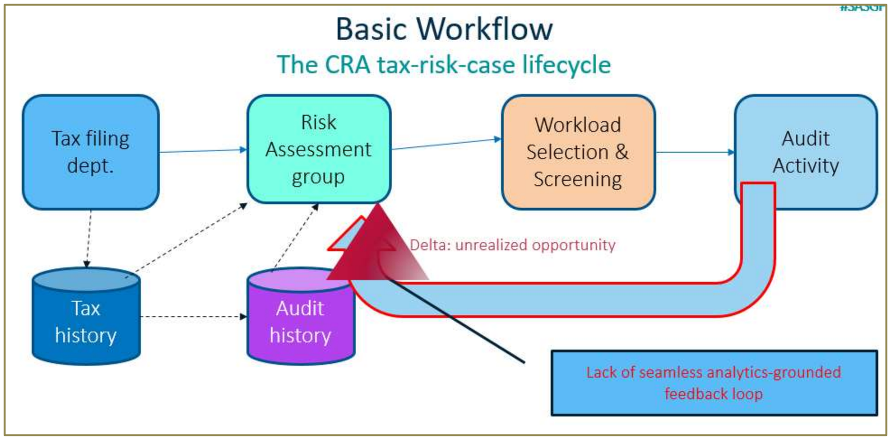
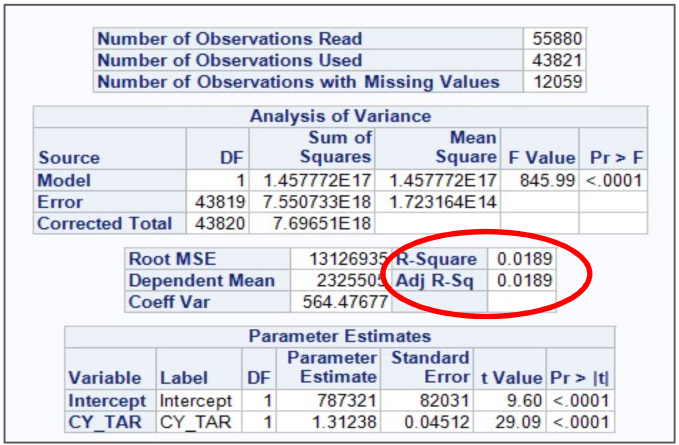

Basic Workflow
The CRA tax-risk-case lifecycle
Tax filing dept.
Risk Assessment group
Workload Selection & Screening
Audit Activity
Tax history
Audit history
Delta: unrealized opportunity
Lack of seamless analytics-grounded feedback loop

Figure 2. Feedback loop disconnect in CRA tax-risk-case lifecycle

There is nothing inherently bad about this *from a day-to-day operational perspective*; after all, the auditing staff are highly preoccupied and devoted to the task at hand to ensure taxpayers are dealt with fairly and expediently. Ergo, they have little time remaining to engage in these peripheral steps beyond the "boots on the ground". We also have to be mindful of the pervasive data model constraint of "one single version of the truth" and maintaining historical integrity [of priority rating]. This is precisely why SAS®, as a trusted partner, has opened so much potential to bridge this gap, as far as a "*predictive* risk level" factor, which we may realize pre-audit or at early stages of audit, and would supplement the traditional priority risk rating.

One observation of note in this regard, is that when using SAS® Enterprise Guide™ in my initial analysis, there was a very low R correlation factor between **TAR** (tax-at-risk) and

TEBA (tax earned by audit)[1]. This was the same pattern for the CRA classes of T1 (small business like sole proprietors or partnerships) or T2 (corporate). In the experiment below, for a T2 dataset, the R^2 value came out to less than 2%.



```
Number of Observations Read                    55880
Number of Observations Used                    43821
Number of Observations with Missing Values     12059

                        Analysis of Variance
                               Sum of          Mean
Source              DF        Squares        Square   F Value   Pr > F
Model                1   1.457772E17   1.457772E17    845.99   <.0001
Error            43819   7.550733E18   1.723164E14
Corrected Total  43820    7.69651E18

Root MSE              13126935   R-Square   0.0189
Dependent Mean         2325505   Adj R-Sq   0.0189
Coeff Var            564.47677

                        Parameter Estimates
                            Parameter   Standard
Variable    Label      DF    Estimate      Error   t Value   Pr > |t|
Intercept   Intercept   1      787321      82031      9.60   <.0001
CY_TAR      CY_TAR      1     1.31238    0.04512     29.09   <.0001
```

Display 1. SAS® Enterprise Guide™ linear regression node output, TAR-TEBA

Note that the "CY_" prefix in TAR means *current year*. (We also have PY_TAR for *past year*. So if the risk data was pulled from 2018, then PY_TAR is from 2017.)

This is one of the reasons why SAS® Enterprise Miner™ has been so helpful to us, due to its automated [and optimal] binning functionality, available from the **Variable Selection** and **Transform Variables** nodes. These assist in uncovering *non-linear* relationships.

On the matter of priority ranking at the risk assessment stage: we have some room for improvement in that regard, such as from observing three years' worth of risk-rated records, where almost two-thirds of them had a priority rating of "High" (but this was not always so at the latter audit stages, as alluded to earlier. The other three levels are "Low", "Medium-Low", and "Medium-High"). Thus, by shrewd scrutiny of our SAS® analytics output, we may open the door to creating more subdivided priority brackets, such as "Very High" and "Top Priority".

ENHANCED TAX MODELING IN SAS® ENTERPRISE MINER™

I began my analysis in SAS® Enterprise Miner™ by importing a file that contained 4,622 observations, with effects that spanned the tax lifecycle as conveyed earlier:

- E-Commerce/web related effects (e.g. site type, DTC, site count)
- Taxpayer Risk-rated profile effects
- Audit case data

This included several class parameters (such as issue codes, industry codes, DTC, and audit classification codes), and interval parameters (such as financial sub-aggregates from tax returns like gross profit, or website count).

The main reason I do not have more than this number is because I only had so much data from the web metrics component – and in any event, we are bound by time considerations as alluded to earlier, meaning that we cannot use several years of risk-profiled data

[1] It should be noted that the term "TEBA" is somewhat of a misnomer, as it doesn't inherently mean that the tax amount has been collected or recovered. It just means that this was the figure concluded in the due course of auditing. It may be regarded as something of an accrual accounting concept.

overlapping with (and ahead of) the timeline of our audit case data. So in this case, I used risk-rated data from 2015-17, and audit case data from the two fiscal years after that. We conduct about 125,000 risk ratings per year with a similar figure for audit completions.

In the course of our analytics exercises, we may consider any of the following types as target (outcome) variables, with examples:

- Binary: tax penalty status, omission of filing status, over a given $ threshold or not
- Interval: TEBA, TAR, Total Hours (spent on audit case)
- Nominal: case selection reason code, project code, audit type code
- Ordinal: end-priority state

PREDICTING ON PENALTY STATE

For our experimental purposes, we are using the binary target of "PENALTY_STATE", which is a derived variable that is set to '1' or TRUE based on the presence of either:

- the application of a penalty (for omissions or gross negligence); or
- the consideration of a penalty (but wasn't actually applied)

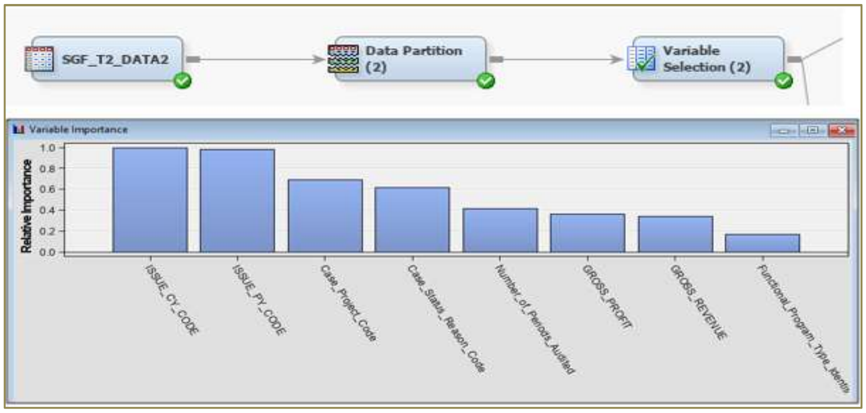
This target binary variable is derived from the field Case Penalty Description – and in the vast majority of cases it is simply "Penalty not considered". So we want to uncover those nuggets, by efficient and automated means, ahead of time – as we have noted a pattern that **even when the status is "Penalty considered", these records are still associated with <u>very high</u> TEBA (and case hours spent).**

To reinforce this point, I observed that within the subset of 37 records in our raw dataset that have PENALTY_STATE = 1, the R correlation factor is ~62.4%. Contrast this with the <2% R factor for the general dataset; even in this reduced set of 4,622 entries with website-related effects, the R factor was ~10.6%. If we take the coefficient of determination, that is the R2 value, the difference becomes even more pronounced.

R-type	Penalty subset, TAR-TEBA	Overall set, TAR-TEBA
R	0.624275978	0.106277
R^2	0.389720496	0.011294801

Table 1. R correlation factors, for data of interest

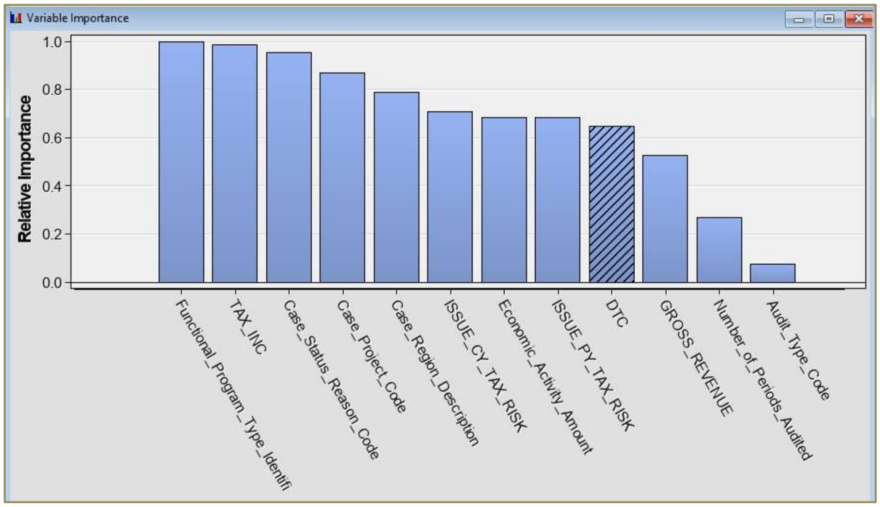
When I first brought in my data to the Enterprise Miner™ workspace diagram, I took care to ensure that all variables were typed accordingly and that irrelevant variables were rejected. I then partitioned it as 70% to Training, and 30% to Validation. As an initial foray into modeling, and some would say as part of best practice, I then inserted a **Variable Selection** node on which I conducted both Chi-sq. and R-sq. significance analysis for the admissible effects:



Display 2. Variable Selection node output, Chi-sq. relative importance

This tells us that the ISSUE_CY_CODE and ISSUE_PY_CODE (current and past year rating, respectively) are very heavily weighted in the model, to the point where we ought to filter them out from consideration so we can uncover less conspicuous (but still useful) predictive inputs. We can do so from the *Edit Variables* (right-click) settings on the node.

On re-running the node, we get this:

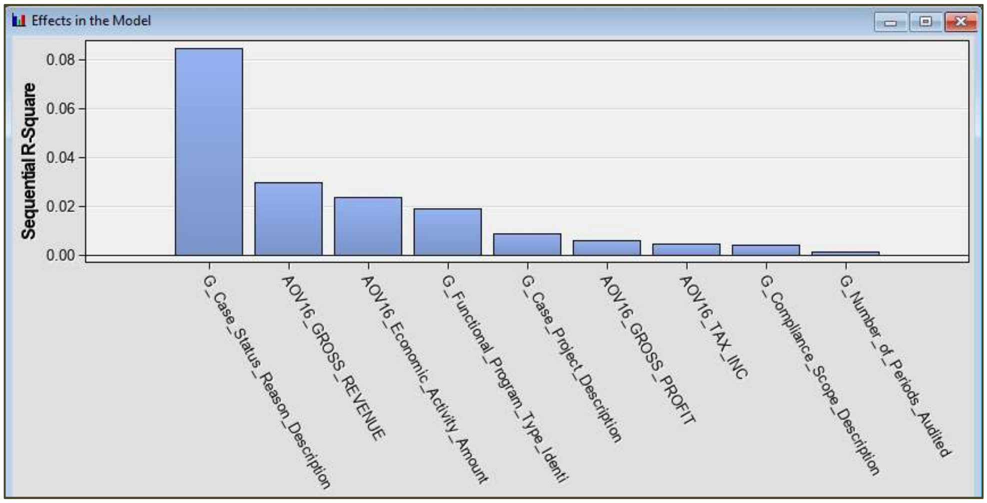


Display 3. Variable Selection, revised

Note that the variable "TAX_INC" is a log-transformed tax variable, but it makes no difference here in terms of significance rating. This is at 98.4% importance. It is, as the name suggests, the amount that a taxpayer reported on their return. Note that in the middle, we have TAR for CY (current year), called "ISSUE_CY_TAX_RISK", at ~70.7% relative importance, which confirms our suspicion that it must be considered in conjunction with other variables as a *non-linear* (and quite likely *non-parametric*) analysis.

Then, as shaded near the end of the revised graph, we have the DTC (Domain Turnover Code), which tells us the website state for a given corporate taxpayer. This is at ~64.6% importance, which emerged as a result of removing the two issue code effects.

We haven't yet examined the Sequential R-square effects, so let us do that also:



Display 4. Sequential R-square graphed effects

Do you notice a pattern from the sequential R-square graph? If you said "it only contains grouped ['G_ABC'] or binned ['AOV16_ABC'] variables", you'd be absolutely right! This is indeed revealing of the type of relationship we're going to be looking at.

Logistic Regression

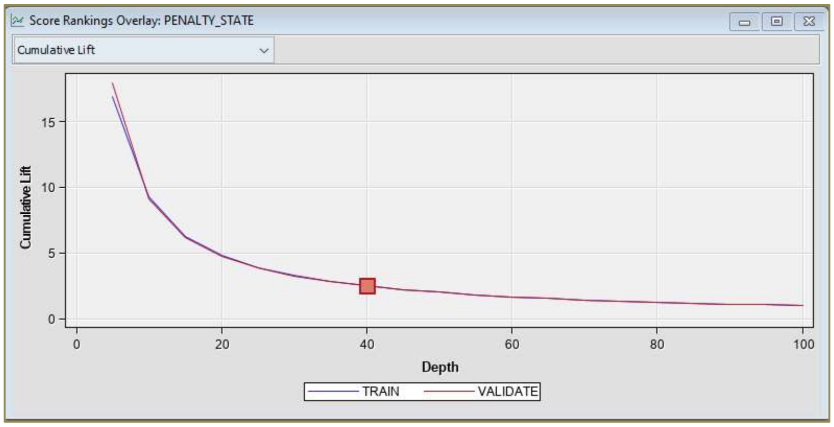
Since we have clearly established that there is a lack of linear relationship at hand, we are going to begin our modelling with a ***logistic regression*** (or, "log reg" as shorthand) node. In so doing, we need to be mindful of four things in the results output:

1. Whether the p-values for entry into the model (based on chi-square value) are acceptable.
2. Whether the minimum R-square value is met for each effect (note that the default for log reg is 0.05, not 0.005 as in the Variable Selection node).
3. Whether *the logit* (i.e. the estimated log-odds) coefficient isn't infinitesimally small, which would contribute nothing to the exponentiated odds value (i.e. $e^{+/-0.001}$ =~ 1.0). As a corollary to this logic, if the confidence interval for the coefficient estimate straddles this 1.0 value, it is also a signal to disqualify the effect from use. In a similar vein, if the coefficient estimate of the log-odds is a pronounced negative number, then the exponentiated odds approximates zero.
4. Whether the confusion matrix at the end of the model output has an acceptable TPR (True Positive Rate) or *Sensitivity* in SAS parlance, and TNR (True Negative Rate), or *Specificity*. A discrepancy in these, between Training & Validation, could mean overfitting.

I don't need to precede this with an *Impute* node, as I have no missing variable instances.

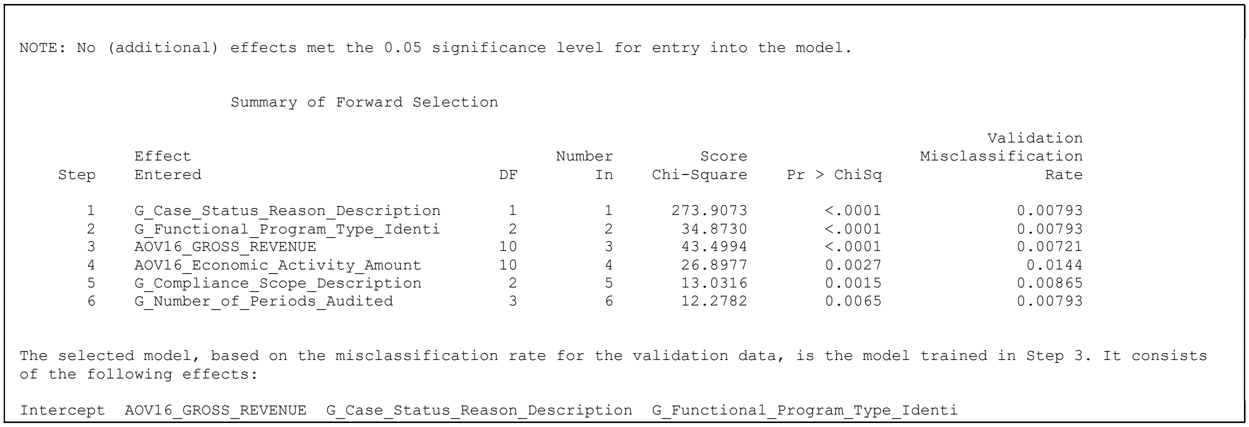
For the initial run, I set the **Selection Model** to "Forward" and the **Selection Criterion** to "Validation Misclassification [Rate]". I refer to the latter in short form as *VMR*. The **Use Selection Defaults** property is set to "Yes".

My Cumulative Lift shows as healthy, as it drops to 4.72 at the 20th percentile and 2.5 at the 40th percentile, but most importantly the training and validation partitions are not offset.



Display 5. Cumulative Lift chart for initial LOG REG model

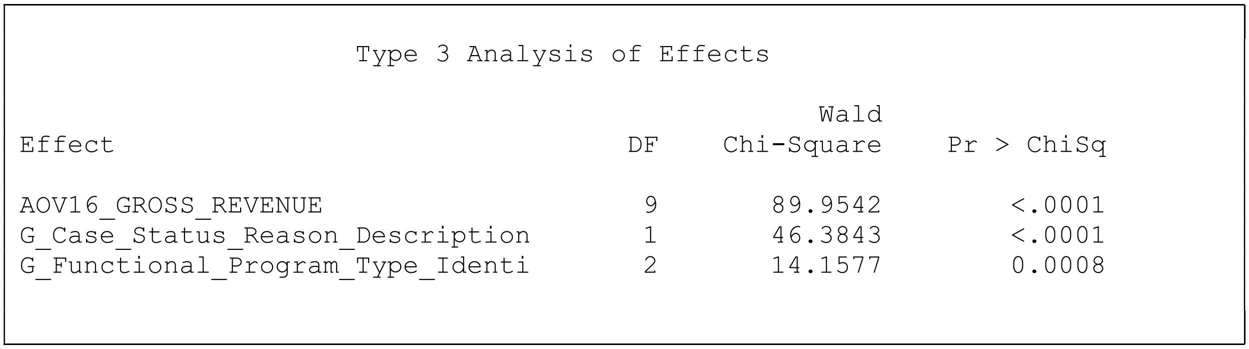
Now here is the output portion, containing our selected model and effects, with the preceding summary of forward selection:



```
NOTE: No (additional) effects met the 0.05 significance level for entry into the model.


                      Summary of Forward Selection

                                                                                   Validation
            Effect                              Number       Score                Misclassification
    Step    Entered                        DF       In    Chi-Square   Pr > ChiSq              Rate

       1    G_Case_Status_Reason_Description    1        1      273.9073       <.0001           0.00793
       2    G_Functional_Program_Type_Identi    2        2       34.8730       <.0001           0.00793
       3    AOV16_GROSS_REVENUE                10        3       43.4994       <.0001           0.00721
       4    AOV16_Economic_Activity_Amount     10        4       26.8977       0.0027            0.0144
       5    G_Compliance_Scope_Description      2        5       13.0316       0.0015           0.00865
       6    G_Number_of_Periods_Audited         3        6       12.2782       0.0065           0.00793


The selected model, based on the misclassification rate for the validation data, is the model trained in Step 3. It consists
of the following effects:

Intercept  AOV16_GROSS_REVENUE  G_Case_Status_Reason_Description  G_Functional_Program_Type_Identi
```

Output 1. Selected model and effects for initial LOG REG model

The variables in the last four steps were not actually selected, due to the Type 3 analysis of effects that followed, which only picked the first three effects:



```
                    Type 3 Analysis of Effects

                                                  Wald
Effect                                  DF     Chi-Square     Pr > ChiSq

AOV16_GROSS_REVENUE                      9        89.9542         <.0001
G_Case_Status_Reason_Description         1        46.3843         <.0001
G_Functional_Program_Type_Identi         2        14.1577         0.0008
```

Output 2. Type 3 Analysis of Effects for initial LOG REG model

Next, we examine the MLE or *Maximum Likelihood Estimates* of each effect – which spans the AOV16 bins of the continuous variable GROSS_REVENUE:

```
                    Analysis of Maximum Likelihood Estimates

                                                   Standard          Wald
Parameter                         DF    Estimate      Error    Chi-Square    Pr > ChiSq    Exp(Est)

Intercept                          1    -11.4084    58.6701          0.04        0.8458       0.000
AOV16_GROSS_REVENUE           1    1      3.5082          .             .             .      33.386
AOV16_GROSS_REVENUE           2    1      6.0763     0.9523         40.71        <.0001     435.402
AOV16_GROSS_REVENUE           3    1     -4.6456      518.0          0.00        0.9928       0.010
AOV16_GROSS_REVENUE           4    1     -4.5168      626.6          0.00        0.9942       0.011
AOV16_GROSS_REVENUE           6    1     -4.5074      623.6          0.00        0.9942       0.011
AOV16_GROSS_REVENUE           7    1     -4.8334      480.9          0.00        0.9920       0.008
AOV16_GROSS_REVENUE           8    1     -4.8334      480.9          0.00        0.9920       0.008
AOV16_GROSS_REVENUE           10   1     -4.9479      449.2          0.00        0.9912       0.007
AOV16_GROSS_REVENUE           12   1      6.5620     1.4465         20.58        <.0001     707.666
AOV16_GROSS_REVENUE           13   1      8.4514     1.1974         49.82        <.0001     999.000
G_Case_Status_Reason_Description 0 1      1.8485     0.2714         46.38        <.0001       6.350
G_Functional_Program_Type_Identi 0 1      5.7158    58.6703          0.01        0.9224     303.634
G_Functional_Program_Type_Identi 1 1      3.7905    58.6698          0.00        0.9485      44.278
```

Output 3. Analysis of Maximum Likelihood Estimates

We can observe that, while the first two bins of AOV16_GROSS_REVENUE contribute quite significantly to the model, the last two bins (12 and 13) contribute even more so. Everything in-between is ruled as dubious to the model. Likewise for both (binary) values of the Grouped FPTI (Functional program type identifier). The Grouped Case Status Reason Description effect also has some contributive value to the model, however.

Lastly, we can examine our **Event Classification Table** (which is the term also used for a confusion matrix). This tells us that we have a sensitivity rate in our training data of 3/26, or ~11.54%, and a specificity rate of 3208/3233 or ~99.23%. We also have a FPR (false positive rate) of 25% or a precision rate of 75%. In the Validation data, the TPR = 1/11, and the specificity =~ 99.28%.

```
Event Classification Table

Data Role=TRAIN Target=PENALTY_STATE Target Label=' '

  False       True        False       True
Negative    Negative    Positive    Positive

   23         3208          1           3


Data Role=VALIDATE Target=PENALTY_STATE Target Label=' '

  False       True        False       True
Negative    Negative    Positive    Positive

   10         1376          .           1
```

Output 4. Event Classification Table for initial LOG REG model

Clearly, there would be some room for improvement; but at least we are not committing a high rate of Type 1 errors (i.e. false positives), and at least we have avoided overfitting our model.

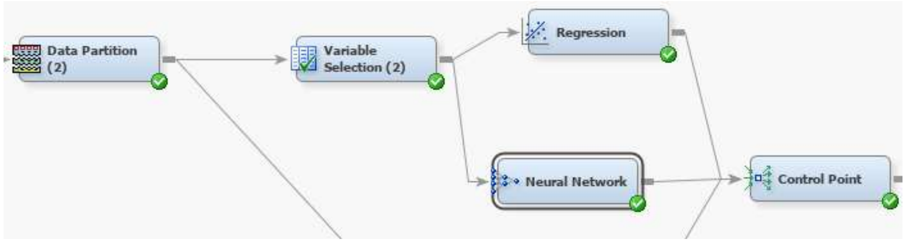
Some other combinations of Selection Model & Criterion that I tried, with results, are as follows:

- Backward / VMR[2], two variables selected (AOV16_GROSS_PROFIT, G_FPTI), *but* it has a 0% sensitivity rate.
- Stepwise / VMR, no different than the forward selection method.
- Forward /AIC, selected variables: AOV16_Economic_Activity_Amount, AOV16_Gross_Revenue, G_Case_Status _Reason, G_Compliance_Scope_Description, G_FPTI, G_Number_of_Periods_Audited. *However*, the model shows signs of overfitting, as the sensitivity in the training data is 5/26, but 0/11 in the validation data.
- Backward/AIC, selected variables: the same as above, plus A0V16_GROSS_PROFIT, AOV16_TAX_INC, and G_Case_Project_Description, less AOV16_Economic_Activity_Amount. The FPR is 50% on the training data and 100% on the validation data. Sensitivity is 5/26 on the training data.
- Stepwise/AIC, selected variables: only the Case_Status_Reason_Description. It has 0% sensitivity for both training and validation datasets.

It is apparent that the Forward/VMR selection model was the best choice.

Neural Network

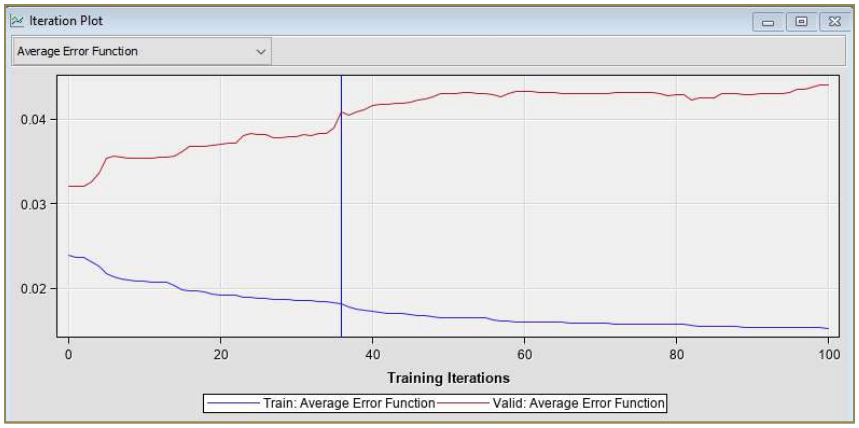
As a matter of typical model comparison convention in SAS®, I went ahead with inputting a **Neural Network** node (again, we don't need to precede this with an Impute node, as I have no missing values for my observations). I connected it to the Variable Selection node, as I did with my Logistic Regression node.



Display 6. Neural Network node in workspace diagram

In the **Network** settings, I went with the default Architecture property of MLP (Multilayer Perceptron), Direct Connection = No, and Number of Hidden Units = 3. I also set the **Model Selection Criterion** to "Misclassification", and ran the node. The iteration plot tells me that it stopped at 36 iterations, well short of the 100 that is standard.

[2] Recall that "VMR" is shorthand for *Validation Misclassification Rate.*



Display 7. Iteration Plot for Neural Network

```
      Optimization Results

Iterations                                        100   Function Calls
376
Gradient Calls                                    215   Active Constraints
26
Objective Function                       0.0152885927   Max Abs Gradient
Element                 0.0025342879
Slope of Search Direction                -0.000171517

QUANEW needs more than 100 iterations or 2147483647 function calls.

WARNING: QUANEW Optimization cannot be completed.
```

Output 5. Optimization Results output for Neural Network

Altogether, the model picked five nominal inputs, and four ordinal ones, but no continuous or binary inputs. The target variable, again, is the binary PENALTY_STATE.

Lastly, we can examine the event classification table output:

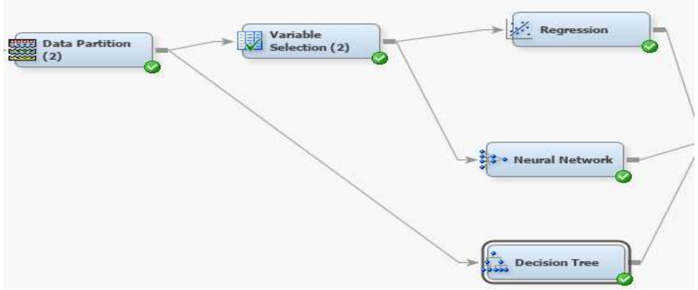
```
      Event Classification Table

Data Role=TRAIN Target=PENALTY_STATE Target Label=' '

  False       True        False       True
Negative    Negative    Positive    Positive

   14         3204          5           12


Data Role=VALIDATE Target=PENALTY_STATE Target Label=' '

  False       True        False       True
Negative    Negative    Positive    Positive

    8         1374          2            3
```

Output 6. Event Classification Table for Neural Network

This gives us a training sensitivity (TPR) of 12/26 or almost 50%, and validation sensitivity of 3/8 or 37.5%. However, things are somewhat less acceptable with regards to *the precision*, which is 12/17 (70.6%) in the training data, and 60% in the validation data.

Decision Tree

Our next step in the model evaluation for predicting on PENALTY_STATE is to introduce a **Decision Tree** into the flow. However, unlike the Log Reg or Neural Net models, we need not put this after our *Variable Selection* node, because Decision Trees (being non-parametric in nature) do optimal "on-the-fly" binning, using a combination of multiple variables. The issue of missing variable values across observations is also moot, as stated before.



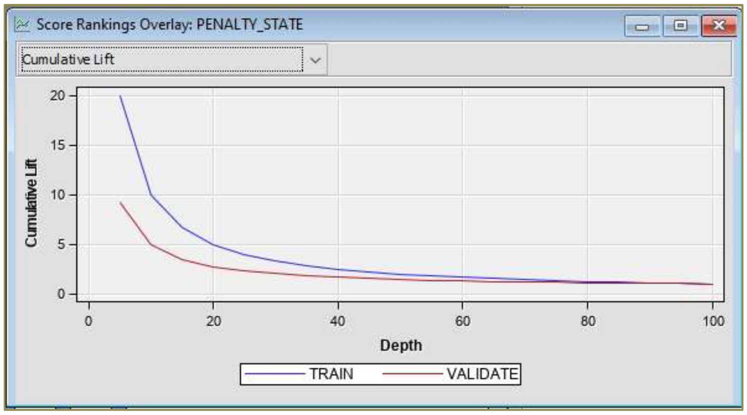
Display 8. Addition of a Decision Tree node to the workflow

I accepted most of the default properties for the Tree, changing these ones:

- Target Criterion = Gini
- Method = Largest [i.e. the maximal tree]
- Assessment Measure = Misclassification (VMR)

For ***the Gini criterion***, this is typically associated with economics i.e. as a measure of wealth disparity in countries or other jurisdictions, but it works well in this context too, as we are ultimately concerned with relative *node purity* in determining the optimal tree splits.

On running the decision tree, I determined that my Cumulative Lift was quite anomalous compared to the logistic regression model earlier.



Display 9. Cumulative Lift, Decision Tree model

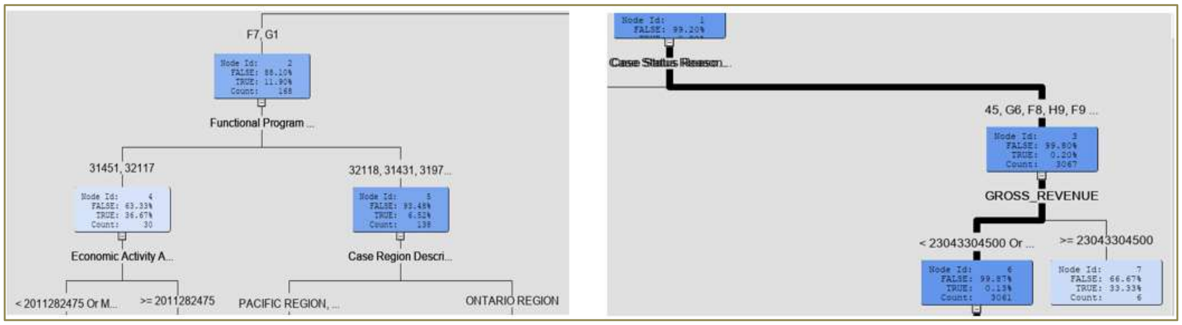
This poses some cause for concern, as it indicates possible overfitting in our model; the gap between training and validation data doesn't really narrow until the 40th percentile, by which point it has plateaued.

At least, our confusion matrix tells us that we are on the right track, when it comes to picking the best model:

```
Event Classification Table

Data Role=TRAIN Target=PENALTY_STATE Target Label=' '

  False       True        False       True
Negative    Negative    Positive    Positive

   18         3207          2           8


Data Role=VALIDATE Target=PENALTY_STATE Target Label=' '

  False       True        False       True
Negative    Negative    Positive    Positive

    8         1374          2           3
```

Output 7. Event Classification Table for Decision Tree

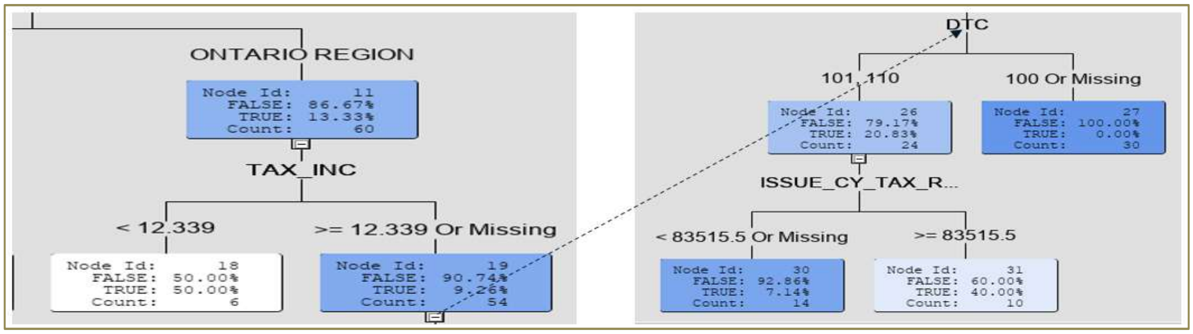
In examining the initial split of the tree itself, it is on the categorical (nominal) effect of Case_Status_Reason_Code, which is F7 or G1 (meaning a re-assessment of the case was necessary). This was true on the left-hand split; the right-hand split removed the bulk of the impurity, i.e. the non-reassessment codes were just tied to six penalty states.



Display 10. First split breakdown of Decision Tree

Continuing down the left branch of the tree, on a subset of FPID [Functional Program ID], then Regional split by ONTARIO, and then [LOG]TAX_INC, then DTC, we pinpoint five penalty instances where the DTC was 101 or 110 (overseas-hosted or masked domain).

Beneath that, the final (leaf node) split is on CY_TAR, where 4/5 positive penalty states are where CY_TAR >= $83,515.50.

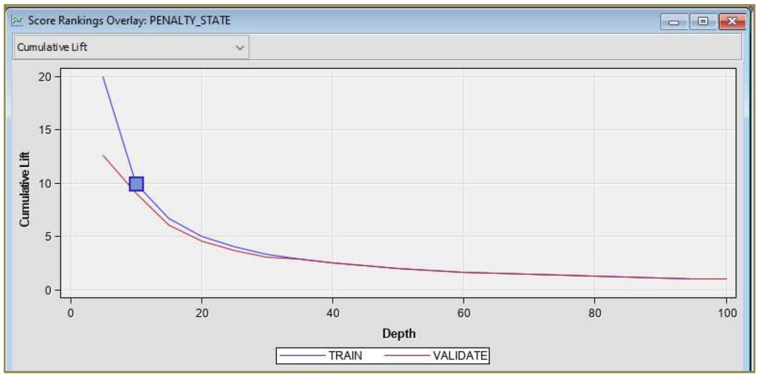


Display 11. Lower-level, left-hand subtree breakdown

Ensemble node

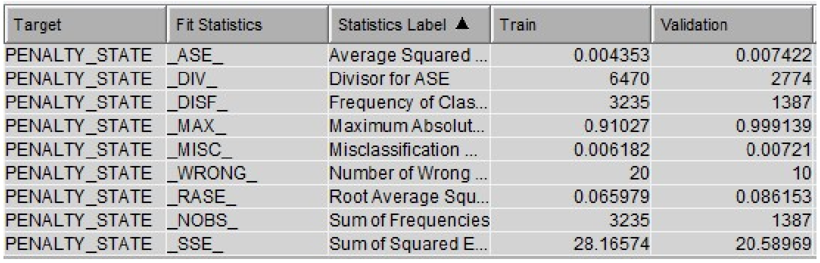
As a final modeling step, we would like to derive what I call "the best of all worlds", using an Ensemble node to create a hybrid model based on our three modeling outputs so far. This will not necessarily give us a better outcome than before, but it is worth trying.

On running the node, I get the familiar asymptotic Cumulative Lift; this is still agreeable, as there is no pronounced difference beyond the 10th percentile between the partitions.



Display 12. Cumulative Lift for Ensemble model

Here is my table of fit statistics:



Target	Fit Statistics	Statistics Label ▲	Train	Validation
PENALTY_STATE	_ASE_	Average Squared ...	0.004353	0.007422
PENALTY_STATE	_DIV_	Divisor for ASE	6470	2774
PENALTY_STATE	_DISF_	Frequency of Clas...	3235	1387
PENALTY_STATE	_MAX_	Maximum Absolut...	0.91027	0.999139
PENALTY_STATE	_MISC_	Misclassification ...	0.006182	0.00721
PENALTY_STATE	_WRONG_	Number of Wrong ...	20	10
PENALTY_STATE	_RASE_	Root Average Squ...	0.065979	0.086153
PENALTY_STATE	_NOBS_	Sum of Frequencies	3235	1387
PENALTY_STATE	_SSE_	Sum of Squared E...	28.16574	20.58969

Table 2. Fit Statistics for Ensemble model

Now we can examine the confusion matrix for this Ensemble model…

```

Event Classification Table
```

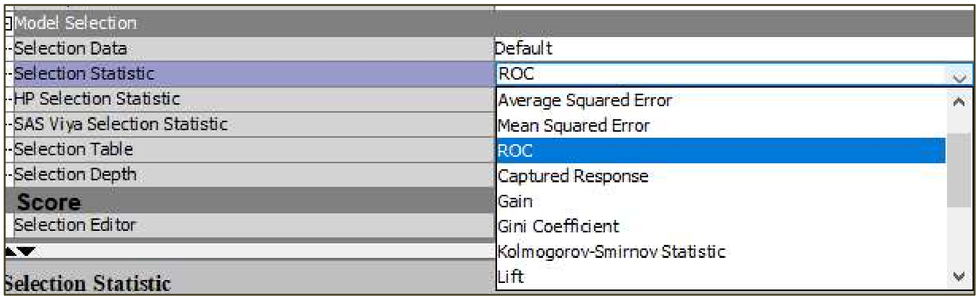
```
Data Role=TRAIN Target=PENALTY_STATE Target Label=' '

  False       True        False       True
Negative    Negative    Positive    Positive

   18         3207          2          8


Data Role=VALIDATE Target=PENALTY_STATE Target Label=' '

  False       True        False       True
Negative    Negative    Positive    Positive

    8         1374          2           3
```

Output 8. Event Classification Table for Ensemble model

Just as it did with the Decision Tree, this gives me a sensitivity rate of 8/26 for the training data, and 3/11 for the validation data (a very slight difference). While the false positive rate is only 20% for the training data, it's 40% on the validation data. All things considered, it's not a serious case of overfitting.

MODEL COMPARISON

Finally, I can run a **Model Comparison** node. This provides a **ROC** [Receiver Operating Characteristic] chart telling me what the preferred model is. As the Selection Statistic, this is what I actually pick (ROC), although I could pick others such as the Misclassification Rate or Cumulative Lift.



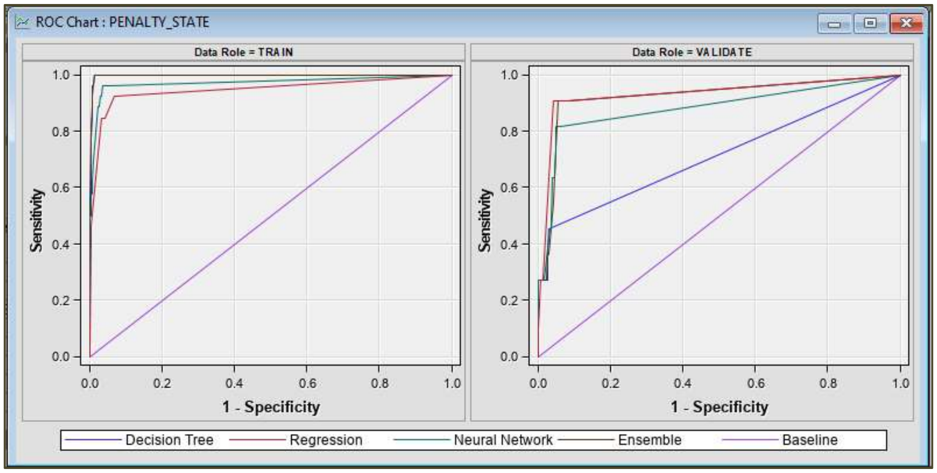
Model Selection

Selection Data: Default

Selection Statistic: ROC

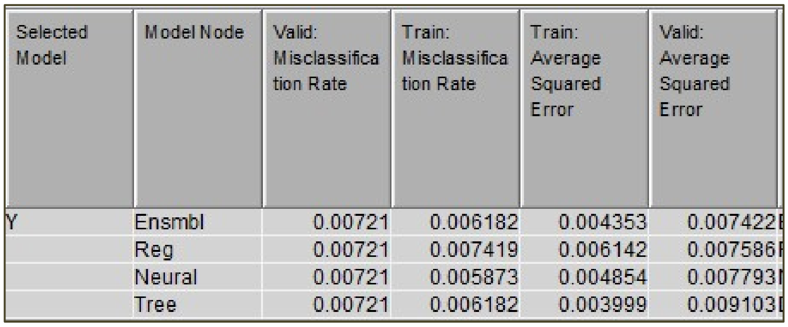
Display 13. Selection Statistic for Model Comparison node

It is perhaps somewhat anticlimactic, but purely on the basis of the ROC index, the Regression model emerges "the winner". This might be true if we didn't give so much leeway to the fact that it has a higher false negative rate relative to our other models.



Display 14. ROC Chart for PENALTY_STATE

We will run the model comparison again with the Misclassification Rate [Validation] as the selection statistic, to see which model is best. While the ROC curve appears the same as above, what we're interested in examining is the table of Fit Statistics.



Selected Model	Model Node	Valid: Misclassification Rate	Train: Misclassification Rate	Train: Average Squared Error	Valid: Average Squared Error
Y	Ensmbl	0.00721	0.006182	0.004353	0.007422
	Reg	0.00721	0.007419	0.006142	0.007586
	Neural	0.00721	0.005873	0.004854	0.007793
	Tree	0.00721	0.006182	0.003999	0.009103

Table 3. Fit Statistics, final model comparison

As it turns out, the Ensemble model was in a "dead heat" with the Decision Tree, having a tied VMR; but based on the preponderance of other selection criteria, it emerged the winner.

While the ASE for the training data was lowest for the Tree, this was not the case for the validation data where the lowest value was for Ensemble.

ADJUSTING MEASURES EXPLORED

Prior to reaching a conclusion, and given the fact that we had pervasive "Type 2" errors (i.e. an otherwise positive instance was predicted as a false negative), I pondered what could be done to improve upon our predictive capacity. I thus decided to engage in some coerced feature selection, in which I filtered out ranges of observations that had no association to the target, more specifically, items that tended to have a large number of superfluous values (like zeroes for continuous effects, or a "General – All other" for a categorical effect).

For starters, in my candidate dataset (containing 4,622 records), I removed all 334 instances where the interval variable "Unassigned_Days" = 0. As all 334 of these were only tied to PENALTY_STATE = "FALSE", it stands to reason that a case that would be tied to something serious like a penalty later on would not have zero unassigned days.

I then observed that, for the class variable "Audited_Income_Category_Description", where it equals Rental, Salaried, Professional, Investment, or Capital Gain/Loss, I may remove these – leaving the two remaining categories of "Business" and "Unknown".

Next, I eliminated any observations for four categories of the "Case_Definition_Description" where it pertained to case screening or quality assurance, all corresponding to not just a false penalty_state, but to zero TEBA and near-zero total hours.

Next, in scrutinizing the class variable "Case_Project_Description", I found that only 7/33 of the values apply where a penalty_state = "TRUE" is concerned, and these are either generics (like "WIP", "Not applicable", or "Not Specified") or real estate related. So I eliminated 219 records from the observations related to the other 26/33 class values.

Finally, to arrive at a total number of records of about 3,700 (so that 37 penalty instances represent ~1% of all observations), I removed 12 ISSUE_CY_CODES which had no relation to a "true" penalty_state. This brought it down to 3,688 observations.

Instead of using the Variable Selection node, I used the **Transform Variables** node and specified Optimal Binning [to maximize relationship to the target].

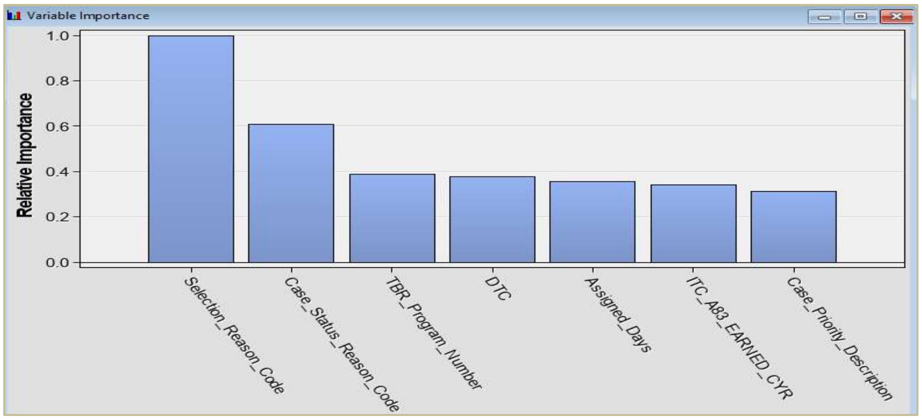
When running my Logistic Regression model with the same settings as accepted earlier (Selection Model = "Stepwise", Selection Criterion = "VMR"), it picked the same three variables as before **but** this time, we see an improvement in the Sensitivity rate at 25% for validation, but with a trade-off in the FPR (1 minus Specificity) at 40% (i.e. 60% precision).

If I run it with the selection criterion of "AIC" rather than VMR, there is drastic overfitting. It gives me "perfect" classification from the training dataset, yet close to 50% Sensitivity and a FPR of 8/13. Clearly, this is not acceptable, and in any event we would only use the Akaike Information Criterion if we put less emphasis on Type 1 errors – which we don't!

The decision tree confusion matrix actually turned out to be exactly the same as with our full dataset. A Gradient Boosting model did nothing to alleviate the dilemma of having a relatively low recall rate.

So as a final effort on coerced feature selection, I observed that 1,491 instances of my streamlined dataset contained class variable Selection Reason Code (SRC) = "Regular". This "Regular" value is a generic one that is not tied to any positive penalty states. Thus I removed it, leaving us with ~2,200 observations.

From *variable selection*, this gave me the following relative importance chart:



Display 15. Revised Relative Importance, streamlined dataset

But, as we might have expected, this gave me drastic overfitting on my logistic regression model, going from a majority true positives and sensitivity (in the training portion) to completely empty for both (in the validation portion).

For *the decision tree*, signs of overfitting were less conspicuous:

- Training data: Recall = 14/22 or ~64%; Precision = 14/19 or ~74%.
- Validation data: Recall = 4/11 or ~36.4%; Precision is the same at 36.4%.

With a validation precision of about half that of the training data, I would have no rational basis to accept this model over the original decision tree with the full dataset, where we got a near-equivalent recall rate, but 60% precision.

CONCLUSION

From what we have seen of our rigorous experimentation, it is clear that coerced feature selection is not always the best remedy; it can introduce overfitting to our model. We also know that using the AIC (Akaike Information Criterion), while providing somewhat of a boon to our sensitivity (recall) rate in both partitions, sacrifices a disproportionate degree of precision, which we can't live with.

Due to the nature of our organization, we are more closely aligned to the famous maxim of our legal system, paraphrased: "it is far better to commit a type 2 error, than it is to commit a type 1 error".

In any event, we can still make tremendous use of what we have mustered, given the fact that all the false negative observations that I encountered have extremely high TEBA (about 10 times the average) and a similarly quantified average number of hours spent. So, one can see how this would assist with case allocation, tax recovery, and workload optimization regardless. (Note that we couldn't use TEBA as a predictive input, given our staircase diagram, because it's not realized until the penultimate stage).

Ergo, based on our discoveries, I believe that we need to take one *or all* of three remedies:

1. Obtain a much larger dataset, preferably one with tens of thousands of records. The constraint here is that we can only reasonably do predictive model building using chronologically-oriented input tables, i.e. the tax-risk-profiling tables can't occur during the entire audit window, nor should they be many years prior.
2. Predict on a factor that is much broader then a penalty state (but one that may well encompass this factor to some degree – so we basically invert the penalty as predictor rather than target). Our analysis is hampered by the fact that penalty instances are very few and far between.
3. Use another model with SAS code or another SAS tool, perhaps Bayesian Network Analysis (BNA) which is great at evaluating conditional probability across class variables with many levels.

From that point, I would like to also inject a test dataset for scoring with the model assessment. This could be taken from the most recent fiscal quarter.

This exercise has certainly served to narrow down what modeling options we ought to pursue. To paraphrase Thomas Edison, "I haven't failed – I've just discovered many ways NOT to make an ideal light bulb!"

REFERENCES

No references were required or used in the composition of this paper, as the material was all originally-sourced.

ACKNOWLEDGMENTS

None

RECOMMENDED READING

- *OECD Secretary-General Tax Report to G20 Finance Ministers and Central Bank Governors.* October 2019, OECD, Paris.
- *Predictive Modeling with SAS® Enterprise Miner™: Practical Solutions for Business Applications, 3rd edition.* Copyright © 2017 SAS Institute Inc., Cary, NC, USA.
- *The Tax Professional of the Future: Staying relevant in changing times.* PWC publications, June 2017.
- *Text Analytics in Government: Using Automated Analytics to Unlock the Hidden Secrets of Unstructured Data*, Copyright © 2014, SAS Institute Inc., Cary, NC, USA.

CONTACT INFORMATION <HEADING 1>

Your comments and questions are valued and encouraged. Contact the author at:

Jason A. Oliver
Canada Revenue Agency
Jason.oliver@cra-arc.gc.ca