

DSC 433/533 – Class 8 – Decision tree examples for Tayko case

This class exercise is based on the **Tayko Software Reseller** Case (see separate document on the handouts page of the course website) and the Excel datasets **Tayko_allpart.xls** and **Tayko_part.xls** (available on the data page of the course website).

- The **Tayko_allpart.xls** has already been partitioned into Training (800), Validation (700), and Test (500) samples. Fit a decision tree model to classify “purch.” In particular:
 - Select XLMiner > Classification > Classification Tree.
 - Step 1: Move all the variables from “usa” to “res” to the “Input variables” box and “purch” to the “Output variable” box.
 - Step 2: Select “Prune tree” and “Minimum # records in a terminal node” and set # records at “80.”
 - Step 3: Select “Best pruned tree” and “Minimum error tree.”
 - Select “Summary report” for “Score training data.”
 - Select “Detailed report,” “Summary report,” and “Lift charts” for “Score validation data.”
 - De-select “Summary report” for “Score test data” (we won’t be using test data for this exercise).

The best pruned tree is drawn on the sheet labeled “CT_PruneTree1” and summarized in tabular form on the sheet labeled “CT_Output1.” Assuming an initial cutoff probability value for success (purchaser) of 0.5, the decision rules resulting from the best pruned tree are:

- If $freq \geq 2$, then predict purchaser;
- If $freq = 1$ and customer placed at least 1 order via web then predict purchaser;
- If $freq = 1$ and customer has not placed at least 1 order via web then predict non-purchaser.
- If $freq = 0$ then predict non-purchaser;

- We can use the results of the decision tree analysis to estimate the probability that a customer will purchase based on the training sample proportions of purchasers in each decision category:
 - $211/278 = 0.75899$;
 - $99/178 = 0.55618$;
 - $69/180 = 0.38333$;
 - $0/164 = 0$.

The “error reports” on the Output worksheet summarize the results for the training and validation samples based on a 0.5 probability cut-off:

Training Error Report				Validation Error Report			
Class	# Cases	# Errors	% Error	Class	# Cases	# Errors	% Error
1	379	69	18.21	1	377	81	21.49
0	421	146	34.68	0	323	86	26.63
Overall	800	215	26.88	Overall	700	167	23.86

Another summary of model performance is the lift in the first decile, which is 1.59 in the validation sample.

- (i) Our next task is to see if models with a smaller minimum # records in a terminal node can outperform this first model. To simplify matters we will use just these two measures of model performance: “% validation error using a 0.5 cut-off” (23.86) and “validation lift in the first decile” (1.59). (A more sophisticated analysis might consider different costs for the two types of error: predicting purchase for non-purchasers and predicting non-purchase for purchasers.)
 - The best pruned model with 20 as the minimum # records in a terminal node has 22.00% validation error and 1.70 validation lift in the first decile.
 - The best pruned model with 4 as the minimum # records in a terminal node has 21.00% validation error and 1.65 validation lift in the first decile.

(iv) The best pruned model with 1 as the minimum # records in a terminal node has 20.43% validation error and 1.65 validation lift in the first decile.

The final model (1 as the minimum # records in a terminal node) has the smallest validation error, but lift is lower than in the second model (20 as the minimum # records in a terminal node), which has validation error 22.00% and lift 1.70. While the second model is probably the best of these decision tree models for this case study (where first decile lift is more important than error over the whole sample), it doesn't quite beat the best logistic regression model from class 10, which has validation error 21.14% and lift 1.80. We will see later if neural networks (class 12) can beat this model's error and lift results.

4. In the first two decision tree analyses, the best pruned tree and minimum error tree are the same. However, the third and fourth decision tree analyses have minimum error trees that are somewhat larger than the best pruned trees. For example, the best pruned model with 4 as the minimum # records in a terminal node has 21.00% validation error, while the minimum error tree has 20.29% validation error. However, this is not statistically distinguishable since 21.00% is within one standard error (1.52% in this case) of 20.29%.
5. Now, for the purposes of illustration, we will digress from the Case and consider lift charts and profit calculations using the second model results and some alternate assumptions about costs and revenues. First, consider the "classification confusion matrix" for the validation data for the best pruned tree on the "CT_Output2" sheet:

Classification Confusion Matrix		
	Predicted Class	
Actual Class	1	0
1	293	84
0	70	253

Expected net profits		
	Predicted Class	
Actual Class	1	0
1	1172	0
0	-140	0
	1032	

You can manually add a table for expected net profits based on it costing \$2 to send a catalog to a customer and Tayko receiving \$6 on average for each purchaser. For example, 293 customers sent a catalog purchased something and resulted in a profit of $293 \times (6 - 2) = \$1,172$. Conversely, 70 customers sent a catalog did not purchase anything and resulted in a loss of $70 \times 2 = \$140$. The other $84 + 253$ customers have a modeled probability of purchase less than 0.5 and so would not be sent a catalog. Net profit is therefore $1,172 - 140 = \$1,032$.

You can change the cut-off probability from 0.5 to some other number to see whether setting a different cut-off can lead to better profitability – in XLMiner it is the number in the blue cell just above the "classification confusion matrix" for the validation data for the best pruned tree (cell F185?).

Complete the following table:

Cut-off probability	1	0.9	0.5	0.38	0.11	0
Catalogs sent	0	225	363	543	562	700
Decision tree model 2 profit	0	\$786	\$1,132	\$1,158	\$1,138	\$862

6. The **Tayko_part.xls** has already been partitioned into Training (379), Validation (377), and Test (244) samples. Fit a decision tree model to predict "spend." In particular:
 - Select XLMiner > Prediction > Regression Tree.
 - Step 1: Move all the variables from "usa" to "res" to the "Input variables" box and "spend" to the "Output variable" box.
 - Step 2: Select "Minimum # records in a terminal node" to be "38," and "Using Best prune tree" for "Scoring option."
 - Step 3: Select "Pruned tree" and "Minimum error tree."
 - Select "Summary report" for "Score training data."
 - Select "Detailed report," "Summary report," and "Lift charts" for "Score validation data."
 - De-select "Summary report" for "Score test data" (we won't be using test data for this exercise).

Results are RMSE of 196.5 in the training sample, RMSE of 184.6 in the validation sample, and first decile lift of 2.66.

7. Similar results for 20 minimum # records in a terminal node are RMSE of 183.7 in the training sample, RMSE of 175.9 in the validation sample, and first decile lift of 2.58.

Similar results for 10 minimum # records in a terminal node are RMSE of 146.7 in the training sample, RMSE of 188.2 in the validation sample, and first decile lift of 2.66.

The first decision tree with 38 minimum # records in a terminal node seems to be the best of the three, but none are as good as the best multiple linear regression model from homework 3 which had RMSE of 163.0 in the validation sample, and first decile lift of 2.73.