Building and Evaluating an ‘Optimal’ Decision Tree

2018-09-23

When we build a model that classifies the cases in the training set really accurately, but doesn’t classify very accurately in “future data sets” (test sets) we have **overfitted** the model. Our goal in building a model is to describe the actual relationships between age and support of same-sex marriage. Overfitting is when we describe the random variation rather than the relationships. This is a problem for generalization, and makes us overoptimistic about how our model will perform. To try to reduce overfitting the model to our training data, we will use a more systematic method of creating the decision tree.

Consider the following two tree models fitted to the training data: (1) a tree based on no decision rules (left) and (2) a tree based on a single decision rule (right):

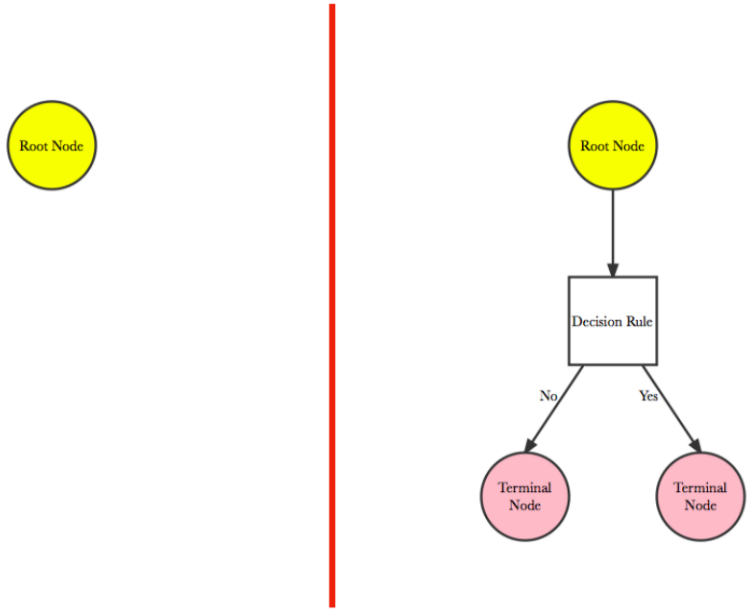
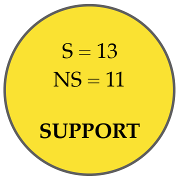


Figure 1: Left tree shows a classification tree based on no decision rules. Right tree shows a classification tree that uses a single decision rule.

Model with No Decision Rules

In the tree for the model that includes no decision rules, there is only a single node—the **root node**. This is the simplest classification model we can have. Using this model we would classify every case as the dominant class.

In our training data, 13 of the 24 (0.542) cases support same-sex marriage and 11 of the 24 cases (0.458) do not support same-sex marriage. Here, the dominant class is “Support”, so this model would classify everyone as supporting same sex marriage. This model would then have a Model Error Rate of 0.458.



Model Error Rate = 0.458

Figure 2: Model using no decision rules on the training data.

Model with One Decision Rule

The tree for a model with a single decision rule takes the root node and splits (or partitions) the data into two **terminal nodes**. This is a more complex model than the no decision rules model, which only had one terminal node (the root node). In this model, every case in the right terminal node would be classified as supporting same-sex marriage and every case in the left terminal node would be classified as not supporting same-sex marriage.

If we used the decision rule `{Age < 32}`, there would be four cases sent to the right terminal node. All four of those cases, classified as supporting same-sex marriage, were classified correctly; error rate of 0.000. The remaining 20 cases would be classified as not supporting same-sex marriage. Of these, 11 were correctly classified. This node has an error rate of 0.45.

To compute the overall error rate of this model, we compute the weighted average of the error rates from all terminal nodes. We use a weighted average because the node sizes are not equal. For this example, we would compute the Model Error Rate as:

$$\text{Model Error Rate} = \frac{\bigg(20(0.45) + 4(0)\bigg)}{24} = 0.375$$

Note that this is the same as the model's overall mis-classification rate. Since we can write each node's error rates as fractions,

$$\begin{aligned}\text{Model Error Rate} &= \frac{20(\frac{9}{20}) + 4(\frac{0}{4})}{24} \\ &= \frac{9}{24} \\ &= 0.375\end{aligned}$$

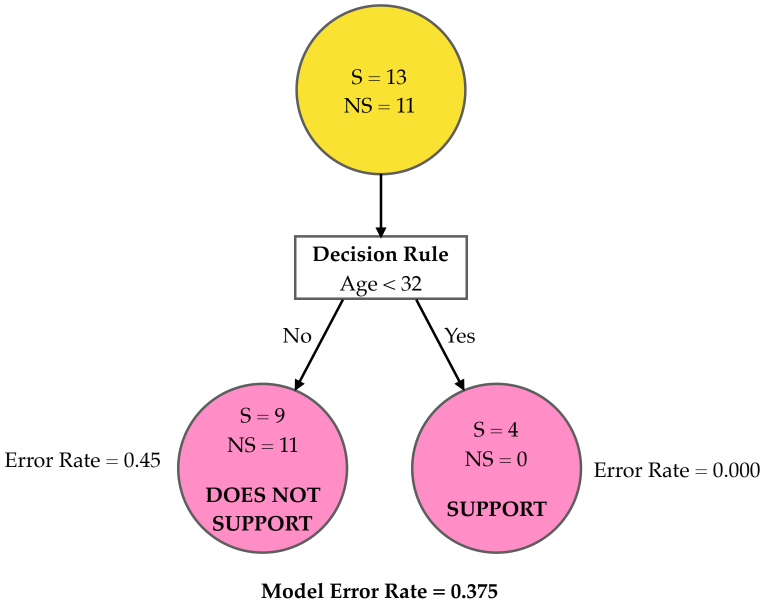


Figure 3: Model using a single decision rule (Age < 32) on the training data.

In general, for any classification model, we can compute the Model Error Eate by determining the weighted average of the error rates from all terminal nodes—which is the overall model mis-specifiction rate.

1. Verify that the Model Error Rate of 0.458 that we computed for the model with no decision rules can be computed by determining the weighted average of the error rates from all terminal nodes.

Model Selection: Choosing Between the Two Models

In summary, we have two models that we are trying to choose between

Model	Error Rate	Complexity
No decision rules	0.458	1 terminal node
One decision rule	0.375	2 terminal nodes

Comparing the models and the two error rates:

- The model with one decision rule is more complex than the model with no decision rules; it has two terminal nodes as opposed to one terminal node.
- The model with one decision rule has a lower error rate than the model with no decision rules. By adopting the model with one decision rule (`{Age < 32}`) we could decrease our mis-classification rate by

0.083.

If we choose the more complex model with one decision rule, we get a lower Model Error Rate, but that comes at a cost of additional complexity, and as we saw earlier, more complex models are less generalizable than simpler models (they are prone to overfit). So the question we need to ask is: *Is the decrease in error rate worth the increase in complexity?*

Rather than perform a statistical test, we will use a pre-determined criterion to evaluate whether the added complexity is worth it. For example, if by adding an additional terminal node we can decrease the Model Error Rate by at least 0.10, it is worth it.

Because we will be using this criterion to evaluate any of our models, it is important that this is specified (and held to) **before looking at the data**.

2. Set the criterion you will use to evaluate your models by filling in the following sentence.

   If by adding an additional terminal node, the Model Error Rate decreases by at least ____ we will adopt the more complex model. If it does not decrease by at least that amount, we will adopt the simpler model.

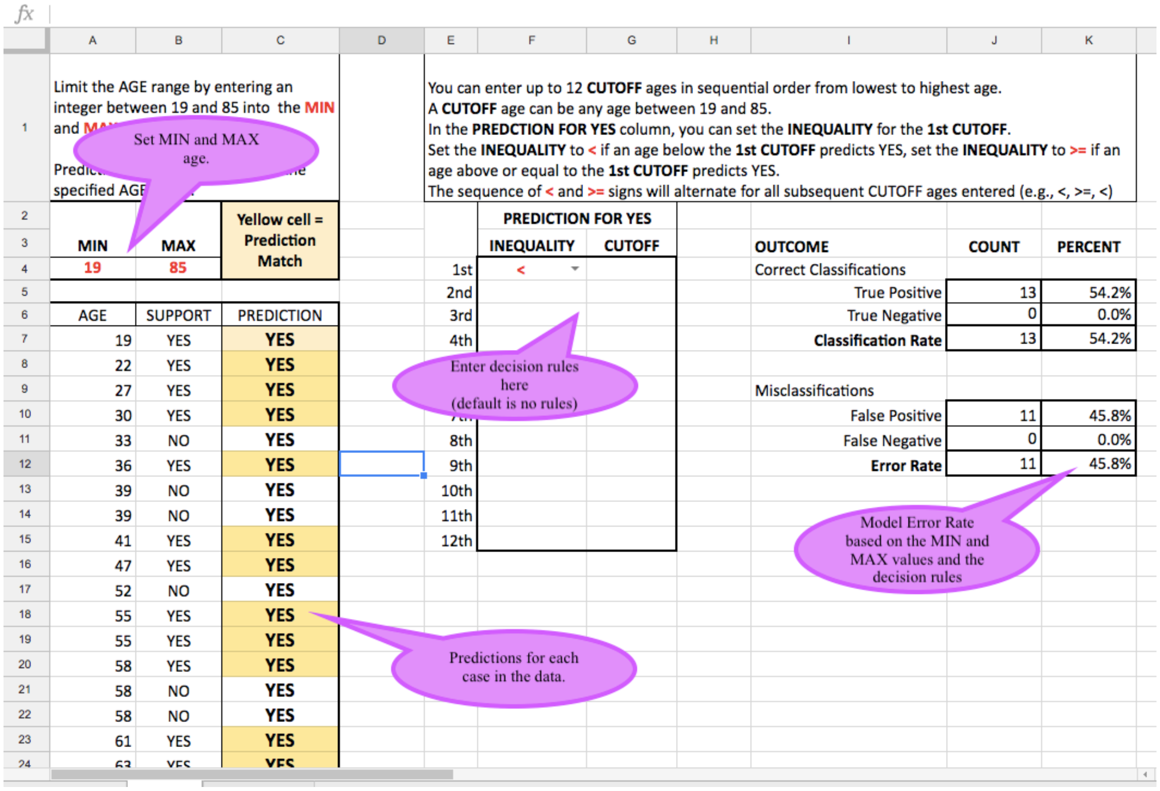
3. Find the Model Error Rate for the model with one decision model based on your first decision rule (e.g., ignore all the other decision rules in your tree and imagine you had stopped after the first decision rule).

4. How much does the model that uses your first decision rule reduce the Model Error Rate from the model with no decision rules?

5. Based on the two Model Error Rates you computed, will you adopt the model with no decision rules or the model that uses your first decision rule? Explain.

Using a Google Sheets Tool to Compute Model Error Rate

Open the *same-sex-marriage* Google Sheet (it should be shared with your UMN account). This document will automate the computation of the Model Error Rate based on predictions for a set of user-entered decision rules. The default model, upon opening the Google Sheet, is the model with no decisions rules—which predicts support for everyone (based on the majority class).


fx
Set MIN and MAX age.
Enter decision rules here (default is no rules)
Predictions for each case in the data.
Model Error Rate based on the MIN and MAX values and the decision rules

Limit the AGE range by entering an integer between 19 and 85 into the **MIN** and **MA**… Predic… …ne specified AGE …

You can enter up to 12 **CUTOFF** ages in sequential order from lowest to highest age.
A **CUTOFF** age can be any age between 19 and 85.
In the **PREDCTION FOR YES** column, you can set the **INEQUALITY** for the **1st CUTOFF**.
Set the **INEQUALITY** to **<** if an age below the **1st CUTOFF** predicts YES, set the **INEQUALITY** to **>=** if an age above or equal to the **1st CUTOFF** predicts YES.
The sequence of **<** and **>=** signs will alternate for all subsequent CUTOFF ages entered (e.g., <, >=, <)

MIN	MAX	Yellow cell = Prediction Match
19	85	

AGE	SUPPORT	PREDICTION
19	YES	YES
22	YES	YES
27	YES	YES
30	YES	YES
33	NO	YES
36	YES	YES
39	NO	YES
39	NO	YES
41	YES	YES
47	YES	YES
52	NO	YES
55	YES	YES
55	YES	YES
58	YES	YES
58	NO	YES
58	NO	YES
61	YES	YES
63	YES	YES

	PREDICTION FOR YES	
	INEQUALITY	CUTOFF
1st	<	
2nd		
3rd		
4th		
8th		
9th		
10th		
11th		
12th		

OUTCOME	COUNT	PERCENT
Correct Classifications		
True Positive	13	54.2%
True Negative	0	0.0%
Classification Rate	13	54.2%
Misclassifications		
False Positive	11	45.8%
False Negative	0	0.0%
Error Rate	11	45.8%

Figure 4: Screenshot showing the Google Sheets tool. The default model is the no decision rules model.

6. Use the Google Sheets tool to compute the Model Error Rate for the model with the decision rule of `{Age < 32}`. Verify this is 0.375; the same value we computed earlier.

7. Use the Google Sheets tool to compute the Model Error Rate for the model based on your first decision rule. Verify this is the same value you computed in Question #3.

Model Building: Build Your Model Using the Training Data

There are many models with one decision rule that could be created from the root node. for example:

- `{Age < 20}`
- `{Age < 24}`
- `{Age < 28}`
- Etc.

In fact with $N = 19$ unique cases there are $N - 1 = 18$ unique models with one decision rule. Classification algorithms fit each of those models and then choose the decision rule that would decrese the Model Error Rate the most. That becomes the first split.

8. Use the Google Sheets tool to evaluate each of the potential models with one decision rule. Which rule leads to the largest decrease in Model Error Rate?

9. Based on your earlier criterion, will you adopt the model with no decision rules or this model with one decision rule? Explain.

To determine if we should adopt a model with two decision rules, we have to examine the Model Error Rates for all possible two decision rule models. There are now $N = 17$ potential models (across both nodes). *Convince yourself that this is true.*

We then choose the model that has the lowest Model Error Rate and evaluate it relative to the Model Error Rate from the "best" one decision rule model. Again, we will use the model evaluation criterion based on reduction of Model Error Rate to determine whether we will adopt the model with two decision rules (three terminal nodes) or the model with one decision rule (two terminal nodes).

10. Use the Google Sheets tool to evaluate each of the potential models with two decision rules. Which set of rules leads to the model that has the largest decrease in Model Error Rate from the model with one decision rule?

11. Based on your earlier criterion, will you adopt the “best” model with one decision rule model or the “best” model with two decision rules? Explain.

Continue to repeat this process until you can no longer decrease the Model Error Rate more than the evaluation criterion.

12. What is the Model Error Rate for your final adopted model?

13. Draw the classification tree for your final adopted model.

Model Evaluation: Evaluate Your Model Using the Test Data

While we can compute the Model Error Rate on the training data, this (as we saw earlier) can be quite misleading. For a more accurate picture of how well your model performs, it is better to compute the Model Error Rate on a test set of data that we have not used for training the model. This will give us a better sense for how the model will actually perform on new data.

14. Use your final adopted model to compute the average Model Error Rate across the 10 test sets of data. Report that value.

15. Based on the Model Error Rate you computed for the training data and the average Model Error Rate you computed for the 10 tests datasets, was your final adopted model overfitted to the training data? Explain.