

Branching Out: Using Decision Trees to Inform Education Decisions

Regional
Educational
Laboratory
Appalachia

At SRI International

REL 2022-133
U.S. DEPARTMENT OF EDUCATION

A Publication of the National Center for Education Evaluation and Regional Assistance at IES



Branching Out: Using Decision Trees To Inform Education Decisions

Neil Seftor, Lisa Shannon, Stephanie Wilkerson, and Mary Klute

December 2021

Classification and Regression Tree (CART) analysis is a statistical modeling approach that uses quantitative data to predict future outcomes by generating decision trees. CART analysis can be useful for educators to inform their decisionmaking. For example, educators can use a decision tree from a CART analysis to identify students who are most likely to benefit from additional support early—in the months and years before problems fully materialize. This guide introduces CART analysis as an approach that allows data analysts to generate actionable analytic results that can inform educators' decisions about the allocation of extra supports for students. Data analysts with intermediate statistical software programming experience can use the guide to learn how to conduct a CART analysis and support research directors in local and state education agencies and other educators in applying the results. Research directors can use the guide to learn how results of CART analyses can inform education decisions.

CONTENTS

| | |
|---|--------------|
| An Introduction to CART Analysis | 1 |
| Overview of the guide | 3 |
| Preparing for CART analysis with data for prior and current cohorts of students | 5 |
| Developing the CART decision tree using data for a prior cohort of students | 6 |
| Applying the CART decision tree to data for the current cohort of students and analyzing results | 9 |
| Using CART results to inform education decisions | 12 |
| Advantages and limitations of CART analysis | 14 |
| Appendix A: Conducting a CART Analysis..... | A-1 |
| CART framework | A-1 |
| CART software requirements | A-2 |
| Stage 1: Prepare the data..... | A-3 |
| <i>Select and process data for a prior cohort of students.....</i> | A-3 |
| <i>Partition the data into training and testing data</i> | A-5 |
| Stage 2: Develop the model..... | A-6 |
| <i>Train and tune the CART analysis to determine the optimal model</i> | A-10 |
| Stage 3: Analyze the results..... | A-17 |
| <i>Consider options.....</i> | A-17 |
| <i>Choose and implement a policy</i> | A-21 |
| Appendix B: Performance Measures..... | B-1 |
| Appendix C: Customization | C-1 |
| Glossary of Terms | Glo-1 |
| References | Ref-1 |

EXHIBITS

| | |
|---|------|
| Exhibit 1. An example of using CART analysis | 1 |
| Exhibit 2. CART analysis splits students into groups that form a decision tree..... | 2 |
| Exhibit 3. CART analysis uses data with the same set of characteristics for prior and current cohorts of students | 5 |
| Exhibit 4. CART analysis creates a decision tree to split students into groups, represented by the terminal nodes..... | 7 |
| Exhibit 5. The analyst identifies 10 options for providing the intervention to different groups of students | 10 |
| Exhibit 6. Decision tree to identify current kindergarten students for intervention | 13 |
| Exhibit A1. CART analysis processes by stage | A-1 |
| Exhibit A2. Install and load R packages | A-3 |
| Exhibit A3. Load data and assign variables | A-4 |
| Exhibit A4. Variable names and details for ECLS-K data | A-4 |
| Exhibit A5. Partition the data into training and testing data | A-5 |
| Exhibit A6. Fully grown tree with complexity parameter of zero | A-7 |
| Exhibit A7. Partitioning data and k-fold cross-validation | A-8 |
| Exhibit A8. Model error and tree size for different values of the complexity parameter..... | A-10 |
| Exhibit A9. Train and tune the CART analysis to determine the optimal model | A-12 |
| Exhibit A10. Output from code used to train and tune the CART analysis to determine the optimal model | A-13 |
| Exhibit A11. Print the nodes and decision rules from the optimal model | A-14 |
| Exhibit A12. Plot the nodes and decision rules from the optimal model | A-15 |
| Exhibit A13. Predict probabilities using the model and plot the ROC curve | A-18 |
| Exhibit A14. Extract results to inform options | A-19 |
| Exhibit A15. Implications of providing the intervention to different groups of students | A-20 |
| Exhibit A16. Plot the final decision tree for identifying students for intervention | A-22 |
| Exhibit B1. Classification matrix and related measures used for evaluating classification models ... | B-1 |
| Exhibit B2. Receiver operating characteristic (ROC) curve | B-3 |
| Exhibit C1. Outline of the tuning process with additional customizations | C-1 |
| Exhibit C2. CART analysis with additional customizations..... | C-2 |

AN INTRODUCTION TO CART ANALYSIS

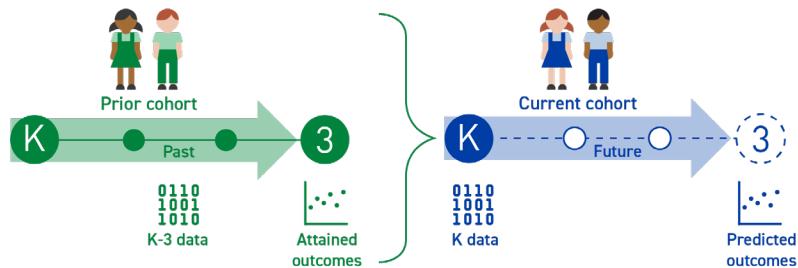
Educators often face the challenge of identifying students early who might benefit from a targeted intervention to improve their likelihood of experiencing positive academic outcomes in an area of need (Lakkaraju et al., 2015). To intervene early with students who might benefit from extra support, educators need the ability to distinguish them based on data available in the months and years before issues fully materialize.

Classification and Regression Tree (CART) analysis is a statistical modeling approach that uses prior quantitative data to predict future outcomes. CART analyses can be used to answer a wide variety of research questions for both binary and continuous outcomes. For example, CART can be used to predict which people are most likely to buy a particular product, a binary outcome. It can also be used to examine how characteristics of streams predict the size of the population of an endangered fish, a continuous outcome. This report focuses on the application of CART in educational contexts. Educators and data analysts can use CART for a wide variety of purposes, such as predicting which teachers are most likely to leave the teaching profession, which schools are most at risk of not meeting accountability expectations, or which students are at the greatest risk of dropping out of school.

This report focuses on a specific use of CART in educational contexts: to help educators identify students who are at risk of not achieving desired learning outcomes. In this use case, CART analysis can use data for a prior cohort of students to identify relationships between a set of student characteristics—such as proficiency on kindergarten reading and math assessments—and a binary student outcome of interest—such as proficiency on a grade 3 state math assessment (exhibit 1). Educators can use the results of the CART analysis to predict outcomes for a different group of students with similar characteristics, such as the current cohort (Gomes & Almedia, 2017; Polat, 2018; Quadril & Kalyankar, 2010). In prior research, CART analysis has predicted which students were at risk of struggling with reading comprehension (Koon & Petscher, 2015, 2016; Koon et al., 2014), not meeting college-readiness benchmarks (Koon & Davis, 2019), and dropping out of college (Dekker et al., 2009).

Exhibit 1. An example of using CART analysis

CART uses **K-3 data** from a **prior cohort** to predict **grade 3 outcomes** for the **current cohort** of **kindergarten** students.

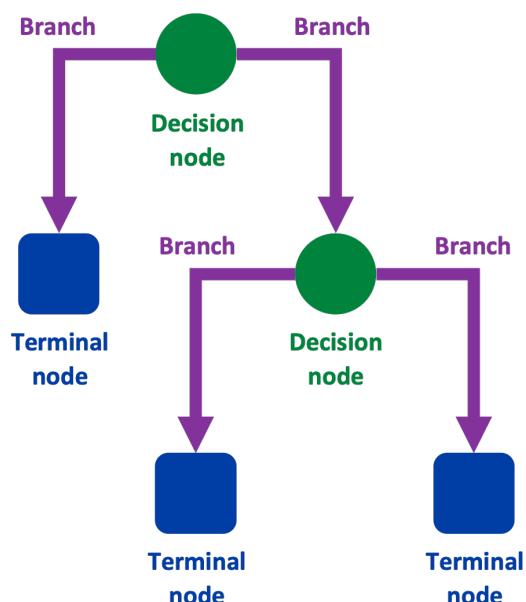


Note: K = kindergarten, 3 = grade 3.

Source: Authors' creation.

Using data for a prior cohort of students, CART analysis splits students into groups that form a [decision tree](#). Each group includes students with similar outcomes based on a set of characteristics, such as assessment scores. A CART decision tree illustrates the splitting of students into groups with a set of [nodes](#) connected by [branches](#) (exhibit 2). A [decision node](#) (green circle) is a point at which students are split based on the value of one of the characteristics. A [terminal node](#) (blue square) is an ending point with no more splits. The terminal nodes in a decision tree visually represent the final set of mutually exclusive student groups for the prior cohort. For example, educators might use a CART analysis to split students from a prior cohort into groups using assessment data from kindergarten and grade 3. The analysis would create a decision tree that visually represents a set of mutually exclusive groups of students with similar assessment scores in kindergarten and similar outcomes in grade 3.

Exhibit 2. CART analysis splits students into groups that form a decision tree



Source: Authors' creation.

Educators can apply the same decision tree to data on the current cohort of students to inform their decisions about which students might benefit from additional support. Applying the same decision tree to data on the current cohort of students will sort them into the same set of groups into which the prior cohort was split. The outcomes for students in each group in the data for the prior cohort provide an estimate of the outcome one might expect to see for students from the current cohort with the same characteristics. This estimate of the outcome is called the predicted outcome. Educators can use these predicted outcomes for the current or future cohorts of students to inform decisions about interventions to support the same educational outcomes they examined in the CART analysis.

It is important to note that CART analysis does not yield perfect predictions about student outcomes. Therefore, when using CART analysis to identify students to receive a targeted intervention, educators should consider other information they have about students, such as information that students share with teachers, knowledge of available resources, and educational priorities.

Overview of the guide

This guide introduces CART analysis as a statistical modeling approach that allows research directors at local and state education agencies, other educators, and data analysts to generate actionable findings to inform education decisions.

This guide has two objectives: 1) to increase research directors' and data analysts' awareness and understanding of CART analysis as one method to identify students who might benefit from early intervention, and 2) to guide data analysts on how to conduct CART analysis.

To meet the first objective, the guide delineates how research directors and data analysts can use CART analysis to support educators in addressing critical questions in educational settings. The guide describes:

- The kinds of data CART analyses require,
- How the method creates a decision tree based on data for one or more prior cohorts of students,
- How analysts can apply the decision tree to data for the current cohort of students, and
- How research directors and data analysts can support educators in using CART results to inform decisions about the allocation of extra supports to students who may need them.

Each section of the guide presents a general description of an aspect of CART analysis followed by an example to illustrate its application (see green boxes with school icon ). The example focuses on the problem of low grade 3 math proficiency in the hypothetical district of Eduphonia. The guide concludes with a discussion of some advantages and limitations of CART analysis.



Problem: One-quarter of Eduphonia students score Below Proficient on grade 3 math assessments

Research shows that mastery of mathematical knowledge from preK through elementary grades is a strong predictor of later success. For example, kindergartners' math skills in pattern recognition, measurement, and advanced number understanding can predict math achievement in grade 8, and early mathematical skills have predicted reading and science achievement as well as grade retention (Claessens & Engel, 2013). Likewise, students' knowledge of fractions and division in elementary school can predict their high school achievement in algebra, regardless of other mathematical ability, working memory, family income, or education (Siegler et al., 2012). Persistent problems with math are the best predictor of failure to graduate from high school or enter college (Duncan & Magnuson, 2011). The math journey begins early, and young children who have more opportunities to develop and apply their mathematical knowledge are more likely to succeed in school and life (Frye et al., 2013).

Educators in the district of Eduphonia are concerned that students who fall behind early in math may have difficulty recovering. In recent years, one-quarter of students in Eduphonia have scored Below Proficient on the state math assessment at the end of grade 3. Educators expect that, without intervention, 250 of the 1,000 current kindergarten students may score Below Proficient on the state math assessment at the end of grade 3. Leaders want to identify the students who are at risk of failing to attain proficiency in grade 3 math when they are younger. If educators can identify those students early enough, it may be possible to introduce targeted interventions to prevent them from falling behind. Eduphonia's data analyst recognizes that CART analysis is well suited for this problem.

To meet the second objective, to guide data analysts on how to conduct CART analysis, the guide includes several technical appendixes. Specifically, [appendix A](#) provides instructions for replicating the analyses described in the Eduphonia example, including annotated R programming code and associated output. It uses publicly available data, which allows data analysts to replicate the analysis and findings, learn about ways to customize the [model](#), and adapt the code for their own use. The appendix provides details on how the CART analysis [algorithm](#) works and analytic issues to consider when creating decision trees. For data analysts who want to delve deeper, [appendix B](#) describes a set of metrics for evaluating the performance of the CART analysis and [appendix C](#) offers sample code for additional approaches to customize the CART analysis. Readers can find definitions of technical terms in a glossary to support their use of this guide. Each term is hyperlinked at its first mention.

The amount of time data analysts need to complete a CART analysis will vary depending upon several factors. These factors include the number of different datasets that need to be accessed, the extent to which the data are clean and ready for analysis, and the data analysts' familiarity with the analysis software.

Preparing for CART analysis with data for prior and current cohorts of students

Before conducting a CART analysis, data analysts need to access data and software. To estimate a CART model, data analysts need characteristics for the prior cohort of students that are linked over time with the outcome. To use the CART analysis results to inform decisionmaking, analysts will need the same characteristics for the current cohort. Because CART analysis uses data on previous cohorts to predict relationships for current cohorts, the datasets for both cohorts will need to include the same set of characteristics used to predict student outcomes. For example, if a data analyst is interested in using kindergarten assessment scores as characteristics in the analysis, these data will need to be available in the data for both the prior and current cohorts of students.

Data analysts will also need to use a statistical software package that includes procedures to conduct CART analysis; this guide provides examples using R, an open-source statistical software package. Data analysts should familiarize themselves with the software to prepare the data and understand the options that each procedure provides.



Eduphonia data analysts identify data needed for the CART analysis

Eduphonia has a longitudinal data system that contains the student-level information linked over time necessary for CART analysis (exhibit 3).

For the prior cohort of students (students who have already completed grade 3), the CART analysis will use assessment scores from the beginning of kindergarten and the proficiency level from the state math assessment at the end of grade 3. The CART analysis will use the data for the prior cohort to identify relationships between the set of characteristics and the outcome of interest.

For the 1,000 Eduphonia students currently in kindergarten, the data include scores from their kindergarten assessments. The CART analysis will use these characteristics from the current cohort of students to predict their probability of scoring Below Proficient on the state math assessment when they complete grade 3.

Exhibit 3. CART analysis uses data with the same set of characteristics for prior and current cohorts of students

| | Prior cohort | Current cohort |
|---------------------|--|---|
| Population | Students who have completed grade 3 | Students currently in kindergarten |
| Characteristics | Kindergarten subject proficiency scores for math and reading; kindergarten teachers' ratings of students' academic achievement in math, reading, and general knowledge | |
| Outcome of interest | Indicator of scoring Below Proficient on the state math assessment at the end of grade 3 | The probability of current students scoring Below Proficient on the state math assessment at the end of grade 3 |

Developing the CART decision tree using data for a prior cohort of students

In developing the decision tree, CART identifies relationships between the set of student characteristics and a student outcome of interest, which is a binary outcome in the Eduphonnia example. The relationship between the characteristics and the student outcome is expressed in terms of predictive accuracy. As a baseline, the CART analysis determines the [accuracy](#) of making predictions using only the student outcome data. In the Eduphonnia example, 75 percent of prior cohort students scored Proficient or above. Because the majority of students scored Proficient or above, the best prediction for any student from the prior cohort would be that they would score Proficient or above. This approach leads to a correct prediction for 75 percent of students and an incorrect prediction for 25 percent of students. The goal of the CART analysis is to improve the predictive accuracy.

To improve predictive accuracy, the CART analysis asks whether there is a characteristic it can use to split the students into two groups, one that is predicted to score Proficient or above and one that is predicted to score Below Proficient, such that the rate of correct prediction is higher than 75 percent. If there are multiple ways to split the data that can improve the rate of correct prediction, the analysis identifies the split that leads to the best improvement in that rate. That split becomes the first decision node. The analysis repeats this process to create additional decision nodes that each use a value of one characteristic to split the students who reached that node into two groups that again improve the overall predictive accuracy.

If the CART analysis continued splitting students until it could make no more splits, it would create a decision tree that would likely align too closely with the data from the prior cohort (Bramer, 2007). Such a tree would have many [decision rules](#) created to deal with idiosyncrasies of the students in the prior cohort. Generally, the resulting decision tree would not be as useful for predicting what will happen in the current cohort as a tree with fewer nodes.

CART analysis has a feature called a [stopping rule](#), which helps to balance the goal of making better predictions for the prior cohort with the goal of making accurate predictions for current or future cohorts of students. The analyst specifies a value for the stopping rule, often a minimum amount of improvement in the predictive accuracy, to guard against fitting a model too closely to the data of the prior cohort. For each split, the CART analysis considers many different potential cut points for each characteristic and identifies the combination of characteristic and cut point that leads to the largest improvement in prediction error. Provided that reduction in prediction error satisfies the stopping rule, CART adds the split to the decision tree and the process continues to try to identify the next split. At some point, the CART analysis will not be able to identify a way to split a group that results in an improvement large enough to satisfy the stopping rule, so the group will remain together in a terminal node, with no further splits.

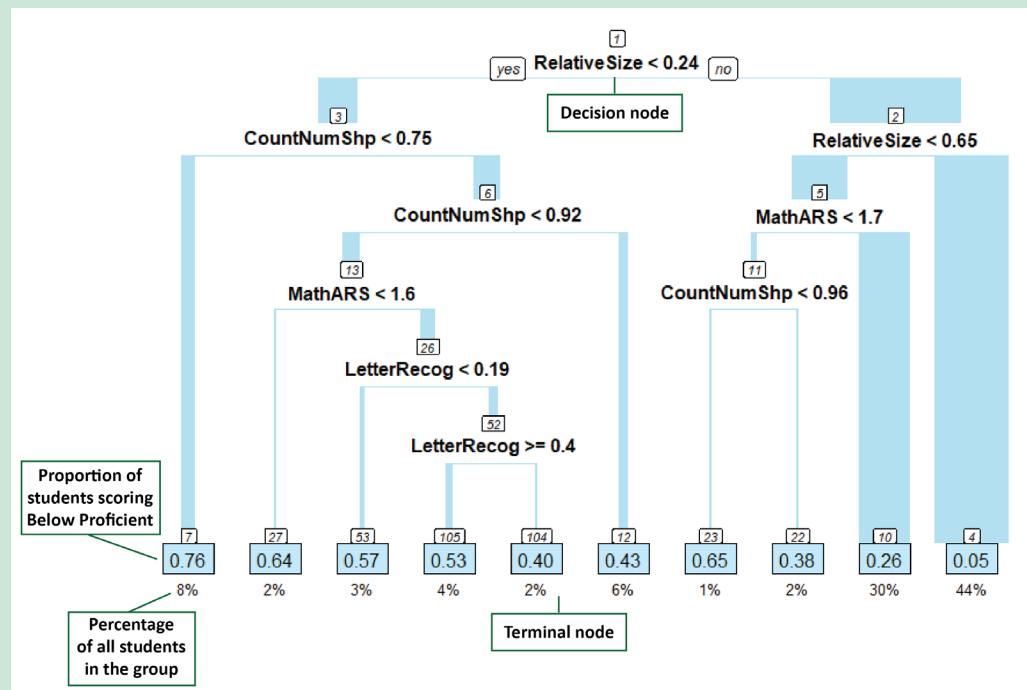
The CART analysis output displays two pieces of information about each terminal node that can be useful when considering which students might benefit from an intervention. Specifically, the output presents the share of students in that group who did not achieve the target outcome and the percentage of all students who end up in that node.



Decision tree splits students from the prior cohort based on kindergarten measures

In Eduphonia, the data analyst enters data for the prior cohort of students into the statistical software and uses the CART analysis to split students into groups using associations between their kindergarten characteristics and an indicator for whether they were Below Proficient on the state math assessment at the end of grade 3. The result is a decision tree that splits students based on four variables in the dataset from the fall of kindergarten: RelativeSize is a proficiency score for the early mathematics concept of relative size; CountNumShp is a proficiency score for the early mathematics concepts of count, number, and shape; LetterRecog is a proficiency score for the early literacy concept of letter recognition; and MathARS is a mathematical thinking academic rating scale. The statistical software package R created the decision tree in exhibit 4.

Exhibit 4. CART analysis creates a decision tree to split students into groups, represented by the terminal nodes



Note: The percentages of all students in each group do not sum to 100 because of rounding.

Source: Authors' analyses of U.S. Department of Education's Early Childhood Longitudinal Study, Kindergarten Class of 1998/99 (ECLS-K) data.

Each decision rule splits students based on their value for a particular characteristic. For example, the first split in the tree is based on whether a student's value for `RelativeSize` is less than 0.24. CART analysis writes all decision rules so that students with a value of Yes split to the left and students with a value of No split to the right. For the first decision rule, the result is Yes if a student has a value of `RelativeSize` that is less than 0.24. The left branch includes all students with values of `RelativeSize` less than 0.24.

The width of each vertical branch represents the proportion of students who follow it. At the first split, the vertical bar on the right side depicts students with `RelativeSize` of at least 0.24. The vertical bar on the left depicts students who have a value of `RelativeSize` less than 0.24. The vertical bar on the right is about three times as wide as the vertical bar on the left, indicating that three-quarters of students have a value of `RelativeSize` of at least 0.24 and about one-quarter of students have values of `RelativeSize` less than 0.24.

The splits continue to a set of mutually exclusive terminal nodes at the bottom of the figure that contain all students who started at the top of the tree. These terminal nodes vary in size and in the extent to which the students in the group scored Below Proficient on the state math assessment at the end of grade 3. The white boxes show the numerical label for each node. The number inside the blue box for each terminal node is the share of kindergarten students in that group who scored Below Proficient on the state math assessment at the end of grade 3, and the number under the terminal node is the percentage of all kindergarten students who end up in that group. For example, the leftmost terminal node (terminal node 7) contains 8 percent of all students from the prior cohort. Of those students, 76 percent scored Below Proficient on the state math assessment at the end of grade 3.

The CART analysis splits the data from the prior cohort based on a set of characteristics and the student outcome. Educators can use information from the analysis to inform predictions about what will happen for students in the current cohort. For predictions based on the prior cohort to apply to the current cohort, the relationships between the characteristics and outcomes must be similar for the two groups. Otherwise, characteristics that predicted outcomes well in the past might no longer be good predictors. Additionally, the predictions will be more accurate if the distributions of characteristics in the current cohort are similar to the distributions of characteristics in the prior cohort. When the distributions of characteristics for the current and prior cohorts are very different, it may imply that the prior cohort is quite dissimilar to the current cohort, which would be less useful for making predictions. For example, if the prior cohort included predominately gifted students, it would be less useful for making predictions about a current cohort that included all students in a district.

Together, similar relationships between student characteristics and student outcomes and similar distributions of characteristics for prior and current cohorts allow the results from the analysis of prior-cohort data to be informative for decisions related to the current cohort of students.

Applying the CART decision tree to data for the current cohort of students and analyzing results

Once the CART analysis creates a decision tree using data from a prior cohort of students, data analysts can apply its decision rules to predict outcomes for the current cohort of students. As described earlier, the CART analysis decision tree includes a set of decision rules that split previous students into groups based on a set of characteristics measured during kindergarten and reported on the average grade 3 outcome for each group. Analysts can apply the same set of rules to data for the current cohort of students to split them into groups based on characteristics measured during kindergarten. This process generates predictions about the future grade 3 outcomes for current students.

After using CART to generate predicted outcomes for current students, research directors and data analysts can support educators in considering various options for each group of students and the implications of choosing each option. This section of the guide describes how educators could use the results of the CART analysis to make decisions about the groups of students the analysis identifies. In practice, educators may consider other information they have about individual students when making decisions about which students receive the intervention. When considering options based on the results of the CART analysis, the question for educators becomes: “Given the split of students into groups of the sizes and predicted outcomes resulting from the CART analysis, what action do we want to take for each group?”



Educators consider the implications of intervention provision

The CART analysis of Eduphonia’s data for the prior cohort of students used kindergarten measures to split previous students into groups. The analysis ended up splitting the prior cohort into 10 groups. The 10 terminal nodes in exhibit 4 represent these 10 groups. The groups vary in how likely the students were to subsequently score Below Proficient on the state math assessment at the end of grade 3, ranging from 5 percent to 76 percent. Educators want to use this information to help identify current students who could benefit from an evidence-based intervention. They start by looking at the students most likely to benefit.

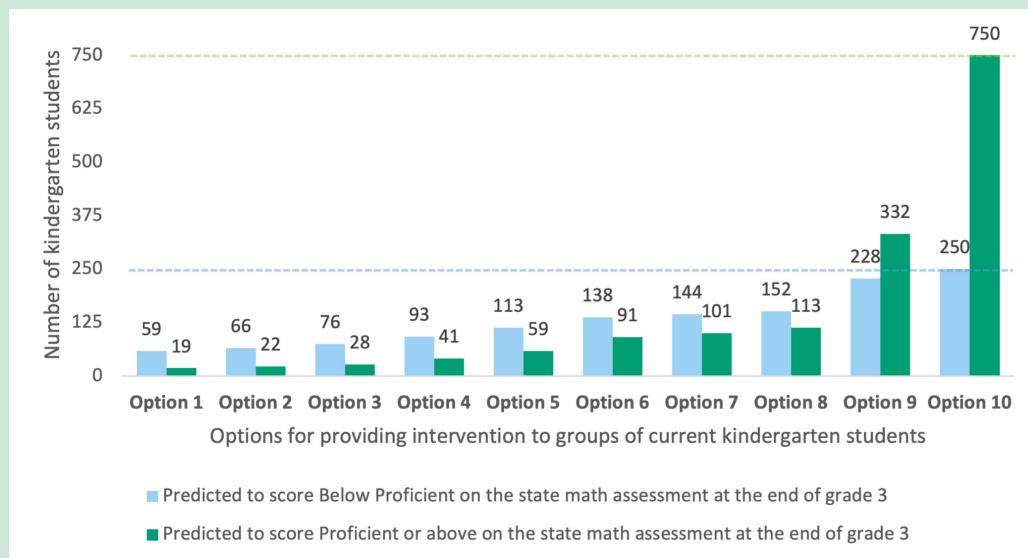
Consider the group of students with the highest likelihood of scoring Below Proficient on the state math assessment at the end of grade 3. Across the groups, the highest likelihood is 76 percent in the leftmost group (see exhibit 4), which contains 8 percent of all students from the prior cohort. The CART analysis uses the average outcome for students in that group from the prior cohort to predict outcomes for students from the current cohort who are in the same group based on their characteristics.

After the CART analysis uses data for a prior cohort of students to split that cohort into groups, data analysts can apply the same decision tree rules to the 1,000 students currently in kindergarten in Eduphonia to help inform options about who should receive early intervention. Using those

decision tree rules, the data analyst can sort the current cohort of students into the same 10 terminal nodes.

The analyst identifies 10 options for the current cohort of students (exhibit 5). The analyst orders the options for providing the intervention, from option 1 (providing the intervention only to students in the group with the highest likelihood of scoring Below Proficient on the state math assessment at the end of grade 3) to option 10 (providing the intervention to all kindergarten students). Each option adds the group with the next-highest likelihood of scoring Below Proficient to the group(s) included in the previous option. For each option, the pair of bars represents the number of students who could receive the kindergarten intervention under that option. The bar on the left in each pair (in blue) presents the number of students predicted to score Below Proficient on the state math assessment at the end of grade 3. These are the students under that option that educators would expect to benefit from the intervention. The column on the right in each pair (in green) presents the number of students predicted to score Proficient or above. Educators would not expect these students to benefit from the intervention.

Exhibit 5. The analyst identifies 10 options for providing the intervention to different groups of students



Note: Options are cumulative, such that each option adds students to the students included in the previous option. For example, the 66 students predicted to score Below Proficient in option 2 include the 59 students predicted to score Below Proficient in option 1.

Source: Authors' analyses of U.S. Department of Education's Early Childhood Longitudinal Study, Kindergarten Class of 1998/99 (ECLS-K) data.

Thus, for option 1, the CART analysis identifies 78 students with the highest probability of scoring Below Proficient on the state math assessment at the end of grade 3. The decision tree in exhibit 4 predicts that 76 percent of these students (59 students) will score Below Proficient and 19 students will not. If the district provides the intervention to students in this group, 59 students who are expected to increase their chances of scoring Proficient as a result of the intervention will receive it. In addition, 19 students will receive the intervention even though they would likely score Proficient or above even without it.

For option 2, the CART analysis identifies students with the next-highest likelihood of scoring Below Proficient on the state math assessment at the end of grade 3. This group has a 65 percent likelihood of scoring Below Proficient and contains 1 percent of all students. Thus, when the data analyst applies the decision tree rules to the current cohort of students, the CART analysis identifies 10 additional students: 7 who are predicted to score Below Proficient and 3 who are not. If the district provides the intervention to the 10 students in this group along with the 78 in the group from option 1, it will provide the intervention to 66 students expected to increase their chances of scoring Proficient due to the intervention and 22 expected to score Proficient or above without the intervention.

To generate the full set of options, the data analyst repeats this process by incrementally adding the group with the next-highest likelihood of scoring Below Proficient on the state math assessment at the end of grade 3. Finally, the data analyst creates a figure that illustrates the implications of providing the intervention to different groups of current kindergarten students (exhibit 5).

Historically, one-quarter of Eduphonia kindergarten students have scored Below Proficient on the state math assessment at the end of grade 3. Therefore, without intervention, educators expect 250 of the 1,000 current kindergarten students in Eduphonia to score Below Proficient on the state math assessment at the end of grade 3 (blue dashed line) and 750 to score Proficient or above (green dashed line).

Providing the intervention to students in additional groups would enable the district to provide the intervention to more students who may benefit from it, but only by also providing it to an increasing number of students who have a low likelihood of needing it. The only option in exhibit 5 that provides the intervention to all 250 students whom the district expects to score Below Proficient at the end of grade 3 is option 10, which also provides the intervention to the other 750 students.

You can estimate a CART model using a set of prior cohorts—for example, those who attended kindergarten between 2015 and 2017—and then use those results to predict outcomes of a number of later cohorts—for example, those who attend kindergarten between 2018 and 2021—as long as you believe the relationships between the characteristics and outcomes have remained fairly stable. You can choose to provide the intervention to students in a different set of terminal nodes for each of the later cohorts, depending on the percentage of students in each terminal node and available resources each year.

Using CART results to inform education decisions

After the CART analysis has generated the predicted outcomes for groups of students with similar characteristics, educators should review the results with the data analyst or research director and decide which of the groups will receive the intervention.¹ Educators will need to consider other contextual information, such as knowledge of available resources and education priorities, in making their final decision. For example, they might not have enough resources to provide the intervention to all students who could potentially benefit. In addition to the high level of resources required, over-provision may misdirect the learning time of students already expected to score Proficient or above, potentially preventing them from engaging in other activities that may enable them to continue progressing academically.



Educators choose groups of students to receive the intervention

The Eduphonia educators want to provide early intervention to students likely to score Below Proficient on the state math assessment at the end of grade 3. Their resources are limited, and they do not want to burden students who are likely to attain proficiency without it.

After reviewing their options, the Eduphonia educators choose option 8 (see exhibit 5), which allows them to provide the intervention to:

- More than 60 percent (152 out of 250) of students predicted to score Below Proficient on the state math assessment at the end of grade 3; and
- Only 15 percent (113 out of 750) of students predicted to score at or above Proficient on the state math assessment at the end of grade 3.

They believe this strikes the right balance of serving most students likely to benefit while limiting the use of resources on those likely not to need the intervention. Though the next option, option 9, would have allowed educators to serve more than 90 percent of students likely to benefit from the intervention, it would have required serving more than half the students in the district, an expensive proposition.

Once educators have made their decision, the data analyst can update the decision tree with indicators that reflect which groups of students will receive the intervention. Educators at the schools can then apply the decision tree rules to data for the current cohort of students to identify which individual students will receive the intervention and which students will not.

1. This section describes how to use the results of the CART analysis to make decisions about whole groups of students in a systematic way. In practice, educators may want to consider other information they have about individual students, beyond group membership, when deciding which students should receive the intervention. However, that approach reduces transparency and consistency and may introduce bias into the process.



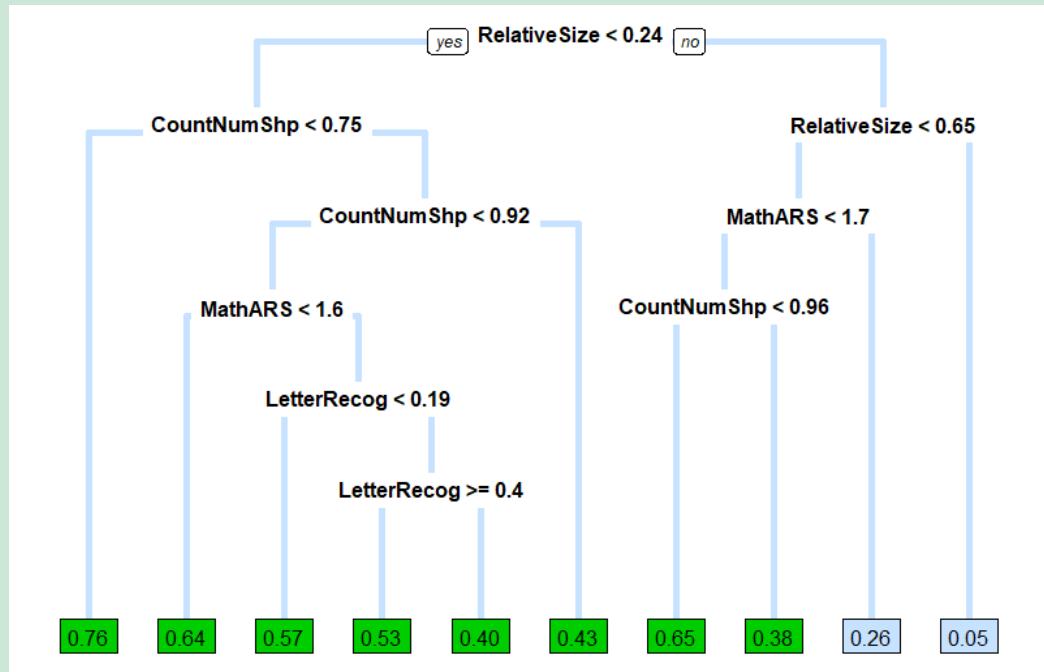
Identifying current kindergarten students for intervention using the decision tree

After the Eduphonnia educators choose option 8, the data analyst adds information about which groups will receive the intervention to the decision tree (exhibit 6). Educators will provide the intervention to eight of the 10 groups of students, indicated by the green boxes. The district will not provide the intervention to students who meet the criteria represented in the blue boxes (two boxes on the far right of exhibit 6).

Educators can use this decision tree to create guidelines for determining which students should receive the intervention. In this case, the decision rules illustrate that a student will receive the intervention if they have:

- A `RelativeSize` score of less than 0.24; or
- A `RelativeSize` score between 0.24 and 0.65 and a `MathARS` score of less than 1.7.

Exhibit 6. Decision tree to identify current kindergarten students for intervention



Note: The numbers in the boxes at the bottom of the figure indicate the proportion of students in each node scoring Below Proficient.

Source: Authors' analyses of U.S. Department of Education's Early Childhood Longitudinal Study, Kindergarten Class of 1998/99 (ECLS-K) data.

Regardless of the additional splits, all of the groups that have students with `RelativeSize` scores less than 0.24 end up in groups that are shaded green at the bottom of exhibit 6, indicating that they will receive the intervention. The green boxes in exhibit 6 also include two groups of students who scored greater than or equal to 0.24 on `RelativeSize`. All of these students also scored less than 0.65 on `RelativeSize` and less than 1.7 on `MathARS`, and are in groups that will receive the intervention, as indicated by the green shaded boxes.

Advantages and limitations of CART analysis

CART analysis is both a useful approach for educators to examine practical issues of interest and a robust analytical method without many of the constraints imposed by other methods. Unusually small or large values of characteristics, such as one student's reading assessment score being much higher than others', do not affect the results of a CART analysis (Gordon, 2013; Song & Lu, 2015). CART analysis does not make assumptions about the distributions of variables, such as requiring that an assessment score be normally distributed. In a bell-shaped normal distribution, most students would be predicted to have, for example, an assessment score near the average and fewer students would score much lower or higher than average.² CART analysis can also accommodate characteristics that are closely related, such as two different kindergarten reading measures, as well as characteristics that may have complex interactions without having to specify which interactions to consider.

Along with its practical applications, studies have demonstrated that CART analysis results are consistent with those of other statistical procedures, such as logistic regression (Dekker et al., 2009; Koon & Petscher, 2015; Polat, 2018). By identifying relationships between a set of characteristics and the outcome in the data, CART analysis provides information to guide decisions.

One limitation of CART analysis is that decision trees created with different samples of students can cause the structure of the tree to change dramatically (James et al., 2015). As described in [appendix A](#), analysts conduct the CART analysis on a subset of the data for the prior cohort. Using an even slightly different subset of data to identify relationships can result in a decision tree that may have different rules and use different characteristics to create splits, which could result in identifying different students for the intervention. Data analysts can deal with this challenge using the methods described in appendix A.

Another limitation is that if the CART analysis allows for data to be split into too many small groups, it runs the risk of being too closely aligned with the specific data used to create the decision tree (Bramer, 2007). CART analysis uses a stopping rule to address this issue and avoid trees with many decision rules created to deal with idiosyncrasies in the original data (see box A2 in appendix A for more information about this potential issue, called "[overfitting](#)").³ When this happens, the predictions for other data are not as accurate. See appendix A for more details on commonly used methods for addressing issues of decision tree instability and overly close alignment to specific data.

CART analysis is not an all-purpose method. CART analysis is particularly well suited for addressing education problems that require identifying students who might be at risk of an adverse outcome, as illustrated by the Eduphonia example. However, CART analysis is inappropriate for informing some types of decisions. It is not appropriate for examining

2. Some other methods, such as regression-based approaches, require assumptions about the distributions of variables, and they do not perform well with large numbers of categorical variables or variables that are highly correlated with each other.

3. This issue also applies to many other methods used to create early warning systems.

An Introduction to CART Analysis

the effectiveness of an intervention because its findings are not causal. In addition, CART analysis would not be able to inform decisions about how or why an outcome occurs. It also may not be as effective as some other methods for identifying which predictors best predict an outcome.

As with other analytic methods, external factors—such as data availability, data quality, and the types of relationships in the data—affect the extent to which CART analysis can inform decisions.

“Our philosophy in data analysis is to look at the data from a number of different viewpoints. Binary trees give an interesting and often illuminating way of looking at the data in classification or regression problems. They should not be used to the exclusion of other methods. We do not claim that they are always better. Like any tool, its greatest benefit lies in its intelligent and sensible application.”

—Breiman, Friedman, Olshen, and Stone (1984)

APPENDIX A: CONDUCTING A CART ANALYSIS

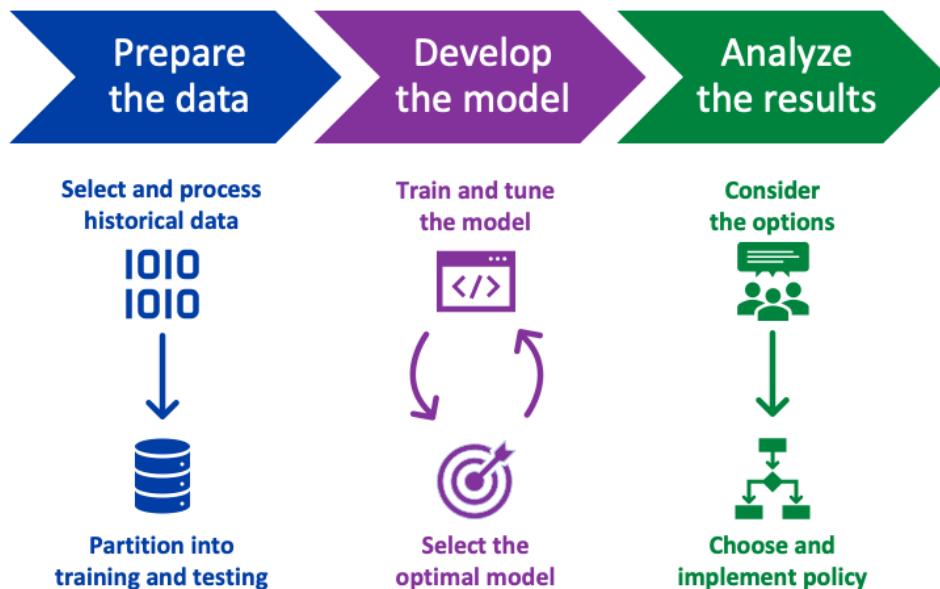
Data analysts can use this appendix to gain basic knowledge of programming in R and replicate the CART analysis presented through the Eduphonnia example in the guide. This appendix provides several types of information, delineated by icon and color, to help data analysts navigate the replication process.

-  Light blue boxes contain code, including brown italic text annotations on lines starting with `#`. References to code in the body of the appendix are highlighted blue.
-  Gray boxes contain output from the code, including graphs.
-  Orange boxes contain technical explanations of issues related to the CART analysis. The information is for interested readers and is not required to implement the CART analysis.

CART framework

CART analysis relies on a predictive algorithm that uses a model to identify the relationships between a set of characteristics and an outcome of interest. When conducting a CART analysis, work proceeds in three broad stages—preparing the data, developing the model, and analyzing the results (exhibit A1). This appendix describes each of these stages in detail.

Exhibit A1. CART analysis processes by stage



Source: Authors' creation.

Prepare the data. In the prepare the data stage, data analysts must obtain data for a prior cohort that contain the set of characteristics and outcome of interest; select data observations and variables to include in the analysis;⁴ process the data for use in the analysis, including cleaning the data, dealing with missing values, and transforming or creating new variables; and partition it into training data and testing data for use in separate parts of the analysis.

Develop the model. In the develop the model stage, the CART analysis runs an algorithm repeatedly as part of a training and tuning process to identify the optimal model. When training the model, the CART analysis runs the algorithm on the training data to generate a set of decision rules. In tuning the model, the data analyst changes the parameters of the algorithm and trains the model for each set of parameters to identify the optimal model. The CART analysis evaluates the model at multiple points. As the algorithm constructs the decision tree, it considers many possible ways to split the groups. In each step of the process, it identifies all of the possible splits, with each split based on a single variable. The algorithm then uses an internal metric to identify the split that leads to the greatest improvement in the predictive accuracy to identify the best way to split the data. Then the process repeats and the CART analysis again considers many possible ways to split the groups. The CART analysis will consider all of the variables when determining how to split the data. Because it only makes the best of all possible splits, CART may not use all of the variables included in the model in the resulting analysis or the final tree. The tuning process uses a different metric to evaluate the overall fit of the model from each training. The CART analysis selects the model with the best overall fit as the optimal model.

Analyze the results. When analyzing the results, educators create a set of options derived from the final model, consider the implications of each option, select an option, and then implement the decision using the model on data for the current cohort of students.

CART software requirements

To replicate the analyses described in this guide, users will need R (version 3.6 or above) and RStudio, along with a basic understanding of both.⁵ Users may need to consult other resources for assistance with the software. Several packages in R implement CART analysis. This example uses the rpart package, which is included in base R (exhibit A2).⁶

4. For more information on factors to consider when selecting variables for a CART analysis, see Lemon et al. (2003).

5. R is a programming language and free software environment for statistical computing and graphics supported by the R Foundation for Statistical Computing. You can download R from the [Comprehensive R Archive Network \(CRAN\)](#). RStudio is an integrated development environment for R. You can download it directly from [RStudio](#).

6. The rpart package implements CART analysis as described by Breiman and colleagues (1984). See the Recursive Partitioning section of CRAN Task View: Machine Learning & Statistical Learning for the full list of related packages. CRAN provides many R packages, as well as detailed help pages and vignettes. You can find additional information through the ??package, ?procedure, and vignette('package') commands in R.

Exhibit A2. Install and load R packages



```
# If these packages have not been installed previously, use install.packages() to
# install them before loading them for use in your work session.
install.packages("caret")
install.packages("rpart")
install.packages("rpart.plot")
install.packages("ROCR")

# Load the packages.
library(caret)      # Functions to streamline model training and tuning processes
library(rpart)       # Implementation of CART analysis algorithm
library(rpart.plot)  # Procedure to plot the results of rpart
library(ROCR)        # Tool for creating curves of performance measures
```

Stage 1: Prepare the data

To begin, the user selects and processes data for the prior cohort for use in the CART analysis. The data analyst then partitions the data into training data and testing data (see exhibit A1).

Select and process data for a prior cohort of students

For demonstration purposes, this analysis uses publicly available data from the U.S. Department of Education's [Early Childhood Longitudinal Study](#), Kindergarten Class of 1998/99 (ECLS-K), as the prior cohort. The ECLS-K data focus on children's early school experiences. Our analysis sample follows ECLS-K students from kindergarten through middle school, linking data across time. The data are from the National Center for Education Statistics [Online Codebook](#), which allows for selection of variables and access to documentation. Visit the [Inter-university Consortium for Political and Social Research](#) for additional information.

Users will begin by identifying the data elements necessary to answer key questions of interest. Users commonly examine the distributions of variables to identify potential issues, such as invalid values, high rates of missing data, or problematic distributions (for example, lack of variability). This section does not provide a comprehensive description of all the steps in data preparation; for such a description, please see the Toolkit for Effective Data Use created by the Harvard Strategic Data Project.

The example presented in this appendix uses a dataset that has been cleaned and prepared for this analysis. To prepare the dataset, analysts selected variables of interest from among all of the variables available in the ECLS-K data; examined patterns of missingness in the data; and transformed the continuous outcome variable, a grade 3 mathematics test score, into a dichotomous variable that indicates whether or not each student scored Below Proficient. Users can access the dataset used in this guide [here](#).

Appendix A: Conducting a CART Analysis

In this step, users will need to load the data into R and assign the variables needed to run the CART analysis. The blue box in exhibit A3 presents the R code that accomplishes this. The gray box in exhibit A3 depicts the output for the `dim` command, which provides the dimensions of the data. The first number in the gray box indicates that the dataset includes 14,374 rows, one for each student. The second number in the gray box indicates that there are 12 columns, one for each variable. Exhibit A4 presents the variable names and details.

Exhibit A3. Load data and assign variables

```
> # Use the load() command to read in the data.  
load("eclsk.rdata") # Change the file name to access your data file.  
  
# Create a copy of your data in mydata to allow you to use the remaining code more easily.  
mydata <- eclsk # Change the dataset name to access your data.  
  
# Assign the outcome to depvar, the dependent variable.  
depvar <- "AtRisk" # Change the dependent variable to your outcome of interest.  
  
# Define the set of independent variables for the analysis.  
indepvar <- c( # Change the independent variables to your characteristics.  
  "LiteracyARS", "MathARS", "GeneralARS", "LetterRecog", "BeginSounds",  
  "EndSounds", "SightWords", "CountNumShp", "RelativeSize", "OrdinalSeq",  
  "AddSubtract")  
  
# Use dim() to show the dimensions of the data.  
dim(mydata)
```

[1] 14374 12

Exhibit A4. Variable names and details for ECLS-K data

| Variable | Variable details (ECLS-K variable name) |
|--------------|---|
| LiteracyARS | Fall kindergarten mathematical thinking academic rating scale (T1RARSMA) |
| MathARS | Fall kindergarten language and literacy academic rating scale (T1RARSLI) |
| GeneralARS | Fall kindergarten general knowledge academic rating scale (T1RARSGE) |
| LetterRecog | Fall kindergarten proficiency probability score for letter recognition (C1R4RPB1) |
| BeginSounds | Fall kindergarten proficiency probability score for beginning sounds (C1R4RPB2) |
| EndSounds | Fall kindergarten proficiency probability score for ending sounds (C1R4RPB3) |
| SightWords | Fall kindergarten proficiency probability score for sight words (C1R4RPB4) |
| CountNumShp | Fall kindergarten proficiency probability score for count, number, shape (C1R4MPB1) |
| RelativeSize | Fall kindergarten proficiency probability score for relative size (C1R4MPB2) |
| OrdinalSeq | Fall kindergarten proficiency probability score for ordinality, sequence (C1R4MPB3) |
| AddSubtract | Fall kindergarten proficiency probability score for add/subtract (C1R4MPB4) |
| AtRisk | Indicator of scoring Below Proficient on state math assessment at the end of grade 3 (C5R4MTSC) |

Notes: AtRisk was created as an indicator based on whether C5R4MTSC was in the lowest quartile. Observations with a missing value for C5R4MTSC were dropped. Other missing values were replaced by variable means.

Source: U.S. Department of Education's Early Childhood Longitudinal Study, Kindergarten Class of 1998/99 (1998–2002).

Partition the data into training and testing data

After preparing the data and before any analysis takes place, users must partition the data for the prior cohort into training and testing data. The CART analysis uses training data to train and tune the model in stage 2 (see exhibit A1). The CART analysis uses testing data in stage 3 to evaluate the predictive accuracy of the final model created with the training data.

A conventional way to split a dataset between training and testing data is to randomly assign 80 percent of the observations to the training data and 20 percent to the testing data. As a result, the CART analysis uses the bulk of the data to fit the model, which should lead to greater accuracy. The CART analysis uses a smaller proportion of data to evaluate the predictive accuracy of the model.

To split the data randomly between training and testing data, users will use a procedure that randomly selects 80 percent of observations for each outcome category to ensure that the proportion of students who score Below Proficient on the state math assessment at the end of grade 3 is the same in both datasets (exhibit A5). That is, the software will randomly select 80 percent of students who scored Proficient or above on the state math assessment for the training data, along with 80 percent of students who scored Below Proficient. Exhibit A5 presents the R code for partitioning the data into training and testing data.

Exhibit A5. Partition the data into training and testing data



```
# Use set.seed() to set the seed for R's random number generator. This is useful
# for creating sets of random numbers in order to reproduce and replicate analyses.
# The seed can be set to any value, but you will need to leave the seed unchanged
# to replicate the results in this appendix.
set.seed(101010)

# Use createDataPartition() to randomly select 80% of the data (p=.8), while
# maintaining the same proportion of Yes for the outcome (depvar) as in the full
# data. This only needs to be done once (times=1) and you do not need to print each
# of the observations (list=FALSE). You will need to use as.vector() to save the
# results as a vector for the next step.
train_index <- as.vector(createDataPartition(mydata[[depvar]], p=.8, list=FALSE, times=1))

# Assign the 80% of observations identified above to the training data and the
# remaining 20% of observations to the testing data.
mytrain <- mydata[train_index, ]
mytest <- mydata[-train_index, ]
```

Stage 2: Develop the model

The CART analysis develops the set of decision rules to split individuals into groups through an iterative process called training and tuning the model (box A1). When training the model, the CART analysis algorithm runs on the training data and generates the best possible model based on a metric that measures the similarity of individuals within groups ([appendix B](#)). Data analysts tune the model by making changes to the parameters that control the algorithm, training the model for each set of parameters, and finding the model that performs best across all trainings.

The process of tuning the model identifies values for parameters that improve the model's performance. There is usually a tradeoff between the degree to which a model fits the data used in creating it and how well it makes predictions for different data. Overfitting (box A2) occurs when a model aligns too closely with the specific data used for training, causing it to perform worse on data that are not exactly like the training data. Users can reduce the likelihood of overfitting the model by adjusting the value of the [complexity parameter \(cp\)](#) to stop the analysis from making more splits (box A2).



Box A1: The CART analysis splitting algorithm

CART analysis creates a decision tree that splits individuals into distinct groups based on relationships between a set of characteristics and an outcome of interest. Two elements play key roles in creating the tree: an algorithm for determining how best to split data at each decision node based on one characteristic at a time and a stopping rule for determining when to stop creating more decision nodes.

The algorithm examines multiple values for each characteristic and identifies thousands of potential ways of splitting the data based on these values. For each possible split, the algorithm computes a measure of [impurity](#),^a or predictive error within a node, for nodes created by the split. The algorithm identifies the split that leads to the largest reduction of impurity. Before determining whether to make the split, the CART algorithm considers the stopping rule.

The stopping rule is a constraint separate from the splitting algorithm and is set by the data analyst or controlled through the tuning process. Data analysts can adjust it by specifying a value for the complexity parameter (cp), which sets a minimum impurity reduction required for a split. A lower value of the complexity parameter tells CART to continue to make splits, even with smaller improvements to impurity. Alternatively, data analysts can impose constraints on other specific elements of the algorithm, such as the maximum number of nodes in the tree. Allowing more nodes in the tree increases the number of splits.

The analysis proceeds in many steps. The algorithm considers multiple ways to split the data and identifies which one leads to the greatest reduction in impurity. If the reduction in impurity exceeds the threshold set by the stopping rule, the algorithm makes the split. Then the process repeats, with the algorithm considering many ways to split the groups, identifying which one leads to the greatest reduction in impurity, and making the split if it exceeds the stopping rule. When the algorithm cannot identify any additional splits that exceed the stopping rule, the algorithm stops splitting. The result of this process is a set of decision nodes that use a decision rule to split the data and terminal nodes that do not split the data further.

^a See Breiman (1996) for more information on how to measure impurity.

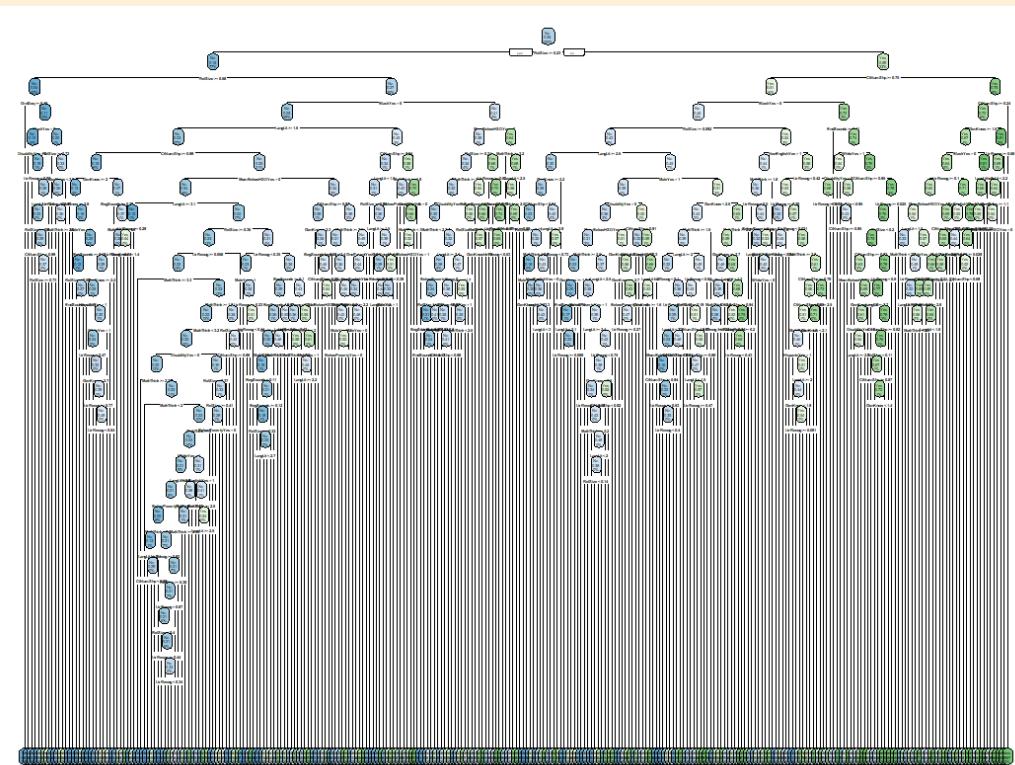


Box A2: Overfitting

One way that data analysts can adjust the stopping rule to stop the CART algorithm from creating more splits is to adjust the value of cp , the complexity parameter, which is the minimum improvement necessary to implement a split. For example, if cp is 0.01, the CART analysis will split a group further only if the split reduces the group's impurity by at least 0.01. If all possible splits considered for a node lead to improvements less than 0.01, the CART analysis will not make a split, and the node will become a terminal node.

When data analysts lower the value of cp , nodes split with smaller gains, leading to a deeper tree with more branches. At the extreme, if the data analyst sets cp equal to zero, the tree would grow to the point that no additional split would lead to any improvement. This is known as a fully grown tree (exhibit A6). Deep trees like this one better fit the unique features of the training data but are unlikely to be optimal for the testing data.

Exhibit A6. Fully grown tree with complexity parameter of zero



Source: Authors' analysis of U.S. Department of Education's Early Childhood Longitudinal Study, Kindergarten Class of 1998/99 (ECLS-K) data.

To identify the optimal value for cp, CART analysis uses [cross-validation](#). This process involves running the algorithm multiple times on different subsets of the training data with different values of cp (box A3). It is common for the CART algorithm to run 10 times, which results in 10 sets of results. For each set of results, the analysis assesses the predictive accuracy with data not used when creating the model. There are different ways to evaluate predictive accuracy, called performance measures. This example uses a [receiver operating characteristic \(ROC\) curve](#) (see [appendix B](#) for more information about performance measures).



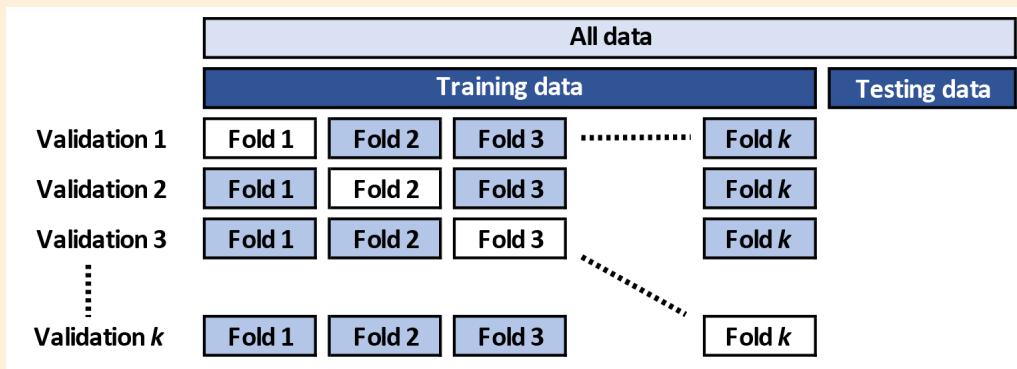
Box A3: Cross-validation

After the data analysts partition the data, they set the testing data aside and work only with the training data to identify the optimal model. Data analysts use the testing data only to evaluate the predictive accuracy of the optimal model.

CART also uses a partitioning approach with the training data during the model-development process. That is, the analysis uses some data to develop the model and the rest to evaluate it. “Validation data” refers to the portion used to evaluate model accuracy during the development process. By developing a series of models and evaluating them all using the validation data, the algorithm can determine the optimal model.

However, it is possible that the way the CART analysis partitions the data will affect the development of the model. To reduce the influence of the data partitioning method, data analysts frequently include multiple rounds of [validation](#). One specific type of validation is k -fold cross-validation, in which users randomly partition the original training sample into k equal-sized subsamples, called folds (exhibit A7).^a In the first validation, the CART analysis holds out fold 1 to be used as validation data for evaluating the model and uses folds 2 through k to develop the model. In the second validation, the CART analysis holds out fold 2 as validation data. This CART analysis repeats this process k times, once for each validation row, with each of the k folds used exactly once as validation data.

Exhibit A7. Partitioning data and k -fold cross-validation



Note: In each row, the fold in the white box is held out.

Source: Authors' creation.

The CART analysis then averages the k results to produce a single estimation of how well the model works on data not used to generate it. The process works as follows:

- Create a random set of k folds.
- For each of the k folds:
 - Train the model on all but one of the folds.
 - Test the model's prediction accuracy on the remaining fold.
 - Store a measure of model performance.
- Average the k measures of model performance.

The advantage of this method is that it uses all observations for both training and validation, and each observation is used for validation exactly once. For decision tree algorithms like CART analysis, cross-validation also eliminates the concern that using slightly different data can cause the structure of the tree to change dramatically. A common type of cross-validation is ten-fold cross-validation, which randomly partitions the training data into 10 equal-sized subsamples.

To further reduce the influence of the specific set of k folds that were selected, analysts often repeat the entire process n times, with a new set of k random partitions created each time.

^a Bootstrapping is another approach sometimes used for cross-validation.

In the code, the data analyst also specifies how R should determine which is the optimal model from the 10 sets of results. One common approach to selecting the best model is the one standard error rule. As described in box A4, the one standard error rule is one way the data analyst can guard against overfitting.



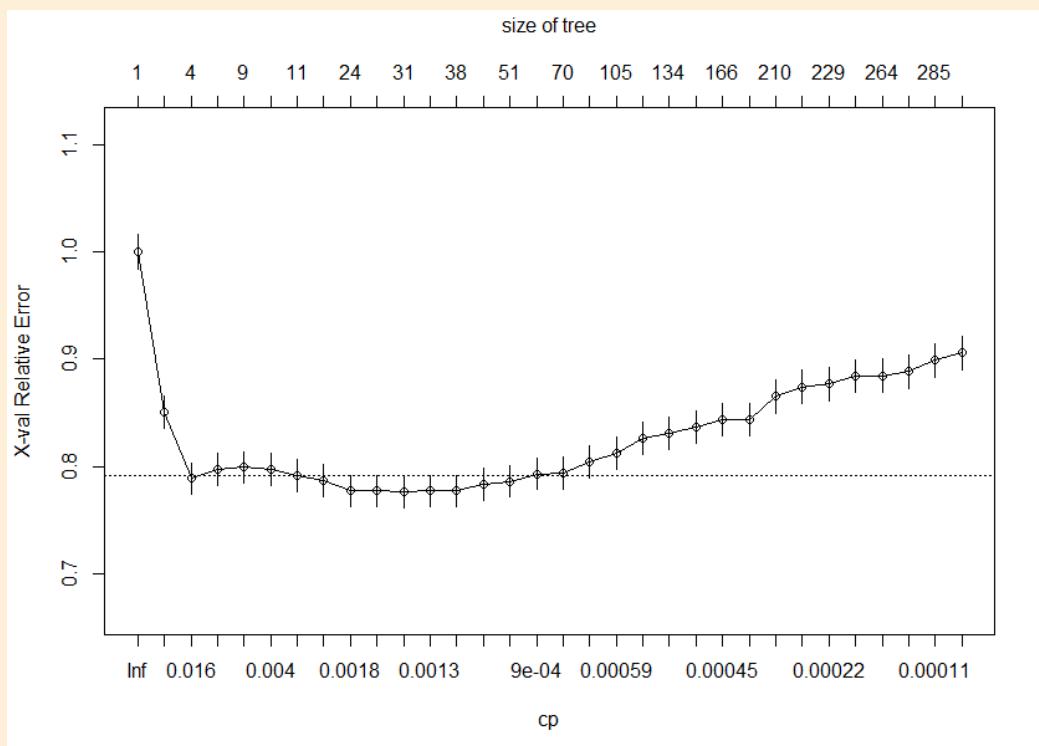
Box A4: The one standard error rule

One common way to identify the optimal model is to use the “one standard error rule,” which posits that the optimal model is the simplest model within one standard error (SE) of the model with the best predictive accuracy (Breiman et al., 1984). This approach can identify an optimal model that is less complicated with minimal loss in the predictive accuracy.

For each value of cp , the CART analysis calculates the average cross-validation error, denoted as X-val Relative Error on the y-axis in exhibit A8. The average cross-validation error indicates how well the model predicts the validation data. As cp increases, moving from left to right along the bottom of the figure, the size of the optimal tree—that is, the number of terminal nodes—increases along the top of the figure. The error rate initially falls sharply and then more slowly. However, as the error begins to rise, the model generates overfitted trees that are worse at making predictions for data not used in determining the model.

R includes a dashed line on the figure, which represents the level of error that is one SE above the lowest. Many trees in exhibit A8 have values for X-val Relative Error that fall on or below the dotted line. The one SE rule would select the simplest model within one SE of the model with the lowest error. To find that model, move from left to right on the figure and identify the first model that is at or below the dotted line (that is, the third model). Referring to the x-axis at the top of the figure, one can see that this tree has four terminal nodes.

Exhibit A8. Model error and tree size for different values of the complexity parameter



Source: Authors' analysis of U.S. Department of Education's Early Childhood Longitudinal Study, Kindergarten Class of 1998/99 (ECLS-K) data.

Train and tune the CART analysis to determine the optimal model

To train and tune the CART analysis to determine the optimal model, users will use the `caret` package, short for **C**lassification and **R**egression **T**raining. Specifically, users will use the `rpart` algorithm, as depicted in exhibit A9.

The first command in the code defines a formula in R to use in each analysis. The formula specifies that the dependent variable (`depvar`) is a function of a set of independent variables (`indepvar`). As specified in previous R code (see exhibit A3), the independent variables for this example are:

- Kindergarten measures of specific math knowledge (`CountNumShp`, `RelativeSize`, `OrdinalSeq`, and `AddSubtract`)
- Kindergarten measures of language knowledge (`LetterRecog`, `BeginSounds`, `EndSounds`, and `SightWords`)
- Kindergarten measures of broad knowledge (`LiteracyARS`, `MathARS`, and `GeneralARS`).

Appendix A: Conducting a CART Analysis

The CART analysis uses these independent variables to predict the indicator for scoring Below Proficient on the state math assessment at the end of grade 3 (AtRisk), which is specified as the dependent variable (see exhibit A3).

The next section of the code uses the `train` function in `caret` to run `rpart` on the training data. The function `trainControl` is where users specify the parameters for the `train` function:

- `Method`, `number`, and `repeats` provide specifications for the cross-validation (see box A3)
- `savePredictions` tells R to save the results from the analysis for use in the next phase
- `selectionFunction` tells R to use the one standard error rule (see box A4)
- `classProbs` and `summaryFunction` provide specifications related to the performance measures, as described in [appendix B](#).

The specifications for the `train` function include `tuneLength`, which specifies that the CART analysis should try 10 values of `cp`, and `metric`, which identifies receiver operating characteristic (ROC) as the performance measure to optimize (appendix B).

In summary, at each of the 10 values of `cp`, the CART analysis runs the training process to find the optimal model and the associated value of the performance `metric`. Then, across all values of `cp`, the CART analysis uses the `selectionFunction` to choose the optimal model.

Exhibit A9. Train and tune the CART analysis to determine the optimal model



```
# Set up the formula for the model in which the dependent variable is a function
# of the set of independent variables. Use paste() to create a summation of the
# variables in indepvar using + between each, and again to add depvar and the ~
# symbol before them. Finally, use as.formula() to save the combination in myformula.
myformula <- as.formula(paste(depvar,paste(indepvar,collapse=" + "),sep=" ~ "))

# Use trainControl() to define parameters for train().
mycontrol <- trainControl(
  method = "repeatedcv",           # Repeated cross-validation
  number = 10,                    # Number of folds (k)
  repeats = 10,                  # Number of repeats (n) of cross-validation
  savePredictions = "final",      # Save predictions for best tuning parameters
  classProbs = TRUE,              # Compute probabilities for each class
  selectionFunction = "oneSE",     # Select model within one standard error of best
  summaryFunction = twoClassSummary # Provide ROC, sensitivity, and specificity
)

# Use caret's train() function to tune using cp and select using ROC.
mytree <- train(
  myformula,                      # Use the formula defined above
  data = mytrain,                 # Use the subset of data for training
  method = "rpart",               # Use the rpart procedure for CART analysis
  trControl = mycontrol,          # Use the controls defined above
  tuneLength = 10,                # Try 10 values of the complexity parameter (cp)
  metric = "ROC"                 # Use the ROC as the metric for choosing the model
)
```

When users run the code in exhibit A9, R delivers the output in exhibit A10. The information at the top of the output describes the analysis conducted with the 11,500 students in the training dataset. It included 11 characteristics, referred to as “predictors” in the output. The outcome variable has two groups: yes and no. The analysis was cross-validated, with 10 folds, repeated 10 times (see box A3). The bottom part of the output displays the results for the complexity parameter; ROC indicates that, using the one standard error rule, it selected an optimal model. The optimal model is the one with a $cp = 0.002433936$, which is the sixth model listed. The value of ROC (0.8061254) is the largest of all the models listed in the output, indicating that this model has the greatest area under the curve. ROC is explained in greater detail in [appendix B](#).

Exhibit A10. Output from code used to train and tune the CART analysis to determine the optimal model



CART

```
11500 samples  
11 predictor  
2 classes: 'No', 'Yes'
```

```
No pre-processing  
Resampling: Cross-Validated (10 fold, repeated 10 times)  
Summary of sample sizes: 10350, 10350, 10350, 10350, 10349, 10350, ...  
Resampling results across tuning parameters:
```

| cp | ROC | Sens | Spec |
|-------------|-----------|-----------|-----------|
| 0.001216968 | 0.8015215 | 0.9161765 | 0.4203375 |
| 0.001390821 | 0.8027753 | 0.9158637 | 0.4225956 |
| 0.001564673 | 0.8042040 | 0.9165479 | 0.4233228 |
| 0.001738526 | 0.8045191 | 0.9169422 | 0.4231139 |
| 0.002086231 | 0.8054273 | 0.9157826 | 0.4285714 |
| 0.002433936 | 0.8061254 | 0.9155858 | 0.4310420 |
| 0.004056560 | 0.7637101 | 0.9155052 | 0.4239251 |
| 0.004520167 | 0.7332805 | 0.9183219 | 0.4125508 |
| 0.015125174 | 0.7139931 | 0.8887995 | 0.4931895 |
| 0.161682893 | 0.6091557 | 0.9268080 | 0.2915033 |

The analysis used ROC to select the optimal model using the one SE rule.
The final value used for the model was cp = 0.002433936.

Note: The white box highlighting a value of cp in the exhibit was added by the authors. It is not part of the original R output.

Users can view the optimal model in several ways. First, users can look at the nodes and decision rules using the code presented in the blue box in exhibit A11.

Exhibit A11. Print the nodes and decision rules from the optimal model



```
# Print the nodes and decision rules from the optimal model.  
mytree$finalModel
```



```
1) root 11500 2876 No (0.74991304 0.25008696)  
2) RelativeSize>=0.2395 8795 1291 No (0.85321205 0.14678795)  
   4) RelativeSize>=0.6465 5057 254 No (0.94977259 0.05022741) *  
   5) RelativeSize< 0.6465 3738 1037 No (0.72257892 0.27742108)  
     10) MathARS>=1.725 3395 874 No (0.74256259 0.25743741) *  
     11) MathARS< 1.725 343 163 No (0.52478134 0.47521866)  
       22) CountNumShp>=0.9575 226 87 No (0.61504425 0.38495575) *  
       23) CountNumShp< 0.9575 117 41 Yes (0.35042735 0.64957265) *  
3) RelativeSize< 0.2395 2705 1120 Yes (0.41404806 0.58595194)  
6) CountNumShp>=0.7465 1808 903 Yes (0.49944690 0.50055310)  
   12) CountNumShp>=0.9225 661 287 No (0.56580938 0.43419062) *  
   13) CountNumShp< 0.9225 1147 529 Yes (0.46120314 0.53879686)  
     26) MathARS>=1.555 970 466 Yes (0.48041237 0.51958763)  
       52) LetterRecog>=0.1905 620 305 No (0.50806452 0.49193548)  
         104) LetterRecog< 0.4045 189 75 No (0.60317460 0.39682540) *  
         105) LetterRecog>=0.4045 431 201 Yes (0.46635731 0.53364269) *  
         53) LetterRecog< 0.1905 350 151 Yes (0.43142857 0.56857143) *  
     27) MathARS< 1.555 177 63 Yes (0.35593220 0.64406780) *  
7) CountNumShp< 0.7465 897 217 Yes (0.24191750 0.75808250) *
```

The output (see the gray box in exhibit A11) describes the decision and terminal nodes of a decision tree. Each line begins with a reference number for the node, followed by the decision rule that led to the node and the number of individuals who reached the node. The first line, referenced as node 1, shows the [root node](#) of the tree, which contains all 11,500 students in the training data.

The node reference numbers have a logical structure, which, along with the indentation in the output, show how the nodes are linked. Nodes 2 and 3 are [child nodes](#) of node 1, and node 1 is the [parent node](#) of nodes 2 and 3. The numbering of the nodes in the output reflects this logical structure. For any node x , the child nodes are $2x$ and $2x+1$. For any node y , the parent node is $y/2$ if y is even or $(y-1)/2$ if y is odd. For example, the child nodes of node 2 (x) are 4 ($2x$) and 5 ($2x+1$), both of which indent one step further than node 2. Similarly, the parent node of node 4 (y) is 2 ($y/2$) since y is even. The node numbers describe the connections between nodes using the rules above. The number 104 indicates only that node 104 is one of the child nodes of node 52; it does not mean there are 104 nodes in the tree.

Appendix A: Conducting a CART Analysis

From the root node, the CART analysis determined that the best way to split these students into nodes 2 and 3 was based on their value for `RelativeSize`: the probability, represented as a decimal between 0 and 1, that a kindergarten student had mastered the early math concept of relative size at the beginning of kindergarten. Those with a `RelativeSize` value greater than or equal to 0.2395 appear in node 2, and those with a `RelativeSize` value less than 0.2395 appear in node 3. That is, the CART analysis split the 11,500 students in node 1 into 8,795 students in node 2 and 2,705 students in node 3.

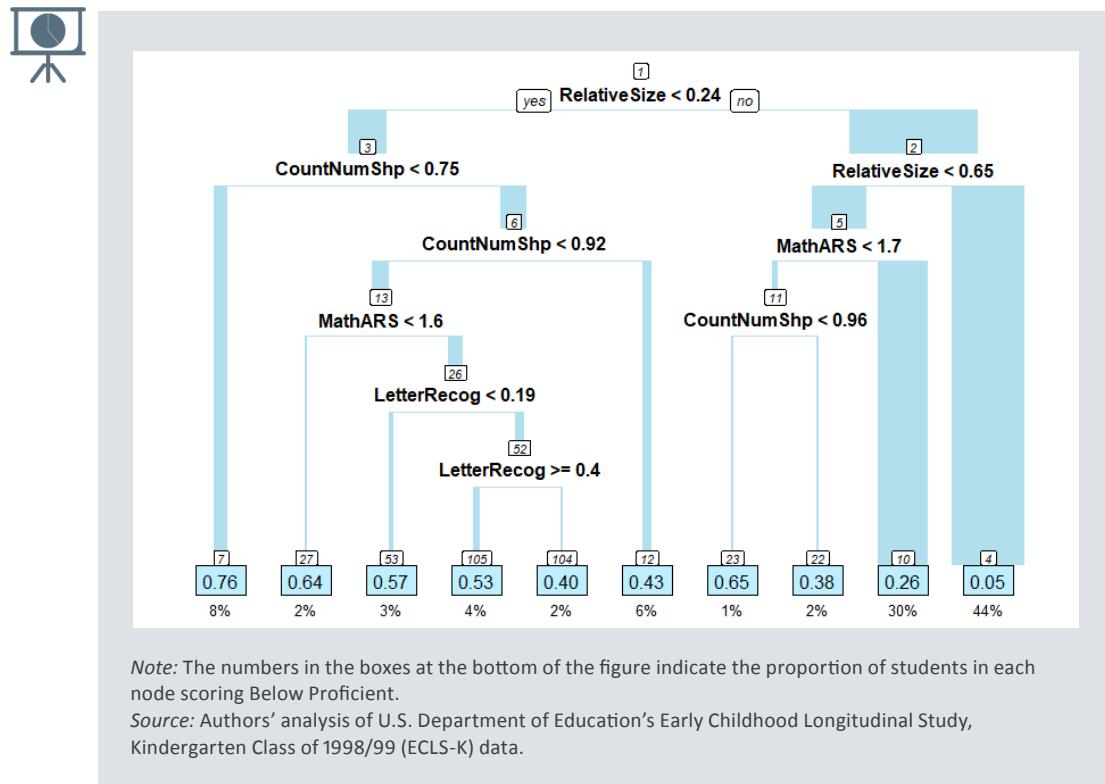
Node 3 is also a decision node. Its decision rule split students into nodes 6 and 7 based on their value of `CountNumShp`: the probability that a student had mastered the early math concepts of count, number, and shape at the beginning of kindergarten. Students with a value greater than or equal to 0.7465 appear in node 6, and the other students appear in node 7. The description of node 7 ends with a `*`, indicating that it is a terminal node with no further splits.

One of the main benefits of CART analysis is the ability to show information graphically. The code in the blue box in exhibit A12 demonstrates how to plot the nodes and decision rules from the optimal model. The gray box presents the output that results from running the code.

Exhibit A12. Plot the nodes and decision rules from the optimal model



```
# Plot the nodes and decision rules from the optimal model.
rpart.plot(
  mytree$finalModel,           # The optimal model from the CART analysis
  box.palette = "lightblue1", # Box color
  type = 0,                  # Draw labels for each split and node
  leaf.round = 0,             # Do not use rounded terminal nodes
  nn = T,                     # Include node numbers
  branch.col = "lightblue2", # Color of the branches
  branch.type = 5,            # Branch width based on share of students
  extra = 107,                # Show % in node at risk and % of all students
  xflip = T,                  # Flip the tree horizontally
  under = T,                  # Place overall percentage under leaf
  cex = 1                     # Size of text
)
```



The nodes in exhibit A11 and the tree plotted in exhibit A12 are two representations of the same decision tree. The tree shows root node 1 plotted at the top. The tree splits students into two nodes based on the value of RelativeSize: students with RelativeSize scores at or above the cutoff take the branch to the right to node 2; students below the cutoff take the branch to the left to node 3. When creating the tree, CART uses the convention that students with a value of Yes split to the left and students with a value of No split to the right. The width of each vertical branch or column represents the proportion of students who follow it.

The splits continue to a set of terminal nodes, presented along the bottom of the figure. The numbers in the white boxes in exhibit A12 are the node numbers. This set of mutually exclusive groups contains all students who started at the top of the tree. In the dataset, each student has a score of 0 or 1 for the outcome variable, where 1 indicates that a student was Below Proficient and 0 indicates that a student was not.⁷ The numbers in the blue boxes are the averages of those values for the students in that group, or the share of kindergarten students in the node who scored Below Proficient on the state math assessment at the end of grade 3. Beneath each terminal node is the percentage of all kindergarten students who

7. The example uses a categorical outcome, whether or not a student scored Below Proficient on the state math assessment at the end of grade 3. The resulting tree is a “classification tree” because the result is typically used to determine the class for each group, such as receiving an intervention or not. Data analysts can use CART analysis to examine continuous outcomes, such as a grade 3 mathematics score that ranges from 0 to 100. The CART analysis would again create the tree by creating decision rules that split students into terminal nodes that would report the average test score for students in the node.

end up in that node. For example, terminal node 7 contains 8 percent of all kindergarten students (920 of the 11,500 students), and 76 percent of students in that node (699 of the 920 students) scored Below Proficient on the state math assessment at the end of grade 3.

Stage 3: Analyze the results

During stage 3, users will apply the rules from the optimal model identified in stage 2 to the testing data to make predictions about outcomes and develop a set of policy options for educators to consider. Recall that the testing data, put aside after users partitioned the data in stage 1, are a random subsample from the same prior cohort of students, including actual outcomes, used for the training and tuning process. In stage 2, users compare model predictions to actual outcomes in the training dataset to evaluate predictive accuracy and determine the optimal model. In stage 3, users will compare model predictions to actual outcomes in the testing data. A research director or data analyst can use the results of this comparison to create a set of options to help educators decide which groups of students should receive the intervention, along with information about their implications, as described in the [“Applying the CART decision tree to data for the current cohort of students and analyzing results”](#) section of the guide. To illustrate how to do this, this section will again use the Eduphonics example presented in the body of the guide.

The optimal model sorts students into 10 groups, each with different probabilities of scoring Below Proficient on the state math assessment at the end of grade 3 (see exhibit A12). Which of these groups should receive the intervention? Educators might consider providing the intervention only to the group at highest risk of scoring Below Proficient (see terminal node 7 in exhibit A12). Alternatively, they might consider providing the intervention to all but the lowest-risk group (all terminal nodes except terminal node 4 in exhibit A12). Using the training data, data analysts can generate a set of options that can help guide this decisionmaking process.

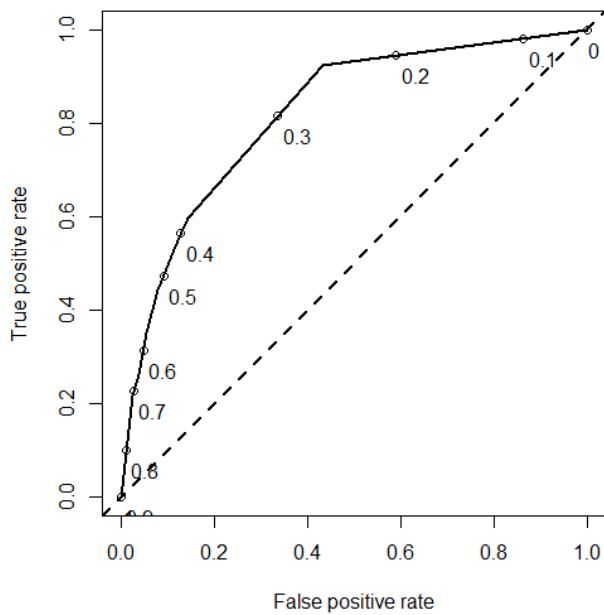
Consider options

In this step, users will assess the predictive accuracy by generating a ROC curve, which plots information about [true positive](#) and [false positive rates](#) (see [appendix B](#) for more information about ROC curves), to inform options for educators to consider. The code in the blue box in exhibit A13 demonstrates how to plot the ROC curve. First, using `predict`, the code generates predicted values for each student by applying the rules from the model in exhibit A12. Imagine each student starting at the root node and following splits based on the student’s scores and characteristics until ending in a terminal node. After doing this for all students, users can compare how the predictions based on the model’s decision rules compare to whether or not they actually scored Below Proficient on the state math assessment. If the model predicted that certain students would score Below Proficient, and they did, these students are true positives, because the prediction matches the actual outcome. Students predicted to score Below Proficient who did not actually score Below Proficient are false positives. In the next section of code, `performance` calculates the true positive and false positive rates in the testing data (exhibit A13). The last section of code creates the ROC curve, presented in the gray box in exhibit A13. See [appendix B](#) for a description of how to interpret the ROC curve.

Exhibit A13. Predict probabilities using the model and plot the ROC curve



```
# Use predict() to predict probabilities of scoring Below Proficient or not using  
# the model from caret and save the at risk probability as the prediction.  
pred <- predict(mytree$finalModel,mytest,type="prob")[,2]  
  
# Use prediction() to transform the predictions and actual values into a format for  
# use by performance(), which calculates the true positive and false positive rates.  
perf <- performance(prediction(pred,mytest[[depvar]]),"tpr","fpr")  
  
# Use par() to format the plotting area as a square. Then use plot() to plot the  
# combinations of true positive and false positive rates and include labels for  
# values of the probability threshold. Finally, use abline() to include a line  
# for reference.  
par(pty="s")  
plot(perf,print.cutoffs.at=seq(0,1,by=0.1),text.adj=c(-0.2,1.7),lwd=2)  
abline(0,1,lty=2,lwd=2)
```



Source: Authors' analysis of U.S. Department of Education's Early Childhood Longitudinal Study, Kindergarten Class of 1998/99 (ECLS-K) data.

Appendix A: Conducting a CART Analysis

The code in exhibit A14 demonstrates how to obtain the true positive rates (`tpr`) and false positive rates (`fpr`) from the ROC curve for the probability thresholds (`pth`) that correspond with each of the 10 terminal nodes in exhibit A12. R will deliver the output of this code, as shown in the gray box. Each of the values in the column labeled `pth` correspond with the proportion of students who scored Below Proficient in each of the terminal nodes in exhibit A12. The output presents these proportions with a much greater level of precision—eight decimal places, rather than the two decimal places in the tree in exhibit A12. Also note that the previous splits determine the order the terminal nodes appear in exhibit A12, whereas in exhibit A14, they are sorted from the highest to the lowest probability of scoring Below Proficient. The columns labeled `tpr` and `fpr` provide the true positive rates and false positive rates associated with each probability threshold (exhibit A14).

Exhibit A14. Extract results to inform options



```
# Extract the x, y, and alpha values from the points along the ROC curve.  
rocpoints <- data.frame(pth=perf@alpha.values[[1]],tpr=perf@y.  
values[[1]],fpr=perf@x.values[[1]])  
  
# Print the values of the measures for each point along the ROC curve.  
rocpoints
```



| | pth | tpr | fpr |
|----|------------|-----------|------------|
| 1 | Inf | 0.0000000 | 0.0000000 |
| 2 | 0.75808250 | 0.2364395 | 0.02516234 |
| 3 | 0.64957265 | 0.2628651 | 0.02991651 |
| 4 | 0.64406780 | 0.3025035 | 0.03722171 |
| 5 | 0.56857143 | 0.3716968 | 0.05473098 |
| 6 | 0.53364269 | 0.4516690 | 0.07803803 |
| 7 | 0.43419062 | 0.5514604 | 0.12140538 |
| 8 | 0.39682540 | 0.5775382 | 0.13462430 |
| 9 | 0.38495575 | 0.6077886 | 0.15074212 |
| 10 | 0.25743741 | 0.9116829 | 0.44306586 |
| 11 | 0.05022741 | 1.0000000 | 1.0000000 |

Note: The orange and white boxes highlighting values of `pth` in the exhibit were added by the authors. They are not part of the original R output.

Next, users will use the true positive rate (`tpr`) and false positive rate (`fpr`) to create options for determining which groups of kindergarten students will receive the intervention and to determine the implications for students. Recall that historically, one-quarter of students in Eduphonia have scored Below Proficient on the state math assessment at the end of grade 3. The district expects this to be the case this year as well; 250 of the current 1,000 kindergarten students are likely to score Below Proficient.

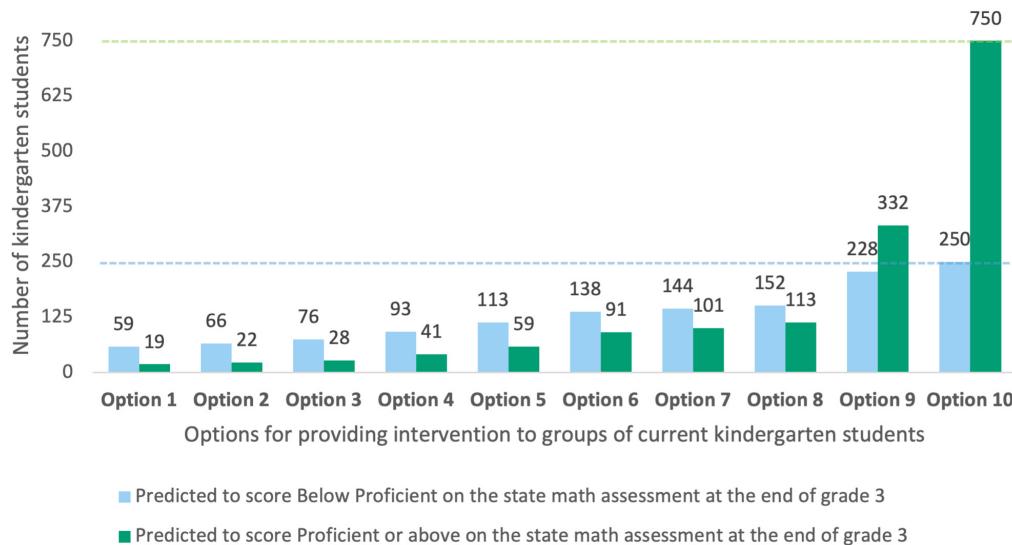
Appendix A: Conducting a CART Analysis

A district decision to give the intervention only to the group of students with the highest predicted probability of scoring Below Proficient on the state math assessment at the end of grade 3 corresponds to row 2, highlighted in yellow in the output above (see exhibit A14).⁸ At this point:

- Only the group with the 75.8 percent predicted probability of scoring Below Proficient on the state math assessment at the end of grade 3 (see terminal node 7 in exhibit A12) would receive the intervention.
- The output in exhibit A14 shows that the true positive rate for this probability threshold is 23.6 percent. This means that 59 of the 250 students predicted to score Below Proficient on the state math assessment at the end of grade 3 would fall into this group and receive the intervention.
- The output in exhibit A14 shows that the false positive rate for this probability threshold is 2.5 percent. This means that 19 of the 750 students predicted to score Below Proficient on the state math assessment at the end of grade 3 would fall into this group and receive the intervention.

Users will repeat this process for each combination of rates from rows of the output above and plot the implications of providing the intervention to different groups of students (exhibit A15).

Exhibit A15. Implications of providing the intervention to different groups of students



Note: Options are cumulative, such that each option adds students to the students included in the previous option. For example, the 66 students predicted to score Below Proficient in option 2 include the 59 students predicted to score Below Proficient in option 1.

Source: Authors' analysis of U.S. Department of Education's Early Childhood Longitudinal Study, Kindergarten Class of 1998/99 (ECLS-K) data.

8. The first row represents not providing the intervention to any students and has not been included in the set of options. Therefore, row 2 in the output corresponds to the first option in the exhibit.

Appendix A: Conducting a CART Analysis

For transparency and consistency, it may be useful for educators to formulate a rule for choosing among the available options before doing the analysis. Some possibilities include:

- Based on the cost of the intervention, the district can provide the intervention to at most 250 students. Their rule may be to choose the option that serves the most students predicted to score Below Proficient on the state math assessment at the end of grade 3, while keeping the total number of students receiving the intervention under 250. In the example, this would lead to selecting option 7, which provides the intervention to 245 students (144 students predicted to score Below Proficient on the state math assessment at the end of grade 3 and 101 students predicted to score at or above Proficient on the state math assessment at the end of grade 3).
- A district less constrained by resources may want to provide the intervention to at least 80 percent of students predicted to score Below Proficient on the state math assessment at the end of grade 3. Their rule would lead to selecting option 9. In that option, 228 students predicted to score Below Proficient will receive the intervention out of a total of 250 students, or 91 percent. In option 8, only 61 percent of students predicted to score Below Proficient will receive the intervention (152/250).
- A district may be very concerned about the potential negative effects of providing the intervention to students who may not benefit from it. They may want to ensure that no more than 20 percent of students predicted to score Proficient or above on the state math assessment at the end of grade 3 receive the intervention. This would lead to selecting option 8, in which the intervention is provided to 16 percent of students (113/750) expected to score Proficient or above.
- Educators might find that none of the options that CART generates align well with their priorities. For example, they may have decided that they want to provide the intervention to at least 80 percent of students expected to score Below Proficient. The CART results may indicate that to achieve this goal, they also have to provide the intervention to a large proportion of students who are predicted to score Proficient or above, which is cost-prohibitive. In this case, they may want to consider other interventions.

Suppose that after reviewing their options, the educators in Eduphonnia choose option 8, in which all groups will receive the intervention except the two with the lowest probabilities for scoring Below Proficient on the state assessment at the end of grade 3. The probability threshold (`pth`) associated with that option (shown in the white box in the output in exhibit A14) is 0.385.

Choose and implement a policy

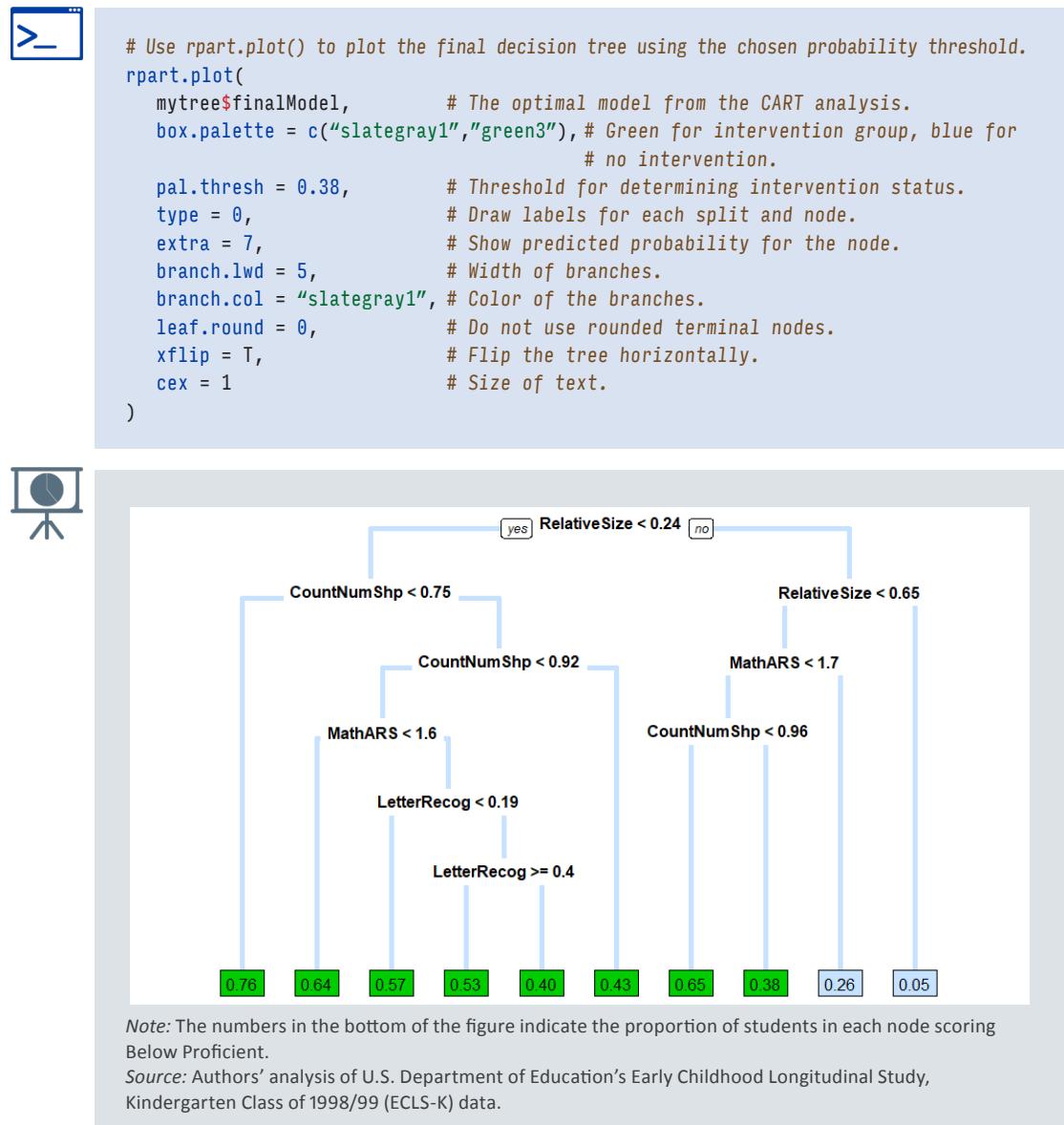
Based on the final decision, the data analyst can plot the final decision tree for use in implementing the choice. To do this, users will use `pal.thresh` to divide students into two groups: those who will receive the intervention and those who will not. The code in exhibit A16 uses a cutoff of 0.38. Any threshold that is less than the selected probability threshold for option 8, which is 0.385, and greater than the probability threshold for option

Appendix A: Conducting a CART Analysis

9, which is 0.257, will yield the same result.⁹ In the output shown in the gray box in exhibit A16, every value above this cutoff is colored green to indicate that students will receive the intervention, while every value below the cutoff is colored blue to indicate that students will not receive the intervention. By examining the decision tree in exhibit A16, users can see that the decision rules illustrate that a student will receive the intervention if they have:

- a `RelativeSize` score of less than 0.24, or
- a `RelativeSize` score between 0.24 and 0.65 and a `MathARS` score of less than 1.7.

Exhibit A16. Plot the final decision tree for identifying students for intervention



9. In this example, there are no nodes with probabilities between 0.257 and 0.385. Thus, using any value between these two numbers to split the students will produce identical results.

APPENDIX B: PERFORMANCE MEASURES

The Eduphonnia example used throughout the guide is a [classification problem](#) for which there are two classes of the outcome of interest: Yes for scoring Below Proficient on the state math assessment at the end of grade 3, and No for scoring Proficient or above. Each observation in the data for a prior cohort of students has an actual value for whether the student scored Below Proficient on the state math assessment at the end of grade 3, and a predicted value from the CART analysis. This appendix describes a range of common metrics that analysts can use when evaluating and choosing a model (see stage 2 of [appendix A](#)).

A [classification matrix](#) provides several measures for evaluating the performance of classification models, with rows representing the values predicted by applying the model rules to the training data and columns representing the actual values observed in the training data (exhibit B1).

Exhibit B1. Classification matrix and related measures used for evaluating classification models

| | | Actual / Reference | | |
|-----------|-----|---|--|--|
| | | No | Yes | |
| Predicted | No | True Negative (TN) | False Negative (FN) | <i>Negative Predictive Value = $TN / (TN + FN)$</i> |
| | Yes | False Positive (FP) | True Positive (TP) | <i>Positive Predictive Value = $TP / (TP + FP)$</i> |
| | | <i>False Positive Rate = $FP / (FP + TN)$</i> <i>Specificity = $1 - False Positive Rate$</i> | <i>True Positive Rate = $TP / (TP + FN)$</i> <i>Sensitivity = True Positive Rate</i> | <i>Accuracy = $(TP + TN) / (TP + TN + FP + FN)$</i> |

The four shaded cells form the 2x2 classification matrix, showing the combinations' actual and predicted values of Yes and No.

The [true positive](#) and [true negative](#) combinations describe students accurately classified:

- True Positives (TP): prediction = Yes and actual = Yes. The model accurately identified these students as scoring Below Proficient on the state math assessment at the end of grade 3.
- True Negatives (TN): prediction = No and actual = No. The model accurately identified these students as not scoring Below Proficient (that is, they scored Proficient or above) on the state math assessment at the end of grade 3.

Appendix B: Performance Measures

The false positive and false negative combinations describe students not accurately classified:

- False Positives (FP): prediction = Yes and actual = No. These students did not score Below Proficient (that is, they scored Proficient or above) on the state math assessment at the end of grade 3, but the model predicted they would.
- False Negatives (FN): prediction = No and actual = Yes. These students scored Below Proficient on the state math assessment at the end of grade 3, but the model predicted they would not.

The five unshaded cells in exhibit B1 contain related performance measures easily computed from the elements of the classification matrix.

The true and false positive rates appear in one column in exhibit B1. The true positive rate is the share of all actual positive events predicted to be positive, and the false positive rate is the share of all actual negative events predicted to be positive.

The negative and [positive predictive values](#) appear in one row in exhibit B1. The positive predictive value is the share of all predicted values of positive that are actually positive, and the [negative predictive value](#) is the share of all predicted values of negative that are actually negative.

Finally, accuracy is the share of all predictions that are correct. Because this is an aggregate measure, it may mask important information about how the model is performing. For example, predicting that all students will score at or above Proficient on the state math assessment at the end of grade 3 results in an accuracy of 75 percent, since 75 percent of all students score at or above Proficient. However, it does not correctly identify any of the students who will score Below Proficient on the state math assessment at the end of grade 3. Now consider making correct predictions for 70 percent of students who score Below Proficient on the state math assessment at the end of grade 3 and 70 percent of students who do not. In this case, accuracy is 70 percent. Though the accuracy is lower than that in the previous example, the predictions are more useful for identifying students who will score Below Proficient on the state math assessment at the end of grade 3.

Setting the `summaryFunction` to `twoClassSummary` in `caret` (see exhibit A9) prompts R to report three related measures in its output: [sensitivity](#) (same as the true positive rate); [specificity](#) (1 minus the false positive rate); and the [area under the receiver operating characteristic \(ROC\) curve](#), which reflects both the true positive and false positive rates. The ROC curve provides a more comprehensive summary of the performance of a classification model but requires more explanation, starting with an understanding of how to create these predictions.

Recall that for each student, the model predicts the probability—between 0 and 1—that he or she will score Below Proficient on the state math assessment at the end of grade 3. This prediction assumes that relationships identified between the characteristics and outcomes in the data for a prior cohort of students will also hold for the current cohort of students

Appendix B: Performance Measures

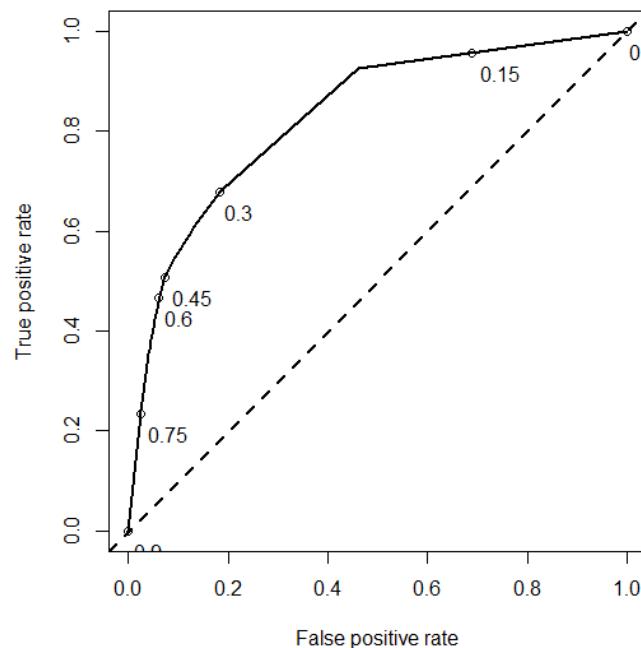
and that data for the current cohort of students have distributions of the characteristics that are similar to students in the prior cohort. Combined, these assumptions allow the observed outcomes for a node of students identified by decision rules in the prior cohort to serve as the predicted outcome for the node of students identified by the same decision rules in the data for the current cohort of students. For students in a terminal node, the probability reported is the average of the probabilities of the students in that node. In the example, these ranged from 0.05 to 0.76 (see exhibit A12).

Every one of the performance measures described in exhibit B1 is based on comparing actual values of Yes and No to predicted values of Yes and No. Probabilities cannot be used to evaluate these performance measures, such as comparing an actual value of Yes for scoring Below Proficient to a 0.70 predicted probability of scoring Below Proficient. Therefore, they each rely on a critical additional assumption: a predicted probability of 0.50 or higher means Yes and a predicted probability lower than 0.50 means No. This level of 0.50 is called the probability threshold, which treats values above it as Yes and values below it as No.

The ROC curve differs from the above measures in that it does not make the same assumption. It looks at how well the model does when the probability threshold that splits Yes and No changes to something else. For example, if all students with a probability equal to or higher than 0.25 are classified as Yes, how would the model perform? What about 0.75?

Specifically, under different probability thresholds, what would the true positive rates and false positive rates be? The plot of those two rates for all possible probability thresholds between 0 and 1 is the ROC curve (exhibit B2).

Exhibit B2. Receiver operating characteristic (ROC) curve



Source: Authors' analysis of U.S. Department of Education's Early Childhood Longitudinal Study, Kindergarten Class of 1998/99 (ECLS-K) data.

Appendix B: Performance Measures

The ROC curve plots the true positive rate on the y-axis against the false positive rate on the x-axis. The solid curved line is the ROC curve, and the points along the ROC curve are the probability thresholds. For example, if the probability threshold is 0.30, predicting that students with a probability above 0.30 will score Below Proficient on the state math assessment at the end of grade 3 and other students will not, the true positive rate would be over 0.65 and the false positive rate would be about 0.20.

As the threshold decreases, moving along the ROC curve away from the origin of the graph (where both the x- and y-axes are equal to 0), the curve predicts more students to score Below Proficient on the state math assessment at the end of grade 3, increasing both the true positive rate and false positive rate. For example, a probability threshold of 0.15 has a true positive rate of about 0.95, and a false positive rate of about 0.70. So, moving the probability threshold from 0.30 to 0.15 increases both the rate of students correctly predicted to score Below Proficient on the state math assessment at the end of grade 3 (from 0.65 to 0.90) as well as the likelihood of an incorrect prediction for students who will score Proficient or above (from 0.20 to 0.70).

This pattern continues all along the ROC curve as the probability threshold declines, moving from the origin to the top right corner of the figure. A decrease in the probability threshold results in an increase in the true positive rate and the false positive rate. However, the relative sizes of these two increases change along the curve. Moving from a probability threshold of 0.75 to 0.60, there is a large increase in the true positive rate and a very small increase in the false positive rate. That is, the true positive rate for correctly predicting that students will score Below Proficient on the state math assessment at the end of grade 3 increases much more quickly between the thresholds of 0.75 and 0.60 than the false positive rate for incorrectly predicting that students will score Proficient or above. As the probability threshold declines, these two rates become more similar. Eventually, the only way the model can increase successful predictions for students who will score Below Proficient on the state math assessment at the end of grade 3 is to make many more incorrect predictions for students who will not, as in moving from a probability threshold of 0.15 to 0.

The measure of model performance **caret** uses and reports (see exhibit A10) as ROC is the area under the curve (AUC). Higher values indicate better model performance. The dashed line represents a completely random model and has an AUC of 0.50. A perfect model would have an AUC of 1, as it would have all true positives and no false positives. Unlike other metrics, like sensitivity or specificity, which examine performance under the assumption that a 0.50 predicted probability is the dividing line between predictions of Yes and No, the AUC provides a summary measure of the model's performance across all possible uses of the model findings. The points on this curve generate the options for educators to consider (see exhibit A15).

APPENDIX C: CUSTOMIZATION

The `rpart` and `caret` packages provide several options for obtaining the optimal model. As data analysts gain more experience with CART and a deeper understanding of the models, they can consider additional ways to customize the training and tuning. This appendix describes how to create a metric for evaluating and choosing a model and introduces a [loss function](#) that can incorporate preferences regarding the relative importance of different types of errors.¹⁰

`Caret` allows users to create their own performance measures, which may be useful if their preferences are both quantifiable and not already calculated. For example, maximizing sensitivity (true positive rate) alone would result in choosing a probability threshold of 0 and classifying everyone as a Yes, while minimizing specificity (true negative rate) alone would result in choosing a probability threshold of 1 and classifying everyone as a No. One approach to incorporate both goals is to create a metric based on both, such as adding them together. This moves the optimal selection to the middle of the ROC curve.

Another customization allows the CART analysis algorithm to change how it considers different types of errors. A loss function imposes a [penalty](#) on certain types of errors, increasing the cost of making that error when evaluating decision rules using a performance measure and encouraging the model to avoid them. For example, a penalty of 2 on false negatives means that failing to correctly predict that a student will score Below Proficient on the state math assessment at the end of grade 3 is twice as costly as failing to correctly predict a student will score Proficient or above.

Putting all the pieces together would involve running the entire training and tuning process for several values of the penalty and using the custom metric defined above to evaluate models (exhibit C1).

Exhibit C1. Outline of the tuning process with additional customizations

```
For each value of the penalty
  For each value of the complexity parameter cp
    For each of the n repeats
      Create a random set of k folds
      For each of the k folds
        Train the model on all but one of the k folds
        Test the model's prediction accuracy on the remaining fold
        Store a measure of model performance
      Average the k measures of model performance
    Average the n measures of model performance
  Select the value of the complexity parameter that produced the highest performance
Select the value of the penalty that produced the highest performance
```

10. This appendix presents two customizations that illustrate some ways of extending the analysis in stage 2 of appendix A, but they are not required, do not need to be used together, and do not cover all possible extensions.

Appendix C: Customization

The code in exhibit C2 explores several values for the penalty and selects the one that leads to the best model as determined by the custom metric.

Exhibit C2. CART analysis with additional customizations



```
# Create a performance metric that is the sum of sensitivity and specificity.
mymetric <- function(data, lev = levels(data$obs), model = NULL) {
  out <- c(twoClassSummary(data, lev = levels(data$obs), model = NULL))
  metric <- out[["Spec"]] + out[["Sens"]]
  c(out, SS = metric)
}

# Use expand.grid() to create a matrix with penalties ranging from 1 to 8, along
# with columns to collect information on the performance metric and complexity parameter.
mygrid <- expand.grid(penalty=seq(1,8,1),ss=0,cp=0)

# Use list() to define a place to store all the models.
models <- list()

# Adjust trainControl() to use the metric above.
mycontrol <- trainControl(
  method = "repeatedcv",
  number = 10,
  repeats = 10,
  savePredictions = "final",
  classProbs = TRUE,
  summaryFunction = mymetric,
  selectionFunction = "oneSE"
)

# There is one row in mygrid for each penalty level. For each penalty level,
# use train() to run the CART analysis model.
for (iteration in 1:nrow(mygrid)) {

  # Store train() results
  models[[iteration]] <- train(
    myformula,
    data = mytrain,
    method = "rpart",
    trControl = mycontrol,
    tuneLength = 25,
    metric = "SS.Spec",

    # Use parms to send the loss parameter to rpart, with the current
    # penalty value for false negatives from this time through the loop.
    parms = list(loss=matrix(c(0,1,mygrid$penalty[iteration],0),byrow=TRUE,nrow=2))
  )

  # Use which.max() to find the model with the highest metric value.
  best <- which.max(models[[iteration]]$results$SS.Spec)

  # Store the values of the metric and the tuning parameter, cp.
  mygrid$ss[iteration] <- models[[iteration]]$results$SS.Spec[best]
  mygrid$cp[iteration] <- models[[iteration]]$results$cp[best]
}

# Use which.max() to select the row with the highest value of the performance
# metric and then extract the model that generated it.
mytree <- models[[which.max(mygrid$ss)]]
```

GLOSSARY OF TERMS

Accuracy. A metric for evaluating classification models. Informally, accuracy is the fraction of outcomes that the model predicted correctly. Formally, it is the sum of true positives and true negatives over the total number of predictions.

Algorithm. The process by which a predictive model is created.

Area under the ROC curve (AUC). AUC measures the two-dimensional area underneath the ROC curve. It provides an aggregate measure of performance across all classification thresholds. It ranges in value from 0 with all predictions incorrect to 1 with all predictions correct.

Branch. A path that connects nodes in a decision tree.

Characteristics. Attributes of an individual that the CART analysis can use to predict the outcome of interest based on relationships determined by the model; also known as predictor variables.

Child node. A node that results when a decision rule splits a larger node.

Classification and Regression Tree (CART). A statistical modeling approach that uses quantitative data for a prior cohort of students to predict an outcome of interest. A decision tree can represent a CART analysis model, in which each decision point is a split based on a predictor variable and each terminal node contains a prediction for the outcome variable.

Classification matrix. A matrix that evaluates the performance of a model for a classification problem by comparing the actual outcome values with those predicted by the model. R software refers to the classification matrix by a common alternative term: confusion matrix.

Classification problem. A predictive modeling problem for which the predicted outcome is a class label or category.

Complexity parameter (cp). A parameter used to control the size of the decision tree and select the optimal tree size. If the cost of adding another split to the decision tree is above the value of cp, then the CART analysis does not implement the split.

Cross-validation. A technique to evaluate models by training several models on subsets of the available input data and evaluating them on the complementary validation subset of data. In k-fold cross-validation, the input data splits into k subsets of data called folds, and each of the k folds serves as the validation data to evaluate the trained model exactly once.

Decision node. A node in a decision tree that uses a decision rule to split data.

Glossary of Terms

Decision rule. The condition evaluated on a characteristic in a decision node used to split data and send portions along different paths.

Decision tree. A visual representation of data split into groups based on a set of decision rules. Classification trees represent categorical outcomes, and regression trees represent continuous outcomes.

False negative. When an observation classified as negative is actually positive, such as predicting that an at-risk student is not at risk.

False positive. When an observation classified as positive is actually negative, such as predicting that a student not at risk is at risk.

False positive rate. The ratio of false positives to the total number of actual negative events.

Impurity. A measure of the amount of predictive error in a group. It is 0 when all members of a group have a predicted value of the outcome that is the same as the actual value of the outcome. CART analysis models use a measure of impurity to determine the optimal split for each node of a classification problem. The most common measures of impurity used by CART analysis are entropy and Gini.

Loss function. A method of evaluating how well a specific algorithm models the given data. The larger the value of the loss function, the larger the deviations between predictions and actual outcomes. An algorithm's default loss function can change so that some errors are more costly than others.

Model. The result of running a machine learning algorithm on a set of data used to make predictions for unseen data.

Negative predictive value. The share of all actual values of negative predicted to be negative.

Node. A point in a decision tree where a group splits.

One standard error rule. A method for selecting the final model as the simplest model within one standard error of the optimal model, allowing for the selection of a less complicated tree with minimal loss in predictive accuracy.

Optimal model. The model that minimizes the loss function using the training data.

Outcome. The feature of interest in a dataset. CART analysis uses data for a prior cohort of students to learn patterns and discover relationships between other features of the dataset and the outcome.

Overfitting. When a model fits the training data too well. It occurs when a model aligns itself to the idiosyncrasies of the training data too closely, thus negatively impacting its performance on new data.

Glossary of Terms

Parameters. Features or rules that a user can change to control the algorithm.

Parent node. The node containing the decision rule that creates child nodes.

Partition. Splitting the data into subsets for specific purposes, such as training and testing.

Penalty. An adjustment to the loss function that specifies how much weight to give to a particular type of error.

Positive predictive value. The share of all actual values of positive predicted to be positive.

Probability threshold. The value that governs the conversion of a predicted probability into one class label or another. The default value of 0.5 may not represent the optimal way of applying classifications.

Receiver operating characteristic (ROC) curve. Shows the performance of a classification model at all probability thresholds for classification. It plots the true positive rate against the false positive rate.

Root node. The starting node in a decision tree that contains the entire sample for the analysis.

Sensitivity. The measure of the proportion of actual positive cases predicted as positive, also known as true positive rate or recall.

Specificity. The measure of the proportion of actual negative cases classified as negative, also known as the true negative rate.

Stopping rule. For CART analysis, the criterion that specifies when to stop splitting the data. It is frequently based on the complexity parameter but potentially controlled by other requirements such as minimal node size.

Terminal node. A node at the bottom of the tree which does not split the data further.

Testing data. A set of data observations used at the end of model training and tuning to assess the model's predictive power on unseen data.

Training data. A set of data observations used to determine the optimal model.

Training the model. The process of determining the optimal model by running the algorithm on the training data.

True negative. An observation correctly classified as negative, such as predicting that a student not at risk is not at risk.

Glossary of Terms

True positive. An observation correctly classified as positive, such as predicting that an at-risk student is at risk.

True positive rate. The ratio of true positives to the total number of actual positive events.

Tuning the model. The process of maximizing the model's performance without overfitting, accomplished by adjusting the parameters of the process and identifying the optimal model across all trainings.

Validation. The set of processes intended to verify that models are performing as expected.

REFERENCES

- Breiman, L. (1996). Technical note: Some properties of splitting criteria. *Machine Learning*, 24, 41–47. <https://link.springer.com/content/pdf/10.1007/BF00117831.pdf>
- Breiman, L., Friedman, J. H., Olshen, R. A., & Stone, C. (1984). *Classification and regression trees*. Chapman & Hall CRC.
- Bramer, M. (2007). *Principles of Data Mining*. Springer, Verlag London.
- Claessens, A., & Engel, M. (2013). How important is where you start? Early mathematics knowledge and later school success. *Teachers College Record*, 115(6). <https://eric.ed.gov/?id=EJ1020177>
- Dekker, G. W., Pechenizkiy, M., & Vleeshouwers, J. M. (2009, July). *Predicting students drop out: A case study*. Paper presented at the Second International Conference on Educational Data Mining, Cordoba, Spain. <https://eric.ed.gov/?id=ED539082>
- Duncan, G. J., & Magnuson, K. (2011). The nature and impact of early achievement skills, attention skills, and behavior problems. In G. J. Duncan & R. J. Murnane (Eds.), *Whither Opportunity? Rising Inequality, Schools, and Children's Life Chances*, (pp. 47–69). Russell Sage Foundation and Spencer Foundation.
- Frye, D., Baroody, A. J., Burchinal, M., Carver, S. M., Jordan, N. C., & McDowell, J. (2013). *Teaching math to young children: A practice guide* (NCEE 2014–4005). U.S. Department of Education, Institute of Education Sciences, National Center for Education Evaluation and Regional Assistance (NCEE). <https://ies.ed.gov/ncee/wwc/PracticeGuide/18>
- Gomes, C. M. A., & Almedia, L. S. (2017). Advocating the broad use of the decision tree method in education. *Practical Assessment, Research and Evaluation*, 22(10), 1–10. <https://eric.ed.gov/?id=EJ1160667>
- Gordon, L. (2013). *Using classification and regression trees (CART) in SAS Enterprise Miner for applications in public health* (Paper No. 089–2013). University of Kentucky. <https://support.sas.com/resources/papers/proceedings13/089-2013.pdf>
- James, G., Hastie, T., Witten, D., & Tibshirani, R. (2015). *An introduction to statistical learning*, 6th ed. Springer.
- Koon, S., & Davis, M. (2019). *Math course sequences in grades 6–11 and math achievement in Mississippi* (REL 2019–007). U.S. Department of Education, Institute of Education Sciences, National Center for Education Evaluation and Regional Assistance, Regional Educational Laboratory Southeast. <https://eric.ed.gov/?id=ED597299>

References

- Koon, S., & Petscher, Y. (2015). *Comparing methodologies for developing an early warning system: Classification and regression tree model versus logistic regression* (REL 2015-077). U.S. Department of Education, Institute of Education Sciences, National Center for Education Evaluation and Regional Assistance, Regional Educational Laboratory Southeast.
<https://eric.ed.gov/?id=ED554441>
- Koon, S., and Petscher, Y. (2016). *Can scores on an interim high school reading assessment accurately predict low performance on college readiness exams?* (REL 2016-124). U.S. Department of Education, Institute of Education Sciences, National Center for Education Evaluation and Regional Assistance, Regional Educational Laboratory Southeast. <https://eric.ed.gov/?id=ED565632>
- Koon, S., Petscher, Y., & Foorman, B. R. (2014). *Using evidence-based decision trees instead of formulas to identify at-risk readers* (REL 2014-036). U.S. Department of Education, Institute of Education Sciences, National Center for Education Evaluation and Regional Assistance, Regional Educational Laboratory Southeast. <https://eric.ed.gov/?id=ED545225>
- Lakkaraju, H., Aguiar, E., Shan, C., Miller, D., Bhanpuri, N., Ghani, R., & Addison, K.L. (2015, August). A machine learning framework to identify students at risk of adverse academic outcomes. Paper presented at the 21st ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Sydney, NSW, Australia.
<http://www.dssgfellowship.org//wp-content/uploads/2016/04/montgomery-kd2015.pdf>
- Lemon S. C., Roy, J., Clark, M. A., Friedmann, P.D., & Rakowski, W. (2003, December). Classification and regression tree analysis in public health, Methodological review and comparison with logistic regression. *Annals of Behavioral Medicine*, 26(3), 172–181.
https://doi.org/10.1207/S15324796ABM2603_02
- Polat, C. (2018). Performance evaluation of logistic regression, linear discriminant analysis, and classification and regression trees under controlled conditions. (Doctoral dissertation, University of Denver). Retrieved from University of Denver Electronic Theses and Dissertations.
<https://digitalcommons.du.edu/etd/1503/>
- Quadril, M. N., & Kalyankar, N. V. (2010). Drop out feature of student data for academic performance using decision tree techniques. *Global Journal of Computer Science and Technology*, 10(2), 2–5.
<https://pdfs.semanticscholar.org/56f0/4ece793c3b4c8f329300fc805d622fc7e029.pdf>
- Siegler, R. S., Duncan, G. J., Davis-Kean, P. E., Duckworth, K., Claessens, A., Engel, M., Susperreguy, M. I., & Chen, M. (2012). Early predictors of high school mathematics achievement. *Psychological Science*, 23(7), 691–697. <https://eric.ed.gov/?id=ED552898>
- Song, Y., & Lu, Y. (2015). Decision tree methods: Applications for classification and prediction. *Shanghai Archives of Psychiatry*, 27(2), 130–135.
<https://www.ncbi.nlm.nih.gov/pmc/articles/PMC4466856/pdf/sap-27-02-130.pdf>

Acknowledgments

The authors offer special thanks to Aaron Butler, Sharon Koon, Mya Martin-Glenn, and Deborah Jonas, who provided critical feedback.

REL 2022-133

December 2021

This resource was prepared for the Institute of Education Sciences (IES) under Contract ED-IES-17-C-0004 by Regional Educational Laboratory Appalachia administered by SRI International. The content of the publication does not necessarily reflect the views or policies of IES or the U.S. Department of Education nor does mention of trade names, commercial products, or organizations imply endorsement by the U.S. Government.

This REL resource is in the public domain. While permission to reprint this publication is not necessary, it should be cited as:

Seftor, N., Shannon, L., Wilkerson, S., & Klute, M. (2021). *Branching out: Using decision trees to inform education policy choices* (REL 2022-133). U.S. Department of Education, Institute of Education Sciences, National Center for Education Evaluation and Regional Assistance, Regional Educational Laboratory Appalachia. Retrieved from <http://ies.ed.gov/ncee/edlabs>.

This resource is available on the Regional Educational Laboratory website at <http://ies.ed.gov/ncee/edlabs>.