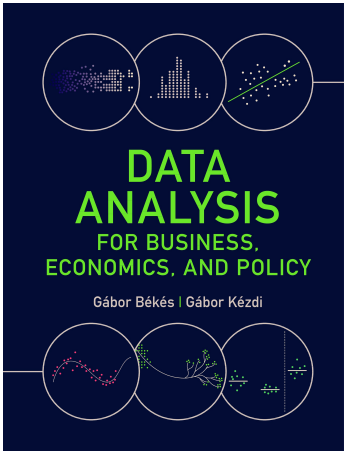
15. Introduction to Machine Learning

Gábor Békés

Data Analysis 3: Prediction

2020

Slideshow for the Békés-Kézdi Data Analysis textbook



- Cambridge University Press, 2021
- **gabors-data-analysis.com**
  - Download all data and code: gabors-data-analysis.com/data-and-code/
- This slideshow is for **Chapter 15**

Regression, LASSO, machine learning

- Regression
  - Analyst defines variables via feature engineering, including functional form selection (polynomials, splines, interactions)
  - Analyst defines a set of possible models
  - Analyst selects best model trying out various options (Best model selected by cross-validated RMSE)

Regression, LASSO, machine learning

- Regression
  - Analyst defines variables via feature engineering, including functional form selection (polynomials, splines, interactions)
  - Analyst defines a set of possible models
  - Analyst selects best model trying out various options (Best model selected by cross-validated RMSE)
- Regression with LASSO
  - Analyst defines variables via feature engineering, including functional form selection (polynomials, splines, interactions)
  - Analyst defines broadest model
  - Algorithm selects best model

Regression, LASSO, machine learning

- Regression
  - Analyst defines variables via feature engineering, including functional form selection (polynomials, splines, interactions)
  - Analyst defines a set of possible models
  - Analyst selects best model trying out various options (Best model selected by cross-validated RMSE)
- Regression with LASSO
  - Analyst defines variables via feature engineering, including functional form selection (polynomials, splines, interactions)
  - Analyst defines broadest model
  - Algorithm selects best model
- Machine learning (Random Forest)
  - Analyst defines set of variables. No functional form selection.
  - Algorithm defines a variety of possible models.
  - Algorithm selects best model (Best model selected by cross-validated RMSE)

How does ML work?

- Regression – finds a single solution.
  - You run an OLS regression and it yields one single output (estimated coefficients), calculated by formulae.
  - LASSO a numerical algorithm finds coefficients and the tuning parameter.
- Machine learning is different
  - No single best solution by formulae
  - Search through a set of possible prediction models
    - That best captures the relationship.
- In terms of prediction
  - Both estimate the model on the training set
  - And we avoid overfitting on the test set (and hope for external validity).

Typical ML prediction procedure

- We start with the data at hand.
- First, cut off the hold-out set.
- For what remains, we create k-times a training and test datasets, building the model on train set, and evaluating it on the test sets.
- We take the average of test set loss functions (e.g. MSE) and select the model with the best fit.
- Then, we take that very model, re-estimate it on our dataset save the holdout sample to get parameter estimates.
- Finally we evaluate it on the hold-out sample to estimate the fit we can expect on the live data.

Roadmap

- Growing one tree [lecture 3]
  - The idea of regression trees (or CART)
  - Understand ML basics - finding an optimal relationship by trying
  - How to develop (grow) a tree
- How to make it less prone to overfitting [lecture 4]
  - Go back to random sampling, and bootstrap
  - Random forest (or RF)
  - Boosting: GBM

Machine learning: Regression Trees with CART

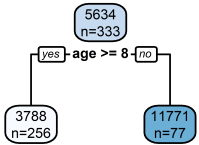
- Regression Tree: Basic idea is that relationships are modelled as a series of binary decisions (splits)
- Classification and Regression Trees (CART) is a regression tree algorithm

- Introduce cart with the case study

Case Study: CART for used cars price prediction

- Case study: used-cars
- The used-cars data includes data on offers of used Toyota Camry cars advertised in the Chicago and Los Angeles areas, in 2018.
- The dataset has N=477 observations.
- Small dataset + simplicity: Single training–test split instead of k-fold cross-validation. Our training set is a random 70 percent of the original data (N = 333).

Case Study: CART for used cars price prediction



- Relationships between y and an x are modeled as a series of binary decisions (splits)
  - Is the car age below or above 8

Regression tree basics

How are trees and cuts created?

- You can't try out *all* possible segmentation combinations.
  - Why? Because, even if we limit the number of nodes, it is computationally infeasible in most of the cases.
- CART offers a *process* to try out *many* options and pick a set of decision rules
- The outcome of CART is a set of a prediction **rules** as well as predicted values for y

- CART has no formulae
- Not coefficients anymore

Today: Understanding CART in a few steps

- Single predictor
- Multiple predictors
- Looking into the process - which variable matters?
- Dealing with overfitting

- Will show case study along with process

CART with a single predictor

- CART is based on a **binary splitting algorithm**.
- Take a predictor, find a cut-off
- Create two bins - below, above the cutoff
- What is the optimal cut-off?
- The one that creates a separating rule that yields the greatest predicting power
- Predicted value is calculated as mean price at each bin.
  - Why mean? Because it minimizes RMSE given cutoff values.

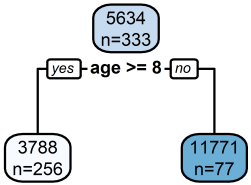
CART with a single predictor

- Separation process
- We have a set of predictors, $x_1 ... x_p$
- Start with x_1
  - Look through all values.
    - For $x_1 = k$, see calculate RMSE
  - Pick the value that yields the model with smallest RMSE.
- Repeat for all predictors
- Pick the predictor AND the cutoff, that yields the the smallest RMSE.

CART with a single predictor

- How we make predictions?
- For each bin, the predicted price will be the simple average of observations in that bin.

CART with a single predictor



- We have a set of predictors, $x_1 ... x_p$
  - Start with x_1 and pick the value that yields the model with smallest RMSE.
  - Repeat for all predictors.
  - Pick the predictor AND the cutoff, that yields the the smallest RMSE.
- Here: check on dozen variables we have
- Pick **age** of car AND 8 yr as cut-off
  - 1-7 – right group, average price is 12 thousand dollars (77 obs).
  - 8 and more - left group, average price is 4 thousand (256 obs).

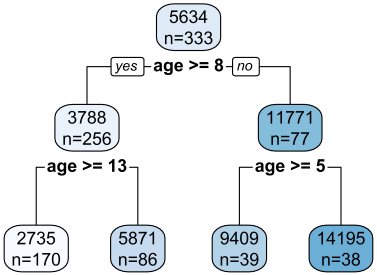
CART vs Linear Regression 1

- Nr.1. Fitting rule: similar

- There is a basic similarity.
- Both aim at finding a model to minimize a loss function.
- It is the same loss function of sum of squared residuals (MSE).

- But regression finds a global optimum while CART does not
  - because it thinks one step ahead only

CART with a single predictor



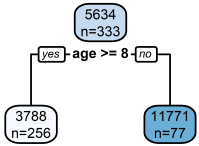
- As with bins, we can have more than 2.
- Once again, separate step by step
- We now have 3 bins (called nodes).
- We start again, and separate at a node, creating 4 bins (nodes).
- We repeat this, till a stopping point.

Measuring fit and stopping rules

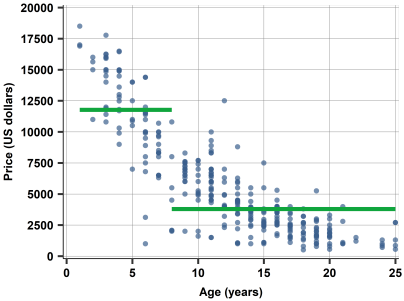
- A regression tree is the result of a series of binary splits of the sample
- The subsamples that result from splits are called bins and are represented as nodes of the tree
- With a single binary x, there is only one split possible; thus the tree has two levels with two terminal nodes
- With a single non-binary x, the algorithm starts with one split that improves the fit the most,and carries on to further splits within each bin
- The splitting algorithm stops as dictated by a stopping rule
- The result of the algorithm is a set of bins (the terminal nodes) that cover the entire sample. The predicted $\hat{y}$ values are the average y value within each bin

CART with a single predictor (age)

CART-decision tree

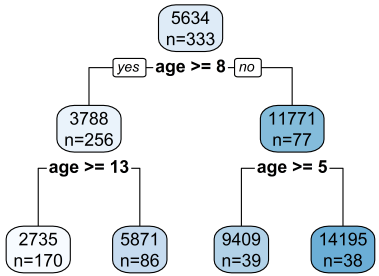


Step function

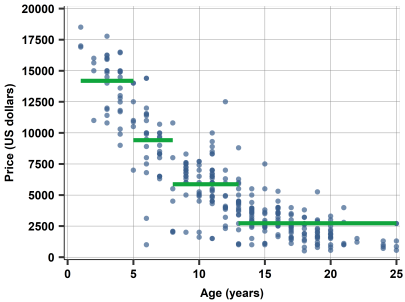


CART with a single predictor (age)

CART-decision tree



Step function



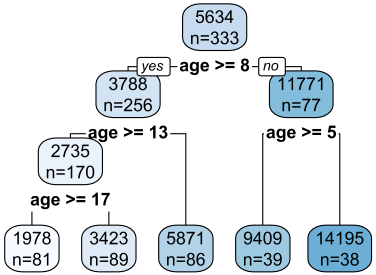
CART M2 output summary

Category	Number of observations	Average_price
Age 0-4	38	14194.84
Age 5-7	39	9408.56
Age 8-12	86	5870.52
Age 13 or more	170	2734.54

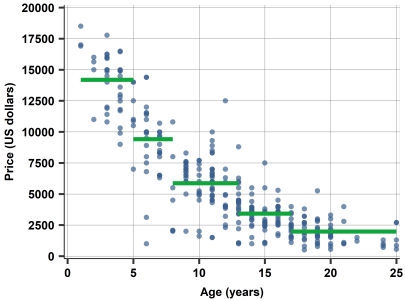
Note: The predicted y values from CART M2, a regression tree grown by CART allowing for three levels.

CART with a single predictor (age)

CART-decision tree

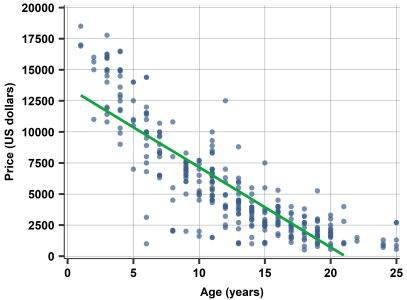


Step function

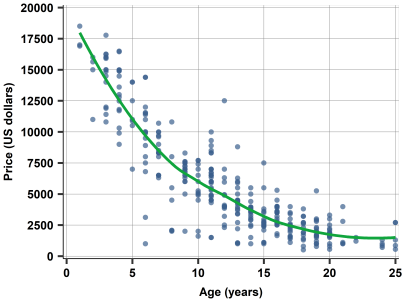


CART with a single predictor (age)

Linear fit



Step function



Predicting price with age: a comparison of CART and OLS models

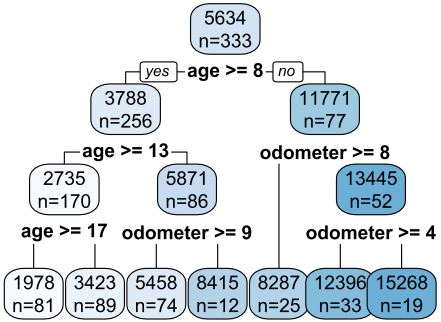
Model	Description	RMSE
CART M1	2 term. nodes	2781.06
CART M2	4 term. nodes	2074.46
CART M3	5 term. nodes	1969.52
OLS M1	1 variable only	2357.01

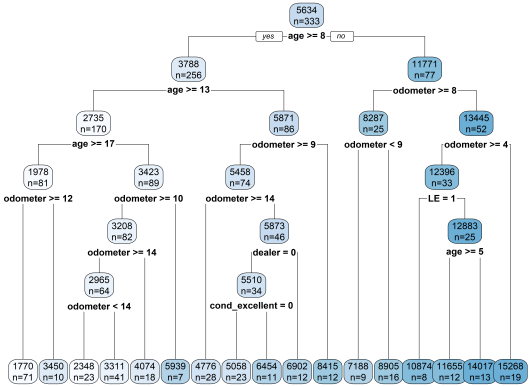
Note: Age is the only predictor. RMSE from single 30% test set.

Multiple predictors

- With multiple predictors, as we grow trees, at each step, we consider
  - all the terminal nodes
  - all predictors and
  - all possible cut-offs for each and every predictor
- Pick the node-predictor-cutoff that leads to the largest drop in MSE.
- Repeat.

Multiple predictors





Multiple predictors

- With multiple predictors, we consider all predictors and possible cut-offs step by step and grow trees.
- The hard part is to know when to stop.
- Every split will improve fit.
- But: overfitting...
- So we employ a stopping rule.

Stopping rule and prediction

- The binary splitting process continues until a stopping criterion is reached.
- Minimum number of observations in a bin for further splitting
- The number of observations in any terminal node.
- Minimum fit improvement - a split is made only if it improves the fit of the by a minimum amount = the complexity parameter(cp)
- Stopping rule strictness will define size of tree.

CART process review with some jargon

- The method is called **binary splitting**. It is a **top-down**, **greedy** approach.
- **Top-down** because it begins at the top of the tree and then successively splits the predictor space; each split is indicated via two new branches further down on the tree.
- **Greedy** because at each step of the tree-building process, the best split is made at that particular step, rather than looking ahead and picking a split that will lead to a better tree in some future step.
  - Disadvantage: Myopic–does not take into account how a split(r) will affect a split (r+1, r+2, etc)
  - Advantage: it's (very) fast

CART vs Linear regression

- . Output: different but with a similarity

- Output for OLS will be a set of coefficient estimates
- Output for CART will be a set of decision rules.

- Similarity
- We can take output that we get in the training data and use it on the test data.
  - We also use it on the live data.

CART vs OLS: logic

Predictions: different logic

- For regression
  - if $x_1 \neq x_2$
  - then $\hat{y}_1 \neq \hat{y}_2$
- For CART
  - if $x_1 \neq x_2$
  - BUT x_1, x_2 in $R_i x$
  - then $\hat{y}_1 = \hat{y}_2$

- examples for cars
  - age=2, age=3
  - price will be different by 0.5K
    - age=2, age=3
    - both in node age<3.5
    - price is same (14K)

CART overview

- The advantage of CART is pattern discovery and easy interpretation.
- Sometimes easier than linear regression
- Decision trees (=bins) - easy to explain

- As a prediction method, performance is not so great.
- But CART is building block of some top performing ML tools.

What's wrong with trees?

- Two key problems
- Overfitting
  - Remember the price - age step-function with 9 nodes.
- Splitting is not robust.
  - Change a few variables, and a split might alter at another point
  - Leading to a completely different tree!
  - This comes from the fact that a split at t, does note take into account future options.

Improving on CART

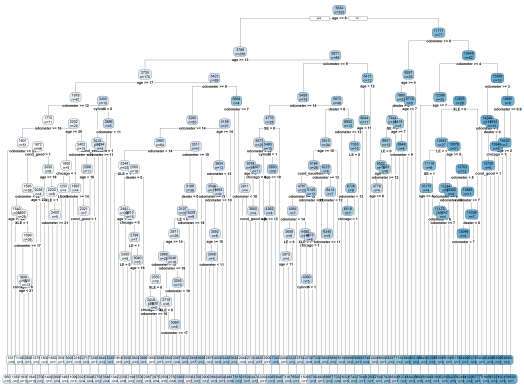
- A smaller tree with fewer splits might be a better idea
  - More stability
  - Worse in sample fit
- Solution 1: early (tough) stopping rule
  - Stop when drop in RSS is below a threshold (cp)
  - Short-sighted: a split now may be not so important but followed by an important split later
- Solution 2: grow a tree and cut back = pruning
  - Grow a tree as large as possible
  - Cut back
  - Turns out, this is better...

Pruning the tree

- Grow a large tree first, with a stopping rule that lets it grow big
- and prune it *afterwards* by deleting some of the splits, with the goal of arriving at a better prediction..
- **Cost complexity pruning** (= weakest link pruning) is used to do this.
- Pruning produces a better model

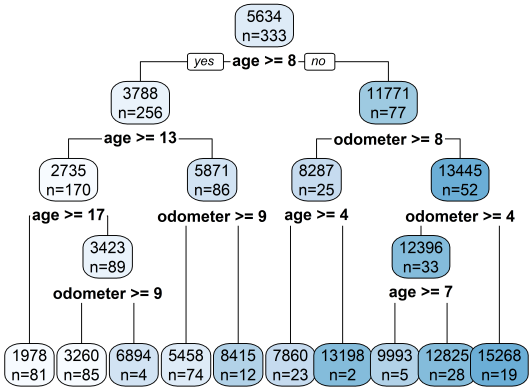
Pruning

- Grow a very large tree:
- Cp very low and/or stop bucket very low



Pruning

- Grow a very large tree
- Cp very low and/or stop bucket very low
- Prune: get to fewer nodes.



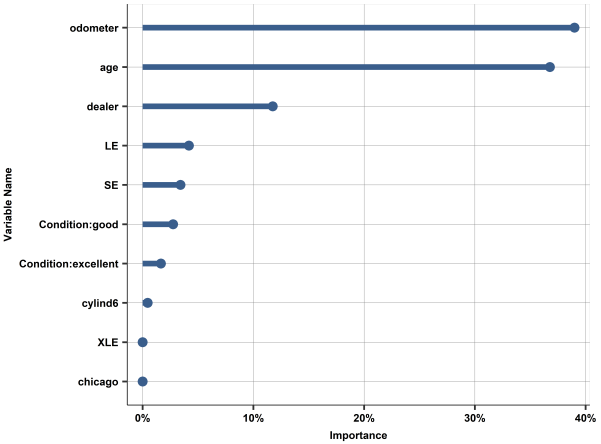
Variable importance

- Learning how predictions are made is rather difficult.
  - Unlike for a regression model, we do not have coefficient estimates.
  - A variable appear at many splits.

- **variable importance:** measures how much fit improved when a particular x variable is used for splitting, summed across all splits in which that variable occurs.
  - Expressed as the share of fit improvement (MSE reduction) due to the particular x variable
  - relative to the overall improvement of fit achieved by the regression tree
  - as opposed to a prediction that doesn't use any x variables
- The measures how important a variable is in terms of helping improve prediction
- High variable importance means that given variable plays an important role in prediction.

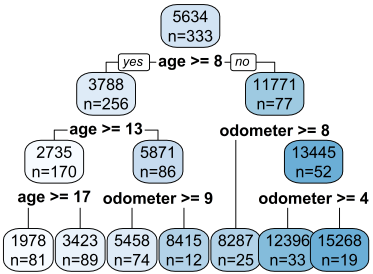
Variable importance plot

- Variable importance for a regression tree on the holdout set.
- Odometer and age are about equally important predictors while other variables help substantially less



Variables used

- Important variable are used early and frequently
- Here: **age:odometer**



A regression tree is a non-parametric regression

- A regression tree with multiple x variables considers all splits across all x variables and chooses the split that improves the fit the most
- The regression tree is a non-parametric regression, one which approximates any functional form with a step function and includes interactions
- A regression tree grown with CART includes only the most important interactions and the most important steps of a step function
- Pruning means first growing a large regression tree with a lenient stopping rule, and then erasing final splits one by one to improve fit in the test set (or cross-validated test sets)
- Pruning tends to produce a better fit in the test sets than a strict stopping rule, but even pruning tends to leave regression trees overfitting the original data

Pros and Cons of Using a Regression Tree for Prediction

Feature	Linear regression (OLS)	Regression tree (CART)
Solution method	Formula	Algorithm without a formula
Solution goal	Minimize loss (MSE)	Minimize loss (MSE)
Solution optimality	Finds best possible linear regression, but only given the included x variables	Greedy algorithm; does not find best possible tree
Variable selection	Pre-defined list of x variables	No pre-defined list
Main results	Set of coefficient estimates; prediction by plugging into formula	Set of terminal nodes; prediction by specifying values of x variables
Predicted y value and x values	Different $\hat{y}$ for different x values	May have same $\hat{y}$ for different x values if in the same bin
Linear relationship between x and average y	Captures a linear relationship	Approximates linearity by step function
Nonlinear relationship between x and average y	Need to pre-specify functional form to approximate nonlinearity	Approximates nonlinearity by step function

Case Study: Regression tree summary and discussion

Model	Number of variables	Describe	RMSE
CART M1	1	2 levels	2781.06
CART M2	1	3 levels	2074.46
CART M3	1	4 levels	1969.52
CART M4	7	cp = 0.01	1892.96
CART M5	7	cp = 0.002	1892.35
CART M6	7	cp = 0.0001	2072.60
CART M7	7	pruned	1818.09
OLS M2	7	linear	1905.85
OLS M3	7	w/ polynomial terms	1636.50

Predictive performance of all regression tree models by the test set RMSE.

Case Study: Regression tree summary and discussion

- Pruned CART or small CART are best, pruned is marginally best
- OLS with some feature engineering (functional form for key vars) are better than any CART
- Advantage of CART is that it is automatic, no need to look for functional form. But worse performance than OLS *with* feature engineering.

Main takeaways

- A regression tree is a method for predicting y based on x variables using an automated algorithm that approximates any functional form and any interaction between the x variables
  - A regression tree splits the sample into many bins according to values of the x variables and predicts y as the average within those bins
  - Regression trees can be thought of as non-parametric regressions
  - A regression tree is prone to overfitting the data even after pruning. For this reason, it is rarely used for prediction in itself