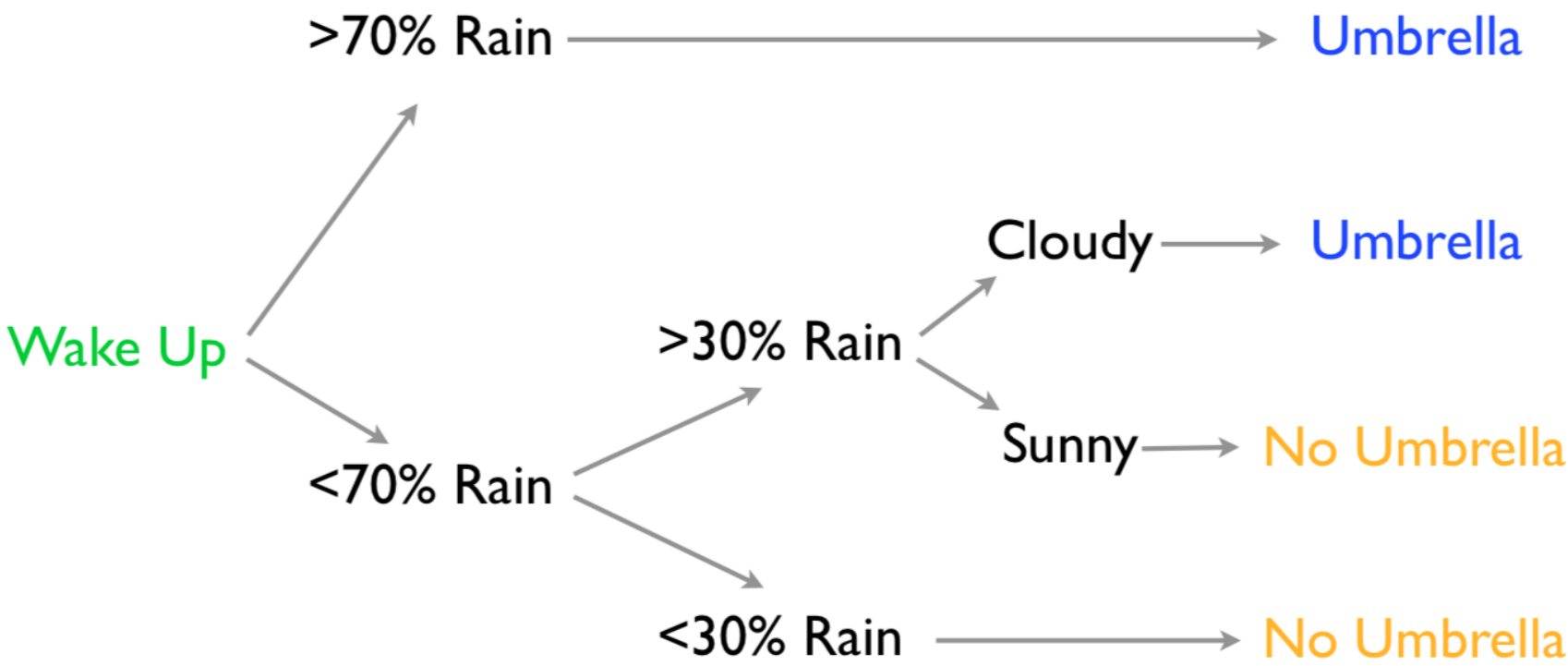
Machine Learning: Trees and Forests

Introduction

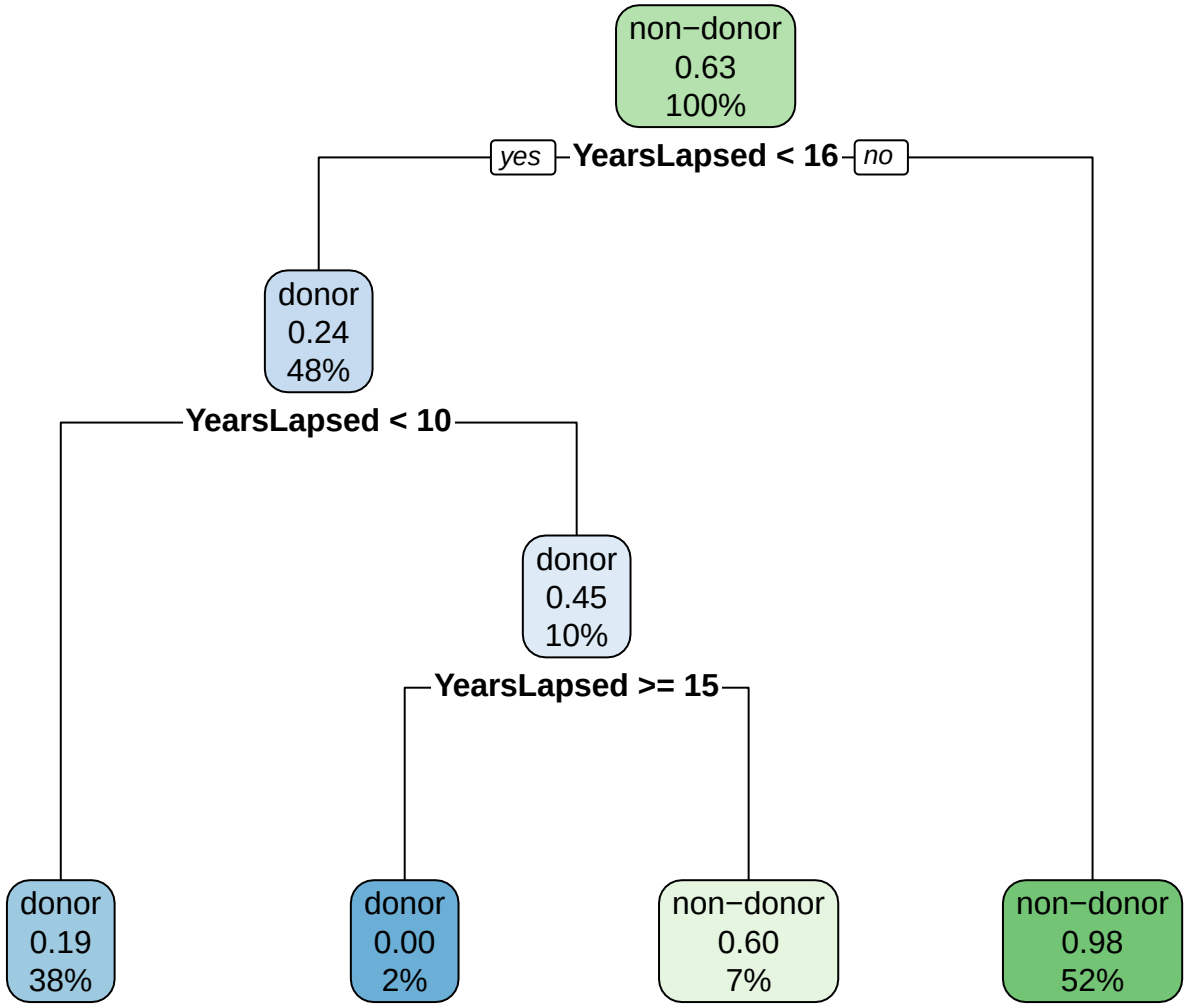
- Decision Trees: Using tree-logic to make predictions
  - CART: Classification and Regression Trees
  - Random Forests: averaging over many possible trees

What is a Decision Tree?



- Tree-logic uses a series of steps to come to a conclusion.
- The trick is to have mini-decisions combine for good choices.
- Each decision is a node, and the final prediction is a leaf node

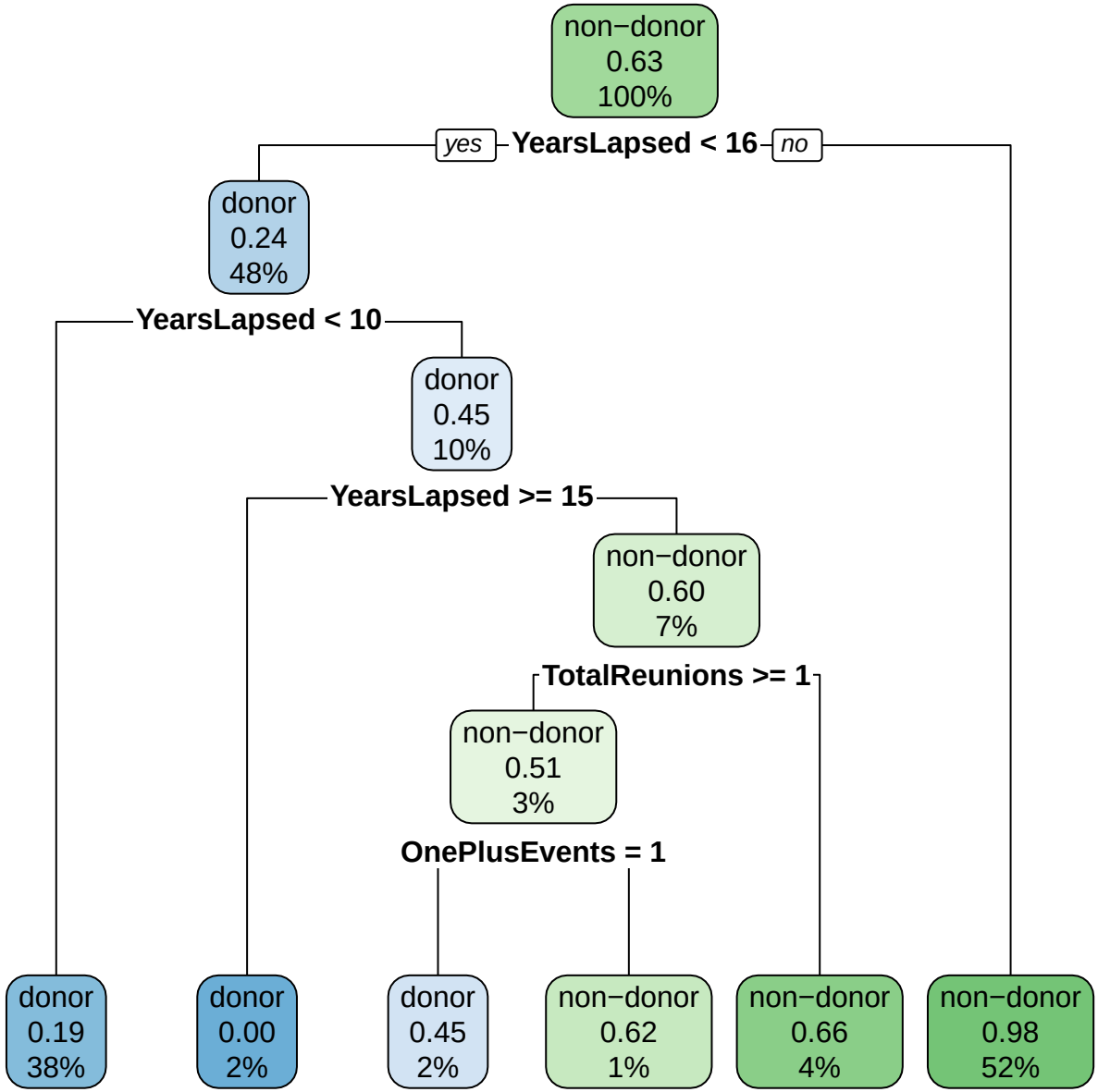
Example Tree Based on Alumni Data



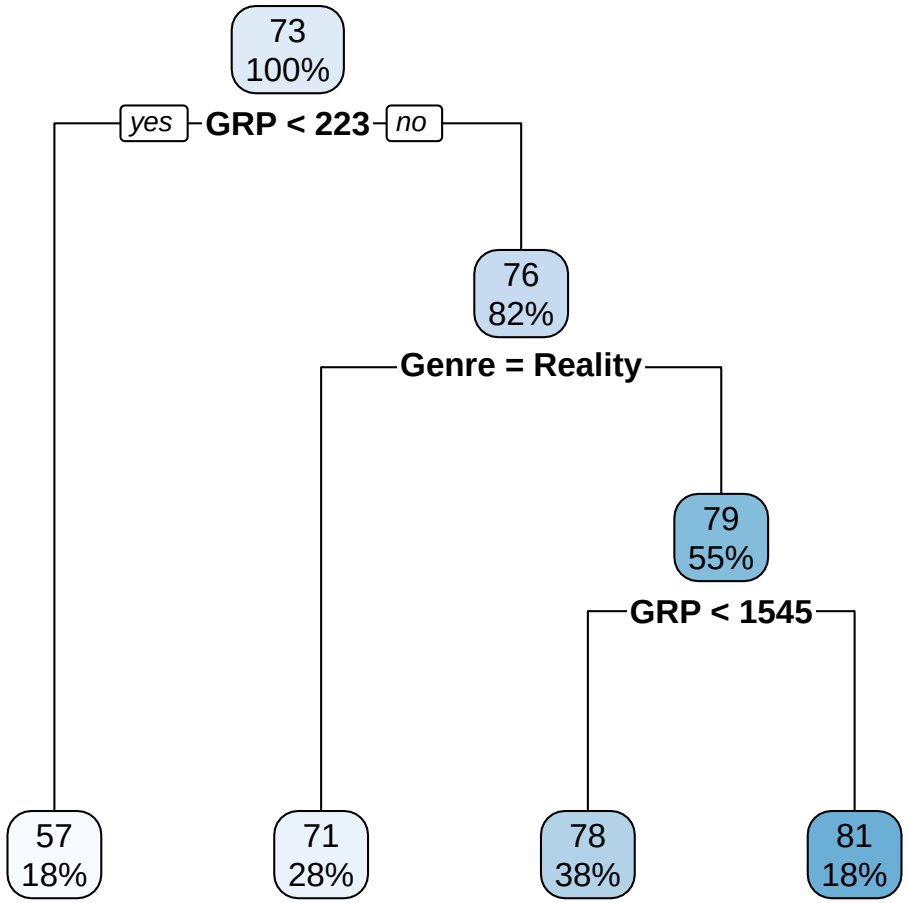
Decision Tree Basics

- The plot above is called a **dendrogram** and shows:
  - Probability of being a non-donor (middle line in each node)
  - Share of the total population in a given node (last line in the node)
- Standard elements of a Tree
  - **Root Node**: the starting point; the entire population
  - **Internal nodes**, labeled with one of the inputs
  - **Terminal nodes** or "leaves"
- **Splitting Rules**
  - These create the two branches out of a given internal node
  - E.g. $<$70% Rain
- **Predictions**
  - Outcomes are predicted by using the average y_i in each terminal node

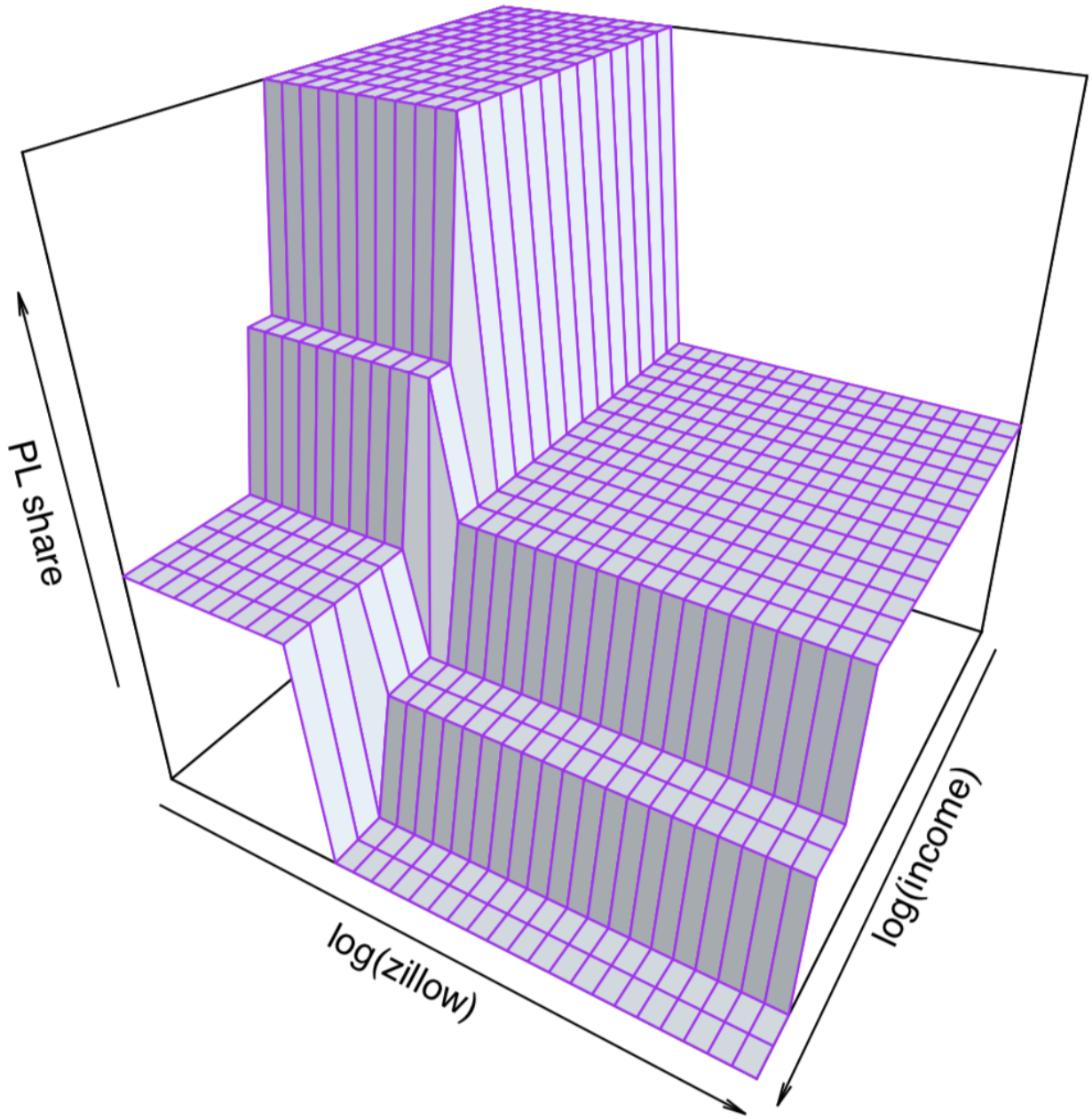
A More Complex Tree for Alumni



Continuous Outcomes: Engagement in TV shows



Trees capture complex interactions between variables



Algorithm for Growing a Regression Tree

- Start at the root node and successfully add splits and thus additional nodes
- How to add splits:
  - Focus on one terminal node in the current tree, and its data $\mathscr{R}$
  - For each X variable in the data set, search for a split point that minimizes the RSS (Sum of Squared Residuals):

$$\text{RSS} = \sum_{i \in \mathscr{R}_a} (y_i - \hat{y})^2 + \sum_{i \in \mathscr{R}_b} (y_i - \hat{y})^2$$

  - where $\mathscr{R}_a$ is all the data with $x_i < s$ and $\mathscr{R}_b$ is all the data with $x_i >= s$ for some split point s
  - pick the x_i and s that together minimize the RSS and use the x_i and s to split the node

Regression Trees: Discussion

Benefits of regression trees:

- Flexible (non-parametric) regression functions
- Allows for interactions between inputs
- Built-in variable selection. (Unpredictive variables are not used for splits.)
- Can be easily interpreted if the tree size is small
- Tress can be displayed graphically – this is great to communicate the estimation results (if the tree size is not too large)

Polling questions

Access the poll here:

www.pollEv.com/mthomas

- Log in with your NUS ID
- This will allow me to give credit for your participation.

Overfit

- Prediction in a regression tree is straightforward, but especially in a large regression tree (small node size) the prediction quality will be poor because of over-fitting.
- Why this happens is easy to see: In the limit, if each node is based on only one observation, the in-sample fit will be perfect! Out-of-sample, not so much.
- **Pruning** is a common method to reduce the variance of the prediction by adding a cost for growing a tree with many leaves.

Tree Pruning

First, grow a very large, complex tree. Then prune the tree back by eliminating branches to obtain a **subtree**. For any such pruned tree T with number of leaves L_T we evaluate the following **cost complexity criterion**:

$$\sum_{l=1}^{L_T} \sum_{i \in \mathscr{R}_l} (y_i - \hat{y}_{\mathscr{R}_l})^2 + \alpha L_T$$

α is the cost complexity parameter. This criterion introduces a tradeoff between in-sample fit and the cost of a complex tree with many leaves. Note that this is very similar to the penalty term in a ridge regression or LASSO that allows for regularization.
For any given *cost complexity parameter* we can find the subtree (of the original, big tree) that minimizes the cost complexity criterion.
How do we determine a good value of the tuning parameter α? — Cross-validation

Random Forests

- We discussed the advantages of trees (flexibility, tool to communicate and visualize the results), but in practice they are a poor tool for prediction.
- A random forest
  - estimates relationship between Xs and Y values
  - inherits the flexibility of trees (non-parametric estimator)
  - provides a much better out-of-sample fit than trees
- Random forests are an **ensemble method**
  - Instead of one algorithm to predict $\hat{y}$, we use an ensemble of B algorithms
  - the final $\hat{y}$ prediction is the average prediction across all of the B algorithms

Interlude: The bootstrap

- The bootstrap is a **resampling method** (like cross-validation)
  - Used to simulate the sampling distribution of an estimator
  - Often we don't know the true distribution (unless $N \to \infty$)
- How to run an **idealized** bootstrap:
  1. Obtain B (say, $B = 1000$) independent samples with N data points
  2. Calculate the estimator, $\hat{\theta}_b$, for each of the samples $b = 1, ..., B$
  3. Predict the standard error of $\hat{\theta}_b$ using the standard deviation of the B $\hat{\theta}_b$ values, calculate a 95-percent confidence interval based on the 2.5th and 97.5th percentile of the $\hat{\theta}_b$ values, etc.

The Bootstrap

- The ideal solution is infeasible because we only have one data set.
- Instead we have to create B data sets from the original population:
  1. Obtain B data sets by randomly sampling N observations from the original data (which has N observations) *with replacement*
  2. Calculate the estimator, $\hat{\theta}_b$, for each of the samples $b = 1, ..., B$
  3. Predict the standard error of $\hat{\theta}_b$ using the standard deviation of the B $\hat{\theta}_b$ values, calculate a 95-percent confidence interval based on the 2.5th and 97.5th percentile of the $\hat{\theta}_b$ values, etc.
- The bootstrap is easy to implement (on a modern computer) and tends to provide better estimates (e.g. a confidence interval) than an asymptotic approximation

Bagging

- Bagging means "bootstrap aggregation"
- The bagging algorithm:
  1. Create B training data sets using the bootstrap
  2. Estimate the prediction algorithm for each training set b
  3. Average over all B predictions
- Purpose of bagging is to reduce the variance of the prediction without making the bias (much) worse

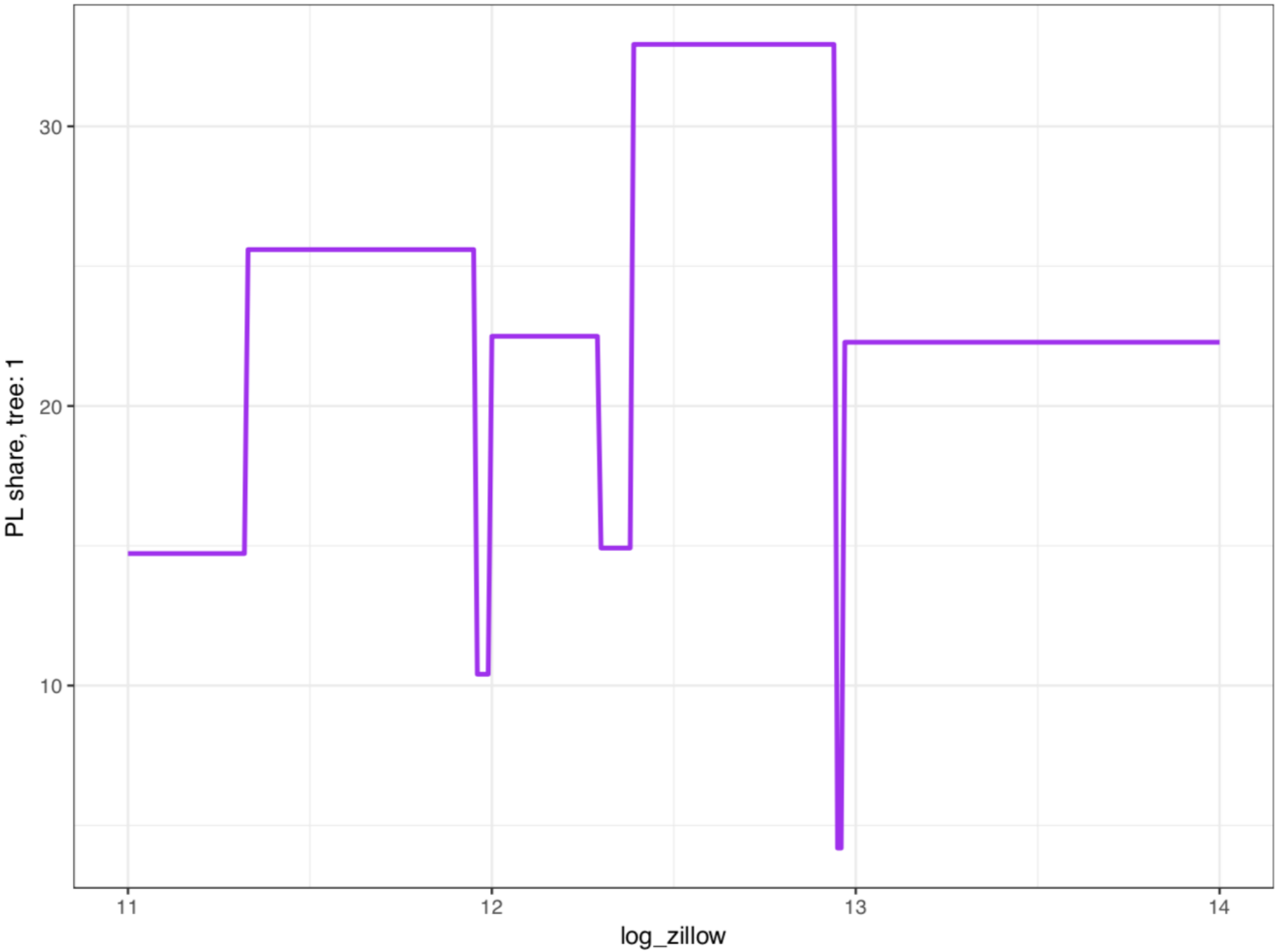
Selection of random subset of inputs

- When building a tree
  - At each attempted split the random forest algorithm will only split based on a randomly chosen subset of all p inputs (e.g., $\sqrt{p}$ or $p/3$ randomly chosen inputs).
  - This is sometimes called "feature bagging" or a "random subspace method"
- Goal:
  - Break reliance of prediction on a possible small number of "dominant" features
  - explore a wider set of inputs to determine splits
- Achieves:
  - reduced overfitting
  - reduced variance of the predictions
  - improved out-of-sample predictive power

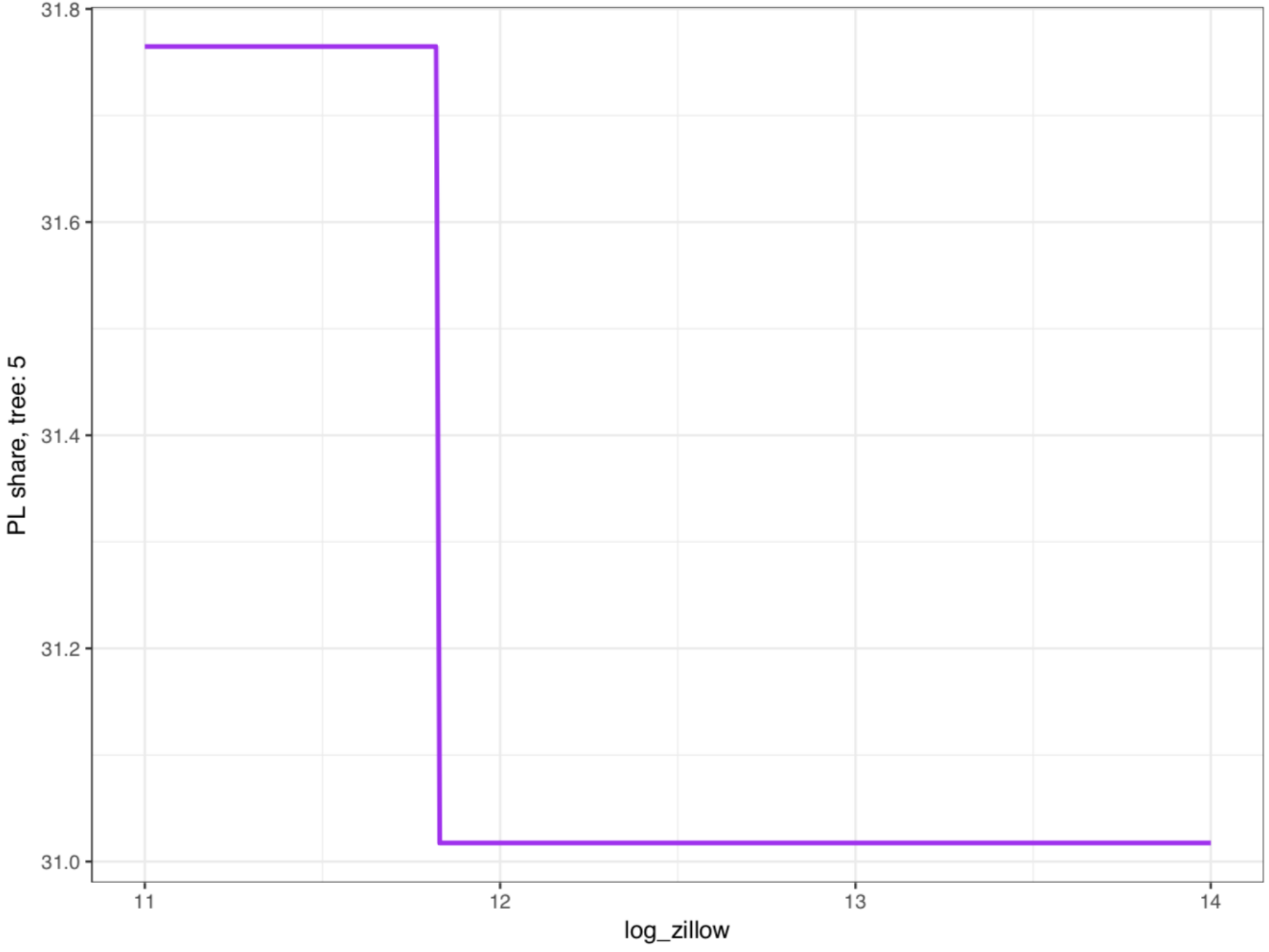
Example

- Predicting Private Label purchase share using housing prices (from Zillow)
  - Notice what each individual tree looks like
  - Notice how they aggregate to make the Random Forest prediction

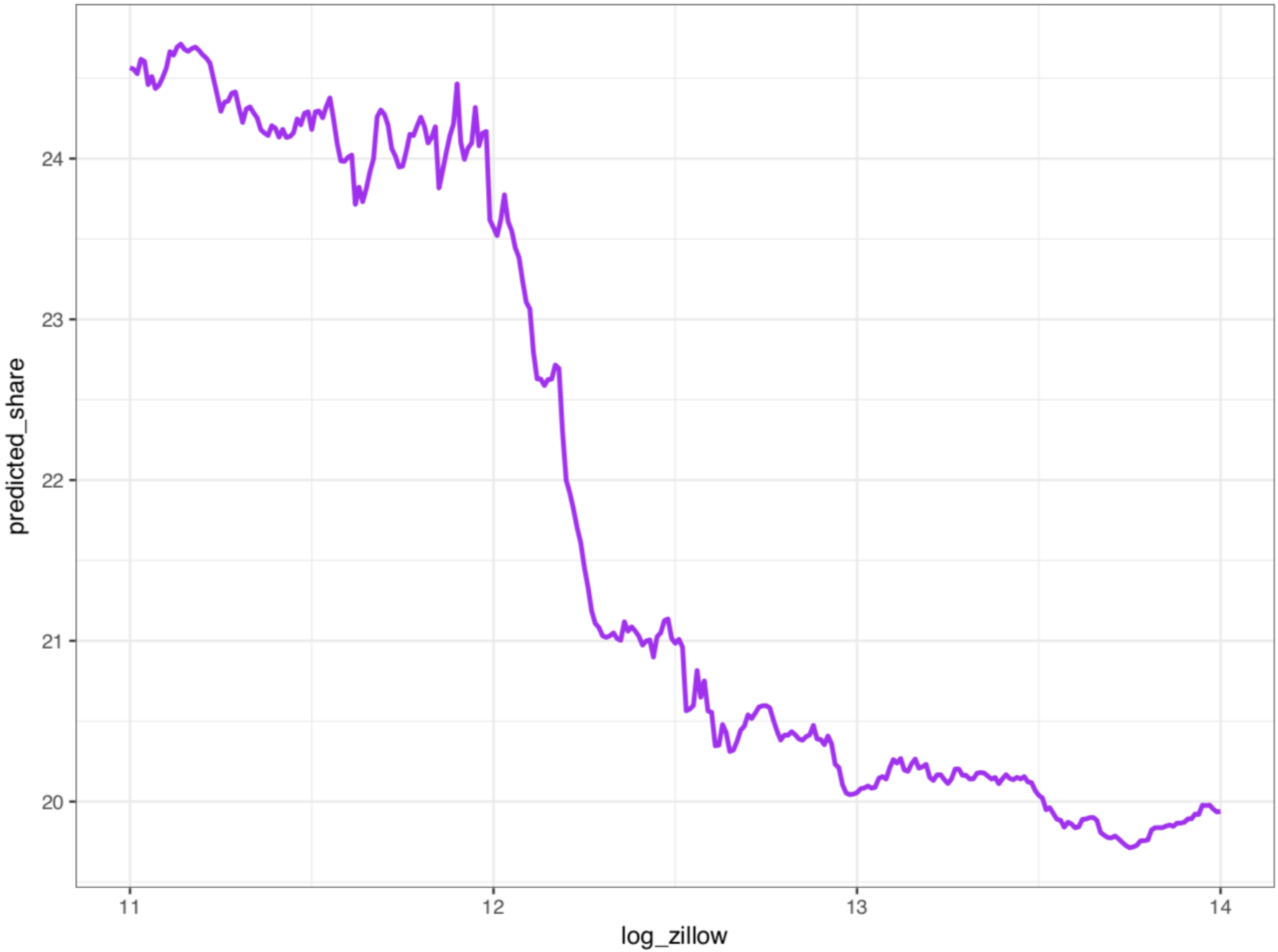
Tree 1



Tree 5



Random Forest Prediction averaged over 2000 trees



Discussion

- Note how a random forest can approximate a "continuous" relationship, even though each individual tree can't — because of bagging and random subset of features selection
- The "price" of random forest:
  - Computational costs—even on a fast computer building 1000 trees may take several hours (or more) if the data size (N and p) is large

Polling questions

Access the poll here:

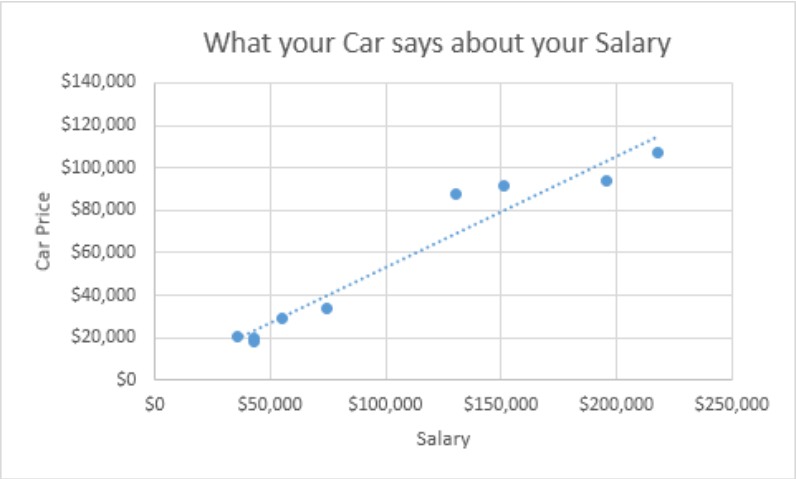
www.pollEv.com/mthomas

- Log in with your NUS ID
- This will allow me to give credit for your participation.

Summary

- Classification and regression trees (CART)
  - Flexible approximation of regression functions (non-linearities, interactions)
  - Good for visualization and communication
  - Often poor out-of-sample predictions
- Random forests
  - State-of-the-art non-parametric estimator
  - Bagging
  - Random feature subset selection
  - Computational cost can be high, but feasible to implement on a modern computer
  - Difficult to interpret how the model made its predictions

Practice question



What would be a natural split point if you were fitting a tree to these data?

* $0
* $50,00
* Ans: $100,000
* $150,000
* $200,000
* $250,000

Practice question

When splitting the training data to create a tree, we find the split that:

* Ans: Minimizes the sum of squared residuals
* Gives the best predictions in cross-validation
* Gives the best predictions in the holdout sample.
* Matches the linear model most closely.

Practice question

What is a natural way to measure the complexity of a tree?

* Ans: Count the number of leaves
* Sum the values of all the split points.
* Count the number of observations in the most populated leaf.
* Count the number of observations in the least populated leaf.

Practice question

What makes Random Forest an ensemble method?

* Ans: Averaging across many models
* Using trees
* Feature bagging
* Using bootstrapping.

Practice question

How does Random Forest make better predictions than a a single tree?
* Ans: Using the average prediction across many trees.
* Applying cross validation.
* Imposing a penalty on model complexity.
* Using a holdout sample.

Practice question

How does Random Forest approximate a continuous function?

* Ans: Averaging across the predictions of many trees.
* Using cross validation.
* Mixing trees with linear models.
* Drawing random predictions from trees.