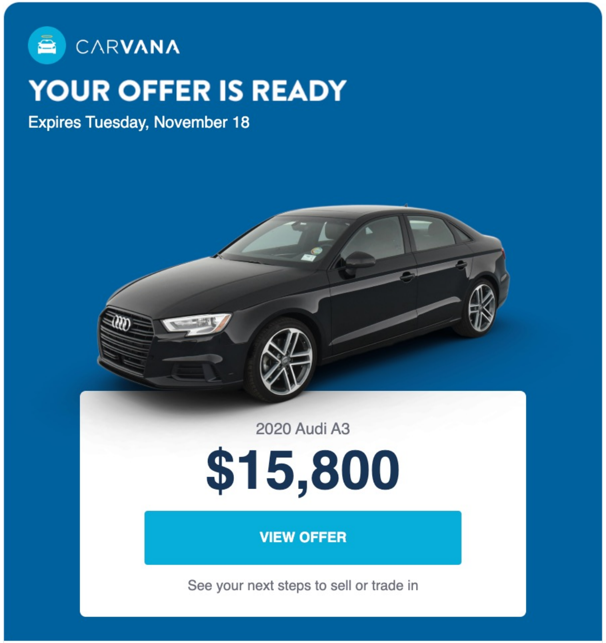
MGMT298D
Science and Strategy of AI

Week 2

Tree-Based Models: CART, Random Forests, XGBoost

Auyon Siddiq
UCLA Anderson School of Management

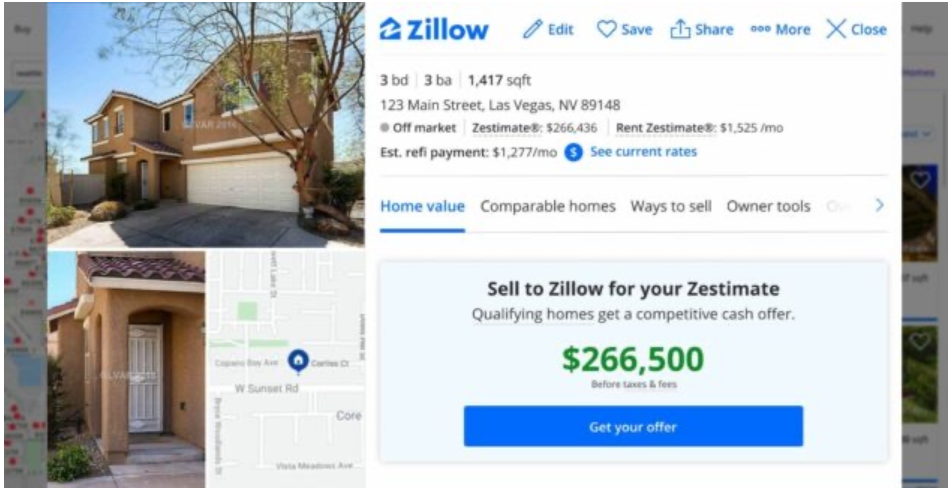

ALWAYS KEEP PACE
WITH YOUR CAR'S VALUE.
Instantly check your vehicle's current value and track it over time.
CARVANA
Value Tracker
Your Car's Value
$25,990


CARVANA
YOUR OFFER IS READY
Expires Tuesday, November 18
2020 Audi A3
$15,800
VIEW OFFER
See your next steps to sell or trade in

Zillow will now make cash offers for homes based on its ‘Zestimates’

By Clare Duffy, CNN Business

3 min read · Updated 10:50 AM EST, Thu February 25, 2021



Zillow's home-buying debacle shows how hard it is to use AI to value real estate

By Rachel Metz, CNN Business

7 min read · Published 7:32 AM EST, Tue November 9, 2021

The decision, announced last week, marks a stunning defeat for Zillow. The real estate listing company took a $304 million inventory write-down in the third quarter, which it blamed on having recently purchased homes for prices that are higher than it thinks it can sell them. The company saw its stock plunge and it now plans to cut 2,000 jobs, or 25% of its staff.

Algorithmic Pricing: Carvana and Zillow

Case Handout

Week 1 Recap

Machine learning is a branch of AI focused on prediction from tabular data

ML and AI models are trained by minimizing a **loss function** – a measure of error between true labels and model predictions

Training optimization process gives us the model **weights**

Linear regression is an example of an ML model. Feature engineering allows for richer models (recall H&M sales) and regularization prevents overfitting to data

Regularization hyperparameters (LASSO, Ridge, ElasticNet) selected via **cross-validation**

Today's Class

We'll cover an alternative framework to linear regression for standard machine learning predictive tasks

Industry-strength "gold standard" method: XGBoost

Classification and Regression Trees (CART)

Classification and Regression Trees (CART)

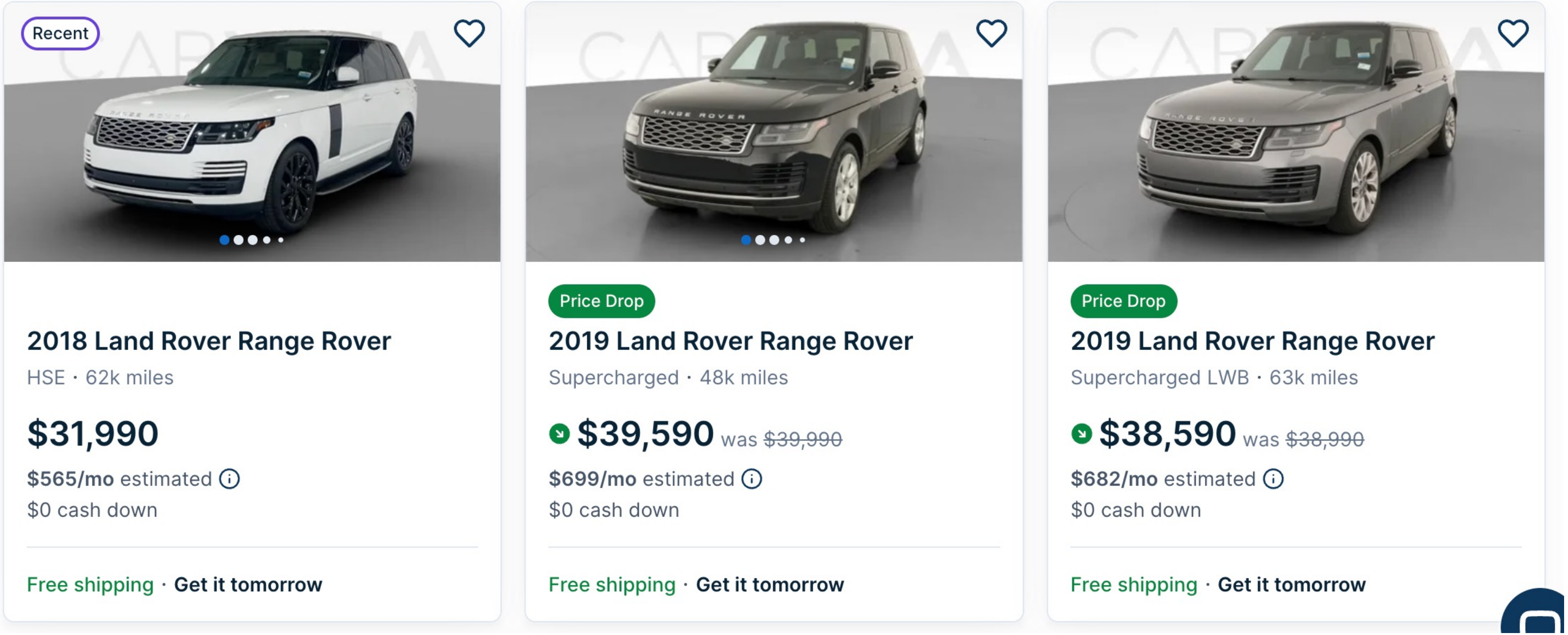
Tree-based models: Alternative to linear regression for predictions

In addition to regression (predicting a number), can be used for classification (predicting a category)

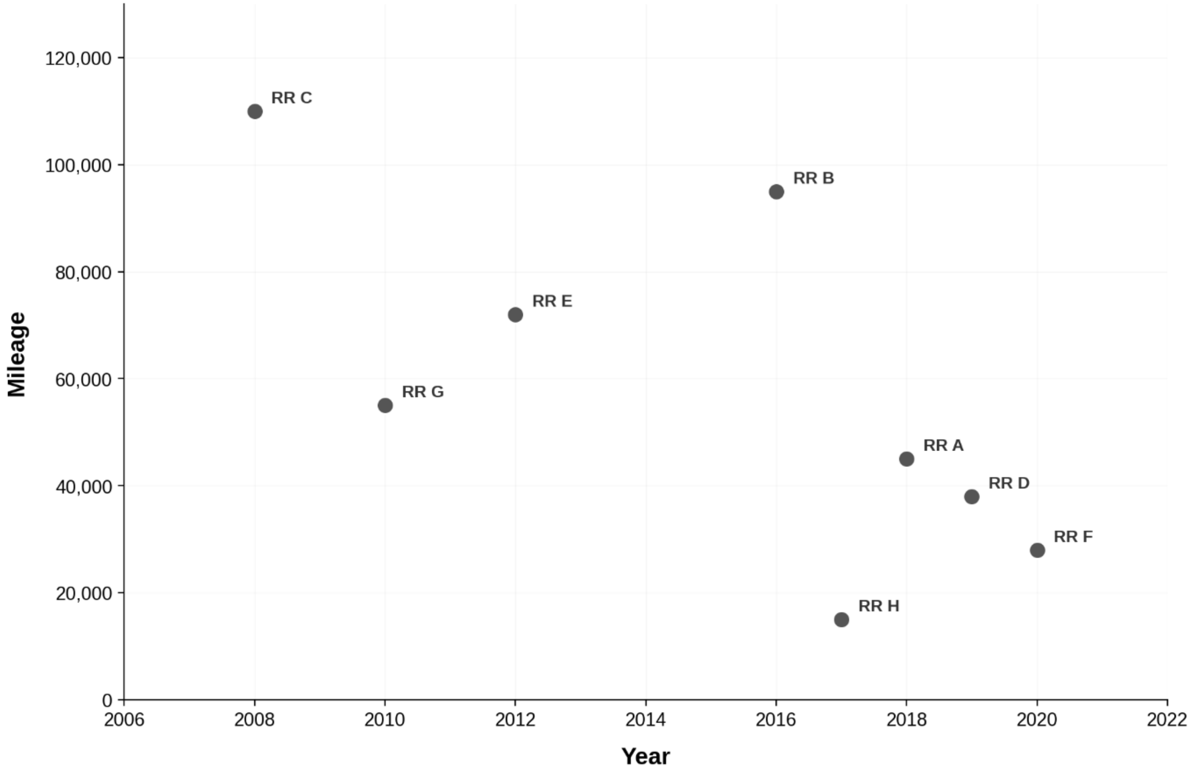
Together called “CART”, sometimes also “decision tree”

Regression Tree Example

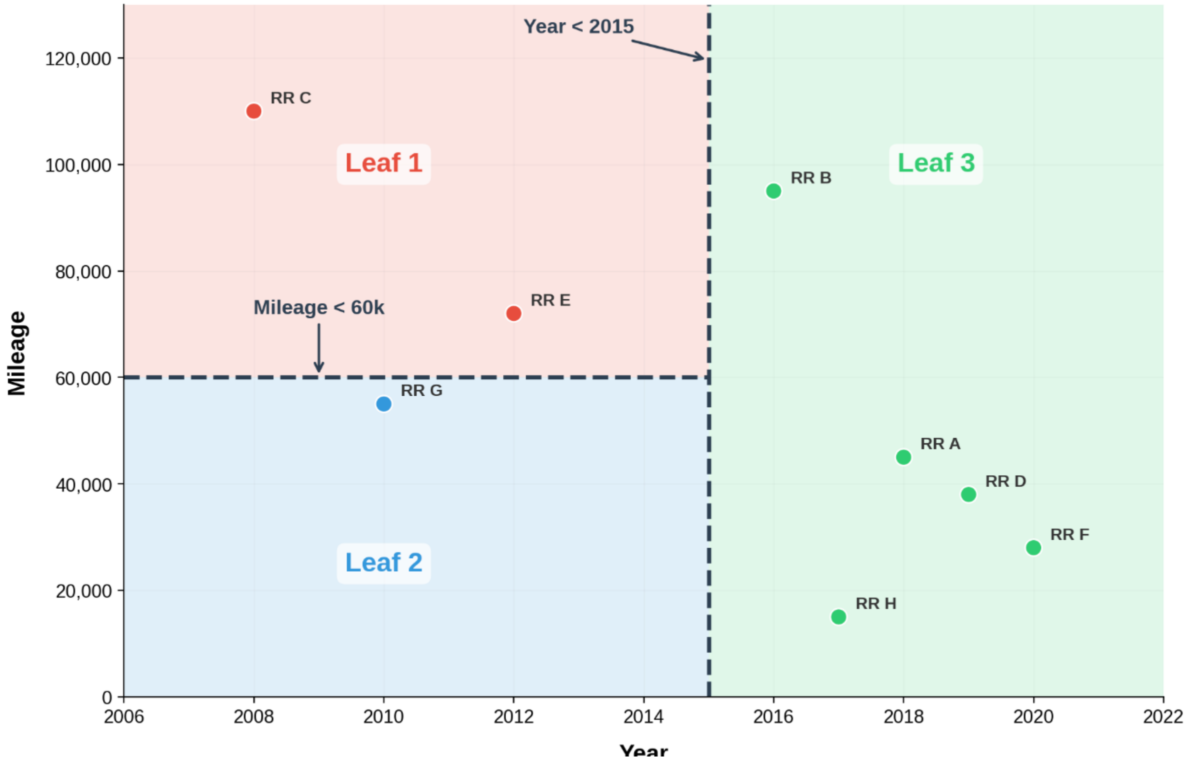
Sample of 375 used Range Rovers



Vehicle	Mileage	Year	Price
RR A	45,000	2018	$48,000
RR B	95,000	2016	$32,000
RR C	110,000	2008	$14,000
RR D	38,000	2019	$52,000
RR E	72,000	2012	$18,000
RR F	28,000	2020	$58,000
RR G	55,000	2010	$30,000
RR H	15,000	2017	$55,000

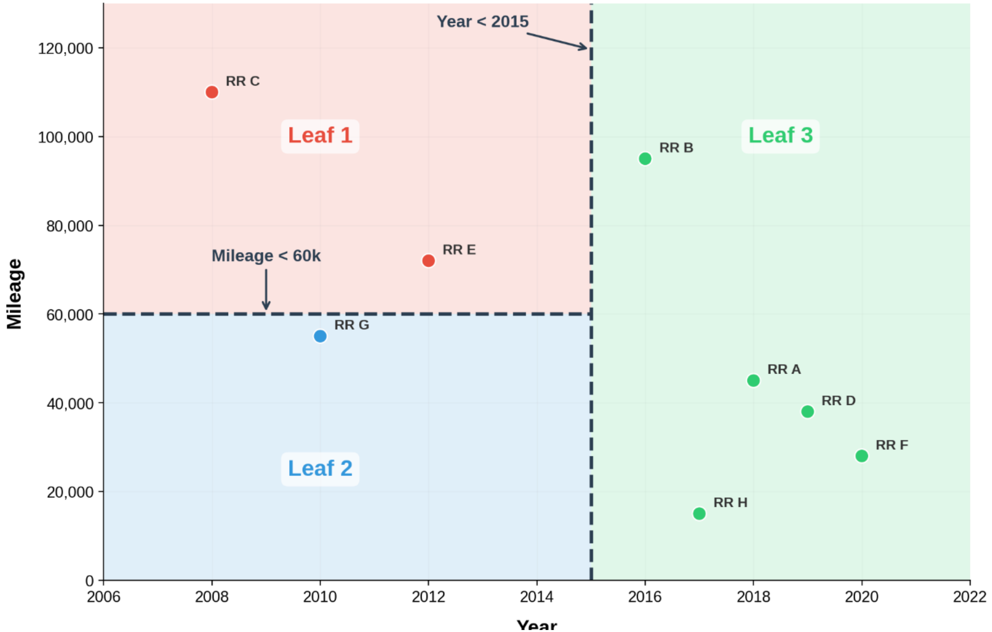


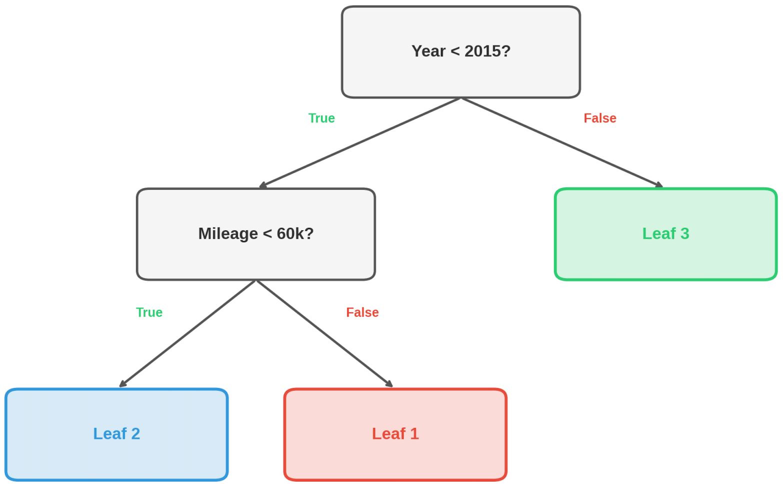
Vehicle	Mileage	Year	Price
RR A	45,000	2018	$48,000
RR B	95,000	2016	$32,000
RR C	110,000	2008	$14,000
RR D	38,000	2019	$52,000
RR E	72,000	2012	$18,000
RR F	28,000	2020	$58,000
RR G	55,000	2010	$30,000
RR H	15,000	2017	$55,000



Each leaf = a prediction

average price of the vehicles in that leaf





Vehicle	Mileage	Year	Actual Price	Leaf	Predicted	Error
RR A	45,000	2018	$48,000	Leaf 3	$49,000	-$1,000
RR B	95,000	2016	$32,000	Leaf 3	$49,000	-$17,000
RR C	110,000	2008	$14,000	Leaf 1	$16,000	-$2,000
RR D	38,000	2019	$52,000	Leaf 3	$49,000	+$3,000
RR E	72,000	2012	$18,000	Leaf 1	$16,000	+$2,000
RR F	28,000	2020	$58,000	Leaf 3	$49,000	+$9,000
RR G	55,000	2010	$30,000	Leaf 2	$30,000	$0
RR H	15,000	2017	$55,000	Leaf 3	$49,000	+$6,000

Training Regression Trees

Training linear regression = finding model weights

Training regression trees = finding split locations (“weight-free”)

Main idea: At each split point, choose a feature and threshold value that minimizes combined error from each new leaf

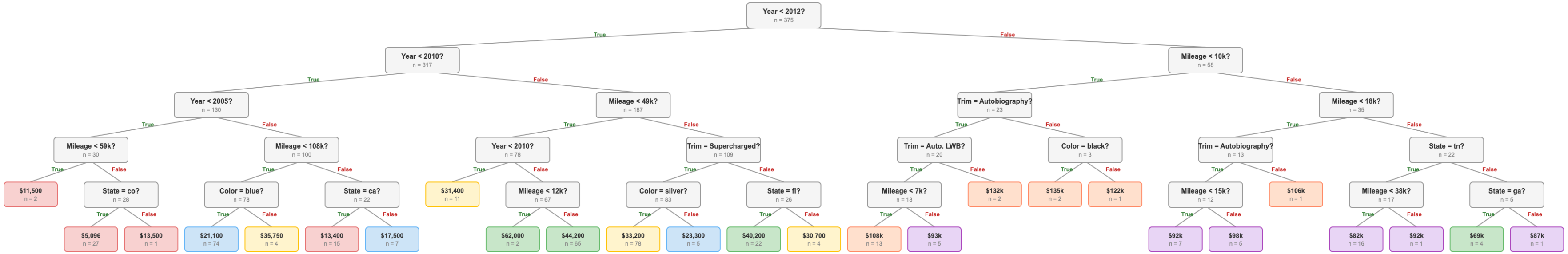
Regression Tree Splits

Depth & Overfitting in Trees

Model complexity determined by tree depth

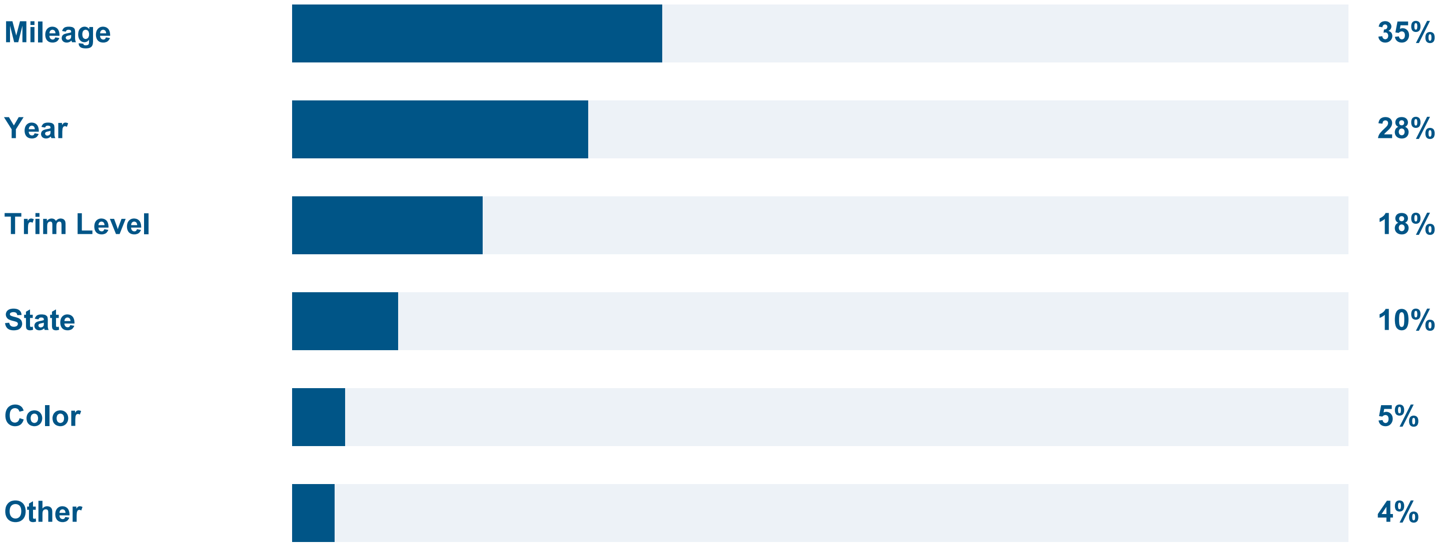
Deep trees have stronger explanatory power, but also risk overfitting

Depth of 4-5 is standard in many applications



Feature Importance

Tree-based models tell you which features drive predictions the most, e.g.,



Trees vs Linear Regression

Linear regression:

Based on model weights, assumes linear relationship

Requires feature engineering to capture non-linear relationships

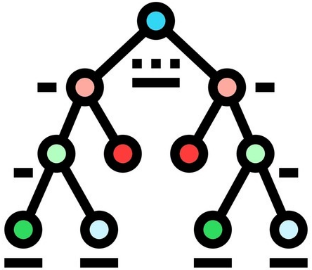
Generally does not overfit unless very small dataset and/or many features

Regression trees:

No model weights, based on splitting the feature data “directly”

Super flexible, no feature engineering required to capture non-linearity or variable interactions

Tree depth is key for model’s power – deeper trees catch richer patterns in data but more prone to overfitting



Can we do better?

Ensembles in Machine Learning

One of the most powerful ideas in ML: **ensembles** of models

Key idea: Train *multiple* models to produce predictions that outperform individual models

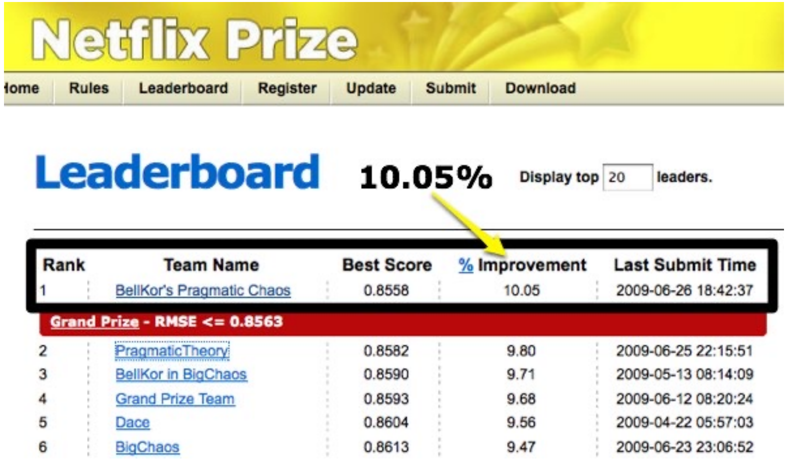
Intuition: Individual errors in models tend to cancel out, stabilizes predictions

“Wisdom of the crowds”

The Netflix Prize

Competition run by Netflix in late 2000s to beat in-house algorithm for recommendations by at least 10%

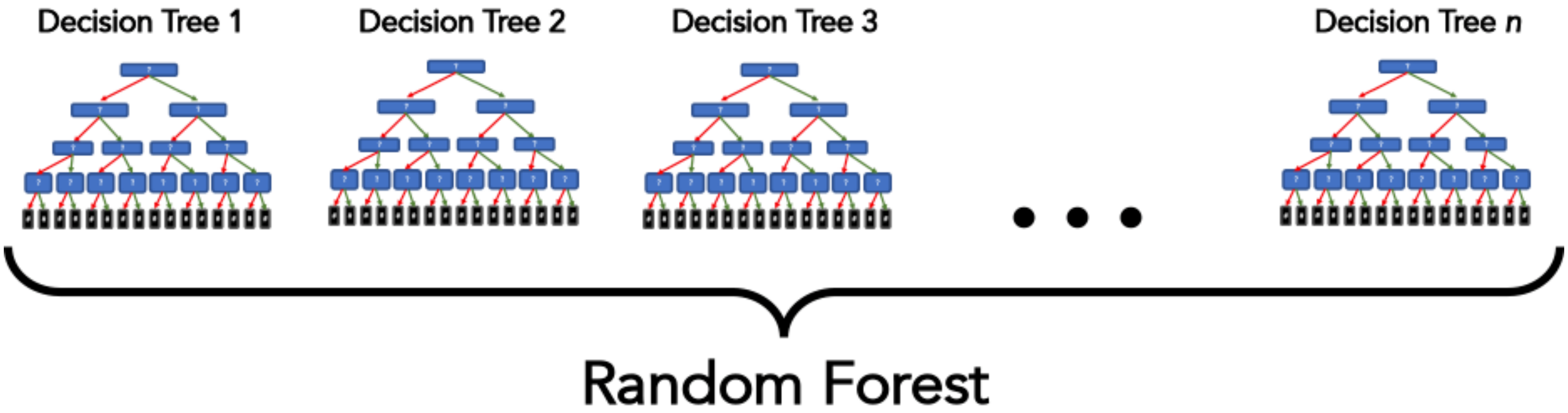
After ~3 years, winning team was an ensemble of multiple top models



Rank	Team Name	Best Score	% Improvement	Last Submit Time
1	BellKor's Pragmatic Chaos	0.8558	10.05	2009-06-26 18:42:37
Grand Prize - RMSE <= 0.8563				
2	PragmaticTheory	0.8582	9.80	2009-06-25 22:15:51
3	BellKor in BigChaos	0.8590	9.71	2009-05-13 08:14:09
4	Grand Prize Team	0.8593	9.68	2009-06-12 08:20:24
5	Dace	0.8604	9.56	2009-04-22 05:57:03
6	BigChaos	0.8613	9.47	2009-06-23 23:06:52

Random Forests

Random Forest: Parallel Tree Building



Training and Prediction

Training:

1. At each split, consider only a random subset of features (default is 1/3)
2. Train a single (deep) regression tree on this sample
3. Repeat steps 1-3 for as many trees as you like (20, 100, 500)

Prediction:

Each tree makes one prediction, final prediction is simply the average:

$$\text{Prediction} = (\text{Tree}_1 + \text{Tree}_2 + \ldots + \text{Tree}_{500}) / 500$$

Random Forest: Numerical Example

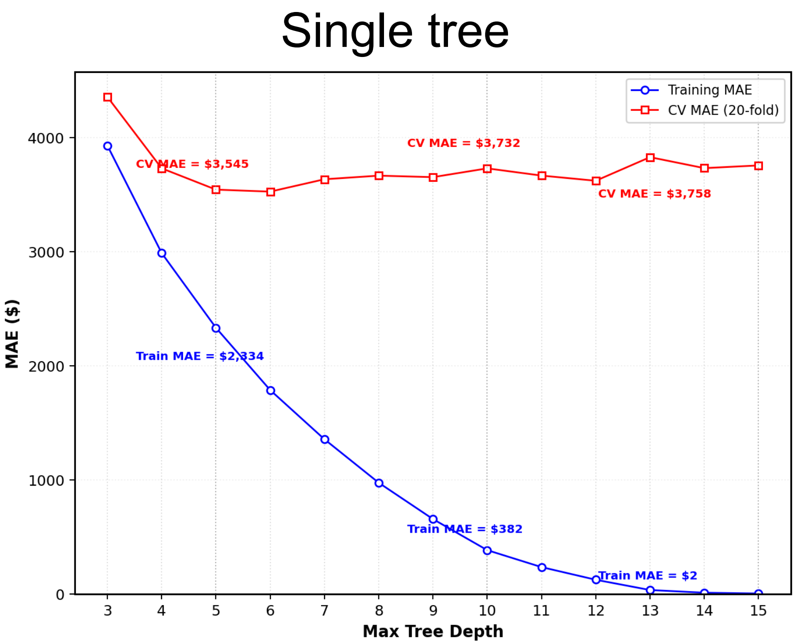
Each tree gives a slightly different prediction; the average is more stable:

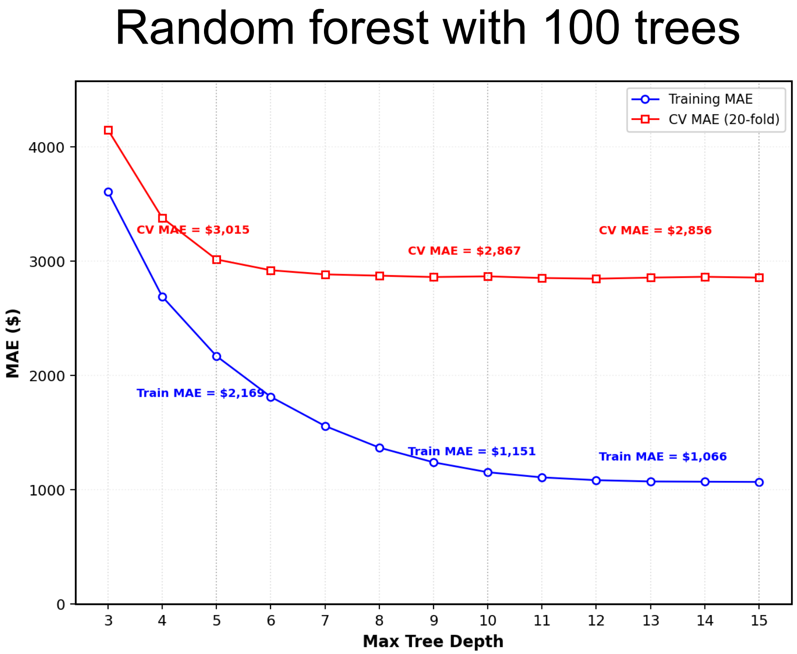
Vehicle	Mileage	Year	Actual Price	Individual Tree Predictions (500 Trees)					Avg. Prediction	Error
				Tree 1	Tree 2	Tree 3	· · ·	Tree 500		
RR A	45,000	2018	$48,000	$51,000	$46,000	$49,000	· · ·	$47,000	**$47,800**	-$200
RR B	95,000	2016	$32,000	$35,000	$30,000	$34,000	· · ·	$31,000	**$33,200**	+$1,200
RR C	110,000	2008	$14,000	$13,000	$18,000	$15,000	· · ·	$14,000	**$15,400**	+$1,400
RR D	38,000	2019	$52,000	$50,000	$53,000	$49,000	· · ·	$51,000	**$50,600**	-$1,400
RR E	72,000	2012	$18,000	$16,000	$20,000	$19,000	· · ·	$17,000	**$17,800**	-$200
RR F	28,000	2020	$58,000	$55,000	$60,000	$56,000	· · ·	$54,000	**$56,200**	-$1,800
RR G	55,000	2010	$30,000	$28,000	$32,000	$31,000	· · ·	$29,000	**$30,400**	+$400
RR H	15,000	2017	$55,000	$53,000	$57,000	$52,000	· · ·	$54,000	**$53,800**	-$1,200

MAE: $875

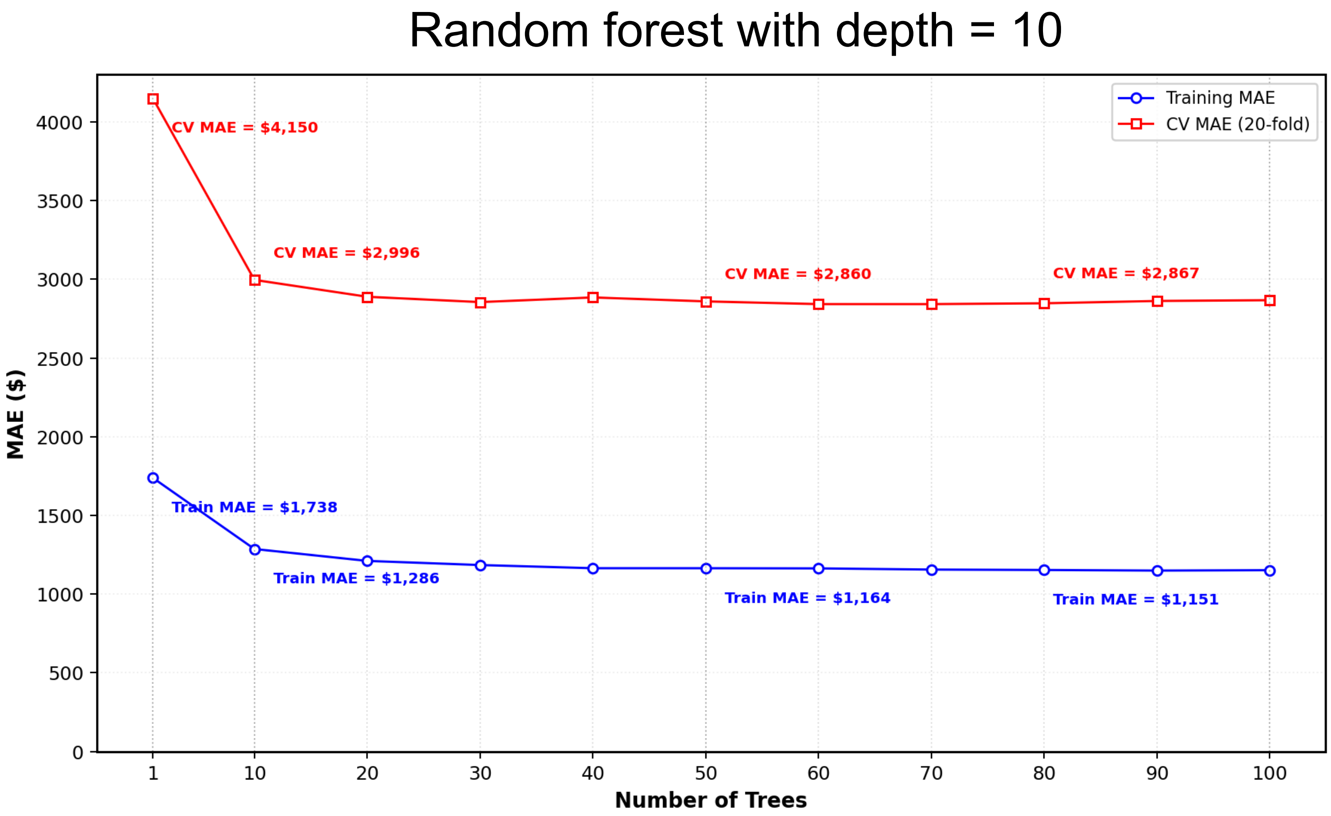
Averaging reduces noise — more stable than any single tree

Random Forest: Numerical Example





Random Forest: Numerical Example



Random Forest Hyperparameters

Number of trees: Typical: 50–500. Usually diminishing returns. Easier prediction problems need fewer trees

Maximum tree depth: Typical: 10–15. Deeper OK since averaging predictions controls overfitting

Maximum features: Features to consider per split. Typical: n/3.

Summary: Trees vs. Forests vs. XGBoost

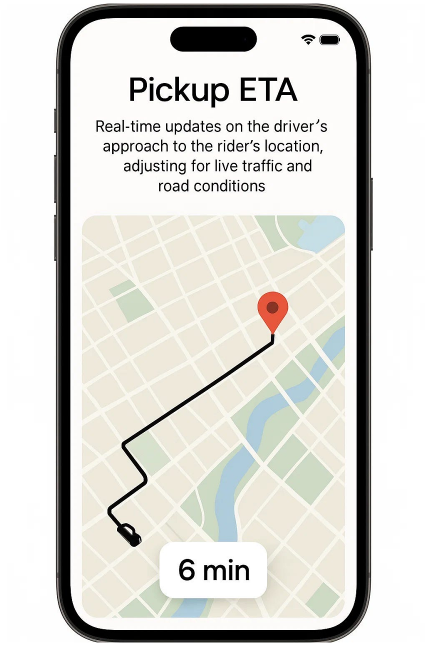
Single Tree	Random Forest
Interpretability **High**	Interpretability **Low**
Predictive Accuracy **Moderate**	Accuracy **High**
Overfitting Risk **High**	Overfitting Risk **Low**
Tuning Required **Moderate**	Tuning Required **Low**

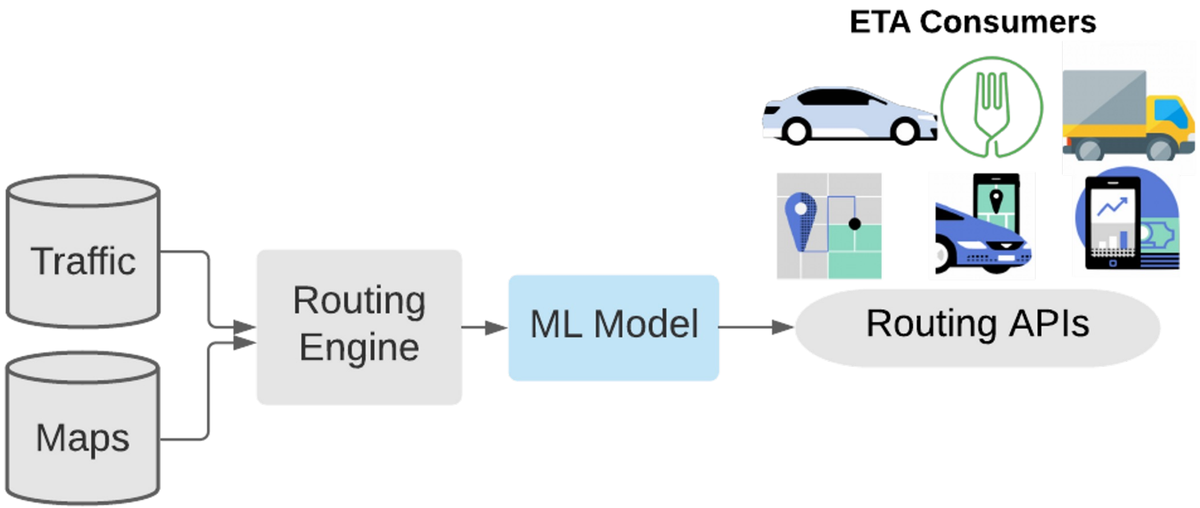
15-min Break

Tree Models in Practice

Uber: ETA Predictions

Prediction task: Predict estimated time of arrival (ETA) for Uber trips
Features: traffic conditions, route + map details
Label: travel time

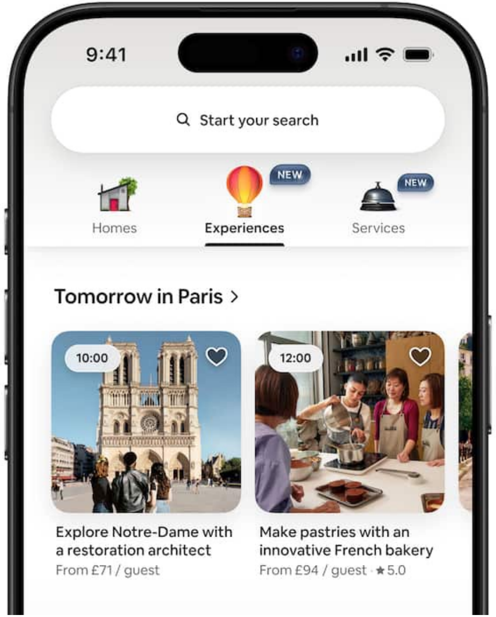


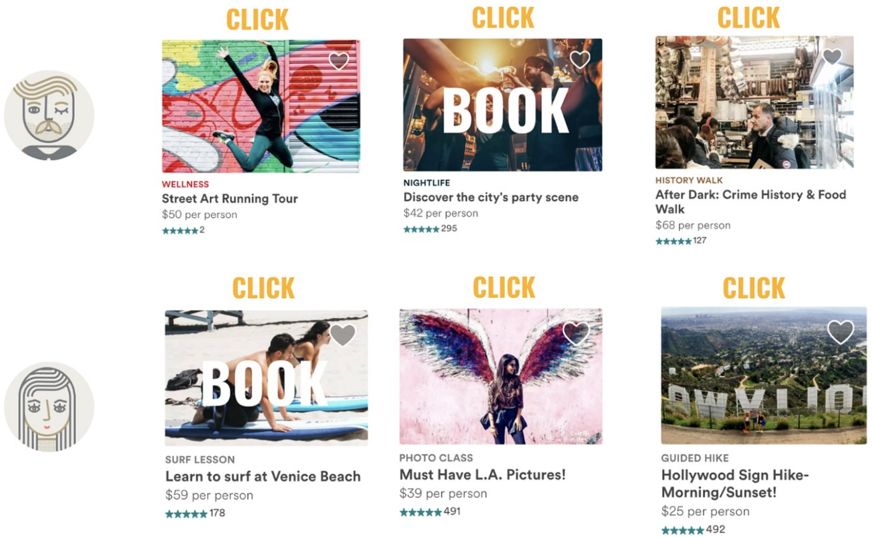


https://www.uber.com/us/en/blog/deepeta-how-uber-predicts-arrival-times/

Airbnb: Search Ranking for Experiences

Prediction task: Predict which experiences users will click or book
Features: experience duration, price, category, reviews, occupancy, number of bookings, user click history
Label: if user clicks, if user books

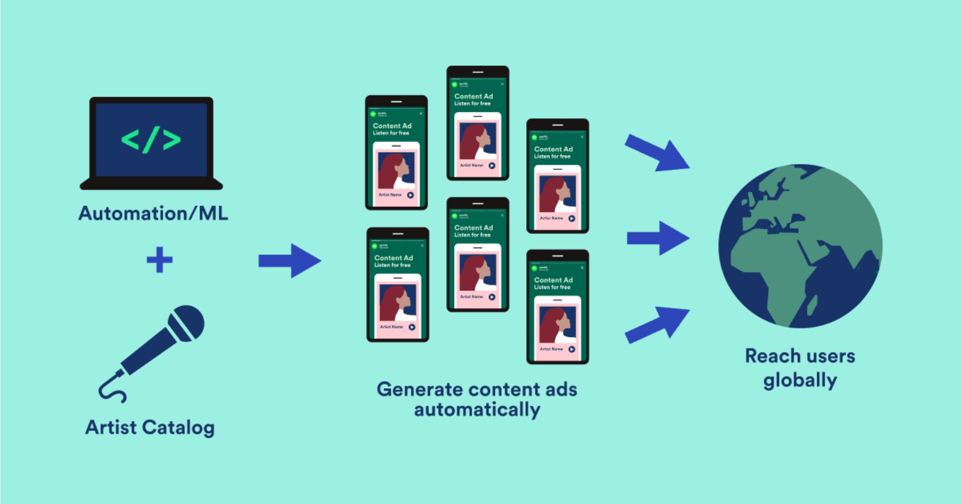




https://medium.com/airbnb-engineering/machine-learning-powered-search-ranking-of-airbnb-experiences-110b4b1a0789

Spotify: Content Marketing

Prediction task: Predict success rate of online ad campaigns
Features: artist shown, ad creative design details, market
Label: conversion rate (e.g., new user sign-ups)



https://engineering.atspotify.com/2023/11/how-we-automated-content-marketing-to-acquire-users-at-scaleAr

DMLC
XGBoost

Parallel vs Sequential Trees

Two common approaches to ensemble methods: parallel vs sequential models

Parallel: train independent ML models and average / combine predictions
→ Random Forest

Sequential: Train models one at a time, with each model building on the last
→ **XGBoost**

XGBoost: The Algorithm

Instead of predicting the target directly, each new tree **predicts the error from all previous trees**

$$\text{Prediction} = \text{Base} + \eta \cdot \text{Tree}_1 + \eta \cdot \text{Tree}_2 + \eta \cdot \text{Tree}_3 + \ldots$$

The **learning rate** η determines how much each new tree's prediction contributes to overall prediction

Lower $\eta \rightarrow$ more trees needed, but more accurate / less overfitting risk

Overall training error shrinks with each tree

XGBoost: Numerical Example

XGBoost: Each tree fits the residual left by all previous trees

Each tree uses a random subsample of data

				Sequential Trees ($\eta = 0.1$)									
Vehicle	Actual Price	Base	$Resid_0$	Tree 1	$\eta \times T1$	$Resid_1$	Tree 2	$\eta \times T2$	$Resid_2$	Tree 3	$\eta \times T3$	$Resid_3$	**Final Pred.**
RR A	$48,000	$36,500	$11,500	$7,500	$750	$10,750	$7,000	$700	$10,050	$6,500	$650	$9,400	**$38,600**
RR B	$32,000	$36,500	-$4,500	-$2,500	-$250	-$4,250	-$2,500	-$250	-$4,000	-$2,000	-$200	-$3,800	**$35,800**
RR C	$14,000	$36,500	-$22,500	-$15,500	-$1,550	-$20,950	-$14,500	-$1,450	-$19,500	-$13,500	-$1,350	-$18,150	**$32,150**
RR D	$52,000	$36,500	$15,500	$9,500	$950	$14,550	$8,500	$850	$13,700	$8,000	$800	$12,900	**$39,100**

XGBoost Hyperparameters

XGBoost is very sensitive to tuning parameters and more prone to overfitting than random forests

Three main parameters:	*Typical range*
1. **learning rate**	(0.01 to 0.3)
2. number of trees	(50 to 1000)
3. max tree depth	(3-6)

An optional parameter is data **subsample**: Each tree fits to a random sample (e.g. 70%) of the training data. Helps a bit with regularization

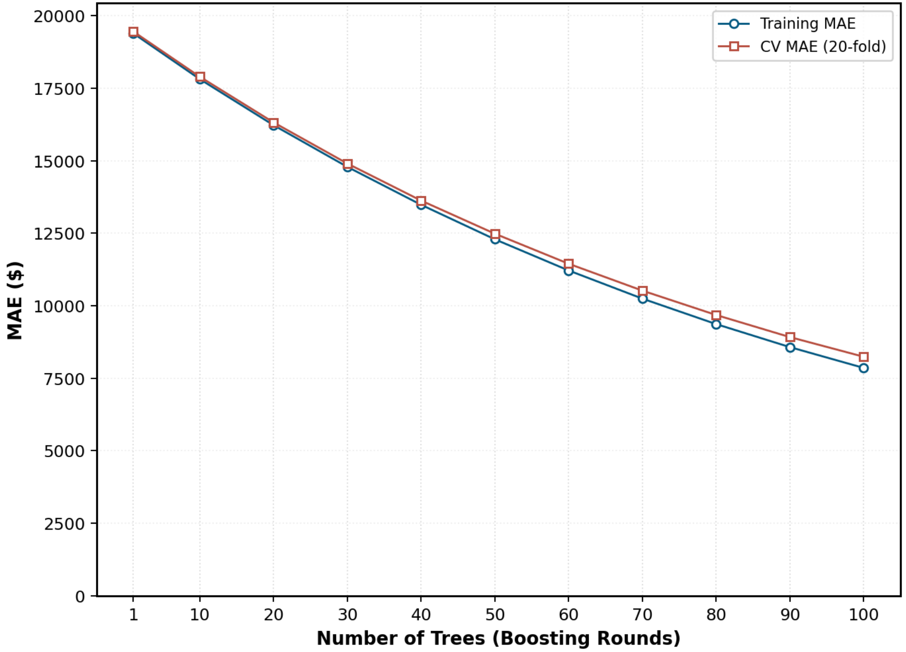
Summary: Trees vs. Forests vs. XGBoost

	Single Tree	Random Forest	XGBoost
Interpretability	High	Low	Low
Accuracy	Moderate	High	Very High
Overfitting Risk	High	Low	Moderate
Tuning Required	Moderate	Low	High

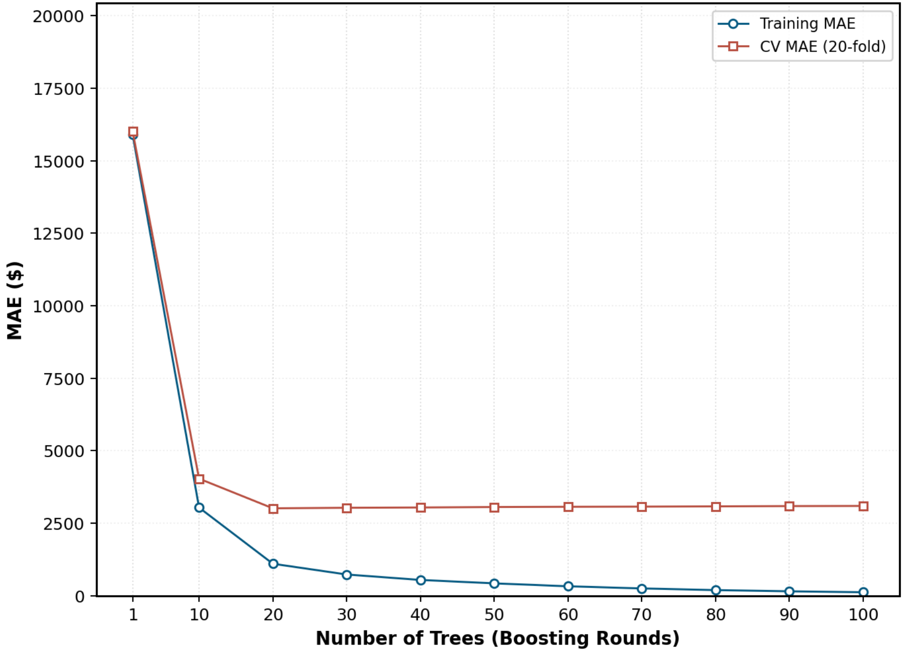
Best Practices:
Hyperparameter Tuning

learning_rate (η)

η = 0.01



η = 0.2



Best Practices: Hyperparameter Tuning

1. **Start with defaults.** If it works good enough “out of the box”, use as is

2. **Tune hyperparameters.** Try a handful of different combinations; **number of trees** and **learning rate** matter more than **tree depth**

3. **Use proper validation.** Use cross-validation to choose hyperparameters, keep test set for “final” evaluation

4. **Monitor for overfitting.** Keep an eye on train vs. validation error. Excessive divergence = overfitting

Which model should you use?

	Single Tree	Random Forest	XGBoost
Interpretability	High	Low	Low
Accuracy	Moderate	High	Very High
Overfitting Risk	High	Low	Moderate
Tuning Required	Moderate	Low	High

colab

Assignment 2

Glossary (1/2)

Term	Definition
Decision Tree (CART)	A model that makes predictions by splitting data with yes/no questions, forming a tree of decision rules.
Leaf Node	The smallest unit of a decision tree that contains a prediction value, such as the average price of all data points that landed there.
Tree Depth	The number of levels of splits in a tree; deeper trees are more complex and more prone to overfitting.
Hyperparameter	A setting chosen before training that controls model behavior, such as tree depth or learning rate. Not learned from data.
Feature Importance	A score showing how much each input variable contributed to the model's predictions.
Random Forest	An ensemble of many decision trees, each trained on a random sample of data, whose predictions are averaged for stability.

Glossary (2/2)

Ensemble	A method that combines multiple models to produce predictions that outperform any individual model, leveraging the “wisdom of the crowds.”
XGBoost	A sequential ensemble algorithm that builds trees one at a time, where each new tree focuses on correcting the errors of previous trees.
Residuals	The difference between actual values and current predictions; in XGBoost, each new tree is trained to predict these gaps.
Learning Rate (eta)	A shrinkage factor that controls how much each new tree contributes to the overall prediction; lower values reduce overfitting but require more trees.