NYU

Decision Trees: Interpretable Classification (and Regression)

Week 2

January 29, 2024

Spring 24 | CUSP-GX 7033 – Machine Learning for Cities | Dr. Anton Rozhkov

Today's Outline

- The prediction problem (classification and regression)
- Overview of classification methods; discussion of interpretable vs. black-box predictors; feature importance
- Interpretable prediction: Rule-based, instance-based, and model-based
- Rule-based Learning: Decision trees for classification and regression

Overview of Classification Methods

NYU

The Prediction Problem

Most machine learning tasks are about making predictions.

This means that given a number of training examples, the system needs to be able to make good predictions for (generalize to) examples it has never seen before.

Having a good performance measure on the training data is good but insufficient; the true goal is to perform well on new instances.

NYU

Source: Geron 2022

Classification vs Regression

Classification predictive modeling is the task of approximating a mapping function (f) from input variables (X) to discrete output variables (y).

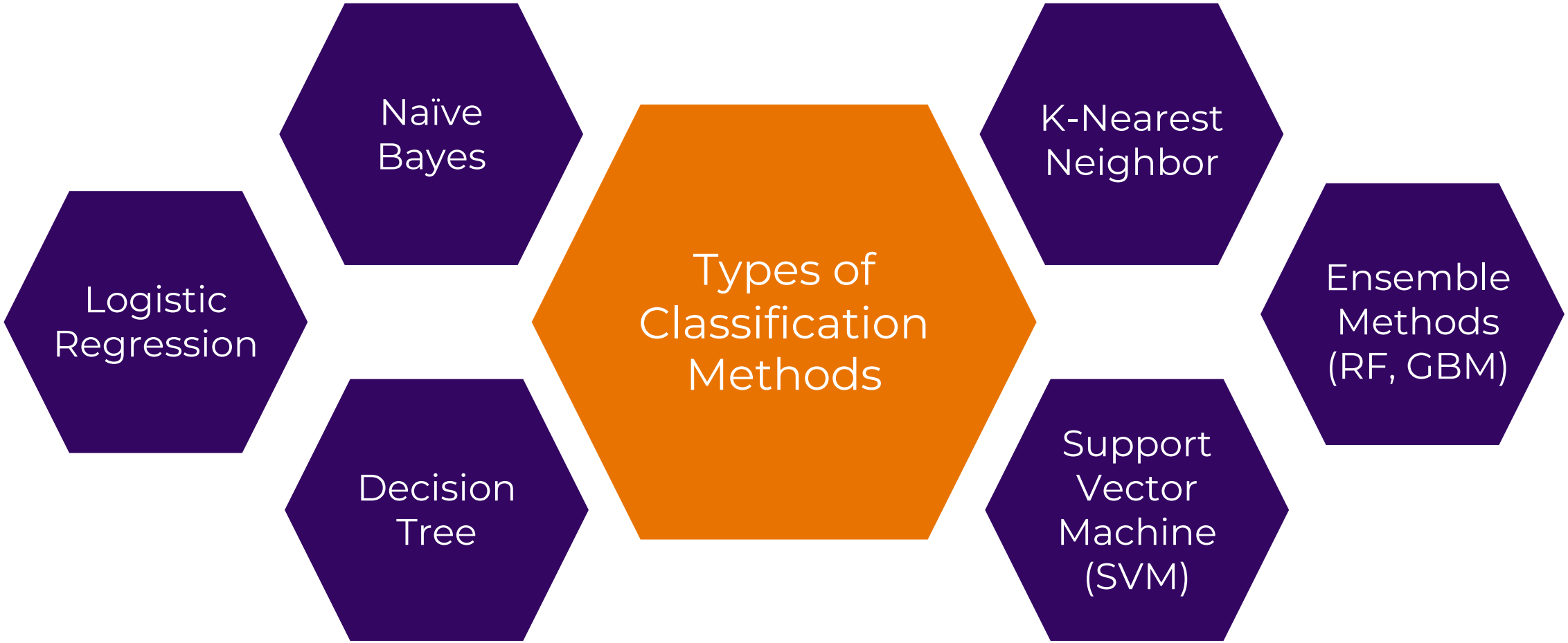
Regression predictive modeling is the task of approximating a mapping function (f) from input variables (X) to a continuous output variable (y).

Classification vs Regression

Classification predictive modeling is the task of approximating a mapping function (f) from input variables (X) to discrete output variables (y).

Regression predictive modeling is the task of approximating a mapping function (f) from input variables (X) to a continuous output variable (y).

Classification Methods



Evaluating Classification Methods

- **Accuracy**
  - classifier accuracy: predicting the class label
  - regression accuracy: guessing the value of predicted attributes
- **Speed**
  - time to construct the model (training time)
  - time to use the model (classification/prediction time)
- **Robustness**: handling noise and missing values
- **Scalability**: efficiency on large databases
- **Interpretability**
  - understanding and insight provided by the model
- **Other measures**, e.g., the goodness of rules, such as decision tree size or compactness of classification rules

Source: Jianlin Cheng; Data Mining: Concepts and Techniques

Interpretable Prediction

Idea of a Black Box

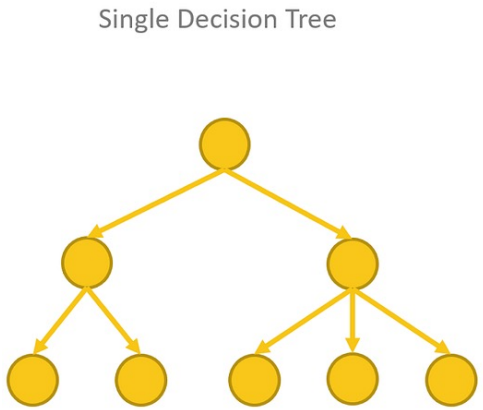
NYU

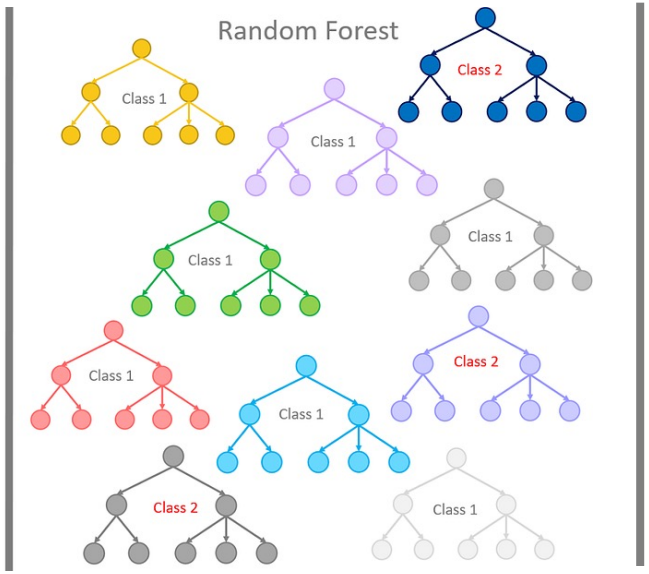
Introductions

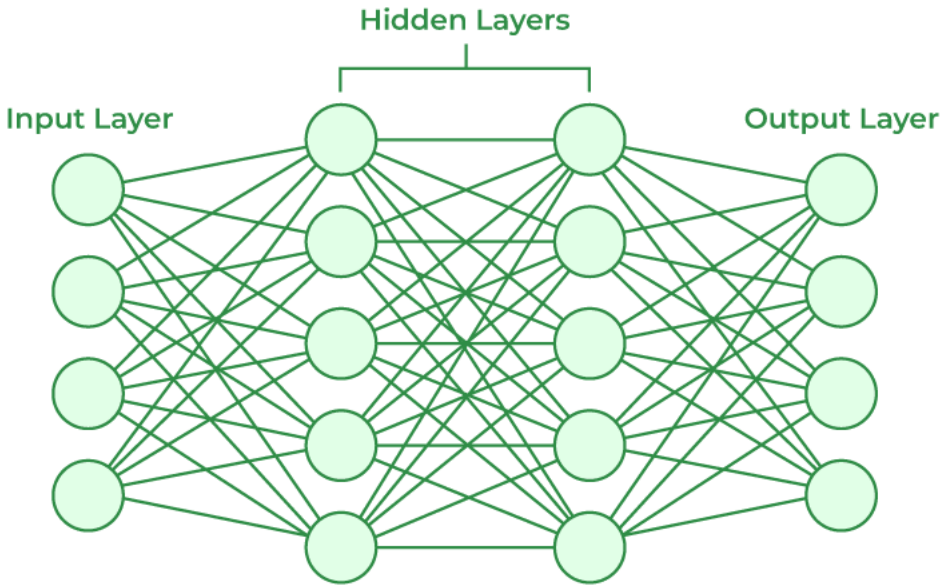
Logistic regression
Decision trees
...

... Random forest ...

Artificial Neural Networks







Interpretability (transparent) → **"Black box" (opaque)**

NYU

Image source: https://media.geeksforgeeks.org/wp-content/cdn-uploads/20230602113310/Neural-Networks-Architecture.png

Idea of a Black Box

NYU

Statistical Science
2010, Vol. 25, No. 3, 289–310
DOI: 10.1214/10-STS330

To Explain or to Predict?

Galit Shmueli

Abstract. Statistical modeling is a powerful tool for developing and testing theories by way of causal explanation, prediction, and description. In many disciplines there is near-exclusive use of statistical modeling for causal explanation and the assumption that models with high explanatory power are inherently of high predictive power. Conflation between explanation and prediction is common, yet the distinction must be understood for progressing scientific knowledge. While this distinction has been recognized in the philosophy of science, the statistical literature lacks a thorough discussion of the many differences that arise in the process of modeling for an explanatory versus a predictive goal. The purpose of this article is to clarify the distinction between explanatory and predictive modeling, to discuss its sources, and to reveal the practical implications of the distinction to each step in the modeling process.

Key words and phrases: Explanatory modeling, causality, predictive modeling, predictive power, statistical strategy, data mining, scientific research.

1. INTRODUCTION

Looking at how statistical models are used in different scientific disciplines for the purpose of theory building and testing, one finds a range of perceptions regarding the relationship between causal explanation and empirical prediction. In many scientific fields such as economics, psychology, education, and environmental science, statistical models are used almost exclusively for causal explanation, and models that possess high explanatory power are often assumed to inherently possess predictive power. In fields such as natural language processing and bioinformatics, the focus is on empirical prediction with only a slight and indirect relation to causal explanation. And yet in other research fields, such as epidemiology, the emphasis on causal explanation versus empirical prediction is more mixed. Statistical modeling for description, where the purpose is to capture the data structure parsimoniously, and which is the most commonly developed within the field of statistics, is not commonly used for theory building and testing in other disciplines. Hence, in this article I focus on the use of statistical modeling for causal explanation and for prediction. My main premise is that the two are often conflated, yet the causal versus predictive distinction has a large impact on each step of the statistical modeling process and on its consequences. Although not explicitly stated in the statistics methodology literature, applied statisticians instinctively sense that predicting and explaining are different. This article aims to fill a critical void: to tackle the distinction between explanatory modeling and predictive modeling.

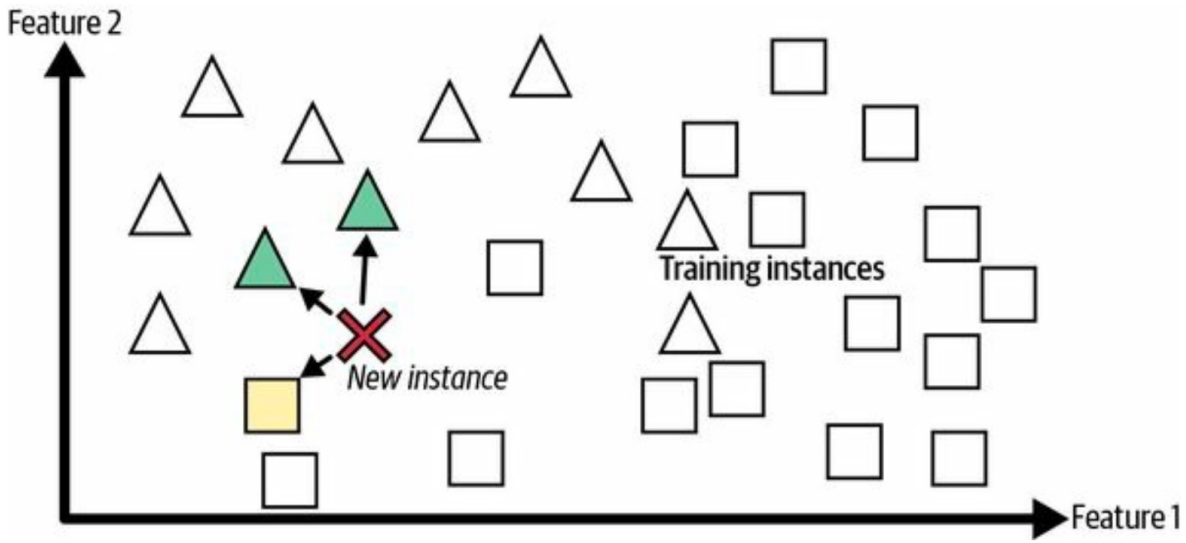
Clearing the current ambiguity between the two is critical not only for proper statistical modeling, but more importantly, for proper scientific usage. Both explanation and prediction are necessary for generating and testing theories, yet each plays a different role in doing so. The lack of a clear distinction within statistics has created a lack of understanding in many disciplines of the difference between building sound explanatory models versus creating powerful predictive models, as well as confusing explanatory power with predictive power. The implications of this omission and the lack of clear guidelines on how to model for explanatory versus predictive goals are considerable for both scientific research and practice and have also contributed to the gap between academia and practice.

I start by defining what I term *explaining* and *predicting*. These definitions are chosen to reflect the dis-

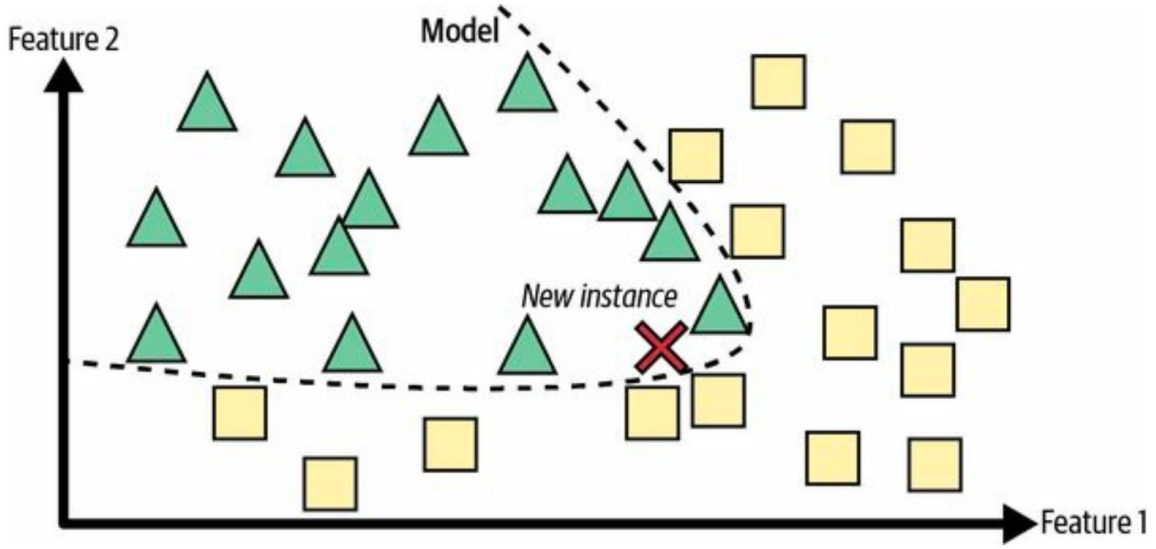
Galit Shmueli is Associate Professor of Statistics, Department of Decision, Operations and Information Technologies, Robert H. Smith School of Business, University of Maryland, College Park, Maryland 20742, USA (e-mail: gshmueli@umd.edu).

Source: Shmueli, Galit. "To explain or to predict?." (2010): 289-310.

Instance-based and Model-based Learning



Instance-based



Model-based

Source: Geron 2022

Instance-based and Model-based Learning

	Instance-based	Model-based
Generalization	Simply memorize the training examples	Learn a generalizable model
Scalability	Slow and memory-intensive with large datasets	More scalable without storing the training data
Interpretability	Effective for small or medium-sized datasets	Good for all datasets

Rule-based Learning

Rule-based systems are computer programs utilizing *if-then* rules for decision-making and task execution.

Strengths:

- Structured Decision Making: offers a clear and structured approach to problem-solving.
- Transparency: the logic behind decisions is explicit, providing transparency.

Challenges:

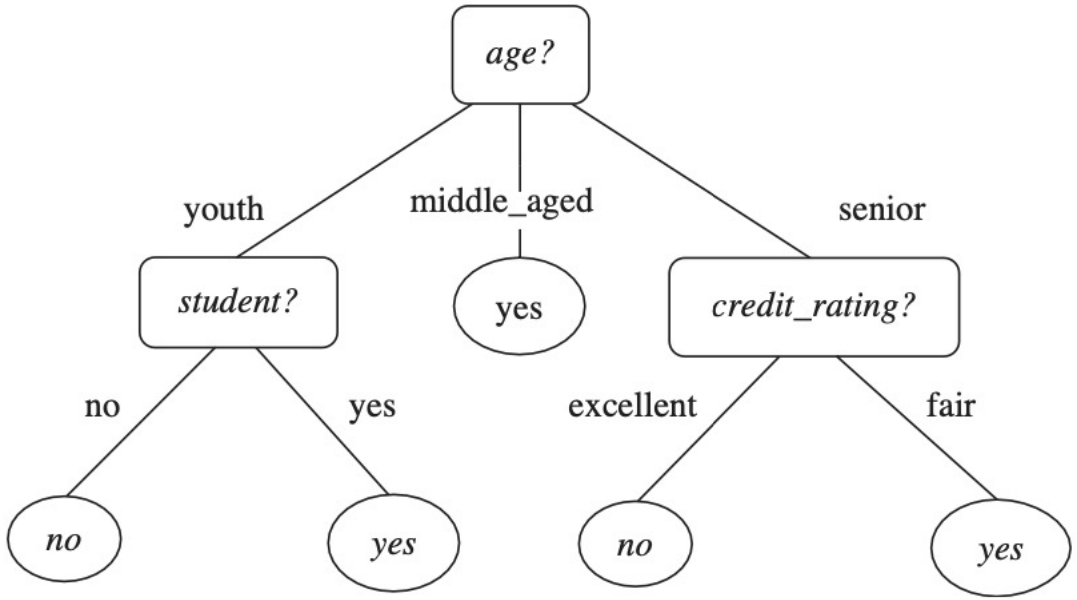
- Optimal Solutions: rule-based systems may not always find the most optimal solution.
- Dependency on Rule Quality: effectiveness is tied to the accuracy and comprehensiveness of rules.

Decision Trees

Decision Trees

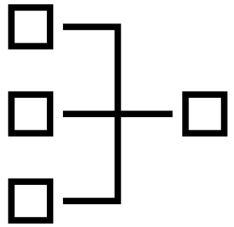
Decision Trees (DTs) are a non-parametric supervised learning method used for classification and regression.

- The simplest method to find rules.
- Provide “if condition, then consequence” rules.
- Simple to understand and to interpret. Trees can be visualized.
- Requires little data preparation.



NYU

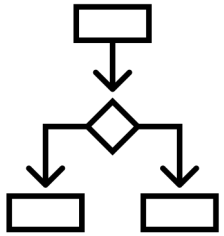
History



Concept learning systems

by E. B. Hunt, J. Marin, and P. T. Stone

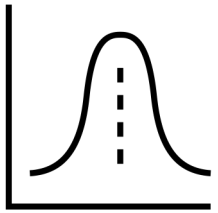
Late 1970s and early 1980



Development of a decision tree algorithm known as *ID3 (Iterative Dichotomiser)* + C4.5

By J. Ross Quinlan

1984



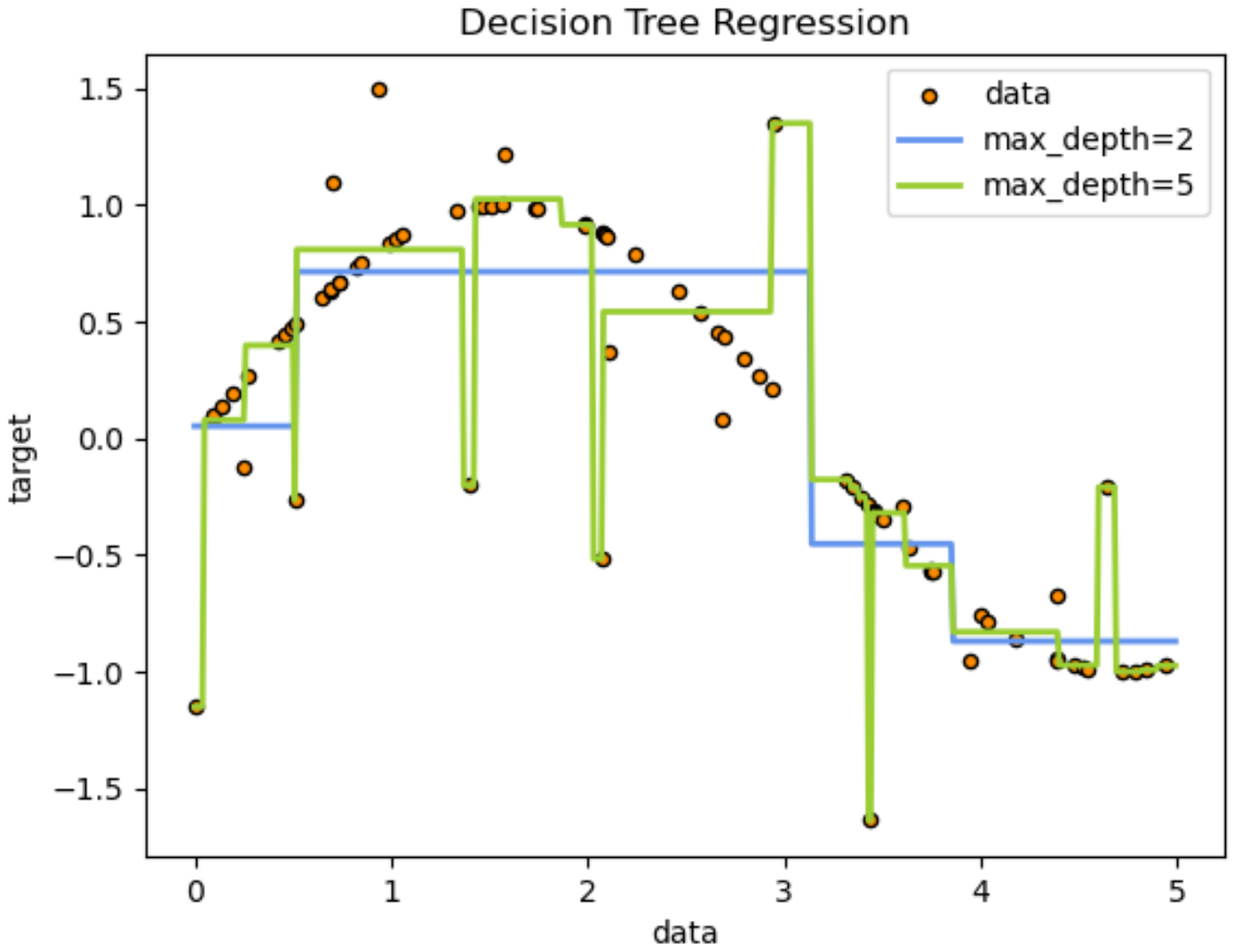
Book: Classification and Regression Trees (CART), which described the generation of binary decision trees

by L. Breiman, J. Friedman, R. Olshen, and C. Stone

NYU

Source: Han 2011, ch. 8

Example


Decision Tree Regression
data
max_depth=2
max_depth=5
target
data
1.5
1.0
0.5
0.0
−0.5
−1.0
−1.5
0
1
2
3
4
5

NYU

When to Use Decision Trees? (1/2)

We have a dataset, with some attribute we want to predict

- We can do either classification or regression.
- Datasets with lots of attributes and/or lots of records are okay. But if there are lots of attributes and not that many records, you should probably use feature selection first to avoid overfitting.
- Datasets with discrete or real values, or both, are okay.

We want to be able to explain our prediction using a simple and interpretable set of rules

- We want to distinguish relevant from irrelevant attributes.
- We want to provide a set of decision steps (e.g., an "expert system" for medical diagnosis; we don't want to perform irrelevant tests).
- If all we care about is prediction accuracy and not interpretability, we might want to use some "black box" classifier instead (e.g., neural network, support vector machine).

When to Use Decision Trees? (2/2)

Can represent complex, non-linear functions

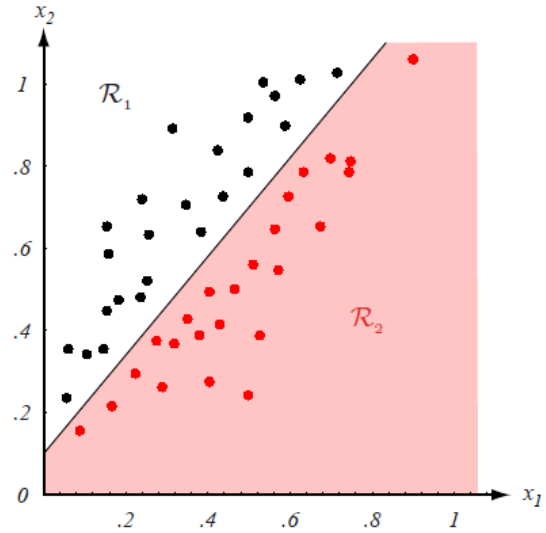
Performance is better when a tree structure makes sense (see next slide).

NYU

Inductive Bias

Inductive bias describes how a prediction method generalizes to previously unseen examples.

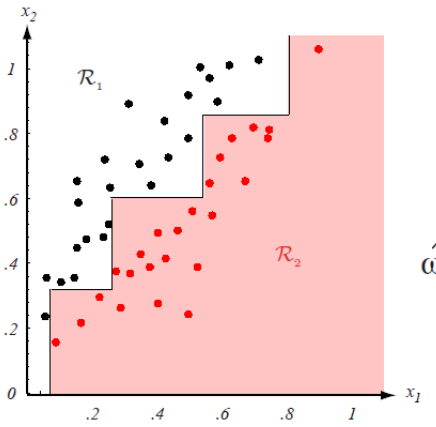
For example: binary logistic regression assumes a linear decision boundary between $R_1(y = 1) > 0.5$ and $R_2(y = 1) < 0.5$.

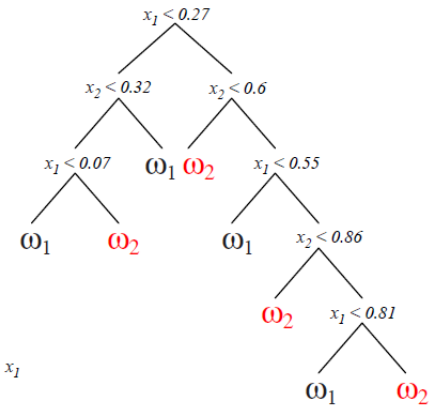


What are the inductive biases of a decision tree?

- Flexible – any function can be represented.
- Prefers shorter trees to longer trees.
- Prefers attributes with higher information gain.
- Splits on one attribute at a time.
- Splits are axis-aligned.

Suboptimal for linear functions



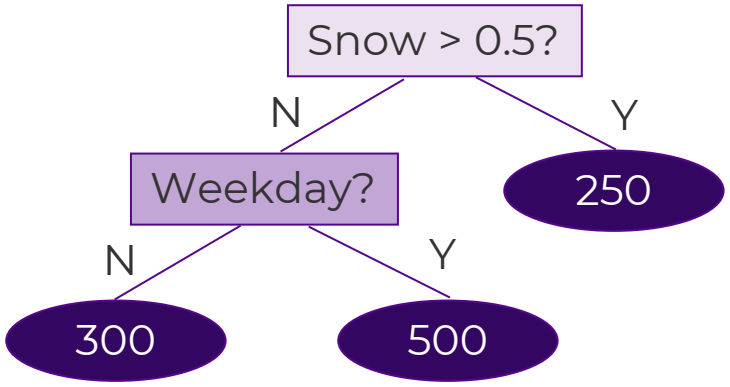


NYU

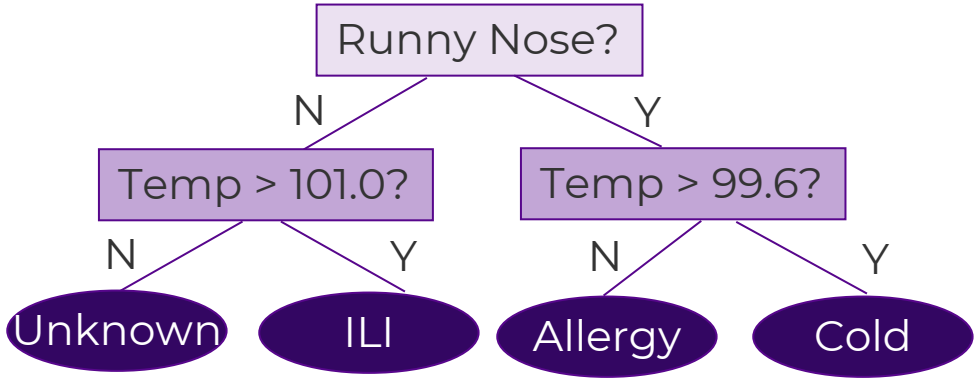
Source: Duda, Hart, and Stork 2nd ed., fig. 8.5.

Rule-based Learning with Decision Trees

- A decision tree is a set of rules that can be learned from data and used to predict an unknown value.
  - Unknown real value (regression): What is the expected incidence of car thefts in NYC on a given day?
  - Unknown category value (classification): What type of illness does patient X have, given their symptoms and demographic data?



How many thefts on Tuesday, January 3 (0.2 inches of snow)?



What do we predict for a patient with Temp = 100 and a runny nose?

Learning Binary Decision Trees

- Step 1: Start with all data points in a single node. Predict the most common value (classification) or mean value (regression).

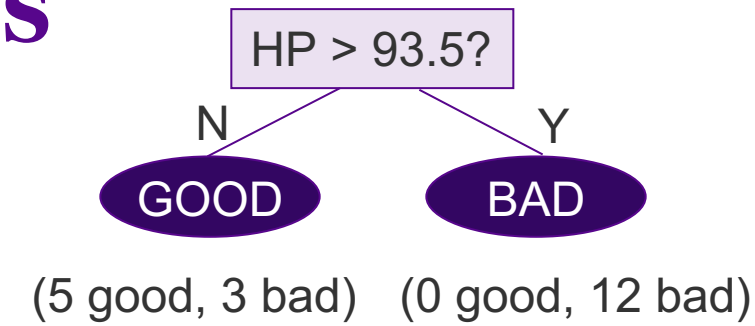
MPG, cylinders, HP, weight

good, 4, 75, light
bad, 6, 90, medium
bad, 4, 110, medium
bad, 8, 175, weighty
bad, 6, 95, medium
bad, 4, 94, light
bad, 4, 95, light
bad, 8, 139, weighty
bad, 8, 190, weighty
bad, 8, 145, weighty
bad, 6, 100, medium
good, 4, 92, medium
bad, 6, 100, weighty
bad, 8, 170, weighty
good, 4, 89, medium
good, 4, 65, light
bad, 6, 85, medium
bad, 4, 81, light
bad, 6, 95, medium
good, 4, 93, light

NYU

Learning Binary Decision Trees

- Step 1: Start with all data points in a single node. Predict the most common value (classification) or mean value (regression).
- Step 2: Choose the "best" binary decision rule and use it to split the data into two groups.



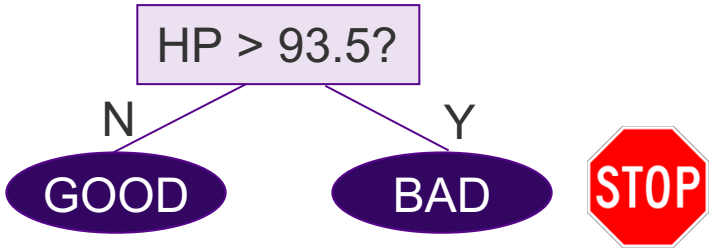
MPG, cylinders, HP, weight

good, 4, 75, light
bad, 6, 90, medium
bad, 4, 110, medium
bad, 8, 175, weighty
bad, 6, 95, medium
bad, 4, 94, light
bad, 4, 95, light
bad, 8, 139, weighty
bad, 8, 190, weighty
bad, 8, 145, weighty
bad, 6, 100, medium
good, 4, 92, medium
bad, 6, 100, weighty
bad, 8, 170, weighty
good, 4, 89, medium
good, 4, 65, light
bad, 6, 85, medium
bad, 4, 81, light
bad, 6, 95, medium
good, 4, 93, light

NYU

Learning Binary Decision Trees

- Step 1: Start with all data points in a single node. Predict the most common value (classification) or mean value (regression).
- Step 2: Choose the "best" binary decision rule and use it to split the data into two groups.
- Step 3: Repeat step 2 on each group until some stopping criterion is reached.



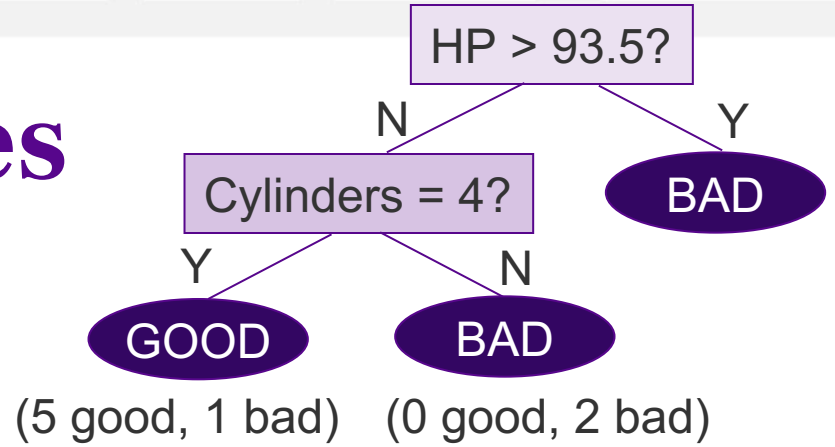
(5 good, 3 bad) (0 good, 12 bad)

MPG, cylinders, HP, weight

good, 4, 75, light
bad, 6, 90, medium
bad, 4, 110, medium
bad, 8, 175, weighty
bad, 6, 95, medium
bad, 4, 94, light
bad, 4, 95, light
bad, 8, 139, weighty
bad, 8, 190, weighty
bad, 8, 145, weighty
bad, 6, 100, medium
good, 4, 92, medium
bad, 6, 100, weighty
bad, 8, 170, weighty
good, 4, 89, medium
good, 4, 65, light
bad, 6, 85, medium
bad, 4, 81, light
bad, 6, 95, medium
good, 4, 93, light

NYU

Learning Binary Decision Trees



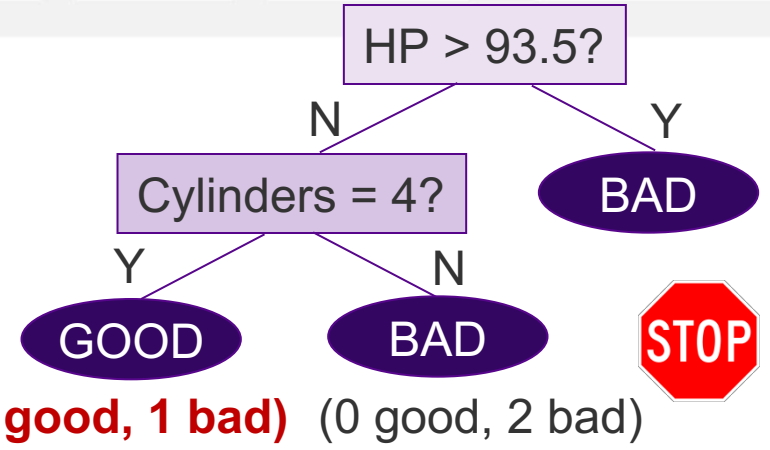
- Step 1: Start with all data points in a single node. Predict the most common value (classification) or mean value (regression).
- Step 2: Choose the "best" binary decision rule and use it to split the data into two groups.
- Step 3: Repeat step 2 on each group until some stopping criterion is reached.

MPG, cylinders, HP, weight

good, 4, 75, light
bad, 6, 90, medium
bad, 4, 110, medium
bad, 8, 175, weighty
bad, 6, 95, medium
bad, 4, 94, light
bad, 4, 95, light
bad, 8, 139, weighty
bad, 8, 190, weighty
bad, 8, 145, weighty
bad, 6, 100, medium
good, 4, 92, medium
bad, 6, 100, weighty
bad, 8, 170, weighty
good, 4, 89, medium
good, 4, 65, light
bad, 6, 85, medium
bad, 4, 81, light
bad, 6, 95, medium
good, 4, 93, light

NYU

Learning Binary Decision Trees



- Step 1: Start with all data points in a single node. Predict the most common value (classification) or mean value (regression).
- Step 2: Choose the "best" binary decision rule and use it to split the data into two groups.
- Step 3: Repeat step 2 on each group until some stopping criterion is reached.

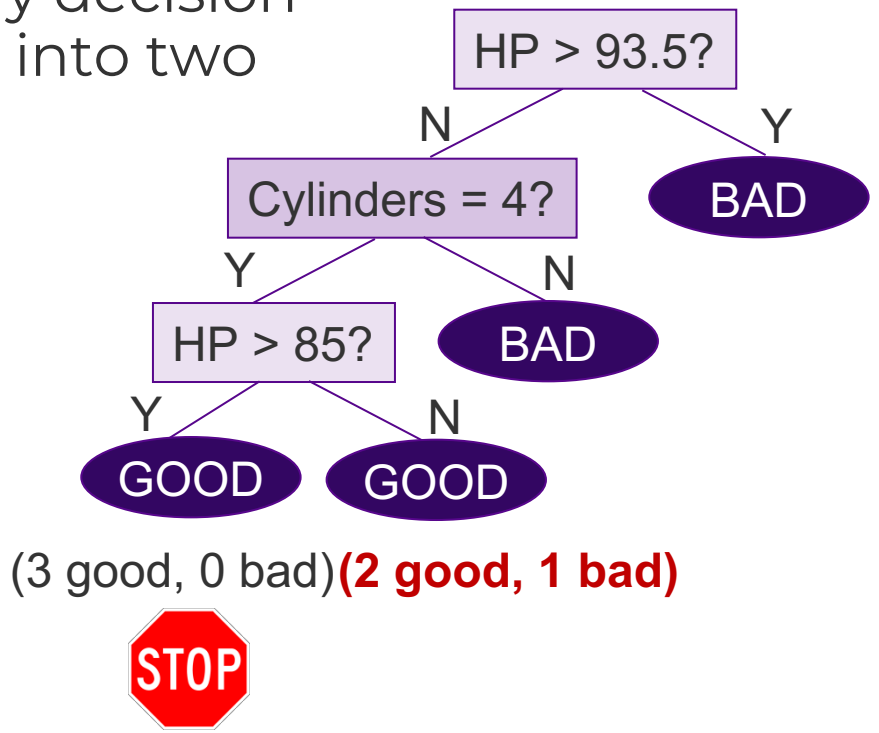
MPG, cylinders, HP, weight

good, 4, 75, light
bad, 6, 90, medium
bad, 4, 110, medium
bad, 8, 175, weighty
bad, 6, 95, medium
bad, 4, 94, light
bad, 4, 95, light
bad, 8, 139, weighty
bad, 8, 190, weighty
bad, 8, 145, weighty
bad, 6, 100, medium
good, 4, 92, medium
bad, 6, 100, weighty
bad, 8, 170, weighty
good, 4, 89, medium
good, 4, 65, light
bad, 6, 85, medium
bad, 4, 81, light
bad, 6, 95, medium
good, 4, 93, light

NYU

Learning Binary Decision Trees

- Step 1: Start with all data points in a single node. Predict the most common value (classification) or mean value (regression).
- Step 2: Choose the "best" binary decision rule and use it to split the data into two groups.
- Step 3: Repeat step 2 on each group until some stopping criterion is reached.



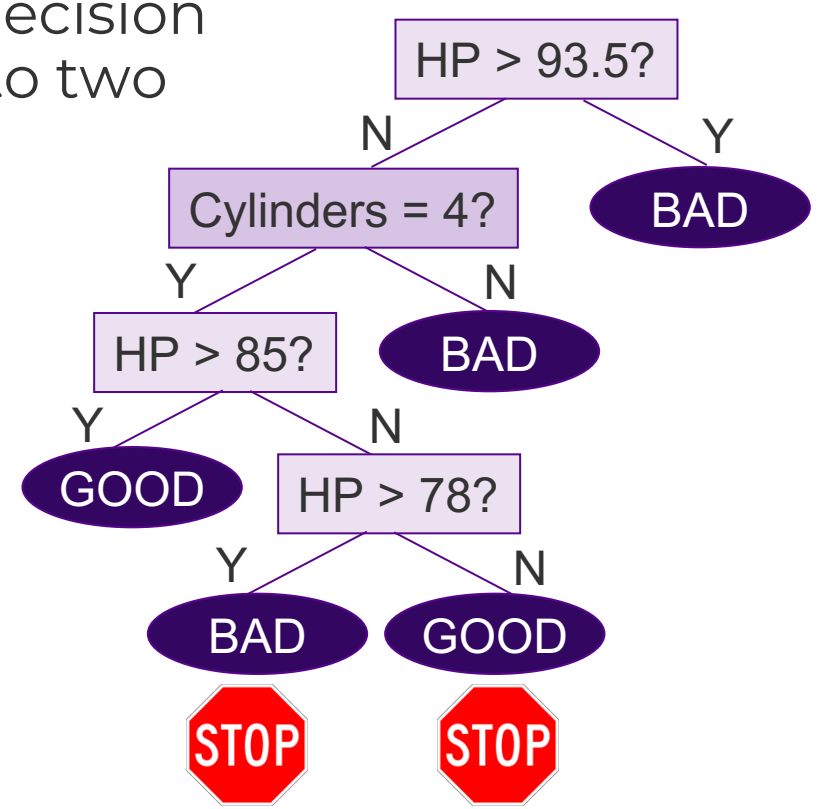
MPG, cylinders, HP, weight

good, 4, 75, light
bad, 6, 90, medium
bad, 4, 110, medium
bad, 8, 175, weighty
bad, 6, 95, medium
bad, 4, 94, light
bad, 4, 95, light
bad, 8, 139, weighty
bad, 8, 190, weighty
bad, 8, 145, weighty
bad, 6, 100, medium
good, 4, 92, medium
bad, 6, 100, weighty
bad, 8, 170, weighty
good, 4, 89, medium
good, 4, 65, light
bad, 6, 85, medium
bad, 4, 81, light
bad, 6, 95, medium
good, 4, 93, light

NYU

Learning Binary Decision Trees

- Step 1: Start with all data points in a single node. Predict the most common value (classification) or mean value (regression).
- Step 2: Choose the "best" binary decision rule and use it to split the data into two groups.
- Step 3: Repeat step 2 on each group until some stopping criterion is reached.



MPG, cylinders, HP, weight

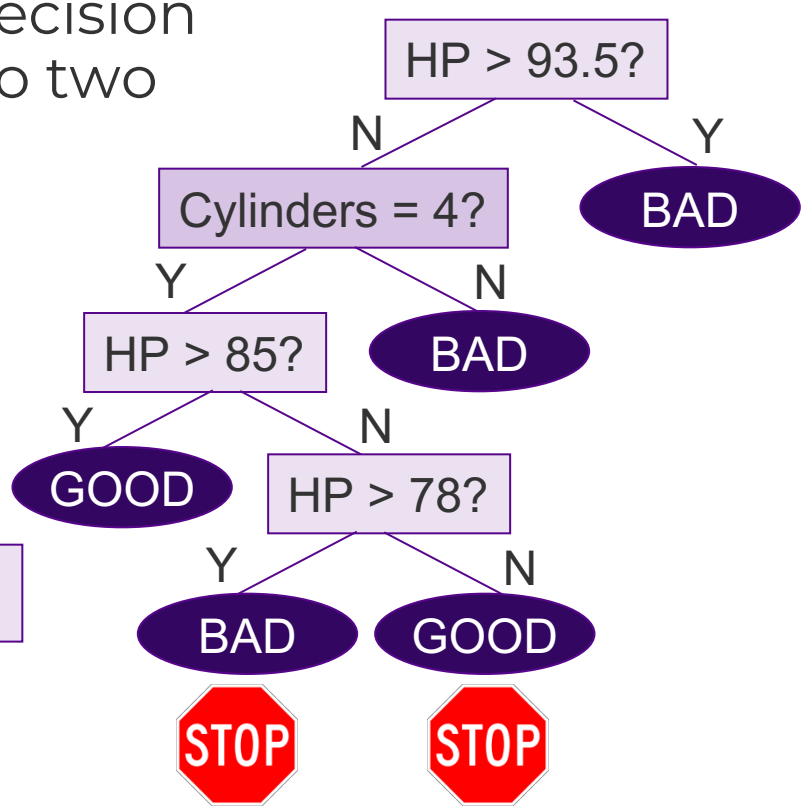
good, 4, 75, light
bad, 6, 90, medium
bad, 4, 110, medium
bad, 8, 175, weighty
bad, 6, 95, medium
bad, 4, 94, light
bad, 4, 95, light
bad, 8, 139, weighty
bad, 8, 190, weighty
bad, 8, 145, weighty
bad, 6, 100, medium
good, 4, 92, medium
bad, 6, 100, weighty
bad, 8, 170, weighty
good, 4, 89, medium
good, 4, 65, light
bad, 6, 85, medium
bad, 4, 81, light
bad, 6, 95, medium
good, 4, 93, light

NYU

Learning Binary Decision Trees

- Step 1: Start with all data points in a single node. Predict the most common value (classification) or mean value (regression).
- Step 2: Choose the "best" binary decision rule and use it to split the data into two groups.
- Step 3: Repeat step 2 on each group until some stopping criterion is reached.
  - All outputs same?
  - All inputs same?

bad, 4, 81, light
good, 4, 81, light

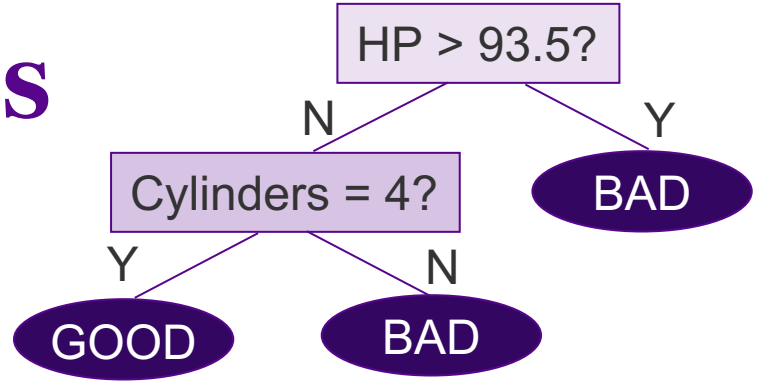


MPG, cylinders, HP, weight

good, 4, 75, light
bad, 6, 90, medium
bad, 4, 110, medium
bad, 8, 175, weighty
bad, 6, 95, medium
bad, 4, 94, light
bad, 4, 95, light
bad, 8, 139, weighty
bad, 8, 190, weighty
bad, 8, 145, weighty
bad, 6, 100, medium
good, 4, 92, medium
bad, 6, 100, weighty
bad, 8, 170, weighty
good, 4, 89, medium
good, 4, 65, light
bad, 6, 85, medium
bad, 4, 81, light
bad, 6, 95, medium
good, 4, 93, light

NYU

Learning Binary Decision Trees



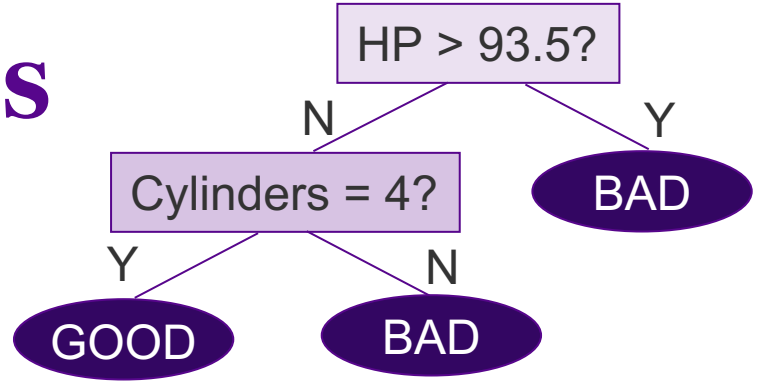
- Step 1: Start with all data points in a single node. Predict the most common value (classification) or mean value (regression).
- Step 2: Choose the "best" binary decision rule and use it to split the data into two groups.
- Step 3: Repeat step 2 on each group until some stopping criterion is reached.
- Step 4: Prune the tree to remove irrelevant rules.

MPG, cylinders, HP, weight

good, 4, 75, light
bad, 6, 90, medium
bad, 4, 110, medium
bad, 8, 175, weighty
bad, 6, 95, medium
bad, 4, 94, light
bad, 4, 95, light
bad, 8, 139, weighty
bad, 8, 190, weighty
bad, 8, 145, weighty
bad, 6, 100, medium
good, 4, 92, medium
bad, 6, 100, weighty
bad, 8, 170, weighty
good, 4, 89, medium
good, 4, 65, light
bad, 6, 85, medium
bad, 4, 81, light
bad, 6, 95, medium
good, 4, 93, light

NYU

Learning Binary Decision Trees



- Step 1: Start with all data points in a single node. Predict the most common value (classification) or mean value (regression).
- Step 2: Choose the "best" binary decision rule and use it to split the data into two groups.
- Step 3: Repeat step 2 on each group until some stopping criterion is reached.
- Step 4: Prune the tree to remove irrelevant rules.

Question 1: How to choose the best decision rule for a given node?

Question 2: How to prune the tree (and why bother?)

MPG, cylinders, HP, weight

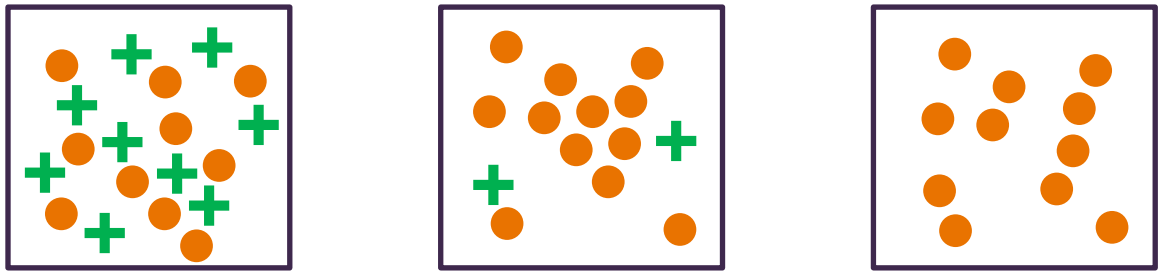
good, 4, 75, light
bad, 6, 90, medium
bad, 4, 110, medium
bad, 8, 175, weighty
bad, 6, 95, medium
bad, 4, 94, light
bad, 4, 95, light
bad, 8, 139, weighty
bad, 8, 190, weighty
bad, 8, 145, weighty
bad, 6, 100, medium
good, 4, 92, medium
bad, 6, 100, weighty
bad, 8, 170, weighty
good, 4, 89, medium
good, 4, 65, light
bad, 6, 85, medium
bad, 4, 81, light
bad, 6, 95, medium
good, 4, 93, light

NYU

Q1: How to Choose Best Split

Q1: How to Choose Best Split

Very impure group | Less impure | Minimum impurity



Impurity (Diversity) Measures:

- Entropy (information gain)
- Information Gain Ratio
- Gini (population diversity)
- Sum of Square Errors

Want to measure "purity" of the split

More certain about Yes/No after split:

- Pure set (**5 yes/15 no**), then completely certain (100%)
- Impure set (**10 yes/10 no**), then completely uncertain (50%)

Can't use Purity ("yes"| set):

- Must be symmetric: **5 yes/0 no** as pure as **0 yes/5 no**

NYU

Source: Witten and Frank 2005, ch. 4 and Soheili, Business Data Analytics, 2019

Q1: How to Choose Best Split

Impurity (Diversity) Measures:

- **Entropy (information gain)**
- Information Gain Ratio
- Gini (population diversity)
- Sum of Square Errors

NYU

Source: Witten and Frank 2005, ch. 4 and Soheili, Business Data Analytics, 2019

Choosing a Decision Rule

- We can use any input attribute to split.
  - If discrete: choose a class, split into = and ≠.
  - If real: choose a threshold, split into > and ≤.

- To choose a threshold for a real attribute, sort the values and use midpoints.

Split	Group Y	Group N
Cylinders = 4?	5+ / 4-	0+ / 11-
Cylinders = 6?	0+ / 6-	5+ / 9-
Cylinders = 8?	0+ / 5-	5+ / 10-
HP > 78?	2+ / 0-	3+ / 15-
HP > 87?	2+ / 2-	3+ / 13-
HP > 89.5?	3+ / 2-	2+ / 13-
HP > 91?	3+ / 3-	2+ / 12-
HP > 93.5?	5+ / 3-	0+ / 12-
Weight = light?	3+ / 3-	2+ / 12-
Weight = medium?	2+ / 6-	3+ / 9-
Weight = heavy?	0+ / 6-	5+ / 9-

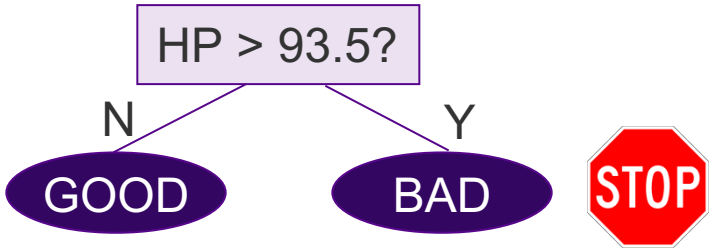
MPG, cylinders, HP, weight

good, 4, 75, light
bad, 6, 90, medium
bad, 4, 110, medium
bad, 8, 175, weighty
bad, 6, 95, medium
bad, 4, 94, light
bad, 4, 95, light
bad, 8, 139, weighty
bad, 8, 190, weighty
bad, 8, 145, weighty
bad, 6, 100, medium
good, 4, 92, medium
bad, 6, 100, weighty
bad, 8, 170, weighty
good, 4, 89, medium
good, 4, 65, light
bad, 6, 85, medium
bad, 4, 81, light
bad, 6, 95, medium
good, 4, 93, light

NYU

Choosing a Decision Rule

- We can use any input attribute to split.
  - If discrete: choose a class, split into = and ≠.
  - If real: choose a threshold, split into > and ≤.



(5 good, 3 bad) (0 good, 12 bad)

- To choose a threshold for a real attribute, sort the values and use midpoints.
- Choose the split with the highest information gain.

Split	Group Y	Group N	Gain
Cylinders = 4?	5+ / 4-	0+ / 11-	0.365
Cylinders = 6?	0+ / 6-	5+ / 9-	0.153
Cylinders = 8?	0+ / 5-	5+ / 10-	0.123
HP > 78?	2+ / 0-	3+ / 15-	0.226
HP > 87?	2+ / 2-	3+ / 13-	0.054
HP > 89.5?	3+ / 2-	2+ / 13-	0.144
HP > 91?	3+ / 3-	2+ / 12-	0.097
HP > 93.5?	**5+ / 3-**	**0+ / 12-**	**0.430**
Weight = light?	3+ / 3-	2+ / 12-	0.097
Weight = medium?	2+ / 6-	3+ / 9-	0.000
Weight = heavy?	0+ / 6-	5+ / 9-	0.153

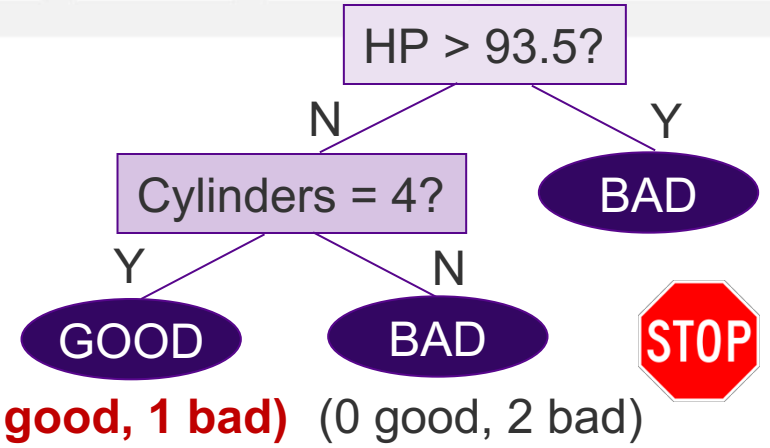
MPG, cylinders, HP, weight

good, 4, 75, light
bad, 6, 90, medium
bad, 4, 110, medium
bad, 8, 175, weighty
bad, 6, 95, medium
bad, 4, 94, light
bad, 4, 95, light
bad, 8, 139, weighty
bad, 8, 190, weighty
bad, 8, 145, weighty
bad, 6, 100, medium
good, 4, 92, medium
bad, 6, 100, weighty
bad, 8, 170, weighty
good, 4, 89, medium
good, 4, 65, light
bad, 6, 85, medium
bad, 4, 81, light
bad, 6, 95, medium
good, 4, 93, light

NYU

Choosing a Decision Rule

- We repeat this process for each non-terminal node of the tree.



Split	Group Y	Group N	Gain
Cylinders = 4?	**5+ / 1-**	**0+ / 2-**	**0.467**
HP > 78?	2+ / 0-	3+ / 3-	0.204
HP > 87?	2+ / 2-	3+ / 1-	0.049
HP > 89.5?	3+ / 2-	2+ / 1-	0.003
HP > 91?	3+ / 3-	2+ / 0-	0.204
Weight = light?	3+ / 1-	2+ / 2-	0.049

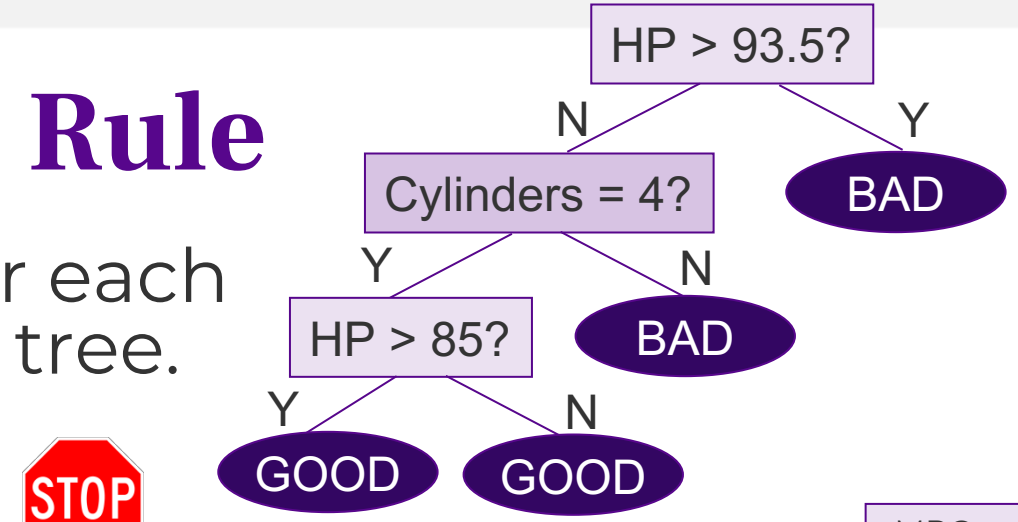
MPG, cylinders, HP, weight

good, 4, 75, light
bad, 6, 90, medium
bad, 4, 110, medium
bad, 8, 175, weighty
bad, 6, 95, medium
bad, 4, 94, light
bad, 4, 95, light
bad, 8, 139, weighty
bad, 8, 190, weighty
bad, 8, 145, weighty
bad, 6, 100, medium
good, 4, 92, medium
bad, 6, 100, weighty
bad, 8, 170, weighty
good, 4, 89, medium
good, 4, 65, light
bad, 6, 85, medium
bad, 4, 81, light
bad, 6, 95, medium
good, 4, 93, light

NYU

Choosing a Decision Rule

- We repeat this process for each non-terminal node of the tree.



(3 good, 0 bad)**(2 good, 1 bad)**

Split	Group Y	Group N	Gain
HP > 78?	3+ / 1-	2+ / 0-	0.109
HP > 85?	**3+ / 0-**	**2+ / 1-**	**0.191**
Weight = light?	3+ / 1-	2+ / 0-	0.109

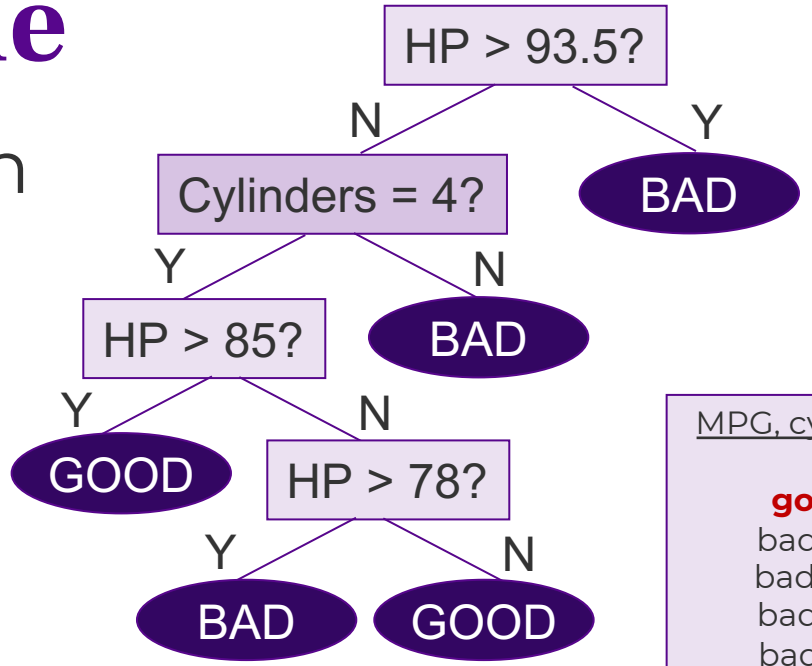
MPG, cylinders, HP, weight

good, 4, 75, light
bad, 6, 90, medium
bad, 4, 110, medium
bad, 8, 175, weighty
bad, 6, 95, medium
bad, 4, 94, light
bad, 4, 95, light
bad, 8, 139, weighty
bad, 8, 190, weighty
bad, 8, 145, weighty
bad, 6, 100, medium
good, 4, 92, medium
bad, 6, 100, weighty
bad, 8, 170, weighty
good, 4, 89, medium
good, 4, 65, light
bad, 6, 85, medium
bad, 4, 81, light
bad, 6, 95, medium
good, 4, 93, light

NYU

Choosing a Decision Rule

- We repeat this process for each non-terminal node of the tree.



Split	Group Y	Group N	Gain
HP > 78?	**0+ / 1-**	**2+ / 0-**	**0.918**

MPG, cylinders, HP, weight

good, 4, 75, light
bad, 6, 90, medium
bad, 4, 110, medium
bad, 8, 175, weighty
bad, 6, 95, medium
bad, 4, 94, light
bad, 4, 95, light
bad, 8, 139, weighty
bad, 8, 190, weighty
bad, 8, 145, weighty
bad, 6, 100, medium
good, 4, 92, medium
bad, 6, 100, weighty
bad, 8, 170, weighty
good, 4, 89, medium
good, 4, 65, light
bad, 6, 85, medium
bad, 4, 81, light
bad, 6, 95, medium
good, 4, 93, light

NYU

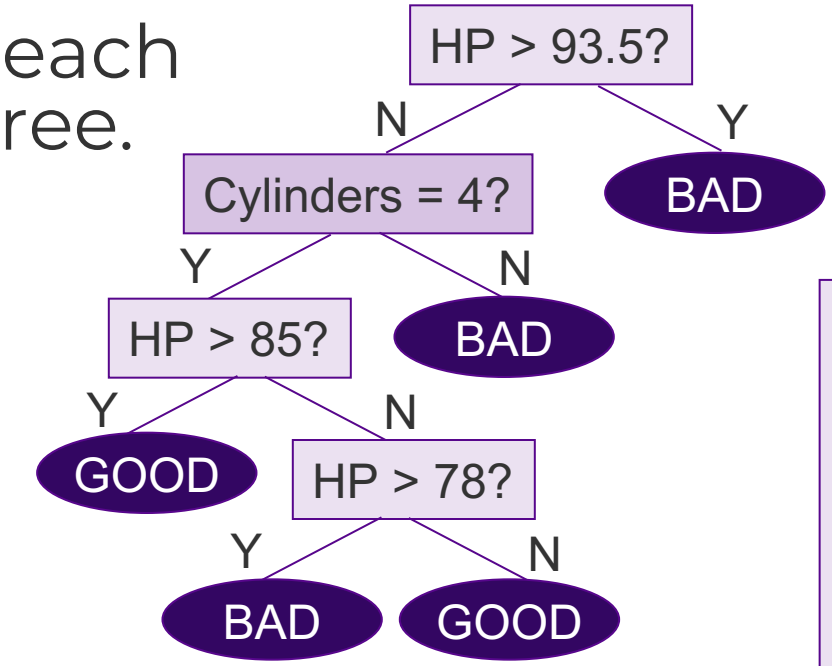
Choosing a Decision Rule

- We repeat this process for each non-terminal node of the tree.

Information gain is an information-theoretic measure of how well the split separates the data.

It can be computed as a function of the numbers of + and – examples in each group.

Group Y	Group N
A+ / B-	C+ / D-



MPG, cylinders, HP, weight

good, 4, 75, light
bad, 6, 90, medium
bad, 4, 110, medium
bad, 8, 175, weighty
bad, 6, 95, medium
bad, 4, 94, light
bad, 4, 95, light
bad, 8, 139, weighty
bad, 8, 190, weighty
bad, 8, 145, weighty
bad, 6, 100, medium
good, 4, 92, medium
bad, 6, 100, weighty
bad, 8, 170, weighty
good, 4, 89, medium
good, 4, 65, light
bad, 6, 85, medium
bad, 4, 81, light
bad, 6, 95, medium
good, 4, 93, light

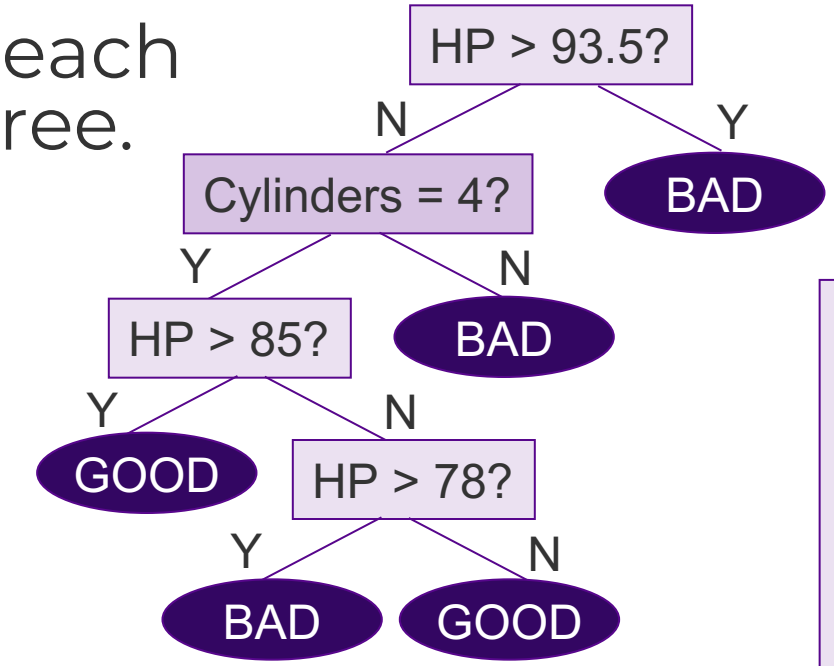
NYU

Choosing a Decision Rule

- We repeat this process for each non-terminal node of the tree.

Information gain is an information-theoretic measure of how well the split separates the data.

It can be computed as a function of the numbers of + and – examples in each group.



MPG, cylinders, HP, weight

good, 4, 75, light
bad, 6, 90, medium
bad, 4, 110, medium
bad, 8, 175, weighty
bad, 6, 95, medium
bad, 4, 94, light
bad, 4, 95, light
bad, 8, 139, weighty
bad, 8, 190, weighty
bad, 8, 145, weighty
bad, 6, 100, medium
good, 4, 92, medium
bad, 6, 100, weighty
bad, 8, 170, weighty
good, 4, 89, medium
good, 4, 65, light
bad, 6, 85, medium
bad, 4, 81, light
bad, 6, 95, medium
good, 4, 93, light

Group Y	Group N
A+ / B-	C+ / D-

$$\text{Gain} = \frac{F((A+C),(B+D)) - F(A,B) - F(C,D)}{A+B+C+D}$$

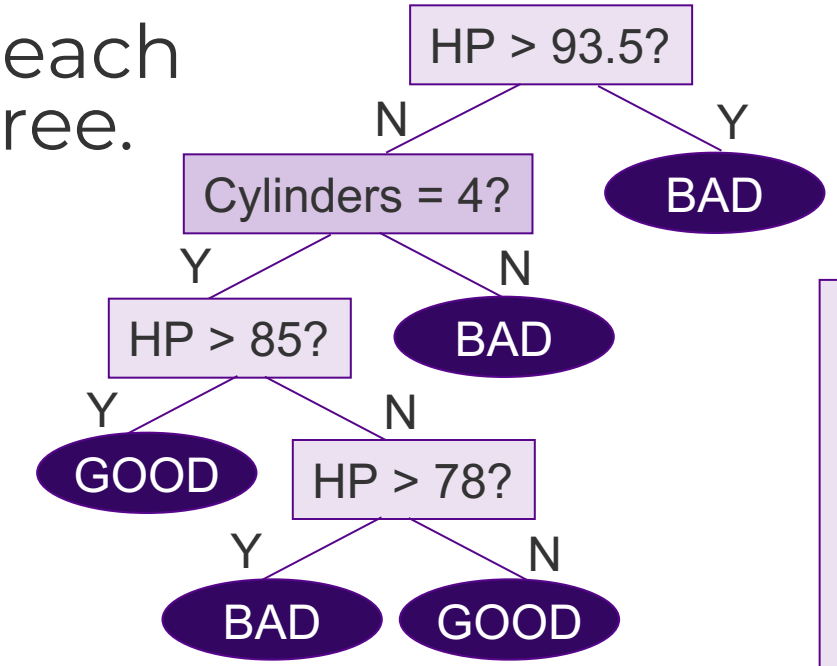
where: $F(X,Y) = X\log_2\frac{X+Y}{X} + Y\log_2\frac{X+Y}{Y}$

(You don't have to memorize this formula.)

NYU

Choosing a Decision Rule

- We repeat this process for each non-terminal node of the tree.



Information gain is an information-theoretic measure of how well the split separates the data.

It can be computed as a function of the numbers of + and – examples in each group.

Group Y	Group N	Gain
3+ / 1-	6+ / 2-	0.000
8+ / 1-	1+ / 2-	0.204
9+ / 0-	0+ / 3-	0.811

Intuitively, the information gain is large when the proportions of positive examples in the two groups are very different and zero when they are the same.

MPG, cylinders, HP, weight

good, 4, 75, light
bad, 6, 90, medium
bad, 4, 110, medium
bad, 8, 175, weighty
bad, 6, 95, medium
bad, 4, 94, light
bad, 4, 95, light
bad, 8, 139, weighty
bad, 8, 190, weighty
bad, 8, 145, weighty
bad, 6, 100, medium
good, 4, 92, medium
bad, 6, 100, weighty
bad, 8, 170, weighty
good, 4, 89, medium
good, 4, 65, light
bad, 6, 85, medium
bad, 4, 81, light
bad, 6, 95, medium
good, 4, 93, light

NYU

Q1: How to Choose Best Split

Impurity (Diversity) Measures:

- Entropy (information gain)
- **Information Gain Ratio**
- Gini (population diversity)
- Sum of Square Errors

The solution to this problem is used to somehow penalize the attributes that lead to a very high number of branches.

- One option is to take into account the number and size of the children nodes, regardless of what classes they contain.
- Instead of using "gain" to determine which attribute to use for the next branch, we shall use "gain ratio," which we shall define as the division between gain and the information value of the attribute.

$$\text{GainRatio}(T) = \text{Gain}(T)/\text{SplitInfo}(T)$$

NYU

Source: Witten and Frank 2005, ch. 4 and Soheili, Business Data Analytics, 2019

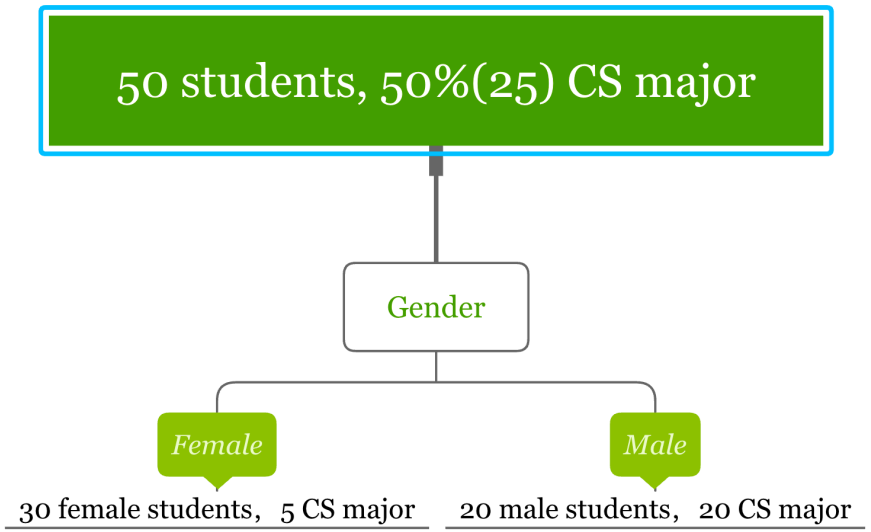
Q1: How to Choose Best Split

Gini impurity is a measure of non-homogeneity.

$$\text{Gini Index} = \sum_i p_i(1 - p_i)$$

Impurity (Diversity) Measures:

- Entropy (information gain)
- Information Gain Ratio
- **Gini (population diversity)**
- Sum of Square Errors



Let's calculate the Gini impurity for splitting node "Gender":
1. Gini impurity for "Female" = 16×56+56×16=518
2. Gini impurity for "Male" = 0×1+1×0=0

The Gini impurity for the node "Gender" is the following weighted average of the above two scores: 35×518+25×0=16

NYU

Source: https://scientistcafe.com/ids/splitting-criteria

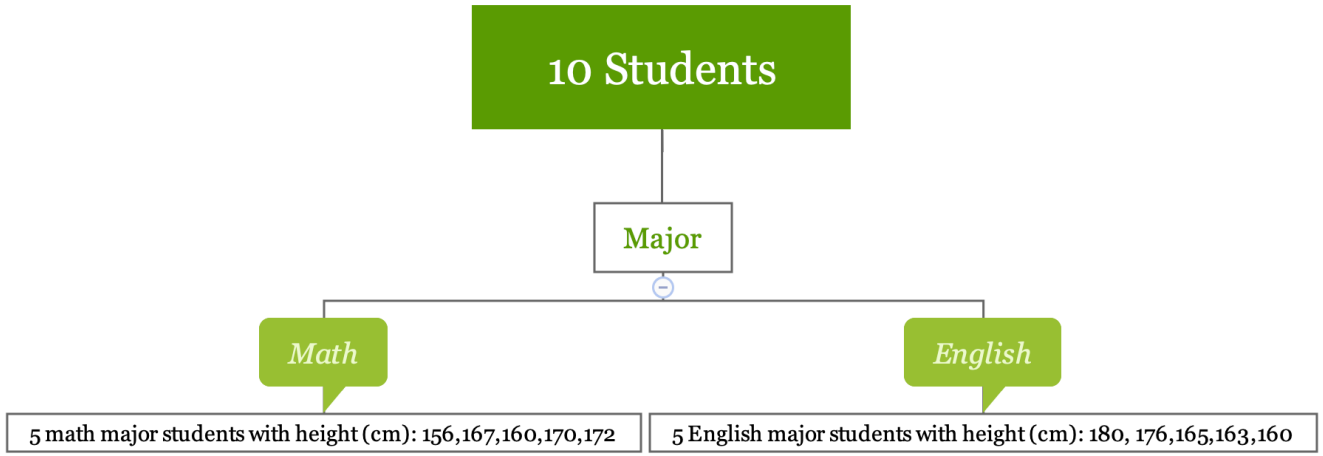
Q1: How to Choose Best Split

The SSE is the most widely used splitting metric for regression.

$$SSE=\Sigma_{i\in S1}(y_i-\bar{y}_1)^2+\Sigma_{i\in S2}(y_i-\bar{y}_2)^2$$

Impurity (Diversity) Measures:

- Entropy (information gain)
- Information Gain Ratio
- Gini (population diversity)
- **Sum of Square Errors**



In this situation:
1. SSE for “Math” is 184
2. SSE for “English” is 302.8
3. SSE for splitting node “Major” is the sum of the above two numbers, which is 486.8

NYU

Source: https://scientistcafe.com/ids/splitting-criteria

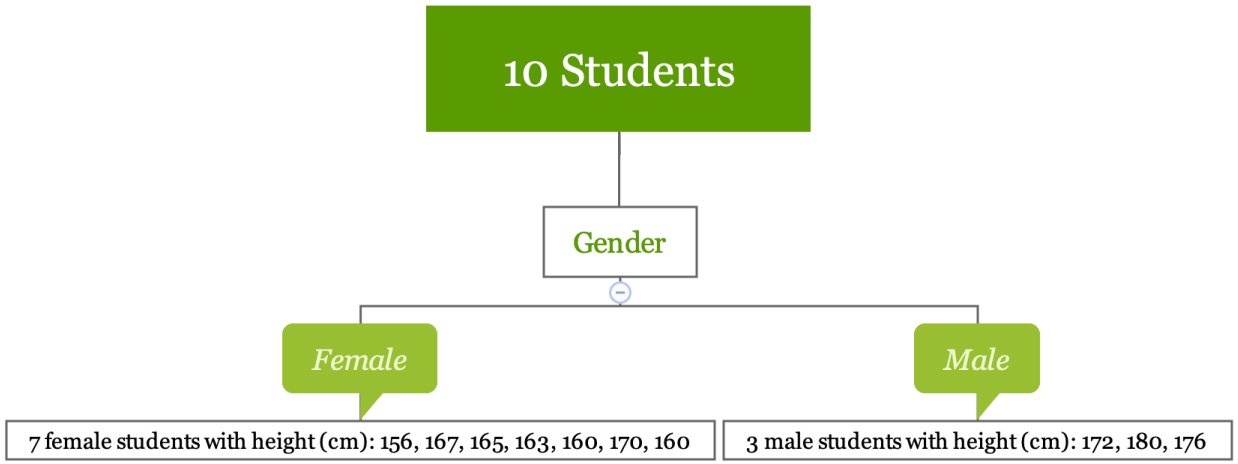
Q1: How to Choose Best Split

The SSE is the most widely used splitting metric for regression.

$$SSE=\Sigma_{i\in S1}(y_i-\bar{y}_1)^2+\Sigma_{i\in S2}(y_i-\bar{y}_2)^2$$

Impurity (Diversity) Measures:

- Entropy (information gain)
- Information Gain Ratio
- Gini (population diversity)
- **Sum of Square Errors**



In this situation:
1. SSE for “Female” is 136
2. SSE for “Male” is 32
3. SSE for splitting node “Gender” is the sum of the above two numbers which is 168
4. SSE for the 10 students in root node is 522.9. After the split, SSE decreases from 522.9 to 168.

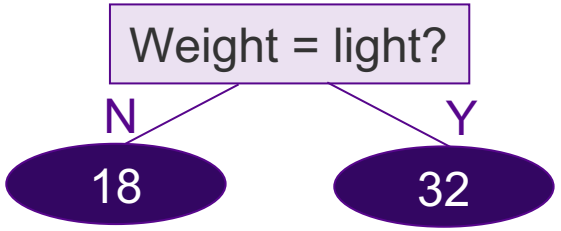
NYU

Source: https://scientistcafe.com/ids/splitting-criteria

Choosing a Decision Rule

- If we are trying to predict a real-valued attribute (i.e. doing regression), minimize the sum of squared errors instead of maximizing information gain.

MPG, cylinders, HP, weight
32, 4, 75, light
20, 6, 95, medium
20, 4, 115, medium
14, 6, 95, medium

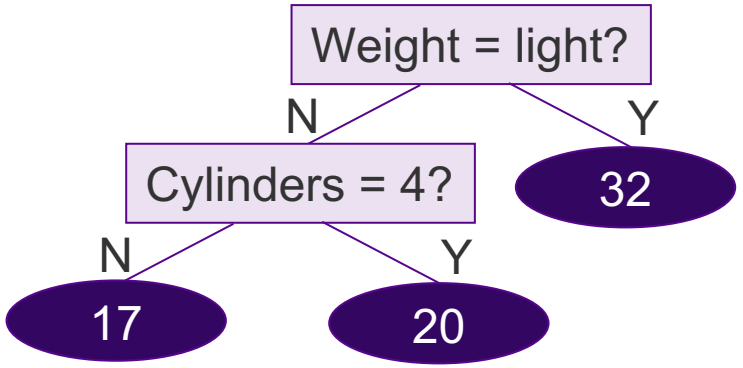


Split	Group Y	Group N	Total SSE
Cylinders = 4?	Predict: 26 SSE: 72	Predict: 17 SSE: 18	90
Weight = light?	**Predict: 32 SSE: 0**	**Predict: 18 SSE: 24**	**24**
HP > 85?	**Predict: 18 SSE: 24**	**Predict: 32 SSE: 0**	**24**
HP > 105?	Predict: 20 SSE: 0	Predict: 22 SSE: 168	168

Choosing a Decision Rule

- If we are trying to predict a real-valued attribute (i.e. doing regression), minimize the sum of squared errors instead of maximizing information gain.

MPG, cylinders, HP, weight
32, 4, 75, light
20, 6, 95, medium
20, 4, 115, medium
14, 6, 95, medium



Split	Group Y	Group N	Total SSE
Cylinders = 4?	**Predict: 20 SSE: 0**	**Predict: 17 SSE: 18**	**18**
HP > 105?	**Predict: 20 SSE: 0**	**Predict: 17 SSE: 18**	**18**

Mathematical Formulation

1. Given training vectors x_i and a label vector y. Let data at tree node m be represented by Q.

$$x_i \in R^n \qquad y \in R^l$$

2. Consider a set of candidate binary splits, each consisting of a feature j and threshold t_m, and partitioning Q into subsets Q_{left} and Q_{right}.

$$\theta = (j, t_m)$$

$$Q_{left}(\theta) = (x, y) | x_j <= t_m$$

$$Q_{right}(\theta) = Q \setminus Q_{left}(\theta)$$

3. Select the split that minimizes the average *impurity* of Q_{left} and Q_{right}, and recurse.

$$\theta^* = \operatorname{argmin}_\theta G(Q, \theta)$$

$$G(Q, \theta) = \frac{n_{left}}{N_m} H(Q_{left}(\theta)) + \frac{n_{right}}{N_m} H(Q_{right}(\theta))$$

4a. For classification, use *cross-entropy.* (p_{mk} are class proportions at node m).

$$p_{mk} = 1/N_m \sum_{x_i \in R_m} I(y_i = k)$$

$$H(X_m) = -\sum_k p_{mk} \log(p_{mk})$$

4b. For regression, use *mean squared error* (c_m is the mean at node m).

$$c_m = \frac{1}{N_m} \sum_{i \in N_m} y_i$$

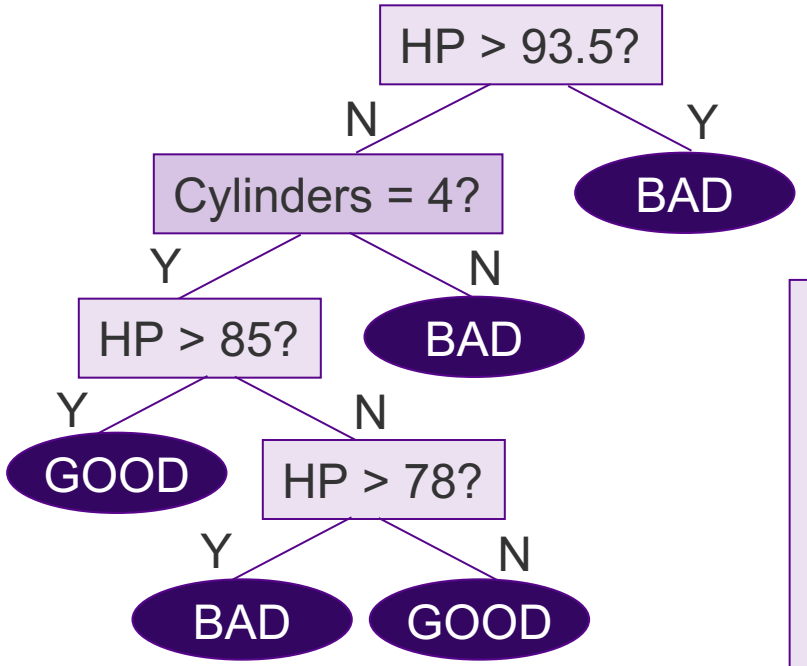
$$H(X_m) = \frac{1}{N_m} \sum_{i \in N_m} (y_i - c_m)^2$$

Source: http://scikit-learn.org/stable/modules/tree.html

Q2: How to Prune the Tree

Pruning Decision Trees

- Notice that the unpruned decision tree classifies every training example perfectly.
- This will always be the case (unless there are records with the same input values and different output values, as in the regression example).
- Does this mean we've found the best tree?



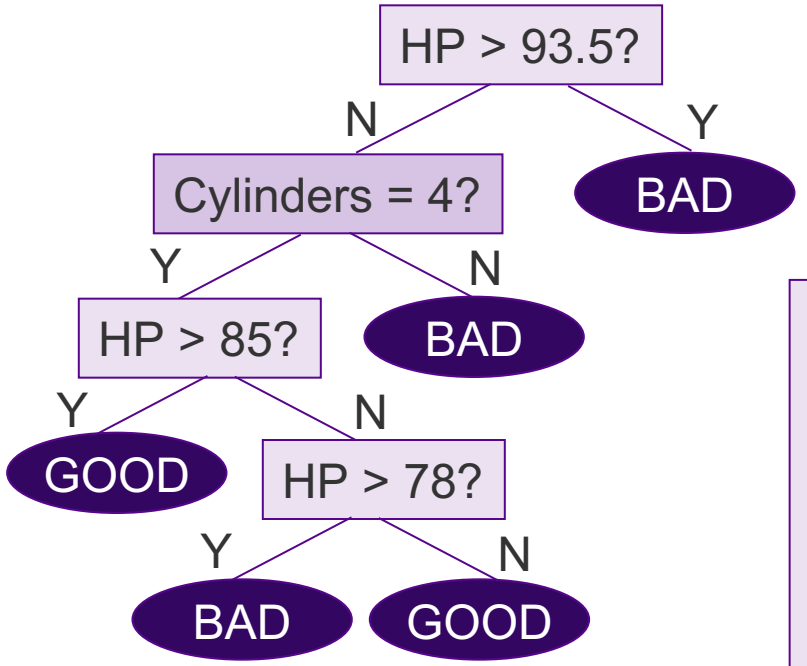
MPG, cylinders, HP, weight

good, 4, 75, light
bad, 6, 90, medium
bad, 4, 110, medium
bad, 8, 175, weighty
bad, 6, 95, medium
bad, 4, 94, light
bad, 4, 95, light
bad, 8, 139, weighty
bad, 8, 190, weighty
bad, 8, 145, weighty
bad, 6, 100, medium
good, 4, 92, medium
bad, 6, 100, weighty
bad, 8, 170, weighty
good, 4, 89, medium
good, 4, 65, light
bad, 6, 85, medium
bad, 4, 81, light
bad, 6, 95, medium
good, 4, 93, light

What we really want to know:
How well does this classifier predict values for new data that we haven't already seen?

Pruning Decision Trees

- One way to answer this question:
  - Hide part of your data (the “test set”).
  - Learn a tree using the rest of the data (the “training set”).
  - See what proportion of the test set is classified correctly.
- Consider pruning each node to see if it reduces test set error.



MPG, cylinders, HP, weight

good, 4, 75, light
bad, 6, 90, medium
bad, 4, 110, medium
bad, 8, 175, weighty
bad, 6, 95, medium
bad, 4, 94, light
bad, 4, 95, light
bad, 8, 139, weighty
bad, 8, 190, weighty
bad, 8, 145, weighty
bad, 6, 100, medium
good, 4, 92, medium
bad, 6, 100, weighty
bad, 8, 170, weighty
good, 4, 89, medium
good, 4, 65, light
bad, 6, 85, medium
bad, 4, 81, light
bad, 6, 95, medium
good, 4, 93, light

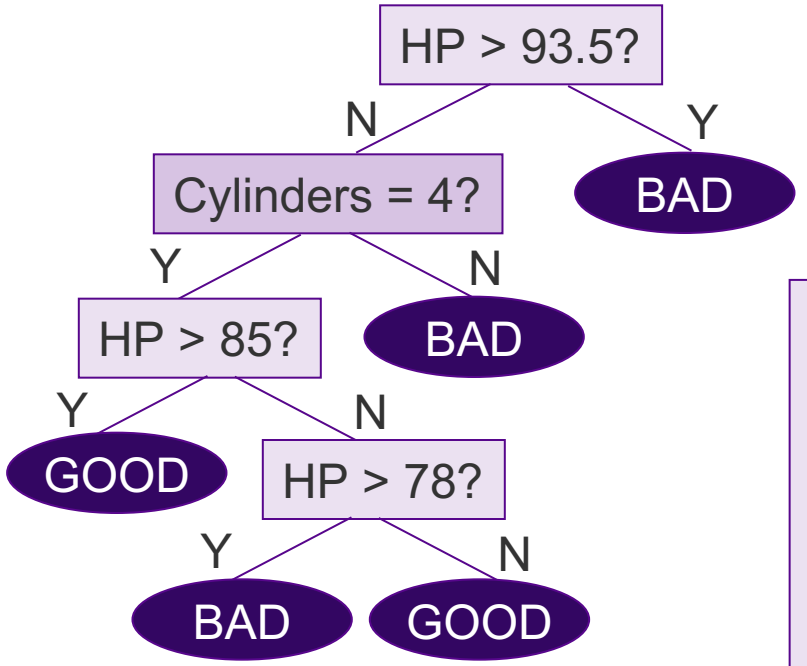
Training set (20 examples):
100% correct, **0%** incorrect

Test set (100 examples):
86% correct, **14%** incorrect

NYU

Pruning Decision Trees

- One way to answer this question:
  - Hide part of your data (the “test set”).
  - Learn a tree using the rest of the data (the “training set”).
  - See what proportion of the test set is classified correctly.
- Consider pruning each node to see if it reduces test set error.



MPG, cylinders, HP, weight

good, 4, 75, light
bad, 6, 90, medium
bad, 4, 110, medium
bad, 8, 175, weighty
bad, 6, 95, medium
bad, 4, 94, light
bad, 4, 95, light
bad, 8, 139, weighty
bad, 8, 190, weighty
bad, 8, 145, weighty
bad, 6, 100, medium
good, 4, 92, medium
bad, 6, 100, weighty
bad, 8, 170, weighty
good, 4, 89, medium
good, 4, 65, light
bad, 6, 85, medium
bad, 4, 81, light
bad, 6, 95, medium
good, 4, 93, light

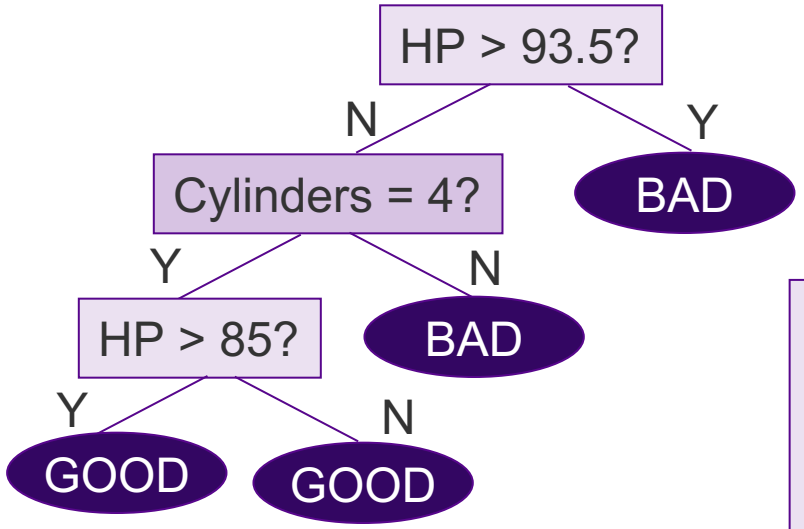
Training set (20 examples):
100% correct, **0%** incorrect

Test set (100 examples):
86% correct, **14%** incorrect

What we really want to know:
How well does this classifier predict values for new data that we haven’t already seen?

Pruning Decision Trees

- One way to answer this question:
  - Hide part of your data (the "test set").
  - Learn a tree using the rest of the data (the "training set").
  - See what proportion of the test set is classified correctly.
- Consider pruning each node to see if it reduces test set error.



Training set (20 examples):
95% correct, **5%** incorrect

Test set (100 examples):
89% correct, **11%** incorrect

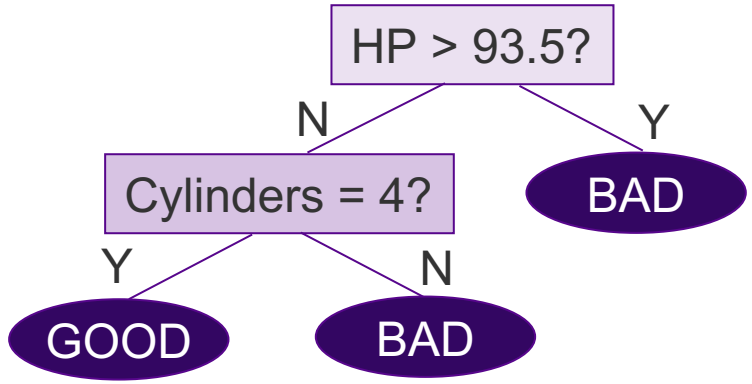
What we really want to know:
How well does this classifier predict values for new data that we haven't already seen?

MPG, cylinders, HP, weight

good, 4, 75, light
bad, 6, 90, medium
bad, 4, 110, medium
bad, 8, 175, weighty
bad, 6, 95, medium
bad, 4, 94, light
bad, 4, 95, light
bad, 8, 139, weighty
bad, 8, 190, weighty
bad, 8, 145, weighty
bad, 6, 100, medium
good, 4, 92, medium
bad, 6, 100, weighty
bad, 8, 170, weighty
good, 4, 89, medium
good, 4, 65, light
bad, 6, 85, medium
bad, 4, 81, light
bad, 6, 95, medium
good, 4, 93, light

Pruning Decision Trees

- One way to answer this question:
  - Hide part of your data (the “test set”).
  - Learn a tree using the rest of the data (the “training set”).
  - See what proportion of the test set is classified correctly.
- Consider pruning each node to see if it reduces test set error.



Training set (20 examples):
95% correct, **5%** incorrect

Test set (100 examples):
89% correct, **11%** incorrect

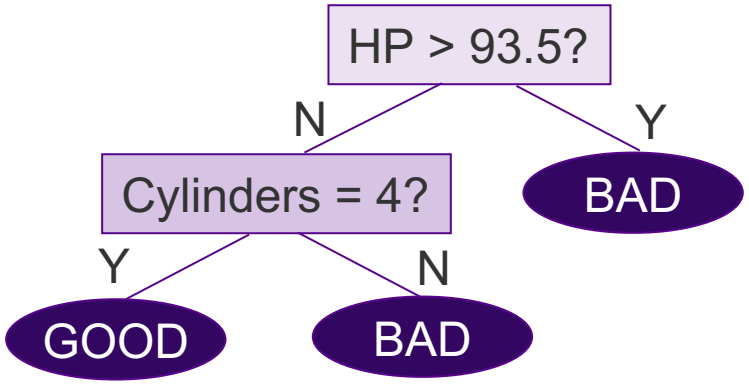
What we really want to know:
How well does this classifier predict values for new data that we haven’t already seen?

MPG, cylinders, HP, weight

good, 4, 75, light
bad, 6, 90, medium
bad, 4, 110, medium
bad, 8, 175, weighty
bad, 6, 95, medium
bad, 4, 94, light
bad, 4, 95, light
bad, 8, 139, weighty
bad, 8, 190, weighty
bad, 8, 145, weighty
bad, 6, 100, medium
good, 4, 92, medium
bad, 6, 100, weighty
bad, 8, 170, weighty
good, 4, 89, medium
good, 4, 65, light
bad, 6, 85, medium
bad, 4, 81, light
bad, 6, 95, medium
good, 4, 93, light

Pruning Decision Trees

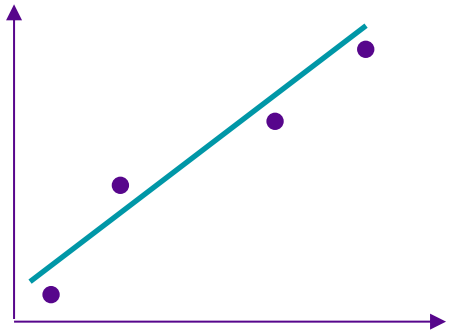
- Q: Why does pruning the tree reduce test set error?
- A: Because the unpruned tree was overfitting the training data (paying attention to parts of the data that are not relevant for prediction).



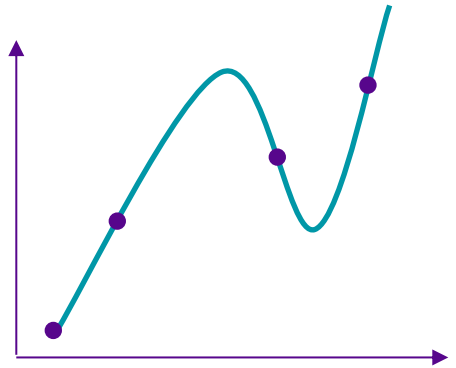
MPG, cylinders, HP, weight

good, 4, 75, light
bad, 6, 90, medium
bad, 4, 110, medium
bad, 8, 175, weighty
bad, 6, 95, medium
bad, 4, 94, light
bad, 4, 95, light
bad, 8, 139, weighty
bad, 8, 190, weighty
bad, 8, 145, weighty
bad, 6, 100, medium
good, 4, 92, medium
bad, 6, 100, weighty
bad, 8, 170, weighty
good, 4, 89, medium
good, 4, 65, light
bad, 6, 85, medium
bad, 4, 81, light
bad, 6, 95, medium
good, 4, 93, light

Here's another example of overfitting, this time for regression:



A line fits pretty well...

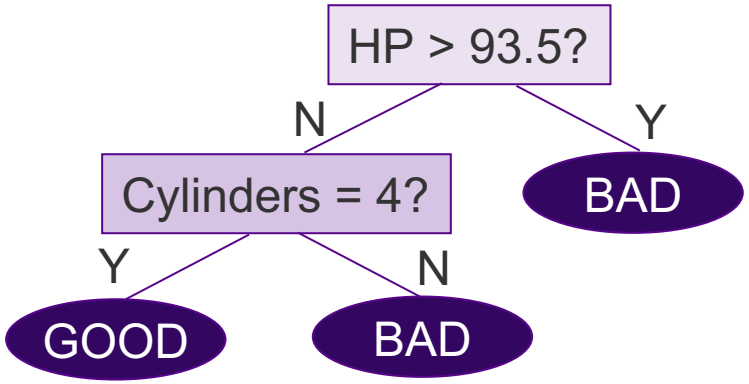


A cubic is probably overfitting.

NYU

Pruning Decision Trees

- Q: What if we don't have enough data points?
- A: Another way to prevent overfitting is to do a certain kind of significance test (chi-squared) for each node and prune any nodes that are not significant.



MPG, cylinders, HP, weight

good, 4, 75, light
bad, 6, 90, medium
bad, 4, 110, medium
bad, 8, 175, weighty
bad, 6, 95, medium
bad, 4, 94, light
bad, 4, 95, light
bad, 8, 139, weighty
bad, 8, 190, weighty
bad, 8, 145, weighty
bad, 6, 100, medium
good, 4, 92, medium
bad, 6, 100, weighty
bad, 8, 170, weighty
good, 4, 89, medium
good, 4, 65, light
bad, 6, 85, medium
bad, 4, 81, light
bad, 6, 95, medium
good, 4, 93, light

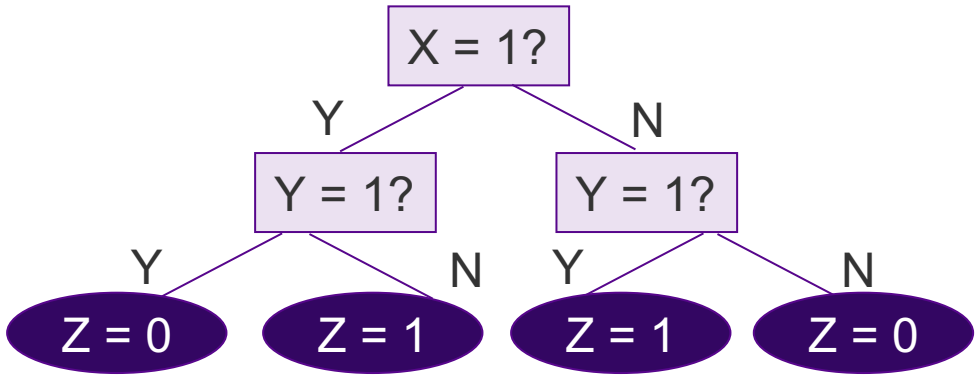
NYU

Pruning Decision Trees

- Q: Why bother building the whole tree if we're just going to prune it?
- A: We could do significance testing while building the tree, but for many datasets, post-pruning gives better performance.

Consider the XOR function:

X	Y	Z
0	0	0
0	1	1
1	0	1
1	1	0

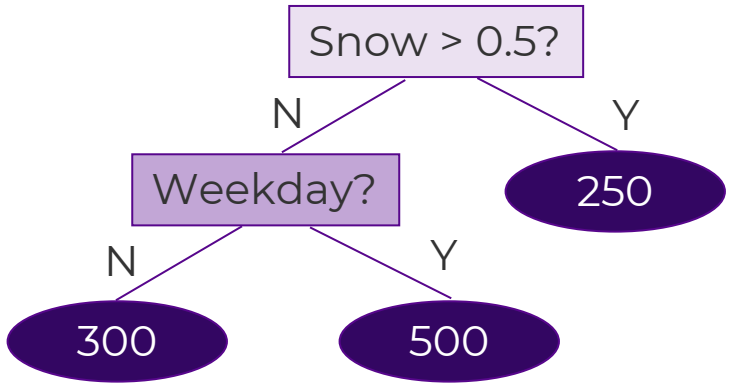


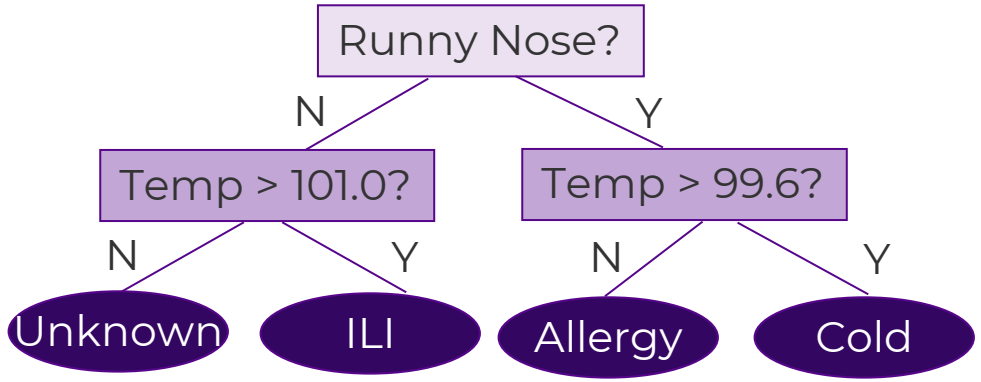
Each second-level split has high information gain. The first level split has no information gain but is needed to make the second level splits possible.

NYU

Some Advantages of Decision Trees

- Easy to learn the tree automatically from a dataset.
- Very easy to predict the target value given the tree.
- Generally good classification performance (though some fancier methods may do better, depending on the dataset).
- It can do both classification and regression and can use both real and discrete inputs.
- It gives an idea of which variables are important in predicting the target value.
  - More important variables tend to be toward the top of the tree.
  - Unimportant variables are not included.





NYU

Some Issues with Decision Trees

1) Multi-way instead of binary splits are okay, but we generally prefer binary.

If using multi-way, use gain ratio instead of information gain to adjust for the arity (# of distinct values) of an attribute. Otherwise, the dataset gets fragmented, leading to poor generalization accuracy (e.g., splitting on country name in the illness dataset).

2) Decision trees can be easily used to predict multiple outputs.

Create a single tree where the split criterion is the average impurity over all outputs.

3) Lots of different ways to handle missing data:
(a) Delete missing observations (not so good); (b) Treat "missing" as its own value (good only if missingness is informative); (c) Surrogate splits (Duda et al. Ch. 8.3.10); (d) Propagate examples with missing values down both branches as partial observations

4) Can create a rule list from the leaves of a decision tree.
Advantages: possibly more interpretable; can prune the rule list directly.

Lab Time

NYU

For the Next Week (Week 3)

1. Read Python documentation at http://scikit-learn.org/stable/modules/tree.html
2. Check readings and review them

References

1. L. Breiman, J.H. Friedman, R.A. Olshen, and C.J. Stone. Classification and Regression Trees, Wadsworth, 1984.
2. J.R. Quinlan. C4.5 : Programs for Machine Learning, Morgan Kaufmann, 1993.
3. T.M. Mitchell. Machine Learning, McGraw-Hill, 1997. (Chapter 3)
4. T. Hastie, R. Tibshirani, J. Friedman. The Elements of Statistical Learning, 2001.