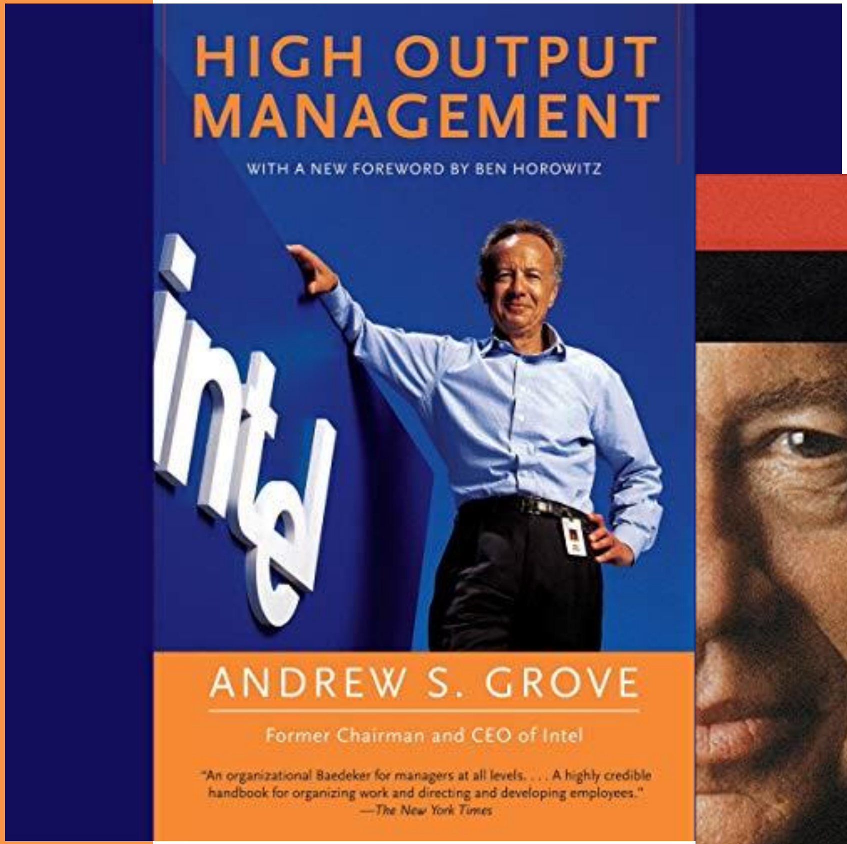
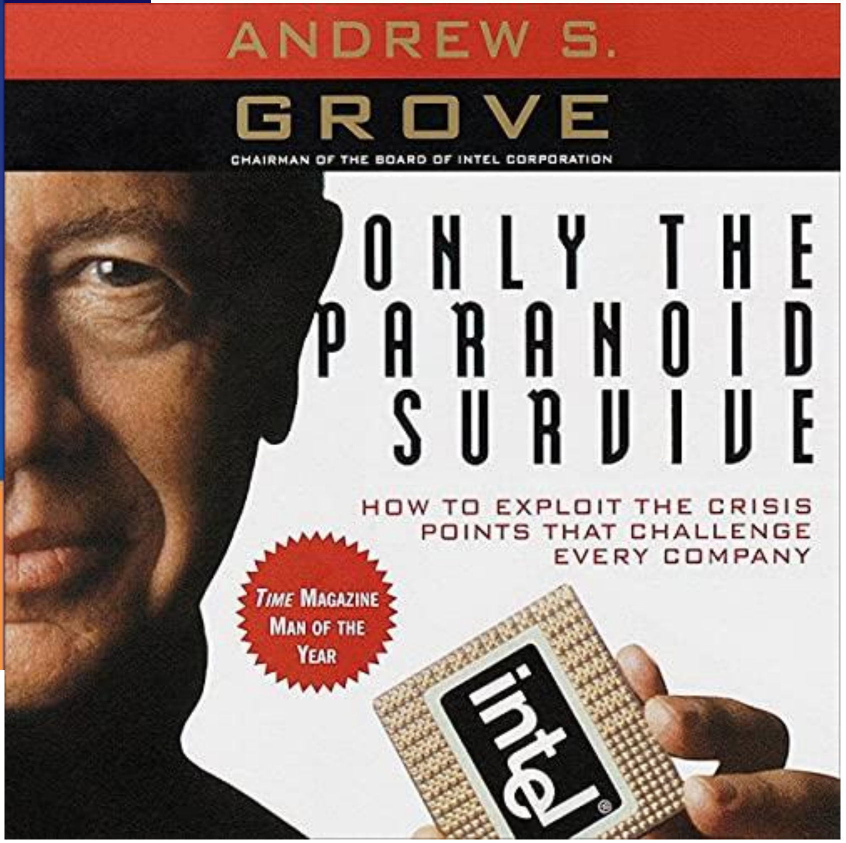
THE UNIVERSITY of TENNESSEE KNOXVILLE

Lecture 05: Decision Trees, Flowers, and the Curious Case of Oxide Growth

Sergei V. Kalinin

What’s after grad school?


HIGH OUTPUT MANAGEMENT
WITH A NEW FOREWORD BY BEN HOROWITZ
ANDREW S. GROVE
Former Chairman and CEO of Intel
"An organizational Baedeker for managers at all levels. . . . A highly credible handbook for organizing work and directing and developing employees." —The New York Times


ANDREW S. GROVE
CHAIRMAN OF THE BOARD OF INTEL CORPORATION
ONLY THE PARANOID SURVIVE
HOW TO EXPLOIT THE CRISIS POINTS THAT CHALLENGE EVERY COMPANY
TIME MAGAZINE MAN OF THE YEAR

The Staff Engineer's Path
A GUIDE FOR INDIVIDUAL CONTRIBUTORS NAVIGATING GROWTH AND CHANGE
TANYA REILLY
Foreword by Camille Fournier, author of The Manager's Path

Amazon Principal Engineers Tenets

As Amazon's most senior individual contributors, Principal Engineers work on Amazon's hardest problems. As they navigate these complex and ambiguous challenges, the Principal Engineering Community uses the following set of tenets to guide them. Tenets are a key part of Amazon's peculiar culture helping to bring Amazonians together and focused on achieving our mission and vision to be Earth's most customer-centric company.

EXEMPLARY PRACTITIONER
Principal Engineers are hands-on and lead by example. We deliver artifacts that set the standard for engineering excellence, from designs to algorithms to implementations. Only by being close to the details can we earn the respect needed to be effective technical leaders.

TECHNICALLY FEARLESS
Amazon’s startup culture does not admit the luxury of conservatism. Principal Engineers tackle intrinsically hard problems, venturing beyond comfortable approaches when necessary. We acquire expertise as needed, pioneer new spaces, and inspire others as to what’s possible.

Amazon Principal Engineers Tenets

LEAD WITH EMPATHY
Principal Engineers shape an inclusive engineering culture where others are heard, feel respected, and are empowered. We are conscious of how our words and demeanor impact others, especially those with less influence; we take responsibility for that impact, intentional or otherwise. Our work builds productive relationships across teams and disciplines, and across a wide range of life experiences.

BALANCED AND PRAGMATIC
Principal Engineers are pragmatic problem solvers. We apply judgment and experience to balance trade-offs between competing interests. We simplify processes and technologies while advocating a long-term view.

ILLUMINATE AND CLARIFY
Principal Engineers bring clarity to complexity and demonstrate smart ways to simplify. We frame each problem in its customer and business context and boil it down to its essence. We probe assumptions, illuminate pitfalls, and foster shared understanding. We accelerate progress by driving crisp and timely decisions.

Amazon Principal Engineers Tenets

FLEXIBLE IN APPROACH

Principal Engineers adapt our approach to meet the needs of the team, project, and product. We solicit differing views and are willing to change our minds as we learn more. We recognize there are often many viable solutions, and that sometimes the best solution is to solve a different problem, or to not solve the problem at all.

RESPECT WHAT CAME BEFORE

Principal Engineers are grateful to our predecessors. We appreciate the value of working systems and the lessons they embody. We understand that many problems are not essentially new.

LEARN, EDUCATE, AND ADVOCATE

Principal Engineers are constantly learning. We seek technical knowledge and educate the entire organization about trends, technologies, and approaches. We combine vision and discretion to drive fruitful and even game-changing technology choices.

HAVE RESOUNDING IMPACT

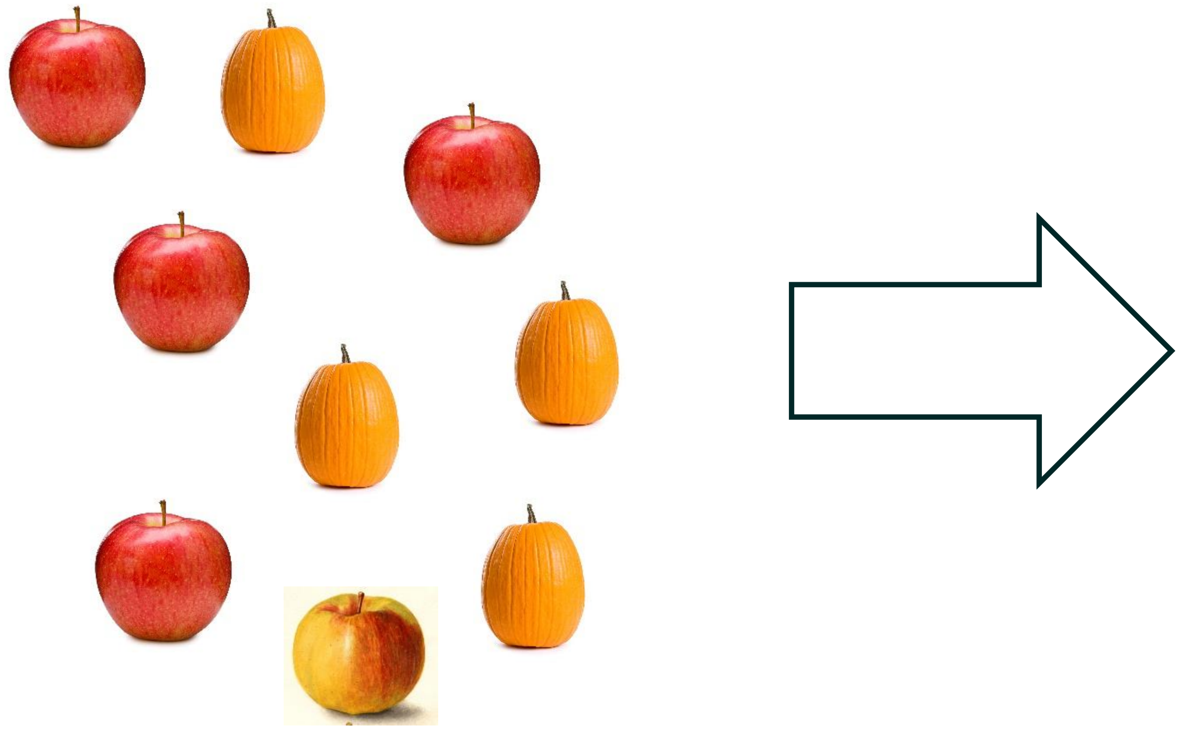
“Deliver Results” is a low bar for a Principal Engineer. Without seeking the spotlight, Principal Engineers make a lasting impact that echoes through the technology, the product, and the company. We amplify our impact by aligning teams toward coherent architectural strategies.

Classification Workflow

1. Selecting features and collecting labeled training examples
2. Choosing a performance metric
3. Choosing a learning algorithm and training a model
    - Can we understand how it works?
    - Does it have any hyperparameters
    - How well does it generalize to new data?
    - How expensive is it?
4. Evaluating the performance of the model
5. Changing the settings of the algorithm and tuning the model.

From S. Raschka, Machine Learning with PyTorch and Scikit-Learn

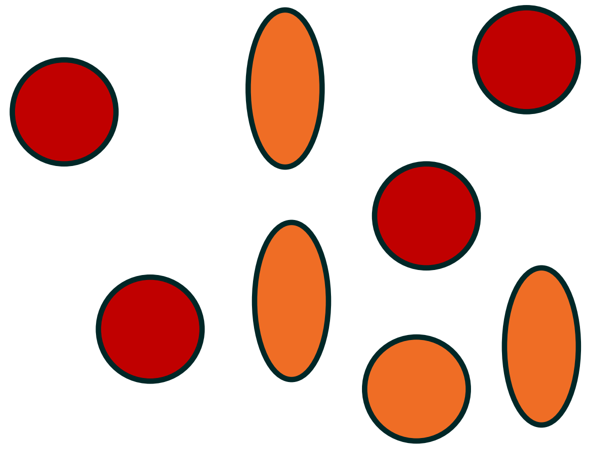
Feature Engineering

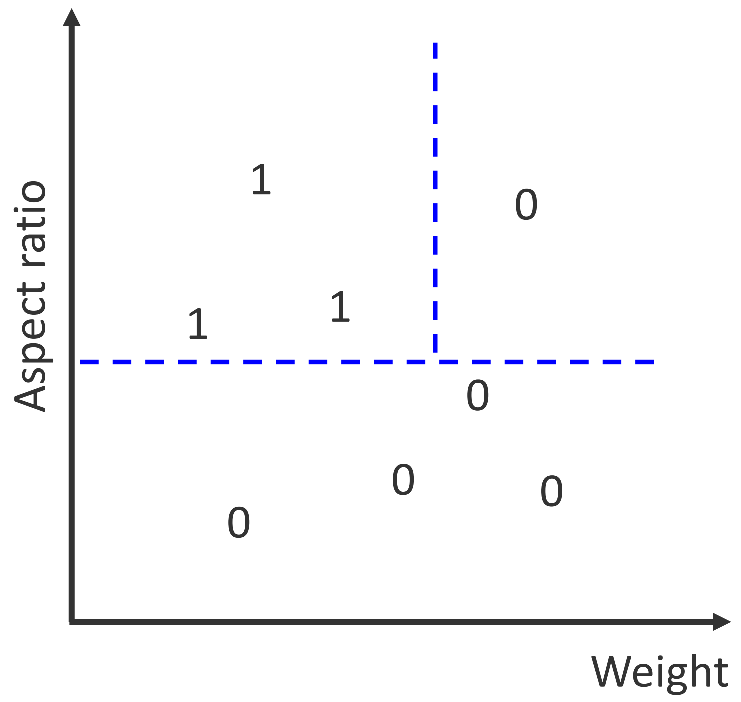


Features: color, shape, weight, aspect ratio

Classification and Regression Tree (CART)

- Create a set of features (shape, color, weight)
- Select a splitting criterion (likelihood):
  - Initialization: create a tree with one node containing all the training data
  - Splitting: find the best way for splitting each terminal node. Split the one terminal node that results in the greatest increase in the likelihood
  - Stopping: if each leaf node contains sample from the same class, stop. Otherwise, continue splitting
  - Pruning: use an independent test set or cross-validation to prune the tree.





How do we split?

Information gain:

$$IG(D_p, f) = I(D_p) - \sum_{j=1}^{m} \frac{N_j}{N_p} I(D_j)$$

- f is the feature to perform the split
- D_p and D_j are the dataset of the parent and jth child node
- I is our **impurity** measure
- N_p is the total number of training examples at the parent node
- N_j is the number of examples in the jth child node

Binary split:

$$IG(D_p, f) = I(D_p) - \frac{N_{left}}{N_p} I(D_{left}) - \frac{N_{right}}{N_p} I(D_{right})$$

From S. Raschka, Machine Learning with PyTorch and Scikit-Learn

What are impurity measures?

Entropy:

$$I_H(t) = -\sum_{i=1}^{c} p(i|t) \log_2 p(i|t)$$

Gini impurity:

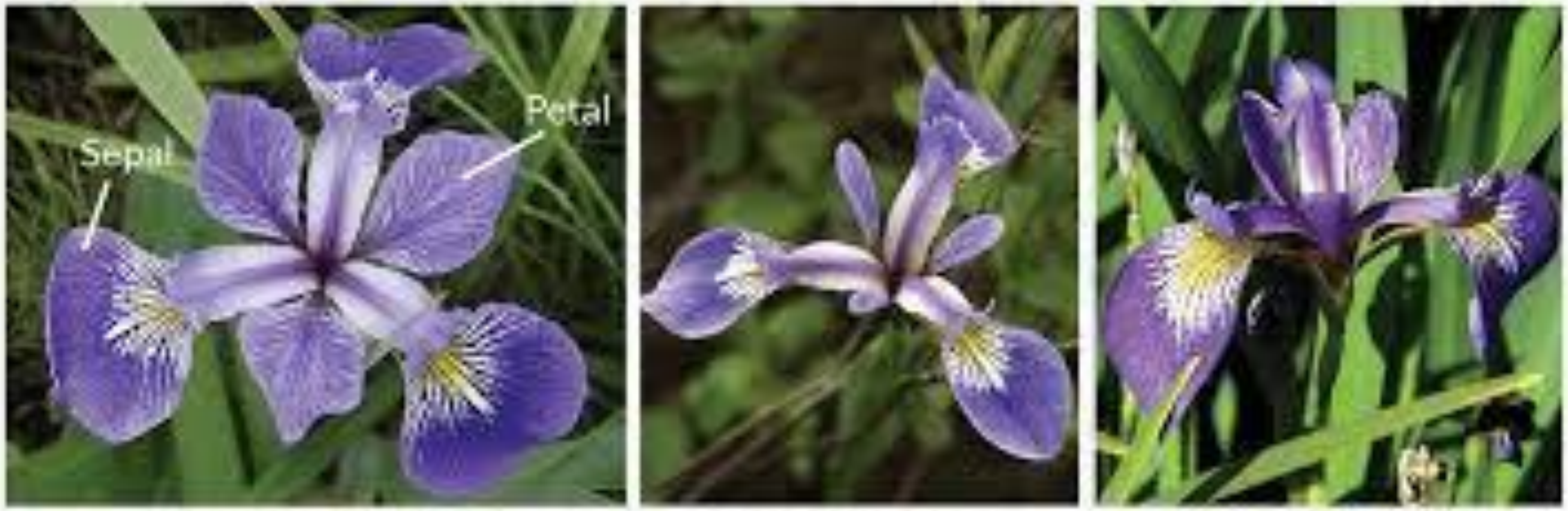
$$I_G(t) = \sum_{i=1}^{c} p(i|t)\,(1 - p(i|t)) = 1 - \sum_{i=1}^{c} p(i|t)^2$$

Classification error: $I_E(t) = 1 - \max\{p(i|t)\}$

- $p(i|t)$ is the proportion of the examples that belong to class i for a node, t.

From S. Raschka, Machine Learning with PyTorch and Scikit-Learn

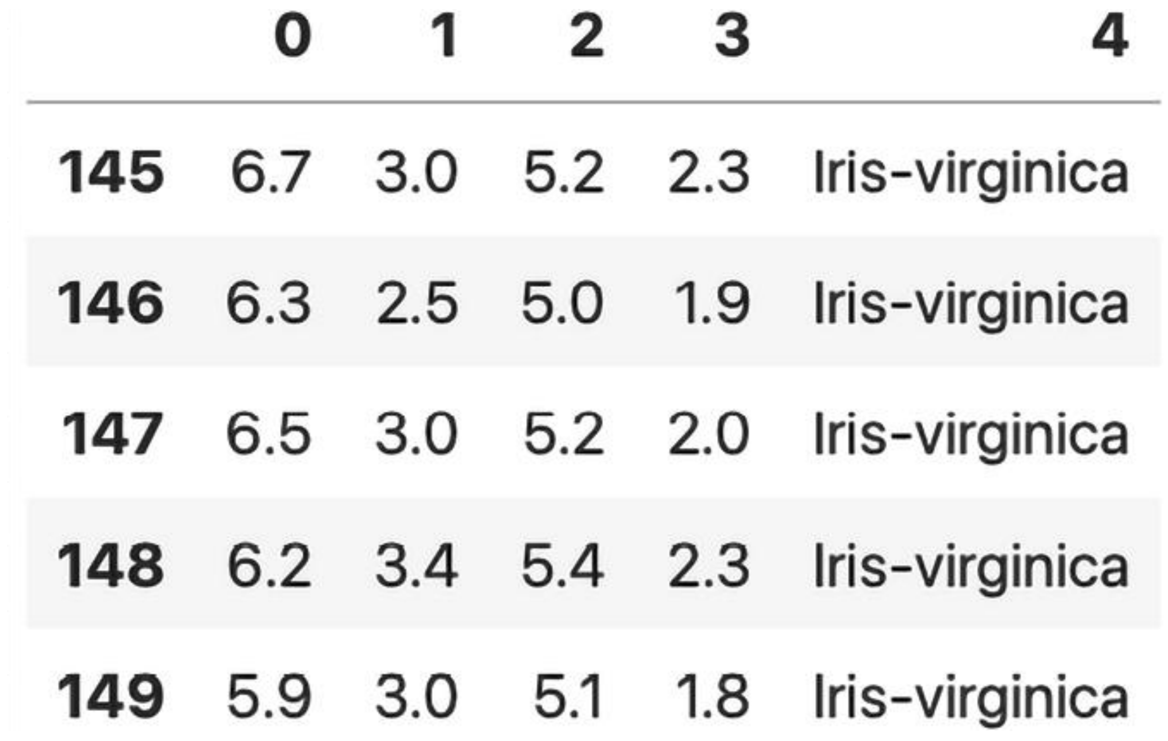
Let’s do some classification!



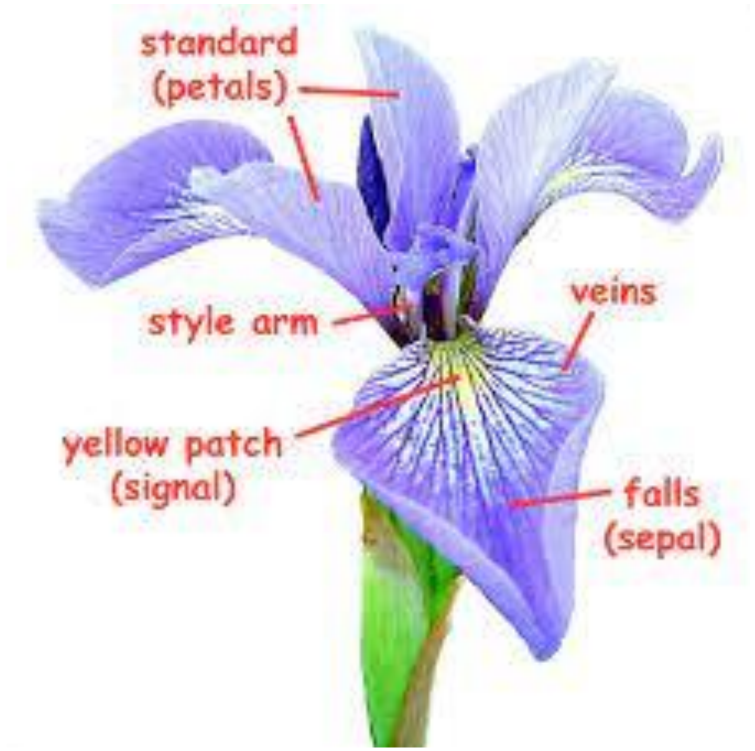
Iris Versicolor Iris Setosa Iris Virginica

http://www.lac.inpe.br/~rafael.santos/Docs/CAP394/WholeStory-Iris.html

Let’s do some classification!



	Sepal length	Sepal width	Petal length	Petal width	
	0	1	2	3	4
145	6.7	3.0	5.2	2.3	Iris-virginica
146	6.3	2.5	5.0	1.9	Iris-virginica
147	6.5	3.0	5.2	2.0	Iris-virginica
148	6.2	3.4	5.4	2.3	Iris-virginica
149	5.9	3.0	5.1	1.8	Iris-virginica



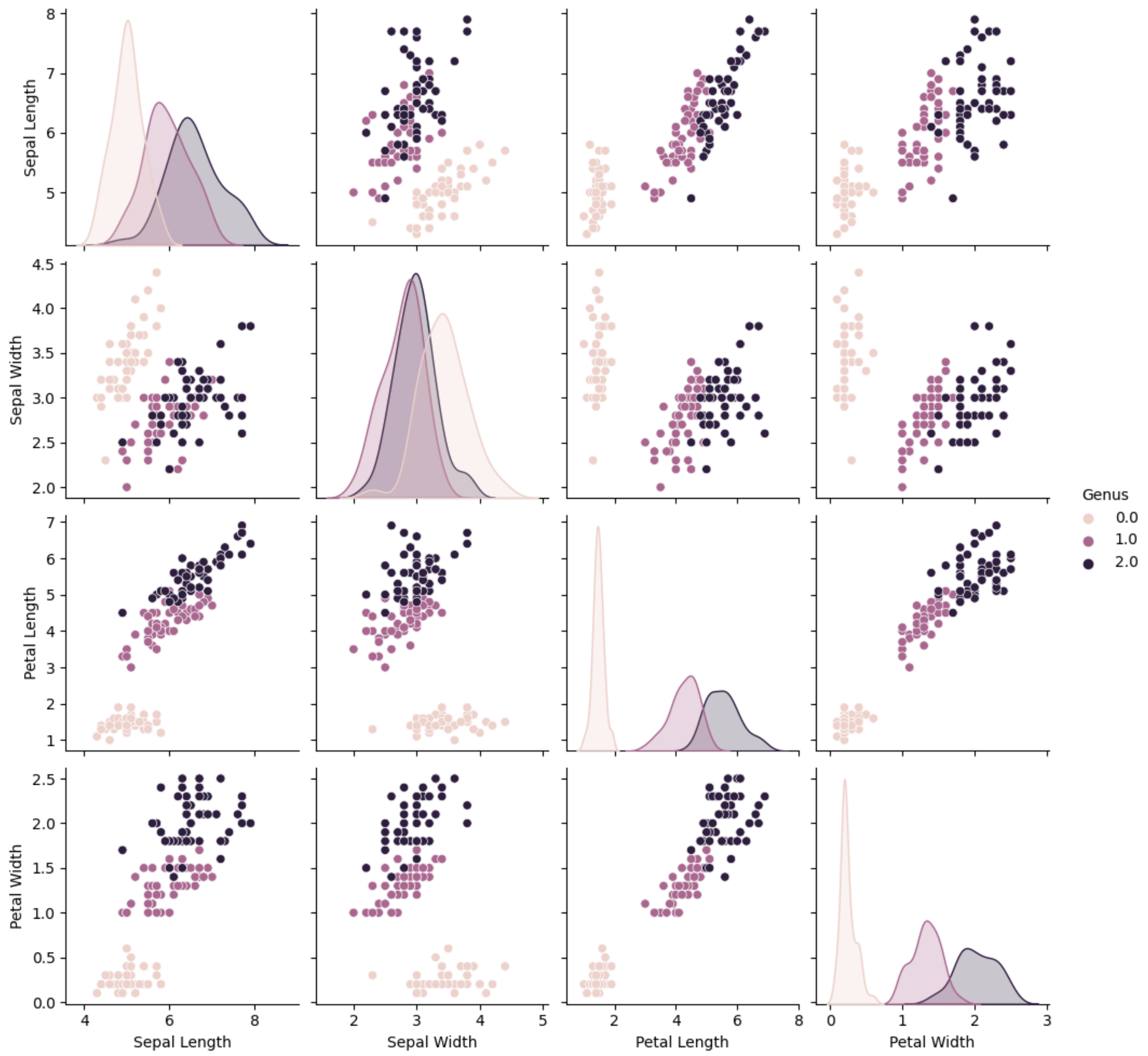
In scikit-learn, Iris-setosa, Iris-versicolor, and Iris-virginica, are already stored as integers

From S. Raschka, Machine Learning with PyTorch and Scikit-Learn

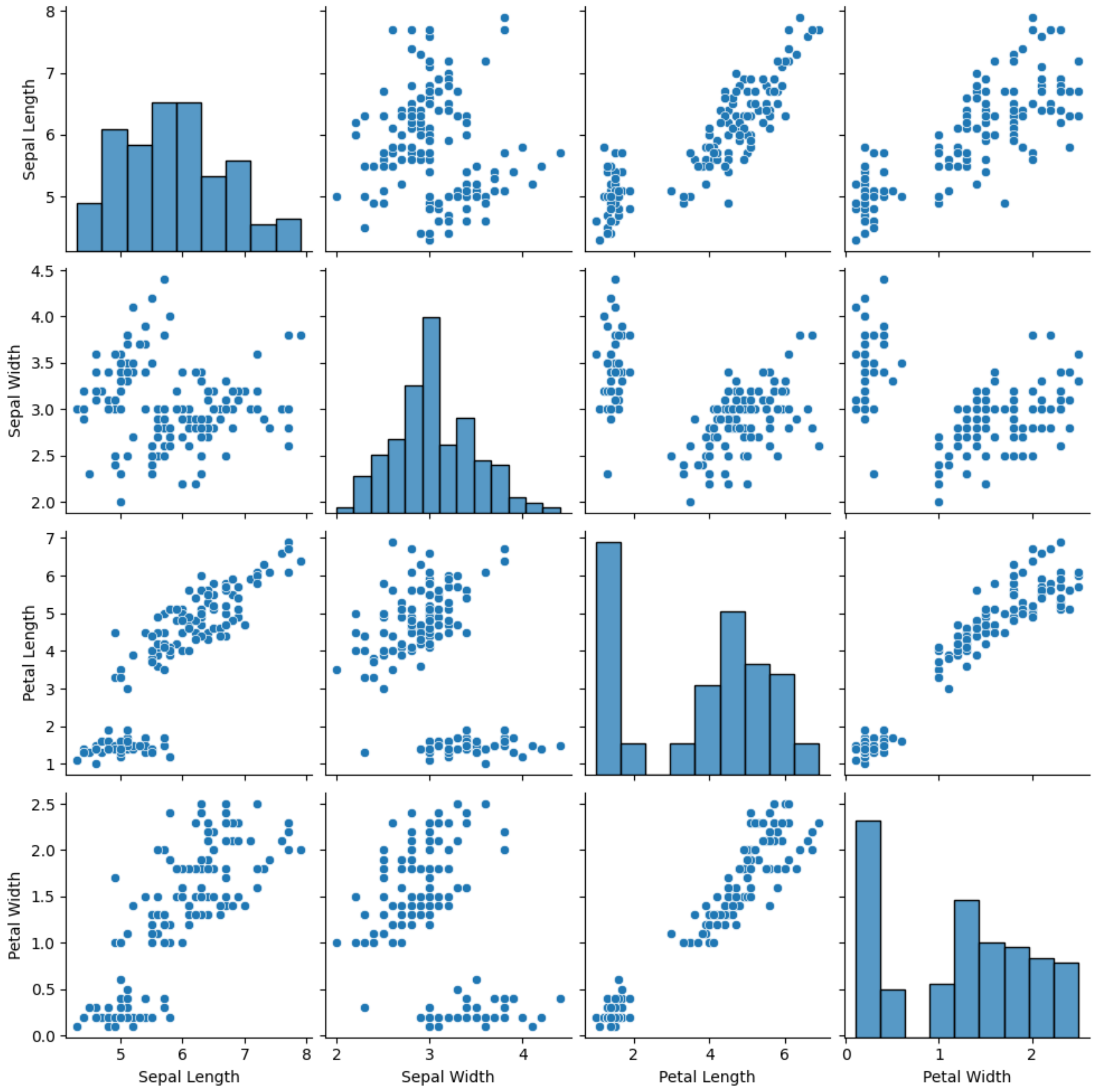
Homework Colabs: please share with

[redacted]@gmail.com

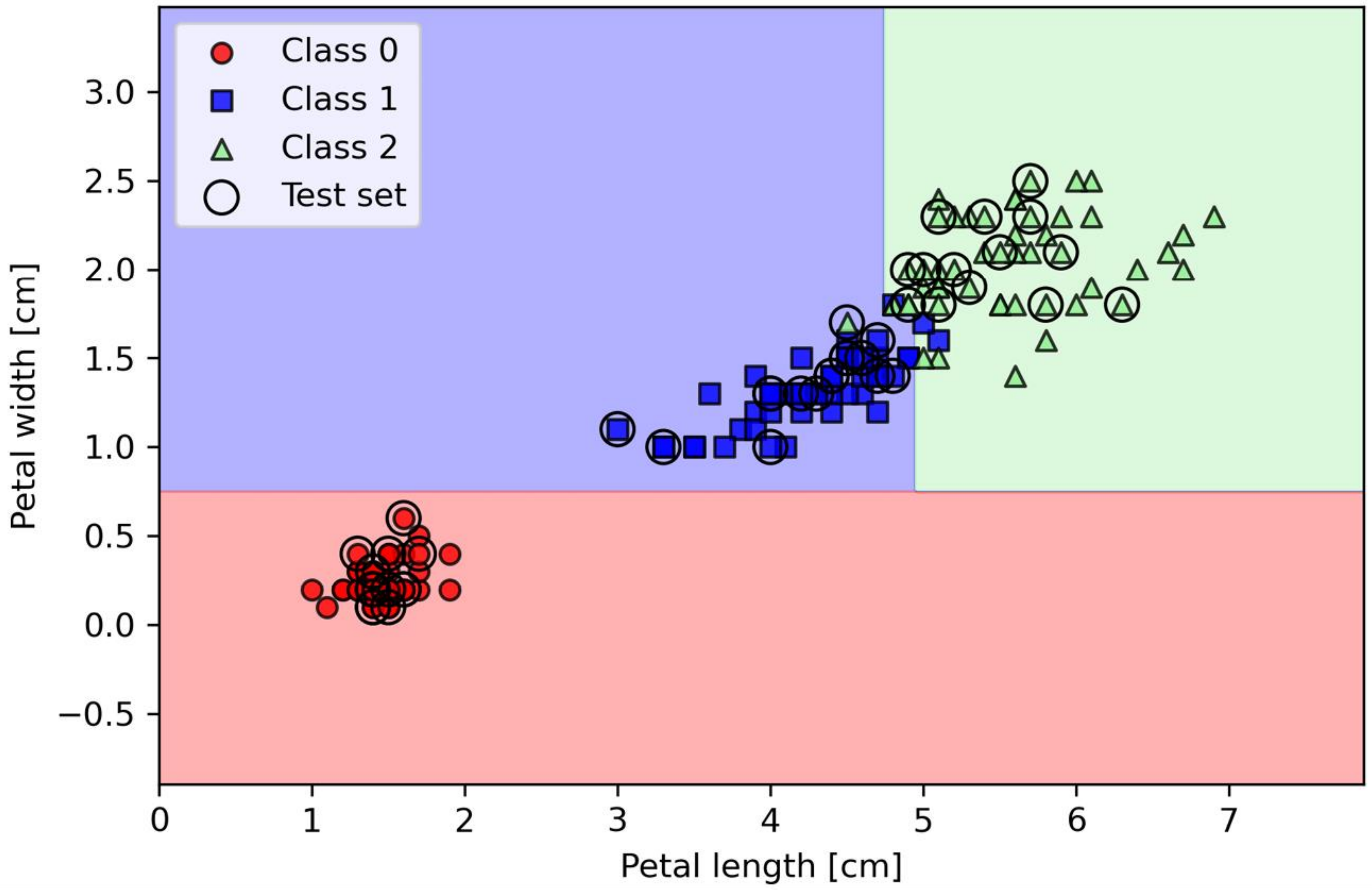
But first, exploratory data analysis…



Can we do it if there is no labels?

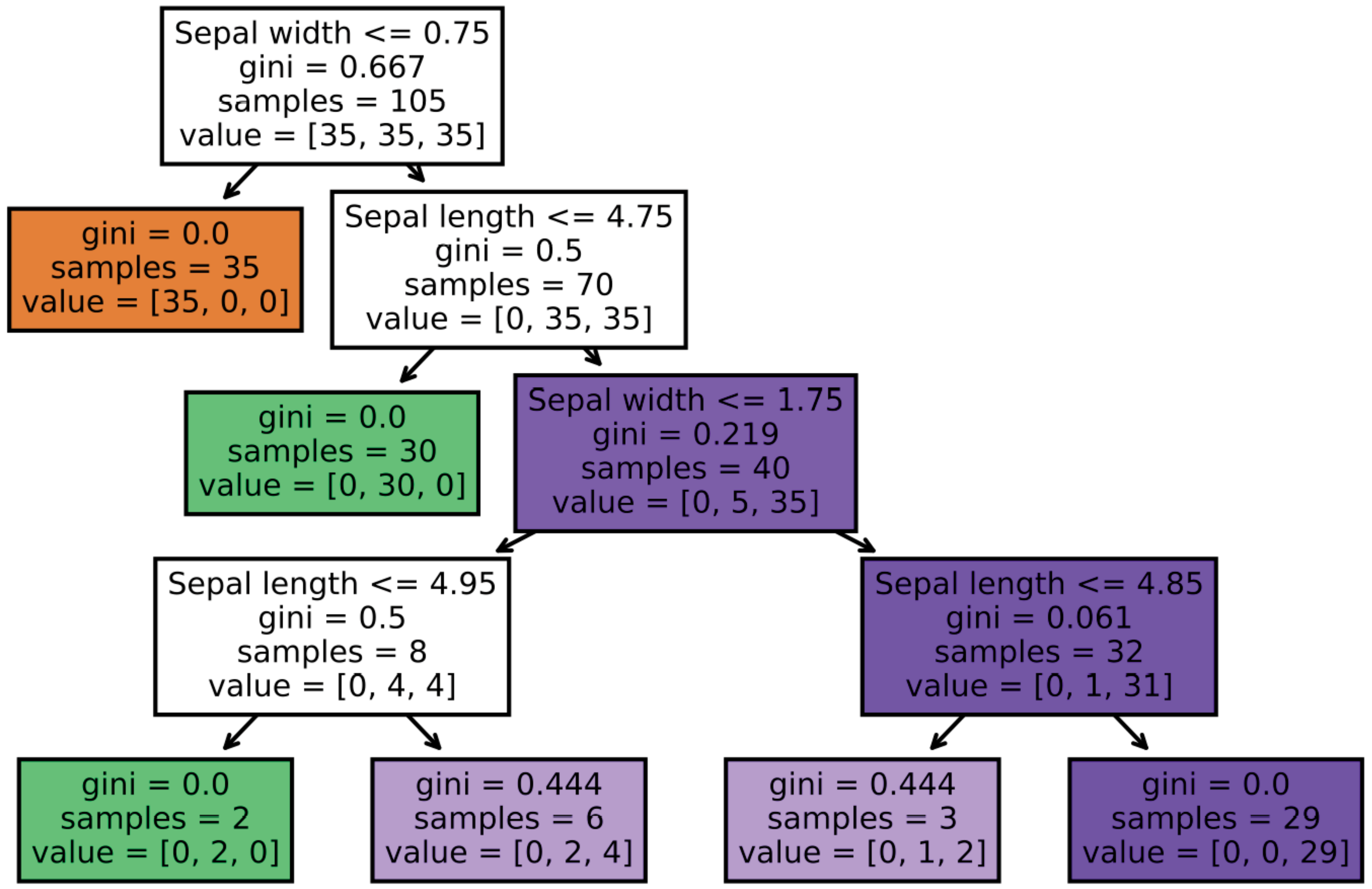


Visualization



From S. Raschka, Machine Learning with PyTorch and Scikit-Learn

Running Decision Tree Algorithm



From S. Raschka, Machine Learning with PyTorch and Scikit-Learn

Conclusion:

- Decision trees are the single most popular data mining tool
  - Easy to understand
  - Easy to implement
  - Easy to use
  - Computationally cheap

- It’s possible to get in trouble with overfitting

- They do classification: predict a categorical output from categorical and/or real inputs

Advantages of Decision Trees - I

- Simple to understand and interpret. People are able to understand decision tree models after a brief explanation. Trees can also be displayed graphically in a way that is easy for non-experts to interpret.

- Able to handle both numerical and categorical data.

- Requires little data preparation. Other techniques often require data normalization. Since trees can handle qualitative predictors, there is no need to create dummy variables.

- Uses a white box or open-box model. If a given situation is observable in a model the explanation for the condition is easily explained by boolean logic. By contrast, in a black box model, the explanation for the results is typically difficult to understand, for example with an artificial neural network.

- Possible to validate a model using statistical tests. That makes it possible to account for the reliability of the model.

https://en.wikipedia.org/wiki/Decision_tree_learning

Advantages of Decision Trees - II

- Non-parametric approach that makes no assumptions of the training data or prediction residuals; e.g., no distributional, independence, or constant variance assumptions

- Performs well with large datasets. Large amounts of data can be analyzed using standard computing resources in reasonable time.

- Mirrors human decision making more closely than other approaches.

- Robust against co-linearity, particularly boosting.

- In built feature selection. Additional irrelevant feature will be less used so that they can be removed on subsequent runs. The hierarchy of attributes in a decision tree reflects the importance of attributes. It means that the features on top are the most informative.

- Decision trees can approximate any Boolean function e.g. XOR.

https://en.wikipedia.org/wiki/Decision_tree_learning

Disadvantages of Decision Trees

- Trees can be very non-robust. A small change in the training data can result in a large change in the tree and consequently the final predictions.
- The problem of learning an optimal decision tree is known to be NP-complete under several aspects of optimality and even for simple concepts. Consequently, practical decision-tree learning algorithms are based on heuristics such as the greedy algorithm where locally optimal decisions are made at each node. Such algorithms cannot guarantee to return the globally optimal decision tree.
- Decision-tree learners can create over-complex trees that do not generalize well from the training data (overfitting.) Mechanisms such as pruning are necessary to avoid this problem
- The average depth of the tree that is defined by the number of nodes or tests till classification is not guaranteed to be minimal or small under various splitting criteria.
- For data including categorical variables with different numbers of levels, information gain in decision trees is biased in favor of attributes with more levels.

https://en.wikipedia.org/wiki/Decision_tree_learning