KNIGHTEC

FORWARD TOGETHER

INTRODUCTION TO ML: CLASSIFICATION, TREES, FORESTS

BY JONAS DÜRANGO AND OLOF RÄNNBÄCK GARPINGER

HANDS ON DATA SCIENCE, FOO CAFÉ, 6/12 - 2018

Before we start

- Presentation, exercises and code examples available on https://github.com/olofgarpinger/hands_on_data_science_Dec_6_2018
- Dataset from Kaggle. We also use Kaggle to evaluate all classifiers. Requires signup!

Strong values take Knightec further

Culture of diversity, consideration and teamwork

SVERIGES BÄSTA ARBETSGIVARE
SVERIGE
universum
2018

Knightec
Position 3

ÅRETS VD
2014, 2018 Dimitris Gioulekas

Quality & Management

Project Management

Quality Assurance

Business development

Risk Management

Specialist area: Compliance. Optimized.

Systems

Software

Electronics

Automation

Machine Learning and Data Science

Specialist area: Connected Device Security

Technology

Mechanical Engineering

Machine Design

Calculation

Certifications

Specialist area: Sustainable Plastic Design

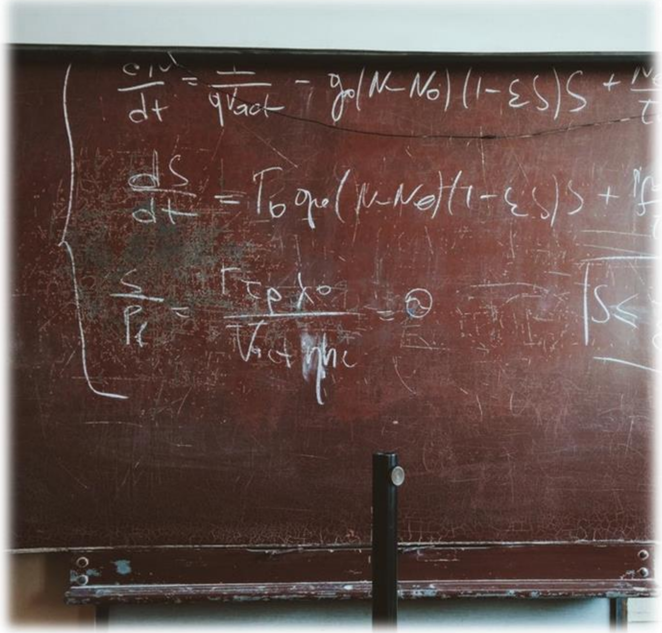
Classification (non ML...)

- Simple concept: assign a new observation to one of several classes.
- How? Some tasks are simple, such as the toy.
  - Pieces have distinct properties such as color, shape, size.
- Other tasks are more complex, such as recognizing a dog.
  - Basic properties not enough to uniquely classify.
- All in all: we classify objects based on
  - The objects specific properties.
  - Experience and knowledge!

Classification (ML)

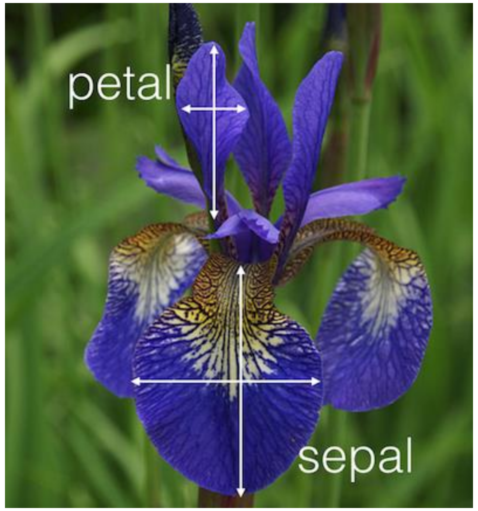
Height	Color	Shape	Target
3 cm	Green	Triangle	

- Same basic concept in ML.
- Representing experience and knowledge?
    - Mathematical models (*classifiers*) that encode experience and knowledge.
    - Tuned by letting them *train* on historical data.
- **Data:** set of measurable properties (*features*) and output (*target*).



- **Main takeaway:** an observations class is determined by it's features.
- This knowledge is fundamental in most ML applications!

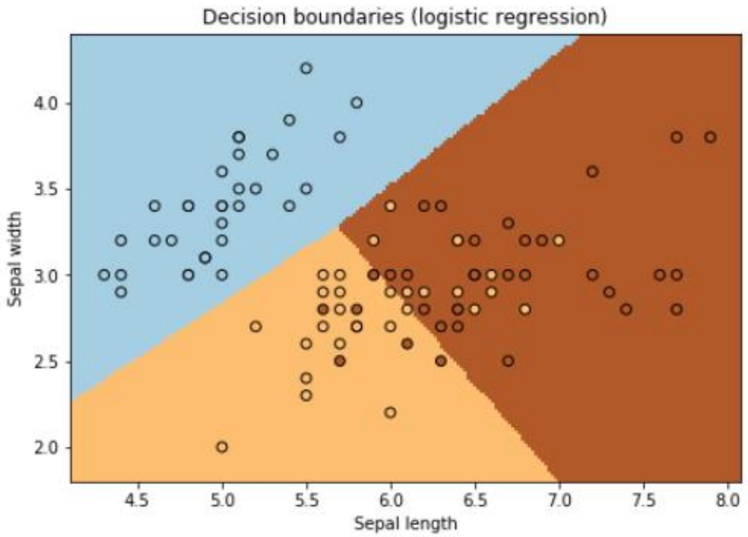
Example: Iris dataset



- Famous simple dataset introduced by Fisher 1936.
- Length, width of petals and sepals of 150 flowers from three Iris subspecies.
- => Classification problem: 4 features and target with 3 classes.

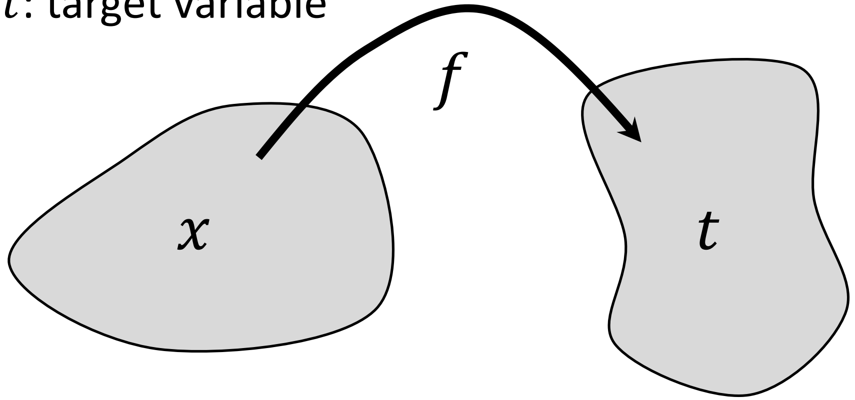
Sepal length (cm)	Sepal width (cm)	Petal length (cm)	Petal width (cm)	Subspecies
5.1	3.5	1.4	0.2	Iris setosa
7.0	3.2	4.7	1.4	Iris versicolor

- Scatter plot and classifier decision boundaries for two features.



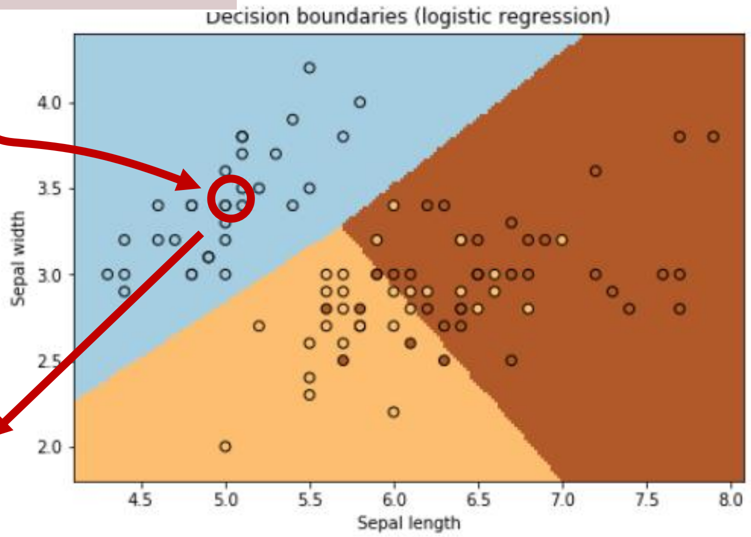
Classification, contd.

- x: feature vector
- t: target variable



- Training classifier = discover mapping function $f: x \to t$.
- Algorithms differ in how f is constructed during training.
    - Often by solving an optimization problem, but not always.
    - Today we focus on a handful of tree-based methods.
- Trained classifier: features as input => class prediction $\hat{t} = f(x)$
- **End goal:** be able to predict class of unseen data based on features.

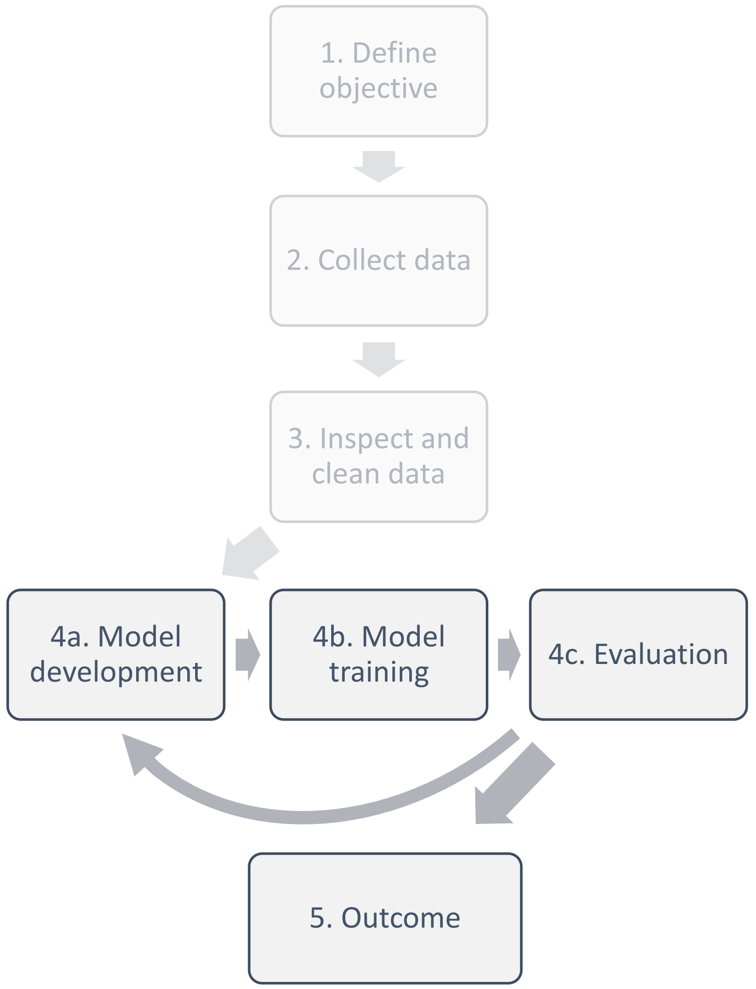
Sepal length (cm)	Sepal width (cm)
5.1	3.5



Subspecies
Iris setosa

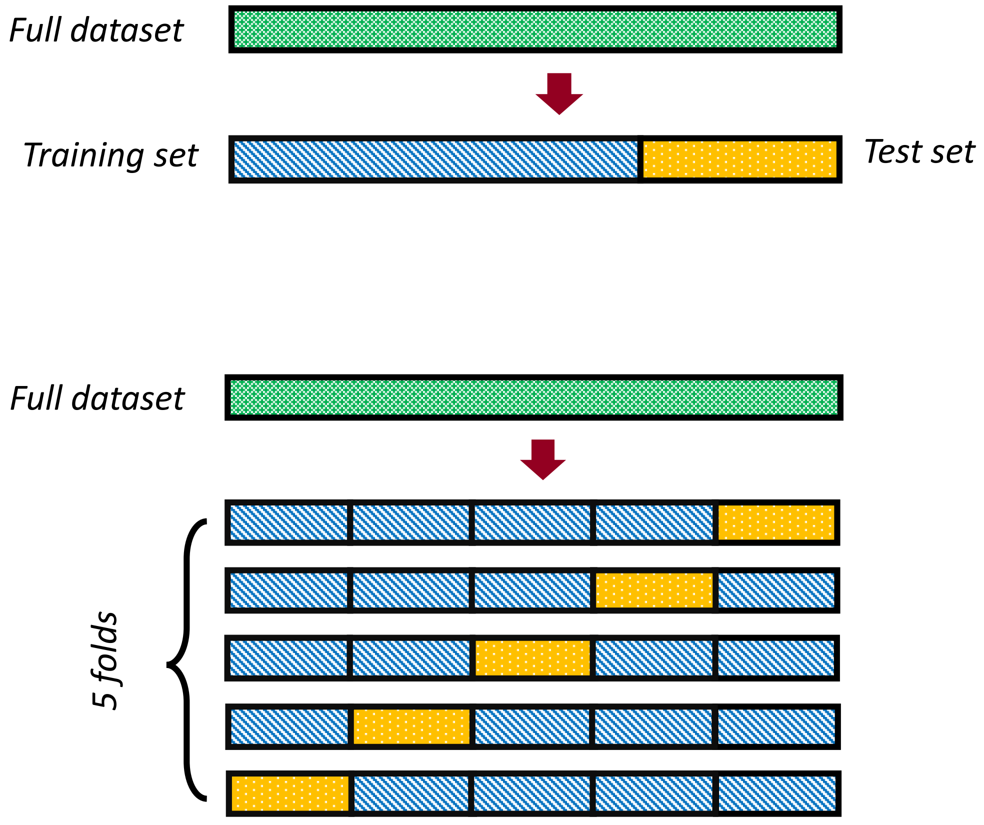
Methodology & workflow

- ML projects typically follow a variant of this workflow.
- Focus of today is on the second half.
- Modeling stage a three step process:
  - **Development**: picking classifier, setting structure.
  - **Training**: fit to historical data.
  - **Evaluation**: performance, cross validation.
- Iterative process!



Cross validation

- Classifiers must generalize to unseen data.
- Holdout method
  - Most basic form of cross validation.
  - Part of data is set aside in a test set and not used in training.
- K-fold cross validation
  - Particularly useful for small datasets
  - Data split in *K* folds. *K-1* folds for training, 1 fold for testing.
  - Repeat train+eval loop *K* times, rotating test folds.



Evaluating performance

- Plenty of performance metrices. Appropriate choice problem dependent!
- Classification accuracy: $\frac{\#\, correct\ classifications}{\#\, classifications}$
  - Obvious choice. Intuitive!
  - Used by Kaggle for todays dataset.
  - What happens if one class occurs rarely, and is very important?
- Other common performance measures (can be useful today!):
  - Precision, recall and F1 score
  - Confusion matrices.

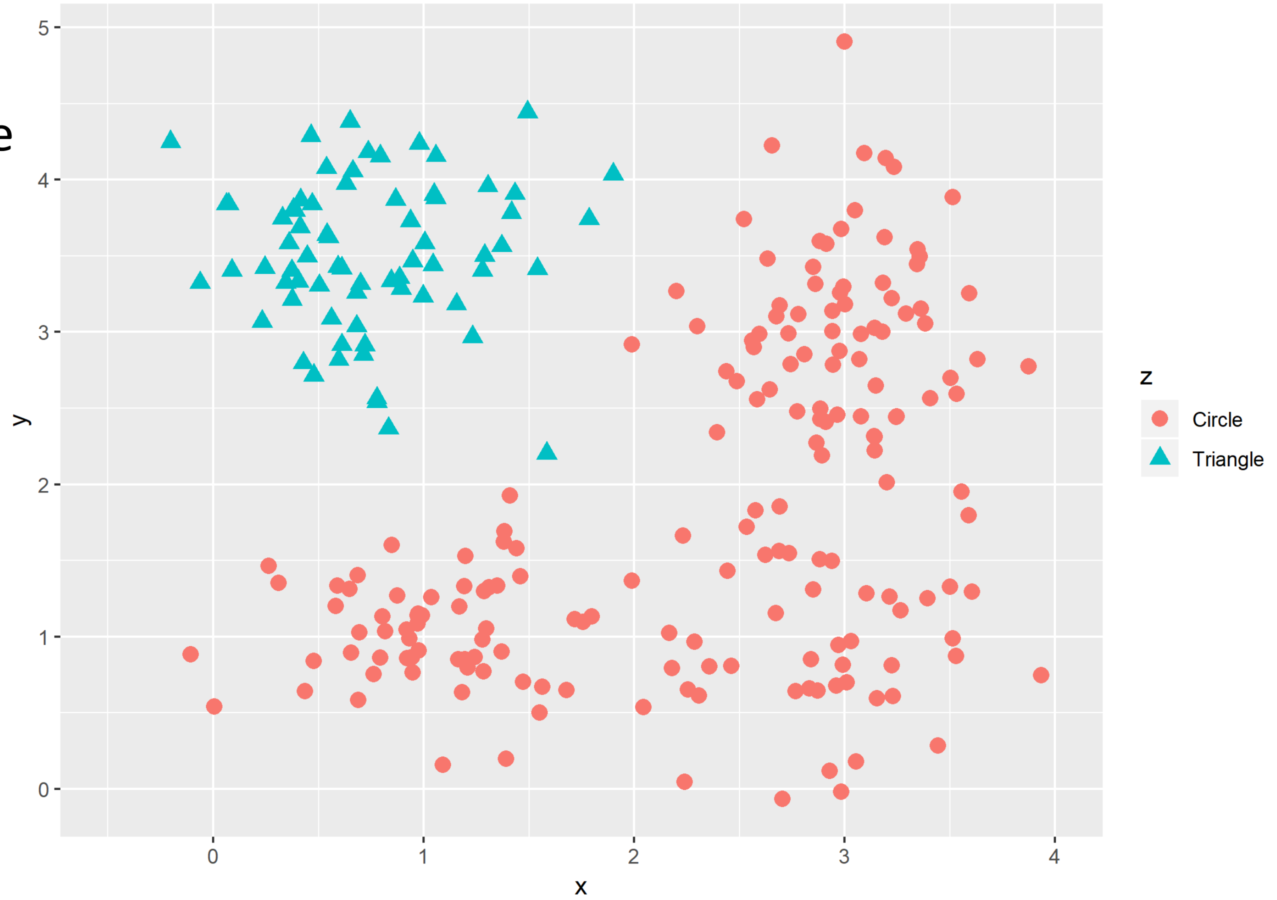
The dataset – Scary monsters

- Made up Kaggle set for practicing classification
  - https://www.kaggle.com/c/ghouls-goblins-and-ghosts-boo
  - Five features (variables):
    - bone_length, rotting_flesh, hair_length, has_soul, color
  - Classify 3 monster types: Ghosts, Goblins, and Ghouls
  - Maximize prediction accuracy
  - Rather small and very tidy set:
    - 371 training observations (rows), 529 test
    - Makes it ideal for practicing different classification methods
    - Not necessarily the most advanced methods that are the best

Decision trees and forests for classification

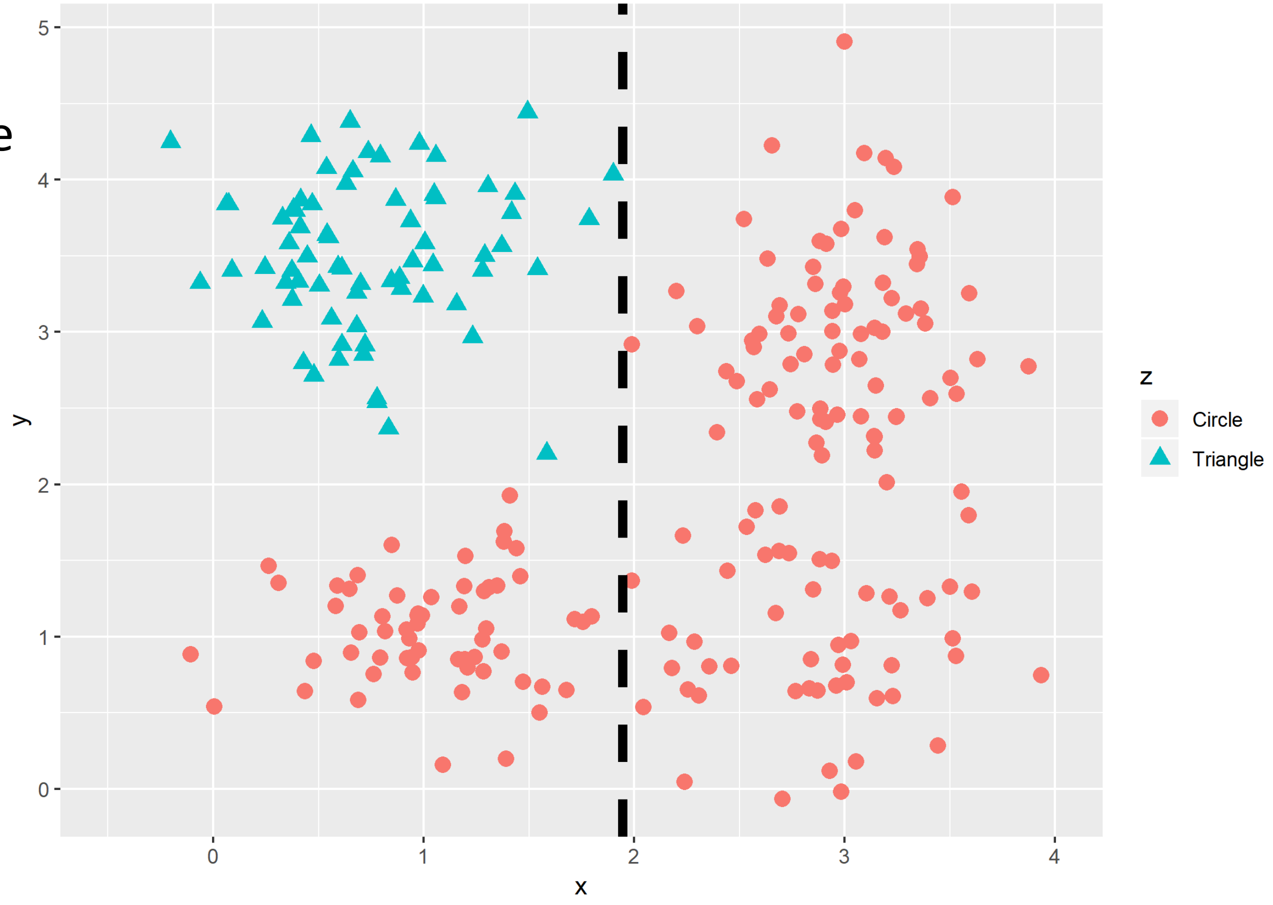
What is a decision tree and how to build it?

- Segment observations into a number of regions in feature space
  - High-dimensional boxes
  - Split previous regions to improve a measure
    - Purity measures, such as accuracy
  - Greedy approach
    - One region at a time
    - Best feature split selected



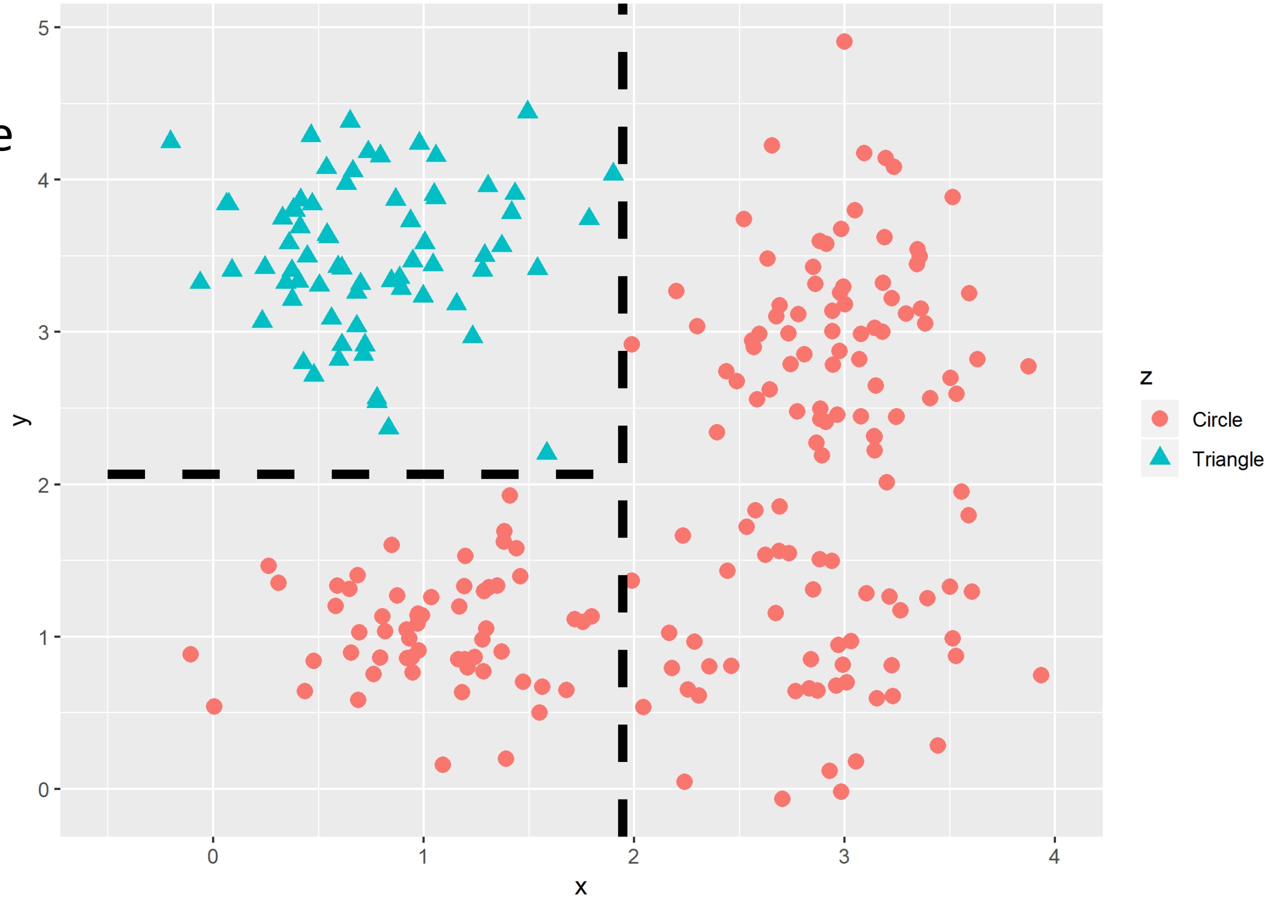
What is a decision tree and how to build it?

- Segment observations into a number of regions in feature space
  - High-dimensional boxes
  - Split previous regions to improve a measure
    - Purity measures, such as accuracy
  - Greedy approach
    - One region at a time
    - Best feature split selected

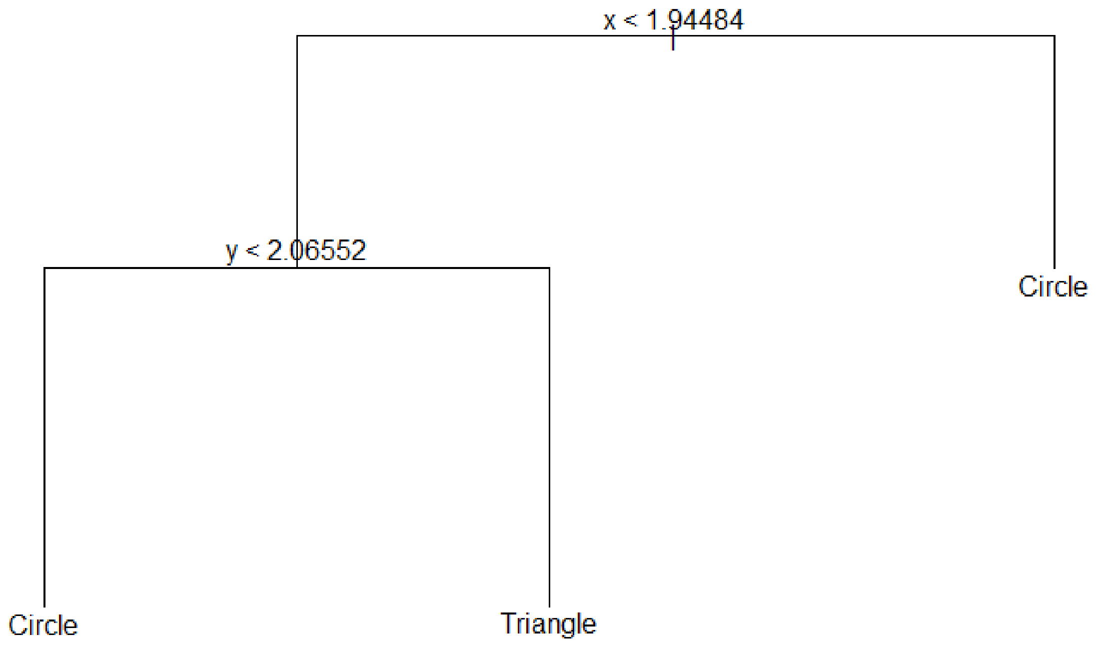


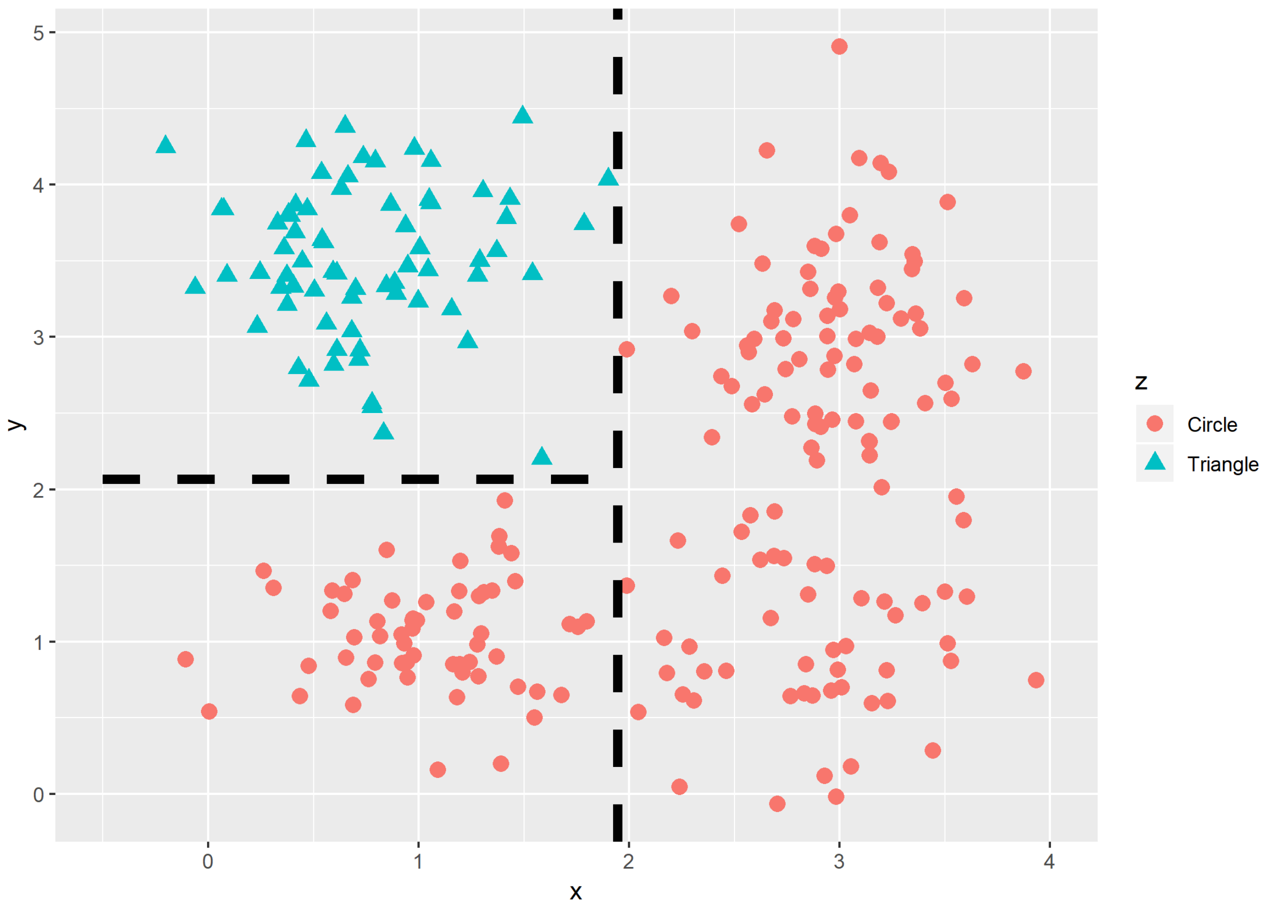
What is a decision tree and how to build it?

- Segment observations into a number of regions in feature space
  - High-dimensional boxes
  - Split previous regions to improve a measure
    - Purity measures, such as accuracy
  - Greedy approach
    - One region at a time
    - Best feature split selected



What is a decision tree and how to build it?





What are the strengths and weaknesses of trees?

Advantages

- + Easily handles mixed data types
- + Handles missing values
- + Robust to outliers
- + Automatic feature (variable) selection
- + Interpretability (small trees)

Disadvantages

- - Inability to extract linear combinations of features
- - Poor prediction accuracy
  - - Big minus

Building forests of decision trees

- Form a powerful committee of weak classifiers
  - Trees in our case
- Grow many trees
  - Each tree votes on class, majority wins
- Keeps most of the advantages of trees
  - Loses some interpretability
- Easy to use immediately on a dataset
  - Quick and Dirty
  - Gives an initial estimate of best performance
- Gains competitive prediction accuracy!
- Very popular methods both in industry and for winning competions

Bootstrap sampling

Training data

Row	x1	x2	x3	y
1				
2				
3				
4				
5				
6				
7				

Randomly sample ***B*** training sets <u>with</u> replacement

Rows in sample

1. 2, 3, 6, 1, 5, 7, 3
2. 7, 7, 5, 5, 3, 7, 5
3. 1, 2, 2, 6, 5, 3, 2

⋮

⋮

B. 2, 7, 1, 1, 5, 6, 6

Bagging (1996) and Random Forest (1999)

Bagging:

- Build *B* trees from *B* bootstrap samples of the training data
  - The trees should be large/bushy with many splits
  - Use all *p* features (variables) in your dataset at every split

Random Forest

- Same procedure as for Bagging, but…
  - At each split, only choose between *m* features
    - Randomly selected at each split
    - Makes the trees less correlated
    - Typically: $m = \sqrt{p}$
- Two tuning parameters, *B* and *m*, easy to apply!

Gradient boosting (2001)

Boosting in general

- General method where additive models are used
- Most often these models are trees

Gradient Boosting

- Iterative process, typically adding small trees
  - Fit a new tree to the residuals of the old model
    - Start with no model (f(x) = 0)
  - Shrink the tree with a factor λ (think step size gradient descent)
  - Add shrunken tree to the previous model and calculate new residuals
- 3 tuning parameters: Number of trees, shrinkage factor, tree depth
  - Many parameters, takes time to tune

Reading material

- An Introduction to Statistical Learning (Book, R, ML)
- Applied Predictive Modeling (Book, R, ML)
- Python Machine Learning (Book, Python, ML)
- The Elements of Statistical Learning (Book, ML)
- Pattern Recognition and Machine Learning (Book, ML)
- Gradient Boosting Machine Learning (T. Hastie video, Boosting)
- http://uc-r.github.io/gbm_regression (Article, Boosting, R)
- http://uc-r.github.io/random_forests (Article, Random Forest, R)

Now, over to:

https://github.com/olofgarpinger/hands_on_data_science_Dec_6_2018

KNIGHTEC