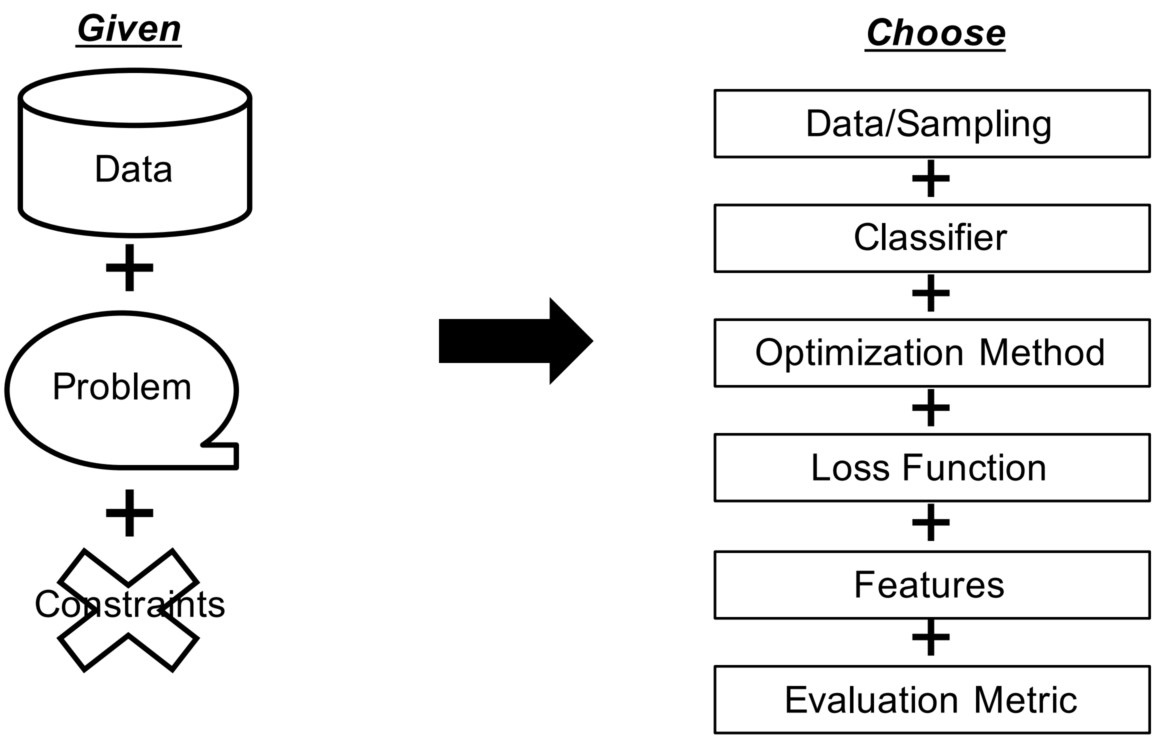
Introduction to Data Science

SOLUTION ENGINEERING

BRIAN D'ALESSANDRO

A COMMON THEME

Few problems have out of the box solutions



The Data Scientist has to navigate these choices

CLASSIFICATION ALGORITHMS

The following is a non-exhaustive list of popular algorithms used in classification problems:

Classic & Simpler Methods

- Decision Tree
- Naïve Bayes
- K- Nearest Neighbors
- Linear Hyperplane

Black Box but Powerful Methods

- Random Forests
- Non-Linear SVM
- Neural Networks

We will NOT discuss each of these algorithms in detail in this course, but we will cover the process of how to choose one.

BUT WHICH ONE SHOULD I USE?

If world free of constraints, then (e.g. a data mining competition):

Try them all, choose best performer

Else:

Consider all constraints on your problem.

Choose best performer subject to constraints

TRY THEM ALL???

Train = Training Data
Val = Validation Data

For each Algorithm in <set of all algorithms>:

Build a classifier, $F^A(X)$ using **Train**

Get out-of-sample error of $F^A(X)$ using **Val**

Choose the Algorithm with the best out-of-sample error.

BAKEOFF RULES

1. Training data must always be disjoint from validation data.

2. Use the same training data and validation data for each hypothesis being tested.

3. Given a tie (statistical or exact), choose the simpler model (sometimes this is subjective).

4. Use this methodology for all design decisions (feature selection, hyper-parameter selection, model selection, etc.)

CONSTRAINTS TO CONSIDER

Do you have the right data?

- Production data is often biased
- The data for your problem might not exist (cold start, new product, etc.)
- The right metric/outcome is unmeasurable (i.e., buys orange juice, or consumer happiness)
- The outcome is so rare you barely observe it

Too little/too much data?

- There are few observations but many variables (generally an estimation problem)
- Too much data (generally a computation problem)

CONSTRAINTS TO CONSIDER

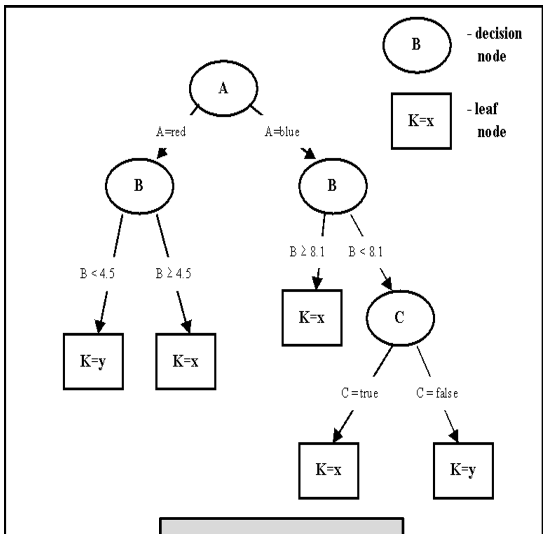
Do you understand the algorithm?

- Your own personal knowledge is a constraint worth admitting to
- You don't have to master every algorithm to be a good data scientist
- Getting the "best-fit" of an algorithm often requires intimate knowledge of said algorithm

CONSTRAINTS TO CONSIDER

Do you need to interpret the model?

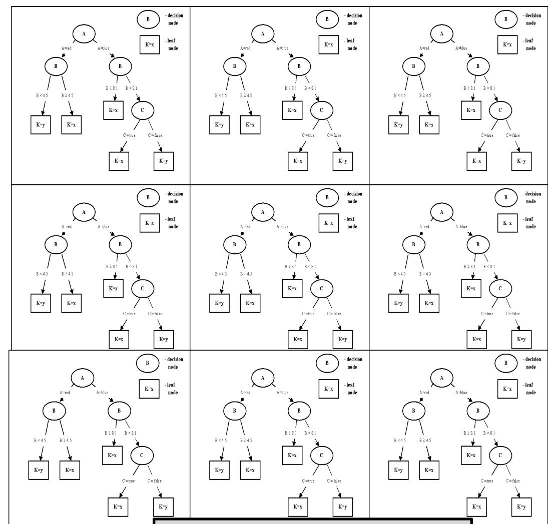
Decision Tree



Very Easy

Vs.

Random Forest

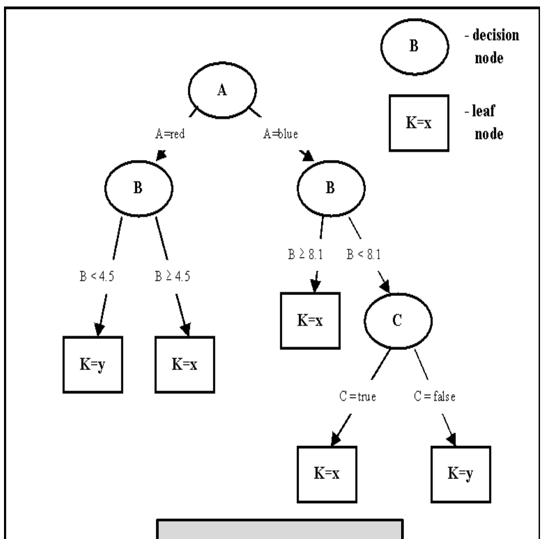


Very Difficult

CONSTRAINTS TO CONSIDER

Does scalability matter (learning time, scoring time, model storage)?

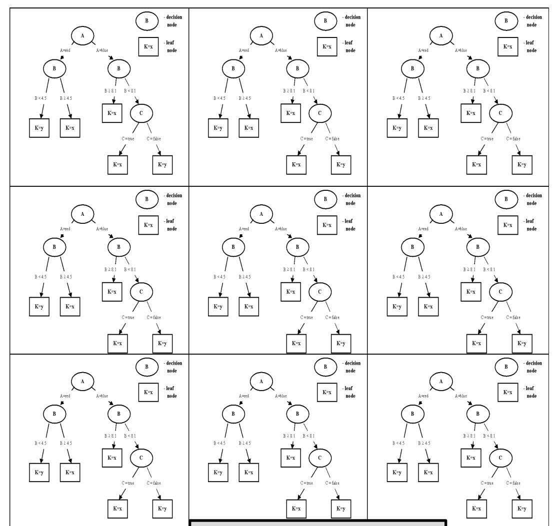
Decision Tree



Fast/Small

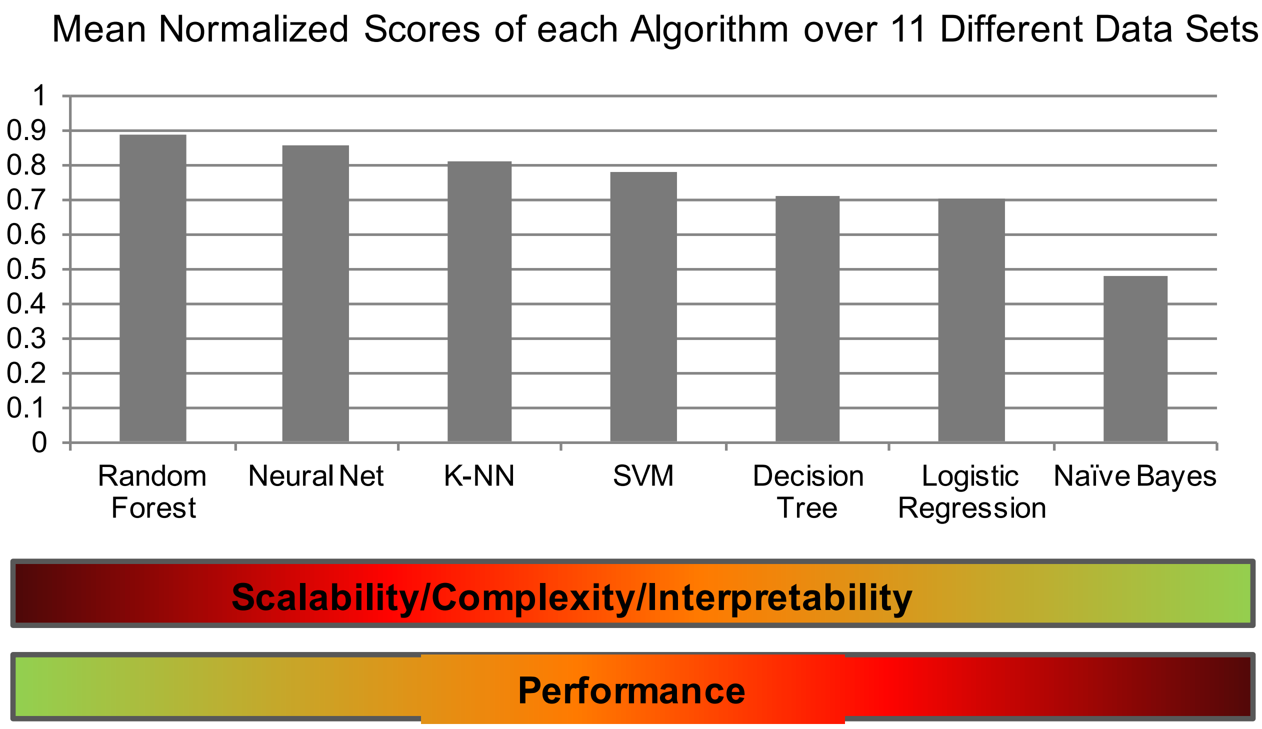
Vs.

Random Forest



Slow/Large

AN EMPIRICAL COMPARISON OF CLASSIFICATION ALGORITHMS



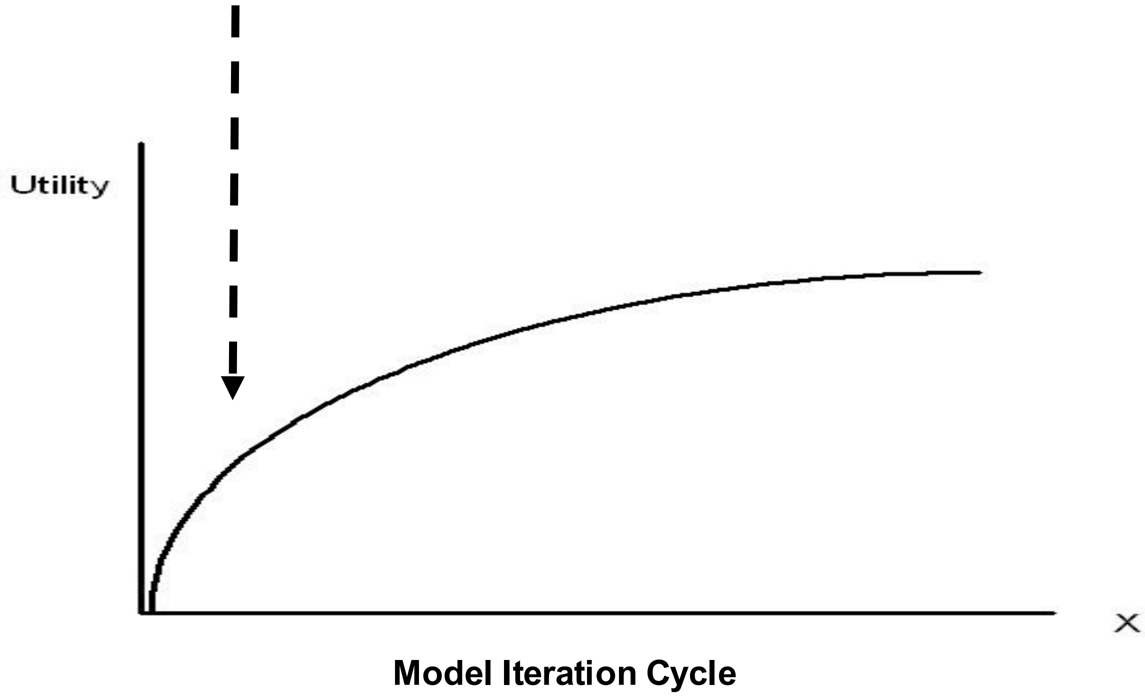
No free lunch: there is no single algorithm that is universally better on all problems (so don't start with Deep Learning on every problem)

Always start simple with a reasonable baseline, and move from there.

Source: An Empirical Comparison of Supervised Learning Algorithms http://www.niculescu-mizil.org/papers/comparison.tr.pdf

ALWAYS BE AGILE: ITERATE

Start with a reasonable baseline model. Should be one with little effort but sophisticated enough to capture signals if they exist.



ALWAYS BE AGILE: ITERATE

Iterate towards better models: in steps 2 – N, try new features and new algorithms. Always start with a good evaluation framework that is grounded in experimental design.