Sponsors:

ansir
innovation
center
startup accelerator

Intro to Random Forests in R

San Diego R Users Group | October 2013

Kevin Davenport
kldavenport.com
kevin.davenport@compliancemetrix.com

CMX
ComplianceMetrix

What is Random Forests?

RF is an classification and regression ensemble learning method that constructs a multitude of decision trees at training time and outputs the class that is the mode of the classes output by individual trees.
It has few parameters to tune and can be used efficiently with defaultsettings (i.e. they are effectively non-parametric). RF are good to use as a first cut when you don't know the underlying model, or when you need to produce a decent model under time pressure.

This ease of use allows people lacking a background in statistics to produce fairly strong predictions free from many common mistakes, with only a small amount of research and programming.

CRAN Task View: Machine Learning & Statistical Learning (7/31/13)
http://cran.r-project.org/web/views/MachineLearning.html

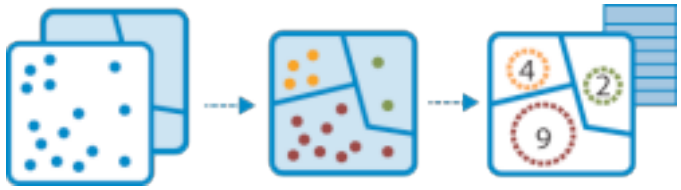
Some Practical Reasons for using it.

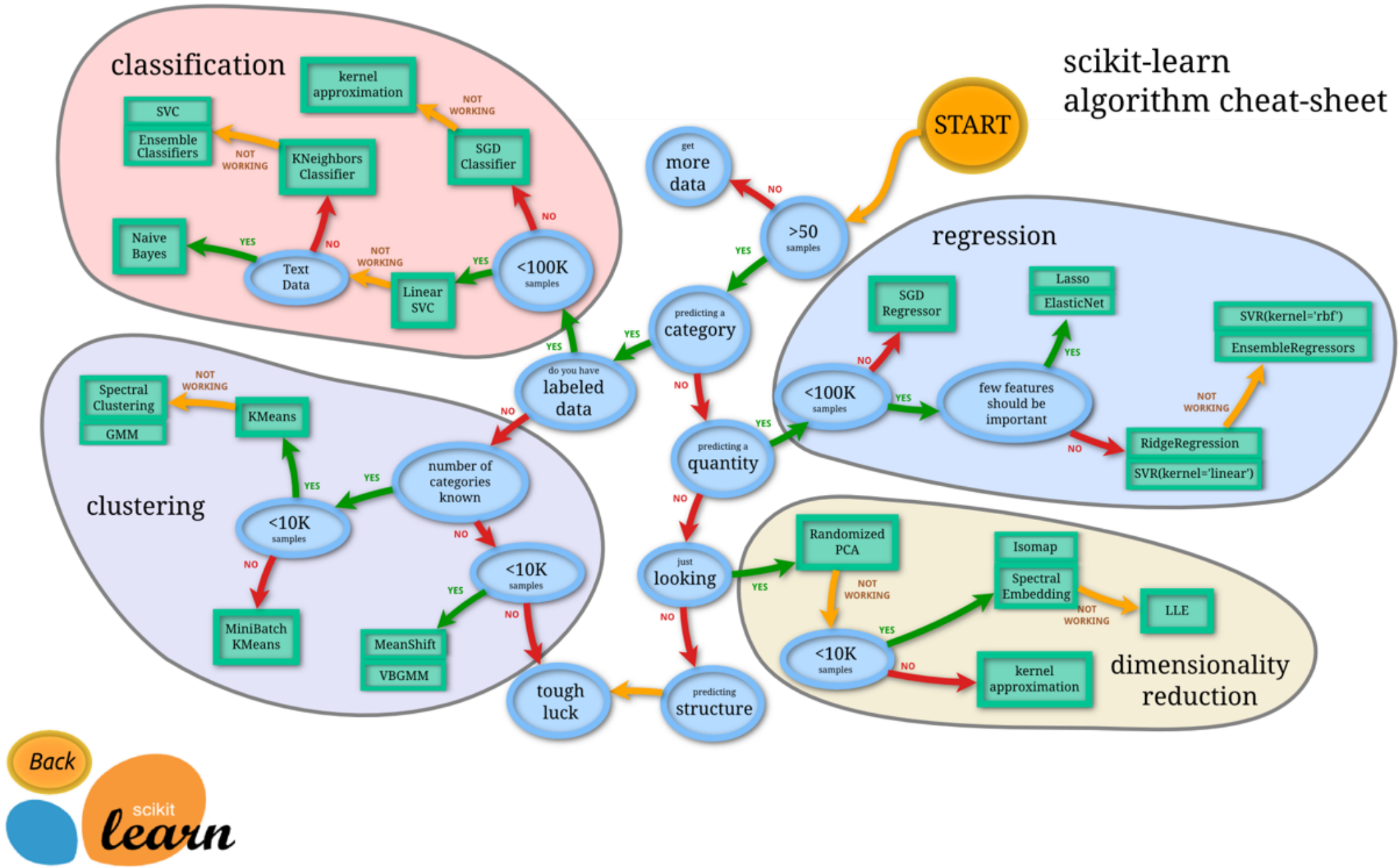
Preprocessing is not needed:
Decision Trees do not care if the data is continuous, discrete, or contains character values. You won't have to scale variables to have a mean of 0 and a standard deviation of 1.

Classification and regression are easy:
The same dataset can be used in times when it's not clear if you want to use regression or classification (boolean output problem).

Easy tuning:
Parameters: mtry & ntree. Address overfitting with extremely randomized trees. Variable importances is easy.

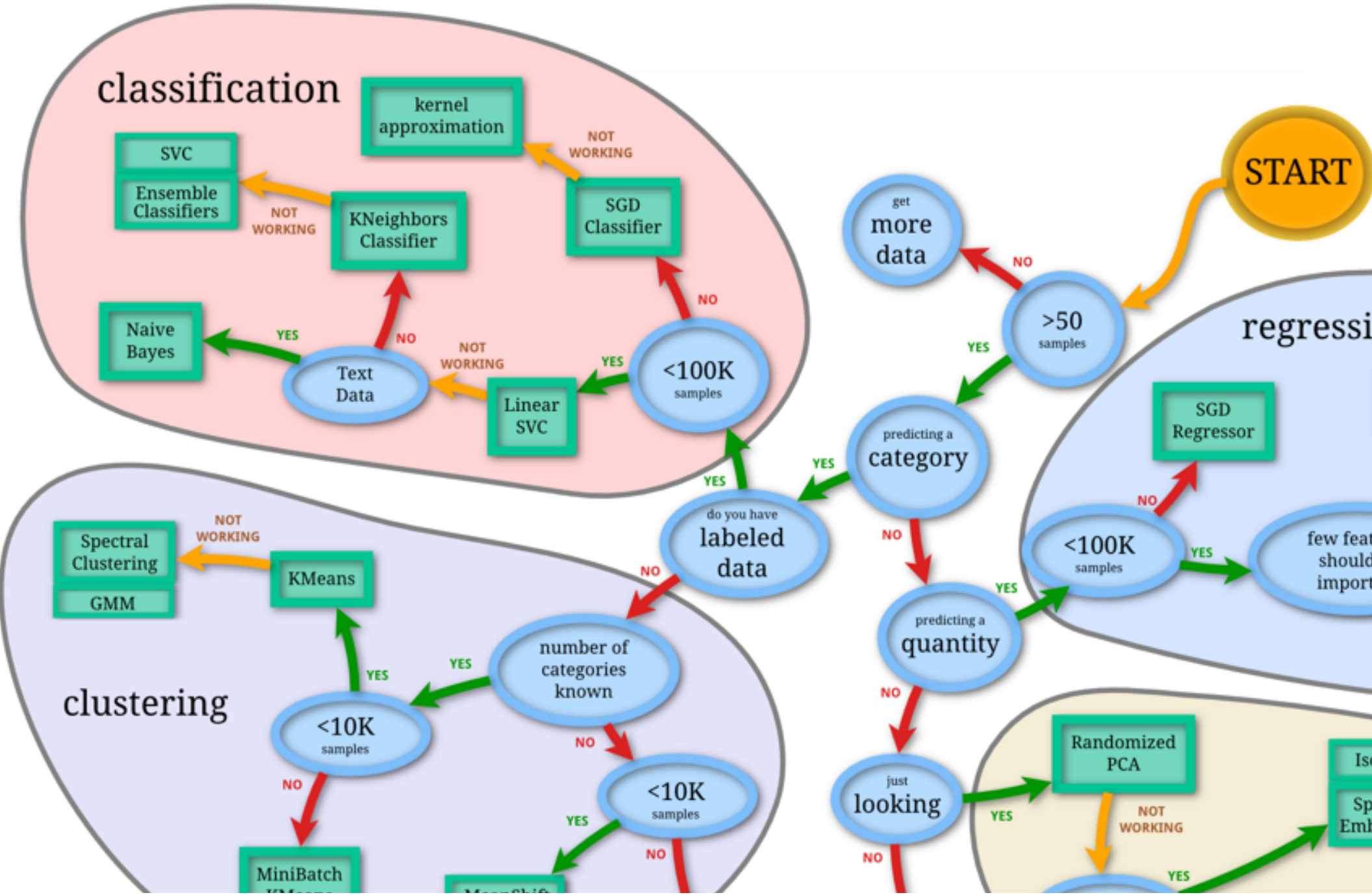
Great Parallelization



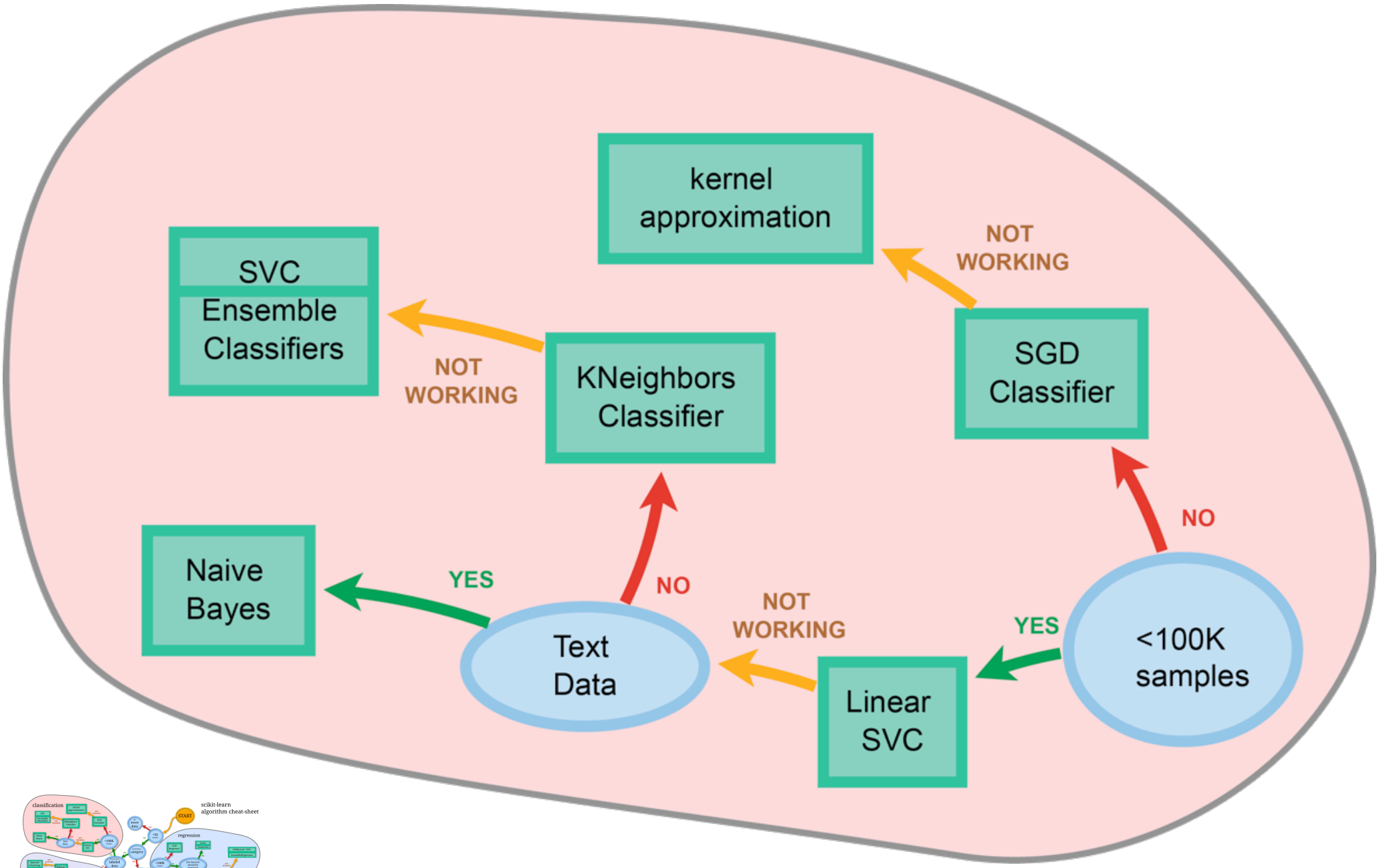


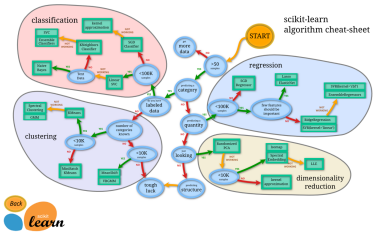
http://peekaboo-vision.blogspot.com/2013/01/machine-learning-cheat-sheet-for-scikit.html

http://peekaboo-vision.blogspot.com/2013/01/machine-learning-cheat-sheet-for-scikit.html

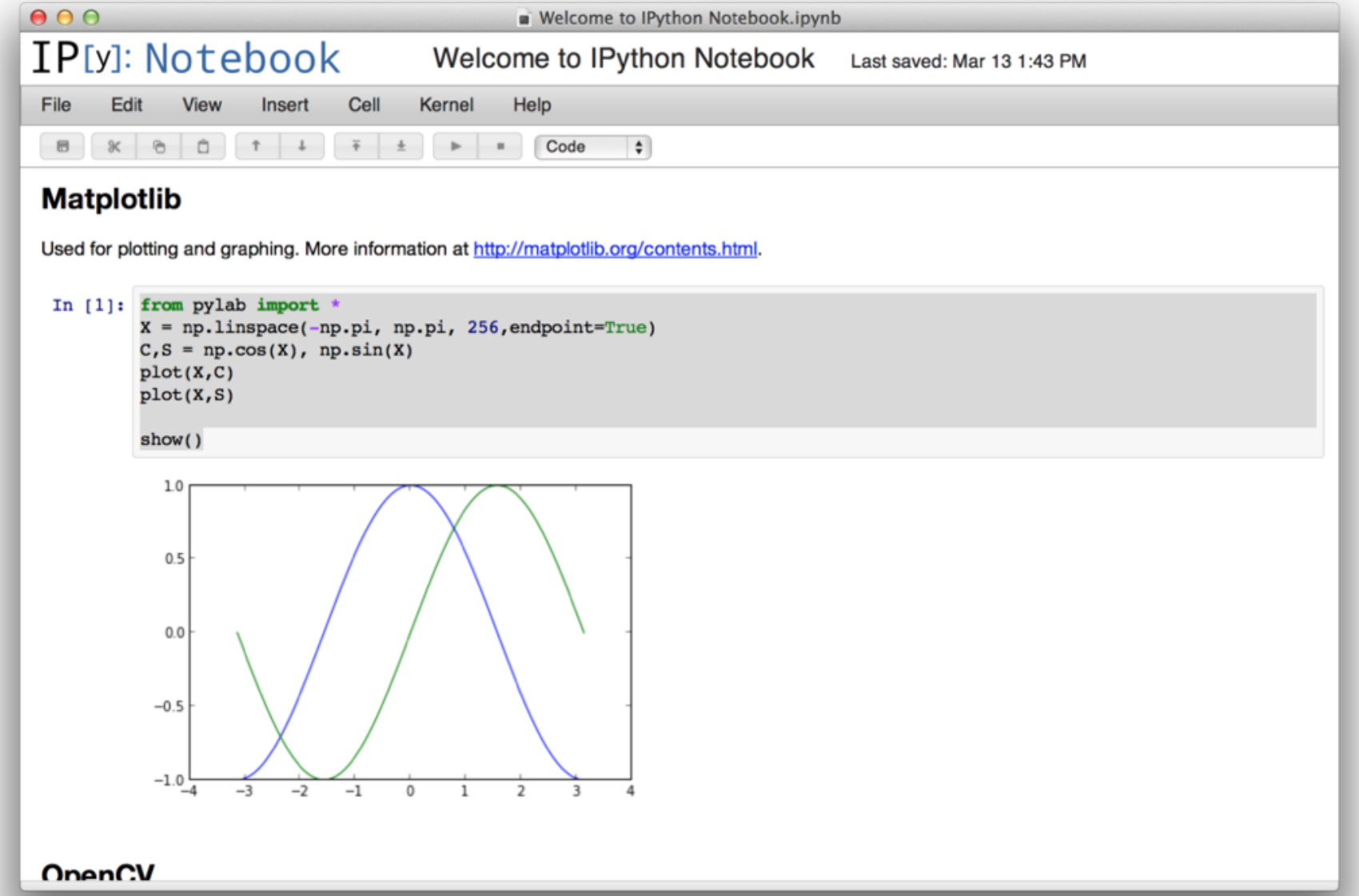


http://peekaboo-vision.blogspot.com/2013/01/machine-learning-cheat-sheet-for-scikit.html





http://ipython.org/notebook.html
http://docs.continuum.io/anaconda/index.html
http://pandas.pydata.org
http://scikit-learn.org/stable/
http://blog.yhathq.com/posts/random-forests-in-python.html



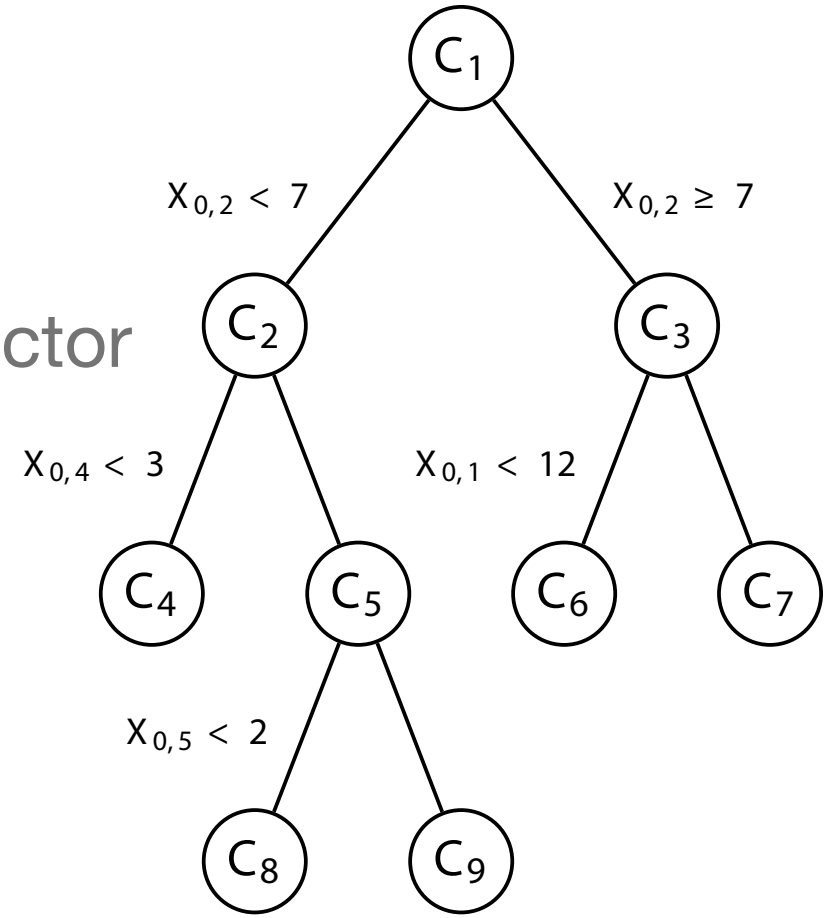
Decision Trees

Can be used for classification or regression

Node splits on any of the attributes

Prune tree to avoid over-fitting

Given **one** data set, get **one** predictor

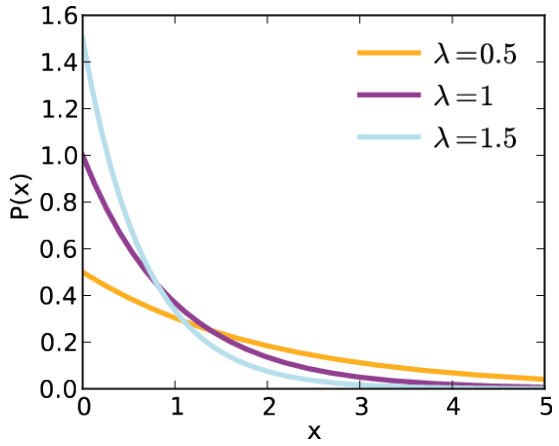


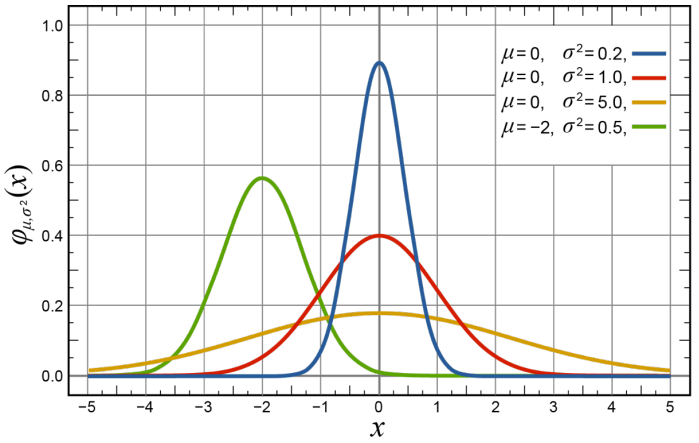
The Strong Law of Large Numbers

The sample average converges almost surely to the expected value

$$\overline{X}_n \xrightarrow{a.s.} \mu \qquad \text{when } n \to \infty.$$

This result explains why random forests do not overfit as more trees are added, but produce a limiting value of the generalization error.





http://www.stat.berkeley.edu/~breiman/random-forests.pdf

Bootstrapping

Tool for inference when the underlying distribution is unknown

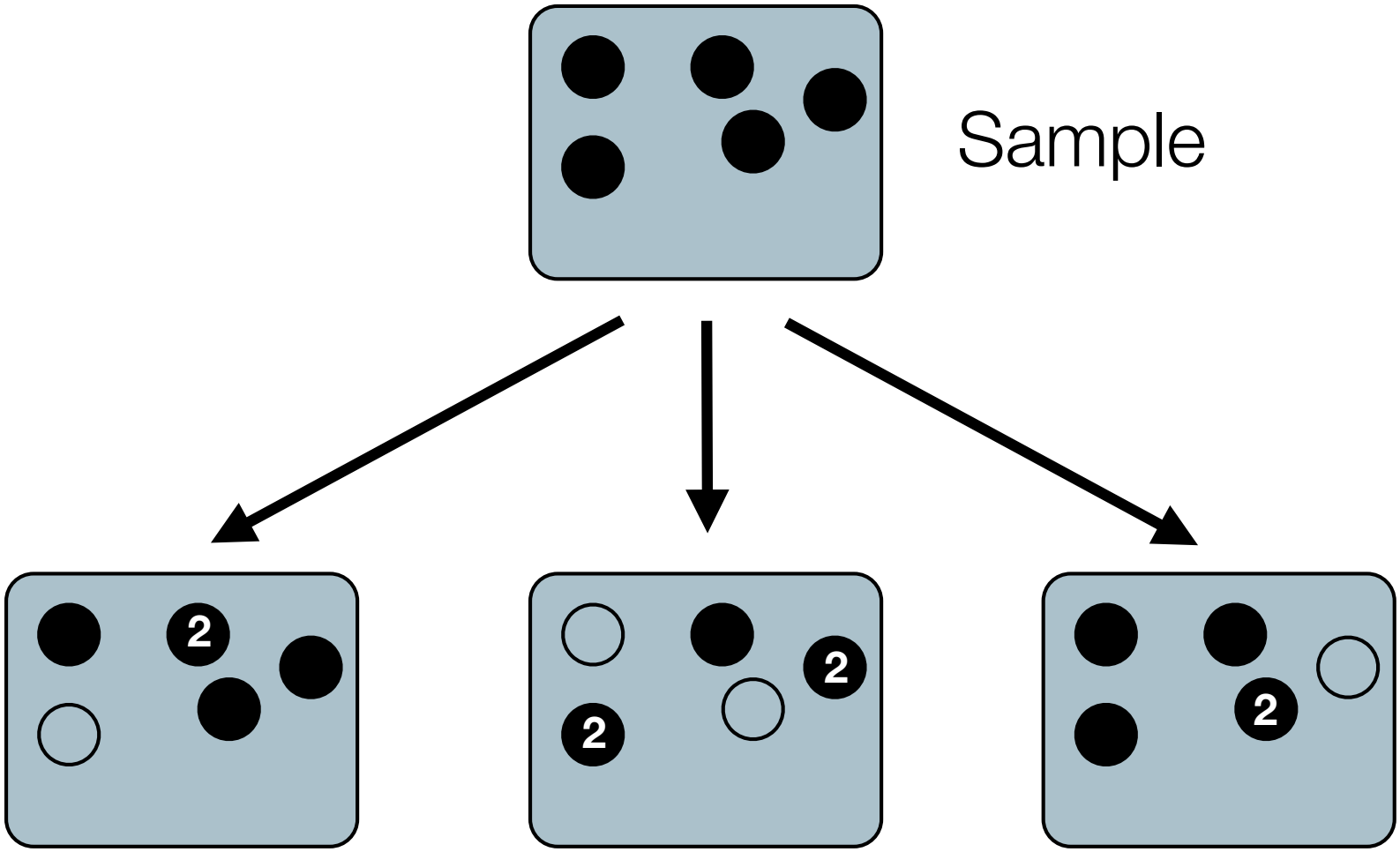
Create replicate data sets from one data set

Sample N times with replacement from a data set of N observations → one replicate

Similar to drawing samples from the same underlying distribution

```
sample(x, size, replace, prob)
library(boot)
boot(data= , statistic= , R=, ...)
```

Bootstrapping



Resample the sample with replacement

Bagging

Bagging = **B**ootstrap **Agg**regat**ing**

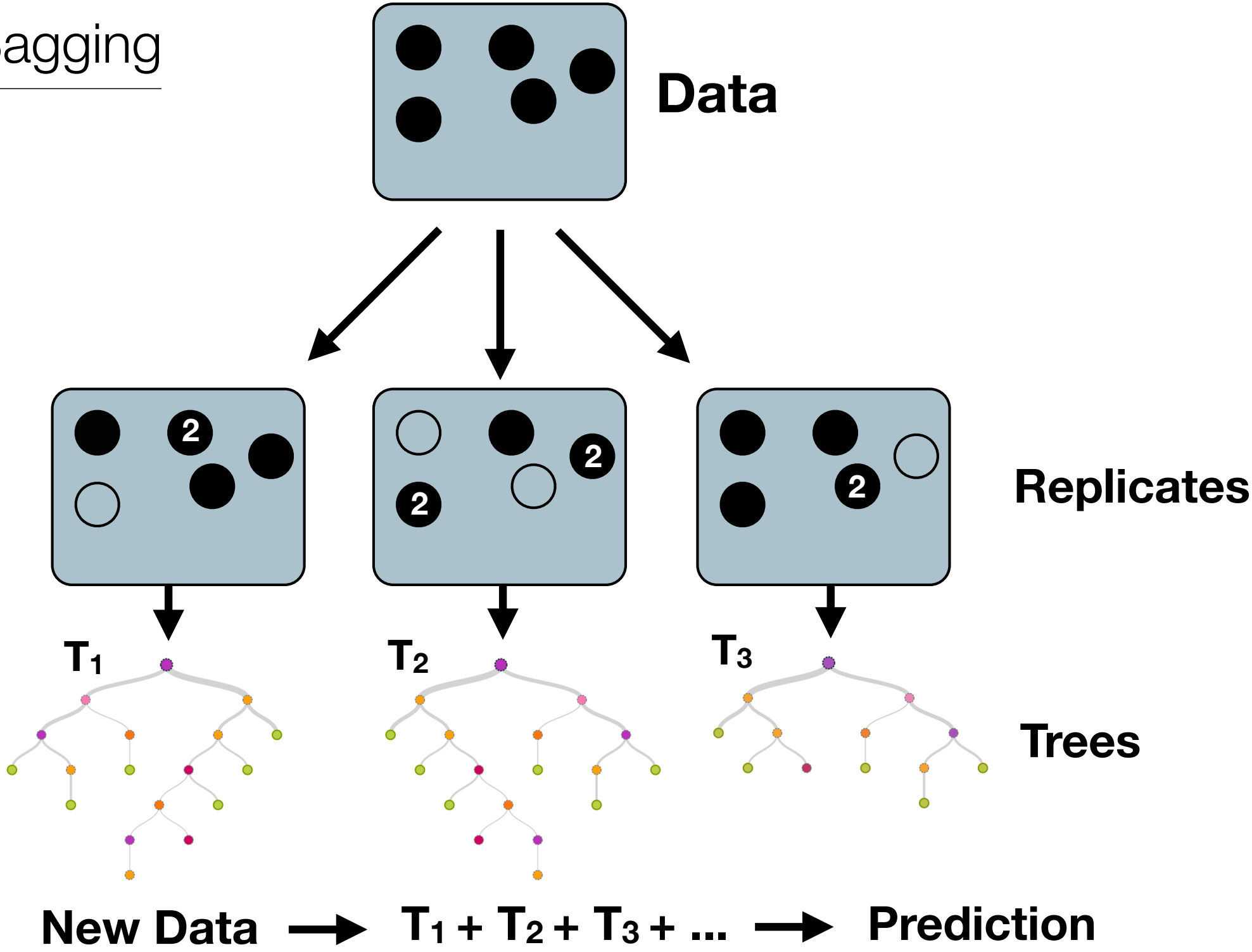
Use bootstrapping to generate a lot of replicate data sets

Grow a decision tree for each replicate

Aggregate tree predictions to get a forest prediction

```
library(ipred)
```

Bagging


Data
2
2
2
2
Replicates
T1
T2
T3
Trees
New Data
T1 + T2 + T3 + ...
Prediction

Ensemble / Aggregate

Average

Weighted average

Vote

Weighted vote

Others...

Bagging Mechanics

Unstable:

ANNs

classification & regression trees

Subset selection (linear regression)

Stable:

kNN

Random Forests

Bagging of trees
\+ Random variable selection
= **Random Forest**

Input: one data set

Bootstrap: many replicates

Train: a decision tree for each replicate
At each node, use one of "mtry" randomly sampled variables

Predict: new data through aggregate

```
library(randomForest)
library(party)
library(randomForestSRC)
```

Tunable Parameters

mtry: randomly selected sample variable for each split

ntree: number of trees (default: 500)

nodesize: minimum size leaf nodes

sampsize: number of observation for each replicate (N)

importance - variable importance (default: False)

```
library(randomForest)
```

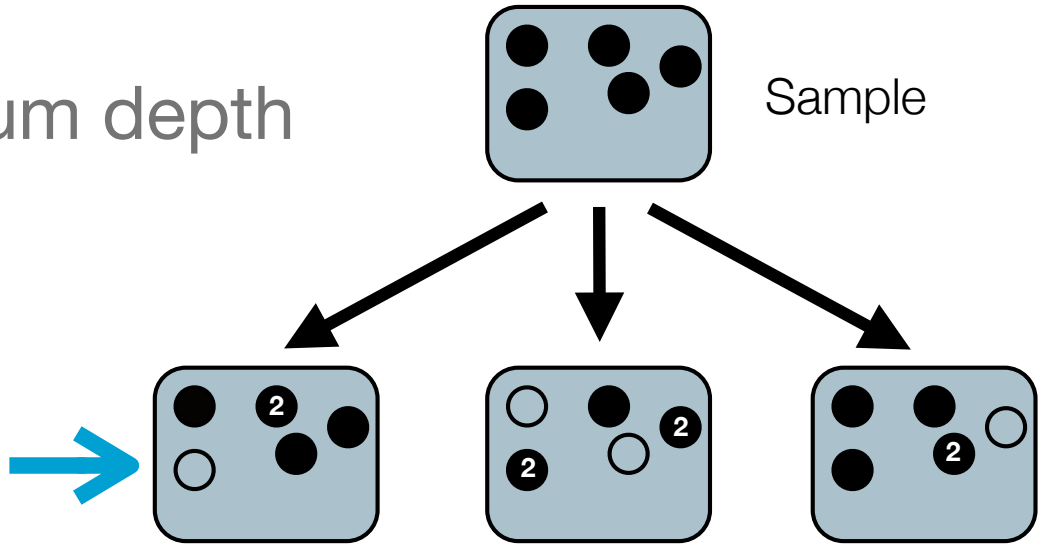
“Out of bag”

For each replicate, data not sampled is "out of bag" (OOB)

Can estimate error rate from OOB data

Proximity measure

Build each tree to maximum depth



ML resources

CRAN Task View: Machine Learning & Statistical Learning (7/31/13)
http://cran.r-project.org/web/views/MachineLearning.html

A comparison of random forest and its Gini importance with standard chemometric methods for the feature selection and classification of spectral data http://www.biomedcentral.com/1471-2105/10/213