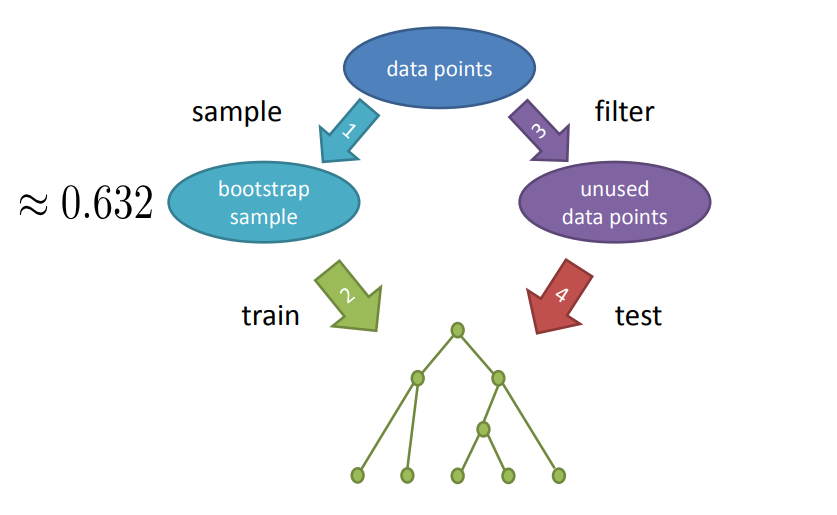
# Random Forest

* All trees are fully grown
* No pruning
* Two parmeters
  + Number of trees
  + Number of features
* Random forests are the bagging idea plus the random subset of features

## Error Rate

* Error depends on
  + Correlation between trees (higher is worse)
  + Strength of single trees (higher is better)
* Increasing number of features for each split:
  + Increases correlation
  + Increases strength of single trees

## Out of Bag Error



* Very similar to cross-validation
  + But you should still use cross-validation regardless of any tool that you use.
* Measured during training
* Can be too optimistic

## Variable Importance

* Use out of bag samples
* Predict class for these samples
* Randomly permute values of one feature
* Predict classes again
* Measure decrease in accuracy
* Measure split criterion improvement
* Record improvements for each feature
* Accumulate over whole ensemble

### Gini vs Randomization

* Normally go with randomization but try both.

# Regression

* What is the difference between regression and classification?
  + Regression preserves the distance between errors, classification doesn’t
  + Classification doesn’t care if 1 star is closer to 2 star when it’s really 3 stars, it’s still a classification error

## KNN for Regression

* Average the values of the K nearest neighbors
* Or build a weighted average of the K nearest neighbors
* There is aKNeighborsRegressor()

## Regression Tree

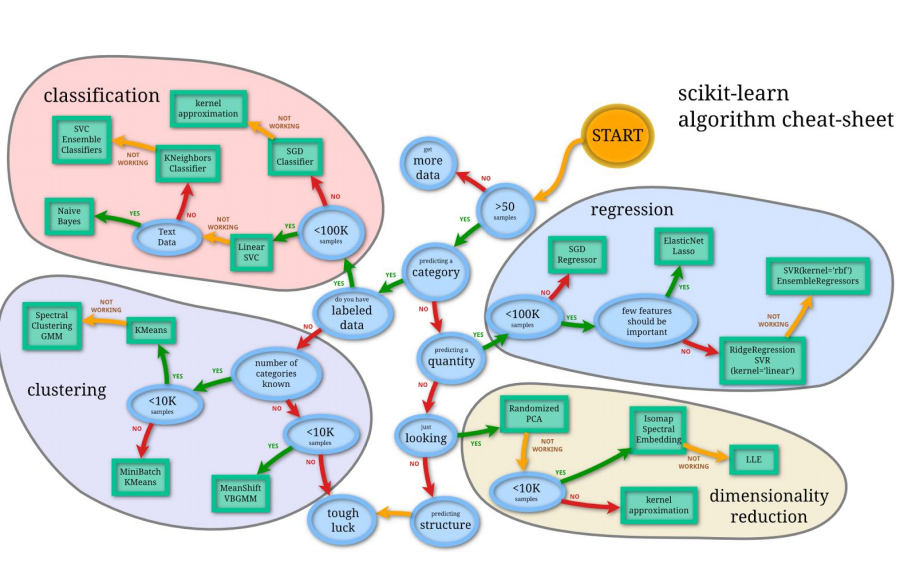
* We average, this time over all points in one of the cells
* During training, split in the way that reduces the squared error the most

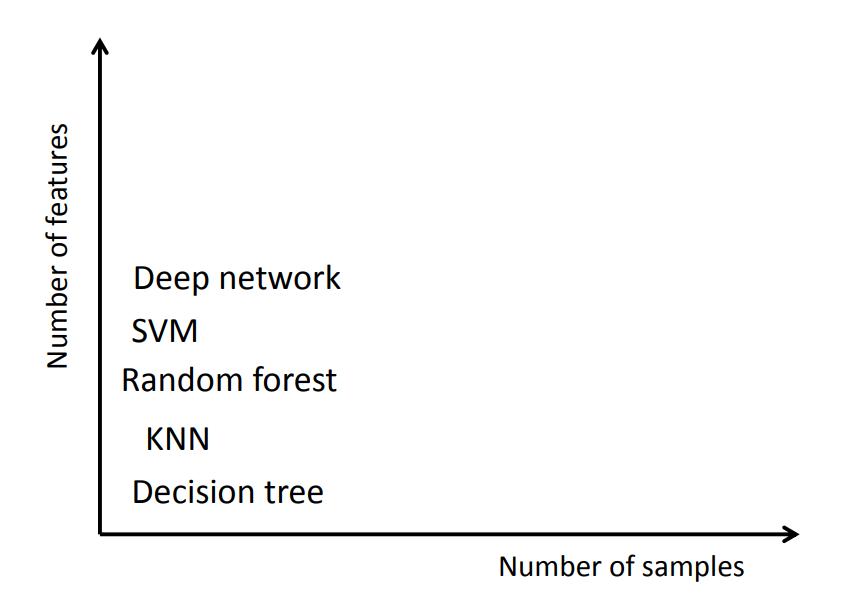
## Random Forest for Regression

* Same idea as before
* Train multiple trees in parallel and average
* Different defaults
* Max\_features = n\_features
* Uses Square error instead of gini

# General Best Practices

* It will be harder than it looks
* Know your application:
  + Zero values
  + Outliers
  + Where do labels come from
    - Gold standard is to get consensus from at least 5 people
* Document, document, document
* Commit, pull, push, repeat





* Deep network requires lots of features and samples
* KNN requires low # of features and samples
* SVM requires lots of features and the number of samples lower than the features
* Random forest requires lots of samples, but medium amount of features

## Cross Validation

* What if your hyper-parameters are all over the place?
  + Then your folds aren’t big enough. You would have to increase the amount of folds.
* What if your hyper-parameters are at the border of your grid search window?
  + Widen your border and run it again.