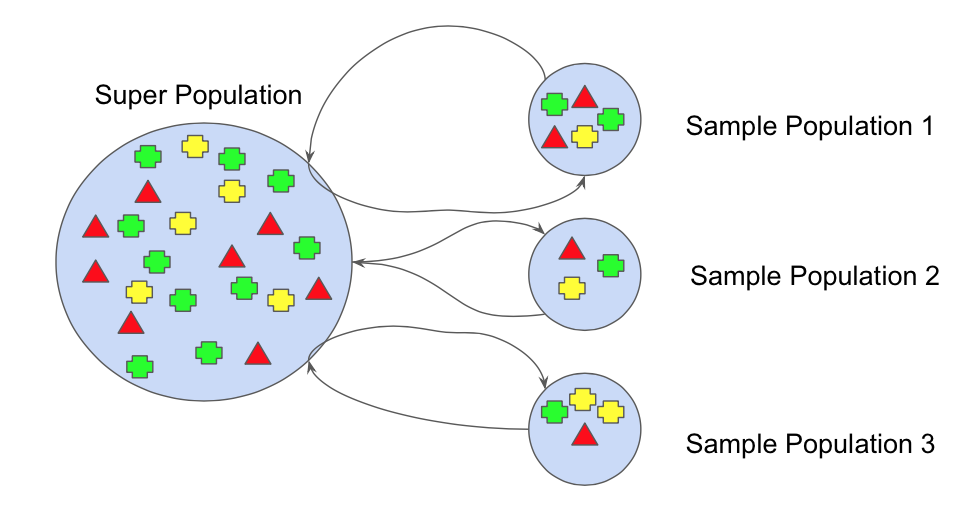
Biggest problem with building a decision tree is high variance.

Solution is ensembling: We can use multiple trees to get more accurate predictions and lower the variance

**Ensemble 1 - Bagging – Bootstrap Aggregation**

* Step 1bootstrap the data, create data set 1 > build dec tree 1
* Step 2, BS data set 2 > dec tree 2
* ….
* Step n, BS data set n > dec tree n

Aggregate all the predictions from all bootstrapped trees.

Regression: average all the predictions (mean)

Classification: majority vote.(mode)

OOB Out of Bag observations/error

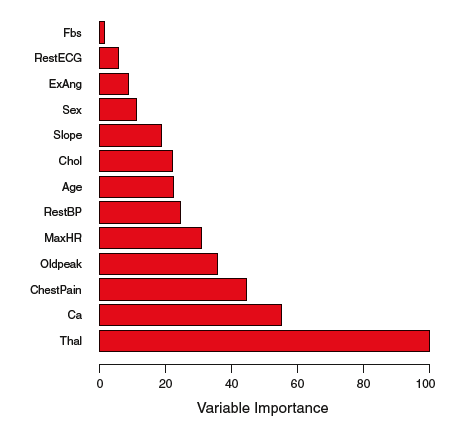
Bootstrap use 2/3

1/3 for test set. .

Here, we grow the trees fully. Don’t limit max\_depth

Interpretation of the features is lost because we have many trees different trees combining to give the predicition

SO, we use **feature importance**



Problem w bagging. Correlated trees: important variable, is present in almost all the trees and at the top.

Solution is Random Forest.

Random forest is very similar to bagging

Difference is in how we make out splits (which features we consider)

Every time we make a split, we take a random sample of features.

100 predictors. Sqrt =10 predctors

**Boosting**

XGBoost example package

Still ensemble of trees.

Difference is trees are built sequential. Residual output of the first tree is input to the next tree.

Short trees, stump (1 split)

Slow learning. Progressing. Parameter lambda (learning rate. Typically .01 or 0.001)

Dependent on each other.

Difficult to use on Distributed cluster system. RF are better for that.

