Random Forests

Data Sets

Attrition

```r
attrition <- attrition %>% mutate_if(is.ordered, factor, order = F)
attrition.h2o <- as.h2o(attrition)

churn <- initial_split(attrition, prop = .7, strata = "Attrition")
churn.train <- training(churn)
churn.test <- testing(churn)
```

Ames, Iowa housing data.

```r
set.seed(123)

ames <- AmesHousing::make_ames()
ames.h2o <- as.h2o(ames)

ames.split <- initial_split(ames, prop =.7, strata = "Sale_Price")

ames.train <- training(ames.split)
ames.test <- testing(ames.split)
```

Random Forest Overview

Random forests are modifications of bagged decision trees that build a large collection of *de-correlated* trees to further improve the predictive performance.

Extended Bagging

The bootstrap aggregation procedure (bagging) has a limited effect on the variance reduction of decision trees.

Random forests help reduce tree correlation by injecting more randomness into the tree-growing process. More specifically, while growing a decision tree during the bagging process, random forests perform split-variable randomization where each time a split is to be peformed, the search for the split variable is limited to a random subset of m_{try} of the original p features.

Typical default: $m_{try} = \frac{p}{3}$ (regression) and $m_{try} = \sqrt{p}$ for classification.

Basic algorithm is as follows::

- 1.) Given a training data set
- 2.) Select number of trees to build (n_trees)

- 3.) for i = 1 to n_trees do:

+ 4.) Generate a bootstrap sample of the original data

```
# clean up
rm(list = ls())
```