Decision Trees Lab 2: Ensembles

4/17/23

```
# Helper packages
library(dplyr)       # for data wrangling
library(ggplot2)     # for awesome plotting
library(doParallel)  # for parallel backend to foreach
library(foreach)     # for parallel processing with for loops

# Modeling packages
library(caret)       # for general model fitting
library(rpart)       # for fitting decision trees
library(ipred)       # for fitting bagged decision trees
```

Bagging trees

The first example is adapted from (**Boehmke2020?**), also available online.

This exercise relies on the well-known `AmesHousing` dataset on hose prices in Ames, IA.

```
if(!require(AmesHousing))
  install.packages("AmesHousing", dep=TRUE)
ames <- AmesHousing::make_ames()
```

```
if(!require(rsample))
  install.packages("rsample", dep=TRUE)
# Stratified sampling with the rsample package
set.seed(123)
split <- rsample::initial_split(ames, prop = 0.7,
                        strata = "Sale_Price")
ames_train  <- training(split)
ames_test   <- testing(split)
```

Building a decision trees to predict the sales price for the Ames housing data yields a poor perfrmance classifier/predictor that is beated by alternatives such as MARS or KNN (check it!)

In this example, rather than use a single pruned decision tree, we can use, say, 100 bagged unpruned trees (by not pruning the trees we're keeping bias low and variance high which is when bagging will have the biggest effect).

As the below code chunk illustrates, we gain significant improvement over our individual (pruned) decision tree (RMSE of 26,462 for bagged trees vs. 41,019 for the single decision tree).

The `bagging()` function comes from the `ipred` package and we use `nbagg` to control how many iterations to include in the bagged model and `coob=TRUE` indicates to use the OOB error rate.

- By default, bagging() uses rpart::rpart() for decision tree base learners but other base learners are available.
- Since bagging just aggregates a base learner, we can tune the base learner parameters as normal.
- Here, we pass parameters to rpart() with the control parameter and we build deep trees (no pruning) that require just two observations in a node to split.

```
# make bootstrapping reproducible
set.seed(123)

# train bagged model
ames_bag1 <- bagging(
```

```
  formula = Sale_Price ~ .,
  data = ames_train,
  nbagg = 100,
  coob = TRUE,
  control = rpart.control(minsplit = 2, cp = 0)
)

show(ames_bag1)
```

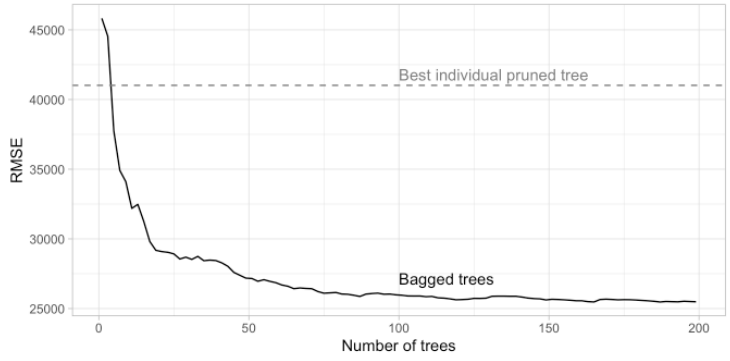
```
Bagging regression trees with 100 bootstrap replications

Call: bagging.data.frame(formula = Sale_Price ~ ., data = ames_train,
    nbagg = 100, coob = TRUE, control = rpart.control(minsplit = 2,
        cp = 0))

Out-of-bag estimate of root mean squared error:  26216.47
```

Bagging tends to improve quickly as the number of resampled trees increases, and then it reaches a platform.

```
knitr::include_graphics("images/baggingRSME.png")
```



References