Interpreting Black-Box Models

```r
library(ISLR)
library(caret)
library(ranger)
library(plotmo)
library(pdp)
library(lime)
```

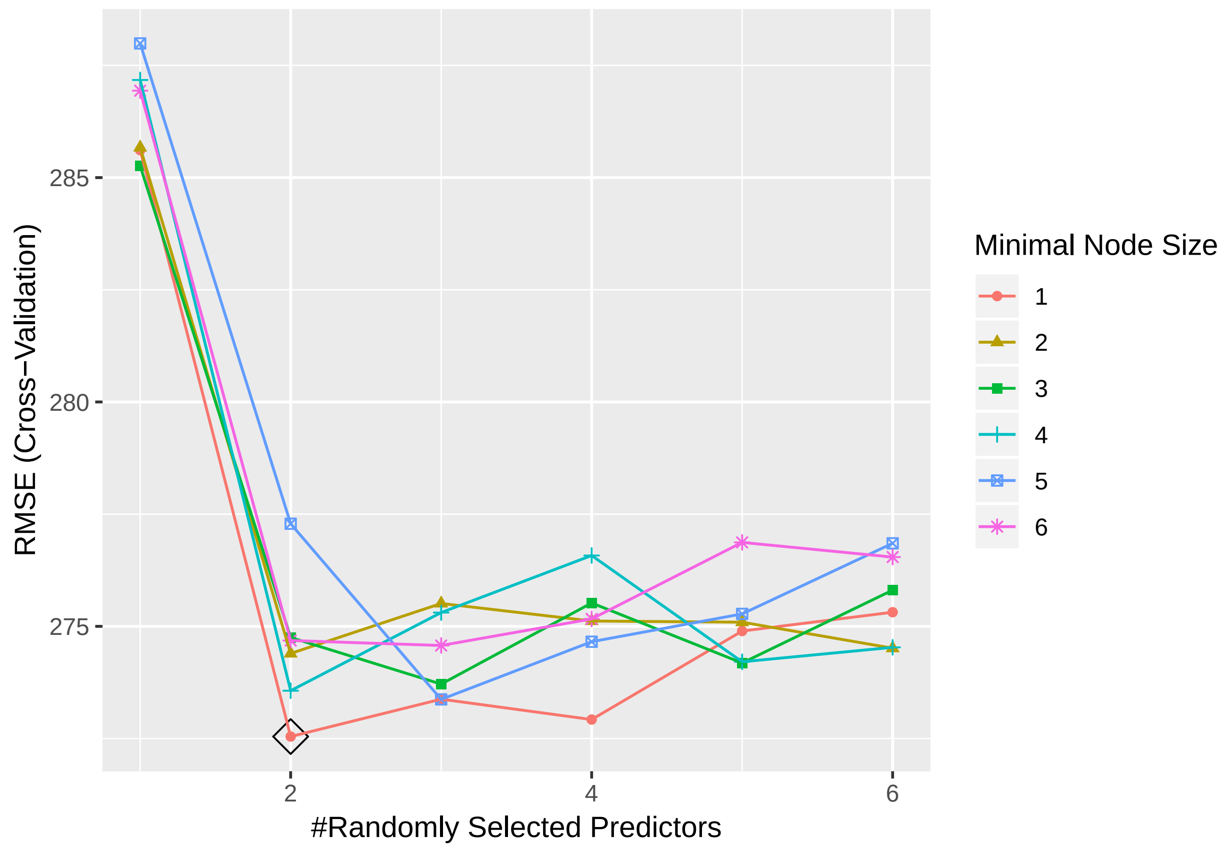
Predict a baseball player's salary on the basis of various statistics associated with performance in the previous year. Use `?Hitters` for more details.

```r
data(Hitters)
Hitters2 <- Hitters[is.na(Hitters$Salary),] # players with missing outcome
Hitters <- na.omit(Hitters)
```

Building a random forest using `caret`

We use the fast implementation of random forest when tuning the model.

```r
set.seed(1)
ctrl <- trainControl(method = "cv")

# Try more if possible
rf.grid <- expand.grid(mtry = 1:6,
                       splitrule = "variance",
                       min.node.size = 1:6)

rf.fit <- train(Salary~., Hitters,
                method = "ranger",
                tuneGrid = rf.grid,
                trControl = ctrl)

ggplot(rf.fit, highlight = TRUE)
```

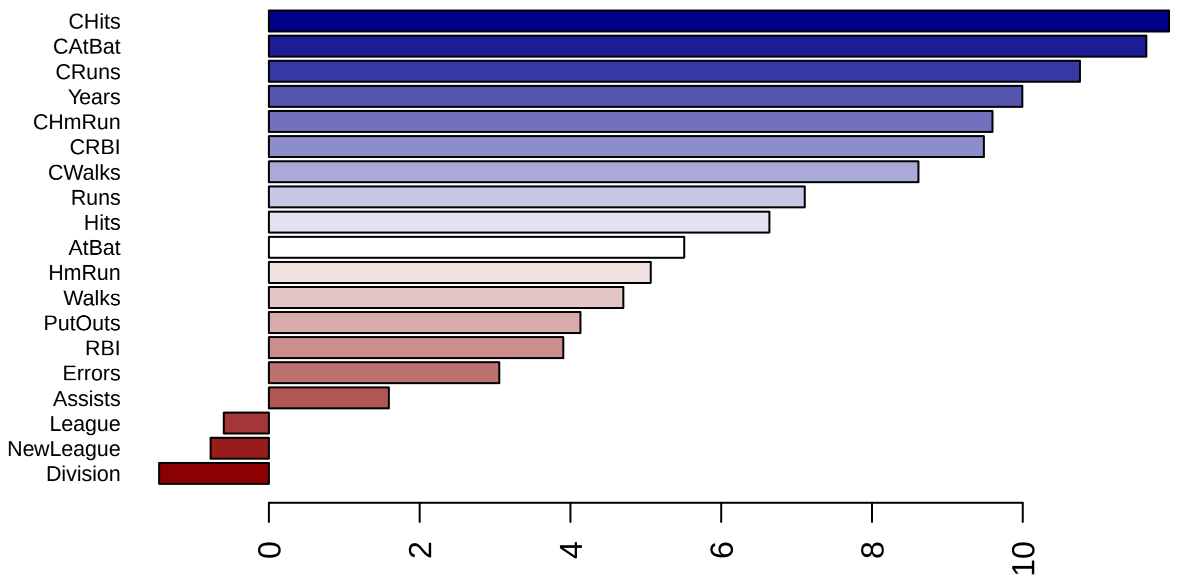


Explain the black-box models

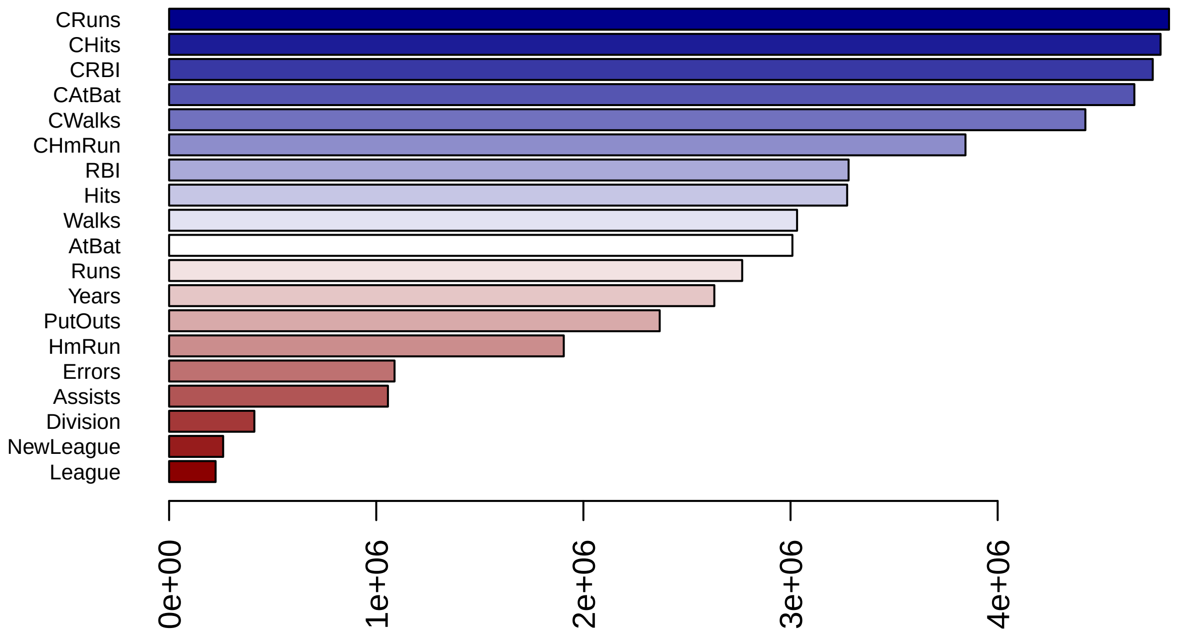
Variable importance

We can extract the variable importance from the fitted models. In what follows, the first measure is computed from permuting OOB data. The second measure is the total decrease in node impurities from splitting on the variable, averaged over all trees. For regression, node impurity is measured by residual sum of squares.

```r
set.seed(1)
rf.final.per <- ranger(Salary~., Hitters,
                       mtry = rf.fit$bestTune$mtry, splitrule = "variance",
                       min.node.size = rf.fit$bestTune$min.node.size,
                       importance = "permutation",
                       scale.permutation.importance = TRUE)

barplot(sort(ranger::importance(rf.final.per), decreasing = FALSE),
        las = 2, horiz = TRUE, cex.names = 0.7,
        col = colorRampPalette(colors = c("darkred","white","darkblue"))(19))
```



```r
set.seed(1)
rf.final.imp <- ranger(Salary~., Hitters,
                       mtry = rf.fit$bestTune$mtry, splitrule = "variance",
                       min.node.size = rf.fit$bestTune$min.node.size,
                       importance = "impurity")

barplot(sort(ranger::importance(rf.final.imp), decreasing = FALSE),
        las = 2, horiz = TRUE, cex.names = 0.7,
        col = colorRampPalette(colors = c("darkred","white","darkblue"))(19))
```

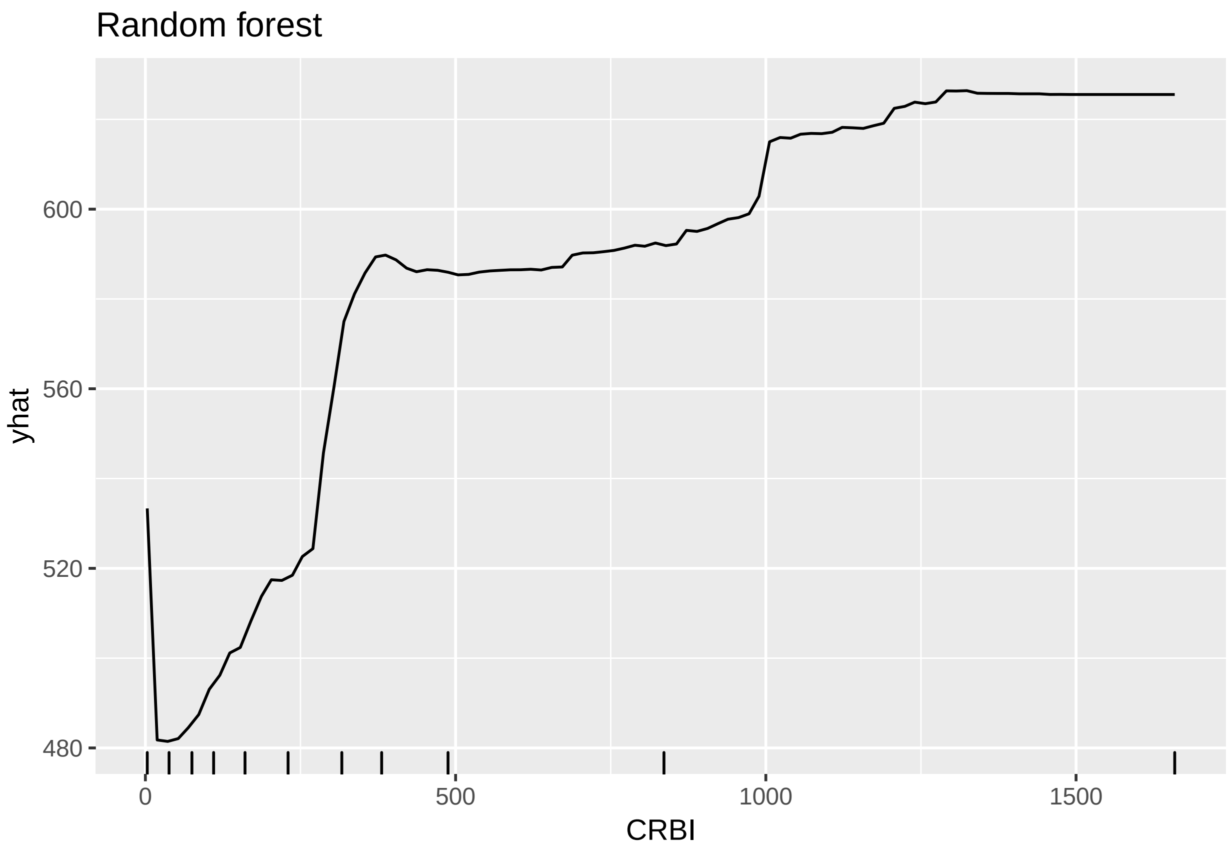


Partial dependence plots and individual conditional expectation curves

After the most relevant variables have been identified, the next step is to attempt to understand how the response variable changes based on these variables. For this we can use partial dependence plots (PDPs) and individual conditional expectation (ICE) curves.

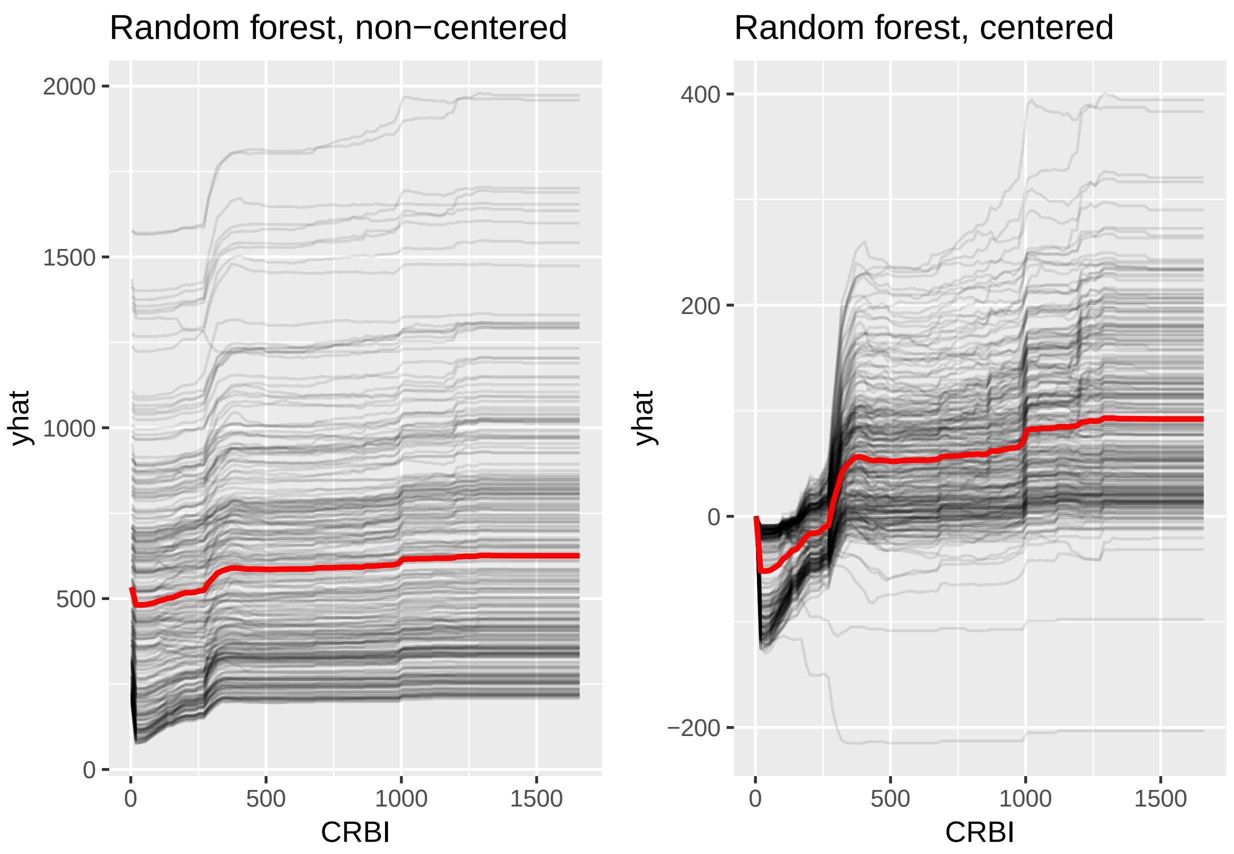
PDPs plot the change in the average predicted value as specified feature(s) vary over their marginal distribution. The PDP plot below displays the average change in predicted `Salary` as we vary `CRBI` while holding all other variables constant. This is done by holding all variables constant for each observation in our training data set but then apply the unique values of `CRBI` for each observation. We then average the `Salary` across all the observations.

```r
pdp.rf <- rf.fit %>%
  partial(pred.var = "CRBI",
          grid.resolution = 100) %>%
  autoplot(rug = TRUE, train = Hitters) +
  ggtitle("Random forest")

pdp.rf
```



ICE curves are an extension of PDP plots but, rather than plot the average marginal effect on the response variable, we plot the change in the predicted response variable for each observation as we vary each predictor variable.

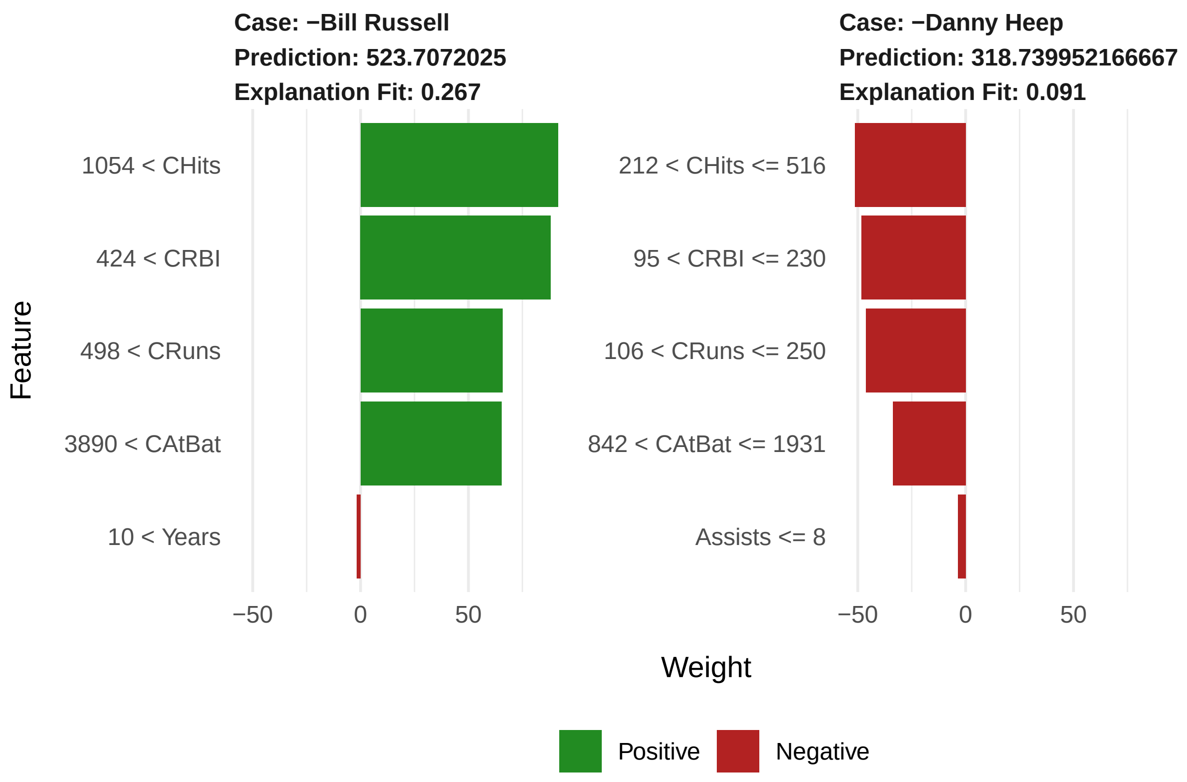
```r
ice1.rf <- rf.fit %>%
  partial(pred.var = "CRBI",
          grid.resolution = 100,
          ice = TRUE) %>%
  autoplot(train = Hitters, alpha = .1) +
  ggtitle("Random forest, non-centered")

ice2.rf <- rf.fit %>%
  partial(pred.var = "CRBI",
          grid.resolution = 100,
          ice = TRUE) %>%
  autoplot(train = Hitters, alpha = .1,
           center = TRUE) +
  ggtitle("Random forest, centered")


grid.arrange(ice1.rf, ice2.rf,
             nrow = 1, ncol = 2)
```



Plot the features in an explanation

The function `plot_features()` creates a compact visual representation of the explanations for each case and label combination in an explanation. Each extracted feature is shown with its weight, thus giving the importance of the feature in the label prediction.

```
new_obs <- Hitters2[c(10,20),-19]
explainer.rf <- lime(Hitters[,-19], rf.fit)
explanation.rf <- explain(new_obs, explainer.rf, n_features = 5)
plot_features(explanation.rf)
```


Case: −Bill Russell
Prediction: 523.7072025
Explanation Fit: 0.267
Case: −Danny Heep
Prediction: 318.739952166667
Explanation Fit: 0.091
Feature
1054 < CHits
424 < CRBI
498 < CRuns
3890 < CAtBat
10 < Years
212 < CHits <= 516
95 < CRBI <= 230
106 < CRuns <= 250
842 < CAtBat <= 1931
Assists <= 8
−50
0
50
Weight
Positive
Negative