Predicting Situational Time On Ice

Kyle Joecken

Question

Can we use data from previous seasons to predict how much time a particular NHL skater will get on the ice (TOI) in various situations (even strength, short-handed, power play) in the upcoming 2014-2015 season?

Data

As before, we load our database of season-wide NHL data into a data frame called `skaterstats`. We'll also grab our games played predictions and store it in a data frame called `skatpred15`; these two steps are done with the `setup.R` file. Finally, we will load our previous model for games played (`GPModel`).

Feature Selection

We'd like to compare a few different attempts. First, we'd like to model time on ice directly, with and without our predicted games played. Then, we'd like to model time on ice per game and multiply it by our predicted games played numbers and compare the accuracy.

We'll start again by modelling even-strength TOI using all possible predictors, then try to ascertain which are the most important factors.

```
gbmdata <- nhlShape(2012, 2013, outcome = 38)
rfdata <- nhlShape(2013, 2013, outcome = 38)
```

First, we build random forest models from several seeds.

```
esrfmod1 <- nhlBuild(data = rfdata, perc = 0.7, seed = 9112)
```

```
## [1] 0.7332324
```

```
esrfmod2 <- nhlBuild(data = rfdata, perc = 0.7, seed = 2857)
```

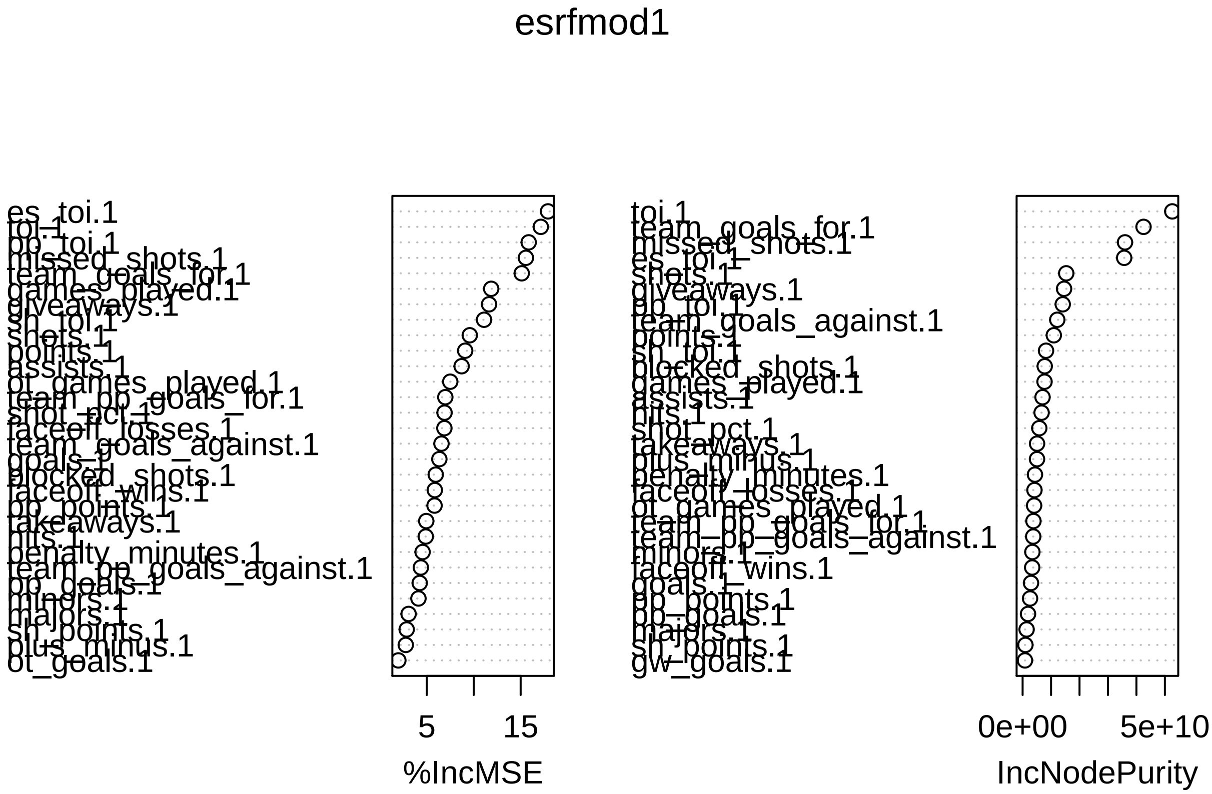
```
## [1] 0.7851254
```

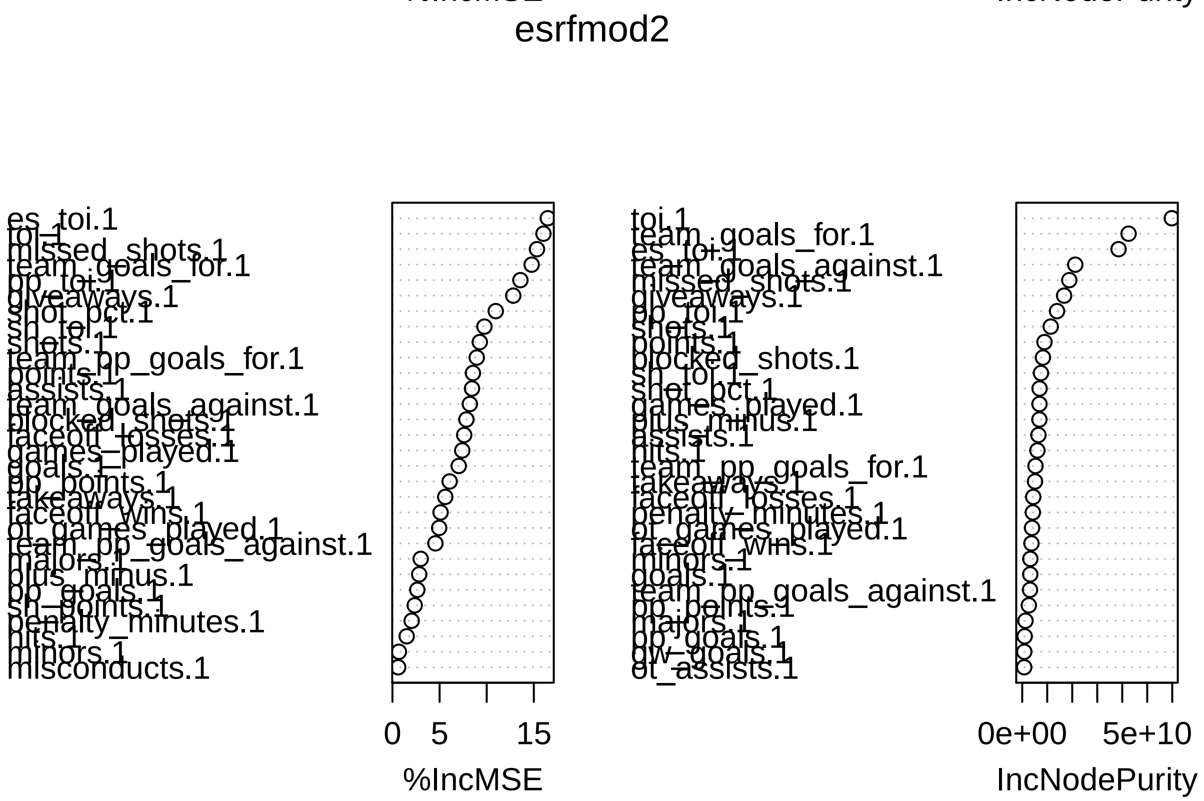
```
esrfmod3 <- nhlBuild(data = rfdata, perc = 0.7, seed = 31415)
```

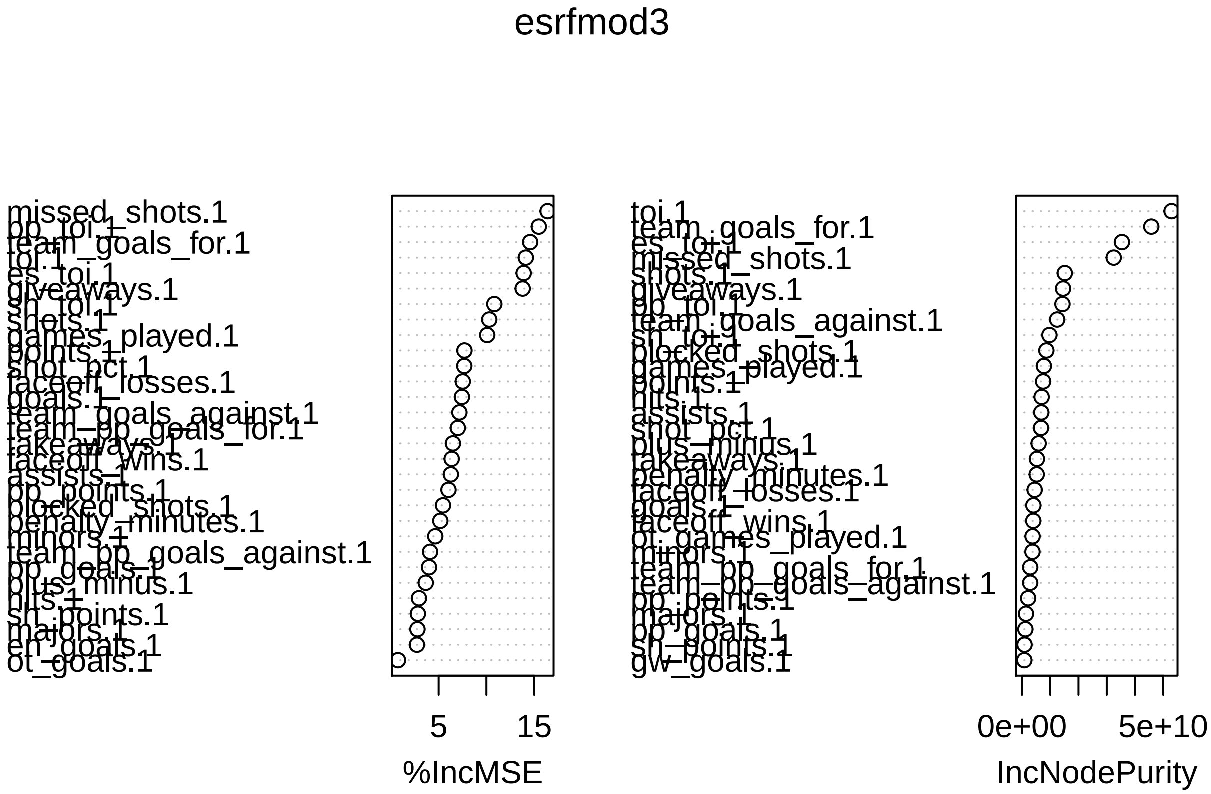
```
## [1] 0.7582238
```

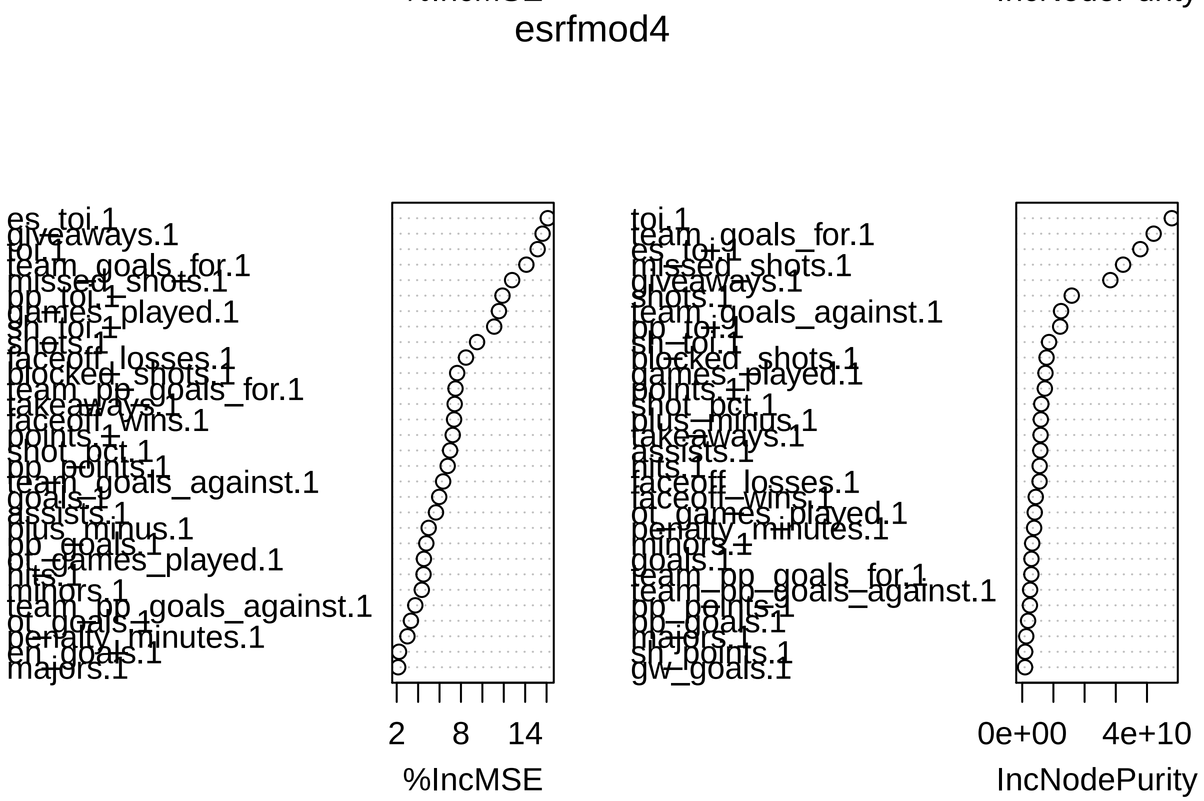
```
esrfmod4 <- nhlBuild(data = rfdata, perc = 0.7, seed = 28182)
```

```
## [1] 0.7335663
```


esrfmod1
es_toi.1
toi.1
pp_toi.1
missed_shots.1
team_goals_for.1
games_played.1
giveaways.1
sh_toi.1
shots.1
points.1
assists.1
ot_games_played.1
team_pp_goals_for.1
shot_pct.1
faceoff_losses.1
team_goals_against.1
goals.1
blocked_shots.1
faceoff_wins.1
pp_points.1
takeaways.1
hits.1
penalty_minutes.1
team_pp_goals_against.1
pp_goals.1
minors.1
majors.1
sh_points.1
plus_minus.1
ot_goals.1
5
15
%IncMSE
toi.1
team_goals_for.1
missed_shots.1
es_toi.1
shots.1
giveaways.1
pp_toi.1
team_goals_against.1
points.1
sh_toi.1
blocked_shots.1
games_played.1
assists.1
hits.1
shot_pct.1
takeaways.1
plus_minus.1
penalty_minutes.1
faceoff_losses.1
ot_games_played.1
team_pp_goals_for.1
team_pp_goals_against.1
minors.1
faceoff_wins.1
goals.1
pp_points.1
pp_goals.1
majors.1
sh_points.1
gw_goals.1
0e+00
5e+10
IncNodePurity


esrfmod2
es_toi.1
toi.1
missed_shots.1
team_goals_for.1
pp_toi.1
giveaways.1
shot_pct.1
sh_toi.1
shots.1
team_pp_goals_for.1
points.1
assists.1
team_goals_against.1
blocked_shots.1
faceoff_losses.1
games_played.1
goals.1
pp_points.1
takeaways.1
faceoff_wins.1
ot_games_played.1
team_pp_goals_against.1
majors.1
plus_minus.1
pp_goals.1
sh_points.1
penalty_minutes.1
hits.1
minors.1
misconducts.1
0
5
15
%IncMSE
toi.1
team_goals_for.1
es_toi.1
team_goals_against.1
missed_shots.1
giveaways.1
pp_toi.1
shots.1
points.1
blocked_shots.1
sh_toi.1
shot_pct.1
games_played.1
plus_minus.1
assists.1
hits.1
team_pp_goals_for.1
takeaways.1
faceoff_losses.1
penalty_minutes.1
ot_games_played.1
faceoff_wins.1
minors.1
goals.1
team_pp_goals_against.1
pp_points.1
majors.1
pp_goals.1
gw_goals.1
ot_assists.1
0e+00
5e+10
IncNodePurity






```
esgbmmod1 <- nhlBuild(data = gbmdata, type = "gbm", perc = 0.7, seed = 9112, n.trees = 10000, cv.folds =
esgbmmod2 <- nhlBuild(data = gbmdata, type = "gbm", perc = 0.7, seed = 2857, n.trees = 10000, cv.folds =
esgbmmod3 <- nhlBuild(data = gbmdata, type = "gbm", perc = 0.7, seed = 31415, n.trees = 10000, cv.folds
```

```
esgbmmod4 <- nhlBuild(data = gbmdata, type = "gbm", perc = 0.7, seed = 28182, n.trees = 10000, cv.folds
```

```
##                  var var.inf.tot
## 1  team_goals_for.1   57.983263
## 2    missed_shots.1   55.538200
## 3             toi.1   50.723774
## 4          es_toi.1   36.647614
## 5  team_goals_for.2   28.477084
## 6          es_toi.2   18.466111
## 7             toi.2   16.286954
## 8          pp_toi.1   14.056329
## 9       giveaways.1   10.440052
## 10         sh_toi.1    9.118568
## 11          shots.1    7.953250
## 12         pp_toi.2    6.804834
## 13      takeaways.2    6.158915
## 14   games_played.1    5.725170
## 15      giveaways.2    5.269562
## 16  blocked_shots.1    4.489300
## 17  blocked_shots.2    4.332368
## 18           hits.2    4.273493
```

```
col1 <- c(1:3, 15, 24, 30:31, 38:41)
col2 <- c(28:29, 32)
esDataRF1 <- nhlShape(2013, 2013, cols = c(col1, col2[1]), outcome = 38, rm.nhlnum = F)
esDataRF2 <- nhlShape(2013, 2013, cols = c(col1, col2[2]), outcome = 38, rm.nhlnum = F)
esDataRF3 <- nhlShape(2013, 2013, cols = c(col1, col2[3]), outcome = 38, rm.nhlnum = F)
esDataRF4 <- nhlShape(2013, 2013, cols = c(col1, col2), outcome = 38, rm.nhlnum = F)
esDataGBM1 <- nhlShape(2012, 2013, cols = c(col1, col2[1]), outcome = 38,
                       rm.nhlnum = F, rm.NA = FALSE)
esDataGBM1 <- subset(esDataGBM1, !is.na(games_played.1))
esDataGBM2 <- nhlShape(2012, 2013, cols = c(col1, col2[2]), outcome = 38,
                       rm.nhlnum = F, rm.NA = FALSE)
esDataGBM2 <- subset(esDataGBM2, !is.na(games_played.1))
esDataGBM3 <- nhlShape(2012, 2013, cols = c(col1, col2[3]), outcome = 38,
                       rm.nhlnum = F, rm.NA = FALSE)
esDataGBM3 <- subset(esDataGBM3, !is.na(games_played.1))
esDataGBM4 <- nhlShape(2012, 2013, cols = c(col1, col2), outcome = 38,
                       rm.nhlnum = F, rm.NA = FALSE)
esDataGBM4 <- subset(esDataGBM4, !is.na(games_played.1))
```

Here are the low level models.

```
esrfmod1 <- nhlBuild(esDataRF1[, -1], perc = 1, seed = 77677)
esrfmod2 <- nhlBuild(esDataRF2[, -1], perc = 1)
esrfmod3 <- nhlBuild(esDataRF3[, -1], perc = 1)
esrfmod4 <- nhlBuild(esDataRF4[, -1], perc = 1)
esgbmmod1 <- nhlBuild(esDataGBM1[, -1], type = "gbm", perc = 1, n.trees = 10000, cv.folds = 5, n.cores =
```

```
## Distribution not specified, assuming gaussian ...
```

```
esgbmmod2 <- nhlBuild(esDataGBM2[, -1], type = "gbm", perc = 1, n.trees = 10000, cv.folds = 5, n.cores =
```

```
## Distribution not specified, assuming gaussian ...
```

```
esgbmmod3 <- nhlBuild(esDataGBM3[, -1], type = "gbm", perc = 1, n.trees = 10000, cv.folds = 5, n.cores =
```

```
## Distribution not specified, assuming gaussian ...
```

```
esgbmmod4 <- nhlBuild(esDataGBM4[, -1], type = "gbm", perc = 1, n.trees = 10000, cv.folds = 5, n.cores =
```

```
## Distribution not specified, assuming gaussian ...
```

We build our data frame with the low level predictions in it.

```
esrf1 <- predict(esrfmod1, esDataRF1)
esrf2 <- predict(esrfmod2, esDataRF2)
esrf3 <- predict(esrfmod3, esDataRF3)
esrf4 <- predict(esrfmod4, esDataRF4)
esrf <- as.data.frame(cbind(esDataRF1[, 1], esrf1, esrf2, esrf3, esrf4))
names(esrf) <- c("nhl_num", "esrf1", "esrf2", "esrf3", "esrf4")
esgbm1 <- predict(esgbmmod1, esDataGBM1)
```

```
## Using 9999 trees...
```

```
esgbm2 <- predict(esgbmmod2, esDataGBM2)
```

```
## Using 9991 trees...
```

```
esgbm3 <- predict(esgbmmod3, esDataGBM3)
```

```
## Using 9994 trees...
```

```
esgbm4 <- predict(esgbmmod4, esDataGBM4)
```

```
## Using 9919 trees...
```

```
esgbm <- as.data.frame(cbind(esDataGBM1[, 1], esgbm1, esgbm2, esgbm3, esgbm4))
names(esgbm) <- c("nhl_num", "esgbm1", "esgbm2", "esgbm3", "esgbm4")
esData <- merge(esrf, esgbm, all = TRUE)
esData <- merge(esData, esDataGBM1[, c(1, 22)])
```

The final even strength model is built.

```
esModel <- nhlBuild(esData[, -1], type = "gbm", perc = 0.7, seed = 98765, n.trees = 10000, cv.folds = 5
```

```
## Distribution not specified, assuming gaussian ...
## Using 10000 trees...
## [1] 0.9711687
```

We repeat the overall process for short-handed time next.

```r
gbmdata <- nhlShape(2012, 2013, outcome = 39)
rfdata <- nhlShape(2013, 2013, outcome = 39)
```

```r
shrfmod1 <- nhlBuild(data = rfdata, perc = 0.7, seed = 9112)
```

```
## [1] 0.7279542
```

```r
shrfmod2 <- nhlBuild(data = rfdata, perc = 0.7, seed = 2857)
```

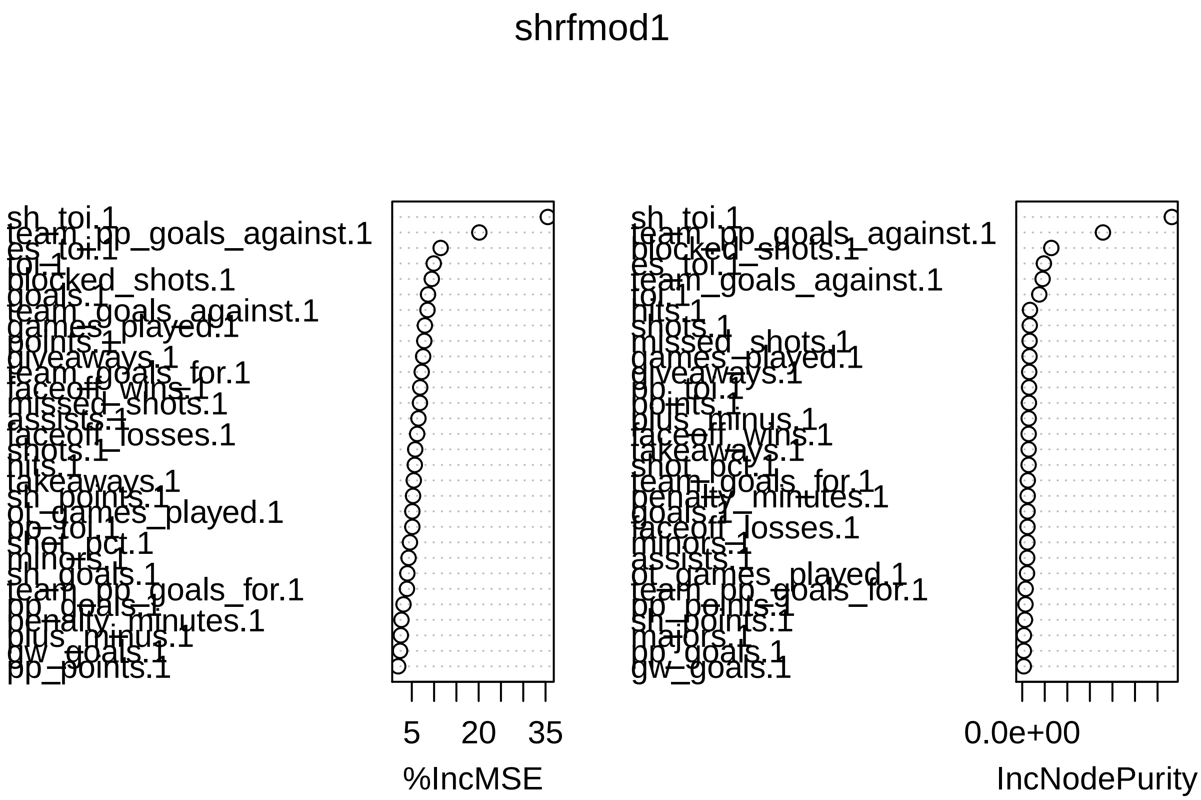
```
## [1] 0.778058
```

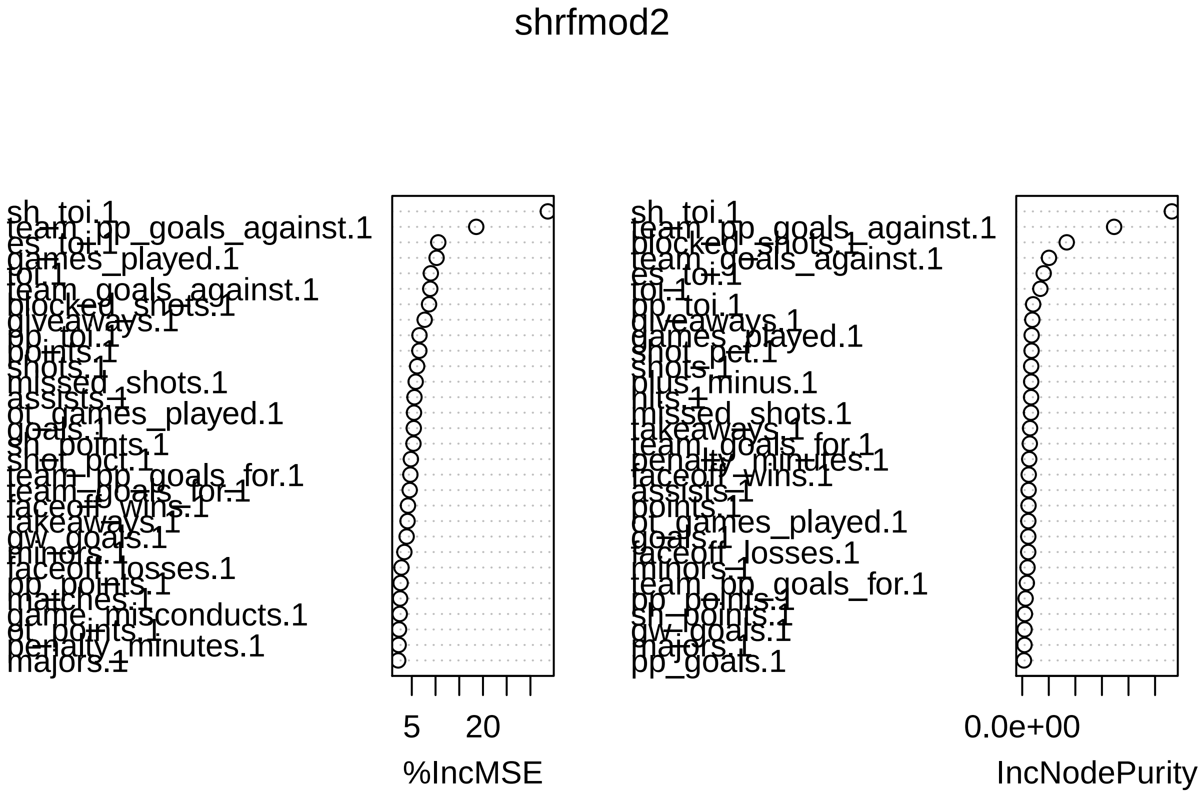
```r
shrfmod3 <- nhlBuild(data = rfdata, perc = 0.7, seed = 31415)
```

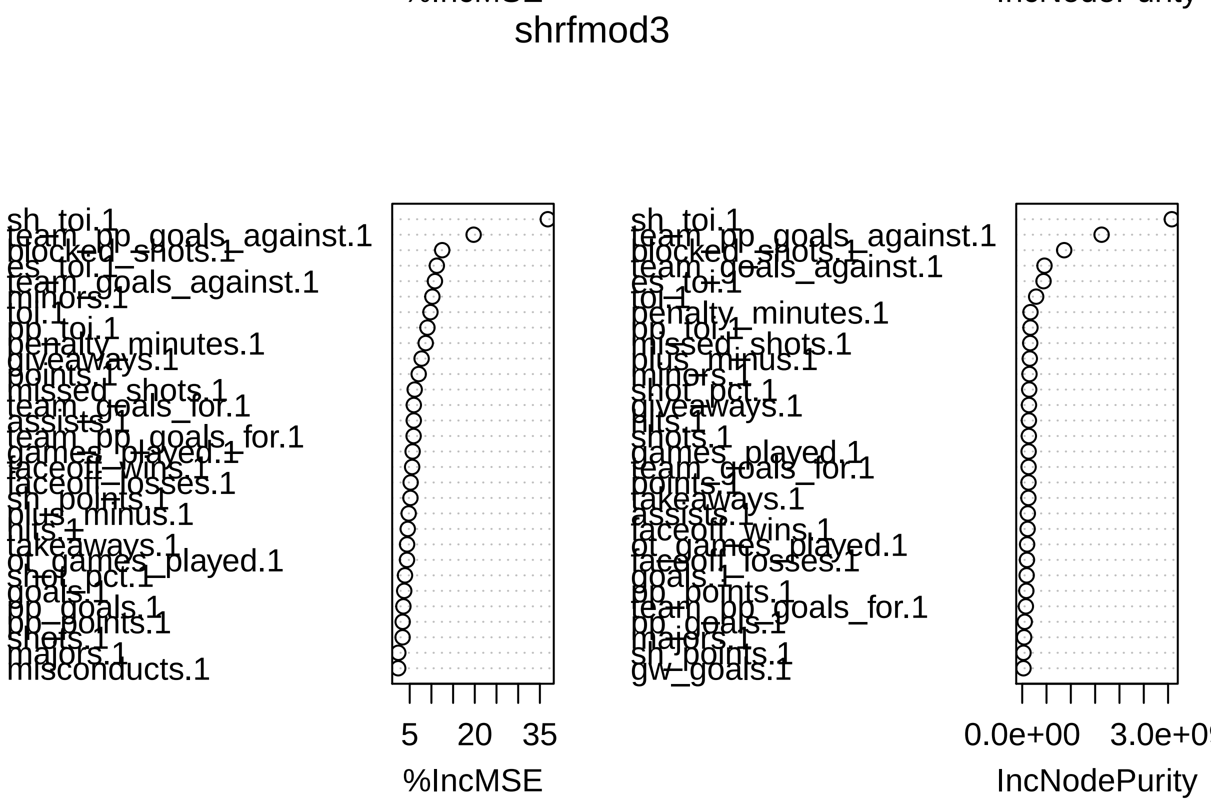
```
## [1] 0.7535301
```

```r
shrfmod4 <- nhlBuild(data = rfdata, perc = 0.7, seed = 28182)
```

```
## [1] 0.7803814
```




shrfmod2
sh_toi.1
team_pp_goals_against.1
es_toi.1
games_played.1
toi.1
team_goals_against.1
blocked_shots.1
giveaways.1
pp_toi.1
points.1
shots.1
missed_shots.1
assists.1
ot_games_played.1
goals.1
sh_points.1
shot_pct.1
team_pp_goals_for.1
team_goals_for.1
faceoff_wins.1
takeaways.1
gw_goals.1
minors.1
faceoff_losses.1
pp_points.1
matches.1
game_misconducts.1
ot_points.1
penalty_minutes.1
majors.1
5
20
%IncMSE
sh_toi.1
team_pp_goals_against.1
blocked_shots.1
team_goals_against.1
es_toi.1
toi.1
pp_toi.1
giveaways.1
games_played.1
shot_pct.1
shots.1
plus_minus.1
hits.1
missed_shots.1
takeaways.1
team_goals_for.1
penalty_minutes.1
faceoff_wins.1
assists.1
points.1
ot_games_played.1
goals.1
faceoff_losses.1
minors.1
team_pp_goals_for.1
pp_points.1
sh_points.1
gw_goals.1
majors.1
pp_goals.1
0.0e+00
IncNodePurity


shrfmod3
sh_toi.1
team_pp_goals_against.1
blocked_shots.1
es_toi.1
team_goals_against.1
minors.1
toi.1
pp_toi.1
penalty_minutes.1
giveaways.1
points.1
missed_shots.1
team_goals_for.1
assists.1
team_pp_goals_for.1
games_played.1
faceoff_wins.1
faceoff_losses.1
sh_points.1
plus_minus.1
hits.1
takeaways.1
ot_games_played.1
shot_pct.1
goals.1
pp_goals.1
pp_points.1
shots.1
majors.1
misconducts.1
5
20
35
%IncMSE
sh_toi.1
team_pp_goals_against.1
blocked_shots.1
team_goals_against.1
es_toi.1
toi.1
penalty_minutes.1
pp_toi.1
missed_shots.1
plus_minus.1
minors.1
shot_pct.1
giveaways.1
hits.1
shots.1
games_played.1
team_goals_for.1
points.1
takeaways.1
assists.1
faceoff_wins.1
ot_games_played.1
faceoff_losses.1
goals.1
pp_points.1
team_pp_goals_for.1
pp_goals.1
majors.1
sh_points.1
gw_goals.1
0.0e+00
3.0e+09
IncNodePurity


```
shgbmmod1 <- nhlBuild(data = gbmdata, type = "gbm", perc = 0.7, seed = 9112, n.trees = 10000, cv.folds =
shgbmmod2 <- nhlBuild(data = gbmdata, type = "gbm", perc = 0.7, seed = 2857, n.trees = 10000, cv.folds =
shgbmmod3 <- nhlBuild(data = gbmdata, type = "gbm", perc = 0.7, seed = 31415, n.trees = 10000, cv.folds
shgbmmod4 <- nhlBuild(data = gbmdata, type = "gbm", perc = 0.7, seed = 28182, n.trees = 10000, cv.folds
```

```
##                        var var.inf.tot
## 1                 sh_toi.1  235.904451
## 2                 sh_toi.2   52.866135
## 3 team_pp_goals_against.1   19.982456
## 4          blocked_shots.1    8.464144
## 5 team_pp_goals_against.2    6.716257
## 6                    toi.1    5.656339
## 7          blocked_shots.2    5.498642
## 8                 es_toi.1    5.282992
```

Both models seem to agree; the previous season's `sh_toi` is far and away the best predictor, with `team_pp_goals_against` making a surprising showing in second place. Perhaps there is a possession argument to be made for the inclusion of this factor.

Unsurprising entries that show up every time are remaining TOI stats and `blocked_shots`. Additional factors of value that show up consistently in random forest models are `team_goals_against`, `giveaways`, and `points`; we'll use these as minor factors.

This is going to be quite the simple model.

```
col1 <- c(1:2, 27, 29, 38:41)
col2 <- c(6, 26, 31)
shDataRF1 <- nhlShape(2013, 2013, cols = c(col1, col2[1]), outcome = 39, rm.nhlnum = F)
```

```r
shDataRF2 <- nhlShape(2013, 2013, cols = c(col1, col2[2]), outcome = 39, rm.nhlnum = F)
shDataRF3 <- nhlShape(2013, 2013, cols = c(col1, col2[3]), outcome = 39, rm.nhlnum = F)
shDataRF4 <- nhlShape(2013, 2013, cols = c(col1, col2), outcome = 39, rm.nhlnum = F)
shDataGBM1 <- nhlShape(2012, 2013, cols = c(col1, col2[1]), outcome = 39,
                       rm.nhlnum = F, rm.NA = FALSE)
shDataGBM1 <- subset(shDataGBM1, !is.na(sh_toi.1))
shDataGBM2 <- nhlShape(2012, 2013, cols = c(col1, col2[2]), outcome = 39,
                       rm.nhlnum = F, rm.NA = FALSE)
shDataGBM2 <- subset(shDataGBM2, !is.na(sh_toi.1))
shDataGBM3 <- nhlShape(2012, 2013, cols = c(col1, col2[3]), outcome = 39,
                       rm.nhlnum = F, rm.NA = FALSE)
shDataGBM3 <- subset(shDataGBM3, !is.na(sh_toi.1))
shDataGBM4 <- nhlShape(2012, 2013, cols = c(col1, col2), outcome = 39,
                       rm.nhlnum = F, rm.NA = FALSE)
shDataGBM4 <- subset(shDataGBM4, !is.na(sh_toi.1))
```

Here are the low level models.

```r
shrfmod1 <- nhlBuild(shDataRF1[, -1], perc = 1, seed = 77677)
shrfmod2 <- nhlBuild(shDataRF2[, -1], perc = 1)
shrfmod3 <- nhlBuild(shDataRF3[, -1], perc = 1)
shrfmod4 <- nhlBuild(shDataRF4[, -1], perc = 1)
shgbmmod1 <- nhlBuild(shDataGBM1[, -1], type = "gbm", perc = 1, n.trees = 10000, cv.folds = 5, n.cores =
```

```
## Distribution not specified, assuming gaussian ...
```

```r
shgbmmod2 <- nhlBuild(shDataGBM2[, -1], type = "gbm", perc = 1, n.trees = 10000, cv.folds = 5, n.cores =
```

```
## Distribution not specified, assuming gaussian ...
```

```r
shgbmmod3 <- nhlBuild(shDataGBM3[, -1], type = "gbm", perc = 1, n.trees = 10000, cv.folds = 5, n.cores =
```

```
## Distribution not specified, assuming gaussian ...
```

```r
shgbmmod4 <- nhlBuild(shDataGBM4[, -1], type = "gbm", perc = 1, n.trees = 10000, cv.folds = 5, n.cores =
```

```
## Distribution not specified, assuming gaussian ...
```

We build our data frame with the low level predictions in it.

```r
shrf1 <- predict(shrfmod1, shDataRF1)
shrf2 <- predict(shrfmod2, shDataRF2)
shrf3 <- predict(shrfmod3, shDataRF3)
shrf4 <- predict(shrfmod4, shDataRF4)
shrf <- as.data.frame(cbind(shDataRF1[, 1], shrf1, shrf2, shrf3, shrf4))
names(shrf) <- c("nhl_num", "shrf1", "shrf2", "shrf3", "shrf4")
shgbm1 <- predict(shgbmmod1, shDataGBM1)
```

```
## Using 8075 trees...
```

```r
shgbm2 <- predict(shgbmmod2, shDataGBM2)
```

```
## Using 9996 trees...
```

```r
shgbm3 <- predict(shgbmmod3, shDataGBM3)
```

```
## Using 9999 trees...
```

```r
shgbm4 <- predict(shgbmmod4, shDataGBM4)
```

```
## Using 8737 trees...
```

```r
shgbm <- as.data.frame(cbind(shDataGBM1[, 1], shgbm1, shgbm2, shgbm3, shgbm4))
names(shgbm) <- c("nhl_num", "shgbm1", "shgbm2", "shgbm3", "shgbm4")
shData <- merge(shrf, shgbm, all = TRUE)
shData <- merge(shData, shDataGBM1[, c(1, 16)])
```

The final short-handed model is built.

```r
shModel <- nhlBuild(shData[, -1], type = "gbm", perc = 0.7, seed = 98765, n.trees = 10000, cv.folds = 5
```

```
## Distribution not specified, assuming gaussian ...
## Using 10000 trees...
## [1] 0.9660476
```

And once more for the power play.

```r
gbmdata <- nhlShape(2012, 2013, outcome = 40)
rfdata <- nhlShape(2013, 2013, outcome = 40)
```

```r
pprfmod1 <- nhlBuild(data = rfdata, perc = 0.7, seed = 9112)
```

```
## [1] 0.8262312
```

```r
pprfmod2 <- nhlBuild(data = rfdata, perc = 0.7, seed = 2857)
```

```
## [1] 0.8410991
```

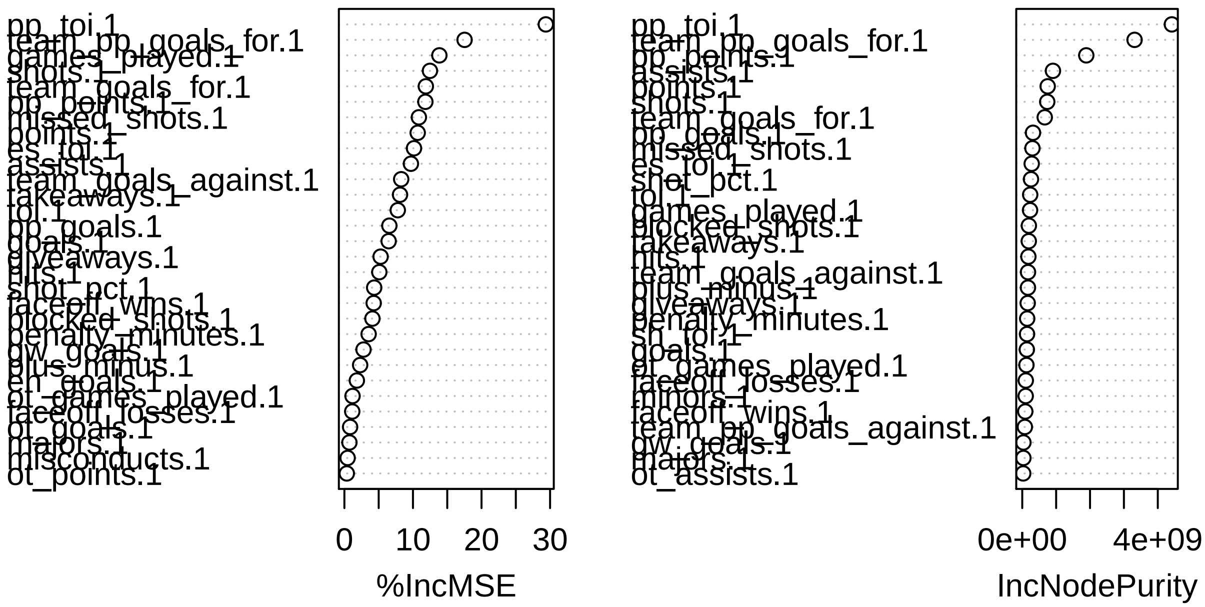
```r
pprfmod3 <- nhlBuild(data = rfdata, perc = 0.7, seed = 31415)
```

```
## [1] 0.8528829
```

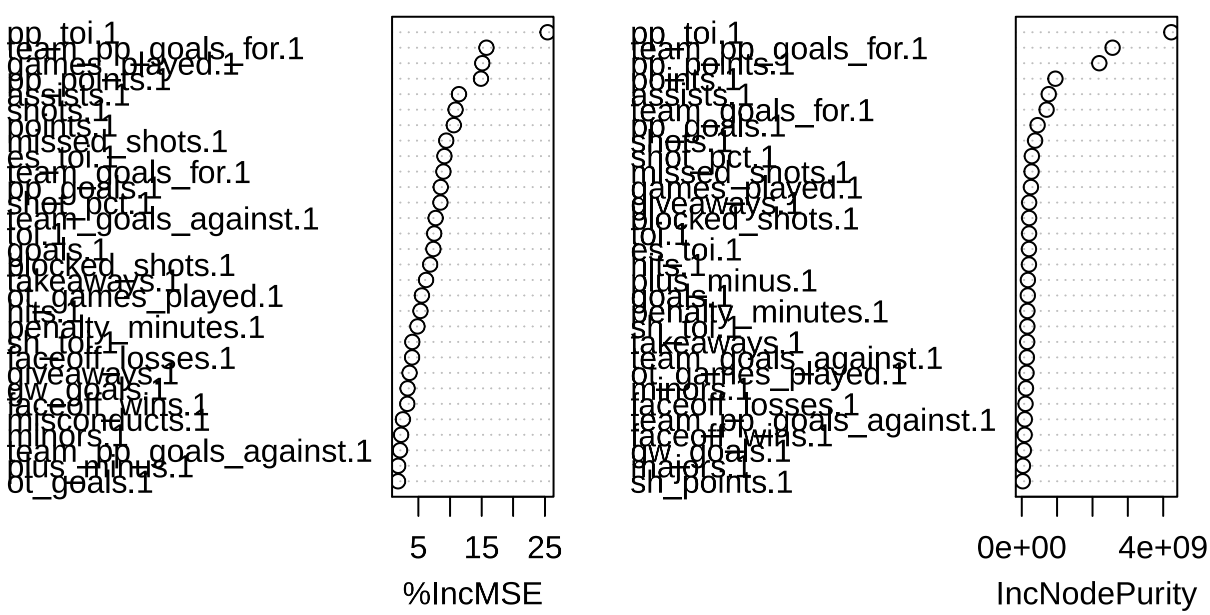
```r
pprfmod4 <- nhlBuild(data = rfdata, perc = 0.7, seed = 28182)
```

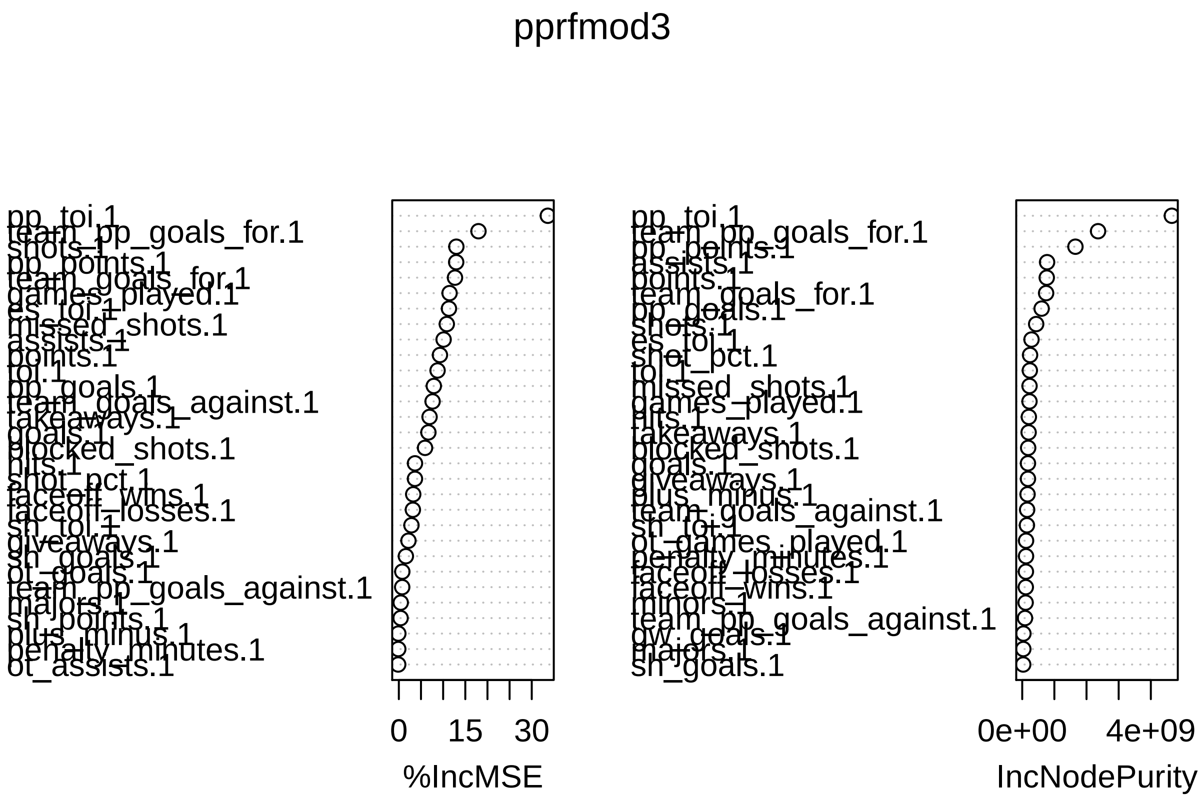
```
## [1] 0.8280485
```

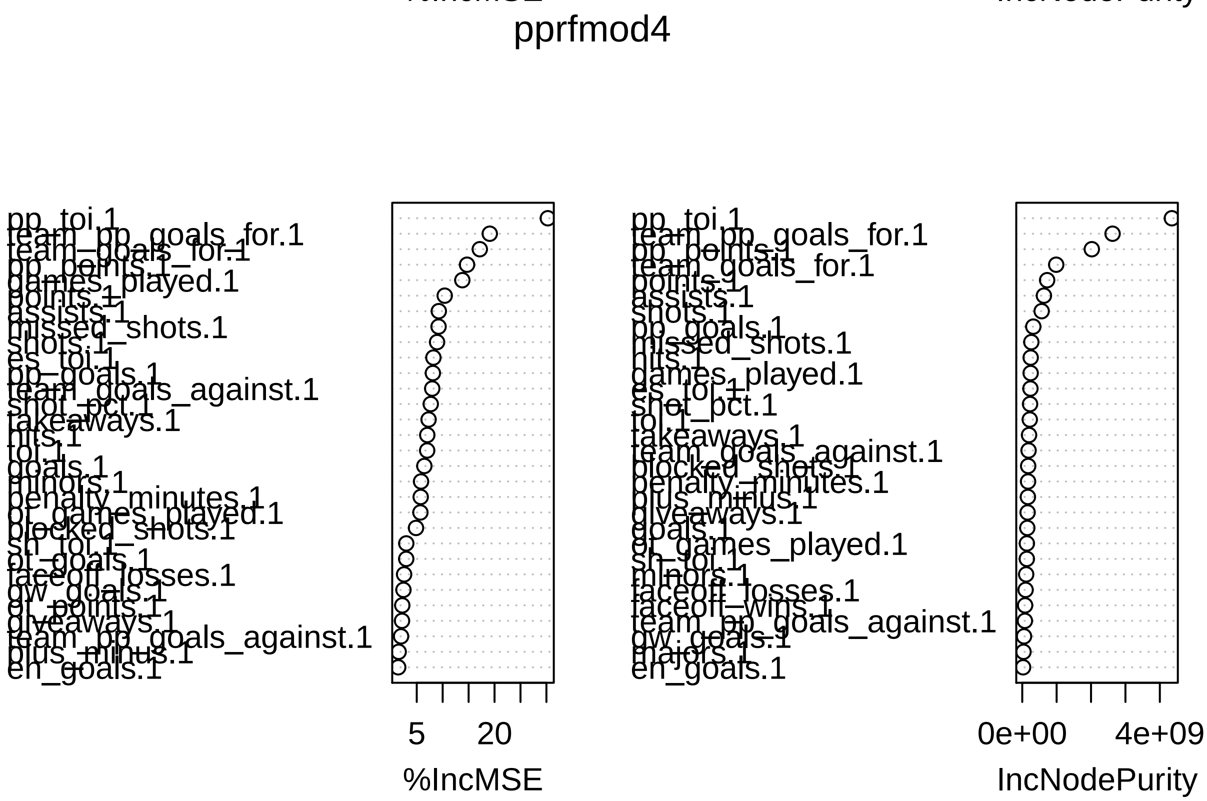
pprfmod1



pprfmod2








```
ppgbmmod1 <- nhlBuild(data = gbmdata, type = "gbm", perc = 0.7, seed = 9112, n.trees = 10000, cv.folds =
ppgbmmod2 <- nhlBuild(data = gbmdata, type = "gbm", perc = 0.7, seed = 2857, n.trees = 10000, cv.folds =
ppgbmmod3 <- nhlBuild(data = gbmdata, type = "gbm", perc = 0.7, seed = 31415, n.trees = 10000, cv.folds
```

```r
ppgbmmod4 <- nhlBuild(data = gbmdata, type = "gbm", perc = 0.7, seed = 28182, n.trees = 10000, cv.folds
```

```
##                    var var.inf.tot
## 1             pp_toi.1  182.942573
## 2             pp_toi.2   40.428422
## 3          pp_points.1   22.297918
## 4  team_pp_goals_for.1   19.674872
## 5  team_pp_goals_for.2   16.254380
## 6     team_goals_for.1   15.934968
## 7            assists.1   12.609585
## 8          pp_points.2    7.777941
## 9            assists.2    6.756863
## 10             shots.1    6.433354
## 11      missed_shots.2    6.258376
## 12          pp_goals.2    6.076638
## 13    team_goals_for.2    4.708492
## 14            points.1    4.642754
```

Not surprisingly, overall and power play offensive metrics reign supreme here. In particular, team success metrics (*e.g.*, `team_goals_for`) are important.

Shot statistics and points have lesser importance to both models types, and we also add `games_played` and `es_toi` from the random forest importance graphs as second-tier predictors.

```r
col1 <- c(1:2, 5, 10, 24:25, 40)
col2 <- c(3, 6, 15, 30, 38)
ppDataRF1 <- nhlShape(2013, 2013, cols = c(col1, col2[1]), outcome = 40, rm.nhlnum = F)
ppDataRF2 <- nhlShape(2013, 2013, cols = c(col1, col2[2]), outcome = 40, rm.nhlnum = F)
ppDataRF3 <- nhlShape(2013, 2013, cols = c(col1, col2[3]), outcome = 40, rm.nhlnum = F)
ppDataRF4 <- nhlShape(2013, 2013, cols = c(col1, col2[4]), outcome = 40, rm.nhlnum = F)
ppDataRF5 <- nhlShape(2013, 2013, cols = c(col1, col2[5]), outcome = 40, rm.nhlnum = F)
ppDataRF6 <- nhlShape(2013, 2013, cols = c(col1, col2), outcome = 40, rm.nhlnum = F)
ppDataGBM1 <- nhlShape(2012, 2013, cols = c(col1, col2[1]), outcome = 40,
                       rm.nhlnum = F, rm.NA = FALSE)
ppDataGBM1 <- subset(ppDataGBM1, !is.na(pp_toi.1))
ppDataGBM2 <- nhlShape(2012, 2013, cols = c(col1, col2[2]), outcome = 40,
                       rm.nhlnum = F, rm.NA = FALSE)
ppDataGBM2 <- subset(ppDataGBM2, !is.na(pp_toi.1))
ppDataGBM3 <- nhlShape(2012, 2013, cols = c(col1, col2[3]), outcome = 40,
                       rm.nhlnum = F, rm.NA = FALSE)
ppDataGBM3 <- subset(ppDataGBM3, !is.na(pp_toi.1))
ppDataGBM4 <- nhlShape(2012, 2013, cols = c(col1, col2[4]), outcome = 40,
                       rm.nhlnum = F, rm.NA = FALSE)
ppDataGBM4 <- subset(ppDataGBM4, !is.na(pp_toi.1))
ppDataGBM5 <- nhlShape(2012, 2013, cols = c(col1, col2[5]), outcome = 40,
                       rm.nhlnum = F, rm.NA = FALSE)
ppDataGBM5 <- subset(ppDataGBM5, !is.na(pp_toi.1))
ppDataGBM6 <- nhlShape(2012, 2013, cols = c(col1, col2), outcome = 40,
                       rm.nhlnum = F, rm.NA = FALSE)
ppDataGBM6 <- subset(ppDataGBM6, !is.na(pp_toi.1))
```

Here are the low level models.

```r
pprfmod1 <- nhlBuild(ppDataRF1[, -1], perc = 1, seed = 77677)
pprfmod2 <- nhlBuild(ppDataRF2[, -1], perc = 1)
pprfmod3 <- nhlBuild(ppDataRF3[, -1], perc = 1)
pprfmod4 <- nhlBuild(ppDataRF4[, -1], perc = 1)
pprfmod5 <- nhlBuild(ppDataRF5[, -1], perc = 1)
pprfmod6 <- nhlBuild(ppDataRF6[, -1], perc = 1)
ppgbmmod1 <- nhlBuild(ppDataGBM1[, -1], type = "gbm", perc = 1, n.trees = 10000, cv.folds = 5, n.cores
```

```
## Distribution not specified, assuming gaussian ...
```

```r
ppgbmmod2 <- nhlBuild(ppDataGBM2[, -1], type = "gbm", perc = 1, n.trees = 10000, cv.folds = 5, n.cores
```

```
## Distribution not specified, assuming gaussian ...
```

```r
ppgbmmod3 <- nhlBuild(ppDataGBM3[, -1], type = "gbm", perc = 1, n.trees = 10000, cv.folds = 5, n.cores
```

```
## Distribution not specified, assuming gaussian ...
```

```r
ppgbmmod4 <- nhlBuild(ppDataGBM4[, -1], type = "gbm", perc = 1, n.trees = 10000, cv.folds = 5, n.cores
```

```
## Distribution not specified, assuming gaussian ...
```

```r
ppgbmmod5 <- nhlBuild(ppDataGBM5[, -1], type = "gbm", perc = 1, n.trees = 10000, cv.folds = 5, n.cores
```

```
## Distribution not specified, assuming gaussian ...
```

```r
ppgbmmod6 <- nhlBuild(ppDataGBM6[, -1], type = "gbm", perc = 1, n.trees = 10000, cv.folds = 5, n.cores
```

```
## Distribution not specified, assuming gaussian ...
```

We build our data frame with the low level predictions in it.

```r
pprf1 <- predict(pprfmod1, ppDataRF1)
pprf2 <- predict(pprfmod2, ppDataRF2)
pprf3 <- predict(pprfmod3, ppDataRF3)
pprf4 <- predict(pprfmod4, ppDataRF4)
pprf5 <- predict(pprfmod5, ppDataRF5)
pprf6 <- predict(pprfmod6, ppDataRF6)
pprf <- as.data.frame(cbind(ppDataRF1[, 1], pprf1, pprf2, pprf3, pprf4, pprf5, pprf6))
names(pprf) <- c("nhl_num", "pprf1", "pprf2", "pprf3", "pprf4", "pprf5", "pprf6")
ppgbm1 <- predict(ppgbmmod1, ppDataGBM1)
```

```
## Using 9789 trees...
```

```r
ppgbm2 <- predict(ppgbmmod2, ppDataGBM2)
```

```
## Using 9989 trees...
```

```r
ppgbm3 <- predict(ppgbmmod3, ppDataGBM3)
```

```
## Using 9978 trees...
```

```r
ppgbm4 <- predict(ppgbmmod4, ppDataGBM4)
```

```
## Using 6451 trees...
```

```r
ppgbm5 <- predict(ppgbmmod5, ppDataGBM5)
```

```
## Using 9055 trees...
```

```r
ppgbm6 <- predict(ppgbmmod6, ppDataGBM6)
```

```
## Using 9995 trees...
```

```r
ppgbm <- as.data.frame(cbind(ppDataGBM1[, 1], ppgbm1, ppgbm2, ppgbm3, ppgbm4, ppgbm5, ppgbm6))
names(ppgbm) <- c("nhl_num", "ppgbm1", "ppgbm2", "ppgbm3", "ppgbm4", "ppgbm5", "ppgbm6")
ppData <- merge(pprf, ppgbm, all = TRUE)
ppData <- merge(ppData, ppDataGBM1[, c(1, 14)])
```

The power play time model is finally built.

```r
ppModel <- nhlBuild(ppData[, -1], type = "gbm", perc = 0.7, seed = 98765, n.trees = 10000, cv.folds = 5
```

```
## Distribution not specified, assuming gaussian ...
## Using 10000 trees...
## [1] 0.9704033
```