NHL Playoffs Analysis

Pierre du Pont

December 29, 2018

Introduction

The National Hockey League (NHL) is one of the four major sports leagues in the United States. As a professional sports league with team operating incomes ranging from $107 million (New York Rangers) to -$21 million (Florida Panthers), data analytics have become increasingly important as teams look for an edge that can push them to the next level. If a team makes the playoffs, they can expect large amounts of additional revenue through box office, sponsorships, and TV money, as well as an additional payout–part of $15 million, depending on how far the team progresses. So teams look for the stats that can drive them towards wins.

The NHL playoffs take 16 teams, 8 from each conference, based on the total number of points that a team earns. Theoretically, the top 8 teams in each conference make the playoffs, although this is not always the case due to the quirks of playoff qualifications. Teams play 82 games per season, and the best teams will win almost 60 games in a season, while the worst can win as few as 20. Teams earn two points for a win, no points for a loss in regular time, and one point for a loss in overtime or a shootout.

But hockey is not the first sport to turn to analytics–and in fact, it is one of the last. The first sport to see widespread use of analytics was baseball. Bill James pioneered the use of SABRmetrics in the 80s, and since then baseball teams across the professional leagues (the MLB in the US) have used analytics with varying levels of success. Today, we will focus on a specific type of analytics–how can we predict playoff appearances.. The data set is collected from Hockey Reference and goes back to the 2004 season. We used this season because the NHL had a lockout resulting in a missed season. Additionally, a result of the lockout was a series of rule changes that resulted in a very different style of play. Therefore, data from before the lockout may show different trends. We attempt to predict which teams make the playoffs using a series of categorical models, including random forest, discriminant analysis, and nearest neighbors.

The data and code can be downloaded from github, but will also be included (as necessary) in this document. It can be downloaded using the following code (which also installs necessary packages:

```r
if(!require(lattice)) install.packages("lattice")
```

```
## Loading required package: lattice
```

```r
if(!require(tidyverse)) install.packages("tidyverse")
```

```
## Loading required package: tidyverse

## -- Attaching packages -----------------------------------------------------------------------------

## v ggplot2 3.1.0     v purrr   0.2.5
## v tibble  1.4.2     v dplyr   0.7.8
## v tidyr   0.8.2     v stringr 1.3.1
## v readr   1.2.1     v forcats 0.3.0

## -- Conflicts ----------------------------------------------------------------------------------
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```r
if(!require(caret)) install.packages("caret")
```

```
## Loading required package: caret

##
## Attaching package: 'caret'

## The following object is masked from 'package:purrr':
##
```

```
##     lift
```

```r
if(!require(readxl)) install.packages("readxl")
```

```
## Loading required package: readxl
```

```r
if(!require(ggalt)) install.packages("ggalt")
```

```
## Loading required package: ggalt
```

```r
raw_data_file <- "https://raw.github.com/psdupvi/nhl-analysis/master/raw-data.csv"
raw_data <- read_csv(raw_data_file)
```

```
## Parsed with column specification:
## cols(
##   .default = col_double(),
##   team = col_character(),
##   playoff = col_character(),
##   champ = col_character(),
##   runner = col_character()
## )
```

```
## See spec(...) for full column specifications.
```

```r
raw_data <- raw_data[rowSums(is.na(raw_data)) != ncol(raw_data),]
```

Methods

Data Exploration and Cleaning

Let's look at the raw data

```
## # A tibble: 6 x 37
##   team   rank av_age    gp     w     l    ol   pts pts_perc    gf    ga
##   <chr> <dbl>  <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>    <dbl> <dbl> <dbl>
## 1 Nash~     1   28.4    82    53    18    11   117    0.713   267   211
## 2 Winn~     2   26.8    82    52    20    10   114    0.695   277   218
## 3 Tamp~     3   27.5    82    54    23     5   113    0.689   296   236
## 4 Bost~     4   28.6    82    50    20    12   112    0.683   270   214
## 5 Vega~     5   28      82    51    24     7   109    0.665   272   228
## 6 Wash~     6   28.4    82    49    26     7   105    0.64    259   239
## # ... with 26 more variables: sow <dbl>, sol <dbl>, simp_rat_sys <dbl>,
## #   strength_of_sched <dbl>, goal_game <dbl>, evgf <dbl>, evga <dbl>,
## #   pp <dbl>, ppo <dbl>, pp_perc <dbl>, ppa <dbl>, ppoa <dbl>,
## #   pk_perc <dbl>, sh <dbl>, sha <dbl>, pm_game <dbl>, o_pm_game <dbl>,
## #   shots <dbl>, shot_percent <dbl>, shots_against <dbl>,
## #   sv_percent <dbl>, pdo <dbl>, year <dbl>, playoff <chr>, champ <chr>,
## #   runner <chr>
```

Our data contains 37 variables, although we only take some of them into our actual analysis. Lets look at playoffs vs points and see if we notice any trends about the points required. First, we turn playoffs into a factor, then find the maximum number of points that misses the playoffs each year. Lets also find the average.

```
## # A tibble: 13 x 2
##     year missed
##    <dbl>  <dbl>
##  1  2005     92
##  2  2006     95
##  3  2007     92
##  4  2008     93
##  5  2009     92
##   6  2010      95
##   7  2011      92
##   8  2012      55
##   9  2013      92
##  10  2014      96
##  11  2015      93
##  12  2016      94
##  13  2017      96
```

We notice that something weird is happening in 2010. The season was actually shortened due to a lockout. So our first step was to normalize the data to 82 games using the following code

```
raw_data_short <- raw_data %>% filter(gp < 82) %>%
  mutate(pts = pts*82/gp,w = w*82/gp, l = l*82/gp,
         ga = ga*82/gp,gf = gf*82/gp, sow = sow*82/gp,
         sol = sol*82/gp, evgf = evgf*82/gp,
         evga = evga*82/gp,
         pp = pp*82/gp, ppo = ppo*82/gp,
         ppoa = ppoa*82/gp,
         ppa = ppa*82/gp,
         sh = sh*82/gp, sha = sha*82/gp,
         shots = shots*82/gp, shots_against = shots_against*82/gp,
         gp = 82)

raw_data_not_short <- raw_data %>% filter(gp == 82)

nhl_data <- rbind(raw_data_not_short,raw_data_short)
```

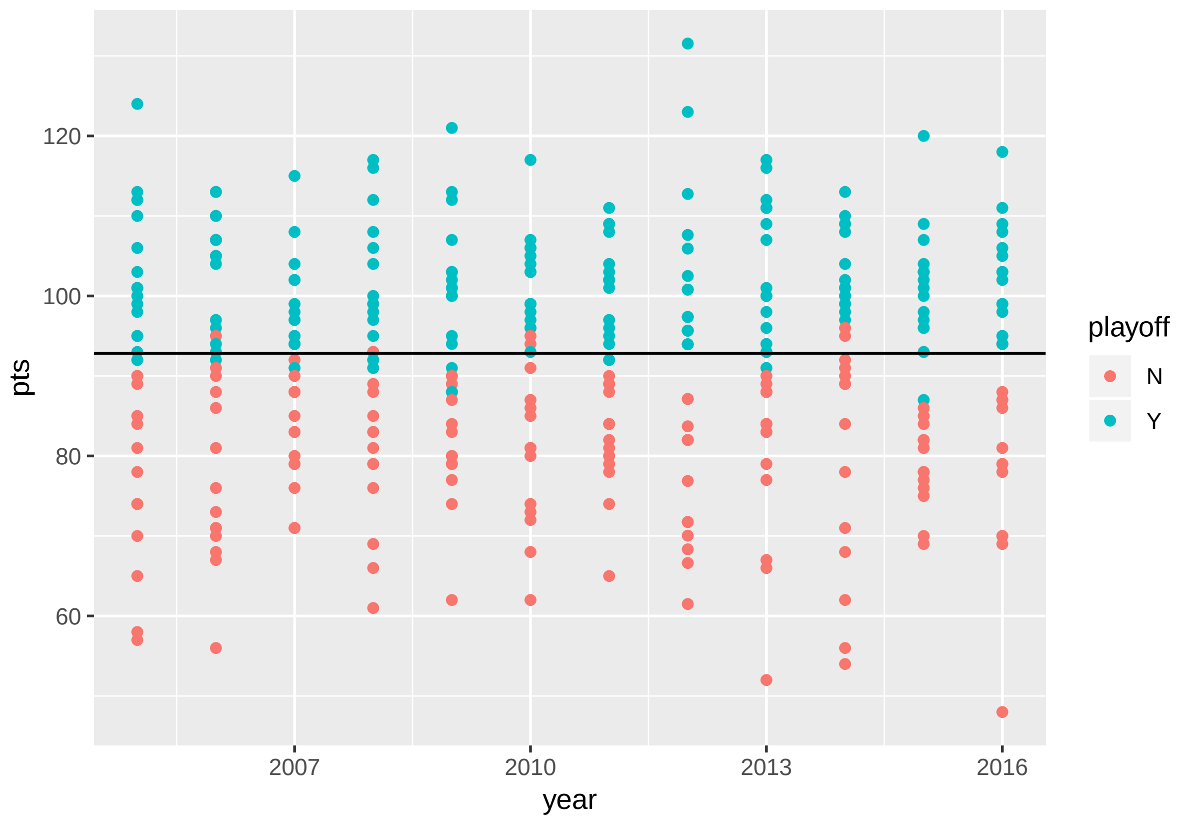
Before we run the analysis again, let's make our train and test sets, as well as select relevant, non linearly combined columns from the sets. We also remove any rows containing the league average. By linearly combined, we mean, for example, that wins are directly proportional to points, as are losses and points percentage.

```
test_data_playoffs <- filter(nhl_data, year == 2017) %>%
  filter(team != "League Average")
train_data_playoffs <- filter(nhl_data, year != 2017) %>%
  filter(team != "League Average")

train_set <- train_data_playoffs %>% select(-year, -champ, -runner,
                                            -team, -rank, -pts_perc,
                                            -gp, -year, -pk_perc,
                                            -pp_perc, -w, -l, -sol, -sow)
```

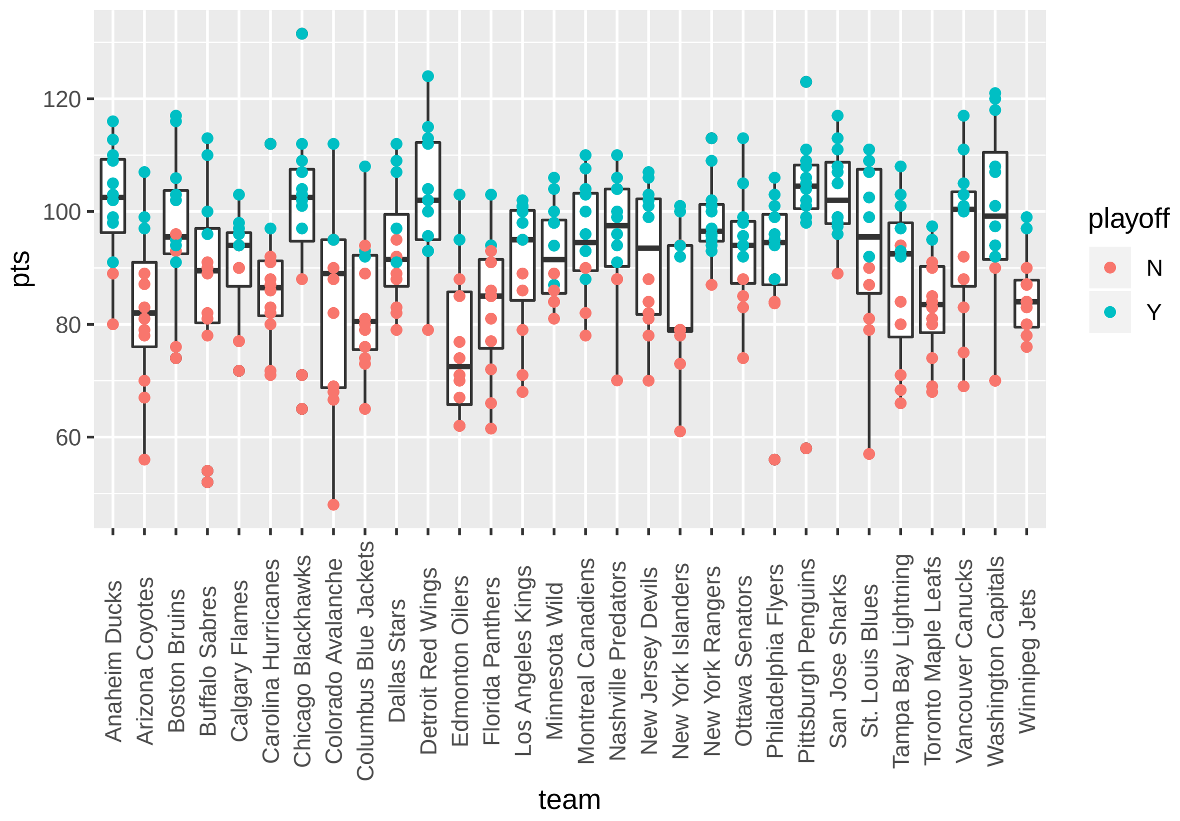
Let's try the analysis again, and this time plot the results as well, with the playoff vs. non playoff teams colored and the average points to make the playoffs included.

```
## # A tibble: 12 x 2
##     year missed
##    <dbl>  <dbl>
##   1  2005   92
##   2  2006   95
##   3  2007   92
##   4  2008   93
##   5  2009   90
##   6  2010   95
##   7  2011   90
##   8  2012   94.0
##   9  2013   90
##  10  2014   96
##  11  2015   93
##  12  2016   94
```



Now we explore the results by team. Note that some teams have more success than others

```
## # A tibble: 30 x 5
##    team                  avg playoffs finals  cups
##    <chr>               <dbl>    <int>  <int> <int>
##  1 San Jose Sharks     103.        11      1     0
##  2 Detroit Red Wings   103.        11      2     1
##  3 Pittsburgh Penguins 102         11      4     3
##  4 Anaheim Ducks       101.        10      1     1
##  5 Chicago Blackhawks   99.2        9      3     3
##  6 Washington Capitals  99.0        9      0     0
##  7 New York Rangers     98.6       11      1     0
##  8 Boston Bruins        96.9        8      2     1
##  9 Nashville Predators  95.8        9      1     0
## 10 Vancouver Canucks    95.4        7      1     0
## # ... with 20 more rows
```



Modeling Approach

At this point, lets start actually testing models. As a classification problem, our first attempt is with a random forest method. Luckily, our data set is small, so we don't lose much time even with the longer fitting methods. We trained our model with the training set, and then tested our accuracy on the test set.

```
fit <- train(playoff ~ ., data = train_set, method = "rf")
pred <- predict(fit,newdata = test_data_playoffs)
```

This gives an accuracy of 0.9, which seems like a good start. But there's a problem… A quick analysis of the predictions shows that the number of teams predicted to make the playoffs is 19, which does not fit the NHL rules. We actually went through approximately 10 models before we noticed this flaw.

Luckily there's a workaround. We set the type option in predict to "prob", which returns the probability of each classification instead of the actual predicted class.

```
fit <- train(playoff ~ ., data = train_set, method = "rf")
pred <- predict(fit,newdata = test_data_playoffs, type = "prob")
head(pred)
```

```
##       N     Y
## 1 0.012 0.988
## 2 0.014 0.986
## 3 0.086 0.914
## 4 0.012 0.988
## 5 0.118 0.882
## 6 0.126 0.874
```

We then selected the top 16 probabilities as our playoff teams (i.e., "Y"), and the remainder as missing the playoffs ("N"). Since we had 31 teams, this was an easy trick using median. Then we test the accuracy again

```
pred$Y[pred$Y >= median(pred$Y)] <- "Y"
pred$Y[pred$Y < median(pred$Y)] <- "N"
```

```r
acc <- mean(pred$Y == test_data_playoffs$playoff)
```

Our new accuracy is 0.94, which is actually an improvement. This makes sense because our model was over predicting the number of playoff teams, and any over prediction is automatically wrong.

Models and Function

We used a series of models of different types with an ensemble method as well to try to maximize our accuracy. We also created a data frame to store our predictions in

```r
models <- c("rf","lda",'naive_bayes','kknn','loclda',
            'wsrf','avNNet','monmlp','adaboost','gbm','hda')

combined_preds <- setNames(data.frame(matrix(ncol = length(models) + 1,
                                             nrow = length(test_data_playoffs$w))),
                           c(models,"Overall"))
```

Our function for quickly applying the models is as follows. It fits the train data for each model, and then uses a for loop to turn the probability predictions into 16 playoff teams and 15 misses. Then we use the predictions from each model to create our ensemble prediction.

As the for loop is the most vague aspect, here is a quick rundown: the loop calculates a set of probabilities for each fit on the test set. It then turns those probabilities into "Y" and "N", before inputting those into the combined predictions data frame.

```r
fits <- lapply(models, function(model){
  print(model)
  train(playoff ~ . , method = model, data = train_set)
})

for(i in 1:length(models)){
  pred <- as.data.frame(predict(fits[i], newdata = test_data_playoffs, type = "prob"))
  pred$Y[pred$Y >= median(pred$Y)] <- "Y"
  pred$Y[pred$Y < median(pred$Y)] <- "N"
  combined_preds[i] <- pred$Y
}

votes <- rowMeans(combined_preds == "Y", na.rm = TRUE)
combined_preds$Overall <- ifelse(votes > 0.5, "Y", "N")
```

Results

```r
combined_preds
```

```
##    rf lda naive_bayes kknn loclda wsrf avNNet monmlp adaboost gbm hda
## 1   Y   Y           Y    Y      Y    Y      Y      Y        Y   Y   Y
## 2   Y   Y           Y    Y      Y    Y      Y      Y        Y   Y   Y
## 3   Y   Y           Y    Y      Y    Y      Y      Y        Y   Y   Y
## 4   Y   Y           Y    Y      Y    Y      Y      Y        Y   Y   Y
## 5   Y   Y           Y    Y      Y    Y      Y      Y        Y   Y   Y
## 6   Y   Y           Y    Y      Y    Y      Y      Y        Y   Y   Y
## 7   Y   Y           Y    Y      N    Y      Y      Y        Y   Y   Y
## 8   Y   Y           Y    Y      Y    Y      Y      Y        Y   N   Y
## 9   Y   Y           Y    N      Y    Y      Y      Y        Y   Y   Y
## 10  Y   Y           Y    Y      Y    Y      N      Y        Y   Y   Y
## 11  Y   Y           Y    Y      Y    Y      Y      Y        Y   Y   Y
## 12  Y   Y           Y    Y      Y    Y      Y      Y        Y   Y   N
## 13  Y   Y           Y    Y      Y    Y      Y      Y        Y   Y   Y
## 14  Y   Y           Y    Y      Y    Y      Y      Y        N   Y   Y
## 15  Y   Y           N    N      N    Y      N      Y        Y   Y   Y
## 16  N   N           N    N      Y    N      N      Y        Y   Y   Y
## 17  N   N           N    Y      N    N      Y      Y        Y   Y   Y
## 18  N   Y           Y    Y      Y    N      Y      N        Y   N   N
## 19  Y   N           Y    Y      Y    Y      Y      N        Y   N   N
## 20  N   N           N    N      N    N      N      N        Y   N   N
## 21  N   N           N    N      N    N      N      N        N   N   N
## 22  N   N           N    N      N    N      N      N        N   N   N
## 23  N   N           N    N      N    N      N      N        Y   N   N
## 24  N   N           N    N      N    N      N      N        N   N   N
## 25  N   N           N    N      N    N      N      N        N   N   N
## 26  N   N           N    N      N    N      N      N        N   N   N
## 27  N   N           N    N      N    N      N      N        N   N   N
## 28  N   N           N    N      N    N      N      N        N   N   N
## 29  N   N           N    N      N    N      N      N        Y   N   N
## 30  N   N           N    N      N    N      N      N        N   N   N
## 31  N   N           N    N      N    N      N      N        N   N   N
##    Overall
## 1        Y
## 2        Y
## 3        Y
## 4        Y
## 5        Y
## 6        Y
## 7        Y
## 8        Y
## 9        Y
## 10       Y
## 11       Y
## 12       Y
## 13       Y
## 14       Y
## 15       Y
## 16       N
## 17       Y
## 18       Y
## 19       Y
## 20       N
## 21       N
## 22       N
## 23       N
## 24       N
## 25       N
## 26       N
## 27       N
## 28       N
## 29       N
## 30       N
## 31       N
```

```r
test_data_playoffs$playoff <- as.character(test_data_playoffs$playoff)
## Need to switch back to a character for the test to actually work
acc <- colMeans(combined_preds == test_data_playoffs$playoff)

acc
```

```
##          rf         lda naive_bayes        kknn      loclda        wsrf
##   0.9354839   0.9354839   0.8709677   0.8709677   0.8064516   0.9354839
##      avNNet      monmlp    adaboost         gbm         hda     Overall
##   0.8709677   0.9677419   0.7741935   0.9354839   0.9354839   0.9354839
```

From this, we can see that most of our models have accuracy between 85% and 95%, although a few show accuracy of 100%. Our ensemble does not do as well as some individual models, which makes some sense–the ensemble is weighed down by some low scorers.

In general, it appears that the Heteroscedastic Discriminant Analysis model is the most accurate.

Conclusions and Next Steps

It would appear that we can accurately predict playoff teams with a mean accuracy of 0.8978495. However, that is not as interesting as predicting which teams will be in the playoffs before the season ends. Unfortunately, there is no way for us to test our predictions at this point, but a next step is to re-run this analysis at the current point of the NHL season and compare to the final results.

Additionally, this model does not take one more factor into account: the conference and division of each team. A better way to do this in the future would be to break the teams up into their conferences and divisions, then to more accurately model 8 teams from each conference. However, this requires a much more serious analysis and we are not quite sure how to do that at this point.