Predictive Hockey Analytics

Erik Hannesson

February 2020

Abstract

It's pretty straightforward, really. We used the National Hockey League's (NHL) API to steal whatever data we could get our grimyc hands on, and then:

Linear Algebra.[1]

And that's about all you need to know.

1 Motivation and Overview

2 Data

So I'm pretty sure we already explained this in the abstract, but I guess we'll vomit it back up here.

The NHL API provides both historical data and live feed data for games currently in play. This allows for incredible diversity in approach and application.

There are two main categories of data that can be analyzed: team data and player data. These can be approached independently, or simultaneously.

Here is a rundown of the main features we use.

Team Statistics

Statistic	Abbreviation
Goals For	GF
Goals Against	GA
Power Play Goals	PPG
Short Handed Goals	SHG
Power Play Percentage	PPP
Penalty Kill Percentage	PK
Blocked Shots	BLK
Shots on Goal	SOG

Skater Statistics

Statistic	Abbreviation
Goals For	GF
Goals Against	GA
Power Play Goals	PPG
Short Handed Goals	SHG
Power Play Percentage	PPP
Penalty Kill Percentage	PK
Blocked Shots	BLK
Shots on Goal	SOG

[1] See Figure A for a helpful infographic.

2.1 Historical Data

2.2 Live Feed

3 Methodology

3.1 Previous Research and Efforts

Uhmm...regression????

Also, apparently game classification accuracy is currently (?) "capped" out at around 66% (reference coming soon...). Surely we can beat this, no?

3.2 Basic Statistical Analysis

Uhmm, it's probably pretty important to have a section analyzing some "basic" statistical properties before liberally applying machine learning.

3.2.1 Team and Player Variance

We try to create a metric for measuring the variance in both team and individual player performance. This is an incredibly helpful statistic, as it can be a hugely influential factor in the models that we build.

3.3 Classification Models

3.3.1 Linear Regression

The first thing we wanted to analyze was which events contributed to the number of wins a team gets. Due to the large number of variables we have that participate in our model, we wanted to avoid overfitting by finding the best OLS model that minimizes the AIC. This was done by implementing the "Best Subset Method" which iterates through all subsets of independent variables, and finds the combination that minimizes the AIC. Although this is computationally complex, it is accurate in finding the variables that minimizes AIC since it takes all possible combinations. Without using this method, we found the following as our OLS model: (we are aware that there is currently nothing here; we're working on porting jupyter notebook stuff to here...)

Once we computed the best subset, the following was determined to be the best OLS model that minimizes AIC. Note that while the R-squared value in this model decreased by .008 from the model using all of the variables, we also went from 26 variables to 6 variables. This greatly simplifies our model, while indicating the most important variables in predicting how many games a team will win in a season. (Again, working on transfering from jupyter)

This OLS model suggest that the most important thing a team can do to win more games is not allow any in a game. This is closely followed by the total number of shots a team takes per game. We found it very interesting that it was "shots" that this model used, rather than goals per game scored. We are unsure of the significance of this at the moment, but will continue with our analysis.

3.3.2 Random Forest

For random forest we put our data into a random forest model and got the following results. Note that a tree diagram indicating the process of this random forest will be uploaded later. This random forest found a mean accuracy score of .929, which shows an incredible amount of accuracy for a ML algorithm predicting playoff teams.

3.3.3 Naive Bayes

Next we implemented a naive bayes algorithm, which resulted in very similar results as the random forest. Here we got an accuracy of .89, which is also extremely good for our analysis.

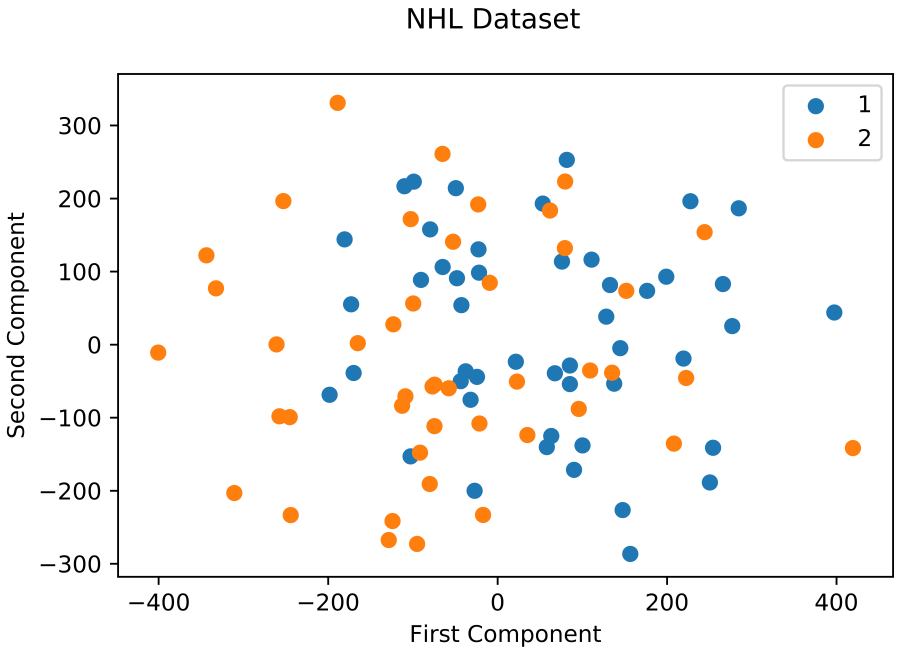

Figure 1: Sadly, this is pretty useless...

3.4 Dimensionality Reduction

There are a billion and one different statistics that are tracked. Figuring out what is important and what is not is pretty important (and nontrivial). Dimensionality reduction can significantly improve signal-to-noise ratios, etc.

3.4.1 Principal Component Analysis

We now moved on to classifying whether or not we can predict playoff teams based on the data. We proceed first by implementing a dimension reduction through PCA. We first find best number of components to include by calculating the variance for each respective number of components. This gives us that we should proceed with 2 components. Now, by putting our data into a PCA model we plot the data which results in the scatter plot below. We notice that there isn't any clustering of results, which suggests that PCA doesn't work the best with our dataset.

3.5 Time-Series Forecasting

3.5.1 ARIMA Modeling

Team Level

(i) General performance - wins, goals for, etc.

(ii) Specific team performance metrics - PPP, PK, PIM, etc. - to use in classification models for game outcomes *as if* the game already happened (does this make sense?)

Individual Level (and line level??)

Not particularly different from the team level, just for individuals. Obviously, this will encompass different statistics, and therefore different analysis. Also, there is quite a bit more that *could* be done here compared with the team level, I just can't really remember what at the moment.

3.6 State Estimation

3.6.1 Overview of Various States

This is where we explain what we mean by "state" in the context of hockey. This includes fully observable states, quasi-observable states (quasi-latent??), and latent states.

One example would be team build or style of play. Some teams are built to be quick on their feet while others are designed to play physically, with the goal of wearing out the other team. We would consider this a quasi-observable state, since you either use ML techniques to determine these states entirely autonomously (unsupervised) or semi-autonomously (semi-supervised), or manually label them based on personal familiarity.[2]

Fully Observable States Description of the fully observable states we deal with.

Latent (Hidden) States Description of the latent (and quasi-latent) states we deal with.

3.6.2 Hidden Markov Models

3.6.3 Mixture Models (Does this go here?)

3.6.4 Kalman Filter (Is this useful...?)

3.7 Reinforcement Learning

Could we run reinforcement learning on historical data but as if it were a live stream??? That is, could we train a model that would predict stuff about a game as its unfolding by using the very large amount of historical data available....?????? That would be awesome.

4 Analysis

5 Conclusion

[2]Or by whether or not a team's name sounds fierce. However, since there is a lack of academic research concerning such methods, it is hard to say how effective it would be.

A Appendix

Figure 2: For the less technical readers, this is all you really need to know about machine learning and artificial intelligence. (Source: https://xkcd.com/1838/)