Predictive Hockey Analytics

Zach Chase*
Erik Hannesson†

April 2020

Abstract

Predicting the outcome of sporting events has long been a fascinating yet challenging problem. While progress has been made with pregame predictions and in-game probability models for several sports, analysis of hockey has lagged behind. The purpose of this paper is to explore what is needed to create an accurate hockey predictor for both pregame and in-game predictions. We used the National Hockey League's (NHL) API to gather data grouped into one of two categories: Historical Data and Live Data. Using the Historical Data we attempted several Gaussian processes to predict winners. Next, using the Live Data, we used several classification algorithms to calculate the effect individual events have on the outcome of a match, with the best result of 75% accuracy using random forests. We then used random forests to build an in-game probability model for live data.

1 Motivation and Overview

The market for sports analytics is expected to reach almost $4 billion by the year 2022, and has impacted the industry all the way from driving customer support to helping expand partnerships. However, one of the more fascinating aspects of sports analytics is understanding the game at a more complex level. When teams understand themselves and their sport better, the result is an increase in performance. Billy Beane famously demonstrated this as the general manager of the Oakland Athletics's when he used mathematics and statistics to transform baseball by finding undervalued players [Lew03]. His ideas revolutionized the sporting industry and has led to teams using advanced software and mathematics to perform better. In addition to improving team performance, understanding sports better through analytics has transformed the gambling industry. On May 14, 2018 the Supreme Court struck down a 1992 federal law that banned commercial sports betting in most states. As a result, data showed a 175% increase in revenue from sports betting in the past year, with predictions showing that the industry will reach $8 billion by 2025 [Pre19]. With this massive surge in popularity, the sports gambling industry has relied on mathematics and statistics more, to create better pregame betting odds (bets made before the start of a a game) and in play betting odds (bets made throughout the game).

While some sports are easier to predict than others, hockey is notorious for being one of the most difficult sports to predict - both for pregame predictions and in play predictions. While in game predictions for the sport suffer from a lack of distinct, discrete events like in football or baseball, pregame predictions suffer from the low scoring yet fast flowing nature of the game. In

*Brigham Young University, Department of Mathematics
†Brigham Young University, Department of Mathematics

fact, it has been theorized by Joshua Weissbock from the University of Ottawa that predicting the winner of NHL games has an upper bound of 62% [Wei14].

With all of this in mind, we have decided to attempt to solve the problem of calculating in play probabilities for NHL games. To do this we need to accomplish two tasks:

1. Determine a base pregame probability metric for two teams,
2. Determine how events in a game change the probability outcome of a match.

2 Previous Research and Efforts

Previous research is grouped into two ideas: previous research on sport prediction models and on hockey predictions. Both have been explored but not extensively together. For example, ESPN uses an in game predictor for the NFL, MLB, and NBA, but not the NHL. While others have created their own versions of NHL predictions (along with in game predictions) no single accepted in game probability metric currently exists.

Figure 1: NFL In-Play Win Probability

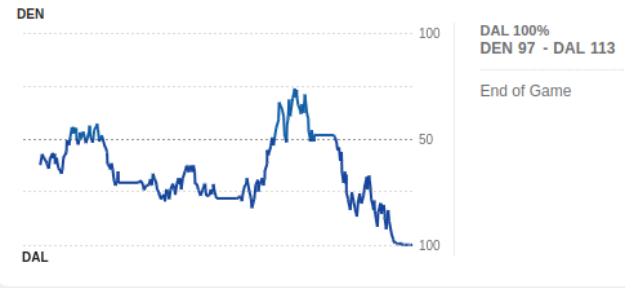

Figure 2: NBA In-Play Win Probability

Additionally, as cited earlier, Joshua Wiessbock's 2014 PhD thesis compared observed win percentage and simulated leagues to find a theoretical upper bound of approximately 62% for single game prediction in the NHL. This difficulty of single game predictions led to predicting the winner of the best-of-seven series with a resulting prediction accuracy of almost 75% [Wei14].

3 Ethics

Before we continue with our analysis, several ethical points must be considered. The main one we will discuss is the ethics behind player tracking devices. Starting in the 2019-2020 season, the NHL deployed Puck and Player Tracking technology that tracks data collected by sensors embedded in shoulder pads and inside the puck itself. Detailed information such as a player's coordinate during shots, puck speed, and player speed are all now easily available, along with numerous other statistics. While the intent of gathering this data is valid - it enhances team and player's performance while also helping viewers understand and enjoy the game more - it is important to realize that there needs to be some boundary that exists in regards of collecting player data. If the desire to continue collecting more data continues, then the next logical step is to access more detailed player information, such as their biometric data. This technology is already present with the NBA using an Kinexon wristband so coaches understand performance and wellness on the

court. While for many they don't have issues with coaches and trainers accessing this data, the concern grows when this data is handled carelessly or endangers player's privacy. We will ensure that this pursuit of knowledge doesn't grow to this point.

4 Data

The NHL API provides both historical data and live feed data for games currently in play. Additionally, there are a few independent websites dedicated to the careful collection of game data specifically for the purpose of statistical analysis.[1] As we measure the accuracy and validity of our data we first look at our sources. The NHL is one of the "Big 4" major professional sporting leagues, and uses precise technology to measure general game data (total shots, goals, etc.) and detailed event data (event, coordinate location, etc.). Additionally the independent websites are highly rated for their accuracy in data, and are a valid source as well. Both this validity and range of data allows for incredible diversity in approach and application. Details for both the historical data and live feed data are as follows.

4.1 Historical Data

The NHL tracks and aggregates various statistics during games and makes them available online and through their API. We refer to this collection of aggregated statistics as historical data. The unprocessed historical data we used in our analysis is relatively straightforward, consisting of basic features such as *goals for/against* (GF/GA), *shots for/against* (SF/SA), *power play/penalty kill percentage*[2] (PPP/PKP), and a few more standard statistics.

4.1.1 Preprocessing Historical Data

In addition to normalizing each feature, we applied many transformations to the raw to obtain a richer set of features. In particular, we transformed most "stale" metrics into rates grouped by game situations. That is, we considered the rate at which a team generated shots while at even strength, on a power play, and on a penalty kill. We applied this transformation to nearly every feature we collected, and then dropped the original feature.

4.2 Live Feed

This section of data is used to calculate and update in-game probabilities of a team winning. Again gathered from the NHL API, it was 4 separate datasets that collectively tracked the following 8 events in a hockey game: *hits*, *penalties*, *shots*, *missed shots*, *blocked shots*, *goals*, *takeaways*, and *giveaways*. For each event, other important key information about the event was tracked, including players involved, coordinates of where the event took place on a hockey rink, details about the game, and the time when the event occurred.

4.2.1 Preprocessing Live Data

The first task was to merge the 4 datasets together and then group by game. Next we converted the time given into seconds based on the period and time remaining in the period. For example, if

[1] In particular, we found the data curated by evolving-hockey to be invaluable.

[2] In hockey, a *power play* is a situation in which your team is allowed to have one or two additional skaters on the ice; in such a situation, the other team is on a *penalty kill*.

it was the 2nd period with 10:30 (10 minutes and 30 seconds) remaining in the period, then this converts to 1,830 seconds remaining in the game. This process enables us to sort the events by time to understand the game as it progresses.

In addition to using this dataset of all events in an entire game, we also grouped by event to create several different sets of data that will analyze as well.

5 Methodology

5.1 Pregame Prediction

5.1.1 Naive Bayes

We first implemented a Naive Bayes classifier on historical data to predict winners. This was done to analyze how well models perform under the "naive" assumption of independence among predictors, and to know how accurate predictions are if they know the end stats of a game.

5.2 Latent Variable Models

We attempted to implement a few latent variable models, hoping to use the historical data to estimate distributions of features that cannot be directly observed. In particular, we considered two stochastic process models - Gaussian process latent variable and Dirichlet process mixtures - though we were unable to get either model to converge properly. The cause for this lack of convergence is not immediately clear. However, we have no reason to believe that the data we have is incompatible with the models, and so we believe that our models simply need further tuning and more careful development.

5.2.1 Gaussian Process Latent Variable Model (GPLVM)

A stochastic process $(X_t)_{t\in T}$ is called a *Gaussian process* if for every finite subset $\tau \subset T$, the random variables $(X_t)_{t\in\tau}$ have a normally distributed joint probability distribution. Interestingly, in the case that T is infinite, a Gaussian process is an *infinite-dimensional* extension of normal distributions.

A GPLVM is an MCMC method that effectively uses Gibbs sampling to estimate an arbitrary probability distribution. They are very similar to the GMMHMMs we considered in the speech recognition lab.

5.2.2 Dirichlet Process Mixtures

A *Dirichlet process* is a stochastic process $(X_t)_{t\in T}$ with sample space equal to the space of probability distributions. Similarly to how Gaussian processes are infinite-dimensional extensions of normal distributions, Dirichlet processes are infinite-dimensional generalizations of Dirichlet distributions. Just as the Dirichlet distribution is a conjugate prior to the categorical distribution, a Dirichlet process is a conjugate prior to infinite categorical distributions (more technically, infinite, nonparametric discrete distributions).

5.3 In-Game Prediction

For in-game predictions we first wanted to determine if there was a way to predict the winner of a game by looking at a single event, the time it occurred, and where it occurred on the rink. Afterwards, we wanted to do the same thing but add a little more information about the game at

the time of the event, such as the current score. To analyze this we used the following classification algorithms: logistic regression, decision trees, naive Bayes, k-nearest neighbors, support vector machine, and random forest. Here is further detail about each machine learning algorithm we used:

5.3.1 Logistic Regression

Currently, famous probability metrics used by the NFL and fivethirtyeight use logistic regression at the core of their algorithms. When implementing our logistic regression we also included hyperparameter tuning with an exhaustive grid search. This iterates through different hyperparameters while fitting a model, and returns the model with the best parameters. The parameters we tuned for were penalty (l1 and l2 - which penalizes the loss function based on number of features) and the solver (such as 'newton-cg', 'lbfgs', 'liblinear', 'sag', and 'saga'). In the results section this will be labeled LR.

5.3.2 Decision Trees

Decision Trees were also used for our classifications since it uses a flowchart-like structure to determine classification. As we did with logistic regression, we also tuned our hyperparameters using exhaustive grid search. Two of the key parameters we looked at were the criterion and max depth parameters. For criterion, this measures the quality of a split and uses either the Gini impurity or the entropy for the information gain. As for max depth we considered different levels (including the default of none) to limit the complexity of our model. In the results section this will be labeled DT.

5.3.3 Gaussian Naive Bayes

The next algorithm that was considered is Gaussian Naive Bayes. We were curious how the naive assumption that each event is conditionally independent would have on our results. Also, since Naive Bayes doesn't have any hyperparameters to choose from and thus we just used it as it is. In the results section this will be labeled NB.

5.3.4 K-Nearest Neighbors

We also looked at the k-nearest neighbors algorithm, since it is non-parametric and measures distances, which our other algorithms don't do. As for tuning the hyperparameters, we first looked at n-neighbors - the number of neighbors compared with. Additionally, we looked at the parameter weights by testing both uniform weights and weights that make closer neighbors have a greater influence than further ones. In the results section this will be labeled KNN.

5.3.5 Support Vector Machine

Additionally, we decided to try using Support Vector Machines. This was done since they are *non-probabilistic* linear classifiers, and therefore offer an interesting alternative approach in comparison with the other probabilistic classifiers we have used. The hyperparameters that went into consideration with this algorithm were kernels and degree. For the kernel we chose between 'linear', 'poly', 'rbf', 'sigmoid', and 'precomputed' to calculate the kernel matrix. In addition, if the kernel was 'poly' then we compared different degrees of the polynomial using our typical exhaustive grid search method. In the results section this will be labeled SVM.

5.3.6 Random Forest

The last classifier algorithm we used was random forest. As an ensemble learning method for classification that constructs a multitude of decision trees at training time before outputting the class during classification. The hyperparameters that we chose to look at are max features and n-estimators. Max features is useful since a higher number of options are considered at each node. If this value gets too high however, the complexity of the algorithm increases too much and thus decreases the speed. As for n-estimators, when the number of trees increases then performance increases, but again makes the algorithm slower. As with the other machine learning algorithms we used an exhaustive grid search to find the best hyperparameters. In the results section this will be labeled RF.

6 Results

6.1 Pregame Prediction Results

As a baseline we found the results of our Naive Bayes classifier (based on all historical data from games) to be 89%. This means if we know the stats of a game then our confidence in predicting the winner is nearly 90%. Although we wanted to use this moving forward, we had issues with the new machine learning algorithms which prevented us from doing so. The different Gaussian models failed to converge, and prevented us from getting the results needed.

6.2 In-Game Prediction Results

We will now look at the results of different algorithms in predicting the outcome of a game given a random event, time remaining, and coordinate of event:

	Hits	Giveaways	Takeaways	Shots	Missed Shots	Blocked Shots	Goals	Penalty	Average
LR	0.508	0.535	0.522	0.520	0.544	0.668	0.542	0.626	0.545
DT	0.510	0.503	0.506	0.514	0.504	0.555	0.522	0.769	0.515
NB	0.515	0.537	0.522	0.523	0.539	0.668	0.531	0.577	0.544
KNN	0.495	0.502	0.516	0.512	0.515	0.598	0.494	0.477	0.517
SVM	0.487	0.536	0.528	0.523	0.538	0.668	0.519	0.517	0.539
RF	0.502	0.517	0.507	0.515	0.506	0.616	0.482	0.810	0.518
Average	0.503	0.522	0.517	0.518	0.524	0.629	0.515	0.629	0.530

Figure 3: Results for individual events

Since most results are around 50% this suggests that very little information is gathered when looking at a single event. Despite this, several results stand out. First, goals are far and away the best single event predictor of who wins. This is basic knowledge and it would be bizarre if this wasn't the case. Additionally, out of all of our machine learning algorithms logistic regression and Gaussian Naive Bayes performed the best (albeit still not very good) with the highest classification rate. Surprisingly, decision trees, k-nearest neighbors, and random forests all performed the worst with the lowest classification rates.

Now, we will add more features to our data to see if knowing more of the situation helps improve results. In addition to the features already listed, we also included the score of the game at the time of the event.

	Hits	Giveaways	Takeaways	Shots	Missed Shots	Blocked Shots	Goals	Penalty	All Events	Average
LR	0.590	0.539	0.456	0.533	0.529	0.567	0.696	0.626	0.534	0.567
DT	0.706	0.723	0.722	0.674	0.675	0.666	0.702	0.769	0.673	0.705
NB	0.570	0.538	0.495	0.545	0.541	0.552	0.699	0.577	0.540	0.565
KNN	0.501	0.514	0.505	0.517	0.509	0.523	0.609	0.477	0.502	0.519
SVM	0.514	0.493	0.522	0.529	0.523	0.532	0.696	0.517	0.499	0.541
RF	0.747	0.754	0.765	0.722	0.730	0.719	0.750	0.810	0.717	0.750
Average	0.605	0.594	0.578	0.587	0.585	0.593	0.692	0.629	0.577	0.608

Figure 4: Results given individual event and game scenario

These results are much more accurate at predicting a game's outcome. In particular, goals (which have been discussed previvously) and penalties are very useful in predicting a game's winner. This makes sense because in hockey players are removed from the ice when they commit a penalty, and thus the team must play with fewer players. It's also interesting to note that random forests predicted the winner of the game 81% of the time when the event was a penalty. In fact, random forests in general performed the best with an average classification rate of 75%. Decision trees were also successful with 70.5% classification rate. Note that these are very reasonable and are an increase from the "upper-bounded" pregame prediction of 62%.

6.3 Creation of In-Game Model

Using these results, we constructed an algorithm that calculates the in-game probability a team will win based on events. It combines the historical data of the teams, and of the results based on the given situation (such as score of the game). Then it calculates the probability in 10 second intervals by considering the historical data, game situation, and the events during that interval, with important events scaled accordingly. This is then iteratively inserted into our trained random forest classifier, where the probability is calculated by predicted number of wins over the iterations.

Consider the October 25, 2019 game between the Buffalo Sabres (away) and Detroit Red Wings (home). Plugging this game into our algorithm we get the following predictions along with detailed plots associated with certain events found in the figures below.

7 Analysis

For the live data, several results can be drawn. First, looking at isolated events is very difficult when predicting the winner of a game. While this may seem like common knowledge, it is still important to verify. Naive Bayes and Logistic Regression did the best, but were still only around 54% accurate. It is interesting to note that these two algorithms are both probabilistic in nature, which may have led to their success. Equally as fascinating was seeing just how poor decision trees and random forests were, especially considering how good they did later on when taking in the game scenario. This principle is an excellent application of the "no free lunch" theorem, which states

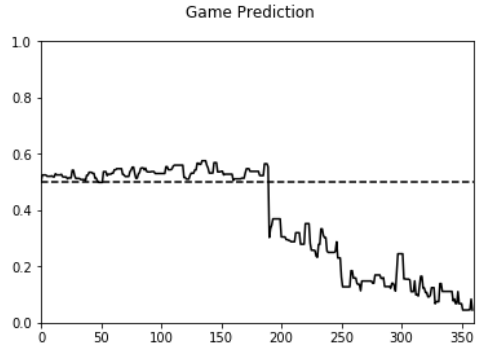

Figure 5: Probability throughout game

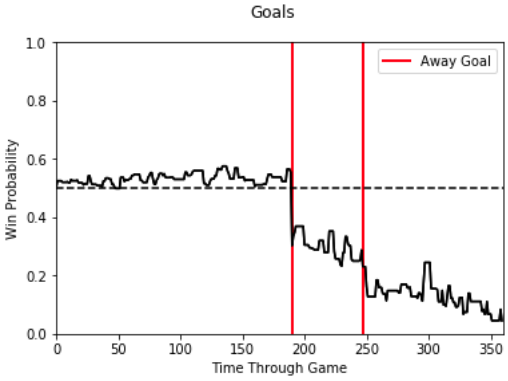

Figure 6: Goals throughout game

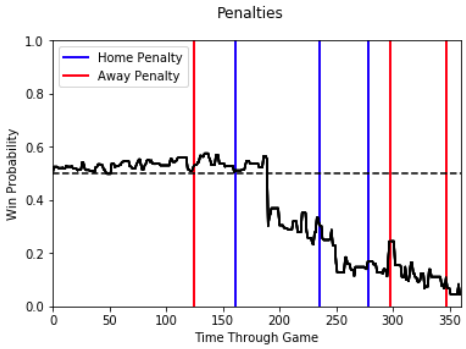

Figure 7: Penalties throughout game

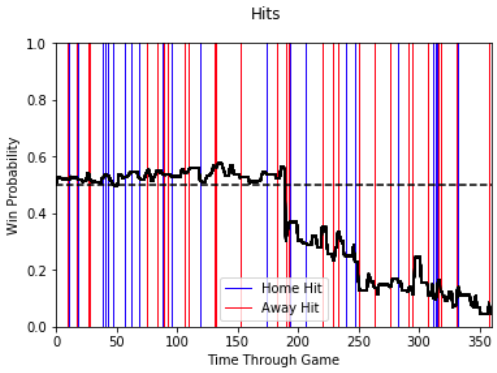

Figure 8: Hits throughout game

that any two optimization algorithms are equivalent when their performance is averaged across all possible problems.

We also briefly want to discuss why random forests did so well when given individual events and the game scenario. It is known that random forests perform better in higher dimensions. Basically when more information is known, better random forests are made. The same thing happened here where the given information was the exact thing needed to perform more accurately. However, the same cannot be said about k-nearest neighbors. Having an increased knowledge of the game scenario did not impact its results at all, with it's average performance staying around 52% for both sets of data. We find this fascinating that distances played very little in predicting success.

8 Conclusions

Calculating sport probabilities is hard - particularly in sport's as dynamic as hockey. Our attempts with several Gaussian processes were unsuccessful, but may confirm a limit on pregame predictions. While an upper bound my exist here, accuracy increases as the game progresses and more information is obtained. Classification algorithms, particularly random forests, can somewhat confidently predict the outcome of a match based on the knowledge of a single event and the game information. Using this, we constructed a model calculating in-game probabilities for hockey matches.

References

[Lew03] Michael Lewis. *Moneyball. The Art of Winning an Unfair Game*. W. W. Norton Company, 2003.

[Wei14] Joshua Weissbock. "Forecasting Success in the National Hockey League using In-Game Statistics and Textual Data". PhD thesis. 75 Laurier Ave E, Ottawa, ON K1N 6N5, Canada: University of Ottawa, 2014.

[Pre19] Associated Press. *Sports betting market expected to reach $8 billion by 2025*. 2019. URL: https://www.marketwatch.com/story/firms-say-sports-betting-market-to-reach-8-billion-by-2025-2019-11-04 (visited on 04/14/2010).