Puck Predictions: Unraveling the NHL Game Forecasting Riddle

Jason Vasquez Dylan Skinner Jeff Hansen
Benjamin McMullin

March 29, 2024

Abstract

The goal of this project is simple: predict the outcomes of NHL games from any given state. As simple as the problem statement is, however, the solution is not so straightforward. To solve this problem, we will use a variety of machine learning techniques, including logistic regression, XGBoost, and ARIMA models. Additionally, we utilize a form of MCMC to simulate the outcomes of games from any given state. Our hypothesis is that we will be able to successfully predict the outcomes of NHL games with a high degree of accuracy using these tools.

1 Problem Statement and Motivation

In the world of sports analytics, predicting the outcomes of games is a common and challenging problem, with live win predictions adding an extra layer of complexity. For most sports, there are a plethora of widely accepted—yet hidden—predictive models and methods that are used to predict games. In addition to this, most sports have easily accessible statistics and graphics that give current win probabilities for any live game.

Hockey, however, is a different story. While there are some methods used to predict the outcome of National Hockey League (NHL) games, these models typically belong to sport books and their nuances are not publicly disclosed. Additionally, hockey analytics is not as developed as it is in other sports, such as basketball or baseball. This lack of model transparency and public interest in hockey analytics makes predicting the outcomes of NHL games a very underdeveloped and challenging problem. Previous attempts and research into predicting NHL games has relied on methods such as

decision trees and artificial neural networks [3] (from 2014), naïve bayes and support vector machines [4] (from 2013), and Monte Carlo simulations [5] (from 2014).

In addition to model research, some research has also gone into developing new features that can be used to better predict the game outcomes. The two biggest engineered classes of features are the Corsi and Fenwick[1] metrics (both around 2007).

Our project seeks a similar outcome to the research mentioned above: predict the outcomes of NHL games. Not only this, but we seek to provide live, acturate win probabilities for any given game state. Despite the simplicity of the problem statement, as mentioned, the solution is not so straightforward. The NHL provides fast-paced games with many events occuring in quick succession. Our goal is to use this abundance of data and new approaches to build upon previous research.

Our motivation for this project exists strictly as fans of the sport and as data scientists. Our model is not intended to be used for gambling or any other nefarious purposes—any use of this model for such purposes is a misuse of our work.

2 Data

Our data came from the hockeyR Github repository [2]. This repository contains an abundance of data about every NHL game that has occured since the 2010-11 season. This data includes information about the events that transpire in a game (hits, shots, goals, etc.), which teams are playing, who is on the ice, and the final score of the game. The data is stored in a series of `.csv.gz` files, allowing for easy access and manipulation.

Each game in a season is given a unique identifier (`game_id`), which is constant across all events in a game. Every event that occurs in a game will be stored in the `event_type` column. There are 17 unique event types, including things such as game start, faceoff, shot, hit, and goal. Most of these event types are not relevant to our analysis, so we remove them from the dataset. After removing the unnecessary events, we are left with nine events: blocked shot, faceoff, giveaway, goal, hit, missed shot, penalty, shot, and takeaway. These events are attributed to the team and player that

[1] These metrics were created by sports bloggers Tim Barnes and Mark Fenwick, respectively. We were unable to locate the original blog posts talking about these metrics, but a good article to learn more about the math can be found here `https://thehockeywriters.com/corsi-fenwick-stats-what-are-they/`.

performs the event. We only take into consideration the team that performs the event and discard the player information.

The data also contains information about when the event occured. This appears in a variaty of formats, but we only use the `game_time_remaining` column. `game_time_remaining` starts at 3600 (60 minutes) and counts down to 0. If the game goes into extra time, i.e., it is tied after 60 minutes, `game_time_remaining` will be a negative value.

We found that our data did not contain any missing values that was not easily explainable. For example, if a game is starting, there will be no events for penalties, which will result in a `NaN` value in the penalties column. Additionally, any data that was confusing or not easily explainable (for example the home team having 7 players on the ice and the away team having 5), was manually verified by watching a clip of the game where the event occured to make sure the event was recorded correctly. We did not find any incorrectly recorded events, so we did not remove any strange events from out dataset.

3 Methods

3.1 Bayesian Network

We first used a Bayesian Network to establish a benchmark for probability using several key features.

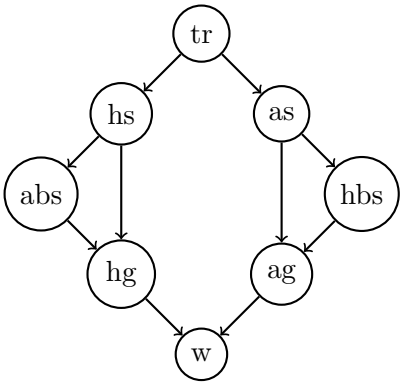
Bayesian networks are probabilistic graphical models that represent probabilistic relationships among a set of variables using a directed acyclic graph (DAG). In a Bayesian network, nodes represent random variables, and directed edges between nodes represent probabilistic dependencies between the variables. Each node in the graph is associated with a conditional probability distribution that quantifies the probability of that variable given its parent variables in the graph.

For our purposes, we predefined the structure of the network, and used the data to calculate the conditional probabilities for each node. We then used the network to calculate the probability of a team winning given the current state of the game.

The computational complexity of Bayesian Network inference is high, with exact inference being an NP-hard problem [1]. Using the python package pgmpy, we originally tried to fit a network with all 26 of our features, but our computational resources failed to fit this network. Then, to get a baseline for our future predictions, we simply fitted the model with the base features of time remaining (tr), home goals (hg), away goals (ag), home

shots (hs), away shots (as), home blocked shots (hbs), and away blocked shots (abs) in order to predict wins (w). These features were chosen as priors because of our opinion that they are the most important to the game, based upon our knowledge of hockey.

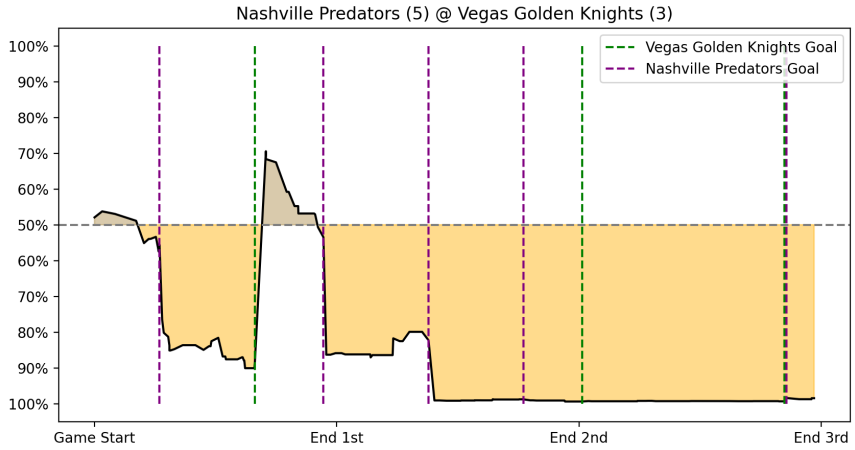
The conditional dependencies of the chosen network are shown in the DAG below:



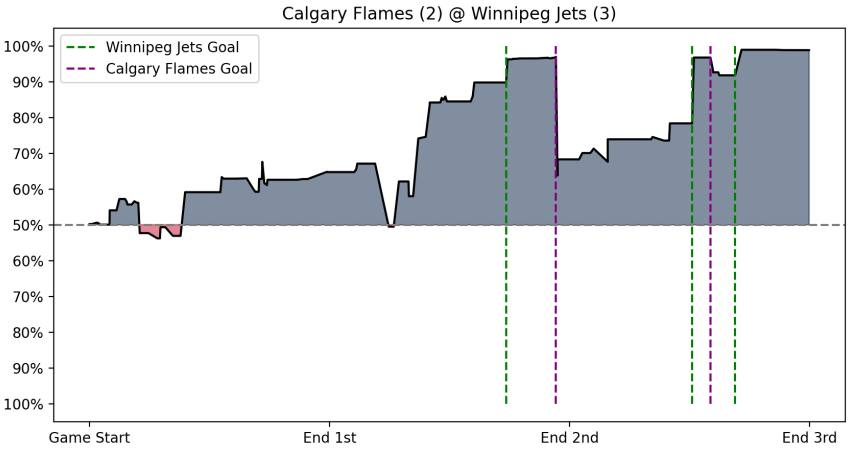
This model was chosen for the task because different stats in hockey are conditionally dependent of each other, so by modeling those conditional dependencies and feeding them into the model, we can hopefully acheive a more accurate prediction of the outcome of the game.

3.2 XGBoost

In our NHL analysis research, we partitioned the dataset into segments corresponding to individual teams' games over multiple seasons. This enabled the creation of time series data in the form of a state vector, capturing the play-by-play dynamics of each team's matches. We then trained separate XGBoost models for each team to learn their unique patterns and strategies. Leveraging the `.predict_proba()` method of XGBoost, we generated probabilistic predictions at various stages of a game, allowing us to plot the evolving probability of each team winning throughout the match (see 1). This approach provided insights into momentum shifts (when one team has an advantage due to them having more players on the ice due to a penalty by the opposing team), key moments (such as goals and penalties, and critical plays influencing game outcomes).



(a)



(b)

Figure 1: For both of these plots, the home team is above the the 50% line, with the away team below. The title has the final score of the game.
(a) The live win probabilities for the Nashville Predators at the Vegas Golden Knights. Notice the probability shifts after one team scores a goal.
(b) The live win probabilities for the Calgary Flames at the Winnipeg Jets. Notice the stark changes in probabilities after each goal. Also notice the shift in probabilities even when no goals are scored. This suggests the XG-Boost model is able to pick up on momentum shifts.

3.3 MCMC Game Simulation

To simulate hockey games, we created a Markov Chain where the states are a tuple of three consecutive events that occurred. For example, if the home team won a faceoff, the home team lost the puck, and then the away team shot the puck and missed, the markov chain would look like figure 2.

	curr_state	next_event	probability
39970	FACEOFF_HOME,GIVEAWAY_HOME,MISSED_SHOT_AWAY	SHOT_AWAY	0.1853207
39969	FACEOFF_HOME,GIVEAWAY_HOME,MISSED_SHOT_AWAY	BLOCKED_SHOT_AWAY	0.1138666
...	...	...	...
39981	FACEOFF_HOME,GIVEAWAY_HOME,MISSED_SHOT_AWAY	PENALTY_AWAY	0.0167210
39976	FACEOFF_HOME,GIVEAWAY_HOME,MISSED_SHOT_AWAY	GOAL_HOME	0.0085793

Figure 2: Sample Markov Chain for the current state (home team won faceoff, home team lost puck, away team missed shot) with the probabilities of the next event sorted from highest to lowest

The probabilities of transitioning from one triple-state to another triple-state is calculated by:

$$P(s_{t+1} = (B, C, D) \mid s_t = (A, B, C)) = P(D \mid (A, B, C)) = \frac{\{\text{\# of times (A,B,C,D) happend}\}}{\{\text{\# of times (A,B,C) happened}\}}$$

Where A, B, C, D represent events that can occur in a game, and the tuple (A,B,C,D) represents that "A then B then C then D" happened right after each other in a game.

To simulate a game, we performed this Monte Carlo algorithm that acted like a random walk through the Markov Chain:

Algorithm 1 Simulation Algorithm

time_remaining $\leftarrow$ 3600 ▷ NHL games are 3600 seconds
$s_0 \leftarrow$ "#" ▷ The "#" symbol represents the start of a game
$s_1 \leftarrow$ "#"
$s_2 \leftarrow$ "#"
while time_remaining > 0 **do**
    next_state $\leftarrow$ sample from MC(curr_state)
    $s_0 \leftarrow s_1$
    $s_1 \leftarrow s_2$
    $s_2 \leftarrow$ next_state
    event_time $\leftarrow$ sample from KDE_times()
    time_remaining $\leftarrow$ (time_remaining $-$ event_time)
end while

The KDE_times() is a KDE model fit on the amount of seconds that transpired between each hockey event for all NHL games over the course of 13 years. This KDE model resembled a poisson distribution with $\lambda \approx 18$. During the course of the algorithm, we maintained a count of all of the events that transpired. Upon termination of the simulation, we compared home goals to away goals and termed a winner. If there was a tie, we would run the algorithm until another team scored0 a goal, thus simulating overtime per NHL rules.

To predict a probability of a certain team winning a hockey game, we would run 50 simulations with our algorithm, but we would initialize the starting states (s_0, s_1, s_2) to be the most current events in the hockey game. Out of the 50 simulations, we would compute $P(\text{home winning}) = \{\#\text{ home simulation wins}\}/50$ and $P(\text{away winning}) = 1 - P(\text{home winning})$.

To evaluate our simulation's effectiveness at accurately modeling actual hockey games, we combined a dataset with the final event counts for 1500 actual hockey games and for 1500 simulated games. We then performed PCA, t-SNE and UMAP dimensionality reductions using various perplexity and neighbor hyperparameters to see if these algorithms clustered the synthetic games and actual games in different clusters. As shown below in 3, the simulated games and actual games are all clustered together, thus demonstrating that our simulation effectively emulates live NHL games.

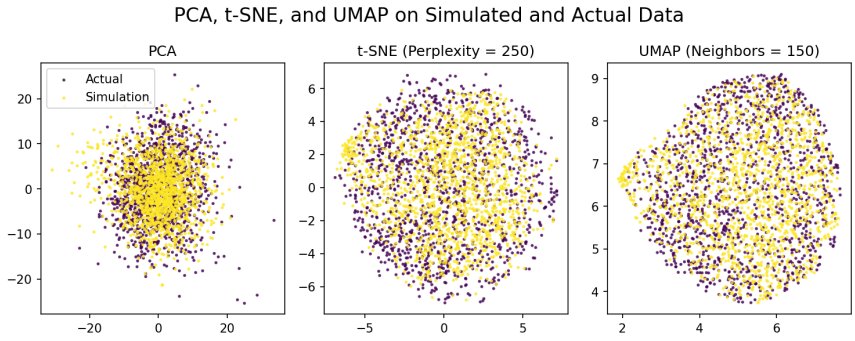


Figure 3: PCA, t-SNE, and UMAP performed on a dataset with final event counts for 1500 simulated and 1289 actual NHL games. The joined grouping demonstrates that our simulation accurately emulates live NHL games

4 Results and Analysis

4.1 Bayesian Network

Overall, the Bayesian Network performed well, and was able to produce realistic win probabilities. Because it defined a joint distribution using a DAG and then used that distribution to fit the data, it was able to correctly predict accuracies using the few features provided. However, the Bayesian Network struggled to capture the intricate dependencies of factors other than goals that could affect the win probabilities, so the predicted probabilities are little more than an over probability calculation of goals and time reamining given historical data. We see this in the probability graph below, where the changes in probability correspond to the goals scored in the game.

5 Ethical Considerations

Predicting win percentages or outcomes in hockey games, like any sport, raises several ethical considerations. Here are some key points we want to address:

- **Gambling and Addiction**: Our win percentages and predictions might be used by those who wish to gamble, which could lead to addiction and financial harm, especially if undue trust in placed in these methods. Any publication of these methods or predictions would be

accompanied by promoting healthy and responsible gambling practices.

- **Fairness and Integrity of the Game**: Sometimes, coaches and players becoming aware of their chance of winning can affect how the game is played, potentially harming the integrity of the sport. We must be careful to not provide an unfair advantage to any team or player. Inaccurate predictions could lead a team to believe that the game is out of reach when it isn't, and we want to avoid that.

Overall, our predictions, like any, should not be considered declarative for gambling or performance purposes, but are rather an interesting exposition into the complexity of sporting events.

6 Conclusions

References

[1] David Maxwell Chickering, Dan Geiger, and David Heckerman. Learning bayesian networks: Search methods and experimental results. In Doug Fisher and Hans-Joachim Lenz, editors, *Pre-proceedings of the Fifth International Workshop on Artificial Intelligence and Statistics*, volume R0 of *Proceedings of Machine Learning Research*, pages 112–128. PMLR, 04–07 Jan 1995. Reissued by PMLR on 01 May 2022.

[2] Dan Morse. hockeyR-data: A collection of hockey datasets for use with the hockeyR package. `https://github.com/danmorse314/hockeyR-data`, 2024. Accessed: March 2024.

[3] Gianni Pischedda. Predicting nhl match outcomes with ml models. *International Journal of Computer Applications*, 101:15–22, 09 2014.

[4] Jason Weissbock, Herna Viktor, and Diana Inkpen. Use of performance metrics to forecast success in the national hockey league. In *European Conference on Machine Learning: Sports Analytics and Machine Learning Workshop*, 2013.

[5] Josh Weissbock. Forecasting success in the national hockey league using in-game statistics and textual data. 2014.