Because It's the Cup: Predicting the Stanley Cup Playoffs

Mason Swofford, Shuvam Chakraborty, Vineet Kosaraju

Introduction

The National Hockey League (NHL)'s Stanley Cup Playoffs, has long been known for its unpredictability. In this study, by using game data and learning techniques and analyses that are not commonly used we will attempt to find if game prediction as applied to playoff outcomes can be accurately ascertained. In order to do so, our project has two main goals. The first, referred to hereafter as Goal 1, uses a variety of features that measure two hockey team's performances upto a point in a season (discussed in *Dataset and Features*), trains a classification model to predict the winner of the game between those two teams, and uses this to specifically predict the result of playoff series using a binomial distribution. Our second main objective, referred to as Goal 2, attempts to use our classification model and the probabilities of wins it learns to construct a gambling agent to maximize the probability of a minimum reward when betting on playoff games, in order to determine if hockey can successfully be bet upon. As we are working on this project jointly for CS221, we share the datasets and overall problem statement, but we focus more on Goal 1 for CS 229, i.e. the model we construct and the error analysis we perform, and Goal 2 for CS 221.

Related Work

Advanced statistical analysis and machine learning techniques have a recent precedent in several sports other than hockey. For instance, in American Football, researchers have used neural networks to predict NFL game outcomes with up to 75% accuracy (Kahn). In basketball, support vector machines were successfully used to predict regular season games with more than 86% accuracy (Yang et. al.). However, much of hockey performance prediction still remains outdated, primarily due to a scarcity and lack of interest in data compared to the NFL/NBA. For example, a paper published in 2016 trained a machine learning model not for predicting games, but for determining which experts were best at predicting game results, and then using those experts to predict game results (Gu). There have been a few analytical approaches to analyzing the overall sport of hockey. Researchers at the University of Cincinnati created a regression to rank player's contributions to their team's success (Morgan et. al.), while more recently, a paper published by researchers in the University of Ottawa (Weissbock et. al.) did attempt to predict hockey game results based on team stats, but their work has several limitations that we will improve on. Specifically, their work does not include important features, such as shot quality, and they only use data from a portion of one season, whereas our data will come from several seasons, multiplying the amount of data by almost a factor of 10.

Goal 1: Dataset and Features

We created two main datasets for our project: a training set, consisting of regular season games, and a test set, consisting of playoff games, because predicting playoff games was the ultimate goal in our project. Our data for this project was scraped from three main sources: 1) playoff results reported by the NHL, 2) daily team stats scraped from corsica.hockey, and 3) Vegas Odds for every NHL playoff game. The first and third sources gave us specific game information, such as the teams playing, the score, and the betting odds. The second source gave us statistics for each team on a daily basis, which we had to consolidate. By the end of our scraping and processing, we created a training and test dataset including thousands of games, where each example consists of 1 game result. The features for each example consists of the team stats averaged over the season, upto that game, while the label is 0 or 1, where 0 represents the away team winning, and 1 represents the home team winning, or in the case of softmax regression, the goal differential with respect to the home team. Table 2 gives a preliminary collection of variables, though we later collected a larger set after performing initial error analysis (discussed more in *Experiments, Results, and Discussion*).

Feature	Description	Feature	Description
CF	Corsi For, shot attempts for a team,	CA	Corsi Against, shot attempts against a team, including

	includes blocked, and not on goal		blocked and not on goal shots.
GF	Goals For	**GA**	Goals Against
xGF	Expected Goals For, based on quality of shot attempts for	**xGA**	Expected Goals Against, based on quality of shot attempts against

Table 2: Basic features, consisting of simple statistics that analyze shot attempt and quality. Not all features shown.

Goal 1: Methods

To solve the binary classification problem for our first goal of predicting game results, we developed several categories of models, including a logistic regression, support vector machines, and artificial neural nets, and we implemented all these features through python's sklearn module. We evaluated these models on our training set using 10-fold cross-validation and results are presented in the *Experiments, Results, and Discussion* section. A more detailed description of these models and our reasons for using them follows:

Logistic Regression and Softmax Regression:

A logistic regression model models the data as a Bernoulli distribution with parameter being the logistic function (Eq 1), which is derived by modelling the Bernoulli distribution as an exponential family distribution and deriving the link function. The weights vector is trained by maximizing the log likelihood over all training examples (Eq. 2). We used logistic regression to obtain the probability of a team winning a hockey game given their stats, and their opponent's stats. For softmax regression, the data is modelled as a multinomial distribution, and once the weights vector is trained, which is done in sklearn by minimizing the cross entropy loss, class probabilities are given by the softmax function, which can be derived from the link/response functions of modelling a multinomial distribution as an exponential family distribution. We used softmax regression on a modified training set (with the labels being the goal differential for each game) and tested whether it could accurately predict which team was the winner, where the winner was decided by comparing the combined probabilities of the positive class labels and the negative class labels.

$$h_\theta(x) = g(\theta^T x) = \frac{1}{1 + e^{-\theta^T x}}$$

$$\begin{aligned} \ell(\theta) &= \log L(\theta) \\ &= \sum_{i=1}^{m} y^{(i)} \log h(x^{(i)}) + (1 - y^{(i)}) \log(1 - h(x^{(i)})) \end{aligned}$$

Eq. 1 (Left): Logistic function, used to calculate p(y=1|x). Eq. 2 (Right): Log likelihood for logistic regression over a training set. This is maximized using gradient descent/Newton's method to determine the weights specified in Eq. 1.

Support Vector Machines:

Support Vector Machines attempt to classify data through a decision boundary that maximizes the minimum distance of any training example to the boundary(concept of margin). Specifically, SVM's attempt to maximize the geometric margin and solve the constrained optimization problem. In practice, this is done by solving the derived Lagrangian dual problem using the SMO algorithm. Due to our high dimensional feature set, we applied a variety of Kernels, such as Gaussian, Polynomial, and Sigmoid. Similar to logistic regression, we used SVM's to derive a probability for a certain team winning the game, which were obtained through the python sklearn framework, which uses Platt scaling to convert margins to probabilities.

Neural Networks:

Our final category of models was artificial neural nets. Due to our primitive knowledge of neural network theory, we varied hidden layers and network size to match other papers in sports, while for activation functions we tried several (logistic, relu, tanh). These activation functions enable the network to map the input to different domains, as logistic is between 0

and 1, relu is 0 and inf, and tanh is -1 to 1. We experimented with various combination of hidden layer sizes and activation functions to see which produced better results, as again our intuition was not strong for neural networks.

Goal 2: Methods

Regarding our second goal, the gambling problem was formulated as a Markov Decision Process (MDP). Each state encodes the money we currently have and the game number (e.g. if there are 4 games to be played the domain for game number is {1, 2, 3, 4, 5} where 5 is the end state). The starting state is game 1 with current money equal to the user specified starting money. End states are reached when it runs out of money or reaches the last game. Our actions are the amount of money to bet (discretized into constant amounts, set by the user), and the team we are betting on. The amount of money won for an action is derived from the Vegas odds. Our transition probabilities are given by our predictions from our model from Goal 1, and we set the discount equal to 1, as we are not focused on earning the money quickly within a day. Finally, we set our reward to 1 if we reach an end state and we have at least the desired amount of money, and we set it to 0 otherwise. Since the MDP was acyclic, it was solved using dynamic programming, and we decided the optimal policy is one that is the most likely to yield a certain return or higher. For example, if we start with $1000, we might say we want to find the policy that will yield at least $1200 with the highest probability. Note that this is most likely not the policy with the highest expected value. Both the starting amount and the desired amount are parameters configurable by the user.

Experiments, Results, and Discussion

In this section we present the results of our various models: as noted before, reports come from 10-fold cross validation, with the training and development error equal to the average over all 10 folds. Further, the main metric used to evaluate different models was accuracy (the number of correct classifications divided by the total). Initially, with our preliminary results, we reached a low performance of around 0.56. These low accuracies are on the training sets, indicating a problem with high bias, as the models are underfitting and are unable to generalize relationships between our features. For most of the models, the training error was almost equivalent to the training-development (validation) error, indicating that the variance is relatively low. However, these initial results don't tell us whether the main problem with our low accuracies can be attributed to our choice of model, or our choice of features. In order to attribute error, we performed an ablative error analysis on the features, removing each one by one, to see what impact each feature had on the results.

One striking result from this error analysis was that explicitly removing one feature from our feature set didn't seem to markedly change the performance of our model, potentially indicating that none of our features were particularly important in our model accuracy. This indicates that the main problem was with feature selection, as none of the features were essential. To further investigate this issue with our features, we analyzed the weights learned for each of the features from our logistic regression model with the background of an intuitive understanding of what those features mean to a hockey expert. For features whose sign mirrored each other, and the signs of the features were also appropriate (i.e. if this feature indicated a positive outcome for a team, its weight should be positive for the home team and negative for the away team and vice versa), it seems the model learned the significance of these features correctly. These features included Goals For/Against/Differential and Corsi Against. On the other hand, for the other features, these features did not correspond to winning games as strongly, so they did not seem to receive weights that intuitively would make sense. This is likely an issue with either our choice of model, as potentially the logistic regression is not powerful enough to classify the complex multi-dimensional feature set, or our choice of features, as previously indicated by the ablative analysis. To remediate these two issues, we extended our feature set to a more advanced collection, and trained new models, including experimenting with artificial neural networks. These new results are presented in Figure 2.

We note that with our advanced set of features and more complex models, we achieve slightly higher accuracies, peaking at 0.58, which is not a substantial improvement from our previous attempt. Not shown are the results of varying different hyperparameters of our models, which achieved comparable performances. Further, as we suspected that our data was too high-dimensional to be accurately learned, we attempted using PCA to simplify the feature space to a few principal components, but this achieved slightly lower accuracies. In order to better understand our models, we plotted

a confusion matrix to investigate where the majority of our errors were coming for our models. One such confusion matrix is in Figure 3, for our ANN (h=5/10, identity) model; the other confusion matrices were similar. This confusion matrix indicates that the main source of error in our model is with false positives where we predict the home team to win, but the visitor team actually wins. This error is likely a result of the models learning to favor home teams when it is less confident about the results (effectively learning the "home field advantage" common to several sports). We see that this intuition about the model's source of error is validated by Figure 4, which demonstrates that our model is less accurate the more close a game is, and the less confident it is.

In Figure 4, we note that the model is more accurate the more confident it is (when it predicts that games are less close), and when games are really close (the probability of a team winning is less than 0.1), it is often wrong. Since there is an abundant amount of data present when our model is less confident, it is possible that training a more fine-grained classifier on this subset of the data might improve the accuracy of our model. Another potential source of error was the dataset itself; as noted by Weissbock et. al. hockey might have a theoretical limit for predictions of 60% as it is an inherently difficult sport to predict due to the low number of goal events. To verify this theoretical threshold, we collected statistics about teams' winning percentages for an entire season of regular gameplay and determined the variance of this distribution. We then conducted several Monte Carlo simulations of a season, pitting teams against each other, calculating winning percentages, and determining the variance of these simulated distributions. In these simulated team matchups, teams either won due to "skill", which were randomly assigned and stayed constant throughout a season, or due to "luck", which was determined by a coin toss. From these simulations, we see that when luck accounts for 73% of games, the simulated distribution most closely matches the observed one. As such, a perfect model would be able to successfully predict the 27% of game results accounted for by skill, and half of the games accounted for luck, reaching an accuracy of 63.5%, confirming the theoretical limit mentioned in previous literature. Although this theoretical limit applies to regular season games, we still attempted to apply our model to playoff games. When doing so, we found out that our model had a 54.66% accuracy on our test set of playoff games. For reference, ESPN experts were 51% accurate (ESPN). Although this is only a small increase in accuracy, because Vegas betting odds are similar in accuracy to ESPN, this small increase helps when betting, as discussed below in *MDP Results*.

To investigate this cause of variance between training and test sets, we looked at how the quality of teams playing in a game impacted our model's accuracy. We see that our model performs best when one team is categorized as "good" and one team is categorized as "bad" (0.595 accuracy); this is likely either because this scenario is easy to predict, or because we have three times as many games for this scenario. However, we notice that when both teams are "good", as is the case in playoff games, our model performs with the least accuracy (0.5588 accuracy). As such, it is possible that we could improve our results by training a specific model on a subset of our data for when both teams are "good" teams, as those are the most common matches in playoffs. We confirmed this assumption by training and testing a SVM on playoff games specifically and were able to achieve 70% validation error, indicating that the theoretical threshold could be higher for playoff games.

MDP Results

Despite our low accuracy in goal 1, our MDP was able to achieve novel results. With a starting amount of $1,000 dollars, desired amount of $1,200, and discretization of $50, we were able to achieve the desired amount in 91% of simulations with at least 3 games. Using our testing platform, we dove into our MDP to understand why it was able to achieve significant returns. We noticed that our MDP was sometimes making unorthodox bets on teams which our model predicted to lose. However, these teams were only slight underdogs in the eyes of our model (e.g. 45% chance of winning), while the vegas odds had them pegged as huge underdogs. We see from this that the primary benefit of our combined prediction to MDP pipeline is that we received a probability estimate that each team wins; so, even if we incorrectly predict the winning team, we can approximate how certain we are each team will win, and take advantage of experts' overconfidence in certain teams, which is reflected in the vegas odds. Again simulating on days with at least 3 games and a starting amount of $1,000, we were able to achieve a desired amount of $1,600 82% of the time.

Conclusions and Future Work

In conclusion, through an extensive process of iteratively improving the performance through error analysis techniques, we were able to reach slightly under a 60% accuracy on predicting regular season games. This low accuracy is likely due to the fact that hockey is a very challenging sport to predict, due to variability inherent in the sport, as illustrated in our Monte Carlo simulation. In theory, perfect data and the perfect model could allow reaching the theoretical accuracy limit around 60-63%. These additional statistics could come from analyzing specific player injuries throughout the season, analyzing how player performance changes, and using the travel time and home vs away field advantage as factors. Furthermore, our error analysis shows that predicting certain matchups by training on games featuring similar teams could produce better results. Our model reached a lower accuracy on predicting playoff games, but it still outperformed the predictions performed by experts on ESPN. This slight improvement in predictions along with having probabilities for each game, allowed our gambling agent to successfully gamble off of hockey playoff games. Ultimately, despite low accuracies, relative to the theoretical threshold we successfully made progress towards our first goal and made novel advancements in the field of hockey predictions and sports gambling.

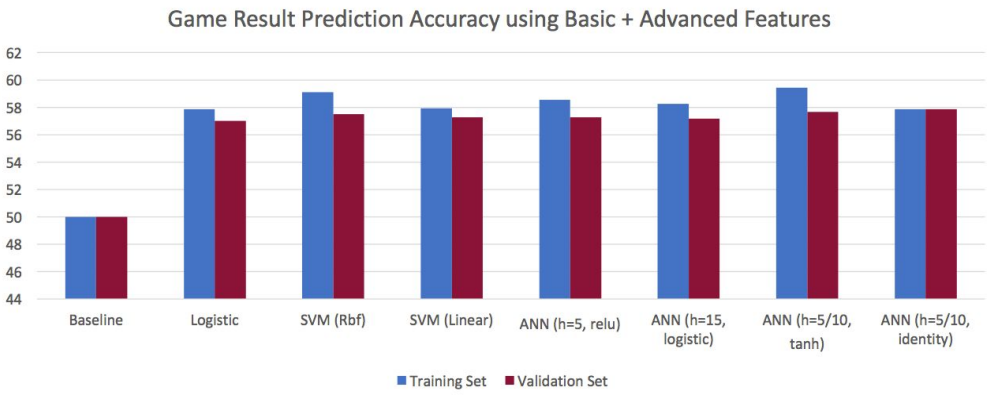

Figure 2: Accuracy of various game result prediction models using our new set of advanced features (in Table 3, Appendix). The accuracy on the validation set increased to 0.58.

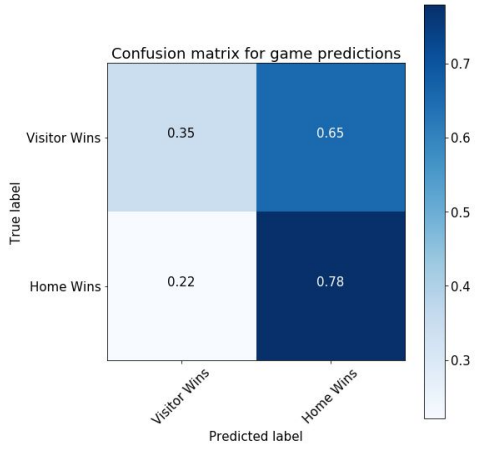

Figure 3: Confusion matrix for ANN (h=5/10, identity). The visualization indicates that the main source of error is false positive (where positive refers to prediction of the home team winning).

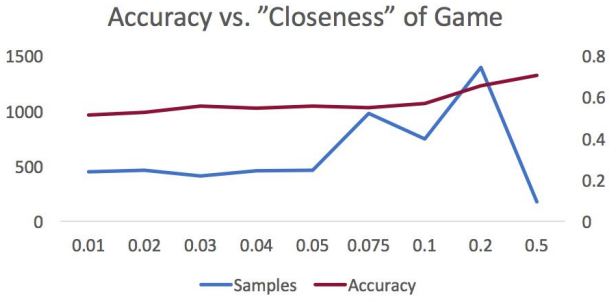

Figure 4: Accuracy (red) and number of samples (blue) versus closeness of game for our model's predictions. Closeness of game on the x-axis is represented by the numerical difference between our model's predicted probability of winning and being completely unsure (0.5).

Contributions

All members in the group worked on the above project equally, as we each divided up the work for each of the various steps of our project. For instance, since our data collection and preprocessing step consisted of data from three sources, we each took the task of collecting data from one source, and worked on amalgamating them together. Similarly, we each implemented several different models for our first goal. Each error analysis step was discussed as a group and then worked on as individuals. For our second goal, we formulated the general idea as a group, with the help of TAs in office hours, and then split it up into three subproblems: concrete formulation, strategy optimization, and results/error analysis.

References

Kahn, Joshua. Neural Network Prediction of NFL Football Games. 19 Dec. 2003, homepages.cae.wisc.edu/~ece539/project/f03/kahn.pdf.

Yang, Jackie B, and Ching-Heng Lu. PREDICTING NBA CHAMPIONSHIP BY LEARNING FROM HISTORY DATA. Proceedings of Artificial Intelligence and Machine Learning for Engineering Design.

Gu, Wei, et al. Expert System for Ice Hockey Game Prediction: Data Mining with Human Judgment . International Journal of Information Technology & Decision Making, www.worldscientific.com/doi/abs/10.1142/S0219622016400022?journalCode=ijitdm&

S Morgan, and C Barnes. Applying decision tree induction for identification of important attributes in one-versus-One player interactions: a hockey exemplar. www.ncbi.nlm.nih.gov/pubmed/23409787.

Hipp, Adam, and Lawrence J Mazlack. Mining Ice Hockey: Continuous Data Flow Analysis. IMMM 2011 : The First International Conference on Advances in Information Mining and Management.

Weissbock, Joshua, et al. Use of Performance Metrics to Forecast Success in the National Hockey Leagu. www.ecmlpkdd2013.org/wp-content/uploads/2013/09/mlsa13_submission_2.pdf.

www.nhl.com

www.corsica.hockey

www.sportsbookreviewsonline.com/scoresoddsarchives/nhl/nhloddsarchives.htm

http://scikit-learn.org/stable/

http://www.espn.com/nhl/story/_/id/19121664/nhl-2017-stanley-cup-playoffs-first-round-predictions

http://www.espn.com/nhl/story/_/id/19229538/2017-stanley-cup-playoffs-second-round-predictions

http://www.espn.com/nhl/story/_/id/19466503/2017-stanley-cup-finals-experts-predictions