Using Machine Learning Methods to Predict Winning Teams in the National Hockey League

SENG 474, University of Victoria

Anar Kazimov - V00884552

Bryce Mennie - V00909758

Evan Woronuk - V00899058

Jodhan Ghangheri - V00851957

Noah Serr - V00891494

Tommy Lay - V00855688

Abstract

Sports betting has become immensely profitable in the past few years which has skyrocketed the popularity of machine learning for the purpose of predicting winning players and teams. However, while there are numerous existing studies for solo sports and other team sports leagues, relatively little attention has been given to NHL hockey. Our approach uses metrics such as outcome, goal differential, and shot differential, among others as well as two popular classification models, Logistic Regression and Neural Networks, to find an optimal classification model. Our results show that Logistic Regression was the most accurate for this purpose, which is parallel with similar studies.

1 - Introduction

The sports betting industry has seen rapid growth in recent years due to its profitability and increasing legality (Casino, 2020). When making bets, it is incredibly important to have confident and accurate estimations before placing wagers down. While there have been many previous studies using machine learning methods to predict the outcome of both individual and team sports, there have been relatively few for hockey. One of the biggest challenges in predicting hockey games is the lack of scoring events (Weissbock). The goal of this study is to evaluate to what degree we can successfully and accurately predict game outcomes, utilizing two common classification methods in machine learning. Our baseline for successful results relies on the accuracy of our predictions being superior to the 50% guessing chance. If we fail to surpass this baseline, then our system fails to provide any meaningful benefit or advantage to gamblers and the sports betting industry. Additionally, the p-value calculated for the results needs to be at least below 0.05 for our predictions to be considered statistically significant.

1.1 - Problem Statement

Assume t_1 and t_2 are two opposing NHL teams. Both teams have a set of games they have already played against other teams. Using the statistics from previous games (explained in detail in the data collection section), the system has to predict who will be the winner of a specific game. Given the sample size (n) of 1082 which represents the number of unique games in the season, a p-value less than 0.05 as our threshold for the system to be legitimate, and the null hypothesis that the system predicts by chance ($p_0 = 0.5$), we have calculated the accuracy that we need from the system as given below:

Given:

P-value < 0.05 → P-value ≈ 0.4960 → z = 0.01
n = 1082
$p_0 = 0.5$

$$z \times \sqrt{\frac{p0\ (1-p0)}{n}} + p0 = 0.01 \times \sqrt{\frac{0.5\ (1-0.5)}{1082}} + 0.5 = 0.500152004$$

Formula 1: Accuracy for training data

As we can see the minimum accuracy we need for the system to produce statistically significant predictions is 50.02% for training data. For test data with n = 216:

$$z \times \sqrt{\frac{p0\ (1-p0)}{n}} + p0 = 0.01 \times \sqrt{\frac{0.5\ (1-0.5)}{216}} + 0.5 = 0.500340206$$

Formula 2: Accuracy for test data

Thus for the predictions the system will make on the test data the accuracy will have to be at least 50.03% for it to be considered statistically significant.

2 - Background Information and Related Work

2.1 - Basics of NHL Hockey

NHL hockey sees two opposing teams with five players at a time try to score goals against one another by shooting a rubber puck using a wooden or composite stick into the opponent's net while skating on a bed of ice. Standard games consist of three 20-minute periods, as well as a 15 minute intermission after the first and second periods. In the case of a tie, an additional 5 minute sudden death overtime period is played where each team has three players on the ice at a time, and the first team to score wins the game. If neither team scores in the 5 minutes, a shootout commences where each team takes turns sending out one player to attempt to score on the opposing goaltender. If the shootout score is level after three shooters have gone from each team, it turns into a sudden death format to finally decide the victor.

2.2 - NHL Case Study

Weissbock's "Use of Performance Metrics to Forecast Success in the National Hockey League" case study concluded that between Neural Networks and Support Vector Machines, Neural Networks were most accurate at predicting outcomes of NHL games with an accuracy of 59.38%. However, Logistic Regression was never explored in this study. This meant we would not have any direct NHL comparisons for our regression results. The data used in the Weissbock study was also a much smaller sample size than what we used for our project.

2.3 - NFL Case Study

Bouzianis' "Predicting the Outcome of NFL Games Using Logistic Regression" study found that Logistic Regression could be up to 70% accurate at predicting the outcomes of NFL football games. While this may initially appear promising to our study, it is important to note that there are many more scoring events in football - between touchdowns, field goals, and even touchbacks, football becomes much easier to predict than hockey which has just one scoring event, goals.

3 - Approach

3.1 - Data Collection and Usage

We decided to limit our data to a single season (2019-2020) as team composition can change substantially from season to season, and we did not take that into account. QuantHockey.com was an excellent resource for gathering data as they provide game logs for each game and each team throughout each entire season. These were easily accessible in spreadsheet format, so we created a sheet for each team's game logs for the season. The data was then reduced to use only the metrics we saw fit for our project. This included date of game, location (home or away, as home teams are often favoured), outcome (win or loss), goal differential (goals for minus

goals against), shot differential (shots for minus shots against), shot percentage (number of goals over number of shots), faceoff percentage (faceoffs won over faceoffs lost), and save percentage (number of saves over number of shots). At each game, these statistics were calculated for all games up until that date, so we were able to analyze a team's performance throughout the played season for every game. These metrics also allowed us to look at multiple different facets of a team's comprehension including offence, defence, goaltending, and fatigue during away games.

This data still had to be processed and cleaned to transform it into a usable format. We constructed a parser which processes each game from each team and consolidates a team's statistics at a given point in the season. We created objects for each game where the differential statistics were from the perspective of the home team, and after parsing through each game and collecting these metrics, all game data was appended to a pandas dataframe. This allowed for us to have a clean and easily manipulatable set of data to use as the input for our classification methods.

4 - Results

Our initial results for both classification methods are as shown in the table below. These results prove to be promising with higher accuracy than we had originally predicted, even before hypertuning for both. With Logistic Regression, it seems to be underfitting the data resulting in a higher accuracy for testing than training. However, the Neural Network displayed a fair amount of overfitting demonstrated by the higher training accuracy compared to the test accuracy.

We used the Sklearn libraries for both of the methods.

Classification Method	Training Accuracy	Testing Accuracy
Neural Network	0.6597	0.6050
Logistic Regression	0.6239	0.6281

Figure 1: Accuracy for both classification before hyperparameters

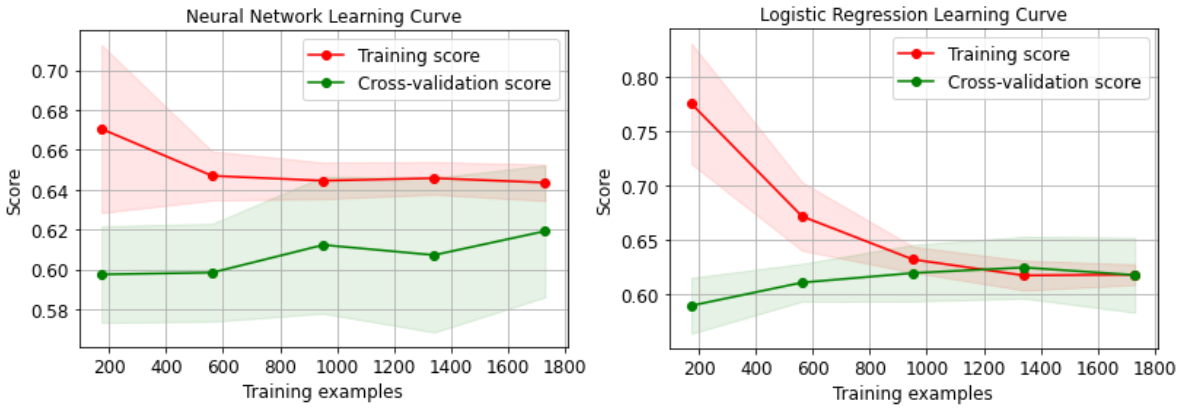

Figure 2: Neural Network and Logistic Regression accuracy before tuning

4.1 - Hyperparameter Tuning

For each classification, we used GridSearchCV method with K-fold cross-validation to search over these specified parameters and help determine the ideal parameters for each model. In K-fold validation, the nested loops iterated through the variety of parameters and calculated the K-fold accuracy score until the set of parameters producing the best accuracy results was decided. We set K=5 to follow the recommended 80/20 split. This means that at any given point, we would have 4 training sets and 1 validation set. The hyperparameters we derived can be found in the tables below. Figure 3 shows the accuracy of both methods after finding the optimal parameters. We were also able to rectify the underfitting for Logistic Regression and increase the testing accuracy for Neural Networks.

Classification Method	Training Accuracy	Testing Accuracy
Neural Network	0.6320	0.6236
Logistic Regression	0.6343	0.6328

Figure 3: Accuracy for both classification after hyperparameters

Logistic Regression doesn't have any critical hyper parameters to tune, as Figure 4 shows all the different parameters that we tested for Logistic Regression. We found that the C parameter made the biggest difference in terms of overall effect on both the testing and training accuracy. We also found that the 'saga' solver which is a variant of the Stochastic Average Gradient Descent along with the 'l1' penalty proved to be time-efficient and worked best for larger datasets such as our own.

Hyperparameter	Description	Value After Tuning
solver	Algorithm to use in the optimization problem	saga
C	Regularization penalty strength	10
penalty	Used to specific the norm used in the penalization	l1
max_iter	Maximum number of iterations taken for the solvers to converge.	3000
multi_class	Classification task with more than two classes	Multinomial
class_weight	Weights associated with classes	balanced
random_state	Random permutations to generate split	1

Figure 4: Logistic Regression Hyperparameters

The biggest gains were made when tuning the solver with the Neural Network. Through some preliminary research, the solver 'lbfgs' was suggested for use by the sklearn documentation and a few articles/forum posts, but in practice it performed the worst of the three available (lbfgs, sgd, adam). Adam is a stochastic solver whereas lbfgs is a "quasi-Newton method" optimized for low bandwidth according to the documentation.

Hyperparameter	Description	Value After Tuning
max_iter	Maximum number of iterations. The solver iterates until convergence (ideal), or until this number of iterations.	300
hidden_layer_sizes	Determines amount of neurons in each hidden layer.	(14,)
solver	Algorithm for weight optimization	adam
alpha	L2 penalty	0.1
random_state	Random permutations to generate split	5
epsilon	Value for numerical stability when using the stochastic solver Adam	0.00000001
learning_rate_init	Initial learning rate. Only used with stochastic solvers.	0.001

Figure 5: Neural Network Hyperparameters

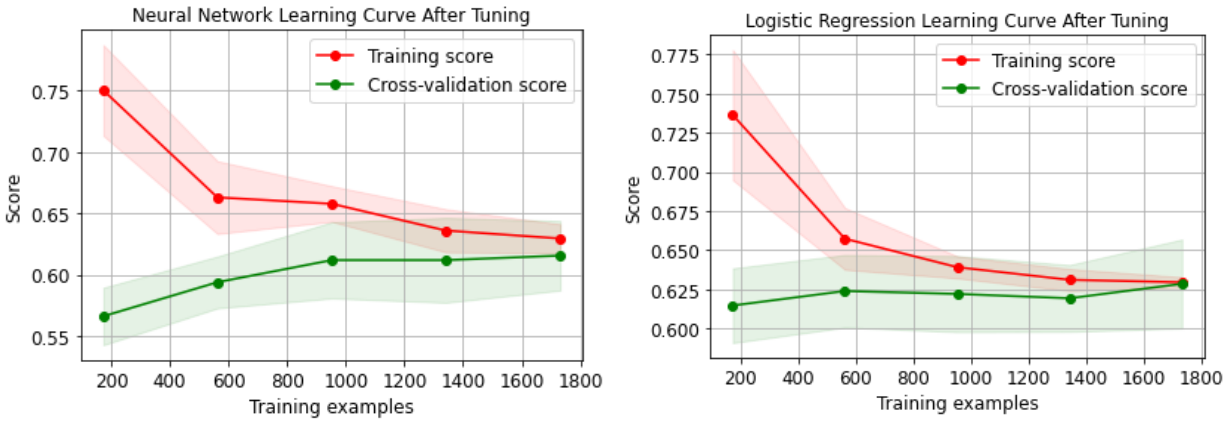

Figure 6:Neural Network and Logistic Regression accuracy after hypertuning

5 - Limitations

5.1 - Data Size Limitations

One of the primary limitations surrounding our project was the limited data size. Both the training and test data were constructed from a single NHL season, which encompasses 1082 unique games played. Had we opted to include additional seasons as well, the dataset could have been substantially larger. However, this introduces many other complications and variables such as new team members across seasons, general team performance changes across seasons, etc.

5.2 - Uncontrollable Variables

There are a multitude of uncontrollable variables that we could not account for in our data, such as strong players getting injured or suspended, overall team fatigue from traveling and time zone changes, or mid-season trades.

5.3 - Unreliability of Initial Training Data

The system learns based on previous games, and we start the training process from game 2 where there has only been one game to base predictions on. This means that our initial predictions are very inaccurate when compared to mid-season predictions. To rectify this at the beginning of the season, it could possibly be beneficial to start with data from the previous season.

6 - Conclusion

Due to the data size limitations and relative lack of previous work on the prediction of NHL game outcomes, this was an interesting problem to tackle. Looking at the training data first, both the Neural Network and Logistic Regression produce similar results with 63.20% and 63.43% respectively. The Logistic Regression model produced the most successful results, giving a respectable 63.30% in test accuracy. Our Neural Network gave us 61.20% on test accuracy. While these accuracies don't seem impressive, they are still statistically significant (given that our accuracy exceeded the necessary 50.03% based on our p-value calculation in Formula 2), and they fall in line with the results of other studies on the topic. However, there is certainly still room for improvement. We would have liked to implement a mechanism to account for an injured player, or even engineer a feature that represents team fatigue. Another possible feature we would have liked to add is a streak factor, which observes if a team is on a win/loss streak and slightly favours them more/less. It would be interesting to apply our findings to sports with similarly few scoring events such as soccer which would both diversify sports betting pools and exaggerate any significant patterns and metrics.

References

Casino. (2020). Global Gambling Industry in Recent Years. Retrieved from https://www.casino.org/features/gambling-statistics/

J. Weissbock (n.d.) Use of Performance Metrics to Forecast Success in the National Hockey League. Retrieved from http://ceur-ws.org/Vol-1969/paper-06.pdf

S. Bouzianis (2019) Predicting the Outcome of NFL Games Using Logistic Regression. Retrieved from https://scholars.unh.edu/honors/474