Predicting the Outcome of NHL Games Using Machine Learning Methods

Michael Ronayne[1]

I. PROBLEM DESCRIPTION

The National Hockey League is the premier professional ice hockey league in North America. The league is composed of 31 teams, with 24 located in the United States, and 6 in Canada. Since the league's inception in 1917, game results and team-level statistics have been accurately collected and made publicly available.

While online sports betting is only legal in a few states, there would be great value to knowing the outcome of an NHL game before it starts. In this paper, a method for predicting the outcome of NHL games using machine learning techniques will be outlined in detail.

II. DATA DESCRIPTION

Individual game result data was collected for the seasons spanning from 2000-2001 to 2016-2017. The 2004-2005 and 2012-2013 seasons were excluded due to labor negotiations. Therefore, game result data was collected for 15 seasons.

Each NHL team competes in 82 games during the regular season, with the total number of games totaling 1,271. Therefore, this dataset contained 19,065 game results.

III. METHODS

A. Approach / Theory

As a season progresses, the amount of data available for each team also increases. Therefore, in theory, as a season progresses, predictions about the outcome of a game should become more accurate. In the extreme example, predicting the outcome of the second game of the season (where the only information available for each team comes from their first game) should be much more difficult than predicting the outcome of the last game of the season (where information from essentially the team's entire season is available).

Using this thought process, the data was filtered to only include games played in the second half of the season.

B. Data Collection and Preprocessing

The data was collected via a single call to the NHL.com/stats API. The resulting JSON object was stored in a Python Pandas DataFrame. Each row in the data represents one team's performance in a single game. Therefore, there are two rows associated with each game (one for the home team, another for the road team).

From the original data, some irrelevant fields were dropped. Other metrics, such as winStreak, were generated from the data. Table I shows the final metrics used.

TABLE I
METRICS

Feature	Description
faceoffWinPctg	Face-off win percentage
goalsFor	Goals scored
goalsAgainst	Goals allowed
points	2 points for a win, 1 for an overtime or shootout loss, 0 for a loss
ppGoalPctg	Percentage of power-plays resulting in a goal
pkSavePctg	Percentage of penalty-kills not resulting in a goal
shotsFor	Shots taken
shotsAgainst	Shots by opposing team
shNumTimes	Number of penalites taken by the team
daysBetweenGames	Time since the team last played
winStreak	Number of consecutive games won
loseStreak	Number of consecutive games lost

The features for each game were transformed into the cumulative sum of those features for every game prior to the current game.

As previously mentioned, all games in which at least one of the teams had not played 41 games (the halfway point of the season) were discarded. Games resulting in ties were also discarded (as of the 2005-2006 season, games can no longer end in a tie).

The two rows associated with each game were then combined into one row using the following method:

$$Home\ team\ features - Road\ team\ features \quad (1)$$

The class label for each sample becomes the boolean value $Home\ team\ won$.

The final step of preprocessing was normalizing each column to a zero mean. Table II demonstrates a simplified example of the transformation from raw data to the final dataset.

C. Machine Learning Techniques

The data from the 2000-2001 to 2014-2015 seasons were used as the training data. These totaled 7,845 game samples.

The data from the 2015-2016 and 2016-2017 seasons were then used as the testing data.

As a fact of sports, the home team wins more frequently than the road team. To account for this class imbalance, games in which the home team won were randomly discarded in the training set until the number of games in which the home and road teams won were equal.

[1]Michael Ronayne is a computer science student in the Honors College of Michigan State University ronayne1@msu.edu

TABLE II

PREPROCESSING STEPS

A. Original Data

Game ID	Team	Location	Goals	Winner
1	Detroit	Home	6	1
1	New York	Away	2	0
2	Detroit	Home	1	0
2	New York	Away	3	1
3	Detroit	Away	2	0
3	New York	Home	1	1

B. Cumulative Sum Data

Game ID	Team	Location	Sum Goals	Sum Wins	Winner
2	Detroit	Home	6	1	0
2	New York	Away	2	0	1
3	Detroit	Away	7	1	0
3	New York	Home	5	1	1

C. Final Data

Game ID	Goal Difference	Win Difference	Winner
2	4	1	0
3	-2	0	1

The Python Scikit-Learn package was used to apply three machine learning algorithms to the dataset as shown in Table III.

TABLE III

MACHINE LEARNING ALGORITHMS AND HYPERPARAMETERS

Logistic Regression

Parameter	Tested Values
C	0.01, 0.1, 1, 10, 100
dual	True, False
penalty	l1, l2
tol	1e-3, 1e-4
max_iter	100, 500, 1000

Random Forest

Parameter	Tested Values
n_estimators	100, 500, 1000
max_features	sqrt, None
max_depth	3, 4, None
min_samples_split	2, 3, 10
criterion	gini, entropy

Support Vector Machine

Parameter	Tested Values
C	0.01, 0.1, 1, 10, 100
kernel	linear, poly, rbf, sigmoid
degree	2, 3, 4
tol	1e-3, 1e-4

Scikit-learn's GridSearchCV was used to perform a 3-fold cross-validation on the training set. The GridSearchCV does an exhaustive trial of the specified parameter values, and returns the best estimator trained using the tuned hyper-parameters. The learning script was run on a high-performing computing cluster (HPCC).

IV. RESULTS

Table IV summarizes the optimal tuned parameters for each estimator after being trained on the testing set. When reviewing the accuracies, note that a baseline estimator that simply predicts $Home\ team\ won$ for each testing sample would have an accuracy of 53.3%.

TABLE IV

BEST PARAMETERS OF EACH CLASSIFIER

Logistic Regression

Parameter	Best Value
C	0.01
dual	False
max_iter	100
penalty	l2
tol	0.0001
Accuracy	56.9%

Random Forest

Parameter	Best Value
criterion	entropy
max_depth	3
max_features	sqrt
min_samples_split	3
n_estimators	100
Accuracy	57.4%

Ultimately, the Random Forest Classifier performed the best in terms of overall accuracy. Detailed performance metrics for the Random Forest Classifier are shown in Table V.

In addition to overall binary prediciton accuracy, for each sample in the testing set, a *probability estimate* was made for the likelihood that $Home\ team\ won$. Table VI summarizes these results for the Random Forest Classifier.

V. DISCUSSION

While the best accuracy achieved was only 56.9%, the results grouped by probability show promise. In the case of the Random Forest Classifier, if a bettor were to only bet on games where the prediction probability was greater than or equal to 60%, the bettor would win 70.6% of the time. This suggests that a profitable betting strategy might involve not betting on *all* game result predictions, but only those where the classifier is at least 60% certain of its prediciton.

TABLE V

RANDOM FOREST PERFORMANCE METRICS

Class	Precision	Recall	F1-Score	Support
0	0.54	0.58	0.56	585
1	0.61	0.57	0.59	669
Avg / Total	0.58	0.57	0.57	1254

Confusion Matrix

Class	0	1
0	58%	42%
1	43%	57%

TABLE VI
PREDICTION ACCURACY BY PROBABILITY

Random Forest			
Prediction Probability	**Correct**	**Total**	**Percentage**
>= 0.65	8	13	61.5%
>= 0.60	**101**	**143**	**70.6%**
>= 0.55	398	653	60.9%
>= 0.50	720	1254	57.4%

However, the calculation of the profitability of such a strategy would require collecting historical money line betting data, which is beyond the scope of this report.

VI. FUTURE IMPROVEMENTS

This model uses a relatively small number of features. Future improvements could include advanced statistics at the team and player level (such as Corsi and Fenwick metrics). Different transformations of the data could also be tested, such as keeping the home and away team metrics as separate features (as opposed to the difference between the two), or expressing the home team metrics as a percentage of the away team metrics. More seasons of data could also be included (however, rule changes throughout the years would add complexity). Finally, games earlier in the season could be included and compared to the results from the current method of training only on games at least half-way through the season.

FURTHER READING

Joshua Weissbock's thesis *Forecasting Success in the National Hockey League using In-Game Statistics and Textual Data* was a great reference for this project. This project's results were originally discouraging, but Weissbock's paper proposes that there is a theoretical upper-bound to the prediction accuracy of the outcome of NHL games.