Modeling NHL Win Probabilities

James Benasuli

https://github.com/jbenasuli/nhl-predictons

Outline

- Overview
- Data
- Methods
- Results
- Conclusions

Overview

- With sports betting becoming increasingly popular and mainstream, data science can be used to make superior decisions over gut intuitions.
- Moneyline betting (betting who will win with no caveats) is the most common type of sports bet
- Can bettors gain an edge?
  - This project aims to answer that question by testing the ability to better predict outcomes than naive choices
  - Naive choice for this test is predicting whether the home team will win
- Goal: Build a model that outputs more accurate probabilities to validate hypothesis and greenlight more in depth testing

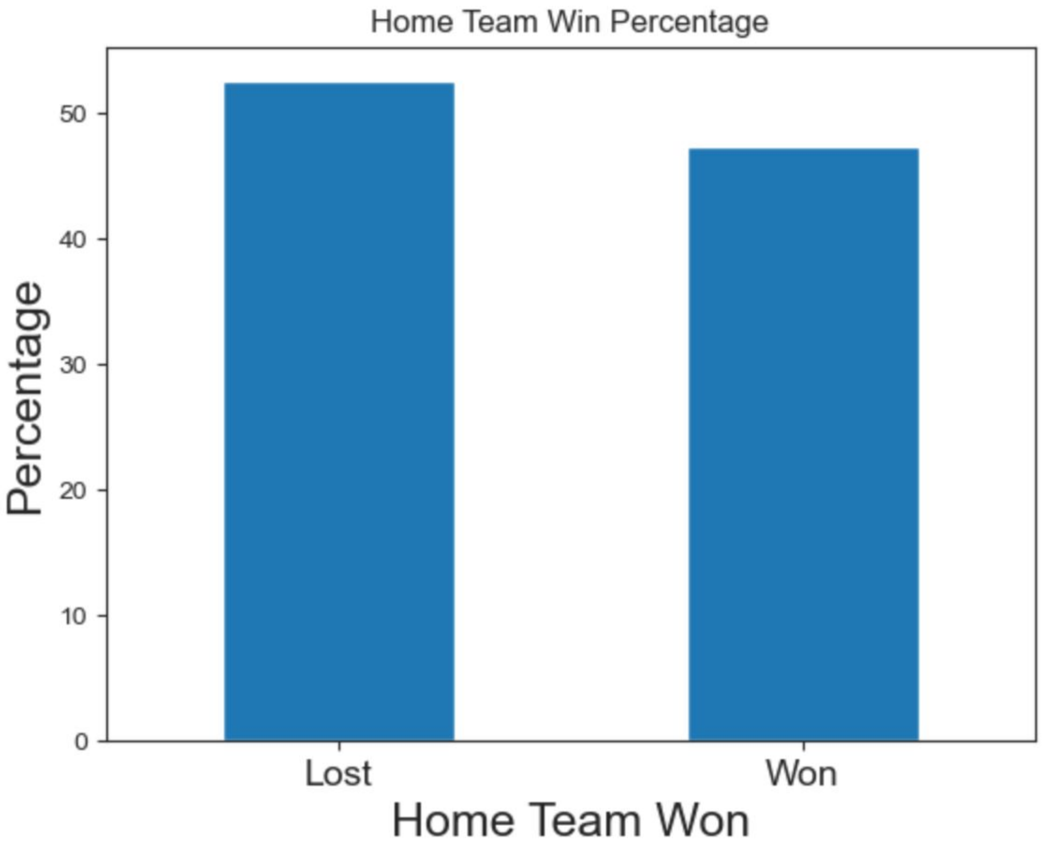
Data

- 2019-2020 season through the current season (2022-2023)
- Team performance stats at the game level from Natural Stat Trick
- Official game results scraped from the NHL API
- Target variable is whether or not the home team one
- Features include a mix of advanced offensive, defensive and goaltending stats
  - Stats are in game share rate form
    - Highlights team performance relative to opponents for both the home and away sides
  - Stats are transformed by taking the prior ten game rolling average and shifted back one game
    - The shifting is necessary to predict future results, as we will not know game stats until after a game begins/ends
  - 10 games was chosen for the window given the streakiness of NHL play and how much can change over bigger horizons due to injuries, trades, etc

Naive Choice on the Home Team to win

Home advantage has always been considered important in sports

- Picking the home team every time would result in being correct 52.67% of the time



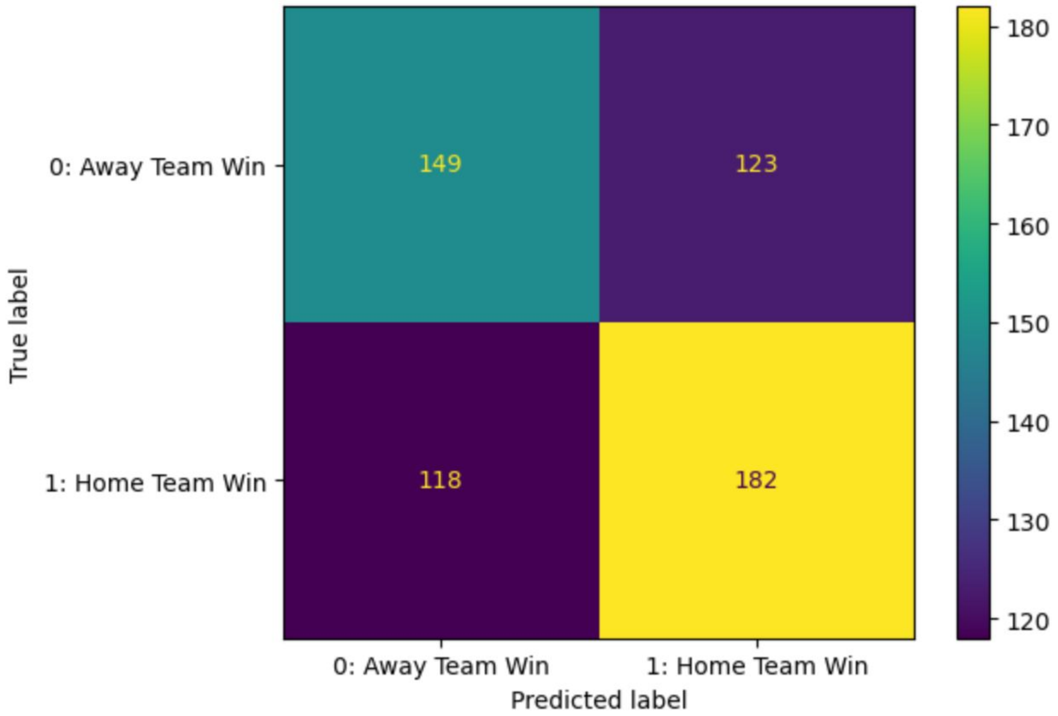
Methods

- Optimize for log loss
- Train on the 19-20 season through the first half of this season
- Evaluate and predict on the second half of this season
- Use rolling means of features with a window of 10 games
- Classification modeling
  - Logistic Regression
  - Gradient Boosting
  - AdaBoost
  - Neural Network
- GridsearchCV to identify best model parameters

Baseline Logistic Regression Model Results

Out of the box logistic regression

- Test accuracy: 0.58
- Test AUC-ROC score: 0.621
- Test log loss score: 0.665



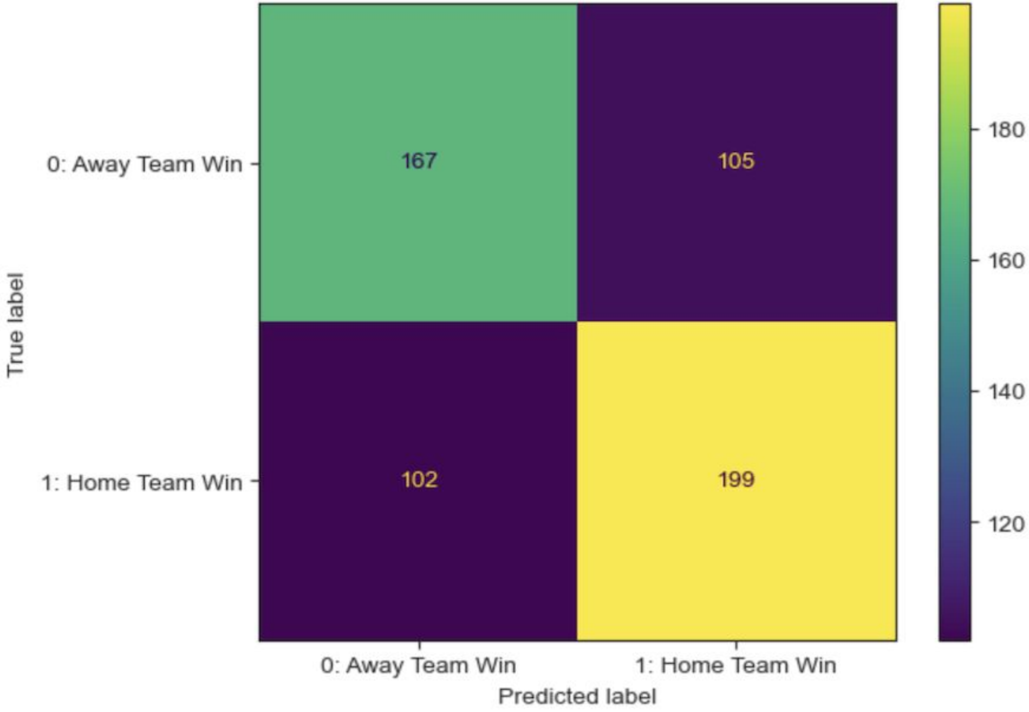
Best Param Results

	Training Cross Validation Accuracy	Training Cross Validation Log Loss	Test Accuracy	Test Log Loss	Paramters
Logistic Regression	0.608610	0.656062	0.638743	0.652721	{'logreg__C': 0.1, 'logreg__class_weight': None, 'logreg__penalty': 'l1', 'logreg__solver': 'liblinear'}
Neural Network	0.608599	0.661920	0.621291	0.654427	{'nn__activation': 'linear', 'nn__dropout_rate': 0.3, 'nn__epochs': 12, 'nn__neurons': 36, 'nn__optimizer': 'Adam', 'nn__weight_constraint': 5}
Gradient Boost	0.609763	0.661689	0.600349	0.659406	{'gb__learning_rate': 0.01, 'gb__max_depth': 3, 'gb__n_estimators': 400}
AdaBoost	0.610931	0.671000	0.633508	0.669250	{'ada__base_estimator': SVC(kernel='linear', probability=True), 'ada__learning_rate': 0.1, 'ada__n_estimators': 25}

Logistic Regression performed the best in terms of accuracy and was second in log loss

Logistic Regression Best Model CM on Unseen Preds

- Test accuracy: 0.64
- Test AUC-ROC score: 0.663
- Test log loss score: 0.653



Conclusions

- Modeling publicly available data can beat naive choices
- Percentage of times prediction correctly had the home team winning was 63.87%
- Next steps:
  - Now we know we can beat that naive prediction, test layering in info from sportsbooks
    - Would framing this from the favorite be more beneficial or cause overfitting?
      - The relatively efficient betting market could provide valuable predictive power
      - Or would it be noisy or cause overfitting?
    - Once that test is completed work on modeling a profitable betting strategy
  - Revisit feature selection
    - Test different team stats and/or try to incorporate individual stats
    - Try different rolling windows and see how predictions fare at different points in the season