Predicting NHL Player Performance

Matt Feller

March 2024

1 Problem statement

Predicting outcomes in sports is inherently difficult, but building a model which performs just a little bit better than someone else's can make all of the difference, whether one is a participant in a low-stakes fantasy league or is an analytics expert for a professional team.

The goal for this project is to build a machine-learning model that predicts NHL skaters' stats based on their recent performance. The scope of the project is deliberately limited, with the aim being to create a working model for just a few stats: goals, assists, and shots on goal. The best-performing model for each of these stats turns out to be a simple linear regression model with no regularization. The linear models perform as well on training sets as on test sets despite, suggesting that there is room to build a more advanced model without over-fitting. However, after trying a handful of more complex models using the same features and seeing no improvement, it appears that the best way to increase complexity would be to engineer more features. This would be a fruitful avenue for future work, in addition to expanding the range of predicted stats beyond just goals, assists, and shots.

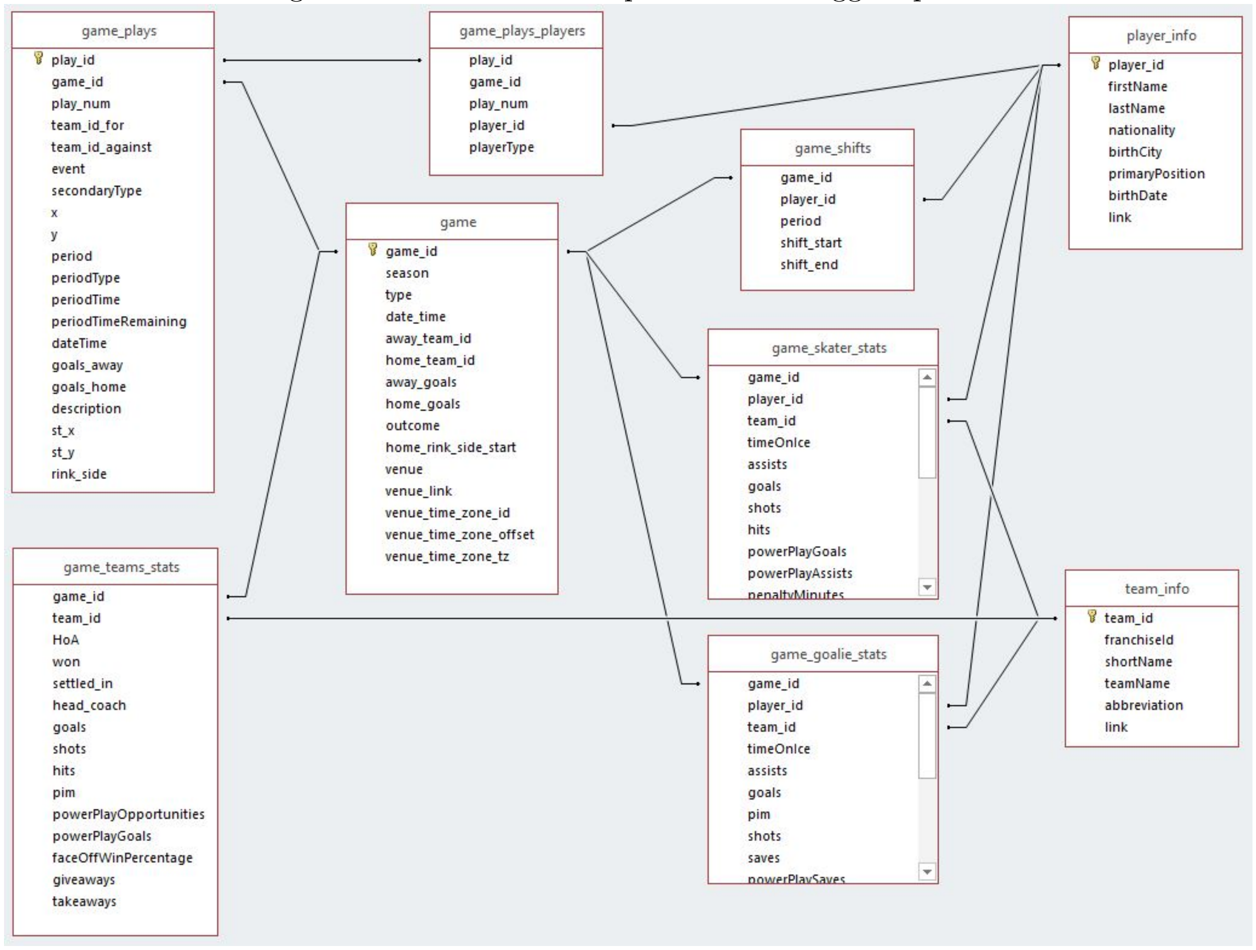
2 Data wrangling

Our raw data is from this Kaggle upload. The data is quite detailed, which leaves plenty of room to engineer some relatively sophisticated features, but due to the scope of the project much of the data is not used.

The majority of the work for this project was in transforming the various tables in the Kaggle upload into a form where linear regression would be possible. Some of this work was in cleaning and imputing the given data, but a lot of the work was also in aggregating desired data which was spread across multiple tables. The former task was the focus of the data wrangling step, while the latter overlapped with the exploratory data analysis step as it pertained more to feature engineering.

The original data covers all seasons from 2000 to 2020, with detailed information about each game. For example, there are over 4 million entries in the game_plays table. The data generally appears to be trustworthy, but there are a number of missing or duplicated values in some of the tables, and some inconsistencies that are not immediately obvious. In particular, there are a few stats which the NHL did not track at all prior to 2010, so in the end we focus only on the seasons starting in 2010-2011.

Figure 1: Table relationships from the Kaggle upload.



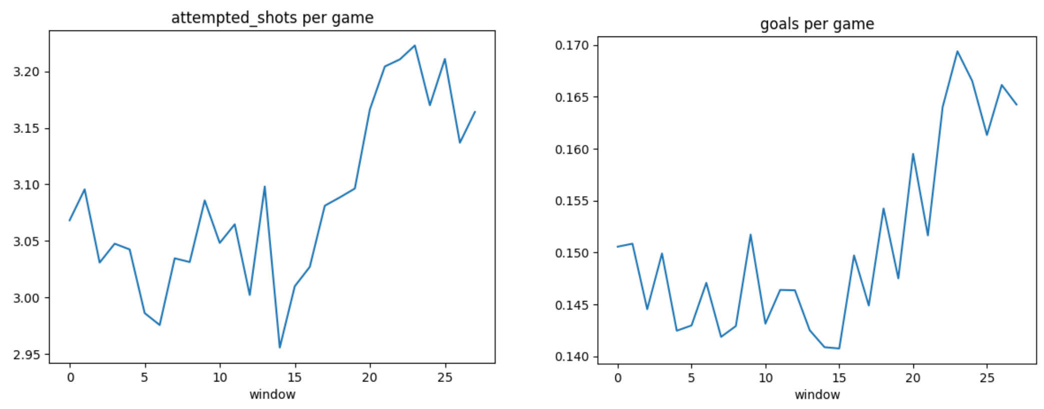
The table relationships in the original upload are shown in Figure 1. We inspected each of the tables in the dataset that could be relevant to predicting a skater's performance, dropping duplicate rows and a number of columns that we don't anticipate using in our analysis. We imputed the heights and weights of three skaters with missing data by taking the mean across all other players, and we also imputed about 700 of the game_shifts table to add in missing shift_end values by adding the mean shift length to the shift_start value. Furthermore, we dropped duplicated shift data which had multiple entries for the same player at the same start time in the same game, but different shift end times, electing to keep the rows with later values. There were no major issues with the datatypes for each column.

3 Exploratory data analysis

The main goal during this step is to create new dataframes which contain the relevant statistics we would like to use in feature creation. For example, in order to use "shot attempts" as a feature in our model, we create a function which, given a time window, outputs a pandas series giving each player's number of shot attempts during that time window. A more sophisticated function does something similar, but counts the number of attempted shots by a player's team when that player was on the ice. These functions involve some relatively sophisticated pandas merges, since the information about shot attempts, players, and teams are all stored in different tables. Using functions like this, we create dataframes for these statistics for seasons from 2010 to 2020, breaking each season roughly into thirds.

When comparing (across the whole league) over the decade, we see that attempts, shots-on-goal, and goals are indeed roughly correlated, justifying our interest in attempts as a useful feature.

Figure 2: Shot attempts vs goals.



We see that shot attempts are a worthwhile statistic to track, since they are more common than goals or shots-on-goal, but are still correlated with goals.

4 Preprocessing

The goal of this step is to get the data ready for modeling, which involves imputing missing values and splitting into training/test sets, but also means transforming it into a form where the features in X contain the "past" windows while the data in Y is the "current" window. We create a new dataframe multi-indexed by time-window and player_id, with columns for the player's stats in that window and in the preceding five windows. Because each player has a different number of games in each window, we also transform each stat into a per-game version.

After all of these transformations, our dataset had 13757 rows. We split it into a training set (80% of the total), a test set of all of the most recent two windows (9% of the total), and a randomized test set of the older windows (11% of the total). The reasoning for taking separate test sets is that each comes with its own pros/cons: testing with the most recent two windows gives us a glimpse of how a model does with 100% new information in the future, but testing with randomized windows avoids any bias from the particularities of the most recent season.

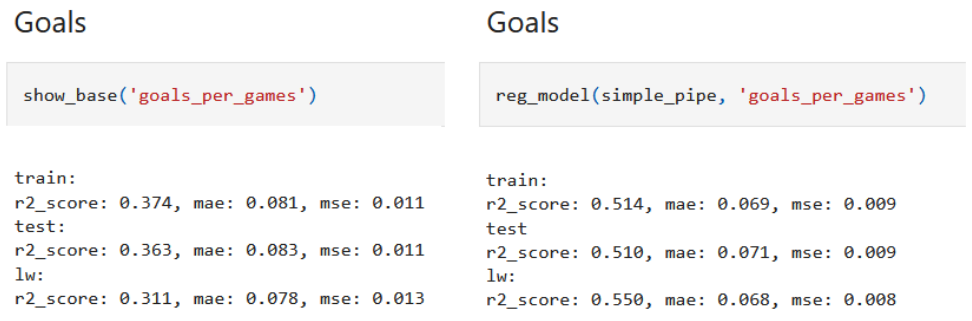
At this step, we impute missing values with zeros, since most seem to come from players who did indeed play during a certain window but had no recorded plays of that type during that window.

5 Modeling

We trained various models on our prepared data, starting with simplistic baseline models, then a simple linear regression model, and then some more complex models. In each case, we are actually training three models: one for shots on goal, one for assists, and one for goals. However, we see that the performance for each type of model follows a similar pattern throughout, regardless of which stat we are considering.

We consider three overly-simplistic models to establish a baseline: the first is a dummy model which takes the median of each value in Y_train and uses that as the prediction, the second is a model which selects just the most recent window feature as the prediction, and the third takes the average of each player's performance over all five recent windows. For each of the three stats, these baselines were progressively better-performing. For example, the dummy model for goals had a mean absolute error (MAE) of about 0.100 goals per game (which is almost at the median of 0.115 goals per game!) but the simple model taking an average of goals per game across all five windows yielded a MAE of about 0.080 goals per game.

Figure 3: Baseline vs Linear Regression.



Goals

```
show_base('goals_per_games')
```

```
train:
r2_score: 0.374, mae: 0.081, mse: 0.011
test:
r2_score: 0.363, mae: 0.083, mse: 0.011
lw:
r2_score: 0.311, mae: 0.078, mse: 0.013
```

Goals

```
reg_model(simple_pipe, 'goals_per_games')
```

```
train:
r2_score: 0.514, mae: 0.069, mse: 0.009
test
r2_score: 0.510, mae: 0.071, mse: 0.009
lw:
r2_score: 0.550, mae: 0.068, mse: 0.008
```

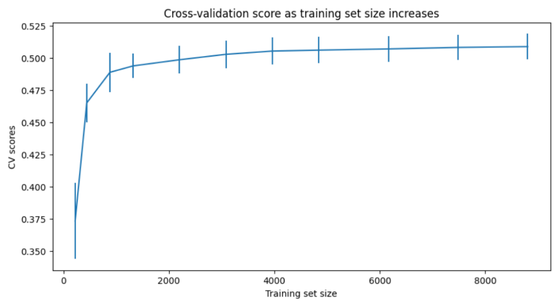
The simple linear regression model yields a MAE of around 0.70 goals per game. Figure 3 shows the full results, with "lw" referring to the test set containing the last two windows (i.e., the 2019-2020 season). There was similarly meaningful improvement over the baseline for the other two stats as well.

Trying a random forest model, with a cross-validation grid search on the number of estimators, yielded a model which drastically over-fit the data yet seemed to perform roughly the same on the tests sets. Training these models took orders of magnitude longer than linear regression, so it did not seem to be a promising avenue to pursue.

Increasing the linear model's complexity by adding polynomial features also turned out to be a dead end, as the models' performance on test sets was slightly worse than the original linear model due to over-fitting.

6 Conclusion

Our simple linear regression model for each statistic performed meaningfully better than our baselines, while matching other more complex models. None of our attempts to increase complexity yielded any promise for improvement via further tweaking, so we have settled on the simple linear regression model.



We also saw that our model is comfortably along the learning curve, with more rows unlikely to yield significantly better results, as seen in the graph on the right.

Since more data and increased model complexity do not seem to be good paths to a better model, the most promising way forward would appear to be in engineering more features by going and making more full use of the original dataset.

It is also interesting to note that the features age, age_squared, and isForward are among the largest coefficients in the chosen model, suggesting that acquiring mode data for each player which is not a direct measure of their on-ice performance could be worthwhile. One idea is to include contract information; it is a well-discussed phenomenon that players tend to perform better in the years that their contracts are set to expire, as they have extra incentive to improve their negotiating power in a new contract.