You Play to Win (or Tie) the Game

I. Introduction

The great Herman Edwards once said, “You play to win the game. Hello! You PLAY to WIN the GAME!” Many statistics can help show how a team won the game (e.g. a high completion percentage over showing they passed well or a high expected goals number showing they obtained great chances at scoring) but ultimately the most important statistic at the end of the day is the checkmark next to the win column. For that reason, I asked this question: how can I calculate the contributions that each individual player makes to winning? In this paper you will find that I was able to construct an expected match points probability model based on the score differential, current expected goals, and amount of time left in the game that can give us the match points probability added for each of the Erie Otters players over their 40 games.

II. Data and Methodology

For this competition, I used the 40 game Stathletes’ translated data sample from the Otters’ 2019-2020 season. Originally, I wanted to create a traditional two class win probability model but hockey’s propensity to tie changed that plan. Therefore, I needed to include the possibility of tying in my model. I then created a three class model that predicted the probability of winning, losing, or tying at any given moment in the game based on the score differential, current expected goals, and amount of time left in the game. Since a team gets two points in the standings for a win (I will denote them as match points to differentiate from goals sometimes being called points), one for a tie, and zero for a loss, I can construct the expected match points, or xMP, at any given moment from the following equation,

$$xMP = 0 * Pr(Lose) + 1 * Pr(Tie) + 2 * Pr(Win)$$

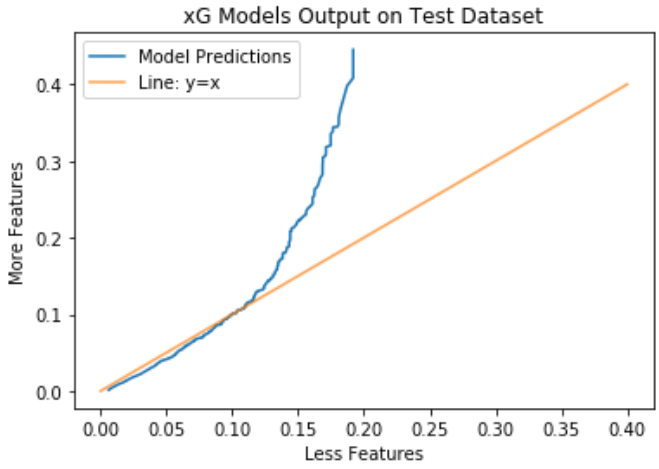
Once I had the probabilities from the three class model, I applied this equation to each event in the dataset and obtained the xMP at that moment. From there I could subtract the current event’s xMP from the next event’s xMP to get the expected match points added, or xMPA, for each event. In order to give credit to the correct player, I needed to subtract the current event from the next event

rather than subtracting the previous event from the current. If I subtracted the previous event's xMP from the current event then the person who received the puck from the man who won the faceoff would get the credit (and similarly for passes). From this, I simply needed to add up the xMP for the player on each event and I got the overall xMP added.

III. Models

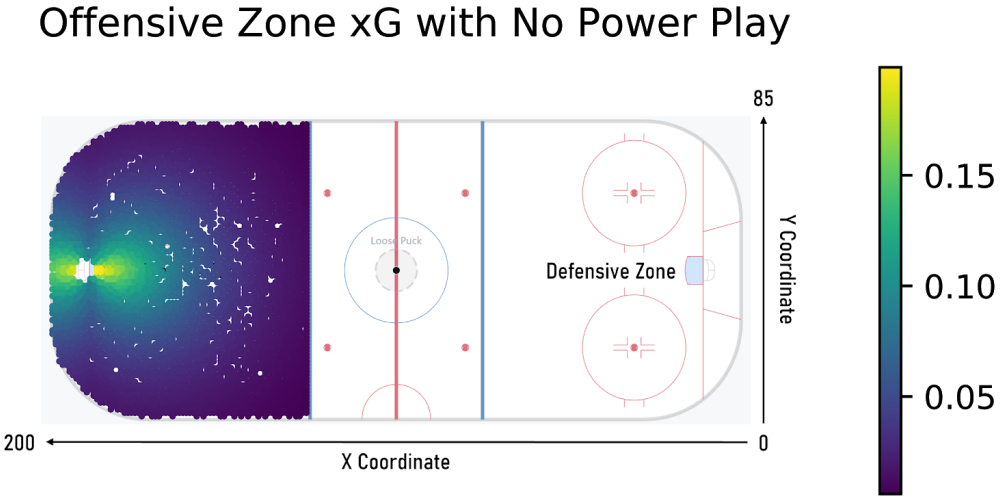
To create the three class win/lose/tie probability model, I needed to create three models: expected goals (xG), expected completion percentage (xCP), and expected pass to point percentage (xPtP).

The xG model was the most obvious one to create since xG is an input for the three class model. From this, I created two xgboost models. One was more complex with parameters such as the time on possession, the amount of extra skaters if on the power play or penalty kill, the type of shot, whether there is traffic, whether the shot is a one timer, the distance to the goal, and the surface area of the net seen. The other was simpler with the only parameters being the amount of extra skaters, if either net was empty, the distance to goal, and the surface area of the net seen. These two models were pretty consistent with each other when it came to predicting low probability shots. It was more conservative when predicting higher probability shots which was fine due to the survivorship bias in training.



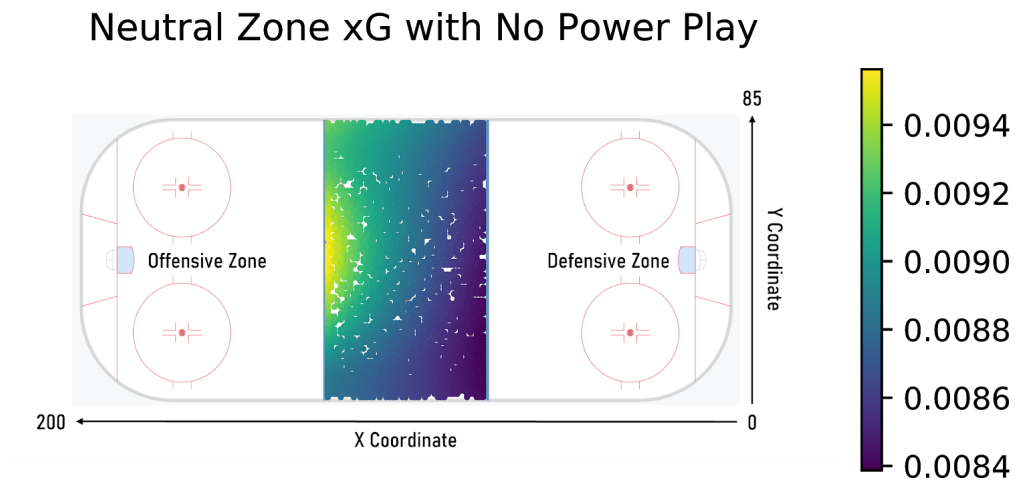
Since the model was only trained on actual shots, we can expect the model to output a higher expected goal number than what may be observed at a given

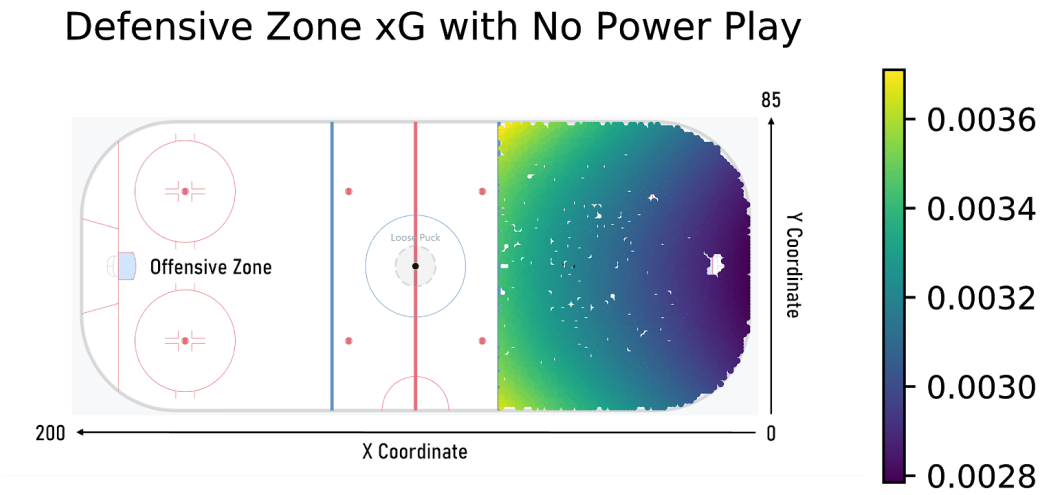
point. This is due to players inherently knowing when the best time is to shoot from years of playing and actually doing so. If a player has the puck at what seems like an optimal place according to the model and doesn't shoot, there generally is a good reason for it. For that reason, I was more than happy to use the more conservative simple model when in the offensive zone.



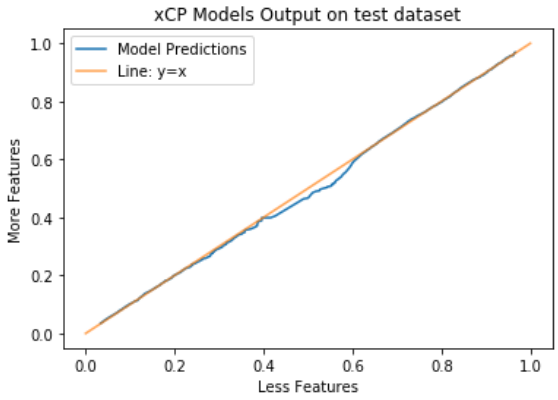
The only problem was that it was based on the location of shots, which meant it was not going to be feasible to use this model on the neutral/defensive zones of the ice.

To get the xG for each point in the neutral zone, I needed to take a grid of points in the offensive zone and, for each of the points in the grid, sum the product of the xG, xCP, and xPtP. I also had to subtract the product of the xG, expected incompletion percentage (1-xCP), and xPtP for each of the points to account for the fact that the pass may be intercepted. This would give the xG for each point in the neutral zone. The same process was used to get the xG for the points in the defensive zone (except the grid of points was in the neutral zone). This process obviously does not account for the chance that a skater could just continue carrying the puck up the ice, but I could not think of a great way to account for that. In the future, one way to improve on this model would be to account for that possibility and adjust the xG as necessary.

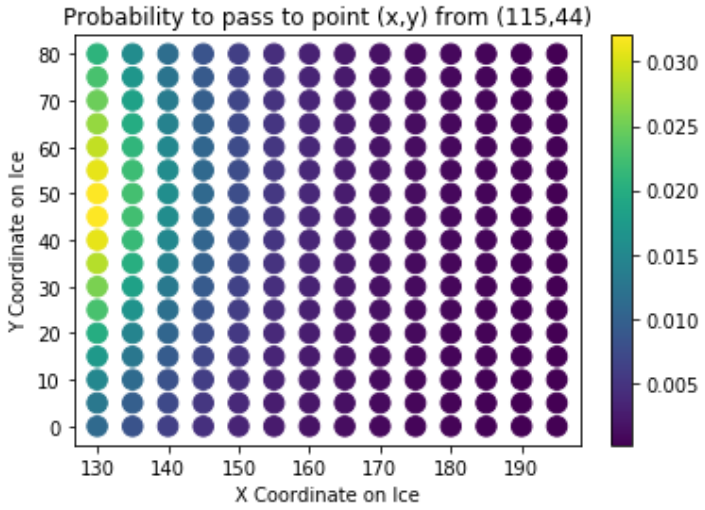




To get the xG for the neutral and defensive zones, I constructed the xCP and xPtP models. The xCP model was constructed very similarly to the xG model with an xgboost framework. Additionally, I created a simple model which included parameters such as the amount of extra skaters for the passing team, total distance of pass, and starting and ending X/Y coordinates, among other features. The complex model had all of those features plus the time on the current possession, whether the pass was direct, and the pass angle. The results for the two were strikingly similar again which led me to conclude that I could use the simple model again.



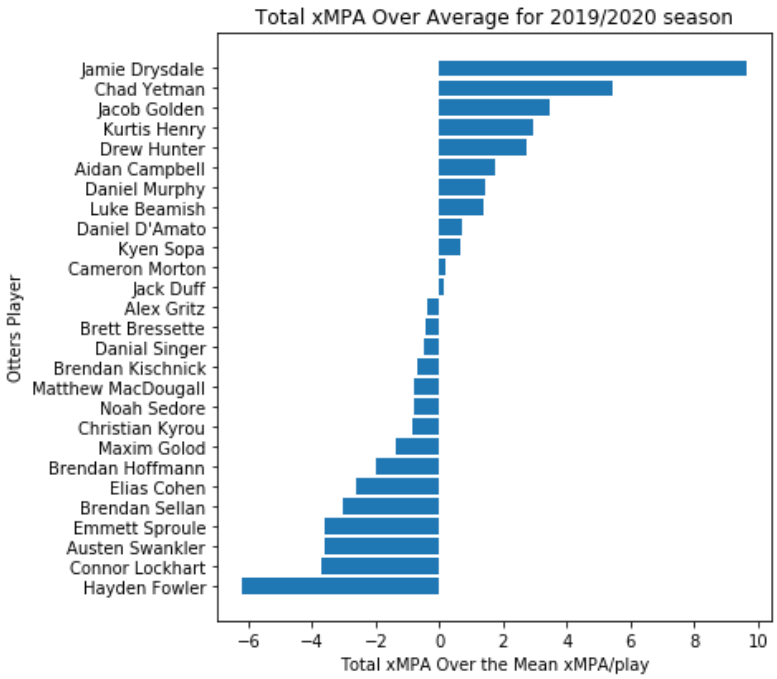
The xPtP model was also an xgboost regression model that took the initial X/Y coordinates, ending X/Y coordinates, and the amount of extra skaters as parameters. For each pass in the dataset, I also had to make a row for all the points that did not get passed to. Thus, I got the expected pass to point model.



We then had the expected goals at every point on the ice and could create the three class win/lose/tie probability model with parameters xG, current score differential, and time left in the game.

IV. Results

Below you will find the results of the expected match points added over the mean xMPA/play for each of the players on the Otters. Over the 40 games, we see a fairly large spread of xMPA between the players with players being more than 9 xMPA below the mean and as much 5 xMPA above the mean. Jamie Drysdale was the highest performer whereas Hayden Fowler was on the lower end.



V. Applications

Some ways we can apply this information is how we construct our lines. Based on this data, players like Jamie Drysdale and Chet Yetman would warrant places on higher lines whereas players like Conner Lockhart and Hayden Fowler would possibly be placed on the lower lines. Fundamentally, there is more information that needs to be considered, such as, who plays well with whom but this could at minimum be used as a tiebreaker if a coach was not sure who to put on the higher line. Additionally, it could be used as an indicator to shake up the various lines. If one or two lines were putting up poor xMPA numbers together over the course of some games, the coach could use that information to change up the groupings.

VI. Appendix

- All of the code for this project can be found at https://github.com/JCHampton/BigDataCup
- XGBoost model usage and some code was used from Akshay Easwaran who took inspiration from Ben Baldwin and Sebastian Carl. Code is at https://github.com/akeaswaran/nfl-cpoe-model