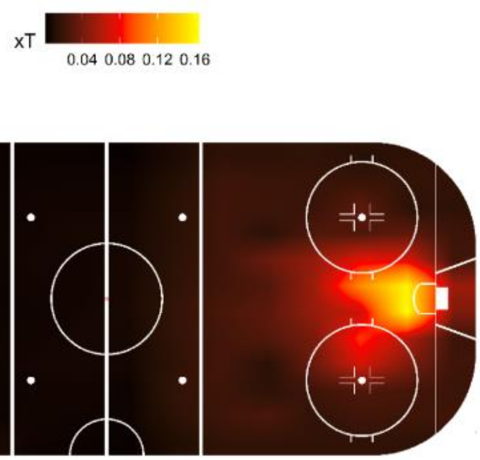
Projecting Win Probability in Women's Hockey

Evan Cheney, Adrian Cadena, Pablo Landeros

Which Players Help Us Win Games?

What is the best way to measure a player's contribution to the game? This was our question when developing our win probability model using both NWHL and Women's Olympic data provided by Stathletes. From previous research done in this space in other sports, such as the NBA, we can assign a number to each play made based on the win probability it added or subtracted from a team's chances of winning the game [1]. We can look at Win Probability Added (WPA) from two different perspectives. First, we can look at it as a total, assessing who the best overall players were from the game. We can also evaluate WPA as a measure of efficiency, considering which players averaged the highest WPA on average. In our opinion, the purposes of player evaluation should look at WPA as a tool to measure player efficiency rather than as a total. This allows for better personnel decisions when looking at which players a coach should provide more or less ice time.

One of the key metrics to our model for WPA is a new metric that we created called Net Expected Threat (nxT). This new metric is a derivative of the Expected Threat (xT) statistical prevalent by Karun Singh [2]. What we add to the original statistic is more in-game context that is missing from xT. For example, we add time remaining in the game to account for the increased aggression of play when a team is losing toward the end of a game. Or, in the inverse, the conservatism that a team will show when they are winning. In the following graphic, we can see how xT will increase as a player gets closer to the net, coinciding with high WPA plays.

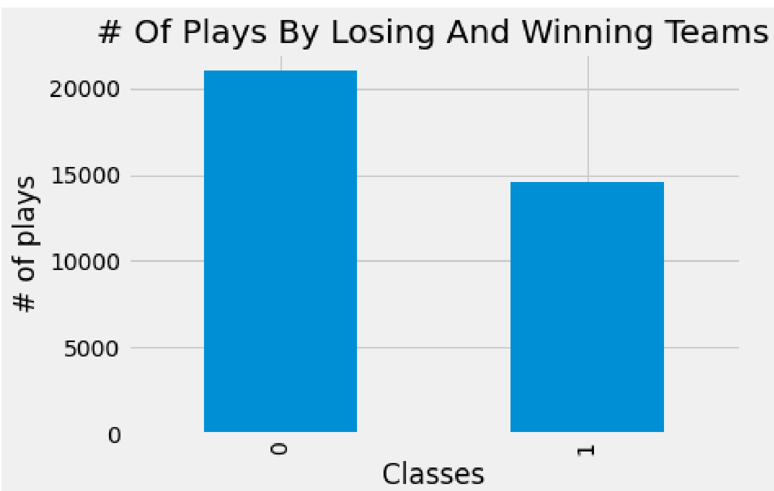


Explaining WPA

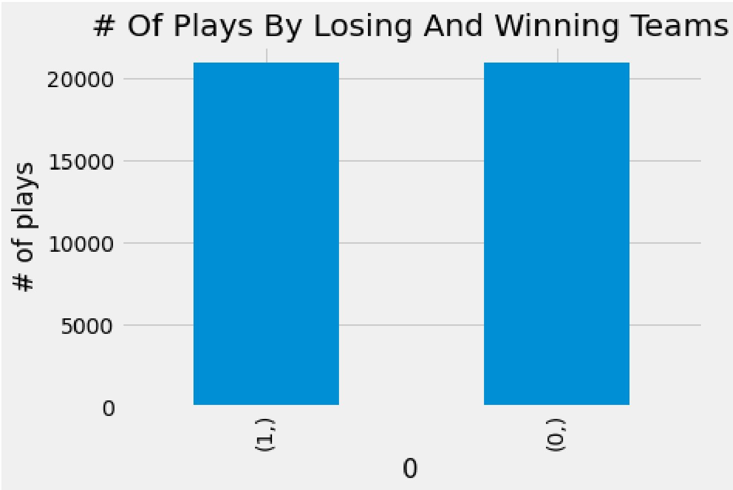
Data

The data that we used for this experiment included variables included in the original dataset provided by Stathletes. This data includes X and Y coordinate data for shot location, event type, player name, and action team name. We also created variables that would better account for in-game contextual information to simulate what a coach would evaluate while the game was still occurring. This data would include the time mentioned earlier in the game, the score differential by the current possessing team, the skater differential to account for power-play and short-handed situations, a binary variable to account for whether or not the goalie was currently pulled, and nxT. We split the data using a 70%-30% training/testing split after withholding Game ID #60 to use as a validation set to test a complete game for modeling.

One of our first observations from the data is that teams that eventually lost their respective games were involved in more in-game events than winning teams. Losing teams are represented by a "0" while winning teams are classified with a "1".



This class imbalance poses a challenge to training a classification model; without any data balancing, the model will learn more about losing teams than winning teams. To handle this, we used the SMOTE library in Python. This method uses K-Nearest Neighbors (KNN) to provide our training data with more similar instances of data [3] that would lead to a team winning their respective game. The following chart shows our class balance after using SMOTE, now with the exact number of instances of data from winning and losing teams.

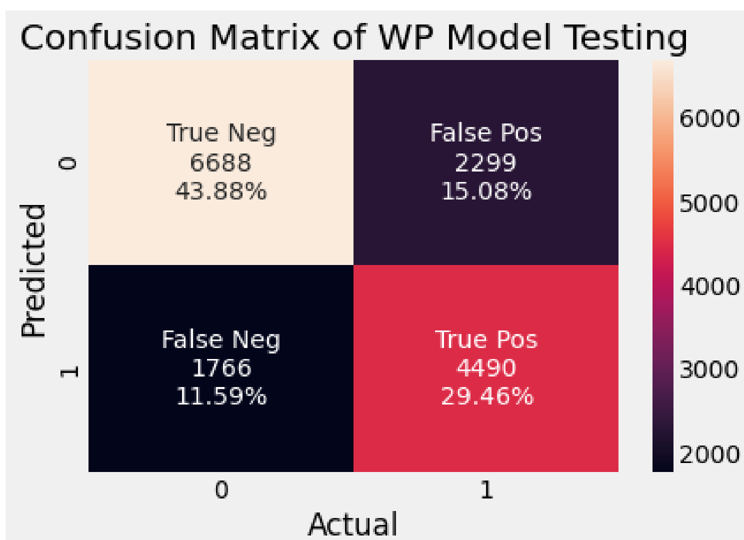


Win Probability Model

With a balanced dataset, we can build a model that we are confident can learn all tendencies of data that would lead to both a winning and losing team. Using an XgBoost classification model [4], we can look at in-game probability distributions as to how likely the current possessing team is to win the game based on the current situation based on the following variables.

1. Time Remaining (Sec)
2. Skater Differential
3. Score Differential
4. Goalie Pulled?
5. xT

To tune our model, we used Bayesian Optimization in the Python package Hyperopt [5] to search the entire parameter grid for the Xgboost algorithm, optimizing for our F1-Score. Our model performed well enough to generate an F1-Score [6] of 73%, minimizing false negatives and false negatives.



Testing on Full Game

Our validation dataset includes the game between the Minnesota Whitecaps and the Metropolitan Riveters on January 26th, 2021. We can look at who the top players were based on how much win probability they added on a per-play basis from this game.

player	team	WPA
Brooke White	Minnesota Whitecaps	2.87
Kendall Cornine	Metropolitan Riveters	1.44
Jonna Curtis	Minnesota Whitecaps	1.27
Stephanie Anderson	Minnesota Whitecaps	1.11
Cailey Hutchison	Metropolitan Riveters	0.94

From this data, we can see that Brooke White was by far the best positive contributor to win probability, adding, on average, 2.87% to Minnesota's win probability per play that she was the active participant. A coach will best use this data to provide a player such as White with more playing time in future games.

Conclusion

Our analysis established that it is possible to evaluate player performance by adding to their respective teams' win probability. With the use of pre-existing statistics such as xT, we can create new statistics such as nxT, with numerous modulations of the provided dataset, to make predictions about which types of plays would add or subtract from a team's win probability. Future iterations of this model would include finding which players most affect their teams' win probability on the defensive end. This can be in the form of a quantitative metric of how, for example, defensive positioning may affect a team's chances of winning the game. Other future work can include which line combinations will add to a team's win probability the most based on optimal line matching for a coach to create these optimal lines, which, right now, are made in a sub-optimal fashion—relying on intuition more than combinations of players that have added to win probability in the past.

References

[1] S. K. Deshpande and S. T. Jensen, "Estimating an NBA Player's Impact on his Team's Chances of Winning," Wharton Faculty Research, 2016.

[2] K. Singh, "Introducing Expected Threat (xT)".

[3] N. A. Lemaître, "Imbalanced-learn: A Python Toolbox to Tackle the Curse of Imbalanced Datasets in Machine Learning," 2017.

[4] Xgboost, "XGBoost Python Package".

[5] Hyperopt, "Hyperopt: Distributed Asynchronous Hyper-parameter Optimization".

[6] T. Wood, "What is the F-Score?".