MGT 6203 Group Final Report

Team #: 62

Team Members:

1. Travis Roemhild; troemhild3

Hello, I am Travis and I am currently working as a Software Engineer where I mainly work in Ruby/Postgres. Previously I have worked on applications in the Microsoft stack (C#, JS, Angular, SQL) and am currently in my third semester at OMSA.

2. Alexander Pastor; apastor8

My name is Alex, and I am an Analytics Systems Engineer that works in the materials manufacturing industry. I come from a mechanical engineering background and have slowly made my way into analytics, working with millions of data points each day from our manufacturing technologies and connected devices in order to optimize operations.

3. Nick Taylor; ntaylor70

I am Nick Taylor, a full-time student in OMSA in Nashville, TN. I finished my undergraduate studies in Applied Mathematics during the summer of 2021. I am interested in working in healthcare as an analyst and am building skills and domain knowledge towards being a Certified Healthcare Data Analyst (CHDA).

4. Nathan Crane; ncrane7

I am an Aerospace Engineer concentrated in aircraft design of future military and civilian aircraft. I completed an MSAE from Georgia Tech in 2020 and a BSAE from Embry-Riddle Aeronautical University in Daytona Beach in 2018.

5. Baris Kopruluoglu; bkopruluoglu3

I am a full time student in OMSA and this is my 6th class in the program. I earned a PhD degree in Mathematics from Auburn University in 2020 and worked as a Math faculty for two years. I want to expertise in data science field using my Mathematics and Statistics background.

1. *Project Overview*

a. *Background Information and Business Impact*

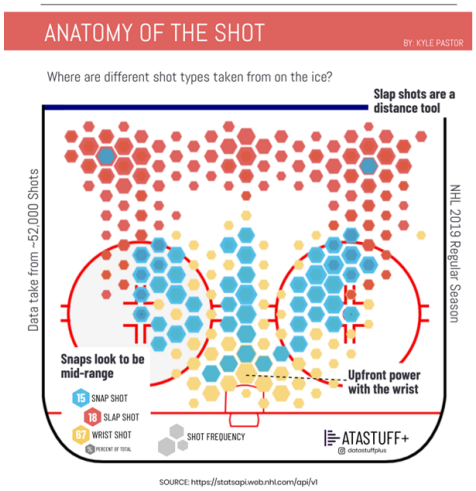
It is known that goal-based metrics can be used to more effectively predict team success within the sport of hockey [3]. With this in mind, it is natural to consider what identifiable factors are correlated with the success of a shot, as increasing goals per game is correlated with winning games. The proper identification of goal-achieving factors within hockey and the creation of more accurate goal-based metrics is of value to team management, investors, and bettors. For instance, if it was found that a certain combination of factor values related to shooting were correlated with a higher goal chance, teams could adjust their practice protocol around these values and develop more effective strategies. Investors and bettors could better predict a team's likely future success by comparing their previous shot data to ideal conditions. That is, they would be able to see whether a team or player was attempting shots with a higher likelihood of success than their competition and make informed investing decisions as a result.

Literature has found that the Fenwick rating (shots plus missed shots) and Corsi rating (shots plus missed shots plus blocked shots) can be good estimators when predicting team or player goals across a longer time frame (half a season or more) [2,3,4]. While these performance indicators are useful for longer duration goal prediction, these ratings have not been included for this study because we are predicting the probability of a shot being a goal.

b. *Overview of Problem and General Approach*

The NHL collects a plethora of different statistical measurements in relation to NHL games, plays, and players. This steady collection process has resulted in several official datasets with standardized IDs spanning many years. Using these substantial sets, with millions of rows, we wanted to explore the development of models to predict whether a shot would likely be a goal. We were primarily interested in the effect the shooting player's position or distance from the goal may have on the probability of shot success. We also considered alternative facets of NHL gameplay to explore in the way of interaction terms. We used a Kaggle dataset that contains many hockey statistics [1].

The data we selected was filtered and cleaned by our team. Even before joining the three datasets by ID, the original collection of data was so large as to be unwieldy, so we decided to only look at data collected from 2010-2011 to 2018-2019 seasons. These seasons were selected for their consistency to one another. During data exploration, NHL seasons before 2010 were found to not include both regular and playoff games, so they were no longer considered. We used two selected seasons (2010-2011 & 2018-2019) to compare models and results to identify seasonal differences. We also looked at a model encompassing the entire time period. Regression analysis was used to identify statistically significant predictors with respect to shot success and predict the probability of success of a shot given input parameters.



Source: Kyle Pastor https://i.redd.it/x2jyfyiy82t41.png

c. *Initial Hypotheses:*

We hypothesized that certain pairs of predictors would exhibit strong collinearity. For instance, interaction terms such as 'angle' and 'distance' are functions of a shot's coordinates and may capture similar variance to an event's 'st_x' and 'st_y' predictors. These are a shot's standardized x and y coordinates as they are from the perspective of the shot's attacker. We also hypothesized that horizontal distance to the net, certain shot types, and time of the shot would be the most significant factors

when predicting if a shot will be a goal. It is reasonable to anticipate that the order of significance of these factors will vary from 2010 and 2018, as the game might have changed over years.

2. *Overview of Data*

a. Cleaning Process

The NHL dataset is very large and contains many columns of descriptors. Some of these descriptors were more populated than others. When cleaning, we had to ensure that the retained columns were well populated. This data also had repeated data, which was acknowledged as human error, so we began by removing any repeat data. Next, we removed all N/A or null values in the dataset.

We performed an initial filtering of columns that could alter our results, specifically focusing on non-standard plays and play type as we wanted to center on goals during normal play. Empty net and strength of the play were removed. We then removed the description of the play column as this was a uniquely worded description for each play. We also filtered out the 'event' column of the game_plays data to only include events related to goal attempts, i.e. missed shot, shot, and goal.

This dataset is very large, consisting of data from 2000-2019 with around 4 million records. To limit the size of the data, we filtered the datasets between two seasons: 2010 and 2018. While this filtering still has many rows, as previously mentioned, this data was more consistent and only limited data slightly. For the final model we also separated the data into 2010 and 2018 individual seasons. This filtering by seasons enabled the comparison of the two models to see how the game has changed over eight years.

The three datasets we started with were: game_goals, game_plays, and game_plays_players. Each of these contain slightly different data to be used in the final model. After our initial cleaning, we joined these three datasets onto play id of the games as the key. As we continued to look at different variables to improve our models' performance, we brought in a fourth data set: games. The purpose was to create variables related to whether or not the attacking player's team was winning, losing, or tied, and the information was not available in the others (only goal differential). It turned out that these factors were statistically significant.

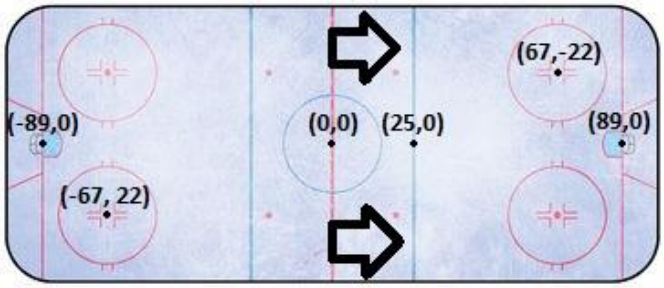
```
> str(game_plays_players)
'data.frame':   6362804 obs.
 $ play_id   : chr  "2016020
 $ game_id   : int  20160200
016020045 2016020045 ...
 $ player_id : int  8473604
 $ playerType: chr  "Winner"
 $ shooter   : num  0 0 8473
 $ goalie    : num  0 0 0 84
 $ blocker   : num  0 0 0 0
>
> str(game_goals)
'data.frame':   133345 ob
 $ play_id         : chr
 $ strength        : chr
 $ gameWinningGoal: logi
 $ emptyNet        : logi
>
```

```
> str(game_plays)
'data.frame':   4217063 obs.
 $ play_id             : chr
 $ game_id             : int
6020045 2016020045 2016020045
 $ team_id_for         : int
 $ team_id_against     : int
 $ event               : chr
 $ secondaryType       : chr
 $ x                   : int
 $ y                   : int
 $ period              : int
 $ periodType          : chr
 $ periodTime          : int
 $ periodTimeRemaining: int
 $ dateTime            : chr
1:40:50" ...
 $ goals_away          : int
 $ goals_home          : int
 $ description         : chr
st Claude Giroux" ...
 $ st_x                : int
 $ st_y                : int
```

b. Data Context

In order to help the reader better understand the data that we worked with, contextual information is provided here for the x- and y-coordinate variables. The st_x and st_y columns contain the standardized locations for each event on the ice, in that they have already been transformed to take into consideration the relative direction of each team. The st_x value of 0 is located at the center of the ice length-wise, where positive is the attacking direction and negative is the defending direction. Similarly the st_y value of 0 is at the center of the ice widthwise, where positive is towards the player's right when attacking and negative is towards the player's left when attacking. Additionally, clarification on the shot types needs to be provided. There are many different types of shots in hockey. Wrist shots

are quick-release shots using the wrist and forearms on the concave side of the hockey stick, while backhand shots use the convex side of the hockey stick. Snap shots are similar to wrist shots, except the stick doesn't remain in contact with the puck the entire time the stick is on the ice. Slap shots require more preparation, raising the stick off the ice and winding up to use the whole body to propel the puck. Tipped shots are shots diverted off of an offensive player's stick, while deflected shots are diverted off of other extensions of the player, such as gear (skates, helmet, gloves) or a body part. Wraparound shots require a player starting on one side of the net, quickly going behind the net, and then wrapping the puck around the other side of the net's post to slip the puck into the net.



c. Distance metric

As it was mentioned before, the datasets include the coordinates of the location of the shots but not their distance to the goal net. We anticipated the distance to be one of the most significant factors, if not the most significant. We wanted to choose the distance norm that would give us better accuracy. We looked at the Manhattan distance metric, l_1, and the Euclidean distance metric, l_2. Here is a simple demonstration of each metric computation based on our location setup:

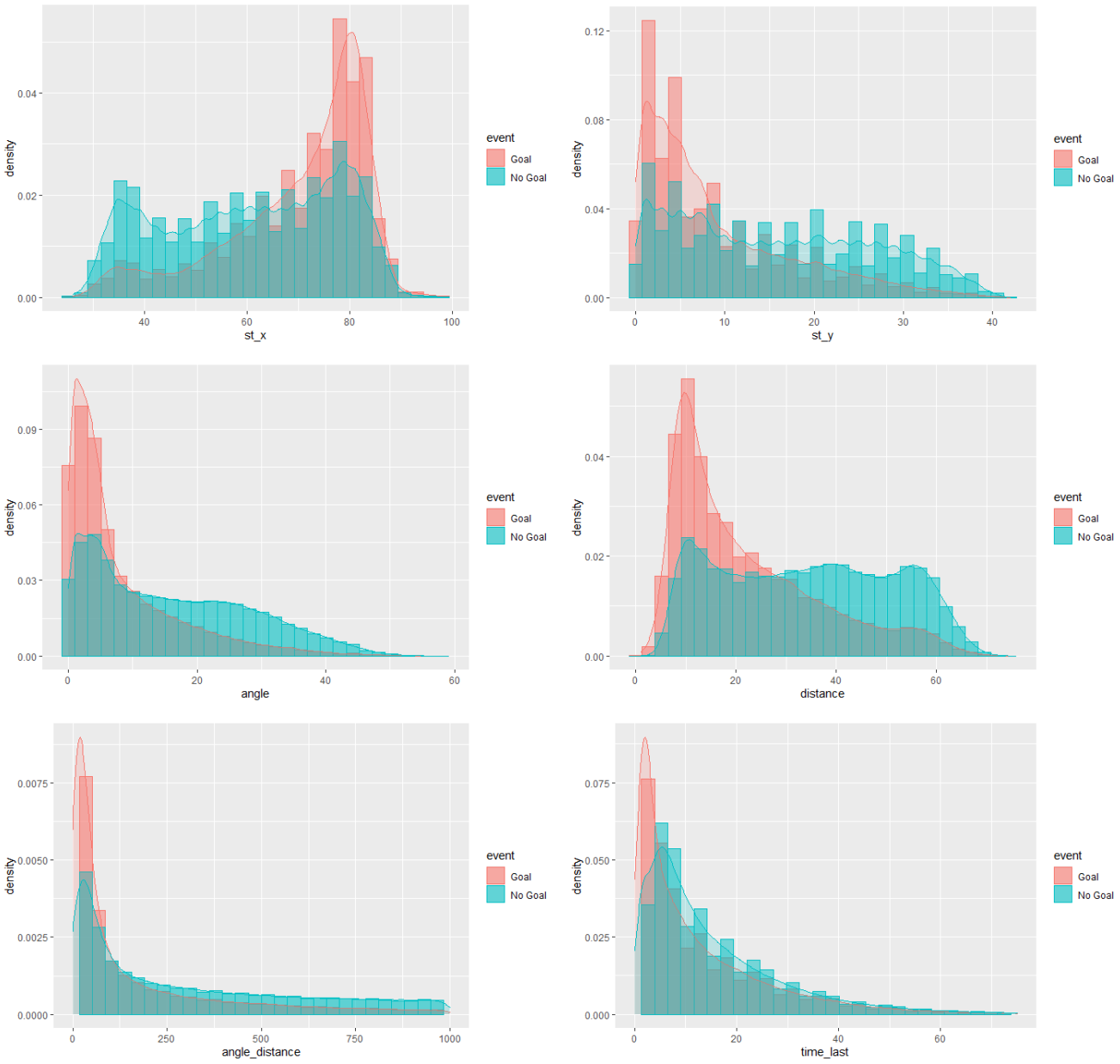
$$l_1\colon\ d(67,-22) = |89-67| + |-22| = 44$$

$$l_2\colon\ d(67,-22) = \sqrt{(89-67)^2 + (-22)^2} \approx 31.113$$

We trained our model with each metric norm separately and obtained that both metrics are statistically significant features and they gave very similar accuracy. Since the Euclidean norm is more practical and more commonly used, we used the Euclidean norm as our distance metric.

d. Data Exploration

We have explored the density plots of the variables to make contextual sense of the estimates. As st_x gets closer to 90, the chance of a goal increases. This makes sense as the shot is taken closer to the net, up until the point where the shot is behind the net. As st_y increases, the chance of a goal starts to decrease because the shot is taken further to the left or right instead of being straight on at the net. Taking the two into consideration, the chance of a goal significantly decreases as the angle of the shot on goal increases. Likewise, the chance of a goal decreases as the distance from the net increases. We also included the interaction variable, angle_distance, which shows a similar trend. Lastly, time_last also shows that the chance of a goal decreases as time since the last event increases. This can be contextualized as goals happen more frequently immediately after other events, such as other shots, faceoffs, penalties, etc. than they do when the game has gone on for a long period of time without an event.



Lastly, looking at the estimates for the shot type variables and comparing them to the base-case, wrist shots and snap-shots have an increased chance of scoring a goal, backhand shots have a decreased chance of scoring a goal, slap shots have an increased chance of scoring a goal, tipped shots are not statistically significant, wraparounds have a decreased chance of scoring a goal, and deflected shots have an increased chance of scoring a goal. This is contextually important, since a lot of teams intentionally try to tip and deflect shots in front of the goal in order to confuse the goalie or prevent them from cleanly seeing the shot. It also shows that if an individual has time to take a snap-shot or slap-shot instead of a quick wrist-shot, the chances of scoring increases.

e. Key Variables

The key variables that we explore in our project contain three location factors and three gameplay factors.

The key location factors are 1: distance between shot location and the net; 2: the horizontal distance between the shot location and the net; and 3: the angle between the shot location and the net. We believe these to be key because as distance and angle increase, a shot becomes much more difficult to score.

The key gameplay factors come from our knowledge of the game and sources investigated during this project. These factors are 4: the shot type; 5: the goal differential at the time of the shot; and 6: the time since the last event. This time since factor is time since any event, not just a previous shot. This includes play stoppages, penalties, etc. We investigated all of these key variables in our models.

3. *Overview of Modeling / Methodology*

a. Model Selection and Hyperparameter Optimization Overview

We used logistic regression models to determine the likelihood of a shot being a goal probabilistically. Predicting whether a shot is a goal is a binary classification problem. The dataset contains the shot results (labels) making the problem a supervised classification problem. The target output (shot result) depends on multiple predictor variables. Because we wanted to predict which shots would be goals, we selected a regression model, and logistic regression enables both a probabilistic result, to identify which factors increase the probability of a shot being a goal, and modeling of binary results. To validate these models, we used confusion matrices, ROC curves, and AUC.

A major issue we encountered after performing initial data cleaning is that the Kaggle dataset versions have significantly different data. Initially using the most recently released version, we found that this version had non-standardized x and y coordinates of the location of shots. This had been fixed in version 4, but the most recent dataset was an expansion of version 2 of the data. To fix this, we gave up the two extra years (2019 and 2020) present in the most recent version 2 data in order to have the consistency of standardized x and y coordinates.

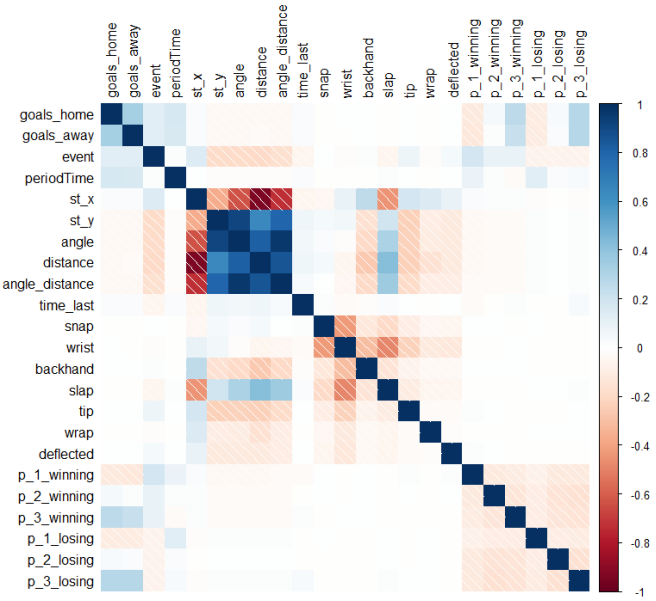
We ran into issues with collinearity of columns such as period time and period time remaining. While both of these columns could be impactful individually, they present the same data in slightly different ways. We had to investigate our data, check for collinearities, and select the columns we wanted to keep between two columns that were highly related. Having multicollinearity among predictor variables creates issues with models as regression coefficients may end up with the wrong sign and adding or removing a variable can greatly alter model fit and regression coefficient values.

At this point, we wanted to add variables related to score differential from the shooter's perspective. The original data sets used to create our first models did not include the correct factors in which to determine this so we brought in the *games* data set, which contained home and away team ids. We joined *games* to *game_plays* by game_id, and used if-else logic to determine which team belonged to the shooter. Using this, we created categorical variables, combining the period number and whether or not the team was 'winning', 'losing', or 'tied.' Each of these factors turned out to be statistically significant, except for those related to the teams being tied. Noting this, we elected to remove the 'tied' variables from our final model.

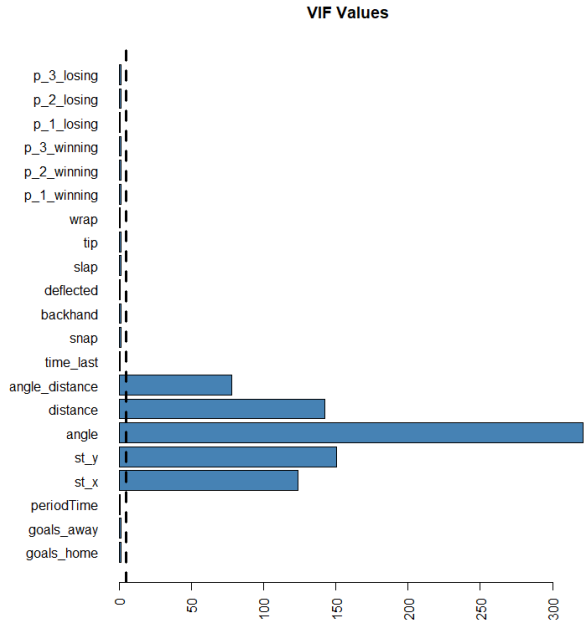
We also created indicator variables for each of the shot types in order to utilize those in the logistic regression model. Indicator variables were created for wrist, backhand, slap, tip, snap, wrap, and deflected, with wrist being utilized as the base case.

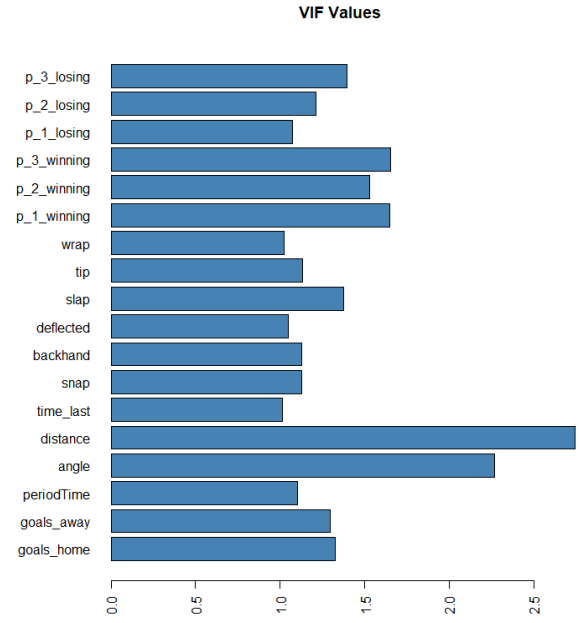
b. *Pre-Fitting Model Exploration*

After cleaning and adding variables, we identified collinearities by creating a correlation matrix, fitting an initial model, and looking at Variance Inflation Factors. The correlation matrix is shown on the right and initial VIF shown below on the left.



These two plots show that we have independent variables that are correlated. VIF greater than 5 (5 being the dashed line on the bottom left chart) means that the variables are highly correlated. We can see this in the correlation matrix as well with the deep blues and reds seen between st_x, st_y, angle, distance, and angle_distance. These variables are highly intercorrelated, so we need to remove some before generating the full models. We decided to remove st_x, st_y, and angle_distance. After removal, the VIF values, shown below on bottom right, are more reasonable.





c. *Initial Model Development*

With correlation removed, we began building usable models. We decided to develop models for three different datasets: data from 2010-2018, data from the 2010-2011 season, and data from the 2017-2018 season. These time ranges will enable visualization of differences across time. All three datasets were separated into Training (70%) and Test (30%) sets for model performance analysis. Due to the random splitting, reruns will yield slightly different results.

Three models were made using the Training sets: 2010-2018, 2010-2011, and 2017-2018.

2010-2018	2010-2011	2017-2018

```
call:
glm(formula = event ~ goals_home + goals_away + periodTime +
    angle + distance + time_last + snap + deflected + backhand +
    slap + tip + wrap + p_1_winning + p_2_winning + p_3_winning +
    p_1_losing + p_2_losing + p_3_losing, family = binomial(),
    data = training10to17)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-2.3433  -0.4148  -0.2556  -0.1469   3.7736

Coefficients:
              Estimate Std. Error z value Pr(>|z|)
(Intercept) -2.071e+00  2.157e-02 -96.025  < 2e-16 ***
goals_home   3.109e-01  4.505e-03  69.016  < 2e-16 ***
goals_away   3.509e-01  4.705e-03  74.586  < 2e-16 ***
periodTime  -4.150e-04  1.835e-05 -22.610  < 2e-16 ***
angle       -4.066e-02  1.026e-03 -39.629  < 2e-16 ***
distance    -3.103e-02  6.894e-04 -45.003  < 2e-16 ***
time_last   -1.085e-02  4.542e-04 -23.886  < 2e-16 ***
snap         2.277e-01  1.798e-02  12.665  < 2e-16 ***
deflected    3.511e-01  3.790e-02   9.264  < 2e-16 ***
backhand    -3.410e-01  2.127e-02 -16.030  < 2e-16 ***
slap         5.074e-01  2.076e-02  24.443  < 2e-16 ***
tip          1.782e-01  2.286e-02   7.797 6.33e-15 ***
wrap        -1.237e+00  6.289e-02 -19.669  < 2e-16 ***
p_1_winning  2.231e+00  1.953e-02 114.237  < 2e-16 ***
p_2_winning  1.028e+00  1.810e-02  56.776  < 2e-16 ***
p_3_winning  4.692e-01  1.985e-02  23.636  < 2e-16 ***
p_1_losing  -7.711e-01  4.932e-02 -15.634  < 2e-16 ***
p_2_losing  -5.167e-01  2.653e-02 -19.479  < 2e-16 ***
p_3_losing  -9.144e-01  2.522e-02 -36.251  < 2e-16 ***
---
signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
call:
glm(formula = event ~ goals_home + goals_away + periodTime +
    angle + distance + time_last + snap + deflected + backhand +
    slap + tip + wrap + p_1_winning + p_2_winning + p_3_winning +
    p_1_losing + p_2_losing + p_3_losing, family = binomial(),
    data = training10)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-2.2906  -0.4168  -0.2542  -0.1457   3.8364

Coefficients:
              Estimate Std. Error z value Pr(>|z|)
(Intercept) -2.0409751  0.0594878 -34.309  < 2e-16 ***
goals_home   0.2959030  0.0119338  24.795  < 2e-16 ***
goals_away   0.3374522  0.0126131  26.754  < 2e-16 ***
periodTime  -0.0004396  0.0000508  -8.653  < 2e-16 ***
angle       -0.0427413  0.0028533 -14.980  < 2e-16 ***
distance    -0.0297962  0.0019318 -15.424  < 2e-16 ***
time_last   -0.0131885  0.0012870 -10.247  < 2e-16 ***
snap         0.1314776  0.0518017   2.538 0.011146 *
deflected    0.3732760  0.1058321   3.527 0.000420 ***
backhand    -0.2634026  0.0564580  -4.665 3.08e-06 ***
slap         0.4194432  0.0570392   7.354 1.93e-13 ***
tip          0.2273024  0.0626417   3.629 0.000285 ***
wrap        -1.4041695  0.1771408  -7.927 2.25e-15 ***
p_1_winning  2.2126110  0.0534796  41.373  < 2e-16 ***
p_2_winning  1.0757529  0.0498635  21.574  < 2e-16 ***
p_3_winning  0.5492362  0.0548120  10.020  < 2e-16 ***
p_1_losing  -0.8910555  0.1459401  -6.106 1.02e-09 ***
p_2_losing  -0.4915256  0.0731184  -6.722 1.79e-11 ***
p_3_losing  -0.7800508  0.0677018 -11.522  < 2e-16 ***
---
signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

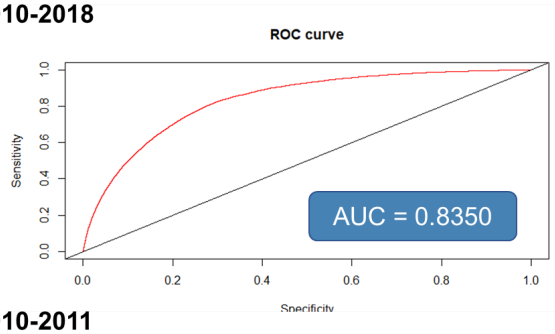
```
call:
glm(formula = event ~ goals_home + goals_away + periodTime +
    angle + distance + time_last + snap + deflected + backhand +
    slap + tip + wrap + p_1_winning + p_2_winning + p_3_winning +
    p_1_losing + p_2_losing + p_3_losing, family = binomial(),
    data = training17)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-1.7148  -0.4244  -0.2622  -0.1547   3.4233

Coefficients:
              Estimate Std. Error z value Pr(>|z|)
(Intercept) -2.032e+00  5.703e-02 -35.629  < 2e-16 ***
goals_home   2.939e-01  1.134e-02  25.926  < 2e-16 ***
goals_away   3.330e-01  1.187e-02  28.046  < 2e-16 ***
periodTime  -3.406e-04  4.814e-05  -7.076 1.49e-12 ***
angle       -3.329e-02  2.704e-03 -12.311  < 2e-16 ***
distance    -3.225e-02  1.827e-03 -17.650  < 2e-16 ***
time_last   -1.260e-02  1.230e-03 -10.246  < 2e-16 ***
snap         1.929e-01  4.739e-02   4.071 4.67e-05 ***
deflected    1.986e-01  9.637e-02   2.060   0.0394 *
backhand    -3.680e-01  5.676e-02  -6.483 8.99e-11 ***
slap         4.620e-01  5.561e-02   8.308  < 2e-16 ***
tip          7.205e-02  5.971e-02   1.207   0.2275
wrap        -1.083e+00  1.672e-01  -6.477 9.39e-11 ***
p_1_winning  2.158e+00  5.132e-02  42.052  < 2e-16 ***
p_2_winning  9.996e-01  4.758e-02  21.007  < 2e-16 ***
p_3_winning  3.756e-01  5.272e-02   7.124 1.05e-12 ***
p_1_losing  -8.396e-01  1.271e-01  -6.605 3.98e-11 ***
p_2_losing  -6.186e-01  7.167e-02  -8.631  < 2e-16 ***
p_3_losing  -9.710e-01  6.637e-02 -14.630  < 2e-16 ***
---
signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

The model summaries show that all parameters are significant for the 2010-2018 dataset. As we move to 2010-2011, snap becomes an insignificant parameter; in 2017-2018, deflected and tip become insignificant.

Goodness of fit parameters (ROC, AUC, and Confusion Matrix) are shown on the right for each of these models. Overall, each model has an AUC of ~0.83 and Confusion Matrix accuracy of ~0.91 with the 2010-2018 data having the highest values.

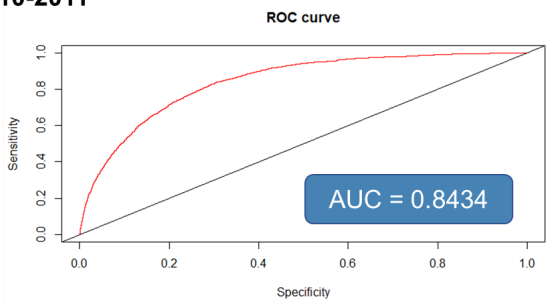
The Confusion Matrices show concerning information: the models predict very few True Positives. The models predict almost equal False Positives as True Positives, and many of the goals are lost in the False Negatives. This conclusion is also seen in the very low Sensitivity and Positive Prediction Values in the Confusion Matrices.

2010-2018



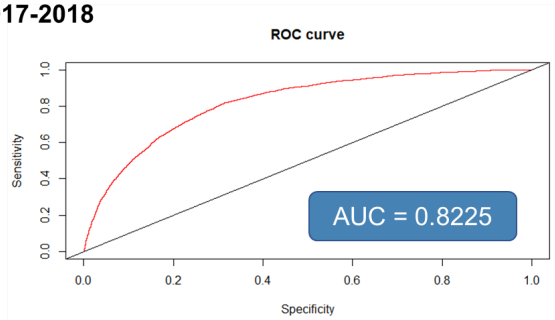

```
Confusion Matrix and Statistics

          Reference
Prediction      0      1
         0 154700  14342
         1   1049   1283

               Accuracy : 0.9102
                 95% CI : (0.9088, 0.9115)
    No Information Rate : 0.9088
    P-Value [Acc > NIR] : 0.02484

                  Kappa : 0.1221

 Mcnemar's Test P-Value : < 2e-16

            Sensitivity : 0.082112
            Specificity : 0.993265
         Pos Pred Value : 0.550172
         Neg Pred Value : 0.915157
             Prevalence : 0.091175
         Detection Rate : 0.007487
   Detection Prevalence : 0.013608
      Balanced Accuracy : 0.537688
```

2010-2011




```
Confusion Matrix and Statistics

          Reference
Prediction     0     1
         0 20478  1891
         1   131   184

               Accuracy : 0.9109
                 95% CI : (0.9071, 0.9145)
    No Information Rate : 0.9085
    P-Value [Acc > NIR] : 0.113

                  Kappa : 0.1331

 Mcnemar's Test P-Value : <2e-16

            Sensitivity : 0.088675
            Specificity : 0.993644
         Pos Pred Value : 0.584127
         Neg Pred Value : 0.915463
             Prevalence : 0.091474
         Detection Rate : 0.008111
   Detection Prevalence : 0.013886
      Balanced Accuracy : 0.541159

       'Positive' Class : 1
```

2017-2018



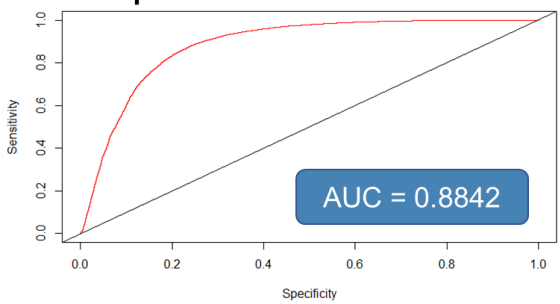

```
Confusion Matrix and Statistics

          Reference
Prediction     0     1
         0 21300  2104
         1   133   157

               Accuracy : 0.9056
                 95% CI : (0.9018, 0.9093)
    No Information Rate : 0.9046
    P-Value [Acc > NIR] : 0.3024

                  Kappa : 0.1036

 Mcnemar's Test P-Value : <2e-16

            Sensitivity : 0.069438
            Specificity : 0.993795
         Pos Pred Value : 0.541379
         Neg Pred Value : 0.910101
             Prevalence : 0.095425
         Detection Rate : 0.006626
   Detection Prevalence : 0.012239
      Balanced Accuracy : 0.531616

       'Positive' Class : 1
```

d. Oversampled Model Development

The models have decent AUC and Accuracy, but they do not predict goals well. This issue stems from the data being heavily weighted towards “no-goal” events. In hockey, many more shots are taken than scored. When looking further into the data, we actually see that the ratio of no-goals to goals was nearly 9 to 1. This large data discrepancy allowed our models to accurately predict when a goal was not going to be made, leading to a good overall predictive quality, while not effectively predicting when a goal would be scored.

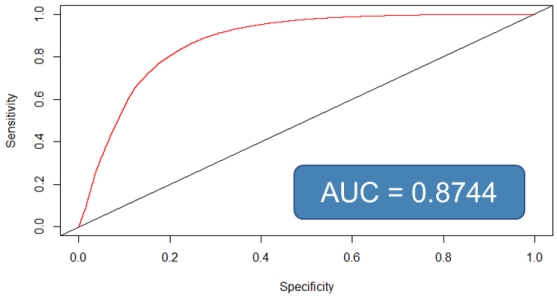
To investigate this issue, we developed two additional models using datasets with near-even scores and no-scores. We started by oversampling the positive data using the Majority Weighted Minority Oversampling TEchnique (MWMOTE). This proven and tested technique assigns each minority class weights according to their euclidean distance from the nearest majority samples. It then generates synthetic samples from the weighted minority samples using clustering [5]. We used MWMOTE to generate enough samples to have a near 50-50 split of positive and negative data. Because there is so much data in the 2010-2018 dataset, we limited this oversampling to the 2010-2011 and 2017-2018 datasets respectively.

2010-2011 Oversampled




```
Confusion Matrix and Statistics

          Reference
Prediction     0     1
         0 10441  1852
         1  3957 17515

               Accuracy : 0.828
                 95% CI : (0.8239, 0.832)
    No Information Rate : 0.5736
    P-Value [Acc > NIR] : < 2.2e-16

                  Kappa : 0.6416

 Mcnemar's Test P-Value : < 2.2e-16

            Sensitivity : 0.9044
            Specificity : 0.7252
         Pos Pred Value : 0.8157
         Neg Pred Value : 0.8493
             Prevalence : 0.5736
         Detection Rate : 0.5187
   Detection Prevalence : 0.6359
      Balanced Accuracy : 0.8148

       'Positive' Class : 1
```

2017-2018 Oversampled




```
Confusion Matrix and Statistics

          Reference
Prediction     0     1
         0 10940  2208
         1  4155 17520

               Accuracy : 0.8173
                 95% CI : (0.8132, 0.8213)
    No Information Rate : 0.5665
    P-Value [Acc > NIR] : < 2.2e-16

                  Kappa : 0.6222

 Mcnemar's Test P-Value : < 2.2e-16

            Sensitivity : 0.8881
            Specificity : 0.7247
         Pos Pred Value : 0.8083
         Neg Pred Value : 0.8321
             Prevalence : 0.5665
         Detection Rate : 0.5031
   Detection Prevalence : 0.6224
      Balanced Accuracy : 0.8064

       'Positive' Class : 1
```

Fitting models, we see much better positive predictive ability. The AUC values for both datasets improve slightly, but Confusion Matrix Sensitivity and Positive Prediction Value drastically improve. Overall Confusion Matrix Accuracy drops from ~0.90 to closer to ~0.80, but the other improvements make up for this loss in accuracy.

Looking at the responses from these models, we can see that the oversampling changed the significant variables. In 2010-2011, all parameters became significant. In 2017-2018, periodTime became insignificant while deflected became significant.

2010-2011 Oversampled

```
Call:
glm(formula = event ~ goals_home + goals_away + periodTime +
    distance + angle + time_last + snap + deflected + backhand +
    slap + tip + wrap + p_1_winning + p_2_winning + p_3_winning +
    p_1_losing + p_2_losing + p_3_losing, family = binomial(),
    data = training10_oversampled)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-3.1763  -0.5061   0.3436   0.6469   4.2783

Coefficients:
              Estimate Std. Error z value Pr(>|z|)
(Intercept)  1.357e+00  3.477e-02  39.033  < 2e-16 ***
goals_home   2.385e-01  8.678e-03  27.489  < 2e-16 ***
goals_away   2.480e-01  9.093e-03  27.279  < 2e-16 ***
periodTime  -1.068e-04  3.103e-05  -3.443 0.000575 ***
distance    -4.906e-02  1.209e-03 -40.569  < 2e-16 ***
angle       -1.120e-01  1.989e-03 -56.320  < 2e-16 ***
time_last   -2.010e-02  7.739e-04 -25.970  < 2e-16 ***
snap         1.630e-01  3.332e-02   4.893 9.94e-07 ***
deflected    4.123e-01  7.407e-02   5.567 2.60e-08 ***
backhand    -3.433e-01  3.473e-02  -9.882  < 2e-16 ***
slap         3.525e-01  4.057e-02   8.687  < 2e-16 ***
tip          1.881e-01  4.211e-02   4.468 7.90e-06 ***
wrap        -1.660e+00  8.477e-02 -19.586  < 2e-16 ***
p_1_winning  2.617e+00  3.918e-02  66.807  < 2e-16 ***
p_2_winning  1.328e+00  3.286e-02  40.425  < 2e-16 ***
p_3_winning  1.018e+00  3.636e-02  27.985  < 2e-16 ***
p_1_losing  -1.186e+00  6.445e-02 -18.404  < 2e-16 ***
p_2_losing  -5.623e-01  3.919e-02 -14.347  < 2e-16 ***
p_3_losing  -6.116e-01  3.860e-02 -15.846  < 2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

2017-2018 Oversampled

```
Call:
glm(formula = event ~ goals_home + goals_away + periodTime +
    distance + angle + time_last + snap + deflected + backhand +
    slap + tip + wrap + p_1_winning + p_2_winning + p_3_winning +
    p_1_losing + p_2_losing + p_3_losing, family = binomial(),
    data = training17_oversampled)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-3.0207  -0.5795   0.3558   0.6800   3.5769

Coefficients:
              Estimate Std. Error z value Pr(>|z|)
(Intercept)  1.232e+00  3.397e-02  36.277  < 2e-16 ***
goals_home   2.291e-01  8.081e-03  28.345  < 2e-16 ***
goals_away   2.485e-01  8.530e-03  29.133  < 2e-16 ***
periodTime   1.139e-05  2.945e-05   0.387  0.69885
distance    -5.150e-02  1.155e-03 -44.581  < 2e-16 ***
angle       -9.272e-02  1.839e-03 -50.407  < 2e-16 ***
time_last   -1.992e-02  7.553e-04 -26.377  < 2e-16 ***
snap         2.637e-01  3.104e-02   8.495  < 2e-16 ***
deflected    1.999e-01  6.727e-02   2.971  0.00297 **
backhand    -4.711e-01  3.522e-02 -13.374  < 2e-16 ***
slap         5.234e-01  3.939e-02  13.288  < 2e-16 ***
tip          6.870e-02  3.861e-02   1.779  0.07518 .
wrap        -1.368e+00  8.724e-02 -15.682  < 2e-16 ***
p_1_winning  2.477e+00  3.719e-02  66.593  < 2e-16 ***
p_2_winning  1.377e+00  3.137e-02  43.883  < 2e-16 ***
p_3_winning  8.807e-01  3.468e-02  25.392  < 2e-16 ***
p_1_losing  -1.196e+00  6.065e-02 -19.717  < 2e-16 ***
p_2_losing  -6.604e-01  3.879e-02 -17.027  < 2e-16 ***
p_3_losing  -7.371e-01  3.771e-02 -19.548  < 2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

4. Results and Conclusion

a. Model Deployment Test

Once we had our relevant factors and confirmed our model, the goal was then to test it against the 2018 season to simulate a new season's data. This will show which models will be most useful for actual deployment in predicting goals.

2010-2018	2010-2011	2017-2018	2010-2011 Oversampled	2017-2018 Oversampled
Confusion Matrix and Statistics Reference Prediction 0 1 0 71647 7050 1 760 730 Accuracy : 0.9026 95% CI : (0.9005, 0.9046) No Information Rate : 0.903 P-Value [Acc > NIR] : 0.6425 Kappa : 0.1304 Mcnemar's Test P-Value : <2e-16 Sensitivity : 0.093830 Specificity : 0.989504 Pos Pred Value : 0.489933 Neg Pred Value : 0.910416 Prevalence : 0.097023 Detection Rate : 0.009104 Detection Prevalence : 0.018582 Balanced Accuracy : 0.541667 'Positive' Class : 1	Confusion Matrix and Statistics Reference Prediction 0 1 0 71734 7090 1 673 690 Accuracy : 0.9032 95% CI : (0.9011, 0.9052) No Information Rate : 0.903 P-Value [Acc > NIR] : 0.4226 Kappa : 0.1256 Mcnemar's Test P-Value : <2e-16 Sensitivity : 0.088689 Specificity : 0.990705 Pos Pred Value : 0.506236 Neg Pred Value : 0.910053 Prevalence : 0.097023 Detection Rate : 0.008605 Detection Prevalence : 0.016998 Balanced Accuracy : 0.539697 'Positive' Class : 1	Confusion Matrix and Statistics Reference Prediction 0 1 0 71848 7221 1 559 559 Accuracy : 0.903 95% CI : (0.9009, 0.905) No Information Rate : 0.903 P-Value [Acc > NIR] : 0.503 Kappa : 0.1038 Mcnemar's Test P-Value : <2e-16 Sensitivity : 0.071851 Specificity : 0.992280 Pos Pred Value : 0.500000 Neg Pred Value : 0.908675 Prevalence : 0.097023 Detection Rate : 0.006971 Detection Prevalence : 0.013942 Balanced Accuracy : 0.532065 'Positive' Class : 1	Confusion Matrix and Statistics Reference Prediction 0 1 0 51269 1931 1 21138 5849 Accuracy : 0.7123 95% CI : (0.7092, 0.7154) No Information Rate : 0.903 P-Value [Acc > NIR] : 1 Kappa : 0.2188 Mcnemar's Test P-Value : <2e-16 Sensitivity : 0.75180 Specificity : 0.70807 Pos Pred Value : 0.21673 Neg Pred Value : 0.96370 Prevalence : 0.09702 Detection Rate : 0.07294 Detection Prevalence : 0.33655 Balanced Accuracy : 0.72993 'Positive' Class : 1	Confusion Matrix and Statistics Reference Prediction 0 1 0 51811 1914 1 20596 5866 Accuracy : 0.7193 95% CI : (0.7162, 0.7224) No Information Rate : 0.903 P-Value [Acc > NIR] : 1 Kappa : 0.2267 Mcnemar's Test P-Value : <2e-16 Sensitivity : 0.75398 Specificity : 0.71555 Pos Pred Value : 0.22168 Neg Pred Value : 0.96437 Prevalence : 0.09702 Detection Rate : 0.07315 Detection Prevalence : 0.33000 Balanced Accuracy : 0.73477 'Positive' Class : 1

From these results, we can see that the original models do not predict very many True Positives, but they also do not predict many False Positives. Due to the rarity of predicting a goal, these models could be good for learning the specific instances where a goal is predicted to be a goal and capitalizing on these. This could also be used for betting as the rarity of these types of shots could reduce risk.

The oversampled data predicts many more True Positives, but they also predict four times as many False Positives as True Positives. This model would be useful for practice and strategies for coaches and players. Even though there is only a one in five chance the shot will be a goal, there are many more shot types and locations in the oversampled data than the normal data.

Also looking at the different data sets, the individual seasons are not significantly better or worse at predicting goals than each other or the multiple season data. This lack of time-dependent models is beneficial as there will not be a need for frequent model updates.

b. Conclusion

While advanced statistics and analysis became household topics when it came to sports through baseball, with the abundance of available data and the documented success of those who bought in, other sports have obviously followed in the time since. There are many parties interested in using statistics to improve the chance of success that we have talked about in our report thus far, including the stakeholders for the teams themselves - owners, coaches, players - and betting websites. It is a testament to how far this industry has come in such a short, relative amount of time, that we (and others like us) can perform statistical analysis basically for free that was nearly impossible (either in the amount of time it takes, the software and data required to do so, or all of the above) a mere 15-20 years ago.

Not to say our models and analysis are the same level as those employed by the teams or betting websites to make others money through success today, but the underlying questions we ask to guide our analysis and the tools and software are probably not all that different.

So what comes next? As you've seen, we have used data from all games across the years. The next logical step would be to modify the models to account for an upcoming game: filtering for a specific home team versus a specific away team. As a member of a coaching staff, filtering for your team and players in the data and creating strategy based on optimized shots: shots that give the shooter the

best chance at success. The other side of that coin is strategizing the defense, specifically looking at what situations the team would decrease the shots to goals conversion rate for the opponent. For the gambling industry, this type of modeling can be used to predict the number of goals and when they could occur, based on which teams are playing.

We can also investigate other oversampling techniques to find a balance between the original models, which tend to not predict any goals, and the MWMOTE models, which predict many goals. Different techniques provide different results, and depending on the model's deployment, some techniques would be more useful than others.

Data science and analytics are flooding every field these days, with sports analytics leading the way. This project was an investigation into the use of data science in hockey to predict goals from shots. We developed multiple models that each have their predictive benefits. Further investigation could be performed into team and player based variables, oversampling methods, and tailoring models for specific use type.

References

1. *Ellis, M., NHL Game Data. Available at:* https://www.kaggle.com/datasets/martinellis/nhl-game-data
2. *Younggren, J. , Younggren, L. 2021. A New Expected Goals Model for Predicting Goals in the NHL. Available at:* A New Expected Goals Model for Predicting Goals in the NHL
3. *Found, R. 2011.Goal-based Metrics Better Than Shot-based Metrics at Predicting Hockey Success. Available at: Goal-based Metrics Better Than Shot-based Metrics at Predicting Hockey Success – The Sport Journal*
4. *Macdonald, B. 2012. An Expected Goals Model for Evaluating NHL Teams and Players. Available at: https://www.researchgate.net/publication/236687040_An_Expected_Goals_Model_for_Evaluating_NHL_Teams_and_Players*
5. *Barua, S., Islam, M., Yao, X., and Murase, K. 2014. MWMOTE–Majority Weighted Minority Oversampling Technique for Imbalanced Data Set Learning. Available at:* https://www.semanticscholar.org/paper/MWMOTE--Majority-Weighted-Minority-Oversampling-for-Barua-Islam/c3c0aaa9f961f0c81d664b0f9a030871de215b79