Christopher Boatto

Big Data Cup: Keep Possession or Dump and Chase?

Hockey is widely known as Canada's pastime. Over the years it has morphed into what it is today. From playing without being paid, to playing without goalie masks or helmets to now at the fastest pace ever, hockey is ever evolving. The next wave of change will be from the increasing amount of data being collected within each game. Coming from a baseball background, I have seen how much the game has changed on the field due to the analytics within the front offices. Previously, bunting and stealing bases were fairly common, but recently those aspects have been slowly reduced and almost removed from the game of baseball due to their inability to provide run creation. Using the Erie Otters data set provided by Stathletes, I set out to determine what event would lead most to scoring goals.

In baseball, in order to win you have to score and prevent runs, hockey is no different with goals. The statistic that is most correlated to run creation is On Base Plus Slugging (OPS.) This metric combines how many times a batter gets on base divided by at bats and his slugging percentage, which is determined by the following formula: (Single + Double * 2 + Triple * 3 + Homerun *4) / the number of at bats. It essentially shows how well a batter hits for power while drawing walks. This has caused baseball to shift from a contact-oriented game to a high walk, high power, high strikeout game. The statistical importance of each at bat in creating runs has become the forefront of every front office's priorities.

When I played hockey we were taught to dump the puck in, chase after it, work hard around the end boards to gain possession and coax a shot on goal. This never made sense to me as you would give up possession of the puck to potentially get it back in the end. Keeping possession to enter the offensive zone would make more sense to me. I saw puck possession as a valuable commodity, much like an at bat in baseball. My objective is to determine whether keeping possession would be more effective in creating goals.

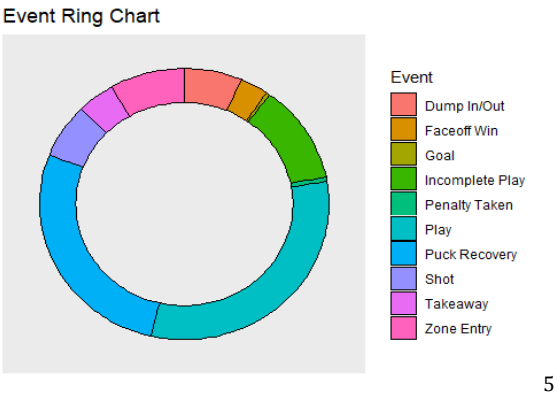
The packages used to complete the project were dplyr, ggplot2, randomForest, corrplot and pROC. Each package has its own functions as the dplyr package allows for a much quicker means to cleaning the data; ggplot2, corrplot and pROC allowed me to plot more intricate graphs and charts; and randomForest allowed me to create my predictive model. [1]

[1] Packages

I downloaded the data from The Big Data Cup website, https://raw.guthubusercontent.com/bigdatacup/Big-Data-Cup-2021/main/hackathon_scouting.csv [2]into my global environment and checked the structure and summary [3]to get a better understanding of the data. The biggest attribute to this project is 'Event'. It contains the result of each observation that occurred within the dataset. To understand the attribute better I used the unique function to gain a better grasp of what each unique outcome was. I then created a ring chart to display the frequency of each event occurring on a percentage basis compared to the entire data set. I noticed that the two outcomes that occurred the most were 'Play' and 'Puck Recovery'. I also saw that 'Dumped In' and 'Carried In' are very similar in regards to frequency. Checking the frequency of each event also allows me to understand what parameters I should set the aggregate function to when I am completing the assignment. This will be shown later in the project.

```
[1] "Faceoff Win"     "Puck Recovery"
[3] "Incomplete Play" "Play"
[5] "Dump In/Out"     "Zone Entry"
[7] "Shot"            "Takeaway"
[9] "Goal"            "Penalty Taken"
```

4



5

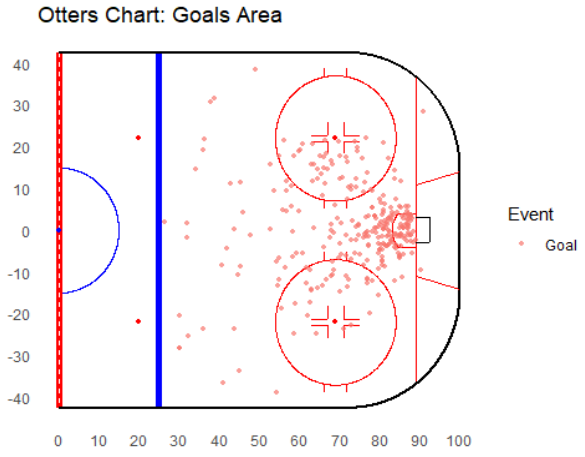
When looking at the structure and summary I noted the NA values only exist in two columns within the data set. This makes them easy to remove as later I just removed the two columns of 'X.Coordinate.2' and 'Y.Coordinate.2'. I then further cleaned the data to create separate columns for each event occurring. I found this prudent because it allowed me to isolate in binary fashion the results of each play separately. This would allow the creation of my random forest model to be a lot easier as I would simply call on each outcome as an attribute separately.[6]

Using the code found on 'https://github.com/mtthwastn/statswithmatt/tree/master/hockey-with-r' the below piece of code creates a hockey rink for my goal plot. [7]I then created a separate data set where I filtered out only the goals within the ottersData dataset to create the shot plot below. Notice the majority of the goals come from within six feet from in front of the net. This would make sense as it would give the goalie less time to react relative to a shot further away.

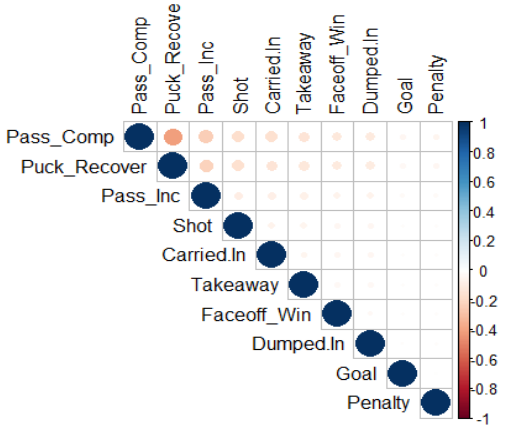
To know how each attribute works with each other, I designed a correlation plot. A correlation plot shows how well each attribute correlates to one another; the deeper the color blue the more

[2] Import Data From Website
[3] Structure and Summary
[4] Unique Outcomes in Event Column
[5] Creating Ring Chart
[6] Cleaning The Data
[7] Rink Function Creation

positive the correlation is and the deeper red the color gets the more negative the correlation is. Notice that none of the attributes really correlate well to the 'Goal' attribute. This is good for the random forest model as we should obtain a well predicted model at first glance and no one attribute should dominate.



8



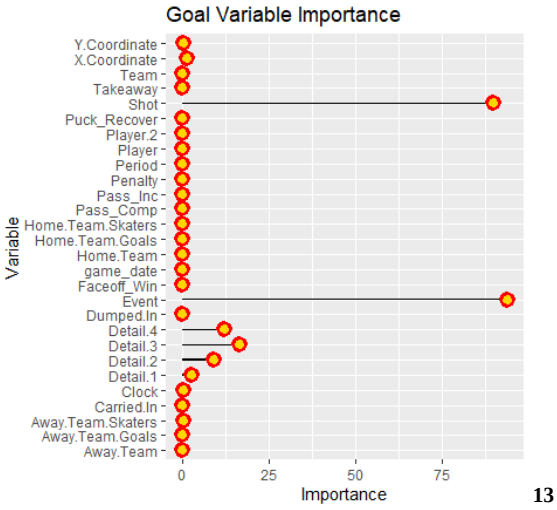
9

To begin my model creation, I split the data on a 75:25 train:test basis on a random selection of observations[10]. This is done to avoid any bias within the splitting and model creation. Without randomizing, the train function would simply take the top 75% of the data set and place it in the train data set. This would cause a potential bias and not produce an optimal prediction.

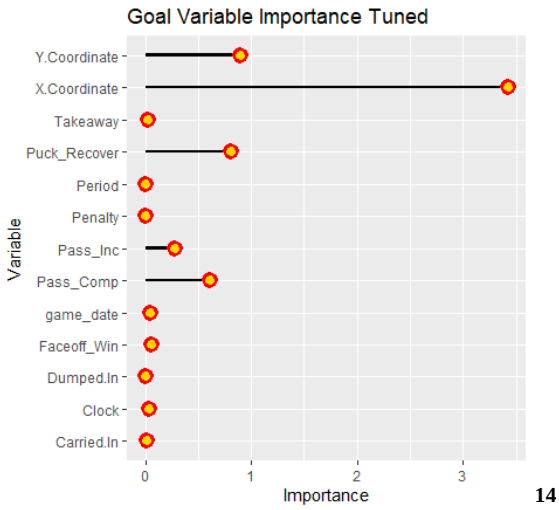
I create a base line model to gain an understanding of attribute domination[11]. I then created a new model that is tuned properly based on the important features above.[12] I also removed attributes that I thought would have no impact on the task at hand. Each attribute depicts a different way that a player could keep possession or lose it therefore having all of them in would allow for a more diverse and better trained model. Check out the difference in importance levels between the basic model and the tuned model. The tuned model has its attributes similarly weighted throughout while the basic one has two attributes that would dominate the model. I notice that in fact there was some domination from the 'Event' attribute and 'Shot' attribute. 'Event' makes sense as it contains the 'Goal' outcome as previously shown but 'Shot' is a little surprising based on the correlation matrix. It makes sense that in order to score you have to shoot the puck and the more shots you have the more goals you should score. I find it interesting that they are not correlated highly in this data set but 'Shot' would dominate the model when attempting to predict goals. Considering most goals were scored within six feet of the net I wanted to see if turnovers at specific spots would cause an increase in goals.

---

[8] Inputting the data into Rink Function
[9] Correlation Plot
[10] Training the Data
[11] Base Model
[12] Tuned Model



13



14

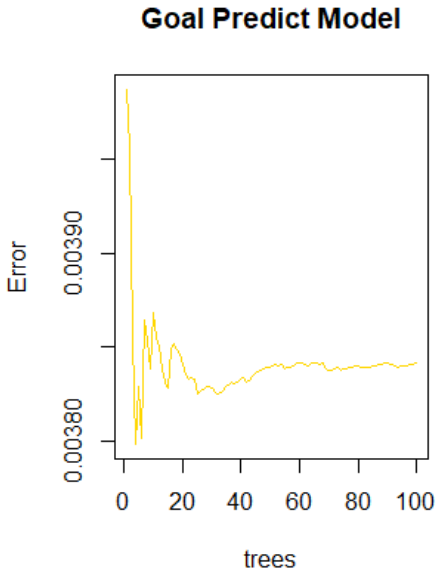
I then predicted the scores from the model created onto the train and test data sets using the response method to get a score. I used the response method because my model was based off regression and not classification. I wanted each observation to receive a score rather than it be assigned a number. This would give me a much better understanding of predicting goals.

I then bound both the predictions to the train and test data respectively, changed the prediction column name and bound both train and test sets back together to create a full data set. This allows me to then use one solid data set for aggregating later in the project to obtain my final scores.[15]

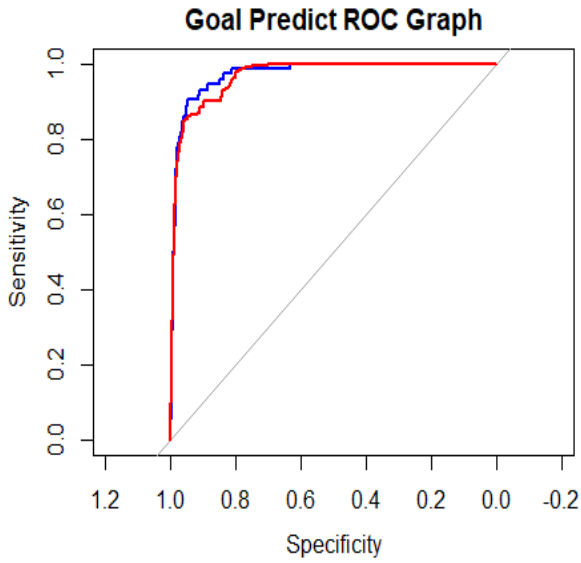
The following plot shows how well the model was created. The trace shows the confidence level in the output from the model. The chart may seem unappealing to the eye but looking along the Y axis you will notice that the model ran at a 99.99% confidence interval throughout. This is the closest you would be able to get to being 100% accurate without using the goal metric in the model itself.

The Receiver Operating Characteristic (ROC) graph below shows how the model did at predicting for the train and test data sets. The graph below shows that both the train (red line) and test (blue line) performed extremely well with no deviation. This is key because we want to see both lines move in unison to prove that the model worked just as well on each data set when predicting. Both lines went towards the upper left corner on a curve away from the diagonal line thus showing that the model worked really well in removing chance from its output.

[13] Importance Chart
[14] Importance Tuned Chart
[15] Predictions and Bindings



16



17

To complete the project, I aggregated the completed data set to determine if whether keeping possession would be more effective in creating goals. As you can see below on the first chart, all the metrics seem relatively close in range thus showing no real statistical difference.

Event <chr>	GoalPred <dbl>
Carried In	0.001575051
Dump In/Out	0.002457354
Dumped In	0.002178955
Faceoff Win	0.007583020
Incomplete Play	0.005168335
Play	0.001831421
Puck Recovery	0.002516954
Takeaway	0.005946603

18

[16] Error Plot
[17] Receiver Operating Curve
[18] Aggregate to Begin Final Steps

I then created a second table to determine if there was a more concise way to determine if puck possession led to more goals. Here I labeled each ‘Event’ as a ‘Possession’ or ‘Lost’ outcome based on their definition. I then aggregated the scores again based on the mean. I found it fair to use the mean once again as there were more ‘Event’ outcomes that swayed to a particular side thus using the sum would have caused an incomplete number. In the end there was a very slight edge to keeping possession of the puck as much as possible but no clear statistical difference between the two.[19]

Event <chr>	GoalPred <dbl>	Outcome <chr>
Carried In	0.001575051	Puck Possession
Dump In/Out	0.002457354	Puck Lost
Dumped In	0.002178955	Puck Lost
Faceoff Win	0.007583020	Puck Possession
Incomplete Play	0.005168335	Puck Lost
Play	0.001831421	Puck Possession
Puck Recovery	0.002516954	Puck Possession
Takeaway	0.005946603	Puck Possession

Outcome <chr>	GoalPred <dbl>
Puck Lost	0.003268215
Puck Possession	0.003890610

[20]

[19] Complete Aggregation for Final Charts
[20] Conclusion