Using Machine Learning Algorithms to Predict Salaries in the National Hockey League (NHL)

By Matthew Gilbody

04/27/2023

California State University, Dominguez Hills

Department of Computer Science

Committee Chair: Dr. Jianchao “Jack” Han

Committee Members: Dr. Bin Tang and Dr. Alexander Chen

Abstract

The world of analytics has grown over the last few years. Most companies are making decisions based off of data driven insights and the major sports leagues are no different. Ever since the release of *Moneyball*, the idea of using statistics to find value in players has become increasingly popular. Teams across all the major sports leagues are investing heavily into analytics and using the insights that they gain to make decisions, such as, strategy, marketing, salary, trades, etc. This paper aims to take data from the current NHL season and use various machine learning algorithms, such as, random forest, lasso regression, and multiple linear regression in order to predict player salary based on player statistics. We will look at how each algorithm works, how well they performed, and what factors go into making their predictions.

I. Introduction

Performance is a big factor when it comes to financial compensation. No matter the profession, performance is often analyzed and can lead to increased compensation or being terminated, and nowhere is this more true than the world of sports. Players are signed to term contracts and the value of those contracts is highly dependent on what kind of value the player brings to the franchise. Player salary has become an intense topic. Players across the various major sports leagues feel that their value has increased and have turned to tactics such as requesting trades, holding out, and refusing to engage in team activities in order to increase pressure on the team to give them the financial compensation that they feel they deserve. On the other side of the coin, teams often have to navigate hard salary caps, making it tricky to sign top players, while still fielding competitive teams. This paper is going to focus on the National Hockey League (NHL). The goal of this paper is to find out what goes into determining a player's salary. In the upcoming sections, I will discuss the data that is used and how the data has been transformed and broken down. I will also discuss the various algorithms that were used to obtain the predictions, the results, and the conclusions that I drew from the project.

II. The Problem and Background

Background

The NHL is one of four major sports leagues in the United States and pulls in billions of dollars each year. While still not as popular as football, baseball, or basketball, the league has boosted its popularity and according to Forbes, "Hockey team owners are scoring big, with the average NHL team value now $1.03 billion, topping $1 billion for the first time and 19% more

than a year ago" ("NHL Team Values 2022", 2022). Hockey has always been a violent sport, but with the influx of highly skilled players and lower levels of fighting, hockey has seen an increase in popularity. According to a Frontiers article by Fortney et. Al, "NHL fighting rates have diminished during the past two decades, while fan attendance has increased. A significant negative correlation exists between fan attendance and fights per game, casting doubt on fighting's entertainment value" (Fortney et. Al). With this influx of talent, comes an influx in player salaries. According to a hockeyskillstraining.com article, "In 1990, Wayne Gretzky was the highest paid player in the NHL at $3 million per season" ("NHL Salaries Over the Years") and according to the data set used in this project, the average salary is around $3 million. Wayne Gretzky is considered to be the greatest player to ever step on the ice, and even though $3 million was considered great money in 1990, when it is adjusted for inflation to the modern day, it's only around $7 million. Some players are currently making $12.5 million and that number is only going to increase as time goes on.

The biggest problems that teams have to face is constructing a competitive roster while staying within the constraints of the salary cap. This can be hard to reconcile as investing a lot of money into a few players can have lasting impacts on the long term success of a franchise. Allocating large sums of money to a few players limits the flexibility a team has, meaning they'll have to sign more average performing players. As previously mentioned, I am going to explore what goes into determining a player's salary.

Description and Objectives

The goal of this paper is to determine which factors are most important in determining a player's salary, how accurate the models are at predicting player salary, and if there are other factors outside of player performance that go into determining a player's value. Player

performance data will be run through 3 machine learning algorithms: random forest, multiple linear regression, and lasso regression, and the results of these algorithms will be compared. Some of the objectives include:

- To see which player stats are most useful in predicting player salary.
- To see which algorithm performs the best and is the most accurate in predicting player salary.
- To locate the most important performance stats and drop the least important to see if our model improves.
- To separate players based upon the positions they play to produce more accurate results.
- To explore how the algorithms pick their most important features.

Design and Implementation

Figure 1 (Flowchart)

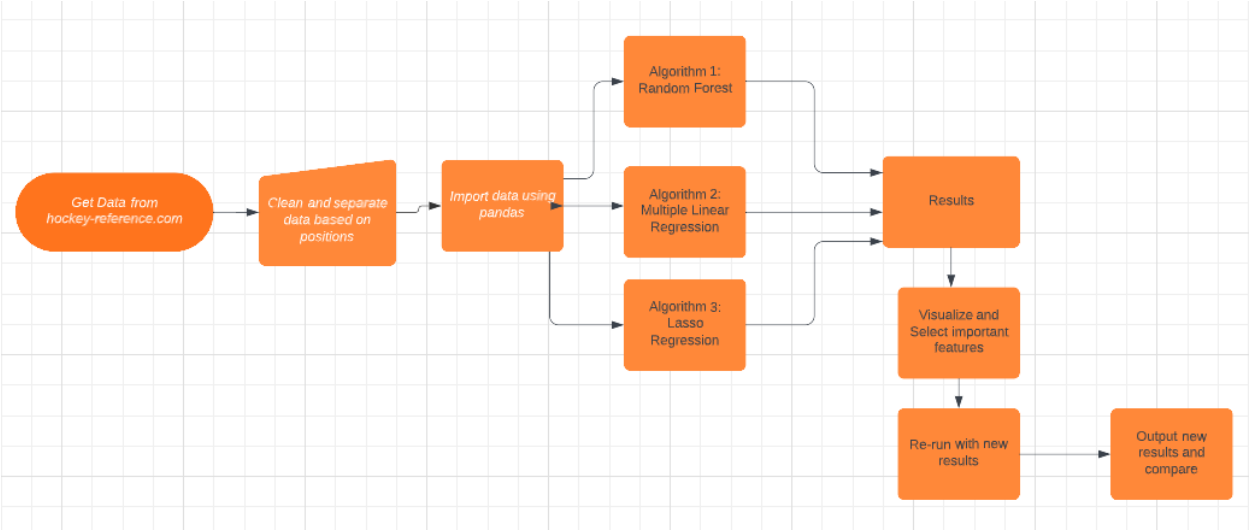


Figure 1 represents the process of this project. Our first step is to gather and load the dataset into excel. From there, I separated the players based upon their position to obtain more

accurate results in the predictions. The next section goes into further detail on how and why the data was separated and cleaned. The next steps were to import the data and run them through the 3 machine learning algorithms. From there we produced preliminary results and showed which features were most prominent in determining player salary. From there, I dropped the features that had a negative correlation to the dependent variable (salary) and re-ran the data through the algorithms to produce new results in the hopes that our predictions could be more accurate and confident. Each data set was run through each algorithm. There were a total of 3 sets of data used (centers, wingers, and defensemen). Pycharm was used as the IDE for this project and all the data was cleaned using excel.

III. Data

The data for this project was collected from hockey-reference.com and includes players' basic statistics, advanced statistics and their respective cap hit from the current 2022-2023 season. I chose cap hit versus their actual salary, because cap hit is the annual average value (AAV) of their contract. Using their salary isn't as efficient, because the way some contracts are set up, a player's actual salary can be different each year depending on the structure of the contract. I broke up the data into 3 separate parts: defense, centers and wingers. I did this because each of these positions contributes differently and I wanted to see how differently each algorithm would determine the variables that are most important for that position. Each of these positions could be broken down even more. I could separate based upon the lines they play on, how much power play time the players get, handedness they are, where they were drafted, etc. but I chose to not break it down that far. Separating them into their broad positional categories is sufficient for this study. Players who have played less than 10 games were excluded from the project as that is

not enough data to draw any meaningful conclusions. Goalies were not included in this project as well. In the sections that follow, I will mainly show visualizations from the NHL Centers data. This is to reduce redundancy and to make this paper more concise. I will share the results for all positions in the results section.

The data was cleaned using excel. I was able to connect to the web and load the data into an excel worksheet. From there, I was able to merge queries, rename the columns, delete certain columns, and separate the various positions. Once I was satisfied with the data, I converted the file from an excel workbook to a CSV (comma separated file). Another method of doing this is by using the pandas library. I mainly used pandas to read in the CSV files, get the basic data from the file, show a correlation matrix, and format the results of the predictions. Pandas is a very powerful data manipulation tool, but I chose to work in excel to clean the data. This allowed me to focus on implementing the algorithms on the data.

Figure 2 (NHL Centers Data Summary)

	Age	GP	G	PLMI	CAPHIT
count	222.000000	222.000000	222.000000	222.000000	2.220000e+02
mean	27.279279	68.621622	16.000000	-1.108108	3.117166e+06
std	4.189345	15.586602	11.558625	14.193142	2.770609e+06
min	19.000000	11.000000	0.000000	-38.000000	4.500000e+05
25%	24.000000	60.250000	7.000000	-9.000000	8.941670e+05
50%	27.000000	74.000000	13.500000	-1.000000	1.762500e+06
75%	30.000000	81.000000	22.000000	8.000000	5.000000e+06
max	38.000000	84.000000	64.000000	42.000000	1.250000e+07

Figure 1 represents a small summary of the “NHL Centers” data table. Centers have the most responsibility on the ice as they are tasked with face offs, providing offense, while also

helping on defense. Figure 1 gives a breakdown of 5 categories from the data on centers. There are 222 centers that were a part of this study. I would expect faceoff win percentage, their scoring ability, their plus/minus, and some defensive stats to play prominent roles in determining their value. We can see that the highest paid center is $12.5 million, while the lowest paid center is $450,000 and the average sits at just above $3 million.

Figure 3 (NHL Wingers Data Summary)

	Age	GP	G	PLMI	CAPHIT
count	196.000000	196.000000	196.000000	196.000000	1.960000e+02
mean	27.448980	64.923469	15.326531	-1.239796	3.271559e+06
std	3.867345	18.454358	11.153728	13.182687	2.630222e+06
min	20.000000	12.000000	0.000000	-33.000000	7.500000e+05
25%	25.000000	56.000000	7.000000	-9.000000	9.250000e+05
50%	27.000000	72.000000	13.000000	-3.000000	2.500000e+06
75%	30.000000	80.000000	21.000000	7.000000	5.135417e+06
max	38.000000	84.000000	61.000000	41.000000	1.164286e+07

Figure 2 represents a small summary of the “NHL Wingers” data table. Wingers play alongside centers and are mainly responsible for providing offense. They do have some defensive responsibility, but wingers are more expected to provide scoring. We have slightly less players in this data pool than we do centers, but we see that there are similar averages across the board. For wingers, I would expect goals scored, assists, points, plus/minus, and other offensive minded stats will be more prominent in determining their value.

Figure 4 (NHL Defense Data Summary)

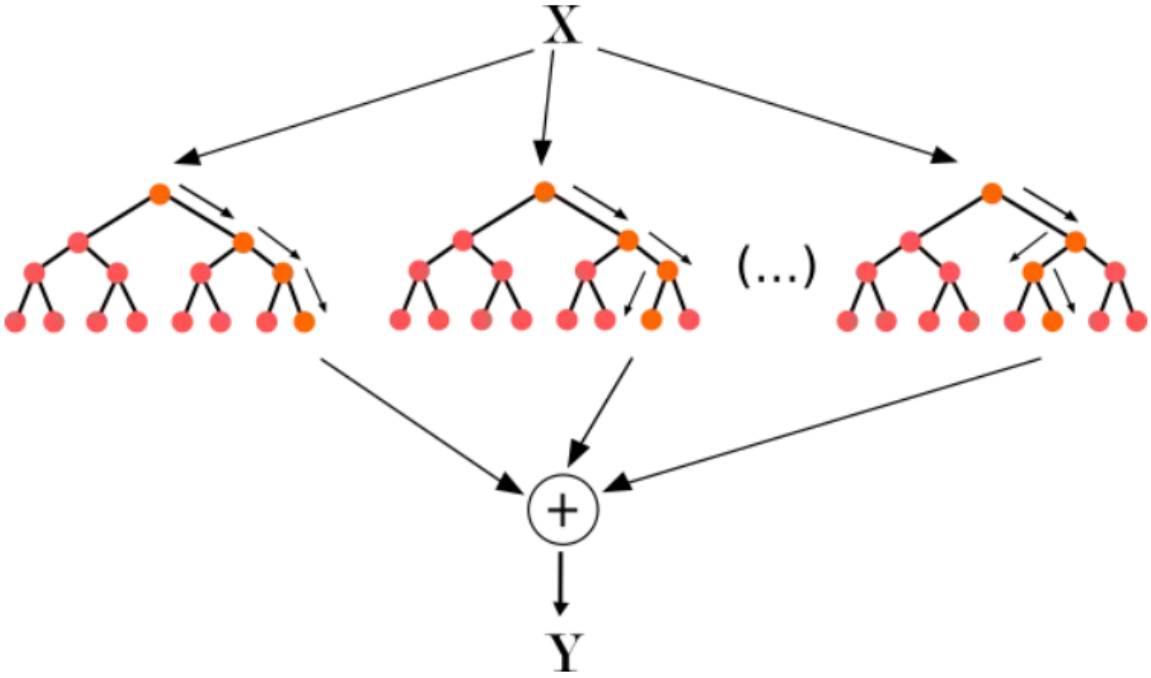
	Age	GP	PTS	PLMI	CAPHIT
count	228.000000	228.000000	228.000000	228.000000	2.280000e+02
mean	27.754386	63.403509	22.399123	2.298246	3.030652e+06
std	4.025385	19.548750	17.487904	15.225141	2.536367e+06
min	20.000000	10.000000	1.000000	-41.000000	7.333330e+05
25%	24.000000	51.000000	10.000000	-7.250000	8.599998e+05
50%	28.000000	70.500000	18.000000	3.000000	2.347075e+06
75%	31.000000	79.000000	31.000000	12.000000	4.500000e+06
max	39.000000	85.000000	101.000000	49.000000	1.150000e+07

Figure 3 represents a small summary of the “NHL Defense" data table. Defensemen focus on keeping the puck out of their own net more than they do about putting the puck in the net of their opponent. They are the last line of defense before it gets to the goalie. Higher paid defensemen bring a higher grade of offense than most defensemen in the league, but it isn’t their primary responsibility. We see similar averages in age and cap hit for all positions, but we won’t see the same when it comes to scoring. For defensemen, I would expect to see plus/minus, total time on ice, age, and other defensive stats to be more prominent in determining their value.

IV. Algorithms and Methods

Random Forest

Figure 5 (Random Forest Architecture)



The first method that I used was the random forest algorithm. Random Forest is as exactly as the name implies. It's a collection of decision trees where independent variables are chosen at random and split until a final node is reached. Once each tree has made its prediction, it will combine the results from all the trees and output a final prediction. Figure 5 is a pictorial representation of the Random Forest architecture where "x" represents the starting node. Each tree then makes a binary split and produces a result, "k". Once each tree has produced its result, all the results are combined and the average produces our final result. It can be represented in an equation as follows:

k(final) = (k1 + k2 + k3…+kn)/n

Random Forest is considered to be one of the most effective algorithms when it comes to classification and regression problems. The reason being, features are randomly selected, regardless of which features are deemed the most important, making each tree independent of one another. Li writes, "RF will take random selection for features (predictors in the research,

like age, goal.....) rather than always using all features to train the decision trees" (Li, pg. 10). In this case, the final prediction is a result of the average taken from all the individual predictions from each tree. If we were doing classification, then the final prediction would be the result of a majority vote. As previously mentioned, there are 3 different categories of NHL players. Each category was run through the algorithm.

Figure 5 (Decision Tree from our Random Forest)

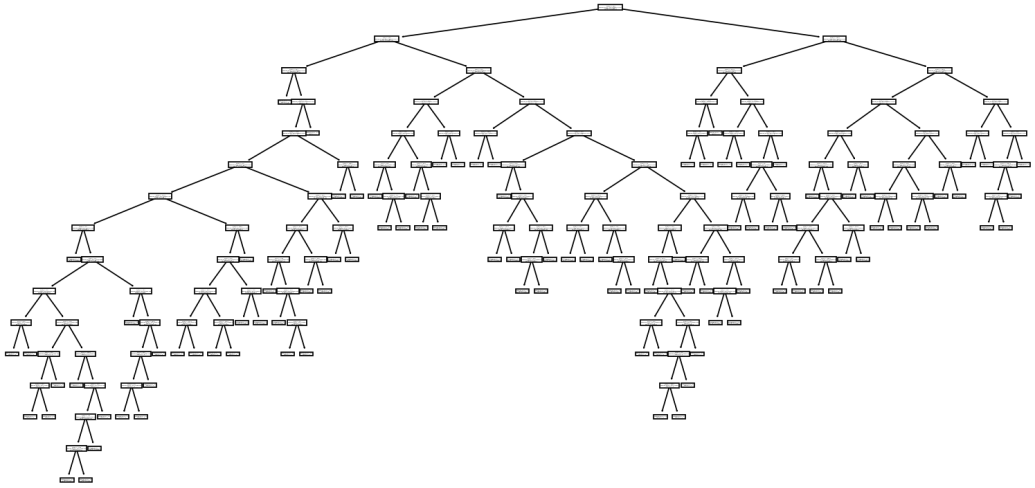
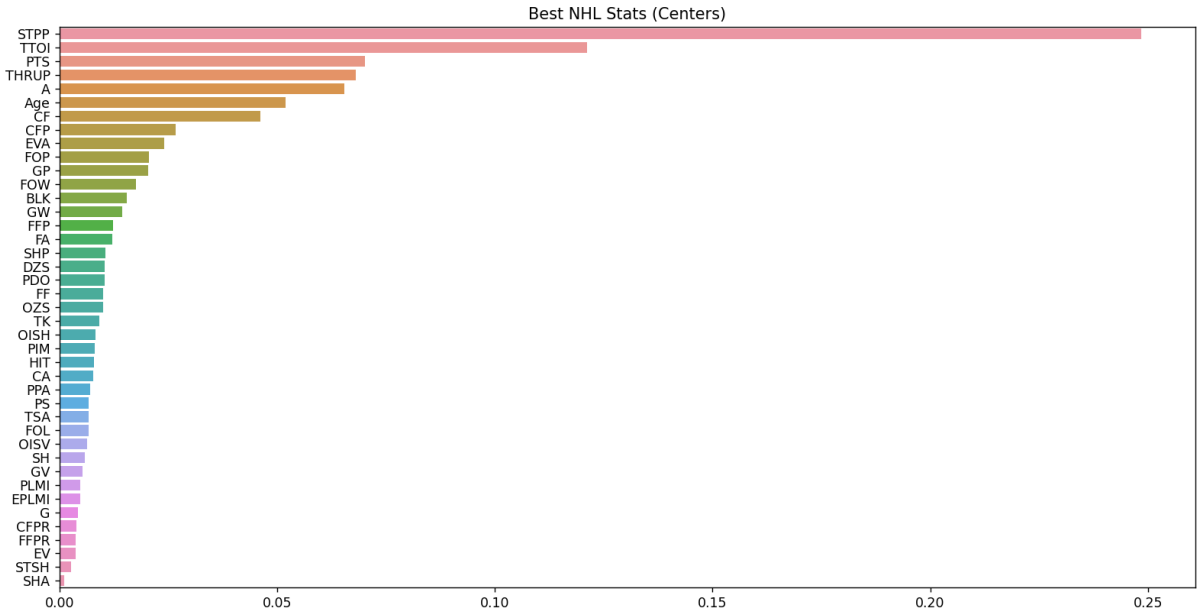


Figure 4 is an example of a decision tree from our Random Forest. Unfortunately, the image was too large and I couldn't see what each node was split at, but it offers a glimpse into the structure of the tree and what happens during the process. Each tree begins with a randomly chosen condition and bakes a binary split at each node until an ending condition is reached. Once each tree outputs its prediction, they can all be averaged out to form the final prediction. For this project, I created 100 trees for all centers, wingers, and defensemen. I used 80% of the data as the training data and the remainder was used as the test data. This was done for centers, wingers, and defensemen.

Figure 6 (Feature Selection for Centers for our Random Forest)



The above graph depicts the most important features ranked from most important to least important for centers in the NHL. The Random Forest model will calculate the features that bring less impurity to the model. The scores above are an indication of how well the feature decreases the impurity of the tree when a node is split at that feature. This can be calculated by the gini importance. According to an article from Medium, “Gini Importance or Mean Decrease in Impurity (MDI) calculates each feature importance as the sum over the number of splits (across all trees) that include the feature, proportionally to the number of samples it splits” (Lee).

There are similar figures for wingers and defensemen. The top 3 features that the model selected for centers are power play goals (STPP), total time on ice (TTOI), and points (P), while the bottom 3 features are even strength (EV), short-handed goals (STSH), and short-handed assists (SHA). While face-off win (FOW) looked to be among the better features, I would have expected it to rank in the top 3. Centers have a large responsibility with face-offs. Face-offs

really impact the game in many ways, so a player with a high face-off win percentage proves himself to be highly effective. The model was run once before feature selection was performed and then once after. Once the features were selected and ranked, the bottom two features were dropped to see if that improved our initial prediction.

Multiple Linear Regression

Multiple Linear Regression (MLR) is a machine learning algorithm that takes more than one independent variable to make a prediction on the dependent variable. Multiple linear regression uses all the features in the dataset to output its predictions, unless features are manually removed. In this case, our dependent variable is the player's cap hit, since this is what we are trying to predict, and the independent variables are the quantifiable stats, such as the amount of goals they scored. In their paper, Tonack used a MLR model for evaluating centers. Tonack states, "The multiple linear regression model solves the question of an NHL center's value by predicting salary based on various on-ice performance metrics by taking the respective coefficient values of each independent variable in the regression and combining them to determine the deserved salary based on the player's statistics" (Tonack). Tonack describes doing an experiment where 17 out of a possible 500 variables are used to predict a player's salary. The variables that were chosen need to have a decent correlation between them and the player's salary, because in order for MLR to work, variables need to be chosen based upon how they correlate to the dependent variable. The equation looks like this with 'Y' representing our dependent variable, 'B0' being our intercept, and 'B1/Bn' represent the coefficient for each independent variable, 'X1/Xn' represent our independent variables, and 'e' represents the margin of error:

Y = B0 + B1X1 + …BnXn + e

```
('Age', 158270.55439183803), ('GP', -82566.75912215574)
```

```
The intercept is: 7529099.045588573
```

Each coefficient is measured based upon the slope of the best fit line between the independent variable and the dependent variable. The above image represents the coefficients for the stats 'Age' and 'GP' (games played) and the bottom image is our y-intercept from our wingers data. For example, if we were to only use these two variables for our test, our equation would look like this:

Y = 7,529099.05 + 158270.55(Age of the player) - 82,566.76(games the player has played)

The result of the above equation would give us our 'Y' value, which in our case, is the prediction of their salary. In our dataset, we have over 30 different features that our model will calculate a coefficient for.

Figure 7 (Correlation Map for Centers)

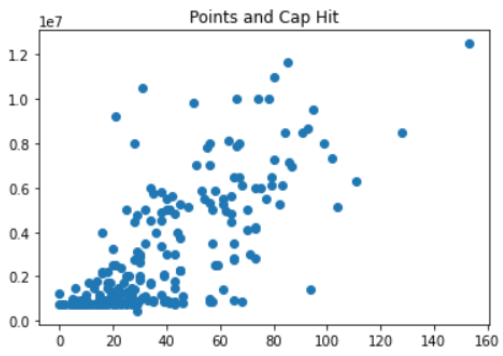
	Age	GP	G	A	PTS	PLMI	PIM	PS	EV	STPP
Age	1.000000	0.088038	0.069710	0.071823	0.073987	0.033233	0.051328	0.068181	0.014404	0.137425
GP	0.088038	1.000000	0.567018	0.579177	0.598760	0.113784	0.288527	0.515304	0.586903	0.361020
G	0.069710	0.567018	1.000000	0.834177	0.942402	0.232818	0.087300	0.957289	0.934347	0.833833
A	0.071823	0.579177	0.834177	1.000000	0.970596	0.253488	0.100895	0.917308	0.725150	0.774358
PTS	0.073987	0.598760	0.942402	0.970596	1.000000	0.255360	0.099297	0.974179	0.847622	0.833594
PLMI	0.033233	0.113784	0.232818	0.253488	0.255360	1.000000	-0.157389	0.402802	0.262763	0.102188
PIM	0.051328	0.288527	0.087300	0.100895	0.099297	-0.157389	1.000000	0.042473	0.093332	0.050562
PS	0.068181	0.515304	0.957289	0.917308	0.974179	0.402802	0.042473	1.000000	0.874949	0.827301
EV	0.014404	0.586903	0.934347	0.725150	0.847622	0.262763	0.093332	0.874949	1.000000	0.595108
STPP	0.137425	0.361020	0.833833	0.774358	0.833594	0.102188	0.050562	0.827301	0.595108	1.000000

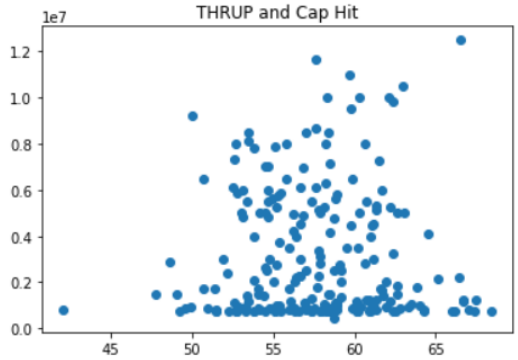
Figure 6 represents a partial snippet of a correlation map between all the different variables. The closer the score is to 1, the more the two variables show correlation between them. Unfortunately, I couldn't show the entire correlation map due to size constraints, but for purposes of this paper, showing a snippet will be sufficient. I ran the algorithm once without removing any of the variables. Once I produced this correlation map, I was able to see which variables did not show a correlation with the cap hit. I removed the variables that showed a negative correlation score and ones that were less than 0.2. By doing this, I was able to use only the variables that showed some amount of correlation, which would hopefully improve the results of the prediction. Correlation can be calculated by the following formula:

$$\text{Correlation} = \frac{n(\sum(x*y) - (\sum x * \sum y)}{\sqrt{(n\sum x^2 - (\sum x)^2)(n\sum y^2 - (\sum y^2))}}$$

In our equation, 'n' represents the amount of observations we have in our dataset, 'x' represents the number of independent variables, and y represents our dependent variable. These scores are always between -1 and 1, -1 being a perfect negative slope and 1 being a perfect slope. In our case, we want the stats that are going to show a correlation as close to 1 as possible and we want to delete the stats that show a correlation closest to 0 and below. The closer we get to 0, the more apparent it becomes that the two variables are not correlated.

Figure 8 (Scatterplots of highly correlated variable and variable with no correlation)





The above two scatter plots show 2 different variables, points (PTS) and thru percentage (THRUP) and how they correlate to cap hit for centers. As mentioned above, the algorithm was run with all the variables included and once again with the variables that showed a correlation score of less than 0.2. The MLR model showed that mostly offensive stats for centers showed the most correlation with a player's cap hit. Face-off stats showed some correlation, but it was not as high as I would have expected. Later on, we analyze the results of this algorithm.

Lasso Regression

Lasso Regression works similarly to linear regression, but aims to improve it by performing feature selection while it is fitting the model. According to Glen, "Lasso regression is a type of linear regression that uses shrinkage. Shrinkage is where data values are shrunk towards a central point, like the mean" (Glen). In our MLR model, the coefficient values aren't normalized, whereas in Lasso regression, they are. This means that the coefficient values are moved towards a value for 0. In order to calculate the penalty, we use the following formula:

$$\sum_{i=1}^{n}(y_i - \sum_{j} x_{ij}\beta_j)^2 + \lambda \sum_{j=1}^{p} |\beta_j|$$

The equation above represents how the coefficients are normalized towards 0. By normalizing the coefficients towards 0, feature selection is being performed, thus eliminating the features that aren't important. In the above equation, 'y' represents the dependent variable, 'x' represents the independent variable observations, 'β' represents our coefficients, and 'λ' represents our alpha. The second part of the equation will represent the penalty that is incurred to each coefficient and this is what brings the coefficients down to 0. While our MLR model will use all of the features, the pool of features is reduced significantly using Lasso. Our dataset features over 30 various stats, and when we use the Lasso model, about half, to more than half of the features get eliminated from consideration.

For the Lasso model, I started by assigning an alpha score of 0.1. Once I got the first prediction, I seeked to get a more optimal alpha score to see if that would improve upon the prediction. The process is very similar to the MLR model, the only difference is, the process of scaling down the coefficients and getting an alpha score that is most optimal for our data set.

V. Results and Algorithm Comparison

Each algorithm was scored by their r-squared value, mean absolute error, and the accuracy score. The r-squared value shows us how well our models explain the data from our dataset. A perfect r-squared of 1 would mean that all of the observed data can be explained by our model, which means it would perform very well. R-squared can be calculated by:

$$r^2 = 1 - \frac{\sum(yi - yp)^2}{\sum(yi - ya)^2}$$

The value 'yi' is the value that we are trying to predict, 'yp' is the predicted value, and 'ya' is the average of the sample. For the purposes of salary prediction, I would consider a r-squared value

of above 0.6 to be a decent score. The mean absolute error is used to describe the average of the difference between the actual value and the predicted value. The formula is as follows:

$$Mae = \frac{\sum(yi - yp)}{N}$$

The value of 'yi' represents our actual value, 'yp' represents the predicted value, and 'n' represents the predicted values from our test set. For the purposes of salary prediction, if the model is around 1 million, I would consider that fairly successful. The last metric used to score the model is the accuracy score. The accuracy score simply takes the number of accurate predictions and divides it by the total number of predictions. The accuracy score is the least metric I am worried about for this project, but having an accuracy score that's above 35% is what I would value to be successful.

A. Random Forest

Random Forest Results

Position	R-Squared Value	0	Mean Absolute error	1	Accuracy Score	2
	Test 1	Test 2	Test 1	Test 2	Test 1	Test 2
Centers	0.63	0.62	1421149	1430719	57.27%	57.52%
Defenseme	0.59	0.6	1322208	1288749	34.16%	34.65%
Wingers	0.45	0.45	1446732	1450383	28.36%	29.14%

The random forest model showed strong predictions for the centers and defense as their r-squared values were above or around 0.6. This means that 60% of the predictions can be explained by the model. This model was accurate due to the fact that it samples random data points from the dataset and never uses the whole set. For each position, our mean squared error was around 1.2 million - 1.4 million, meaning that our predictions were off by those numbers. Considering that salaries can range from 750,000 to 12.5 million, this amount of error is not

small, but it isn't large enough to say that the predictions were off. Each tree operates independent of the others and each tree samples random data points from the dataset. Due to this, we can get more variety of data being used, thus helping us achieve higher accuracy in our predictions. Below are the outputs of the various predictions. Each one shows what the actual salary is, the first prediction, the difference of the actual salary and first prediction, the second prediction from removing certain features, and the difference between the actual salary and second prediction.

Defense Predictions

```
First Prediction vs New Prediction:
      Salary  First Prediction  Difference 1  New Prediction  Difference 2
102  2500000        4844950.25    2344950.25      4976111.67    2476111.67
89    750000        1110983.32     360983.32      1114908.33     364908.33
197  5250000        5411990.85     161990.85      5251587.52       1587.52
204  1250000        2469727.51    1219727.51      2445000.02    1195000.02
101   750000        1506358.34     756358.34      1534300.00     784300.00
```

Wingers Predictions

```
First Prediction vs New Prediction
      Salary  First Prediction  Difference 1  New Prediction  Difference 2
31    750000        2363666.68    1613666.68      2260708.35    1510708.35
61   3000000        1156967.49   -1843032.51      1172283.33   -1827716.67
19   6650000        3741439.27   -2908560.73      3820759.51   -2829240.49
95   6500000        5018172.14   -1481827.86      5043043.21   -1456956.79
180  1450000        1604642.10     154642.10      1633533.35     183533.35
```

Centers Predictions

First Prediction vs New Prediction

	Salary	First Prediction	Difference 1	New Prediction	Difference 2
72	6500000	6347621.49	-152378.51	6357052.32	-142947.68
4	2166667	934611.65	-1232055.35	893421.67	-1273245.33
163	1875000	3753596.67	1878596.67	3462117.50	1587117.50
33	4900000	3074901.28	-1825098.72	3143229.16	-1756770.84
57	10000000	4105402.51	-5894597.49	3990491.66	-6009508.34

B. Multiple Linear Regression Results

Multiple Linear Regression Results

Position	R-Squared Value	0	Mean Absolute error	1	Accuracy Score	2
	Test 1	Test 2	Test 1	Test 2	Test 1	Test 2
Centers	0.32	0.39	1634126	1617917	13.82%	15.27%
Defenseme	0.58	0.59	1283620	1252264	34.40%	38.36%
Wingers	0.34	0.34	1585959	1523121	10.60%	16.36%

Overall, the MLR algorithm performed fairly low compared to the other algorithms. This could be explained by the fact that the entire dataset is used. Random forest provides the flexibility of using random data points and splitting them. MLR is stuck with the coefficient of each feature. Even with removing less important features, the r-squared value is fairly low for the positions, with the exception of defense. This could be explained by the higher variation in stats between centers and wingers. Defenseman stats, when it comes to offense, typically don't vary far. It could be harder for the model to use stats such as goals, points, and assists for centers and wingers, due to the fact that there is a large disparity. The mean absolute error for this model was also fairly high, ranging from 1.2 million - 1.6 million. Below are the outputs of the various predictions. Each one shows what the actual salary is, the first prediction, the difference of the

actual salary and first prediction, the second prediction from removing certain features, and the difference between the actual salary and second prediction.

Defense Predictions

	Salary	First Prediction	Difference 1	Second Prediction	Difference 2
39	750000	8.113659e+05	-6.136586e+04	6.264417e+05	1.235583e+05
169	6150000	3.478960e+06	2.671040e+06	3.732333e+06	2.417667e+06
93	7875000	5.626141e+06	2.248859e+06	6.026962e+06	1.848038e+06
62	3250000	3.100824e+06	1.491756e+05	2.445002e+06	8.049984e+05
198	5300000	5.320395e+06	-2.039461e+04	4.554373e+06	7.456274e+05

Wingers Predictions

	Salary	First Prediction	Difference 1	Second Prediction	Difference 2
127	835833	6.952434e+05	1.405896e+05	4.587068e+05	3.771262e+05
11	1000000	1.492187e+06	-4.921866e+05	2.196933e+06	-1.196933e+06
110	5250000	2.878219e+06	2.371781e+06	2.384124e+06	2.865876e+06
124	3000000	6.659274e+06	-3.659274e+06	6.138679e+06	-3.138679e+06
18	5400000	3.402453e+06	1.997547e+06	3.421048e+06	1.978952e+06

Centers Predictions

	Salary	First Prediction	Difference 1	Second Prediction	Difference 2
177	4000000	2.693520e+06	1.306480e+06	2.781398e+06	1.218602e+06
34	1125000	4.276592e+05	6.973408e+05	5.898233e+05	5.351767e+05
89	4500000	3.609401e+06	8.905993e+05	3.690375e+06	8.096254e+05
116	863333	4.882561e+05	3.750769e+05	3.920999e+05	4.712331e+05
67	750000	1.980548e+06	-1.230548e+06	2.218674e+06	-1.468674e+06

C. Lasso Regression Results

Lasso Regression Results

Position	R-Squared Value	0	Mean Absolute error	1	Accuracy Score	2
	Test 1	Test 2	Test 1	Test 2	Test 1	Test 2
Centers	0.51	0.63	1513222	1360718	30.85%	37.65%
Defenseme	0.59	0.61	1124918	1136815	19.82%	21.47%
Wingers	-0.4	-0.03	1666787	1440604	-14.14%	5.15%

The Lasso model produced fairly accurate results, with the exception of the wingers stats. This was very surprising to see, but the model found no correlation between player productivity and their salaries. However, the model showed tremendous improvement when applied to centers and stayed fairly consistent with defensemen. Even though defensemen improved very little and the wingers were still negative, the model showed improvement. This can be attributed to its feature selection and the elimination of stats that weren't as important. The model identified these stats by reducing their coefficients down to 0, making it so we didn't have to consider those stats the second time we ran the algorithm. As stated before, there were over 30 different player stats in each of the datasets, and the lasso algorithm cut the number of features to around half. This offered a better outcome because we only saw the stats that mattered to our prediction. The mean absolute error was also the lowest as it ranged from 1.1 million - 1.5 million. Below are the outputs of the various predictions. Each one shows what the actual salary is, the first prediction, the difference of the actual salary and first prediction, the second prediction from removing certain features, and the difference between the actual salary and second prediction.

Defense Predictions

	Salary	First Prediction	...	New Prediction	Second Difference
139	9500000	4.238510e+06	...	4.489925e+06	5.010075e+06
200	2750000	1.609576e+06	...	1.953997e+06	7.960033e+05
122	1400000	1.740916e+06	...	2.231264e+06	-8.312638e+05
185	850000	1.189863e+06	...	1.943429e+06	-1.093429e+06
141	6750000	5.803816e+06	...	5.180130e+06	1.569870e+06

Wingers Predictions

	Salary	First Prediction	...	New Prediction	Second Difference
83	2625000	7.505020e+06	...	6.894327e+06	-4.269327e+06
12	1200000	1.230420e+06	...	1.050258e+06	1.497421e+05
33	4000000	2.421590e+06	...	3.664933e+06	3.350669e+05
113	750000	7.647115e+05	...	1.155211e+06	-4.052106e+05
171	3250000	1.256779e+06	...	1.749042e+06	1.500958e+06

Centers Predictions

	Salary	First Prediction	...	New Prediction	Second Difference
130	766667	2.324867e+05	...	4.107614e+05	3.559056e+05
203	2750000	2.424278e+06	...	1.644988e+06	1.105012e+06
170	925000	3.472901e+06	...	2.675665e+06	-1.750665e+06
66	800000	3.166553e+06	...	2.905836e+06	-2.105836e+06
181	6125000	4.975696e+06	...	5.673656e+06	4.513437e+05

VI. Conclusion

This project aimed to investigate whether or not a player's stats could be used to predict their salary. From the results, I believe we can give a fairly accurate prediction of what a player should make, based upon their production. Out of all the algorithms, Random Forest produced the most consistent results across the board. Even though the r-squared was lower for wingers, it still out performed MLR and the lasso algorithms. I believe this can be attributed to the

flexibility of the Random Forest algorithm. As stated before, Random Forest picks random stats from the dataset and produces a series of decision trees that are independent of each other. The summary of these results was then used to output the final prediction. This allowed for more variety of the dataset to be used and considered. The Lasso algorithm showed improvement for all datasets once the coefficients that were least important were removed. This acted as feature selection, but it still was not as flexible as Random Forest. The MLR algorithm was by far the worst of the 3 algorithms used. This was due to the fact that the whole dataset was used each time, meaning there was no room for variety. I did initiate some feature selection by choosing stats that showed a correlation of higher than 0.2, but it didn't contribute to much improvement the second time the algorithm was run.

A limitation of this study was my decision to leave players with entry level contracts. Other studies argued that having them in would reduce the accuracy of the model because their salary isn't dependent on their performance. The problem with this is that some players are near the end of their entry level contracts and could be entering free agency where they can sign a higher value deal, meaning that their performance does affect their salary.

This project's goal was to see how stats were used to determine what a player should be paid. While the models showed effectiveness, I would argue that stats alone cannot be the only factor that goes into determining a player's salary. Other factors are fan attendance, a player's jersey sales, their leadership, feeling amongst teammates and coaches, and what the player means to the city and organization that he plays for. Jersey sales and ticket sales are quantifiable stats and could improve upon determining a player's salary, but what the player means to the city, leadership traits and respect from players and coaches can be a determining factor as well, and these stats cannot be quantified. What fans see is the production, or lack thereof, on the ice, but

the intangible factors play a bigger role than we might think. There is no doubt that there is a correlation between a player's production and their salary, but their salary can't be solely based on how they play.

Works Cited

Fortney, Thomas, et al. "National Hockey League Fights per Game and Viewership Trends: 2000-2020." *Frontiers*, 30 Jun. 2022, www.frontiersin.org/articles/10.3389/fspor.2022.890429/full#:~:text=Conclusions%3A%20NHL%20fighting%20rates%20have,doubt%20on%20fighting's%20entertainment%20value. Accessed 20 Apr. 2023.

Glen, Stephanie. "Lasso Regression: Simple Definition" From StatisticsHowTo.com: Elementary Statistics for the rest of us! https://www.statisticshowto.com/lasso-regression/

Lee, Ceshine. “Feature Importance Measures for Tree Models — Part I.” Medium, 8 Sept. 2020, medium.com/the-artificial-impostor/feature-importance-measures-for-tree-models-part-i-47f187c1a2c3.

Li, Chenyao, et al. *Machine Learning Modeling to Evaluate the Value of Football Players*. 2022. University College London, Final Project.

Morthi, Aparna. "How Lasso Regression Works in Machine Learning." *Dataaspirant*, 26 Nov. 2020, dataaspirant.com/lasso-regression/#t-1606404715787. Accessed 19 Apr. 2023.

Ozanian, Mike, and Justin Teitelbaum. "NHL Team Values 2022." *Forbes*, 22 Dec. 2022, www.forbes.com/sites/mikeozanian/2022/12/14/nhl-team-values-2022-new-york-rangers-on-top-at-22-billion/?sh=eb74d967deb1. Accessed 18 Apr. 2023.

Stephanie. “Lasso Regression: Simple Definition - Statistics How To.” Statistics How To, 27 Apr. 2021, www.statisticshowto.com/lasso-regression.

Tonack, Austin. *Determining an NHL Center’S Value: Salary Prediction Based on Performance Data*. 2018. Project.

hockey-reference.com

"NHL Salaries Over the Years." *HockeySkillsTraining*, www.hockeyskillstraining.com/nhl-salaries-over-the-years/. Accessed 19 Apr. 2023.

Glossary of Terms

A	assists - first or second person to pass to the eventual goal scorer
Age	age - how old the person is

BLK	blocks - how many shots a person blocked
CA	corsi
CAPHIT	cap hit - a player's aav of their contract
CF	corsi at even strength (shots+blocks+misses)
CFP	corsi for percentage
CFPR	relative corsi for percentage
DZS	defensive zone start percentage
EPLMI	expected plus minus
EV	even strength goals - how many goals a player has when it is 5 on 5
EVA	even strength assists - how many assists a player has when it is 5 on 5
FA	fenwick against (shots + misses)
FF	fenwick for (shots+misses)
FFP	fenwick for percentage
FFPR	relative fenwick for percentage

G	goals - total number of goals a player has scored
GP	games played - total number of games a player has played
GV	giveaways
GW	game winning goals - how many goals a player has that resulted in winning the game
HIT	hit - how many hits or checks a player has
OISH	team on ice shooting percentage when player is on the ice
OISV	team on ice save percentage when player is on the ice
OZS	offensive zone start percentage
PDO	shooting percentage plus save percentage
PIM	penalty in minutes - total number of minutes a player spends in the penalty box
Player	player - name of the player
PLMI	plus minus

Pos	position - what position a player plays
PPA	power play assists - how many assists a player has on the power play
PS	point share
PTS	points - goals plus assists
S	shots - how many shots a player has
SHA	short handed assists - how many assists a player has when they are short handed
SP	shot percentage - total goals divided by total shots taken
STPP	special teams power play - how many goals a player has while their team is on the power play
STSH	special teams shots - how many shots a player has on the power play
THRUP	thru percentage - number of shots that record as a shot on goal
TK	takeaways

Tm	team - what team a player plays for
TSA	total shots attempted
TTOI	total time on ice - total minutes a player has spent on the ice for the season