Prediction of NHL Rookie Salary with ML Models

Chandler Brooks John Carmack Lillian Coar Brian Horsburg

***Abstract*— NHL player salaries are capped by a set amount per team. It is then imperative that recruiting managers and talent scouts determine the appropriate salary to offer new recruits. This can be difficult given the subjective nature of 'scoring' candidate recruits. One way to do so is to look at how the NHL at large is allocating their salary budget. In this paper, several machine learning models designed to predict the salary of NHL players given some common stats among the players are presented and discussed.**

I. Introduction

The NHL has a salary cap on each team, meaning every team has the same total salary to be split among the active players on each team. Better players earn a larger portion of their teams cap, so a player's salary is normally reflective of their importance to the team.

There is also a lot of information available on the players like shot percentage, goals scored, and shots blocked that may indicate this importance and in turn their salary. The goal of this work is to use a regression model to predict player salaries from a set of statistics about each player.

II. Data Discussion

The data acquired for this report was collected from a popular dataset sharing resource, Kaggle.com. The data consists of 874 samples of rookie NHL player salaries and their corresponding statistics. Each sample contains 153 features. These features contain a mixture of categorical and numeric data.

Initially the dataset was broken into multiple files, one purpose-split for training and another for testing. The two files were conglomerated into a single file such that it could be split again using a standard ratio of 80% training data and 20% testing data. The code that does this is contained in the python script 'concatenate_data.py'.

The author of the data set conveyed that the data set was incomplete, and some features were missing for some of the represented samples. To remedy this, mean imputation was applied to the data set such that all missing features were replaced with the mean.

After doing this the data was then visualized to gain some intuition behind the features. A histogram of the salary values was produced and is presented. It demonstrates a non-normal distribution of salaries heavily weighted towards smaller values.

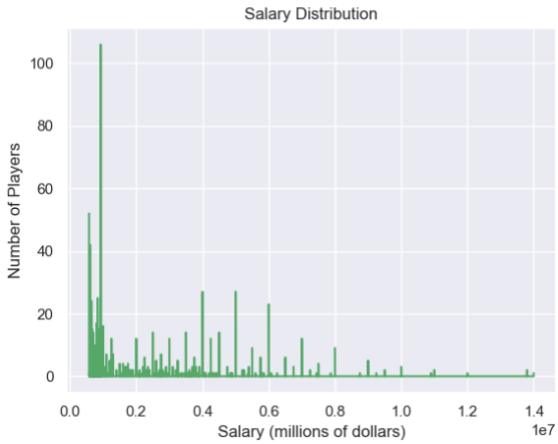

Figure 1 - Histogram of Salaries

Plots were also produced to determine the correlation between select features and salary. The features selected were ones believed to be correlated with athletic performance, such as points scored and overall draft pick.

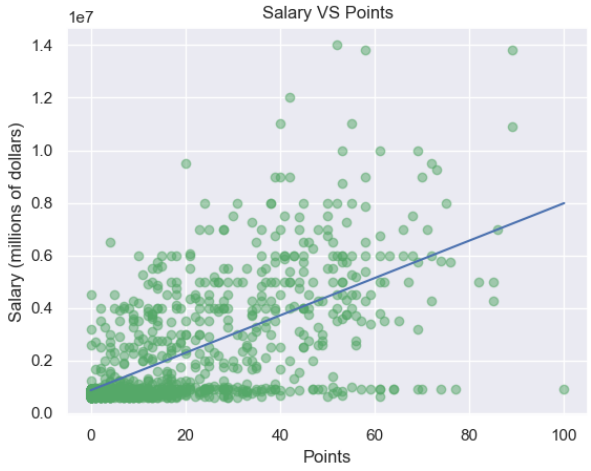

Figure 2 Salary vs. Points Scored

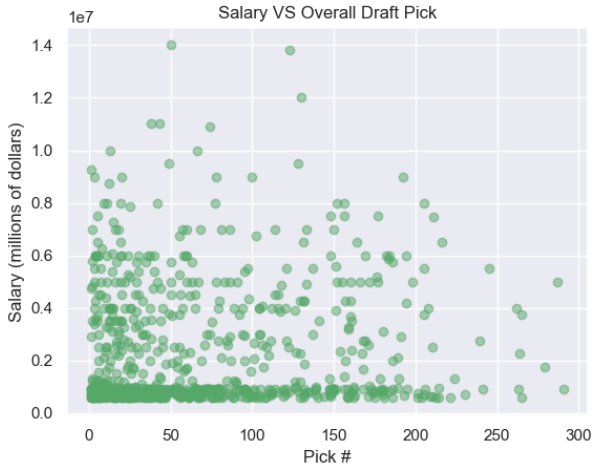

Figure 3 - Salary vs Overall Draft Pick

Of these two features, only points scored showed a marginal correlation to salary, but one is present. These plots were produced with the python script 'data_analysis.py'. It was then decided that some features must be highly correlated. With this in mind, and considering the relatively high dimensionality of the data, the decision was made to perform some measure of feature selection.

Firstly, all non-numeric features were stripped from the data. This reduced the dimensionally from 153 features to 144. Highly correlated features (>90% correlated) were compared and trimmed stochastically. This further reduced the number of features to 74. To further control for high dimensionality and feature correlation, the number of features considered limited to a certain number of the most highly correlated features. These numbers were 10, 20 and 30 This step of the cleaning process was carried out by the python script 'correlate.py.

III. Model Discussion

Three models using linear regression were chosen. These models are the ridge regression model, the lasso regression model, and the elastic-net regression model. Due to the data featuring highly correlated data, all models employ some form of regularization. This is done to counteract the redundancies caused by the correlated data in addition to combat over-fitting.

Ridge regression works by using a penalty that is based on the size of the coefficients. This encourages smaller coefficients which, in turn, improves performance of the model even with high collinearity.

Lasso regression works by adding a penalty based on the number of coefficients that are non-zero. In practice, this effectively eliminates coefficients with the goal of ultimately eliminating redundant features. Whereas ridge regression improves the performance of the model regardless of collinearity, lass regression removes the collinearity altogether.

The elastic-net model is unique in that it mixes both ridge regression and lasso regression. The idea behind this model is that by using both forms of regularization, the model can have both the sparse coefficients of lasso regression but also the minimized coefficients of ridge regression.

A grid search was performed with sklearn's GridsearchCV utility for each of these regression models, with the grid search controlling for values of alpha. These values were 0.001, 0.01, 0.1, 1, 10, 100, and 1000.

IV. Results

The first trials of the models used the 10 features most correlated with the salary, which produced an R2 score of 28% for the best hyperparameters from a grid search. Then by changing to 30 features, the R2 score achieved was greater than 99%. The accuracy is extremely high so there may be an issue or a pattern in the underlying data.

To determine if this was the case, intermediate values of most correlated features were used and compared by graphing the regression lines generated. Values of 10, 20, and 30 most correlated features were selected. Across the board, the best regression technique was ridge regression by a small margin. All graphs shown are derived from the ridge regression classifier implementation in sklearn.

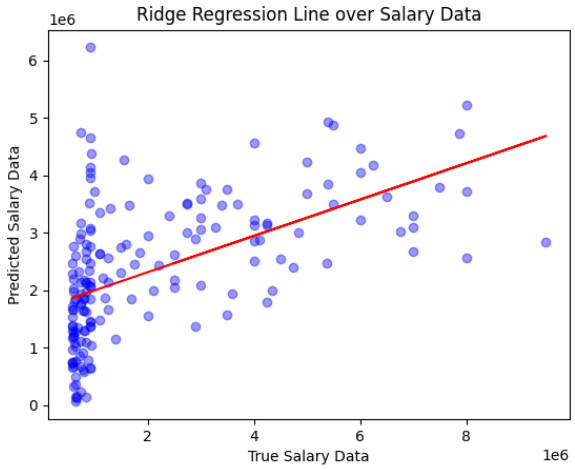

Figure 4 - Ridge Regression Line with 10 Correlated Features

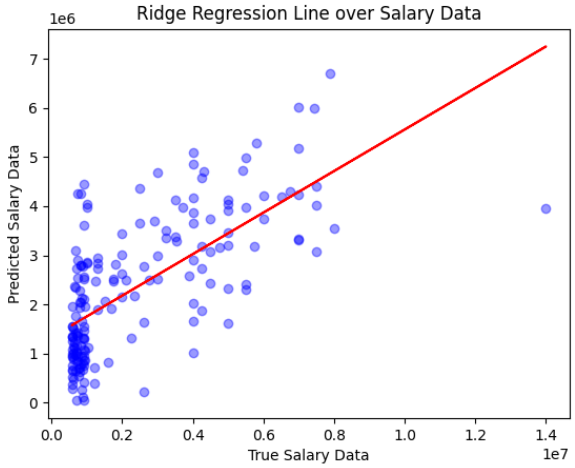

Figure 5 - Ridge Regression Line with 20 Correlated Features

Figure 6 - Ridge Regression Line with 30 Correlated Features

As the dimensionality of the data increases, the accuracy of the classifier does too until finally converging at approximately 100% accuracy at about 30 correlated features. A table of accuracies and other data collected for each value of correlated features selected in presented. This data and the corresponding graphs produced was obtained with the python script 'regression.py'.

"Classifier": "Ridge",
"Parameters": {
"alpha": 100
},
"Test Score": 0.3642436773282113,
"Training Score": 0.31189176627275983

"Classifier": "Ridge",
"Parameters": {
"alpha": 1
},
"Test Score": 0.4231577747383698,
"Training Score": 0.3552342048915308

"Classifier": "Ridge",
"Parameters": {
"alpha": 0.001
},
"Test Score": 0.9999999999954579,
"Training Score": 0.999999999991679

Figure 7 - Table of Best Performing Classifiers for 10, 20, and 30 Correlated Features

The salary caps being the same on each team in the NHL means that teams must be precise to pay a player exactly what they are worth. Any higher would damage the salary cap and any lower may let them get a better contract elsewhere. A player's value is backed up by all the data we use in our models so this precision may help explain the very high accuracy in our predictions.

V. CONTRIBUTIONS

Chandler Brooks	• Data Acquisition • Documentation • Regression Plots
John Carmack	• Data Processing • Model Selection • Documentation
Lillian Coar	• Model Selection • Documentation
Brian Horsburg	• Data Processing • Model Selection • Documentation