Predicting NHL Player Salaries (Statistical Learning Case Study)

MATH 5P87 Statistical Learning

Department of Mathematics and Statistics

Faculty of Mathematics and Science Brock University

St. Catharines, Ontario

Andrew Kenig, Brad Klassen, Kinza Faiyaz

Prof. William Marshall

March 27, 2020

Contents

1 Abstract

This project aims to build statistical learning models that will predict the salaries of NHL players during the 2016-2017 NHL season. The project explores linear models including ridge regression and LASSO regression, as well as, non-linear models including kernel smoothing regression, XGBoost regression and decision tree regression. K-fold cross-validation is used to rigorously test and evaluate the models. The LASSO regression model is the most optimal model as judged by root mean squared error, with an RMSE of 1447595. These results can be used by sports agents and general managers to accurately determine the value of their athletes salaries.

2 Introduction

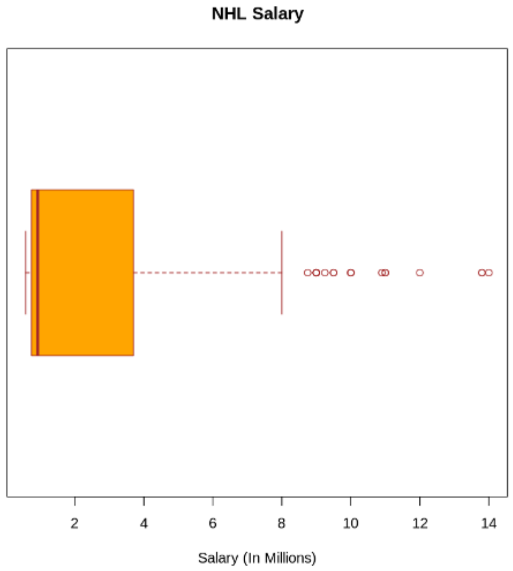
In the age of big data, the field of sports analytics is becoming increasingly transdisciplinary, combining domain specific knowledge from sports management with the statistical and computational tools of data science. Hockey is a sport that generates massive amounts of data with no shortage of opportunity for analysis. As statistics becomes more integrated into professional sports, having an analytical edge gives both athletes and teams a competitive advantage. The goal of the project is to use machine learning to predict the salaries of NHL players during the 2016-2017 season. This end-to-end machine learning project involves data manipulation, exploratory data analysis, model training, model optimization, model evaluation and predictive modelling.

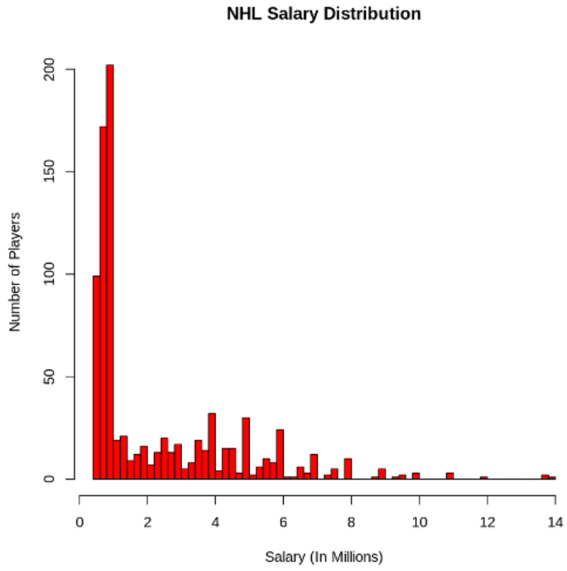
Along with the other professional sports leagues such as the NFL and the NBA, the NHL is a cap dominated market which means that only a certain amount of money can be spent on players' salaries. General managers find it challenging to optimize the allocation of the salaries to the players. A league wide salary cap system helps in reducing the instances of players being paid far more than their worth. However, it is not fully preventing managers from over estimating players' worth. By using regression methods, teams can prevent overspending on player contracts and thus build a more competitive team.

In this project, we use supervised learning regression techniques to develop models that can be used by the general managers to accurately determine the worth of the player. General managers can also use these statistical learning models to sign players to a lower salary early in their career before they reach their full potential.

3 Data

The data set was obtained from Kaggle, a popular data science website. The user acknowledged that the data set was acquired from a popular hockey analytics website called Hockey Abstract.[1] The data set contains 874 observations, 153 inputs, and 1 output. The inputs are a mixture of categorical data (country born, year drafted, etc.), discrete data (goals scored, penalties incurred, faceoffs won, etc.) and continuous data (assist percentage, player height, time on ice, etc.). The output variable is the salary of the player (in USD). The box plot and histogram below show the summary of the salary of the players.





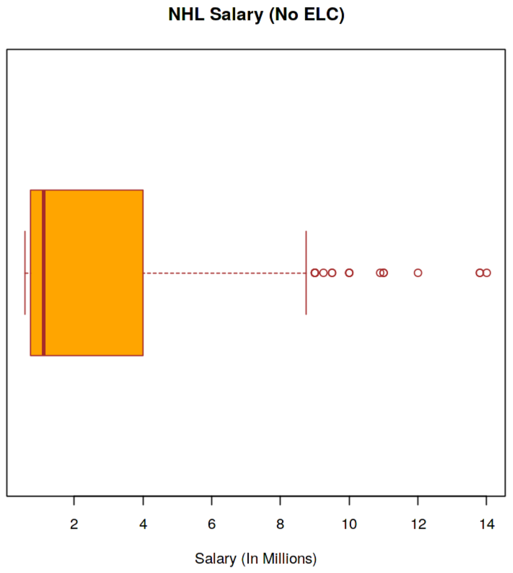
The lowest salary of the players in the 2016-2017 NHL season is $575,000, the average salary is $2,325,289, and the highest salary is $14,000,000. During the given season, the league minimum salary was $575,000, meaning no team is allowed to pay a player less than the amount stated. The salary cap for each team during the given season was $73,000,000, this amount must cover all of the players salaries on the team. The maximum salary that a player may receive is 20% of the team's salary cap. Using this information we are able to determine the range of the output variable. The minimum a player can earn in the given season is $575,000 and the maximum a player can earn is $14,600,000.

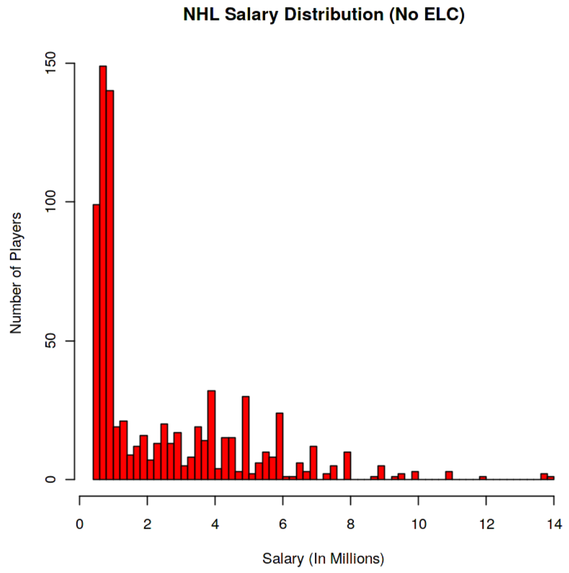
3.1 Entry-Level Contracts

From the histogram above, we can see that the data is skewed right. This means that our model may give the most reliable predictions for lower salaries. To overcome this, we decided to remove players with entry-level contracts from the data set. Since entry-level contracts are fixed

[1] *Hockey Abstract*. URL: `http://www.hockeyabstract.com/`.

by the league and not negotiated by the player agents and general managers, this may affect our predictions and introduce a bias. We reviewed 2014, 2015, and 2016 NHL drafts to determine the age of the players at the time of the draft and any player whose contract extended into the 2016 season was removed. Some of the athletes we removed were players from age 18 to 21 with 3 year contracts, players age 22 to 23 with 2 years contracts, and players age 24 with 1 year contracts. Removing the entry-level contracts will improve the predictive power as well as the accuracy of the machine learning models. Unfortunately, many of the birth dates in the original data set were incorrect. According to the data, there were many occurrences where players were born in the year 2000 or later, yet were still drafted before the 2016 season. For example, in the data set it states that Connor McDavid was born on January 14, 2001. Yet Mcdavid was actually born on January 13, 1997. As a result, the entry-level contracts needed to be removed manually. We have removed entry-level contracts using excel and have created a new data set to be used for further analysis. The box plot and histogram below show the summary of the salary of the athletes with entry-level contracts removed.



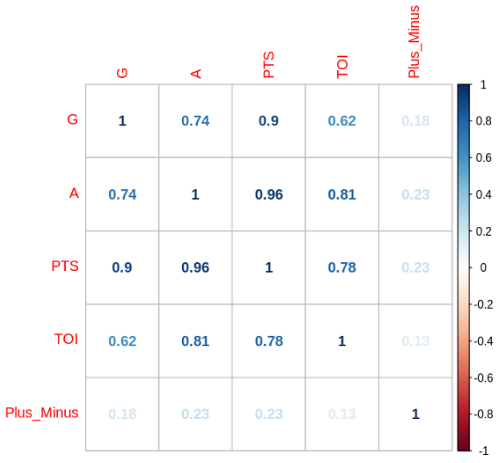


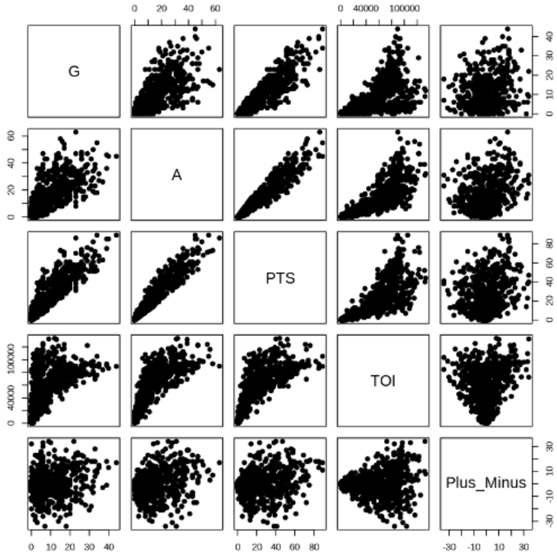
3.2 Missing Value Imputation

Many statistical learning models are incapable of handling data with missing values, therefore we must impute new data. We have used a common method for imputing missing data by filling the missing value with the mean of the statistic.

3.3 Correlation

The correlation plot below shows the correlation matrix for five selected inputs that were thought to be important as judged by hockey knowledge. (Goals (G), Assists (A), Points (PTS), Time on Ice (TOI) and Plus/Minus (+/-)). These statistics will be described in a table below.





Below is a table of notable inputs and their correlation coefficients with respect to salary. Rows one to three are inputs with the largest positive correlation, rows four and five are inputs with the largest negative correlation, and rows six to nine are inputs that are chosen using hockey knowledge. Due to the large number of inputs in the data set, only a few important inputs have been described below.

Statistics	Description	Correlation Coefficient
xGF	Team's expected goals while this player was on the ice	0.6902
GF	Team's goals while this player was on the ice	0.6859
SCF	Team's scoring chances while this player was on the ice	0.6839
DftYr	Year the player was drafted	-0.4224
Ovrl	Where the player was drafted overall (position)	-0.2775
PTS	Number of points, measured by goals plus all assists	0.6723
G	Number of goals	0.5543
A	Number of goals the player has assisted	0.6792
TOI	Time on Ice	0.6322

4 Methods

Due to the output variable being continuous, this project explores regression methods including two linear models and three non-linear models. The linear models tested include LASSO regression (L1 loss function with absolute penalty term) and ridge regression (L2 loss function with quadratic penalty term). The non-linear models that are tested include decision trees (node impurity uses a L2 loss function), kernels (data driven approach with no loss function) and a bonus gradient boosting method known as XGBoost (L2 loss function). The project uses k-fold cross validation as the resampling technique that is used to estimate any hyperparameter and assess the machine learning models. The evaluation metric that is used to judge the performance of the regression models is root mean squared error (RMSE). RMSE was chosen over MSE due to the massive size of the model errors.

4.1 Ridge Regression

In our data set, there are 153 input variables. From correlation plots shown above, we noticed that these variables are highly correlated, as a result multicollinearity seems to exist in our data set. Multicollinearity makes coefficients highly sensitive to the small changes in the model, increases the variance of regression coefficients, and reduces the precision of the estimated coefficients which weakens the power of the statistical model. As a remedy, ridge regression can be used to add some degree of biasedness to the regression estimates for reducing the variance.[2] Lambda is a hyperparameter used in ridge regression that controls the amount of shrinkage. Lambda places a penalty on the beta coefficients and shrinks unimportant beta coefficients towards zero. Ridge regression adds a penalty that is equal to the square of the magnitude of the coefficients, shrinking them to small values but not eliminating them entirely. Prior to performing ridge regression, the data must be standardized because the regularization technique depends on the magnitude of each variable. Using k-fold cross validation with 10 folds, the program performs ridge regression for a sequence of lambda values.

4.2 LASSO Regression

In ridge regression, all the predictors are included in the final model. The penalty, lambda, shrinks coefficients towards zero but it does not set any of them exactly equal to zero. LASSO regression is an alternative to ridge regression that overcomes this disadvantage. LASSO regression adds a penalty to the absolute value of the magnitude of the coefficients. This penalty has the effect of forcing some of the coefficient estimates to be exactly equal to zero when the tuning parameter lambda is sufficiently large. Hence, LASSO is capable of performing variable selection. LASSO yields sparse models that involve only a subset of variables.

[2] Bahadır Yüzbaşı, Mohammad Arashi, and S Ejaz Ahmed. "Shrinkage Estimation Strategies in Generalised Ridge Regression Models: Low/High-Dimension Regime". In: *International Statistical Review* ().

4.3 Decision Tree Regression

Decision trees are significant predictors and embody an explicit representation of the structure in a data set. They explore the structure of the data while developing an easy to visualize decision rule for predicting outcome. Decision trees are constructed via an algorithmic approach that identifies a way to split a data set based on different conditions. We created a fully grown decision tree using the training data. The training data is in the root node which gets divided into the branches to define two new nodes. This process continues until the stopping criteria is met. Fully grown decision trees tend to overfit the model. Hence, we prune the tree by trimming its branches. We prune the tree to the nodes where the RMSE is the minimum. Pruning reduces the complexity of the model and improves the prediction accuracy.

4.4 XGBoost Regression

In addition to decision trees and kernel smoothing regression, Extreme Gradient Boost (XGBoost) is another non-linear technique explored in this project. The main idea of boosting is to combine a series of weak classifiers with low accuracy to build a strong classifier with better performance. XGBoost improves predictive performance and reduces the variance of the final model by using only a random subset of data to fit each new tree.[3]

4.5 Kernel Smoothing Regression

Kernel smoothing regression is one of the non-linear techniques we used to further look into the structural features of the data. The basic principle is that local averaging or smoothing is performed with respect to a kernel function. Since kernel methods cannot be applied naively to higher dimensional data, we used an additive model of only the top five best inputs as judged by the XGBoost model. An additive model estimates five one-dimensional kernel regression functions. We used the epanechnikov kernel function because it has the highest efficiency relative to other common kernels. A constant bandwidth of 0.5 was estimated using the back fitting algorithm and 10-fold cross-validation. Bandwidth refers to the maximum distance from the kernel's center at which mass is spread.

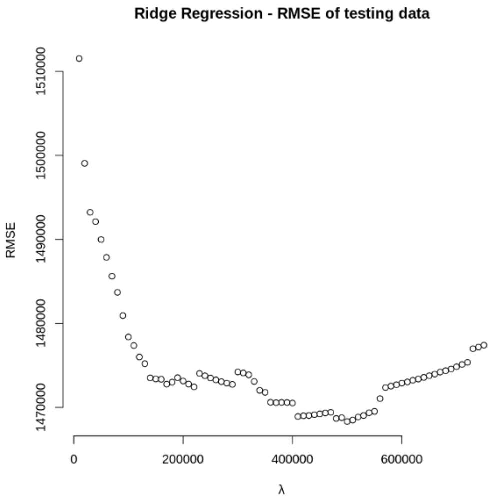
[3] Minghua Chen et al. "XGBoost-based algorithm interpretation and application on post-fault transient stability status prediction of power system". In: *IEEE Access* 7 (2019), pp. 13149–13158.

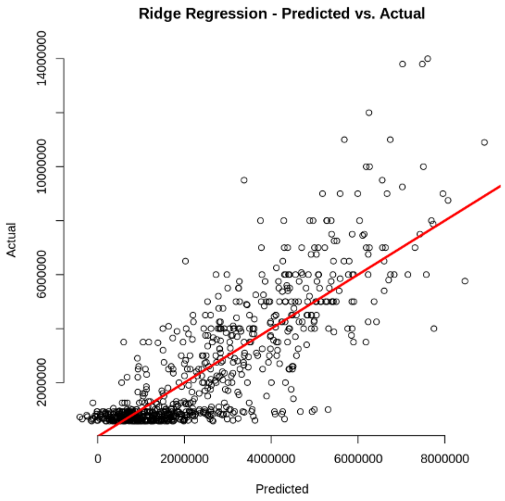
5 Results

Each model explored in the project had various hyperparameters that were tuned to allow for optimal performance.

5.1 Ridge Regression

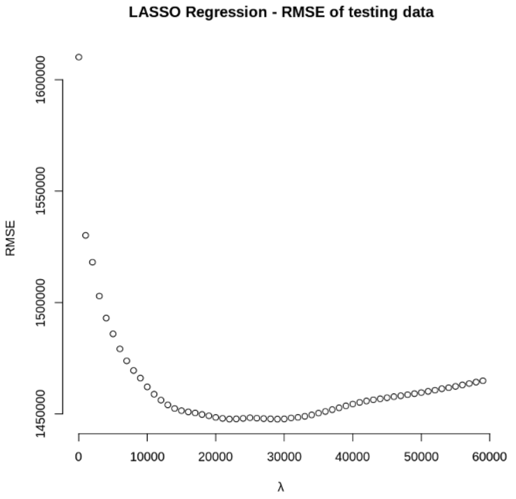
The optimal ridge regression model had a lambda value of 489312 with a corresponding RMSE of 1468157. The plot below highlights how the lambda value affects the RMSE.

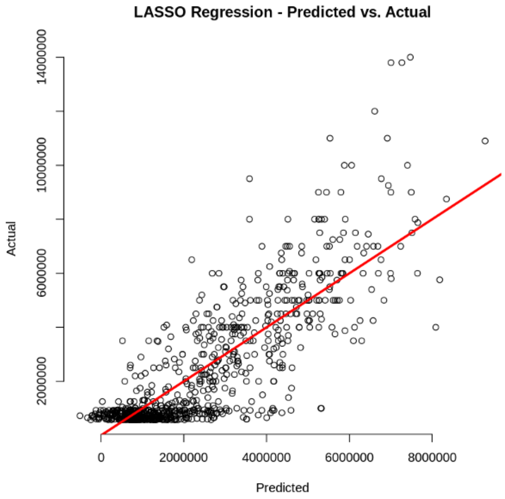




5.2 LASSO Regression

The optimal LASSO regression model had a lambda value of 28392 with a corresponding RMSE of 1447595. The plot below highlights how the lambda value affects the RMSE.

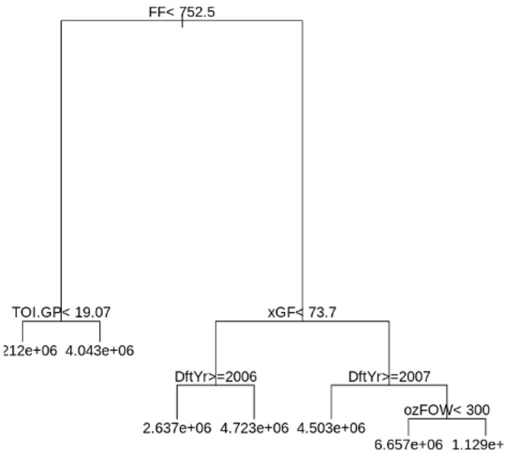


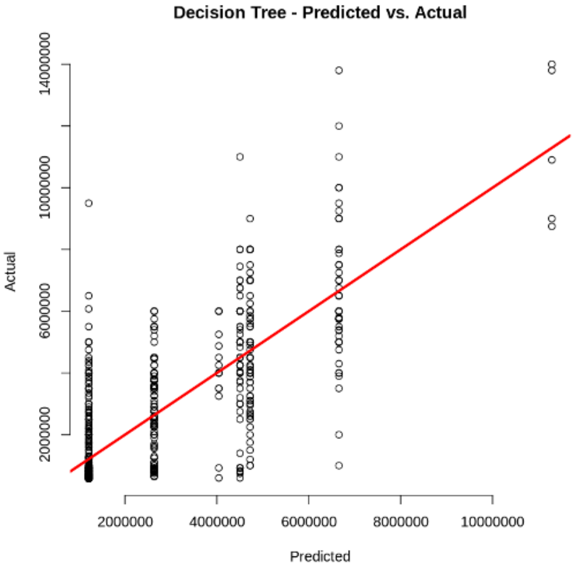


5.3 Decision Tree Regression

The optimal decision tree regression model had a complexity parameter of 0.01908 with a corresponding RMSE of 1463413. The plot below highlights the tree-like structure of the model and the inputs that were used to create the tree. Shown below is a pruned decision tree with the optimal complexity parameter. The table below consists of the variables that were used to build the pruned decision tree.

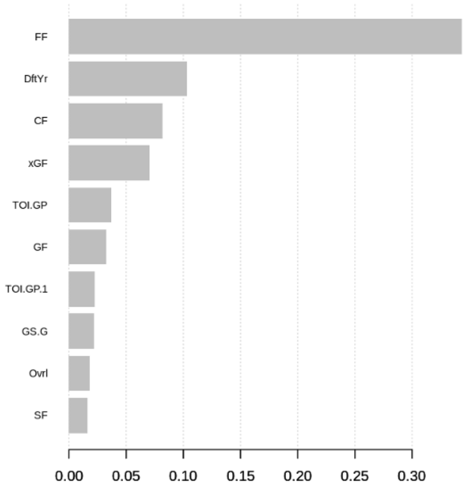
Statistics	Description
FF	The team's unblocked shot attempts (Fenwick, USAT) while this player was on the ice
TOI.GP	Time on ice divided by games played
xGF	The team's expected goals (weighted shots) while this player was on the ice, which is shot attempts weighted by location
DftYr	Year the player was drafted
ozFOW	Faceoffs won in the offensive zone
GS.G	The player's average game score





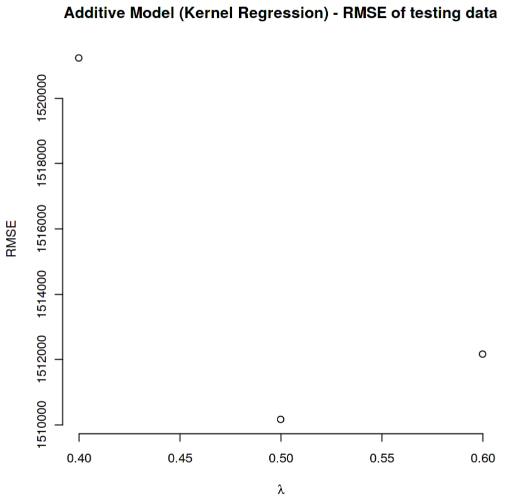
5.4 XGBoost Regression

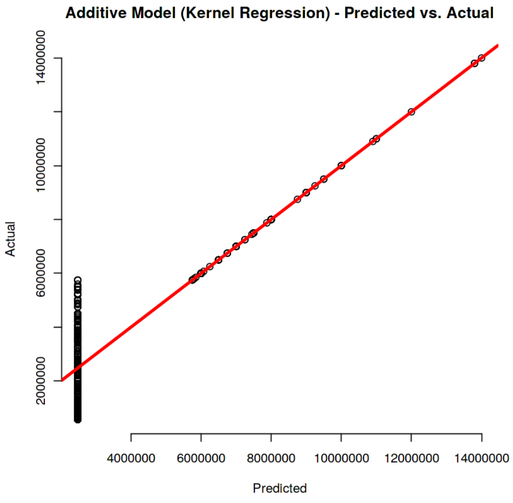
The XGBoost model resulted in an RMSE of 1512315. The bar chart below shows the most important features in the XGBoost model as measured by gain. Gain is the average training loss reduction gained when using a feature for splitting. The top five variables obtained from XGBoost to predict the salaries of the NHL players are similar to the variables from the decision tree model, with the addition of the third most important feature, CF. CF is the team's shot attempts (Corsi, SAT) while this player was on the ice.



5.5 Kernel Smoothing Regression

Due to kernel smoothing regression being computationally expensive, we wanted to limit the number of models being created and were selective of the lambda values tested in the model. The plot below shows the RMSE for a narrow range of lambda values after determining the region where the minimum occurs. The lambda value of 0.5 led to the optimal RMSE of 1510169. This was obtained using the back fitting algorithm for the additive model.





5.6 Best Model

In summary, the table below highlights the models tested in the project and their respective RMSE values.

Model	RMSE
LASSO Regression	1447595
Decision Tree Regression	1463413
Ridge Regression	1468157
Kernel Smoothing Regression	1510169
XGBoost Regression	1512315

After testing multiple statistical learning models, the best model as judged by RMSE is LASSO regression, with a lambda value of 28392 giving an RMSE of 1447595.

6 Conclusion

This project provided us with great real world experience for creating and evaluating predictive models. The project introduced us to many of the common problems that are faced when working with imperfect data.

One concept explored in the lectures that commonly appeared in our project is model complexity. With such a large number of inputs, there was plenty of opportunity to overfit the models. Therefore, regularization was crucial for this specific problem. We needed to reduce the contribution of less important coefficients to simplify the models. With no regularization the complex models may have performed well in training but a complex model would never perform well in testing. In particular, it is well known that local methods are affected by a large number of inputs. This is called the curse of dimensionality, which was discussed in lecture. To overcome this problem, we decided to use an additive model with kernel smoothing after reducing the model to the five best inputs determined by the XGBoost model.

6.1 Future Applications

Handling missing data is a common problem in machine learning that can be solved using many different techniques. It would be interesting to test multiple methods and analyze which leads to the best performance. A second way that we would improve the final model is by tuning multiple hyperparameters. In the project we explored tuning the most important hyperparameter in the respective models. However, for XGBoost and decision tree models specifically, there are multiple other very important hyperparameters that could also be tuned. Using a common library in R, a user can provide a sequence of values for each hyperparameter used in the model and R will output the optimal model with the corresponding RMSE and it's parameters. This method is computationally expensive, however it leads to a stronger predictive model. There are multiple methods that could have been explored beyond the models used in this project. It would have been interesting to explore other boosting techniques such as AdaBoost. If we had wanted to increase the number of inputs, we could have created interaction terms between existing inputs. However, increasing the number of inputs beyond the already large number currently in the model would also increase the amount of time required to train the models. If computing power and time was not important it would be interesting to see how the interaction terms would impact the model. Another method we could have used to increase the number of inputs is categorical encoding. It would have been valuable to encode string and categorical data to allow all inputs to be used in the model. Overall there are multiple future applications that could be explored to improve predictive power.

References

Chen, Minghua et al. "XGBoost-based algorithm interpretation and application on post-fault transient stability status prediction of power system". In: *IEEE Access* 7 (2019), pp. 13149–13158.

Hockey Abstract. URL: `http://www.hockeyabstract.com/`.

Yüzbaşı, Bahadır, Mohammad Arashi, and S Ejaz Ahmed. "Shrinkage Estimation Strategies in Generalised Ridge Regression Models: Low/High-Dimension Regime". In: *International Statistical Review* ().