Application of performance statistics to predict future performance in the National Hockey League

I. Introduction

When analyzing data from sports it is crucial to have an understanding of the sport and what the statistics mean. Due to this necessary understanding of the sport, this introduction is best suited to explain the basics of hockey and everything needed in fully grasping the sport. Ice hockey is a sport played on ice widely regarded as one of the fastest and most physical games in the world, with players reaching speeds up to 30 mph and puck speeds reaching close to 100 mph (Fischler et. al, 2021). These stats make it incredibly hard to predict future statistics in ice hockey due to the fast paced nature of the game itself. While there have been a few hockey metrics created in recent years to try and quantify the impact a player has on the ice, it is severely difficult to put these into action due to the pace hockey plays at. For example, baseball is a sport which has seen the greatest involvement from sports analytics but this is due to baseball having stop times between each play. This stop time allows the stat keeper to reset and signify a new play, whereas in hockey play keeps moving and even on a turnover everything happens so quickly it is hard to truly evaluate a performance.

This difficulty in evaluating hockey performance led us to focus our analysis on the simpler hockey statistics (ie. goals, assists, etc.) avoiding complicated metrics which have not been proven to accurately evaluate a players performance. Analyzing the past data was helpful in providing a clear result as we took the prior statistics of players from 2008-2010 from the website Hockey Reference (Wallace, 2021) and used this to see if prior performance helped to predict future goals scored in 2011. Taking such a large amount of data and using it to cast possibilities allowed us to see the greater picture. With this introductory knowledge about hockey and the process in which we conducted our research the reader should be better able to understand the following portions of the report.

This paper is trying to analyze if using multivariate linear regression models to predict the goals for a player in the NHL is a good method when compared to deep learning using neural networks. The motivation of the project was to further analyze a project completed by a group member in a different course where a similar analysis was done by purely using neural networks.

II. Methods

The Data has 2000 observations of players in the National Hockey League (NHL) through 2008-2010 predicting the response variable of goals next season (G_NS).

Table 1: **Variables Table**

S.NO	Variable Name	Description	Type	Measurement units
1.	AGE	Age of player	Discrete	How old player is
2.	GP	Games Played	Discrete	Number of games
3.	G	Number of goals	Discrete	Number of goals
4.	A	Number of assists	Discrete	Number of assists
5.	PTS	Goals plus assists	Discrete	Number of goals plus assists
6.	PlusMinus	Goal differential on ice	Continuous	Goals differential scored when on ice
7.	PIM	Penalty minutes	Continuous	Number of minutes spent in penalty box
8.	PPG	Power play goals	Discrete	Number of goals scored against a shorthanded defense
9.	PPA	Power play assists	Discrete	Number of assists scored against a shorthanded defense
10.	PPP	Power play points	Discrete	Number of points scored against a shorthanded defense
11.	SHG	Shorthanded goals	Discrete	Number of goals scored with a shorthanded team
12.	SHA	Shorthanded assists	Discrete	Number of assists scored with a shorthanded team
13.	SHP	Shorthanded points	Discrete	Number of points scored with a shorthanded team

14.	S	Shots taken	Discrete	Number of shots taken by player
15.	HIT	Hits on other players	Discrete	Number of hits a player has on others
16.	BLK	Blocks	Discrete	Number of blocks a player has on shots

As a preliminary analysis, we created scatterplots for all of the variables together and noticed a closely linear relationship with most of the variables (See figure 1). The total number of points, shots, and points per game being the most strongly correlated together as these are all interrelated. Examining these scatterplots we also saw that variables like age and shorthanded statistics were extremely variable to the other parts as they had little or no relation to other variables. This preliminary analysis allowed us to gauge how useful each variable would be in our analysis and informed us to exclude variables like player name and year of statistics from our model. However, the graph of goals scored showed up as a chi square distribution leading us to be wary of our distribution moving forward in the project (See figure 2). Furthermore, we calculated the mean, median and variance to get more preliminary information about the data. The results are summarized in the Appendix B (See table 6). From the table reported, we can clearly see that the data for almost all statistics have a very high variance.

Our model building process was lengthy but started using the full model with every variable as it was the most logical starting point and our group decided to look for the adjusted R-squared value for each model and got .585 for the full model. Not being satisfied we then built model 2 where we dropped the variables points, points per game, power play points, short handed goals, short handed assists, and short handed points. This was because we deemed these variables insignificant from the full model summary where they lacked significance as opposed to others. But the adjusted R-squared value for this model was lower at 0.5848 telling us that this model was not the correct choice.

Next we tried a new model selection method by stepwise selection to check the scope of the variables. The function works backwards, meaning it removes variables in each iteration to check which model is the most statistically significant. The main difference between stepwise and multiple regression is that multiple regression considers all the variables at the same time and stepwise regression keeps removing variables to find the most statistically significant model. This led us to the model of removing points, power play goals, power play points, and all shorthanded statistics. With this final model we saw our highest adjusted R-squared value yet at 0.5852. Finally we decided to use the best subsets model which puts each model against each other and sees which are the most accurate using Adjusted R-Squared, CP, & BIC. On top of this we use cross validations finding that model 11 gave us the smallest error. This model ended up being the same model as the previous where we saw an adjusted R-squared value of 0.5852. We decided to move forward with the models that we got using best-subsets and stepwise selection

due to a higher adjusted R-Squared value as well as getting the same model using 2 different methods to continue with the analysis.

Before checking for model assumptions, the data was transformed to get a better adjusted R-Squared value. To do this, using trial and error on the final model, we found that a log transformation on the age of players (age) and a square root transformation on the goals for previous season (g) yielded a better adjusted R-Squared value of 0.5857.

For any given model, there are certain assumptions that need to be met:

1. Linearity: It is the relationship between the independent variable X and the mean of the dependent variable Y. Our data was initially not ideal for regression, however, upon normalizing the data we were able to make the data more linear. By viewing the residual vs fitted value plot, we can observe that the points follow linearity and do not show a curvature in plot points. Very few outliers are visible from the central ban, indicating the variance is constant.The assumption that linearity exists between the predictor and the mean of the response variable.The QQ plots were heavy-tailed for some initial models but the final model has a qq plot which has more appreciable normality. The points lie on the diagonal indicating that the data is normally distributed. As seen in the QQ plot in Appendix B figure 3, the data is relatively close to the line of theoretical probability hence we can say this assumption is met.
2. Homoscedasticity: The variance for residuals is the same for any values of our independent variable X. This can be seen in the residual plot in Appendix B figure 3, and we can say that the assumption is met.
3. Independence: Observations are independent from each other. As seen in the data, each stat is from a different player and hence they are independent.
4. Normality: Our dependent variable Y is normal for any fixed value of X.

In this research paper, we are trying to predict the goals for next season for an NHL player given their statistics for the most recent season. The research question that we accessed is to find out if predicting goals using multiple linear regression is better than using deep learning. We used the best model selected from the previous model selection methods to predict the goals using the predict function in R after fitting the model. We also used deep learning in R with the help of neural networks. Using the same response variables as the best models, we scaled the data and prepared it for training. After that, we trained the Feed Forward ANNs using the scaled data and calculated the predicted values.

We compared the 2 models by evaluating a few statistics such as the mean squared error (MSE) and root mean squared error (RMSE). We also used some non-statistical factors such as time taken to commute the fitting of the multiple linear regression models and the time taken to train the neural network. The results that we found were that the multiple linear regression model

found using the model selection methods performed better than the neural networks. The results are discussed in detail in the Results and Discussion sections.

III. Results

Table 2: Best subsets model selection table

Variables Included	CV Error
g_ns ~ g	6.3487
g_ns ~ g + blk	6.2714
g_ns ~ g + blk + s	6.2076
g_ns ~ g + blk + s + age	6.1673
g_ns ~ g + blk + s + age + a	6.1432
g_ns ~ g + blk + s + age + a + gp	6.1196
g_ns ~ g + blk + s + age + a + gp + ppa	6.1068
g_ns ~ g + blk + s + age + a + gp + ppa + PlusMinus	6.1024
g_ns ~ age + gp + g + a + PlusMinus + pim + ppa + s + hit + blk + year	6.0964
g_ns ~ g + blk + s + age + a + gp + ppa + PlusMinus + hit	6.1004

The ranking of this table is based upon the CV error value, and the model with the least CV error value is the best model.

Table 3: **ANOVA Table**

```
Analysis of Variance Table

Response: g_ns
            Df Sum Sq Mean Sq   F value    Pr(>F)
age          1     57      57    1.5279 0.2165699
gp           1  26535   26535  715.9206 < 2.2e-16 ***
g            1  70824   70824 1910.8475 < 2.2e-16 ***
a            1   2895    2895   78.1039 < 2.2e-16 ***
PlusMinus    1     50      50    1.3559 0.2443935
pim          1      3       3    0.0867 0.7684820
ppa          1    491     491   13.2516 0.0002793 ***
s            1   3144    3144   84.8264 < 2.2e-16 ***
hit          1      0       0    0.0001 0.9916151
blk          1   1027    1027   27.6980 1.571e-07 ***
year         1    145     145    3.9022 0.0483605 *
Residuals 1988  73684      37
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

By looking into the ANOVA table we can see that the variables gp, g, a, ppa, s, blk and year are significant and are considered in our final model. Even though the P-values of age and PlusMinus are not indicating their significance, we took them into our final model because both the variables were shown in the result of stepwise and the best subsets model selection. Thus our final model has nine variables, including age and PlusMinus apart from the seven variables that we obtained from the ANOVA table.

Table 4: Parameter Table

Variable	Estimate	Standard Error	T-Value	P-Value	Confidence Interval
age	-4.76034	8.653325e-01	-5.5011749	4.259954e-08	(-6.4574, -3.0633)
gp	-0.10130	1.218772e-02	-8.3117607	1.724342e-16	(-0.1252, -7.7399)
g	2.63928	2.073409e-01	12.7292226	9.702098e-36	(2.2326, 3.0459)
a	0.19461	3.231205e-02	6.0229333	2.035190e-09	(0.1324, 2.5798)
PlusMinus	-0.01353	1.620832e-02	-0.8348513	4.039018e-01	(-0.0453, 1.8255)
pim	-0.00433	5.101394e-03	-0.8490686	3.959454e-01	(-0.0143, 5.6732)
ppa	-0.19959	5.260522e-02	-3.7940598	1.526351e-04	(-0.30276, -9.6420)
s	0.04500	4.995506e-03	9.0087837	4.760671e-19	(0.0353, 5.4800)
hit	0.00558	3.890416e-03	1.4337251	1.518080e-01	(-0.0021, 1.3208)
blk	-0.02514	4.901137e-03	-5.1297308	3.183706e-07	(-0.0348, -1.5530)
year	-0.34304	1.736564e-01	-1.9754032	4.836053e-02	(-0.6836, 2.4738)

The results from our comparisons to predict goals for NHL skaters are summarized below -

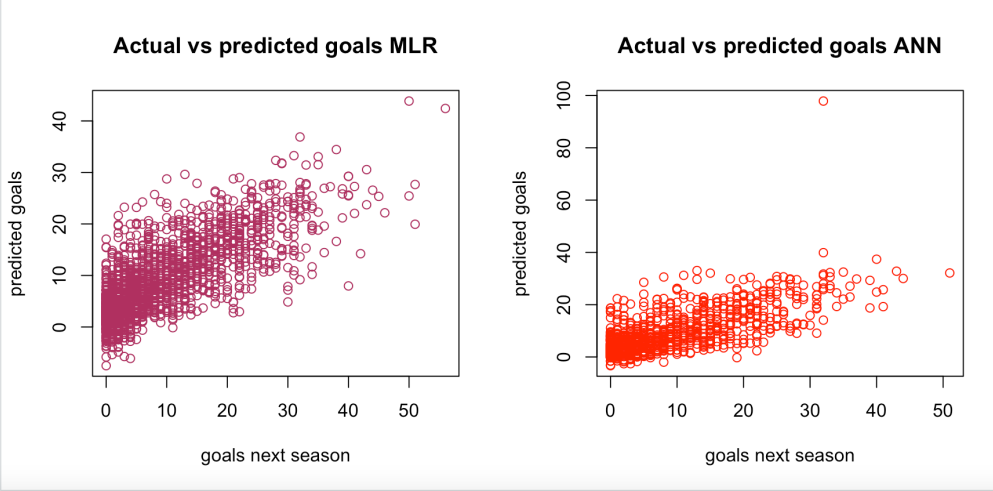
Table 5: Results for model prediction performance

Model	MSE	RMSE	Time to compute (sec)
Multivariate Linear Regression model	36.8417	6.069742	<0.1 sec
Neural Networks	51.502	7.1764	20-30 sec

As we can see, the Multivariate Linear Regression model performed better than the neural networks as the MSE and RMSE are both much lower than the neural network. Additionally, the time taken to compute the Multivariate Linear Regression model took less than 0.1 seconds on average. Compared to this, the neural networks performed worse as the MSE and RMSE were much higher and it also took longer on average to train the model.

Below is the picture of the actual vs predicted for skaters using the test data for both models (see figure 1). As we can see from the pictures, the ANN seems to be 'underfitting' the values compared to the multiple linear regression model.

Figure 1: Actual vs predicted goals for Multivariate Linear Regression and ANN models



IV. Discussion

The motivation behind this project was to compare the results from the predictions of Multivariate Linear Regression and deep learning in R while applying the knowledge that was learned in this class. As summarized briefly in the results, the Multivariate Linear Regression model selected from the model selections performed much better than a neural network (see table 5). There are, however, some limitations to the analysis and the results. Neural networks are a great tool for prediction for regression type problems. However, they require a large amount of data before they are accurate. Although the sample size is quite large (2000 data points), neural networks often require millions of data points to give extremely accurate information. Furthermore, access to powerful computing resources could have helped the group to analyze more hyperparameters for the neural networks to reduce the MSE.

Given the limited amount of time constraint and limited access to resources for implementing more complex machine learning models, a multivariate linear regression model is much better to implement as it is easier to build and fit the model when compared to deep learning. Some future exploration for the group would be to use more complex deep learning algorithms like Multivariate Recurrent Neural Network (MRNN) and compare the results to the existing multivariate linear regression model.

V. References

Please note that data was retrieved from the website using a scraping function in R - https://stathead.com/hockey/ppbp_finder.cgi?__hstc=88549636.24265637e9a47d8c4fddfb452b8dfa3d.1639154496281.1639154496281.1639154496281.1&__hssc=88549636.1.1639154496281&__hsfp=304880126

Player Advanced Stat Finder. (n.d.). Stathead.Com. https://stathead.com/hockey/ppbp_finder.cgi?__hstc=88549636.24265637e9a47d8c4fddfb452b8dfa3d.1639154496281.1639154496281.1639154496281.1&__hssc=88549636.1.1639154496281&__hsfp=304880126

R file 'scraper' (2021, March 7) In Ventresca, M (Ed.), in IE 33200 Computing in Industrial Engineering. Purdue University

Shah, M,. et al (2021) IE 332 Group 4 report. Purdue University.

Multiple Linear Regression in R. (2018, March 10). Articles - STHDA. http://www.sthda.com/english/articles/40-regression-analysis/168-multiple-linear-regression-in-r/

Best Subsets Regression Essentials in R. (2018, March 11). Articles - STHDA. http://www.sthda.com/english/articles/37-model-selection-essentials-in-r/155-best-subsets-regression-essentials-in-r/

Best Subsets Regression Essentials in R. (2018b, March 11). Articles - STHDA. http://www.sthda.com/english/articles/37-model-selection-essentials-in-r/155-best-subsets-regression-essentials-in-r/

Admin, J. (2020, May 24). *Basics Of Neural Network | Neural Network in R*. Analytics Vidhya. https://www.analyticsvidhya.com/blog/2017/09/creating-visualizing-neural-network-in-r/

Tutorials, A. P. R. (n.d.). *neuralnet: Train and Test Neural Networks Using R | DataScience+*. Datascienceplus.Com. https://datascienceplus.com/neuralnet-train-and-test-neural-networks-using-r/

Cakaloglu, T., & Xu, X. (2019). *MRNN: A Multi-Resolution Neural Network with Duplex Attention for Document Retrieval in the Context of Question Answering*. Arrive.Org. https://arxiv.org/pdf/1911.00964.pdf

Fischler, S. I. , Fischler, . Shirley W. and Eskenazi, . Gerald (2021, August 4). ice hockey. Encyclopedia Britannica. https://www.britannica.com/sports/ice-hockey

Wallace, H. (2021). *2020-21 NHL skater statistics*. Hockey. Retrieved December 9, 2021, from https://www.hockey-reference.com/leagues/NHL_2021_skaters.html.

VI. Appendix A: Code (Code found in R file "STAT 512 project.R")

```
#STAT 512 Project
#2.

# loading the data set in r
install.packages("readxl")
install.packages("neuralnet")
install.packages("tidyverse")
install.packages("leaps")
install.packages("corrplot")
install.packages("plyr")
install.packages("readr")
install.packages("dplyr")
install.packages("caret")
install.packages("ggplot2")
install.packages("mltools")
install.packages("repr")

library(repr)
library(readxl)
library(neuralnet)
library(tidyverse)
library(leaps)
library(corrplot)
library(plyr)
library(readr)
library(dplyr)
library(caret)
library(ggplot2)
library(mltools)
library(repr)

#############################################################################################
# Function to scrape season skater statistics from Hockey-reference.com

# The scraping function was developed by a group member in a different class
# and the appropriate citations can be found in the references of the report. To
# test the scraping function, please un comment the function, run the the function and
```

```
# call it using the season to extract from the website. e.g. to get data for season
# 2008, use scrapeSkaters(2008). The group pre-scraped the data and saved it as an excel
#file for grading convenience
#################################################################################################
# scrapeSkaters <- function(S) {
#   # The function takes parameter S which is a string and represents the season (YYYY)
#   # Returns: data frame
#
#   require(XML)
#   require(httr)
#
#   # Define certificate file, needed since website is HTTPS
#   cafile <- system.file("CurlSSL", "cacert.pem", package = "RCurl")
#
#   cafile <- "/etc/ssl/cert.pem"
#
#
#   # Read secure page
#   ## create the URL to scrape data from
#   URL <- paste("https://www.hockey-reference.com/leagues/NHL_",S, "_skaters.html", sep="")
#   page <- GET(URL,config(cainfo=cafile))
#
#   # Use regex to extract the desired table from the page
#   x <- text_content(page) #will give a deprecation warning, but that is OK
#   tab <- sub('.*(<table class="sortable stats_table".*?>.*</table>).*', '\\1', x)
#
#   ## grab the data from the page
#   tables <- readHTMLTable(tab)
#   ds.skaters <- tables$stats
#
#   ds.skaters <- ds.skaters[which(ds.skaters$Rk!="Rk"),]
#
#   ## Convert to lower case character data (otherwise will be treated as factors)
#   for(i in 1:ncol(ds.skaters)) {
#     ds.skaters[,i] <- as.character(ds.skaters[,i])
#     names(ds.skaters) <- tolower(colnames(ds.skaters))
#   }
#
#   ## finally fix the columns - NAs forced by coercion warnings
#   for(i in c(1, 3, 6:19)) {
#     ds.skaters[,i] <- as.numeric(ds.skaters[, i])
#   }
#
#   cn <- colnames(ds.skaters)
#   ds.skaters <- cbind(ds.skaters,ppp=rowSums(ds.skaters[,which(cn=="pp")]))
#   ds.skaters <- cbind(ds.skaters,shp=rowSums(ds.skaters[,which(cn=="sh")]))
#   cn <- colnames(ds.skaters)
#
#   ## fix a couple of the column names
#   #colnames(ds.skaters)
#   names(ds.skaters)[11] <- "pim"
#   names(ds.skaters)[18] <- "ppa"
#   names(ds.skaters)[14] <- "ppg"
#   names(ds.skaters)[15] <- "shg"
#   names(ds.skaters)[19] <- "sha"
#   names(ds.skaters)[10] <- "PlusMinus"
#
#
#   ## remove the header and totals row
#   ds.skaters <- ds.skaters[!is.na(ds.skaters$rk), ]
#
```

```
#   ## add the year too
#   ds.skaters$season <- S
#
#   ## remove any ' from players names (will case parsing issues later otherwise)
#   ds.skaters$player <- gsub("'","",ds.skaters[,"player"])
#
#   ## return the dataframe of subset of all categories
#   return(ds.skaters[,c(2:11,14,18,29,15,19,30,20,25,24)])
#   #ds.skaters
# }


data <- read_excel("~/Downloads/data_stat-1.xlsx")
str(data)

df <- data
df <- df[,-1]
#plot(df)

#preliminary analysis
mean_df <- colMeans(df[,2:ncol(df)])
median_df <- apply(df[2:ncol(df)], 2, median)
var_df <- apply(df[2:ncol(df)], 2, var)
std_df <- apply(df[2:ncol(df)], 2, sd)

ggplot(data=df, aes(g_ns)) +
  geom_histogram(aes(y =..density..), fill = "orange") +
  geom_density()

#Model selection
#full model
m1 <- lm(g_ns ~., data = df)
summary(m1)
summary(m1)$coefficient

#model 2 dropped variables -
m2 <- lm(g_ns ~ age + gp + g + a + PlusMinus + pim + ppa + s + hit + blk, data = df)
summary(m2)
summary(m2)$coefficient

#model 3 stepwise selection -
m3 <- lm(g_ns ~ . , data = df)
selectedMod <- step(m3)
summary(selectedMod)

#model 4 Best subsets
m4 <- regsubsets(g_ns ~.  , data = df, nvmax = 17)
summary(m4)

m4 <- regsubsets(g_ns ~ age + gp + g + a + PlusMinus + pim + ppa + ppg + s + sha + shg +  hit + blk + year, data = df, nvmax = 14)
summary(m4)

res.sum <- summary(m4)
data.frame(
  Adj.R2 = which.max(res.sum$adjr2),
  CP = which.min(res.sum$cp),
  BIC = which.min(res.sum$bic)
)

#k folds cross validations -
get_model_formula <- function(id, object, outcome){
```

```
  # get models data
  models <- summary(object)$which[id,-1]
  # Get outcome variable
  #form <- as.formula(object$call[[2]])
  #outcome <- all.vars(form)[1]
  # Get model predictors
  predictors <- names(which(models == TRUE))
  predictors <- paste(predictors, collapse = "+")
  # Build model formula
  as.formula(paste0(outcome, "~", predictors))
}

get_cv_error <- function(model.formula, data){
  set.seed(1)
  train.control <- trainControl(method = "cv", number = 5)
  cv <- caret::train(model.formula, data = data, method = "lm",
           trControl = train.control)
  cv$results$RMSE
}

model.ids <- 1:14
cv.errors <-  map(model.ids, get_model_formula, m4, "g_ns") %>%
  map(get_cv_error, data = df) %>%
  unlist()
cv.errors

which.min(cv.errors)

m4_selected <- lm(g_ns ~ age + gp + g + a + PlusMinus + pim + ppa + s + hit + blk +
                  year, data = df)

summary(m4_selected)
plot(m4_selected)

#transformations and linearity assumptions plots-

df2 <- df[,- c(5, 8, 10:13)]
plot(df2)

dft  <-  data.frame(log(df2$age),  df2$gp,  sqrt(df2$g),  df2$a,  df2$PlusMinus,  df2$pim,  df2$ppa,  df2$s,  df2$hit,  df2$blk,  df2$year,
df2$g_ns)
plot(dft)
colnames(dft) <- c("age", "gp", "g", "a", "PlusMinus", "pim", "ppa", "s", "hit", "blk", "year", "g_ns")

m5 <- lm(g_ns ~ age + gp + g + a + PlusMinus + pim + ppa + s + hit + blk +
           year, data = dft)
summary(m5)
summary(m5)$coefficient
confint(m5)

par(mfrow = c(2, 2))
plot(m5)

# anova table for selected model
anova(m5)

# predictions
#machine learning
#scaling the data
set.seed(100)
```

```
df_ <- dft[,-c(11)]
scaled <- as.data.frame(df_[,1:ncol(df_)])
maxs <- as.data.frame(apply(df_, 2, function(x){ max(x)})) %>% t()
mins <- as.data.frame(apply(df_, 2, function(x){ min(x)})) %>% t()
scaled <- as.data.frame(scale(scaled, center = mins, scale = maxs - mins))

# Train-test random splitting
index <- sample(1:nrow(df_),round(0.5*nrow(df_)))
train_ <- scaled[index,]
test_ <- scaled[-index,]

st_g <- g_ns ~ age + gp + g + a + PlusMinus + pim + ppa + s + hit + blk

#training the neural networks.
#Please note that the training takes around 20-25 seconds

nn <- neuralnet(st_g, data=train_ , hidden = c(8),stepmax = 1e+40,learningrate = 10,act.fct = "tanh", linear.output=T)
pr <- neuralnet::compute(nn,test_)
pr_nn <- (pr$net.result*(max(data$g_ns)-min(data$g_ns)))+min(data$g_ns)
test.nn <- ((test_$g_ns)*(max(data$g_ns)-min(data$g_ns)))+min(data$g_ns)
MSE_nn <- sum((test.nn - pr_nn)^2)/nrow(test_)
print(MSE_nn)

# Step 2 - predicting and evaluating the model on train data
predictions = predict(m5, newdata = dft)
mse(predictions, dft$g_ns)

#plots
par(mfrow=c(1,2))
plot(dft$g_ns, predictions, xlab = "goals next season", ylab = "predicted goals", col = "maroon", main = "Actual vs predicted goals
MLR")
plot(test.nn, pr_nn, xlab = "goals next season", ylab = "predicted goals", col = "red", main = "Actual vs predicted goals ANN")


confint(m5)

par(mfrow = c(2, 2))
plot(m5)

# anova table for selected model
anova(m5)

# predictions

#machiene learning
#scaling the data
set.seed(100)

df_ <- dft[,-c(11)]
scaled <- as.data.frame(df_[,1:ncol(df_)])
maxs <- as.data.frame(apply(df_, 2, function(x){ max(x)})) %>% t()
mins <- as.data.frame(apply(df_, 2, function(x){ min(x)})) %>% t()
scaled <- as.data.frame(scale(scaled, center = mins, scale = maxs - mins))

# Train-test random splitting
index <- sample(1:nrow(df_),round(0.5*nrow(df_)))
train_ <- scaled[index,]
test_ <- scaled[-index,]

st_g <- g_ns ~ age + gp + g + a + PlusMinus + pim + ppa + s + hit + blk
```

```
#training the neural networks.
#Please note that the training takes around 20-25 seconds

nn <- neuralnet(st_g, data=train_ , hidden = c(8),stepmax = 1e+40,learningrate = 10,act.fct = "tanh", linear.output=T)
pr <- neuralnet::compute(nn,test_)
pr_nn <- (pr$net.result*(max(data$g_ns)-min(data$g_ns)))+min(data$g_ns)
test.nn <- ((test_$g_ns)*(max(data$g_ns)-min(data$g_ns)))+min(data$g_ns)
MSE_nn <- sum((test.nn - pr_nn)^2)/nrow(test_)
print(MSE_nn)


# Step 2 - predicting and evaluating the model on train data
predictions = predict(m5, newdata = dft)
mse(predictions, dft$g_ns)


#plots
par(mfrow=c(1,2))
plot(dft$g_ns, predictions, xlab = "goals next season", ylab = "predicted goals", col = "maroon", main = "Actual vs predicted goals MLR")
plot(test.nn, pr_nn, xlab = "goals next season", ylab = "predicted goals", col = "red", main = "Actual vs predicted goals ANN")
```

VII. Appendix B: Output

Figure 2: Initial Scatterplot

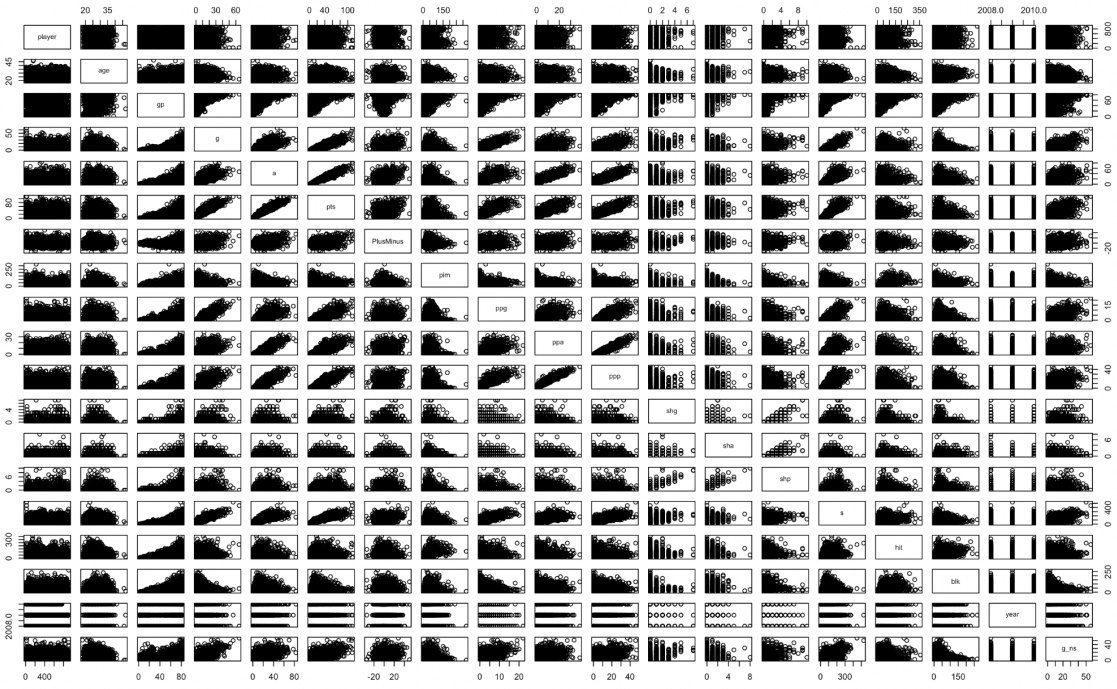


Figure 3: Preliminary analysis of goals scored in the next season used to build the model

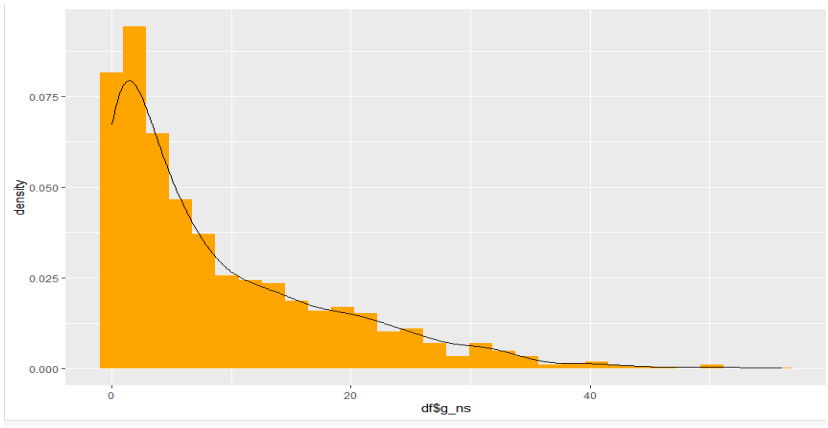


Figure 4: Diagnostic Plots

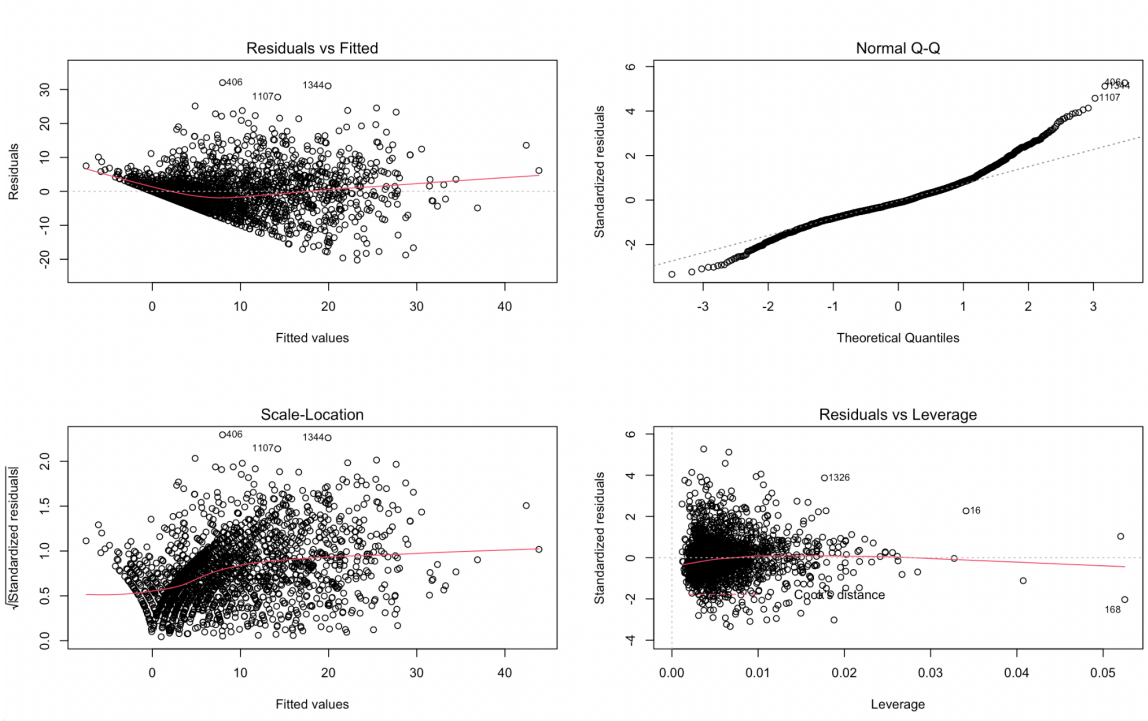


Figure 5: Plot of trained Neural Network for goals prediction

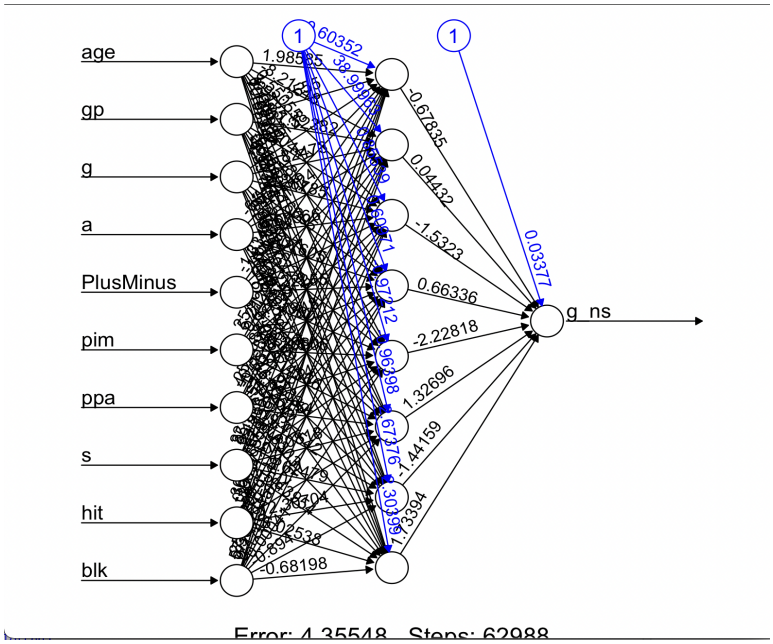


Table 6: Preliminary analysis for all statistics -

Statistic/ measure	gp	g	a	pts	PlusMinus	pim	ppg	ppa	ppp
mean	53.1665	8.6925	14.7920	23.4845	0.1960	39.4280	2.3645	4.4385	6.8030
median	63	5	11	17	-1	32	0	1	2
variance	760.2519	92.0240	195.967 7	494.8502	87.5834	1309.0373	12.8721	40.5025	85.7971

Statistic/ measure	shg	sha	shp	s	hit	blk	year	g_ns
mean	0.2850	0.3150	0.6000	91.5295	61.6815	38.7640	2008.9430	8.7830
median	0	0	0	78	50	24	2009	5
variance	0.5521	0.5000	1.3667	5603.102 7	2791.9360	1691.1629	0.6351	89.4716