msheridan econ 1042 goalie project

Matt Sheridan

2023-05-06

```
library("xtable")
```

```
library("broom")
```

```
library("stargazer")
```

```
library("MASS")
```

```
library(dplyr)
```

```
## Warning: package ’dplyr’ was built under R version 4.2.3
```

```
##
## Attaching package: ’dplyr’
```

```
## The following objects are masked from ’package:stats’:
##
##     filter, lag
```

```
## The following objects are masked from ’package:base’:
##
##     intersect, setdiff, setequal, union
```

```
team_stand = data.frame(readxl::read_excel("TeamStandingsFinal.xlsx"))
```

```
## New names:
## * ‘GF/GP‘ -> ‘GF/GP...13‘
## * ‘GF/GP‘ -> ‘GF/GP...14‘
```

```
team_stand$WinsPerGame = team_stand$W / team_stand$GP
years = unique(team_stand$Year)
coefs_1 = rep(NULL,length(years))

for (i in 3:length(years)){
  current = years[i]
  dta = data.frame(subset(team_stand, Year %in% c(years[i-2],years[i-1], years[i])))
  dta$weight = 0
  dta[dta$Year == years[i-2], ]$weight = 1
  dta[dta$Year == years[i-1], ]$weight = 2
  dta[dta$Year == years[i], ]$weight = 3
  coefs_1 = c(coefs_1, summary(lm(GD.GP ~ WinsPerGame, data = dta, weights = weight))$coefficients[2,1]
}

GPW = data.frame(readxl::read_excel("GoalsPerWinStat.xlsx"))[,c(1,3)]

GPW = rbind(GPW, data.frame(Year = years[6:7], Goals.Per.Win = coefs_1[4:5]))[2:16,]

GPW$Year = as.numeric(GPW$Year)
GPW
```

```
##    Year Goals.Per.Win
## 2  2008      5.525000
## 3  2009      5.525000
## 4  2010      5.600000
## 5  2011      5.733000
## 6  2012      5.389000
## 7  2013      5.279000
## 8  2014      5.252000
## 9  2015      5.182000
## 10 2016      5.312000
## 11 2017      5.132000
## 12 2018      5.364000
## 13 2019      5.620000
## 14 2020      5.571000
## 15 2021      5.250578
## 16 2022      5.543946
```

```
#Data loading and cleaning
goalie_lagged = data.frame(readxl::read_excel("goaliedata2.xlsx"))

goalie_lagged = goalie_lagged[goalie_lagged$ongoal > 0, ]
#GSAX variables
goalie_lagged$GSAX = goalie_lagged$XGA - goalie_lagged$GA
goalie_lagged$lagged_GSAX = goalie_lagged$lagged_xga - goalie_lagged$lagged_ga
#flurry adjusted
goalie_lagged$flurry_GSAX = goalie_lagged$flurryAdjustedxGoals - goalie_lagged$GA
goalie_lagged$lagged_flurry_GSAX = goalie_lagged$lagged_flurryadjxg - goalie_lagged$lagged_ga
goalie_lagged$flurryGSAXper60 = (60 * goalie_lagged$flurry_GSAX) / (goalie_lagged$TOI/60)
goalie_lagged$lagged_flurryGSAXper60 = (60 * goalie_lagged$lagged_flurry_GSAX) / (goalie_lagged$lagged_

#GSAX Per Game
goalie_lagged$GSAXper = (goalie_lagged$XGA - goalie_lagged$GA) / goalie_lagged$GP

goalie_lagged$GSAXper_lagged = (goalie_lagged$lagged_xga - goalie_lagged$lagged_ga) / goalie_lagged$lag
#GSAX per 60
goalie_lagged$GSAXper60 = (60 * goalie_lagged$GSAX) / (goalie_lagged$TOI/60)
goalie_lagged$lagged_GSAXper60 = (60 * goalie_lagged$lagged_GSAX) / (goalie_lagged$lagged_toi/60)
#GP Percentage
goalie_lagged$GPPCT = goalie_lagged$GP / 82
goalie_lagged$lagged_GPPCT = goalie_lagged$lagged_gp / 82
```

```
#lockout adjusting - this year is weird because the gppcts could be higher since there were less games.
goalie_lagged[goalie_lagged$Year==2012,]$GPPCT = goalie_lagged[goalie_lagged$Year==2012,]$GPPCT * 82/48
goalie_lagged[goalie_lagged$Year==2012,]$lagged_GPPCT = goalie_lagged[goalie_lagged$Year==2012,]$lagged_

#covid adjusting - this year is weird because the gppcts could be higher since there were less games.
goalie_lagged[goalie_lagged$Year==2012,]$GPPCT = goalie_lagged[goalie_lagged$Year==2012,]$GPPCT * 82/70
goalie_lagged[goalie_lagged$Year==2012,]$lagged_GPPCT = goalie_lagged[goalie_lagged$Year==2012,]$lagged_

#SVPCT
goalie_lagged$SVPCT = (goalie_lagged$ongoal - goalie_lagged$GA) / goalie_lagged$ongoal
goalie_lagged$lagged_SVPCT = (goalie_lagged$lagged_ongoal - goalie_lagged$lagged_ga) / goalie_lagged$lag
#GAA
goalie_lagged$GAA = (60*goalie_lagged$GA) / (goalie_lagged$TOI/60)
goalie_lagged$lagged_GAA = (60*goalie_lagged$lagged_ga) / (goalie_lagged$lagged_toi/60)
#low danger goals saved above expected
goalie_lagged$LDGSAX = goalie_lagged$lowDangerxGoals - goalie_lagged$lowDangerGoals
goalie_lagged$lagged_LDGSAX = goalie_lagged$lagged_ldxg - goalie_lagged$lagged_ldg
#medium danger goals saved above expected
goalie_lagged$MDGSAX = goalie_lagged$mediumDangerxGoals - goalie_lagged$mediumDangerGoals
goalie_lagged$lagged_MDGSAX = goalie_lagged$lagged_mdxg - goalie_lagged$lagged_mdg
#high danger goals saved above expected
goalie_lagged$HDGSAX = goalie_lagged$highDangerxGoals - goalie_lagged$highDangerGoals
goalie_lagged$lagged_HDGSAX = goalie_lagged$lagged_hdxg - goalie_lagged$lagged_hdg
#low danger goals saved above expected per 60
goalie_lagged$LDGSAXper = (60 * goalie_lagged$LDGSAX) / (goalie_lagged$TOI/60)
goalie_lagged$lagged_LDGSAXper = (60 * goalie_lagged$lagged_LDGSAX) / (goalie_lagged$lagged_toi/60)
#medium danger goals saved above expected per 60
goalie_lagged$MDGSAXper = (60 * goalie_lagged$MDGSAX) / (goalie_lagged$TOI/60)
goalie_lagged$lagged_MDGSAXper = (60 * goalie_lagged$lagged_MDGSAX) / (goalie_lagged$lagged_toi/60)
#high danger goals saved above expected per 60
goalie_lagged$HDGSAXper = (60 * goalie_lagged$HDGSAX) / (goalie_lagged$TOI/60)
goalie_lagged$lagged_HDGSAXper = (60 * goalie_lagged$lagged_HDGSAX) / (goalie_lagged$lagged_toi/60)
#Win PCT
goalie_lagged$WGP = goalie_lagged$W / goalie_lagged$GP
```

```
par(mfrow = c(2,2))
goalies_2022 = subset(goalie_lagged, (Year == 2022) & !is.na(W))
goalies_not_2022 = subset(goalie_lagged, (Year != 2022) & !is.na(W))
goalies_not_2022_nowin = subset(goalie_lagged, (Year != 2022))
boxplot(goalies_not_2022$GAA, goalies_not_2022_nowin$GAA,
names=c("Only top 50", "Not in top 50 GP"), main = "GAA Comparison", ylab = "GAA")
boxplot(goalies_not_2022$SVPCT, goalies_not_2022_nowin$SVPCT,
names=c("Only top 50", "Not in top 50 GP"), main = "SVPCT Comparison", ylab = "SVPCT")
print(xtable(t(summary(1:8))), type="html", file="xt.html", include.rownames=FALSE)
colnames = c("WGP", "GPPCT")
goalies_not_2022[,colnames(goalies_not_2022) %in% colnames]
```

```
boxplot(goalies_not_2022$WGP, main = "Wins Per Games Played")
boxplot(goalies_not_2022$GPPCT, main = "Total Games Played Percentage")
boxplot(goalies_not_2022$GAA, main = "Goals Against Average")
boxplot(goalies_not_2022$SVPCT, main = "Save Percentage")
boxplot(goalies_not_2022$Votes, main = "Total Votes")
boxplot(goalies_not_2022[goalies_not_2022$Votes>0,]$Votes, main = "Votes Among Vote Receivers")
```

```r
nb_model_1 = glm.nb(Votes ~ WGP + GPPCT + SVPCT + GAA, data = goalies_not_2022)
summary(nb_model_1)
stargazer(nb_model_1, type='latex')
mean((predict(nb_model_1) - goalies_not_2022$Votes)^2)

preds = predict(nb_model_1, newdata = goalies_2022, type = 'response')
#288 * (preds / sum(preds))
preds_df_22 = data.frame(Name = goalies_2022$Name, pred_votes = 288 * (preds / sum(preds)))
top_10 = head(preds_df_22[order(preds_df_22$pred_votes, decreasing = T),],10)
top_10$Name = factor(top_10$Name, levels = top_10$Name)
predict_year = function(year, mod){
data = subset(subset(goalie_lagged, (Year == year) & !is.na(W)))
predictions = predict(mod, data, type='response')
predictiondf = data.frame(Name = data$Name, pred_votes = 288 * (predictions / sum(predictions)))
top_10 = head(predictiondf[order(predictiondf$pred_votes, decreasing = T),],10)
top_10$Name = factor(top_10$Name, levels = top_10$Name)
ggplot(top_10, mapping = aes(x =forcats::fct_rev(Name),y=pred_votes)) +
geom_bar(stat='identity', fill=rainbow(10)) + coord_flip() +
labs(title=paste("Predicted Votes for", year, "Goalies")) + ylab("Predicted Votes") +
xlab("Goaltender")
}
ggplot(top_10, mapping = aes(x =forcats::fct_rev(Name),y=pred_votes)) +
geom_bar(stat='identity', fill=rainbow(10)) + coord_flip() +
labs(title="Predicted Votes for 2022 Goalies") + ylab("Predicted Votes") +
xlab("Goaltender")
library("ggpubr")
```

```r
predict_year(2013,nb_model_1)
predict_year(2018,nb_model_1)
predict_year(2022,nb_model_1)
```

```r
#Data Expoloration
boxplot(GPPCT~Votes>0 , data = goalie_lagged)
goalies_lagged_contenders = subset(goalie_lagged, (GPPCT>0.28) & !is.na(W))
goalies_2022 = subset(goalies_lagged_contenders, Year == 2022)
goalies_not_2022 = subset(goalies_lagged_contenders, Year != 2022)
train_goalies = subset(goalies_lagged_contenders, ((Year!=2022) & (Year %% 2 == 0)))
test_goalies = subset(goalies_lagged_contenders, ((Year!=2022) & (Year %% 2 != 0)))
pois_model = glm(Votes ~ WGP + GPPCT + SVPCT + GAA, data = train_goalies, family = poisson)
summary(pois_model)
```

```r
preds_df_22 = data.frame(Name = goalies_2022$Name, pred_votes = predict(pois_model, newdata =
                                                                          goalies_2022))
preds_df_22[order(preds_df_22$pred_votes, decreasing = T),]




pred_year = function(year, model){

  year = 2020
  pred_df = cbind(subset(goalies_not_2022, Year == year)[,c(2,3,4,5,42,44,46,60)],
                  data.frame(pred_votes = predict(model, newdata = subset(goalies_not_2022, Year == yea
  pred_df$actual_votes = subset(goalies_not_2022, Year == year)$Votes

  pred_df[order(pred_df$pred_votes, decreasing = T),]
}
pred_year(2019, pois_model)
```

```
#CREATING MY OWN WINS ABOVE REPLACEMENT VALUE FOR NHL GOALTENDERS
Complete_Data = subset(goalie_lagged, !is.na(W) & !is.na(Team_Wins))
Complete_Data = left_join(Complete_Data, GPW, by="Year")
Complete_Data$Goalie_WARs = (Complete_Data$GSAX / Complete_Data$Goals.Per.Win) * Complete_Data$W.TW
display_year = function(y){
dta = subset(Complete_Data, Year == y)
dta[order(dta$Goalie_WARs, decreasing=T), c(1,2,26,69)]
}
for (i in 1:length(unique(Complete_Data$Year))){
print(display_year(unique(Complete_Data$Year)[i]))
}
by(Complete_Data$Goalie_WARs, Complete_Data$Year, summary)
summaries = cbind(aggregate(Complete_Data$Goalie_WARs, by=list(Complete_Data$Year), min),
aggregate(Complete_Data$Goalie_WARs, by=list(Complete_Data$Year), max)$x,
aggregate(Complete_Data$Goalie_WARs, by=list(Complete_Data$Year), median)$x,
aggregate(Complete_Data$Goalie_WARs, by=list(Complete_Data$Year), mean)$x)
colnames(summaries) = c("Year", "Min", "Max", "Median", "Mean")
summaries$Year = as.character(summaries$Year)
stargazer(t(summaries), type = 'latex')
dta = subset(Complete_Data, Year == 2022)
dta = head(dta[order(dta$Goalie_WARs, decreasing=T), c(2,69)], 5)
stargazer(t(t(dta)))
```