Pitcher Data

Danny Kim

6/25/2021

This project uses machine learning to predict pitcher WAR (Wins Above Replacement)

```
# load data
pitcherData <- read.csv('pitching.csv', fileEncoding = "UTF-8-BOM")
```

Conduct a correlation analysis to find the variables with the greatest correlation with WAR

```
cor(subset(pitcherData, select=-c(Season,Name,Team,playerid)))
```

```
##               W           L         G        GS        IP          H
## W    1.00000000 -0.23962935 0.4906802 0.5112760 0.6170719 0.29419092
## L   -0.23962935  1.00000000 0.4615225 0.4723297 0.3537170 0.59316362
## G    0.49068016  0.46152253 1.0000000 0.9612645 0.8848625 0.79747591
## GS   0.51127602  0.47232973 0.9612645 1.0000000 0.9129434 0.81098734
## IP   0.61707186  0.35371698 0.8848625 0.9129434 1.0000000 0.78284665
## H    0.29419092  0.59316362 0.7974759 0.8109873 0.7828466 1.00000000
## R    0.01551775  0.71706009 0.6810726 0.6812255 0.5347605 0.82737506
## ER   0.02719858  0.69593679 0.6763726 0.6760027 0.5246820 0.81267763
## HR   0.08254809  0.38845756 0.4704309 0.4677397 0.3520328 0.43814038
## BB   0.16402669  0.35575813 0.5520149 0.5578072 0.4149881 0.32145700
## SO   0.53983054 -0.01306392 0.5035731 0.5418612 0.6342158 0.20560750
## WAR  0.51508605 -0.25591758 0.1976540 0.2415900 0.4573257 0.05633516
##               R          ER          HR         BB          SO         WAR
## W    0.01551775  0.02719858  0.08254809  0.1640267  0.53983054  0.51508605
## L    0.71706009  0.69593679  0.38845756  0.3557581 -0.01306392 -0.25591758
## G    0.68107256  0.67637265  0.47043086  0.5520149  0.50357314  0.19765401
## GS   0.68122545  0.67600270  0.46773971  0.5578072  0.54186124  0.24159001
## IP   0.53476046  0.52468199  0.35203277  0.4149881  0.63421578  0.45732572
## H    0.82737506  0.81267763  0.43814038  0.3214570  0.20560750  0.05633516
## R    1.00000000  0.98204159  0.60547427  0.4718852  0.07162203 -0.26083123
## ER   0.98204159  1.00000000  0.62956429  0.4654406  0.07384345 -0.26759643
## HR   0.60547427  0.62956429  1.00000000  0.1554401  0.19030585 -0.28845823
## BB   0.47188517  0.46544062  0.15544008  1.0000000  0.27001880 -0.12545270
## SO   0.07162203  0.07384345  0.19030585  0.2700188  1.00000000  0.74106437
## WAR -0.26083123 -0.26759643 -0.28845823 -0.1254527  0.74106437  1.00000000
```

From the WAR column, the best variables are wins, losses, innings pitched, earned runs, home runs allowed, strikeouts, and WAR (correlation coefficient greater than .25 or less than -.25)

```
variables <- c('W','L','IP','ER','HR','BB','SO','WAR')
library(caret)
```

```
## Loading required package: lattice

## Loading required package: ggplot2
```

```
# select 70 percent of data
inTrain <- createDataPartition(pitcherData$WAR,p=0.7,list=FALSE)
# create train and test sets
training <- pitcherData[inTrain, variables]
testing <- pitcherData[-inTrain, variables]
```

After creating the train and test sets, we want to model WAR adjusting for these variables using linear regression with cross-validation.

```
# cross validation
method = 'lm'
ctrl <- trainControl(method = 'repeatedcv',number = 10, repeats = 10)

# fit the model
modelFit <- train(WAR ~ ., method=method, data=training, trControl=ctrl)
summary(modelFit)
```

```
##
## Call:
## lm(formula = .outcome ~ ., data = dat)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.79548 -0.35754  0.02021  0.35426  1.55142
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  0.2404387  0.1425266   1.687   0.0922 .
## W            0.0067712  0.0103114   0.657   0.5117
## L           -0.0204286  0.0118724  -1.721   0.0859 .
## IP           0.0152176  0.0015092  10.083   <2e-16 ***
## ER           0.0021347  0.0025464   0.838   0.4022
## HR          -0.1138800  0.0049634 -22.944   <2e-16 ***
## BB          -0.0369569  0.0017400 -21.239   <2e-16 ***
## SO           0.0257063  0.0007639  33.653   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.5613 on 561 degrees of freedom
## Multiple R-squared:  0.8794, Adjusted R-squared:  0.8778
## F-statistic: 584.1 on 7 and 561 DF,  p-value: < 2.2e-16
```

The adjusted R-squared for this model is .8675, and there may be room for improvement. We saw that IP, HR, BB, and SO were the most significant variables contributing to WAR, so we will fit another model with these 4 variables to see if there is any improvement in our model.

```
model2 <- train(WAR ~ L + IP + HR + BB + SO, method=method, data=training, trControl=ctrl)
summary(model2)
```

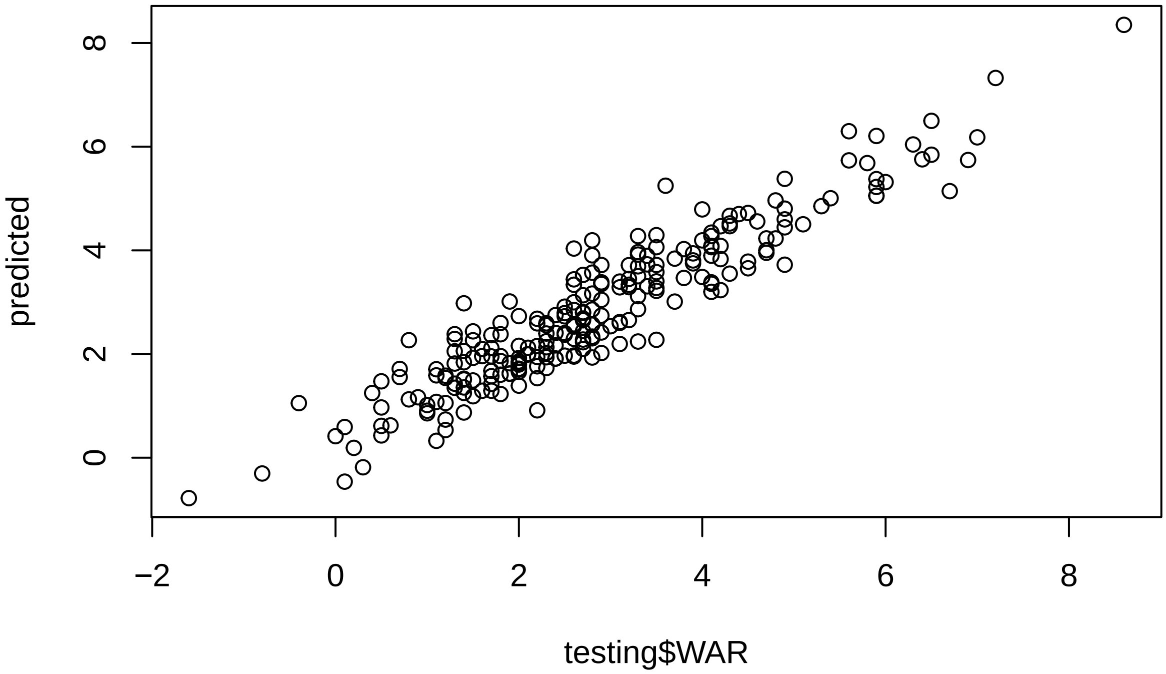
```
##
## Call:
## lm(formula = .outcome ~ ., data = dat)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.80669 -0.36025  0.02847  0.36350  1.56668
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  0.2531765  0.1418145   1.785   0.0748 .
## L           -0.0204061  0.0090992  -2.243   0.0253 *
## IP           0.0162136  0.0010664  15.204   <2e-16 ***
## HR          -0.1115359  0.0040818 -27.325   <2e-16 ***
## BB          -0.0364013  0.0016114 -22.589   <2e-16 ***
## SO           0.0255044  0.0007117  35.837   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.5608 on 563 degrees of freedom
## Multiple R-squared:  0.8791, Adjusted R-squared:  0.8781
## F-statistic:   819 on 5 and 563 DF,  p-value: < 2.2e-16
```

The R-squared value of both models is similar. Since the second model adjusts for less variables but fits the data just as well, we will continue with our second model. Lastly, we need to apply our model to the test see if it is good.

```
# Apply to test set
predicted <- predict(model2,newdata=testing)
# R-squared
cor(testing$WAR,predicted)^2
```

```
## [1] 0.874692
```

```
# Plot the predicted values vs. actuals
plot(testing$WAR,predicted)
```



Correlation coefficient is high, and seems to fit the test set very well.

Final model to predict pitcher WAR

WAR $= 0.2191834 - 0.0245542(L) + 0.0161916(IP) - 0.111120(HR) - 0.0344068(BB) + 0.0252364(SO)$