Analysis of Major League Baseball Starting Pitchers and the Cy Young Award

STAT 473 Final Report

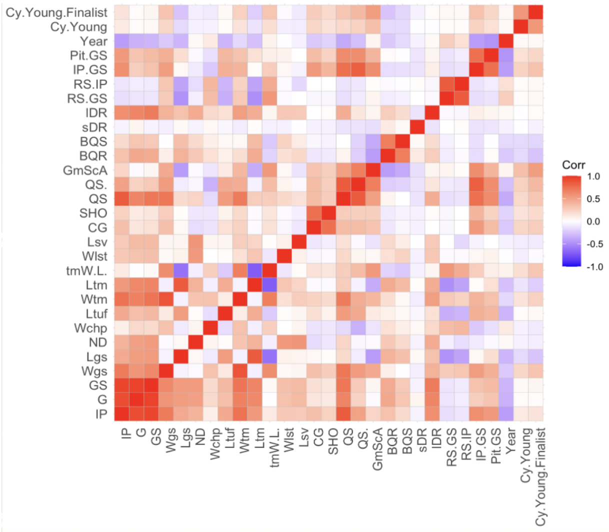
Mario A. Leon
Derek Garcia

California State University, Long Beach
May 12, 2023

Abstract

The Cy Young award is given to the two best pitchers in Major League Baseball at the conclusion of the Major League Baseball season. To see what makes a pitcher a Cy Young award favorite, uncommon predictors like quality starts, short and long rest between starts, and losses in games started were analyzed. The results showed that consistency and longevity influenced quality starts, losses in games started, bequeathed runners, and short-day rest. Pitchers that appeared in as many games as possible saw influenced the outcome of a game more often over a pitcher that did not appear in games.

Fig. 1. Correlation Matrix of all our variables in the data set.



Introduction

Major League Baseball began in the late 1800's with the sport slowly becoming more and more developed as time went on. This includes the rules, the skill level of the players, and even how to award certain players. The best team, the World Series champion, is not the only one who gets awarded. Players who had an incredible season deserve recognition like being recognized as an All Star or a Silver Slugger for best hitter. The award we will put our attention on is the Cy Young award, named after Hall of Fame pitcher Cy Young who pitched from 1890 to 1911. The award was introduced in 1956 to award the best pitcher in both the American League and the National League, so there will be 2 Cy Young award winners per season. It is worth noting that Major League Baseball consists of 30 teams divided into two leagues with each league consisting of 15 teams, the American League and the National League. Every season since 1956, two Cy Young awards are given, each one to the best pitcher in both the American League and National League.

Questions of Interest

One of our goals is to figure out which predictors have an association with the probability of winning the Cy Young Award. Then we also would like to see which of these predictors contribute to an increase in the probability of winning the Cy Young Award (i.e. positive coefficients). After obtaining this, we would like to know how accurately our models classify Cy Young Award winners and we will use Decision Trees, Random Forest, Linear Discriminant Analysis, and Quadratic Discriminant Analysis to determine this. We will use confusion matrices and ROC curves to analyze how ideal our predictors are. The reason we decided this combination of prediction, inference, and classification questions is because like mentioned earlier, our variables are not common or obvious. We are looking into features that aren't really talked about when discussing a Cy Young Award winner so we would like to see if we can identify any significant and high accuracy in our models. To check the accuracy of the LDA fit, a percentage was produced using our selected variables. Additionally, an ROC curve for the LDA model was created to check how ideal our selected predictors are. A QDA model was applied using the same predictors from the LDA model, repeating what we did for the LDA model.

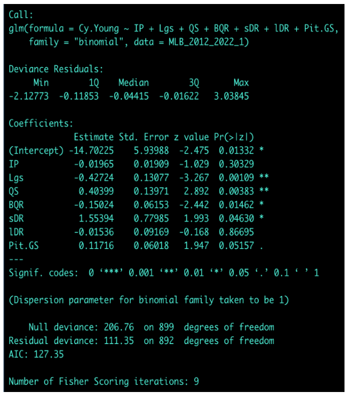
Analysis

Exploratory Data Analysis

A lot of factors determine what makes the "best pitcher". These can range from having a low Earned-Run Average, striking out many hitters, allowing little to no walks and runs, and so on. However, we consider these features to be too obvious. Instead, our data set shows the regular season statistics for each Major League Baseball starting pitchers from 2012-2022 who has pitched at least 150 innings (50 in 2020), such as short days rest, long days rest, quality starts, game score, team losses in games started, team wins, runners left on base when the pitcher left the game, and so on. Postseason statistics are not considered. We will exclude the year 2020 though since that was the shortened 60 game season. We would like to keep working with full seaosns (162 games).

It also important to add a little caveat about Game Score. When Game Score was included in our models, it ended up being either the only significant predictor or one of the only significant predictors and the reason is due to how it is calculated. According to MLB, Game Score is calculated as follows: a player starts with 40 points and has 2 points added per out recorded, 1 point per strikeout, 2 points will be lost for each walk and hit allowed, 3 points will be lost for each run allowed, and 6 points will be lost for each home run allowed. Basically, the higher the game score, the more likely a pitcher would be considered for the Cy Young award. This variable overpowers all the other variables. We could have easily just said that Game Score has a big effect on Cy Young winners and call it a day but that is not the point of our analysis. The point is to determine what uncommon features has a significant effect on predicting and classifying Cy Young award winners so due to

Fig. 2. Logistic Regression Output



```
Call:
glm(formula = Cy.Young ~ IP + Lgs + QS + BQR + sDR + lDR + Pit.GS,
    family = "binomial", data = MLB_2012_2022_1)

Deviance Residuals:
     Min        1Q    Median        3Q       Max
-2.12773  -0.11853  -0.04415  -0.01622   3.03845

Coefficients:
            Estimate Std. Error z value Pr(>|z|)
(Intercept) -14.70225    5.93988  -2.475  0.01332 *
IP           -0.01965    0.01909  -1.029  0.30329
Lgs          -0.42724    0.13077  -3.267  0.00109 **
QS            0.40399    0.13971   2.892  0.00383 **
BQR          -0.15024    0.06153  -2.442  0.01462 *
sDR           1.55394    0.77985   1.993  0.04630 *
lDR          -0.01536    0.09169  -0.168  0.86695
Pit.GS        0.11716    0.06018   1.947  0.05157 .
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 206.76  on 899  degrees of freedom
Residual deviance: 111.35  on 892  degrees of freedom
AIC: 127.35

Number of Fisher Scoring iterations: 9
```

this, we will remove Game Score. Another obvious variable was Wgs, Wins on Games Started, but that one is self explanatory. Of course a pitcher's chances look a lot better if the team wins when he starts. The inverse, Lgs (Losses on Games Started), isn't as obvious and will be included in the model.

After much consideration and analysis of each variable in our dataset, including avoiding multicollinearity among other variables using **Fig.1**, we decided to work with the following variables: IP (Innings Pitched), Lgs (Team Losses in Games Started (Pitched)), QS (Quality Starts), BQR (Bequeath Runners (number of runners on base when pitcher left game)), sDR (Short Days Rest), lDR (Long Day Rest), and Pit.GS (Pitches per Games Started).

Which of our selected predictors have an association with the probability of winning the Cy Young Award? Which of our selected predictors contribute to an increase in the log-odds of winning the Cy Young Award?

We ran a Logistic Regression model here using the selected predictors mentioned earlier. According to the output, we have the following model as shown in **Fig.2**:

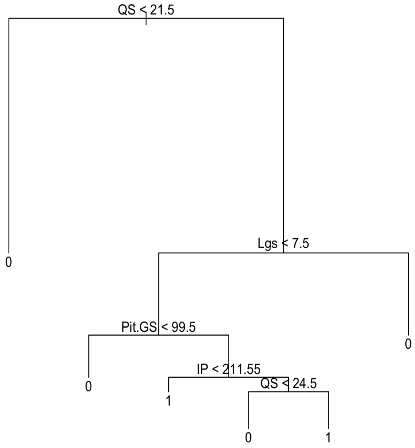
$log(\frac{\hat{p}(X)}{1-\hat{p}(X)}) = -14.70225 - 0.01965 * IP - 0.42724 * Lgs + 0.40399 * QS - 0.15024 * BQR + 1.55394 * sDR - 0.01536 * lDR + 0.11716 * Pit.GS$

Here, we see that team losses in games started, quality starts, runners left on base when the pitcher left the game, and short days rest all have an association with the probability of winning the Cy Young Award. This is very informative as it tells us the more the pitcher starts (i.e. short days rest), the more of a chance they'll have at winning the Cy Young Award. There's also been a debate that a team losing in that pitcher's start shouldn't matter as long as the pitcher pitches a great game but our model suggests that it may have a significant impact. Had we left the GmScA variable, that variable along with another variable would have been the only significant variables, which honestly do not tell us much. We already knew before the research that Game Score has an association with the probability of winning the Cy Young Award. In terms of the coefficients, the ones that are positive are quality starts, short days rest, and pitches per game. Each of these contribute to an increase in the log odds of winning the Cy Young award. For example, if we increase Short Days Rest of a pitcher by 1 day, holding all other predictors constant, this means the pitcher will rest for one less day and will translate to a 1.55 increase in the log-odds in winning the Cy Young Award. This could likely lead to injury for a pitcher but if the pitcher is durable and pitches a lot, then their chances look good. Hypothetical, this model means that if a pitcher had 5 team losses in games he started, 30 quality starts, 30 bequeathed runners, and 5 occurrences of short days rest gives the pitcher a 16.7% chance of winning the Cy Young Award.

How accurate do our models classify Cy Young award winners given our selected predictors? What is the probability of a pitcher winning the Cy Young award?

Here, we tested the classification using several models: Regular Decision Trees, Random Forests, Linear Discriminant Analysis, and Quadratic Discriminant Analysis. First, we'll go over Decision Trees:

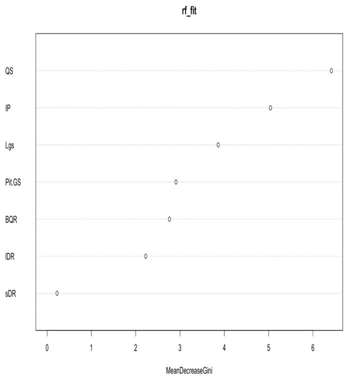
Fig. 3. Final Decision Tree Used



Our original tree had 7 terminal nodes featuring quality starts, losses ingames started, innings pitched, and pitches per game. When we tried to prune the tree, it simply went down to 6 terminal nodes but this gives us more information than the trees shown in the **Fig.3**. The confusion matrix for this tree correctly predicted one Cy Young Award winner and misclassified only 11 pitchers, presenting an accuracy of 95.94%. It was at this point we thought maybe one Decision Tree was not enough so we chose to go with the Random Forest algorithm to see if it improved the results. With the Random Forest model, it turns out that while accuracy increases (only misclassifying 7 pitchers), only one Cy Young Award winner is correctly predicted once again. At least these models show us the importance of each variable as shown in **Fig.4**. It makes sense that Quality Starts is the most important one but it is interesting that short day's rest is the least important one of our selected variables.

Next, we decided to go with a Linear Discriminant Analysis and Quadratic Disciminant Analysis models. The LDA model shows that the probability of winning the Cy Young award is slim because pitchers had a 2.2% chance of winning. The outputs are shown in the Appendix. Through the LDA model, we can also see that the data set is unbalanced and biased towards pitchers that did not win the award. We can also see from the LDA model that as the probability of winning the Cy Young increases by one-unit, certain predictors such as losses in game started, bequeathed runners, and long rest days decreased. Otherwise as the probability of winning increases by one-unit, IP, QS, sDR, and Pit.Gs increased. The LDA model had a 97.9% accurate fit, slightly less than the QDA's 99.6% accurate fit. Overall, the predictors were ideal as shown in the ROC curves of the LDA and QDA models in **Fig.5**. The confusion matrix produced (shown in the Appendix) classified more Cy Young Award winners than the tree methods as well, especially the QDA model.

Fig. 4. Variable Importance Plot for Random Forest Model.



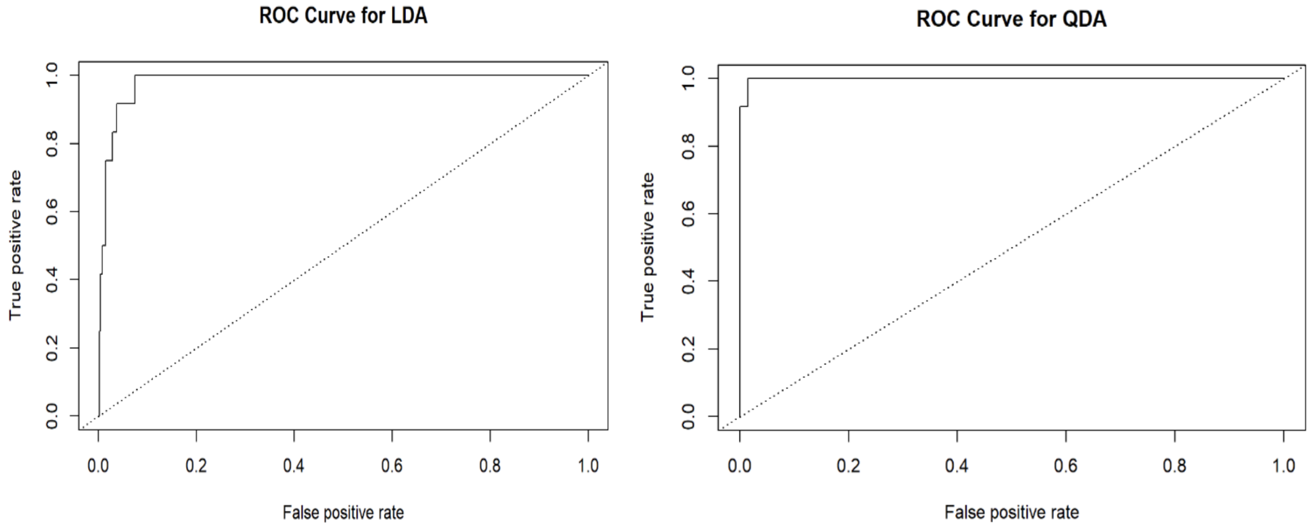


Fig. 5. ROC Curves for both LDA and QDA Models.

Conclusion

Based on our analysis, we see that short days rest, team losses in games started, bequeathed runners, and quality starts were all significant predictors. Also, QDA and LDA did a much better job at correctly classifying who won the Cy Young Award winner as opposed to the tree-based methods only correctly classifying one. In summation, these results showed us that if a Major League pitcher wants to win the Cy Young award, they must not only be really good but also consistent and durable. A pitcher that is consistently pitching will have a better chance at influencing a game as long as that pitcher remains healthy throughout the MLB season.

One of the major challenges we had was extracting the appropriate predictors that we believe will give us more information and avoid the multicollinearity problem. This required a lot of EDA but we believe we got the results that we were hoping to obtain. Also, due to how incredibly unbalanced our data was (only 2% won the Cy Young Award), we had to critically think about how to appropriately split up our data in training and testing. For future studies, perhaps we could either look at Cy Young Award finalists or pitchers who make the All-Star Game so that the data is not as unbalanced. It may also be important to consider some other variables that were not included in the data set.

Appendix

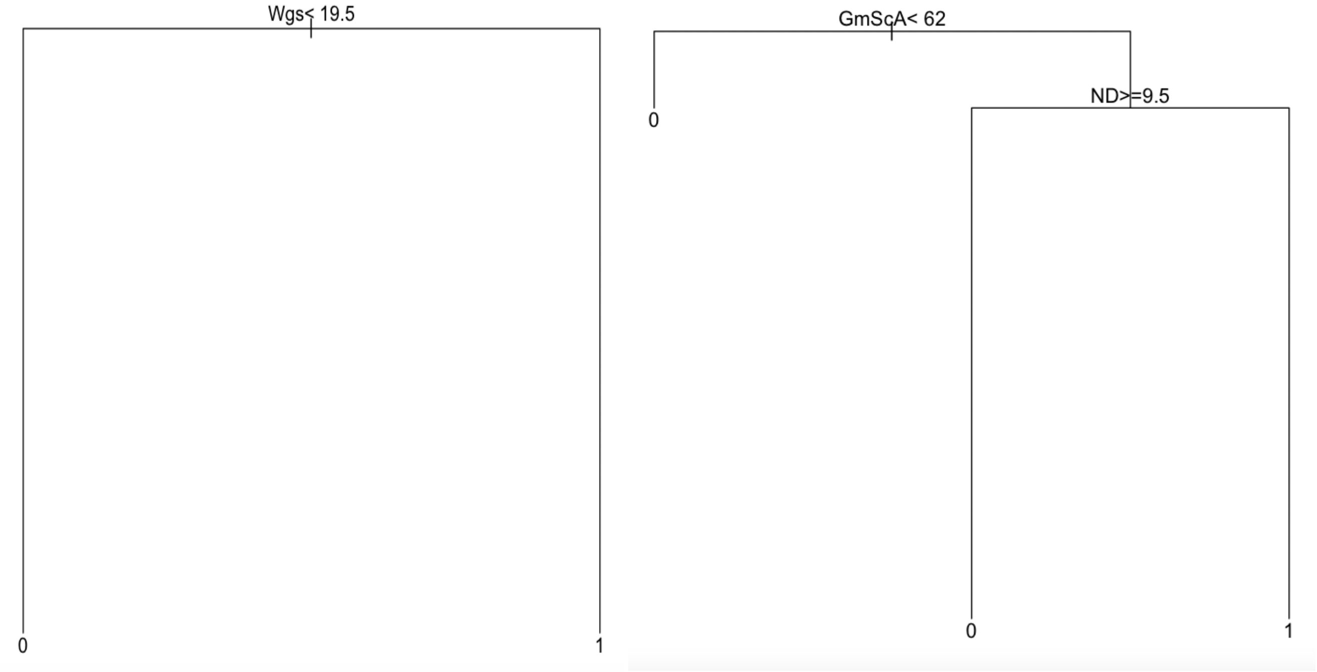


Fig. 6. How our trees looked like had GmScA or Wgs was included in the model.

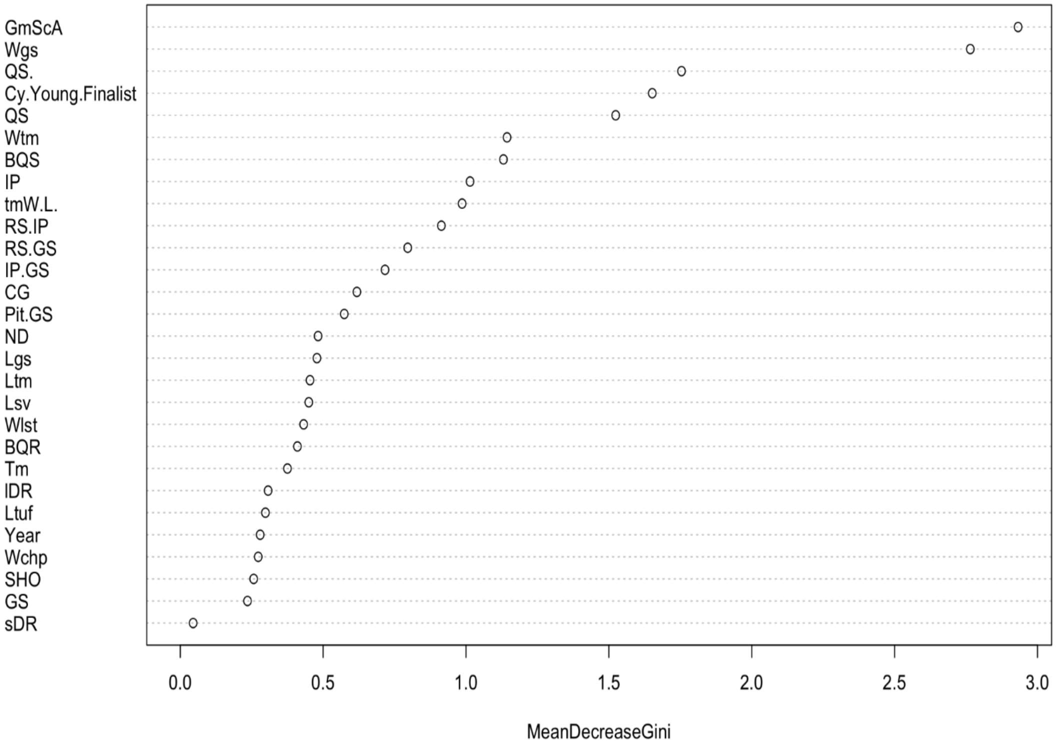
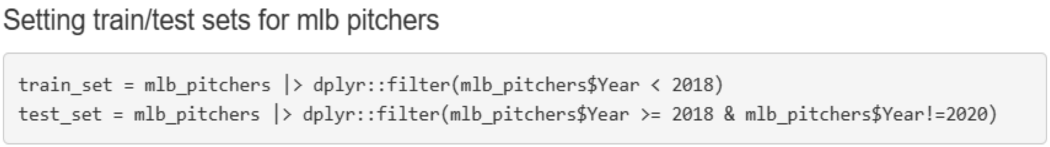
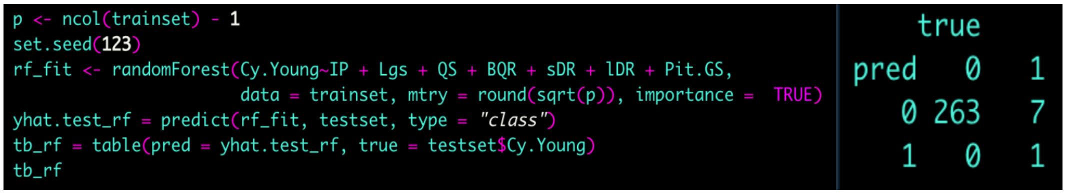


Fig. 7. Variable Importance Plot including every predictor.



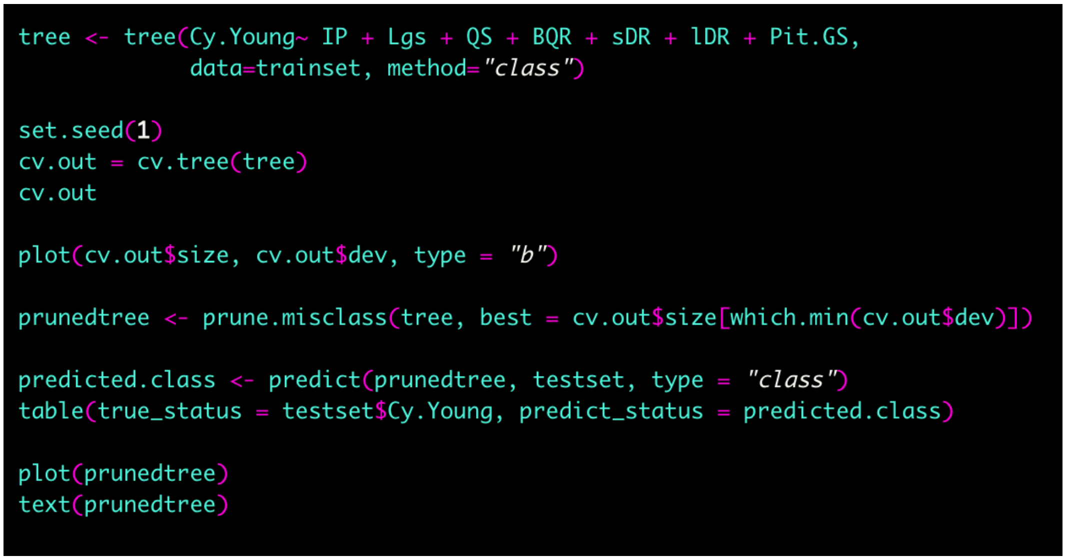
Setting train/test sets for mlb pitchers

```
train_set = mlb_pitchers |> dplyr::filter(mlb_pitchers$Year < 2018)
test_set = mlb_pitchers |> dplyr::filter(mlb_pitchers$Year >= 2018 & mlb_pitchers$Year!=2020)
```

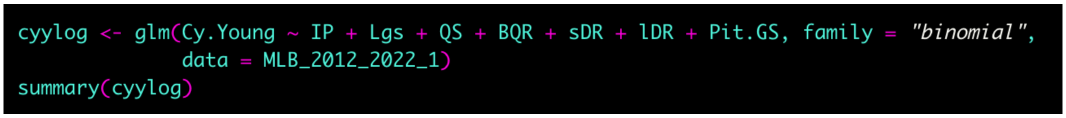


```
p <- ncol(trainset) - 1
set.seed(123)
rf_fit <- randomForest(Cy.Young~IP + Lgs + QS + BQR + sDR + lDR + Pit.GS,
                       data = trainset, mtry = round(sqrt(p)), importance =  TRUE)
yhat.test_rf = predict(rf_fit, testset, type = "class")
tb_rf = table(pred = yhat.test_rf, true = testset$Cy.Young)
tb_rf
```

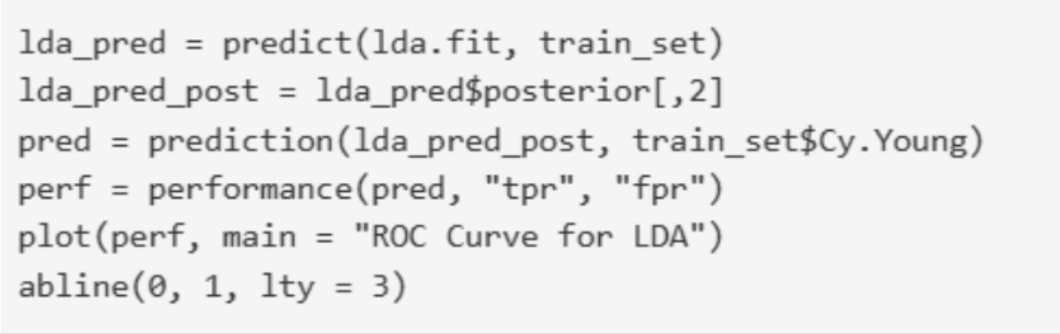
```
     true
pred    0   1
   0  263   7
   1    0   1
```



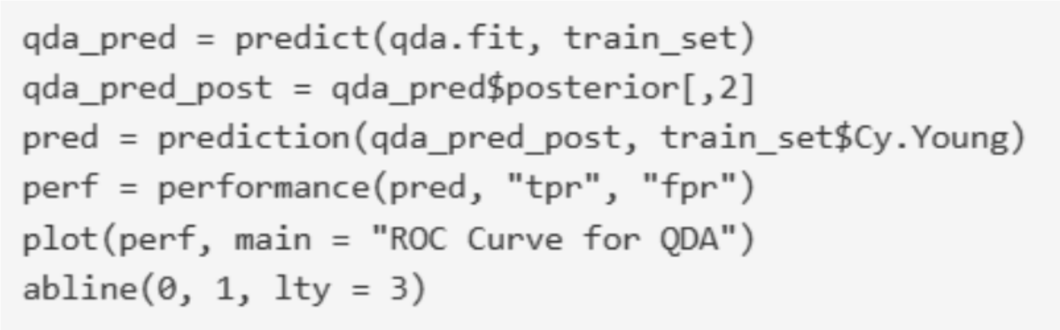
```r
tree <- tree(Cy.Young~ IP + Lgs + QS + BQR + sDR + lDR + Pit.GS,
             data=trainset, method="class")

set.seed(1)
cv.out = cv.tree(tree)
cv.out

plot(cv.out$size, cv.out$dev, type = "b")

prunedtree <- prune.misclass(tree, best = cv.out$size[which.min(cv.out$dev)])

predicted.class <- predict(prunedtree, testset, type = "class")
table(true_status = testset$Cy.Young, predict_status = predicted.class)

plot(prunedtree)
text(prunedtree)
```



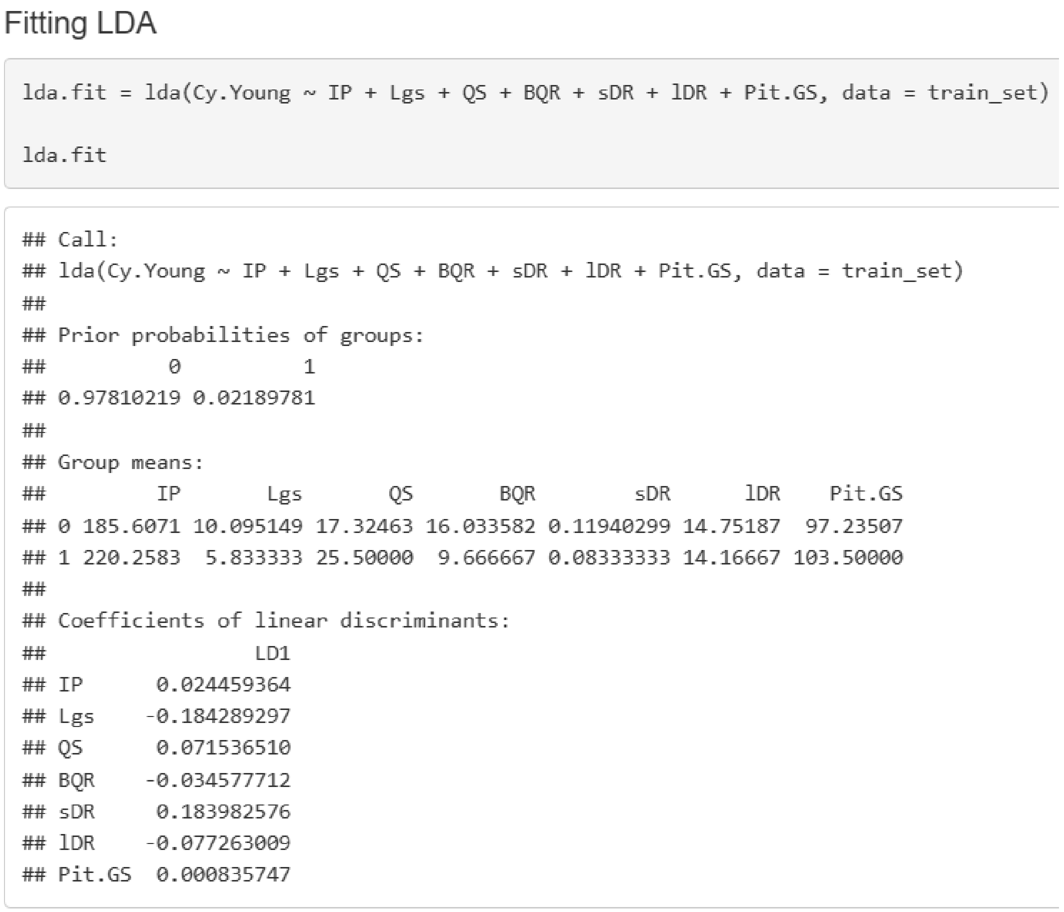
```r
cyylog <- glm(Cy.Young ~ IP + Lgs + QS + BQR + sDR + lDR + Pit.GS, family = "binomial",
              data = MLB_2012_2022_1)
summary(cyylog)
```



```r
lda_pred = predict(lda.fit, train_set)
lda_pred_post = lda_pred$posterior[,2]
pred = prediction(lda_pred_post, train_set$Cy.Young)
perf = performance(pred, "tpr", "fpr")
plot(perf, main = "ROC Curve for LDA")
abline(0, 1, lty = 3)
```



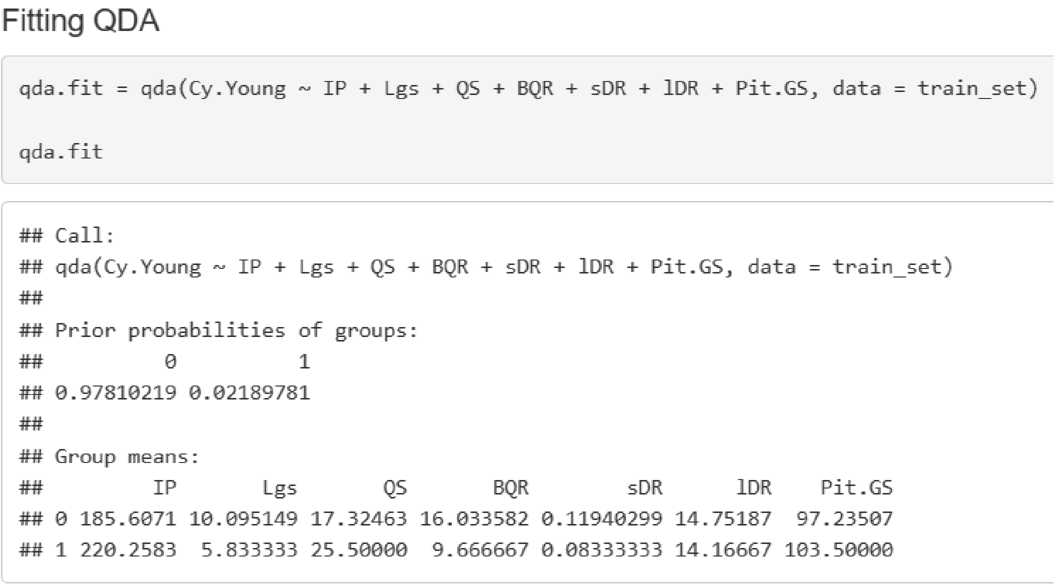
```r
qda_pred = predict(qda.fit, train_set)
qda_pred_post = qda_pred$posterior[,2]
pred = prediction(qda_pred_post, train_set$Cy.Young)
perf = performance(pred, "tpr", "fpr")
plot(perf, main = "ROC Curve for QDA")
abline(0, 1, lty = 3)
```



Fitting LDA

```
lda.fit = lda(Cy.Young ~ IP + Lgs + QS + BQR + sDR + lDR + Pit.GS, data = train_set)

lda.fit
```

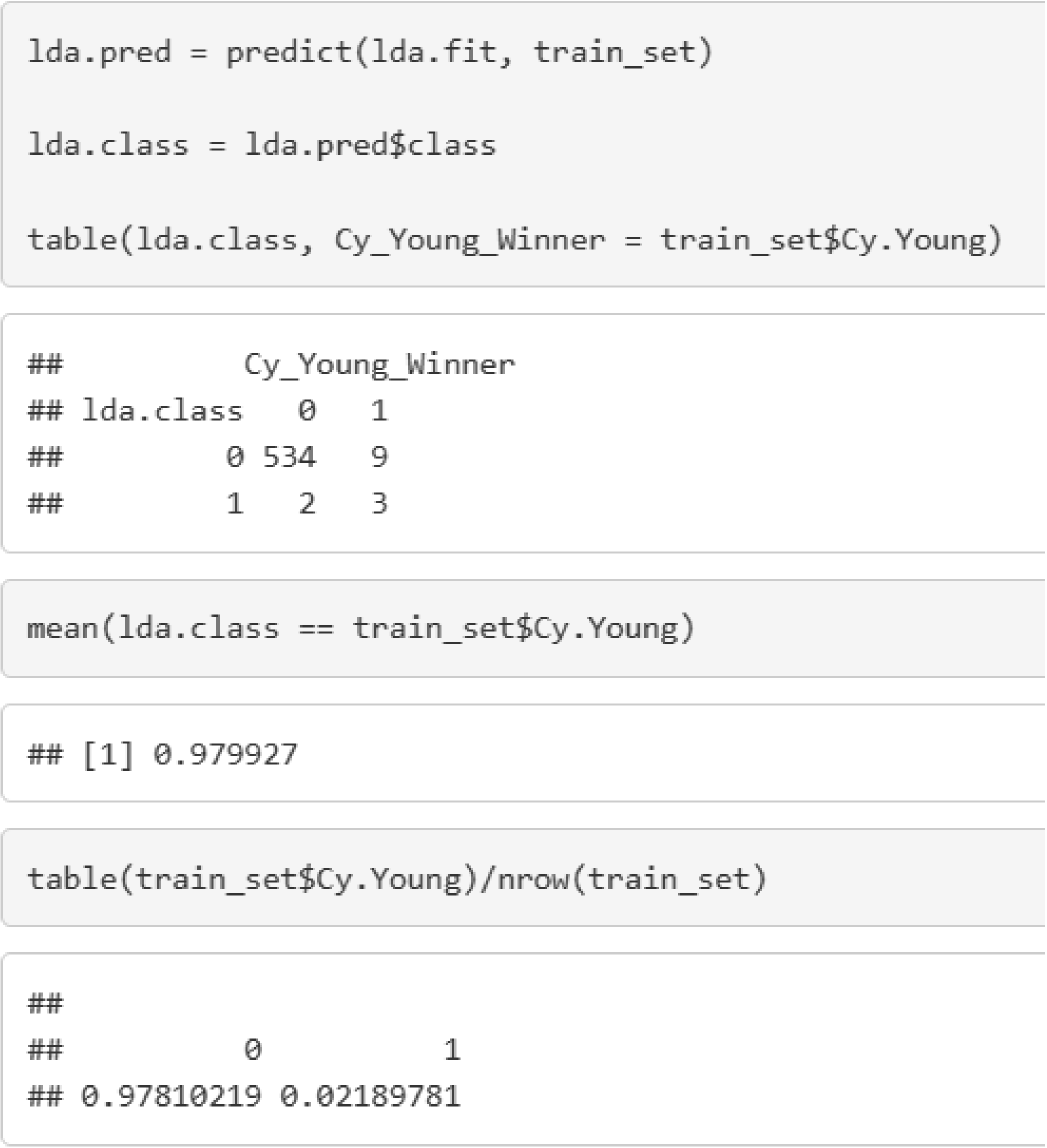
```
## Call:
## lda(Cy.Young ~ IP + Lgs + QS + BQR + sDR + lDR + Pit.GS, data = train_set)
##
## Prior probabilities of groups:
##          0          1 
## 0.97810219 0.02189781 
##
## Group means:
##         IP       Lgs       QS       BQR        sDR      lDR    Pit.GS
## 0 185.6071 10.095149 17.32463 16.033582 0.11940299 14.75187  97.23507
## 1 220.2583  5.833333 25.50000  9.666667 0.08333333 14.16667 103.50000
##
## Coefficients of linear discriminants:
##                 LD1
## IP      0.024459364
## Lgs    -0.184289297
## QS      0.071536510
## BQR    -0.034577712
## sDR     0.183982576
## lDR    -0.077263009
## Pit.GS  0.000835747
```



Fitting QDA

```
qda.fit = qda(Cy.Young ~ IP + Lgs + QS + BQR + sDR + lDR + Pit.GS, data = train_set)

qda.fit
```

```
## Call:
## qda(Cy.Young ~ IP + Lgs + QS + BQR + sDR + lDR + Pit.GS, data = train_set)
##
## Prior probabilities of groups:
##          0          1 
## 0.97810219 0.02189781 
##
## Group means:
##         IP       Lgs       QS       BQR        sDR      lDR    Pit.GS
## 0 185.6071 10.095149 17.32463 16.033582 0.11940299 14.75187  97.23507
## 1 220.2583  5.833333 25.50000  9.666667 0.08333333 14.16667 103.50000
```



```
lda.pred = predict(lda.fit, train_set)

lda.class = lda.pred$class

table(lda.class, Cy_Young_Winner = train_set$Cy.Young)
```

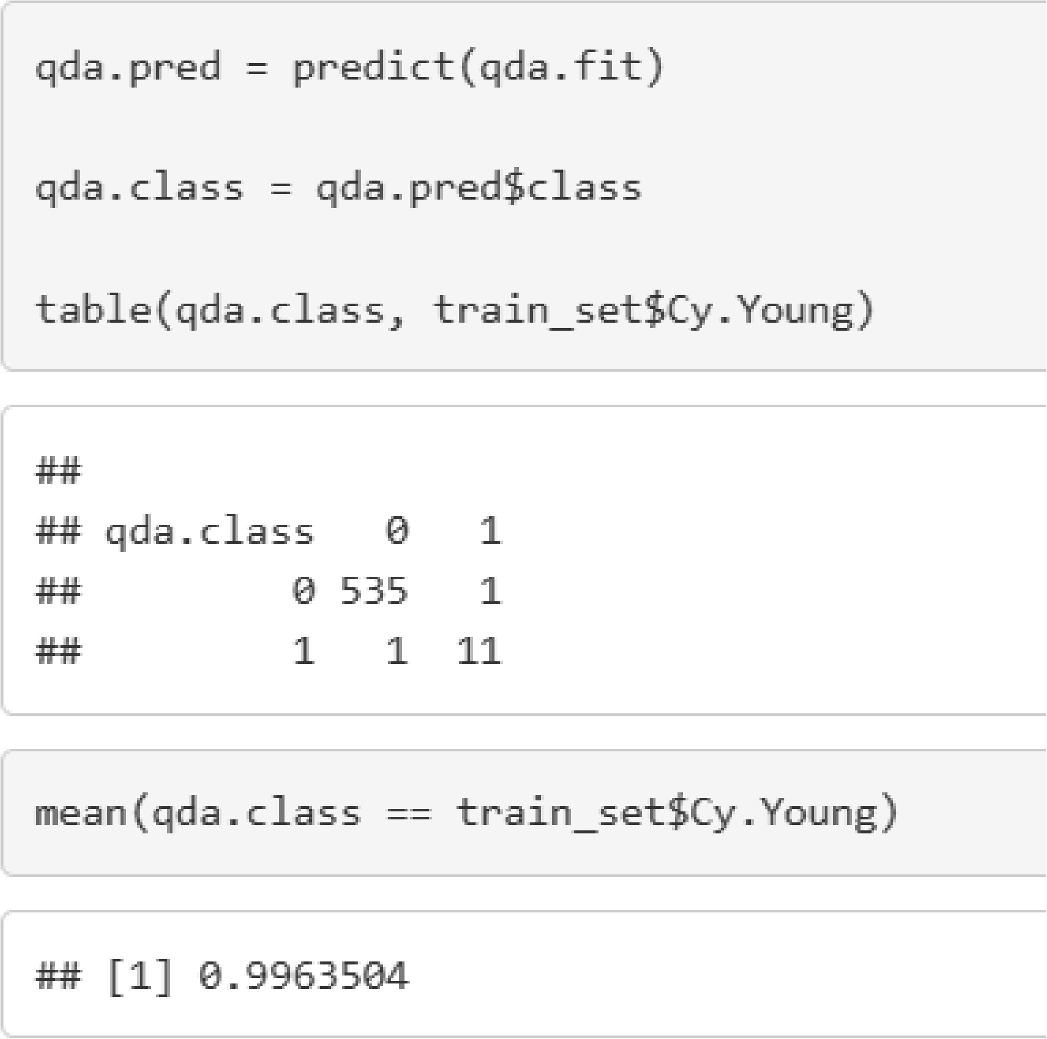
```
##           Cy_Young_Winner
## lda.class   0   1
##         0 534   9
##         1   2   3
```

```
mean(lda.class == train_set$Cy.Young)
```

```
## [1] 0.979927
```

```
table(train_set$Cy.Young)/nrow(train_set)
```

```
## 
##          0          1 
## 0.97810219 0.02189781
```



```
qda.pred = predict(qda.fit)

qda.class = qda.pred$class

table(qda.class, train_set$Cy.Young)
```

```
##
## qda.class   0   1
##         0 535   1
##         1   1  11
```

```
mean(qda.class == train_set$Cy.Young)
```

```
## [1] 0.9963504
```