UNITED STATES MILITARY ACADEMY

SCORE MODULE

MA388: SABERMETRICS

SECTION D1

LTC MICHAEL POWELL

BY

CADET GARRETT S SALISBURY ‘25, CO F3

WEST POINT, NEW YORK

10 MAY 2024

GS I CERTIFY THAT I HAVE COMPLETELY DOCUMENTED ALL SOURCES THAT I USED TO COMPLETE THIS ASSIGNMENT AND THAT I ACKNOWLEDGED ALL ASSISTANCE I RECEIVED IN THE COMPLETION OF THIS ASSIGNMENT.

_____ I CERTIFY THAT I DID NOT USE ANY SOURCES OR RECEIVE ANY ASSISTANCE REQUIRING DOCUMENTATION WHILE COMPLETING THIS ASSIGNMENT.

SIGNATURE: Garrett Salisbury

Can you predict whether or not a baseball player will be an All Star given his prior season's statistics?

Model Assessment through TPR/FPR, PPV/NPV, and F1 Score

Garrett Salisbury

2024-05-08

Learning Objectives

- Understand True Positive (Sensitivity) and True Negative (Specificity) Rates (TPR and TNR, respectively)
- Understand Positive and Negative Predictive Values (PPV and NPV, respectively)
- Understand F1 Score
- Train a predictive model and test it on new data
- Utilize TPR, TNR, PPV, NPV, and F1 Score to assess model adequacy

Introduction

There are many players in the MLB that are considered "perennial All Stars." However, every year there are a handful of players that make the All Star game in an unexpected fashion. All Star game selection does not have a set standard for a player's performance, but rather seems to rely on their popularity among fans. This module aims to see if there is a way to predict which players are poised to make the All Star game based on the statistics of their previous season.

In order to accomplish this, a model must be trained by a data set. This trained model is then applied to "unseen" data that has been split from the same data set as the training data. After the new data has been run against the model, the adequacy of the model can be judged through measures including True Positive and True Negative Rates (Sensitivity and Specificity, respectively), Positive and Negative Predictive Values, and the F1 Score. A further explanation of each measure is provided in the Methods/Instructional Content section.

In order to calculate these measures, a logistic regression model can be used. The different statistics that will be used to classify whether a player would be anticipated to make the All Star game or not are batting average (AVG), home runs (HR), on-base percentage (OBP), and slugging percentage (SLG). A further explanation of AVG, OBP, and SLG will be provided in the Data section.

Data

The Lahman database is a set of data frames including pitching, hitting, fielding, and other information. All statistics from 1871 to 2022 are present in the Lahman database. One major benefit of using the Lahman database is the ability to easily join the tables on year or player names. In order to use the Lahman database, the first step is to install its package. Then, both the Batting and AllstarFull tables must be obtained.

```r
#install.packages("Lahman")
library(Lahman)
library(tidyverse)
library(knitr)
library(caret)

#Save the Batting and AllstarFull tables
#Under usable names
all_hitters <- Batting
asg_apps <- AllstarFull
```

The next step is to mutate the provided columns in the Batting table to produce columns for AVG, OBP, and SLG. The formulas for each are provided below:

Batting Average = $AVG = \frac{H}{AB}$.

On Base Percentage = $OBP = \frac{H+BB+HBP}{AB+BB+HBP+SF}$.

Slugging Percentage = $SLG = \frac{TB}{AB} = \frac{1B+2*2B+3*3B+4*HR}{AB}$

```r
#Mutate the above stats into the hitters table
all_hitters <- all_hitters %>%
  mutate(AVG = H/AB,
          OBP = (H + BB + HBP)/(AB + BB + HBP + SF),
         SLG = (H - X2B - X3B - HR +
             2 * X2B + 3 * X3B + 4 * HR)/AB)
```

The next step is to subtract 1 from the yearID column in the All Star Game table. Also, add a new column that has a value of 1 for all members in the All Star table to show that a given player was an all star. This allows us to see if a player's statistics in one season correspond to their selection to the All Star game in the next. After this, join the two tables by year and player.

```r
#Subtract one year from the all star table
#To enable it to be joined correctly and then
#add a column that marks every player in the table
#with a "1" for being an all star in that season
asg_apps <- asg_apps %>%
  mutate(yearID = yearID - 1) %>%
  mutate(allstar = 1)

#Join the two tables by year and player
all_star_hitters <- all_hitters %>%
  left_join(asg_apps, by = c("yearID", "playerID"), relationship = "many-to-many")
```

Filter out all seasons prior to 2000, as well as the 2022 season (because this season does not have a corresponding column in the All Star table). Select the variables needed to make the regression model (playerID, yearID, AVG, HR, OBP, SLG, and gameNum) Finally, turn all NAs in the data to zeroes to ensure the regression model can run.

```r
#Filter the data set so it contains the
#2000-2021 seasons
#Filter out all players with less than 200 ABs
#select the stats to be used for model
all_star_hitters <- all_star_hitters %>%
  filter(yearID >= 2000) %>%
  filter(yearID < 2022) %>%
  filter(AB > 200) %>%
  select(playerID, yearID, AVG, HR, OBP, SLG, allstar)

all_star_hitters[is.na(all_star_hitters)] <- 0
#Used FavTutor source to implement above line.

all_star_hitters %>%
  summary() %>%
  kable()
```

	playerID	yearID	AVG	HR	OBP	SLG	allstar
	Length:6885	Min. :2000	Min. :0.1429	Min. : 0.00	Min. :0.1745	Min. :0.1867	Min. :0.0000
	Class :character	1st Qu.:2005	1st Qu.:0.2449	1st Qu.: 7.00	1st Qu.:0.3087	1st Qu.:0.3768	1st Qu.:0.0000
	Mode :character	Median :2010	Median :0.2654	Median :12.00	Median :0.3316	Median :0.4228	Median :0.0000
	NA	Mean :2010	Mean :0.2650	Mean :14.09	Mean :0.3332	Mean :0.4285	Mean :0.1217
	NA	3rd Qu.:2015	3rd Qu.:0.2860	3rd Qu.:20.00	3rd Qu.:0.3564	3rd Qu.:0.4742	3rd Qu.:0.0000
	NA	Max. :2021	Max. :0.3724	Max. :73.00	Max. :0.6094	Max. :0.8634	Max. :1.0000

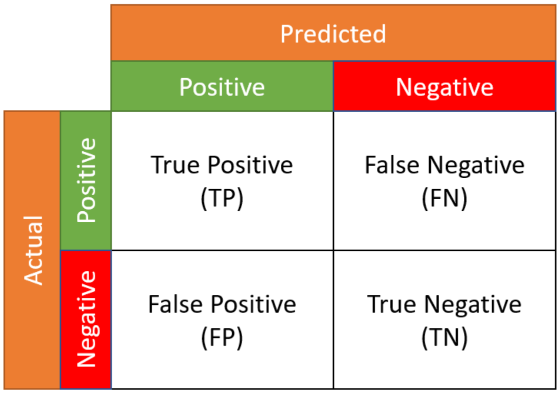
playerID	yearID	AVG	HR	OBP	SLG	allstar
abbotje01	2000	0.274	3	0.343	0.395	0
abreubo01	2000	0.316	25	0.416	0.554	0
agbaybe01	2000	0.289	15	0.391	0.477	0
alfoned01	2000	0.324	25	0.425	0.542	0
alicelu01	2000	0.294	6	0.365	0.404	0
alomaro01	2000	0.310	19	0.378	0.475	1
alomasa02	2000	0.289	7	0.324	0.404	0
aloumo01	2000	0.355	30	0.416	0.623	1
anderbr01	2000	0.257	19	0.375	0.421	0
anderga01	2000	0.286	35	0.307	0.519	0

Methods/Instructional Content

There are many measures that can be used to assess the performance of a model. Some of these measures include Sensitivity, Specificity, Positive Predictive Value (PPV), Negative Predictive Value (NPV), and F1 Score.

The Monaghan et. al journal article provides a great deal of insight into what the sensitivity, specificity, PPV, and NPV represent.

A "True Positive" represents a data point that is predicted to be True by the model and is actually True. A "True Negative" represents a data point that is predicted to be False by the model and is actually False. A "False Positive" is a data point that is predicted to be True by the model, but is in fact False. A "False Negative" is a data point that is predicted to be False by the model, but is actually True. The confusion matrix presented below provides of visualization of these four concepts.



		Predicted	
		Positive	Negative
Actual	Positive	True Positive (TP)	False Negative (FN)
	Negative	False Positive (FP)	True Negative (TN)

Figure 1: Confusion Matrix (source: medium.com)

The Crandon article summarizes what each measure shown above represents. Sensitivity indicates how likely a model is to classify something as true if the data point is actually true. Specificity shows how likely a model is to classify something as false if the data point is actually false. The PPV represents how likely a data point is to be true if the model classifies it as true. The NPV indicates how likely a data point is to be false if the model classifies it as false. The formulas for each are as follows:

$Sensitivity = \frac{TruePositives}{TruePositives+FalseNegatives}$.

$Specificity = \frac{TrueNegatives}{TrueNegatives+FalsePositives}$.

$PPV = \frac{TruePositives}{TruePositives+FalsePositives}$.

$NPV = \frac{TrueNegatives}{TrueNegatives+FalseNegatives}$.

Finally, the Kundu article summarizes how the calculate the F1 score of a model and what this score represents. The F1 score is used to assess the accuracy of a model, relating to the number of correct predictions a model made. The F1 score is a combination of the Precision and Recall of the model. Precision looks at how many True predictions made by the model were True in reality. The Recall of a model is how often values that are True in reality were classified as True by the model. The formula for Precision is the same as that of PPV, and the formula of Recall is the same as Sensitivity. A higher F1 Score for a model indicates high values for both Precision and Recall. The F1 Score indicates how well a model is at classifying values based on what they are in reality and that the predictions are in fact correct.

Using the formulas for Precision and Recall, the formula for F1 Score is

$F1Score = \frac{2}{(Precision)^{-1}+(Recall)^{-1}}$

Or simplified as

$F1Score = \frac{TruePositives}{TruePositives+.5(FalsePositives+FalseNegatives)}$

All three of these sources provided background and explanations of the aforementioned scores and how each is used to assess a model.

Exercises/Activities

First, split the data set into a training and testing set.

```r
set.seed(679)
split<- createDataPartition(all_star_hitters$allstar, p = .75, list = FALSE)
train <- all_star_hitters[split,]
test <- all_star_hitters[-split,]
```

Next, create a logistic regression model that classifies a hitter as an All Star or not using AVG, HR, OBP, and SLG.

```r
all_star_mod <- glm(allstar ~ AVG + HR + OBP + SLG, data = train,
                    family = "binomial")
```

Now that you have trained a logistic regression model that quantifies a player's All Star status, use the test data set created above to predict whether a hitter would be an All Star or not using the model.

```r
pred <- predict(all_star_mod, newdata = test, type = "response")
summary(pred)
```

```
##     Min.  1st Qu.   Median     Mean  3rd Qu.     Max.
## 0.004244 0.041694 0.075333 0.115808 0.141457 0.845290
```

Create a matrix that contains the number of each value (True Positive etc.). Use a threshold of 0.2 to classify whether a player is an All Star or not.

```r
threshold <- 0.2
allstar_matrix <- table(test$allstar, pred >= threshold)
allstar_matrix %>%
  kable(col.names = )
```

	FALSE	TRUE
0	1344	163
1	124	90

Finally, calculate each of the metrics from above (Sensitivity etc.). Reference the example Confusion Matrix to determine which values are needed to compute each value.

```r
sensitivity <- allstar_matrix[2,2]/(allstar_matrix[2,2] + allstar_matrix[2,1])

specificity <- allstar_matrix[1,1]/(allstar_matrix[1,1] + allstar_matrix[1,2])

ppv <- allstar_matrix[2,2]/(allstar_matrix[2,2] + allstar_matrix[1,2])

npv <- allstar_matrix[1,1]/(allstar_matrix[1,1] + allstar_matrix[2,1])

fscore <- allstar_matrix[2,2]/(allstar_matrix[2,2] + .5*(allstar_matrix[1,2] + allstar_matrix[2,1]))
```

Show a table of each value, labeled with its respective measure.

```r
values <- c(sensitivity, specificity, ppv, npv, fscore)
measures <- c("Sensitivity", "Specificity", "PPV", "NPV", "F1 Score")
stats <- data.frame(`measures`, `values`)
stats %>%
  kable(digits = 4)
```

measures	values
Sensitivity	0.4206
Specificity	0.8918
PPV	0.3557
NPV	0.9155
F1 Score	0.3854

As shown above, the model is exceptionally good at predicting false values. This is shown by high Specificity and NPV. As a reminder, Sensitivity indicates how likely a model is to classify something as true if the data point is actually true while NPV represents how likely a data point is to be false if the model classifies it as false.

However, the model is extremely lacking in its ability to predict true values, as shown by low Sensitivity and PPV values. To reiterate, Sensitivity indicates how likely a model is to classify something as true if the data point is actually true and PPV represents how likely a data point is to be true if the model classifies it as true.

Additionally, the F1 Score of this model is very low, indicating that this model is not very accurate. Again, F1 score represents the number of correct predictions a model made. This indicates that this model may not be very useful at classifying players as All Stars or not depending on their prior season's statistics.

Wrap-Up/Conclusions

This lesson introduced how to create a logistic regression model, with this model then being used to classify unseen data. Finally, the model's adequacy was assessed on its ability to classify the unseen data through measures of Sensitivity, Specificity, Positive Predictive Value, and Negative Predictive Value. This was all applied to how well a logistic regression model could classify a player as an All Star or not depending on his statistics the previous season.

As shown, logistic regression is not very successful at classifying a player as an All Star by using the previous season's statistics. This is likely because of season-to-season fluctuations in "successful" player performance. Statistics that could be considered good in one year may be considered average in another. One way to make this model stronger would be to standardize all statistics in order to have a common comparison from season to season.

The biggest limitation of this exercise is that the data set contains a much larger number of players that are not All Stars than players that are All Stars. This is because there are very few players that are selected to the All Star Game every year. One potential mitigation of this issue would be to increase the minimum number of at bats for a player to be considered in the analysis. This would filter out even more players that had no chance of making the All Star game, likely leading to higher values of the performance measures described above.

Another sports application for this would be classifying a player as a Hall of Famer or not depending on their career statistics and then assessing how adequate the logistic regression model is by testing it on unseen data. This might work better because although no performance benchmarks have been set to be selected for the Hall of Fame, the performance of inductees is much more consistent than players selected to be All Stars.

An additional skill that builds on this idea is to calculate the Out-of-Sample R Squared, Root Mean Square Error, and Mean Absolute Error to further assess how strong a model is at classifying unseen data. These values could then be compared to other models using the same data, but with altered filters on the minimum number of ABs to see what benchmark for ABs is the most appropriate.

Works Cited

Crandon, Saul. “Sensitivity and Specificity Explained: A Cochrane UK Trainees Blog.” *Cochrane.org*, 2019, uk.cochrane.org/news/sensitivity-and-specificity-explained-cochrane-uk-trainees-blog.

Kundu, Rohit. “F1 Score in Machine Learning: Intro & Calculation.” *Www.v7labs.com*, 16 Dec. 2022, www.v7labs.com/blog/f1-score-guide.

Monaghan, Thomas F et al. “Foundational Statistical Principles in Medical Research: Sensitivity, Specificity, Positive Predictive Value, and Negative Predictive Value.” *Medicina (Kaunas, Lithuania)* vol. 57,5 503. 16 May. 2021, doi:10.3390/medicina57050503

“Replace NA with 0 (Zero) in R | 5 Methods (with Code).” *FavTutor*, favtutor.com/blogs/replace-na-with-zero-r.

This source was used to turn all NAs in the data into zeroes. It was implemented in line 79.