Using Regression Analysis to Predict MLB Attendance

Josh Hancock

2017-06-02

The standard MLB season is 162 games long and each team typically plays 81 games at home and 81 games in the stadiums of the opposing teams. Using data from historical MLB seasons, we can build a model to predict the total attendance at the 81 home games for each team. In many business applications, it's of particualr interest to stakeholders to understand which factors influence the revenue-creating side of the business. For this reason, we'll choose to use a linear regression model, which may fall short of deep-learning models when it comes to predictive power, but will provide an output that is interpritable and may allow for a better understaning of the business as a whole.

First, we'll obtain and clean data. Attendance can be influenced by many factors beyond the playing field, so it will be important to include data on both team performance factors and the demographics of the city and fan base for each team. For team data, we can use Baseball Reference to grab results between 2006 and 2014 for each of the 30 teams, giving us 270 observations. In addition to the win/loss records of each team, we have home attendance, wins, payroll, stadium name, stadium capacity, stadium age, number of years a team has been in a city, playoff appearances for each team, the number of all-stars for each team, the number of home-runs hit by each team, and the number of professional sports teams in in each city for each observation.

Using publically available demographic data for each city, we can also include the number of people, the number of households, and the median income for various drive times (15, 30, 45, and 60 minutes) from each stadium. We then imported the data into R and inspected it for any obvious problems (graphically and using the *summary* command).

```r
mlbattendance_final = read.csv("mlbattendance_final.csv", header = TRUE)
attach(mlbattendance_final)
summary(mlbattendance_final)
```

```
##                   observationName   nextAttend    currentAttend
##  arizona diamondbacks_2006:  1    Min.   :1287   Min.   :1164
##  arizona diamondbacks_2007:  1    1st Qu.:1952   1st Qu.:1969
##  arizona diamondbacks_2008:  1    Median :2435   Median :2422
##  arizona diamondbacks_2009:  1    Mean   :2498   Mean   :2506
##  arizona diamondbacks_2010:  1    3rd Qu.:3035   3rd Qu.:3046
##  arizona diamondbacks_2011:  1    Max.   :4299   Max.   :4299
##  (Other)                  :264
##     currentW          priorW         X.capacity        payrollM
##  Min.   : 51.00   Min.   : 51.00   Min.   :0.3730   Min.   : 14.67
##  1st Qu.: 73.00   1st Qu.: 73.00   1st Qu.:0.5663   1st Qu.: 67.82
##  Median : 81.00   Median : 81.00   Median :0.6750   Median : 87.45
##  Mean   : 80.99   Mean   : 81.03   Mean   :0.7052   Mean   : 94.23
##  3rd Qu.: 90.00   3rd Qu.: 90.00   3rd Qu.:0.8588   3rd Qu.:110.18
##  Max.   :103.00   Max.   :103.00   Max.   :1.0680   Max.   :258.12
##
##    stadiumCap      stadiumAge      yearsInCity          playoffs
##  Min.   :31042   Min.   :  1.00   Min.   :  2.00   lcs       : 18
##  1st Qu.:40941   1st Qu.:  9.00   1st Qu.: 38.00   lds       : 41
##  Median :42319   Median : 15.00   Median : 48.50   no_playoff:193
##  Mean   :43893   Mean   : 23.88   Mean   : 65.07   ws        : 18
##  3rd Qu.:48647   3rd Qu.: 25.00   3rd Qu.:111.00
##  Max.   :57333   Max.   :103.00   Max.   :139.00
##
```

```
##  playoffsBin    proTeams        allstars         hrs
##  no :193     Min.   : 2.000   Min.   :1.000   Min.   : 91
##  yes: 77     1st Qu.: 3.000   1st Qu.:1.000   1st Qu.:137
##              Median : 4.000   Median :2.000   Median :160
##              Mean   : 4.733   Mean   :2.474   Mean   :160
##              3rd Qu.: 6.000   3rd Qu.:3.000   3rd Qu.:180
##              Max.   :11.000   Max.   :8.000   Max.   :257
##
##      pop15             pop30             pop45              pop60
##  Min.   : 172347   Min.   : 455797   Min.   : 1119922   Min.   : 1678338
##  1st Qu.: 349174   1st Qu.:1149847   1st Qu.: 1951126   1st Qu.: 2625301
##  Median : 514956   Median :1637438   Median : 2746328   Median : 3733633
##  Mean   : 685767   Mean   :2235462   Mean   : 3663535   Mean   : 4810492
##  3rd Qu.: 692890   3rd Qu.:2286908   3rd Qu.: 3857654   3rd Qu.: 5102564
##  Max.   :3081588   Max.   :8610868   Max.   :12555131   Max.   :14786653
##
##   households15      households30      households45      households60
##  Min.   :  70449   Min.   : 198319   Min.   : 475711   Min.   : 665944
##  1st Qu.: 144285   1st Qu.: 481509   1st Qu.: 763531   1st Qu.: 968038
##  Median : 210898   Median : 626307   Median :1057057   Median :1395412
##  Mean   : 269708   Mean   : 845642   Mean   :1353362   Mean   :1764438
##  3rd Qu.: 300642   3rd Qu.: 857852   3rd Qu.:1443046   3rd Qu.:1916014
##  Max.   :1203728   Max.   :3238724   Max.   :4610213   Max.   :5393836
##
##     medInc15        medInc30        medInc45        medInc60
##  Min.   :24819   Min.   :40628   Min.   :45596   Min.   :49146
##  1st Qu.:41533   1st Qu.:50821   1st Qu.:55944   1st Qu.:56823
##  Median :45979   Median :53680   Median :59741   Median :63115
##  Mean   :50822   Mean   :58755   Mean   :63297   Mean   :64909
##  3rd Qu.:58953   3rd Qu.:63638   3rd Qu.:68737   3rd Qu.:69686
##  Max.   :78935   Max.   :90308   Max.   :95777   Max.   :94637
##
```

reserve 10% of our data for testing purposes before starting our analysis.

```
nrow(testdata)
```

```
## [1] 27
```

```
nrow(traindata)
```

```
## [1] 243
```

We begin our analysis with 243 observations x 27 variables (including observation names) for the training data set.We started our analysis with a base model:

```
basemod <- lm(nextAttend ~ currentAttend + currentW + priorW + X.capacity + payrollM +
                stadiumCap + stadiumAge + yearsInCity + playoffsBin + proTeams +
                allstars + hrs + pop15 + pop30 + pop45 + pop60 + households15 +
                households30 + households45 + households60 + medInc15 + medInc30 + medInc45 +
                medInc60,data=traindata)
```

Note: See Appendix A for an explanation of the data and variable names

Initially, we suspected a strong correlation between many variables in our data set. We started by looking at the correlation matrix (See Appendix B).

There were many strong correlations, especially with the demographic data. We decided to build a different base model for each level of drive time (15,30,45,60) data to determine which one has the most significance in the current model:

```
lmod15<-lm(nextAttend ~ currentAttend + currentW + priorW + X.capacity + payrollM +
             stadiumCap + stadiumAge + yearsInCity + playoffsBin + proTeams + allstars +
             hrs + pop15 + households15 + medInc15,data=traindata)


lmod30<-lm(nextAttend ~ currentAttend + currentW + priorW + X.capacity + payrollM +
             stadiumCap + stadiumAge + yearsInCity + playoffsBin + proTeams + allstars +
             hrs + pop30 + households30 + medInc30,data=traindata)


lmod45<-lm(nextAttend ~ currentAttend + currentW + priorW + X.capacity + payrollM +
             stadiumCap + stadiumAge + yearsInCity + playoffsBin + proTeams + allstars +
             hrs + pop45 + households45 + medInc45,data=traindata)


lmod60<-lm(nextAttend ~ currentAttend + currentW + priorW + X.capacity + payrollM +
             stadiumCap + stadiumAge + yearsInCity + playoffsBin + proTeams + allstars +
             hrs + pop60 + households60 + medInc60,data=traindata)
```

All models were similar in adjusted r^2, so we selected the 60-minute model, which seemed to have the most significance in the individual drive-time variables. Even after selecting a single level of demographic data, there still seemed to be issues with correlated predictors, so we decided to view the variance inflation factor(VIF) for the 60-minute model:

```
vif(lmod60)
```

```
## currentAttend      currentW        priorW    X.capacity      payrollM
##    154.316052      3.063643      1.412947    122.221320      3.017898
##    stadiumCap    stadiumAge   yearsInCity   playoffsBin      proTeams
##     30.163883      1.310847      2.035686      2.189777      8.621885
##      allstars           hrs         pop60  households60       medInc60
##      1.774643      1.323476    193.680221    223.787559      1.781134
```

There seems to be two issues that need to be addressed. There is a very large VIF for currentAttend, X.capacity, pop60, and households60. stadiumCap also has a large VIF, but we will choose to address that after addressing the higher values. We start by removing X.capacity and households60.

```
lmod <- lm(nextAttend ~ currentAttend + currentW + priorW + payrollM + stadiumCap + stadiumAge + yearsI
vif(lmod)
```

```
## currentAttend      currentW        priorW      payrollM    stadiumCap
##      2.811035      3.010444      1.402670      2.856213      1.723035
##    stadiumAge   yearsInCity   playoffsBin      proTeams      allstars
##      1.170548      1.848468      2.155888      6.465883      1.734736
##           hrs         pop60      medInc60
##      1.239714      6.656949      1.627345
```

All the VIF levels are under 10 (including *stadiumCap*), so we now move to graphically checking the variance (see Appendix C for plot).

From the plot we can safely assess that the variance appears to be constant and no further investigation is needed. Next, we will check normality assumptions (see Appendix C for plot).

The qqplot appears to show that the data is short-tailed, which is acceptable. We can conclude that no transformation of our model is needed because there doesn't appear to be any problems with variance or linearity.

Next we will check for high leverages in our data:

```
hatv <- hatvalues(lmod)
threshold <- (2*14)/243
hatv_true <- hatv>threshold
which(hatv_true, useNames = TRUE)
```

```
## 146 154 155 157 158
## 132 139 140 142 143
```

There are a few observations that have a higher leverage than the *2p/n* ratio of 0.1037 (see Appendix C for plot). At this point we decide that they were not severe enough to immediately remove and should be assessed in presence of other tests. Next, we looked for outliers using the Bonferoni Correction:

```
stud <- rstudent(lmod)
```

This gives us studentized residuals. Now, calculate the Bonferoni critical value:

```
bonf <- qt((0.05/243*2),232)
abs_bonf <- abs(bonf)
abs_stud <- abs(stud)
bonf_points <- abs_stud > abs_bonf
abs_bonf
```

```
## [1] 3.389387
```

As we an see, there is one observation that exceeds the Bonferoni critical value. Because the data set is fairly large (n=243), we are not overly concerned with outliers. In order to test for influential points, we calculated the Cooks Distance for the data.

```
cook <- cooks.distance(lmod)
```

We then plotted the half normal plot of Cooks Distance and use the $4/(n-p-1)$ rule of thumb to check for any influential points in the data (see Appendix C for plot). From this we identified 12 points that appear to be influential points. We then removed those points and moved on with our diagnostics.

```
newtrain <- subset(traindata, cook < 0.01754)
```

Confident that our data and model assumptions are sound, we chose to move on to the shrinkage phase of the diagnostics.

```
b <- regsubsets((nextAttend ~ currentAttend + currentW + priorW + payrollM + stadiumCap + stadiumAge + 
rs <- summary(b)
```

Note: See Appendix E for predictor logic matrix

```
AIC <- 231*log(rs$rss/231) + (2:14)*2
which.min(AIC)
```

```
## [1] 11
```

AIC suggests the nine predictor model (see Appendix C for plot). Next, we'll look at the *Mallows CP* criterion (see Appendix C for plot). *Mallows CP* suggests a model with ten predictors. Next, we looked at *adjusted* r^2.

```
which.max(rs$adjr2)
```

```
## [1] 11
```

Adjusted r^2 suggests 10 predictors. As we did further analysis to decide on a final model, something became clear. The model that we were considering had one predictor that was much more significant than the others:

currentAttend. Including this predictor in our model gave us a higher degree of accuracy, which is desirable in a prediction model. However, this predictor appeared to already contain much of the information we sought to include in our model by using additional variables and would possibly limit the amount of inference that could be achieved compared to a model that uses more predictors. We decided to branch our model into two different versions: one with *currentAttend* and one without. We checked the Cooks Distances for this model with the original training data set and found that slightly fewer points seemed to be influential, so we made a separate subset for the second model.

```
cook2 <- cooks.distance(lmod2)
newtrain2 <- subset(traindata, cook2 < 0.01746)
```

Here is the second version of the model:

```
lmod2 <- lm(nextAttend ~ currentW + priorW + payrollM + stadiumCap + stadiumAge + yearsInCity + playoffs
```

After running the same diagnostics on the second model as we did on the original, *AIC* suggested ten predictors, *CP* suggested 11 predictors, and *adjusted* r^2 suggested 11.

After considering the suggested number of predictors from each criterion, we removed predictors using the logic matrix and came up with the following models:

```
final_lmod <- lm(nextAttend ~ currentAttend + currentW + priorW + payrollM + stadiumCap + stadiumAge + y

final_lmod2 <- lm(nextAttend ~ hrs + stadiumAge + pop60 + priorW + playoffsBin + medInc60 + payrollM + c
```

Taking a look at the coefficients, we decided to scale *stadiumCap* to the same units as the attendance numbers.

```
final_lmod <- lm(nextAttend ~ currentAttend + currentW + priorW + payrollM + I(stadiumCap/1000) + stadiu

final_lmod2 <- lm(nextAttend ~ hrs + stadiumAge + pop60 + priorW + playoffsBin + medInc60 + payrollM + c

sumary(final_lmod)
```

```
##                      Estimate  Std. Error t value  Pr(>|t|)
## (Intercept)        -502.802796  173.147081 -2.9039  0.004063
## currentAttend         0.926007    0.031199 29.6808 < 2.2e-16
## currentW              7.780165    1.342999  5.7931 2.383e-08
## priorW               -3.459886    1.344174 -2.5740  0.010713
## payrollM             -2.189897    0.541452 -4.0445 7.267e-05
## I(stadiumCap/1000)    9.281591    2.917450  3.1814  0.001678
## stadiumAge            1.007931    0.547017  1.8426  0.066740
## yearsInCity           0.705325    0.362104  1.9479  0.052711
## proTeams             13.101610    7.294260  1.7962  0.073848
##
## n = 228, p = 9, Residual SE = 189.69973, R-Squared = 0.92
```

```
sumary(final_lmod2)
```

```
##                  Estimate  Std. Error t value  Pr(>|t|)
## (Intercept)    -1.9514e+03  4.4582e+02 -4.3772 1.841e-05
## hrs            -1.3754e+00  8.8423e-01 -1.5555 0.1212341
## stadiumAge      1.8179e+00  1.1481e+00  1.5834 0.1147250
## pop60           1.7194e-05  9.9079e-06  1.7353 0.0840504
## priorW          5.4880e+00  2.7919e+00  1.9657 0.0505628
## playoffsBinyes  2.0216e+02  8.3258e+01  2.4280 0.0159659
## medInc60        7.3176e-03  2.7352e-03  2.6754 0.0080134
## payrollM        3.3668e+00  1.0381e+00  3.2432 0.0013613
## currentW        1.3343e+01  3.8875e+00  3.4323 0.0007122
```

```
## yearsInCity     4.0868e+00  8.0721e-01  5.0628 8.589e-07
## I(stadiumCap)   4.3410e-02  5.7577e-03  7.5395 1.149e-12
##
## n = 236, p = 11, Residual SE = 399.71649, R-Squared = 0.65
```

For an interpretation of the coefficients, we start with the model containing *currentAttend*:

intercept: no meaningful interpretation (not possible to have negative attendance)
currentAttend: for every 1000 people that attend in the current year, 926 people can be expected to attend next year (ceteris paribus)
currentW: for each additional game a team wins, we can expect and additional 7780 people to attend the next year (ceteris paribus)
priorW: for each game a team won last season, the expected attendance will drop by 3459 people in two seasons(ceteris paribus). This is counterintuitive and may be due to a correction effect resulting from other predictors
payrollM: for each additional million dollars a team spends on payroll, the attendance of the next season can be expected to drop by 2189 (ceteris paribus). This is also counterintuitive and could also be a correcting effect.
stadiumCap: for each additional 1000 seats in capacity, the attendance can be expected to increase by 9281 people over the course of a season(ceteris paribus)
stadiumAge: for each additional year in stadium age, the attendance can be expected to increase by 1007(ceteris paribus)
yearsInCity: for each additional year a franchise has been located in its current city, we can expect an additional 705 people to attend (ceteris paribus)
proTeams: for every additional professional team in the metro area, attendance will increase by 13101 per season (ceteris paribus)

There are a few differences in the coefficients between the two models. Most notably, all of the coefficients became positive (except for the intercept and *hrs*) in the second model. This leads us to believe that the *currentAttend* data was a very powerful vacuum, so to speak, and all the information that is sucked up by and contained inside of it needs to be corrected by other covariates contained in the model. In the absence of *currentAttend*, many of the other predictors' significance levels increased as they absorbed some of the significance abandoned by *currentAttend*. Additionally, *hrs*, *pop60*, and *medInc60* are included in the model and are significant at the 0.15, 0.10, and 0.01 levels, respectively.

We decided to use both models to fit values to the test data. First, we define a function that calculates *rmse*:

```r
rmse <- function(x,y)sqrt(mean((x-y)^2))
```

Now we will compare the two models.

The model with *currentAttend*:

```r
rmse(fitted(final_lmod),newtrain$nextAttend)
```

```
## [1] 185.918
```

```r
rmse(predict(final_lmod,testdata),testdata$nextAttend)
```

```
## [1] 206.7607
```

The model without *currentAttend*:

```r
rmse(fitted(final_lmod2),newtrain2$nextAttend)
```

```
## [1] 390.2899
```

```r
rmse(predict(final_lmod2,testdata),testdata$nextAttend)
```

```
## [1] 425.695
```

As we expected, the model that includes *currentAttend* has the lower *rmse* value for the train and test cases. Graphically: