DATA 621 HW 1

Last edited September 13, 2023

Business Analytics and Data Mining

Homework #1 Assignment Requirements

Overview

In this homework assignment, you will explore, analyze and model a data set containing approximately 2200 records. Each record represents a professional baseball team from the years 1871 to 2006 inclusive. Each record has the performance of the team for the given year, with all of the statistics adjusted to match the performance of a 162 game season.

Your objective is to build a multiple linear regression model on the training data to predict the number of wins for the team. You can only use the variables given to you (or variables that you derive from the variables provided).

Below is a short description of the variables of interest in the data set:

Variable Names	Definition	Theoretical Effect
INDEX	Identification Variable (do not use)	None
TARGET_WINS	Number of wins	$12
TEAM_BATTING_H	Base Hits by batters (1B,2B,3B,HR)	Positive Impact on Wins
TEAM_BATTING_2B	Doubles by batters (2B)	Positive Impact on Wins
TEAM_BATTING_3B	Triples by batters (3B)	Positive Impact on Wins
TEAM_BATTING_HR	Homeruns by batters (4B)	Positive Impact on Wins
TEAM_BATTING_BB	Walks by batters	Positive Impact on Wins
TEAM_BATTING_HBP	Batters hit by pitch (get a free base)	Positive Impact on Wins
TEAM_BATTING_SO	Strikeouts by batters	Negative Impact on Wins
TEAM_BASERUN_SB	Stolen bases	Positive Impact on Wins
TEAM_BASERUN_CS	Caught stealing	Negative Impact on Wins
TEAM_FIELDING_E	Errors	Negative Impact on Wins
TEAM_FIELDING_DP	Double Plays	Positive Impact on Wins
TEAM_PITCHING_BB	Walks allowed	Negative Impact on Wins
TEAM_PITCHING_H	Hits allowed	Negative Impact on Wins
TEAM_PITCHING_HR	Homeruns allowed	Negative Impact on Wins
TEAM_PITCHING_SO	Strikeouts by pitchers	Positive Impact on Wins

Deliverable:

- A write-up submitted in PDF format. Your write-up should have four sections. Each one is described below. You may assume you are addressing me as a fellow data scientist, so do not need to shy away from technical details.
- Assigned predictions (the number of wins for the team) for the evaluation data set.
- Include your R statistical programming code in an Appendix.

Write Up:

1. **DATA EXPLORATION (25 Points)** Describe the size and the variables in the moneyball training data set. Consider that too much detail will cause a manager to lose interest while too little detail will make the manager consider that you aren't doing your job. Some suggestions are given below. Please do NOT treat this as a check list of things to do to complete the assignment. You should have your own thoughts on what to tell the boss. These are just ideas.

a. Mean / Standard Deviation / Median
b. Bar Chart or Box Plot of the data
c. Is the data correlated to the target variable (or to other variables?)
d. Are any of the variables missing and need to be imputed "fixed"?

2. **DATA PREPARATION (25 Points)** Describe how you have transformed the data by changing the original variables or creating new variables. If you did transform the data or create new variables, discuss why you did this. Here are some possible transformations.

a. Fix missing values (maybe with a Mean or Median value)
b. Create flags to suggest if a variable was missing
c. Transform data by putting it into buckets
d. Mathematical transforms such as log or square root (or use Box-Cox)
e. Combine variables (such as ratios or adding or multiplying) to create new variables

3. **BUILD MODELS (25 Points)** Using the training data set, build at least three different multiple linear regression models, using different variables (or the same variables with different transformations). Since we have not yet covered automated variable selection methods, you should select the variables manually (unless you previously learned Forward or Stepwise selection, etc.). Since you manually selected a variable for inclusion into the model or exclusion into the model, indicate why this was done. Discuss the coefficients in the models, do they make sense? For example, if a team hits a lot of Home Runs, it would be reasonably expected that such a team would win more games. However, if the coefficient is negative (suggesting that the team would lose more games), then that needs to be discussed. Are you keeping the model even though it is counter intuitive? Why? The boss needs to know.
4. **SELECT MODELS (25 Points)** Decide on the criteria for selecting the best multiple linear regression model. Will you select a model with slightly worse performance if it makes more sense or is more parsimonious? Discuss why you selected your model. For the multiple linear regression model, will you use a metric such as Adjusted R2 , RMSE, etc.? Be sure to explain how you can make inferences from the model, discuss multi-collinearity issues (if any), and discuss other relevant model output. Using the training data set, evaluate the multiple linear regression model based on (a) mean squared error, (b) R2, (c) F-statistic, and (d) residual plots. Make predictions using the evaluation data set.

Evaluation

Load the data

```
git_url<-
  "https://raw.githubusercontent.com/melbow2424/Data621_HW1/main/"

df_train <-
  read.csv(paste0(git_url,"moneyball-training-data.csv"))

df_evaluation <-
  read.csv(paste0(git_url,"moneyball-evaluation-data.csv"))
```

Review Data

```
print(skim(df_train))
```

```
## -- Data Summary ------------------------
##                            Values
## Name                       df_train
## Number of rows             2276
## Number of columns          17
## ________________________
## Column type frequency:
##   numeric                  17
## ________________________
## Group variables            None
##
## -- Variable type: numeric ------------------------------------------------------
##    skim_variable    n_missing complete_rate   mean     sd   p0    p25   p50
##  1 INDEX                    0        1      1268.   736.     1  631. 1270.
##  2 TARGET_WINS              0        1        80.8   15.8    0   71     82
##  3 TEAM_BATTING_H           0        1      1469.   145.   891 1383   1454
##  4 TEAM_BATTING_2B          0        1       241.    46.8   69  208    238
##  5 TEAM_BATTING_3B          0        1        55.2   27.9    0   34     47
##  6 TEAM_BATTING_HR          0        1        99.6   60.5    0   42    102
##  7 TEAM_BATTING_BB          0        1       502.   123.     0  451    512
##  8 TEAM_BATTING_SO        102        0.955   736.   249.     0  548    750
##  9 TEAM_BASERUN_SB        131        0.942   125.    87.8    0   66    101
## 10 TEAM_BASERUN_CS        772        0.661    52.8   23.0    0   38     49
## 11 TEAM_BATTING_HBP      2085        0.0839   59.4   13.0   29   50.5    58
## 12 TEAM_PITCHING_H          0        1      1779.  1407.  1137 1419   1518
## 13 TEAM_PITCHING_HR         0        1       106.    61.3    0   50    107
## 14 TEAM_PITCHING_BB         0        1       553.   166.     0  476    536.
## 15 TEAM_PITCHING_SO       102        0.955   818.   553.     0  615    814.
## 16 TEAM_FIELDING_E          0        1       246.   228.    65  127    159
## 17 TEAM_FIELDING_DP       286        0.874   146.    26.2   52  131    149
##      p75  p100 hist
##  1 1916.  2535
##  2   92    146
##  3 1537.  2554
##  4  273    458
##  5   72    223
##  6  147    264
##  7  580    878
##  8  930   1399
##  9  156    697
## 10   62    201
## 11   67     95
## 12 1682. 30132
## 13  150    343
## 14  611   3645
## 15  968  19278
## 16  249.  1898
## 17  164    228
```

```
train_means<-sapply(df_train, function(x) round(mean(x, na.rm = TRUE)))
train_means
```

Get the Means of columns in Data

```
##            INDEX      TARGET_WINS   TEAM_BATTING_H  TEAM_BATTING_2B
##             1268               81             1469              241
##  TEAM_BATTING_3B  TEAM_BATTING_HR  TEAM_BATTING_BB  TEAM_BATTING_SO
##               55              100              502              736
##  TEAM_BASERUN_SB  TEAM_BASERUN_CS TEAM_BATTING_HBP  TEAM_PITCHING_H
##              125               53               59             1779
## TEAM_PITCHING_HR TEAM_PITCHING_BB TEAM_PITCHING_SO  TEAM_FIELDING_E
##              106              553              818              246
## TEAM_FIELDING_DP
##              146
```

```
eval_means<-sapply(df_evaluation, function(x) round(mean(x, na.rm = TRUE)))
eval_means
```

```
##            INDEX   TEAM_BATTING_H  TEAM_BATTING_2B  TEAM_BATTING_3B
##             1264             1469              241               56
##  TEAM_BATTING_HR  TEAM_BATTING_BB  TEAM_BATTING_SO  TEAM_BASERUN_SB
##               96              499              709              124
##  TEAM_BASERUN_CS TEAM_BATTING_HBP  TEAM_PITCHING_H TEAM_PITCHING_HR
##               52               62             1813              102
## TEAM_PITCHING_BB TEAM_PITCHING_SO  TEAM_FIELDING_E TEAM_FIELDING_DP
##              552              800              250              146
```

```
train_medians<-sapply(df_train, function(x) round(median(x, na.rm = TRUE)))
train_medians
```

Get the Medians of columns in data

```
##            INDEX      TARGET_WINS   TEAM_BATTING_H  TEAM_BATTING_2B
##             1270               82             1454              238
##  TEAM_BATTING_3B  TEAM_BATTING_HR  TEAM_BATTING_BB  TEAM_BATTING_SO
##               47              102              512              750
##  TEAM_BASERUN_SB  TEAM_BASERUN_CS TEAM_BATTING_HBP  TEAM_PITCHING_H
##              101               49               58             1518
## TEAM_PITCHING_HR TEAM_PITCHING_BB TEAM_PITCHING_SO  TEAM_FIELDING_E
##              107              536              814              159
## TEAM_FIELDING_DP
##              149
```

```
eval_medians<-sapply(df_evaluation, function(x) round(median(x, na.rm = TRUE)))
eval_medians
```

```
##            INDEX   TEAM_BATTING_H  TEAM_BATTING_2B  TEAM_BATTING_3B
##             1249             1455              239               52
##  TEAM_BATTING_HR  TEAM_BATTING_BB  TEAM_BATTING_SO  TEAM_BASERUN_SB
##              101              509              686               92
##  TEAM_BASERUN_CS TEAM_BATTING_HBP  TEAM_PITCHING_H TEAM_PITCHING_HR
##               50               62             1515              104
## TEAM_PITCHING_BB TEAM_PITCHING_SO  TEAM_FIELDING_E TEAM_FIELDING_DP
##              526              745              163              148
```

```
# Replace NA values in 'column_name' with 'mean'
df_train_mn <- df_train %>%
  mutate(TEAM_BATTING_SO =
           ifelse(is.na(TEAM_BATTING_SO),
                  train_means[8],TEAM_BATTING_SO))%>%
  mutate(TEAM_BASERUN_SB =
           ifelse(is.na(TEAM_BASERUN_SB),
                  train_means[9], TEAM_BASERUN_SB))%>%
  mutate(TEAM_BASERUN_CS =
           ifelse(is.na(TEAM_BASERUN_CS),
                  train_means[10], TEAM_BASERUN_CS))%>%
  mutate(TEAM_BATTING_HBP =
           ifelse(is.na(TEAM_BATTING_HBP),
                  train_means[11],TEAM_BATTING_HBP))%>%
  mutate(TEAM_PITCHING_SO =
           ifelse(is.na(TEAM_PITCHING_SO),
                  train_means[15], TEAM_PITCHING_SO))%>%
  mutate(TEAM_FIELDING_DP =
           ifelse(is.na(TEAM_FIELDING_DP),
                  train_means[17], TEAM_FIELDING_DP))
```

```
df_evaluation_mn <- df_evaluation %>%
  mutate(TEAM_BATTING_SO =
           ifelse(is.na(TEAM_BATTING_SO),
                  eval_means[8],TEAM_BATTING_SO))%>%
  mutate(TEAM_BASERUN_SB =
           ifelse(is.na(TEAM_BASERUN_SB),
                  eval_means[9], TEAM_BASERUN_SB))%>%
  mutate(TEAM_BASERUN_CS =
           ifelse(is.na(TEAM_BASERUN_CS),
                  eval_means[10], TEAM_BASERUN_CS))%>%
  mutate(TEAM_BATTING_HBP =
           ifelse(is.na(TEAM_BATTING_HBP),
                  eval_means[11],TEAM_BATTING_HBP))%>%
  mutate(TEAM_PITCHING_SO =
           ifelse(is.na(TEAM_PITCHING_SO),
                  eval_means[15], TEAM_PITCHING_SO))%>%
  mutate(TEAM_FIELDING_DP =
           ifelse(is.na(TEAM_FIELDING_DP),
                  eval_means[17], TEAM_FIELDING_DP))
```

Replace NA values in columns with their respective Mean

```
# Replace NA values in 'column_name' with 'median'
df_train_md <- df_train %>%
  mutate(TEAM_BATTING_SO =
           ifelse(is.na(TEAM_BATTING_SO),
                  train_medians[8],TEAM_BATTING_SO))%>%
  mutate(TEAM_BASERUN_SB =
           ifelse(is.na(TEAM_BASERUN_SB),
                  train_medians[9], TEAM_BASERUN_SB))%>%
  mutate(TEAM_BASERUN_CS =
           ifelse(is.na(TEAM_BASERUN_CS),
                  train_medians[10], TEAM_BASERUN_CS))%>%
  mutate(TEAM_BATTING_HBP =
           ifelse(is.na(TEAM_BATTING_HBP),
                  train_medians[11],TEAM_BATTING_HBP))%>%
  mutate(TEAM_PITCHING_SO =
           ifelse(is.na(TEAM_PITCHING_SO),
                  train_medians[15], TEAM_PITCHING_SO))%>%
  mutate(TEAM_FIELDING_DP =
           ifelse(is.na(TEAM_FIELDING_DP),
                  train_medians[17], TEAM_FIELDING_DP))
```

```
df_evaluation_md <- df_evaluation %>%
  mutate(TEAM_BATTING_SO =
           ifelse(is.na(TEAM_BATTING_SO),
                  eval_medians[8],TEAM_BATTING_SO))%>%
  mutate(TEAM_BASERUN_SB =
           ifelse(is.na(TEAM_BASERUN_SB),
                  eval_medians[9], TEAM_BASERUN_SB))%>%
  mutate(TEAM_BASERUN_CS =
           ifelse(is.na(TEAM_BASERUN_CS),
                  eval_medians[10], TEAM_BASERUN_CS))%>%
  mutate(TEAM_BATTING_HBP =
           ifelse(is.na(TEAM_BATTING_HBP),
                  eval_medians[11],TEAM_BATTING_HBP))%>%
  mutate(TEAM_PITCHING_SO =
           ifelse(is.na(TEAM_PITCHING_SO),
                  eval_medians[15], TEAM_PITCHING_SO))%>%
  mutate(TEAM_FIELDING_DP =
           ifelse(is.na(TEAM_FIELDING_DP),
                  eval_medians[17], TEAM_FIELDING_DP))
```

Replace NA values with their respective Medians

```
df_train_0 <- df_train %>%
  replace_na(list(
    INDEX = 0,
    TARGET_WINS = 0,
    TEAM_BATTING_H = 0,
    TEAM_BATTING_2B = 0,
    TEAM_BATTING_3B = 0,
    TEAM_BATTING_HR = 0,
    TEAM_BATTING_BB = 0,
    TEAM_BATTING_SO = 0,
    TEAM_BASERUN_SB = 0,
    TEAM_BASERUN_CS = 0,
    TEAM_BATTING_HBP = 0,
    TEAM_PITCHING_H = 0,
    TEAM_PITCHING_HR = 0,
    TEAM_PITCHING_BB = 0,
    TEAM_PITCHING_SO = 0,
    TEAM_FIELDING_E = 0,
    TEAM_FIELDING_DP = 0
  ))
```

```
df_evaluation_0 <- df_evaluation %>%
  replace_na(list(
    INDEX = 0,
    TARGET_WINS = 0,
    TEAM_BATTING_H = 0,
    TEAM_BATTING_2B = 0,
    TEAM_BATTING_3B = 0,
    TEAM_BATTING_HR = 0,
    TEAM_BATTING_BB = 0,
    TEAM_BATTING_SO = 0,
    TEAM_BASERUN_SB = 0,
    TEAM_BASERUN_CS = 0,
    TEAM_BATTING_HBP = 0,
    TEAM_PITCHING_H = 0,
    TEAM_PITCHING_HR = 0,
    TEAM_PITCHING_BB = 0,
    TEAM_PITCHING_SO = 0,
    TEAM_FIELDING_E = 0,
    TEAM_FIELDING_DP = 0
  ))
```

Replace NA values with 0

Remove all rows with NA's

```
df_train_rm<- na.omit(df_train)
df_evaluation_rm<- na.omit(df_evaluation)
```

```
#print(skim(df_train_mn))
```

```
#print(skim(df_train_md))
```

```
#print(skim(df_train_0))
```

Models

```
model_initial <- lm(TARGET_WINS ~ TEAM_BATTING_H+TEAM_BATTING_2B +TEAM_BATTING_3B+TEAM_BATTING_HR+TEAM_
TEAM_BASERUN_SB+ TEAM_BASERUN_CS + TEAM_BATTING_HBP +TEAM_PITCHING_H+ TEAM_PITCHING_HR+TEAM_PITCHING_BB

summary(model_initial)
```

```
##
## Call:
## lm(formula = TARGET_WINS ~ TEAM_BATTING_H + TEAM_BATTING_2B +
##     TEAM_BATTING_3B + TEAM_BATTING_HR + TEAM_BATTING_BB + TEAM_BATTING_SO +
##     TEAM_BASERUN_SB + TEAM_BASERUN_CS + TEAM_BATTING_HBP + TEAM_PITCHING_H +
##     TEAM_PITCHING_HR + TEAM_PITCHING_BB + TEAM_PITCHING_SO +
##     TEAM_FIELDING_E + TEAM_FIELDING_DP, data = df_train)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -19.8708  -5.6564  -0.0599   5.2545  22.9274
##
## Coefficients:
##                   Estimate Std. Error t value Pr(>|t|)
## (Intercept)       60.28826   19.67842   3.064  0.00253 **
## TEAM_BATTING_H     1.91348    2.76139   0.693  0.48927
## TEAM_BATTING_2B    0.02639    0.03029   0.871  0.38484
## TEAM_BATTING_3B   -0.10118    0.07751  -1.305  0.19348
## TEAM_BATTING_HR   -4.84371   10.50851  -0.461  0.64542
## TEAM_BATTING_BB   -4.45969    3.63624  -1.226  0.22167
## TEAM_BATTING_SO    0.34196    2.59876   0.132  0.89546
## TEAM_BASERUN_SB    0.03304    0.02867   1.152  0.25071
## TEAM_BASERUN_CS   -0.01104    0.07143  -0.155  0.87730
## TEAM_BATTING_HBP   0.08247    0.04960   1.663  0.09815 .
## TEAM_PITCHING_H  -1.89096    2.76095  -0.685  0.49432    
## TEAM_PITCHING_HR  4.93043   10.50664   0.469  0.63946    
## TEAM_PITCHING_BB  4.51089    3.63372   1.241  0.21612    
## TEAM_PITCHING_SO -0.37364    2.59705  -0.144  0.88577    
## TEAM_FIELDING_E  -0.17204    0.04140  -4.155 5.08e-05 ***
## TEAM_FIELDING_DP -0.10819    0.03654  -2.961  0.00349 ** 
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## 
## Residual standard error: 8.467 on 175 degrees of freedom
##   (2085 observations deleted due to missingness)
## Multiple R-squared:  0.5501, Adjusted R-squared:  0.5116 
## F-statistic: 14.27 on 15 and 175 DF,  p-value: < 2.2e-16
```

Reference

- "Pythagorean Theorem of Baseball." Baseball Reference, https://www.baseball-reference.com/bullpen/Pythagorean_Theorem_of_Baseball. Accessed 11 September 2023.
- No author listed. "Pythagorean Expectation in Major League Baseball." Digital Commons @ Cal Poly, https://digitalcommons.calpoly.edu/cgi/viewcontent.cgi?article=1067&context=statsp. Accessed 11 September 2023.