Data 621 HW1: MLB Regression Project Write Up

(Group 4) Mohamed Hassan-El Serafi, Chun Shing Leung, Keith Colella, Yina, Qiao, Eddie Xu

2024-09-29

Contents

Introduction

Sport analytics becomes popular in recent decades as it is used to help athletes and teams perform better or predict future outcomes using historical data. For this assignment, our group reviews baseball data sets with records containing professional team and its performance from the year 1871 to 2006.

In this assignment, we first explored the data sets for better understand the data points and address any potential challenges lies ahead. With insights from our data exploration, we leverage that to conduct a data preparation for our data modeling on 3 different models. We evaluated all linear models and determined the best model that offers the best accuracy and completeness.

Data Exploration

Describe the size and the variables in the money ball 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 checklist of things to do to complete the assignment. You should have your own thoughts on what to tell the boss. These are just ideas.

Data Sets

The training data set was used to explore any potential variables that can be used for the test data set. The `INDEX` column was removed because it is not a variable. After the data set was reviewed, there were multiple variables with missing (NA) values, with the `TEAM-BATTING_HBP` having the highest amount. BATTING_HBP variable shows a very close mean and median value. This might be due to less data value to be computed. Out of variables, `PITCHING_H`, `PITCHING_BB`, `PITCHING_SO` and `FIELDING_E` are extremely skewed to the right. Interestingly, `BATTING_BB` is not right skewed, compared to all other variables. There were high correlations between batting and pitching stats that were measuring the same feature. For instance, home runs (HR), Walks (BB), Strikeouts (SO), and Hits (H) for batting and pitching were highly correlated with their respective features. This will be important to consider when selecting features for modeling.

Data Preparation

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.

After examining the density and histogram plots, as well as the correlation matrix for the independent and dependent features, we decided to perform a log transformation of the training data. Before we transformed the data, we noticed that singles (1B) were not represented in the dataset. Therefore, we created a new variable by subtracting the batting columns for home runs, triples, and doubles from the total amount of hits represented by the TEAM_BATTING_H column. Next, we evaluated what the new variable's correlation coefficient was with TARGET_WINS. It had the 4th-highest coefficient with 0.22, behind batting hits (0.39), batting doubles (0.29), and batting walks (0.23). In addition, we noticed that the correlation coefficients had decreased for both the dependent and independent variables. Thus, imputing the data and creating a new variable may have helped decrease multicollinearity in the train data.

Next, we decided to use a log-transformation method to address non-normal distribution and positive skewness that was apparent in some of the independent variables for the training data. Since transforming the variables before imputing missing values helps preserve the relationships between the variables in the regression model, we decided to transform the data first before using our KNN imputation method. After log-transforming the data, there were a considerable amount of infinite values created. To address this, we turned those values into NA values and imputed the NA values. We created density, histogram, and point plots to re-assess how the log transformation changed the distribution for each independent variable, as well as their respective relationships with TARGET_WINS. Variables that previously showed a distinct right skewness (TEAM_PITCHING_BB, TEAM_PITCHING_SO, and TEAM_FIELDING_E) showed improvement in their distribution, with TEAM_PITCHING_BB showing the biggest improvement and TEAM_PITCHING_H not showing much change from its previous right-skewed distribution. Because we log-transformed and imputed the training data, we did the same for the test data to ensure that the model using the log-transformed train data would be able to make accurate predictions with test data that was also log-transformed.

Build Models

Select Models

Appendix

```
knitr::opts_chunk$set(echo = TRUE)

## Data Exploration
mlb_training <- read.csv("https://raw.githubusercontent.com/moham6839/Data_621_HW1/main/moneyball-traini

mlb_test <- read.csv("https://raw.githubusercontent.com/moham6839/Data_621_HW1/main/moneyball-evaluatio

mlb_training <- mlb_training %>%
  dplyr::select(-INDEX)

mlb_test <- mlb_test %>%
  dplyr::select(-INDEX)

## Glimpse and Summary of MLB Training Set
head(mlb_training)

mlb_training %>%
  glimpse() %>%
  kable() %>%
  kable_styling()

mlb_training %>%
  summary() %>%
  kable() %>%
  kable_styling()

head(mlb_training)

### Glimpse and Summary of MLB Test Set
head(mlb_test)

mlb_test %>%
  glimpse() %>%
  kable() %>%
  kable_styling()

mlb_test %>%
  summary() %>%
  kable() %>%
  kable_styling()

head(mlb_test)

mlb_training %>%
  gather(variable, value, TARGET_WINS:TEAM_FIELDING_DP) %>%
  ggplot(., aes(value)) +
  geom_density(fill = "blue", color="blue") +
  facet_wrap(~variable, scales ="free", ncol = 4) +
  labs(x = element_blank(), y = element_blank())


ggplot(gather(mlb_training), aes(value)) +
    geom_histogram(bins = 8) +
    facet_wrap(~key, scales = 'free_x')


mlb_training %>%
  ggplot(aes(TARGET_WINS)) +
  geom_histogram(bins = 50, fill = 'blue', color="black",) +
  geom_vline(aes(xintercept = mean(TARGET_WINS, na.rm = T)), col = "red", lty = 2) +
  geom_vline(aes(xintercept = median(TARGET_WINS, na.rm = T)), col = "yellow", lty = 2) +
  labs(x = element_blank(),
       y = "Count",
       title = "Distribution of Wins",
       caption = "* Red line is the mean value and yellow is the median") +
  theme_classic()

mlb_training %>%
  gather(-TARGET_WINS, key = "var", value = "value") %>%
  ggplot(aes(x = value, y = TARGET_WINS)) +
    facet_wrap(~ var, scales = "free") +
    geom_point(fill = "blue", color="blue") +
    geom_smooth(method = "lm", se = FALSE, color = "black") +
  labs(x = element_blank(), y = "Wins")

ggplot(stack(mlb_training), aes(x = ind, y = values)) +
  geom_boxplot() +
  coord_cartesian(ylim = c(0, 5000))+
  labs(x = element_blank(), y = element_blank()) +
  theme(axis.text.x = element_text(angle = 45, hjust = 1))

## Measuring Correlation of Features

cor_matrix <- mlb_training %>%
  cor(., use = "complete.obs")


ggcorrplot::ggcorrplot(cor_matrix, type = "lower",
          lab = TRUE, lab_size = 2.1, tl.cex = 8)

cor_matrix[lower.tri(cor_matrix, diag=TRUE)] <- ""
cor_matrix <- cor_matrix %>%
  as.data.frame() %>%
  rownames_to_column() %>%
  gather(Variable, Correlation, -rowname) %>%
  filter(Variable != rowname) %>%
  filter(Correlation != "") %>%
  mutate(Correlation = as.numeric(Correlation)) %>%
  rename(` Variable` = rowname) %>%
  arrange(desc(abs(Correlation)))

cor_matrix %>%
  filter(abs(Correlation) > .5) %>%
  kable() %>%
  kable_styling()
```

```r
**Missing Data**

mlb_training %>%
  gather(variable, value) %>%
  filter(is.na(value)) %>%
  group_by(variable) %>%
  tally() %>%
  mutate(percent = n / nrow(mlb_training) * 100) %>%
  mutate(percent = paste0(round(percent, ifelse(percent < 10, 1, 0)), "%")) %>%
  arrange(desc(n)) %>%
  rename(`Variable Missing Data` = variable,
         `Number of Records` = n,
         `Share of Total` = percent) %>%
  kable(caption="<center>Missing Training Data Count and Percentage", align = "c") %>%
  kable_styling(latex_options="scale_down", c("striped", "hover", "condensed", full_width=F))

mlb_test %>%
  gather(variable, value) %>%
  filter(is.na(value)) %>%
  group_by(variable) %>%
  tally() %>%
  mutate(percent = n / nrow(mlb_test) * 100) %>%
  mutate(percent = paste0(round(percent, ifelse(percent < 10, 1, 0)), "%")) %>%
  arrange(desc(n)) %>%
  rename(`Variable Missing Data` = variable,
         `Number of Records` = n,
         `Share of Total` = percent) %>%
  kable(caption="<center>Missing Test Data Count and Percentage", align = "c") %>%
  kable_styling(latex_options="scale_down", c("striped", "hover", "condensed", full_width=F))

### Flaws of Imputing `TEAM_BATTING_HBP`

set.seed(123)
mlb_train_imp2 <- mlb_training %>%
  kNN(variable = c("TEAM_BASERUN_CS", "TEAM_FIELDING_DP", "TEAM_BASERUN_SB", "TEAM_BATTING_SO", "TEAM_P
      k = 5, numFun = weighted.mean, weightDist = TRUE, imp_var = FALSE)

set.seed(123)
mlb_train_imp2 <- mlb_train_imp2 %>%
  dplyr::mutate(TEAM_BATTING_1B = TEAM_BATTING_H - dplyr::select(., TEAM_BATTING_2B:TEAM_BATTING_HR) %>
  dplyr::mutate(TEAM_BATTING_AB = TEAM_BATTING_H + TEAM_PITCHING_BB + TEAM_BATTING_SO + TEAM_BATTING_HB
  dplyr::mutate(TEAM_BATTING_AVERAGE = TEAM_BATTING_H/TEAM_BATTING_AB) %>%
  dplyr::mutate(TEAM_BATTING_OBP = (TEAM_BATTING_H + TEAM_BATTING_BB + TEAM_BATTING_HBP)/(TEAM_BATTING_
  dplyr::mutate(TEAM_BATTING_SLG = (TEAM_BATTING_1B + 2*TEAM_BATTING_2B + 3*TEAM_BATTING_3B + 4*TEAM_BAT
  relocate(TEAM_BATTING_1B, .before = TEAM_BATTING_2B)

head(mlb_train_imp2)

max(mlb_train_imp2$TEAM_BATTING_AVERAGE)

### Dropping `TEAM_BATTING_HBP` and Imputing Missing Values

## Training
```

```r
set.seed(123)
mlb_training_no_hbp <- mlb_training %>%
  dplyr::select(-TEAM_BATTING_HBP)

set.seed(123)
mlb_train_imp <- mlb_training_no_hbp %>%
  kNN(variable = c("TEAM_BASERUN_CS", "TEAM_FIELDING_DP", "TEAM_BASERUN_SB", "TEAM_BATTING_SO", "TEAM_P
      k = 5, numFun = weighted.mean, weightDist = TRUE, imp_var = FALSE)

head(mlb_train_imp)

sum(is.na(mlb_train_imp))
sum(is.nan(as.matrix(mlb_train_imp)))
sum(is.infinite(as.matrix(mlb_train_imp)))

### Test Set
set.seed(123)
mlb_test_no_hbp <- mlb_test %>%
  dplyr::select(-TEAM_BATTING_HBP)

set.seed(123)
mlb_test_imp <- mlb_test_no_hbp %>%
  kNN(variable = c("TEAM_BASERUN_CS", "TEAM_FIELDING_DP", "TEAM_BASERUN_SB", "TEAM_BATTING_SO", "TEAM_P
      k = 5, numFun = weighted.mean, weightDist = TRUE, imp_var = FALSE)

head(mlb_test_imp)

sum(is.na(mlb_test_imp))
sum(is.nan(as.matrix(mlb_test_imp)))
sum(is.infinite(as.matrix(mlb_test_imp)))

## Data Preparation

mlb_train_imp <- mlb_train_imp %>%
  dplyr::mutate(TEAM_BATTING_1B = TEAM_BATTING_H - dplyr::select(., TEAM_BATTING_2B:TEAM_BATTING_HR) %>
  relocate(TEAM_BATTING_1B, .before = TEAM_BATTING_2B)

mlb_train_imp %>%
  ggplot(aes(TEAM_BATTING_1B)) +
  geom_histogram(bins = 50, fill = 'blue', color="black",) +
  geom_vline(aes(xintercept = mean(TEAM_BATTING_1B, na.rm = T)), col = "red", lty = 2) +
  geom_vline(aes(xintercept = median(TEAM_BATTING_1B, na.rm = T)), col = "yellow", lty = 2) +
  labs(x = element_blank(),
       y = "Count",
       title = "Distribution of Singles",
       caption = "* Red line is the mean value and yellow is the median") +
  theme_classic()

cor_matrix2 <- mlb_train_imp %>%
  cor(., use = "complete.obs")

ggcorrplot::ggcorrplot(cor_matrix2, type = "lower",
          lab = TRUE, lab_size = 2.1, tl.cex = 8)
```

```
### Log-Transforming Variables

### Training Set

mlb_training_no_hbp <- mlb_training_no_hbp %>%
  dplyr::mutate(TEAM_BATTING_1B = TEAM_BATTING_H - dplyr::select(., TEAM_BATTING_2B:TEAM_BATTING_HR) %>
  relocate(TEAM_BATTING_1B, .before = TEAM_BATTING_2B)

mlb_train_log <- log(mlb_training_no_hbp)

set.seed(123)
mlb_train_log <- mlb_train_log%>%
  kNN(variable = c("TEAM_BASERUN_CS", "TEAM_FIELDING_DP", "TEAM_BASERUN_SB", "TEAM_BATTING_SO", "TEAM_P
      k = 5, numFun = weighted.mean, weightDist = TRUE, imp_var = FALSE)

sum(is.na(mlb_train_log))
sum(is.nan(as.matrix(mlb_train_log)))
sum(is.infinite(as.matrix(mlb_train_log)))

mlb_train_log[sapply(mlb_train_log, is.infinite)] <- NA

set.seed(123)
mlb_train_log <- mlb_train_log %>%
  kNN(k = 5, numFun = weighted.mean, weightDist = TRUE, imp_var = FALSE)

mlb_train_log %>%
  gather(-TARGET_WINS, key = "var", value = "value") %>%
  ggplot(aes(x = value, y = TARGET_WINS)) +
    facet_wrap(~ var, scales = "free") +
    geom_point(fill = "blue", color="blue") +
    geom_smooth(method = "lm", se = FALSE, color = "black") +
  labs(x = element_blank(), y = "Wins")

ggplot(gather(mlb_train_log), aes(value)) +
    geom_histogram(bins = 8) +
    facet_wrap(~key, scales = 'free_x')

mlb_train_log %>%
  gather(variable, value, TARGET_WINS:TEAM_FIELDING_DP) %>%
  ggplot(., aes(value)) +
  geom_density(fill = "blue", color="blue") +
  facet_wrap(~variable, scales ="free", ncol = 4) +
  labs(x = element_blank(), y = element_blank())

**Test Set**

mlb_test_no_hbp <- mlb_test_no_hbp %>%
  dplyr::mutate(TEAM_BATTING_1B = TEAM_BATTING_H - dplyr::select(., TEAM_BATTING_2B:TEAM_BATTING_HR) %>
  relocate(TEAM_BATTING_1B, .before = TEAM_BATTING_2B)

mlb_test_log <- log(mlb_test_no_hbp)

set.seed(123)
```

```
mlb_test_log <- mlb_test_log%>%
  kNN(variable = c("TEAM_BASERUN_CS", "TEAM_FIELDING_DP", "TEAM_BASERUN_SB", "TEAM_BATTING_SO", "TEAM_P
      k = 5, numFun = weighted.mean, weightDist = TRUE, imp_var = FALSE)

sum(is.na(mlb_test_log))
sum(is.nan(as.matrix(mlb_test_log)))
sum(is.infinite(as.matrix(mlb_test_log)))

mlb_test_log[sapply(mlb_test_log, is.infinite)] <- NA

set.seed(123)
mlb_test_log <- mlb_test_log %>%
  kNN(k = 5, numFun = weighted.mean, weightDist = TRUE, imp_var = FALSE)

sum(is.na(mlb_test_log))
sum(is.nan(as.matrix(mlb_test_log)))
sum(is.infinite(as.matrix(mlb_test_log)))

ggplot(gather(mlb_test_log), aes(value)) +
    geom_histogram(bins = 10) +
    facet_wrap(~key, scales = 'free_x')

## Model Building

### Model 1 - Using findCorrelation function from caret package

set.seed(123)
highlyCorDescr <- findCorrelation(cor(mlb_train_imp), cutoff = .50, verbose = TRUE, names=TRUE)

set.seed(123)
keep_these <- names(mlb_train_imp)[!(names(mlb_train_imp) %in% colnames(mlb_train_imp)[highlyCorDescr])
mlb_train_features <- mlb_train_imp[, keep_these]

head(mlb_train_features)

set.seed(123)
m1 <- lm(TARGET_WINS ~., data = mlb_train_features)
summary(m1)

set.seed(123)
m1_train_revised <- lm(TARGET_WINS ~ TEAM_BATTING_2B + TEAM_BATTING_BB + TEAM_BASERUN_SB + TEAM_PITCHIN
summary(m1_train_revised)

ggplot(m1_train_revised, aes(x = .fitted, y = .resid)) +
  geom_point() +
  geom_hline(yintercept = 0, linetype = "dashed") +
  labs(title="Residual vs. Fitted Values Plot") +
  xlab("Fitted values") +
  ylab("Residuals")

ggplot(data = m1_train_revised, aes(x = m1_train_revised$residuals)) +
    geom_histogram(bins = 10, fill = 'steelblue', color = 'black') +
    labs(title = 'Histogram of Residuals', x = 'Residuals', y = 'Frequency')
```

```
ggplot(data = m1_train_revised, aes(x = .resid)) +
  geom_histogram(binwidth = 0.4) +
  xlab("Residuals")

qqnorm(resid(m1_train_revised))
qqline(resid(m1_train_revised))

### Model 2 - Using findCorrelation function for Log-Transformed Train Set

set.seed(123)
highlyCorDescr2 <- findCorrelation(cor(mlb_train_log), cutoff = .50, verbose = TRUE)

set.seed(123)
keep_these2 <- names(mlb_train_log)[!(names(mlb_train_log) %in% colnames(mlb_train_log)[highlyCorDescr2
mlb_train_features2 <- mlb_train_log[, keep_these2]

head(mlb_train_features2)

set.seed(123)
m2_log <- lm(TARGET_WINS ~., data = mlb_train_features2)
summary(m2_log)

set.seed(123)
m2_log2 <- lm(TARGET_WINS ~ TEAM_BASERUN_SB + TEAM_BATTING_2B + TEAM_PITCHING_H + TEAM_PITCHING_BB + TE
summary(m2_log2)

ggplot(m2_log2, aes(x = .fitted, y = .resid)) +
  geom_point() +
  geom_hline(yintercept = 0, linetype = "dashed") +
  labs(title="Residual vs. Fitted Values Plot") +
  xlab("Fitted values") +
  ylab("Residuals")

ggplot(data = m2_log2, aes(x = .resid)) +
  geom_histogram(binwidth = 0.04) +
  xlab("Residuals")

qqnorm(resid(m2_log2))
qqline(resid(m2_log2))

### Model 3 - Using Recursive Feature Elimination on Imputed Train Set

set.seed(123)
filterCtrl <- rfeControl(functions=rfFuncs, method="cv", number=3)
results <- rfe(x= mlb_train_imp[,2:16],y= mlb_train_imp[,1], sizes=c(2:16), rfeControl=filterCtrl)
results

set.seed(123)
m3_train <- lm(TARGET_WINS ~ TEAM_FIELDING_E + TEAM_BATTING_H + TEAM_BASERUN_CS + TEAM_BATTING_BB + TEAM
summary(m3_train)

ggplot(m3_train, aes(x = .fitted, y = .resid)) +
  geom_point() +
  geom_hline(yintercept = 0, linetype = "dashed") +
  labs(title="Residual vs. Fitted Values Plot") +
  xlab("Fitted values") +
  ylab("Residuals")

ggplot(data = m3_train, aes(x = m3_train$residuals)) +
    geom_histogram(bins = 10, fill = 'steelblue', color = 'black') +
    labs(title = 'Histogram of Residuals', x = 'Residuals', y = 'Frequency')

ggplot(data = m3_train, aes(x = .resid)) +
  geom_histogram(binwidth = 0.4) +
  xlab("Residuals")

qqnorm(resid(m3_train))
qqline(resid(m3_train))

### Model 4 - Using Recursive Feature Elimination for Log-Transformed Imputed Train Set

set.seed(123)
filterCtrl2 <- rfeControl(functions=rfFuncs, method="cv", number=3)
results2 <- rfe(x= mlb_train_log[,2:16],y= mlb_train_log[,1], sizes=c(2:16), rfeControl=filterCtrl2)
results2

set.seed(123)
m4_log <- lm(TARGET_WINS ~ TEAM_FIELDING_E + TEAM_BATTING_H + TEAM_BASERUN_SB + TEAM_PITCHING_SO + TEAM_
summary(m4_log)

set.seed(123)
vip(m4_log)

ggplot(m4_log, aes(x = .fitted, y = .resid)) +
  geom_point() +
  geom_hline(yintercept = 0, linetype = "dashed") +
  labs(title="Residual vs. Fitted Values Plot") +
  xlab("Fitted values") +
  ylab("Residuals")

ggplot(data = m4_log, aes(x = m4_log$residuals)) +
    geom_histogram(bins = 10, fill = 'steelblue', color = 'black') +
    labs(title = 'Histogram of Residuals', x = 'Residuals', y = 'Frequency')

ggplot(data = m4_log, aes(x = .resid)) +
  geom_histogram(binwidth = 0.2) +
  xlab("Residuals")

qqnorm(resid(m4_log))
qqline(resid(m4_log))

## Model Selection
set.seed(123)
compare_performance(m1_train_revised, m2_log2, m3_train, m4_log)

set.seed(123)
```

```r
predictions <- predict(m4_log, newdata = mlb_test_log)
summary(predictions)

set.seed(123)
mlb_test_log$predicted_wins <- round(exp(predictions))

set.seed(123)
mlb_test_log_final <- mlb_test_log %>%
  relocate(predicted_wins)

head(mlb_test_log_final)
```

References

- Masterindatascience Link
- Stefvanbuuren Link
- Analyticsvidhya Link
- Datacamp Link
- Statmuse Link
- JTF13 Github Link
- Sthda Link
- Bookdown Link