Capstone Modeling

Danielle Contreras

2023-03-26

R Markdown

```r
data <- read.csv("C:\\Users\\danie\\Downloads\\472\\imdb_data_22.csv")
data <- na.omit(data)
data$Director <- as.factor(data$Director)
data$Lead <- as.factor(data$Lead)
```

```r
#new_data <- read.csv("C:\\Users\\danie\\Downloads\\test_data - newmovie_data.csv")
#new_data <- new_data %>% mutate(
     #Rating_range = case_when(
         #Rating >= 4 & Rating < 6 ~ 1,
         #Rating >= 6 & Rating < 8 ~ 2,
         #Rating >= 8 & Rating < 10 ~ 3,
         #TRUE ~ 0
     #)
#)
#new_data$Rating_range <- as.factor(new_data$Rating_range)
#new_data$Sequel <- as.factor(new_data$Sequel)

#cols <- c(12:31)
#new_data <- new_data %>%
       #mutate_each_(funs(factor(.)),cols)
#modeling_data_new <- new_data %>%
 # select(-Lead, -Director, -X, -Title, -Rating, -Votes)

#names(modeling_data_new)[names(modeling_data_new) == 'Dir_avg'] <- 'Avg_rating_director'
#names(modeling_data_new)[names(modeling_data_new) == 'Lead_avg'] <- 'Avg_rating_lead'


#set.seed(472)
#fit.knn_n <- train(Rating_range~., data=modeling_data_new, method="knn",
                 #metric=metric ,trControl=trainControl)
#prediction <- predict(fit.knn_n, newdata = modeling_data_new)
#cf <- confusionMatrix(prediction, modeling_data_new$Rating_range)
#print(cf)
#fit.knn.k1 <- knn(train=modeling_data3, test=modeling_data_new, cl=modeling_data_new$Rating_range, k=k
#cf <- confusionMatrix(modeling_data_new$Rating_range,fit.knn.k1)

#plot.df <- data.frame(modeling_data_new, predicted = fit.knn.k1)
#plot.df1 <- data.frame(x = plot.df$Runtime,
                      #y = plot.df$Budget,
                      #predicted = plot.df$predicted)

#find_hull <-  function(df) df[chull(plot.df1$x, plot.df$y), ]
#boundary <- ddply(plot.df1, .variables = "predicted", .fun = find_hull)


#ggplot(plot.df, aes(x=Avg_rating_director,y=Avg_rating_lead , color = predicted, fill = predicted)) +
  #geom_point(size = 5) +
  #geom_polygon(data = boundary, aes(x,y), alpha = 0.5)  +
   # scale_fill_manual(values=c("#56B4E9", "#D55E00", "#F0E442")) +
   # scale_color_manual(values=c("#56B4E9", "#D55E00", "#F0E442"))
```

```
#new_data_2 <- new_data %>% mutate(
     #Rating_range = case_when(
         #Rating > 4 & Rating <= 6 ~ 1,
         #Rating > 6 & Rating <= 8 ~ 2,
         #Rating > 8 & Rating <= 10 ~ 3,
         #TRUE ~ 0
      #)
#)
#new_data_2 <- new_data_2 %>% mutate(
     #Avg_rating_director = case_when(
     #Dir_avg > 4 & Dir_avg <= 6 ~ 1,
        #Dir_avg > 6 & Dir_avg <= 8 ~ 2,
         #Dir_avg > 8 & Dir_avg <= 10 ~ 3,
         #TRUE ~ 0
      #)
   #)

#new_data_2 <- new_data_2 %>% mutate(
     #Avg_rating_lead = case_when(
     #Lead_avg > 4 & Lead_avg <= 6 ~ 1,
        #Lead_avg > 6 & Lead_avg <= 8 ~ 2,
         #Lead_avg > 8 & Lead_avg <= 10 ~ 3,
         #TRUE ~ 0
      #)
   #)
#new_data_2$Avg_rating_director <- as.factor(new_data_2$Avg_rating_director)
#new_data_2$Avg_rating_lead <- as.factor(new_data_2$Avg_rating_lead)


#names(modeling_data3)[names(modeling_data3) == 'Avg_rating_director'] <- 'Dir_avg'
#names(modeling_data3)[names(modeling_data3) == 'Avg_rating_lead']<- 'Lead_avg'
#modeling_data3 <- modeling_data3 %>%mutate(
     #Avg_rating_director = case_when(
     #Dir_avg > 4 & Dir_avg <= 6 ~ 1,
       #Dir_avg > 6 & Dir_avg <= 8 ~ 2,
         #Dir_avg > 8 & Dir_avg<= 10 ~ 3,
         #TRUE ~ 0
      #)
   #)
```

```
#modeling_data3 <- modeling_data3 %>%mutate(
     #Avg_rating_lead = case_when(
     #Lead_avg > 4 & Lead_avg <= 6 ~ 1,
       #Lead_avg > 6 & Lead_avg <= 8 ~ 2,
         #Lead_avg > 8 & Lead_avg<= 10 ~ 3,
         #TRUE ~ 0
      #)
  # )

#modeling_data3 <- modeling_data3 %>%
  #select(-Dir_avg, -Lead_avg)
#names(modeling_data3)[names(modeling_data3) == 'Dir_avg.1'] <- 'Avg_rating_director'




#new_data_2$Rating_range <- as.factor(new_data_2$Rating_range)
#new_data_2$Sequel <- as.factor(new_data_2$Sequel)
#modeling_data3$Avg_rating_director <- as.factor(modeling_data3$Avg_rating_director)
#modeling_data3$Avg_rating_lead <- as.factor(modeling_data3$Avg_rating_lead)


#cols <- c(12:31)
#new_data_2 <- new_data_2 %>%
       #mutate_each_(funs(factor(.)),cols)
#modeling_data_new2 <- new_data_2 %>%
  #select(-Lead, -Director, -X, -Title, -Rating, -Votes, -Dir_avg, -Lead_avg)



#set.seed(472)


#trainControl <- trainControl(method="repeatedcv", number=10, repeats=3)
#metric <- "Accuracy"
#fit.knn <- train(Rating_range~., data=modeling_data3, method="knn",
                 #metric=metric ,trControl=trainControl)
#knn.k1 <- fit.knn$bestTune
#print(fit.knn)
#plot(fit.knn)
#ggplot(fit.knn, aes(x=k, y=Accuracy)) + geom_point() + labs(x="# Neighbors", y="Accuracy (Repeated Cro


#prediction <- predict(fit.knn, newdata = modeling_data_new2)
#cf <- confusionMatrix(prediction, modeling_data_new2$Rating_range)
#cf_sep <- cf$overall

#cf_sep %>% kable(booktabs=TRUE, digits=3, caption="Overall Confusion Matrix Results") %>% kable_stylin

#cf_sep2 <- cf$table
#cf_sep2 <- as.data.frame(cf_sep2)
#cf_sep2 <- cf_sep2 %>%
  #filter(!Freq == 0)
```

```
#cf_sep2 %>% kable(booktabs=TRUE, digits=3, caption="Predicted vs. Actual Confusion Matrix Results") %>



#fit.knn.k1 <- knn(train=modeling_data3, test=modeling_data_new2, cl=modeling_data3$Rating_range, k=knn

#cf <- confusionMatrix(modeling_data_new2$Rating_range,fit.knn.k1)
#cf_sep <- cf$overall

#cf_sep %>% kable(booktabs=TRUE, digits=3, caption="Overall Confusion Matrix Results") %>% kable_stylin

#cf_sep2 <- cf$table
#cf_sep2 <- as.data.frame(cf_sep2)
#cf_sep2 <- cf_sep2 %>%
  #filter(!Freq == 0)
#cf_sep2 %>% kable(booktabs=TRUE, digits=3, caption="Predicted vs. Actual Confusion Matrix Results") %>


#set.seed(472)

#plot.df <- data.frame(modeling_data_new2, predicted = fit.knn.k1)
#plot.df1 <- data.frame(x = plot.df$Avg_rating_director,
                      #y = plot.df$Avg_rating_lead,
                      #predicted = plot.df$predicted)

#find_hull <-  function(df) df[chull(plot.df1$x, plot.df$y), ]
#boundary <- ddply(plot.df1, .variables = "predicted", .fun = find_hull)


#ggplot(plot.df, aes(x=Avg_rating_director,y=Avg_rating_lead , color = predicted, fill = predicted)) +
  #geom_point(size = 5) +
  #geom_polygon(data = boundary, aes(x,y), alpha = 0.5)  +
    #scale_fill_manual(values=c("#56B4E9", "#D55E00", "#F0E442")) +
    #scale_color_manual(values=c("#56B4E9", "#D55E00", "#F0E442")) + labs(x="Average Rating for Directo


#plot.df.2 <- data.frame(modeling_data_new2, predicted = fit.knn.k1)
#plot.df2 <- data.frame(x = plot.df.2$Budget,
                      #y = plot.df.2$Runtime,
                      #predicted = plot.df.2$predicted)

#find_hull_2 <-  function(df) df[chull(plot.df2$x, plot.df2$y), ]
#boundary2 <- ddply(plot.df2, .variables = "predicted", .fun = find_hull_2)


#ggplot(plot.df.2, aes(x=Budget,y=Runtime , color = predicted, fill = predicted)) +
  #geom_point(size = 5) +
  #geom_polygon(data = boundary2, aes(x,y), alpha = 0.5)  +
    #scale_fill_manual(values=c("#56B4E9", "#D55E00", "#F0E442")) +
    #scale_color_manual(values=c("#56B4E9", "#D55E00", "#F0E442")) + labs(x="Budget", y= "Runtime(minut
```

```r
data <- read.csv("C:\\Users\\danie\\Downloads\\472\\imdb_data_22.csv")
new_data <- read.csv("C:\\Users\\danie\\Downloads\\test_data - newmovie_data.csv")
new_data_2 <- read.csv("C:\\Users\\danie\\Downloads\\new_movies_apr18 - Sheet1.csv")
data <- na.omit(data)

data$Director <- as.factor(data$Director)
data$Lead <- as.factor(data$Lead)









data <- data %>%
 mutate(across(30:31, round, 1))
```

```
## Warning: There was 1 warning in ‘mutate()‘.
## i In argument: ‘across(30:31, round, 1)‘.
## Caused by warning:
## ! The ‘...‘ argument of ‘across()‘ is deprecated as of dplyr 1.1.0.
## Supply arguments directly to ‘.fns‘ through an anonymous function instead.
##
##   # Previously
##   across(a:b, mean, na.rm = TRUE)
##
##   # Now
##   across(a:b, \(x) mean(x, na.rm = TRUE))
```

```r
data <- data %>%
  mutate_at(4:23, as.factor) %>%
  mutate_at(29, as.factor) %>%
  mutate(Budget = Budget/1000000)

data <- data %>% group_by(Director) %>% filter(n() >= 3) %>% ungroup()
#write.csv(data, "mo_three.csv")


modeling_data_nume <- data %>%
  select(Runtime, Rating, Budget, Sequel, Avg_rating_director, Avg_rating_lead)

cols <- c(12:31)
new_data <- new_data %>%
  mutate(Budget = Budget/1000000)

names(new_data)[names(new_data) == 'Dir_avg'] <- 'Avg_rating_director'
names(new_data)[names(new_data) == 'Lead_avg'] <- 'Avg_rating_lead'



test_data <- new_data %>%
  select(Title, Rating, Runtime, Budget, Avg_rating_director, Avg_rating_lead)
```

```
modeling_data_nume <- modeling_data_nume %>% mutate_if(is.numeric, round, 1)
test_data <- test_data %>% mutate_if(is.numeric, round, 1)

#colnames(new_data_2) <- c("Title", "Avg_rating_director", "Avg_rating_lead", "Budget", "Runtime", "Rat
#new_data_2 <- na.omit(new_data_2)

#new_data_2 <- new_data_2 %>% mutate_if(is.numeric, round, 1)
#new_data_2 <- new_data_2 %>%
  #slice(4)

#test_data <- rbind(new_data_2, test_data)
```

```
set.seed(472)
library(FNN)
```

```
## Warning: package ’FNN’ was built under R version 4.2.3
```

```
##
## Attaching package: ’FNN’
```

```
## The following objects are masked from ’package:class’:
##
##     knn, knn.cv
```

```
library(fastDummies)
```

```
## Warning: package ’fastDummies’ was built under R version 4.2.3
```

```
data_reg <- modeling_data_nume
rating_outcome <- data_reg %>% select(Rating)
rating_test <- test_data %>% select(Rating)


test_data_new <- test_data %>% select(-Title, -Rating)
data_reg <- data_reg %>% select(Runtime, Budget, Avg_rating_director, Avg_rating_lead)





i=1                          # declaration to initiate for loop
k.optm=1                     # declaration to initiate for loop
for (i in 1:28){
    knn.mod <-  knn.reg(data_reg, test_data_new, rating_outcome$Rating, k=i)
    k.optm[i] <- sqrt(mean((knn.mod$pred - rating_test$Rating)^2))
    k=i
    #cat(k,'=',k.optm[i],'\n')       # to print % accuracy
}

reps <- c(1:28)
reps <- as.data.frame(reps)
k_opt <- as.data.frame(k.optm)
```

```r
k_values <- cbind(reps, k_opt)
colnames(k_values) <- c("K", "RMSE")

k_values <- round(k_values, digits=3)
k_values_sort <- k_values %>%
  arrange(RMSE)
kt <- function(data) {
  knitr::kable(data, digits=3,linesep='',booktabs=TRUE, caption="RMSE for K's") %>% kable_styling(boots
}
#head(k_values_sort) %>% kt
reg_results <- knn.reg(data_reg, test_data_new, rating_outcome$Rating, k = 5)
rmse_test <- sqrt(mean((reg_results$pred - rating_test$Rating)^2))
#cat('Testing RMSE:' , rmse_test)


all_errors <- as.data.frame((reg_results$pred - rating_test$Rating)^2)

#ggplot(all_errors, aes(`(reg_results$pred - rating_test$Rating)^2`)) + geom_histogram()

reg_results_pred <- reg_results$pred
reg_results_pred <- as.data.frame(reg_results_pred)

reg_results_pred <- round(reg_results_pred, 1)
pred_actual <- cbind(reg_results_pred, test_data$Rating)



colnames(pred_actual) <- c("Prediction", "Rating")
#ggplot(pred_actual, aes(x=Prediction, y=Rating)) + geom_point() + labs(x= "Predicted Ratings", "Rating
  #stat_smooth(method = "lm",
              #formula = y ~ x,
              #geom = "smooth")


pred_act_table <- test_data %>%
  select(Title, Rating) %>%
  left_join(pred_actual, by="Rating") %>%
  distinct() %>%
  slice(-2,-6,-9,-11)
```

```
## Warning in left_join(., pred_actual, by = "Rating"): Detected an unexpected many-to-many relationshi
## i Row 1 of ‘x‘ matches multiple rows in ‘y‘.
## i Row 1 of ‘y‘ matches multiple rows in ‘x‘.
## i If a many-to-many relationship is expected, set ‘relationship =
##   "many-to-many"‘ to silence this warning.
```

```r
kt <- function(pred_act_table) {
  knitr::kable(pred_act_table, digits=3,linesep='',booktabs=TRUE, caption="KNN IMDB Ratings Predictions
}
#pred_act_table %>% kt
```

```
pred_act_table <- pred_act_table %>%
  slice(-2,-6,-9,-11)




kt <- function(pred_act_table) {
  knitr::kable(pred_act_table, digits=3,linesep='',booktabs=TRUE, caption = "IMDB KNN Ratings") %>% kabl
}

#pred_act_table %>% kt()
```

```
options(scipen = 999)
data_eda <- read.csv("C:\\Users\\danie\\Downloads\\472\\imdb_data_22.csv")

data_eda_fil <- data_eda %>%
  mutate_at(4:23, as.factor) %>%
  mutate_at(29, as.factor) %>%
  select(Runtime, 4:23, Rating, Budget, Sequel, Avg_rating_director, Avg_rating_lead)

#ggplot(data_eda, aes(x=Runtime, y=Rating)) + geom_point() + labs(x="Runtime(minutes)", y="Rating", tit
             # formula = y ~ x,
              #geom = "smooth") + theme(axis.text = element_text(size = 15))+ theme(axis.title=element_

#ggplot(data_eda, aes(x=Budget, y=Rating)) + geom_point() + labs(x="Budget", y="Rating", title = "Budge
              #formula = y ~ x,
              #geom = "smooth") + theme(axis.text = element_text(size = 15))+ theme(axis.title=element_

#ggplot(data_eda, aes(x=Avg_rating_director, y=Rating)) + geom_point() + labs(x="Average Rating per Dir
              #formula = y ~ x,
              #geom = "smooth") + theme(axis.text = element_text(size = 15))+ theme(axis.title=element_

#ggplot(data_eda, aes(x=Avg_rating_lead, y=Rating)) + geom_point() + labs(x="Average Rating per Lead", 
              #formula = y ~ x,
              #geom = "smooth") + theme(axis.text = element_text(size = 15))+ theme(axis.title=element_


#ggplot(data_eda, aes(x=Rating, y=Sequel))+ geom_point() + labs(x="Rating", y="Sequel", title = "Sequel
#geom_smooth(method = "glm",
#method.args = list(family = "binomial")) + theme(axis.text = element_text(size = 15))+ theme(axis.titl


data_genres <- data_eda %>%
  select(4:23)

data_genres_t <- as.data.frame(t(data_genres))

genre_counts <- as.table(rowSums(data_genres_t))

genre_counts <- as.data.frame(genre_counts)

colnames(genre_counts) <- c("Genre", "Count")
```

```
genre_counts_table <- genre_counts %>%
  arrange(desc(Count))

genre_tab1 <- genre_counts_table %>%
  slice(1:10)
genre_tab2 <- genre_counts_table %>%
  slice(11:20)

kt <- function(genre_counts_table) {
  knitr::kable(genre_counts_table, digits=3,linesep='',booktabs=TRUE, caption="Genres by Count") %>% ka
}
#genre_tab1 %>% kt
#genre_tab2 %>% kt
```