Saivee Phatak
OPSM324
Prof. Deepak Srivastav
30 April 2024

OPSM324- Box office Analysis Final Code

```
# Clean up the environment
rm(list=ls())

#install libraries
library(Ecdat)
library(ggplot2)
library(ggcorrplot)
library(GGally)
library(dplyr)
library(caret)
library(lmtest)
library(caret)
library(rpart)
library(rpart.plot)
library(car)

#get csv file
setwd("/Users/saiveephatak/Desktop")
getwd()
moviedata <- read.csv("Box_office.csv", stringsAsFactors = TRUE)

str(moviedata)

#Q1 a. Best season to release a movie. Explain the possible reason behind the same
summary_data <- moviedata %>%
  group_by(Release_Date, S_F) %>%
  summarise(count = n()) %>%
  mutate(prop = count / sum(count))

# Now plot the summarized data
ggplot(summary_data, aes(x = Release_Date, y = prop, fill = factor(S_F))) +
  geom_bar(stat = "identity", position = "stack") +
  labs(x = "Release Date", y = "Proportion", title = "Proportion of Success/Failure vs. Release
Date",
       fill = "Success/Failure") +
  scale_fill_manual(values = c("blue", "red")) +  # Adjust fill colors if needed
  theme(axis.text.x = element_text(angle = 45, hjust = 1))
```

```
moviedata %>%
  group_by(Release_Date) %>%
  summarise(average_Box_Office_Collection = mean(Box_Office_Collection)) %>%
  ggplot(aes(x = Release_Date, y = average_Box_Office_Collection)) +
  geom_bar(stat = "identity", fill = "blue") +
  labs(x = "Release Date", y = "Box Office Collection", title = "Success of a movie vs. Box Office
Collection") +
  theme(axis.text.x = element_text(angle = 45, hjust = 1))

#Q1 b. Does an item song make a difference in the budget/box office collection?
# Summary statistics
summary_stats <- moviedata %>%
  group_by(Item_Song) %>%
  summarise(
    mean_box_office = mean(Box_Office_Collection),
    median_box_office = median(Box_Office_Collection),
    sd_box_office = sd(Box_Office_Collection),
    min_box_office = min(Box_Office_Collection),
    max_box_office = max(Box_Office_Collection)
  )

print(summary_stats)

ggplot(moviedata, aes(x = factor(Item_Song), y = Box_Office_Collection)) +
  geom_boxplot() +
  labs(x = "Item Song", y = "Box Office Collection", title = "Box Office Collection vs. Item Song")

summary_stats_1 <- moviedata %>%
  group_by(Item_Song) %>%
  summarise(
    mean_box_office = mean(Budget),
    median_box_office = median(Budget),
    sd_box_office = sd(Budget),
    min_box_office = min(Budget),
    max_box_office = max(Budget)
  )

print(summary_stats_1)

ggplot(moviedata, aes(x = factor(Item_Song), y = Budget)) +
  geom_boxplot() +
  labs(x = "Item Song", y = "Box Office Collection", title = "Budget vs. Item Song")
```

```
#T Test to find out whether Item songs make a difference in the box office collection or not
t_test_result <- t.test(Box_Office_Collection ~ Item_Song, data = moviedata)
print(t_test_result)

#Q1 c. How does digital medium impact box office collection?
ggplot(moviedata, aes(x = Youtube_Views, y = Box_Office_Collection)) +
  geom_point() + geom_smooth(method = "lm", se = FALSE, color = "blue")+
  labs(x = "Youtube Views", y = "Box Office Collection") +
  ggtitle("Scatter Plot of Youtube Views vs Box Office Collection")

# Combine the data for Youtube_Likes and Youtube_Dislikes
combined_data <- rbind(
  transform(moviedata, Variable = "Youtube Likes"),
  transform(moviedata, Variable = "Youtube Dislikes")
)

# Plot both scatter plots in one graph
ggplot(combined_data, aes(x = ifelse(Variable == "Youtube Likes", Youtube_Likes,
Youtube_Dislikes), y = Box_Office_Collection)) +
  geom_point() +
  geom_smooth(method = "lm", se = FALSE, aes(color = Variable)) +
  labs(x = "Youtube Interaction", y = "Box Office Collection") +
  ggtitle("Scatter Plot of Youtube Likes/Dislikes vs Box Office Collection") +
  facet_wrap(~ Variable, scales = "free") +
  scale_color_manual(values = c("Youtube Likes" = "green", "Youtube Dislikes" = "red"))

anova_result_2 <- aov(Box_Office_Collection ~ Youtube_Views, data = moviedata)
summary(anova_result_2)


#Q1 d. What is the estimated difference in box office collection for different lead actor
categories?
anova_model <- aov(Box_Office_Collection ~ Lead_Actor, data = moviedata)
summary(anova_model)
TukeyHSD(anova_model)

#Q1 e. Is there a significant difference in the budget of different movie types by content?
# Fit ANOVA model
anova_model_2 <- aov(Budget ~ Movie_Content, data = moviedata)
summary(anova_model_2)

#Q2. Create a logistic regression model using budget as an independent variable and success
as a dependent.
attach(moviedata)
```

```
set.seed(456)
train=sample(1:nrow(moviedata),nrow(moviedata)*0.70)
training_data=moviedata[train,]
testing_data=moviedata[-train, ]
training_data$S_F <- factor(training_data$S_F, levels = c("0", "1"))

#logistic regression
logm <- glm(S_F ~ Budget, data = moviedata, family = "binomial")
summary(logm)
varImp(logm)

#Q2 a. Calculate the budget for which box office success and failure are equally likely
# Coefficients from the logistic regression model
intercept <- -0.525651
budget_coefficient <- 0.005356

budget_likely <- -intercept / budget_coefficient
budget_likely

#Q2 b. -	Is there sufficient evidence to conclude that higher-budget movies are more likely to fail at the box office - Standard error and P value

#Q2 c. A production house is making a movie with a 100-crore budget. What is the probability of success for this movie?
new_data <- data.frame(Budget = 100)
predicted <- predict(logm, newdata = new_data, type = "response")
predicted

#Q2 d. What is the sensitivity and specificity of the classification model used? Interpret your findings

log_pred <- predict(logm, testing_data, type = "response")
log_pred_class <- ifelse(log_pred > 0.5, "1", "0")
testing_data$S_F <- factor(testing_data$S_F, levels = c("0", "1"))
log_pred_factor <- factor(log_pred_class)

conf_matrix <- confusionMatrix(log_pred_factor, testing_data$S_F)
print(conf_matrix)

#Q3. Create a logistic regression model using Item song as an independent variable and success as a dependent variable
logm_2 <- glm(S_F ~ Item_Song, data = moviedata, family = "binomial")
summary(logm_2)
```

```
prob_song <- predict(logm_2,
                            newdata = data.frame(Item_Song = 1),
                            type = "response")
prob_no_song <- predict(logm_2,
                               newdata = data.frame(Item_Song = 0),
                               type = "response")
difference_in_probabilities <- prob_song - prob_no_song
difference_in_probabilities

ggplot(moviedata, aes(x = Item_Song, y = S_F, fill = Item_Song)) +
  geom_bar(stat = "identity") +  # Use stat = "identity" to represent raw data
  labs(x = "Item Song", y = "Count of success", title = "Success vs. Item Song") +
  theme_bw()


#Q3 b. comparison
anova(logm, logm_2, test = "Chisq")

AIC(logm, logm_2)

library(pROC)
roc_1 <- roc(moviedata$S_F, predict(logm, type = "response"))
roc_2 <- roc(moviedata$S_F, predict(logm_2, type = "response"))
plot(roc_1, col = "blue")
plot(roc_2, col = "red", add = TRUE)


#Q4.    Develop a model to predict the success of the movie using all the variables provided.
Explain the factors affecting the success/failure of a movie. What are the rules that can be used
to predict the success or failure
logm_3 <- glm(S_F ~ . - Movie_Name, data = training_data, family = "binomial")
vif(logm_3)


dt_model <- rpart(S_F ~ Release_Date + Genre + Movie_Content + Director + Item_Song +
Lead_Actor + Production_House + Music_Dir + Box_Office_Collection + Profit + Budget +
Youtube_Views,
                  data = training_data,
                  method = "class")

rpart.plot(dt_model)

var_importance <- varImp(dt_model)
var_importance
```

```
dt_model_1 <- rpart(S_F ~ Movie_Content + Production_House + Box_Office_Collection +
Profit + Youtube_Views,
            data = training_data,
            method = "class",
            control = rpart.control(minsplit = 10,
                         minbucket = 5,
                         cp = 0.01,
                         maxdepth = 5))

rpart.plot(dt_model_1)

#confusion matrix- decision tree
tree_pred <- predict(dt_model_1, testing_data, type = "class")
tree_pred_factor <- factor(tree_pred)
testing_data$S_F <- factor(testing_data$S_F, levels = c("0", "1"))
conf_matrix_dt <- confusionMatrix(tree_pred_factor, testing_data$S_F)

print(conf_matrix_dt)
```