# Group Number - Group E (Tim Tieng, Vanessa Caldera, Matthew Hill)

# Subject - IST 657 Intro to Data Science

# Assignment - Group Project

# Code Author - Tim Tieng

# Load Packages

library(tidyverse)

library(ggplot2)

library(jsonlite)

library(readr)

library(lattice)

library(caret)

library(measures)

library(rpart)

library(rpart.plot)

library(kernlab)

library(quanteda)

library(quanteda.textplots)

# Create a dataframe from the movies.csv file

setwd("/Users/timtieng/Library/CloudStorage/OneDrive-Personal/Desktop/Masters in Applied Data Science/IST-687 Intro to Data Science/Project/TMDB 5000 Movie/Datasets")

movieFile <- "tmdb\_5000\_movies.csv"

moviesDF <- data.frame(read\_csv(movieFile))

# Inspect to see if moviesDF was successfully created

str(moviesDF) # 4803 Rows, 20 Columns

# Columns: budget, genres, homepage, id, keywords, orginal\_language, original\_title, overview, popularity, production\_companies, production\_countries, release\_date, revenue, runtime,

# spoken\_languages, status, tagline, title, vote\_average, vote\_count,

# Task 1 - Analyse how budget correlates with movie popularity

# Dependent Variable - popularity

# Independent Variable - budget

# Determine if there are any NA values in the two columns -- Expecting 0's

numNABudgetColumn <- sum(is.na(moviesDF$budget))

numNABudgetColumn

numNAPopularityColumn <- sum(is.na(moviesDF$popularity))

numNAPopularityColumn

# Visualize using Scatterplot

budgetVSPopularityPlot <- ggplot(moviesDF) +

geom\_point(aes(x=budget, y=popularity)) +

ggtitle("Budget Versus Popularity Rating")

budgetVSPopularityPlot

# Linear Regression Model

budgetLM <- lm(formula = popularity ~ budget, data=moviesDF)

summary(budgetLM)

# Visualize Populartiy vs additional ind. variables

genresVSPopularityPlot <- ggplot(moviesDF) +

geom\_point(aes(x=genres, y=popularity)) +

ggtitle("Genres Versus Popularity Rating")

genresVSPopularityPlot

voteAvgVSPopularityPlot <- ggplot(moviesDF) +

geom\_point(aes(x=vote\_average, y=popularity)) +

ggtitle("Vote Average Versus Popularity Rating")

voteAvgVSPopularityPlot

voteCountVSPopularityPlot <- ggplot(moviesDF) +

geom\_point(aes(x=vote\_count, y=popularity)) +

ggtitle("Vote Count Versus Popularity Rating")

voteCountVSPopularityPlot

releaseDateVSPopularityPlot <- ggplot(moviesDF) +

geom\_point(aes(x=release\_date, y=popularity)) +

ggtitle("Release DateVersus Popularity Rating")

releaseDateVSPopularityPlot

revenueVSPopularityPlot <- ggplot(moviesDF) +

geom\_point(aes(x=revenue, y=popularity)) +

ggtitle("Revenue Versus Popularity Rating")

revenueVSPopularityPlot

# Linear Regrssion Model - Multi Variable analysis (budget + genres + runtime)

budgetMultiVariable <- lm(formula = popularity ~ budget + genres + runtime + vote\_average + vote\_count + release\_date + revenue, data = moviesDF)

summary(budgetMultiVariable)

# Task 2 - Use another technique to dtermine which attributes have the most influence on the dependent variable

# Create a predictable sample

set.seed(111)

# Create the indices to use to create two additional data sets

trainList <- createDataPartition(y = moviesDF$popularity, p = .80, list = FALSE)

#Create test and train data sets

trainSet <- moviesDF[trainList, ]

testSet <- moviesDF[-trainList, ]

# Inspect

str(trainSet)

str(testSet)

dim(trainSet)

dim(testSet)

head(trainSet)

# Build treebag model

fit1 <- train(popularity ~ budget, runtime, vote\_average, vote\_count, data = trainSet, method = "treebag",preProc = c("center", "scale"))

cartTree <- rpart(popularity ~ budget , runtime, vote\_average, data = trainSet, method = "class")

prp(cartTree, faclen = 0, cex = 0.8, extra = 1)

# Analyze tagline column via wordcloud to understand initially what words are used - potential to shed light on type of genre

movieTaglingCorpus <- corpus(moviesDF$tagline)

#summary(movieCorpus) # 43803 documents

# Format text to remove punctuation and numbers from corpous variable

movieTaglineCorpus <- tokens(movieTaglingCorpus, remove\_punct = TRUE, remove\_symbols = TRUE, remove\_numbers = TRUE)

head(movieCorpus, 10)

# Ensure all words are the same case

movieTaglineCorpus <- tokens\_tolower(movieTaglineCorpus)

# remove stop words

movieTaglineCorpus <- tokens\_select(movieTaglineCorpus, selection = "remove", pattern = stopwords("en"))

head(movieCorpus)

# Create Document Matrix

taglineMatrix <- dfm(movieTaglineCorpus)

taglineMatrix

taglineWordCloud <- textplot\_wordcloud(taglineMatrix, min\_count =25)

# Analyze genres column

genreCorpus <- corpus(moviesDF$genres)

summary(genreCorpus) # 43803 documents

head(genreCorpus)

# Format text to remove punctuation and numbers from corpous variable

genreCorpus <- tokens(genreCorpus, remove\_punct = TRUE, remove\_symbols = TRUE, remove\_numbers = TRUE)

head(genreCorpus, 10)

# Ensure all words are the same case

genreCorpus <- tokens\_tolower(genreCorpus)

# remove stop words

genreCorpus <- tokens\_select(genreCorpus, selection = "remove", pattern = stopwords("en"))

head(movieCorpus)

# Create Document Matrix

genreMatrix <- dfm(genreCorpus)

genreMatrix

genreWordCloud <- textplot\_wordcloud(genreMatrix, min\_count = 20)

#Code Author: Vanessa Caldera

#Librarying the Packages

library(ggplot2)

library(tidyverse)

library(imputeTS)

library(readr)

#Reading the file

df <- read.csv("/Users/vanessacaldera/Desktop/Syracuse University/Intro to Data Science/Final Project/Final Project Dataset.csv", header=TRUE,stringsAsFactors = FALSE)

df$runtime <- na\_interpolation(df$runtime)

df$popularity <- na\_interpolation(df$popularity)

#Put it all together

df <- df %>% arrange(df$runtime)

#Take away the 0

df <- df[-1:-35, ]

#Mean of the original df after cleaning

mean(df$runtime)

* 107.6543

mean(df$popularity)

* 2.15165

#Creating short and long runtime subsets

shortestaruntimes <- df[df$runtime > 150, ]

longestruntimes <- df[df$runtime <90, ]

#How many movies fall in each

length(shortetruntimes$popularity)

* 671

length(longestruntimes$popularity)

* 17

#Mean of the shortest variables

mean(shortestruntimes$runtime)

* 83.77124

mean(shortestruntimes$popularity)

* 12.15499

#Mean of the longest variables

mean(longestruntimes$runtime)

* 173.6725

mean(longestruntimes$popularity)

* 42.35088

#Linear models

shortmodel <- lm(formula= popularity ~ runtime, data= shortestruntimes)

longmodel <- lm(formula=popularity ~ runtime, data=longestruntimes)

originalmodel <- lm(formula=popularity ~ runtime, data=df)

#Summary of linear models

summary(shortmodel)

* R-Squared: 0.01491

summary(longmodel)

* R-Squared: 0.009518

summary(originalmodel)

* R-Squared: 0.04889

#Plotting

ggplot(df) + aes(x=runtime, y=popularity) + geom\_point()

ggplot(shortestruntimes) + aes(x=runtime, y=popularity) + geom\_point(aes(color=popularity))

ggplot(longestruntimes) + aes(x=runtime, y=popularity) + geom\_point(aes(color=popularity))

ggplot(middleruntimes,aes(x=runtime, y=popularity)) + geom\_point(aes(color=popularity))

#Looking at the top10 most popular movies

df2 <- df %>% arrange(desc(df$popularity))

top10 <- head(df2, 10)

mean(top10$runtime)

* 123.1

**#Creating a Word Cloud for Top 10 Popular Movie overviews**

#Librarying the packages

library(arules)

library(quanteda)

library(quanteda.textplots)

glimpse(df)

#Creating Top 10 df

popdf <- df %>% arrange(desc(df$popularity))

popdf <- popdf[1:10,]

glimpse(popdf)

#Creating Corpus

popdfCorpus <- corpus(popdf$overview, docnames = popdf$id)

popdfCorpus<- tokens(popdfCorpus, remove\_punct = TRUE)

popdfCorpus <-tokens\_tolower(popdfCorpus, keep\_acronyms = FALSE)

head(popdfCorpus,5)

popdfCorpus <- tokens\_remove(popdfCorpus, pattern = stopwords("en"))

popdfCorpus <- tokens(popdfCorpus, remove\_symbols = TRUE)

popdfCorpus <- tokens(popdfCorpus, remove\_punct = TRUE)

#Turning it into Matrix

popdfDFM <- dfm(popdfCorpus)

#Viewing DFM

popdfDFM

textplot\_wordcloud(popdfDFM, min\_count = 2)

#Code Author: Matthew Hill

#read in the csv

Movies<-read.csv(file='IMDB\_Dataset.csv')

#Select only the columns wanted for this research question

Movies<-select(Movies, Main.Genre,Secondary.Genre,vote\_average,vote\_count)

#Change the column names to easier to reference names

colnames(Movies)<- c("MainGenre","AltGenre","VoteAvg","VoteCount")

#Checking dataframe

head(Movies)

#Use dply to group the main genre column and then find the average vote score of those grouped genres

GenreAvg <- Movies %>%

group\_by(MainGenre) %>%

summarize(VoteAvg=mean(VoteAvg))

#Remove first blank line (movies with no genre)

GenreAvg=GenreAvg[-1,]

GenreAvg

#Place VoteAvg numeric value in a set to be used by barplot

Genrebar<-GenreAvg$VoteAvg

#create barplot using Genre as X axis and avg vote rating as y value. Include text of avg for each column

Genreplot<-barplot(Genrebar, names.arg=GenreAvg$MainGenre, xlab="Genre", las=2, ylab="Average Rating", main="Ratings by Main Genre")

text(Genreplot, Genrebar-.3, labels=round(Genrebar, digits=2))

#Use dply to group the main genre column and then find the Total number of votes in those grouped genres

GenreSum <- Movies %>%

group\_by(MainGenre) %>%

summarize(VoteCount=sum(VoteCount))

#Remove first blank line (movies with no genre)

GenreSum=GenreSum[-1,]

GenreSum

#Place VoteCount numeric value in a set to be used by barplot

GenreSumbar<-GenreSum$VoteCount

#create barplot using Genre as X axis and total vote count as y value. Include text of total count for each column

GenreSumplot<-barplot(GenreSumbar, names.arg=GenreSum$MainGenre, xlab="Genre", las=2, ylab="Total votes", main="Vote Counts by Main Genre")

text(GenreSumplot, GenreSumbar, labels=round(GenreSumbar, digits=2))