**Movie Analysis**

setwd("E:/R Directory/Movie\_Analysis")

getwd()

**#Read a file**

Movie = read.csv("Section6-Homework-Data.csv",stringsAsFactors = FALSE)

str(Movie)

colnames(Movie) = c("DayReleased","Director","Genre","MovieName","ReleaseDate",

"Studio","Adj.Gross\_Mill","Budjet\_Mill","Gross\_Mill","IMDB","MovieLens",

"Overseas\_Mill","Overseas\_Per","Profit\_Mill","Profit",

"Runtime\_Min","US\_Mill","Gross\_US\_Mill")

summary(Movie)

Movie$Adj.Gross\_Mill = gsub('[^a-zA-Z0-9.]', '',Movie$Adj.Gross\_Mill)

Movie$Adj.Gross\_Mill = as.numeric(Movie$Adj.Gross\_Mill)

Movie$Gross\_Mill = gsub('[^a-zA-Z0-9.]', '',Movie$Gross\_Mill)

Movie$Gross\_Mill = as.numeric(Movie$Gross\_Mill)

Movie$Overseas\_Mill = gsub('[^a-zA-Z0-9.]', '',Movie$Overseas\_Mill)

Movie$Overseas\_Mill = as.numeric(Movie$Overseas\_Mill)

Movie$Profit\_Mill = gsub('[^a-zA-Z0-9.]', '',Movie$Profit\_Mill)

Movie$Profit\_Mill = as.numeric(Movie$Profit\_Mill)

Movie$DayReleased = factor(Movie$DayReleased)

Movie$Director = factor(Movie$Director)

Movie$Genre = factor(Movie$Genre)

Movie$MovieName = factor(Movie$MovieName)

Movie$Studio = factor(Movie$Studio)

Movie$ReleaseDate = as.Date(x = Movie$ReleaseDate,"%d/%m/%Y")

Movie$ReleaseDate = format(x = Movie$ReleaseDate,"%Y-%b-%d")

Movie$ReleaseDate = factor(Movie$ReleaseDate)

saveRDS(Movie,file = "Movie.rds")

#Load packages

library("dplyr")

library("tidyr")

library("ggplot2")

#Seperate date and save in new dataset

Data = separate(data = Movie,col = ReleaseDate,into = c("Year","Month","Day"),sep ="-" )

Data$Year = factor(Data$Year)

Data$Month = factor(Data$Month)

Data$Day = factor(Data$Day)

str(Data)

saveRDS(Data,file = "Data.rds")

Data = readRDS("Data.rds")

#Expolore DayReleased variable

Day = table(Data$DayReleased)

install.packages("xlsx")

library(xlsx)

write.xlsx(Day,"Analysis.xlsx",sheetName = "Day",row.names = FALSE)

Day\_Year = table(Data$Year,Data$DayReleased)

Year = table(Data$Year)

write.xlsx(Day\_Year,"Analysis.xlsx",sheetName = "Day\_Year",row.names = FALSE,

append = TRUE)

write.xlsx(Year,"Analysis.xlsx",sheetName = "Year",row.names = FALSE,append = TRUE)

View(Day\_Year)

#Day wise average budjet and profit

Av = Data %>% group\_by(DayReleased) %>% summarise(Budjet\_Avg = mean(Budjet\_Mill),Gross\_Avg=mean(Gross\_Mill),

Profit\_Avg=mean(Profit\_Mill))

Av$Profit\_Per\_Avg = (Av$Profit\_Avg/Av$Budjet\_Avg)\*100

#Plot Day vs Profit percent

Av = arrange(Av,desc(Profit\_Per\_Avg))

Av$DayReleased = as.character(Av$DayReleased)

ggplot(Av,aes(DayReleased,Profit\_Per\_Avg))+

geom\_bar(stat = "identity",fill="DarkBlue",width = 0.5)+

scale\_x\_discrete(limits = Av$DayReleased)+ggtitle("Day vs Profit Average")+

xlab("Day Released")+ylab("Profit Percentage")

str(Av)

library(scales)

install.packages("tidyverse")

library(tidyverse)

#Overseas and US

Ou = Data %>% group\_by(DayReleased) %>% summarise(Overseas\_Avg=mean(Overseas\_Mill),US\_Avg = mean(Gross\_US\_Mill),Budjet\_Avg=mean(Budjet\_Mill))

Ou$Overseas\_Profit\_Per = (Ou$Overseas\_Avg/Ou$Budjet\_Avg)\*100

Ou$US\_Profit\_Avg = (Ou$US\_Avg/Ou$Budjet\_Avg)\*100

#Plot Overseas gross and US gross

ggplot(Data,aes(Overseas\_Mill,Gross\_US\_Mill))+geom\_point(colour="DarkGreen")+

ggtitle("Overseas vs Profit")+xlab("Overseas\_Gross")+ylab("US\_Gross")

#Plot Imdb vs Movielens

ggplot(Data,aes(IMDB,MovieLens))+geom\_point(colour="Red")+ggtitle("IMDB vs MovieLens")+

xlab("IMBD Rating")+ylab("Movielens Rating")

#Plot IMDB vs Profit

ggplot(Data,aes(IMDB,Profit\_Mill))+geom\_point(colour="Violet")+

scale\_x\_continuous(breaks = seq.int(0,10,1))+

scale\_y\_continuous(breaks = seq.int(0,3000,200))+

ggtitle("IMDB vs Profit")+xlab("IMDB Rating")+ylab("Profit(US Million Dollar")

#Plot IMDB vs Profit

ggplot(Data,aes(MovieLens,Profit\_Mill))+geom\_point(colour="Orange")+

scale\_x\_continuous(breaks = seq.int(0,5,1))+

scale\_y\_continuous(breaks = seq.int(0,3000,200))+

ggtitle("Movielens vs Profit")+xlab("Movielens Rating")+ylab("Profit(US Million Dollar")

#Create Table for Genre

Genre = table(Data$Genre)

#Bar plot for Genre

a =table(Data$Genre)

View(a)

b =as.data.frame(a)

f = Data

ggplot(f,aes(Genre,fill=Profit.Percent))+geom\_bar()+coord\_flip()+

ggtitle("Profit Percent in Genre")+xlab("Genre")+ylab("No of Movies")+

scale\_y\_continuous(breaks=seq(0,250,25))

?reorder

install.packages("forcats")

library(forcats)

f$Genre =fct\_rev(f$Genre)

f$Profit\_Catefory = "50"

for (i in 1:nrow(f)) {

if(f$Profit[i]<=50){

f$Profit\_Catefory[i] ="0-50"

}else if(f$Profit[i]>50 & f$Profit[i]<=100){

f$Profit\_Catefory[i] ="51-100"

}else if(f$Profit[i]>100 & f$Profit[i]<=500){

f$Profit\_Catefory[i] = "101-500"

}else if(f$Profit[i]>500 & f$Profit[i]<=1000){

f$Profit\_Catefory[i] = "501-1000"

}else if(f$Profit[i]>1000 & f$Profit[i]<=5000){

f$Profit\_Catefory[i] = "1001-5000"

}else if(f$Profit[i]>5000 & f$Profit[i]<=10000){

f$Profit\_Catefory[i] = "5001-10000"

}else{

f$Profit\_Catefory[i] = "Above 10000"

}

}

str(f)

f$Profit.Percent = factor(f$Profit.Percent,levels = c("0-50","51-100",

"101-500","501-1000","1001-5000","5001-10000","Above 10000"))

colnames(f)[21] = "Profit.Percent"

saveRDS(f,"f.rds")

#Plot Box plot for Genre and budjet

ggplot(f,aes(Genre,Budjet\_Mill))+geom\_boxplot(fill=rainbow(length(unique(f$Genre))))+

ggtitle("Genre vs Budjet")+xlab("Genre")+ylab("Budjet(US Million Dollars)")

#Top 20 studio

install.packages("glue")

library(glue)

library(dplyr)

library(tidyr)

Studio = sort(table(f$Studio),decreasing = TRUE)

View(Studio)

Studio = as.data.frame(Studio)

colnames(Studio) = c("Studio","No\_of\_Movies")

str

str(Studio)

Top10\_Studios = top\_n(Studio,10)

#Plot studio and Genre

ggplot(Top10\_Studios,aes(reorder(Studio,No\_of\_Movies),No\_of\_Movies))+

geom\_bar(mapping = aes(fill=Genre),data = f,stat = "identity")+coord\_flip()

library(forcats)

f$Studio = fct\_rev(f$Studio)

table(f$Studio,f$Genre)

l = f %>% group\_by(Studio) %>% summarise(Budjet = sum(Budjet\_Mill),Gross=sum(Gross\_Mill),n = n())

l$Profit.Percent = (l$Gross/l$Budjet)\*100

l = gather(data = l,"Finance","Amount",2:3)

l = arrange(l,desc(Profit.Percent))

Top5.ProfitPercent = l[1:10,]

#plot Studio vs Amount

ggplot(Top5.ProfitPercent,aes(Studio,Amount,fill=Finance))+

geom\_bar(stat = "identity",position = "dodge",width = 0.5)+

scale\_y\_continuous(breaks = seq(0,1200,100))+

ggtitle("Studios generated high profit percentage")+

xlab("Studio")+ylab("US Million Dollar")

#identify the movie names of top 5 studios in profit percent but produced 1 movies

T5 = f %>% filter(Studio%in%c("Art House Studios","IFC","Lionsgate Films","Vestron Pictures"))

#Multiple Linear Regression

#Exporing and preparing the data

str(Data)

summary(Data$Gross\_Mill)

ggplot(data = Data,aes(Gross\_Mill))+geom\_histogram(binwidth = 125,fill="Purple")+

scale\_x\_continuous(breaks = seq(0,3000,250))+ggtitle("Budjet Histogram")+

ylab("No of Movies")

table(Data$DayReleased)

table(Data$Genre)

#Expolring relationships among features-the correlation matrix

cor(Data[c("Budjet\_Mill","IMDB","MovieLens","Runtime\_Min","Gross\_Mill")])

#Visualising relationships among features-the scatterplot matrix

pairs(Data[c("Budjet\_Mill","IMDB","MovieLens","Runtime\_Min","Gross\_Mill")])

pairs.panels(Data[c("Budjet\_Mill","IMDB","MovieLens","Runtime\_Min","Gross\_Mill")])

Data$Rating = (Data$IMDB+(Data$MovieLens\*2))/2

cor(Data[c("Budjet\_Mill","Rating","Runtime\_Min","Gross\_Mill")])

pairs.panels(Data[c("Budjet\_Mill","Rating","Runtime\_Min","Gross\_Mill")])

#Prepare training and test dataset

set.seed(123)

sa = sample(608,480)

multiple\_training = Data[sa, ]

multiple\_test = Data[-sa, ]

#Train a model on the data(include IMDB and Movielens)

multiple\_model = lm(formula = Gross\_Mill~DayReleased+Genre+Budjet\_Mill+

IMDB+MovieLens+Runtime\_Min,data = multiple\_training)

multiple\_model

lm(formula = Gross\_Mill ~ DayReleased + Genre + Budjet\_Mill +

IMDB + MovieLens + Runtime\_Min, data = multiple\_training)

Coefficients:

(Intercept) DayReleasedSaturday DayReleasedSunday

-474.8178 -275.9945 -46.5980

DayReleasedThursday DayReleasedTuesday DayReleasedWednesday

61.1513 -40.4433 32.0201

Genreadventure Genreanimation Genrebiography

114.6584 73.4779 -141.6491

Genrecomedy Genrecrime Genredocumentary

40.8129 -111.2522 -84.6678

Genredrama Genrefantasy Genrehorror

62.4705 222.8007 83.5511

Genremusical Genremystery Genreromance

189.4600 -47.8656 -0.6609

Genresci-fi Genrethriller Budjet\_Mill

137.8359 5.3640 1.9947

IMDB MovieLens Runtime\_Min

71.9444 -27.6112 2.1980

#Evaluating model performance

summary(multiple\_model)

Call:

lm(formula = Gross\_Mill ~ DayReleased + Genre + Budjet\_Mill +

IMDB + MovieLens + Runtime\_Min, data = multiple\_training)

Residuals:

Min 1Q Median 3Q Max

-579.61 -118.48 -17.06 75.89 1832.16

Coefficients:

Estimate Std. Error t value Pr(>|t|)

(Intercept) -474.8178 91.7817 -5.173 3.45e-07 \*\*\*

DayReleasedSaturday -275.9945 216.4462 -1.275 0.202917

DayReleasedSunday -46.5980 214.0503 -0.218 0.827763

DayReleasedThursday 61.1513 49.4547 1.237 0.216905

DayReleasedTuesday -40.4433 88.5368 -0.457 0.648035

DayReleasedWednesday 32.0201 25.6628 1.248 0.212772

Genreadventure 114.6584 38.1316 3.007 0.002785 \*\*

Genreanimation 73.4779 33.6517 2.183 0.029509 \*

Genrebiography -141.6491 74.9323 -1.890 0.059344 .

Genrecomedy 40.8129 31.8125 1.283 0.200172

Genrecrime -111.2522 89.4175 -1.244 0.214070

Genredocumentary -84.6678 153.1129 -0.553 0.580551

Genredrama 62.4705 38.1652 1.637 0.102353

Genrefantasy 222.8007 108.3330 2.057 0.040291 \*

Genrehorror 83.5511 90.0457 0.928 0.353963

Genremusical 189.4600 213.9995 0.885 0.376446

Genremystery -47.8656 124.1433 -0.386 0.699997

Genreromance -0.6609 97.9040 -0.007 0.994617

Genresci-fi 137.8359 63.9913 2.154 0.031765 \*

Genrethriller 5.3640 69.9661 0.077 0.938923

Budjet\_Mill 1.9947 0.2040 9.776 < 2e-16 \*\*\*

IMDB 71.9444 22.1463 3.249 0.001245 \*\*

MovieLens -27.6112 43.2288 -0.639 0.523325

Runtime\_Min 2.1980 0.5817 3.779 0.000179 \*\*\*

---

Signif. codes: 0 ‘\*\*\*’ 0.001 ‘\*\*’ 0.01 ‘\*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 212.2 on 456 degrees of freedom

Multiple R-squared: 0.3814, Adjusted R-squared: 0.3502

F-statistic: 12.23 on 23 and 456 DF, p-value: < 2.2e-16

#Improve model Performance-Remove Day Release as it is not seems to be stastiscally significant

multiple\_Imp1 = lm(formula = Gross\_Mill~Genre+Budjet\_Mill+IMDB+MovieLens+Runtime\_Min,

data = multiple\_training)

summary(multiple\_Imp1)

#Remove Genre

multiple\_Imp2 = lm(formula = Gross\_Mill~Budjet\_Mill+IMDB+MovieLens+Runtime\_Min,data = multiple\_training)

summary(multiple\_Imp2)

#Include Rating and remove IMDB,movielens

multiple\_final = lm(formula = Gross\_Mill~Budjet\_Mill+Rating+Runtime\_Min,data = multiple\_training)

summary(multiple\_final)

Call:

lm(formula = Gross\_Mill ~ Budjet\_Mill + Rating + Runtime\_Min,

data = multiple\_training)

Residuals:

Min 1Q Median 3Q Max

-491.36 -122.69 -30.49 72.23 1937.10

Coefficients:

Estimate Std. Error t value Pr(>|t|)

(Intercept) -423.5618 83.2101 -5.090 5.16e-07 \*\*\*

Budjet\_Mill 2.0257 0.1727 11.733 < 2e-16 \*\*\*

Rating 64.7500 11.9860 5.402 1.04e-07 \*\*\*

Runtime\_Min 1.8139 0.4783 3.792 0.000169 \*\*\*

---

Signif. codes: 0 ‘\*\*\*’ 0.001 ‘\*\*’ 0.01 ‘\*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 216.5 on 476 degrees of freedom

Multiple R-squared: 0.3278, Adjusted R-squared: 0.3235

F-statistic: 77.36 on 3 and 476 DF, p-value: < 2.2e-16

#Predict model perfomance

cor(Data$Gross\_Mill,Data$Rating)

multiple\_Predict\_Imp2 = predict(object = multiple\_Imp2,newdata = multiple\_test)

summary(multiple\_Predict\_Imp2)

summary(multiple\_test$Gross\_Mill)

multiple\_Predict\_final = predict(object = multiple\_final,newdata = multiple\_test)

summary(multiple\_Predict\_final)

summary(multiple\_Predict\_final)

Min. 1st Qu. Median Mean 3rd Qu. Max.

91.96 307.67 406.28 423.72 505.47 967.64

multiple\_test$Multiple\_IMDB\_MovieLens = multiple\_Predict\_Imp2

multiple\_test$Multiple\_Rating = multiple\_Predict\_final

View(multiple\_test)

#Decision Tree Regression

#Use the same training and test data set

library(rpart)

?rpart

#Train a model

str(multiple\_training)

saveRDS(multiple\_training,"Multiple\_Training.rds")

saveRDS(multiple\_test,"Regression\_Test.rds")

View(multiple\_test)

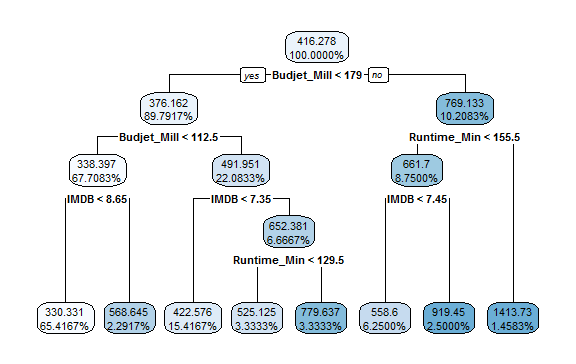
Decision\_model = rpart(formula = Gross\_Mill~Budjet\_Mill+IMDB+MovieLens+Runtime\_Min,data = multiple\_training)

Decision\_model

#Visualise decision Trees

library(rpart.plot)

rpart.plot(Decision\_model,digits = 6)



#Evaluate Model Performance

Decision\_Predict = predict(Decision\_model,newdata = multiple\_test)

summary(Decision\_Predict)

summary(multiple\_test$Gross\_Mill)

cor(Decision\_Predict,multiple\_test$Gross\_Mill)

> summary(Decision\_Predict)

Min. 1st Qu. Median Mean 3rd Qu. Max.

330.3 330.3 330.3 417.2 422.6 1413.7

> summary(multiple\_test$Gross\_Mill)

Min. 1st Qu. Median Mean 3rd Qu. Max.

201.6 241.4 330.0 402.9 458.0 1371.6

> cor(Decision\_Predict,multiple\_test$Gross\_Mill)

[1] 0.5387679

#Measure performance with mean absolute error

MAE = function(actual,predicted){

mean(abs(actual-predicted))

}

MAE(multiple\_test$Gross\_Mill,Decision\_Predict)

mean(multiple\_training$Gross\_Mill)

MAE(416.2779,multiple\_test$Gross\_Mill)

multiple\_test$Decision = Decision\_Predict

#Improving model performance using model Tree

library(RWeka)

M5\_Model = M5P(formula =Gross\_Mill~Budjet\_Mill+IMDB+MovieLens+Runtime\_Min,data = multiple\_training)

M5\_Model

summary(M5\_Model)

M5\_predict = predict(object = M5\_Model,multiple\_test)

summary(M5\_predict)

summary(multiple\_test$Gross\_Mill)

summary(Decision\_Predict)

summary(multiple\_Predict\_final)

multiple\_test$Model\_M5P = M5\_predict

cor(M5\_predict,multiple\_test$Gross\_Mill)

Anal = multiple\_test[c(11,22:25)]

View(Anal)

M5\_Model

M5 pruned model tree:

(using smoothed linear models)

Budjet\_Mill <= 112.5 :

| MovieLens <= 3.55 : LM1 (226/45.325%)

| MovieLens > 3.55 : LM2 (99/69.943%)

Budjet\_Mill > 112.5 :

| Runtime\_Min <= 135.5 : LM3 (109/74.658%)

| Runtime\_Min > 135.5 :

| | Budjet\_Mill <= 182.5 : LM4 (24/75.053%)

| | Budjet\_Mill > 182.5 :

| | | IMDB <= 7.55 : LM5 (16/104.981%)

| | | IMDB > 7.55 : LM6 (6/246.023%)

LM num: 1

Gross\_Mill =

0.5995 \* Budjet\_Mill

+ 6.049 \* IMDB

+ 0.0772 \* Runtime\_Min

+ 229.8822

LM num: 2

Gross\_Mill =

2.5382 \* Budjet\_Mill

+ 102.4402 \* IMDB

+ 0.0772 \* Runtime\_Min

- 551.2833

LM num: 3

Gross\_Mill =

1.0989 \* Budjet\_Mill

+ 86.1482 \* IMDB

+ 0.7807 \* Runtime\_Min

- 367.8006

LM num: 4

Gross\_Mill =

-0.6341 \* Budjet\_Mill

+ 127.0365 \* IMDB

+ 44.2018 \* MovieLens

+ 4.0902 \* Runtime\_Min

- 906.1059

LM num: 5

Gross\_Mill =

1.3344 \* Budjet\_Mill

+ 279.2295 \* IMDB

- 267.4119 \* MovieLens

+ 7.7529 \* Runtime\_Min

- 1667.6621

LM num: 6

Gross\_Mill =

1.3344 \* Budjet\_Mill

+ 349.2452 \* IMDB

- 354.1601 \* MovieLens

+ 9.4286 \* Runtime\_Min

- 1999.9366

Number of Rules : 6

> summary(M5\_Model)

=== Summary ===

Correlation coefficient 0.7164

Mean absolute error 125.0772

Root mean squared error 184.4002

Relative absolute error 69.3915 %

Root relative squared error 70.1293 %

Total Number of Instances 480

> M5\_predict = predict(object = M5\_Model,multiple\_test)

> summary(M5\_predict)

Min. 1st Qu. Median Mean 3rd Qu. Max.

228.3 316.0 346.3 431.3 490.0 1471.1

> summary(multiple\_test$Gross\_Mill)

Min. 1st Qu. Median Mean 3rd Qu. Max.

201.6 241.4 330.0 402.9 458.0 1371.6

> summary(Decision\_Predict)

Min. 1st Qu. Median Mean 3rd Qu. Max.

330.3 330.3 330.3 417.2 422.6 1413.7

> summary(multiple\_Predict\_final)

Min. 1st Qu. Median Mean 3rd Qu. Max.

91.96 307.67 406.28 423.72 505.47 967.64

> multiple\_test$Model\_M5P = M5\_predict

> cor(M5\_predict,multiple\_test$Gross\_Mill)

[1] 0.6664073

> Anal = multiple\_test[c(11,22:25)]