STA 141A Final Project

Analysis of Disney+ Shows

6/06/2022

Name	Contribution	Email
Aurora Travers	Linear Regression Analysis	autravers@ucdavis.edu
Hugo Moncada	k-NN Algorithm Analysis	hmamoncada@ucdavis.edu
Timothy Shen	Report Write Up, Data Visualization	tzshen@ucdavis.edu
Maha Shafaeen	Logistic Regression Analysis	mshafeen@ucdavis.edu
Jeff Lee	Descriptive Analysis, Report Write Up	fejlee@ucdavis.edu

Contents

Introduction

Filmmaking and television has been one of the most volatile careers in recent years with its unemployment rate consistently higher than the national average, especially during the pandemic, when it reached a peak of 39%.[1] Thus the ability of sustained success is relatively low and risky. We will look at movie and TV production specifically and attempt to quantify their likelihood of success. As a movie or TV producer, the goal is to earn a high audience rating as well as high popularity. Thus popularity and audience will be used as the parameter of success. There are a multitude of factors such as runtime, genre, and board certifications that may affect these parameters. So it is essential to analyze the relationship between these factors to be able to quantify the success of movies and tv shows.

Dataset Introduction

The data set[2] on Disney+ was obtained from Kaggle:
https://www.kaggle.com/datasets/timmofeyy/-current-available-disney-projects.
The data is composed of 7850 movies/shows and contains all the projects on Disney+, including projects after Disney company take-overs. Additionally, each project includes 8 variables: title of project, year(s) of runtime, certification, runtime length (in minutes), audience rating (based from IMDb), number of votes from the audience rating, and director or star actor/actress. Since the director or star actor/actress variable is not consistent, with some projects including the director while others include the star actor/actress, it will be dropped for our analysis. Furthermore the year(s) of the runtime variable offers different unit measures for tv shows (length of show runtime) and movies (year of release), so we are dropping this variable from our analysis as well. Finally, we ran the following data manipulation:

1. In order to standardize the certification[3], as some were based from Canada's rating system while others from the US', we subjected them to the following global factors:

Table 1a: Standardization of Certification Variable

New Certification	Original Certifications
Suitable for All	G
Suitable for Kids	6+, PG, TV-Y, TV-G, TV-Y7, TV-Y7-FV, TV-PG
Suitable for Teens	PG-13, TV-14
Suitable for Adults	R, TV-MA
Not Rated	Not Rated, No Data, Unrated, Approved, Passed

2. There are 304 unique genres as some projects list multiple. Thus we simplified the genres by the first genre listed, its primary.

Research Questions

In order to quantify the likelihood of success for movies and tv shows, we will answer these questions:

1. How do the different runtimes/genres/certificates determine the audience rating? Which factor best predicts the audience rating?
2. How do the different runtimes/genres/certificates affect the popularity? Which factor has the highest significance on popularity?
3. How does popularity affect the rating? What is the relationship between popularity and audience rating?

Descriptive Analysis

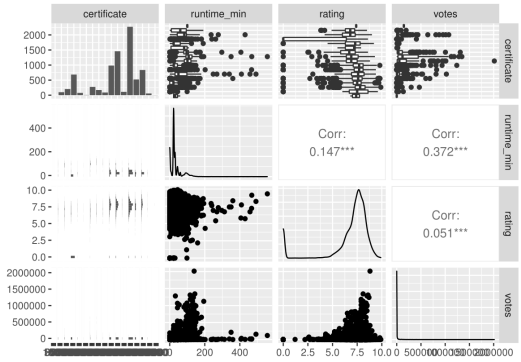


Figure 2a: Correlation between variables without genre.

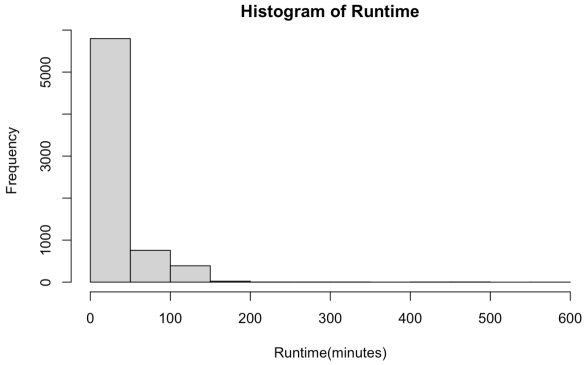
The figure on the left shows that there is not much correlation between the variables. The correlation coefficient between votes(popularity) and runtime(minutes) is the highest with 0.372. The p-values for all the correlations were very close to 0, making the results statistically significant.

The data contains three different variables that will be analyzed. These are runtime length, genres, and certificates. Runtime length is a quantitative variable while genre and certificate are qualitative variables.

Table 2: Summary of Runtime Length

Min.	1st Q	Median	Mean	3rd Q.	Max.
0	21	23	35.29	41	557

Figure 2b: Histogram of Runtime Length



The table above shows the general information on the runtime length variable. Although its mean is 35.29 with the 3rd Quartile being 41, the max is much higher than both of these. This suggests that there are likely outliers in the runtime portion of the data. The figure above shows the frequency of different runtimes. As we can see, the majority of the data was below fifty minutes, leading us to believe that these were most likely tv-shows. It is skewed heavily to the left, and quickly dies out after it goes over 100 minutes.

Figure 2c: Bar Plot of Genres

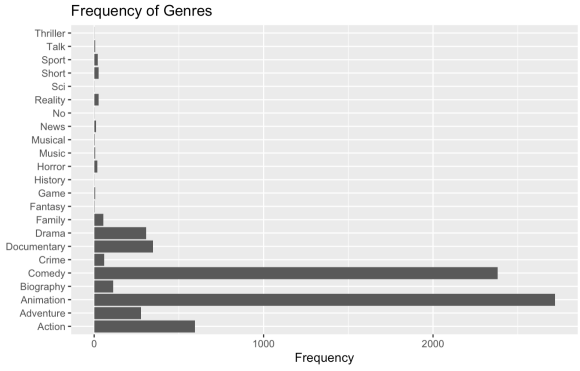


Figure 2d: Bar Plot of Certificate

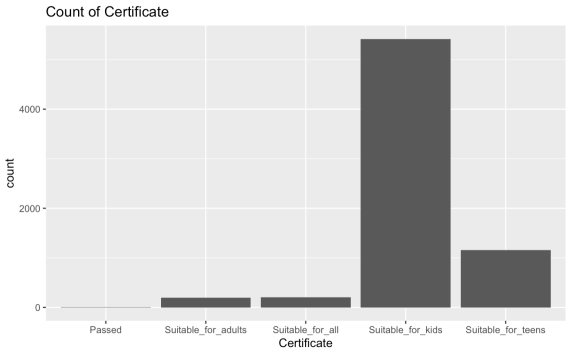


Figure 2c above shows the frequency of genres. As mentioned before, the genre was classified by its primary one, leading to the most common genres to be animation and comedy by a large margin. Figure 2d shows that the most common certificate from the Disney+ dataset is “Suitable for kids”. The second most common is “Suitable for teens”. This is to be expected because Disney is typically catered towards children.

Data Analysis

Methodology

To find which factors best predict the audience rating of Disney+ projects, we used the supervised learning method of Logistic Regression. We try to find which predictor best can classify *audience rating*. Furthermore, we will use the k-NN algorithm to further test the effectiveness of *runtime* as a classifier to *audience rating*. Then we will use linear regression to understand the relationship between predictors *genre, certificate,* and *runtime* and the response *votes. Votes* will be used as the quantifier of popularity. Finally we will run another linear regression to examine how significant popularity, *votes,* is in effecting *audience rating*.

Logistic Regression Analysis

First, we converted the *audience rating* to a binary predictor variable with its indicator as one if the rating is above 7 (good), and zero otherwise (bad). For our first model, we focused on the relationship between the *certificate* predictors and *audience rating* response variable. Since *certificate* is a categorical variable with 5 levels, we used the level *No Rating* as our reference variable to run the logistic regression model.

Table 3a: Summary of Logistic Regression of First Model

	Estimate	**Standard Error**	**P-Value**
Intercept	-1.25276	0.08270	< 2e-16
Suitable for Adults	1.61190	0.16675	< 2e-16
Suitable for All	0.94494	0.16436	8.97e-09
Suitable for Kids	2.30068	0.08921	< 2e-16
Suitable for Teens	2.14642	0.10921	< 2e-16

All predictors appear to be important in predicting audience rating compared to not having a rating as their corresponding p-values are significantly low (Table 3a). It appears all *certificate* variables have a positive association with audience rating However, when there are no ratings there appears to be an expected decrease in the likelihood of a good audience rating.

For our second model, we focused on the relationship between *genre* predictors and the *audience rating* response variable. Like *certificate*, *genre* is a categorical variable with 23 levels, so we used the level *No Genre* as our reference variable to run the logistic regression model.

Table 3b: Summary of Logistic Regression of Second Model

	P-Value		**P-Value**		**P-Value**		**P-Value**
Intercept	0.941	*Crime*	0.939	*History*	0.974	*Reality*	0.943
Action	0.939	*Documentary*	0.941	*Horror*	0.939	*Science Fiction*	0.941
Adventure	0.941	*Drama*	0.939	*Music Show*	0.943	*Short Film*	0.942
Animation	0.938	*Family*	0.944	*Musical*	0.943	*Sports*	0.938
Biography	0.942	*Fantasy*	0.943	*Mystery*	~1	*Talk*	0.940
Comedy	0.937	*Game Show*	0.943	*News*	0.940	*Thriller*	0.974

The result contains all of the predictor variables having a p-value greater than 0.941, we conclude that the predictor coefficients are not significantly different than a coefficient of 0.

Based on the conclusion of the first model and the second model, we excluded the *genre* variable from our third model and included *certificate variables* given their low p-values. We also included the predictor of *runtime* in our model. *Audience rating* is still the response variable.

Table 3c: Summary of Logistic Regression of Third Model

	Estimate	**Standard Error**	**P-Value**
Intercept	-1.09418	0.08432	< 2e-16
Suitable for Adults	2.130196	0.17921	< 2e-16
Suitable for All	1.367268	0.17213	1.97e-15
Suitable for Kids	2.396316	0.09074	< 2e-16
Suitable for Teens	2.396821	0.11420	< 2e-16
Runtime (in min)	-0.00794	0.00082	<2e-16

Similar to the first model, all of the predictors variables appear to have low p-values (Table 3c). All of the predictors with exception to *Runtime* have a positive association with *audience rating*.

All of the models were developed through the training set of the first 6,849 projects. Using the test data of the remaining 1,001 projects, we compare between the models. The first model had the lowest error rate 19.88%, although the difference in error rate between the first and third model is very trivial. However, it is important to note that across all models, the false negatives rates were all high with 88%, 96%, 88% across the models respectively (Table 3d).

Table 3d: Confusion Matrices of All Models

Model 1	***Predicted***	
True	**Bad**	**Good**
Bad	26	2
Good	197	776
Error Rate	19.88%	

Model 2	***Predicted***	
True	**Bad**	**Good**
Bad	8	52
Good	215	726
Error Rate	26.67%	

Model 3	***Predicted***	
True	**Bad**	**Good**
Good	25	2
Bad	198	776
Error Rate	19.90%	

The difference in AIC between Model 1 and Model 3 were relatively trivial as well with levels 7923.9 and 7823.1 respectively. Although still very high, it is much better than Model 2 which has an AIC of 8569.6.

Figure 3e: ROC Curve of Model 1

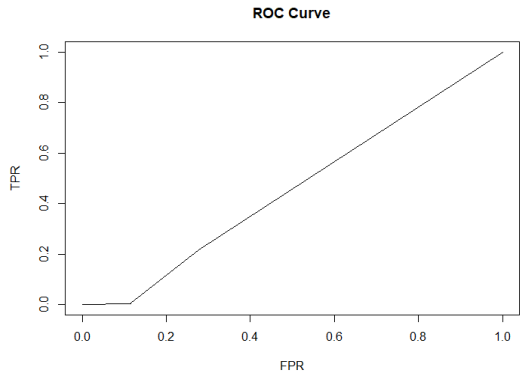
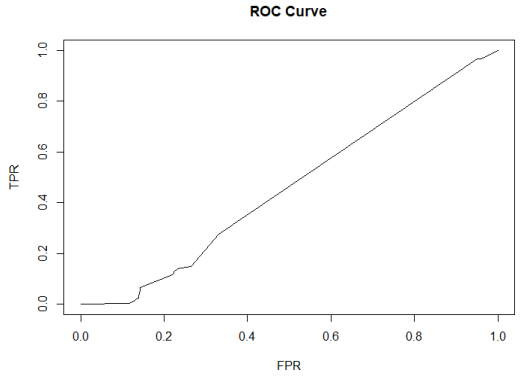


Figure 3f: ROC Curve of Model 3

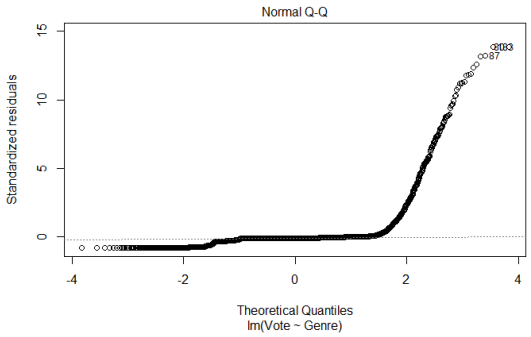


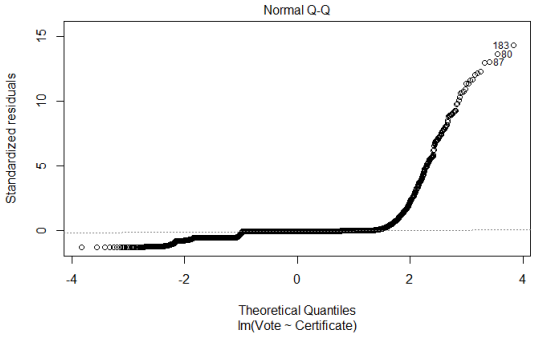
Despite Model 1 and Model 3 having a better fit than Model 2, based on their ROC Curves, it is still very random and not a good binary classifier (Figures 3e, 3f).

Linear Regression Analysis

In order to examine the factors that significantly impact popularity we used the supervised learning method of Linear Regression. For our first two models we set *genre* as our predictor for the first and *certificate* for the second. *Votes* were the response for both.

Figure 4a: Normal Probability Plot of the first two Linear Regression Models

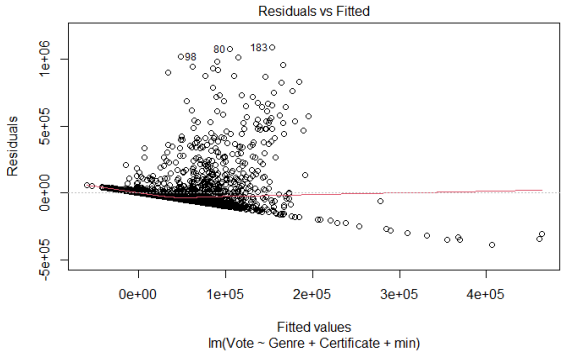


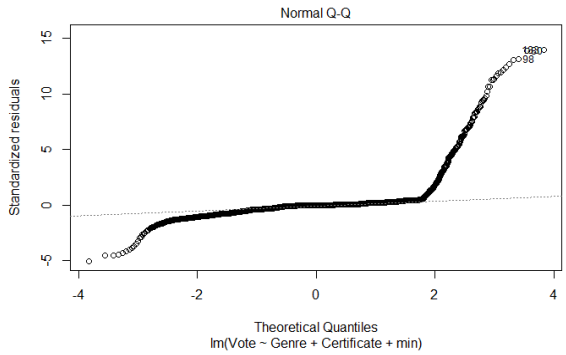


Model 1 (Votes ~ Genre) *Model 2 (Votes ~ Certificate)*

Both models appear to have significant departures from the straight line pattern (Figure 4a), thus suggesting non-normality. Furthermore in our third model, we ran the full model with *genre*, *certificate*, and *runtime* as predictors and *votes* still as the response. Similarly it produced a non-normal QQ plot, and a pattern appeared in its Residual vs. Fitted Plot (Figure 4b).

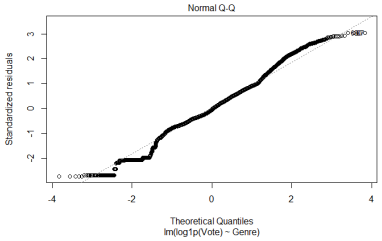
Figure 4b: Residual vs Fitted Plot and Normal Probability Plot (QQ Plot) of Third Model

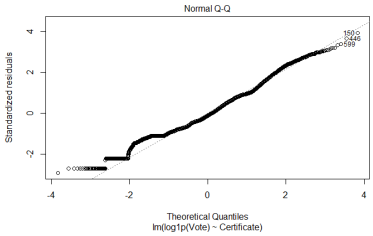


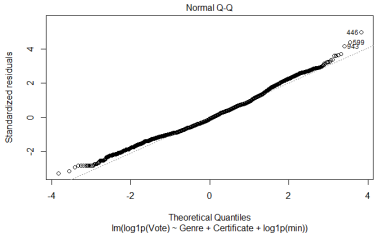


Thus we performed *log* transformation to the response variable, *votes*, as well as to the numerical predictor variables. The following were the normal probability plots of the new models. They appear to look much more normal (Figure 4c).

Figure 4c: Normal Probability Plots of the models post-transformation







Model 1: *log*(Votes) ~ Genre **Model 2**: *log*(Votes) ~ Certificate **Model 3**: *log*(Votes) ~ Genre + Cert. + *log*(Runtime)

Removing outliers, we focused our first model of *log*(Votes) ~ Genres. From the result all *genre* predictors are significant given that they all have very low p-values. The only insignificant coefficient is the intercept. However its adjusted R-squared is very low at 0.07219.

Again removing outliers, the second model produced the following output:

Table 4a: Summary of Logistic Regression of Second Model

	Estimate	**Standard Error**	**P-Value**
Intercept	2.87319	0.08672	< 2e-16
Suitable for Adults	6.76208	0.20252	< 2e-16
Suitable for All	6.01606	0.19925	< 2e-16
Suitable for Kids	2.81158	0.09344	< 2e-16
Suitable for Teens	4.07408	0.11497	< 2e-16

It appears all *certificate* predictors are significant given their low p-values (Table 4a). Its adjusted R-squared is 0.21.

Finally in our third model, the full model, it produces the following output:

log(Votes) ~ Genre + Certificate + *log*(Runtime)

Table 4b: Summary of Logistic Regression of Third Model (G - Genre, C - Certificate)

	Estimate	**P-Value**		**Estimate**	**P-Value**
Intercept	-0.19220	0.685	*History (G)*	2.62378	0.238221
Action (G)	2.91149	2.50e-09	*Horror (G)*	2.23301	0.000880
Adventure (G)	2.06886	2.74e-05	*Music Show (G)*	0.71747	0.391071
Animation (G)	2.12796	1.00e-05	*Musical (G)*	1.91794	0.076500
Biography (G)	1.89982	0.000239	*Mystery (G)*	1.60139	0.176633
Comedy (G)	1.52683	0.001542	*News (G)*	0.22775	0.744771
Crime (G)	3.25330	2.68e-09	*Reality (G)*	0.60116	0.293186
Documentary (G)	1.01600	0.037050	*Science Fiction (G)*	1.61044	0.316552
Drama (G)	1.07159	0.030105	*Short Film (G)*	1.54088	0.010164
Family (G)	1.22144	0.22437	*Sports (G)*	1.76447	0.006296
Fantasy (G)	0.51789	0.632262	*Talk (G)*	1.55550	0.101717
Game Show(G)	2.16269	0.009705	*Thriller (G)*	0.39581	0.859086
Suitable for All (C)	3.31206	< 2e-16	*Suitable for Kids (C)*	1.06738	< 2e-16
Suitable for Adults (C)	3.96098	< 2e-16	*Suitable for Teens (C)*	2.10387	< 2e-16
log(*Runtime*)	0.97515	< 2e-16			

Based on the output, for the *genre* predictor, levels *family*, *fantasy*, *history*, *music show*, *musical*, *mystery*, *news*, *reality*, *science fiction*, *talk*, *thriller* all fail to reject the null hypothesis that its coefficient is not zero at a significance level of 0.05 (Table 4b). However, all *certificate* levels are statistically significant as well as the *runtime* predictor. The adjusted R-squared is also much improved with a value of 0.4341.

Finally we ran a linear regression model comparing the impact of *audience rating* to *votes*. After running the Breusch-Pagan Test, it produced a p-value of 0.001. Thus rejected the assumption of homoscedasticity. After *log* transformations, the Breush-Pagan test produced a p-value of 0.604. The model appears to be much more normal. The following is the summary output of the transformed model:

log(Votes) ~ *log*(Rating)

Table 4c: Summary of Logistic Regression of Fourth Model

	Estimate	**Standard Error**	**P-Value**
Intercept	1.50462	0.18854	1.69e-15
Rating	2.20693	0.08929	< 2e-16

Rating predictor has a low p-value, thus it is statistically significant in explaining *audience rating* (Table 4c). However the adjusted R-squared is 0.08.

K-NN Algorithm

For the K-NN Algorithm, we split the data by runtime into 2 groups. We can see from table 1 that the majority of the data has a runtime under 100 minutes. Therefore, our final data will have a runtime of under 100 minutes with 6042 observations. A test set was created containing 20% of the data along with a training set containing the remaining 80%. The confusion matrix was then created with AM being "above mean" and BM being "below mean". The results are shown below.

Table 5: Confusion Matrix of Test Set

k-NN	***True***	
Predicted	**Bad**	**Good**
AM	82	239
BM	220	667

Figure 5: k-NN of Test Set

Taking the above mean as the positive value, the confusion matrix shows a low true positive, but has a high true negative. On the other hand, the K-NN table shows an error rate of 0.38 with the false negative and false positive being relatively the same. The K-NN model is not accurate enough to use in a real situation. While it is not very good for classifying those whose rating is above the mean, it is much better at predicting those who are below the mean. This is an interesting observation due to the fact that we have more observations with ratings above the mean than below the mean in the training set. As we can see in the figure 5 above, the results that were obtained are most likely because the training set had double the AM observations over BM while the test set had triple the BM observations over the AM. Therefore, the predictions of the BM observations are more accurate. Overall, although the K-NN model is fairly accurate with predicting values below the mean, the error rate is too high to use.

Discussion of Results

1. How do the different runtimes/genres/certificates determine the audience rating? Which factor best predicts the audience rating?

From the summary of the second logistic regression model (Table 3b), none of the *genre* variables are significant with all having p-values greater than 0.9. Thus concluding that none of the *genre* predictors have a significant impact in predicting the audience rating compared to having no genres. Furthermore, looking at the summary of the first logistic regression model (Table 3a), all of the *certificate* predictors have a p-value that is significantly low. Thus they all have a significant impact in predicting the audience rating. Specifically it appears that the odds of a project with a certificate that is suitable for kids having a good rating is $e^{2.3} = 9.9801$ times more likely than that of projects without a certification. All of the certificate attributes have positive association to audience rating suggesting that they all have a greater chance in having a good rating, whereas projects without a certificate (the intercept) only has an expected $e^{-1.253}/(1 + e^{-1.23}) = 22.22\%$ chance of having a good rating. Based on the third model of

logistic regression (Table 3c), it also appears that runtime is significant in predicting audience rating. However despite it being significant, with all other factors fixed, increasing runtime by one minute leads to the odds of having a good rating $1 - e^{-0.00794} = 0.0079$ times lower. Which is not that impactful. This is further proved through the kNN algorithm, in which runtime has an error rate 39% in predicting if its rating is above the mean rating. Thus concluding only the certificate variable significantly impacts and best predicts audience rating. However it is also important to note that despite the certificate variable being the better predictor, its error rate is still high with 19.88% (Table 3d) and an ROC curve (Figure 3e) suggesting it is not a good classifier for audience rating as well.

2. How do the different runtimes/genres/certificates affect the popularity? Which factor has the highest significance on popularity?

Using the number of votes as the quantifier for *popularity.* From the linear regression analysis, it appears that both genre and certificate are significant on votes when it is the lone predictor variable. However when inthe full model, it appears a lot of *genre* variables are not significant. Suggesting a possible collinearity in predictors which means a possible overlap between *genre* and the other variables in explaining *votes*. Furthermore, *certificate* variables remain significant in explaining *votes* in both the full model and the reduced model of just *certificate*. *Runtime* also is highly significant from the full model. Since both *certificate* and *runtime* have equally low p-values, they both are the highest significance on popularity. In particular, for every 1% increase in *runtime*, all other factors fixed, leads to a *97.5%* increase in votes. For *certificate* all the predictors are positive in association, thus having a *certificate* classification will lead to an increase in popularity compared to no certification. In particular, having a certification for adults leads to the highest increase where having a project certified in that level will lead to a $e^{3.96098} - 1 = 5,151\%$ increase in *votes*.

3. How does popularity affect the rating? What is the relationship between popularity and audience rating?

From the linear regression, it appears that for every 1% increase in *votes*, the *audience rating* increases by 220%. Thus suggesting that more popular movie/tv show projects will lead to a better rating. However, only 22.51% of variance in *audience ratings* is explained by *votes*. Thus although there may be a positive association between them, its relationship is still relatively weak.

Conclusion

To conclude, from the Disney+ projects dataset, it appears that having a certification leads to an increase in both audience rating and popularity. Specifically, projects suitable for kids and adults have the greatest impact in audience rating and popularity respectively. Also, from the analysis, genre appears to be a weak indicator of determining the likelihood of success of projects. It is also unclear whether there is a relationship between popularity and audience ratings. However it is important to note that we only considered a limited number of factors given the constraints of our datasets, thus there may be other factors (actors, directors, budget, etc.) that may also play a role. Thus predicting the likelihood of success in movies and projects would likely require more research in order to get a complete scope of the best quantifiers of success.

Appendix

Bibliography

[1] U.S. Bureau of Labor Statistics. (n.d.). *About the motion picture and Sound Recording Industries Subsector*. U.S. Bureau of Labor Statistics. Retrieved June 5, 2022, from https://www.bls.gov/iag/tgs/iag512.htm

[2] Efimpolianskii. (2022, March). Current Available Disney+ Projects Dataset. Kaggle. Retrieved May 19,2022, from https://www.kaggle.com/datasets/timmofeyy/-current-available-disney-projects

[3] IMDb.com. (2022, January 4). *Certificates*. IMDb. Retrieved June 5, 2022, from https://help.imdb.com/article/contribution/titles/certificates/GU757M8ZJ9ZPXB39?ref_=helpart_nav_27#usa

R-Code

```
data<-read.csv("Disney.csv")
factor_certificate = factor(data$certificate)
library(forcats)
certificate_group = matrix(fct_collapse(factor_certificate, Suitable_for_all
= c("G"), Suitable_for_kids = c("TV-Y", "TV-G", "TV-Y7", "TV-Y7-FV", "TV-PG",
"6+", "PG"), Suitable_for_teens = c("PG-13", "TV-14"),
                                    Suitable_for_adults = c("R",
"TV-MA"), Not_rated = c("No data", "Not Rated", "Unrated", "Approved",
"Passed")))
data$certificate = factor(certificate_group)
factor_genre = factor(data$genre)
library(stringr)
data$genre = matrix((str_extract(data$genre, "[aA-zZ]+")))
data$genre = relevel(factor(data$genre), ref = "No")
data$good_rating = as.factor(ifelse(data$rating > 7, "Good", "Bad"))
training.data = data[1:6849,]
test.data = data[6850:7850,]
confusion_matrix = function(pred, actual) {
  table(pred, actual, dnn = c("Predicted Rating", "Actual Rating"))
}

thresholds <- seq(0.01, 0.99, 0.005)
TPR <- numeric(length(thresholds))
FPR <- numeric(length(thresholds))
attach(data)

# Descriptive Analysis
library(ggplot2)
require(GGally)
aldat=disney[-c(1,2,5,8)]
```

```
ggpairs(data=aldat,cardinality_threshold=17)
summary(as.numeric(df2$Runtime_min))
table(df2$Genre)
table(df2$Certificate)
hist(as.numeric(df2$Runtime_min), main = "Histogram of Runtime", xlab =
"Runtime(minutes)")
ggplot(df2, aes(x = Genre)) +
  geom_bar() +
  labs(x = "",
       y = "Frequency",
       title = "Frequency of Genres") +
  coord_flip()
table(df2$Certificate)
ggplot(df2, aes(x = Certificate)) +
  geom_bar() +
  ggtitle("Count of Certificate")

# Logistic Regression
certificate = relevel(factor(certificate), ref = "Not_rated")
data.lr.cert = glm(good_rating ~ certificate, data = training.data, family =
"binomial")
summary(data.lr.cert) # Model 1
prediction1 = ifelse(predict(data.lr.cert, newdata =
test.data[c("certificate")], type = "response") > 0.5, "Good", "Bad")
confusion1 = confusion_matrix(prediction, test.data$good_rating)
er1<-1 - sum(diag(confusion1))/sum(confusion1)

genre = relevel(factor(genre), ref = "No")
data.lr.genre = glm(good_rating ~ genre, data = training.data, family =
"binomial")
summary(data.lr.genre) # Model 2
prediction2 = ifelse(predict(data.lr.genre, newdata = test.data[c("genre")],
type = "response") > 0.5, "Good", "Bad")
confusion2 = confusion_matrix(prediction, test.data$good_rating)
er2<-1 - sum(diag(confusion2))/sum(confusion2)

data.lr.reduced = glm(good_rating ~ certificate + runtime_min, data =
training.data, family = "binomial")
summary(data.lr.reduced) # Model 3
prediction3 = ifelse(predict(data.lr.reduced, newdata =
test.data[c("certificate", "runtime_min")], type = "response") > 0.5, "Good",
"Bad")
confusion3 = confusion_matrix(prediction, test.data$good_rating)
er3<-1 - sum(diag(confusion3))/sum(confusion3)

ROC<-function(model,test,level,true){ # Function for plotting ROC Curve
  thresholds <- seq(0.01, 0.99, 0.005)
  TPR <- numeric(length(thresholds))
  FPR <- numeric(length(thresholds))
```

```
  for (i in 1:length(thresholds)) {
    predicted <- factor(ifelse(predict(model, test, type = "response") >
thresholds[i],level[1], level[2]),
                        level)
    confusion <- table(as.character(true),
                       predicted,
                       dnn = c("True Rating", "Predicted Rating"))
    TPR[i] <- confusion[2,2]/(confusion[2,2]+confusion[2,1])
    FPR[i] <- confusion[1,2]/(confusion[1,2]+confusion[1,1])
  }
  plot(x = FPR, y = TPR, type = 'l', main = 'ROC Curve')
}

levels<-c("Good","Bad")
ROC(data.lr.cert,test.data[c("certificate")],levels,test.data$good_rating) #
ROC Curve for Model 1
ROC(data.lr.reduced, test.data[c("certificate",
"runtime_min")],levels,test.data$good_rating) # ROC Curve for Model 2

# k-NN Algorithm
dataset2=data[data$runtime_min<100,]
dataset2=dataset2[,-c(2,3,5,7,8)]
dataset2=dataset2[dataset2$runtime_min!=0,]
dataset2=dataset2[dataset2$rating!=0.0,]
Mean2=mean(dataset2$rating)
testset=dataset2[1:1208,]
trainset=dataset2[1209:6042,]
BAvec=ifelse(trainset$rating>=Mean2,"AM","BM")
trainset=cbind(trainset,BAvec)
trainset=trainset[-c(5:12)]
testsetmod=testset[,2]
trainsetmod=trainset[2]
knnfunction=function(index){ # k-NN Algorithm Function
  dis=abs(testsetmod[index]-trainsetmod)
  dis2=sort(dis[,1])
  dis3=min(dis2[dis2>0])
  dis4=head(rownames(which(dis==dis3,arr.ind=TRUE)),1)
  dis5=trainset$BAvec[row.names(trainset)==dis4]
  pbm=0
  pam=1
  if (dis5=="AM"){
    cp=pam/1
  }else {
    cp=pbm/1
  }
  fr=ifelse(cp>.5,"AM","BM")
  return(fr)
}
preval=NULL
```

```
for (i in 1:nrow(testset)){
  preval[i]=knnfunction(i)
}
new_testset=cbind(testset,ifelse(testset$rating>=Mean2,"AM","BM"))
new_testset=new_testset[,4]
cm=table(preval,new_testset,dnn = c("Predicted rating","True rating"))
(sum(diag(cm))/sum(cm))
newdat=data.frame(new_testset,preval)
ptestset=data.frame(testset,new_testset)
ptestset1=data.frame(ptestset,preval)
ptestset2=ptestset1[ptestset1$new_testset!=ptestset1$preval,]
require(ggplot2)
ggplot()+

geom_point(data=ptestset,mapping=aes(x=rating,y=1:1208,colour=new_testset))+
  labs(x="Rating",y="Index",colour="Above or Below")+
  geom_point(data=ptestset1,mapping =
aes(x=rating,y=1:1208,colour=preval),shape=2)+
  geom_vline(xintercept = Mean2,linetype="dashed",color="red")

# Linear Regression
df1 <- data.frame(cbind(certificate_group, genre_group, data$title,
data$year, data$runtime_min, data$rating, data$votes, data$director_star))
colnames(df1) <- c("Certificate", "Genre", "Title", "Year", "Runtime_min",
"Rating", "Votes", "Director_star")
attach(df1)
df2<-df1[!(df1$Certificate=="Not_rated"),]
df2<-df2[,-8]
df2<-df2[,-4]
df2<-df2[,-3]
attach(df2)
df2$Runtime_min <- as.numeric(df2$Runtime_min)
df2$Votes <- as.numeric(df2$Votes)
df2$Rating<- as.numeric(df2$Rating)
attach(df2)

df1$Genre = relevel(factor(Genre), ref = "No")
model1 <- lm(Vote~Genre, data = df1) # Model 1
summary(model1)
plot(model1)

df1$Certificate = relevel(factor(Certificate), ref = "Not_rated")
model2 <- lm(Vote~Certificate, data = df1) # Model 2
summary(model2)
plot(model2)

model3 <- lm(Vote~Genre + Certificate + min, data = df1)
summary(model3)
plot(model3)
```

```
df2 <- df1[-c(21, 27, 88, 85,59, 119, 1515),] # Outliers removed

df2$Genre = relevel(factor(df2$Genre), ref = "No")
model7 <- lm(log1p(Vote)~Genre, data = df2) # Post-Transform Model 1
summary(model7)
plot(model7)

df2$Certificate = relevel(factor(df2$Certificate), ref = "Not_rated")
model8 <- lm(log1p(Vote)~Certificate, data = df2) # Post-Transform Model 2
summary(model8)
plot(model8)

model9 <- lm(log1p(Vote)~Genre + Certificate + log1p(min), data = df2) #
Post-Transform Model 3
summary(model9)
plot(model9)

library(lmtest)
df2$rating <- as.numeric(df2$Rating)
model20 <- lm(Votes~Rating, data = df2) # Model 4
bptest(model20)
model11 <- lm(log1p(Vote)~log1p(rating), data = df2) # Post-Transform Model 4
bptest(model11)
plot(model11)
summary(model11)
```