Predicting a Movie's Revenue

Melissa Marsh and Kai Hartley

April 17, 2020

Introduction:

The oldest surviving film is *Roundhay Garden Scene* from 1888; a 2.11 second long film showing the director's family walking in a garden. Film has come a long way from where it started over 100 years ago. When it first started, it was mostly for the elite who could afford to attend. Nowadays, going to the movies is a popular pastime across the world and, with increased interest comes increased money.

Movies can reach larger audiences then ever before, and producers are continually trying to create new movies that will generate a large amount of revenue. The question remains, can you predict how well a movie will be received before it is released? This compilation of work will attempt to answer the question a hand and use linear regression to predict a movie's revenue.

Initial Model:

The dataset for our modelling was collected from TMDB. In this dataset, we are trying to determine the effects of multiple prediction variables on a movie's revenue. The full model includes eight prediction variables. All variables are described in *Table 1* below.

Variable	**Description**
Revenue	Amount of revenue generated by movie in US dollars adjusted to 2010 inflation
Budget	Budget of movie in US dollars adjusted to 2010 inflation
Popularity	Popularity score determined by TMDB
Runtime	Length of movie in minutes
Vote Average	Average voting score given by TMDB users on a scale from 1-10
Vote Count	Number of votes received by TMDB users
Genre	Primary genre of the movie (1=Action, 2=Adventure, 3=Animation, 4=Comedy, 5=Crime, 6=Documentary, 7=Drama, 8=Family, 9=Fantasy, 10=Foreign, 11=History, 12=Horror, 13=Music, 14=Mystery, 15=Romance, 16=Science Fiction, 17=Thriller, 18=War, 19=Western
English	Whether or not the original language for the movie was English (1=English 0=not English)
US Production	Whether or not the country of the primary production company was the United States (1=US production, 0=not US production)

Table 1: Description of variables in TMDB dataset.

Before fitting a model using OLS regression, we first examined the distribution of the data by creating the scatterplots of revenue vs. the quantitative variables budget, popularity, runtime, vote average, and vote count. These scatterplots are shown in *Figure 1*. The scatterplots show a potential linear relationship between the quantitative variables and revenue; however, the distribution of these data points is not even. We have some points that appear to look like potential influential points or outliers. Although these points exist, their influence can be reduced through a variety of techniques.

We also examined histograms and boxplots to determine the distribution of the data. Not surprisingly, the histograms for the quantitative variables showed non-normality for all variables except for vote average which was distributed normally. Additionally, the variables budget, popularity and vote count had prominent right skew. The boxplots showed a handful of potential outliers.

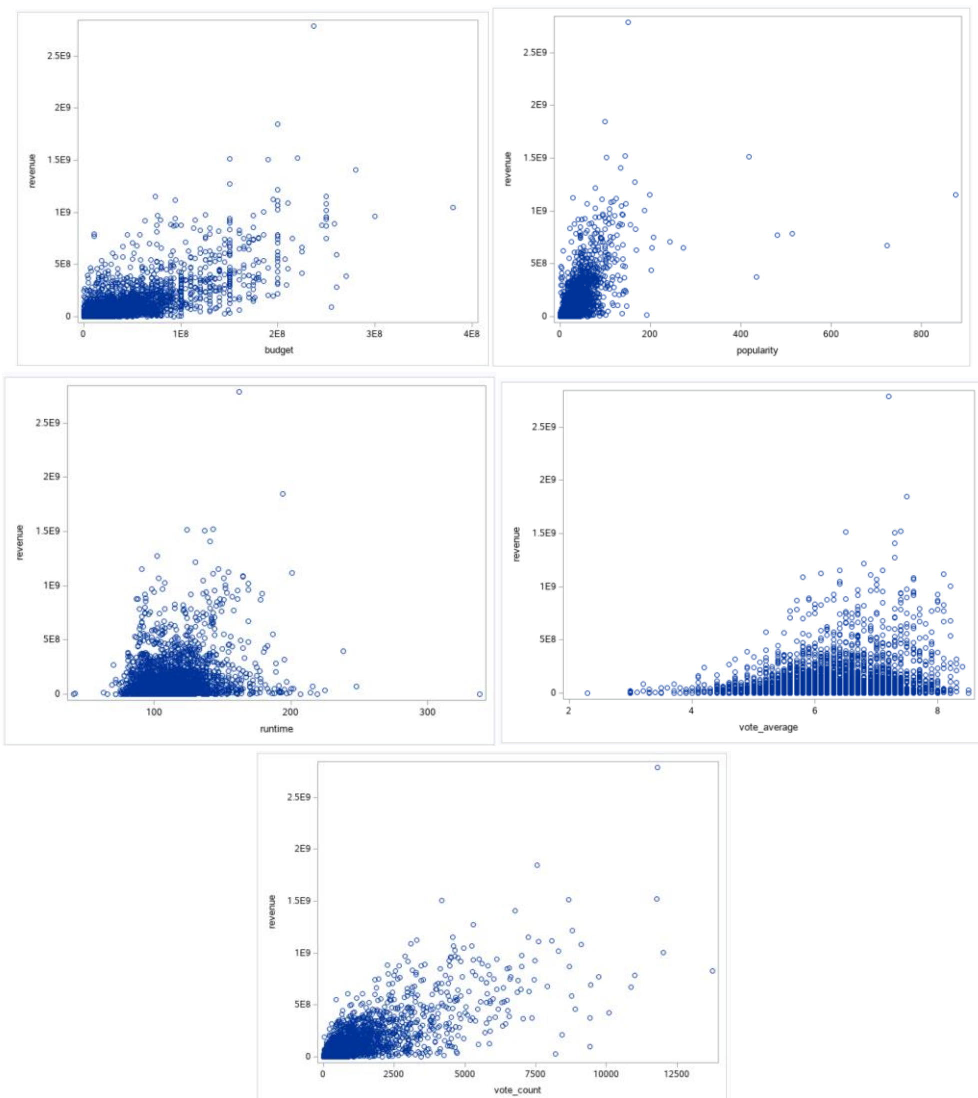

Figure 1: Scatterplots of Revenue vs. Quantitative Variables.

Despite the challenges presented with the original data set, we decided to fit a model using OLS regression. This is due to the fact that the quantitative variables do more or less show a linear relationship with revenue and the issues presented by the outlier points and distribution of the data may be fixed through statistical techniques.

Before fitting the model, we needed to "clean-up" the data. Some values had not transferred over correctly, which was stated in the original data's description, resulting in values equaling zero. This was nonsensical in some cases such as budget equaling zero. These values were removed before fitting the model. Additionally, we split our data into a "train" group and a "test" group so that we could test how well a reduced model would perform on unseen data when compared to the full model.

The initial model did not fit model assumptions, as it was non-normally distributed and had nonconstant variance. To fix this, we examined transformations for revenue in addition to some of the explanatory variables. We ended up performing a fourth root transformation on revenue, a cubed root transformation on budget, a cubed root transformation on popularity and a fourth root transformation on vote count. Other transformations were also attempted but they did not significantly improve the model enough to warrant their inclusion. Shown in *Figure 2* is the QQ plot before and after transformation. Before transformation, the model did not fit the line well and was nonnormally distributed. After transformation, the model more closely fits the line with though the points deviate more towards the ends.

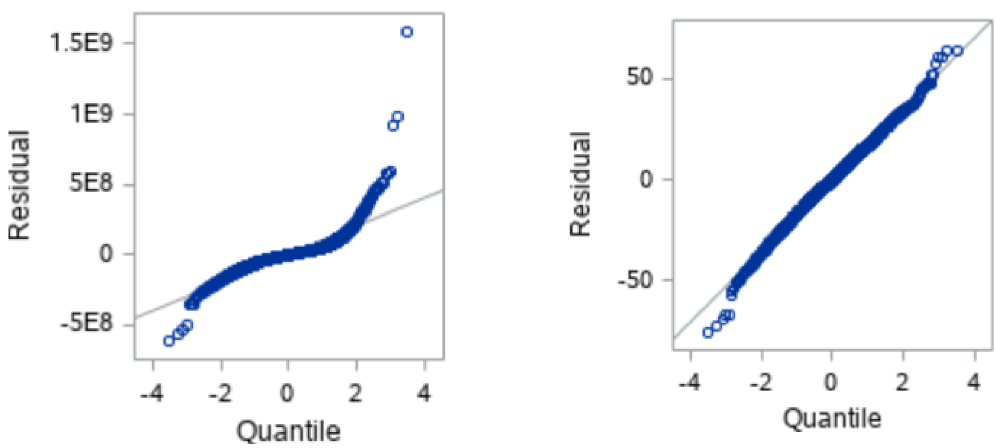

Figure 2: QQ Plots of Initial Model (left) and Transformed Model (right).

After these transformations were completed, our model fit assumptions of normality shown above and constant variance. The BF test had a p-value of 2.8502E-75 before transformations and the BF test had a p-value of 0.099744 after transformation. Included below in *Figure 3* are the residual plots before and after transformations.

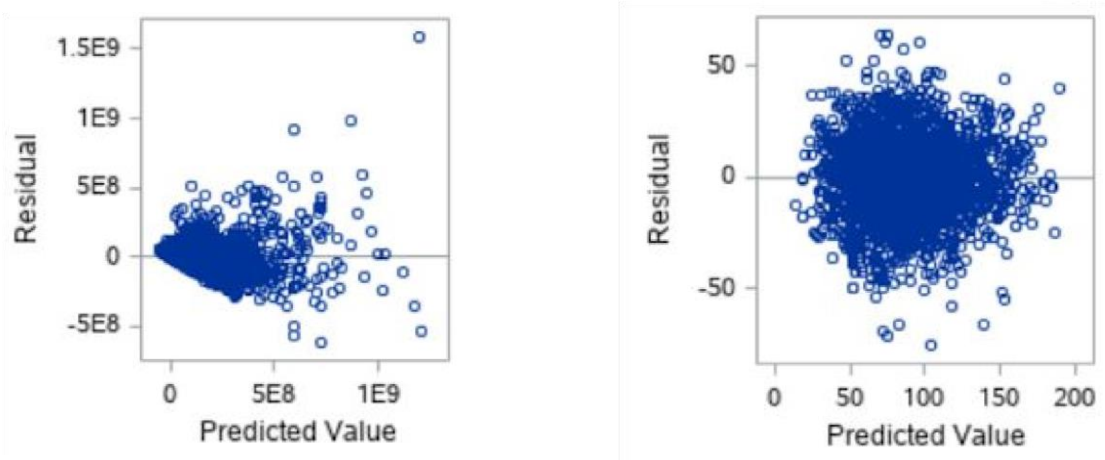

Figure 3: Residual Plots of Initial Model (left) and Transformed Model (right).

Our model now fits the assumptions required for OLS regression, and therefore we can investigate variable selection to reduce the size of our model. It is vital to note that our model does have some issues with multicollinearity due to the variance inflation factors being greater than zero. This is not surprising due to the nature of some of our variables being closely related. Although this may not be ideal, multicollinearity does not cause problems with predictive ability of our model - it only inflates the beta coefficients making them less interpretable. Due to the full model presenting possibly problematic multicollinearity, when choosing a reduced model, we were conscious to reduce the amount of multicollinearity in the quantitative variables. The variance inflation factors for each variable are shown in *Table 2* below.

Variable	**Variance Inflation Factor**
$Budget^{1/3}$	2.15233
$Popularity^{1/3}$	6.29899
Runtime	1.55886
Vote_Average	2.00431
$Vote\_Count^{1/4}$	7.24806
Genre Action	24.04704
Genre Adventure	13.67903
Genre Animation	5.57303
Genre Comedy	25.15483
Genre Crime	7.39798
Genre Documentary	2.00785
Genre Drama	27.85040
Genre Family	2.67122
Genre Fantasy	5.51754
Genre Foreign	1.06352

Genre History	1.88086
Genre Horror	9.46038
Genre Music	1.99833
Genre Mystery	2.17818
Genre Romance	4.34440
Genre Science Fiction	4.26034
Genre Thriller	6.27115
Genre War	1.82280
Genre Western	
Original Language English	1.14340
Original Language Not English	
Not US Production	1.12531
US Production	

Table 2: Variance Inflation Factors of Full Model Variables.

<u>Reduced Model:</u>

To create a reduced model, we used stepwise selection and backwards elimination techniques. We looked at different inclusion levels when stepwise was 0.05 and 0.01. Backwards elimination had exclusion values of 0.05 and 0.01. Stepwise selection and backwards elimination on both 0.01 and 0.05 levels found that budget, runtime, vote count and non US production all were significant so these variables were added to our reduced model. Additionally, backwards elimination and stepwise selection found that different genres were significant. Due to the difference in which genres were significant, and the fact that more than one genre was significant, it was decided to include all levels of genre into our reduced model.

The reduced model was fit using the variables found in *Table 3* below.

Variable	Description	Parameter Estimate	Variance Inflation
	Intercept	-9.88834	0
X_1	$\text{Budget}^{1/3}$	0.11918	1.6674
X_2	Runtime	0.07743	1.34280
X_3	$\text{Vote\_Count}^{1/4}$	10.99158	1.45894
X_4	Genre Action	-0.78443	23.83631
X_5	Genre Adventure	4.32999	13.62297
X_6	Genre Animation	10.58384	5.57046
X_7	Genre Comedy	4.80043	24.96929
X_8	Genre Crime	-5.67921	7.37118
X_9	Genre Documentary	8.09680	2.00059
X_{10}	Genre Drama	-0.86569	27.76485

X_{11}	Genre Family	10.47295	2.66984
X_{12}	Genre Fantasy	2.57804	5.49062
X_{13}	Genre Foreign	-2.95688	1.06090
X_{14}	Genre History	5.26040	1.87773
X_{15}	Genre Horror	5.79456	9.30486
X_{16}	Genre Music	7.89712	1.99360
X_{17}	Genre Mystery	-6.28039	2.17105
X_{18}	Genre Romance	1.96881	4.32930
X_{19}	Genre Science Fiction	-3.55806	4.23662
X_{20}	Genre Thriller	-5.89554	6.21499
X_{21}	Genre War	-8.00001	1.81897
X_{22}	Genre Western		
X_{23}	US_Production = 0	-3.92985	1.04176

Table 3: Variable Description, Parameter Estimate, and VIF for Reduced Model

After choosing which variables to include in our reduced model, we then examined our model to see if it fit model assumptions for OLS regression. *Figure 4* shows the QQ plot of the reduced model. We see that the QQ plot closely follows the line with a few deviations along the ends. This is similar to the QQ plot found for the full model and shows a normal distribution. Additionally, the correlation test of normality found that the model has a value of 0.99837 showing normality assumptions are met.

In addition to meeting assumptions regarding normality, the reduced model also fits assumptions of constant variance. The residual vs. predicted value plot is also shown in *Figure 4*. This plot shows that the residuals do have constant variance although some points fall outside of the large cloud of data. This is expected due to the large dataset. The Brown-Forsythe test confirms that our reduced model does have constant variance with a p-value of 0.072. This p-value is close to the border of being significant, but this expected as large datasets tend to be on the edge of significance due to a few observations affecting the residual plot.

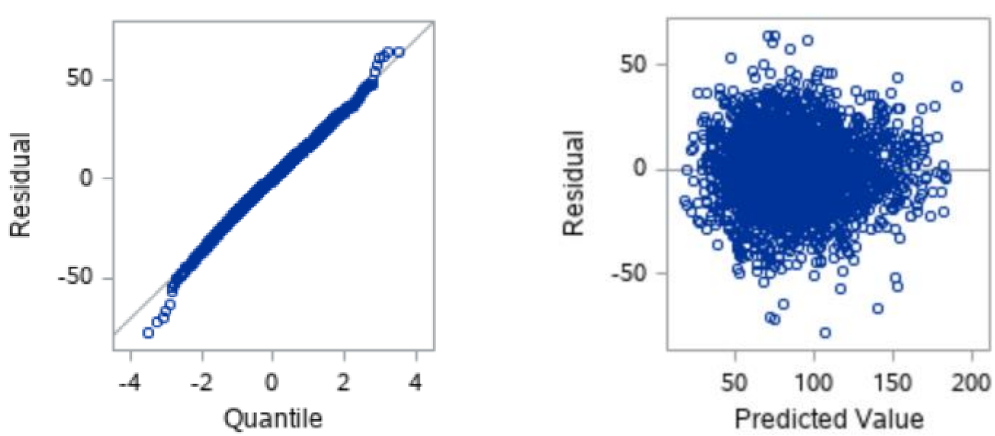

Figure 4: QQ plot for reduced model (left) and Residual plot for reduced model (right).

The final reduced model has the following equation:

$$Revenue^{1/4} = -9.888 + 0.119X_1 + 0.077X_2 + 10.992X_3 - 0.784X_4 + 4.330X_5 + 10.584X_6 + 4.800X_7 - 5.679X_8 + 8.097X_9 - 0.866X_{10} + 10.473X_{11} + 2.578X_{12} - 2.957X_{13} + 5.260X_{14} + 5.795X_{15} + 7.897X_{16} - 6.280X_{17} + 1.969X_{18} - 3.558X_{19} - 5.896X_{20} - 8.000X_{21} + 0X_{22} - 3.930X_{23}$$

In the model we chose to keep every genre in an effort to distinguish which kind of movie is most profitable. We interpret our equation so that any value shown in the reduced linear model with a positive multiplier suggests an increase of the fourth root of revenue, while a negative suggests a decrease. For example, for every unit increase in the runtime, we expect the fourth root of revenue of that movie to increase by 0.077 on average, holding all other variables constant. Also, when a movie is not a US production, we expect that it will decrease revenue by 3.930 on average, holding all other variables constant. For the genre, we consider that a movie can have only one main genre. Each score is based on Western being the dummy variable of the genre, which does not influence the revenue of a movie in our model. Due to the significant amount of multicollinearity associated with the genre variable, we cannot accurately state the affect that a specific genre will have on the fourth root of revenue.

Multicollinearity is present in our final model. The presence of multicollinearity in our model is mostly due to the genre variables having high variance inflation factors. The other quantitative variables – budget, runtime, and vote_count had relatively low variance inflation values which had an average close to 1 as seen in *Table 3*. The variable describing that a movie was not a US production also had a low variance inflation factor close to one. While some of the genre values did have high variance inflation factors all genres were included in our model to make it more interpretable. If we only included the genres that did not have high variance inflation factors, our model would not be easy to interpret and due to the nature of the genre variable, multicollinearity was unavoidable. Even though multicollinearity is present, it does not affect the predictive power of our model. It only makes the coefficients related to genre difficult to interpret with their effect on $revenue^{1/4}$.

When creating this reduced model, we examined possible interaction terms. We examined interaction terms with all the quantitative variables including the following: $X_1 * X_2$, $X_1 * X_3$, $X_2 * X_3$. From this analysis, we found that the interaction term of $X_1 * X_3$ was significant and all other interaction terms were not significant. Due to this values significance, we examined the effect it had on our reduced model. We found that when the interaction term was included in our model, it resulted in a higher MSPR then when our model did not have the interaction term. For this reason, we chose not to include any of the interaction terms in our final model.

Influential points were examined in the reduced model by examining the Cook's D plot and DIFFITS shown in *Figure 5* and *Figure 6*. From these figures we can see that there are a good amount of observations that could be considered influential points. However, due to no one point having a much larger Cook's D or DIFFITS than the others, and the fact that our data set is so large, there is not enough evidence to justify the removal of these influential points. Removing valid values due to their high influence is seen as a last-ditch effort and because our values are not too influential, this is not necessary.

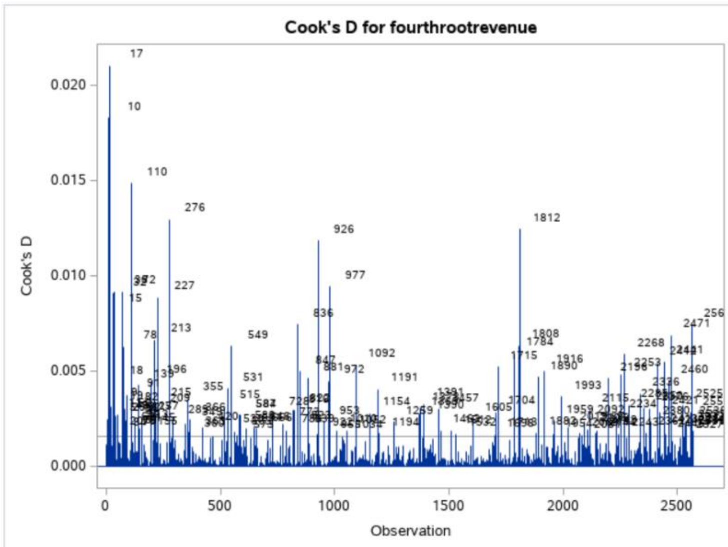

Figure 5: Cook's D plot for reduced model.

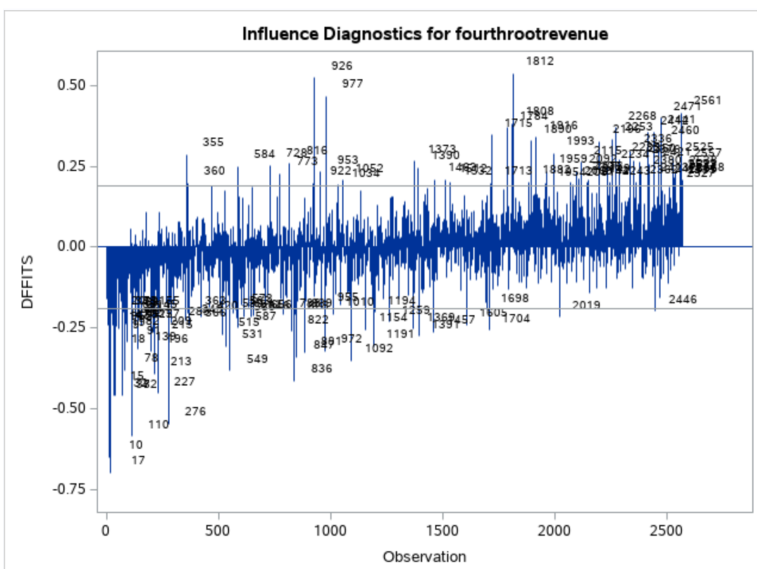

Figure 6: DIFFITS plot for reduced model.

Outliers in the reduced model were examined using the studentized residual plot shown in *Figure 7*. From this plot we can see that there is a good amount of observations that are considered outliers and some that do show to have leverage on our model. Although, these points do exist, they are closely clustered so removal of one would not make sense unless we removed an entire cluster. Also, although these points exist, our model fits the requirements of OLS so we do not have a strong enough reason to remove them.

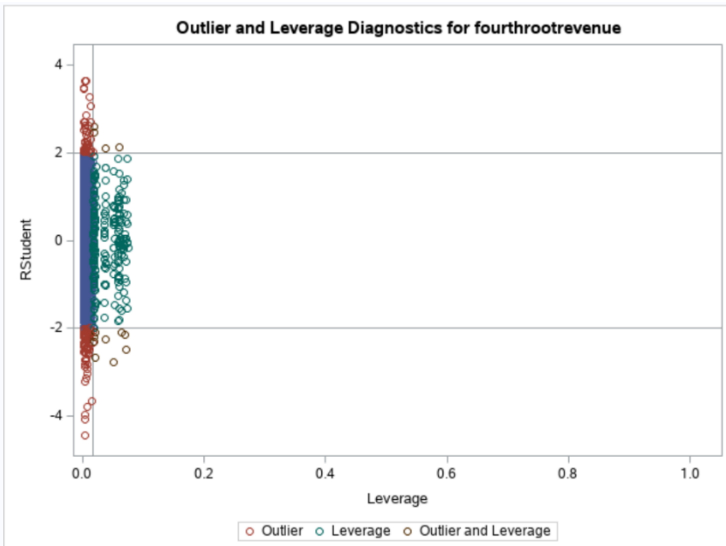

Figure 7: Studentized residual plot for reduced model.

The reduced model was compared to the full model to test how well it would predict new data. This was done by calculating MSPR values using the test set. We found the following results in *Table 4* for the reduced model, full model, and null model (model that only included the intercept). We see that the reduced model has an MSPR value similar to the full model meaning that the predictive power is similar between the two models. Additionally, both the full model and the reduced model outperformed the null model when predicting on new data.

Model	Full Model	Null Model	Reduced Model
MSPR	345.53	1263.46	344.82

Table 4: MSPR for full, null, and reduced models.

Overall, the use of OLS on our dataset was able to look at our model in a structured setting that could easily be interpreted. However, it presented challenges when working with multicollinearity and we are only able to look at two-way interaction terms. To overcome these limitations, we decided to explore the use of a regression tree to model our dataset.

<u>**Alternative Approach:**</u>

As an alternative approach to OLS regression we looked at a regression tree of our data. Regression trees have the benefit of looking at high power interactions that are difficult to model in OLS regression. We are interested in looking at how these high-power interactions can affect the predicted revenue of a movie, so we chose to explore this approach.

In our regression tree we decided to use the fourth root of revenue so that we could more easily compare the regression tree to our linear model, and we could work with smaller values of revenue. Our regression tree suggested 40 nodes; the subset regression tree is shown below in *Figure 8*.

Our regression tree allows us to look at high power interaction terms. We can see that when a movie has a vote count greater than 413, the budget becomes important in determining the revenue. Also, once we look at budget on the second node, we see that a movie with a budget less than $72,200,000 depends on the budget again for determining revenue while if the budget is above 72,200,000 a movie depends on its vote count.

Col6	Vote_count
Col2	Budget
Col3	Popularity

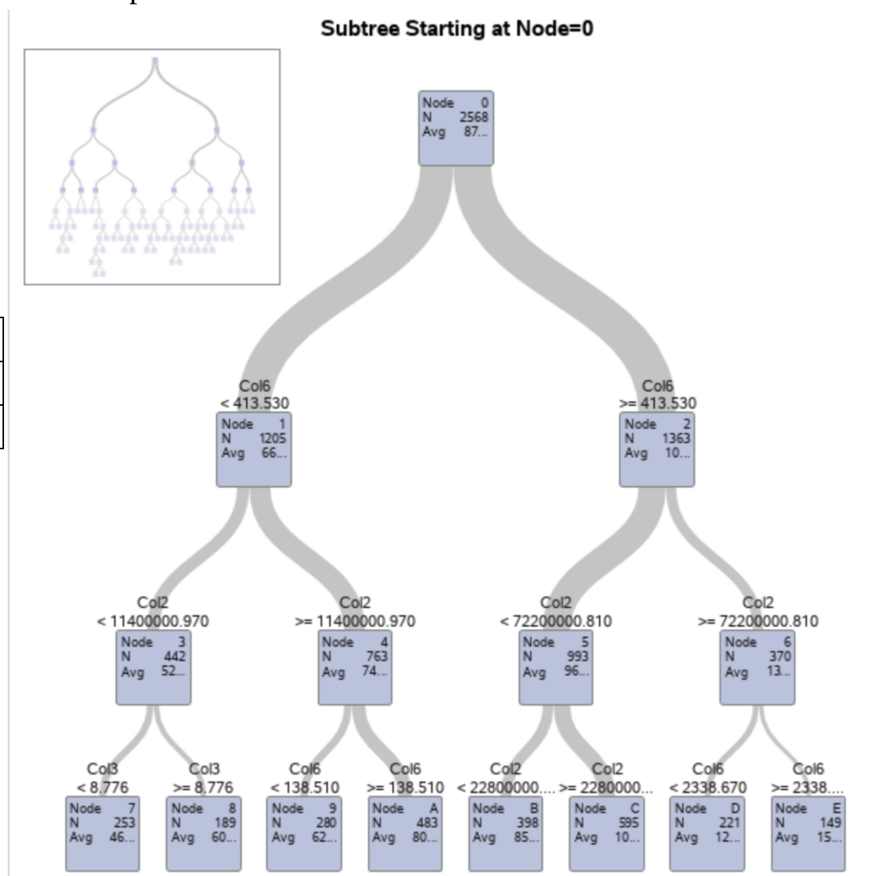

Figure 8: Descriptor of Variables (left) and Regression Subtree Starting at Node = 0 (right).

The relative importance of each rule is shown in the following *Figure 9*. We can interpret this to mean that vote count and budget are the most important variables in our regression tree. This means that as you decide to make a movie, you would select each branch to determine which kind of movie is most profitable, with revenue increasing as you proceed further down the tree and to the right.

Variable Importance				
		Training		
Variable	Variable Label	Relative	Importance	Count
Col6	vote_count	1.0000	1160.8	12
Col2	budget	0.7215	837.5	13
Col5	vote_average	0.1353	157.0	8
Col3	popularity	0.1281	148.8	1
Col4	runtime	0.0918	106.5	4
Col8	genre Adventure	0.0715	82.9700	1
Col27	us_production 0	0.0713	82.7363	1
Col10	genre Comedy	0.0673	78.1470	1
Col7	genre Action	0.0450	52.1827	1
Col21	genre Romance	0.0446	51.7872	1
Col11	genre Crime	0.0431	50.0126	1
Col24	genre War	0.0342	39.6791	1

Figure 9: Variable importance from Regression Tree

Regression trees are useful to look at high power interactions while linear regression models can easily be interpreted. Both these models have advantages and disadvantages. In terms of our data, we examined how the regression tree performed in comparison to our OLS model. The summary of this is found in *Table 5* below. In this table we see that our reduced model had an MSPR of 344.82 while our regression tree had a value of 422.07. This shows that our regression tree was not able to better predict on new data as our reduced model was so it is not preferable to linear regression in the case of our dataset.

Model	Reduced Model	Regression Tree
MSPR	344.82	422.07

Table 5: Comparison of Reduced Model and Regression Tree.

Conclusion:

Predicting a movie's revenue is of important interest to producers and directors alike. In this work we examined how revenue for a movie could be predicted using a few explanatory variables with linear regression and an alternative approach of a regression tree.

Using these techniques, we found that longer movies with a higher vote count and a higher budget provide the largest revenue. Our tree model shows that vote count and budget are the most important variables, so this further confirms our OLS model.

In the future, this data could be used to find cultural trends in movie success over time, if the data was measured season by season. Also, this data could model expected revenues of new movies to help determine the best budget to maximize potential profit.

To improve the validity of our model we would want to look a more possible explanatory variables as we were limited to only examining a few. Also, although our linear regression model did outperform the regression tree, examining a more in-depth neural network would be advantageous in the case of this data. A neural network could look at high power interaction terms with many different combinations and could have better predictive power than simple OLS regression. OLS regression is a powerful tool because it can easily be interpreted but it does have many limitations. Predicting the success of a movie is a complicated problem due to the influence of many different factors.

Appendix

```
/* This first line of code will need to be changed */
FILENAME REFFILE '/home/u45031672/my_courses/STAT 5100/Final
Project/melissa_movies_update_edited.csv';
PROC IMPORT DATAFILE=REFFILE replace
	DBMS=CSV
	OUT=WORK.melissa_movies_update_edited;
	GETNAMES=YES;
RUN;
/*Examine Scatterplots, Boxplots and Histograms for quantitative variables.
proc sgplot data=melissa_movies_update_edited;
	scatter x=budget y=revenue;
run;
proc univariate data=melissa_movies_update_edited nonprint;
histogram budget;
run;
proc sgplot data=melissa_movies_update_edited;
	vbox budget;
run;
proc sgplot data=melissa_movies_update_edited;
	scatter x=popularity y=revenue;
run;
proc univariate data=melissa_movies_update_edited nonprint;
histogram popularity;
run;
proc sgplot data=melissa_movies_update_edited;
	vbox popularity;
run;
proc sgplot data=melissa_movies_update_edited;
	scatter x=runtime y=revenue;
run;
proc univariate data=melissa_movies_update_edited nonprint;
histogram runtime;
run;
proc sgplot data=melissa_movies_update_edited;
	vbox runtime;
run;
proc sgplot data=melissa_movies_update_edited;
	scatter x=vote_average y=revenue;
run;
```

```
proc sgplot data=melissa_movies_update_edited;
	vbox vote_average;
run;
proc univariate data=melissa_movies_update_edited nonprint;
histogram vote_average;
run;
proc sgplot data=melissa_movies_update_edited;
	scatter x=vote_count y=revenue;
run;
proc univariate data=melissa_movies_update_edited nonprint;
histogram vote_count;
run;
proc sgplot data=melissa_movies_update_edited;
	vbox vote_count;
run;
data melissa_movies_update_edited; set melissa_movies_update_edited;
if budget in (0) then delete;
/*Remove if vote_average or vote_count equal zero to do variable tranfromation*/
if vote_average in (0) then delete;
if vote_count in (0) then delete;
run;
proc glmmod data=melissa_movies_update_edited outdesign=GLMDesign outparm=GLMParm
NOPRINT;
	class  release_date genre us_production;
	model revenue=budget popularity runtime vote_average vote_count genre english
us_production;
run;

/* Separate Into Training and Test Sets.
Only Fit Models to the Training Set. The variable
"Selected" separates training (0) from test (1) */
proc surveyselect data=GLMDesign seed=12345 out=movie
    rate=0.2 outall; /* Withold 20% for validation */
run;
data train; set movie;
if Selected = 0;
run;
data test; set movie;
if Selected = 1;
run;
```

```
/*Crude Regression Model*/
proc reg data=train
 plots =(CooksD RStudentByLeverage DFFITS DFBETAS);
 model revenue = COL1-COL28/vif;
 output out=out0 r=resid p=pred;
store regModel;
run;
%resid_num_diag(dataset=out0, datavar=resid, label ='Residual',
predvar=pred, predlabel = 'Predicted Value Initial Model');
run;
/*Transformation for each variable*/
/*COL2 lambda equals 0.35*/
proc transreg data=train;
	model boxcox(COL2/lambda=-0.6 to 0.6 by 0.05)
		=identity(revenue);
	title1 'Box-Cox Transformation';
run;
/*COL3 lambda equals 0.3*/
proc transreg data=train;
	model boxcox(COL3/lambda=-0.6 to 0.6 by 0.05)
		=identity(revenue);
	title1 'Box-Cox Transformation';
run;
/*COL4 lambda equals -0.65*/
proc transreg data=train;
	model boxcox(COL4/lambda=-2 to 2 by 0.05)
		=identity(revenue);
	title1 'Box-Cox Transformation';
run;
/*COL5 lambda equals 1.85 This tranfromation is not included as it doesn't make sense*/
proc transreg data=train;
	model boxcox(COL5/lambda=-2 to 2 by 0.05)
		=identity(revenue);
	title1 'Box-Cox Transformation';
run;
/*COL6 lambda equals 0.25*/
proc transreg data=train;
	model boxcox(COL6/lambda=-0.6 to 0.6 by 0.05)
		=identity(revenue);
```

```
        title1 'Box-Cox Transformation';
run;

/*Fit data using interpretable transformations that have significant effect*/
proc transreg data=train;
        model boxcox(revenue/lambda=-0.2 to 0.4 by 0.05)
                =identity(COL1 cubedrootCOL2 cubedrootCOL3 COL4 COL5 fourthrootCOL6
                COL7-COL28);
        title1 'Box-Cox Transformation';
run;
data train; set train;
        fourthrootrevenue = (revenue)**(1/4);
        cubedrootCOL2 = (COL2)**(1/3);
        cubedrootCOL3 = (COL3)**(1/3);
        fourthrootCOL6 = (COL6)**(1/4);
        cubedrootCOL2_fourthrootCOL6 =cubedrootCOL2*fourthrootCOL6;
run;
proc reg data=train plots =(CooksD RStudentByLeverage DFFITS DFBETAS);
        model fourthrootrevenue = cubedrootCOL2 cubedrootCOL3 COL4 COL5
        fourthrootCOL6 COL7-COL28 /vif;
        output out=out6 r=resid p=pred;
        title1 'Simple model for Tranfromed Data';
store intialmodel;
run;
%resid_num_diag(dataset=out6, datavar=resid, label ='Residual',
predvar=pred, predlabel = 'Predicted Value Tranformed');
run;

/*Variable selection*/
/*Stepwise Selection*/
proc reg data=train;
        model fourthrootrevenue = COL1 cubedrootCOL2 cubedrootCOL3 COL4 COL5
        fourthrootCOL6 COL7-COL28
                /selection=stepwise slentry=.05 slstay=.05;
        title1 'Stepwise Selection';
run;
proc reg data=train;
        model fourthrootrevenue = COL1 cubedrootCOL2 cubedrootCOL3 COL4 COL5
        fourthrootCOL6 COL7-COL28
                /selection=stepwise slentry=.01 slstay=.01;
```

```
        title1 'Stepwise Selection';
run;

/*Backwards Elimination*/
proc reg data=train;
        model fourthrootrevenue = COL1 cubedrootCOL2 cubedrootCOL3 COL4 COL5
        fourthrootCOL6 COL7-COL28
                /selection=backward slstay=0.05;
        title1 'Backward Elimination';
run;
proc glmselect data=train plots=(criterion ase);
        model fourthrootrevenue = COL1 cubedrootCOL2 cubedrootCOL3 COL4 COL5
        fourthrootCOL6 COL7-COL28 /
                selection=backward slstay=0.5;
        title1 'Backwards Variable Selection';
run;

/*Model with variable selection*/
proc reg data=train plots (label) =(CooksD RStudentByLeverage DFFITS DFBETAS);
        model fourthrootrevenue = cubedrootCOL2 COL4 fourthrootCOL6 COL7-COL24
        COL27 /vif;
        output out=out6 r=resid p=pred;
        title1 'Simple model for Reduced variables'
store mymodel6;
run;
%resid_num_diag(dataset=out6, datavar=resid, label ='Residual',
predvar=pred, predlabel = 'Predicted Value Reduced');
run;
/*****Look at interaction terms******/
/**BUDGET and VOTE COUNT**/
/* Define higher-order predictors */
data train; set train;
cubedrootCOL2_fourthrootCOL6 =cubedrootCOL2*fourthrootCOL6;
sv2 = cubedrootCOL2**2;
fr2 = fourthrootCOL6**2;
run;
proc reg data=train;
model fourthrootrevenue = cubedrootCOL2 fourthrootCOL6 cubedrootCOL2_fourthrootCOL6
/vif;
        title1 "Interaction model Budget and Vote Count";
```

```
run;
proc reg data=train;
        model fourthrootrevenue = cubedrootCOL2 fourthrootCOL6
cubedrootCOL2_fourthrootCOL6 sv2 fr2 /vif;
        highercheck: test cubedrootCOL2_fourthrootCOL6=sv2=fr2=0;
        title1 'Check for higher-order predictors Budget and Vote Count';
run;
proc reg data=train;
        model fourthrootrevenue = cubedrootCOL2 fourthrootCOL6/vif;
        title1 'Lower-order model';
run;
/* Now look at higher-order variables with standardized data */
proc stdize data=train out=std_train
        method=std mult=.0197372692;
run; /* Note that mult = 1/sqrt(n-1) */
data std_train; set std_train;
cubedrootCOL2_fourthrootCOL6 =cubedrootCOL2*fourthrootCOL6;
  sv2 = cubedrootCOL2**2;
  fr2 = fourthrootCOL6**2;
run;
proc reg data=std_train;
        model fourthrootrevenue = cubedrootCOL2 fourthrootCOL6
cubedrootCOL2_fourthrootCOL6 / vif;
        title1 'Check for interaction (standardized scale) Budget and Vote Count';
run;
proc reg data=std_train;
model fourthrootrevenue = cubedrootCOL2 fourthrootCOL6 cubedrootCOL2_fourthrootCOL6
sv2 fr2 /vif;
highercheck: test cubedrootCOL2_fourthrootCOL6=sv2=fr2=0;
title1 'Check for higher-order predictors (standardized scale) Budget and Vote Count';
run;

/*Model with variable selection AND interaction term*/
proc reg data=train plots =(CooksD RStudentByLeverage DFFITS DFBETAS);
        model fourthrootrevenue = COL1 cubedrootCOL2 COL4 fourthrootCOL6
cubedrootCOL2_fourthrootCOL6 COL7-COL25 COL27 /vif;
        output out=out13 r=resid p=pred;
        title1 'Simple model for Reduced variables'
store mymodel13;
run;
```

```
%resid_num_diag(dataset=out13, datavar=resid, label ='Residual',
predvar=pred, predlabel = 'Predicted Value Reduced');
run;

/*Add in transformations to test data to caclulate MSPR*/
data test; set test;
	fourthrootrevenue = (revenue)**(1/4);
	cubedrootCOL2 = (COL2)**(1/3);
	cubedrootCOL3 = (COL3)**(1/3);
	fourthrootCOL6 = (COL6)**(1/4);
	cubedrootCOL2_fourthrootCOL6 =cubedrootCOL2*fourthrootCOL6;
run;

/*MSPR for full model*/
proc plm restore=intialmodel;
 score data=test out=newTest predicted;
 run;
data newTest; set newTest;
MSE = (fourthrootrevenue - Predicted)**2;
run;
proc means data = newTest;
var MSE;
run;

/*****MSPR for null model****/
proc reg data=train
 plots =(CooksD RStudentByLeverage DFFITS DFBETAS);
 model fourthrootrevenue =  ;
 output out=out2 r=resid p=pred;
store modelintercept;
run;

proc plm restore=modelintercept;
 score data=test out=newTest10 predicted;
 run;
data newTest10; set newTest10;
MSE = (fourthrootrevenue - Predicted)**2;
run;
proc means data = newTest10;
var MSE;
```

```
run;

/*MSPR for reduced model NO Interaction*/
proc plm restore=mymodel6;
 score data=test out=newTest predicted;
 run;
data newTest; set newTest;
MSE = (fourthrootrevenue - Predicted)**2;
run;
proc means data = newTest;
var MSE;
run;

/*MSPR for reduced model Yes Interaction*/
proc plm restore=mymodel13;
 score data=test out=newTest predicted;
 run;
data newTest; set newTest;
MSE = (fourthrootrevenue - Predicted)**2;
run;
proc means data = newTest;
var MSE;
run;

/*Regression Tree*/
proc hpsplit data=train seed=123 maxdepth=10 maxbranch=2;
	model fourthrootrevenue=COL1-COL28;
  output out=out20;
  code file='/home/u45031672/my_courses/STAT 5100/Final Project/tree2.sas';
  /* This saves the tree to a file (need to change the path) */
run;
/**Call the test data and include the tree, this will make predictions on the tree */
data scored;
set test;
%include '/home/u45031672/my_courses/STAT 5100/Final Project/tree2.sas';
run;
/* Now calculate the MSPR as we did in OLS */
data testTree;
set scored;
ASE = (fourthrootrevenue - P_fourthrootrevenue)**2;
```

```
run;
proc means data = testTree;
var ASE;
run;
```