The Man Who Loved His Little Girl

A Movie Pitch to Film Executive William Moneybags

Sabrinia Crouch - scrouch@syr.edu

Maria Ng - mng103@syr.edu

Cherngywh (Charlie) Lee - clee61@syr.edu

Timothy Rivers - tarivers@syr.edu

Table of Contents

Project Abstract

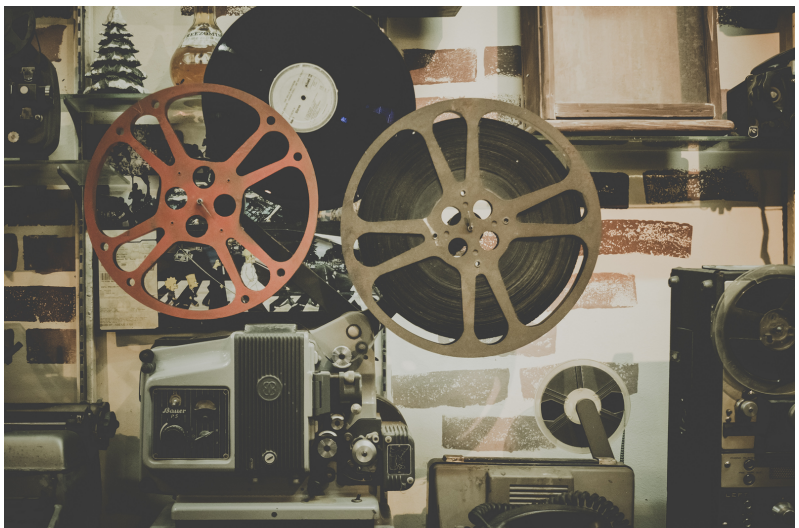
In this project, we would like to encourage a movie producer to invest in our film. Using the Kaggle dataset, "The Movie Dataset", we spent extensive time cleaning and curating the dataset into a workable dataset that would allow us to investigate several potential models to predict revenue. In additional to predicting revenue, we also optimized the key properties of the film such as genre, movie title, release date, runtime and language of release.

For the movie, after assessing which films had the best correlation between revenue and genre, it was determined that both animation and music had the only significant values, but neither genre produced a model with a high correlation or high R2. We produced a word cloud of all of the titles used across all genres to generate a title (The Man Who Loved his Little Girl) that would perform well in our selected genre of music or animation and would be be appealing enough to pull in audiences from other genres.

We also generated a histogram of release dates across all genres and identified that summer time was the best time to release a new movie. Finally, we identified that the top languages for release were English, Chinese, German and Japanese and of these languages, average movie run time was 100 minutes for English, German and Japanese and 125 minutes for Chinese language films.

For predicting revenue, we focussed on four primary attributes in the dataset, budget, popularity, runtime and vote count. After evaluating these attributes in linear, support vector machine and neural net models, we were only able to build a well fitting model using a linear model using vote count to predict revenue. Unfortunately this is not ideal, since vote count would be collected after the movie would be released. Therefore we recommend that this parameter be monitored post release to guide market expansion or contraction.

Based on this analysis, we decided to create an animated movie that tells the story of the Dad from Montana and his daughter who leaves the snow for sunny California. The movie will be released in the summer (May-June) to an English speaking audience with a runtime of 100minutes. We will monitor the early signs of the vote count to extrapolate potential sales and monitor for markets of expansion based on the model.



Contribution Statements

“Sabrinia Crouch: Coordinating the weekly Zoom meetings, organizing the final presentation, generating the dataset and function to generate the text mining, generating the function for the svm and neural net models, building the language and runtime graphs .”

“Maria Ng: Maintaining the Excel metadata, separating multiple genres, sorting dataset to a specific genre (documentary, animation, and musical), data munging, and plotting distribution graphs. Drafting and sending out team update. Generating the codes for the best month to release a movie.”

“Timothy Rivers: Initial data exploration and analyses, regressions, initial SVM and neural network models, data interpretation and exploration, report building"

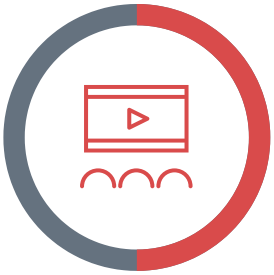
“Cherngywh (Charlie) Lee: Conducting data munging by python, separate the column like company that contains multiple values. Cleaned and Created bag of words and word cloud of titles, overviews and tagline to see the frequency and sentiment analysis.”

The Crazy Saturdays

Named for our weekly discussions

Introduction

The Movie Dataset



100

Average runtime in minutes

Focal attributes

Revenue	Genre
Budget	Release Date
Popularity	Runtime
Vote Count	Original Language
Title	ROI

Introduction

A lucrative movie producer will be in town and our team would like to pitch to him a movie we have designed. Using advanced analytical techniques, we optimized our movie properties to be popular and yield the highest revenue. Following our analysis we found the following:

1. The most popular genres were animation and music
2. Most movies are released in the summer in english with an average runtime of 100 minutes
3. Revenue could not be proactively predicted
4. Revenue could be predicted using vote count, so we recommended that the movie be monitored post release and then revenue could be projected in real time

Based on our data we will create an animated film that tells the story of the antics of a rugged Dad from Montana and his daughter who leaves the snow for sunny California. The film will be released in the summer to an English speaking audience with a runtime of 100-160 minutes. We will monitor early signs of the vote count to extrapolate potential sales and monitor markets for expansion.

The Movie Dataset from Kaggle

The Movies DataSet was used to perform our analysis (https://www.kaggle.com/rounakbanik/the-movies-dataset#movies_metadata.csv). These files contain metadata for all 45,000 movies listed in the Full MovieLens Dataset. The dataset consists of movies released on or before July 2017. Data points include cast, crew, plot keywords, budget, revenue, posters, release dates, languages, production companies, countries, TMDB vote counts and vote averages.This dataset also has files containing 26 million ratings from 270,000 users for all 45,000 movies. Ratings are on a scale of 1-5 and have been obtained from the official GroupLens website.

Data Analysis Methods

The group utilized multiple techniques throughout the presentation. The data analysis was divided into two primary groups: 1) Data analysis to support the movie design and 2) Data modelling to predict revenue.

Data Analysis to support the movie design

Linear model-In order to narrow in which genre we should make our movie, we generated a linear model to try to predict which genre was most closely linked to revenue. Linear models use linear regression to predict the value of a variable, y (revenue), based on one or more input variables, x. Using the regression, we hoped to generate a mathematical equation that we could use to predict which genre would be most closely linked to revenue. The summary of statistics would include a F-statistic (goodness of fit of model) in which we would look for a p-value <0.05. Each variable will have a coefficient which measures the relationship between X and Y, or measures the intercept. We would also look at the individual genres of significance (t-statistic p-value <0.05) indicating that the X variable is a significant term in the model (slope is significant).

Once the genre was selected, association rule mining was performed on the dataset to determine if "animation" and "music" appeared together as designated genres. Association rule mining is a common technique used to find associations between many variables. Association rule mining is often used when looking for relationships between items (i.e when a shopper buys milk they also buy bread). The association rule mining is performed by installing the "arules" package and then transposing a matrix dataset into a transactional dataset. Transforming to a transactional dataset allows the user to convert continuous variables into discrete variables. Once the datasets are converted to transactions, the "apriori" function can be employed to calculate the confidence, support and lift. Confidence is an indication of how often the rule has been found to be true. Support is an indication of how frequently the itemset appears in the dataset. Lift is the factor by which our two variables (music and animation)exceeds the expected probability of those same variables occuring together. Therefore the higher the lift, the higher the chance our variables (music and animation) will occur together.

Histograms of release date and language were generated to understand what time of year would be the best time to release the film and in what language. A histogram is a graphical representation of the frequency of a variable, x. The histograms are generated with the package , ggplot, with the function ((geom_bar())). For the release date, frequencies were binned by month, regardless of year of release. For the language, due to the large number of languages represented, the survey was narrowed to the top 20 languages with the highest frequencies. Languages were represented using the ISO language codes.

Violin plots were generated for the runtime of the top ten languages. Violin plots are similar to box plots, but instead of boxes, data is shown as density plots. In our violin plots, the runtime is represented in minutes with a marker for the median and boxes for the quartiles. The violin plots are generated using ggplot with the function (geom_violin()).

In order to identify a title that would resonate with all movie goers regardless of genre, a word cloud was generated for all of the titles in the movie dataset. Using the TM and wordcloud packages, matrices of the title words were designed and then the frequencies of each of these words was calculated. Then using the wordcloud function, these words were graphed with the relative size of the word corresponding to the magnitude of the frequency.

Data Analysis to support the movie design

The original dataset only provided "budget" and "revenue" as values. However, intuitively, it seems more appropriate to calculate success based on return of investment (ROI) rather than "budget" or "revenue" alone.

$$ROI = \frac{(revenue - budget)}{budget}$$

Histograms of these three parameters were generated using R-basic and it was observed that there was a disproportionate right-handed skewness for "budget" and "revenue" resulting in calculated ROI being substantially right skewed. As a consequence, the ROI was difficult to predict due to the narrow swath of calculated results. Therefore the team decided to focus on "revenue" as our parameter to optimize.

Before deciding what parameters on which to build models, four parameters budget, popularity, vote count and runtime were plotted using a scatter plot matrix. Budget and runtime were two prospective parameters that were used to design models to predict revenue before release of the film. If models were built using popularity or vote-count, these would be used as retrospective predictors or revenue and would only be used to guide marketing.

The scatter plot matrix allows the team to visualize potential relationships (i.e linear, logistic, power, etc) with reference to the dependent variable, "revenue". The scatterplot matrix was generated using "lattice" and "car" packages with the function, scatterplotMatrix. Using the function we were able to visualize the data point distribution and include a linear fit trend line. For parameters with a strong linear correlation, the data points will align with the trend-line.

Linear models- Following analysis with the scatterplot matrix, linear models were built using all of the parameters as main effects and as moderating effects. Much like the linear models for genre, F-statistic, t-statistic and R2 were evaluated for significance and goodness of fit. Similarly to the linear models for genre, using all parameters resulted in very poor model fit. As a follow up, linear models were also generated for individual main effect variables, such a budget. Unfortunately these models were also a poor predictor of revenue.

SVM-Since linear models were not predictive of revenue, support vector machine (SVM) models were generated. Although traditionally used for classification models, we attempted to use the SVM for continuous data. In the classification model, SVM clusters the data based on properties or attributes on different hyperplanes. For the continuous data, we generated a SVM regression in which the data is clustered similarly to a classification model but then a linear function is applied to that data in the non-linear space which theoretically corresponds to the nonlinear function in the original space. This is achieved primarily by clustering the data into kernels (KSVM).

To build the model, the dataset is divided into training and test sets. Then using the resultant model the predicted values are compared to the original data and a Root Mean Square Error (RMSE) is calculated for each predicted value. Ideally, the RMSE should be small and evenly distributed. However in our model, although the training model error is low, the predicted RMSE is large and uniform across the dataset.

$$\mathrm{RMSE} = \sqrt{\frac{\sum_{i=1}^{n} (P_i - O_i)^2}{n}}$$

In a final effort to identify a prospective parameter in which to build a model, neural net regressions were performed with multiple combinations of nodes and layers. Neural nets utilize weight optimizations to target optima to generate a regression similar to a linear regression for each node with the weights being the "coefficients" of the model. The models can increase in complexity by adding more nodes within a layer and by adding additional hidden layers within the model. These additional layers are then further optimized using the previous layer as the optimized variables using the same weighting scheme as the first layer. This iterative optimization continues until a single value is identified. The challenge with neural nets is that it takes several iterations of the model to minimize the error. Additionally because the model is optimizing on a maximum value, this maxima maybe a local maxima and not a global maxima, so the model can predict very different value from run to run. When evaluating a neural net model, the modeller should minimize the error to as small as possible. Most well predictive simple models will have an error <20

Analysis Summary

Several linear models were generated to identify which genres were best correlated with revenue. Our first model was a linear model including all genres as factors without moderating effects (Appendix pg.19).

```
lm(formula = revenue ~ Action + Adventure
+ Animation + Comedy + Crime +
Documentary + Drama + Family + Fantasy +
Foreign + History + Horror + Music +
Mystery + Romance + Science.Fiction +
Thriller + TV + Western, data = genre)
```

This model was well predictive with a F-statistic of 7.537e-11 and showed that the intercept, action, adventure, animation, horror and music were significant variables. However, the adjusted R2 indicated that there was only a 4% confidence (R2 = 0.04428) in predicting revenue on these significant variables. Further examination showed that animation and music were most significant, so a new model was generated using only these parameters. (Appendix pg.19)

The new linear model with revenue as a function of music and animation was predictive with a F-statistic of 7.964e-14 and both variables were highly significant with t-statistic p-values of 0.0000000130 and 0.0000000297 for animation and music, respectively. However, similarly to the model containing all genres as independent variables, the music/animation model has a low R2 of 0.03684 suggesting that the model is only able to predict about 3% of the variation seen in model. Additionally the variables were checked for collinearity (VIF) and were determined to not be collinear.

```
vif(LinearModel.2)  Animation     Music
                    1.001299    1.001299
```

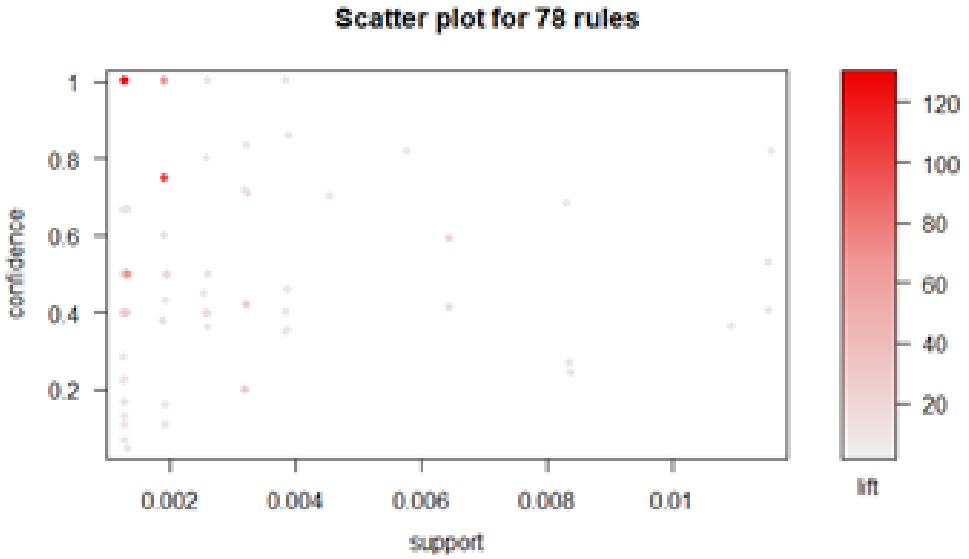
Based on these two models, although the R-squares were low, both models indicated that these genres were significant and correlated. Therefore we selected animation as our genre for our feature film.

Associated Text Mining

In order to determine if “animation” and “music” appeared together frequently and could potentially skew the results of the linear model, the complete dataset was evaluated for incidences were the variables occur together using associated text mining. Using a cutoff of lift >5, the function generated 78 rules. (Appendix pg.20)

```
goodrules_md_TDM1 <- aruled_md_TDM1[quality(rules_md_TDM1)$lift >5]
```

In order to determine if “animation” and “music” appeared together frequently and could potentially skew the results of the linear model, the complete dataset was evaluated for incidences were the variables occur together using associated text mining. Using a cutoff of lift >5, the function generated 78 rules.

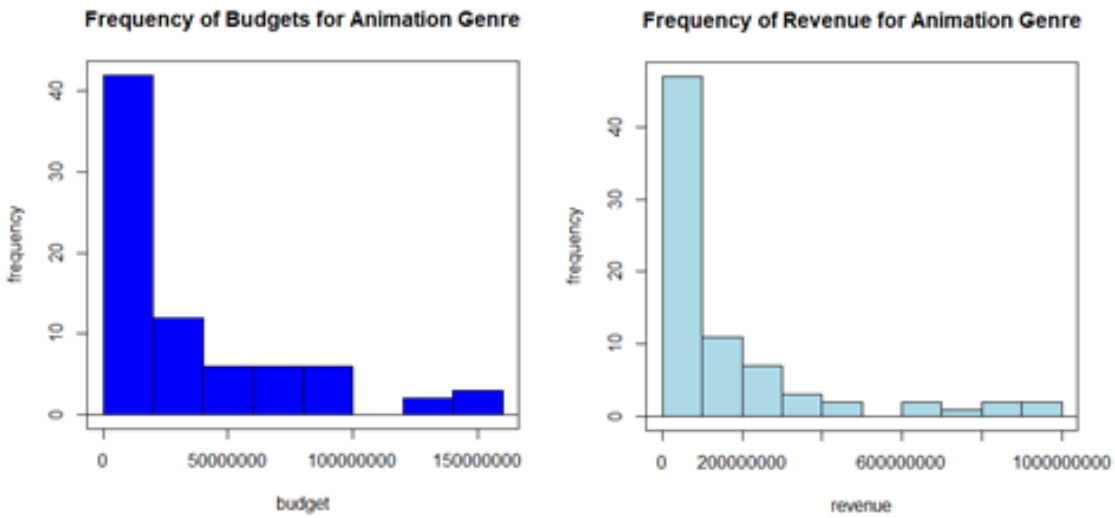


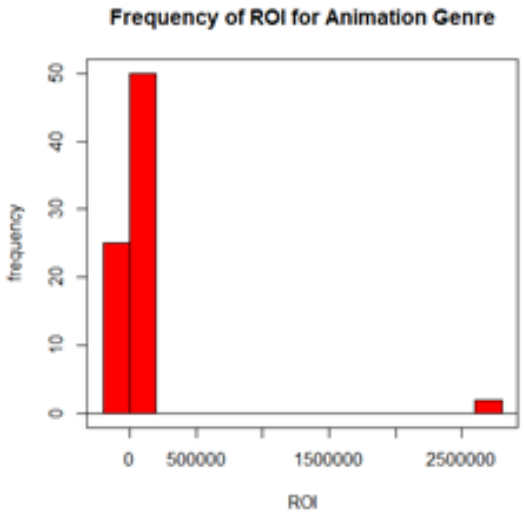
	lhs		rhs	support	confidence	lift	count
[1]	{Thriller}	=>	{Drama}	0.003211304	0.41666667	25.950000	5
[2]	{Drama}	=>	{Thriller}	0.003211304	0.20000000	25.950000	5
[3]	{Thriller}	=>	{Foreign}	0.001284522	0.16666667	8.370968	2
[4]	{Foreign}	=>	{Thriller}	0.001284522	0.06451613	8.370968	2
[5]	{History}	=>	{War}	0.008349390	0.24074074	7.809028	13
[6]	{War}	=>	{History}	0.008349390	0.27083333	7.809028	13
[7]	{Horror, Thriller}	=>	{Drama}	0.001284522	0.40000000	24.912000	2
[8]	{Drama, Horror}	=>	{Thriller}	0.001284522	1.00000000	129.750000	2
[9]	{Crime, Thriller}	=>	{Drama}	0.001926782	1.00000000	62.280000	3
[10]	{Crime, Drama}	=>	{Thriller}	0.001926782	0.75000000	97.312500	3
[11]	{Action, Thriller}	=>	{Drama}	0.001284522	0.40000000	24.912000	2
[12]	{Action, Drama}	=>	{Thriller}	0.001284522	0.50000000	64.875000	2
[13]	{Science.Fiction, Thriller}	=>	{Adventure}	0.001284522	1.00000000	7.344340	2

Top 13 Rules of the 78 Total Rules

Budget, Revenue and ROI

Observed large differences in the data in how lower values are reported for budget and revenue. These differences skews the ROI calculation. Future projections were made on revenue. (Appendix pg.21)



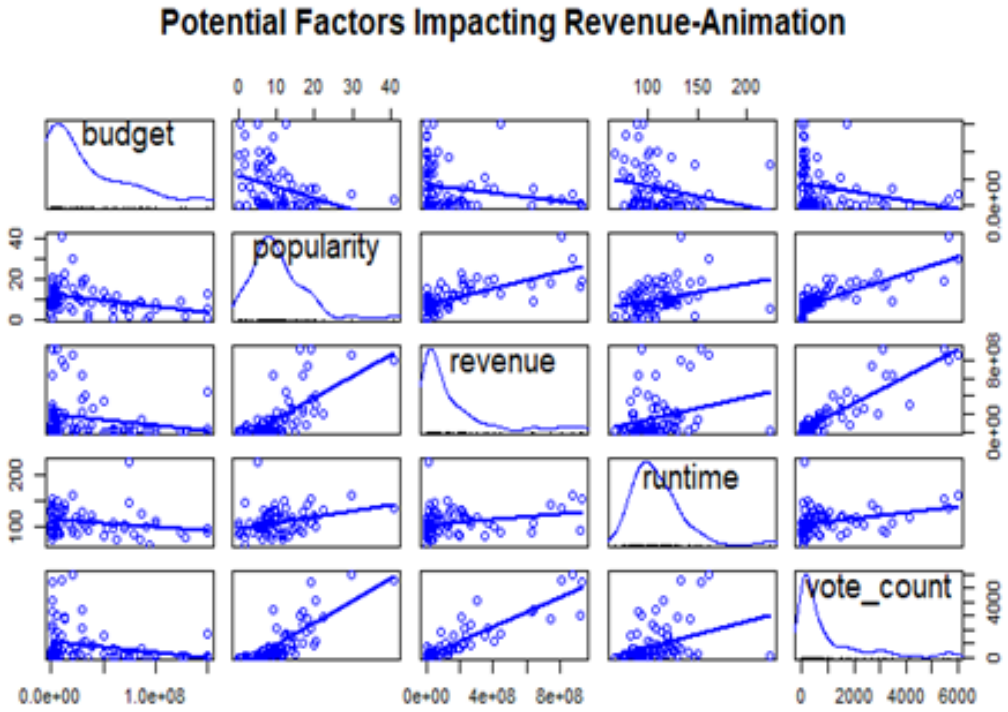


Histogram of Budget (a), Revenue (b), and ROI (c).

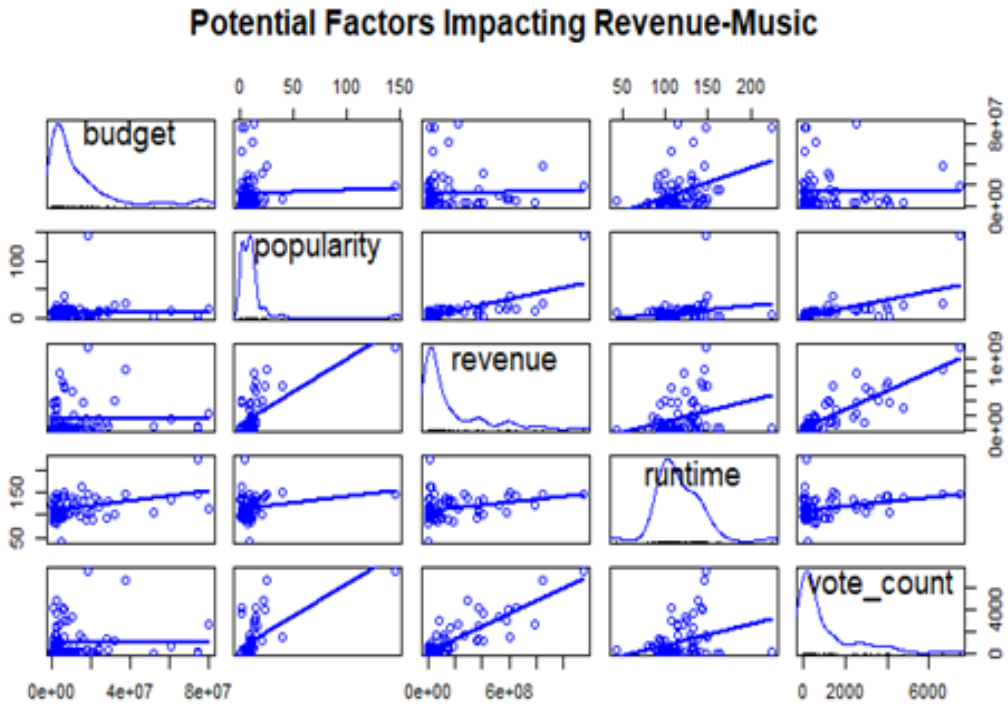
Factors impacting Revenue-Animation and Music

Popularity, runtime and vote count were the three factors that were identified as the most impactful on revenue for animation and music. Popularity incorporates number of votes, views per day, positive reviews, and release date. Runtime shows the length of the film. Vote count represents the number of people who watched the film- good or bad

Scatter plots were generated using animation and music data to looking at our top parameters:budget, popularity, runtime, vote count, and revenue. Both scatter plots showed strong linear correlation between revenue and vote count. Less strong linear correlation was observed with popularity and runtime. (Appendix pg.22)



Animation Scatter Plot



Music Scatter Plot

Models Predicting Revenue

Linear- Linear models were generated for both the animation and music datasets (full datasets that have been queried to include only the genres animation or music). Using the same parameters highlighted in the scatterplot matrix, linear models including full main effects models with and without moderating effects were generated.

Following evaluation of the various linear models, it was clear that neither prospective parameter (budget, runtime) would generate a predictive model. For example, after building a model where revenue is a function of budget, all of the statistics of fitness (F-statistic, t-statistic and R2) were poor or not significant. A similar exercise was performed for the music dataset as well and these models were less predictive.

In order to build a predictive model for revenue, we decided to look at the retrospective parameters of popularity and vote count. These two parameters would not allow the team to predict the revenue prior to the release of the film, but we would be able to monitor the vote count and popularity as the film is released and the revenue could be projected based on early readouts. Unfortunately the model utilizing popularity and vote_count indicated that only vote_count was significant. Therefore a linear model of revenue as a function of vote_count was performed. This final analysis resulted in a good fit (F-statistic p-value of 2.2e-16) where the variable vote_count is significant (t-statistic p-value of <2e-16) with a strong correlation (R2 of 0.8031). The model generated the following predictive equation:

$$\text{Revenue} = (154994 * \text{votecount}) + 13123426$$

In an effort to identify a model that could be prospectively predictive of revenue including runtime or budget, SVM and neural net models were built using the chosen parameters. (Appendix pg.22)

This resulted in the following predicted revenues based on vote count:

Predicted Revenue	Vote count
$ 13,123,436.00	0
$ 168,117,436.00	1000
$ 323,111,436.00	2000
$ 478,105,436.00	3000
$ 633,099,436.00	4000
$ 788,093,436.00	5000
$ 943,087,436.00	6000

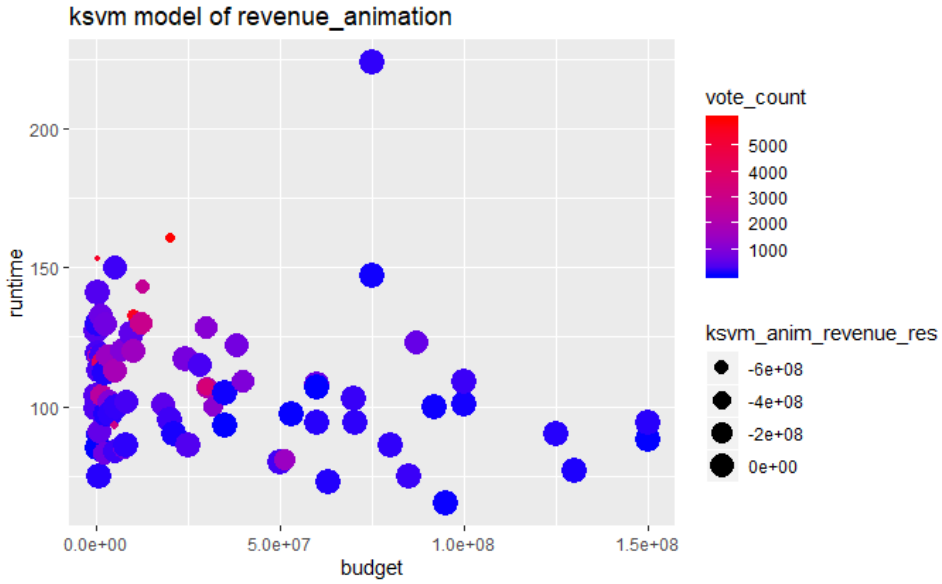
SVM

Our Support Vector Machine (SVM) model was an exploration of the relationship between the budget of the movie, the vote count and the revenue for the type of movie (in this case, animation). What we found is that, like the linear regression, there doesn't appear to be any obvious relationship between these variables.

The model was developed by using the animation dataset and dividing it into training (⅔ dataset) and test (⅓ dataset) sets.

```
Support Vector Machine object of class "ksvm"

SV type: eps-svr  (regression)
 parameter : epsilon = 0.1  cost C = 3

Gaussian Radial Basis kernel function.
 Hyperparameter : sigma =  0.47022135669666

Number of Support Vectors : 40

Objective Function Value : -21.3946
Training error : 0.059256
Cross validation error : 2.815407e+16
Laplace distr. width : 120186909
```

The training set had a low training error of 0.059 and cross validation error of 2.8 e16. We utilized an epsilon support vector regression since we are modelling continuous data with an epsilon of 0.1 and a cost of 3 which resulted in 40 support vectors.



In this graph of the SVM model, the budget is plotted on the x-axis and the runtime is plotted on the y-axis. The coloration of each point is indicative of the vote count, where the closer to red a point is, the more votes it received. The size of each point indicates the root mean squared residuals of the generated revenue. Ideally we would like the residuals to be small. However our residuals are fairly consistent regardless of parameter, suggesting that the underlying model is uniformly unpredictive.

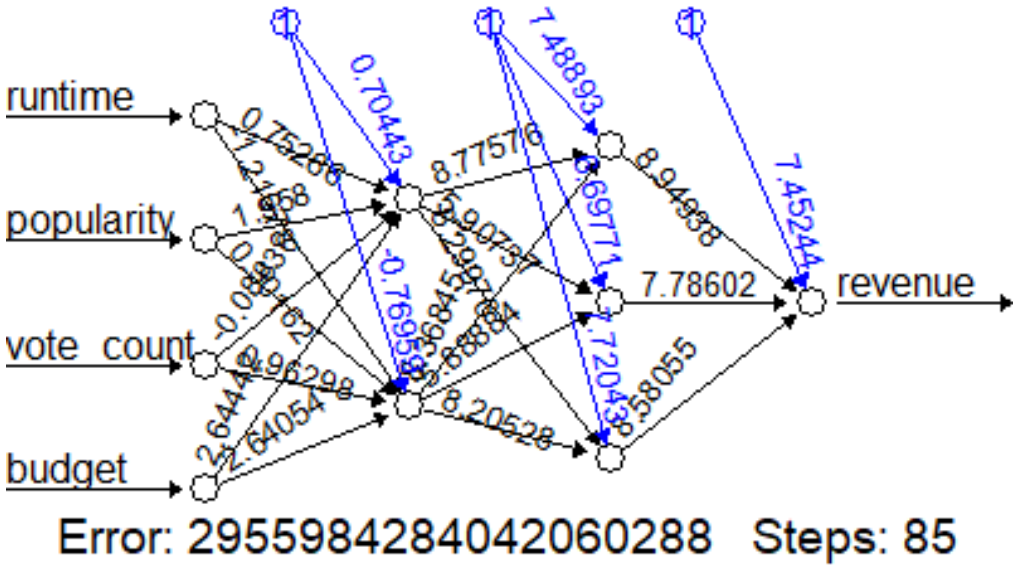
However, one of the advantages of ksvm models is the ability to cluster data into kernels.There is significant grouping on the lower end of the budget spectrum, suggesting that most animated movies had a lower budget. This is also where you will find the largest vote counts. This suggests that animated movies may not require a larger budget to be well received among audiences (Appendix pg.26) .

Neural Networks

Multiple neural networks were trained using our four parameters (runtime, popularity, vote_count, and budget) as a function for revenue. Below is a summary of one of more complex models. In this model, there are 2 hidden layers with 2 nodes in the first layer and 3 nodes in the second layer with a treshold of 0.1%. The formula used for the neural network was:

```
movie_net <- neuralnet(revenue~
(runtime+popularity+vote_count+budget), PRRB_1, hidden =
c(2, 3), lifesign="minimal", linear.output=FALSE,
threshold=0.0001)
```

Unfortunately, following training of our model, our neural network did not appear to produce a better model than either our regression analysis or our SVM. After training the neural network, the error was still astronomical indicating that the model is not predictive. Therefore we did not predict values using this model (Appendix pg.27).



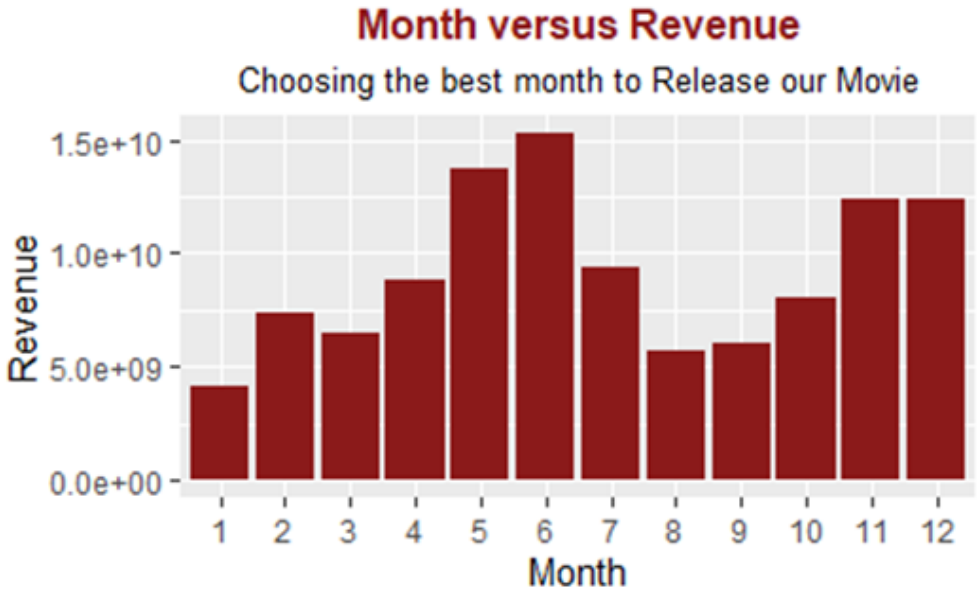
Suggested Characteristics of the Movie

A histogram of month of release vs revenue (in dollars) was generated using ggplot (geom_bar()). However before generating the frequency plot, The date was reformatted using the as.Date function and then converted to a factor in order to generate the monthly "bins".

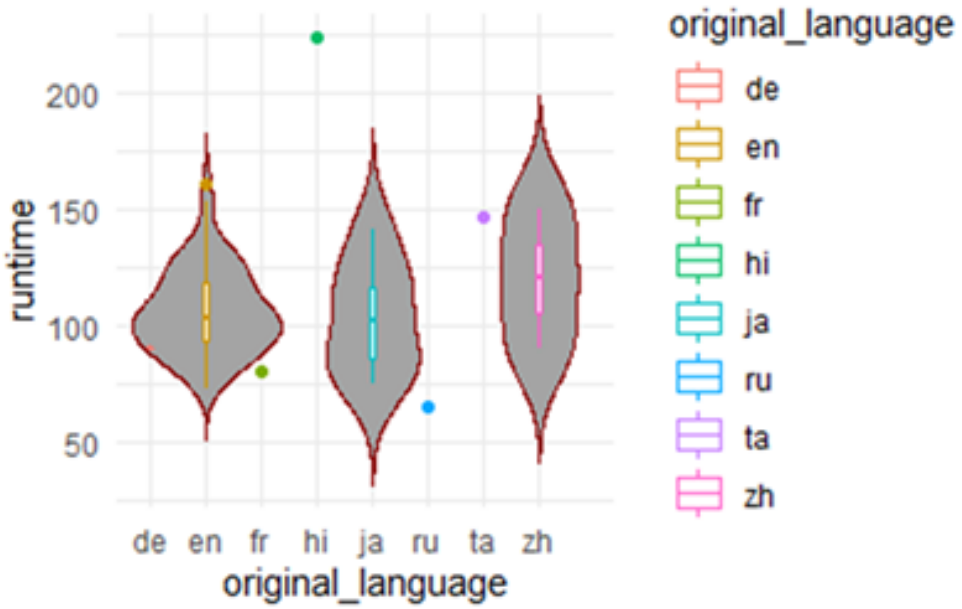
```
#convert date from 1/2/1920 to 1920-1-2
moviedata6$budget<-as.integer(moviedata6$budget)
moviedata6$release_date<-as.Date(moviedata6$release_date)
moviedata6#break date to three column
moviedata6$month <-month(ymd(moviedata6$release_date))
moviedata6$month<-as.factor(moviedata6$month)
```

Release Date, Language and Runtime

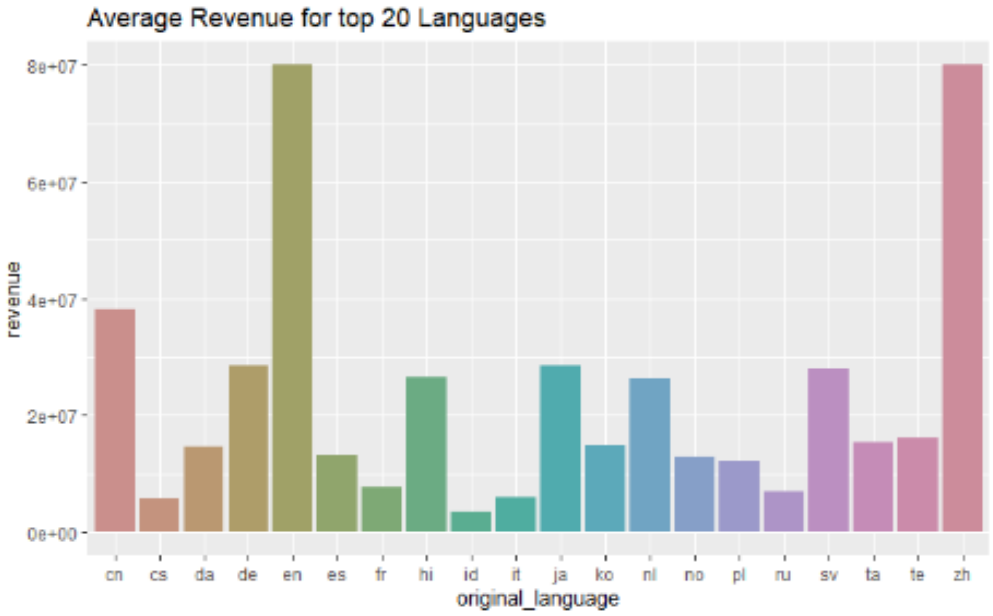
Based on months that generate the most revenue in the animation data, our movie should be released in May-June. We will launch our movie in English first. Then, we will release the movie in Chinese, German and Japanese with a runtime of ~100minutes and no longer than 160 minutes. (Appendix pg.27)



Violin plots of language vs runtime were generated for the top eight languages using the geom_violin() function, a minimal theme (grey and white background) with a dark red trim (Appendix pg.28). Based on the graph English and Japanese both have a median runtime of approximately 100 minutes; whereas, Chinese movies run about 25 minutes longer. Certain languages such as French and Russian are represented by dots, because there is not a sufficient number of points or variability in the data. Therefore we would release our movie with a runtime of ~100 minutes.



A histogram of original language vs revenue (in dollars) was generated using ggplot (geom_bar()) wioth hue theme to color the bars. Before graphing the data, the revenue was averaged for each language type. Then, due to the large number of languages represented , we plotted only the top 20 languages vs revenue. Based on the plot, we will launch our movie in English first, followed by Chinese, German and Japanese (Appendix pg.28).



Text Mining for Potential Movie Title

In order to derive a title that would be attractive to a wide audience, a wordcloud of all complete movie titles was designed using the complete dataset. In order to generate the wordcloud, the body of the text was normalized by converting the text to lowercase, removing punctuation and any symbols or numbers. Additionally common words such as “the” and “and” were removed.

```
words.corpus <- tm_map(words.corpus, content_transformer(tolower))
words.corpus <- tm_map(words.corpus, removePunctuation)
words.corpus <- tm_map(words.corpus, removeWords, stopwords("english"))
```

The resulting works were then transformed into a matrix and then each unique term was counted and summed. These summaries were then converted to the visual using the wordcloud function (Appendix pg.29). Based on the output, words such as “man”, “love”, and “little” exhibited high frequency across genres. Therefore we titled our movie “The Man who Loved his Little Girl” to capture as many of these words as possible.

Potential Movie Titles, based on words found most commonly in movie titles.

Conclusions

Data preparation was key for this project since the dataset was so large. Even with large datasets, we rarely had models that fit well and unable to design practically predictive models for our revenue data. Only once we limited the scope of our data could we identify models with good fit and minimal error. Although the dataset was not informative from a revenue perspective, it provided a lot of details to help design the movie- release date, movie title, language etc.

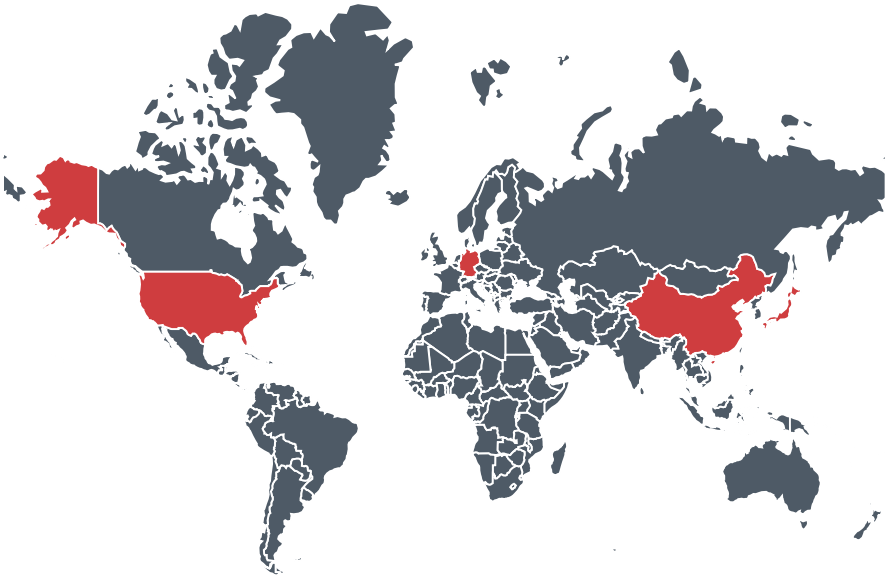
With a comprehensive data analysis by the Crazy Saturday team, here are our following findings from the “The Movie Dataset”:

a. Determined the best correlation between revenue and genre were animation and music which had the only significant values. We selected animation genre based on most significance with revenue for producing our movie.
b. Identified four primary attributes on which to build our models : budget, popularity, runtime and vote count
c. Conducted linear regression, support vector machine and neural net models to predict revenue. Support vector machine found to have a poor model fit with large residuals. Then, using more complex modeling such as neural nets did not improve our predictivity of revenue.
d. We were able to generate a retrospective predictive linear regression model of revenue using vote count as a function of revenue
e. Generated histogram of release dates across all genres and identified that summer time was the best time to release a new movie.

f. Found that the top languages for release were English, Chinese, German and Japanese and of these languages, average movie run time was 100 minutes for English, German and Japanese and 125 minutes for Chinese language films.
g. Populated a word cloud to create an attractive movie title.

Based on our data we will create a cartoon that tells the story of the antics of a rugged Dad from Montana and his daughter who leaves the snow for sunny California. Here are our following movie recommendations:

a. The animation will be released in the summer to an English-speaking audience with a runtime of 100-160 minutes.
b. We will monitor early signs of the vote count to extrapolate potential sales and monitor markets for expansion.
c. We generate a title (The Man Who Loved his Little Girl) that would perform well in our selected genre of music or animation and would be appealing enough to pull in audiences from other genres



References

- Language codes: https://www.andiamo.co.uk/resources/iso-language-codes/
- RMSE equation graphic: http://statweb.stanford.edu/~susan/courses/s60/split/node60.html
- Saltz, J. & Stanton, J. (2017). An Introduction to Data Science. Syracuse, NY: Sage Publication. Retrieved from: https://cjacks04.github.io/687/introDataScienceV4e.pdf
- Tutorial on Support Vector Machine: http://www.svms.org/regression/SmSc98.pdf

Appendix

Generating the linear model with all main effects

lm(formula = revenue ~ Action + Adventure + Animation + Comedy + Crime + Documentary + Drama + Family + Fantasy + Foreign + History + Horror + Horror + Music + Mystery + Romance + Science.Fiction + Thriller + TV + War + Western, data = genre)

Residuals:

Min	1Q	Median	3Q	Max
-184057800	-61219616	-38423532	6428147	1314679617

Coefficients:

	Estimate	Std. Error	t value	Pr(>|t|)
(Intercept)	65191944	6934859	9.401	< 2e-16***
Action[T.Action]	25532132	9264133	2.756	0.00592**
Adventure[T.Adventure]	-30185237	11304236	-2.670	0.00766**
Animation[T.Animation]	115162655	22097209	5.212	0.000000213***
Comedy[T.Comedy]	-5091593	7942304	-0.641	0.52157
Crime[T.Crime]	-9660223	10445820	-0.925	0.35522
Documentary[T.Documentary]	-30336071	43425976	-0.699	0.48493
Drama[T.Drama]	-27644033	28653104	-0.965	0.33481
Family[T.Family]	26486735	14739326	1.797	0.07253 .
Fantasy[T.Fantasy]	14186490	14233723	0.997	0.31908
Foreign[T.Foreign]	-19742531	24788642	-0.796	0.42590
History[T.History]	-5986473	19685597	-0.304	0.76109
Horror[T.Horror]	-27165791	12710285	-2.137	0.03273 *
Music[T.Music]	83393574	16711912	4.990	0.000000672 ***
Mystery[T.Mystery]	-21710059	14053108	-1.545	0.12259
Romance[T.Romance]	15167636	9696784	1.564	0.11798
Science.Fiction[T.Science Fiction]	-774902	11301490	-0.069	0.94534
Thriller[T.Thriller]	-3932663	41415654	-0.095	0.92436
TV[T.TV]	66379472	68484233	0.969	0.33256
War[T.War]	-15519048	20771216	-0.747	0.45509
Western[T.Western]	-41479353	27052637	-1.533	0.12541 ---

Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 135500000 on 1536 degrees of freedom
Multiple R-squared: 0.05657, Adjusted R-squared: 0.04428
F-statistic: 4.605 on 20 and 1536 DF, p-value: 7.537e-11

Linear Model of Music and Animation (No Moderating Effects

```
lm(formula = revenue ~ Animation + Music, data = genre)

Residuals:
      Min        1Q    Median        3Q       Max
-187238558  -61415459  -47864548     157763 1342234256

Coefficients:
                          Estimate Std.         Error          t value          Pr(>|t|)
(Intercept)                63169438          3580342        17.643        < 2e-16 ***
Animation[T.Animation]     124665121         21807778       5.717         0.0000000130 ***
Music[T.Music]             90938107          16321340       5.572         0.0000000297 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 136100000 on 1554 degrees of freedom
Multiple R-squared:  0.03807,  Adjusted R-squared:  0.03684
F-statistic: 30.75 on 2 and 1554 DF,  p-value: 7.964e-14
```

Associated Text Mining

```
#install.packages("readxl")
library(readxl)
#import the downloaded excel
md_TDM<_read.csv("C:/Users/scrouch/Desktop/SyracuseUniversity/IST687/Project/movie_data__genre_ROI_
na rm_TDM.csv")
 #TDM<-(termDocMatrix)
md_TDM<-data.frame(md_TDM)
#create a new dataframe (TDM) of the imported dataset "termDocMatrix"
summary(md_TDM)
# view the summary of TDM
#TDM1<-t(TDM)
#transpose TDM into a new dataset TDM1
#View(md_TDM)
# view the new dataframe TDM1
summary(md_TDM)
#summary of TDM1
#install.packages("arules")
#install.packages("arulesViz")
library(arules)
library(arulesViz)
md_TDM<-as.matrix(md_TDM)
class(md_TDM)
```

```
#confirm that TDM1 is a matrix
md_TDM1 <- as(md_TDM, "transactions")
#convert TDM1 into a transactions dataset called TDM2
itemFrequencyPlot(md_TDM1, support =0.05, cex.names= 0.75)
#generate a frequency plot of the itesm with support 0.05 (items appear together at least 5% of the time) and cex.names adjust the size of the x axis font for legibility
data(md_TDM1) #calling the data in TDM2
rules_md_TDM1 <- apriori(md_TDM1, parameter=list(support=0.001, confidence=0.001))
# generating an apriori rule form the list where items appear together at least 7.5% of the time and  has a confidence of 7.5% meaning that the items on the RHS appear in all of the carts when at least 7.5% of of the items appear together
summary(rules_md_TDM1)
#view the summary of the rule
inspect(rules_md_TDM1)
#inspect the ruleplot(rules_md_TDM1)
#plot the support, confidence and lift for rules_TDM2
goodrules_md_TDM1<-rules_md_TDM1[quality(rules_md_TDM1)$lift > 5]
#create a new variablegoodrules_TDM2 that indexes a subset of the data in rules_TDM2 where the lift value is > 2.0; anything less than 2 only indexes 2 items
inspect(goodrules_md_TDM1) # inspect the new indexing
plot(goodrules_md_TDM1) #plot the new indexing
```

Budget, Revenue and ROI

```
Animation <- read.table("C:/Users/scrouch/Desktop/Syracuse University/IST 687/Project/animation_1.csv",header=TRUE, sep="", na.strings="NA", dec=".", strip.white=TRUE)
Budget_hist<-hist(Animation$budget,main="Frequency of Budgets for Animation Genre",
scale= "frequency",
xlab= "budget",
breaks= 10,
col="blue")
Revenue_hist<-hist(Animation$revenue,main="Frequency of Revenue for Animation Genre",
scale= "frequency",
xlab= "revenue",
breaks= 10,
col="lightblue")
ROI_hist<-hist(Animation$ROI,main="Frequency of ROI for Animation Genre",
scale= "frequency",
xlab= "ROI",
breaks= 10,
col="red")
```

Scatterplot Matrix - Animation

```
Animation <- read.table("C:/Users/scrouch/Desktop/Syracuse University/IST
687/Project/animation_1.csv",header=TRUE, sep="", na.strings="NA", dec=".", strip.white=TRUE)
#install.packages("lattice")
#install.packages("car")
library(car)
library(lattice)
sp_matrix_animation<-scatterplotMatrix(~ budget+popularity+revenue+runtime+vote_count, data=Animation,
main="Potential Factors Impacting Revenue", regLine=TRUE, smooth=FALSE, diagonal=list(method="density"))
```

Scatterplot Matrix - Music

```
Music <- read.table("C:/Users/scrouch/Desktop/Syracuse University/IST 687/Project/music_1.csv",header=TRUE,
sep="", na.strings="NA", dec=".", strip.white=TRUE)
#install.packages("lattice")
#install.packages("car"
)library(car)
library(lattice)
sp_matrix_animation<-scatterplotMatrix(~ budget+popularity+revenue+runtime+vote_count, data=Music,
main="Potential Factors Impacting Revenue", regLine=TRUE, smooth=FALSE, diagonal=list(method="density"))
```

Animation - Linear Model (Main Effects Only)

```
lm(formula = revenue ~ budget + popularity + runtime + vote_count,    data = animation)

Residuals:
      Min        1Q    Median        3Q       Max
-357227468  -38268520  -16562917   38063033  416286071

Coefficients:
                Estimate   Std.            Error              t value        Pr(>|t|)
(Intercept)   56878002.2892     62427974.8092      0.911        0.365
budget           0.1255              0.3124              0.402        0.689
popularity    -827788.0070        3007846.7162      -0.275        0.784
runtime       -427906.1667         528999.4902       -0.809        0.421
vote_count     162010.0608          15205.9619        10.654       <2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 104500000 on 72 degrees of freedom
Multiple R-squared:  0.8086, Adjusted R-squared:  0.798
F-statistic: 76.06 on 4 and 72 DF,  p-value: < 2.2e-16
```

Animation - Linear Model with Moderating Effects

```
lm(formula = revenue ~ (budget + popularity + runtime + vote_count)^2,    data = animation)
Residuals:
      Min        1Q    Median        3Q       Max
-345369119 -24236736  -6361656  24169745 393928399
Coefficients:
                          Estimate Std.          Error      t value     Pr(>|t|)
(Intercept)             1.028e+08         2.066e+08     0.498     0.6204
budget                   -1.100e+00        1.865e+00    -0.590     0.5571
popularity              -4.788e+06         2.227e+07    -0.215     0.8304
 runtime                 -9.450e+05        2.042e+06    -0.463     0.6451
vote_count                2.097e+05        9.439e+04     2.221     0.0298 *
budget:popularity       2.575e-02         1.077e-01     0.239     0.8118
budget:runtime           9.347e-03        1.876e-02     0.498     0.6200
budget:vote_count        5.598e-04        6.756e-04     0.829     0.4103
popularity:runtime       4.072e+04        2.025e+05     0.201     0.8412
popularity:vote_count   -1.137e+03        1.405e+03    -0.810     0.4211
runtime:vote_count       -2.480e+02        7.082e+02    -0.350     0.7273
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 104400000 on 66 degrees of freedom
Multiple R-squared:  0.825, Adjusted R-squared:  0.7985
F-statistic: 31.11 on 10 and 66 DF,  p-value: < 2.2e-16
```

Animation - Linear Model (Budget Only)

```
lm(formula = revenue ~ budget, data = animation)

Residuals:
      Min        1Q    Median        3Q       Max
-191229250 -132784285  -74693120  29587311 742126730

Coefficients:
                          Estimate    Std.          Error      t value     Pr(>|t|)
(Intercept)        191901634.4917         33806374.9684         5.676     0.000000246 ***
budget                   -1.1528                0.6379        -1.807     0.0748 .
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 229100000 on 75 degrees of freedom
Multiple R-squared:  0.04172, Adjusted R-squared:  0.02895
F-statistic: 3.266 on 1 and 75 DF,  p-value: 0.07476
```

Music (Main Effects Only)

```
lm(formula = revenue ~ budget + popularity + runtime + vote_count,    data = music)

Residuals:
      Min        1Q    Median        3Q       Max
-311024550  -51110676  -13208463   33505800  459418182
Coefficients:
                Estimate  Std.          Error          t value  Pr(>|t|)
(Intercept)   -35434352.1116   67924592.1666   -0.522   0.60359
budget           -0.1396            0.8421            -0.166   0.86879
popularity    2866149.6206   1019895.6319    2.810   0.00646 **
runtime        426716.6955    629263.1488     0.678   0.50000
vote_count     107518.8419     11669.5147     9.214   1.39e-13 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 115400000 on 68 degrees of freedom  (1 observation deleted due to missingness)
Multiple R-squared:  0.7762, Adjusted R-squared:  0.7631
F-statistic: 58.98 on 4 and 68 DF,  p-value: < 2.2e-16
```

Music (with Moderating Effects)

```
lm(formula = revenue ~ (budget + popularity + runtime + vote_count)^2,    data = music)
Residuals:
      Min        1Q    Median        3Q       Max
-310810974  -51062448    6907826   33564602  449762631
Coefficients:
                          Estimate Std.          Error          t value  Pr(>|t|)
(Intercept)             -2.212e+07   1.452e+08   -0.152   0.8794
budget                   4.519e-01   3.382e+00    0.134   0.8941
popularity              -3.769e+06   1.535e+07   -0.246   0.8068
runtime                  2.224e+04   1.247e+06    0.018   0.9858
vote_count               1.371e+05   9.700e+04    1.413   0.1625
budget:popularity        8.774e-02   2.204e-01    0.398   0.6920
budget:runtime          -5.832e-03   2.124e-02   -0.275   0.7846
budget:vote_count       -3.183e-04   9.432e-04   -0.337   0.7369
popularity:runtime       9.588e+04   1.186e+05    0.809   0.4218
popularity:vote_count   -1.232e+03   5.714e+02   -2.157   0.0349 *
runtime:vote_count      -1.749e+02   7.768e+02   -0.225   0.8226
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
Residual standard error: 114800000 on 62 degrees of freedom  (1 observation deleted due to missingness)
Multiple R-squared:  0.7762, Adjusted R-squared:  0.7631
F-statistic: 58.98 on 4 and 68 DF,  p-value: < 2.2e-16
```

Music (with Budget Only)

```
lm(formula = revenue ~ budget, data = music)
Residuals:
      Min        1Q    Median        3Q       Max
-158516998 -145440046 -115955404   27155725 1000071032
Coefficients:
                    Estimate Std.          Error          t value        Pr(>|t|)
(Intercept)     149007083.6325       34277790.0601        4.347       0.0000446 ***
budget          0.2348               1.5750               0.149       0.882
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 237700000 on 72 degrees of freedom
Multiple R-squared:  0.0003086, Adjusted R-squared:  -0.01358
F-statistic: 0.02222 on 1 and 72 DF,  p-value: 0.8819
```

Linear Model for Animation Dataset Using Retrospective Variables

```
lm(formula = revenue ~ popularity + vote_count, data = genre)

Residuals:
      Min                 1Q             Median             3Q                Max
-354495899          -36718315          -15140331          36356148          425559270

Coefficients:
                    Estimate Std.          Error          t value        Pr(>|t|)
(Intercept)         20962627           23397798           0.896          0.373
popularity          -1227837           2905303            -0.423         0.674
vote_count          160040             14855              10.773         <2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 103700000 on 74 degrees of freedom
Multiple R-squared:  0.8062, Adjusted R-squared:  0.8009
F-statistic: 153.9 on 2 and 74 DF,  p-value: < 2.2e-16
```

Linear Model for Animation using Vote Count Only

```
lm(formula = revenue ~ vote_count, data = genre)
Residuals:
      Min        1Q    Median        3Q       Max
-350724426  -38589091  -13419307   34104232  429023854

Coefficients:
             Estimate Std.      Error    t value  Pr(>|t|)
(Intercept)  13123436        14183143      0.925    0.358
vote_count     154994           8788       17.636   <2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
Residual standard error: 103100000 on 75 degrees of freedom
Multiple R-squared:  0.8057, Adjusted R-squared:  0.8031
F-statistic:   311 on 1 and 75 DF,  p-value: < 2.2e-16
```

Support Vector Machine

```
anim <- read.csv("C:/Users/scrouch/Desktop/Syracuse University/IST 687/Project/animation_1.csv")
dim(anim)
randIndex_anim<-sample(1:dim(anim)[1])
summary(randIndex_anim)
length(randIndex_anim)

x2_3cut<-floor(2*dim(anim)[1]/3)
x2_3cut

train_anim<-anim[randIndex_anim[1:x2_3cut],]
test_anim<-anim[randIndex_anim[(x2_3cut+1):dim(anim)[1]],]

ksvm_anim<-ksvm(revenue ~ budget+runtime+popularity + vote_count, data=train_anim, kernel="rbfdot", kpar =
"automatic", C=3, cross = 2, prob.model =TRUE)
print(ksvm_anim)
str(test_anim)ksvm_anim
Pred<-predict(ksvm_anim, test_anim, type = "decision")
ksvm_anim_df<-data.frame(test_anim, ksvm_animPred)
ksvm_anim_revenue_rmse<- sqrt(mean((ksvm_animPred-test_anim$revenue)^2))
anim1<-cbind(anim, ksvm_anim_revenue_rmse)
ksvm_anim_revenue_res<-(ksvm_anim_revenue_rmse-anim1$revenue)
anim2<-cbind(anim1, ksvm_anim_revenue_res)
scat_ksvm_anim<-ggplot(anim2, aes(x=budget, y=popularity, color = vote_count)) +
geom_point(aes(size=ksvm_anim_revenue_res)) + ggtitle("ksvm model of
revenue_animation")+scale_color_gradient(low="blue", high="red") scat_ksvm_anim
```

Neural Network

```
# Install packages if necessary
#install.packages("neuralnet")
library(neuralnet)
PRRB_1<-read.csv(file = "C:/Users/scrouch/Desktop/Syracuse University/IST 687/Project/animation_1.csv",
header=TRUE, sep="")#importing the dataset
names(PRRB_1)
movie_net<-neuralnet(revenue ~ (runtime + popularity+vote_count +budget), PRRB_1, hidden = c(2, 3),
lifesign="minimal", linear.output =FALSE, threshold =0.0001)
movie_net$result.matrix
plot(movie_net)
```

Release Date

```
 #convert date from 1/2/1920 to 1920-1-2
moviedata6$budget<-as.integer(moviedata6$budget)
moviedata6$release_date<-as.Date(moviedata6$release_date)
moviedata6
#break date to three column
#install.packages("lubridate"
)library(lubridate)
moviedata6$month <-month(ymd(moviedata6$release_date))
moviedata6$month<-as.factor(moviedata6$month)
moviedata6
#install.packages("ggplot2")
library(ggplot2)
#install.packages("dplyr")
library(dplyr)
bar<-ggplot(data=moviedata6,aes(x = moviedata6$month, y=moviedata6$revenue))+geom_bar(stat="identity",
fill="firebrick4") + xlab("Month") + ylab("Revenue")+  labs(title = "Month versus Revenue",    subtitle = "Choosing
the best month to Release our Movie") + theme(plot.title = element_text(lineheight = 0.9, hjust = 0.5,color =
"firebrick4", size = 12, face = "bold"), plot.subtitle = element_text(lineheight = 0.9, hjust = 0.5,color = "black", size =
10, ))bar
#colnames(moviedata6)
# list of column names in moviedata6
```

Runtime

```
anim_data <- read.csv("C:/Users/scrouch/Desktop/Syracuse University/IST 687/Project/animation_1.csv")
names(anim_data)
anim_data1<-ggplot(anim_data, aes(x = original_language, y = runtime))+  geom_point() +  geom_violin(stat =
"smooth", method = "loess") +
    theme(legend.position="top",
        axis.text=element_text(size = 6))
 anim_data1<-ggplot(anim_data, aes(x = original_language, y = runtime,
color =original_language))+  geom_point() +
  geom_violin(trim=FALSE, fill='#A4A4A4', color="darkred")+
   geom_boxplot(width=0.1) + theme_minimal()
```

Original Language

```
#install.packages("ggplot2")
#install.packages("dplyr")
library(ggplot2)
library(dplyr)
#creating a dataframe of mean revenue and language
revenue_lang_aggr<-aggregate(revenue ~ original_language, moviedata6, mean)
rla.df<-data.frame(revenue_lang_aggr)

# sorting the rla dataframe first ordering revenue in ascending order, taking the top 20 values grouped by
language
rla1.df<-rla.df %>%
arrange(desc(revenue)) %>% slice(1:20)%>%
group_by(original_language)

#generate a bar plot of the top 20 original languages and the mean budget using the "hue" colour scheme
hue_barplot<-ggplot(rla1.df, aes(x=original_language, fill=original_language, y = revenue)) +geom_bar(stat =
"identity")+ scale_fill_hue(c = 40) + theme(legend.position="none")+ggtitle("Average Revenue for top 20
Languages")
print(hue_barplot)
```

Wordcloud

```{r cars}
library(tm)
library(wordcloud)
title_file <- "alltitle.txt"
titles <- readLines(title_file)str(titles)
words.vec <- VectorSource(titles)
words.corpus <- Corpus(words.vec)
words.corpus <- tm_map(words.corpus, content_transformer(tolower))
words.corpus <- tm_map(words.corpus, removePunctuation)
words.corpus <- tm_map(words.corpus, removeWords, stopwords("english"))
titleCloud <- TermDocumentMatrix(words.corpus)
tC <- as.matrix(titleCloud)
wordCounts <- rowSums(tC)
wordCounts <- sort(wordCounts, decreasing=TRUE)
head(wordCounts)
#cloudFrame <- data.frame(word=names(wordCounts), freq=wordCounts)
#wordcloud(cloudFrame$word, cloudFrame$freq)
wordcloud(names(wordCounts), wordCounts, min.freq=2, max.words=50, rot.per=0.3, color=brewer.pal(8, "Dark2"))
```