Can you make a hit movie?

Final Report

SEAS 6402
Dr. Joe Goldfrank

Anulekha Boddu
G40397446

Nicole Xie
G35996333

Problem Statement

“Can you make a hit movie?”

This paper aims to answer the question posed above by first analyzing previously successful movies and building predictive models that will predict the revenue for a given movie. In order to make this interactive we aim to create a front end interface hosted on Tableau. Users can input various features of their films (eg., the main actor, their budget, the director, their source material, etc.) and through our analysis we will be able to predict how each of those features might impact the revenue of their movie.

Big production studios like Netflix, Amazon and Universal Studios can benefit from the expected outcome of this project. We hope to save big studios time, money, and resources by allowing them a tool to outsource a large chunk of their decision making. Our tool will allow them to decide on which projects to pursue and which ones aren’t worth pursuing. Of course, filmmakers and producers should take risks, however, this will allow them to narrow down their prospects by outsourcing their initial screening.

Film students and indie directors making their first films might not have the resources that big studios have to assess the success of their movie. By providing them with a tool that can predict the success of their movie backed by rigorous statistical analyses, they can be rest assured in at least a few of the many filmmaking decisions they must make.

The main decision that these stakeholders make is: Will making this movie be profitable for me? We aim to help provide an answer to that question.

We plan on doing exploratory analysis, web scraping, and one or two machine learning modeling. We hope the information we get through analyzing the dataset can provide some for either data analysts or film maker insight on predicting movie revenues.

Related Work

Multiple Linear Regression(MLR) is the most common one that has been used in several studies with accuracy above 70% (Hsu et al., 2014). Also, Ni and colleagues found for low box office movies, it’s more suitable to use a regression model than any other models(2022), and low box office is also what we aim for in this analysis. Hence, we will use the regression model as our first model. Four assumptions need to be met to ensure the validity of the regression model (Gross, 2003; Abidi Syed et al., 2020), in which we will break down in detail in the following method section to show how we test these assumptions. Although MLR has limitations of linearity restriction between predictors and target variables, we can adapt a two layer stacking

algorithm model to discover both the linear and non-linear aspect of the prediction (Ni et al., 2022).

We find reviews and word of mouth are used in post production, but we will only focus on the pre production of movies, so we decide not to use reviews or keywords for now(Chiranjib & Prabir, 2022). One study concluded that the best machine learning algorithm to use is depending on the dataset one uses, so we decided to try a time series model if time allows. Considering the specific release time is given in our dataset(year, month, and date), we think seasonality would be one another interesting feature to explore. Time series analysis means a method of analyzing a sequence of data points collected over an interval of time (Madsen, 2008).

Few past studies didn't find star powers to have a positive significant influence on predicting movie revenues(Chiranjib & Prabir, 2022; Liu, Mazumdar, & Li, 2015). Thus, we will not pay extra attention to stars or use any classification for star powers. Directors also seems to be an inconclusive predictor based on past studies(Chiranjib & Prabir, 2022), although we have a variable that is writer so we may be able to use these two variables to explore something new about their effects on movie performance.

We believe classification model would be effective to use due to the nature of our dataset contains thousands of variables, however, one study by Cocuzzo and Wu at Stanford University (2013) found that classification model seems to be effective in predicting unsuccessful movies not the other, so we decide not to consider any classification model at this time.

Data

We are using the Kaggle Dataset "Movie Industry" that was originally pulled from The Internet Movie Database (IMDB) using API. It has a total of 6820 worldwide movies (before cleaning) across from 1980 to 2020. We have first checked any NA/NAN entry and found out most of them were under the budget and gross columns. All of the columns are important variables in our analysis, so we decided to just drop any rows that contain a NA entry. When checking for the duplicates, we only checked the movie name and year released as we were aware it's normal to have duplicates in other variables. We decided to only keep the movies from the UK, US, and Canada since these three movie industries are the closest in their budgets, audiences, and cultures.

First, we plotted simple histograms to see the distribution of the data. Most films' budgets fall within this range of $10-50 million, while the average revenue is around $100 million. Average movie runtime is 100 minutes, which is roughly 1 hour and 40 minutes. The year distribution with the initial dataset was more evenly distributed, however, after data cleanup the distribution is more uneven. Older movies have lesser number of films due to the fact that the budget information for a lot of older movies was missing. Average score is about 6.5, with 0

movies that have a rating of 10. This is expected since it's highly improbable that a movie has a perfect rating.

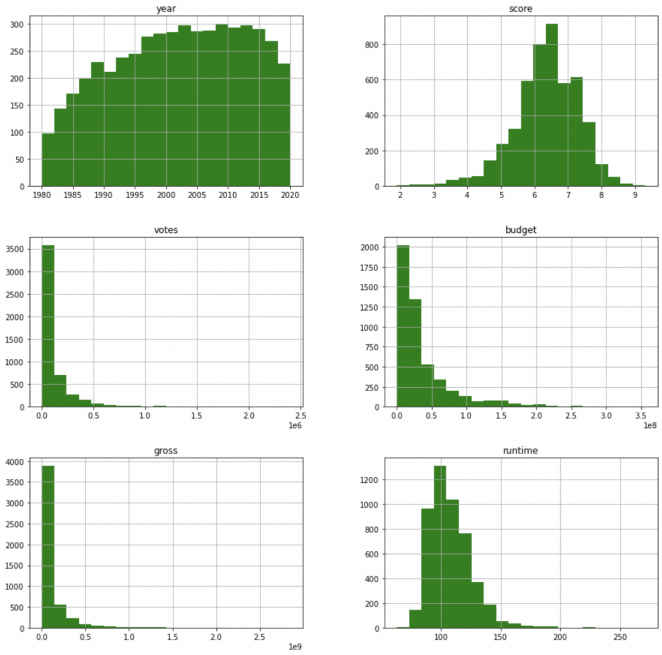


Figure 1: Data distribution plots

Conventionally, it is believed that higher budget movies bring in higher revenue. This correlation plot shows that there is in fact a generally positive relationship between the two factors. These two are the movies with the highest budgets and they are the two Avengers films: Infinity War and Endgame. The movie with the highest revenue is Avatar.

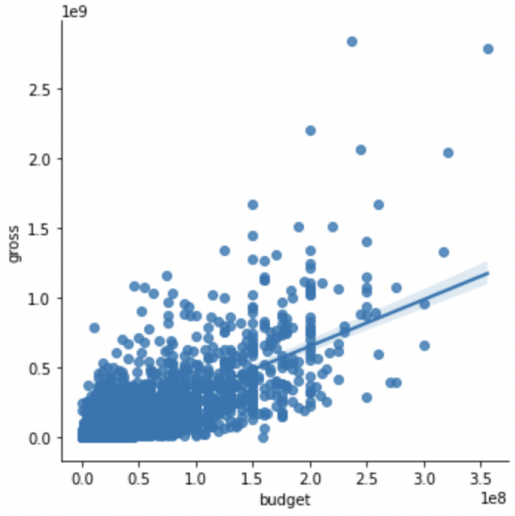


Figure 2: Correlation between Revenue and Budget

	year	score	votes	budget	gross	runtime
count	4922.000000	4922.000000	4.922000e+03	4.922000e+03	4.922000e+03	4922.000000
mean	2001.482934	6.359651	1.158238e+05	3.665873e+07	1.060303e+08	107.682446
std	10.596736	0.951509	1.836579e+05	4.223868e+07	1.904920e+08	17.594351
min	1980.000000	1.900000	1.950000e+02	6.000000e+03	3.090000e+02	63.000000
25%	1993.000000	5.800000	1.800000e+04	1.000000e+07	1.112565e+07	95.000000
50%	2002.000000	6.400000	5.200000e+04	2.200000e+07	3.860006e+07	105.000000
75%	2010.000000	7.000000	1.310000e+05	4.500000e+07	1.160595e+08	117.000000
max	2020.000000	9.300000	2.400000e+06	3.560000e+08	2.847246e+09	271.000000

Figure 3: Summary Statistics

We also plotted the number of votes, or number of ratings, per year. We find that 1994 and 2008 specifically had a significantly larger number of votes. So we took a closer look at those years.

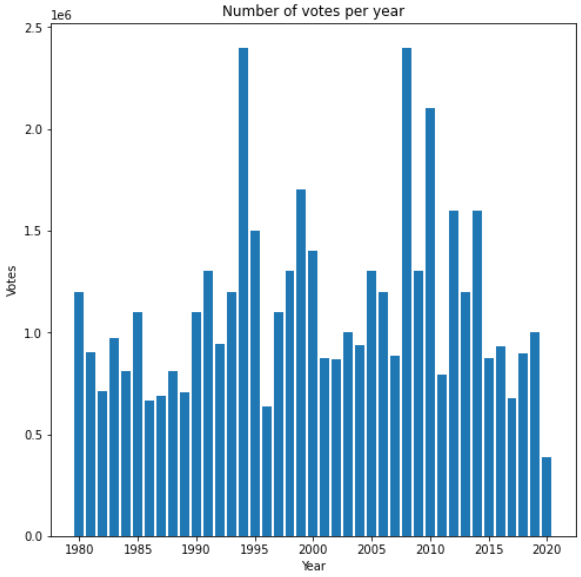


Figure 4: Number of votes per year

	name	rating	genre	year	released	score	votes	director	writer	star	country	budget	gross	company	runtime
1289	The Shawshank Redemption	R	Drama	1994	October 14, 1994 (United States)	9.3	2400000.0	Frank Darabont	Stephen King	Tim Robbins	United States	25000000.0	2.881729e+07	Castle Rock Entertainment	142.0
1291	Forrest Gump	PG-13	Drama	1994	July 6, 1994 (United States)	8.8	1900000.0	Robert Zemeckis	Winston Groom	Tom Hanks	United States	55000000.0	6.782261e+08	Paramount Pictures	142.0
1290	Pulp Fiction	R	Crime	1994	October 14, 1994 (United States)	8.9	1900000.0	Quentin Tarantino	Quentin Tarantino	John Travolta	United States	8000000.0	2.139288e+08	Miramax	154.0
1293	The Lion King	G	Animation	1994	June 24, 1994 (United States)	8.5	970000.0	Roger Allers	Irene Mecchi	Matthew Broderick	United States	45000000.0	1.083721e+09	Walt Disney Pictures	88.0
1298	Dumb and Dumber	PG-13	Comedy	1994	December 16, 1994 (United States)	7.3	361000.0	Peter Farrelly	Peter Farrelly	Jim Carrey	United States	17000000.0	2.472754e+08	New Line Cinema	107.0

Figure 5: 1994 movies with the highest number of votes

	name	rating	genre	year	released	score	votes	director	writer	star	country	budget	gross	company	runtime
3248	The Dark Knight	PG-13	Action	2008	July 18, 2008 (United States)	9.0	2400000.0	Christopher Nolan	Jonathan Nolan	Christian Bale	United States	185000000.0	1.005974e+09	Warner Bros.	152.0
3264	WALL·E	G	Animation	2008	June 27, 2008 (United States)	8.4	1000000.0	Andrew Stanton	Andrew Stanton	Ben Burtt	United States	180000000.0	5.213119e+08	FortyFour Studios	98.0
3250	Iron Man	PG-13	Action	2008	May 2, 2008 (United States)	7.9	969000.0	Jon Favreau	Mark Fergus	Robert Downey Jr.	United States	140000000.0	5.857962e+08	Paramount Pictures	126.0
3269	Slumdog Millionaire	R	Drama	2008	December 25, 2008 (United States)	8.0	812000.0	Danny Boyle	Simon Beaufoy	Dev Patel	United Kingdom	15000000.0	3.784105e+08	Celador Films	120.0
3259	The Curious Case of Benjamin Button	PG-13	Drama	2008	December 25, 2008 (United States)	7.8	605000.0	David Fincher	Eric Roth	Brad Pitt	United States	150000000.0	3.358028e+08	Warner Bros.	166.0

Figure 6: 2008 movies with the highest number of votes

These two years were simply great years for film, with a lot of highly rated and popular films releasing in these years.

Looking at the production companies, we find that there are 1263 unique companies, with Universal Pictures having the most number of films in this dataset.

	company
Universal Pictures	316
Columbia Pictures	284
Warner Bros.	282
Paramount Pictures	265
Twentieth Century Fox	202
New Line Cinema	142
Walt Disney Pictures	105
Touchstone Pictures	103
Metro-Goldwyn-Mayer (MGM)	92
Dreamworks Pictures	73

Figure 7: Most popular production companies

Comedy is the most popular movie genre, making up 28.9% of the movies in this dataset, with Action having 25.9%. Comedy has 1423 movies and Action has 1273. The genre with the lowest number of movies is Western with only 2 films. Family is the second least popular genre with only 4 movies in the dataset. Sci-Fi has only 6 movies and Thriller only 7. These numbers don't seem right, and are leading to an imbalance in the dataset. It might be because movies can often be described using more than one genre (eg. Action and Adventure, Rom-Com, Sci-fi Adventure, etc.). Therefore, it is important to address this issue.

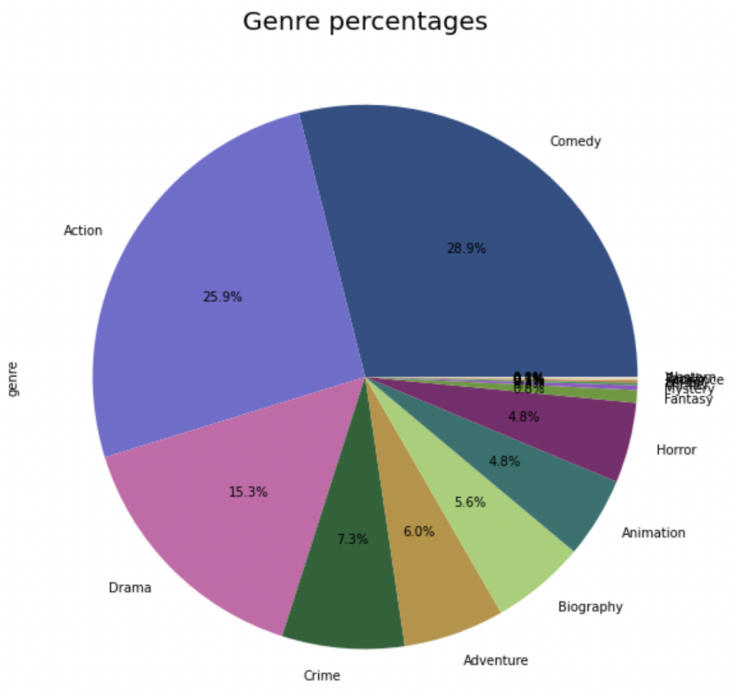


Figure 8: Genre Distribution

Methods

We decided to start with a basic linear regression model. The feature set included: Budget, Runtime, Studio Partner, Director, Writer, Main Actor, and Genre. Since most of these variables are categorical, we used the OneHotEncoder() function to convert them into numerical values so that we can model them. We then ran three regressions: Linear, Ridge, and Lasso. The RMSE results are as follows:

Model	RMSE
Linear Regression	2.59e+21
Ridge Regression	1.92e+8
Lasso Regression	1.11e+8

These scores are exaggerated, indicating poor models. To address this, we first decided to use a dimensionality reduction method, PCA, to address the inflated number of columns. However, since most of the data is one-hot encoded, PCA didn't yield the best results. Therefore, it was quickly dropped.

When it comes to analysis, besides the exploratory data analysis we have talked about above, we also used machine learning modeling and several python libraries to complete the analysis. Sklearn, Numpy, and Panda have been used frequently in our analysis for creating data frames, extract values, slicing, and modeling. Matplotlib and Seaborn have been widely used in visualization. Beautiful soup is mainly used for scraping information out of movies' wikipedia pages.

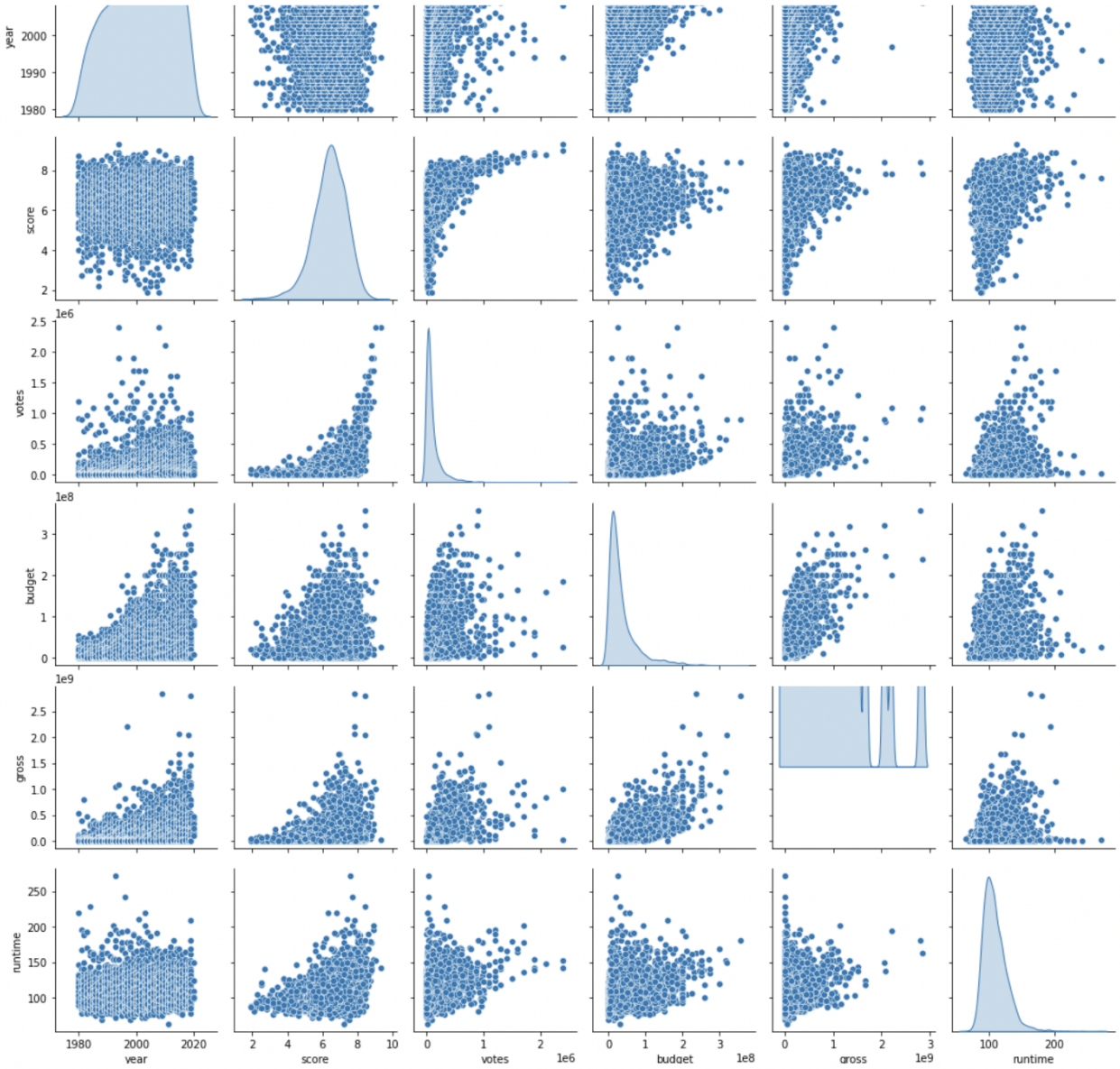
There are minimal to none mathematical theories behind our analysis because of the techniques we use so far. The only analysis that has mathematical theory is the linear regression model. Linear regression model is used to measure the relationship between a dependent variable and one or more independent variables. In our analysis, we have more than one independent variable so we will focus on explaining the math behind Multi Linear Regression. The hypothesis/equation is seen as below(Freedman, 2009):

$$Yi = b0 + b1Xi1 + b2Xi2 + ... + bpXip + ei$$

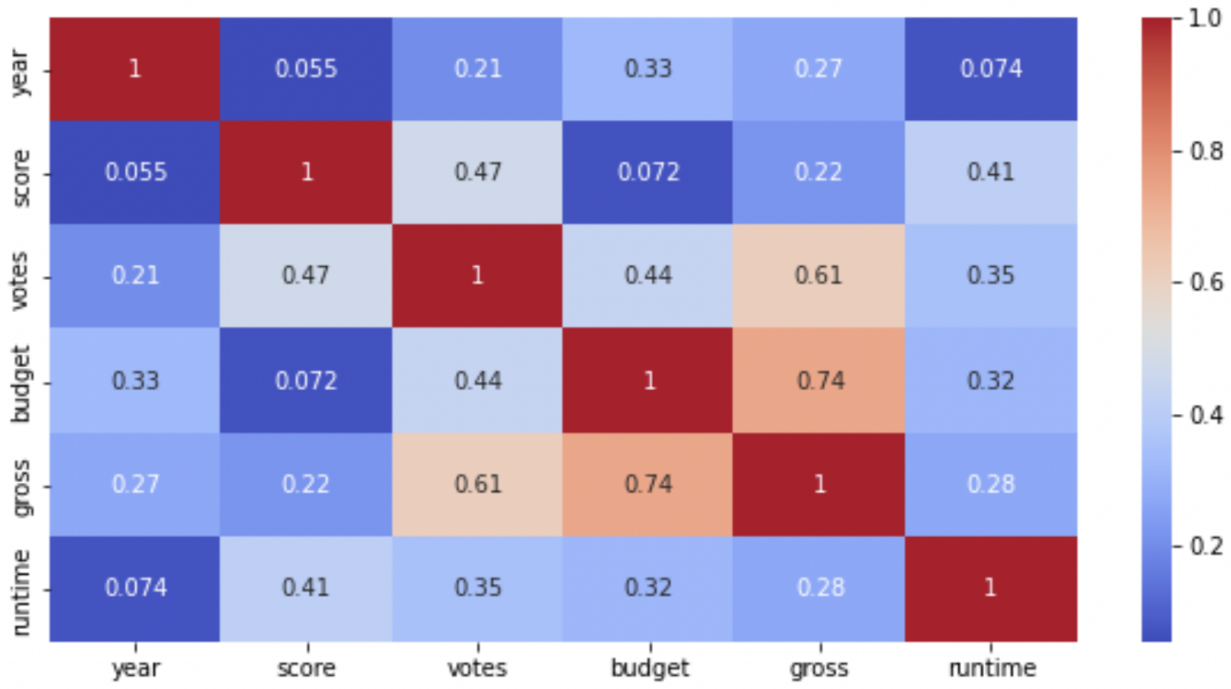
In this equation, we propose a function that uses all known quantities(X) to predict unknown quantities(Y). All the X in the function represents our independent variables and Y on the left side of the function represents the variable we want to predict.

Four assumptions have to be met in order for the linear regression to work. First is the linear relationship, which means all variables need to have a linear relationship or graphically represented as a straight line. To check this assumption, we used a pairwise scatter plot(see

attached figure below) among the variables.



The second assumption is all variables should be multivariate normal. This can be checked through plotting variables in histogram, which we did in the EDA(seen in the previous Data section). The third assumption is multicollinearity, where no independent variables should be highly correlated to each other. We checked through the correlation matrix(see attached figure below), and we then eliminated the variable with a correlation above 0.8 or higher.



The fourth assumption is homoscedasticity, where there is no autocorrelation among residuals/errors. This can be checked by plotting the errors after we do the regression analysis (Statistics Solution).

In the web scraping process, the function was working on a single entry of URL, as you can see the results generated below, where a dictionary has keys and values so we can pull out keys that are "based on" or "story by" to get the value for creating source material. However, it took us too much time to make the lists of URLs to work, because we encountered some issues from getting the movies' infobox out of a whole list.

The first issue we had was creating the URL to get html content. We know for all wikipedia movies' pages, they have a base path that's "https://en.wikipedia.org/wiki/", follow by the base path is the movie name with underscore symbol connected. We created a panda series when we pulled all the movies' names into a list, and in order to split the movies' names with underscore, we needed to use str. before the initial string function split and join to create. This is different from splitting a single string and joining with a special symbol, so it took us a while to find the trick.

The second issue we had was using the function we created for a single URL and iterating the list of URLs through the function. We thought if the function works on a single URL, it should also work on a list of URLs, but in reality, we forgot that some URLs in the list we created through base wiki path and movies name weren't the perfect URLS that can direct you straight to the specific movie page. As we recall, when entering a name into wikipedia search bar, it will return several pages that contain the movie name or remake of the movies in

```
import wikipedia as wiki
wiki.search('The_Shining')
```

```
['Shining',
 'The Shining (film)',
 'Pokémon Brilliant Diamond and Shining Pearl'
 'The Shining (novel)',
 'The Shining (miniseries)',
 'Doctor Sleep (2019 film)',
 'Shining Path',
 'Shining Girls',
 'Sun Is Shining',
 'Shining Force']
```

different years(fig 5). For example, the movie name “The Shining” gave us not only the film but also the novel and the remake of the movie.

To solve this issue, we at first didn’t use the regex r".{30}[b][o][x][_][f][i][l][m]" expression used to filter if the page is a film movie page, because we thought if the page has no infobox vevent then the page must not be a movie page. However, it turned out infobox vevent were not unique to movie pages, some media franchise pages also have it. Then we went back to compare line by line what’s unique about a movie page compared to other pages, and it turned out to be a string beginning with “Templatestyle” and following with a series of numbers that can be used to identify movie pages. In order to select that string, using “soup.select” was not working because the media franchise pages also had the same Template styles, so we used the regex expression, which contains the word ‘film’ and might filter out movies pages. But it’s still not working and stopped at one of the media franchise pages, which means the regex expression was also not working. We went back and checked raw Html content line by line, and it turned out the media franchise pages also contain “film” words. At this point, we think we should not proceed to web scraping although there might be an alternative solution which we will discuss in the future work section.

Feature Engineering

The next step is feature engineering. The dataset we are using is static in time. It only captures the impact at a certain point in time. However, films have a crucial temporal aspect that is not accounted for. For example, older movies have lower budgets or newer actors have less leverage than veteran actors; it’s the same with directors. Therefore, in an attempt to bring this temporal aspect into our dataset, we decided to create three new features: actor’s previous film’s gross revenue, director’s previous film’s gross revenue, and writer’s previous film’s gross revenue. These capture the impact of the actor/director/writer’s popularity at the time of release.

As mentioned, movie production has changed in more ways than one over the years, one of them being in their budget. If not taken into consideration, this can lead to under-fitting and poor model training. Therefore, to address this, we split the dataset into training and test subsets

based on the release date. All movies released in or after 2018 were established as the test set, and the rest of the films were made the training set. After making these changes, the results were as follows:

	R_Squared	RMSE	MAE
ElasticNet	0.615488	1.211330e+08	6.726418e+07
Ridge	0.613025	1.215204e+08	6.797800e+07
XGBRegressor	0.610764	1.218748e+08	6.411450e+07
Gradient Boosting	0.604755	1.228120e+08	6.483132e+07
KNeighborsRegressor	0.485217	1.401586e+08	7.355483e+07
Extra Tree	0.289625	1.646461e+08	7.920358e+07
Decision Tree	0.268460	1.670809e+08	7.827367e+07
Lasso	-1.345578	2.991800e+08	2.039964e+08
Linear	-170579.741688	8.068124e+10	2.128514e+10

Figure 9: Model Results with Feature Engineering

As can be seen, the ElasticNet model performed the best with an R-squared of 61.5%. While this is a good start, we were aiming for a much higher number.

Upon further investigation of the data, it came to our attention that the ‘Genre’ variable was very imbalanced. Movies are rarely defined as belonging to one specific genre. Action movies are also often adventurous. Romance films can be categorized under either drama or comedy. Comedy and drama are sometimes combined as ‘dramedy’. However, our dataset only assigns one genre to each movie, which created a major imbalance in the variable. Comedy is the most popular genre, making up 28.9% of the entire genre set with 1423 movies. In contrast, only 5 movies are labeled as romance. It is clear that the majority of romantic comedies are labeled as only comedy. Since addressing this issue directly and trying to re-label all the films in the dataset is beyond the scope of this project, we wanted to see how removing ‘Genre’ completely would impact the model performance. Here are the results:

	R_Squared	RMSE	MAE
Ridge	6.156628e-01	1.211055e+08	6.832576e+07
ElasticNet	6.136035e-01	1.214295e+08	6.740045e+07
XGBRegressor	6.047123e-01	1.228186e+08	6.484326e+07
Gradient Boosting	6.043430e-01	1.228760e+08	6.460082e+07
KNeighborsRegressor	4.852166e-01	1.401586e+08	7.355483e+07
Extra Tree	3.758735e-01	1.543277e+08	7.483327e+07
Decision Tree	2.973319e-01	1.637505e+08	7.784809e+07
Lasso	-1.465579e+00	3.067376e+08	2.113246e+08
Linear	-1.115818e+06	2.063500e+11	5.500128e+10

Figure 10: Model Results with No Genre

Surprisingly, the change in the R-square is pretty insignificant with only a 0.02% increase. This suggests that a movie genre has very minimal impact on movie revenue. We also wanted to make sure that the added variables are not a cause for the minimal increase, so we also ran the model after removing both “Genre” and the added features.

	R_Squared	RMSE	MAE
ElasticNet	5.860579e-01	1.256832e+08	7.008344e+07
Gradient Boosting	5.860054e-01	1.256912e+08	6.602158e+07
Ridge	5.151205e-01	1.360268e+08	8.006512e+07
KNeighborsRegressor	5.068799e-01	1.371778e+08	7.388596e+07
XGBRegressor	4.187874e-01	1.489276e+08	6.724080e+07
Decision Tree	3.324687e-01	1.596039e+08	7.767788e+07
Extra Tree	2.681107e-01	1.671207e+08	7.613702e+07
Lasso	-1.361310e+00	3.001816e+08	2.051938e+08
Linear	-2.153718e+08	2.866831e+12	9.437955e+11

Figure 11: Model Results with no Genre and no Feature Engineering

The model performed worse, so we added back both the features and the “Genre” variable.

The next step is to introduce non-linearity into the model. Since there are only 2 originally numerical values, ‘Budget’ and ‘Runtime’, these are the only variables that we can make non-linear transformations to. We tried to add polynomials to the model by squaring the variables, as well as square root. However, they both made the model worse. We then decided to log the budget. That resulted in the following:

	R_Squared	RMSE	MAE
Gradient Boosting	0.590862	2.340613e+08	1.275035e+08
XGBRegressor	0.523908	2.524877e+08	1.291691e+08
ElasticNet	0.477688	2.644598e+08	1.505079e+08
Extra Tree	0.448107	2.718455e+08	1.374680e+08
Ridge	0.404179	2.824571e+08	1.703678e+08
KNeighborsRegressor	0.379561	2.882334e+08	1.608907e+08
Decision Tree	0.310486	3.038548e+08	1.534259e+08
Lasso	-0.795269	4.902972e+08	3.239546e+08
Linear	-34.297756	2.174043e+09	1.550347e+09

Figure 12: Model Results with Log Budget

This also made the model worse but not by too much. So we then also logged the runtime. This resulted in our best model yet:

	R_Squared	RMSE	MAE
Ridge	0.632706	2.217695e+08	1.263198e+08
ElasticNet	0.619000	2.258692e+08	1.250520e+08
Extra Tree	0.614181	2.272932e+08	1.259410e+08
Gradient Boosting	0.576775	2.380566e+08	1.285767e+08
XGBRegressor	0.523956	2.524748e+08	1.290913e+08
KNeighborsRegressor	0.507823	2.567175e+08	1.400935e+08
Decision Tree	0.377883	2.886229e+08	1.514114e+08
Lasso	-0.381813	4.301493e+08	2.809267e+08
Linear	-27.222762	1.943992e+09	1.395779e+09

Figure 13: Model Results with Log Budget and Log Runtime

Challenges and Limitations

One of the main challenges we faced was in the model deployment. We wanted to build an interactive UI hosted on Tableau, which is a visual analytics platform. Tableau recently introduced its own API Extensions that would allow users to integrate and interact with functionality or data from other applications directly in Tableau. Since this is a fairly new addition, it demanded a lot of time and effort that was ultimately misspent because several of the functionalities necessary for this project had limited availability. Therefore, we had to drop this whole aspect of the project. The time spent on this could have been used to improve the model even further.

The film industry is a highly unpredictable and robust industry. There are innumerable amounts of different features that impact movie revenue. In this project, we only focused on a very small aspect of filmmaking. A movie could have the most popular director, writer, and actor, the best production studio, the biggest budget, and still flop in the box office. This is because movies are very subjective and also artistic. It is an almost impossible task to quantify art.

Other aspects that could greatly impact movie revenue include story, cinematography, editing. However, the most important one is marketing. An incredible movie that could do considerably well in the box office might have failed because it wasn't marketed properly. On the other hand, movies that aren't so great in their quality can do extremely well if they had a high

marketing budget. We wanted to only study the impact of the pre-production on movie revenue, which is why we didn't include marketing into our model, yet it is a very powerful one.

For web scraping, the output by now can be seen below in white HTML format, which seemed to be only one movie's information. We didn't know why when iterating the whole list, only

```
{'title': 'Unplanned', 'Directed by': ['Cary Solomon', 'Chuck Konzelman'], 'Screenplay by': ['Cary Solomon', 'Chuck Konzelman'],
```

One movie's url went through, as if other URLs were not valid, so we went back to test a few URLs one by one, and it all worked and generated a dictionary result which can be seen above in black.

We can't find the inline difference of HTML between a movie page versus other pages on wikipedia, we need to go back to refine our urls, which seems to be the only way to direct beautiful soup to wikipedia's movies pages. We knew the professor had suggested early to refine the search term through adding movie year, but we thought the regex expression would work, so we didn't try that path at the beginning. Hence, the next step will be adding '(year_film)' by the end of the urls. We also found that if we don't include movie year, with movies having more than one remake, wikipedia would stop at the search page and not know which link to pick. We attempted the process and sliced the release year out; but we don't know how to add a matched year into the format as seen below. We believe this is worth documenting as in future, if anyone

```
[<tr><th class="infobox-above summary" colspan="2" style="font-size: 125%; font-style: italic;">Doli
<ul><li>Stephen Gaghan</li>
<li>Dan Gregor</li>
<li>Doug Mand</li></ul>
</div></td></tr>, <tr><th class="infobox-label" scope="row" style="white-space: nowrap; padding-righ
<ul><li><a href="/wiki/Joe_Roth" title="Joe Roth">Joe Roth</a></li>
<li><a href="/wiki/Jeff_Kirschenbaum" title="Jeff Kirschenbaum">Jeff Kirschenbaum</a></li>
<li><a href="/wiki/Susan_Downey" title="Susan Downey">Susan Downey</a></li></ul>
</div></td></tr>, <tr><th class="infobox-label" scope="row" style="white-space: nowrap; padding-righ
<ul><li><a href="/wiki/Robert_Downey_Jr." title="Robert Downey Jr.">Robert Downey Jr.</a></li>
<li><a href="/wiki/Antonio_Banderas" title="Antonio Banderas">Antonio Banderas</a></li>
<li><a href="/wiki/Michael_Sheen" title="Michael Sheen">Michael Sheen</a></li>
<li><a href="/wiki/Harry_Collett" title="Harry Collett">Harry Collett</a></li>
<li><a href="/wiki/Emma_Thompson" title="Emma Thompson">Emma Thompson</a></li>
<li><a href="/wiki/Rami_Malek" title="Rami Malek">Rami Malek</a></li>
<li><a href="/wiki/John_Cena" title="John Cena">John Cena</a></li>
<li><a href="/wiki/Kumail_Nanjiani" title="Kumail Nanjiani">Kumail Nanjiani</a></li>
<li><a href="/wiki/Octavia_Spencer" title="Octavia Spencer">Octavia Spencer</a></li>
<li><a href="/wiki/Tom_Holland" title="Tom Holland">Tom Holland</a></li>
<li><a href="/wiki/Ralph_Fiennes" title="Ralph Fiennes">Ralph Fiennes</a></li>
<li><a href="/wiki/Selena_Gomez" title="Selena Gomez">Selena Gomez</a></li>
<li><a href="/wiki/Marion_Cotillard" title="Marion Cotillard">Marion Cotillard</a></li></ul>
</div></td></tr>, <tr><th class="infobox-label" scope="row" style="white-space: nowrap; padding-righ
<ul><li><a href="/wiki/Craig_Alpert" title="Craig Alpert">Craig Alpert</a></li>
<li><a href="/wiki/Chris_Lebenzon" title="Chris Lebenzon">Chris Lebenzon</a></li></ul>
</div></td></tr>, <tr><th class="infobox-label" scope="row" style="white-space: nowrap; padding-righ
<ul><li><a href="/wiki/Roth/Kirschenbaum_Films" title="Roth/Kirschenbaum Films">R/K Films</a></li>
<li><a href="/wiki/Team_Downey" title="Team Downey">Team Downey Productions</a></li>
<li><a href="/wiki/Perfect_World_Pictures" title="Perfect World Pictures">Perfect World Pictures</a>
</div></div></td></tr>, <tr><th class="infobox-label" scope="row" style="white-space: nowrap; paddin
<ul><li>January 17, 2020<span style="display:none"> (<span class="bday dtstart published updated">20
</div></td></tr>, <tr><th class="infobox-label" scope="row" style="white-space: nowrap; padding-righ
<ul><li>United States</li>
<li>United Kingdom</li></ul>
</div></td></tr>, <tr><th class="infobox-label" scope="row" style="white-space: nowrap; padding-righ
```

encounters the same problem, then can skip the previous steps and jump straight to the issue of

refining URLs.

Conclusion and Future Work

Although we faced a lot of setbacks, this project was a great learning experience and simply the beginning of what can be a great venture. There is an immense amount of scope and potential that must be explored. Moving forward, we still believe that creating a front end is the best way to go. We believe that this project deserves a dedicated website that can be used by anyone. Of course, without the constraints of time, we will be able to further develop the model with rigorous feature engineering, bringing in more temporal aspects. A time series might be a better option to pursue since we would be able to capture the impact over time most accurately. Furthermore, adding interaction variables can also prove to be beneficial. Maybe a certain actor-director pair has more influence on movie revenue. There are a countless number of aspects we could bring into our model that could benefit our model.

For web scraping, besides the steps we mentioned above, we believe once we proceed to the scraping part, we can get source material as one another features to add on when training our model.

Extra

1. This Addendum modifies and supplements the attached group project agreement (the "Project Agreement") concerning the project titled _Can You Make a Hit Movie?_____ (including any supplementary materials) in Fall 2022 Capstone SEAS 6402.
2. NiThe parties to the Project Agreement as modified and supplemented by this Addendum are: _Nicole Xie__ (member A), and course Professor Joe Goldfrank, and ____Annulekha Boddu_____ (member B).
3. The parties agree that they will follow the work distribution on the Project Agreement, if there's change needed, then the parties will rearrange the work distribution together until all parties agree. Wherever there is any conflict between this Addendum and the Project Agreement, the provisions of this Addendum will control and the Project Agreement will be construed accordingly.

MEMBER A_____Nicole_____Date

MEMBER B_____Anulekha_____Date

	Anulekha	Nicole
19-Oct	EDA & Basic Model	Web Scraping
26-Oct	Interim Proposal	Interim Proposal
2-Nov		
9-Nov	Model Training & Testing	Data/Feature Engineering
16-Nov		
23-Nov	Functional Visualization	
30-Nov	Add niceties	More modelling/help with viz
7-Dec	Finalize Everything	Finalize

References

Abidi Syed, Muhammad R., et al. "Popularity Prediction of Movies: From Statistical Modeling to Machine Learning Techniques." *Multimedia Tools and Applications*, vol. 79, no. 47-48, 2020, pp. 35583-35617. *ProQuest*, http://proxygw.wrlc.org/login?url=https://www.proquest.com/scholarly-journals/popularity-prediction-movies-statistical-modeling/docview/2473396437/se-2, doi:https://doi.org/10.1007/s11042-019-08546-5.

Chiranjib Paul & Prabir Kumar Das (2022) Predicting movie revenue before committing significant investments, Journal of Media Economics, 34:2, 63-90, DOI: 10.1080/08997764.2022.2066108

Dan Cocuzzo, Stephen Wu, Hit or Flop: Box Office Prediction for Feature Films, Stanford University, 2013

Freedman, David. *Statistical Models : Theory and Practice*. Second edition. Cambridge: Cambridge University Press, 2009. Print.

Gross, Jürgen. Linear Regression. Berlin ;: Springer, 2003. Print.

Hsu, Ping-Yu, Yuan-Hong Shen, and Xiang-An Xie. “Predicting Movies User Ratings with Imdb Attributes.” Lecture Notes in Computer Science (including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics). Vol. 8818. Cham: Springer International Publishing, 2014. 444–453. Web.

“Linearity.” *Wikipedia*, 9 Oct. 2022, en.wikipedia.org/wiki/Linearity#In_mathematics. Accessed 27 Oct. 2022.

Liu, A., Mazumdar, T., & Li, B. (2015). Counterfactual decomposition of movie star effects with star selection. *Management Science*, 61(7), 1704–1721. https://doi-org.proxygw.wrlc.org/10.1287/mnsc.2014.1923

Madsen, Henrik. *Time Series Analysis*. Boca Raton: Chapman & Hall/CRC, 2008. Print.

Ni, Yuan, et al. "Movie Box Office Prediction Based on Multi-Model Ensembles." *Information*, vol. 13, no. 6, 2022, pp. 299. *ProQuest*, http://proxygw.wrlc.org/login?url=https://www.proquest.com/scholarly-journals/movie-box-office-prediction-based-on-multi-model/docview/2679727931/se-2, doi:https://doi.org/10.3390/info13060299.