CS105 Final Project Report
Movies Data Analysis
Team 4: Ellen Yim, Hannah Bach, Connie Pak, Linda Ly, Huiwen Chen

Project Description

Our project goal is to analyze a dataset that focuses on movie statistics and determine how certain factors affect each other. The features of a movie that we would like to focus on especially is score, votes, gross, and budget. We want to perform exploratory data analysis to better understand and capture interesting information about our dataset. We also want to perform KNN Regression to make predictions of a movie's features using other features of the movie.

Data

For data collection, we used the 'movies.csv' file found in a git repository (https://github.com/danielgrijalva/movie-stats/blob/master/movies.csv).

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
1	name	rating	genre	year	released	score	votes	director	writer	star	country	budget	gross	company	runtime
2	The Shinin	R	Drama	1980	June 13, 19	8.4	927000	Stanley Ku	Stephen Ki	Jack Nicho	United Kin	19000000	46998772	Warner Br	146
3	The Blue L	R	Adventure	1980	July 2, 198	5.8	65000	Randal Kle	Henry De V	Brooke Shi	United Sta	4500000	58853106	Columbia I	104
4	Star Wars:	PG	Action	1980	June 20, 19	8.7	1200000	Irvin Kersh	Leigh Brac	Mark Ham	United Sta	18000000	5.38E+08	Lucasfilm	124
5	Airplane!	PG	Comedy	1980	July 2, 198	7.7	221000	Jim Abraha	Jim Abraha	Robert Ha	United Sta	3500000	83453539	Paramoun	88
6	Caddyshac	R	Comedy	1980	July 25, 19	7.3	108000	Harold Ra	Brian Doyl	Chevy Cha	United Sta	6000000	39846344	Orion Pictu	98
7	Friday the	R	Horror	1980	May 9, 198	6.4	123000	Sean S. Cu	Victor Mill	Betsy Palm	United Sta	550000	39754601	Paramoun	95
8	The Blues	R	Action	1980	June 20, 19	7.9	188000	John Landi	Dan Aykro	John Belus	United Sta	27000000	1.15E+08	Universal P	133
9	Raging Bul	R	Biography	1980	December	8.2	330000	Martin Scc	Jake LaMo	Robert De	United Sta	18000000	23402427	Chartoff-V	129
10	Superman	PG	Action	1980	June 19, 19	6.8	101000	Richard Le	Jerry Siege	Gene Hack	United Sta	54000000	1.08E+08	Dovemeac	127
11	The Long F	R	Biography	1980	May 16, 19	7	10000	Walter Hill	Bill Bryden	David Carr	United Sta	10000000	15795189	United Art	100
12	Any Which	PG	Action	1980	December	6.1	18000	Buddy Van	Stanford S	Clint Eastw	United Sta	15000000	70687344	The Malpa	116
13	The Gods I	PG	Adventure	1980	October 2	7.3	54000	Jamie Uys	Jamie Uys	N!xau	South Afric	5000000	30031783	C.A.T. Film	109
14	Popeye	PG	Adventure	1980	December	5.3	30000	Robert Alt	Jules Feiffe	Robin Willi	United Sta	20000000	49823037	Paramoun	114
15	Ordinary P	R	Drama	1980	September	7.7	49000	Robert Re	Judith Gue	Donald Sut	United Sta	6000000	54766923	Paramoun	124
16	Dressed to	R	Crime	1980	July 25, 19	7.1	37000	Brian De P	Brian De P	Michael Ca	United Sta	6500000	31899000	Filmways P	104
17	Somewher	PG	Drama	1980	October 3,	7.2	27000	Jeannot Sz	Richard M	Christophe	United Sta	5100000	9709597	Rastar Pict	103
18	Fame	R	Drama	1980	May 16, 19	6.6	21000	Alan Parke	Christophe	Eddie Bart	United States		21202829	Metro-Gol	134
19	9 to 5	PG	Comedy	1980	December	6.9	29000	Colin Higgi	Patricia Re	Jane Fonda	United Sta	10000000	1.03E+08	IPC Films	109
20	The Fog	R	Horror	1980	February 8	6.8	66000	John Carpe	John Carpe	Adrienne E	United Sta	1000000	21448782	AVCO Emb	89
21	Stir Crazy	R	Comedy	1980	December	6.8	26000	Sidney Poi	Bruce Jay I	Gene Wild	United States		1.01E+08	Columbia I	111

We cleaned the dataset by dropping any unnecessary columns and removing any null rows. The columns we decided to keep are genre, score, votes, budget, and gross. We also replaced the data in the 'genre' column with numerical values.

Description of each of the columns that we will be using for analysis:

- genre: main genre of the movie
- score: IMDb user rating
- votes: number of user votes
- budget: the budget of a movie
- gross: revenue of the movie

name	genre	score	votes	budget	gross
The Shining	Drama	8.4	927000.0	19000000.0	46998772.0
The Blue Lagoon	Adventure	5.8	65000.0	4500000.0	58853106.0
Star Wars: Episode V - The Empire Strikes Back	Action	8.7	1200000.0	18000000.0	538375067.0
Airplane!	Comedy	7.7	221000.0	3500000.0	83453539.0
Caddyshack	Comedy	7.3	108000.0	6000000.0	39846344.0
...	...	...	...	...	...
Bad Boys for Life	Action	6.6	140000.0	90000000.0	426505244.0
Sonic the Hedgehog	Action	6.5	102000.0	85000000.0	319715683.0
Dolittle	Adventure	5.6	53000.0	175000000.0	245487753.0
The Call of the Wild	Adventure	6.8	42000.0	135000000.0	111105497.0
The Eight Hundred	Action	6.8	3700.0	80000000.0	461421559.0

5436 rows × 5 columns

name	genre	score	votes	budget	gross
The Shining	1	8.4	927000.0	19000000.0	46998772.0
The Blue Lagoon	2	5.8	65000.0	4500000.0	58853106.0
Star Wars: Episode V - The Empire Strikes Back	3	8.7	1200000.0	18000000.0	538375067.0
Airplane!	4	7.7	221000.0	3500000.0	83453539.0
Caddyshack	4	7.3	108000.0	6000000.0	39846344.0
...	...	...	...	...	...
Bad Boys for Life	3	6.6	140000.0	90000000.0	426505244.0
Sonic the Hedgehog	3	6.5	102000.0	85000000.0	319715683.0
Dolittle	2	5.6	53000.0	175000000.0	245487753.0
The Call of the Wild	2	6.8	42000.0	135000000.0	111105497.0
The Eight Hundred	3	6.8	3700.0	80000000.0	461421559.0

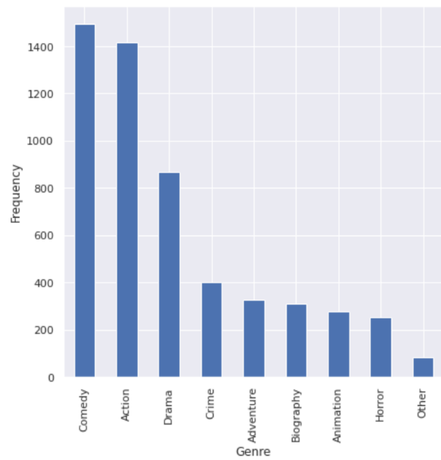
We then used Min-Max Normalization for our preprocessing, except for the ‘genre’ column. For this, we had to drop the ‘genre’ column first, then perform the normalization on the other columns, and then add the ‘genre’ column back so that it would not be affected by the calculations.

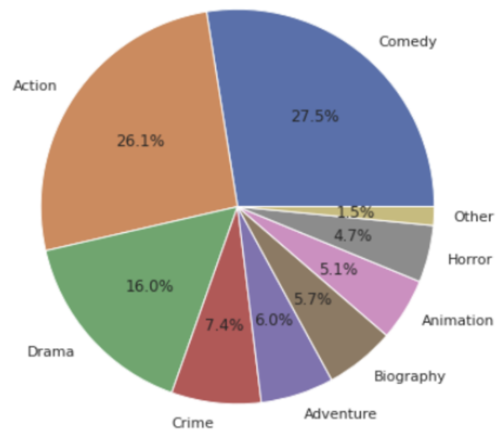
name	score	votes	budget	gross	genre
The Shining	0.878378	0.386200	0.053355	0.016507	1
The Blue Lagoon	0.527027	0.027004	0.012624	0.020670	2
Star Wars: Episode V - The Empire Strikes Back	0.918919	0.499959	0.050546	0.189086	3
Airplane!	0.783784	0.092010	0.009815	0.029310	4
Caddyshack	0.729730	0.044922	0.016837	0.013995	4
...	...	...	...	...	...
Bad Boys for Life	0.635135	0.058257	0.252796	0.149796	3
Sonic the Hedgehog	0.621622	0.042422	0.238751	0.112289	3
Dolittle	0.500000	0.022004	0.491564	0.086219	2
The Call of the Wild	0.662162	0.017420	0.379203	0.039022	2
The Eight Hundred	0.662162	0.001461	0.224706	0.162059	3

EDA

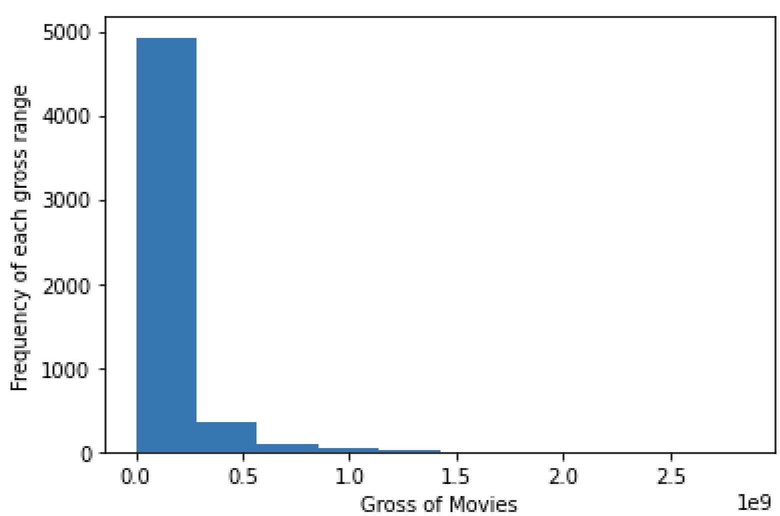
For Exploratory Data Analysis, we explored and analyzed the relationships between features that we need. We used pie charts, bar graphs, histogram, scatter plots, and boxplot using matplotlib.pyplot library to create our visualizations. To start off, we took a look at the frequency of genres, frequency of movie budget, and frequency of gross.

The frequency of genres, in both pie and bar chart below, we see that comedy is the most frequent in three decades of movie statistics data. With Action coming in a close second.

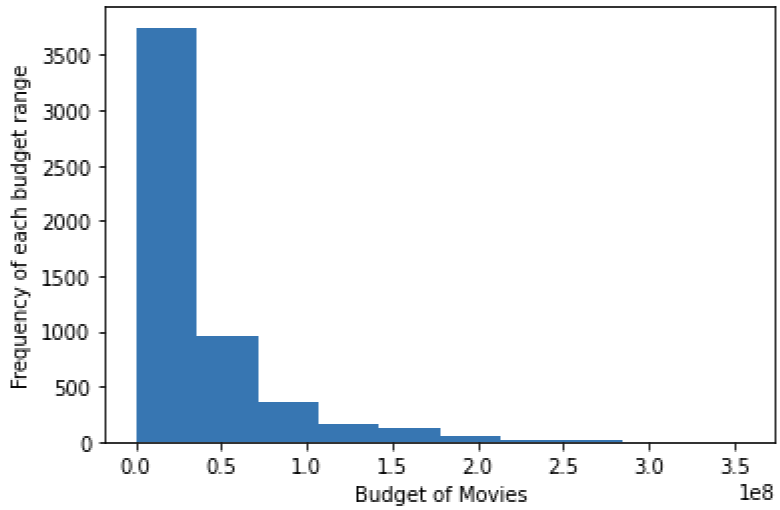




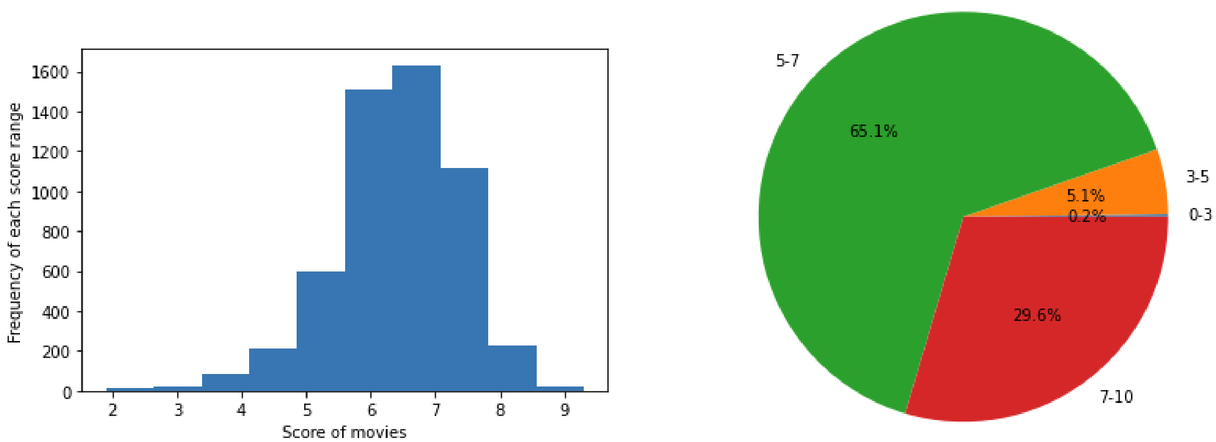
For the gross of movies, the frequency of each gross range, it is skewed right with the approximate bin size of 0.1 to 0.3, where the most common movie gross is between $100 million to $300 million.



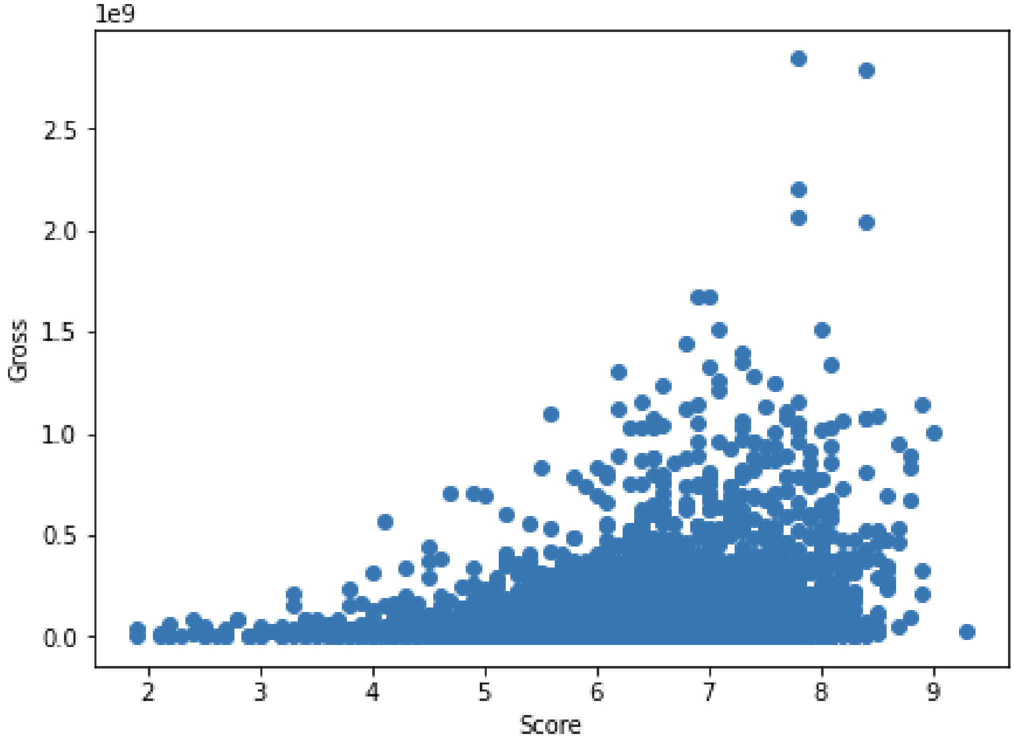
For the budget of movies, and looking at the frequency of each budget range, it is also skewed right, similar to frequency of each gross range, and the approximate bin size is 0.2 to 0.4, where the most common movie gross is between $20 million to $40 million.



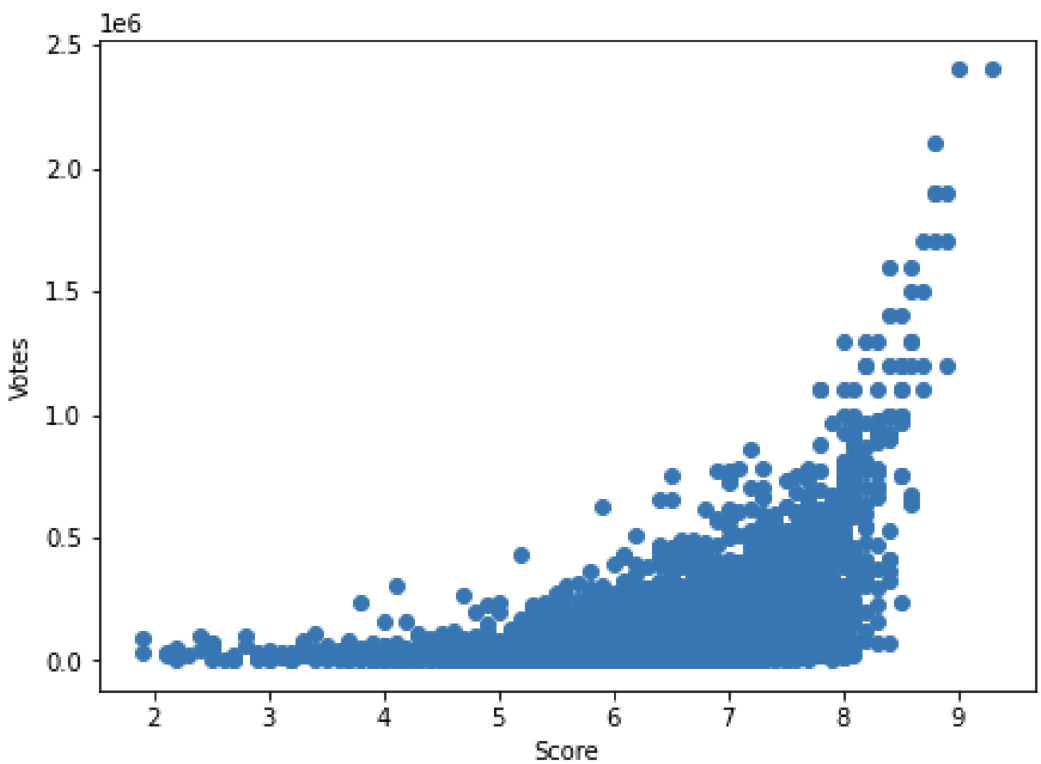
The frequency of movie scores, we see that the majority of the score range lies within 5 to 7 and that the data is skewed fairly to the left.



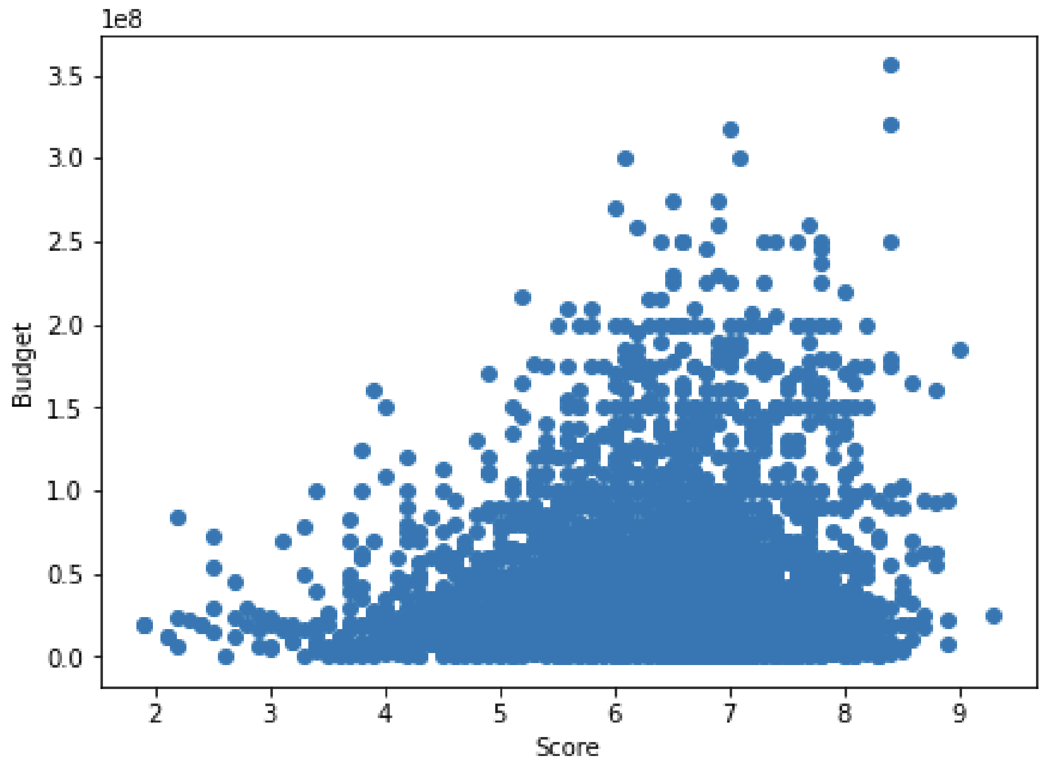
Using scatter plots to observe the relationship between score and gross, as well as with score and budget, and score and votes. Looking at score and gross, it’s a strong, positive relationship that has about 5 outliers for movies that have high scores and high gross.



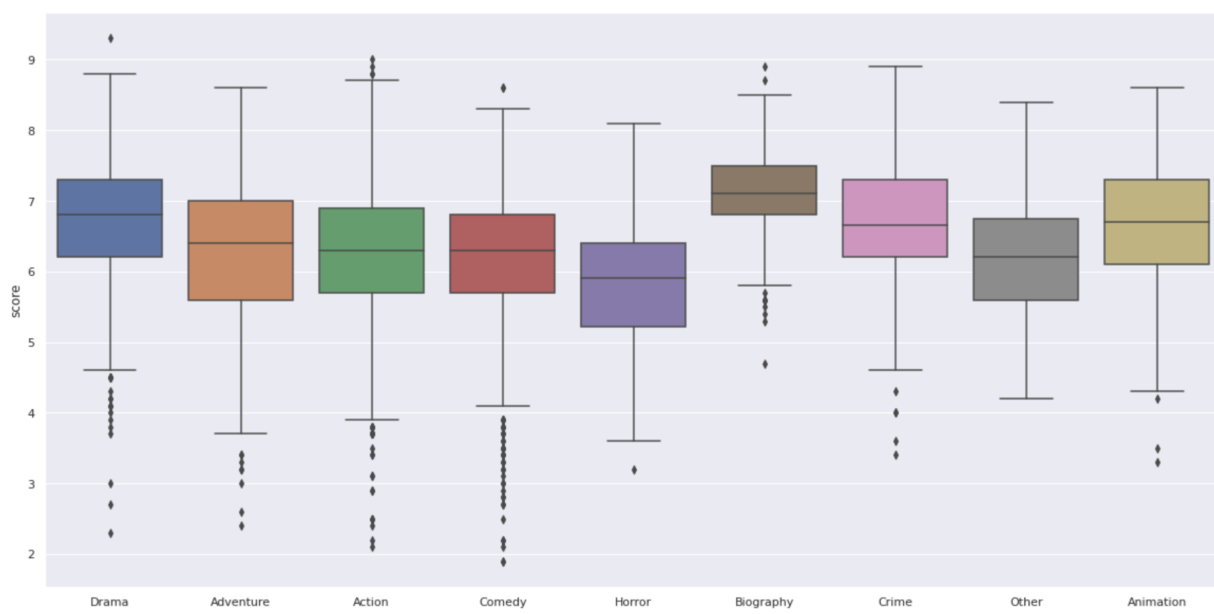
Looking at score and votes, which have a moderately strong, positive, linear relationship with few outliers that are not too far away from the cluster.



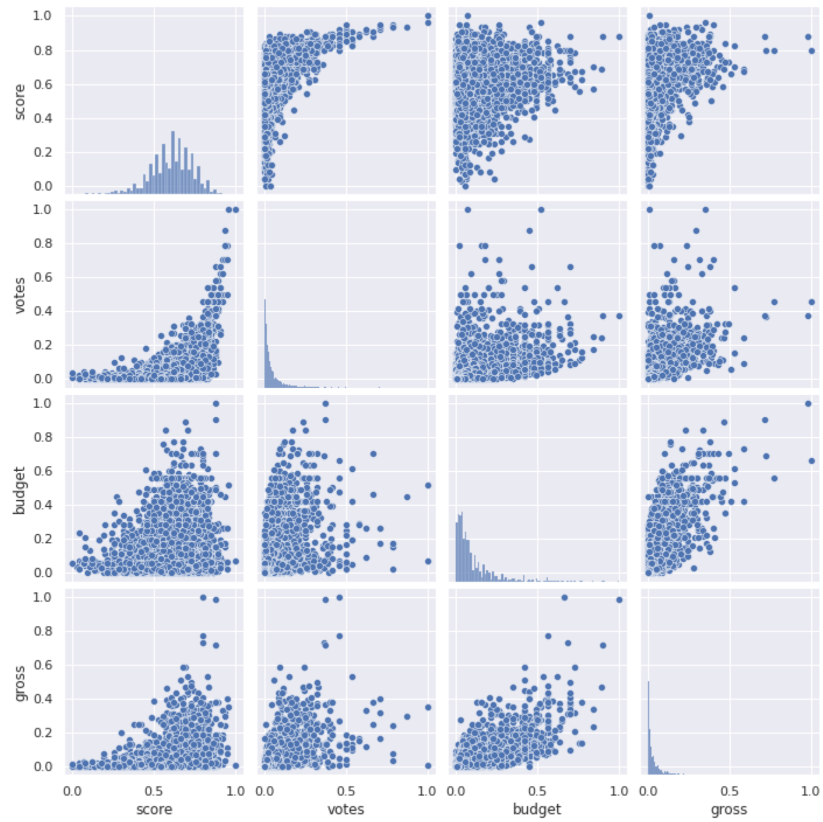
Looking at score and budget, we can see that the cluster is in the score range of 4 to 8 with a small budget. We see that there is a moderate relationship between score and budget.



Using a boxplot, we explored the minimum, maximum, median, and also had outliers for each movie genre. We can see that the average score for each genre is within the range of 6 to 7.



To determine which features are most similar, we looked at a pairplot using the seaborn library to see the relationships between each pair of features that we are using. We saw that budget and gross, score and votes, score and budget, score and gross have a close relationship with each other, seeing that the histogram is equally distributed with score and moderately strong relationship with budget and gross. Scores and votes have a strong, linear relationship.



Technique - KNN Regression

In this project, we decided to use KNN regression to build our machine learning models. KNN regression functions by taking the k nearest values of a test variable and computing the average of these values. We use this technique to predict the score of a movie based on a variety of factors such as the gross and budget. We also used KNN regression to predict the gross of a movie using the budget of a movie as a feature.

One tool we used was sklearn which is a machine learning library in python. We used the train_test_split function to split the data between train and test data with a size of 0.30 meaning that 30% of the data is used as test data. We chose this number because a 70:30 ratio is generally good when splitting training and testing data. To create our model, we used the

KNeighborsRegressor class from sklearn. We included a column with “error” which is the actual value minus the predicted value. We also performed the mean squared error test on each model to analyze their accuracy.

The first KNN Regression calculation was for k = 5, where the model takes the average of its 5 nearest neighbors (i.e. movies) to make the prediction. In this model, we focused on predicting the score from gross and budget.

name	Actual Val	Prediction	Error
What's the Worst That Could Happen?	0.486486	0.605405	-0.118919
The Brady Bunch Movie	0.567568	0.600000	-0.032432
Cop	0.608108	0.613514	-0.005405
Harry Potter and the Half-Blood Prince	0.770270	0.645946	0.124324
My Family	0.729730	0.605405	0.124324
...	...	...	...
50 First Dates	0.662162	0.635135	0.027027
200 Cigarettes	0.554054	0.629730	-0.075676
Man's Best Friend	0.445946	0.548649	-0.102703
Let's Be Cops	0.608108	0.675676	-0.067568
Paddington 2	0.797297	0.751351	0.045946

1088 rows × 3 columns

(Normalized)

name	Actual Val	Prediction
What's the Worst That Could Happen?	5.5	6.38
The Brady Bunch Movie	6.1	6.34
Cop	6.4	6.44
Harry Potter and the Half-Blood Prince	7.6	6.68
My Family	7.3	6.38
...	...	...
50 First Dates	6.8	6.60
200 Cigarettes	6.0	6.56
Man's Best Friend	5.2	5.96
Let's Be Cops	6.4	6.90
Paddington 2	7.8	7.46

1088 rows × 2 columns

(Original Values)

```
Mean Squared Error: 0.01850244464558349
```

For the same prediction, we then focused on k = 7.

name	budget	gross
The Shining	0.053355	0.016507
The Blue Lagoon	0.012624	0.020670
Star Wars: Episode V - The Empire Strikes Back	0.050546	0.189086
Airplane!	0.009815	0.029310
Caddyshack	0.016837	0.013995
...	...	...
Bad Boys for Life	0.252796	0.149796
Sonic the Hedgehog	0.238751	0.112289
Dolittle	0.491564	0.086219
The Call of the Wild	0.379203	0.039022
The Eight Hundred	0.224706	0.162059

5436 rows × 2 columns

(Independent Variables)

name	Score Test	Score Predicted
Troy	7.2	7.242857
Hostel	5.9	6.742857
Ghost in the Shell	6.3	6.242857
Diary of a Wimpy Kid: The Long Haul	4.3	6.271429
House of the Dead	2.1	6.485714
...	...	...
Aliens vs. Predator: Requiem	4.6	6.400000
Elizabethtown	6.4	6.242857
Chernobyl Diaries	5.0	6.542857
The Rocketeer	6.5	6.042857
Horrible Bosses	6.8	6.757143

1631 rows × 2 columns

(Score Predicted)

```
Mean Squared Error: 0.016547675814676893
```

We wanted to find the best k value for our dataset that gives us the minimum error possible. Starting with k = 5 for our first regression model, we used gross and budget as our training dataset to predict the score for a given movie, we saw it gave a mean squared error of 0.0185. In a second attempt with KNN regression using k=7, we saw it gave a mean squared error of 0.0165 which is a smaller error loss than k=5. The more neighbors to the training set for gross and budget, the smaller the error loss would be when predicting score. However, since we found that the accuracy did not make a huge difference when increasing the k from 7, we found that 7 was a good approximation for the number of nearest neighbors.

name	Score Test	gross
Troy	7.2	497409852.0
Hostel	5.9	81979826.0
Ghost in the Shell	6.3	169846945.0
Diary of a Wimpy Kid: The Long Haul	4.3	40140972.0
House of the Dead	2.1	13818181.0
...	...	...
Aliens vs. Predator: Requiem	4.6	130290885.0
Elizabethtown	6.4	52164016.0
Chernobyl Diaries	5.0	38390020.0
The Rocketeer	6.5	46704056.0
Horrible Bosses	6.8	209838559.0

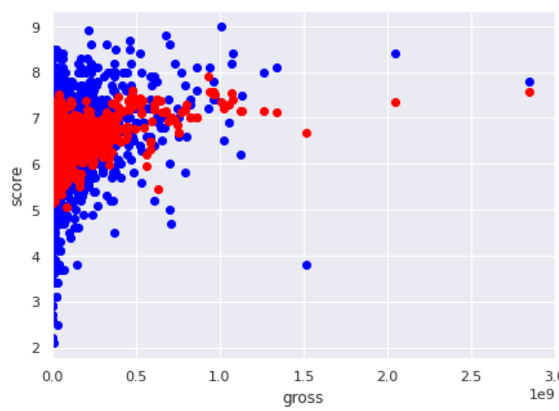
1681 rows × 2 columns

(Actual Score)

name	Score Predicted	gross
Troy	6.771429	497409852.0
Hostel	5.800000	81979826.0
Ghost in the Shell	6.571429	169846945.0
Diary of a Wimpy Kid: The Long Haul	6.957143	40140972.0
House of the Dead	6.071429	13818181.0
...	...	...
Aliens vs. Predator: Requiem	6.528571	130290885.0
Elizabethtown	7.271429	52164016.0
Chernobyl Diaries	5.571429	38390020.0
The Rocketeer	6.385714	46704056.0
Horrible Bosses	7.171429	209838559.0

1681 rows × 2 columns

(Score Predicted)




```
Coefficient of Determination: 0.1919846162948563
Mean Squared Error: 0.017868701538028216
Root Mean Squared Error: 0.1336738625836338
```

From our previous k values, we saw that with k=7, it gave a smaller error so we used that for the following KNN regressions.

For our next KNN calculation, we focused on predicting the score based on gross being an independent variable. After getting our KNN model with k=7, we needed to check our model's prediction accuracy. We saw the mean squared error for gross and predicted score to be 0.0178. Looking above at our tables, we can see that the score predicted and actual score are fairly good, except for a few outliers like the predicted score and actual score for "House of the Dead".

name	Score Test	budget
Troy	7.2	175000000.0
Hostel	5.9	4800000.0
Ghost in the Shell	6.3	110000000.0
Diary of a Wimpy Kid: The Long Haul	4.3	22000000.0
House of the Dead	2.1	12000000.0
...	...	...
Aliens vs. Predator: Requiem	4.6	40000000.0
Elizabethtown	6.4	45000000.0
Chernobyl Diaries	5.0	1000000.0
The Rocketeer	6.5	35000000.0
Horrible Bosses	6.8	35000000.0

1681 rows × 2 columns

(Actual Score & Budget)

name	Score Predicted	budget
Troy	6.771429	175000000.0
Hostel	5.800000	4800000.0
Ghost in the Shell	6.571429	110000000.0
Diary of a Wimpy Kid: The Long Haul	6.957143	22000000.0
House of the Dead	6.071429	12000000.0
...	...	...
Aliens vs. Predator: Requiem	6.528571	40000000.0
Elizabethtown	7.271429	45000000.0
Chernobyl Diaries	5.571429	1000000.0
The Rocketeer	6.385714	35000000.0
Horrible Bosses	7.171429	35000000.0

1681 rows × 2 columns

(Predicted Score & Budget)


```
Coefficient of Determination: 0.005723942638615531
Mean Squared Error: 0.01761090079600409
Root Mean Squared Error: 0.13270606917546798
```

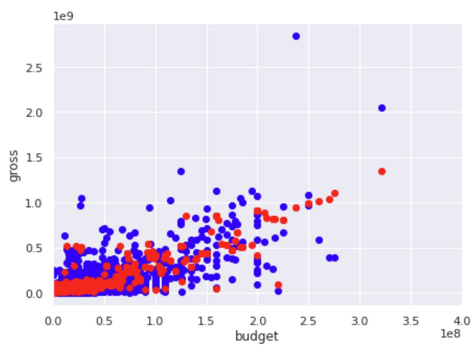
The next KNN regression calculation we performed was to predict the score based on the movie's budget, which was the independent variable here. We performed the mean squared error calculation as well and found it to be 0.01761 which means it's close to 0 so it's very accurate. Following the calculations, we also created a plot as shown above. The blue plots represent the budget vs actual score, while the red plots represent budget vs predicted score. Based on how similar the red and blue plots are to each other, just by looking at it, we can say that the prediction is very accurate as well. With the mean squared error value as well, we can determine that overall this model is accurate in predicting the scores based on a movie's budget.

name	Gross Test	budget
Troy	497409852.0	175000000.0
Hostel	81979826.0	4800000.0
Ghost in the Shell	169846945.0	110000000.0
Diary of a Wimpy Kid: The Long Haul	40140972.0	22000000.0
House of the Dead	13818181.0	12000000.0
...	...	...
Aliens vs. Predator: Requiem	130290885.0	40000000.0
Elizabethtown	52164016.0	45000000.0
Chernobyl Diaries	38390020.0	1000000.0
The Rocketeer	46704056.0	35000000.0
Horrible Bosses	209838559.0	35000000.0

(Actual Gross & Budget)

name	Gross Predicted	budget
Troy	4.727161e+08	175000000.0
Hostel	4.474432e+07	4800000.0
Ghost in the Shell	4.105951e+08	110000000.0
Diary of a Wimpy Kid: The Long Haul	1.293684e+08	22000000.0
House of the Dead	5.872859e+06	12000000.0
...	...	...
Aliens vs. Predator: Requiem	1.068343e+08	40000000.0
Elizabethtown	1.374568e+08	45000000.0
Chernobyl Diaries	2.714686e+07	1000000.0
The Rocketeer	6.723828e+07	35000000.0
Horrible Bosses	6.723828e+07	35000000.0

(Predicted Gross & Budget)




```
Coefficient of Determination: 0.005723942638615531
Mean Squared Error: 0.0022045732489578075
Root Mean Squared Error: 0.0469528832869485
```

The last KNN regression calculation we performed uses budget as the independent variable and predicts the gross of movies. In the plot above, the blue represents the actual gross against budget while the red represents the predicted gross against budget. From just looking at it, we can see that the predictions do fairly well and follow the general trend of the actual gross of movies given the budget. However, we wanted an actual value that could tell us about our model's accuracy. We performed mean squared error as a way to get some sort of account for accuracy. For this model, we got a mean squared error value of 0.0022 which we believe to be a good as it is close to 0. By comparing the actual and predicted values by eye, looking at the plot, and examining the mean squared error value, we believe this model's accuracy is fairly good.

Conclusion

The goal of this project is to analyze the characteristics of a movie such as the budget, score, and genre in order to predict another characteristic. Before building our machine learning models, we used the Min-Max normalization technique to prepare our data for analyses. We did this to ensure that the variables with values of high magnitude would not affect the variables with values of much smaller magnitudes during the training of our models. We used KNN regression to form predictions and calculated mean squared errors to assess the accuracy of our predictions.

Based on our results from the KNN Regression, we found that using two features, the gross and the budget, were the best in predicting the score of a movie. This is because it resulted in having the smallest mean squared error compared to the other two models we built in the prediction of the scores (using either gross or budget to predict a movie's score). Not only did we build models to predict a movie score, we also built a KNN Regression model to predict the gross of a movie using the budget as its feature. We found that the accuracy of this model was quite high when predicting the movie's gross given a budget as its input. For instance, when the model was given the budget of $175000000.00 for the movie "Troy" as test input, it predicted that the gross would be $472716100.00 which is quite close to its actual gross of $497409852.00.

Member's Contribution

We all worked on each part together, collaboratively:

Ellen Yim - Proposal, Data Collection/Cleaning/Preprocessing, EDA, KNN Regression, Report, Presentation, Writing Questions, Recording

Hannah Bach - Proposal, Data Collection/Cleaning/Preprocessing, EDA, KNN Regression, Report, Presentation, Writing Questions, Recording

Connie Pak - Proposal, Data Collection/Cleaning/Preprocessing, EDA, KNN Regression, Report, Presentation, Writing Questions, Recording

Linda Ly - Proposal, Data Collection/Cleaning/Preprocessing, EDA, KNN Regression, Report, Presentation, Writing Questions, Recording

Huiwen Chen - Proposal, Data Collection/Cleaning/Preprocessing, EDA, KNN Regression, Report, Presentation, Writing Questions, Recording

Presentation

Slides:
https://docs.google.com/presentation/d/1gKYbLi1198d1hHdyw7JnUZ0TldAq-sEEU4k0982cZJk/edit?usp=sharing
Recording:
https://drive.google.com/file/d/1vZqR3T2tUNc_8py08zluokjU2unp5uyP/view?usp=sharing

Sources

- https://www.datatechnotes.com/2019/04/regression-example-with-k-nearest.html
- https://www.kaggle.com/code/hamzatanc/k-nearest-neighbors-regression
- https://github.com/danielgrijalva/movie-stats
- https://www.datacamp.com/tutorial/understanding-logistic-regression-python