.

k -means clustering

Movie data analysis to identify popular movies based on budget, gross and movie Facebook likes

Sahidtiasadason Chindu

ABSTRACT

k -means clustering is a type of unsupervised learning, which serves many purposes. This report focuses on using R language to cluster a movie dataset based on budget and gross in order to find a suitable cluster of movies to be added to a company's movie database. 'Movie_metadata' published in Kaggle is the dataset used in this report. The distance measure used by the *k* -means clustering in this report is the Euclidean distance. Within Sum of Squares is used in conjunction with an elbow plot to identify the optimal number of clusters. The first analysis was carried out using the optimal *k* value, after which the *k* value is increased, and the analysis was carried out again. Using the knowledge gained from the two analysis, 100 movies were selected to be added to the company's database.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

1. BUSINESS UNDERSTANDING

1.1 Business objectives/Situation Assessment

Beta Ltd is an Australian company that offers movie streaming over the internet since the early 2000s. This company chooses random movies to be offered to their customers for streaming. In the initial stages of the company, the customer database of the company was gradually increasing.

The competition between movie streaming companies grew over the years and many companies started implementing data mining techniques to understand their customers better and provide them with better services. With these techniques, companies such as Netflix and Amazon started to dominate this industry.

In 2018, the company realised that they had lost over 70% of their customers despite their low monthly subscription. The company understood the importance of changing their marketing strategy. To improve the business, the boss of the company decided to start offering movies which are popular instead of randomly picking movies. The objective would be to attract more customers by increasing the movie database by 100 popular movies. The focus of the study would be to understand customers demand for movies based on gross, budget and Facebook likes.

1.2 Data mining goal/Project plan

The data-mining goal is to identify popular movies released over the past 15year. The idea would be to identify movies that people like rather than just basing it on scores like IMDB or Rotten Tomatoes. Another goal would be to understand the general reaction of the public to low budget, average budget and high budget movies and understanding if high gross and a low gross of the movies have any relation to Facebook likes. By understanding the budget, gross and how much the public loved the movie, questions such as ‘Do people only enjoy high budget movies?’ , ‘Does a low gross necessarily mean that people did not enjoy the movie?’ can be answered. Finally, the main aim would be to group movies based on budget and gross and to identify which group would be a better choice of movies to be offered to customers.

To achieve these goals, a dataset of movies released over the last 15 years has to be obtained. The dataset should include the budget of the movies, the gross of the movies and how people reacted to the movie, example Facebook likes and critics. These three variables would be base of the analysis. Depending on the other variables available on the dataset, further analysis could be carried out.

This project will be carried out in R studio using various packages. Visualisation techniques such as box plot and scatterplots are used for identification of outliers as well as to understand the data. Clustering would be the main techniques used to group the movies as well as to understand the dataset better.

2. DATA UNDERSTANDING

2.1 Data collection/Describe data

The dataset that was used in this analysis was obtained from the Kaggle website (https://www.kaggle.com/carolzhangdc/imdb-5000-movie-dataset). It contains 28 variables for 5043 movies, spanning across in 66 countries. There are 2399 unique director names and thousands of actors/actresses. The dataset contains the three main variables, budget, gross and Facebook likes which are relevant to this analysis. The data collected for each of these observations are from one week after the release date of the movies. All the three variables are numerical variables, and this makes it suitable for cluster analysis. The data contains much extra information, most of which will be eliminated during the data preparation process.

2.2 Explore data/Verify data quality

The first step carried out was to identify if there were any duplicate rows in the dataset. Using an R function, 45 duplicates were identified. These duplicated observations were removed before any data exploration was carried out. After removal, the dataset now contains 4998 rows and 28 variables.

The next step involved identifying missing values in the dataset. Figure 2.1 shows the proportion of data that is missing from each variable.

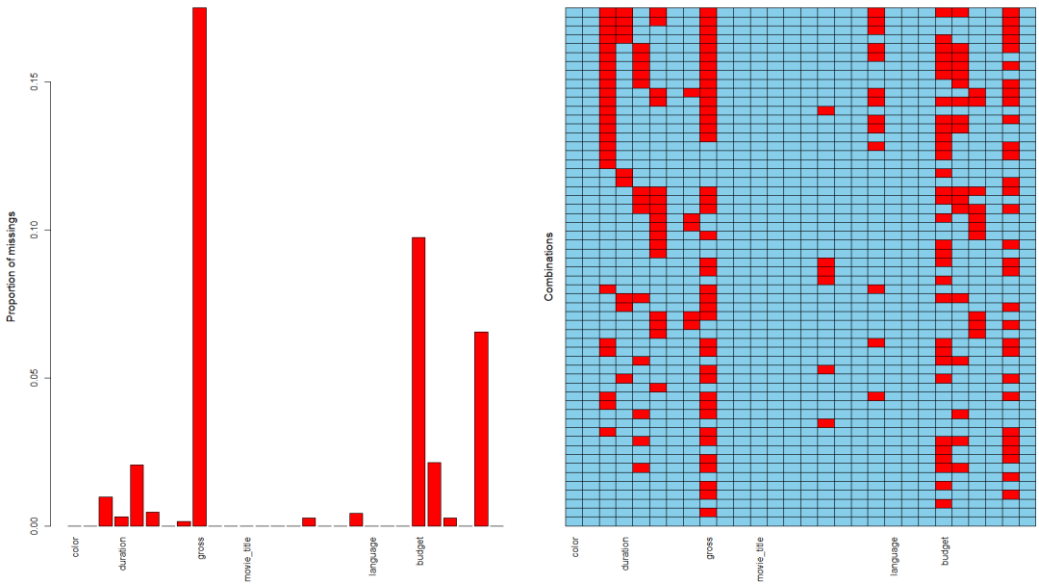


Figure 2. 1 Missing value visualisation

A missing value table(Table 2.1) was created to get a better understanding of the missing value. The tables display the breakdown of how many cases are missing from each variable. The main concern in this analysis is budget, gross and moive_facebook_likes. Both budget and gross has a considerable number of missing values. These have to be dealt with before clustering.

color	0	director_name	0
num_critic_for_reviews	49	duration	15
director_facebook_likes	103	actor_3_facebook_likes	23
actor_2_name	0	actor_1_facebook_likes	7
gross	874	genres	0
actor_1_name	0	movie_title	0
num_voted_users	0	cast_total_facebook_likes	0
actor_3_name	0	facenumber_in_poster	13
plot_keywords	0	movie_imdb_link	0
num_user_for_reviews	21	language	0
country	0	content_rating	0
budget	487	title_year	107
actor_2_facebook_likes	13	imdb_score	0
aspect_ratio	327	movie_facebook_likes	0

Table 2. 1 Summary of missing values

Since gross and budget have too many missing values and these two variables are required for the following analysis, the rows with null values have to be deleted because imputation will not do a good job here.

Visualisation techniques are used to get a better understanding of the three variables used for this analysis. Figure 2.2 and 2.3 shows the scatterplot of both gross and budget. Figure 2.2 reveals some extreme outliers in the variable ‘gross’. This trend is similar for the budget as shown in Figure 2.3. Figure 2.4 shows the scatterplot of movie_facebook_likes. From this scatter plot, it is noted that there is one extreme outlier; this should be eliminated for a more reliable comparison.

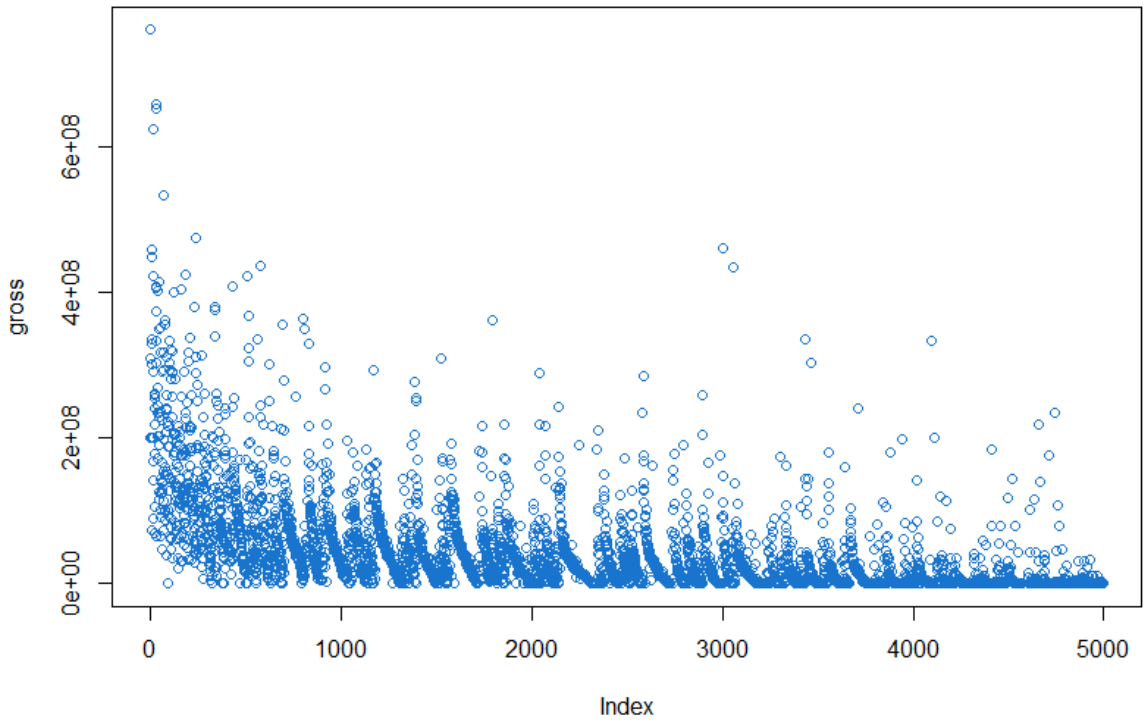


Figure 2. 2 Scatterplot for the variable Gross

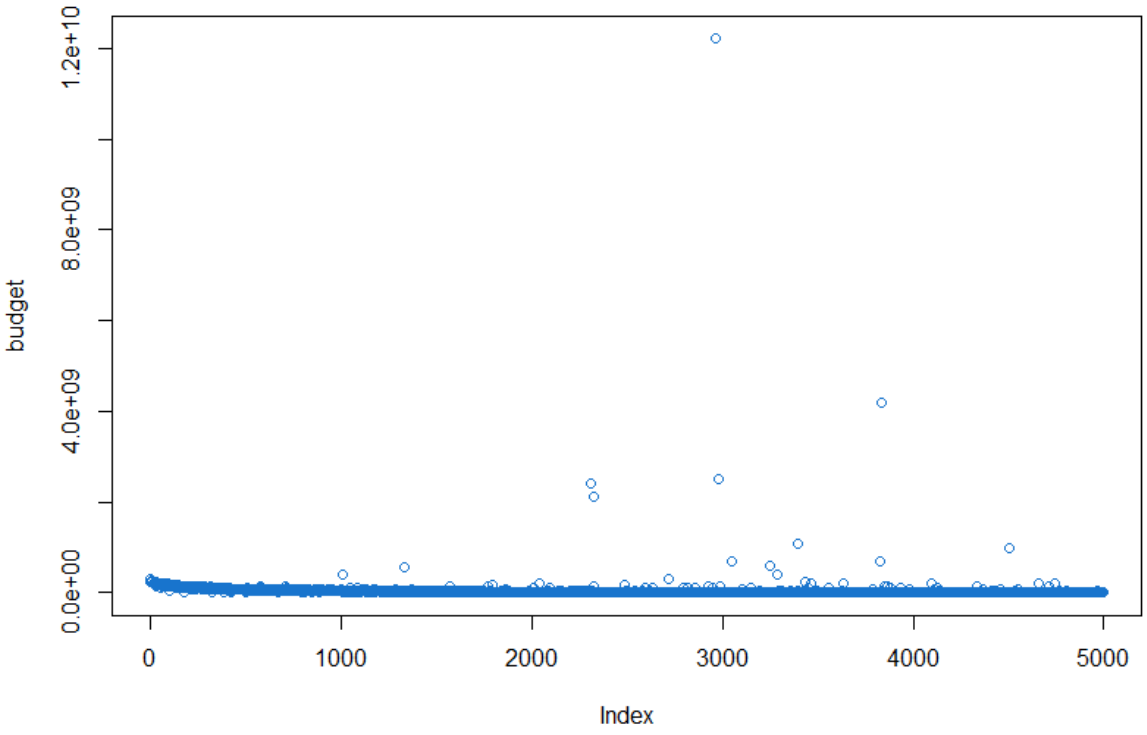


Figure 2. 3 Scatterplot for the variable budget

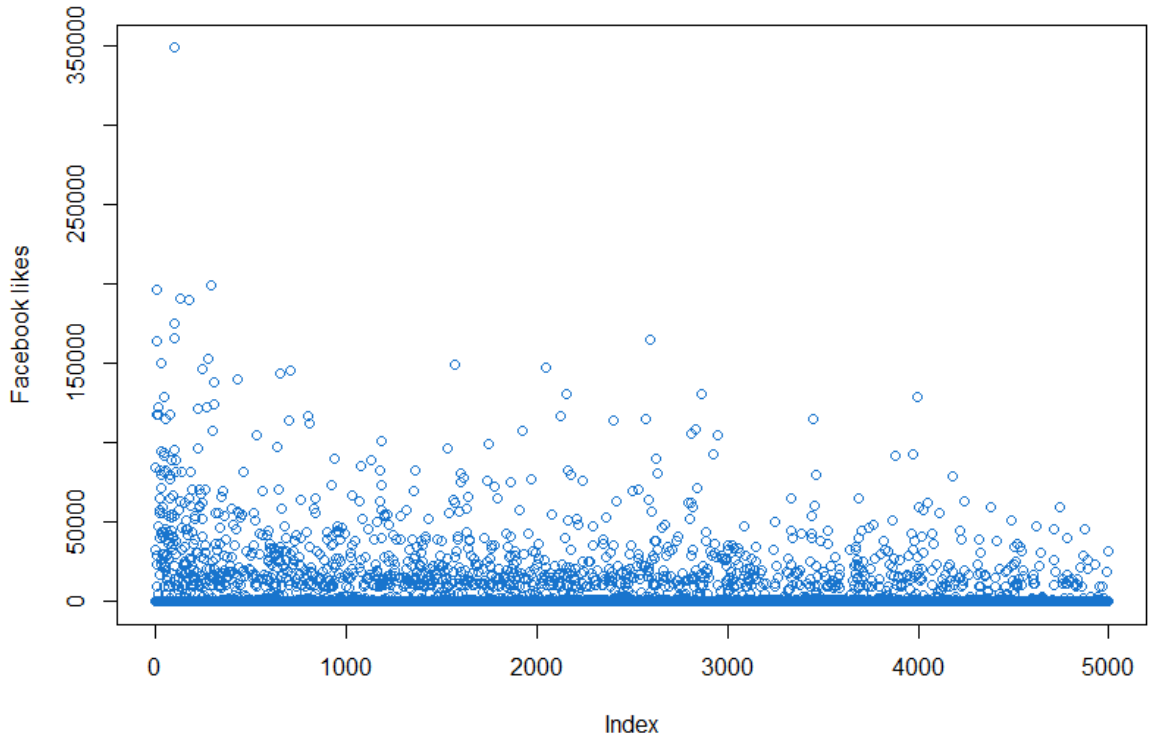


Figure 2. 4 Scatterplot for the variable movie_facebook_likes

The outlier detection is searching for objects in the database that do not obey laws valid for the major part of the data. In clustering, outliers are considered as observations that should be removed in order to make clustering more reliable (Vaishali & Mehta, 2011).

A summary statistics of the crucial variables were obtained using the summary function in R. Table 2.2 shows the summary statistics of the variables. From this table it was identified that this data has either some incorrect entries or the dataset does not consist of only movies. This conclusion was derived from the analysis that the minimum duration is ‘7' and max is ‘511'. Movie duration is usually minimum 60mins and a maximum of 200mins. This needs to be investigated further. Another interesting observation is the gross; the minimum gross is ‘162'. This could be an entry error and should be considered an outlier and removed. Further investigation of the variable ‘gross' is required. These extreme cases will be dealt with during data preparation. Similarly budget has a meagre value. There are also observations with zero Facebook likes, and this needs to be investigated further.

```
num_critic_for_reviews    duration     director_facebook_likes actor_3_facebook_likes     gross
Min.   :  1.0          Min.   :  7.0   Min.   :    0.0         Min.   :    0.0        Min.   :      162
1st Qu.: 50.0          1st Qu.: 93.0   1st Qu.:    7.0         1st Qu.:  133.0        1st Qu.:  5304835
Median :110.0          Median :103.0   Median :   49.0         Median :  369.0        Median : 25445749
Mean   :139.9          Mean   :107.2   Mean   :  688.7         Mean   :  639.9        Mean   : 48325649
3rd Qu.:195.0          3rd Qu.:118.0   3rd Qu.:  192.0         3rd Qu.:  635.0        3rd Qu.: 62319416
Max.   :813.0          Max.   :511.0   Max.   :23000.0         Max.   :23000.0        Max.   :760505847
NA's   :49             NA's   :15      NA's   :103             NA's   :23             NA's   :874
    budget          actor_1_facebook_likes num_voted_users   cast_total_facebook_likes facenumber_in_poster
Min.   :2.180e+02   Min.   :     0.0       Min.   :      5   Min.   :     0            Min.   : 0.000
1st Qu.:6.000e+06   1st Qu.:   611.5       1st Qu.:   8560   1st Qu.:  1406            1st Qu.: 0.000
Median :2.000e+07   Median :   984.0       Median :  34261   Median :  3086            Median : 1.000
Mean   :3.975e+07   Mean   :  6556.9       Mean   :  83470   Mean   :  9677            Mean   : 1.369
3rd Qu.:4.500e+07   3rd Qu.: 11000.0       3rd Qu.:  96121   3rd Qu.: 13740            3rd Qu.: 2.000
Max.   :1.222e+10   Max.   :640000.0       Max.   :1689764   Max.   :656730            Max.   :43.000
NA's   :487         NA's   :7                                                          NA's   :13
num_user_for_reviews    budget.1         actor_2_facebook_likes   imdb_score    movie_facebook_likes
Min.   :   1         Min.   :2.180e+02   Min.   :     0         Min.   :1.600   Min.   :     0.0
1st Qu.:  64         1st Qu.:6.000e+06   1st Qu.:   280         1st Qu.:5.800   1st Qu.:     0.0
Median : 156         Median :2.000e+07   Median :   595         Median :6.600   Median :   162.5
Mean   : 272         Mean   :3.975e+07   Mean   :  1643         Mean   :6.441   Mean   :  7487.4
3rd Qu.: 324         3rd Qu.:4.500e+07   3rd Qu.:   917         3rd Qu.:7.200   3rd Qu.:  3000.0
Max.   :5060         Max.   :1.222e+10   Max.   :137000         Max.   :9.500   Max.   :349000.0
NA's   :21           NA's   :487         NA's   :13
```

Table 2. 2 Summary statistics of all variables

3. DATA PREPARATION

3.1 Data cleaning

In this stage, the variables, ‘budget’, ‘gross’ and ‘movie_facebook_likes’ will be analysed further and the rows with missing value eliminated. Furthermore, a strategy to remove extreme cases(outliers) will be derived and implemented to deal with outliers. Having outliers in the dataset would affect the cluster analysis severely (Rajeev, Vaishali, & Rupa, 2011).

After analysing the dataset further, it was understood that the dataset not only contains movies but also contains TV shows and Documentaries. Furthermore, it was observed that movies with zero Facebook likes are generally movies released before 2005. This is because Facebook started gaining popularity after the year 2005.

To deal with these issues, a set of rules were created and applied to the data set. The rules are as follows:

Rule 1: eliminate observations with movie_facebook_likes < 500
Rule 2: eliminate observations with duration < 60
Rule 3: eliminate observations with duration > 200
Rule 4: eliminate observations with gross and budget < 40000
Rule 5: eliminate movies, which are non-English
Rule 6: eliminate movies released before 2005

Rule 1 was introduced to eliminate movies that have extremely low popularity. Rule 2 and 3 was used to filter out movies from the dataset. An assumption that a movie will be longer than 60mins and less than 200mins was used to create rule 2 and 3. Rule 4 filters out movies from TV series and Documentaries as well as eliminate any movies with a meagre budget or gross. Rule 5 was introduced, as the company only wants to offer English movies. Finally Rule 6, this was introduced because of its relationship with the variable ‘Movie_facebook_likes’. Facebook became more widely used around 2005, to have a reliable comparison of movie Facebook likes in each cluster, only movies released after 2005 were used for this analysis. Applying this set of rules reduced the dataset to 1571 observations.

After applying the set of rules, scatterplots (Figure 3.1) of the three numerical variables were created for outlier identification. From Figure 3.1 it can be observed that each of these three variables have some extreme values. In order to obtain more logical clusters, these variables should be eliminated from the dataset.

There are several ways to identify outliers. The first method tested to identify the outliers was the Box-plot method. Applying this method, detected a considerable number of movies as outliers. Eliminating these movies would defeat the purpose of the analysis as a considerable number of high gross and budget movies will be eliminated. (Refer to the appendix, BoxPlot).

Since it was critical in this analysis to keep the high budget and high gross movies, it was decided that movies with extreme gross, budget and Facebook likes would be removed by manual selection by defining a function in R that removes values larger than a specified number. These outliers will be analysed separately to determine if they should be added to the company's movie database.

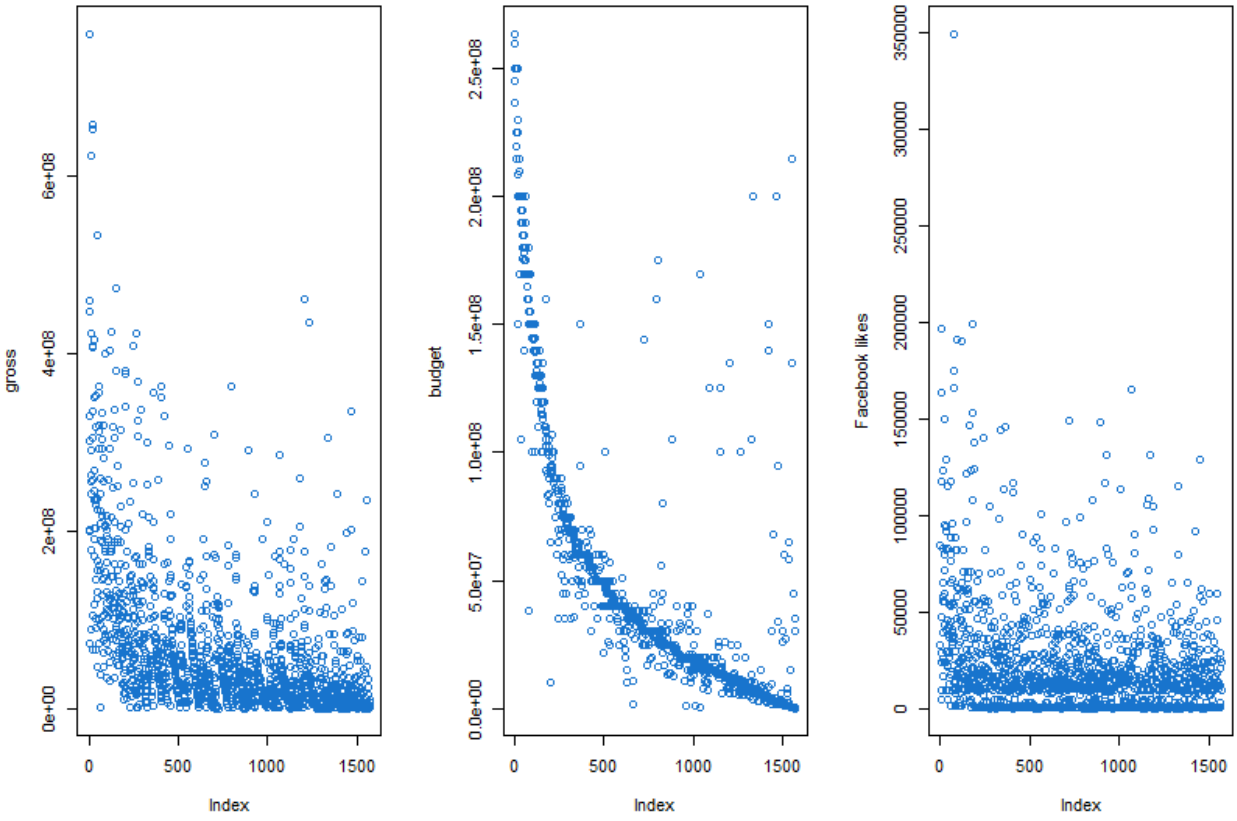


Figure 3. 1 Scatterplot of gross, budget and movie_facebook_likes

The grosses of the movies were arranged from the highest to the lowest. Table 3.1 shows a list of the top 8 gross movies. From the list the movies ‘Avatar’,’Titanic’,’Jurassic World’, ‘The Dark Knight Rises’ and ‘The Avengers' have an extreme gross value; hence these movies are eliminated and reserved for special analysis on if these should be included in the company's database.

gross	movie_title	budget	imdb_score	movie_facebook_likes
760505847	AvatarÂ	237000000	7.9	33000
658672302	TitanicÂ	200000000	7.7	26000
652177271	Jurassic WorldÂ	150000000	7.0	150000
623279547	The AvengersÂ	220000000	8.1	123000
533316061	The Dark KnightÂ	185000000	9.0	37000
474544677	Star Wars: Episode I - The Phantom MenaceÂ	115000000	6.5	13000
460935665	Star Wars: Episode IV - A New HopeÂ	11000000	8.7	33000
458991599	Avengers: Age of UltronÂ	250000000	7.5	118000

Table 3. 1 Data frame (arranged by decreasing gross)

Next, the budgets of the movies were arranged from highest to lowest (Table 3.2), from this list two movies are identified as extreme values, ‘John Carter' and ‘Tangled'. These movies were added to the reserved list for individual analysis

gross	movie_title	budget	imdb_score	movie_facebook_likes
73058679	John CarterÂ	263700000	6.6	24000
200807262	TangledÂ	260000000	7.8	29000
458991599	Avengers: Age of UltronÂ	250000000	7.5	118000
448130642	The Dark Knight RisesÂ	250000000	8.5	164000
407197282	Captain America: Civil WarÂ	250000000	8.2	72000
330249062	Batman v Superman: Dawn of JusticeÂ	250000000	6.9	197000
301956980	Harry Potter and the Half-Blood PrinceÂ	250000000	7.5	10000
255108370	The Hobbit: The Battle of the Five ArmiesÂ	250000000	7.5	65000

Table 3. 2 Data frame (arranged by decreasing budget)

Finally, the movie Facebook likes were arranged from highest to lowest (Table 3.3). From the list, the movie ‘Interstellar’ was identified to have an extreme value and hence added to the reserved list of movies for further analysis.

gross	movie_title	budget	imdb_score	movie_facebook_likes
187991439	InterstellarÂ	1.65e+08	8.6	349000
162804648	Django UnchainedÂ	1.00e+08	8.5	199000
330249062	Batman v Superman: Dawn of JusticeÂ	2.50e+08	6.9	197000
153629485	Mad Max: Fury RoadÂ	1.50e+08	8.1	191000
183635922	The RevenantÂ	1.35e+08	8.1	190000
292568851	InceptionÂ	1.60e+08	8.8	175000
303001229	The Hobbit: An Unexpected JourneyÂ	1.80e+08	7.9	166000

Table 3. 3 Data frame (arranged by decreasing movie Facebook likes)

After removing these observations, the dataset now consists of 1563 observations.

3.2 Variable selection

For this analysis, only four variables were selected from the entire dataset. These variables are Gross, Budget, Movie_facebook_likes and the Movie Title.

Gross and Budget will be used to create the cluster. Movie_facebook_likes is used to understand the cluster. Finally, the Movie Title is used to identify the names of the movies in the cluster.

3.3 Data normalisation

Distance computation in *k* -Means weights each dimension equally and hence care must be taken to ensure that unit of dimension do not distort relative near-ness of observations. If axes have different units and very different scale, normalisation is necessary.

Data Normalization standardises the raw data by converting them into a specific range using linear transformation which can generate good quality clusters and improve the accuracy of clustering algorithms. Some conventional methods of data normalisation are ‘Min-Max scaling' and ‘standardisation'.

Min-Max scaling is a method of rescaling data to have values between 0 and 1 (Stephanie, 2015). Min-Max scaling is defined by,

$$x_{new} = \frac{x - x_{min}}{x_{max} - x_{min}}, \tag{3.1}$$

Standardisation transforms data to have a mean of zero and standard deviation of one (Stephanie, 2015). This standardisation is known as Z-score and is defined by,

$$Z_i = \frac{x_i - \bar{x}}{s}, \tag{3,2}$$

Where x_i is a data point, $\bar{x}$ is the sample mean, and s is the sample standard deviation.

In this dataset, clustering is carried out using the variable budget and gross; these two variables have the same units as well as a very similar scale. Hence, data normalisation is not required. A test was carried out to check if the movies in the clusters varied if data was normalised and the results prove that both clusters formed are almost similar. (Refer to appendix 'Normalised clustering for the cluster result for normalised data)

4. MODELLING

4.1 Overview

The model used for this analysis is *k* -means clustering. Cluster analysis or clustering is the task of grouping a set of objects in such a way that objects in the same group are more similar to each other than to those in other groups (Lech, Yiyu, Piotr, & Davide, 2017). *k* -means clustering is a type of unsupervised learning which is used to label unlabelled data. The goal of this algorithm is to find groups in the data, with the number of groups represented by the variable *k* (Siddhant & Utkarsha, 2018). The algorithm works iteratively to assign each data point to one of the *k* groups based on the features that are provided. Data points are clustered based on feature similarity. The fundamental restriction for *k* -Means algorithm is that the data should be continuous. This method is not suitable for categorical variables, a similar method known as 'K-Modes clustering' works well with categorical data.

k -means clustering intends to partition n objects into *k* clusters in which each object belongs to the cluster with the nearest mean. This method produces exactly *k* different clusters of greatest possible distinction (Ankit, 2015). The best number of clusters leading to the greatest separation is not known as a priori and must be computed from the data. The objective of *k*-means clustering is to minimise total intracluster variance or the squared error function (Saed, 2010). The goal is to assign a cluster to each data point.

Clustering will group similar movies based on budget and gross into groups and analysis can be carried out on each group to understand better how people react to different types of movies as well as to identify a group of movies that can be introduced to the movie database.

4.2 Methodology

Figure 4.1 shows the process flow diagram of *k* -means clustering. This is an iterative process that terminates when the optimal minimum distance is achieved. *k*-Means runs on distance calculations, which uses Euclidean distance for this purpose. Euclidean distance is used to calculate the distance between two given points.

The equation for Euclidean distance is defined as,

$$d(x,y) = \sqrt{\sum_{i=1}^{n}(x_i - \mu_k)^2} \tag{4.1}$$

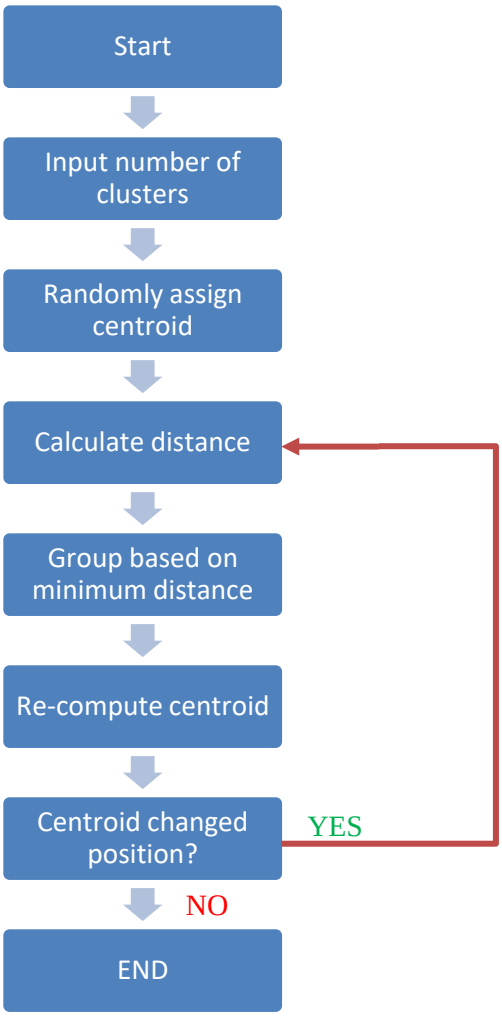


Figure 4. 1 Process flow diagram of K-Means

The process flow of *k* -means:

Step 1: Specify *k* - the number of clusters to be created.

Step 2: Select *k* random objects from the dataset as the initial cluster centres.

Step 3: Calculate Euclidean distance of each observation to the cluster centroids.

Step 4: Assign each observation to their closest centroid, based on the Euclidean distance between the object and the centroid.

Step 5: For each of the *k* clusters re-compute the cluster centroid by calculating the new mean value of all the data points in the cluster.

Step 6: Iteratively minimize the total within sum of square. Repeat Steps 3, 4 and 5, until the maximum number of iterations are reached or when the centroids stops changing position. (R uses ten as the default value for the maximum number of iterations).

When all observations are assigned to a cluster, the centroids have to be recalculated. Using these new centroids, a new binding has to be done between the same dataset points and the nearest new centre. A loop has been generated. Because of this loop, the *k* centres change their location step by step until there are no more changes in the centroids locations. This algorithm aims at minimising an objective function.

There are several *k*-means algorithms available to determine this objective function. The standard algorithm is Hartigan-Wong algorithm (1979), which defines the total within-cluster variation as the sum of squared Euclidean distances between dataset points and the corresponding centroid. The within-cluster variation is calculated as the sum of the Euclidean distance between the data points and their respective cluster centroids.

The within-cluster variation is defined as,

$$W(C_k) = \sum_{x_i \in C_k} d^2, \tag{4.2}$$

Substituting equation 4.1 into equation 4.2 yields the equation,

$$W(C_k) = \sum_{x_i \in C_k} (x_i - \mu_k)^2, \tag{4.3}$$

Where x_i is a data point belonging to the cluster C_k , μ_k is the mean value of the points assigned to the cluster C_k and d refers to Euclidean distance.

The total within-cluster variation can be modified from equation 4.3 and is defined as:

$$\sum_{k=1}^{k} W(C_k) = \sum_{k=1}^{k} \sum_{x_i \in C_k} (x_i - \mu_k)^2, \tag{4.4}$$

The total within-cluster sum of square measures the compactness of the clustering, and it should be as small as possible (Teja, 2015).

The starting assignments of cluster centres are random. A nstart value of 20 was used in R; this means that R will try 20 different random starting assignments and then select the one with the lowest within-cluster variation.

In step 5 of the process flow, the cluster centroid has to be recalculated, the cluster centre can be recomputed by,

$$N_i = \left(\frac{1}{c_i}\right) \sum_{j=1}^{c_i} x_i, \tag{4.5}$$

Where c_i represents the number of data points in cluster i

After re-computation, if no data point was reassigned then stop, otherwise repeat. Repeat these steps until the centroid no longer move.

4.3 Elbow plot

k -Means is relatively an efficient method. However, the number of clusters have to be specified in advance, and the final results are sensitive to initialisation and often terminates at a local optimum. A practical approach is to compare the outcomes of multiple runs with different *k* and choose the best one based on a predefined criterion (Kumbhar, Oza, & Kamat, 2012). There are several techniques developed over the years to identify the suitable number of clusters for a dataset, but the best number of clusters chosen by these methods may not be the right number of clusters for the business needs. In general, a large *k* probably decreases the error but increases the risk of overfitting. (Saed, 2010)

One method to validate the number of clusters is the elbow method. The idea of the elbow method is to run *k* -means clustering algorithm on the dataset for a range of values of *k* and for each value of *k* calculate the internal sum of squares in each cluster.

In this analysis, an elbow plot has been used to identify the optimal number of clusters and this value of k was slowly increased in the hope of finding interesting groups and valuable information.

The elbow plot (Figure 4.2) is a graphic representation of the average internal per cluster sum of squares distance by the number of clusters. The average internal sum of squares is the average distance between points inside of a cluster.

The elbow method computes the clustering algorithm for different values of *k* by using equation 4.4. For instance, by varying *k* from 1 to 10 clusters. For each *k*, calculate the total within-cluster sum of square(WSS). Plot the curve of WSS according to the number of clusters *k*. The location of a bend in the plot is generally considered as an indicator of the appropriate number of clusters. In Figure 4.2 the indicator lies at $k = 3$, which means clustering the dataset into three clusters is a suitable option. The goal is to choose a small value of *k* that has a low WSS, and the elbow usually represents the *k* value where diminishing returns are expected by increasing *k*. (Robert, 2017)

From the elbow plot, the suitable value of *k* for this analysis was determined to be 3. The next step is to assign 3 cluster centres randomly. After assigning the cluster centre, apply the

Euclidean distance function(Equation 4.1) and calculate the distance of each observation from the cluster centres. Assign the observations to the cluster centre with the lowest distance. Apply equation 4.5 to re-compute the cluster centres. If the cluster centre changes, recalculate distance and carry out the iterative process until the cluster centre do not change.

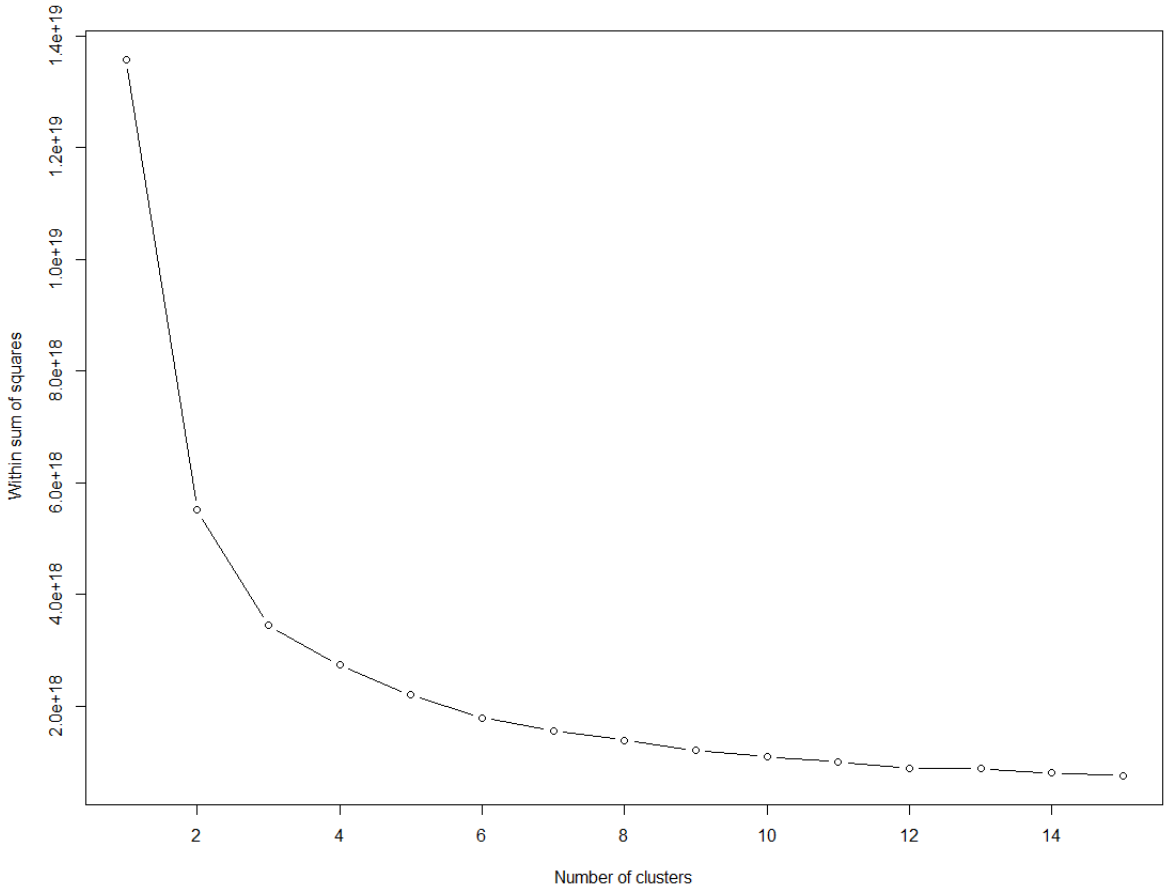


Figure 4. 2 Elbow Plot

4.4 R coding summary

The data was uploaded into RStudio and checked for any duplicate data. These duplicates were removed from the dataset. A visualisation of the missing values was created from which the decision to remove observations with missing gross and budget values was made due to the high amount of missing gross and budget values. A function called 'OutlierReplace' was created to apply the rules stated in section 3.1. A variable selection was carried out to eliminate variable not used in this analysis.

Scatterplots were used to identify outliers. The initial step was to use the boxplot method to eliminate outliers. This method proved inefficient for this particular analysis. Hence, only the extreme outliers were removed using the 'OutlierReplace' function created earlier. These outliers were stored together in a data frame for separate analysis.

A new data frame with only gross and budget was created to be used in the calculation of wss and creation of elbow plot. From the elbow plot, the value of *k* was identified as 3 and clustering was carried out by setting nstart value =20. The cluster values were then saved to the dataset, and the aggregate function was used to identify the mean Facebook likes in each cluster. These steps were carried out again for a *k* value of four.

After identifying the significant clusters (1 and 3) from a *k* value of four, the observations in these clusters were separated from the dataset. The top 46 movies were filtered out based on highest Facebook likes in each of these clusters. The two groups of 46 movies were bind together using the rbind function.

Finally, after determining that the eight movies eliminated because of extreme values are movies that are worth adding to the company's database, these eight movies were bind together with the 92 movies. This ended the selection of 100 movies. The data frame of 100movies was then saved as a CSV file to be handed over to the sales department.

(Refer to appendix "R CODE for clustering and final movie selection" for full R script along with the list of 100 movies obtained from the full analysis)

5. EVALUATION

Based on the elbow plot (Figure 4.2), the optimal number of *k* was identified to be 3. Figure 5.1 shows the plot of *k* -means clustering with 3 clusters. Cluster 1 is represented as red, cluster 2 as green and cluster 3 as blue.

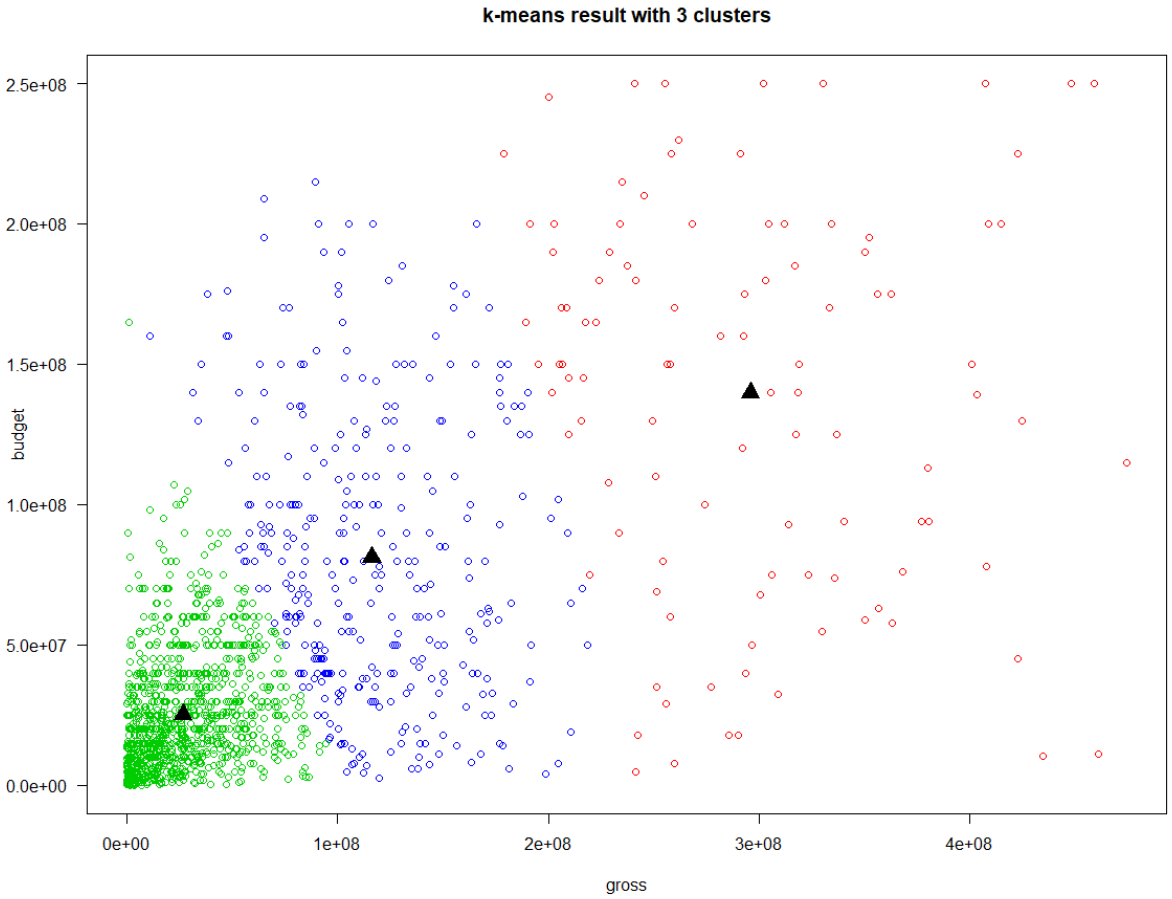


Figure 5. 1 k -means clustering k=3

Cluster	Cluster size	Mean Budget	Mean Gross	Mean Facebook likes
1	107	140m	295m	50 000
2	1106	25m	27m	15 000
3	350	81m	116m	32 000

Table 5. 1 Cluster results, k=3

Cluster 1 is generally made of high gross movies, cluster 2 is made of low gross and budget movies, and cluster 3 is mostly made of average gross movies. From analysing the cluster

centres, it can be identified that movies with a high budget generally obtained high gross and Facebook likes within the first week of release. The majority of the movies are low budget movies which made little profit in the first week and received low Facebook likes. A basic conclusion can be made from cluster 1 that people generally enjoy high budget movies.

The level of information that can be obtained from these clusters is limited. Hence, the data is re-clustered with a new *k* value of 4 (4 clusters) to get a better understanding of the data before selecting the movies.

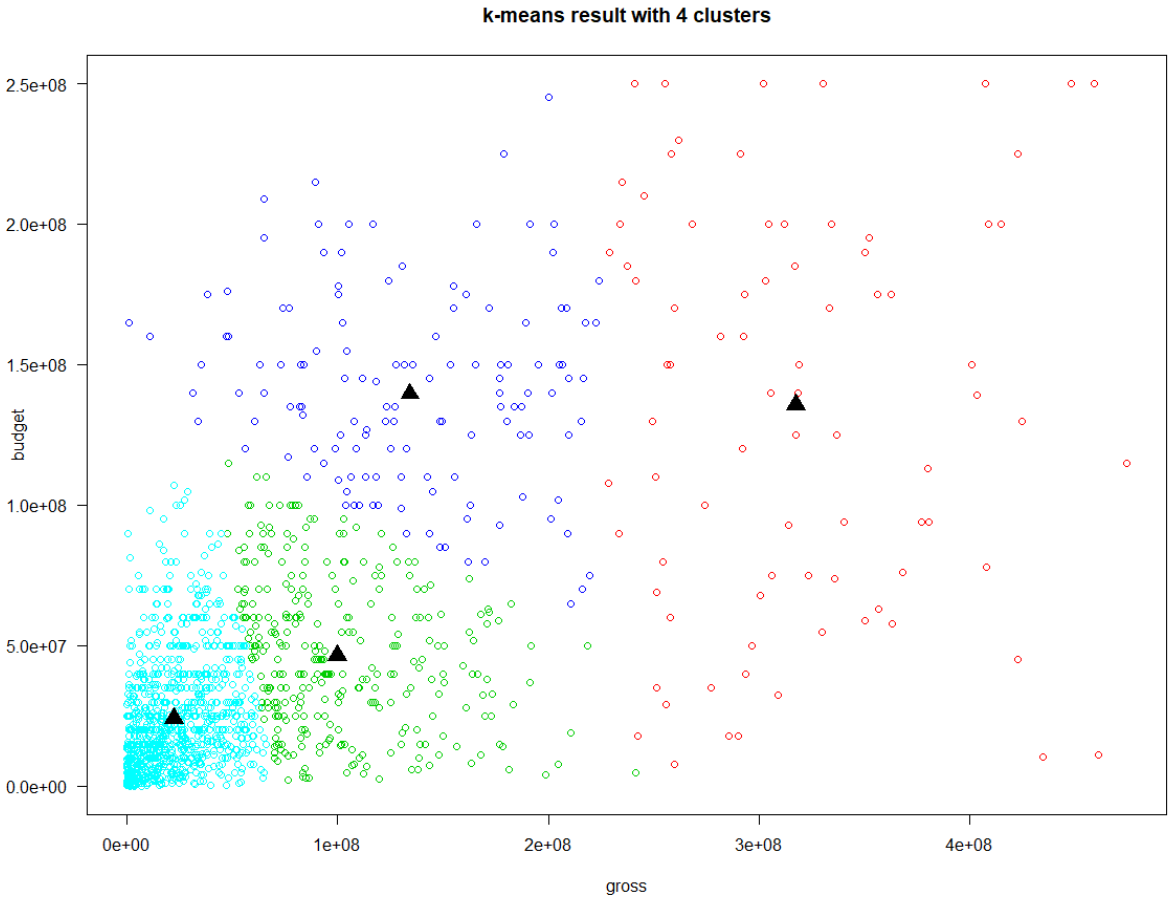


Figure 5. 2 k-means clustering, k=4

Cluster	Cluster size	Mean Budget	Mean Gross	Mean Facebook likes
1	86	136m	317m	53 000
2	327	47m	100m	26 000
3	144	140m	134m	39 000
4	1006	22m	24m	14 000

Table 5. 2 Cluster results k=4

Cluster 1 is represented in red, cluster 2 in green, cluster 3 in blue and cluster 4 in cyan. Comparing to the previous model, this model breaks down high budget movies into two groups, which allows us to do a more in-depth analysis. Cluster 1 is mostly made of high budget movies, which have made huge profits. Cluster 2 is generally made of mid-budget movies, which made a good profit. Cluster 3 is mainly made of high budget movies, which generally made a loss. Cluster 4 is made of low budget movies which barely made any profit.

Cluster 1 has 86 movies mostly made of high budget movies which made a high profit (gross) and has also achieved high Facebook likes. Selecting movies from this cluster may be a good option. Cluster 2 is made mostly of mid-budget movies that made a good profit but the mean number of Facebook likes for this cluster is low. Cluster 3 is mainly made of high budget movies that did not make much profit (low gross), but the number of Facebook like for this cluster was relatively high. Majority of the movies fall into cluster 4; this cluster is made of low budget movies, which barely managed to make any profit. This cluster also has the lowest Facebook likes, which makes this cluster of movies the least preferred movies among movie viewers. Hence, this cluster of movies can be eliminated from the list of movies to be added to the company's database. Similarly, cluster 2 was removed due to a low mean Facebook likes.

From the first analysis with k =3, it was understood that people generally like high budget movies. However, in this model with k =4, there are two groups of high budget movies, cluster 1 generally made a profit while most movies in cluster 3 barely managed to make any profit.

Cluster 3 is a concern in this analysis. This is due to the reason that even though these movies have a low gross, these group of movies still managed to achieve a reasonably high mean Facebook likes. This rose the question of how gross is linked to Facebook likes.

A high gross is achieved when many people go to the movies to watch the films, and high Facebook likes are achieved when people like the movie that they watched. So the question is how did low gross movies in cluster 3 receive high Facebook likes? The only logical explanation is that these movies could have had lousy advertising and bad trailers or released at bad timing. This, in turn, could have caused people not wanting to spend their money or time to go to the movies to watch the Film. So people could have streamed these movies online is low quality when these movies were released and found these movies were good despite the

lousy advertising and liked the movies on Facebook. Considering this concept, cluster 3 could hold a possible selection of movies that could be added to the company's database.

If the logic mentioned above is true, then these high budget movies with low gross would be in higher demand than high budget movies with a high gross profit. People would be searching to watch these movies online in high quality. Therefore offering these type of movies on top of other favourite movies would attract more people to subscribe to Beta company's services.

Another consideration that should be taken into account when selecting movies is to include high budget movies with high gross, and high Facebook likes. These movies are potentially the movies that people liked, and the chances of them wanting to watch them again are high. So, in conclusion, it would be ideal to add movies from both clusters.

In the data preparation stage, eight movies were removed due to extreme values (outliers). After analysing these movies, these movies are movies that were well accepted by people and had earned a huge profit as well as a high amount of Facebook likes. Hence, these movies were added to the selection. The remaining 92 movies were chosen from cluster one and cluster three. Forty-six movies with the highest Facebook likes were chosen from each cluster. This ended the selection of the 100 movies.

6. BENEFITS AND COMMERCIAL RISK

Clustering movies by budget and gross have enabled easy identification of specific groups of movies that people would like to stream online. Selection of movies by this method could increase the popularity of the company as the most searched movies will be shown on the company's website.

This method also enables the quick elimination of low budget movies and mid-budget movies. After this elimination, analysis can be carried out on high budget movies to obtain suitable movies efficiently.

The main commercial risk is that this method is not able to identify movies older than 2005. Another downfall is that this method might not be suitable for movies released before 2008. Facebook gained popularity in 2005, but it because more widely used around 2008, hence movies released from 2005-2008 may not have obtained as many Facebook likes as movies released after 2008.

7. CONCLUSION

In this report, a *k* -means clustering approach is carried out to identify a set of suitable movies to be added to a company's database. *k* -means clustering based on Euclidean distance was used to carry out this process, and the elbow plot was used to determine the initial *k* value. Clustering using the initial *k* value classified the dataset into three groups from which, it was understood that people generally enjoyed movies with high budgets. The *k* value was then increased to understand the dataset further and find meaningful patterns. A *k* value of 4 further classified high budget movies into two groups, high budget movies which made a profit and high budget movies which barely made any profit. After applying a logical conclusion that high budget movies with low gross could be due to lousy advertising or bad release date, it was concluded that movies in this cluster could be potential movies that should be added to the database. The final selection of movies was made up of 46 movies from high budget movies with high gross, 46 movies from high budget movies with low gross and eight movies which were removed as outliers in the cluster analysis. This method has proven to be an efficient method in narrowing down a list of 100 movies for the company.

REFERENCES

Ankit, A. (2015, February 24). All about clustering (Hierarchical and partitive).

Kumbhar, V., Oza, K., & Kamat, R. (2012). *Web Mining, A synergic Approach Resorting to Classifications and Clustering.* Denmark: River Publishers.

Lech, P., Yiyu, Y., Piotr, A., & Davide, C. (2017). *Rough Sets.* Olsztyn, Poland: Springer.

Rajeev, p., Vaishali, & Rupa, M. G. (2011). Impact of Outlier Removal and Normalization Approach in Modified K-Means Clustering Algorithm. *International Journal of Computer Science Issues 8*.

Robert, G. (2017, December 26). *Using the elbow method to determine the optimal number of clusters for K-means clustering*. Retrieved from Robert Gove's Blocks: https://bl.ocks.org/rpgove/0060ff3b656618e9136b

Saed, S. (2010). *An introduction to data science*. Retrieved from K-means clustering: https://www.saedsayad.com/clustering_kmeans.htm

Siddhant, G., & Utkarsha, M. (2018). Predicting Agricultural output by applying Machine Learning. *IJSTE- International Journal of Science Technology & Engineering*, Volume 4, Issue 11.

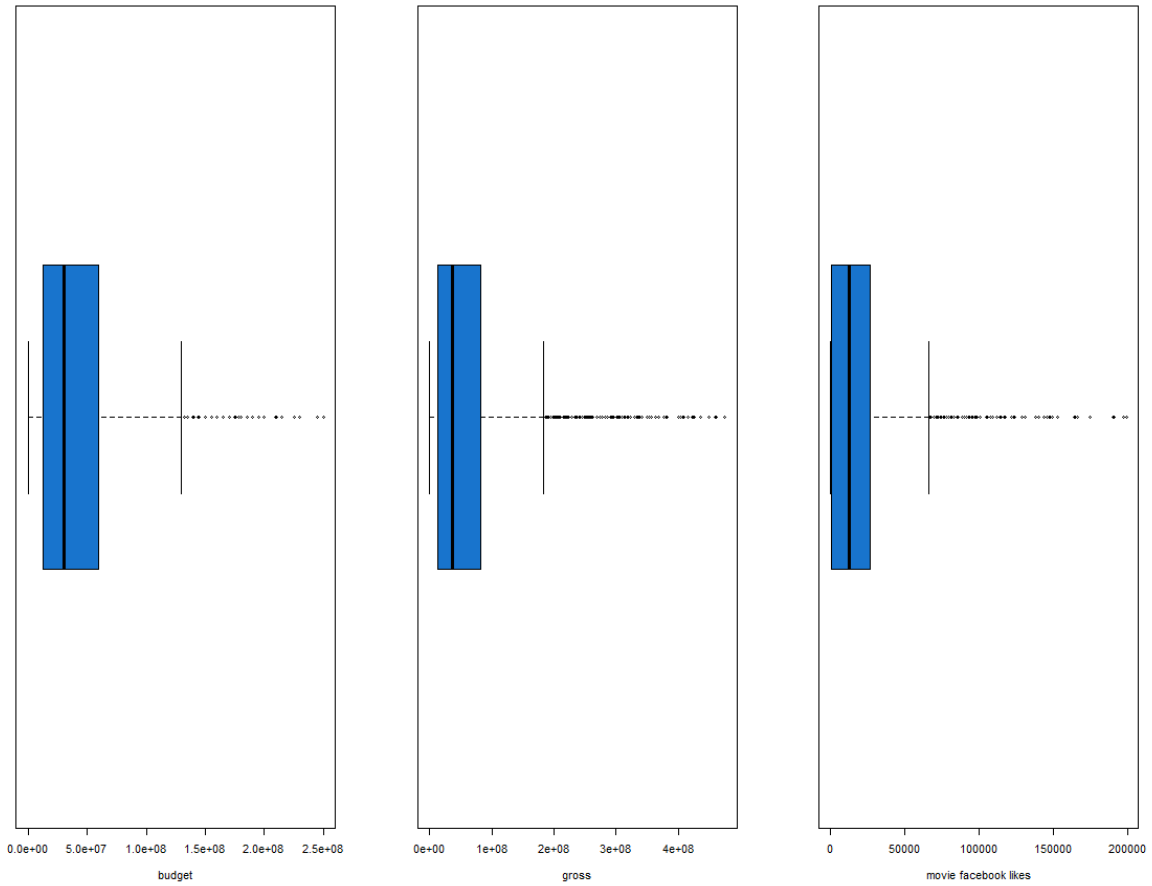
Stephanie. (2015, Novemeber 8). *Normalized Data/Normalization*. Retrieved from Statistics How To: https://www.statisticshowto.datasciencecentral.com/normalized/

Teja, K. (2015, December 28). Advanced modeling. *K Means Clustering in R*.

Vaishali, P., & Mehta, G. R. (2011). Impact of Outlier Removal and Normalization Approach in modified K-Means Clustering Algorithm. *IJCSI International Journal of Computer Science Issues*, vol 8, issue 5.

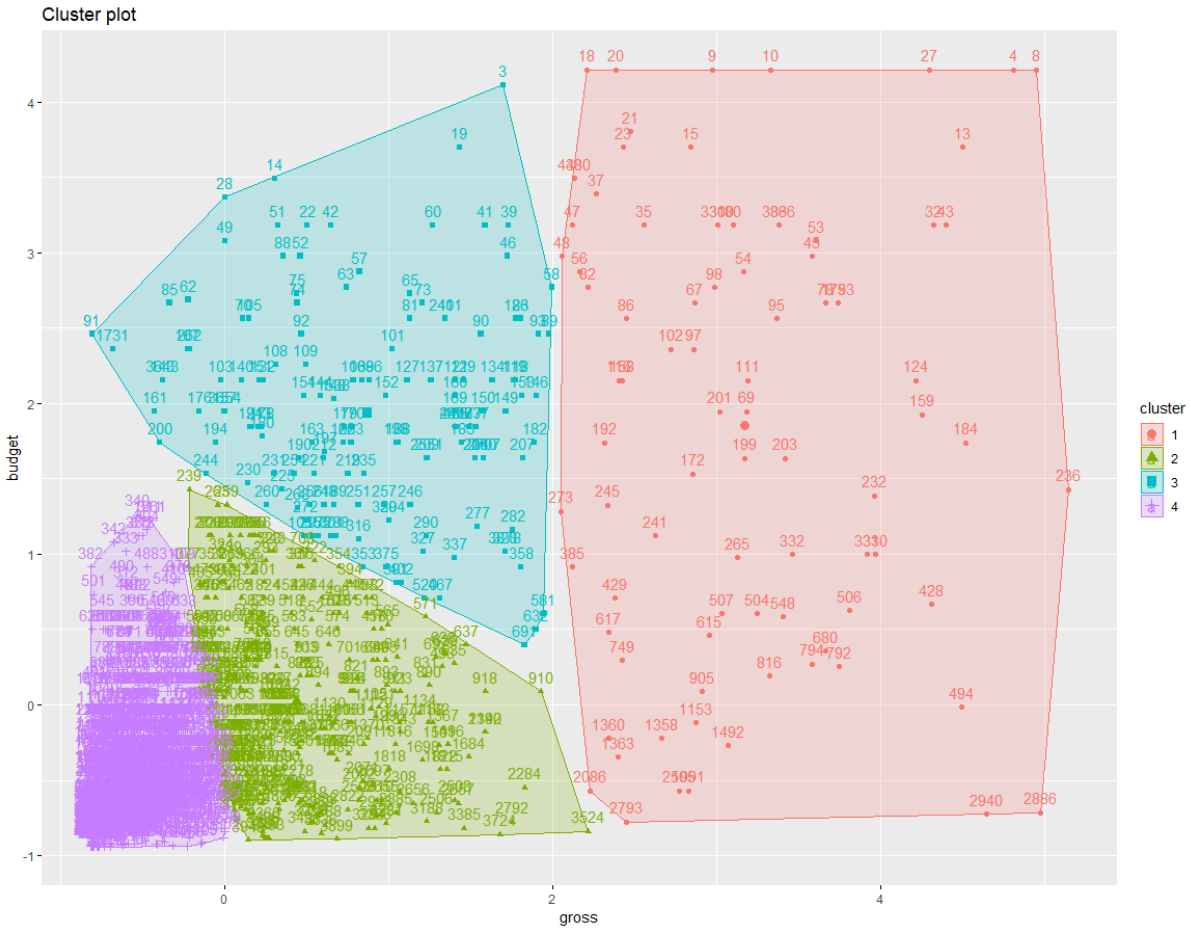
APPENDIX

BOXPLOT



The box plot identified a large number of movies as outliers; points after the right most vertical line are considered as outliers. Hence, this method was not used to remove outlier. Instead, a manual outlier approach was used.

Normalised clustering



The normalised clusters formed similar results to the non-normalised clusters.

R CODE for clustering and final movie selection

```
library(VIM)
library(data.table)
library(clustertend)
library("NbClust")
library(cluster)    # clustering algorithms
library(factoextra)
library(tidyverse)
library(eeptools)

mydata <- read.csv("movie_metadata.csv", header = TRUE)
View(mydata)
# Duplicate rows
sum(duplicated(mydata))
# Delete duplicate rows
mydata <- mydata[!duplicated(mydata),]

# plot of missing values
colSums(sapply(mydata,is.na))
aggr(mydata)

# data exploration
plot(mydata$gross,ylab="gross",col = "dodgerblue3")
plot(mydata$budget,ylab="budget",col = "dodgerblue3")
plot(mydata$movie_facebook_likes,ylab="Facebook likes",col = "dodgerblue3")

summary(mydata)
#filtering out only the movies from the data (year 2005 onwards)

outlierReplace = function(dataframe, cols, rows, newValue = NA)
{
  if (any(rows))
{
    set(dataframe, rows, cols, newValue)
  }
}
outlierReplace(mydata, "title_year", which(mydata$title_year <2005), NA)
outlierReplace(mydata, "duration", which(mydata$duration < 70), NA)
outlierReplace(mydata, "duration", which(mydata$duration > 200), NA)
outlierReplace(mydata, "gross", which(mydata$gross < 40000), NA)
outlierReplace(mydata, "budget", which(mydata$budget < 40000), NA)
mydata=filter(mydata, language=="English")
outlierReplace(mydata, "movie_facebook_likes", which(mydata$movie_facebook_likes <500), NA)

#selecting variables to use

mydata <- mydata[c(9,12,23,28)]
mydata=na.omit(mydata)
```

```
View(mydata)
summary(mydata)

par(mfrow=c(1,3))
# Explore data
plot(mydata$gross,ylab="gross",col = "dodgerblue3")
plot(mydata$budget,ylab="budget",col = "dodgerblue3")
plot(mydata$movie_facebook_likes,ylab="Facebook likes",col = "dodgerblue3")

View(mydata)


b1<- subset(mydata,budget >2.55e+08)
g1<- subset(mydata,gross >5.0e+08)
F1<- subset(mydata,movie_facebook_likes>2.0e+05)
View(F1)
special <- rbind(b1,g1,F1)
View(special)
# removing movies with extreme values for special analysis
outlierReplace(mydata,"budget", which(mydata$budget> 2.55e+08),NA)
outlierReplace(mydata, "gross", which(mydata$gross > 5.0e+08),NA)
outlierReplace(mydata,"movie_facebook_likes", which(mydata$movie_facebook_likes> 2.0e+05),NA)

mydata=na.omit(mydata)
mydata


#outlier detection
#boxplot(mydata$budget,horizontal = T,col ="dodgerblue3", xlab='budget')
#boxplot(mydata$gross,horizontal = T,col ="dodgerblue3", xlab='gross')
#boxplot(mydata$movie_facebook_likes,horizontal = T,col ="dodgerblue3", xlab='movie facebook likes')
#outliner elimination
#summary(mydata$budget)
#bench <- 60000000 + 1.5*IQR(mydata$budget)
#bench
#bench2<- 13000000 - 1.5*IQR(mydata$budget)
#bench2
#outlierReplace(mydata, "budget", which(mydata$budget > 2.1e+08), NA)
#summary(mydata$gross)
#bench <- 67685175 + 1.5*IQR(mydata$gross)
#bench
#bench2<- 12134420 - 1.5*IQR(mydata$gross)
#bench2
#outlierReplace(mydata, "gross", which(mydata$gross > 4.4e+08), NA)
#mydata=na.omit(mydata)
#mydata
#boxplot(mydata$budget,horizontal = T,col ="dodgerblue3", xlab='budget')
#boxplot(mydata$gross,horizontal = T,col ="dodgerblue3", xlab='gross')

# scatterplot after manual removal of extreme values
plot(mydata$gross, ylab="gross",col = "dodgerblue3")
plot(mydata$budget, ylab="budget",col ="dodgerblue3")
plot(mydata$movie_facebook_likes, ylab="Movie facebook likes",col="dodgerblue3")
```

```
# subsetting data with only gross and budget to calculate wss
newdata <- mydata[c(1,3)]
newdata
#############################################################################################
#clustering tendency & number of clusters, elbow plot
wss <- (nrow(newdata)-1)*sum(apply(newdata,2,var))

for (i in 2:15) wss[i]<-sum(kmeans(newdata,centers=i)$withinss)

plot(1:15, wss, type="b", xlab= "Number of clusters" , ylab=" within sum of squares")
#############################################################################################
# clustering the dataset based on gross and budget, with optimal k value found from wss

set.seed(20)
#clusters <- kmeans(mydata[c(1,3)],3, nstart=20)
clusters <- kmeans(mydata[c(1,3)],4, nstart=20)
#save cluster number in the dataset
mydata$cluster <- as.factor(clusters$cluster)
#switch cluster to first column
mydata<- mydata[,c(ncol(mydata),1:(ncol(mydata)-1))]


str(clusters)
clusters$center
clusters$size
#check within sum of squares value (higher the better)
clusters
#plot(newdata, col =(clusters$cluster +9) , main="k-means result with 3 clusters", pch=1, cex=1, las=1)
plot(newdata, col =(clusters$cluster +9) , main="k-means result with 4 clusters", pch=1, cex=1, las=1)
points(clusters$centers, col = "black", pch = 17, cex = 2)
aggregate(data=mydata,movie_facebook_likes~cluster,mean)

# clustering using normalization method
fviz_cluster(clusters,data=mydata[c(2,4)])

aggregate(mydata[c(2,4,5)],by=list(clusters$cluster),FUN = mean)


# extracting only movies from cluster 1 and 3
cluster1x <- subset(mydata, cluster == 1)
cluster3x<- subset(mydata, cluster ==3)

#arranging movies from highest to lowest value
cluster1 <- cluster1x[order(-cluster1x$movie_facebook_likes) , ]
cluster3 <- cluster3x[order(-cluster3x$movie_facebook_likes) , ]

# Selecting top 46 movies in each cluster with the highest facebook likes
Fcluster1<-cluster1[1:46, 2:5]
FCluster3<-cluster3[1:46, 2:5]


# combining the top 46 movies from the two clusters and 8 movies which were removed as outliers
total <- rbind(Fcluster1, FCluster3)
total2<- rbind(total, special)
# save file as excel
write.csv(total2, file = "Selected 100 movies.csv")
```

The result of the code: Selected 100 movies.CSV

gross	movie_title	budget	movie_facebook_likes
3.3E+08	Batman v Superman: Dawn of Justice	2.50E+08	197000
2.93E+08	Inception	1.60E+08	175000
3.03E+08	The Hobbit: An Unexpected Journey	1.80E+08	166000
4.48E+08	The Dark Knight Rises	2.50E+08	164000
2.28E+08	The Martian	1.08E+08	153000
2.74E+08	Gravity	1.00E+08	147000
4.08E+08	The Hunger Games	7.80E+07	140000
4.59E+08	Avengers: Age of Ultron	2.50E+08	118000
2.91E+08	Man of Steel	2.25E+08	118000
3.56E+08	Inside Out	1.75E+08	118000
3.63E+08	Deadpool	5.80E+07	117000
3.5E+08	American Sniper	58800000	112000
3.33E+08	Guardians of the Galaxy	1.70E+08	96000
4.09E+08	Iron Man 3	2.00E+08	95000
3.5E+08	Furious 7	1.90E+08	94000
2.29E+08	Star Trek Into Darkness	1.90E+08	92000
2.41E+08	The Dark Knight	1.80E+08	89000
2.58E+08	The Hobbit: The Desolation of Smaug	2.25E+08	83000
2.34E+08	X-Men: Days of Future Past	2.00E+08	82000
4.25E+08	The Hunger Games: Catching Fire	1.30E+08	82000
3.04E+08	Skyfall	2.00E+08	80000
3.04E+08	Skyfall 2	2.00E+08	80000
4.07E+08	Captain America: Civil War	2.50E+08	72000
3.36E+08	Minions	7.40E+07	70000
2.55E+08	The Hobbit: The Battle of the Five Armies	2.50E+08	65000
3.63E+08	The Jungle Book	1.75E+08	65000
2.92E+08	The Twilight Saga: Breaking Dawn - Part 2	1.20E+08	65000
3.63E+08	The Jungle Book 2	1.75E+08	65000
2.58E+08	The Lego Movie	6.00E+07	64000
2.35E+08	Oz the Great and Powerful	2.15E+08	60000
2.35E+08	Oz the Great and Powerful 2	2.15E+08	60000
3.3E+08	Forrest Gump	5.50E+07	59000
2.41E+08	Pirates of the Caribbean: On Stranger Tides	2.50E+08	58000
4.01E+08	Frozen	1.50E+08	58000
2.62E+08	The Amazing Spider-Man	2.30E+08	56000
2.45E+08	Transformers: Age of Extinction	2.10E+08	56000
2.54E+08	The Hangover Part II	8.00E+07	56000
3.68E+08	Despicable Me 2	7.60E+07	56000
2.6E+08	Captain America: The Winter Soldier	1.70E+08	55000
3.37E+08	The Hunger Games: Mockingjay - Part 1	1.25E+08	52000
3.52E+08	Transformers: Dark of the Moon	1.95E+08	46000
2.68E+08	Monsters University	2.00E+08	44000
2.37E+08	Brave	1.85E+08	39000

2.82E+08	The Hunger Games: Mockingjay - Part 2	1.60E+08	38000
2.56E+08	The Blind Side	2.90E+07	38000
3.24E+08	The Secret Life of Pets	7.50E+07	36000
1.63E+08	Django Unchained	1.00E+08	199000
1.54E+08	Mad Max: Fury Road	1.50E+08	191000
1.84E+08	The Revenant	1.35E+08	190000
1.17E+08	The Wolf of Wall Street	1.00E+08	138000
2.02E+08	World War Z	1.90E+08	129000
1.25E+08	Life of Pi	1.20E+08	122000
1.45E+08	The Great Gatsby	1.05E+08	115000
1.45E+08	Maleficent	1.05E+08	115000
1.26E+08	Prometheus	1.30E+08	97000
46978995	Warcraft	1.60E+08	89000
2E+08	Spectre	2.45E+08	85000
1.02E+08	Pacific Rim	1.90E+08	83000
89732035	Terminator Genisys	1.55E+08	82000
1.61E+08	Suicide Squad	1.75E+08	80000
1E+08	Edge of Tomorrow	1.78E+08	77000
1.01E+08	Noah	1.25E+08	71000
89021735	Oblivion	1.20E+08	71000
1.06E+08	300: Rise of an Empire	1.10E+08	71000
1.33E+08	The Wolverine	1.20E+08	68000
1.02E+08	Independence Day: Resurgence	1.65E+08	67000
2.06E+08	Thor: The Dark World	1.70E+08	63000
1.81E+08	Thor	1.50E+08	63000
1.18E+08	Ghostbusters	1.44E+08	62000
1.91E+08	Teenage Mutant Ninja Turtles	1.25E+08	62000
1.18E+08	Ghostbusters	1.44E+08	62000
1.91E+08	Teenage Mutant Ninja Turtles 2	1.25E+08	62000
1.8E+08	Ant-Man	1.30E+08	61000
93050117	Elysium	1.15E+08	61000
2.01E+08	Cinderella	9.50E+07	56000
2.01E+08	Alice in Wonderland	9.50E+07	56000
1.55E+08	X-Men: Apocalypse	1.78E+08	54000
1.46E+08	X-Men: First Class	1.60E+08	54000
2.1E+08	Fast Five	1.25E+08	54000
1.55E+08	Snow White and the Huntsman	1.70E+08	53000
1.55E+08	San Andreas	1.10E+08	52000
65007045	Exodus: Gods and Kings	1.40E+08	51000
65007045	Gods and Kings	1.40E+08	51000
1.51E+08	Divergent	8.50E+07	49000
89289910	The Lone Ranger	2.15E+08	48000
1.95E+08	Mission: Impossible - Rogue Nation	1.50E+08	47000
1.77E+08	Rise of the Planet of the Apes	9.30E+07	47000
1.77E+08	How to Train Your Dragon 2	1.45E+08	46000
1.77E+08	Captain America: The First Avenger	1.40E+08	46000

2.09E+08	Dawn of the Planet of the Apes	1.70E+08	45000
65173160	Battleship	2.09E+08	44000
47375327	Jupiter Ascending	1.76E+08	44000
73058679	John Carter	263700000	24000
2.01E+08	Tangled	2.60E+08	29000
7.61E+08	Avatar	2.37E+08	33000
6.23E+08	The Avengers	2.20E+08	123000
6.59E+08	Titanic	2.00E+08	26000
6.52E+08	Jurassic World	1.50E+08	150000
5.33E+08	The Dark Knight	1.85E+08	37000
1.88E+08	Interstellar	1.65E+08	349000